

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
ОШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Автоматизированных систем и цифровых технологий»

«Согласовано»
председателем методсовета
факультета МИТ
к.п.н., д.: Д. Зулпукарова

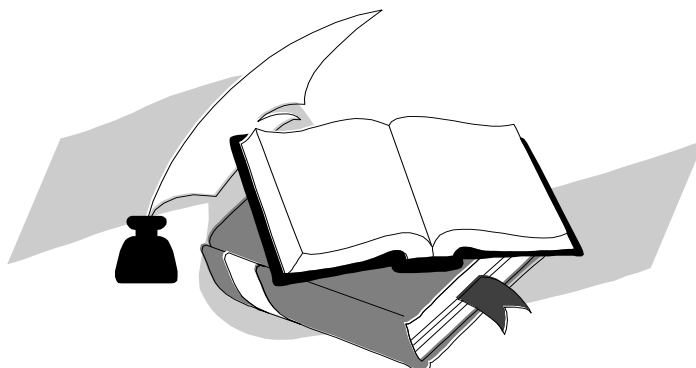
«__» _____ 2020 г.

«Утверждено»
на заседании кафедры АСЦТ
прот. №1 от «28» августа 2020 г.
Зав. каф. АСЦТ: Молдоярв У.

УЧЕБНО- МЕТОДИЧЕСКИЙ КОМПЛЕКС

по дисциплине: «База данных»
для магистрантов очного отделения, обучающихся по специальности:

ИВТ



УМК составлен на основе Государственного образовательного стандарта по направлению 710100 «ИВТ», академическая степень - магистр «ИВТ» для магистрантов, обучающихся в очном отделении

Составитель, доцент каф. АСЦТ: _____ Беделова Н.С.

2020-2021 учебный год

Сведения о преподавателя

Беделова Нургуль Салибаевна – доцент каф. АСЦТ,
ФМИТ ОшГУ, общий стаж работы – 22 года,

образование – высшее, закончила

физико-математический факультет ОшГУ, 2000 г.

Рабочий телефон: 03222-2-11-85,

Рабочее место: 723500. Главный корпус ОшГУ,
ул. Ленина 331, каб. 224.

Мобильный телефон: 0772-43-73-31

E-mail: nbedelova@oshsu.kg.

I. Аннотация к дисциплине

Целью освоения дисциплины «База данных» является формирование у магистрантов профессиональных навыков необходимых для правильного выбора и использования инструментальных средств создания БД и информационных систем, определения подходящей модели данных, организации эффективной структуры хранения данных, организации запросов к хранимым данным и других вопросов от которых зависит эффективность разрабатываемых систем.

Задачи освоения дисциплины состоят в формировании профессиональных компетенций, позволяющих самостоятельно проводить обследование организаций, выявлять и описывать прикладные процессы и информационные потребности пользователей, а также осуществлять ведение баз данных средствами современных СУБД, самостоятельно решать задачи обработки текстовой и нетекстовой информации в БД и поддержку информационного обеспечения решения прикладных задач.

Содержание

Сведения о преподавателя.....	2
I. АННОТАЦИЯ.....	2
Внутренняя и внешняя рецензии на рабочую программу.....	4
Выписка из матрицы компетенций ООП специальности по направлению 710100 «Информатика и вычислительная техника».....	6
II. РАБОЧАЯ ПРОГРАММА.....	10
2.1. Цели освоения дисциплины.....	11
2.2. Результаты обучения и компетенции формируемые в процессе изучения дисциплины «База данных».....	11
2.3. Место дисциплины в структуре ООП.....	11
2.4. Карта компетенций дисциплины.....	12
2.5. Технологическая карта дисциплины.....	12
2.6. Карта накопливаемости баллов по дисциплине.....	12
2.7. Программа дисциплины.....	14
2.8. Тематический план распределения часов по видам занятий.....	17
2.9. Календарно-тематический план по видам занятий.....	18
2.9.1. Лекционные занятия.....	18
2.9.2. Лабораторные занятия.....	18
2.9.3.Задания на самостоятельной работы магистрантов.....	18
2.10. Задании итогового контроля.....	24
2.11. Учебно-методическое обеспечение дисциплины.....	25
2.11.1.Основные литературы.....	25
2.11.2.Дополнительные литературы.....	25
2.11.3.Электронные источники.....	25
2.12. Критерии оценки знаний магистрантов.....	26
III. СИЛЛАБУС.....	27
3.1. Цели освоения дисциплины.....	28
3.2. Результаты освоения дисциплины.....	28
3.3. Политика курса.....	29
3.4. Пререквизиты курса.....	29
3.5. Постреквизиты курса.....	29
3.6. Образовательные технологии.....	30
3.7. Технологическая карта.....	30
3.8. Содержание лекционных и лабораторных занятий.....	32
3.9. Задания на самостоятельной работы магистрантов.....	34
3.10. Учебно-методическое обеспечение дисциплины.....	34
3.11. Примерный перечень вопросов для подготовки к экзамену.....	35
3.12. Критерии оценки знаний магистрантов.....	35
3.13. Политика выставления баллов.....	35
IV. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ.....	38
Критерии оценки на экзамене.....	41
V. ЛЕКЦИИ.....	43
VI. ТЕСТЫ.....	73

Внутренняя и внешняя рецензии на рабочую программу

РЕЦЕНЗИЯ

на рабочую программу по дисциплине «База данных» для магистрантов очного отделения, обучающихся по направлению 710100 «Информатика и вычислительная техника»

Рабочая программа дисциплины «База данных» разработана в соответствии с требованиями ГОС ВПО КР, на основе бюллетеня №19 ОшГУ и ООП специальности по направлению 710100 «Информатика и вычислительная техника».

Рецензируемая программа предназначена для методического обеспечения учебной работы магистрантов очной формы обучения. Содержание представленной на рецензию рабочей учебной программы включает в себя разделы: цели и задачи освоения дисциплины; место дисциплины в структуре ООП; требования к результатам освоения дисциплины; объем дисциплины и виды учебной работы; содержание дисциплины; библиотечно-информационные ресурсы; оценочные средства; материально-техническое обеспечение.

Содержание курса представлено шести разделами, которые в полной мере отражают необходимый объем изучаемого материала. По каждому разделу составлен перечень вопросов, рассмотрение которых позволит сформировать знания, умения и навыки, отвечающие требованиям ГОС ВПО КР.

Информация о видах и объеме учебной работы содержит лабораторных работ и тематику лекционных занятий, призванных помочь магистранту получить теоретические знания и практические навыки создания, внедрения, функционирования, применения создания база данных, обеспечивающих поддержку работы специалиста.

Программа соответствует всем требованиям бюллетеня №19 ОшГУ и ООП специальности по направлению 710100 «Информатика и вычислительная техника».

Рецензент, к.т.н., доцент по спец.
«Информатика и управление»,
проректор по учебной работе ОшГУ:



Матисаков Ж.К.

РЕЦЕНЗИЯ

на рабочую программу по дисциплине «База данных» для магистрантов очного отделения, обучающихся по направлению 710100 «Информатика и вычислительная техника»

База данных является важнейшим средством при решении многих задач. Именно в связи с этим фактором обучение созданию базы данных так необходимо для подготовки инженеров-программистов.

Рабочая программа дисциплины «База данных» разработана в соответствии с требованиями ГОС ВПО КР, на основе бюллетеня №19 ОшГУ и ООП специальности по направлению 710100 «Информатика и вычислительная техника».

Рабочая учебная программа содержит:

- ✓ цели и задачи освоения дисциплины;
- ✓ результаты обучения и компетенции формируемые в процессе изучения дисциплины;
- ✓ место дисциплины в структуре ООП;
- ✓ карта компетенций дисциплины в разрезе тем;
- ✓ технологическая карта дисциплины;
- ✓ карта накопления баллов по дисциплине;
- ✓ тематический план распределения часов по видам занятий;
- ✓ программа дисциплины;
- ✓ требования к результатам освоения дисциплины;
- ✓ объем дисциплины и виды учебной работы;
- ✓ содержание дисциплины;
- ✓ библиотечно-информационные ресурсы;
- ✓ оценочные средства;
- ✓ материально-техническое обеспечение;
- ✓ учебно-методическое обеспечение дисциплины;
- ✓ политика выставления баллов.

На основании вышеизложенного считаю, что рецензируемая программа полностью соответствует требованиям, предъявляемым к рабочей программе по дисциплине «База данных» и может быть рекомендована для обучения магистрантов по направлению 710100 «Информатика и вычислительная техника».

Рецензент, к.ф.-м.н., доцент и
зав. каф. ИСП:



Аркабаев Н.К.

Выписка из матрицы компетенций ООП специальности по направлению 710100 «Информатика и вычислительная техника»

710100 – «Информатика жана эсептөө техникасы» багыты боюнча магистрлерди даярдоодогу билим берүү программасынын максаттары (М):

М1. Магистранттердин социалдык-инсандык сапаттарын, жалпы маданияттуулук деңгээлин, ой-жүгүртүүсүн компетенттик мамиле аркылуу калыптандыруу жана жогорулатуу; бүтүрүүчүлөрдүн мамлекеттик жана расмий тилдерде маалыматка ээ болуусун жана аларды кеңири чөйрөдө колдоно алуу жөндөмдүүлүктөрүн арттыруу; мамлекеттин ар тараптан өнүгүшү үчүн интеллектуалдык потенциалын, билимин жана билгичтиктерин жүзөгө ашырууга даяр экендигине тарбиялоо, өлкөгө болгон патриоттук сезимдерин жогорулатуу.

М2. Сапаттын кепилдигин камсыздоо менен экономикалык, математикалык жана компьютердик техника, тармактар жана технологиялар багытындагы заманбап илимий билимдердин негиздеринде бүтүрүүчүгө тандап алган иш-чөйрөсүндө ийгиликтүү иштөөгө мүмкүндүк берүүчү жогорку кесиптик жактан адистешкен, тереңдетилген, атайын кесиптик билимдерди өздөштүрүүсүн камсыздоо жана аларды практика жүзүндө колдоно алууга даярдоо.

М3. Ишмердүүлүк чөйрөдөгү маселелерди чечүүнүн, эсептөөчү машиналарды, комплекстерди, системаларды, тармактарды иштетүүнүн моделдерин жана алгоритмдерин түзүү, аларды долбоорлоонун ыкмаларын жана методдорун колдоно алуу, эсептөө техникаларын жана тармактарды системалык, прикладдык жана программалык камсыздоо, администрлөө, инструменталдык тейлөө технологияларын өздөштүрүү менен иштете алуу жөндөмдүүлүктөрдү бүтүрүүчүлөрдө калыптандыруу.

М4. Илимдин жана техниканын жаңылыктарын жана жетишкендиктерин колдонуу менен ишкана-мекемедеги эсептөө техникалары жана тармактар менен иштөөнү автоматташтырууга, интранет, интернет тармактарын эффективдүү колдонуу аркылуу берилгендер базасын түзүүгө жана колдонууга, электрондук продукция чыгарууга жана киреше алууга, команда менен иштөөгө, өндүрүштөгү ар түрдүү татаал кырдаалдардан жана конфликттик абалдардан чыгымды азайтуу менен мыйзам чегинде чыга алууга жөндөмдүү болгон бүтүрүүчүлөрдү даярдоо. Илимдин жана техниканын жаңылыктарын пайдалануу менен региондорду өнүктүрүүгө жана санариптештирүүгө салым кошууга даярдоо.

М5. Эмгек базарынын керектөөлөрүн эсепке алууда, ресурстарды рационалдуу жана үнөмдүү пайдаланууда ата мекендик жана чет элдик тажрыйбалардын илимий-техникалык маалыматтарын, анализдөө менен колдоно билүүдө, стратегиялык пландаштырууда информациялык-коммуникациялык технологияларды колдонуу аркылуу чет элдик адистер менен мамилелешип иштөөгө жөндөмдүү болгон бүтүрүүчүлөрдү чыгаруу.

710100 – «Информатика жана эсептөө техникасы» бакалаврды даярдоо багытынын НББПсында төмөнкүдөй күтүлүүчү натыйжалар (КН) пландаштырылат:

№	Күтүлүүчү натыйжалар	Компетенциялар
1	Долбоордук-конструктордук ишмердүүлүк: техникалык каражаттардын мүмкүнчүлүктөрүн, үлгүлөрүн жана колдонуучулардын талаптарын анализдөөнүн негизинде, кесиптик ишмердүүлүктүн объекттеринин өзүнчө компоненттерине коюлуучу талаптарды жана өзгөчөлүктөрдү иштеп чыгуу; аппараттык комплекстердин компоненттерин долборлоо; аппараттык комплекстерди эффективдүү ишке ашыруу үчүн программалоо каражаттарын жана эсептөө	КК-1, КК-2, КК-3, КК-4, КК-5, КК-8

	техника каражаттарын колдонууга жөндөмдүү	
2	Технологиялык ишмердүүлүк: эсептөөчү системаларды, автоматташтырылган системалардын компоненттерин, программаларды, программалык комплекстерди алдыңкы технологияларды пайдалануу менен керектүү сапатта жана мөөнөттө түзүү; аппараттык-программалык комплекстерди тестирлөө жана оңдоо; программаларды жана аларды текшерүүнүн ыкмаларын иштеп чыгуу, кесиптик ишмердүүлүктүн объекттерин текшерүүнү өткөрүү; кесиптик ишмердүүлүктүн объектерин сыноочу программаларды жана сыноо усулдарын талдап иштеп чыгуу, сыноолорду жүргүзүүгө жөндөмдүү	ЖИК-2, КК-11, КК-12
3	Өндүрүштүк ишмердүүлүк: жаңы өндүрүмдү чыгарууда өндүрүштүк технологиялык процесстерди табууга жана өздөштүрүүгө; кесиптик ишмердүүлүктүн объекттерин иштеп чыгуунун технологияларын өздөштүрүү, өндүрүштөгү ар түрдүү татаал кырдаалдардан жана конфликттик ситуациялардан чыгымды азайтуу менен мыйзам чегинде чыга алууга; программалык өндүрүмдөрдүн эсептөөчү жана автоматташтырылган системдердин ишин көзөмөлдөп коштоого; кесипкөй ишмердиктин объекттерин эксплуатациялык мүнөздөмөлөрүн өлчөөнүн усулдарын жана каражаттарын тандоого; эсептөөчү жана автоматташтырылган системдердин системдик, инструменталдык жана колдонмо программалык жабдууларын орнотууга, калыптоого(настройка) жана тейлөөгө үйрөтүүгө жөндөмдүү	ЖОЖК-1
4	Уюштуруу-башкаруу ишмердүүлүгү: кесиптик ишмердүүлүктүн объекттерин иштеп чыгуу жараянынын этаптарын талап кылынган сапатта жана мөөнөттө уюштуруу; кесиптик ишмердүүлүктүн объекттерин иштеп чыгуу жараянын баалоо, көзөмөлдөө жана башкаруу; кесиптик ишмердүүлүктүн объекттерин иштеп чыгуу жараянын уюштурууда технологияны, аспаптык каражаттарды жана ЭТ каражаттарды тандоого жөндөмдүү	КК-11, КК-12
5	Түздөө-оңдоо ишмердүүлүгү: өздүк менчик программалар жана программалык комплекстерди талдап иштеп чыгуу, оңдоп түздөөдөн (отладка) өткөрүү, модернизациялоо; программаларды жана программалык комплекстерди оңдоп түзөө, тестирлөө үчүн усулдарды жана каражаттарды тандай, баалай билүү; эсептөө системаларынын жана автоматташтырылган системалардын аспаптык жана колдонмо программалык каражаттарды адаптациялоо жана тейлөөгө жөндөмдүү	КК-9, КК-10
6	Илимий-изилдөө ишмердүүлүгү: илимий-изилдөөдөгү, долбоордук-конструктордук ишмердүүлүктөгү, башкаруу системаларындагы компьютердик технологиялардын жана чечим кабыл алууну колдоочу системалардын математикалык үлгүлөрүн, ыкмаларын тандоо; кабыл алынган долбоорлоо чечимдерин шартто жана анын тууралыгын аныктоо үчүн эксперименттерди өткөрүү; аткарылган иштердин жыйынтыктарынын негизинде илимий-техникалык отчетторду,	ЖИК-5, КК-6, КК-7,

	презентацияларды даярдоо, изилдөө жыйынтыктарын статья, доклад түрүндө илимий-техникалык конференцияларда чыгарууга жөндөмдүү	
7	Инновациялык ишмердүүлүгү: өз ишмердүүлүк чөйрөсүндө үзгүлтүксүз пайда болуп жаткан жаңы инновацияларды өздөштүрүү, колдоно билүү жана аларга атаандаш идеяларды түзөө алуу; жаңы инновациялардын эффективдүүлүгүн баалай билүү, традициялык жана инновациялык ойлордун айырмачылыктарын изилдөө, түшүнө билүү, жаңы инновацияларды пайдалануу менен заманбап программалардын жана программалык комплекстердин үстүнөн иштөөгө, катышууга жөндөмдүү	ЖИК-4
8	Сервистик-тейлөө ишмердүүлүгү: жөнөкөй жана пайдалануучулардын бардык катмарларын эске алуу менен жасалган интерфейске ээ болгон программалык өндүрүмдөрдү түзүүгө жана пайдалана билүү, программалардын жана программалык комплекстердин үзгүлтүксүз иштеп туруусун жана алардын бүтүндүгүн камсыздап туруу, программалардын жана программалык комплекстерди сервистик зарылчылыкка жараша адаптациялоого жана оңдоп түзөөгө жөндөмдүү	ЖОЖК-2
9	Ресурстарды сарамжалдуу пайдалануу ишмердүүлүгү: ресурстарды сарамжалдуу колдонууга, табиятты коргоого, программалык комплекстерди сарамжалдуу пайдаланууну камсыз кылууга жөндөмдүү	ЖИК-1, СИЖМК-4
10	Мамлекеттик жана расмий тилдерде иштөөжөндөмдүүлүгү: мамлекеттик жана расмий тилдерде өзүнүн оозеки жана жазуу сүйлөмдөрүн логикалык туура, аргументтүү жана ачык-айкын түзүүгө жана иш жүргүзө билүүгө, аткарылган илимий-техникалык жана өндүрүштүк маселелердин отчетторун түзө алат, аларды презентация, статья, доклад формасында баяндай алууга жөндөмдүү	ЖИК-4, АК-2, КК-6, КК-7
11	Дүйнөлүк маалыматтарды колдонуу ишмердүүлүгү: адис өзүнүн ишмердүүлүк чөйрөсү боюнча англис тилинде эркин баарлашууга; башка дүйнөлүк тилдерде жарыяланган маалыматтарды алууга жана аларды өз ишмердүүлүк чөйрөсүндө колдоно билүүгө жөндөмдүү	АК-3, ЖИК-2
12	Мамлекеттин өнүгүшүнө салым кошуу ишмердүүлүгү: жигердүү атуулдук позицияда болууда, атуулдук демократиялык коомдун баалуулуктарынын негизинде диалог жүргүзө билүүгө, экономикалык эффективдүүлүгү жогору болгон өндүрүмдөрдү (программаларды, программалык комплекстерди ж.б.) иштеп чыгууга жөндөмдүү. Региондорду өнүктүрүүрүүсүнө жана санариптештирүүсүнө салым кошууга жөндөмдүү	СИЖМК-3, КК-5, КК-9, КК-10, КК-11
13	Электрондук маалыматтар дүйнөсүндө иштөө жөндөмдүүлүгү: ар кандай электрондук маалыматтарды түзө билүүгө, окуй алууга, колдоно билүүгө, интернет тармагы боюнча демократиялык баалуулуктардын негизинде диалог жүргүзүүгө жөндөмдүү; команданын курамында интранет, интернет тармактарындагы электрондук маалыматтарды иштете алууга жөндөмдүү. региондорду өнүктүрүүсүнө жана	АК-1, СИЖМК-3, КК-1, ЖИК-3, КК-5, КК-9, КК-10, КК-11

	санариптештирүүсүнө салым кошууга жөндөмдүү	
14	Веб чөйрөсүндөгү ишмердүүлүк: интернет тармагында клиент-сервер технологияларын колдонуу менен веб тиркемелерди түзүүгө, берилгендер базаларын түзүүгө жана эн алдынкы технологиядагы берилгендер базасын башкаруу системалар менен иштей билүүгө, интернет тармагы боюнча бизнес долбоорлорду түзө жана ишке ашыра билүүгө, аракеттеги бизнес долбоорлорго катыша билүүгө жөндөмдүү	АК-5, АК-6, КК-4
15	Өздүк өнүгүү ишмердүүлүгү: өздүк жетишкендиктерин баалай билүү жана кемчилдиктерине сын көз менен кароо, жетишкендиктерди андан ары өнүктүрүү жана кемчиликтерди кыска убакытта жоюу; жамааттык ишмердүүлүктү өнүктүрүү, коомдо кабыл алынган моралдык жана укуктук негизде социалдык өз-ара аракеттенишүүгө жөндөмдүү, элге сый, башка маданиятка толеранттуулук жана шериктештик мамилелерди колдоого даярдыкты көрсөтөт, ишмердик баарлашууну жүзөгө ашырууга жөндөмдүү: эл алдына чыгып сүйлөө, сүйлөшүүлөр, кеңешме өткөрүү, ишмердүүлүктө жазуу түрүндө баарлашууга жөндөмдүү	АК-4, СИЖМК-1, СИЖМК-2, СИЖМК-5, ЖИК-6

II. РАБОЧАЯ ПРОГРАММА

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
ОШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ


ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Автоматизированных систем и цифровых технологий»

«Утверждено»

на заседании кафедры АСЦТ

прот. № 1 от «28» августа 2020 г.

Зав. каф. АСЦТ, доц.:  Молдоярв У.

РАБОЧАЯ ПРОГРАММА

Дисциплина: «База данных»

Профиль подготовки: ИВТ
Курс: 2
Группа: ИВТ(м)-1-19

для магистрантов очного отделения, обучающихся по специальности:

ИВТ

Сетка часов по учебному плану

Наименование дисциплин	Количество часов					СРС	Отчетность
	Всего	Аудит. занятия					
		Ауд. зан.	Лекция	Лаб.	Модуль, ТК, РК		экзамен
Базы данных	90	36	20	16	4	54	экзамен
III семестр	90	36	20	16	4	54	1 с. - конс.

Рабочая программа составлена на основе Государственного образовательного стандарта по направлению 710100 «ИВТ» для магистрантов обучающихся в очном отделении.

Составитель, доцент каф. АСЦТ: Беделова Н.С.

2020-2021 учебный год

2.1. Целью учебной дисциплины «Базы данных» является изучение теоретических основ построения и функционирования баз данных, характеристик современных СУБД, современных технологий организации БД и приобретение навыков работы в среде конкретных СУБД.

Задачи учебной дисциплины

Задачами изучения данного курса является формирование у магистранта системы знаний о современных методах хранения и обработки информации и выработка у обучающихся устойчивых навыков их использования, что является неотъемлемым признаком высококлассного специалиста. Особое внимание уделяется вопросам автоматизации проектирования БД, практической работе (проектирование, ведение и использование баз данных) в среде выбранных целевых СУБД. Основной задачей дисциплины является формирование у магистранта четкого понимания о задачах и функциях проектирования баз данных.

2.2. Требования к уровню освоения дисциплины

В результате изучения курса «Базы данных», магистранты должны:

Иметь представление о:

- современных базах и банках знаний;
- принципах построения БД и БЗ;
- логическом и физическом устройстве современных БД;
- средствах защиты данных, перспективах развития БД.

Знать:

- основные понятия БД;
- основы построения и функционирования БД;
- технологии организации БД;
- язык создания и манипулирования данными SQL;
- способы защиты данных;
- приемы работы в распределенных и многопользовательских БД.

Уметь:

- построить модель предметной области;
- создать соответствующую модели базу данных в СУБД ACCESS и SQL SERVER;
- организовать ввод информации в базу данных и вывод отчетов;
- формулировать запросы к БД;
- организовать работу в многопользовательской БД.

2.3. Место учебной дисциплины в структуре ООП ВПО

Изучение дисциплины «База данных» основывается на ранее изученных магистрантами предметах. Дисциплина «Базы данных» дает знания о применении и использовании современных информационных технологий; показывает особенности технологии банков данных как одной из основных новых информационных технологий, с тем, чтобы магистранты

понимали тенденции развития современных информационных технологий, видели их преимущества и недостатки, особенности работы в условиях конкретных технологий в их профессиональной деятельности; ориентирует магистрантов во множестве современных СУБД и связанных с ними технологий; освещает теоретические и организационно-методические вопросы построения и функционирования систем, основанных на концепции баз данных, в том числе различные методологии моделирования и проектирования баз данных.

К началу изучению данной дисциплины магистранты должны изучить дисциплины «Информатика и программирование», «Информационные системы». В свою очередь изучения курса «Базы данных» является базой для изучения профессиональных дисциплин.

Дисциплина включает лекционную часть, практические занятия под руководством преподавателя и самостоятельные занятия. Итоговой формой контроля является экзамен по всем темам учебной дисциплины. В качестве текущих форм контроля знаний магистрантов тестирование по всем темам.

2.4. Технологическая карта

Всего	Ауд. часы	СӨАИ	1-модуль (30 б.)				2 модуль (30 б.)				Итог. конт. (ИК) (30 б.)				Поощ. балл	Всего
			Ауд. часы		СӨАИ	1-текущий конт. (ТК ₁)	Ауд. часы		СӨАИ	2-текущий конт. (ТК ₂)	Лекция	Лаборатория	СРС	Итогов. конт. (ИК)		
			Лекция	Лабор.			Лекция	Лабор.								
90	36	54	10	8	28		10	8	26							
Баллы			30	30	30	30	30	30	30	30	30	30	30	30	10	
Модули и итоговые контроли			$TK_{1,2} = (Лек + Лаб + СРС) / 3,$ $M1 = (TK_1 + TK_2 + PK_1) / 3$				$TK_{3,4} = (Лек + Лаб + СРС) / 3,$ $M2 = (TK_3 + TK_4 + PK_2) / 3$				$ИК = (Лек + Лаб + СРС) / 3,$ $Экз = M_1 + M_2 + ИК + С$				100	

2.5. Карта компетенций дисциплины

Модули и названия	ОК-3	ИК-1	ПК-4	Общее количество компетенций
1 часть. Основные понятия		+	+	2
2 часть. Классификация ЭИС	+	+		2

2.6. Карта накапливаемости баллов по дисциплине

Карта накапливаемости баллов – это информация, предоставляющая магистрантам сведения о количестве баллов, получаемых им по всем видам работ, проверок и контроля по каждой теме изучаемой дисциплины.

На первой занятии, кроме общей информации, проводится ознакомление магистрантов с видами работ, контроля и картой накапливаемости баллов. В ходе занятий магистрантам регулярно проводятся напоминания о накапливаемости баллов.

В процессе изучения дисциплины магистранты должны выполнить 15 лабораторных работ оговоренных рабочей программой. На лабораторных работах магистранты осваивают работу с программным обеспечением путем выполнения плановых и индивидуальных заданий. В итоговой работе каждого рубежа в виде СРС дается выполнение проекта.

Оценивание уровня знания магистрантов в модулях проводится следующим образом:

В первом модуле проводятся два текущих контроля (ТК) и один рубежный контроль (РК). Каждый вид контроля оценивается по 30-балльной системе.

На 4 неделе занятий организуется ТК1, на 8 неделе – ТК2, а РК организуется также на 8 неделе.

ТК1 оценивается по результатам освоения магистрантами лекционного материала, пройденных до 4 недели учебного процесса, выполненных лабораторных занятий и самостоятельных работ. Оценивание производится по средне-арифметической системе:

$$TK1 = \frac{\text{Лек} + \text{Лаб} + \text{СРС}}{3}$$

ТК2 оценивается по результатам освоения магистрантами лекционного материала, пройденных с 4 недели по 8 неделю учебного процесса, выполненных лабораторных занятий и самостоятельных работ. Оценивание производится по средне-арифметической системе:

$$TK2 = \frac{\text{Лек} + \text{Лаб} + \text{СРС}}{3}$$

Результаты оценивания ТК1 и ТК2 фиксируются в групповом журнале и ведомости.

На 8 неделе по материалам первого модуля проводится первый рубежный контроль (РК1). В РК1 включаются все пройденные и освоенные лекционные материалы, выполненные лабораторные и самостоятельные работы первого модуля. Оценивание РК1 производится по средне-арифметической системе:

$$PK1 = \frac{\text{Лек} + \text{Лаб} + \text{СРС}}{3}$$

Оценивание первого модуля (М1) производится путем вычисления средне-арифметического текущих контролей и первого рубежного контроля:

$$M1 = \frac{TK1 + TK2 + PK1}{3}$$

Оценивание второго модуля производится по аналогичной схеме.

Итоговый контроль включает все пройденные лекционные материалы, выполненные лабораторные работы и самостоятельные работы, включенные в текущий семестр. Оценивание производится путем нахождения средне-арифметического:

$$ИК = \frac{Лек + Лаб + СРС}{3}$$

Экзаменационный балл берется от средне-арифметического всех модулей, итогового контроля и дополнительного балла:

$$Экз = M1 + M2 + ИК + Д$$

Д – дополнительный балл. Дополнительный балл (поощрительный) указан в проекте «Система оценивания знаний».

2.7. Программа дисциплины

Тема 1. Введение. Основные понятия

Цели и задачи дисциплины, ее место в учебном процессе. Роль моделей и методов теории ЭИС и БД в процессах разработки, эксплуатации и развития информационных систем. Основные понятия: база данных (БД), банк данных (БнД), система управления базами данных (СУБД), приложения. История развития баз данных (четыре этапа).

Информационная система в общем виде. Понятие информации и данных. Понятие ЭИС, принципы их построения и функционирования. Классификация ЭИС. Компоненты ЭИС. Базы данных (БД). Документальные БД. Фактографические БД. Гипертекстовые и мультимедийные БД. XML-серверы. Объектно-ориентированные БД. Коммерческие БД. Предметная область. Пользователи ЭИС

Тема 2. Введение в банки данных. OLAP и OLTP-системы

Понятие банка данных (БнД). Предпосылки создания БнД. Компоненты БнД. База данных (БД) - ядро БнД. Программные средства БнД. Понятие СУБД. Языковые средства современных СУБД. Классификация языковых средств. Языки четвертого поколения и их свойства. Технические средства БнД. Организационно-методические средства. Классификация БнД.

Хранилища данных (Data Warehouse). Проблема создания и сжатия больших информационных массивов, информационных хранилищ и складов данных. Основы фракталов. Фрактальная математика. Фрактальные методы в архивации. Управление складами данных. Последовательные, списковые структуры. Двоичные деревья сортировки. Индексирование. Хеширование

Тема 3. Принципы построения БД. Инфологическое (концептуальное) моделирование предметной области.

Функции СУБД. Выбор СУБД. Пользователи БД. Администраторы БД (АБД) и их функции. Преимущества и недостатки БД. Этапы проектирования баз данных. Жизненный цикл БД. Архитектура базы данных. Трехуровневая модель СУБД. Схема прохождения запроса пользователя в трехуровневой модели БД. Модели данных. Классификация моделей данных. Понятие предметной области. Состав инфологической модели (ИЛМ). Требования, предъявляемые к ИЛМ. Способы описания предметной области. ER-модели. Объекты и классы объектов. Атрибуты объектов. Типы объектов. Виды связей. Классы членства. Сравнение методик инфологического моделирования. Методология IDEF1X. CASE-средства проектирования БД. Возможности проектирования БД (целевые СУБД, функциональные возможности, оценка алгоритма, оценка CASE-средства в целом). Модели описания предметной области. UML

Тема 4. Теоретико-графовые модели данных. Реляционная модель данных.

Общие сведения о даталогическом моделировании. Особенности даталогических моделей. Факторы, влияющие на проектирование БД. Критерии оценки проекта. Теоретико-графовые модели данных: иерархическая и сетевая, стандарты языков описания данных и манипулирования данными в каждой из моделей. Реляционные модели. Сетевые модели. Особенности СУБД, построенных на основе инвертированных файлов. Объектно-ориентированные БД. Языковые средства ОБД. Особенности создания и использования. Гипертекстовые системы. Характеристика гипертекстовых оболочек. XML - формат данных. Системы документального типа. Объектно-ориентированное проектирование. Основные понятия реляционной алгебры: домен, атрибут, отношение, кортеж. Операции над отношениями. Понятие отношения. Ключи в отношениях. Операции над отношениями. Процедурные и декларативные средства обработки отношений. Реляционная алгебра. Реляционное исчисление. Нормализация отношений. Функциональная зависимость атрибутов. Теоремы о функциональных зависимостях. Многозначные зависимости. Нормальные формы отношений. Функциональные зависимости и корректность операций над отношениями. Характеристика реляционных СУБД. Алгоритм перехода от ER-модели к реляционной; дополнительные рекомендации по проектированию, не вытекающие из ER-модели и теории нормализации Описание баз данных.

Тема 5. Табличные языки запросов

Язык запросов QBE. Общая характеристика языка QBE. Особенности реализации QBE в современных СУБД. Переменные и наполнители. Задание простых и сложных запросов. Влияние типа поля на формулирование запроса. Возможности совместной обработки нескольких файлов, связывание файлов. Вычисляемые поля. Упорядочение данных в ответе. Состав и

порядок следования полей в ответе. Возможности группировки данных, получение подитогов. Использование агрегатных функций. Корректирующие операторы. Запоминание и корректировка запросов. Использование запросов на QBE при обработке данных. Особенности реализации QBE в современных настольных СУБД. Табличные языки как «построители» SQL-запросов.

Тема 6. Язык SQL.

Основные понятия и операторы. Общая характеристика SQL. Стандарты SQL. Реализации SQL в современных СУБД. SQL-серверы. SQL-DDL. Основные операторы SQL: определения данных (DDL), манипулирования данными (DMP), запросов (DQL), администрирования данных. Реализация SQL в конкретном SQL-сервере. Типы данных. Создание баз данных, таблиц, индексов. Ввод и корректировка данных средствами SQL и средствами SQL SERVER. Оператор выбора SELECT: предложения FROM, WHERE, GROUP BY, HAVING, ORDER BY. Вычисляемые поля. Определение состава полей, выводимых в ответ. Вычисляемые поля. Возможности задания условий отбора. Возможности связывания файлов – внутренние (INNER) и внешние объединения. Использование агрегатных функций. Вложенные запросы. Операторы манипулирования данными (DELETE, INSERT, UPDATE). Возможности связывания файлов. Вложенные запросы. Возможности группировки данных, получение подитогов. Использование агрегатных функций. Создание и использование "представлений". Ограничения целостности в стандартах SQL. Хранимые процедуры и триггеры. Встроенный SQL. Понятие курсора. Создание и использование. Особенности реализаций SQL в современных СУБД

Тема 7. Целостность баз данных. Физические модели. Манипулирование данными.

Понятие целостности. Классификация ограничений целостности. Причины, вызывающие нарушение ограничений целостности. Способы задания ограничений целостности в современных СУБД: процедурный и декларативный способы задания ограничений целостности. Ограничения целостности в стандартах SQL. Файловые структуры. Организация файлов БД. Способы упорядочения файлов. Файлы прямого, последовательного доступа. Индексные файлы (плотный, неплотный индекс), В-деревья. Команды поиска. Оптимизация поиска. Инвертированные списки. Корректировка БД. Обработка взаимосвязанных файлов. Бесфайловая организация данных. Структура хранения для MS SQL: типы и структуры страниц.

Тема 8. Организация ввода данных в базу данных. Вывод информации из баз данных. Разработка приложений

Способы ввода данных в базу данных. Оформление экрана. Создание и использование экранных форм. Использование приемов, рационализирующих процесс ввода данных. Контроль вводимых данных. Возможности генераторов отчетов современных СУБД. Задание формы и состава документа. Введение вычисляемых полей. Получение документов,

включающих несколько степеней итогов. Получение документов на основе нескольких связанных файлов. Графическое оформление документа. Возможности получения сложных документов, включающих несколько степеней итогов; создание документов, включающих разнородные части. Получение документов на основе нескольких связанных файлов. Вывод документа на печать, на дисплей и в файл. Создание "этикеток". Использование генераторов форм для получения выходных документов. Генераторы приложений в современных СУБД. Создание меню. Визуальное программирование. Средства документирования проекта.

Тема 9. Распределенные БД

Понятие распределенных БД. Централизованные и распределенные системы. Понятия транзакции, удаленного запроса, распределенной транзакции. Функции приложения и их распределение в различных моделях. Работа с базами данных в режимах "файл-сервер" и "клиент-сервер". Двух- и трехуровневые системы клиент-сервер. Модели транзакций. Журнал транзакций. Проблемы параллельного выполнения транзакций. Блокировки, виды блокировок. Технологии тиражирования. Репликации. Особенности работы в глобальных сетях. Сетевые возможности современных СУБД. Особенности создания баз данных, функционирующих в локальных сетях. Особенности работы с базами данных в многопользовательском режиме. Особенности программирования обработки баз данных при многопользовательском режиме доступа. Проектирование приложений типа "клиент-сервер". Клиентское и серверное ПО. Особенности работы в гетерогенной среде. Стандарты интерфейсов. Особенности администрирования в распределенных БД. Публикация БД в Интернет.

Тема 10. Безопасность данных. Перспективы развития БД и СУБД.

Понятие безопасности данных. Классификация безопасных систем. Уровни защиты. Способы обеспечения безопасности данных в современных СУБД. Способы обеспечения безопасности данных в MS SQL-Server. Пользователи БД. Администраторы БД и их функции. Понятие коммерческих баз данных (КБД). Характеристика секторов рынка информации. Способы распространения КБД. Субъекты информационного рынка. Вопросы ценообразования. Источник информации о КБД. Программные средства поддержки КБД. Критерии выбора КБД. Юридические базы данных.

2.8. Тематический план распределения часов по видам занятий

№	Названия разделов и тем	Виды учеб. занятий		СРС
		Ауд. занятия, в том числе		
		Лекции	Лаб. занятия	
	Введение. Основные понятия	2	1	2

	Введение в банки данных. OLAP и OLTP-системы	2	1	3
	Принципы построения БД. Инфологическое (концептуальное) моделирование предметной области.	2	2	3
	Теоретико-графовые модели данных. Реляционная модель данных.	2	2	3
	Табличные языки запросов	2	2	3
	Язык SQL.	2	2	3
	Целостность баз данных. Физические модели. Манипулирование данными.	2	2	3
	Организация ввода данных в базу данных. Вывод информации из баз данных. Разработка приложений	2	2	3
	Распределенные БД	2	2	3
	Безопасность данных. Перспективы развития БД и СУБД.	2	1	3
	Всего:	20	16	29

2.9.2. Содержание лабораторных занятий

Подготовка к лабораторным занятиям является одной из важнейших форм самостоятельной работы магистрантов. Целью практических занятий является закрепление знаний полученных на лекционных занятиях и в ходе самостоятельной работы, а также выработка навыков работы с учебной и научной литературой. Приступая к решению задачи, магистрант должен, прежде всего, уяснить содержание задачи. Подготовку к занятиям следует начинать с внимательного изучения соответствующих разделов учебных пособий и учебников. По завершению изучения рекомендованной литературы, магистранты могут проверить свои знания с помощью вопросов для самоконтроля, входящих в состав вопросов к зачету (экзамену) по соответствующему разделу (темы) изучаемой дисциплины.

2.9.3. Методические указания по организации самостоятельной работы магистрантов

Самостоятельная работа магистрантов

№	Наименование и план темы	Вопросы, выносимые на самостоятельную работу магистрантов	Виды и содержание самостоятельной работы
1	Введение. Основные понятия	Цели и задачи дисциплины, ее место в учебном процессе. Роль моделей и методов теории ЭИС и БД в процессах разработки,	Проработка учебного материала (по конспектам лекций,

		<p>эксплуатации и развития информационных систем. Основные понятия: база данных (БД), банк данных (БнД), система управления базами данных (СУБД), приложения. История развития баз данных (четыре этапа).</p> <p>Информационная система в общем виде. Понятие информации и данных. Понятие ЭИС, принципы их построения и функционирования. Классификация ЭИС. Компоненты ЭИС. Базы данных (БД). Документальные БД. Фактографические БД. Гипертекстовые и мультимедийные БД. XML-серверы. Объектно-ориентированные БД. Коммерческие БД. Предметная область. Пользователи ЭИС</p>	<p>научной и учебной литературе) Работа с нормативными документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации</p>
2	Введение в банки данных. OLAP и OLTP-системы	<p>Понятие банка данных (БнД). Предпосылки создания БнД. Компоненты БнД. База данных (БД) - ядро БнД. Программные средства БнД. Понятие СУБД. Языковые средства современных СУБД. Классификация языковых средств. Языки четвертого поколения и их свойства. Технические средства БнД. Организационно-методические средства. Классификация БнД. Хранилища данных (Data Warehouse). Проблема создания и сжатия больших информационных массивов, информационных хранилищ и складов данных. Основы фракталов. Фрактальная математика. Фрактальные методы в архивации. Управление складами данных. Последовательные, списковые структуры. Двоичные деревья сортировки. Индексирование. Хеширование</p>	<p>Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Поиск и обзор научных публикаций и электронных источников информации</p>
3	Принципы построения БД. Инфологическое (концептуальное) моделирование предметной области.	<p>Функции СУБД. Выбор СУБД. Пользователи БнД. Администраторы БнД (АБД) и их функции. Преимущества и недостатки БнД. Этапы проектирования баз данных. Жизненный цикл БД. Архитектура базы данных. Трехуровневая модель СУБД. Схема прохождения запроса пользователя в трехуровневой модели БД. Модели данных. Классификация моделей данных. Понятие предметной области. Состав инфологической модели (ИЛМ). Требования, предъявляемые к ИЛМ. Способы описания предметной области. ER-модели. Объекты и классы объектов. Атрибуты объектов. Типы объектов. Виды</p>	<p>Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Поиск и обзор научных публикаций и электронных источников информации</p>

		связей. Классы членства. Сравнение методик инфологического моделирования. Методология IDEF1X. CASE-средства проектирования БД. Возможности проектирования БД (целевые СУБД, функциональные возможности, оценка алгоритма, оценка CASE-средства в целом). Модели описания предметной области. UML	
4	Теоретико-графовые модели данных. Реляционная модель данных.	Общие сведения о даталогическом моделировании. Особенности даталогических моделей. Факторы, влияющие на проектирование БД. Критерии оценки проекта. Теоретико-графовые модели данных: иерархическая и сетевая, стандарты языков описания данных и манипулирования данными в каждой из моделей. Реляционные модели. Сетевые модели. Особенности СУБД, построенных на основе инвертированных файлов. Объектно-ориентированные БД. Языковые средства ОБД. Особенности создания и использования. Гипертекстовые системы. Характеристика гипертекстовых оболочек. XML - формат данных. Системы документального типа. Объектно-ориентированное проектирование. Основные понятия реляционной алгебры: домен, атрибут, отношение, кортеж. Операции над отношениями. Понятие отношения. Ключи в отношениях. Операции над отношениями. Процедурные и декларативные средства обработки отношений. Реляционная алгебра. Реляционное исчисление. Нормализация отношений. Функциональная зависимость атрибутов. Теоремы о функциональных зависимостях. Многочленные зависимости. Нормальные формы отношений. Функциональные зависимости и корректность операций над отношениями. Характеристика реляционных СУБД. Алгоритм перехода от ER-модели к реляционной; дополнительные рекомендации по проектированию, не вытекающие из ER-модели и теории нормализации Описание баз данных.	Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Поиск и обзор научных публикаций и электронных источников информации
5	Табличные языки запросов	Язык запросов QBE. Общая характеристика языка QBE. Особенности реализации QBE в современных СУБД. Переменные и наполнители. Задание	Проработка учебного материала (по конспектам лекций, научной и учебной

		<p>простых и сложных запросов. Влияние типа поля на формулирование запроса. Возможности совместной обработки нескольких файлов, связывание файлов. Вычисляемые поля. Упорядочение данных в ответе. Состав и порядок следования полей в ответе. Возможности группировки данных, получение подитогов. Использование агрегатных функций. Корректирующие операторы. Запоминание и корректировка запросов. Использование запросов на QBE при обработке данных. Особенности реализации QBE в современных настольных СУБД (Access, Paradox и др.). Табличные языки как «построители» SQL-запросов.</p>	<p>литературе) Работа с нормативными документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации</p>
6	Язык SQL.	<p>Основные понятия и операторы. Общая характеристика SQL. Стандарты SQL. Реализации SQL в современных СУБД. SQL-серверы. SQL-DDL. Основные операторы SQL: определения данных (DDL), манипулирования данными (DMP), запросов (DQL), администрирования данных. Реализация SQL в конкретном SQL-сервере. Типы данных. Создание баз данных, таблиц, индексов. Ввод и корректировка данных средствами SQL и средствами SQL SERVER. Оператор выбора SELECT: предложения FROM, WHERE, GROUP BY, HAVING, ORDER BY. Вычисляемые поля. Определение состава полей, выводимых в ответ. Вычисляемые поля. Возможности задания условий отбора. Возможности связывания файлов – внутренние (INNER) и внешние объединения. Использование агрегатных функций. Вложенные запросы. Операторы манипулирования данными (DELETE, INSERT, UPDATE). Возможности связывания файлов. Вложенные запросы. Возможности группировки данных, получение подитогов. Использование агрегатных функций. Создание и использование "представлений". Ограничения целостности в стандартах SQL. Хранимые процедуры и триггеры. Встроенный SQL. Понятие курсора. Создание и использование. Особенности реализаций SQL в современных СУБД</p>	<p>Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Работа с нормативными документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации</p>
7	Целостность баз данных.	<p>Понятие целостности. Классификация ограничений целостности. Причины,</p>	<p>Проработка учебного материала (по</p>

	Физические модели. Манипулирование данными.	вызывающие нарушение ограничений целостности. Способы задания ограничений целостности в современных СУБД: процедурный и декларативный способы задания ограничений целостности. Ограничения целостности в стандартах SQL. Файловые структуры. Организация файлов БД. Способы упорядочения файлов. Файлы прямого, последовательного доступа. Индексные файлы (плотный, неплотный индекс), В-деревья. Команды поиска. Оптимизация поиска. Инвертированные списки. Корректировка БД. Обработка взаимосвязанных файлов. Бесфайловая организация данных. Структура хранения для MS SQL: типы и структуры страниц.	конспектам лекций, научной и учебной литературе) Работа с нормативными документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации
8	Организация ввода данных в базу данных. Вывод информации из баз данных. Разработка приложений	Способы ввода данных в базу данных. Оформление экрана. Создание и использование экранных форм. Использование приемов, рационализирующих процесс ввода данных. Контроль вводимых данных. Возможности генераторов отчетов современных СУБД. Задание формы и состава документа. Введение вычисляемых полей. Получение документов, включающих несколько степеней итогов. Получение документов на основе нескольких связанных файлов. Графическое оформление документа. Возможности получения сложных документов, включающих несколько степеней итогов; создание документов, включающих разнородные части. Получение документов на основе нескольких связанных файлов. Вывод документа на печать, на дисплей и в файл. Создание "этикеток". Использование генераторов форм для получения выходных документов. Генераторы приложений в современных СУБД. Создание меню. Визуальное программирование. Средства документирования проекта.	Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Работа с нормативными документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации
9	Распределенные БД	Понятие распределенных БД. Централизованные и распределенные системы. Понятия транзакции, удаленного запроса, распределенной транзакции. Функции приложения и их распределение в различных моделях. Работа с базами	Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Работа с нормативными

		<p>данных в режимах "файл-сервер" и "клиент-сервер". Двух- и трехуровневые системы клиент-сервер. Модели транзакций. Журнал транзакций. Проблемы параллельного выполнения транзакций. Блокировки, виды блокировок. Технологии тиражирования. Репликации. Особенности работы в глобальных сетях. Сетевые возможности современных СУБД. Особенности создания баз данных, функционирующих в локальных сетях. Особенности работы с базами данных в многопользовательском режиме. Особенности программирования обработки баз данных при многопользовательском режиме доступа. Проектирование приложений типа "клиент-сервер". Клиентское и серверное ПО. Особенности работы в гетерогенной среде. Стандарты интерфейсов. Особенности администрирования в распределенных БД. Публикация БД в Интернет.</p>	<p>документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации</p>
10	<p>Безопасность данных. Перспективы развития БД и СУБД.</p>	<p>Понятие распределенных БД. Централизованные и распределенные системы. Понятия транзакции, удаленного запроса, распределенной транзакции. Функции приложения и их распределение в различных моделях. Работа с базами данных в режимах "файл-сервер" и "клиент-сервер". Двух- и трехуровневые системы клиент-сервер. Модели транзакций. Журнал транзакций. Проблемы параллельного выполнения транзакций. Блокировки, виды блокировок. Технологии тиражирования. Репликации. Особенности работы в глобальных сетях. Сетевые возможности современных СУБД. Особенности создания баз данных, функционирующих в локальных сетях. Особенности работы с базами данных в многопользовательском режиме. Особенности программирования обработки баз данных при многопользовательском режиме доступа. Проектирование приложений типа "клиент-сервер". Клиентское и серверное ПО. Особенности работы в гетерогенной среде. Стандарты интерфейсов. Особенности администрирования в распределенных БД. Публикация БД в</p>	<p>Проработка учебного материала (по конспектам лекций, научной и учебной литературе) Работа с нормативными документами и законодательной базой Поиск и обзор научных публикаций и электронных источников информации</p>

	Интернет.	
--	-----------	--

2.10. Задания итогового контроля.

1. Понятие информации и данных.
2. Понятие ЭИС Классификация ЭИС.
3. Понятия банка данных.
4. Предметная область.
5. Классификация и основные свойства единиц информации.
6. Экономические показатели. Определение и структура показателя.
7. Понятия банка данных. Преимущества и недостатки.
8. Компоненты банков данных и их краткая характеристика.
9. Языковые средства СУБД.
10. Классификация банков данных.
11. Тенденции развития СУБД.
12. Характеристика современных реляционных СУБД.
13. Языки 4-го поколения и их реализация в современных СУБД.
14. Этапы проектирования баз данных.
15. Пользователи банков данных.
16. Функции администраторов банков данных.
17. Технические средства банков данных.
18. Инфологическое моделирование.
19. Сравнение методик ER-моделирования.
20. ERWin. Виды сущностей и их создание.
21. ERWin . Виды связей.
22. ERWin. Особенности построения ER-модели.
23. Базовая ER-модель. Виды сущностей.
24. Базовая ER-модель. Виды связей.
25. Особенности проектирования реляционных БД.
26. Дatalogическое моделирование.
27. Факторы, влияющие на проектирование баз данных.
28. Виды связей между объектами и их отражение в дatalogической модели.
29. Критерии оценки БД.
30. Основные понятия: отношение, ключ, вероятный ключ, внешний ключ.
31. Особенности реляционных моделей
32. Нормальные формы
33. Нормализация отношений.
34. Реляционная алгебра
35. Алгоритм перехода от ER-модели к структуре реляционной базы данных
36. CASE -средства проектирования БД
37. Возможности совместной обработки таблиц в реляционных СУБД
38. Возможности задания ограничений целостности в современных СУБД.
39. Ограничения целостности. Понятие и классификация.
40. Ограничения целостности связи. Понятие. Возможности реализации в современных СУБД.
41. Задание ограничений целостности в ERWin
42. Последовательные и списковые структуры.
43. Двоичные деревья сортировки.
44. Индексирование.
45. Хеширование.
46. Способы организации ввода данных в базу данных
47. Классификация экранных форм

48. Задание ограничений целостности в экранных формах
 49. Характеристика генератора экранных форм конкретной СУБД.
 50. Экранные формы для ввода и корректировки данных.
 51. Табличные языки запросов. Общая характеристика.
 52. Классификация запросов.
 53. Особенности обработки полей разных типов. Работа с вычисляемыми полями.
 54. Использование агрегирующих функций.
 55. Совместная обработка таблиц
 56. Реализация запросов со сложными условиями в табличных языках.
 57. Корректировка данных с помощью табличного языка.
 58. Язык SQL. Общая характеристика.
 59. Стандарты SQL.
 60. Общая структура команды Select языка SQL.
 61. Вложенные запросы в SQL.
 62. Корректировка данных в SQL.
 63. SQL. Создание объектов.
 64. SQL. Встроенный JOIN.
 65. SQL. Понятие курсора.
 66. SQL. Группировка данных. Использование обобщающих функций.
 67. SQL. Возможности совместной обработки таблиц.
 68. SQL. Упорядочение данных. Операция объединения.
 69. SQL. Возможности задания состава колонок, выводимых в ответ.
 70. SQL. Возможности задания условий отбора.
 71. SQL. Создание и использование представлений.
 72. SQL. Хранимые процедуры и триггеры.
 73. Разновидности и структура выходных документов
 74. Использование экранных форм для вывода данных
 75. Возможности генераторов отчетов современных СУБД.
 76. Генераторы приложений в современных СУБД.
 77. Создание меню
 78. Классификация распределенных банков данных.
 79. Особенности проектирования распределенных БД.
 80. Проблемы обеспечения целостности в распределенных БД.
 81. Сравнение централизованных и распределенных систем.
 82. Распределенные БД. Технология клиент - сервер.
 83. Распределенные базы данных. Технология тиражирования.
 84. Сетевые возможности современных СУБД.
 85. Распределенные гетерогенные банки данных.
 86. Проблемы, возникающие при параллельном доступе, и пути их решения.
 87. Определение безопасности данных.
 88. Назовите классы безопасных систем..
 89. Перечислить уровни защиты.
 90. Способы обеспечения безопасности данных в современных СУБД.
- 2.11. Учебно-методическое обеспечение дисциплины

2.11.1. Основная

1. М. Кузнецов, И. Симдянов MySQL 5. – Издат.: БХВ-Петербург, 2016. – 1007 с.
2. Гольцман В. MySQL 5 – Издат.: Питер, 2017. – 253 с.

2.11.2. Дополнительная

3. Бен Форта Освой самостоятельно MySQL. – Издат.: Вильямс, 2015. – 287 с.
4. Грабер, Мартин SQL. Справочное руководство; М.: Лори; Издание 2-е, 2016. – 354 с

5. Шнайдер, Роберт Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных; М.: Лори, 2017. - 361 с.

6. Яргер, Р.Дж.; Риз, Дж.; Кинг, Т. MySQL и mSQL: Базы данных для небольших предприятий и Интернета; СПб: Символ-Плюс, 2018. - 560 с.

2.12. Критерии оценки знаний магистрантов на экзамене.

В III семестре установлен экзамен, результаты по дисциплине общий балл (*Обалл*) ставится следующим образом:

Рейтинг (балл)	Оценка по балльной системе	Оценка GPA по цифровой эквивалентности	Оценка по традиционной системе
87 – 100	A	4,0	Отлично
80 – 86	B	3,33	Хорошо
74 – 79	C	3,0	
68 – 73	D	2,33	Удовлетворительно
61 – 67	E	2,0	
31 -60	FX	0	Не удовлетворительно

В соответствии с действующими нормативными актами и рекомендациями Министерства образования и науки КР устанавливаются следующие критерии выставления оценок на экзаменах:

- оценка **"отлично"** выставляется магистранту, который обнаружил на экзамене всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, который усвоил основную литературу и ознакомился с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется магистрантам, усвоившим взаимосвязь основных понятий дисциплины и их значений для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала;

- оценка **"хорошо"** выставляется магистранту, который на экзамене обнаружил полное знание учебно-программного материала, успешно выполнил предусмотренные в программе задания, усвоил основную литературу, рекомендованную в программе. Как правило, оценка "хорошо" выставляется магистрантам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному выполнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности;

- оценка **"удовлетворительно"** выставляется магистранту, обнаружившему знание основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, предусмотренных программой, который ознакомился с основной литературой, рекомендованной программой. Как правило, оценка "удовлетворительно" выставляется магистрантам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя;

- оценка **"неудовлетворительно"** выставляется магистранту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий, не ознакомившемуся с основной литературой, предусмотренной программой, и не овладевшему базовыми знаниями, предусмотренными по данной дисциплине и определенными соответствующей программой дисциплины.

**КЫРГЫЗ РЕСПУБЛИКАСЫНЫН
БИЛИМ БЕРҮҮ ЖАНА ИЛИМ МИНИСТРЛИГИ
ОШ МАМЛЕКЕТТИК УНИВЕРСИТЕТИ
МАТЕМАТИКА ЖАНА ИНФОРМАЦИЯЛЫК ТЕХНОЛОГИЯЛАР ФАКУЛЬТЕТИ
АСЦТ КАФЕДРАСЫ**

«Макулдашылды»
МИТ факультетинин Методика-
лык кеңешинин төрайымы
п.и.к., доцент: Д. Зулпукарова
«__» _____ 2020-ж.

«Бекитилди»
АСЦТ кафедрасынын
2020-жылдын 28-августунда
өткөрүлгөн №1-протоколунда
Каф. башч., доц.: Молдоярров

у.

МАГИСТРАНТТЫН ОКУУ ПРОГРАММАСЫ (СИЛЛАБУС)

Дисциплина: Берилгендер базасы

Багыты: 710100 «Информатика жана эсептөө техникасы»

Окутуунун формасы: күндүзгү (магистр)

Окуу жылы: 2020-2021

Окуу планы боюнча сааттардын эсеби

Дисциплинанын аталышы	Сааттардын саны					МӨАИ	Отчеттуулук
	Баары	Аудиториялык сааттар					
		Ауд. саат	Лекция	Лаб.	УТ, модуль		II
ББ	90	36	20	16	4	54	экзамен
2-курс, III с., 3 кредит	90	36	20	16	4	54	1 с. - конс.

Силлабус 710100 “Информатика жана эсептөө техникасы” багытынын мамлекеттик билим берүү стандартынын негизги билим берүү программасынын жана ОшМУнун №19-бюллетенин негизинде түзүлдү.

Түзгөн, ф.-м.и.к., доцент: _____ Беделова Н.С.

Ош – 2020

Окутуучу жөнүндө маалымат

Лектор-окутуучу:

Беделова Нургуль Салибаевна – ф.-м.и.к., доцент, ОшМУ, МИТ факультети, АСЦТ кафедрасы. Стажы – 22 жыл, билими – жогорку, ОшМУнун Физика жана математика факультетин 2000-жылы бүтүргөн.

Иш телефону: 03222-2-11-85, Иш орду: 723500. ОшМУнун башкы корпусу, Ленин к. 331, 224-каб.

Моб. телефону: +996(772)437331, e-mail: kireshe78@mail.ru

3.1. Дисциплинанын максаты

Берилгендер базасы дисциплинасын окутуу MySQL программалоо тилинин негизинде жүргүзүлөт.

Курсту окутуунун максаты **НББПнын 4-максатынан** келип чыгат:

- информациялык системалардын берилгендер базасын түзүү, кошуу, автоматташтыруучу жумушчу инструменттерин сабаттуу өздөштүрүү;
- логикалык жана программисттик ой-жүгүртүүнү, интеллектуалдык жана чыгармачылык жөндөмдүүлүктөрдү өстүрүү;
- MySQL берилгендерди башкаруу системасын колдонуу менен берилгендер базасын сапаттуу түзүү жана жыйынтыктарын көрсөтүү.

3.2. Дисциплинаны үйрөтүү процессинде жетишүүчү окутуу натыйжалары жана калыптандырылуучу компетенциялар

3.2.1. Окутуунун натыйжалары

Дисциплинаны окутуунун натыйжалары НББПнын 13-14-окутуу натыйжаларынан келип чыгат:

Электрондук маалыматтар дүйнөсүндө иштөө жөндөмдүүлүгү: ар кандай электрондук маалыматтарды түзө билүүгө, окуй алууга, колдоно билүүгө, интернет тармагы боюнча демократиялык баалуулуктардын негизинде диалог жүргүзүүгө жөндөмдүү; команданын курамында интранет, интернет тармактарындагы электрондук маалыматтарды иштете алууга жөндөмдүү (**ОН₁₃**);

Веб чөйрөсүндөгү ишмердүүлүк: интернет тармагында клиент-сервер технологияларын колдонуу менен веб тиркемелерди түзүүгө, берилгендер базаларын түзүүгө жана эң алдынкы технологиядагы берилгендер базасын башкаруу системалар менен иштей билүүгө, интернет тармагы боюнча бизнес долбоорлорду түзө жана ишке ашыра билүүгө, аракеттеги бизнес долбоорлорго катыша билүүгө жөндөмдүү (ОН₁₄).

3.2.2. Калыптандырылуучу компетенциялар

- жогорку даражадагы өз-алдынчалуулукта заманбап билим берүү жана информациялык технологияларын колдонуу менен жаңы билим алууга жөндөмдүү (ЖИК-3);
- маалыматты кабыл алууга, жалпылоого жана талдоого, максат коюуга жана ага жетүүнүн жолдорун тандоого жөндөмдүү (АК-1);
- маалымат системдеринин компоненттеринин, маалымат базаларынын үлгүлөрүн иштеп чыгуу (КК-4).

Дисциплинаны өздөштүрүүдө магистр төмөнкү окутуу натыйжаларына жетишет:

- *MySQL берилгендерди башкаруу системасынын түшүнүктөрүн билет, кесиптик маселелерди программалоо методдору менен чечүүнүн негизги жолдорун билет;*
- *маселелердин моделдерин, берилгендердин базасын башкаруу системасын түзө алат жана алынган жыйынтыктарды сыноо жолу менен ишке ашыра алат;*
- *MySQL берилгендерди башкаруу системасында берилгендер базасын түзүү менен жаңы инновацияларды өздөштүрүүгө, колдонууга, анализдөөгө жана синтездөөгө жөндөмдүү.*

3.3. НББПнын структурасындагы дисциплинанын орду

«Берилгендердин базасы» дисциплинасы окуу планынын профессионалдык циклинин базалык бөлүмүнө кирет (Б3).

Бул дисциплинаны үйрөнүү профессионалдык циклинин базалык бөлүмүндөгү (Б3) программалоо жана маалыматтарды коргоо дисциплиналарга негизделет.

Дисциплинанын негизги жоболору мындан ары *Web программалоо* дисциплиналарын үйрөнүүдө пайдаланылат.

3.4. Курстун пререквизити: Информатика

3.5. Курстун постреквизити: БББС

3.6. Технологиялык карта

Баары	Ауд. саат	СӨАИ	1-модул (30 б.)				2-модул (30 б.)				Жыйынт. текш. (ЖТ) (30 б.)				Сыйлык балл	Жалпы балл
			Ауд. саат		СӨАИ	1-аралыктагы текш. (АТ ₁)	Ауд. саат		СӨАИ	2-аралыктагы текш. (АТ ₂)	Лекция	Лаборатория	СӨАИ	Жыйынт. тек. (ЖТ)		
			Лекция	Лабор.			Лекция	Лабор.								
90	36	54	10	8	28	10	8	26	30	30	30	30	30	30	10	
Баллдар			30	30	30	30	30	30	30	30	30	30	30	30	10	
Модулдар жана жыйынтык текшерүүлөр			$УТ_{1,2}=(Лек+Лаб+СӨАИ)/3,$ $М1=(УТ_1+УТ_2+АТ_1)/3$				$УТ_{3,4}=(Лек+Лаб+СӨАИ)/3,$ $М2=(УТ_3+УТ_4+АТ_2)/3$				$ЖТ=(Лек+Лаб+СӨАИ)/3,$ $Экз=М_1+М_2+ЖТ+С$				100	

Ауд. – аудиториялык, УТ – учурдагы текш., АТ – аралык текш., М – модулдар, СӨАИ – магистрин өз алдынча иши, ЖТ – жыйынтыктоочу текш., С – сыйлык балл.

3.7. Дисциплина боюнча балл топтоонун картасы

Баллдарды топтоонун картасы – сабактардын бардык түрлөрүндөгү текшерүү боюнча канча балл (максималдуу) ала тургандыгы жөнүндө магистртерге жеткирилүүчү маалымат.

Магистранттар баллдарды модулдарда төмөнкүдөй топтошот:

1-модулда эки учурдагы текшерүү ($УТ_1$, $УТ_2$) жана бир аралыктагы текшерүү ($АТ_1$) уюштурулат. Ар бир текшерүү үчүн 30 баллдык баалоо системасы колдонулат. Баллдар тапшырмалар менен кошо тааныштырылат.

$УТ_1$ текшерүүсү 4-жумада, $УТ_2$ текшерүүсү 8-жумада уюштурулат, ал эми аралыктагы текшерүү дагы 8-жумада уюштурулат.

$УТ_1$ деп 4-жумага чейин өтүлгөн лекциялык материалдарды өздөштүргөндүгү, аткарылган лабораториялык жана өз алдынча иштер боюнча баалоонун арифметикалык орточосун алабыз:

$$УТ_1 = \frac{Лек + Лаб + СӨАИ}{3} .$$

$УТ_2$ деп сабак башталгандан баштап 4-жумадан 8-жумага чейин өтүлгөн лекциялык материалдарды өздөштүргөндүгү, аткарылган лабораториялык жана өз алдынча иштер боюнча баалоонун арифметикалык орточосун алабыз: $УТ_2 = \frac{Лек + Лаб + СӨАИ}{3} .$

Ведомостко жана журналга $УТ_1$, $УТ_2$ лердин жыйынтыктары коюлат.

8-жумада 1-модулдун материалдары боюнча 1-аралыктагы текшерүү уюштурулат. Мында 1-модулда өтүлгөн лекциялык материалдарды өздөштүргөндүгү, аткарылган лабораториялык жана өз

алдынча иштер боюнча баалоонун арифметикалык орточосун алабыз:

$$AT_1 = \frac{Лек + Лаб + С\theta AI}{3} \text{ (аралыктагы текшерүүлөр тест аркылуу алынат).}$$

1-модулда баалоо учурдагы текшерүүлөрдүн жана 1-аралыктагы текшерүүнүн арифметикалык орточосу менен аныкталат:

$$M_1 = \frac{UT_1 + UT_2 + AT_1}{3}.$$

2-модулдагы баалоо 1-модулдагы баалоо сыяктуу эле аткарылат.

Жыйынтыктоочу текшерүүдө семестр ичинде өтүлгөн лекциялык материалдарды өздөштүргөндүгү, аткарылган лабораториялык жана өз алдынча иштер боюнча баалоонун арифметикалык орточосун алабыз:

$$ЖТ = \frac{Лек + Лаб + С\theta AI}{3} \text{ (долбоордун негизинде бааланат).}$$

Экзамендеги баалоо модулдардын жана жыйынтыктоочу текшерүүнүн арифметикалык орточосу менен сыйлык (С) баллдардын суммасы менен аныкталат:

$$Экз = M_1 + M_2 + ЖТ + С,$$

С – сыйлык баллдар. Сыйлык баллдар «Билимди баалоо системасы» жөнүндөгү жободо көрсөтүлгөн.

“Сыйлык балл” - 10 балл магистранттин активдүүлүгүн жогорулатуу максатында берилет. Сыйлык балл төмөндөгү таблица боюнча бөлүштүрүлөт:

№	Иш-чара	Катышуу (балл)	Жеңүү балл
1.	Олимпиадага катышуу (факультеттик деңгээл)	1	2
2.	Олимпиадага катышуу (университеттик деңгээл)	2	3
3.	Олимпиадага катышуу (регионалдык деңгээл)	2	3
4.	Спорттук таймаштарга катышуу	1	3
5.	Коомдук иштерге активдүү катышууга	1	
6.	Долбоорлорго катышууга	1	
7.	Түрдүү конкурстарга катышууга	1	2
8.	“Нур” газетасына макала, ыр чыгарууга	1	
9.	Телеберүүгө катышууга	1	
10.	Практикалардан “Ыраазычылык кат” алып келүүгө	1	
11.	Старостанын аткарган иштерине	1	
12.	Жаштар комитетинде иштегендиги үчүн	2	
13.	Илимий конференциянын уюштуруу иштерине катышууга	1	2
14.	Университеттин жана факультеттин деңгээлиндеги атайын тапшырмаларды аткаргандыгы үчүн	1	
15.	Факультеттин стенгазеталарын чыгарууга		

	активдүү катышкандыгы үчүн		
16.	КВНге катышууга	1	2

3.8. Лекциялык, лабораториялык сабактардын жана өз алдынча иштердин бөлүштүрүлүшүнүн тематикалык планы

№	Лекциялык, лабораториялык сабактардын жана өз алдынча иштердин темаларынын аталыштары	Лек. (саат ы)	Лаб. (саат ы)	СӨА И (саат ы)	Билим берүү техн-ры	Баалоо каражат.
I модуль						
1.	№1 Лекция. Киришүү. Берилгендер базасынын теориясынын негизги түшүнүктөрү №1 лабораториялык жумуш. MySQL 5 серверин орнотуу №1. СӨАИ. SQL тилинин келип чыгышы жана колдонулушу	2	1	4	интерак. метод, ДБТ, КОП, СО	синквейнсуроолор, тест,
2.	№2 лекция. Берилгендердин типтери №2 лабораториялык жумуш. Берилгендер базасын жана таблицаларды түзүү №2. СӨАИ. Linux операциондук системасында берилгендер базасын түзүү	2	1	4	интерак. метод, ДБТ, КОП, СО	суроолор, “миллио-нер” оюну, тест
3.	№3. Лекция. ББ мн иштөө технологиялары №3. Лабораториялык жумуш. Таблицааларды түзүү жана аларды информациялар менен толтуруу №3. СӨАИ. FLOAT, BLOB, TEXT, ENUM жана SET типтеринин өзгөчөлүктөрү	2	2	4	интерак. метод, ДБТ, КОП, СО	гlossарий, суроолор, тест
1-учурдагы текшерүү						
4.	№4. Лекция. Берилгендердин логикалык жана физикалык көз карандысыздыгы. №4. Лабораториялык жумуш. SELECT жана LIKE операторлорунун берилгендер базасында колдонушу.	2	2	6	интерак. метод, ДБТ, КОП, СО	суроолор, гlossарий, “Змеи и лестницы” оюну, тест

	<p>№4. СӨАИ. IN жана BETWEEN операторлорун колдонуп, 25жаштан жогорку кызматкерлердин тизмесин түзгүлө</p>					
5.	<p>№5. Лекция. Берилгендердин моделинин типтери. №5. Лабораториялык жумуш. Камтылган запростор. WHERE ачкыч сөзүнүн колдонулушу №5. СӨАИ. Тексттик берилгендерди шаблондор боюнча изилдөө</p>	2	2	6	интерак. метод, ДБТ, КОП, СО	суроолор, “Змей и лестницы” оюну, тест
6.	<p>№ 6. Лекция. Реляциондук алгебра № 6. Лабораториялык жумуш. Таблицаларды бириктирүү (ички биригүү) №6. СӨАИ. Таблицаларды оперативдик эске жайгаштыруу</p>	2	2	6	интерак. метод, ДБТ, КОП, СО	суроолор, кроссворд, тест
2-учурдагы текшерүү						
7.	<p>№7. Лекция. ББсын проектирлөө №7. Лабораториялык жумуш. Таблицаларды бириктирүү (сырткы биригүү) №7. СӨАИ. 30 жаштан төмөнкү фамилиясы К жана А тамгаларынан башталбаган кызматкерлердин атынын жана жашынын тизмесин түзүү</p>	2	2	6	интерак. метод, ДБТ, КОП, СО	суроолор, “миллио-нер” оюну, тест
8.	<p>№8. Лекция. ББсын концептуалдык долбоорлоо №8. Лабораториялык жумуш. Жазууларды группировкалоо жана COUNT() функциясынын колдонулушу №8. СӨАИ. DISTINCT ачкыч сөзүн пайдаланып, компаниянын кызматкерлеринин кызмат акыларын өсүү тартибинде түзүү</p>	2	2	6	интерак. метод, ДБТ, КОП, СО	суроолор, “дуэль” оюну, тест
9.	<p>№9. Лекция. Берилгендер базасын нормалдаштыруу №9. Лабораториялык жумуш. Берилгендерди редактирлөө,</p>	2	1	6	интерак. метод, ДБТ, КОП,	гlossарий, “Змей и лестницы” оюну, тест.

	жаңылоо жана жок кылуу №9. СӨАИ. HAVING тиркемесинин жардамында түрдүү кызматтагы кызматкерлердин орточо эмгек акысын чыгаруучу программа түзүү				СО		
	1-учурдагы текшерүү						
10	№10. Лекция. ББнын структурасын долбоорлоо каражаттары №10. Лабораториялык жумуш. Жыйынтык функциялар эсептелүүчү мамычалар жана аларды көрсөтүү №10. СӨАИ. «Мугалим» кызматындагы кызматкерлердин эмгек акысынын суммасын жана алардын орто жашын эсептөө	2	1	6	интерак. метод, ДБТ, КОП, ЧТ	гlossарий, “Висели-ца” оюну, тест.	
	2-учурдагы текшерүү						
	Бардыгы	20	16	54			

3.9. Дисциплинанын программасы

MySQL’дин структурасы. Клиент-сервердик архитектура. Берилгендер базасын жана таблицаларды түзүү. Берилгендерди тандап (ылгап) алуу - SELECT оператору. Камтылган запростор. Таблицаларды бириктирүү (ички жана сырткы биригүү). Берилгендерди редактирлөө, жаңылоо жана жок кылуу. Математикалык функциялардын колдонулушу.

3.10. Дисциплинанын окуу-методикалык жана маалыматтык камсыздалышы

3.10.1. Негизги адабияттар

1. М. Кузнецов, И. Симдянов MySQL5. – Издат.: БХВ-Петербург, 2016. – 1007 с.

2. Гольцман В. MySQL5 – Издат.: Питер, 2017. – 253 с.

3.10.2. Кошумча адабияттар

3. Бен Форта Освой самостоятельно MySQL. – Издат.: Вильямс, 2015. – 287 с.

4. Грабер, Мартин SQL. Справочное руководство; М.: Лори; Издание 2-е, 2016. – 354 с

5. Шнайдер, Роберт Microsoft SQL Server 6.5. Проектирование высокопроизводитель-ных баз данных; М.: Лори, 2017. - 361 с.

б. Яргер, Р.Дж.; Риз, Дж.; Кинг, Т. MySQL и mSQL: Базы данных для небольших предприятий и Интернета; СПб: Символ-Плюс, 2018. - 560 с.

3.11. Курстун саясаты

Магистранттерге коюлуучу талаптар:

- а) сабактарга сөзсүз катышуу;
- б) практикалык (семинардык) сабактардагы активдүүлүгү;
- в) сабактарга, үй тапшырмасын жана өз алдынча иштерди аткарууга даярдыгы ж.б.
- г) калтырган сабактардын конспектисин өз алдынча даярдап келип, окутуучуга баяндап бериши.

Төмөнкүлөргө жол берилбейт:

- а) сабактарга кечигүү жана сабактан кетип калуу;
- б) сабак учурунда уюлдук телефонду пайдалануу;
- в) жалганчылык жана көчүрүп алуу (плагиат);
- г) тапшырмаларды өз убагында тапшырбоо ж.б.

3.12. Калтырылган сабакты толуктоо (отработка)

Магистрант калтырылган сабакта өтүлгөн теманы өз алдынча өздөштүрүп, деканаттын уруксат кагазы менен кафедрага келип, предметникке (предметник талап кылган формада) теманы кайрадан тапшырат. Калтырылган сабакты толуктоо аралык текшерүүгө чейин кабыл алынат жана модулдук баллдарга таасир этет. Калтырылган сабак толукталбаса, ар бир калтырылган сабак үчүн предметник модулдан 2 балл кемитет. Предметник кайра тапшырууну атайын журналга каттап, деканаттын уруксат кагазына «калтырылган сабак толукталды» деген белгини коюп берет.

3.13. Магистранттардын өз алдынча иштери үчүн тапшырмалар (МӨАИ)

№1. ӨАИ. SQL тилинин келип чыгышы жана колдонулушу

Өз алдынча иштин планы:

- 1) SQL тили жөнүндө маалыматтар;
- 2) SQL тилинин MySQL'де колдонулушу.

Адабияттар: негизги: [1, 2]; кошумча: [3]

№2. ӨАИ. Linux операциондук системасында берилгендер базасын түзүү

Өз алдынча иштин планы:

- 1) Linux операциондук системасында MySQL'ди орнотуу;
- 2) Linux операциондук системасында берилгендер базасын түзүү.

Адабияттар: негизги: [1, 2]; кошумча: [3]

№3. ӨАИ. FLOAT, BLOB, TEXT, ENUM жана SET типтеринин өзгөчөлүктөрү

Өз алдынча иштин планы:

- 1) FLOAT, BLOB, TEXT, ENUM жана SET типтери жөнүндө түшүнүк;
- 2) MySQL' де алардын колдонуштары.

Адабияттар: негизги: [1, 2]; кошумча: [4]

№4. ӨАИ. IN жана BETWEEN операторлорун колдонуп, 25 жаштан жогорку кызматкерлердин тизмесин түзгүлө**Өз алдынча иштин планы:**

- 1) WHERE ачыкч сөзүнүн IN жана BETWEEN маанилерин колдонуу;
- 2) 25 жаштан жогорку кызматкерлердин тизмесин түзүү.

Адабияттар: негизги: [1, 2]; кошумча: [3, 4]

№5. ӨАИ. Тексттик берилгендерди шаблондор боюнча изилдөө**Өз алдынча иштин планы:**

1. Тексттик берилгендерди шаблондор боюнча изилдөө;
2. Шаблондордун негизинде маалыматтарды табуу.

Адабияттар: негизги: [2]; кошумча: [4]

№6. ӨАИ. Таблицааларды оперативдик эске жайгаштыруу**Өз алдынча иштин планы:**

1. Таблицааларды оперативдик эске жайгаштыруу;
2. Шаблондордун негизинде маалыматтарды табуу.

Билимди текшерүүнүн формалары: демонстрация

Тапшыруу мөөнөтү: 7-жума

Адабияттар: негизги: [1]; кошумча: [4,6]

№7. ӨАИ. 30 жаштан төмөнкү фамилиясы К жана А тамгаларынан башталбаган кызматкерлердин атынын жана жашынын тизмесин түзүү**Өз алдынча иштин планы:**

- 1) 30 жаштан төмөнкү фамилиясы К жана А тамгаларынан башталбаган кызматкерлердин атынын жана жашынын тизмесин түзүү.
- 2) Маселеде SELECT операторунун колдонулушу.

Адабияттар: негизги: [1]; кошумча: [4,6]

№8. ӨАИ. DISTINCT ачыкч сөзүн пайдаланып, компаниянын кызматкерлеринин кызмат акыларын өсүү тартибинде түзүү**Өз алдынча иштин планы:**

- 1) DISTINCT ачыкч сөзүнүн мааниси жана анын колдонулушу;
- 2) DISTINCT ачыкч сөзүн пайдаланып, компаниянын кызматкерлеринин кызмат акыларын өсүү тартибинде түзүү.

Адабияттар: негизги: [1]; кошумча: [3,4]

№9. ӨАИ. HAVING тиркемесинин жардамында түрдүү кызматтагы кызматкерлердин орточо эмгек акысын чыгаруучу программа түзүү**Өз алдынча иштин планы:**

- 1) HAVING тиркемесинин колдонулушу;
- 2) HAVING тиркемесинин жардамында түрдүү кызматтагы кызматкерлердин орточо эмгек акысын чыгаруучу программа түзүү.

Адабияттар: негизги: [1, 2]; кошумча: [3,6]

№10. ӨАИ. «Мугалим» кызматындагы кызматкерлердин эмгек акысынын суммасын жана алардын орто жашын эсептөө

Өз алдынча иштин планы:

- 1) Убакыт жана дата функциялар жөнүндө түшүнүк жана колдонулушу;
- 2) «Мугалим» кызматындагы кызматкерлердин эмгек акысынын суммасын жана алардын орто жашын эсептөө

Адабияттар: негизги: [1, 2]; кошумча: [3,4].

3.14. Упайларды коюу саясаты

Учурдагы, аралыктагы жана жыйынтыктоочу текшерүүлөр «Билимди баалоо» жөнүндөгү жобо менен аныкталат.

Магистртин билим деңгээли 30 жана 100 баллдык системаларда төмөнкү эрежеге ылайык коюлат:

30 баллдык система	Баллдар (Рейтинг)	Тамгалык система боюнча баа	GPA боюнча баалоонун цифралык эквиваленти	Традициялык система боюнча баа
26 - 30	87 – 100	A	4,0	Эң жакшы
24 - 25	80 – 86	B	3,33	Жакшы
22 - 23	74 – 79	C	3,0	
20 - 21	68 – 73	D	2,33	Канааттандыраарлык
18 - 19	61 – 67	E	2,0	
9 - 17	31 -60	FX	0	Канааттандыраарлык эмес
0 - 8	0 - 30	F	0	


Экзаменде бааны коюуда объективдүүлүк жана акыйкаттуулук принциптеринин негизинде магистртин билиминин сапаты бардык тараптан анализделип, модулдук-рейтингдик системанын жобосуна ылайык коюлат.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
КЫРГЫЗСКОЙ РЕСПУБЛИКИ

ОШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА АСЦТ

«Утверждена»
на заседании кафедры АСЦТ
от 28 августа 2020 года, протокол №1
Зав. каф. АСЦТ, доцент:  Молдоярров У.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

База данных

НАПРАВЛЕНИЕ: 710100 «ИВТ»

КВАЛИФИКАЦИЯ (СТЕПЕНЬ) ВЫПУСКА - магистр

ПАСПОРТ

фонда оценочных средств по дисциплине «База данных»

1. Ожидаемые результаты и компетенции

Процесс изучения дисциплины «БД» направлен на формирования следующих **универсальных** компетенций:

общенаучными (ОК):

✓ способен глубоко понимать и критически оценивать новейшие теории, методы и способы, использовать междисциплинарный подход и интегрировать достижения различных наук для приобретения новых знаний (ОК-1);

- **инструментальными (ИК):**

способен к дальнейшему самостоятельному обучению (ИК-4);

- **социально-личностными и общекультурными (СЛК):**

- способен руководить коллективом, в том числе междисциплинарными проектами, влиять на формирование целей команды, воздействовать на ее социально-психологический климат в нужном для достижения целей направлении, корректно оценивать качество результатов деятельности (СЛК-4);

профессиональными компетенциями (ПК):

- ✓ способен анализировать и систематизировать научно-техническую информацию по теме исследования, выбирать методики и средства решения задач (ПК-2);
- ✓ способен к постановке и модернизации отдельных лабораторных работ и практикумов по дисциплинам направления (ПК-6);
- ✓ способен готовить технических заданий на разработку проектных решений (ПК-8);
- ✓ способен проектировать и применять инструментальные средства реализации программно-аппаратных проектов (ПК-13).

Перечень оценочных средств

№	Вид деятельности	Определения	Примечание
1.	Парная работа	<i>Парная работа</i> – взаимодействие участников учебного процесса в парах сменного состава («обучая — учусь»)	Используется в различных формах деятельности
2.	Групповая работа	<i>Групповая работа</i> – это взаимодействие в малых группах, где обсуждение происходит с каждым и решения принимаются сообща	Используется в различных формах деятельности
3.	Работа с контрольными карточками	<i>Контрольные карточки</i> могут содержать вопросы, оформленные в виде тестовых заданий, вопросов соответствий, контрольных заданий и др.	
4.	Исследовательская работа	Исследовательская работа – продукт самостоятельной работы студента, представляющий собой решение определенной учебно-практической научной темы	
5.	Элективный (выбросный) тест	Тест – система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	
6.	Выполнение практических и лабораторных заданий	Практическое (лабораторное) задание – задание задаваемое преподавателем студенту для самостоятельного выполнения, в целях усвоения пройденного материала	Выполняется в тетрадях
7.	Конспектирование	<i>Конспект</i> (лат. conspectus — обозрение, обзор, очерк) — краткое изложение содержания нового материала	в тетради
8.	Выполнение реферата	<i>Реферат</i> – продукт самостоятельной работы студента по определённой теме, в котором собрана информация из одного или нескольких источников, представляющий собой краткое изложение в письменном виде.	На листах формата А4, объемом 2-15 листа

9.	Авторизация системы	В <u>информационных технологиях</u> посредством <i>авторизации</i> устанавливаются права доступа к информационным ресурсам и системам обработки данных.	
10.	Установка программного обеспечения;	<i>Установка программного обеспечения, инсталляция</i> — процесс установки программного обеспечения (Pascal ABC) на компьютер конечного пользователя	
11.	Презентация проектной работы	<i>Презентация</i> , на основе программного обеспечения, для демонстрации конечного продукта, сопровождается устным пояснением	Представление прак-го рез-та. на основе пройденных материалов
12.	Графические (визуальные) организаторы	Генерирование, визуализация, структурирование идей в процессе изучения и организации информации, разрешения проблем, принятия решений	Диаграммы Венна, Т-диаграммы, графические органайзеры сравнения и сопоставления и т.д.
13.	Подготовка портфолио	Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах.	используется весь семестр
14.	Дополни предложение	Дополняя предложение, учащиеся активизируют предварительные знания	
15.	Составление кластера	<i>Кластер</i> представляется, как подмножество результатов поиска, связанных единством темы	понятие используется в итоговых занятиях

Критерии оценивания

№	Вид деятельности	Критерии оценивания	Баллы
1.	Дополни предложение	<ol style="list-style-type: none"> 1. Дополняя предложение, учащиеся активизируют предварительные знания 2. Повторение освоенных понятий и терминов, освоение новых 3. Воспроизведение 4. Продумайте ключевые слова по теме. Это должно быть слово, которое может стать подлежащим в предложении и значение которого известно учащимся, то есть это не должно быть абстрактным понятием. 	<ul style="list-style-type: none"> • <i>Полный объем работы: 5 б;</i> • <i>Неполный: 3 б;</i> • <i>Минимальный: 1 б</i>

2.	Графические организаторы для активизации: мишень практика	<ol style="list-style-type: none"> 1. Учащиеся заполняют мишень идеями и именами людьми, которые относятся к теме урока 2. Запоминание и понимание 3. Письмо (короткие записи) и говорение (неформальное) 4. Раздайте каждому учащемуся рисунок мишени 	в практических уроках используется
3.	Составление кластера	<ol style="list-style-type: none"> 1. Приобретение новых знаний с большой степенью самостоятельности; 2. Восприятие, обобщение и анализ итоговой информации; 3. Выделение и объединение однородных элементов, связанных единством темы; 4. Грамотное составление наглядной схемы; 5. Использование различных методов построений схем 6. Обоснование принимаемых решений 	<ul style="list-style-type: none"> • <i>Полный объем работы: 5 б;</i> • <i>Неполный: 3 б;</i> • <i>Минимальный: 1 б</i>

4.1. Критерии оценки на экзамене

Выставление оценок на экзаменах осуществляется на основе принципов объективности, справедливости, всестороннего анализа качества знаний студентов, и других положений, способствующих повышению надежности оценки знаний обучающихся и устранению субъективных факторов.

В соответствии с действующими нормативными актами и рекомендациями Министерства образования и науки КР устанавливаются следующие критерии выставления оценок на экзаменах по гуманитарным, естественным, техническим и другим дисциплинам:

- оценка **"отлично"** выставляется студенту, который обнаружил на экзамене всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, который усвоил основную литературу и ознакомился с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины и их значений для приобретаемой профессии,

проявившим творческие способности в понимании, изложении и использовании учебно-программного материала;

- оценка **"хорошо"** выставляется студенту, который на экзамене обнаружил полное знание учебно-программного материала, успешно выполнил предусмотренные в программе задания, усвоил основную литературу, рекомендованную в программе. Как правило, оценка "хорошо" выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному выполнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности;

- оценка **"удовлетворительно"** выставляется студенту, обнаружившему знание основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, предусмотренных программой, который ознакомился с основной литературой, рекомендованной программой. Как правило, оценка "удовлетворительно" выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя;

- оценка **"неудовлетворительно"** выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий, не ознакомившемуся с основной литературой, предусмотренной программой, и не овладевшему базовыми знаниями, предусмотренными по данной дисциплине и определенными соответствующей программой курса (перечень основных знаний и умений, которыми должны овладеть магистранты, является обязательным элементом рабочей программы курса).

Лекция 1. Основные понятия теории баз данных

План:

1. Основные понятия теории баз данных
2. История возникновения баз данных
3. История развития баз данных
4. Классификация БД

Компьютеры были созданы для решения вычислительных задач, однако со временем они все чаще стали использоваться для построения систем обработки документов, а точнее, содержащейся в них информации. Такие системы обычно и называют информационными. Информационные системы имеют следующие особенности:

- для обеспечения их работы нужны сравнительно низкие вычислительные мощности
- данные, которые они используют, имеют сложную структуру
- необходимы средства сохранения данных между последовательными запусками системы, ...

Другими словами, информационная система требует создания в памяти ЭВМ *динамически обновляемой* модели внешнего мира с использованием единого хранилища - **базы данных (БД)**.

Предметная область - часть реального мира, подлежащая изучению с целью организации управления и, в конечном счете, автоматизации.

Отличительной чертой баз данных следует считать то, что данные хранятся совместно с их описанием, а в прикладных программах описание данных не содержится. Независимые от программ пользователя данные обычно называются **метаданными**. В ряде современных систем метаданные, содержащие также информацию о пользователях, форматы отображения, статистику обращения к данным и др. сведения, *хранятся в словаре базы данных*.

Таким образом, **система управления базой данных (СУБД)** - важнейший компонент информационной системы. Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор.

Основные функции СУБД:

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти;
- журнализация изменений и восстановление базы данных после сбоев;
- поддержание языков БД (язык определения данных, язык манипулирования данными).

История возникновения БД

В истории вычислительной техники можно проследить развитие двух основных областей ее использования.

Первая область — применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную. Характерной особенностью данной области применения вычислительной техники является наличие сложных алгоритмов обработки, которые применяются к простым по структуре данным, объем которых сравнительно невелик.

Вторая область — это использование средств вычислительной техники в автоматических или автоматизированных информационных системах.

Информационная система представляет собой программно-аппаратный комплекс, обеспечивающий выполнение следующих функций:

1. надежное хранение информации в памяти компьютера;

2. выполнение специфических для данного приложения преобразований информации и вычислений;
3. предоставление пользователям удобного и легко осваиваемого интерфейса.

Обычно такие системы имеют дело с большими объемами информации, имеющей достаточно сложную структуру.

Важным шагом в развитии именно информационных систем явился переход к использованию централизованных систем управления файлами. С точки зрения прикладной программы, файл — это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Правила именования файлов, способ доступа к данным, хранящимся в файле, и структура этих данных зависят от конкретной системы управления файлами и, возможно, от типа файла. Система управления файлами берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным.

Пользователи видят файл как линейную последовательность записей и могут выполнить над ним ряд стандартных операций:

- создать файл (требуемого типа и размера);
- открыть ранее созданный файл;
- прочитать из файла некоторую запись (текущую, следующую, предыдущую, первую, последнюю);
- записать в файл на место текущей записи новую, добавить новую запись в конец файла.

В разных файловых системах эти операции могли несколько отличаться, но общий смысл их был именно таким.

Главное, что следует отметить, это то, что **структура записи файла была известна только программе, которая с ним работала**, система управления файлами не знала ее. И поэтому для того, чтобы извлечь некоторую информацию из файла, необходимо было точно знать структуру записи файла с точностью до бита. Каждая программа, работающая с файлом, должна была иметь у себя внутри структуру данных, соответствующую структуре этого файла. Поэтому **при изменении структуры файла требовалось изменять структуру программы**, а это требовало новой компиляции, то есть процесса перевода программы в исполняемые машинные коды. Такая ситуация характеризовалась как зависимость программ от данных. Для информационных систем характерным является наличие большого числа различных пользователей (программ), каждый из которых имеет свои специфические алгоритмы обработки информации, хранящейся в одних и тех же файлах. **Изменение структуры файла, которое было необходимо для одной программы, требовало исправления и перекомпиляции и дополнительной отладки всех остальных программ, работающих с этим же файлом.** Это было первым существенным недостатком файловых систем, который явился толчком к созданию новых систем хранения и управления информацией. Поскольку файловые системы являются общим хранилищем файлов, принадлежащих, вообще говоря, разным пользователям, системы управления файлами должны обеспечивать авторизацию доступа к файлам. В общем виде подход состоит в том, что **по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю.** И отсутствие централизованных методов управления доступом к информации послужило еще одной причиной разработки СУБД. Следующей причиной стала **необходимость обеспечения эффективной параллельной работы многих пользователей с одними и теми же файлами.** В общем случае системы управления файлами обеспечивали режим многопользовательского доступа. Если операционная система поддерживает многопользовательский режим, вполне реальна ситуация, когда два или более

пользователя одновременно пытаются работать с одним и тем же файлом. Если все пользователи собираются только читать файл, ничего страшного не произойдет. Но если хотя бы один из них будет изменять файл, для корректной работы этих пользователей требуется взаимная синхронизация их действий по отношению к файлу.

В системах управления файлами обычно применялся следующий подход. В операции открытия файла (первой и обязательной операции, с которой должен начинаться сеанс работы с файлом) среди прочих параметров указывался режим работы (чтение или изменение). Если к моменту выполнения этой операции некоторым пользовательским процессом PR1 файл был уже открыт другим процессом PR2 в режиме изменения, то в зависимости от особенностей системы процессу PR1 либо сообщалось о невозможности открытия файла, либо он блокировался до тех пор, пока в процессе PR2 не выполнялась операция закрытия файла. При подобном способе организации одновременная работа нескольких пользователей, связанная с модификацией данных в файле, либо вообще не реализовывалась, либо была очень замедлена.

Эти недостатки послужили тем толчком, который заставил разработчиков информационных систем предложить новый подход к управлению информацией. Этот подход был реализован в рамках новых программных систем, названных впоследствии Системами Управления Базами Данных (СУБД), а сами хранилища информации, которые работали под управлением данных систем, назывались базами или банками данных (БД и БнД).

История развития БД

Концепция БД сложилась в конце 60-х годов прошлого столетия и с тех пор постоянно развивалась.

Первый этап сложился к началу 60-х годов прошлого века и характеризуется следующими признаками:

- информация преимущественно хранится в последовательных файлах на магнитных лентах;
- физическая структура данных строго соответствует логической;
- в качестве архива хранятся несколько копий файлов;
- файлы предназначены для единственной программы;
- программист планирует не только логическую, но и физическую организацию данных;
- при изменении физической или логической организации данных программа должна перерабатываться.

Второй этап относится к середине 60-х годов и имеет следующие особенности:

- появились внешние устройства прямого доступа, позволившие осуществить произвольный доступ к записям (прямой, индексно-последовательный);
- вошли в употребление процедуры поиска записи по ключевому полю (обычно одному);
- стало возможным переносить файлы на другие внешние устройства без изменения прикладных программ, что обычно обеспечивалось средствами языка управления данными соответствующей операционной системы.

Третий этап начался с конца 60-х годов. Основным достижением можно считать осознание необходимости централизации данных для доступа к ним различных приложений. При этом уменьшается избыточность и противоречивость информации, приложения используют стандартные средства доступа к данным. На этом этапе возросла сложность организации данных, был реализован эффективный поиск записей по многим ключам.

Именно на этом этапе появились первые СУБД. Прежде всего развивались теория и практика построения иерархических и сетевых СУБД. В этих моделях связи данных описываются с помощью деревьев и графов общего вида.

Четвертый этап датируется второй половиной 70-х годов. На этом этапе были реализованы следующие основные характеристики СУБД:

- логическая и физическая независимость данных;
- удобство развития БД;
- безопасность, секретность, целостность данных;
- поиск информации по различным запросам;
- языковые средства для администратора, прикладного программиста, пользователя-непрофессионала.

С начала 70-х годов после публикаций Э. Кодда начались активные исследования реляционной модели данных. Основу реляционной СУБД составляют таблицы. Вплоть до 80-х годов реляционные СУБД считались перспективными, но трудными для реализации.

Новый этап в развитии СУБД наступил при появлении персональных компьютеров. На этом этапе на передний план вышли такие особенности СУБД, как:

- дружелюбность и удобство работы пользователя (развитые диалоги, меню, оконный интерфейс, контекстная помощь);
- упрощение громоздких схем СУБД за счет частичной реализации ряда свойств;
- почти полный переход на реляционные СУБД;
- ориентация не только на программиста, но и на пользователя-непрофессионала;
- наличие средств автоматизации программирования в виде генераторов форм, меню, отчетов, запросов.

Классификация БД

Классификация БД может быть произведена по различным признакам, среди которых выделяют:

1. *По форме представления информации:* фактографические и документальные.
2. *По типу используемой модели данных:* иерархические, сетевые, реляционные.
3. *По типологии хранения данных:* локальные (централизованные) и распределённые (удалённые) БД.

Классификация не является полной. Различные источники предоставляют разнообразную классификацию.

Вопросы для самоконтроля:

1. Дайте определения понятиям: информационная система, предметная область.
2. Что называется базой данных и каково ее место в ИС?
3. В чем различие между данными и метаданными?
4. Каково назначение систем управления базами данных?
5. Для чего используется словарь данных?
6. Назовите этапы развития БД.
7. Какую роль в развитии технологии БД сыграло появление ПК?
8. Каковы функции СУБД?

Лекция 2-3. Технологии работы с базами данных

План:

1. Централизованная архитектура

2. Архитектура "файл-сервер"
3. Технология "клиент – сервер"
4. Трехзвенная (многозвенная) архитектура "клиент – сервер"

Понятие базы данных изначально предполагало возможность решения многих задач несколькими пользователями. В связи с этим, важнейшей характеристикой современных СУБД является наличие многопользовательской технологии работы. Разная реализация таких технологий в разное время была связана как с основными свойствами вычислительной техники, так и с развитием программного обеспечения.

Централизованная архитектура

При использовании этой технологии база данных, СУБД и прикладная программа (приложение) располагаются на одном компьютере (рисунок 1). Для такого способа организации не требуется поддержки сети и все сводится к автономной работе.

Рисунок 1 - Централизованная архитектура

Многопользовательская технология работы обеспечивалась либо режимом мультипрограммирования, либо режимом разделения времени.

Архитектура "файл-сервер"

Увеличение сложности задач, появление персональных компьютеров и локальных вычислительных сетей явились предпосылками появления новой архитектуры *файл-сервер*. Эта архитектура баз данных с сетевым доступом предполагает назначение одного из компьютеров сети в качестве выделенного сервера, на котором будут храниться файлы базы данных. В соответствии с запросами пользователей файлы с *файл-сервера* передаются на рабочие станции пользователей, где и осуществляется основная часть обработки данных. Центральный сервер выполняет в основном только роль хранилища файлов, не участвуя в обработке самих данных (рисунок 2). **Технология "клиент – сервер"**

Использование технологии "*клиент – сервер*" предполагает наличие некоторого количества компьютеров, объединенных в сеть, один из которых выполняет особые управляющие функции (является сервером сети).

Так, архитектура "*клиент – сервер*" разделяет функции приложения пользователя (называемого клиентом) и сервера. Приложение-клиент формирует запрос к серверу, на котором расположена БД, на структурном языке запросов SQL (Structured Query Language), являющемся промышленным стандартом в мире реляционных БД. Удаленный сервер принимает запрос и переадресует его SQL-серверу БД.

SQL-сервер – специальная программа, управляющая удаленной базой данных. SQL-сервер обеспечивает интерпретацию запроса, его выполнение в базе данных, формирование результата выполнения запроса и выдачу его приложению-клиенту. При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса; клиентский компьютер лишь отправляет запрос к серверной БД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю. Так как клиентскому приложению посылается результат выполнения запроса, по сети "путешествуют" только те данные, которые необходимы клиенту. В итоге снижается нагрузка на сеть. Поскольку выполнение запроса происходит там же, где хранятся данные (на сервере), нет необходимости в пересылке больших пакетов данных. Кроме того, SQL-сервер, если это возможно, оптимизирует полученный запрос таким образом, чтобы он был выполнен в минимальное время с наименьшими накладными расходами.

Архитектура системы представлена на рисунке 3.

Рисунок 3 - Архитектура "клиент – сервер" **Трехзвенная (многозвенная) архитектура "клиент – сервер"**

Трехзвенная (в некоторых случаях *многозвенная*) **архитектура** (N-tier или multi-tier). представляет собой дальнейшее совершенствование технологии "*клиент – сервер*".

Рассмотрев архитектуру " клиент – сервер ", можно заключить, что она является 2-звенной: первое звено – клиентское приложение, второе звено – сервер БД + сама БД. В *трехзвенной архитектуре* вся бизнес-логика (деловая логика), ранее входившая в клиентские приложения, выделяется в отдельное звено, называемое сервером приложений. При этом клиентским приложениям остается лишь пользовательский интерфейс.

Схематически такую *архитектуру* можно представить, как показано на рисунке 4.

Рисунок 4 - Представление многоуровневой архитектуры "клиент-сервер"

Вопросы для самоконтроля:

1. Назовите достоинства и недостатки существующих многопользовательских технологий с базами данных.

Лекция 4. Логическая и физическая независимость данных

План:

1. Базовые понятия
2. Архитектура базы данных
3. Механизм прохождения запроса к БД

Современные авторы часто употребляют термины "банк данных" и "база данных" как синонимы, однако в общеотраслевых руководящих материалах по созданию банков данных Государственного комитета по науке и технике (ГКНТ), изданных в 1982 г., эти понятия различаются. Там приводятся следующие определения банка данных, базы данных и СУБД:

Банк данных (БНД) — это система специальным образом организованных данных — баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

База данных (БД) — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Система управления базами данных (СУБД) — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Программы, с помощью которых пользователи работают с базой данных, называются *приложениями*. В общем случае с одной базой данных могут работать множество различных приложений. При рассмотрении приложений, работающих с одной базой данных, предполагается, что они могут работать параллельно и независимо друг от друга, и именно СУБД призвана обеспечить работу множества приложений с единой базой данных таким образом, чтобы каждое из них выполнялось корректно, но учитывало все изменения в базе данных, вносимые другими приложениями.

Архитектура базы данных

В процессе научных исследований, посвященных тому, как именно должна быть устроена СУБД, предлагались различные способы реализации.

Самым жизнеспособным из них оказалась предложенная американским комитетом по стандартизации ANSI (American National Standards Institute) трехуровневая система организации БД, изображенная на рисунке 5. Уровень внешних моделей — самый верхний уровень, где каждая модель имеет свое "видение" данных. Этот уровень определяет точку зрения на БД отдельных приложений. Каждое приложение видит и обрабатывает только те данные, которые необходимы именно этому приложению.

Концептуальный уровень — центральное управляющее звено, здесь база данных представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями, работающими с данной базой данных. Фактически концептуальный

уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась база данных. Как любая модель, концептуальная модель отражает только существенные, с точки зрения обработки, особенности объектов реального мира.

Физический уровень — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.

Эта архитектура позволяет обеспечить логическую (между уровнями 1 и 2) и физическую (между уровнями 2 и 3) независимость при работе с данными.

Логическая независимость предполагает возможность изменения одного приложения без корректировки других приложений, работающих с этой же базой данных.

Физическая независимость предполагает возможность переноса хранимой информации с одних носителей на другие при сохранении работоспособности всех приложений, работающих с данной базой данных. **Процесс прохождения пользовательского запроса**

Рисунок 6 иллюстрирует взаимодействие пользователя, СУБД и ОС при обработке запроса на получение данных. Цифрами помечена последовательность взаимодействий:

Рисунок 6 - Схема прохождения запроса к БД

Разумеется, механизм прохождения запроса в реальных СУБД гораздо сложнее, но и эта упрощенная схема показывает, насколько серьезными и сложными должны быть механизмы обработки запросов, поддерживаемые реальными СУБД.

Вопросы для самоконтроля:

1. Каким образом прикладные программы взаимодействуют с БД?
2. Чем банк данных отличается от базы данных?
3. Какие компоненты входят в состав банка данных?
4. Что представляет собой трехуровневая архитектура СУБД?
5. В чем особенность уровня внешних моделей?
6. В чем особенность концептуального уровня?
7. В чем особенность физического уровня?
8. Что означает логическая и физическая независимость данных?

Лекция 5. Типы моделей данных. Реляционная модель данных

План:

1. Иерархическая модель базы данных
2. Сетевая модель базы данных
3. Реляционная модель базы данных

Различают три основные модели базы данных - это иерархическая, сетевая и реляционная. Эти модели отличаются между собой по способу установления связей между данными.

Иерархическая модель базы данных

Иерархические базы данных - самая ранняя модель представления сложной структуры данных. Информация в иерархической базе организована по принципу древовидной структуры, в виде отношений "предок-потомок".

Каждая запись может иметь не более одной родительской записи и несколько подчиненных. Связи записей реализуются в виде физических указателей с одной записи на другую. Основным недостатком иерархической структуры базы данных - невозможность реализовать отношения "многие-ко-многим", а также ситуации, когда запись имеет несколько предков.

Иерархические базы данных графически могут быть представлены как перевернутое дерево, состоящее из объектов различных уровней. Верхний уровень (корень дерева) занимает один объект, второй - объекты второго уровня и так далее.

Организация данных в СУБД иерархического типа определяется в терминах:

элемент, агрегат, запись (группа), групповое отношение, база данных.

Атрибут (элемент данных) - наименьшая единица структуры данных. Обычно каждому элементу при описании базы данных присваивается уникальное имя. По этому имени к нему обращаются при обработке. Элемент данных также часто называют полем.

Запись - именованная совокупность атрибутов. Использование записей позволяет за одно обращение к базе получить некоторую логически связанную совокупность данных. Именно записи изменяются, добавляются и удаляются. Тип записи определяется составом ее атрибутов. Экземпляр записи - конкретная запись с конкретным значением элементов.

Групповое отношение - иерархическое отношение между записями двух типов.

Родительская запись (владелец группового отношения)

называется исходной записью, а дочерние записи (члены группового отношения) -

подчиненными. Иерархическая база данных может хранить только такие древовидные структуры.

Корневая запись каждого дерева обязательно должна содержать ключ с уникальным значением. Ключи некорневых записей должны иметь уникальное значение только в рамках группового отношения. Каждая запись идентифицируется полным сцепленным ключом, под которым понимается совокупность ключей всех записей от корневой, по иерархическому пути.

Для групповых отношений в иерархической модели обеспечивается автоматический режим включения и фиксированное членство. Это означает, что для запоминания любой некорневой записи в БД должна существовать ее родительская запись.

Операции над данными, определенные в иерархической модели:

Добавить в базу данных новую запись. Для корневой записи обязательно формирование значения ключа.

Изменить значение данных предварительно извлеченной записи. Ключевые данные не должны подвергаться изменениям.

Удалить некоторую запись и все подчиненные ей записи.

Извлечь корневую запись по ключевому значению, допускается также последовательный просмотр корневых записей.

Извлечь следующую запись (следующая запись извлекается в порядке левостороннего обхода дерева).

В операции **ИЗВЛЕЧЬ** допускается задание условий выборки.

Все операции изменения применяются только к одной "текущей" записи (которая предварительно извлечена из базы данных). Такой подход к манипулированию данными получил название "навигационного".

В иерархической БД поддерживается только целостность связей между владельцами и членами группового отношения (никакой потомок не может существовать без предка).

Сетевая модель базы данных

Сетевая модель данных определяется в тех же терминах, что и *иерархическая*. Она состоит из множества записей, которые могут быть владельцами или членами групповых отношений. Связь между записью-владельцем и записью-членом также имеет вид 1:N.

Основное различие этих моделей состоит в том, что в сетевой модели запись может быть членом более чем одного группового отношения. 3

Операции над данными в сетевой модели БД

- Добавить** - внести запись в БД и, в зависимости от режима включения, либо включить ее в групповое отношение, где она объявлена подчиненной, либо не включать ни в какое групповое отношение.
- Включить в групповое отношение** - связать существующую подчиненную запись с записью-владельцем.
- Переключить** - связать существующую подчиненную запись с другой записью-владельцем в том же групповом отношении.
- Обновить** - изменить значение элементов предварительно извлеченной записи.
- Извлечь** - извлечь записи последовательно по значению ключа, а также используя групповые отношения - от владельца можно перейти к записям - членам, а от подчиненной записи к владельцу набора.
- Удалить** - убрать из БД запись. Если эта запись является владельцем группового отношения, то анализируется класс членства подчиненных записей. Обязательные члены должны быть предварительно исключены из группового отношения, фиксированные удалены вместе с владельцем, необязательные останутся в БД.
- Исключить из группового отношения** - разорвать связь между записью-владельцем и записью-членом.

В сетевой модели обеспечивается только поддержание целостности по ссылкам (владелец отношения - член отношения).

Реляционная модель базы данных

Реляционная модель была предложена в 1969 году сотрудником фирмы IBM Е. Ф. Коддом (Dr. E. F. Codd), известным исследователем в области баз данных. Впервые основные концепции этой модели были опубликованы в 1970 году.

Набор средств для управления реляционными базами данных называется реляционной системой управления базами данных (РСУБД). Реляционная система управления базами данных может содержать утилиты, приложения, сервисы, библиотеки, средства создания приложений и другие компоненты.

Реляционная база данных представляет собой совокупность двумерных таблиц. Любая таблица реляционной базы данных состоит из строк, называемых *записями*, и столбцов, называемых *полями*. Строки таблицы содержат сведения об объектах. Каждый столбец в таблице должен содержать только определенный тип информации. Каждая *строка* таблицы содержит разнообразную (разного типа) информацию. Все данные, помещенные в одной строке, называют *записью*, каждый *элемент записи* - это *поле*. Таким образом, каждое поле содержит часть информации, находящейся на пересечении соответствующей строки и столбца. В таблице всевозможные значения одного типа в одном столбце называют *доменом*. Поле является элементом записи и представляет собой ячейку таблицы. У каждого столбца есть свое неповторимое имя, описывающее тот вид информации, который содержится в нем. Это имя называют *«именем поля базы данных»*. Каждое поле имеет фиксированную длину, следовательно, и любая запись в таблице имеет фиксированную длину. Каждая запись характеризуется своим уникальным порядковым номером.

Данные в реляционной таблице должны удовлетворять следующим принципам:

1. Каждое значение поля должно быть атомарным, т.е. не расчленимым на несколько значений;
2. Значения данных домена (в одном и том же столбце) должны принадлежать к одному и тому же типу данных, доступному для использования в данной СУБД;
3. Каждая запись в таблице уникальна, т.е. в таблице не существует двух записей с полностью совпадающим набором значений ее полей;
4. Каждое поле имеет уникальное имя;
5. Последовательность полей в таблице несущественна;

6. Последовательность записей в таблице несущественна.

Существенное отличие реляционной модели от обыкновенного последовательного файла заключается в том, что все столбцы в таблице с точки зрения входа предполагаются эквивалентными. Именно это свойство делает эту модель весьма мощной и делает невозможным отображение ее на память в виде последовательного массива данных.

Поскольку записи в таблице неупорядочены, то необходимо указать поле (или набор нескольких полей) для уникальной идентификации каждой записи.

Первичный ключ - это поле или набор полей, которые однозначно идентифицируют (определяют) запись таблицы

Обычно ключом является поле или совокупность полей фиксированной длины. Каждому значению *первичного* ключа соответствует одна и только одна запись. Первичный ключ любой таблицы обязан содержать уникальные непустые значения для каждой записи. Если первичный ключ состоит из нескольких полей, он называется *составным первичным ключом* (prime key).

Поле, указывающее на запись в другой таблице, связанную с данной записью, называется *внешним ключом* (foreign key).

Подобное взаимоотношение между таблицами называется *связью* (relationship). Связь между двумя таблицами устанавливается путем присвоения значений внешнего ключа одной таблицы значениям первичного ключа другой. Таблица, содержащая внешний ключ, называется *второстепенной*, а таблица, содержащая первичный ключ, определяющий возможные значения внешнего ключа второстепенной таблицы, называется *главной*.

Типичная реляционная база данных состоит из нескольких связанных таблиц. **Типы связей между объектами**

Все информационные объекты предметной области связаны между собой. Соответствия, отношения, возникающие между объектами предметной области, называются *связями*. Связанные отношениями таблицы взаимодействуют по принципу *главная, подчиненная*. Возможны следующие отношения между таблицами:

1. Отношение «один – ко – многим» (обозначают 1:M): одной записи из главной таблицы может соответствовать ноль, одна или несколько записей подчиненной таблицы.

2. Отношение «один – к – одному» (обозначают 1:1): одной записи из главной таблицы соответствует только одна запись из подчиненной таблицы.

3. Отношение «многие – ко – многим» (обозначают 1:1): одной записи из главной таблицы может соответствовать ноль, одна или несколько записей подчиненной таблицы и наоборот.

Одним из правил ссылочной целостности (referential integrity) является то, что первичный ключ любой таблицы должен содержать уникальные непустые значения для данной таблицы. Некоторые СУБД могут контролировать уникальность первичных ключей. Если СУБД контролирует уникальность первичных ключей, то при попытке присвоить первичному ключу значение, уже имеющееся в другой записи, СУБД сгенерирует диагностическое сообщение, обычно содержащее словосочетание primary key violation. Это сообщение в дальнейшем может быть передано в приложение, с помощью которого конечный пользователь манипулирует данными.

Если две таблицы связаны соотношением главная-подчиненная, внешний ключ подчиненной таблицы должен содержать только те значения, которые имеются среди значений первичного ключа главной таблицы. Если корректность значений внешних ключей не контролируется СУБД, можно говорить о нарушении ссылочной целостности. Если же СУБД контролирует корректность значений внешних ключей, то при попытке присвоить внешнему ключу значение, отсутствующее среди значений первичных ключей главной таблицы, либо при удалении или модификации записей главной таблицы,

приводящих к нарушению ссылочной целостности, СУБД сгенерирует сообщение, о котором говорилось выше.

Вопросы для самоконтроля:

1. Что такое модель данных?
2. Для чего строится модель данных?
3. Укажите достоинства и недостатки иерархической модели данных.
4. Как организуется физическое размещение данных в БД иерархического типа?
5. Охарактеризуйте сетевую модель данных.
6. Охарактеризуйте реляционную модель данных.
7. Чем отличается реляционная модель данных от предшествующих ей моделей?
8. Что такое простой ключ и составной ключ?
9. Перечислите виды связей между объектами? Охарактеризуйте их.
10. Как проявляется иерархическая подчиненность в связи «один ко многим»?

Лекция 6. Реляционная алгебра

План:

1. Традиционные операции реляционной алгебры
2. Специальные операции реляционной алгебры

С точки зрения внешнего представления объектов реального мира *модель данных* — это основные понятия и способы, используемые при анализе и описании предметной области. Среди многих попыток представить обработку данных на формальном абстрактном уровне реляционная модель, предложенная Э. Ф. Коддом, стала по существу первой работоспособной *моделью данных*, поскольку помимо средств описания объектов имела эффективный инструментарий преобразований этих описаний — операции реляционной алгебры.

Реляционная алгебра в том виде, в котором она была определена Э. Ф. Коддом, состоит из двух групп по четыре оператора.

1. Традиционные операции: объединение, пересечение, разность и декартово произведение.

2. Специальные реляционные операции: выборка, проекция, соединение, деление.

Объединение возвращает таблицу, содержащую все записи, которые принадлежат либо одной из двух заданных таблиц, либо им обоим.

Пересечение возвращает таблицу, содержащую все записи, которые принадлежат одновременно двум заданным таблицам.

Разность возвращает таблицу, содержащую все записи, которые принадлежат первому из двух заданных таблиц и не принадлежат второй.

Произведение возвращает таблицу, содержащую все возможные записи, которые являются сочетанием двух записей, принадлежащих соответственно двум заданным таблицам.

Выборка возвращает таблицу, содержащую все записи из заданной таблицы, которые удовлетворяют указанным условиям.

Проекция возвращает таблицу, содержащую все записи заданной таблицы, которые остались в этой таблицы после исключения из неё некоторых атрибутов

Соединение возвращает таблицу, содержащую все возможные записи, которые представляют собой комбинацию атрибутов двух записей, принадлежащих двум заданным таблицам, при условии, что в этих двух комбинированных записях присутствуют одинаковые значения в одном или нескольких общих для исходных таблиц атрибутах (причем эти общие значения в результирующей записи появляются один раз, а не дважды).

Деление для заданных двух унарных таблиц и одной бинарной возвращает таблицу, содержащую все записи из первой унарной таблицы, которые содержатся также в бинарной таблице и соответствуют всем записям во второй унарной таблице.

Результат выполнения любой операции над таблицами также является таблицей, поэтому результат одной операции может использоваться в качестве исходных данных для другой. Другими словами, можно записывать вложенные реляционные выражения, т. е. выражения, в которых операторы сами представлены реляционными выражениями, причем произвольной сложности. Эта особенность называется свойством *реляционной замкнутости*.

Вопросы для самоконтроля:

1. Сколько реляционных операций образуют реляционную алгебру?
2. Перечислите и охарактеризуйте операции реляционной алгебры. Приведите примеры.

Лекция 7. Основные этапы проектирования БД

План:

1. Жизненный цикл БД
2. Планирование разработки базы данных
3. Определение требований к системе
4. Сбор и анализ требований пользователей
5. Проектирование базы данных
6. Разработка приложений
7. Реализация
8. Загрузка данных
9. Тестирование
10. Эксплуатация и сопровождение

Жизненный цикл БД

Как и любой программный продукт, база данных обладает собственным жизненным циклом (ЖЦБД). Главной составляющей в жизненном цикле БД является создание единой базы данных и программ, необходимых для ее работы.

ЖЦБД включает в себя следующие основные этапы (рисунок 7):

Рисунок 7 - Жизненный цикл БД

Планирование разработки базы данных

Содержание данного этапа — разработка стратегического плана, в процессе которой осуществляется предварительное планирование конкретной системы управления базами данных.

Планирование разработки базы данных состоит в определении трех основных компонентов: объема работ, ресурсов и стоимости проекта.

Важной частью разработки стратегического плана является проверка осуществимости проекта, состоящая из нескольких частей.

Первая часть — проверка технологической осуществимости. Она состоит в выяснении вопроса, существует ли оборудование и программное обеспечение, удовлетворяющее информационным потребностям фирмы.

Вторая часть — проверка операционной осуществимости — выяснение наличия экспертов и персонала, необходимых для работы БД.

Третья часть — проверка экономической целесообразности осуществления проекта. При исследовании этой проблемы весьма важно дать оценку ряду факторов, в том числе и таким:

- целесообразность совместного использования данных разными

- отделами;
- величина риска, связанного с реализацией системы базы данных;
- ожидаемая выгода от внедрения подлежащих созданию приложений;
- время окупаемости внедренной БД;
- влияние системы управления БД на реализацию долгосрочных
- планов организации.

Определение требований к системе

На данном этапе необходимо определить диапазон действия приложения базы данных, состав его пользователей и области применения.

Определение требований включает выбор целей БД, выяснение информационных потребностей различных отделов и руководителей фирмы и требований к оборудованию и программному обеспечению.

Сбор и анализ требований пользователей

На данном этапе необходимо создать для себя модель движения важных материальных объектов и уяснить процесс документооборота. По каждому документу необходимо установить периодичность использования, определить данные, необходимые для выполнения выделенных функций (анализируя существующую и планируемую документацию, выясняют, как получается каждый элемент данных, кем получается, где в дальнейшем используется, кем контролируется.

Собранная информация о каждой важной области применения приложения и пользовательской группе должна включать следующие компоненты: исходную и генерируемую документацию, подробные сведения о выполняемых транзакциях, а также список требований с указанием их приоритетов.

Формализация собранной на этом этапе информации может быть повышена с помощью методов составления спецификаций требований, к числу которых относятся, например, технология структурного анализа и проектирования, диаграммы потоков данных и графики "вход — процесс — выход".

Проектирование базы данных

Полный цикл разработки базы данных включает концептуальное, логическое и физическое ее проектирование.

Концептуальное проектирование базы данных

Первая фаза процесса проектирования базы данных заключается в создании для анализируемой части предприятия *концептуальной модели данных*.

В построении *общей концептуальной модели данных* выделяют ряд этапов.

- Выделение локальных представлений, соответствующих обычно относительно независимым данным. Каждое такое представление проектируется как подзадача.
- Формулирование сущностей, описывающих локальную предметную область проектируемой БД, и описание атрибутов, составляющих структуру каждой сущности.
- Выделение ключевых атрибутов.
- Спецификация связей между сущностями. Удаление избыточных связей.
- Анализ и добавление неключевых атрибутов.
- Объединение локальных представлений.

Созданная концептуальная модель данных предприятия является источником информации для фазы логического проектирования базы данных.

Логическое проектирование базы данных

Цель второй фазы проектирования базы данных состоит в создании логической модели данных для исследуемой части предприятия.

Логическая модель, отражающая особенности представления о функционировании предприятия одновременно многих типов пользователей, называется *глобальной логической моделью данных*.

Процесс проектирования БД должен опираться на определенную модель данных (реляционная, сетевая, иерархическая), которая определяется типом предполагаемой для реализации информационной системы СУБД.

Концептуальное и логическое проектирование — это итеративные процессы, которые включают в себя ряд уточнений, продолжающиеся до тех пор, пока не будет получен наиболее соответствующий структуре предприятия продукт. **Физическое проектирование базы данных**

Физическое проектирование базы данных

Целью проектирования на данном этапе является создание описания СУБД ориентированной модели БД.

Действия, выполняемые на этом этапе, слишком специфичны для различных моделей данных, поэтому их сложно обобщить. Остановимся на реляционной модели данных. В этом случае под физическим проектированием подразумевается:

- создание описания набора реляционных таблиц и ограничений для них на основе информации, представленной в глобальной логической модели данных;
- определение конкретных структур хранения данных и методов доступа к ним, обеспечивающих оптимальную производительность системы с базой данных;
- разработка средств защиты создаваемой системы.

Разработка приложений

Параллельно с проектированием системы базы данных выполняется разработка приложений. Главные составляющие данного процесса — это проектирование транзакций и пользовательского интерфейса.

Проектирование транзакций

Транзакции представляют некоторые события реального мира.

Транзакция может состоять из нескольких операций, однако с точки зрения пользователя эти операции представляют собой единое целое, переводящее базу данных из одного непротиворечивого состояния в другое. Реализация транзакций опирается на тот факт, что СУБД способна обеспечивать сохранность внесенных во время транзакции изменений в БД и непротиворечивость базы данных даже в случае возникновения сбоя.

Проектирование транзакций заключается в определении:

- данных, которые используются транзакцией;
- функциональных характеристик транзакции;
- выходных данных, формируемых транзакцией;
- степени важности и интенсивности использования транзакции.

Проектирование пользовательского интерфейса

Интерфейс должен быть удобным и обеспечивать все функциональные возможности, предусмотренные в спецификациях требований пользователей.

Специалисты рекомендуют при проектировании пользовательского интерфейса использовать следующие основные элементы и их характеристики:

- содержательное название;
- ясные и понятные инструкции;
- логически обоснованные группировки и последовательности полей;
- визуально привлекательный вид окна формы или поля отчета;
- легко узнаваемые названия полей;
- согласованную терминологию и сокращения;

- согласованное использование цветов;
- визуальное выделение пространства и границ полей ввода данных;
- удобные средства перемещения курсора;
- средства исправления отдельных ошибочных символов и целых полей;
- средства вывода сообщений об ошибках при вводе недопустимых значений;
- особое выделение необязательных для ввода полей;
- средства вывода пояснительных сообщений с описанием полей;
- средства вывода сообщения об окончании заполнения формы.

Реализация

На данном этапе осуществляется физическая реализация базы данных и разработанных приложений, позволяющих пользователю формулировать требуемые запросы к БД и манипулировать данными в БД.

База данных описывается на языке определения данных выбранной СУБД. В результате компиляции его команд и их выполнения создаются схемы и пустые файлы базы данных. На этом же этапе определяются и все специфические пользовательские представления. Прикладные программы реализуются с помощью языков третьего или четвертого поколений. Кроме того, на этом этапе создаются другие компоненты проекта приложения — например, экраны меню, формы ввода данных и отчеты.

Реализация этого, а также и более ранних этапов проектирования БД может осуществляться с помощью инструментов автоматизированного проектирования и создания программ, которые принято называть CASE-инструментами (Computer-Aided Software Engineering).

Загрузка данных

На этом этапе созданные в соответствии со схемой базы данных пустые файлы, предназначенные для хранения информации, должны быть заполнены данными. Наполнение базы данных может протекать по-разному, в зависимости от того, создается ли база данных вновь или новая база данных предназначена для замены старой.

Тестирование

Для оценки законченности и корректности выполнения приложения базы данных может использоваться несколько различных стратегий тестирования:

- нисходящее тестирование;
- восходящее тестирование;
- тестирование потоков;
- интенсивное тестирование.

Эксплуатация и сопровождение

Основные действия, связанные с этим этапом сводятся к наблюдению за созданной системой и поддержке ее нормального функционирования по окончании развертывания. Поддержка БД предполагает разрешение проблем, возникающих в процессе эксплуатации БД и связанных как с ошибками реализации БД, так и с изменениями в самой предметной области, созданием дополнительных программных компонент или модернизацией самой БД.

Вопросы для самоконтроля:

1. Перечислите этапы, составляющие жизненный цикл БД.
2. Что является целью каждого этапа?
3. Какие работы ведутся на каждом из этапов?

Лекция 8. Концептуальное проектирование БД

План:

1. Модель "Сущность - Связь"(ERD)
2. Структурный подход при разработке инфологической модели
3. Моделирование локальных представлений
4. Правила преобразования ER-диаграмм в реляционные таблицы

Концептуальная модель отображает реальный мир в некоторые понятные человеку концепции, полностью независимые от параметров среды хранения данных. Существует множество подходов к построению таких моделей: графовые модели, семантические сети, модель "сущность-связь" и т.д. Наиболее популярной из них оказалась модель "сущность-связь".

Модель "Сущность - Связь"(ERD)

Это модель предметной области, которая используется на этапе концептуального проектирования.

Модель использует три основных элемента: сущность, атрибут и связь.

Сущность - это абстракция какого-либо объекта, процесса или явления реального мира, о котором нужно хранить информацию. В качестве сущности могут выступать материальные (предприятие, товар) и нематериальные (описание явления, реферат статьи) объекты.

Тип сущности определяет набор однородных объектов, а **экземпляр сущности** - конкретный объект в наборе.

Каждый тип сущности обладает одним или несколькими атрибутами.

Каждому типу сущности должно быть дано уникальное имя. К одному и тому же имени должна всегда применяться одна и та же интерпретация.

Для идентификации конкретных экземпляров сущностей используются атрибуты – идентификаторы (один или несколько), которые позволяют однозначно отличать один экземпляр сущности от другого.

Каждая сущность может обладать любым количеством связей с другими сущностями модели.

Атрибут - это поименованная характеристика сущности, которая принимает значения из некоторого множества значений.

Чтобы задать атрибут в модели, необходимо присвоить ему наименование, привести смысловое описание атрибута, определить множество его допустимых значений и указать, для чего он используется.

Основное назначение атрибута - описание свойства сущности, а также идентификация экземпляров сущности.

Атрибут может быть либо обязательным, либо необязательным. Обязательность означает, что атрибут не может принимать неопределенных значений (null values). Атрибут может быть либо описательным, либо входить в состав уникального идентификатора (первичного ключа).

Первичный ключ – набор атрибутов, значения которого однозначно определяют экземпляр сущности.

Внешний ключ - это набор атрибутов, используемый для представления связей между сущностями.

Связь - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь – это средство, с помощью которого представляются отношения между сущностями, имеющие место в предметной области. Связи может даваться имя, выражаемое грамматическим оборотом глагола. Имя каждой связи между двумя данными сущностями должно быть уникальным, но имена связей в модели не обязаны быть уникальными.

Связи могут быть между двумя (бинарные), тремя (тернарные) и более сущностями. Чаще всего используются бинарные. Они классифицируются следующим образом:

Связь "один - к - одному" (1:1)

Когда каждому экземпляру сущности А соответствует один и только один экземпляр сущности Б, и наоборот. Связь двунаправленная.

Связь "один - ко - многим" (1:M)

Это такой тип связи, когда каждому экземпляру сущности А может соответствовать ни одного, один или несколько экземпляров сущности Б, однако каждому экземпляру сущности Б соответствует один и только один экземпляр сущности А.

Связь "многие - к - одному" (M:1)

Это отображение обратно предыдущему.

Связь "многие - ко - многим" (отображение M:N)

Это такой тип связи, при котором каждому экземпляру сущности А может соответствовать ни одного, один или несколько экземпляров сущности Б, и наоборот.

Информацию о проекте оформляют составлением спецификаций по сущностям, атрибутам и отношениям с использованием графических диаграмм. На диаграмме обозначают:

- сущности - прямоугольниками;
- атрибуты - овалами, соединяя их с соответствующими сущностями ненаправленными ребрами; идентифицирующие атрибуты подчеркиваются;
- связи (отношения) - ромбами, соединяя их с соответствующими сущностями ненаправленными ребрами, за исключением бинарных связей, которые соединяются направленными ребрами.

При моделировании используются следующие общие правила:

- используются только три типа конструктивных элементов - сущность, атрибут и связь;
- в отдельном проектом представлении каждый компонент информации моделируется только одним конструктивным элементом, то есть необходимо избегать избыточности.

Структурный подход при разработке инфологической модели

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов. При моделировании предметной области проектировщик разбивает ее на ряд локальных областей, моделирует каждое локальное представление, а затем их объединяет.

Моделирование локальных представлений

Выбор локального представления зависит от масштабов предметной области. Для удобства проектирования желательно использовать в отдельных локальных представлениях шесть - семь сущностей.

1. Формулирование сущностей

На этом этапе необходимо указать типы объектов, о которых нужно хранить информацию. Иногда это сложно сделать, так как отдельный объект можно представить и в виде сущности, и в виде атрибута и в виде связи. Тогда следует проработать несколько вариантов моделей и выбрать наиболее гибкий.

Каждой выбранной сущности должно быть присвоено четкое наименование. Общее их количество не должно быть большим.

2. Выбор идентифицирующего атрибута для каждой сущности.

Необходимо для каждой сущности указать идентификатор, позволяющий однозначно распознавать экземпляры сущности. Это один или несколько атрибутов, называемых ключом. Если в наборе атрибутов такого нет, то его надо ввести. Обычно в этом случае для каждого экземпляра сущности вводится внутренний номер, не имеющий смысла вне системы. Он называется **суррогатным ключом**.

Один и тот же набор объектов может иметь несколько ключей. Один из них является **первичным**. Это ключ, который однозначно идентифицирует отдельные экземпляры сущности. Первичный ключ должен включать в свой состав минимальное количество атрибутов. Ключ, состоящий из нескольких атрибутов, называется составным.

3. Назначение сущностям описательных атрибутов

Спецификация атрибутов заканчивается указанием для каждого атрибута множества значений, которые он может принимать. Если это множество бесконечное, то оно задается указанием типа значений (числовой, символьный и пр.) и диапазона значений для чисел и количества символов для алфавитно-цифровых значений.

4. Спецификация связей

Выявляются зависимости между двумя и более сущностями. Определяется какие из них необходимые, какие избыточные. Каждый тип связи именуется.

Рассмотрим понятие **класс принадлежности сущности**.

Если каждый экземпляр сущности А связан с экземпляром сущности В, то класс принадлежности сущности А является обязательным. Это отмечается на ER-диаграмме черным кружком, помещённым в прямоугольник, смежный с прямоугольником сущности А.

Если не каждый экземпляр сущности А связан с экземпляром сущности В, то класс принадлежности сущности является необязательным. Это отмечается на ER-диаграмме черным кружком, помещённым на линии связи возле прямоугольника сущности А.

Правила преобразования ER-диаграмм в реляционные таблицы

Концептуальные модели позволяют более точно представить предметную область, чем реляционные и другие более ранние модели. Но в настоящее время существует немного СУБД, поддерживающих эти модели. На практике наиболее распространены системы, реализующие реляционную модель. Поэтому необходим метод перевода концептуальной модели в реляционную. Такой метод основывается на формировании набора предварительных таблиц из ER-диаграмм.

Для каждой сущности создается таблица. Причем каждому атрибуту сущности соответствует столбец таблицы.

Правила генерации таблиц из ER-диаграмм опираются на два основных фактора – тип связи и класс принадлежности сущности. Изложим их.

Правило 1: Если связь типа 1:1 и класс принадлежности обеих сущностей является обязательным, то необходима только одна таблица. Первичным ключом этой таблицы может быть первичный ключ любой из двух сущностей.

Правило 2: Если связь типа 1:1 и класс принадлежности одной сущности является обязательным, а другой - необязательным, то необходимо построить таблицу для каждой сущности. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Первичный ключ сущности, для которой класс принадлежности является необязательным, добавляется как атрибут в таблицу сущности с обязательным классом принадлежности.

Правило 3: Если связь типа 1:1 и класс принадлежности обеих сущностей необязательный, то необходимо построить три таблицы - по одной для каждой сущности и одну для связи. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключи обеих сущностей.

Правило 4: Если связь типа 1:М и класс принадлежности сущности на стороне М является обязательным, то необходимо построить таблицу для каждой сущности. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне М.

Правило 5: Если связь типа 1:М и класс принадлежности сущности на стороне М является необязательным, то необходимо построить три таблицы - по одной для каждой сущности и одну для связи. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключи обеих сущностей.

Правило 6: Если связь типа М:N, то необходимо построить три таблицы - по одной для каждой сущности и одну для связи. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключи обеих сущностей.

Вопросы для самоконтроля:

1. Что называется концептуальной моделью?
2. Какие базовые понятия используются на этапе концептуального проектирования?
3. Какие задачи решаются на этапе концептуального проектирования?
4. Перечислите шаги концептуального проектирования.
5. Что называется сущностью и экземпляром сущности?

Лекция 9. Нормализация БД

План:

1. Понятие нормализации
2. Первая нормальная форма
3. Вторая нормальная форма
4. Третья нормальная форма
5. Высшие нормальные формы

При проектировании БД могут появиться нежелательные свойства, такие как избыточность, аномалии обновления, аномалии включения, аномалии удаления и др. Для уменьшения нежелательных характеристик БД к схемам отношений применяют процедуры нормализации.

Нормализация - это разбиение таблицы на две или более, обладающие лучшими свойствами при включении, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Нормализация – это процесс разделения информации на структурные единицы, т.е. таблицы.

Нормализация БД должна быть выполнена с учётом следующего правила: таблицы, которые содержат повторяющуюся информацию, для устранения дублирования значений должны быть разделены на отдельные таблицы, что приводит к сокращению размеров БД. В теории реляционных баз данных вводятся понятия так называемых “нормальных форм” — требований к организации данных в таблицах.

Нормальные формы нумеруются последовательно, по мере ужесточения требований. В правильно спроектированной БД таблицы находятся как минимум в третьей нормальной форме.

Так же можно сказать, что процесс нормализации представляет собой приведение таблиц к требуемому уровню нормальности: первый, второй и третий. Каждый уровень нормальности соответствует определённой нормальной форме.

Теория нормализации основана на концепции нормальных форм. Говорят, что таблица находится в данной нормальной форме, если она удовлетворяет определённому набору требований. Теоретически существует пять нормальных форм, но на практике обычно используются только первые три. Первые две нормальные формы являются промежуточными шагами для приведения базы данных к третьей нормальной форме.

Первая нормальная форма (1НФ)

Первая нормальная форма предписывает, что все данные, содержащиеся в таблице, должны быть атомарными (неделимыми). Перечень соответствующих атомарных типов данных определяется СУБД. Требование 1НФ совершенно естественное. Оно означает, что в каждом поле каждой записи должна находиться только одна величина, но не массив и не какая-либо другая структура данных.

Вторая нормальная форма (2НФ)

Говорят, что таблица находится во второй нормальной форме, если она находится в 1НФ и каждый не ключевой столбец полностью зависит от первичного ключа. Другими словами, значение каждого поля должно полностью определяться значением первичного ключа. Важно отметить, что зависимость от первичного ключа понимается именно как зависимость от ключа целиком, а не от отдельной его составляющей (в случае составного ключа).

Чтобы перейти от первой нормальной формы ко второй, нужно выполнить следующую последовательность действий:

- Определить, на какие части можно разбить первичный ключ, так чтобы некоторые из неключевых полей зависели от одной из этих частей (эти части не обязаны состоять из одной колонки).
- Создать новую таблицу для каждой такой части ключа и группы, зависящих от нее полей и переместить их в эту таблицу. Часть бывшего первичного ключа станет при этом первичным ключом новой таблицы.
- Удалить из исходной таблицы поля, перемещенные в другие таблицы, кроме тех, которые станут внешними ключами.

Третья нормальная форма (3НФ)

Говорят, что таблица находится в 3НФ, если она соответствует 2НФ и все не ключевые столбцы взаимно независимы.

Взаимную зависимость столбцов удобно понимать следующим образом: столбцы являются взаимно независимыми, если нельзя изменить один из них, не изменяя другой.

Чтобы перейти от второй нормальной формы к третьей, нужно выполнить следующую последовательность действий:

- Определить все поля, от которых зависят другие поля.
- Создать новую таблицу для каждого такого поля или группы полей и группы зависящих от него полей и переместить их в эту таблицу. Поле или группа полей, от которого зависят перемещенные поля, станет при этом первичным ключом новой таблицы.
- Удалить из исходной таблицы поля, перемещенные в другие таблицы, кроме тех, которые станут внешними ключами.

Высшие нормальные формы

В теории реляционных баз данных рассматриваются и формы высших порядков — нормальная форма Бойса — Кодда, 4НФ, 5НФ и даже выше. Большого практического значения эти формы не имеют, и разработчики, как правило, всегда останавливаются на 3НФ.

Вопросы для самоконтроля:

1. Назовите цели нормализации.
2. Чем опасно избыточное дублирование информации?
3. Назовите основные свойства нормальных форм.
4. Какие ограничения таблиц относят к 1НФ, 2НФ и 3НФ?
5. Приведите примеры таблиц, соответствующих и не соответствующих требованиям нормальных форм.

Лекция 12. Основные понятия языка SQL

План:

1. Появление языка SQL.
2. Типы команд SQL
3. Преимущества языка SQL

Рост количества данных, необходимость их хранения и обработки привели к тому, что возникла потребность в создании *стандартного языка баз данных*, который мог бы функционировать в многочисленных компьютерных системах различных видов. Действительно, с его помощью пользователи могут манипулировать данными независимо от того, работают ли они на персональном компьютере, сетевой рабочей станции или универсальной ЭВМ.

Одним из языков, появившихся в результате разработки реляционной модели данных, является язык SQL (Structured Query Language), который в настоящее время получил очень широкое распространение и фактически превратился в *стандартный язык реляционных баз данных*. Стандарт на язык SQL был выпущен Американским национальным институтом стандартов (ANSI) в 1986 г., а в 1987 г. Международная организация стандартов (ISO) приняла его в качестве международного.

Типы команд SQL

Реализация в SQL концепции операций, ориентированных на табличное представление данных, позволила создать компактный язык с небольшим набором предложений. Язык SQL может использоваться как для выполнения *запросов* к данным, так и для построения прикладных программ.

Основные *категории команд* языка SQL предназначены для выполнения различных функций, включая построение объектов *базы данных* и манипулирование ими, начальную загрузку данных в *таблицы*, обновление и удаление существующей информации, выполнение *запросов* к *базе данных*, управление доступом к ней и ее общее администрирование.

Основные *категории команд* языка SQL:

- DDL – язык определения данных;
- DML – язык манипулирования данными;
- DQL – язык *запросов*;
- DCL – язык управления данными;
- команды администрирования данных;
- команды управления транзакциями

Определение структур базы данных (DDL)

Язык определения данных (Data Definition Language, DDL) позволяет создавать и изменять структуру объектов *базы данных*, например, создавать и удалять *таблицы*. Основными командами языка DDL являются следующие: **CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX.**

Манипулирование данными (DML)

Язык манипулирования данными (*Data Manipulation Language, DML*) используется для манипулирования информацией внутри объектов *реляционной базы данных* посредством трех основных команд: **INSERT, UPDATE, DELETE.**

Выборка данных (DQL)

Язык *запросов* DQL наиболее известен пользователям *реляционной базы данных*, несмотря на то, что он включает всего одну команду **SELECT**. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования *запросов* к *реляционной базе данных*.

Язык управления данными (DCL - Data Control Language)

Команды управления данными позволяют управлять доступом к информации, находящейся внутри *базы данных*. Как правило, они используются для создания объектов, связанных с доступом к данным, а также служат для контроля над распределением привилегий между пользователями. Команды управления данными следующие: **GRANT, REVOKE.**

Команды администрирования данных

С помощью команд администрирования данных пользователь осуществляет контроль над выполняемыми действиями и анализирует операции *базы данных*; они также могут оказаться полезными при анализе производительности системы. Не следует путать администрирование данных с администрированием *базы данных*, которое представляет собой общее управление *базой данных* и подразумевает использование команд всех уровней.

Команды управления транзакциями

Существуют следующие команды, позволяющие управлять транзакциями *базы данных*: **COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION.** **Преимущества языка SQL**

Язык SQL является основой многих *СУБД*, т.к. отвечает за физическое структурирование и запись данных на диск, а также за чтение данных с диска, позволяет принимать SQL-*запросы* от других компонентов *СУБД* и пользовательских приложений. Таким образом, SQL – мощный инструмент, который обеспечивает пользователям, программам и вычислительным системам доступ к информации, содержащейся в *реляционных базах данных*.

Основные достоинства языка SQL заключаются в следующем:

- стандартность;
- независимость от конкретных *СУБД*;
- возможность переноса с одной вычислительной системы на другую;
- реляционная основа языка;
- возможность создания *интерактивных запросов*;
- возможность программного доступа к БД;
- обеспечение различного представления данных;
- возможность динамического изменения и расширения структуры БД;
- поддержка архитектуры *клиент-сервер*.

Любой язык работы с *базами данных* должен предоставлять пользователю следующие возможности:

- создавать *базы данных* и *таблицы* с полным описанием их структуры;

- выполнять основные операции манипулирования данными, в частности, вставку, модификацию и удаление данных из *таблиц*;
- выполнять простые и сложные *запросы*, осуществляющие преобразование данных.

Кроме того, язык работы с *базами данных* должен решать все указанные выше задачи при минимальных усилиях со стороны пользователя, а структура и синтаксис его команд – достаточно просты и доступны для изучения. И наконец, он должен быть универсальным, т.е. отвечать некоторому признанному *стандарту*, что позволит использовать один и тот же синтаксис и структуру команд при переходе от одной *СУБД* к другой. Язык SQL удовлетворяет практически всем этим требованиям.

Вопросы для самоконтроля:

1. Дайте определение понятию SQL.
2. Перечислите категории команд языка SQL.
3. В чем состоит основное достоинство SQL?

Лекция 13. Синтаксис операторов, типы данных

План:

1. Синтаксис SQL-операторов
2. Типы данных SQL

Синтаксис SQL-операторов

Оператор SQL состоит из *зарезервированных слов*, а также из слов, определяемых пользователем. *Зарезервированные слова* являются постоянной частью языка SQL и имеют фиксированное значение. Их следует записывать в точности так, как это установлено, нельзя разбивать на части для переноса с одной *строки* на другую. Слова, определяемые пользователем, задаются им самим (в соответствии с синтаксическими правилами) и представляют собой идентификаторы или имена различных объектов базы данных. Слова в операторе размещаются также в соответствии с установленными синтаксическими правилами.

Идентификаторы языка SQL предназначены для обозначения объектов в базе данных и являются именами таблиц, представлений, столбцов и других объектов базы данных. Символы, которые могут использоваться в создаваемых пользователем идентификаторах языка SQL, должны быть определены как набор символов. Стандарт SQL задает набор символов, который используется по умолчанию, – он включает строчные и прописные буквы латинского алфавита (**A-Z**, **a-z**), цифры (**0-9**) и символ подчеркивания (**_**). На формат идентификатора накладываются следующие ограничения:

- идентификатор* может иметь длину до 128 символов;
- идентификатор* должен начинаться с буквы;
- идентификатор* не может содержать пробелы.

Большинство компонентов языка не чувствительны к регистру. Поскольку у языка SQL свободный формат, отдельные SQL-операторы и их последовательности будут иметь более читаемый вид при использовании отступов и выравнивания.

Язык, в терминах которого дается описание языка SQL, называется *метаязыком*. Синтаксические определения обычно задают с помощью специальной металингвистической символики, называемой *Бэкуса-Науэра формулами* (БНФ). Прописные буквы используются для записи зарезервированных слов и должны указываться в операторах точно так, как это будет показано. Строчные буквы употребляются для записи слов, определяемых пользователем.

Лекция 14. Создание, модификация и удаление таблиц

План:

1. Создание таблицы
2. Модификация таблиц
3. Удаление таблиц

Создание таблицы

```
CREATE TABLE <имя_таблицы>
(<имя_столбца> <тип_столбца>
[NOT NULL]
[UNIQUE | PRIMARY KEY]
[REFERENCES <имя_таблицы> (<имя_столбца>)]
, ...)
```

Пользователь обязан указать имя таблицы и список столбцов. Для каждого столбца обязательно указываются его имя и тип, а также опционально могут быть указаны параметры:

- NOT NULL** - в этом случае элементы столбца всегда должны иметь определенное значение (не NULL)
- один из взаимоисключающих параметров **UNIQUE** - значение каждого элемента столбца должно быть уникальным или **PRIMARY KEY** - столбец является первичным ключом.
- REFERENCES <имя_мастер_таблицы> [<имя_столбца>]** - эта конструкция определяет, что данный столбец является внешним ключом и указывает на ключ какой мастер_таблицы он ссылается.

Контроль за выполнением указанных условий осуществляет СУБД

Модификация таблиц

Как бы тщательно не планировалась структура таблицы, иногда возникает необходимость внести в нее некоторые изменения. Предположим, что в уже сформированную таблицу необходимо добавить столбец. Эту операцию можно выполнять различными путями. Например, можно удалить таблицу со старой структурой и создать вместо нее новую таблицу с нужной структурой. Недостатком этого метода является то, что необходимо будет куда-то скопировать имеющиеся в таблице данные и переписать их в новую таблицу после ее создания.

Специальная команда ALTER TABLE предназначена для модификации структуры таблицы. С ее помощью можно изменять свойства существующих столбцов, удалять или добавлять в таблицу столбцы, а также управлять ограничением целостности, как на уровне столбца, так и на уровне таблицы, т.е. выполнять следующие функции:

- добавить в таблицу определение нового столбца;
- удалить столбец из таблицы;
- изменить значение по умолчанию для какого-либо столбца;
- добавить или удалить первичный ключ таблицы;
- добавить или удалить внешний ключ таблицы;
- добавить или удалить условие уникальности;
- добавить или удалить условие на значение.

Лекция 16-17. Организация запросов на выборку данных при помощи языка SQL

План:

1. Синтаксис оператора SELECT
2. Выборка из нескольких таблиц

Синтаксис оператора SELECT

Для извлечения записей из таблиц в SQL определен оператор **SELECT**. С помощью этой команды осуществляется не только операция реляционной алгебры "выборка" (горизонтальное подмножество), но и предварительное соединение двух и более таблиц. Это наиболее сложное и мощное средство SQL, полный синтаксис оператора **SELECT** имеет вид:

```
SELECT [ALL | DISTINCT] <список_выбора>
FROM <имя_таблицы>
[ WHERE <условие> ]
[ GROUP BY <имя_столбца>,... ]
[ HAVING <условие> ]
[ORDER BY <имя_столбца> [ASC | DESC],... ]
```

Лекция 18. Сортировка и группировка данных при помощи языка SQL

План:

1. Группировка данных
2. Сортировка данных

Группировка данных

Группировка данных в операторе SELECT осуществляется с помощью ключевого слова GROUP BY и ключевого слова HAVING, с помощью которого задаются условия разбиения записей на группы.

GROUP BY неразрывно связано с агрегирующими функциями, без них оно практически не используется. GROUP BY разделяет таблицу на группы, а агрегирующая функция вычисляет для каждой из них итоговое значение.

Ключевое слово HAVING работает следующим образом: сначала GROUP BY разбивает строки на группы, затем на полученные наборы накладываются условия HAVING.

Сортировка данных

Для сортировки данных, получаемых при помощи оператора SELECT служит ключевое слово ORDER BY. С его помощью можно сортировать результаты по любому столбцу или выражению, указанному в <списке_выбора>. Данные могут быть упорядочены как по возрастанию, так и по убыванию.

Ключевое слово DESC задает обратный порядок сортировки, ключевое слово ASC (его можно опускать) - прямой порядок сортировки.

Вопросы для самоконтроля:

1. С помощью какого ключевого слова осуществляется группировка данных в операторе SELECT?
2. Для чего используется ключевое слово HAVING?
3. С помощью какого ключевого слова осуществляется сортировка данных в операторе SELECT?
4. Как отсортировать данные по возрастанию (убыванию)?

Лекция 19. Функции в запросах SQL

План:

1. Агрегатные функции
2. Преобразование текста
3. Работа со строками

SQL позволяет выполнять различные арифметические операции над столбцами результирующего отношения. В конструкции <список_выбора> можно использовать константы, функции и их комбинации с арифметическими операциями и скобками. В арифметических выражениях допускаются операции сложения (+), вычитания (-), деления (/), умножения (*), а также различные функции (COS, SIN, ABS - абсолютное значение и т.д.).

Агрегатные функции

В SQL также определены так называемые агрегатные функции, которые совершают действия над совокупностью одинаковых полей в группе записей. Среди них:

- AVG(<имя поля>)** - среднее по всем значениям данного поля
- COUNT(<имя поля>)** или **COUNT (*)** - число записей
- MAX(<имя поля>)** - максимальное из всех значений данного поля
- MIN(<имя поля>)** - минимальное из всех значений данного поля
- SUM(<имя поля>)** - сумма всех значений данного поля

Следует учитывать, что каждая агрегирующая функция возвращает единственное значение.

Область действия данной функции можно ограничить с помощью логического условия.

Преобразование текста

Часто, текстовые значения заполняются пользователями программного обеспечения по-разному: кто пишет Ф.И.О. с заглавной буквы, кто нет; кто-то пишет все заглавными буквами. Многие отчетные формы требуют унифицированного подхода, да и не только отчетные формы. Для решения этой задачи в SQL есть две функции UCASE - преобразует символы строки в верхний регистр и LCASE - преобразует символы строки в нижний регистр.

Работа со строками

MID(<text>, <start_num>, <num_chars>)- возвращает строку символов из середины текстовой строки с учетом начальной позиции и длины, где text - текстовая строка, из которой нужно извлечь символы, или столбец, содержащий текст; start_num - положение первого символа, который необходимо извлечь (начинаются с 1); num_chars - число возвращаемых символов.

Иногда приходится в качестве аргументов функции MID использовать выражения с функцией LEN:

LEN(column_name)- возвращает длину значения в поле записи.

Функция LEN() исключает из подсчета конечные пробелы.

Вопросы для самоконтроля:

1. Какие операции и функции можно выполнять над данными в SQL?
2. Что такое агрегатные функции? Какие функции входят в эту группу?
3. Какие функции для работы со строками в SQL вам известны?

В настоящее время существует несколько видов СУБД. Наиболее известными и популярными СУБД являются FoxPro, Paradox, MS SQL Server.

Правила выполнения лабораторных работ

Подготовка к лабораторным работам. Лабораторные работы в группах проводятся в соответствии с расписанием учебных занятий в колледже в течение определенного времени. Поэтому для выполнения лабораторных работ магистрант должен руководствоваться следующими положениями:

- Предварительно ознакомиться с графиком выполнения лабораторных работ;
- Внимательно ознакомиться с описанием соответствующей лабораторной работы и установить, в чем состоит основная цель и задача этой работы;
- По лекционному курсу и соответствующим литературным источникам изучить теоретическую часть, относящуюся к данной лабораторной работе;
- Неподготовленные к работе магистранты к выполнению лабораторной работы не допускаются.

Выполнение лабораторных работ. Успешное выполнение лабораторных работ может быть достигнуто в том случае, если экспериментатор отчетливо представляет себе цель эксперимента и ожидаемые результаты, поэтому важным условием обстоятельности проводимых исследований является тщательная подготовка к лабораторной работе.

После окончания работы в лаборатории рабочее место должно быть приведено в порядок. В течение всего времени занятий в лаборатории магистранты обязаны находиться на своих рабочих местах. Выходить из помещения лаборатории во время занятий можно только с разрешения преподавателя.

Оформление отчета по лабораторным работам. Составление отчета о проведенных исследованиях является важнейшим этапом выполнения лабораторной работы. По каждой выполненной работе в рабочей тетради составляют отчет, руководствуясь следующими положениями:

- Указать название и порядковый номер лабораторной работы, а так же краткое сформулировать цель работы;
- Схемы и графики чертить с соблюдением принятых стандартных условий обозначений;
- Отчет по каждой лабораторной работе должен содержать основные выводы. В заголовке отчета указывают номер работы и ее полное наименование. При составлении отчета нужно кратко описать цель работы, ее содержание, указать использованные аппаратуру и оборудование.

При выполнении лабораторных работ необходимо строго следовать правилам техники безопасности.

Тема: Создание и управление базой данных с помощью SQL – операторов.

Цели:

- ознакомиться с принципами создания многотабличной базы данных
- изучить принципы взаимосвязи таблиц
- изучить конструкции SQL - операторов
- изучить принципы создания базы данных при помощи SQL - операторов

В результате выполнения лабораторной работы студент должен:

Знать:

- типы взаимосвязей таблиц
- основные конструкции SQL – операторов
- принципы создания базы данных при помощи SQL

Уметь:

- создавать базу данных при помощи SQL
- строить запросы при помощи языка запросов SQL

Ход выполнения лабораторной работы:

Язык SQL стал фактически стандартным языком доступа к базам данных. Все СУБД, претендующие на название "реляционные", реализуют тот или иной диалект SQL. Многие нереляционные системы также имеют в настоящее время средства доступа к реляционным данным. Целью стандартизации является переносимость приложений между различными СУБД.

Нужно заметить, что в настоящее время, ни одна система не реализует стандарт SQL в полном объеме. Кроме того, во всех диалектах языка имеются возможности, не являющиеся стандартными. Таким образом, можно сказать, что каждый диалект - это надмножество некоторого подмножества стандарта SQL. Это затрудняет переносимость приложений, разработанных для одних СУБД в другие СУБД.

Язык SQL оперирует терминами, несколько отличающимися от терминов реляционной теории, например, вместо "отношений" используются "таблицы", вместо "кортежей" - "строки", вместо "атрибутов" - "колонки" или "столбцы".

Стандарт языка SQL, хотя и основан на реляционной теории, но во многих местах отходит от нее. Например, отношение в реляционной модели данных не допускает наличия одинаковых кортежей, а таблицы в терминологии SQL могут иметь одинаковые строки. Имеются и другие отличия.

Язык SQL является реляционно полным. Это означает, что любой оператор реляционной алгебры может быть выражен подходящим оператором SQL.

Операторы SQL

Основу языка SQL составляют операторы, условно разбитые на несколько групп по выполняемым функциям.

Можно выделить следующие группы операторов (перечислены не все операторы SQL):

Операторы DDL (Data Definition Language) - операторы определения объектов базы данных
CREATE SCHEMA - создать схему базы данных

DROP SCHEMA - удалить схему базы данных

CREATE TABLE - создать таблицу

ALTER TABLE - изменить таблицу

DROP TABLE - удалить таблицу

CREATE DOMAIN - создать домен

ALTER DOMAIN - изменить домен

DROP DOMAIN - удалить домен

CREATE COLLATION - создать последовательность

DROP COLLATION - удалить последовательность

CREATE VIEW - создать представление

DROP VIEW - удалить представление

Операторы DML (Data Manipulation Language) - операторы манипулирования данными

SELECT - отобрать строки из таблиц

INSERT - добавить строки в таблицу

UPDATE - изменить строки в таблице

DELETE - удалить строки в таблице

COMMIT - зафиксировать внесенные изменения

ROLLBACK - откатить внесенные изменения

Операторы защиты и управления данными

CREATE ASSERTION - создать ограничение

DROP ASSERTION - удалить ограничение

GRANT - предоставить привилегии пользователю или приложению на манипулирование объектами

REVOKE - отменить привилегии пользователя или приложения. Кроме того, есть группы операторов установки параметров сеанса, получения информации о базе данных, операторы статического SQL, операторы динамического SQL.

Наиболее важными для пользователя являются операторы манипулирования данными (DML).

Примеры использования операторов манипулирования данными

INSERT - вставка строк в таблицу

Пример 1. Вставка одной строки в таблицу:

```
INSERT INTO
```

```
P (PNUM, PNAME)
```

```
VALUES (4, "Иванов");
```

Пример 2. Вставка в таблицу нескольких строк, выбранных из другой таблицы (в таблицу TMP_TABLE вставляются данные о поставщиках из таблицы P, имеющие номера, большие 2):

```
INSERT INTO
```

```
TMP_TABLE (PNUM, PNAME)
```

```
SELECT PNUM, PNAME
```

```
FROM P
```

```
WHERE P.PNUM > 2;
```

UPDATE - обновление строк в таблице

Пример 3. Обновление нескольких строк в таблице:

```
UPDATE P
```

```
SET PNAME = "Пушников"
```

```
WHERE P.PNUM = 1;
```

DELETE - удаление строк в таблице

Пример 4. Удаление нескольких строк в таблице:

```
DELETE FROM P
```

```
WHERE P.PNUM = 1;
```

Пример 5. Удаление всех строк в таблице: DELETE FROM P;

Примеры использования оператора SELECT

Оператор SELECT является фактически самым важным для пользователя и самым сложным оператором SQL. Он предназначен для выборки данных из таблиц, т.е. он, собственно, и реализует одно из основных назначение базы данных - предоставлять информацию пользователю.

Оператор SELECT всегда выполняется над некоторыми таблицами, входящими в базу данных.

Замечание. На самом деле в базах данных могут быть не только постоянно хранимые таблицы, а также временные таблицы и так называемые представления. Представления - это просто хранящиеся в базе данные SELECT-выражения. С точки зрения пользователей представления - это таблица, которая не хранится постоянно в базе данных, а "возникает" в момент обращения к ней. С точки зрения оператора SELECT и постоянно хранимые таблицы, и временные

получим
таблицу с
колонками,
которых не
было в
исходной
таблице
TOVAR:
TNAME
Болт 10 100 = 1000

ТЕСТЫ

1. Как называется организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения?
 - а) банк данных
 - б) база данных
 - в) информационная система
 - г) реляционная таблица
 - д) СУБД
2. Какая база содержит краткие сведения об описываемых объектах, представленные в строго определённом формате.
 - а) иерархическая
 - б) сетевая
 - в) реляционная
 - г) фактографическая
 - д) документальная
3. Какая база данных содержит обширную информацию самого разного типа: текстовую; графическую, звуковую, мультимедийную?
 - а) документальная
 - б) сетевая
 - в) реляционная
 - г) фактографическая
 - д) распределенная
4. Как называется база данных, разные части которой хранятся на различных ЭВМ компьютерной сети?
 - а) локальная
 - б) распределенная
 - в) сетевая
 - г) иерархическая
 - д) реляционная
5. Что такое реляционная база данных?
 - а) база данных, разные части которой хранятся на различных ЭВМ компьютерной сети.
 - б) базы данных с табличной формой организации
 - в) комплекс аппаратно - программных средств для хранения, изменения и поиска информации, для взаимодействия с пользователем
 - г) база, которая содержит краткие сведения об описываемых объектах, представленные в строго определённом формате.
 - д) организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения.

6. Как классифицируются базы данных по характеру информации?
- а) документальные БД
 - б) сетевые БД
 - в) распределенные БД
 - г) иерархические БД
 - д) фактографические БД
7. Как классифицируются базы данных по структуре данных?
- а) иерархические БД
 - б) сетевые БД
 - в) реляционные БД
 - г) операционные БД
 - д) справочно-информационные БД.
8. Какие бывают базы данных по способу хранения данных?
- а) сетевые БД
 - б) иерархические БД
 - в) централизованные БД
 - г) универсальные БД
 - д) распределенные БД
9. Указать основные понятия реляционной БД?
- а) таблица
 - б) запись
 - в) поле
 - г) тип поля
 - д) главный ключ таблицы
10. Как называется программное обеспечение, предназначенное для работы с базами данных?
- а) СУБД
 - б) ГИБДД
 - в) НКВД
 - г) УСБД
 - д) УФСБ
11. Какие действия можно выполнить с помощью СУБД?
- а) создание структуры базы данных
 - б) заполнение базы данных информацией
 - в) изменение (редактирование) структуры и содержания БД
 - г) поиск информации в базе данных и сортировка данных
 - д) защита данных и проверка целостности БД
12. Назовите главный тип БД. Все остальные разновидности объектов являются производными от неё.
- а) запрос
 - б) форма
 - в) отчёт
 - г) таблица
 - д) модули и макросы
13. Это вспомогательный объект БД, без которого можно, в принципе, и обойтись, они создаются для дополнительного удобства пользователя при просмотре, вводе и редактировании данных.
- а) таблица
 - б) форма
 - в) запрос

- г) отчёт
 - д) макросы и модули
14. Как называется это результат обращения пользователя к СУБД для поиска данных, добавления, удаления, и обновления записей?
- а) отчёт
 - б) таблица
 - в) форма
 - г) запрос
 - д) макросы и модули
15. Как называется документ, предназначенный для вывода на печать, сформированный на основании информации, содержащейся в таблицах и запросах?
- а) отчёт
 - б) форма
 - в) запрос
 - г) таблица
 - д) макросы

Тема «Синтаксис языка SQL»

1. Что обеспечивает опция ADD?
- а) добавление полей
 - б) задаёт условие выполнения запроса
 - в) создаёт или удаляет индексы
 - г) удаление поля таблицы
 - д) объединяет поля
2. Какие запросы SQL бывают?
- а) подчинённые запросы
 - б) запросы к серверу
 - в) запросы на объединение
 - г) синтаксические запросы
 - д) управляющие запросы
3. Что должен сделать пользователь для подчинённого запроса?
- а) создать эквивалентную инструкцию
 - б) создать или удалить индекс
 - в) ввести инструкцию SQL в строку ПОЛЕ
 - г) удалить существующие поля
 - д) добавить новые поля
4. Какие слова используются для поиска значений в основном запросе, которые равны, превышают или меньше значений, возвращаемых подчинённым запросам? .
- а) Anj
 - б) In
 - в) All
 - г) The
 - д) Exist
5. Каким образом можно изменить структуру таблицы?
- а) удалить существующие поля
 - б) генерация поля
 - в) сделать ссылку на внешнюю таблицу
 - г) добавить новые поля
 - д) создать или удалить индексы

6. Что обеспечивает опция DROP? .
- а) индексирование полей
 - б) добавление полей в таблицу
 - в) создание копии таблицы
 - г) создание новой таблицы
 - д) удаление поля таблицы
7. Какая команда вводит ссылку на внешнюю таблицу?
- а) WHERE
 - б) REFERENCES
 - в) ADD
 - г) DISALLOW NULL
 - д) DROP INDEX
8. Какая команда используется для удаления таблицы? .
- а) DISALLOW NULL
 - б) WHERE
 - в) PRIMARY KEY
 - г) ADD
 - д) DROP INDEX
9. Какая команда позволяет указать имена исходных таблиц, участвующих в формировании выборки?
- а) FROM
 - б) DROP
 - в) WHERE
 - г) ICNORE NULL
 - д) SELECT
10. Какие зарезервированные слова используются для проверки наличия результатов подчинённого запроса? .
- а) Exists
 - б) Not Exists
 - в) date
 - г) Create table
 - д) Constraint
11. Какая фраза определяет структуру данных источника передаваемых записей - имена таблицы и полей, содержащих исходные данные для загрузки в таблицу?
- а) DROP
 - б) FOREIGN KEY
 - в) SELECT
 - г) WHERE
 - д) ICNORE NULL
12. Что означает РБД? Создание РБД означает спецификацию состава полей:
- а) указание имени поля
 - б) указание типа поля
 - в) указание длины каждого поля
 - г) указание формата каждого поля
 - д) указание индекса поля
13. Для каких запросов необходимо создать инструкцию SQL непосредственно в окне запроса в режиме SQL?
- а) для запросов к серверу
 - б) для управляющих запросов
 - в) для подчинённых запросов

- г) для запросов на объединение
 - д) для перекрёстных запросов
14. Какой запрос позволяет работать с таблицами на сервере, не прибегая к их связыванию?
- а) подчинённый запрос
 - б) запрос к серверу
 - в) запрос на выборку
 - г) управляющий запрос
 - д) запрос на объединение
15. Какой запрос представляет собой инструкцию SQL Select, вложенную в запрос на выборку или на изменение.
- а) подчинённый запрос
 - б) запрос на изменение
 - в) запрос на выборку
 - г) управляющий запрос
 - д) запрос на объединение

Тема «Развитие архитектуры СУБД»

1. Укажите характеристики, относящиеся к СУБД централизованной архитектуры:
- а) СУБД, база данных и прикладные программы, которые работают с базой данных, функционируют на центральном компьютере.
 - б) процессы, связанные с обработкой данных, производятся на центральном компьютере
 - в) централизованная архитектура СУБД подразумевает доступ из одного узла локальной сети к ресурсам, находящимся на в других узлах
 - г) рабочая станция предназначена для работы пользователя и обладает ресурсами , соответствующими потребностям пользователя
 - д) сервер локальной сети предоставляет ресурсы рабочим станциям и другим серверам.
2. Указать виды серверов сетевой архитектуры БД:
- а) сервер телекоммуникаций
 - б) вычислительный сервер
 - в) дисковый сервер
 - г) файловый сервер
 - д) сервер баз данных
3. Какой сервер обеспечивает услуги по связи данной локальной сети с внешним миром
- а) сервер телекоммуникаций
 - б) файловый сервер
 - в) дисковый сервер
 - г) вычислительный сервер
 - д) сервер баз данных
4. Какой сервер даёт возможность производить вычисления, которые невозможно выполнить на рабочих станциях?
- а) сервер телекоммуникаций
 - б) файловый сервер
 - в) дисковый сервер
 - г) вычислительный сервер
 - д) сервер баз данных

5. Как называется сервер, обладающий расширенными ресурсами внешней памяти и предоставляющий эти ресурсы в использование рабочим станциям и другим серверам?
- а) файловый сервер
 - б) вычислительный сервер
 - в) дисковый сервер
 - г) сервер баз данных
 - д) сервер телекоммуникаций
6. Указать характеристики, относящиеся к многопользовательскому режиму архитектуры файлового сервера:
- а) СУБД может располагаться и работать на нескольких персональных компьютерах, а базы данных располагаются в разделяемых файлах на файловом сервере
 - б) пользователь, работающий на персональном ПК, может обратиться через СУБД к базе данных на файловом сервере
 - в) файловый сервер направляет по сети требуемый блок данных
 - г) недостатком является высокий сетевой трафик
 - д) недостатком является низкий уровень безопасности доступа к данным
7. Какие варианты совместного использования баз данных по технологии файлового сервера существуют?
- а) при использовании средств Access работа БД в сети не зависит от конфигурации и способа размещения на ней СУБД
 - б) совместное использование целой базы данных
 - в) пользователи работают с одними и теми же данными, используя одни и те же формы
 - г) совместное использование только таблиц базы данных Access
 - д) база данных Access размещена на компьютере, выделенном в качестве файлового сервера
8. При совместном использовании только таблиц базы данных Access возможно: .
- а) поместить на сетевой сервер только таблицы и хранить остальные объекты БД на компьютерах пользователей
 - б) работа с базой данных происходит быстрее, так как по сети передаются только данные
 - в) пользователи баз данных могут изменять формы, отчёты и другие объекты баз данных, не влияя на работу других пользователей
 - г) обработка данных осуществляется на сервере
 - д) СУБД Access может быть установлена только на сервере
9. Как называются специальные копии общей базы данных, для которых возможна синхронизация изменений?
- а) нормализация БД
 - б) интеграция БД
 - в) репликация БД
 - г) синхронизация БД
 - д) телекоммуникация БД
10. Преобразование БД в реплицированную БД можно произвести из меню: .
- а) Сервис
 - б) Формат
 - в) Правка
 - г) Вид
 - д) Файл

11. Что происходит с репликами при проведении сеанса синхронизации?
- изменения, сделанные одним пользователем могут автоматически вноситься в общую реплику и реплики других пользователей и наоборот;
 - для отделения таблиц от других объектов базы данных применяется Мастер разделения баз данных
 - при проведении сеанса синхронизации БД пользователи могут менять формы, отчёты и другие объекты БД в соответствии со своими конкретными требованиями
 - если пользователи двух разных реплик по - разному изменили одну и ту же запись, создаётся конфликтная таблица
 - производится обмен обновлёнными записями и объектами между репликами
12. Укажите принципы системной архитектуры "клиент - сервер": .
- система разбивается на две части - клиентскую и серверную
 - в качестве основного интерфейса между клиентской и серверной частью выступает язык БД SQL
 - на рабочих станциях - клиентах работает Access
 - клиентская часть системы при потребности обращается к серверной части
 - SQL - сервер выполняет обработку данных, разме
13. В наиболее полном варианте СУБД может иметь следующие компоненты:
- среда пользователя
 - алгоритмический язык для программирования прикладных систем обработки данных, реализованный как интерпретатор
 - компилятор для придания завершённой программе вида коммерческого продукта в форме независимого exe-файла
 - программы утилиты быстрого программирования рутинных операций (генераторы отчётов, форм, таблиц, экранов, меню и других приложений)
 - информационная система
14. Какие возможности имеются у клиента в системной архитектуре "клиент-сервер SQL"?
- клиент может посылать запросы на сервер SQL
 - клиент может получать с сервера SQL необходимые данные
 - доступ к базе данных от прикладной программы производится путём обращения к клиентской части системы
 - на стороне клиента СУБД работает только такое программное обеспечение, которое не имеет доступа к базам данных, а обращается для этого к серверу с использованием языка БД SQL
 - клиент может посылать обратно н сервер обновлнные данные
15. Укажите серверы баз данных:
- ODBC(Open Database Connectivity)
 - Microsoft Windows NT Serwer
 - SQL Server фирмы Microsoft
 - Oracle Server фирмы Oracle
 - NetWare SQL фирмы Novell

Тема « Разработка многотабличных форм»

1. Как называется форма, созданная на основе нескольких взаимосвязанных таблиц?
- Подчиненная форма
 - Главная форма
 - Многотабличная форма
 - Однотабличная форма
 - Все ответы верны

2. Как называется объект MS Access, предназначенный для просмотра и ввода данных? .
 - а) запрос
 - б) форма
 - в) отчет
 - г) таблица
 - д) макрос
3. Как называется общая структура экранной формы?
 - а) макет формы
 - б) состав полей
 - в) объекты загрузки
 - г) связанная форма
 - д) нет правильного ответа
4. Как может быть построена подчиненная форма?
 - а) на основе только галавной таблице
 - б) на основе только подчиненной формы
 - в) на основе, как подчиненной, так и на основе главной таблицы
 - г) на основе макроса
 - д) на основе связанной таблицы
5. Как может быть создана многотабличная форма?
 - а) в режиме конструктора
 - б) в режиме автоформа
 - в) в режиме мастера форм
 - г) в режиме подчиненной таблицы
 - д) в режиме импорта таблиц
6. Какие типы подключения подчиненной формы нужно выбрать, если таблица - источник основной формы является главной по отношению к другой таблице, тоже выбранной для формы? .
 - а) связанные формы
 - б) подчиненные формы
 - в) одиночные формы
 - г) многотабличная форма
 - д) нет правильного ответа
7. Что надо выбрать в окне Создания форм для непосредственного включения подчиненной формы?
 - а) одиночные формы
 - б) связанные формы
 - в) подчиненные формы
 - г) многотабличные формы
 - д) однотоабличные формы
8. Что надо выбрать в окне Создания форм для включения кнопки, вызывающей связанную форму .
 - а) одиночные формы
 - б) связанные формы
 - в) подчиненные формы
 - г) многотабличные формы
 - д) однотоабличные формы
9. Какую форму автоматически Access создает, если таблица - источник основной части формы является подчиненной по отношению к другой таблице, тоже выбранной для формы?
 - а) многотабличная форма
 - б) одиночная форма

- в) подчиненная форма
 - г) связанная форма
 - д) главная форма
10. Какой вид может иметь подчиненная или связанная форма? .
- а) ленточная
 - б) табличная
 - в) в один столбец
 - г) в одну строку
 - д) в два столбца
11. Какой вид может иметь одиночная форма?
- а) в один столбец
 - б) ленточная
 - в) табличная
 - г) связанная
 - д) все ответы верны
12. Какие режимы создания форм включают для заданной таблицы все поля таблицы и не предоставляют возможности выбора стиля оформления? .
- а) автоформа: в столбец
 - б) мастер форм
 - в) автоформа: ленточная
 - г) автоформа: табличная
 - д) конструктор
13. Какой режим создания форм позволяет пользователю определить, какие поля таблицы включаются в форму, и выбрать стиль ее оформления?
- а) конструктор
 - б) диаграмма
 - в) мастер форм
 - г) автоформа: в столбец
 - д) автоформа: ленточная
14. Какое диалоговое окно представляет возможность выбрать режим создания формы: Конструктор, Мастер форм, Автоформа: в столбец, Автоформа: ленточная, Автоформа: табличная, Диаграмма, Сводная таблица? .
- а) создание форм
 - б) создать
 - в) новая форма
 - г) открыть форму
 - д) открыть
15. Какая форма выводит данные в виде колонок для просмотра только одной записи, поля которой расположены в определенном порядке? .
- а) ленточная
 - б) в один столбец
 - в) табличная
 - г) диаграмма
 - д) в одну строку

Тема «Макросы, их создание»

1. Программа, состоящая из последовательности макрокоманд, это:
- а) макрос
 - б) макрокоманда
 - в) модули

- г) запрос
 - д) отчет
2. Инструкция, ориентированная на выполнение определенного действия:
- а) макрос
 - б) макрокоманда
 - в) запрос
 - г) отчет
 - д) модуль
3. Что позволяет выполнить макрокоманда: ЗАДАТЬ КОМАНДУ МЕНЮ ?
- а) организовывать выполнение макросов
 - б) позволяет задать порядок пережачи макросов
 - в) оперирует объектами, формами , отчетами
 - г) позволяет выполнить часть команд меню
 - д) позволяет выполнить любую команду меню
4. Чем оперируют макросы и модули?
- а) формы
 - б) отчеты
 - в) запросы
 - г) команды
 - д) таблицы
5. Что можно сделать с помощью макрокоманд ?
- а) открыть формы, отчеты
 - б) напечатать запрос
 - в) запустить запрос
 - г) применить фильтр
 - д) отменить фильтр
6. Где можно организовать ветвление?
- а) в программе, состоящей из модулей
 - б) в программе, состоящей из макрокоманд
 - в) в программе, состоящей из таблиц
 - г) в программе, состоящей из запросов
 - д) в программе, состоящей из форм
7. Какие существуют события?
- а) события формы
 - б) события элемента управления
 - в) события записи и раздела формы
 - г) события отчета и раздела отчета
 - д) события запроса и отчета
8. Какие команды используются для организации ветвления программы?
- а) СОЗДАТЬ макрос
 - б) УДАЛИТЬ макрос
 - в) макрокоманды ОСТАНОВИТЬ макрос
 - г) ЗАПУСК макроса
 - д) СОЗДАТЬ модуль
9. Какие бывают средства программирования?
- а) язык макросов
 - б) язык запросов
 - в) язык модулей
 - г) язык SQL
 - д) язык VBA

10. Какую кнопку нужно нажать для выполнения макросов?
- а) кнопку КОНСТРУКТОРА
 - б) кнопку ЗАПУСК
 - в) кнопку МЕНЮ
 - г) кнопку ФОРМАТ
 - д) кнопку СПРАВКА
11. Какую кнопку нужно нажать для просмотра и редактирования имеющегося макроса?
- а) кнопку ФОРМА
 - б) кнопку ЗАПУСК
 - в) кнопку КОНСТРУКТОР
 - г) кнопку МЕНЮ
 - д) кнопку СПРАВКА
12. Где начинается создание макроса?
- а) в отчёте
 - б) в модуле
 - в) в диалоговом окне
 - г) в окне БД
 - д) в таблице
13. Где формируются аргументы макрокоманды (отметить не верные)?
- а) в верхней части окна
 - б) в нижней части окна
 - в) в левой части окна
 - г) в правой части окна
 - д) по центру
14. Укажите примеры событий:
- а) открытие отчета
 - б) ввод новых данных элементов управления в их форме
 - в) щелчок мышки
 - г) удаление фрагмента
 - д) удаление отчета
15. Какая команда открывает окно макросы?
- а) кнопка Меню
 - б) закладка Модули
 - в) кнопка Создать
 - г) кнопка Открыть
 - д) закладка Макросы

Тема «Защита баз данных»

1. Какие традиционные способы защиты имеет база данных?
- а) установка пароля
 - б) защита на уровне пользователя
 - в) ограничение доступа
 - г) шифрование БД
 - д) рабочие группы
2. Кому предоставляет Admins БД право доступа к различным объектам БД? .
- а) администратору
 - б) пользователю
 - в) всем
 - г) профессионалу
 - д) некоторым членам рабочей группы

3. Определить право доступа к некоторому объекту может владелец этого объекта т.е. ...
 - а) кодировщик
 - б) администратор в рабочей группе Admins
 - в) системный пользователь
 - г) пользователю к которому присвоены права администратора именно для этого объекта
 - д) администратор рабочей группы БД
4. К разграничению прав доступа пользователей и групп можно приступить после... .
 - а) создания рабочей группы
 - б) определения администратора
 - в) создания учетных записей
 - г) изменения рабочей группы
 - д) создания пользователя
5. Какой из способов смены владельца существуют для объектов БД:
 - а) копирование всех объектов БД в новый файл
 - б) шифрование
 - в) рабочая группа
 - г) импортирование всех объектов БД в новый файл
 - д) использование вкладки
6. После создания рабочей группы можно приступить к внесению... .
 - а) зашифрованных записей
 - б) учетных записей
 - в) регистрационных записей
 - г) отчетных записей
 - д) пользовательских записей
7. К рабочим группам относятся...
 - а) рабочая группа
 - б) объекты рабочих групп
 - в) элементы рабочей группы
 - г) файл рабочей группы
 - д) администратор рабочих групп
8. Что хранится в файлах рабочей группы? .
 - а) учетные записи
 - б) пароли
 - в) данные о правах доступа к объектам БД
 - г) регистрационные записи
 - д) файлы
9. Просмотр и изменение данных без вставки и удаления - это операция для...
 - а) таблица
 - б) запрос
 - в) макрос
 - г) модуль
 - д) форма
10. Право доступа изменения макета, какие его действия. .
 - а) просмотр
 - б) изменение
 - в) удаление
 - г) переименование
 - д) копирование

11. Право доступа администратора имеет:
- а) база данных
 - б) таблица
 - в) запрос
 - г) форма
 - д) модуль
12. Право доступа открытия Базы данных в монопольном режиме имеет... .
- а) База данных
 - б) программа Access
 - в) формы
 - г) модули
 - д) макросы
13. Право доступа удаления данных имеют..
- а) таблицы
 - б) формы
 - в) отчеты
 - г) запросы
 - д) макросы
14. Право доступа открытия/запуска имеют... .
- а) база данных
 - б) таблица
 - в) форма
 - г) отчет
 - д) макрос
15. Право доступа для чтения макета имеют
- а) таблицы
 - б) запросы
 - в) формы
 - г) отчеты
 - д) макросы