

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Сыров Игорь Анатольевич  
Должность: Директор  
Дата подписания: 20.05.2024 17:06:01  
Уникальный программный ключ:  
b683afe664d7e9f64175886cf9626a198149ad36

ПРИЛОЖЕНИЕ 2

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Стерлитамакский филиал

Колледж

Фонд оценочных средств

по дисциплине **ОЦП.05 Основы проектирования баз данных**

**Общепрофессиональный цикл, вариативная часть**

цикл дисциплины и его часть (обязательная, вариативная)

специальность

**09.02.07**

**Информационные системы и программирование**

код

наименование специальности

квалификация

**Специалист по информационным системам**

Разработчик (составитель)

**Игдисамова Р.Р.**

ученая степень, ученое звание,  
категория, Ф.И.О.

Стерлитамак 2024

## **I Паспорт фондов оценочных средств**

### **1. Область применения**

**Фонд оценочных средств (ФОС)** предназначен для проверки результатов освоения дисциплины ОП.05. Основы проектирования баз данных, входящей в состав программы подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование». **Работа обучающихся во взаимодействии с преподавателем** 68 часов, на самостоятельную работу 10 часов.

### **2. Объекты оценивания – результаты освоения дисциплины**

ФОС позволяет оценить следующие результаты освоения дисциплины в соответствии с ФГОС специальности 09.02.07 «Информационные системы и программирование» и рабочей программой дисциплины ОП.08. Основы проектирования баз данных:

#### **умения:**

- определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска
- применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение
- работать с документами отраслевой направленности.
- собирать, обрабатывать и анализировать информацию на предпроектной стадии.
- работать с современными case-средствами проектирования баз данных.
- создавать объекты баз данных в современных СУБД.
- создавать хранимые процедуры и триггеры на базах данных.

#### **знания:**

- номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации
- современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности
- методы описания схем баз данных в современных СУБД.
- основные положения теории баз данных, хранилищ данных, баз знаний.
- основные принципы структуризации и нормализации базы данных.
- основные принципы построения концептуальной, логической и физической модели данных.
- основные принципы структуризации и нормализации базы данных.
- структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров.
- основные принципы построения концептуальной, логической и физической модели данных.
- современные инструментальные средства проектирования схемы базы данных.
- основные принципы структуризации и нормализации базы данных.
- основные принципы построения концептуальной, логической и физической модели данных.
- структуры данных СУБД.
- методы организации целостности данных.
- модели и структуры информационных систем.

Вышеперечисленные умения, знания направлены на формирование у обучающихся следующих **общих и профессиональных компетенций**:

ОК.02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК.09. Пользоваться профессиональной документацией на государственном и иностранном языках

### **3 Формы контроля и оценки результатов освоения дисциплины**

Контроль и оценка результатов освоения – это выявление, измерение и оценивание знаний, умений и формирующихся общих и профессиональных компетенций в рамках освоения дисциплины.

В соответствии с учебным планом специальности 09.02.07 «Информационные системы и программирование», рабочей программой дисциплины ОП.08. Основы проектирования баз данных предусматривается текущий и промежуточный контроль результатов освоения.

#### **3.1 Формы текущего контроля**

Текущий контроль успеваемости представляет собой проверку усвоения учебного материала, регулярно осуществляемую на протяжении курса обучения.

Текущий контроль результатов освоения дисциплины в соответствии с рабочей программой и календарно-тематическим планом происходит при использовании следующих обязательных форм контроля:

- выполнение и защита лабораторных и практических работ,
- проверка выполнения самостоятельной работы студентов,
- проверка выполнения контрольных работ.

Во время проведения учебных занятий дополнительно используются следующие формы текущего контроля – устный опрос, решение задач, тестирование по темам отдельных занятий.

**Выполнение и защита практических работ.** Практические работы проводятся с целью усвоения и закрепления практических умений и знаний, овладения профессиональными компетенциями. В ходе практической работы студенты приобретают умения, предусмотренные рабочей программой дисциплины, учатся использовать формулы, и применять различные методики расчета, анализировать полученные результаты и делать выводы, опираясь на теоретические знания.

Список практических работ:

- Практическая работа №1. Знакомство с Microsoft Access.
- Практическая работа №2. Создание связей между таблицами.
- Практическая работа №3. Создание формы в режиме автоформы и с помощью мастера.
- Практическая работа №4. Создание запросов на выборку.
- Практическая работа №5. Создание отчётов.
- Практическая работа №6. Практическая работа №6.
- Практическая работа №7. Работа в эмуляторе SQL.
- Практическая работа №8. Связи между таблицами в SQL.
- Практическая работа №9. Знакомство с PostgreSQL.
- Практическая работа №10. Запросы ввода данных в таблицы в PostgreSQL.
- Практическая работа №11. Запросы выборки данных в PostgreSQL.
- Практическая работа №12. Анализ плана запроса - EXPLAIN в PostgreSQL.

**Проверка выполнения самостоятельной работы.** Самостоятельная работа направлена на самостоятельное освоение и закрепление обучающимися практических

умений и знаний, овладение профессиональными компетенциями.

Самостоятельная подготовка обучающихся по дисциплине предполагает следующие виды и формы работы:

- Систематическая проработка конспектов занятий, учебной и специальной технической литературы.
- Самостоятельное изучение материала и конспектирование лекций по учебной и специальной технической литературе.
- Выполнение расчетных заданий.
- Работа со справочной литературой и нормативными материалами.
- Оформление отчетов по лабораторным и практическим работам, и подготовка к их защите.

**Проверка выполнения контрольных работ.** Контрольная работа проводится с целью контроля усвоенных умений и знаний и последующего анализа типичных ошибок и затруднений обучающихся в конце изучения темы или раздела. Согласно календарно-тематическому плану дисциплины предусмотрено проведение следующих контрольных работ:

- *Контрольная работа №1 по теме «MS Access».*
- *Контрольная работа №2 по теме «PostgreSQL».*

Спецификации контрольных работ приведены ниже в данном комплекте ФОС.

**Сводная таблица по применяемым формам и методам текущего контроля и оценки результатов обучения**

| Результаты обучения                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Формы и методы контроля и оценки результатов обучения                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Освоенные умения:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                       |
| <ul style="list-style-type: none"><li>– определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска</li><li>– применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение</li><li>– Работать с документами отраслевой направленности.</li><li>– Собирать, обрабатывать и анализировать информацию на предпроектной стадии.</li><li>– Работать с современными case-средствами проектирования баз данных.</li><li>– Создавать объекты баз данных в</li></ul> | Выполнение и защита практических работ № 1-16<br>Решение задач<br>Устный опрос во время занятия, доклад по выбранной теме.<br>Контрольная работа №1,2 |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>современных СУБД.</p> <ul style="list-style-type: none"> <li>– Создавать хранимые процедуры и триггеры на базах данных.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                             |
| <b>Усвоенные знания:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                             |
| <ul style="list-style-type: none"> <li>– номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</li> <li>– современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности</li> <li>– Методы описания схем баз данных в современных СУБД.</li> <li>– Основные положения теории баз данных, хранилищ данных, баз знаний.</li> <li>– Основные принципы структуризации и нормализации базы данных.</li> <li>– Основные принципы построения концептуальной, логической и физической модели данных.</li> <li>– Основные принципы структуризации и нормализации базы данных.</li> <li>– Структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров.</li> <li>– Основные принципы построения концептуальной, логической и физической модели данных.</li> <li>– Современные инструментальные средства проектирования схемы базы данных.</li> <li>– Основные принципы структуризации и нормализации базы данных.</li> <li>– Основные принципы построения концептуальной, логической и физической модели данных.</li> <li>– Структуры данных СУБД.</li> <li>– Методы организации целостности данных.</li> <li>– Модели и структуры информационных систем.</li> </ul> | <p>Выполнение и защита практических работ № 1-16</p> <p>Решение задач</p> <p>Устный опрос во время занятия, тестирование</p> <p>Контрольная работа №1,2</p> |

|                                                                                                                                                                                                                                                                                                                                                                            |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none"> <li>– номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</li> <li>– современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности.</li> </ul> |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

### 3.2 Форма промежуточной аттестации

Промежуточная аттестация по дисциплине Численные методы – экзамен, спецификация которого содержится в данном комплекте ФОС.

*Обучающиеся допускаются к сдаче экзамена при выполнении всех видов самостоятельной работы, лабораторных, практических и контрольных работ, предусмотренных рабочей программой и календарно-тематическим планом дисциплины /МДК.*

*Дифференцированный зачет/зачет/ итоговая контрольная работа проводится за счет времени отведенного на изучение дисциплины/МДК. При условии своевременного и качественного выполнения обучающимся всех видов работ, предусмотренных рабочей программой дисциплины/МДК.*

#### **Перечень вопросов к экзамену (или дифференцированному зачету, зачету, итоговой контрольной работы)**

##### **Часть А**

1. Что такое информационная система? Из каких компонентов она состоит?
2. Что такое база данных? Какими свойствами она должна обладать? Является ли базой данных бумажная картотека в библиотеке? Ответ обоснуйте.
3. Какие функции выполняет СУБД? Почему произошел переход от множества форматов хранения данных к использованию универсальных СУБД? Приведите примеры и обоснуйте.
4. Удаленные базы данных. Расскажите о файл-серверных и клиент-серверных СУБД, их недостатки и достоинства. Приведите примеры.
5. Объясните значения слов «поле», «запись». Зачем каждому полю присваивают свой тип?
6. Какие типы данных поддерживаются в современных СУБД?
7. Что такое ключ таблицы. Назовите и объясните два свойства ключа. Чем отличаются простой и составной ключи? Чем отличаются понятия «ключ» и «первичный ключ».
8. Объясните, когда одни и те же данные в одной ситуации могут быть ключом, а в другой – нет (например, адрес электронной почты, марка стиральной машины и т.п.). Приведите примеры. В каких случаях в качестве первичного ключа используют номер записи? Можно ли применять такой подход, если в таблице есть другое уникальное поле?
9. Какие методы поиска вы знаете? Чем различаются линейный и двоичный поиск? Назовите их достоинства и недостатки.
10. Что такое индекс в базах данных? Как он строится? Можно ли в одной и той же таблице построить несколько индексов? Объясните принцип поиска с помощью индекса.

11. Что такое целостность базы данных? Какие виды целостности вы знаете? Как обеспечивается физическая целостность данных? Как обеспечивается логическая целостность данных?
12. Почему собрать все данные в одной таблице во многих случаях не выгодно? Ответ обоснуйте. По какому принципу данные разбиваются на несколько таблиц?
13. Что такое внешний ключ таблицы? Приведите примеры.
14. Что такое ссылочная целостность базы данных? Как она обеспечивается?
15. Какие типы связей используются в многотабличных базах данных? Зачем они необходимы? Приведите примеры, ответ обоснуйте.
16. Когда применяется связь «1:1»? Какие поля при этом связываются?
17. Когда применяется связь «1:N»? Какие поля при этом связываются?
18. Опишите проблемы, возникавшие при работе с базами данных в середине XX века. Приведите примеры. Подготовьте сообщение.
19. Расскажите об основных идеях, на которых строится реляционная модель данных.
20. Объясните понятия «кортеж», «отношение». Какие ограничения накладываются на операции с реляционной базой данных?
21. Как связана реляционная модель данных и табличное представление? Какими свойствами должны обладать таблицы в реляционной базе данных? Какие базы данных называются реляционными?
22. Что такое нормализация? Каковы её цели? Как вы понимаете выражение «поле должно быть неделимым»? Приведите примеры.
23. Почему нужно стараться, чтобы структура базы данных (состав таблиц, количество и состав полей) не менялась во время её использования?
24. Что такое запрос в базе данных? Зачем используются запросы? Можно ли строить и изменять запрос, не используя Конструктор? Как создавать запросы в MS Access и PostgreSQL?
25. Что такое отчёт? Откуда берутся данные, которые выводятся в отчёте?
26. Почему иногда запрос приходится добавлять в таблицы, данные из которых не появляются в результатах запроса?
27. Назовите недостатки реляционных баз данных? В каких задачах они проявляются?
28. Что такое экспертная система? Из каких элементов она состоит?
29. Перечислите особенности экспертных систем. В каких областях они применяются?
30. Что такое экспертная система? Почему развитие экспертных систем в последние годы идёт не очень активно?

## Часть В

1. Альпинисты хотят сохранить в базе данных информацию о горных вершинах и о том, кто из них на какую вершину поднимался и в каком году. Помогите им грамотно построить многотабличную базу данных – постройте схему данных с указанием ключевых полей и типов связей между ключевыми полями.
2. Необходимо составить базу данных, в которой хранится школьное расписание. Нужно учесть, что каждый предмет преподают несколько учителей, и каждый учитель может вести занятия по нескольким предметам. Постройте многотабличную базу данных – постройте схему данных с указанием ключевых полей и типов связей между ключевыми полями.
3. Выполните нормализацию однотабличной базы данных по участникам музыкального конкурса:

| Год | Изготовитель | Город | Модель | Скорость | Цена |
|-----|--------------|-------|--------|----------|------|
|-----|--------------|-------|--------|----------|------|

|      |       |                     |       |          |                |
|------|-------|---------------------|-------|----------|----------------|
| 2007 | ВАЗ   | Тольятти            | 1119  | 165 км/ч | 120000 руб     |
| 1995 | ВАЗ   | Тольятти            | 11113 | 130 км/ч | 50000 руб      |
| 1992 | КАМАЗ | Набережные<br>Челны | 5320  | 90 км/ч  | 200000 руб     |
| 2006 | КАМАЗ | Набережные<br>Челны | 55102 | 90 км/ч  | 450000 руб     |
| 2007 | БелАЗ | Жодино              | 75600 | 64 км/ч  | 1200000<br>руб |

4. Создайте в MS Access новую базу данных *Футбол* и сохраните её в своей папке.

Выполните следующие задания:

а) постройте таблицу

|   |           |                 |
|---|-----------|-----------------|
| • | Команда   | Текст [VARCHAR] |
| • | Победы    | Целое [INTEGER] |
|   | Ничьи     | Целое [INTEGER] |
|   | Поражения | Целое [INTEGER] |
|   | Зарплата  | Целое [INTEGER] |

б) заполните таблицу данными;

в) отсортируйте записи по убыванию количества побед;

г) примените фильтр, который отбирает только команды, имеющие более 10 побед и меньше 5 поражений.

5. Создайте в MS Access новую базу данных *Футбол* и сохраните её в своей папке.

Выполните следующие задания:

а) постройте таблицу

|   |           |                 |
|---|-----------|-----------------|
| • | Команда   | Текст [VARCHAR] |
| • | Победы    | Целое [INTEGER] |
|   | Ничьи     | Целое [INTEGER] |
|   | Поражения | Целое [INTEGER] |
|   | Зарплата  | Целое [INTEGER] |

б) заполните таблицу данными;

в) создайте запрос с именем Запрос85, который отбирает всю информацию о командах, имеющих более 8 побед и меньше 5 поражений; Команды должны быть расставлены по убыванию числа побед.

6. Создайте в MS Access новую базу данных *Футбол* и сохраните её в своей папке.

Выполните следующие задания:

а) постройте таблицу



|   |           |                 |
|---|-----------|-----------------|
| • | Команда   | Текст [VARCHAR] |
| • | Победы    | Целое [INTEGER] |
|   | Ничьи     | Целое [INTEGER] |
|   | Поражения | Целое [INTEGER] |
|   | Зарплата  | Целое [INTEGER] |

б) заполните таблицу данными;

в) создайте запрос, который отбирает команды, где зарплата игроков не меньше суммы, введенной пользователем; Команды должны быть расставлены по убыванию зарплат.

7. Создайте в PostgreSQL новую базу данных *Футбол*. Выполните следующие задания:

а) постройте таблицу

|   |           |                 |
|---|-----------|-----------------|
| • | Команда   | Текст [VARCHAR] |
| • | Победы    | Целое [INTEGER] |
|   | Ничьи     | Целое [INTEGER] |
|   | Поражения | Целое [INTEGER] |
|   | Зарплата  | Целое [INTEGER] |

б) заполните таблицу данными;

в) создайте запрос с именем Запрос85, который отбирает всю информацию о командах, имеющих более 8 побед и меньше 5 поражений; Команды должны быть расставлены по убыванию числа побед.

8. Создайте в PostgreSQL новую базу данных. Выполните следующие задания:

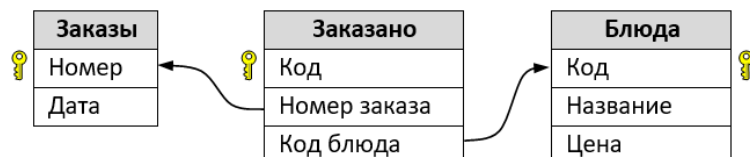
а) постройте таблицу

|   |           |                 |
|---|-----------|-----------------|
| • | Команда   | Текст [VARCHAR] |
| • | Победы    | Целое [INTEGER] |
|   | Ничьи     | Целое [INTEGER] |
|   | Поражения | Целое [INTEGER] |
|   | Зарплата  | Целое [INTEGER] |

б) заполните таблицу данными;

в) создайте запрос, который отбирает команды, где зарплата игроков не меньше суммы, введенной пользователем; Команды должны быть расставлены по убыванию зарплат.

9. Используя запросы PostgreSQL создайте новую базу данных согласно схеме:

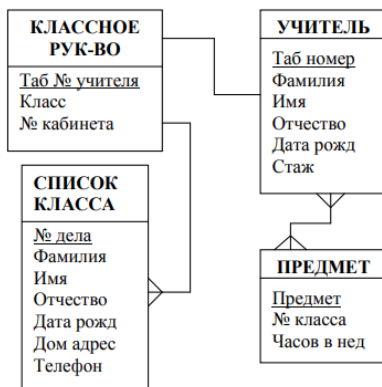


Заполните таблицу данными:

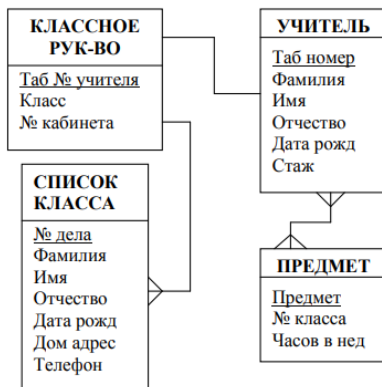
| Заказы |          | Заказано |              |           | Блюда |          |        |
|--------|----------|----------|--------------|-----------|-------|----------|--------|
| Номер  | Дата     | Код      | Номер заказа | Код блюда | Код   | Название | Цена   |
| 1      | 11.04.13 | 1        | 1            | 1         | 1     | борщ     | 80 р.  |
| 2      | 12.04.13 | 2        | 1            | 3         | 2     | бифштекс | 110 р. |
|        |          | 3        | 1            | 4         | 3     | гуляш    | 70 р.  |
|        |          | 4        | 2            | 1         | 4     | чай      | 10 р.  |
|        |          | 5        | 2            | 2         | 5     | кофе     | 50 р.  |
|        |          | 6        | 2            | 2         |       |          |        |
|        |          | 7        | 2            | 5         |       |          |        |

Постройте запрос, который будет выводить минимальную сумму заказа.

10. Постройте в MS Access новую базу данных *Школа*, согласно ER-модели. Объясните, почему ключевые поля выбраны подобным образом, какие типы связей между ними построены.



11. Постройте в PostgreSQL новую базу данных *Школа*, согласно ER-модели. Объясните, почему ключевые поля выбраны подобным образом, какие типы связей между ними построены.



12. Создать БД в MS Access «Прием в ВУЗ» и внести в нее следующие сведения:

|  | Факультеты   | Специальность и | План приема на дневное отделение | Наличие бюджетных мест | Дата начала вступительных экзаменов |
|--|--------------|-----------------|----------------------------------|------------------------|-------------------------------------|
|  | Исторический | История         | 50                               | Да                     | 12.07.08                            |
|  | Исторический | Политология     | 25                               | Да                     | 12.07.08                            |

|  |               |                    |    |     |          |
|--|---------------|--------------------|----|-----|----------|
|  | ий            |                    |    |     |          |
|  | Экономический | Бухгалтерский учет | 40 | Нет | 7.07.08  |
|  | Экономический | Финансы и кредит   | 25 | Да  | 7.07.08  |
|  | Юридический   | Социальная работа  | 25 | Да  | 14.07.08 |
|  | Юридический   | Юриспруденция      | 60 | Нет | 5.07.08  |

1. Создать форму «Информация о приеме», включив в нее все поля БД, кроме поля «№». Откорректировать форму с помощью конструктора.
2. С помощью фильтров найти записи, удовлетворяющие следующим условиям:
  - 1) вся информация по юридическому факультету;
  - 2) факультеты, имеющие бюджетные места, дату начала экзаменов не позднее 13.07.08 и план приема не менее 30 человек.

13. Создать БД в PostgreSQL «Прием в ВУЗ» и внести в нее следующие сведения:

|  | Факультеты    | Специальность и    | План приема на дневное отделение | Наличие бюджетных мест | Дата начала вступительных экзаменов |
|--|---------------|--------------------|----------------------------------|------------------------|-------------------------------------|
|  | Исторический  | История            | 50                               | Да                     | 12.07.08                            |
|  | Исторический  | Политология        | 25                               | Да                     | 12.07.08                            |
|  | Экономический | Бухгалтерский учет | 40                               | Нет                    | 7.07.08                             |
|  | Экономический | Финансы и кредит   | 25                               | Да                     | 7.07.08                             |
|  | Юридический   | Социальная работа  | 25                               | Да                     | 14.07.08                            |
|  | Юридический   | Юриспруденция      | 60                               | Нет                    | 5.07.08                             |

С помощью фильтров найти записи, удовлетворяющие следующим условиям:

- 1) вся информация по юридическому факультету;
- 2) факультеты, имеющие бюджетные места, дату начала экзаменов не позднее 13.07.08 и план приема не менее 30 человек.

14. Создать БД в MS Access «Абитуриенты» и внести в нее следующие сведения:

| Рег. номер | ФИО            | Дата рожд. | Город  | Код специальности | Медаль |
|------------|----------------|------------|--------|-------------------|--------|
| 1012       | Васильева О.Н. | 12.10.90   | Пермь  | 101               | да     |
| 1032       | Быков А.И.     | 24.04.89   | Кунгур | 101               | нет    |
| 1119       | Листьев Д.В.   | 01.12.91   | Кстово | 102               | да     |
| 1120       | Анохин С.П.    | 30.03.90   | Пермь  | 201               | нет    |
| 2010       | Ильин П.В.     | 14.07.8    | Берез  | 101               | да     |

|  |  |   |      |  |  |
|--|--|---|------|--|--|
|  |  | 9 | ники |  |  |
|--|--|---|------|--|--|

1. Создать форму «Абитуриенты», включив в нее все поля БД, кроме поля «Город». Откорректировать форму с помощью конструктора.
2. С помощью фильтров найти записи, удовлетворяющие следующим условиям:
  - 1) вся информация об абитуриентах из Перми;
  - 2) абитуриенты, поступающие на специальность с кодом 101, имеющие медаль и родившиеся не позднее 01.01.90

15. Создать БД в PostgreSQL «Абитуриенты» и внести в нее следующие сведения:

| Рег. номер | ФИО            | Дата рожд. | Город     | Код специальности | Ме даль |
|------------|----------------|------------|-----------|-------------------|---------|
| 1012       | Васильева О.Н. | 12.10.90   | Пермь     | 101               | да      |
| 1032       | Быков А.И.     | 24.04.89   | Кунгур    | 101               | нет     |
| 1119       | Листьев Д.В.   | 01.12.91   | Кстово    | 102               | да      |
| 1120       | Анохин С.П.    | 30.03.90   | Пермь     | 201               | нет     |
| 2010       | Ильин П.В.     | 14.07.89   | Березники | 101               | да      |

1. Создать форму «Абитуриенты», включив в нее все поля БД, кроме поля «Город». Откорректировать форму с помощью конструктора.
2. С помощью запросов найти записи, удовлетворяющие следующим условиям:
  - 1) вся информация об абитуриентах из Перми;
  - 2) абитуриенты, поступающие на специальность с кодом 101, имеющие медаль и родившиеся не позднее 01.01.90.

16. Создать БД в MS Access «Железо» и внести в нее следующие сведения:

|  | Микропроцессор | Тактовая частота (МГц) | ОП (Мб) | Винчестер (Мб) | Цена (тыс.руб) |
|--|----------------|------------------------|---------|----------------|----------------|
|  | Intel 486DX2   | 66                     | 16      | 200            | 4.800          |
|  | Intel 486DX    | 33                     | 8       | 80             | 3.500          |
|  | Intel 486SX2   | 25                     | 4       | 200            | 3.200          |
|  | Pentium        | 66                     | 32      | 1000           | 5.200          |
|  | Intel 486DX    | 33                     | 2       | 50             | 2.500          |

1. Создать форму «Железо», включив в нее все поля БД, кроме поля «№». Откорректировать форму с помощью конструктора.
2. С помощью фильтров найти записи, удовлетворяющие следующим условиям:
  - 1) все процессоры фирмы Intel;
  - 2) процессоры, имеющие объем ОП не менее 4 Мбайт, объем винчестера – более 100 Мбайт и не дороже 5.000 рублей.

17. Создать БД в PostgreSQL «Железо» и внести в нее следующие сведения:

| Микропроцессор | Тактовая частота (МГц) | ОП (Мб) | Винчестер (Мб) | Цена (тыс.руб) |
|----------------|------------------------|---------|----------------|----------------|
| Intel 486DX2   | 66                     | 16      | 200            | 4.800          |
| Intel 486DX    | 33                     | 8       | 80             | 3.500          |
| Intel 486SX2   | 25                     | 4       | 200            | 3.200          |
| Pentium        | 66                     | 32      | 1000           | 5.200          |
| Intel 486DX    | 33                     | 2       | 50             | 2.500          |

С помощью фильтров найти записи, удовлетворяющие следующим условиям:

- 1) все процессоры фирмы Intel;
- 2) процессоры, имеющие объем ОП не менее 4 Мбайт, объем винчестера – более 100 Мбайт и не дороже 5.000 рублей.

18. Создать БД в MS Access «Сотрудники» и внести в нее следующие сведения:

| Фамилия   | Имя   | Дата     | Рост (см) | Вес (кг) | Наличие ПК |
|-----------|-------|----------|-----------|----------|------------|
| Иванов    | Иван  | 5.02.82  | 165       | 52       | Да         |
| Петров    | Петр  | 6.02.81  | 158       | 48       | Да         |
| Сидорук   | Сидор | 10.11.82 | 160       | 42       | Нет        |
| Степанова | Степа | 5.06.82  | 148       | 46       | Да         |
| Стадник   | Стадн | 13.12.81 | 150       | 45       | нет        |

Создать форму «Сотрудники», включив в нее все поля БД, кроме поля «№». Откорректировать форму с помощью конструктора.

С помощью фильтров найти записи, удовлетворяющие следующим условиям:

- 1) все сотрудники, имеющие ПК;
- 2) все мужчины, не ниже 155 см и тяжелее 50 кг.

19. Создать БД в PostgreSQL «Сотрудники» и внести в нее следующие сведения:

| Фамилия   | Имя   | Дата     | Рост (см) | Вес (кг) | Наличие ПК |
|-----------|-------|----------|-----------|----------|------------|
| Иванов    | Иван  | 5.02.82  | 165       | 52       | Да         |
| Петров    | Петр  | 6.02.81  | 158       | 48       | Да         |
| Сидорук   | Сидор | 10.11.82 | 160       | 42       | Нет        |
| Степанова | Степа | 5.06.82  | 148       | 46       | Да         |

|  |             |   |              |     |    |     |
|--|-------------|---|--------------|-----|----|-----|
|  | Стадн<br>ик | 1 | 13.1<br>2.81 | 150 | 45 | нет |
|--|-------------|---|--------------|-----|----|-----|

С помощью фильтров найти записи, удовлетворяющие следующим условиям:

- 1) все сотрудники, имеющие ПК;
- 2) все мужчины, не ниже 155 см и тяжелее 50 кг.

20. Создать БД в PostgreSQL «Почта» и внести в нее следующие сведения:

| Отправит<br>ель  | Адрес<br>отправителя | Вид<br>отправителя | Вид<br>услуги  | Город-<br>адресат | Ве<br>с(кг) |
|------------------|----------------------|--------------------|----------------|-------------------|-------------|
| Готовев<br>А.А.  | Победы, 5            | Юр. лицо           | посылк<br>а    | Москва            | 5,1         |
| Готовев<br>А.А.  | Победы, 5            | Юр. лицо           | посылк<br>а    | Киев              | 7,6<br>3    |
| Готовев<br>А.А.  | Победы, 5            | Юр. лицо           | бандер<br>оль  | Москва            | 1,2         |
| Радостин<br>Е.Л. | Дружбы 15-3          | Физ. лицо          | бандер<br>оль  | Киев              | 0,7         |
| Радостин<br>Е.Л. | Дружбы 15-3          | Физ. лицо          | Зак.<br>письмо | Орёл              | 0,1         |
| Белов<br>И.И.    | Мира, 3              | Юр. лицо           | Зак.<br>письмо | Орёл              | 0,6<br>4    |
| Юдин<br>Г.Г.     | Труда, 7-84          | Физ. лицо          | посылк<br>а    | Москва            | 6,2         |

С помощью фильтров найти записи, удовлетворяющие следующим условиям:

- 1) все отправители с полями «Вид услуги» и «Вес»;
- 2) все отправители, у которых вид услуги бандероль, вес более 1 кг.

#### 4 Система оценивания комплекта ФОС текущего контроля и промежуточной аттестации

##### Критерии оценивания заданий практических работ.

Практическая работа оценивается максимально оценкой «5» (отлично).

Каждое задание оценивается максимально оценкой «5» (отлично).

По результатам оценивания всех заданий оценка соответствует средней.

Критерии оценивания решений задач.

«5» (отлично) – составлен правильный алгоритм решения задачи, в логическом рассуждении, в выборе нормативных источников и решении нет ошибок, получен верный ответ, задача решена рациональным способом.

«4» (хорошо) – составлен правильный алгоритм решения задачи, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор нормативных источников; есть объяснение решения, но задача решена нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ.

«3» (удовлетворительно) – задание выполнено, в логическом рассуждении нет существенных ошибок, но допущены существенные ошибки в выборе нормативных источников; задача решена не полностью или в общем виде.

«2» (неудовлетворительно) – задача решена неправильно.

##### Критерии оценивания ответа по устному опросу.

«5» (отлично) – за глубокое и полное овладение содержанием учебного материала, в котором обучающийся свободно и уверенно ориентируется; за умение практически

применять теоретические знания, высказывать и обосновывать свои суждения; за грамотное и логичное изложение ответа.

«4» (хорошо) – если обучающийся полно освоил учебный материал, владеет научно-понятийным аппаратом, ориентируется в изученном материале, осознанно применяет теоретические знания на практике, грамотно излагает ответ, но содержание и форма ответа имеют отдельные неточности.

«3» (удовлетворительно) – если обучающийся обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности в применении теоретических знаний при ответе на практико-ориентированные вопросы; не умеет доказательно обосновать собственные суждения.

«2» (неудовлетворительно) – если обучающийся имеет разрозненные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

### **Критерии оценивания контрольной работы №1**

Задание к контрольной работе состоит из двух задач, каждая из которых оценивается максимально оценкой «5» (отлично). По результатам оценивания решения двух задач оценка соответствует средней.

### **Критерии оценивания контрольной работы №2**

Задание к контрольной работе состоит из двух задач, каждая из которых оценивается максимально оценкой «5» (отлично). По результатам оценивания решения двух задач оценка соответствует средней.

### **Критерии оценивания ответов по экзаменационным билетам.**

Ответ по экзаменационному билету оценивается максимально оценкой «5» (отлично).

Первый вопрос максимально оценивается оценкой «5» (отлично).

Второй вопрос максимально оценивается оценкой «5» (отлично).

Задача оценивается максимально оценкой «5» (отлично).

По результатам оценивания всех трех вопросов оценка соответствует средней.

При оценивании *практической и самостоятельной работы* студента учитывается следующее:

- *качество выполнения практической части работы;*
- *качество оформления отчета по работе;*
- *качество устных ответов на контрольные вопросы при защите работы.*

Каждый вид работы оценивается по пяти бальной шкале.

«5» (отлично) – за глубокое и полное овладение содержанием учебного материала, в котором обучающийся свободно и уверенно ориентируется; за умение практически применять теоретические знания, высказывать и обосновывать свои суждения. Оценка «5» (отлично) предполагает грамотное и логичное изложение ответа.

«4» (хорошо) – если обучающийся полно освоил учебный материал, владеет научно-понятийным аппаратом, ориентируется в изученном материале, осознанно применяет теоретические знания на практике, грамотно излагает ответ, но содержание и форма ответа имеют отдельные неточности.

«3» (удовлетворительно) – если обучающийся обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности, в применении теоретических знаний при ответе на практико-ориентированные вопросы; не умеет доказательно обосновать собственные суждения.

«2» (неудовлетворительно) – если обучающийся имеет разрозненные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

*Тест оценивается по пяти бальной шкале следующим образом: стоимость каждого вопроса 1 балл. За правильный ответ студент получает 1 балл. За неверный ответ или его отсутствие баллы не начисляются.*

*Оценка «5» соответствует 86% – 100% правильных ответов.*  
*Оценка «4» соответствует 73% – 85% правильных ответов.*  
*Оценка «3» соответствует 53% – 72% правильных ответов.*  
*Оценка «2» соответствует 0% – 52% правильных ответов.*



**Проверка выполнения контрольных работ.** Контрольная работа проводится с целью результатов обучения и последующего анализа типичных ошибок и затруднений обучающихся в конце изучения темы или раздела. Согласно календарно-тематическому плану дисциплины предусмотрено проведение следующих контрольных работ:

### Контрольная работа № 1 Контрольная работа №1

Разработать базу данных согласно описанию предметной области (по вариантам). В работе должны быть четко представлены:

- 1) исследование предметной области;
- 2) ER-модель;
- 3) реляционная модель с описанием функций запросов и отчетов;
- 4) файл базы данных.

Работа делится на теоретическую и практическую части. Теоретическая часть (пункты 1-3.) оформляется обычным образом в формате MS Word, для построения ER-диаграмм можно использовать Paint, можно зарисовать в тетради и загрузить скан в отчет.

Практическая часть (пункт 4) выполняется на компьютере в СУБД MS Access 97 (или позднее).

Создать таблицы и установить связи между ними. Заполнить таблицы. Родительская таблица (соответствующая основному отношению) должна содержать не более 5(!) записей. Остальные – не менее 10 записей. Выполнить запросы и отчеты.

Результат представить в виде файла Access именем, состоящим из номера группы, фамилии, записанной латинскими буквами и номера варианта, согласно образцу:

3ISP11\_Petrov\_12

#### **Вариант 1. Создать базу данных детского сада.**

*В детском саду 4 группы (младшая, средняя, старшая, подготовительная). В каждой группе несколько детей, у которых учитывается фамилия, имя, дата рождения и домашний адрес. Кроме этого необходимо создать таблицу дополнительных сведений о ребенке, где будут учитываться фамилия, имя, отчество одного родителя, его место работы, телефон рабочий и домашний.*

Создать следующие запросы:

- вывести данные детей, фамилии которых начинаются на буквы «А» или «Г»;
- вывести данные детей, возраст, которых равен 5 годам;
- вывести фамилии детей средней группы;
- вывести фамилии детей в возрасте до 5 лет, а также данные их родителей.

Создать следующие отчеты:

- вывести данные всех детей детского сада с указанием домашнего телефона;
  - вывести фамилии детей по группам.

#### **Вариант 2. Создать базу данных туристической базы**

*На туристической базе 5 тренеров, которые занимаются с туристами различными видами спорта («горные лыжи», «слалом», «сноуборд»). За каждым тренером закреплено несколько туристов.*

*У них фиксируются фамилия, имя отчество, дата прибытия и стоимость путевки. Кроме этого необходимо создать для чрезвычайной ситуации таблицу дополнительных сведений о туристе, куда входят домашний адрес, место работы, группа крови.*

Создать следующие запросы:

- вывести данные туристов, фамилии которых начинаются на буквы «В» или «Н»;
- вывести данные туристов, прибывших в январе;
- вывести фамилии туристов, которые занимаются горными лыжами;
- вывести фамилии туристов, которые занимаются слаломом, а также их домашний адрес, место работы и группу крови.

Создать следующие отчеты:

- вывести данные всех туристов;
- вывести фамилии туристов и их тренеров, данные сгруппировать по фамилиям тренеров.

### **Вариант 3. Создать базу данных факультета**

*На факультете несколько групп (94-21, 94-22, 93-21, 93-22, 92- 21). В каждой группе несколько студентов, у которых фиксируются номер зачетной книжки, фамилия, имя, отчество, дата рождения и форма оплаты (бюджетная или внебюджетная). Кроме этого, необходимо создать таблицу дополнительных сведений о студенте, куда следует включить адрес прописки, номер страхового свидетельства, паспортные данные.*

Создать следующие запросы:

- вывести данные студентов, фамилии которых начинаются на буквы «К» или «А»;
- вывести данные студентов, родившихся в феврале;
- вывести данные студентов, которые обучаются с внебюджетной формой оплаты;
- вывести фамилии студентов, прописанных в г. Казань с указанием их домашнего адреса, паспортных данных и номера страхового свидетельства.

Создать следующие отчеты:

- вывести данные всех студентов с указанием адреса прописки, паспортных данных и номера страхового свидетельства;
- вывести фамилии студентов по группам.

### **Вариант 4. Создать базу данных фирмы «Эдельвейс»**

*Фирма «Эдельвейс-М» производит и реализует молочную продукцию. На предприятии существуют следующие отделы:*

*«Отдел продаж», «Технический отдел», «Отдел маркетинга» и*

*«Бухгалтерия». В каждом отделе несколько сотрудников, у которых фиксируются фамилия, имя, отчество, должность (инженер, менеджер, маркетолог, бухгалтер), оклад и дата поступления на работу. Кроме этого необходимо создать таблицу дополнительных сведений о сотруднике, где будут фиксироваться домашний адрес, паспортные данные, дата рождения, телефон.*

Создать следующие запросы:

- вывести данные сотрудников, фамилии которых начинаются на буквы «В» или «М»;
- вывести данные сотрудников, стаж работы которых больше 5 лет;

- вывести данные сотрудников, у которых оклад от 8000 до 1000 рублей;
- вывести фамилии менеджеров, с указанием их домашнего адреса, паспортных данных и телефона.

Создать следующие отчеты:

- вывести фамилии всех сотрудников по отделам;
- вывести данные сотрудников бухгалтерии.

#### **Вариант 5. Создать базу данных отделения социальной защиты**

*В отделении социальной защиты работают 5 патронажных сестер, которые обслуживают лежачих пенсионеров. У каждой патронажной сестры по 2-3 пенсионера, для которых указываются фамилия, имя, отчество, адрес проживания, паспортные данные, сумма пенсии, телефон. Кроме этого имеются дополнительные сведения об основном диагнозе болезни, справочные данные об одном из родственников.*

Создать следующие запросы:

- вывести данные пенсионеров, фамилии которых начинаются на буквы «В» или «М»;
  - вывести данные пенсионеров, у которых пенсия ниже 2000 руб;
  - вывести фамилии пенсионеров, у которых нет телефона
- Создать следующие отчеты:
- вывести фамилии всех пенсионеров, данные сгруппировать по фамилиям патронажных сестер, которые их обслуживают;
  - вывести данные пенсионеров с адресами и телефонами дальних родственников.

#### **Вариант 6. Создать базу данных отделения социальной защиты**

*В отделении социальной защиты работают патронажные сестры, которые обслуживают лежачих пенсионеров. Одна и та же сестра может обслуживать разных пенсионеров. С другой стороны, к каждому пенсионеру могут приходить разные сестры. Для каждого такого визита фиксируются, какая сестра к кому приходила и дата визита. Кроме того, в базе данных должны содержаться сведения о фамилии, имени, отчестве, адресе, дате рождения и телефоне пенсионеров, а также о фамилии и дате поступления на работу патронажных сестер.*

Создать следующие запросы:

- вывести данные пенсионеров, фамилии которых начинаются на буквы «А» или «П»;
- вывести данные патронажных сестер, стаж работы которых больше 5 лет;
- вывести данные о визитах сестер к пенсионерам;
- вывести фамилии пенсионеров, у которых нет телефона.

Создать следующие отчеты:

- вывести фамилии всех патронажных сестер с указанием пенсионеров, которых они обслуживали;
- вывести данные всех пенсионеров.

#### **Вариант 7. Создать базу данных фирмы «М-вид»**

*Фирма «М-вид» реализует товары 4 видов: компьютеры, бытовую технику, оргтехнику и торговое оборудование. Для каждого товара указывается наименование, цена, дата изготовления и производитель. За каждым видом*

товара закреплен свой консультант-менеджер. У менеджера имеется табельный номер, фамилия, имя, отчество, дата поступления на работу и оклад.

Создать следующие запросы:

- вывести данные сотрудников, фамилии которых начинаются на буквы «В» или «М»;
- вывести данные сотрудников, стаж работы которых больше 5 лет;
- вывести фамилии сотрудников, у которых оклад от 8000 до 10000 рублей;
- вывести данные менеджеров, с указанием их домашнего адреса, паспортных данных и телефона.

Создать следующие отчеты:

- вывести данные всех видов товара и отвечающих за него менеджеров;
- вывести товары отдела бытовой техники с указанием производителя.

### **Вариант 8. Создать базу данных фирмы «ЖЖЖ»**

Фирма «ЖЖЖ» реализует гастрономическую продукцию (код, наименование продукта, срок годности) и работает с поставщиками (ИНН, название фирмы). Каждый поставщик поставляет различные продукты. С другой стороны, один и тот же товар может поставляться разными поставщиками. При каждой такой поставке указываются код продукта, ИНН поставщика, цена, дата изготовления продукта, дата поставки и производитель.

Создать следующие запросы:

- вывести данные о продуктах, срок годности которых свыше 10 дней;
- вывести данные фирм, которые поставляли товары в марте;
- вывести данные всех поставок с указанием названий фирм и наименований продуктов.

Создать следующие отчеты:

- вывести данные всех поставок, данные сгруппировать по фирмам;
- вывести данные всех поставок, данные сгруппировать по товарам.

### **Вариант 9. Создать базу данных фирмы «Статус»**

Фирма «Статус» реализует телефоны различных фирм («Nokia», «Siemens», «LG», «Motorola»). Каждая фирма выпускает различные модели, для которых указывается номер модели, код фирмы-изготовителя, год выпуска, вес. При поступлении товара учитываются модель, дата поставки, цена закупочная и цена реализации.

Создать следующие запросы:

- вывести фирмы и модели телефонов, цена реализации которых ниже 5000 рублей;
- вывести фирмы и модели телефонов текущего года выпуска;
- вывести данные телефонов с весом более 200г.

Создать следующие отчеты:

- вывести данные всех телефонов, имеющихся в фирме, сгруппировать по фирмам производителям;
- вывести данные телефонов фирмы «Samsung».

### **Вариант 10. Создать базу данных библиотеки**

В библиотеке книги сортируются по отделам: классическая литература, приключения и фантастика, детективы, женский роман и детская литература. Для книг фиксируются автор, год издания, стоимость. Для авторов – фамилия,

*имя, отчество, страна проживания, годы жизни.*

Создать следующие запросы:

- вывести данные писателей, фамилии которых начинаются на буквы «А» или «К»;
- вывести данные книг, изданных за последние 5 лет;
- вывести данные книг, стоимость которых ниже 100 рублей.

Создать следующие отчеты:

- вывести данные всех книг, сгруппировать по отделам;
  - вывести данные всех книг, написанных писателями из России.

### **Вариант 11. Создать базу данных библиотеки**

*В библиотеке для книг фиксируются автор, год издания, стоимость. Для читателей – номер читательского билета, фамилия, паспортные данные. При получении книги в библиотеке фиксируются номер книги, номер читательского билета, дата получения, отметка о возврате книги.*

Создать следующие запросы:

- вывести данные книг, которые брали читатели, с фамилиями на буквы «С» или «Р»;
- вывести данные книг, которые находятся на руках и данные читателей;
  - вывести данные книг, изданных за последний год.

Создать следующие отчеты:

- вывести все записи получения книг, данные сгруппировать по фамилиям читателей.
- вывести данные всех книг;

### **Вариант 12. Создать базу данных ателье мод**

*Ателье мод представляет собой небольшой комплекс различных залов и служб для оказания услуг населению. В этот комплекс входят зал кроя, пошивочный зал, обметочный зал, зал приемки заказов у населения. В каждом из этих залов работают по несколько сотрудников, у которых в базе данных требуется учесть фамилию, имя, отчество, телефон, дату поступления на работу и специальность. Специальность определяется из штатного расписания, которое представляет собой перечень специальностей и окладов.*

Создать следующие запросы:

- вывести данные сотрудников, фамилии которых начинаются на буквы «С» или «П»;
- вывести данные сотрудников, стаж работы которых больше 5 лет;
- вывести данные сотрудников, у которых оклад от 4000 до 5500 рублей;
- вывести данные сотрудников зала кроя.

Создать следующие отчеты:

- вывести данные всех сотрудников, сгруппировать по отделам;
- вывести данные сотрудников пошивочного зала.

### **Вариант 13. Создать базу данных оптового склада**

*Имеется оптовый склад, в котором 4 отдела: бытовая химия, канцелярские товары, видеокассеты, садовый инвентарь. В каждом отделе работает по несколько сотрудников, у которых фиксируются фамилия, имя, отчество, оклад и дата поступления на работу. За сотрудниками закреплены клиенты. Имеются сведения о месте работы клиентов, их телефоне и фамилии.*

Создать следующие запросы:

- вывести данные сотрудников, фамилии которых начинаются на буквы «В» или «М»;
- вывести данные сотрудников, стаж работы которых больше 5 лет;
- вывести данные сотрудников, у которых оклад от 8000 до 10000 рублей;
- вывести данные клиентов отдела бытовой химии.

Создать следующие отчеты:

- вывести данные всех сотрудников, сгруппировать по отделам;
- вывести данные сотрудников и их постоянных клиентов.

#### **Вариант 14. Создать базу данных факультета**

*На факультете 4 кафедры: «Иностранных языков», «Маркетинга», «Права» и «Охраны труда». Для каждой кафедры фиксируются фамилия заведующего, номер его кабинета, телефон. На каждой кафедре работает по несколько преподавателей. У каждого преподавателя свой табельный номер, должность, предмет, который он ведет. Кроме того, необходимо составить таблицу дополнительных сведений о преподавателе, куда входят, ИНН, номер страхового свидетельства, домашний адрес, домашний телефон.*

Создать следующие запросы:

- вывести данные преподавателей, фамилии которых начинаются на буквы «К» или «А»;
- вывести данные преподавателей кафедры охраны труда;
- вывести данные преподавателей, у которых нет телефона.

Создать следующие отчеты:

- вывести данные всех преподавателей, сгруппировать по кафедрам;
- вывести дополнительные сведения преподавателей кафедры права.

#### **Вариант 15. Создать базу данных ЖЭУ**

*ЖЭУ обслуживает несколько домов, для которых указывается улица, номер дома, этажность, год постройки. Для квартир фиксируются номер, фамилия владельца, площадь (кв. м.), количество проживающих. В ЖЭУ работают управляющие домами (учитывается фамилия), за каждым закреплены несколько домов.*

Создать следующие запросы:

- вывести данные домов по улице Восстания;
- вывести данные домов, которые были построены за последние 2 года;
- вывести данные владельцев квартир, метраж которых менее 40 метров;

Создать следующие отчеты:

- вывести данные всех квартир, сгруппировать по управляющим домами;
- вывести данные всех домов и их управляющих.

### **Контрольная работа № 2**

Задача: создать базу данных, заполнить ее и вывести данные с помощью запросов на экран.

#### **Задание**

Создать и заполнить базу данных для обработки данных по работе книжной лавки,

состоящую из четырех таблиц. Первая таблица должна содержать поля: идентификатор\_жанра, наименование жанра. Вторая: идентификатор\_книги, название книги, автор книги, год выпуска книги, стоимость, количество страниц, серия, идентификатор\_жанра, идентификатор\_издательства, идентификатор\_страны. Третья: идентификатор\_издательства, наименование издателя, адрес издательства и другие поля при необходимости. Четвертая: идентификатор\_страны, наименование страны.

**Не забудьте установить ключевые поля и связи в таблицах!**

На основании созданных таблиц создать запросы:

- ☐ вывести поля: название книги, автор книги, жанр, издательство, количество страниц;
- ☐ вывести книги из Великобритании (автор, название, жанр, издательство, год издания);
- ☐ вывести все книги дороже 500 и 2021 года выпуска (автор, название, стоимость, год издания);
- ☐ вывести все книги в жанре проза, которые были выпущены издательством АСТ;
- ☐ вывести все книги серии Всемирная литература и Библиотека Всемирной Литературы;
- ☐ вывести все книги издательства ЭКСМО, которые стоят более 500;
- ☐ вывести все книги больше чем на 600 страниц, 2022 года выпуска со стоимостью более 500.

Ниже представлены книги, которые нужно внести в базу данных. Не забываем, что данные необходимо распределить по четырём разным таблицам!

| название книги        | автор книги                 | год выпуска | стоимость | жанр  | серия           | количество страниц | издательство |
|-----------------------|-----------------------------|-------------|-----------|-------|-----------------|--------------------|--------------|
| 1. Собачье сердце     | Булгаков Михаил Афанасьевич | 2021        | 3233      | проза | Избранное       | 344                | Нигма        |
| 2. Мастер и Маргарита | Булгаков Михаил Афанасьевич | 2017        | 3600      | проза | Книжный шкаф    | 311                | Верже        |
| 3. Беляя гвардия      | Булгаков Михаил Афанасьевич | 2018        | 163       | проза | Провременем     | 368                | Время        |
| 4. Суламифь           | Куприн Александр Иванович   | 2021        | 368       | проза | Black and White | 112                | Аркадия      |
| 5. Крейцерова соната  | Толстой Лев Николаевич      | 2021        | 692       | проза | ЁЁ Медиа        | 78                 | RUGRAM       |

|                                   |   |                                |      |     |                  |                                 |     |        |
|-----------------------------------|---|--------------------------------|------|-----|------------------|---------------------------------|-----|--------|
| 6. Рассказ неизвестного человека  | Р | Чехов Антон Павлович           | 2011 | 635 | проза            | ЁЁ Медиа                        | 60  | RUGRAM |
| 7. Черный монах                   | Ч | Чехов Антон Павлович           | 2022 | 209 | проза            | Эксклюзив: Русская классика     | 252 | АСТ    |
| 8. Семья вурдалака                | С | Толстой Алексей Константинович | 2022 | 209 | проза            | Эксклюзив: Русская классика     | 320 | АСТ    |
| 9. Проза Мира                     | Р | Андреев Даниил Леонидович      | 2022 | 583 | проза            | Библиотека Всемирной Литературы | 704 | Эксмо  |
| 10. Джейн Эйр                     | Д | Бронте Шарлотта                | 2022 | 228 | проза            | Всемирная литература            | 544 | Эксмо  |
| 11. Карнивола                     | К | Лейпек Дин                     | 2022 | 585 | фэнтези          | Охотники за мирами              | 544 | Эксмо  |
| 12. Вдали от обезумевшей толпы    | В | Гарди Томас                    | 2022 | 649 | зарубежная проза | Элегантная классика             | 679 | Эксмо  |
| 13. Незнакомка из Уайлдфелл-Холла | Н | Бронте Энн                     | 2022 | 300 | зарубежная проза | Всемирная литература            | 480 | Эксмо  |
| 14. 1984. Скотный двор            | 1 | Оруэлл Джордж                  | 2021 | 240 | зарубежная проза | Всемирная литература            | 416 | Эксмо  |



|                                                                                                                  |                                             |      |      |                |                                                                    |     |          |
|------------------------------------------------------------------------------------------------------------------|---------------------------------------------|------|------|----------------|--------------------------------------------------------------------|-----|----------|
| 15. Т<br>аро.<br>Полное<br>руково<br>дство<br>по<br>чтению<br>карт и<br>предск<br>азатель<br>ной<br>практи<br>ке | Лаво<br>Константин                          | 2018 | 734  | Эзоте<br>рика  | Тайн<br>ы<br>таро                                                  | 608 | Эксмо    |
| 16. С<br>ode.<br>Носите<br>ли                                                                                    | Маррс Джон                                  | 2022 | 576  | детек<br>тив   | Альф<br>а-<br>Трилл<br>ер                                          | 352 | Эксмо    |
| 17. З<br>олотые<br>земли.<br>Сокол<br>и<br>Ворон                                                                 | Черкасова<br>Ульяна                         | 2021 | 642  | фэнте<br>зи    | Young<br>Adult.<br>Книж<br>ный<br>бунт.<br>Новы<br>е<br>сказк<br>и | 640 | Эксмо    |
| 18. Д<br>венадц<br>ать<br>месяце<br>в.<br>Сказка<br>для<br>чтения<br>и<br>предста<br>вления                      | Маршак<br>Самуил<br>Яковлевич               | 2021 | 1600 | Сказк<br>и     | Детск<br>ая<br>художеств<br>енная<br>литер<br>атура                | 128 | Лабиринт |
| 19. С<br>транств<br>ующий<br>Цирк<br>Вампи<br>ров                                                                | Лаймон<br>Ричард                            | 2022 | 948  | фанта<br>стика | Леген<br>дарн<br>ые<br>прем<br>ии                                  | 448 | Феникс   |
| 20. Ж<br>иву,<br>пока<br>люблю                                                                                   | Успенская-<br>Ошанина<br>Татьяна<br>Львовна | 2022 | 273  | роман          | Все<br>люди<br>-<br>разны<br>е                                     | 324 | АСТ      |

|                                                                      |                                |      |      |                    |                                           |     |                           |
|----------------------------------------------------------------------|--------------------------------|------|------|--------------------|-------------------------------------------|-----|---------------------------|
| 21. Насколько хватает взгляда . История видения и прогресса человека | Дэнхэм Уэйд Сьюзан             | 2022 | 1344 | Естественные науки | ПроНаука                                  | 464 | Портал                    |
| 22. Приключения попугая Кеши                                         | Курляндский Александр Ефимович | 2022 | 839  | Детская классика   | Детская иллюстрированная классика         | 231 | Малыш                     |
| 23. Приключения Хомы и Суслика. Сказки                               | Иванов Альберт Анатольевич     | 2022 | 548  | сказка             | Лучшая детская книга Лучшая детская книга | 154 | Малыш                     |
| 24. Возрождение Героя Щита. Том 15                                   | Кю Айя, Юсаги Анэко            | 2022 | 343  | манга              | Возрождение Героя Щита                    | 166 | XL Media                  |
| 25. Левша. Повести и рассказы                                        | Лесков Николай Семенович       | 2022 | 551  | Проза              | Классная библиотека                       | 368 | Детская и юношеская книга |
| 26. В запретной синеве                                               | Риера Карме                    | 2022 | 1089 | Роман              | Современная зарубежная проза              | 480 | Книжники                  |

|                                                                                                                                                           |                                   |      |     |                       |                                                  |     |         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|------|-----|-----------------------|--------------------------------------------------|-----|---------|
| 27. Ц<br>веток<br>пустын<br>и. Реальн<br>ая истори<br>я суперм<br>одели<br>Варис<br>Дирие                                                                 | Дирие<br>Варис, Милл<br>ер Кэтлин | 2022 | 326 | Мему<br>ар            | Книг<br>и, о<br>котор<br>ых<br>говор<br>ят       | 352 | Бомбора |
| 28. М<br>etallica.<br>Экстре<br>мальна<br>я<br>биогра<br>фия<br>группы                                                                                    | Уолл Мик                          | 2022 | 464 | Автоб<br>иогра<br>фия | Боги<br>метал<br>-<br>сцены                      | 785 | Бомбора |
| 29. Т<br>есты<br>Айзенк<br>а. IQ.<br>Переза<br>грузка<br>мозга.<br>Лучши<br>й<br>способ<br>развить<br>свои<br>интелл<br>ектуаль<br>ные<br>способ<br>ности | Айзенк Ганс<br>Юрген              | 2022 | 260 | Псих<br>ологи<br>я    | Псих<br>ологи<br>я.<br>Мозг<br>овой<br>штур<br>м | 256 | Эксмо   |
| 30. Т<br>ренинг<br>уверен<br>ного<br>общени<br>я. 56<br>упражн<br>ений,<br>которы<br>е<br>помогу<br>т<br>прокач<br>ать                                    | Биркенбил<br>Вера                 | 2022 | 713 | Псих<br>ологи<br>я    | Псих<br>ологи<br>я<br>обще<br>ния                | 288 | Бомбора |

|                                                                                                                               |                                                                        |      |      |                                                         |                                              |     |                          |
|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|------|------|---------------------------------------------------------|----------------------------------------------|-----|--------------------------|
| навыки<br>комму<br>никации                                                                                                    |                                                                        |      |      |                                                         |                                              |     |                          |
| 31. У<br>чимся<br>готовит<br>ь<br>завтрак<br>и                                                                                |                                                                        | 2022 | 558  | Кули<br>нария                                           | Кули<br>нария<br>. Для<br>начин<br>ающи<br>х | 144 | Эксмо                    |
| 32. Г<br>ерой<br>нашего<br>времен<br>и                                                                                        | Лермонтов<br>Михаил<br>Юрьевич                                         | 2021 | 586  | Проза                                                   | Librar<br>ium                                | 252 | Рипол-Классик            |
| 33. Н<br>овейши<br>й<br>справо<br>чник<br>врача<br>скорой<br>помощ<br>и                                                       | Е. Г.<br>Кодиева                                                       | 2022 | 448  | Энци<br>клопе<br>дия                                    |                                              | 448 | 1000<br>Бестселлеров     |
| 34. С<br>любовь<br>ю к<br>растен<br>иям.<br>Как<br>обустр<br>оить<br>зелены<br>й оазис<br>у себя<br>дома                      | Рейнеберг<br>Андерс, Ше<br>рвен Эрик                                   | 2022 | 1599 | Расте<br>ниевое<br>дство                                | Urban<br>Jungle                              | 232 | Манн, Иванов<br>и Фербер |
| 35. Э<br>лектроа<br>втомат<br>ика<br>универ<br>сальны<br>х и<br>програ<br>ммных<br>станко<br>в.<br>Учебно<br>е<br>пособи<br>е | Чернов<br>Евгений<br>Александров<br>ич, Костенко<br>Андрей<br>Иванович | 2022 | 3575 | Маши<br>ностро<br>ение<br>. Приб<br>орост<br>роени<br>е |                                              | 524 | Инфра-<br>Инженерия      |

|                                                                                                        |                                                                          |      |      |                               |                                                              |     |                     |
|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|------|------|-------------------------------|--------------------------------------------------------------|-----|---------------------|
| 36. П<br>илотир<br>уемые<br>космич<br>еские<br>полеты<br>·<br>Справо<br>чник                           | Карфидов<br>Виктор<br>Юрьевич                                            | 2022 | 2458 |                               |                                                              | 816 | Инфра-<br>Инженерия |
| 37. Л<br>андша<br>фтовед<br>ение.<br>Учебни<br>к                                                       | Торгашев<br>Роман<br>Евгеньевич                                          | 2022 | 939  |                               |                                                              | 192 | Инфра-<br>Инженерия |
| 38. Б<br>езопасн<br>ость<br>жизнед<br>еятельн<br>ости в<br>лесу                                        | Буглаев<br>Анатолий<br>Михайлович<br>, Сиваков<br>Владимир<br>Викторович | 2022 | 822  | Учеб<br>ная<br>литер<br>атура | Осно<br>вы безоп<br>аснос<br>ти<br>жизне<br>деятель<br>ности | 140 | Инфра-<br>Инженерия |
| 39. А<br>лиса в<br>Зазерка<br>лье,<br>или<br>Сквозь<br>зеркало<br>и что<br>там<br>увидел<br>а<br>Алиса | Кэрролл<br>Льюис                                                         | 2020 | 2680 | Сказк<br>а                    | Детск<br>ая<br>худож<br>ествен<br>ная<br>литер<br>атура      | 140 | Лабиринт            |
| 40. Р<br>омео и<br>Джулье<br>тта                                                                       | Шекспир<br>Уильям                                                        | 2021 | 478  | Пьеса                         | Black<br>and<br>White                                        | 160 | Аркадия             |
| 41. П<br>риключ<br>ения<br>Оливер<br>а<br>Твиста                                                       | Диккенс<br>Чарльз                                                        | 2022 | 1008 | Роман                         | Коллек<br>ция                                                | 416 | Качели,             |
| 42. Д<br>евочка<br>из<br>города                                                                        | Воронкова<br>Любовь<br>Федоровна                                         | 2017 | 178  | Проза                         | Школ<br>ьная<br>прогр<br>амма                                | 160 | Стрекоза            |

|                                                                        |                                        |      |      |             |                                         |     |              |
|------------------------------------------------------------------------|----------------------------------------|------|------|-------------|-----------------------------------------|-----|--------------|
| 43. С<br>ахарны<br>й<br>ребёно<br>к.<br>Люди.<br>Событ<br>ия.<br>Факты | Громова<br>Ольга<br>Константино<br>вна | 2020 | 560  | Проза       |                                         | 216 | КомпасГид    |
| 44. О<br>стров<br>Сокров<br>ищ                                         | Стивенсон<br>Роберт<br>Льюис           | 2020 | 2976 | Роман       | Детская<br>художественная<br>литература | 188 | Лабиринт     |
| 45. Е<br>катери<br>на II                                               | Меркулов<br>Дмитрий                    | 2022 | 2420 | История     | Детская<br>художественная<br>литература | 28  | Лабиринт     |
| 46. Б<br>илли<br>Саммерс                                               | Кинг Стивен                            | 2022 | 867  | Проза       | Тёмная<br>башня                         | 544 | АСТ          |
| 47. З<br>ов<br>предков                                                 | Лондон<br>Джек                         | 2019 | 2400 | История     | Детская<br>художественная<br>литература | 120 | Лабиринт     |
| 48. П<br>оллиан<br>на                                                  | Портер<br>Элинон                       | 2021 | 797  | Повест<br>ь | «Romantic»                              | 256 | Качели       |
| 49. Ч<br>удесное<br>путешес<br>твие<br>Нильса<br>с<br>дикими<br>гусями | Лагерлеф<br>Сельма                     | 2017 | 259  | Сказка      | Волшебная<br>книга                      | 176 | ЭНАС-КНИГА   |
| 50. З<br>имний<br>солдат                                               | Мейсон<br>Дэниел                       | 2022 | 920  | Проза       |                                         | 416 | Фантом Пресс |

**Выполнение и защита практических работ.** Практические работы проводятся с целью усвоения и закрепления практических умений и знаний, овладения профессиональными компетенциями. В ходе практической работы студенты приобретают умения, предусмотренные рабочей программой дисциплины, учатся *использовать формулы, и применять различные методики расчета, анализировать полученные результаты и делать выводы, опираясь на теоретические знания.*

### ***Практическая работа №1. Знакомство с Microsoft Access***

**Цель работы:** *научиться создавать новую базу данных и освоить способы создания и редактирования таблиц данных.*

1. Создание новой базы данных. Запустить программу Access 2002 и создать новую базу данных. Для этого воспользоваться пиктограммой на рабочем столе, либо выполнить **Пуск – Программы – Microsoft Access**. В диалоговом окне **Создание файла** (справа) выбрать пункт **Новая база данных**. Присвоить имя (в папке своей группы). Завершить создание БД. В результате будет создан файл новой БД. На экране будет отображено **Окно базы данных**.

2. В левой части **Окна базы данных** расположены вкладки **Таблицы, Запросы, Формы, Отчеты, Макросы, Модули**. Выбрать по очереди различные вкладки.

3. Создание новой таблицы в режиме таблицы. Создать таблицу. Для этого в **Окне базы данных** выбрать вкладку **Таблицы**. Нажать кнопку **Создать**, расположенную в верхней части **Окна базы данных**. Таблицы можно создавать в *режиме таблицы*, в режиме *конструктора*, либо с помощью *мастера таблиц*. В диалоговом окне **Новая таблица** выбрать **Режим таблицы**, нажать кнопку **ОК**. В результате на экране появится таблица, готовая к вводу информации. В верхней части таблицы находятся *заголовки полей (Поле1, Поле2 ...)*. Заккрыть таблицу, сохранив ее под именем *Таблица 1*.

4. Создание новой таблицы в режиме конструктора. Перед вводом данных в таблицу необходимо задать имена полей, их свойства, количество полей в таблице и т.п. Это делается в режиме конструктора. Создать новую таблицу в режиме конструктора. Для этого в **Окне базы данных** выбрать вкладку **Таблицы**. Нажать кнопку **Создать**. В диалоговом окне **Новая таблица** выбрать **Конструктор**, нажать кнопку **ОК**.

При конструировании таблицы необходимо задать имена полей и тип данных (см. рис.12).

**Имя поля** – это заголовки столбцов таблицы. В разделе **Тип данных** можно задать, какие данные и в каком формате будут введены в таблицу (числовой, текстовый, денежный и т.п.). **Описание** поля является необязательным параметром при конструировании таблицы.

Создать в режиме конструктора таблицу:

| № п/п | Фамилия | Дата рождения | Адрес | Паспорт | Сред з/пл | Дети | Пособие |
|-------|---------|---------------|-------|---------|-----------|------|---------|
|-------|---------|---------------|-------|---------|-----------|------|---------|

Для этого в первой строке в разделе **Имя поля** набрать **№ п/п**. Мышкой переключиться в раздел **Тип данных**. Справа от указателя мыши появится стрелка раскрывающегося списка. Развернуть список и выбрать из него тип *Счетчик*. Переключиться на следующую строку.

В разделе **Имя поля** набрать **Фамилия**, в разделе **Тип данных** выбрать *Текстовый*. В нижней половине окна конструктора расположен раздел **Свойства поля**. Его вид зависит от выбранного типа данных (текстовый, числовой, денежный и т.д.). Заполнить свойства для поля *Фамилия*. В разделе **Размер** поля указать 25 (символов), в разделе **Обязательное** поле указать *Да*, в разделе **Пустые строки** – *нет*.

В следующей строке раздела **Имя поля** ввести *Дата рождения*. **Тип данных** задать

*Дата/время*. Заполнить **Свойства поля: Формат поля** – *Краткий формат даты* (см. рис. 12). Для полей *Адрес* и *Паспорт* установить тип *Текстовый*. Для *Средней з/пл* и *Пособия* – денежный с двумя знаками после запятой. Для поля *Дети* установить тип *Логический*, **формат поля** выбрать *Да/нет*.

5. **Задание ключевого поля.** Каждая таблица должна содержать одно или несколько полей, однозначно идентифицирующих каждую запись в таблице. Такое поле называется *ключевым*. Ключ служит для установления связей между таблицами и для предотвращения ввода повторяющихся



Рис. 12. Окно конструктора таблицы.

данных. В данной таблице в качестве ключевого следует использовать поле *№ п/п*. Для того чтобы обозначить поле как ключевое, необходимо в режиме конструктора выделить

поле и нажать кнопку  на панели инструментов.

6. **Сохранение таблицы.** Все объекты в Access, в том числе таблицы, сохраняются стандартным для Windows способом. Сохранить таблицу под именем *Сотрудники*. Закройте таблицу.

7. **Ввод данных в таблицу.** Открыть таблицу *Сотрудники*. Для этого в **Окне базы данных** выделить таблицу и нажать кнопку **Открыть** в верхней части **Окна базы данных**. Ввести 10 записей. Обратите внимание на то, что поле *№ п/п* заполняется автоматически при добавлении новой записи. Сохранить таблицу.

8. **Редактирование таблицы в режиме просмотра таблицы.** Ознакомиться с пунктами меню окна **Таблица**. Используя пункты меню **Формат**, изменить высоту и ширину ячеек для оптимального представления таблицы на экране. Переименовать поле *№п/п* в *Табельный номер*.

9. Переключиться в режим конструктора и для поля *Фамилия* в графе **Формат** поставить знак > (он переводит все символы в ПРОПИСНЫЕ). Переключиться в режим таблицы и посмотреть, как изменились данные.

10. **Заголовок поля.** Переключиться в режим конструктора. В поле *Паспорт* в окне **Свойства поля** в графе **Подпись** ввести: *Паспортные данные*. Переключиться в режим таблицы. Обратите внимание, что заголовок поля изменился. Таким образом, в графе **Подпись** можно задавать *заголовок поля*, если он не совпадает с *именем поля*. Закройте таблицу.

11. **Копирование и удаление таблиц.** Скопировать таблицу. (Для этого таблицу предварительно закрыть!) Копирование таблицы происходит в **Окне базы данных** аналогично копированию файлов в Windows. Отметить таблицу. Скопировать ее в буфер обмена. Вставить таблицу из буфера обмена. При этом присвоить второй таблице имя



Копия таблицы Сотрудники.

Показать работу преподавателю. Удалить скопированную таблицу. Для этого выделить таблицу и нажать клавишу Delete.

### Практическая работа №2. Создание связей между таблицами

Цель работы: научиться создавать и редактировать связи между таблицами БД.

Описание предметной области. Магазин занимается продажей бытовой техники. Бытовые приборы выпускаются несколькими фирмами-изготовителями. Каждая из фирм выпускает несколько приборов. Фирмы расположены в разных странах. На территории одной страны могут располагаться несколько фирм. В каждой из фирм имеется штат менеджеров, осуществляющих продажу продукции.

1. Создать новую базу данных *Техника*. Создать таблицы новой базы данных согласно реляционной модели, представленной на рисунке 13.
2. Создание связей между полями таблиц. Перед созданием связей заккрыть все таблицы! В меню *Сервис* выбрать *Схема данных*. В диалоговом окне *Добавление таблицы* отметить все четыре таблицы и нажать кнопку *Добавить*. Закрыть окно *Добавление таблицы*. В окне *Схема данных* установить связь между соответствующими полями таблиц *Фирмы* и *Приборы*. Для этого щелкнуть мышью по полю *Код фирмы* таблицы *Фирмы* и, удерживая ее, перетащить указатель мыши на поле *Код фирмы* таблицы *Приборы*. На экране появится диалоговое окно *Связи*.

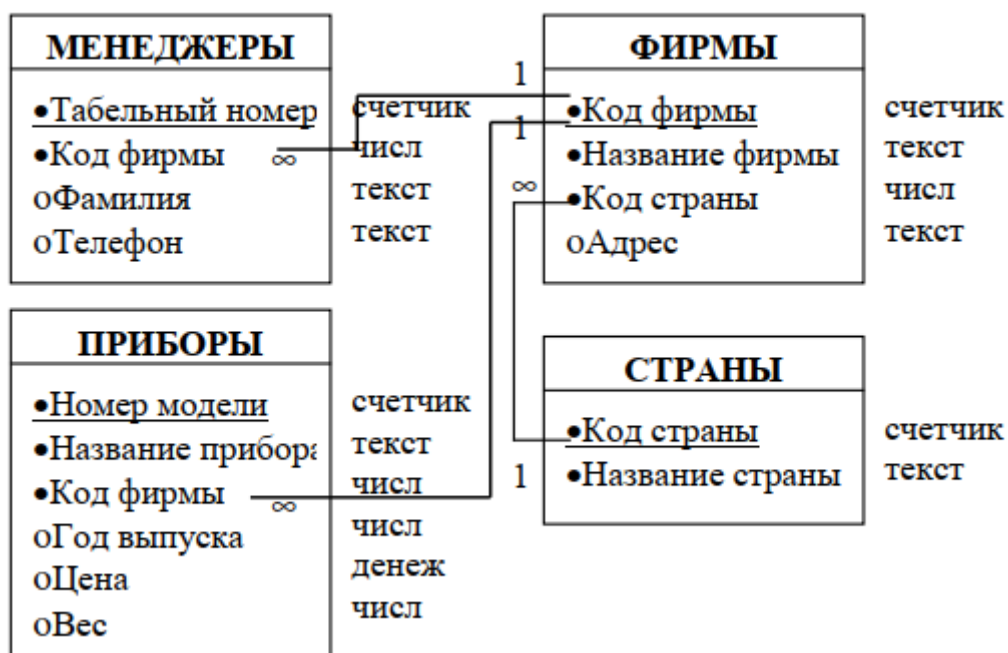


Рис. 13. Реляционная модель БД «Приборы».

В левой части этого окна указано связываемое поле родительской таблицы, а в правой – дочерней. Поставить флажки на пунктах Обеспечение целостности данных и Каскадное обновление, нажать кнопку Создать. В окне *Схемы данных* появится линия связи между таблицами *Фирмы* и *Приборы*. Следует обратить внимание, что на одном конце линии связи стоит знак 1, а на другом ∞. Этот тип связи называется «один-ко-многим». Связь с отношением «один-ко-многим» является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в таблице *Фирмы* могут соответствовать несколько записей в таблице *Приборы*, а запись в таблице *Приборы* не может иметь более одной соответствующей ей записи в таблице *Фирмы*.

3. Аналогичным образом установить связи между остальными полями таблиц. Сравнить с образцом на рисунке 14.

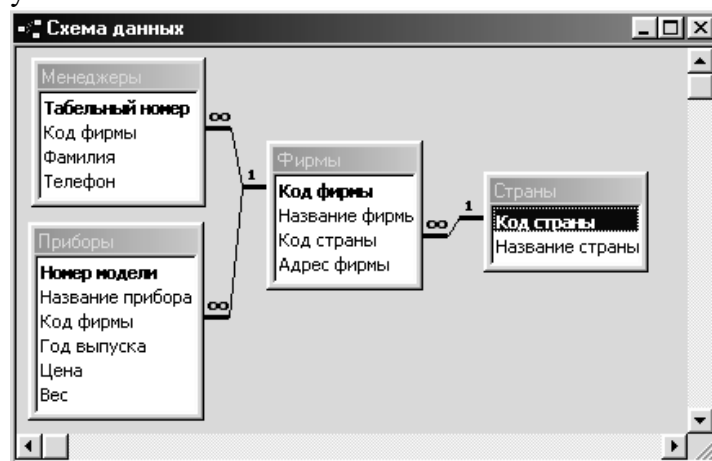


Рис. 14. Схеме данных БД «Приборы»

Сохранить базу данных *Техника*. Показать результат работы преподавателю. Заккрыть базу данных.

### ***Практическая работа №3. Создание формы в режиме автоформы и с помощью мастера***

Цель работы: *научиться создавать формы для работы с данными в режимах автоформы и с помощью мастера форм.*

После построения таблиц в режиме конструктора и установления связей необходимо ввести данные в таблицу (*записи*). Это можно делать как в режиме таблицы, так и используя *Формы*. *Форма* представляет собой *бланк для заполнения таблиц*. Она создает комфорт при работе с Access, защищает базу от действий неквалифицированных пользователей и скрывает конфиденциальную информацию. Кроме того, формы используются при получении ответов на запросы.

#### ***1. Создание формы в режиме Автоформы.***

Открыть базу данных *Техника*. В Окне базы данных выбрать вкладку *Формы*, нажать кнопку *Создать*. Формы можно создавать в *режимах конструктора, мастера форм и автоформы*. При использовании *автоформы* в форме отображаются все поля и записи *базовой таблицы*. Выбрать *Автоформа:* в столбец. В качестве источника данных в раскрывающемся списке в нижней части окна выбрать таблицу *Страны*, нажать кнопку *ОК*. В результате появится автоматически созданная форма в виде столбца. Ввести 5 записей. Закрыть форму.

#### ***2. Создание формы с помощью мастера.***

Создать новую форму с использованием *мастера форм*. Для этого нажать кнопку *Создать* и выбрать *Мастер форм*. В качестве источника данных выбрать таблицу *Фирмы*, нажать кнопку *ОК*. В отличие от автоформы, мастер работает более гибко: есть возможность включить в проектируемую форму *произвольный набор полей* из различных выбранных таблиц.

Используя кнопку *>>*, перенести в форму все поля таблицы *Фирмы*. Обратить внимание, что в таблице присутствует поле *Код страны*, но нет названия страны. Чтобы добавить его в форму, выбрать таблицу *Страны* (см. рис.15). Выделить поле *Название страны* в левой половине окна и *Код страны* в правой половине, нажать *>*. Тогда поле *Название страны* будет помещено после поля *Код страны* в создаваемой форме. Нажать кнопку

Далее. В следующем окне выбрать Вид представления данных - *Фирмы* и далее следовать инструкциям мастера до завершения создания формы. Ввести 5 записей. При вводе данных в поле *Код страны* вводить только те значения, которые уже были введены в таблицу *Страны* (так как поля *Код страны* таблицы *Страны* и *Код страны* таблицы *Фирмы* связаны). Сохранить форму и закрыть.

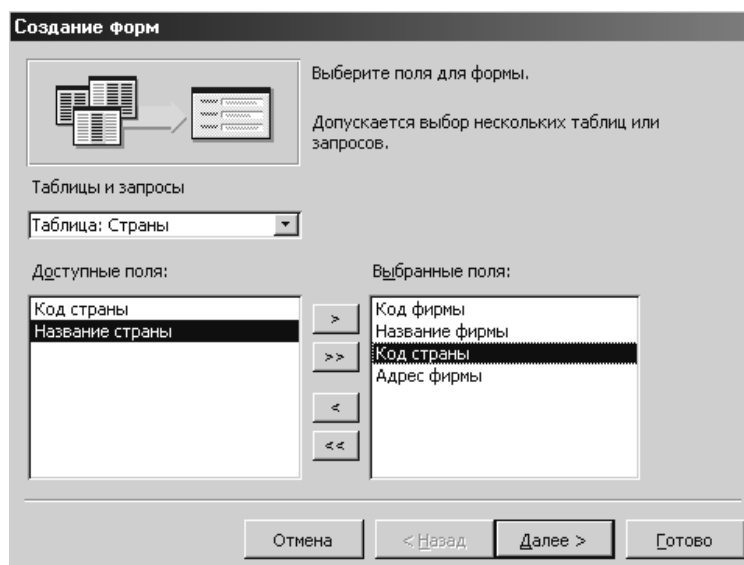


Рис. 15. Окно мастера форм, шаг 2.

3. С помощью мастера форм создать форму *Менеджер*. Включить все поля таблицы *Менеджер* и поле *Название фирмы* таблицы *Фирмы*. Вид представления данных выбрать - *Фирмы*, *Подчиненная форма*. Закончить создание формы, следуя инструкциям. В результате получится *вложенная* форма, где основная форма – *Фирмы*, а подчиненная – *Менеджер*. Для того, чтобы заносить данные о менеджерах, необходимо сначала открыть запись соответствующей фирмы в основной форме, а потом внести записи о менеджерах этой фирмы в подчиненную форму. Для каждой из фирм занести по 3 записи о менеджерах (не дублировать *Номер менеджера*). Сохранить форму и закрыть.

Обратить внимание, что в предыдущем случае (п.2) была создана *одиночная* форма, хотя в ней также использовались поля из двух таблиц. Но в предыдущем случае в качестве основной использовалась таблица *Фирмы*. Связь между таблицами *Фирмы* и *Страны* имеет тип отношения «многие-к-одному», то есть каждой записи таблицы *Фирмы* соответствует *одна* запись таблицы *Страны*. А в последнем случае (п.3) связь *Фирмы* - *Менеджер* имеет тип отношения «один-ко-многим», то есть одной записи таблицы *Фирмы* может соответствовать *несколько* записей таблицы *Менеджер* (у одной фирмы может быть несколько менеджеров).

4. С помощью мастера форм создать форму для ввода данных в таблицу *Приборы*. Включить все поля таблицы *Приборы* и дополнительное поле *Название фирмы* таблицы *Фирмы*. Ввести 10 записей. Сохранить и закрыть форму.

5. Открыть таблицы *Страны*, *Фирмы*, *Менеджер*, *Приборы* и посмотреть внесенные данные. Показать результат работы преподавателю. Закрыть таблицы. Закрыть базу данных.

#### **Практическая работа №4. Создание запросов на выборку**

Цель работы: *ознакомиться с основными типами запросов на выборку и научиться создавать запросы в режиме конструктора запросов.*

Запросы на выборку позволяют извлечь информацию, рассчитать показатели и создать перекрестные ссылки, но *не изменяют данные в таблицах*. Запросы могут создаваться как в режиме конструктора, так и посредством мастера.

1. Создание простых запросов. Простые запросы формируются с помощью мастера и не содержат каких-либо критериев, условий, фильтров. Открыть базу данных *Техника*. Перейти на вкладку Запросы. Выбрать опцию Создать запрос при помощи мастера. На экране появится окно Создание простых запросов. Из таблицы *Страны* внести в запрос поля *Код страны* и *Страна*, из таблицы *Фирмы* – *Название фирмы* и *Адрес*. Далее следуйте инструкциям и завершите создание запроса, сохранив его под именем *Запрос1*. Таким образом, был создан запрос, в результате которого вы получили список страны фирм, которые там находятся.

2. Создание запроса в режиме конструктора. Создание запроса с помощью конструктора - это основной способ формирования запросов. Перейти в Окно базы данных. Активизировать опцию Создать запрос в режиме конструктора. В окне Добавление таблицы добавить таблицы *Страны* и *Фирмы*. Закрыть окно Добавление таблицы. В результате на экране останется окно конструктора запросов (см. рис.16).

Чтобы поместить поля в бланк запроса, надо дважды щелкнуть на соответствующем поле таблицы, либо поставить

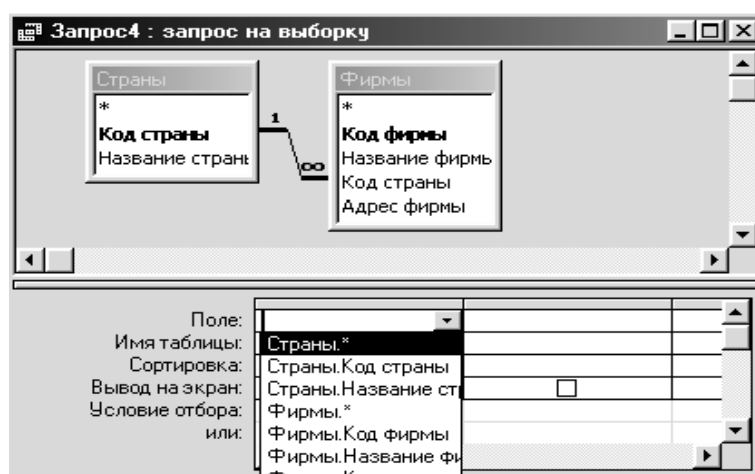


Рис. 16. Окно конструктора запросов.

курсор в строку Поле в нижней части окна и из списка полей выбрать нужные (см. рис. 16). Как и в предыдущем случае, включить в бланк запроса поля с кодом и названием страны, а также с названием и адресом фирмы. Сохранить запрос под именем *Запрос2*. Нажать кнопку запуска запроса . В результате получится таблица, полностью совпадающая с п.1. Закрыть базу данных *Техника*.

3. Создать новую базу данных *Факультет*. В ней в режиме конструктора создать следующую таблицу.

| №п/п           | Фамилия | Имя    | Ср балл аттестата               | Сумма баллов вступит экзамены |
|----------------|---------|--------|---------------------------------|-------------------------------|
| Ключ., счетчик | Текст.  | Текст. | Числ., одинарное с плав. точкой | Числ., длинное целое          |

Сохранить таблицу под именем *Абитуриент*.

4. Ввести 10 записей (желательно, чтобы имена и фамилии начинались в основном на буквы «П» или «Р», средний балл лежал в интервале от 4 до 5, сумма баллов

от 10 до 20).

5. Создание запросов с использованием групповых операций. Запросы с использованием групповых операций предназначены для первичной обработки полученной информации – ее суммирования, осреднения и т.п. В Окне базы данных выбрать Запросы, Создание в режиме конструктора. Добавить таблицу *Абитуриент*. Для того чтобы опция Групповые операции стала доступна, необходимо щелкнуть на кнопке **Σ** на панели инструментов. Тогда в бланке запроса появится дополнительная строка Групповая операция. Включить в бланк запроса поле №n/n. Щелкнуть в строке Групповые операции, развернуть стрелку прокрутки и выбрать функцию **Count**, которая суммирует количество записей, возвращаемых запросом. Сохранить запрос под именем *Запрос5* и запустить его. Закрыть. Выполнить следующие запросы, сохраняя каждый из них под именами *Запрос5\_1*, *Запрос5\_2*...

➤ Рассчитать среднее значение поля *Средний балл* и поля *Сумма баллов*. Для этого для полей *Средний балл* и *Сумма баллов* за экзамены выбрать функцию **Avg**, вычисляющую среднее арифметическое набора чисел поля.

➤ Найти минимальное значение поля *Сумма баллов*.

➤ Показать последнюю запись в таблице *Абитуриент*. В запрос должны быть включены все поля таблицы.

6. Создание запросов с условиями отбора. *Условия отбора*

- это ограничения, которые пользователь накладывает на запрос для определения записей, с которыми он будет работать. Чтобы установить условие отбора для поля в бланке запроса, в ячейку Условие отбора для данного поля необходимо ввести выражение. При написании выражения необходимо учитывать несколько общих правил. При работе с текстовыми данными их заключают в кавычки (= "Иванов"). При работе с датами используют символ # (<#12.03.1998#). Числа набирают без каких-то дополнительных символов. Пустые позиции обозначаются ключевым словом *Null*.

6.1. Отбор записей, содержащих значения, которые больше, меньше, равны, не равны или находятся между указанными значениями. Построить запрос к таблице *Абитуриент*. Включить все поля. Для поля *Средний балл* ввести условие >4,5. Запустить запрос. Вернуться в режим конструктора. Удалить это условие. В поле *Фамилия* ввести условие >="Иванов" (Все фамилии, начиная с «Иванов» и до конца алфавита). Запустить запрос. Вернуться в режим конструктора. Удалить условие. В поле № задать условие *Between 4 And 8* (числа в диапазоне от 4 до 8). Запустить запрос. Сохранить последний запрос под именем *Запрос6\_1*. Закрыть.

6.2. Отбор записей, начинающихся или кончающихся определенными наборами символов. Для этого используются подстановочные знаки: \* - заменяет любое количество цифр или символов, ? – заменяет один символ. Построить запрос к таблице *Абитуриент*. Включить все поля. Используя подстановочные символы, вывести (по очереди) а) все записи с фамилиями, начинающимися на букву «Р»; б) все записи с именами из 4 букв; в) все записи, в которых средний балл аттестата заканчивается на цифру «5». Сохранить последний запрос под именем *Запрос6\_2*. Закрыть.

6.3. Отбор записей, кроме определенных. Для отбора записей *кроме* определенных используется ключевое слово *Not*. Построить запрос к таблице *Абитуриент*. Включить все поля. Вывести: а) фамилии *кроме* начинающихся с буквы «П»: *Not П\**; б) непустые позиции в поле *Имя*: *Not Null*. Сохранить последний запрос под именем *Запрос6\_3*. Закрыть.

6.4. Запросы с дополнительными условиями отбора. Для того же поля или для нескольких полей можно ввести *дополнительные условия отбора*. Если выражения вводятся в несколько ячеек Условие отбора, то они автоматически комбинируются с

помощью операторов **And** или **Or**. Если выражения находятся в *разных* ячейках, но в *одной* строке, то автоматически используется оператор **And**, что означает, что будут возвращены только записи, отвечающие условиям отбора, указанным *во всех ячейках*. Если же выражения находятся в *разных* строках бланка запроса, то автоматически используется оператор **Or**, что означает, что будут возвращены записи, отвечающие *любому* из условий отбора, указанных в ячейках.

Выполнить следующие запросы, сохраняя каждый из них под именами *Запросб\_41*, *Запросб\_42...*

- Вывести записи, у которых *Средний балл* выше 4,6 и *Сумма баллов* >16.
- Вывести записи, у которых *Средний балл* выше 4,6 или *Сумма баллов* >16.
- Вывести записи, у которых *Фамилия* начинается на «П» или на «Р».
- Вывести записи, у которых *Фамилия* начинается на «П» или *Имя* на «Р».
- Вывести записи, у которых (*Фамилия* начинается на «П» и *Средний балл* >4.6) или (*Имя* начинается на «Р» и *Сумма баллов* >16).

### ***Практическая работа №5. Создание отчётов***

Цель работы: *научиться создавать отчеты.*

Технология создания отчета в Access 2002 аналогична созданию формы. Разница между этими двумя объектами заключается в их целевом назначении. Если форма необходима для работы с данными на экране, то отчет предназначен для печати и представляет ценность как материальная копия документа, полученного из базы данных.

Отчеты можно формировать с помощью: *автоотчета*, *мастера отчетов* и *конструктора отчетов*.

1. Создание отчета с помощью автоотчета. Отчет, созданный таким образом, будет отображать *все поля и записи* базовой таблицы или запроса.

Создать отчет по таблице *Страны* базы данных *Техника*. Для этого открыть БД *Техника*. В Окне базы данных выбрать вкладку *Отчеты*. Создать новый отчет в режиме автоотчета в столбец. В качестве источника данных выбрать таблицу *Страны*. Завершить создание отчета. Сохранить отчет под именем *Страны* (через меню *Файл – Сохранить*). Закрыть созданный отчет.

Аналогично в режиме автоотчета создать отчет по таблице *Фирмы*, используя автоотчет: ленточный. Что изменилось в отображении данных? Сохранить отчет под именем *Фирмы*.

2. Создание отчета с помощью мастера. В отличие от форм, в отчетах можно *группировать* и *сортировать* данные. С помощью *мастера отчетов* создать отчет по таблице *Приборы*, включив в нее все поля, кроме поля *Код фирмы*. На следующем шаге добавить уровень группировки *по году выпуска*. Далее добавить сортировку данных *по названию прибора*. Следуя инструкциям мастера, завершить создание отчета. Сохранить его под именем *Приборы*.

4. Создание отчета на основе запроса. Для того чтобы выводить на печать данные, полученные с помощью запросов, используются отчеты, созданные на *базе запросов*. Технология создания таких отчетов полностью совпадает с созданием отчетов на базе таблиц, только в качестве источника необходимо выбрать не таблицу, а соответствующий запрос.

Создать автоотчет: ленточный на базе одного из запросов (любого) БД *Техника*. Сохранить его под именем *Отчет\_Запрос*. Показать работу преподавателю.

### ***Практическая работа №6. Практическая работа №6***

Пример выполнения контрольной работы

## 1. Исследование предметной области.

ЖЭУ обслуживает дома своего района. В ЖЭУ работают управляющие домами, дворники и сантехники. Для домов указывается улица, номер дома, этажность и количество подъездов. Для рабочих – фамилия, имя, отчество, дата поступления на работу, должность и оклад. Каждый сотрудник ЖЭУ может обслуживать несколько домов.

Создать следующие запросы:

- вывести данные сотрудников, обслуживающих улицу Декабристов;
- вывести фамилии сотрудников со стажем более 3 лет;
- вывести фамилии дворников и дома, которые они обслуживают.

Создать следующие отчеты:

- вывести данные всех сотрудников;
- вывести данные всех домов, сгруппировать по управляющим домами.

## 2. ER-модель.

Согласно описанию предметной области можно выделить следующие сущности: «сотрудники» и «дома».

Так как каждый сотрудник может обслуживать несколько домов, то на один экземпляр сущности «сотрудники» приходится несколько экземпляров сущности «дома».

С другой стороны, в каждом доме может работать несколько сотрудников ЖЭУ (дворник, лифтер, сантехник и т.п.). Поэтому на один экземпляр сущности «дома» может приходиться несколько экземпляров сущности «сотрудники».

Таким образом, тип связи между сущностями «дома» и «сотрудники» будет «многие-ко-многим», а ER-модель будет выглядеть следующим образом (рис. 17).

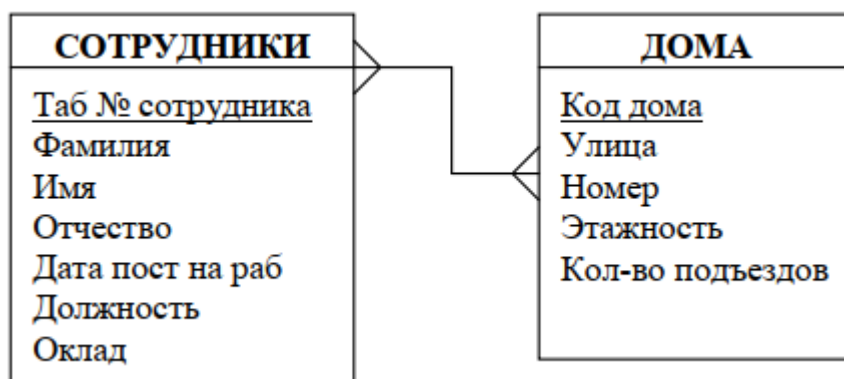


Рис. 17. ER-модель БД ЖЭУ.

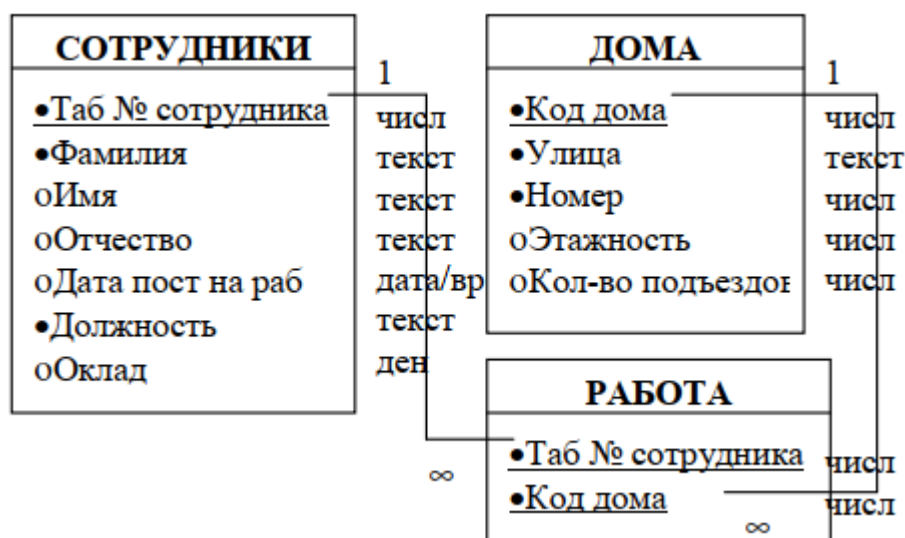
В сущность «сотрудники» в качестве ключевого был добавлен атрибут «Таб № сотрудника». В сущности «дома» – «Код дома».

## 3. Реляционная модель.

Согласно алгоритму преобразования ER-модели в реляционную модель данных сущностям «сотрудники» и «дома» будут соответствовать одноименные отношения с соответствующими атрибутами. Для реализации связи «многие-ко-многим», которая не поддерживается реляционной моделью, необходимо ввести связующее отношение.

Обозначим его как «работа».

Его атрибутами будут «Таб № сотрудника» и «Код дома», а их комбинация будет образовывать составной ключ отношения. Полученная таким образом модель представлена на рисунке 18.



1 18. Реляционная модель БД ЖЭУ.

База данных «ЖЭУ» была разработана в MS Access 2002. Нарисунке 19 представлена схема базы данных «ЖЭУ».

Рис. 19. Схема данных «ЖЭУ».



Родительскими здесь являются таблицы «сотрудники» и «дома», в соответствии с требованиями в них содержится по 5 записей.

В таблице «работа» - не менее 10 записей. Содержание таблиц приведено ниже.

Таблица «Дома».

| Код дома | Улица       | Номер | Этажность | Кол-во подъездов |
|----------|-------------|-------|-----------|------------------|
| 1        | Декабристов | 4     | 2         | 4                |
| 2        | Декабристов | 8     | 2         | 4                |
| 3        | Декабристов | 10    | 9         | 8                |
| 4        | Восстания   | 22    | 5         | 6                |
| 5        | Восстания   | 24    | 5         | 6                |

Таблица «Сотрудники»

| Таб № | Фамилия   | Имя       | Отчество   | Дата пост на раб | Должн     | Оклад   |
|-------|-----------|-----------|------------|------------------|-----------|---------|
| 1     | Салтыков  | Федор     | Петрович   | 10.06.2001       | дворник   | 3 000р. |
| 2     | Малинин   | Николай   | Сергеевич  | 01.01.2004       | дворник   | 3 000р. |
| 3     | Куприянов | Василий   | Николаевич | 05.08.2002       | сантехник | 3 500р. |
| 4     | Никитина  | Маргарита | Семеновна  | 01.01.2000       | управдом  | 3 800р. |
| 5     | Лебедева  | Юлия      | Сергеевна  | 08.05.2004       | управдом  | 3 800р. |



Таблица «Работа»

| Таб № сотрудника | Код дома |
|------------------|----------|
| 1                | 1        |
| 1                | 2        |
| 1                | 3        |
| 2                | 4        |
| 2                | 5        |
| 3                | 1        |
| 3                | 2        |
| 3                | 3        |
| 4                | 1        |
| 4                | 3        |
| 4                | 4        |
| 5                | 2        |
| 5                | 5        |

Были созданы следующие запросы.

Запрос «Сотрудники-Декабристов» выводит данные сотрудников, обслуживающих улицу Декабристов.

| Таб № | Фамилия   | Имя       | Отчество   | Дата поступления | Должность | Оклад  | Улица       | Номер |
|-------|-----------|-----------|------------|------------------|-----------|--------|-------------|-------|
| 1     | Салтыков  | Федор     | Петрович   | 10.06.2001       | дворник   | 3 000. | Декабристов | 4     |
| 1     | Салтыков  | Федор     | Петрович   | 10.06.2001       | дворник   | 3 000. | Декабристов | 8     |
| 1     | Салтыков  | Федор     | Петрович   | 10.06.2001       | дворник   | 3 000. | Декабристов | 10    |
| 3     | Куприянов | Василий   | Николаевич | 05.08.2002       | сантехник | 3 500. | Декабристов | 4     |
| 3     | Куприянов | Василий   | Николаевич | 05.08.2002       | сантехник | 3 500. | Декабристов | 8     |
| 3     | Куприянов | Василий   | Николаевич | 05.08.2002       | сантехник | 3 500. | Декабристов | 10    |
| 4     | Никитина  | Маргарита | Семеновна  | 01.01.2000       | управдом  | 3 800. | Декабристов | 4     |
| 4     | Никитина  | Маргарита | Семеновна  | 01.01.2000       | управдом  | 3 800. | Декабристов | 10    |
| 5     | Лебедева  | Юлия      | Сергеевна  | 08.05.2004       | управдом  | 3 800. | Декабристов | 8     |

Запрос «Сотрудники-Стаж» выводит фамилии сотрудников стажем более 3 лет.

| Фамилия   | Дата поступ на раб |
|-----------|--------------------|
| Малинин   | 01.01.2004         |
| Куприянов | 05.08.2002         |
| Лебедева  | 08.05.2004         |

Запрос «Дворники-Дома» выводит фамилии дворников и дома, которые они обслуживают.

| Фамилия  | Должность | Улица       | Номер |
|----------|-----------|-------------|-------|
| Салтыков | дворник   | Декабристов | 4     |
| Салтыков | дворник   | Декабристов | 8     |
| Салтыков | дворник   | Декабристов | 10    |
| Малинин  | дворник   | Восстания   | 22    |
| Малинин  | дворник   | Восстания   | 24    |

Были созданы следующие отчеты.

Отчет «Сотрудники» выводит данные всех сотрудников.

| Таб № | Фамилия   | Имя       | Отчество   | Дата поступ | Должность | Оклад   |
|-------|-----------|-----------|------------|-------------|-----------|---------|
| 1     | Салтыков  | Федор     | Петрович   | 10.06.2001  | дворник   | 3 000р. |
| 2     | Малинин   | Николай   | Сергеевич  | 01.01.2004  | дворник   | 3 000р. |
| 3     | Куприянов | Василий   | Николаевич | 05.08.2002  | сантехник | 3 500р. |
| 4     | Никитина  | Маргарита | Семеновна  | 01.01.2000  | управдом  | 3 800р. |
| 5     | Лебедева  | Юлия      | Сергеевна  | 08.05.2004  | управдом  | 3 800р. |

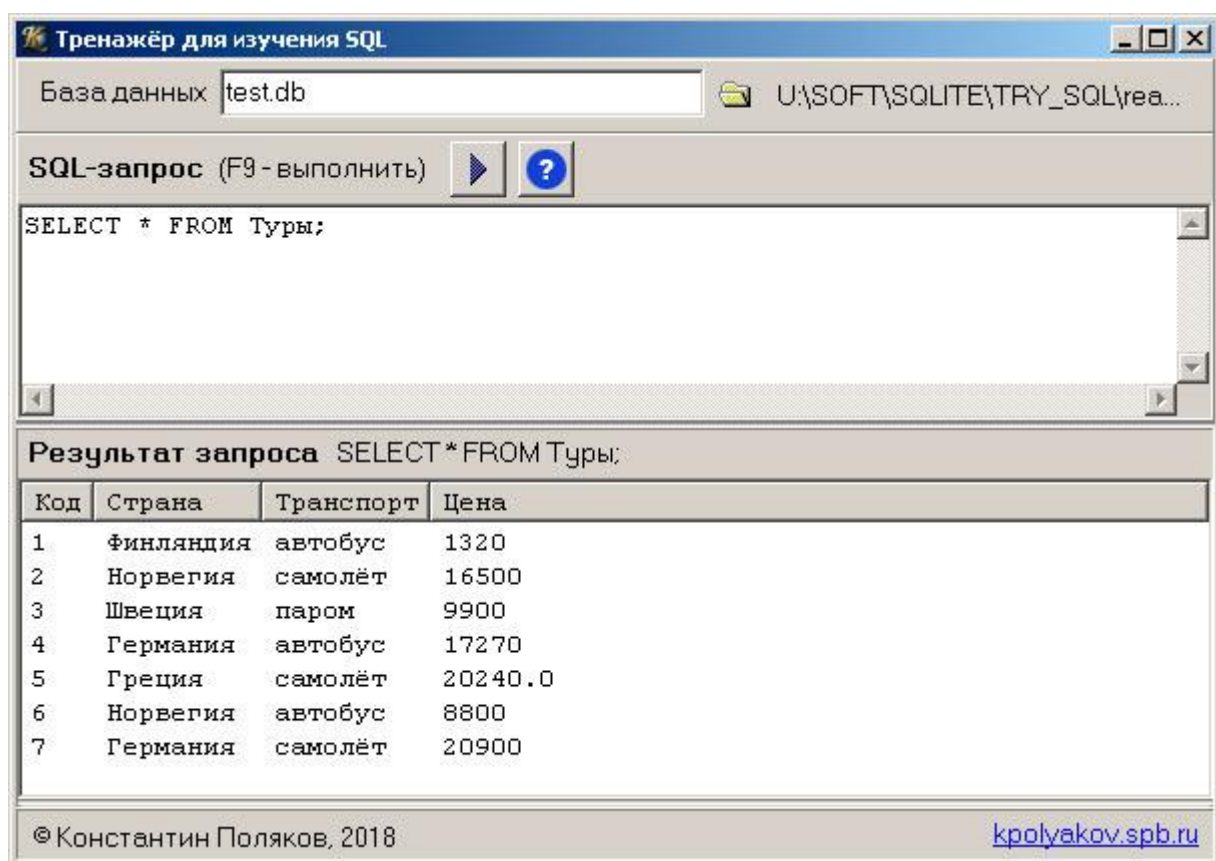
Отчет «Управдомами» выводит данные всех домов, сгруппированных по управляющим домами.

| Фамилия         | Код дома | Улица       | Ном<br>ер | Этаж<br>ность | Кол-во<br>подъез |
|-----------------|----------|-------------|-----------|---------------|------------------|
| <i>Лебедева</i> |          |             |           |               |                  |
|                 | 5        | Восстания   | 24        | 5             | 6                |
|                 | 2        | Декабристов | 8         | 2             | 4                |
| <i>Никитина</i> |          |             |           |               |                  |
|                 | 4        | Восстания   | 22        | 5             | 6                |
|                 | 3        | Декабристов | 10        | 9             | 8                |
|                 | 1        | Декабристов | 4         | 2             | 4                |

### Практическая работа №7. Работа в эмуляторе SQL

#### Что такое Эмулятор SQL?

Программа «Попробуй SQL!» — это тренажёр для экспериментов с простой базой данных при изучении языка SQL. В качестве движка используется встраиваемая СУБД SQLite. Все её функции объединены в одной библиотеке общего доступа `sqlite3.dll`. Её можно скачать, например, с [официального сайта](#). Кроме того, эта библиотека входит в архив с программой, который можно скачать на этой странице (см. далее).



### Как работать с программой?

В верхней части окна вводится имя базы данных. Справа расположена кнопка для выбора каталога, по умолчанию программа работает с базами данных в текущем каталоге. Если базы данных с указанным именем нет в каталоге, она будет создана при первом обращении.

В следующем поле нужно ввести SQL-запрос, возможно, многострочный. При нажатии на клавишу F9 или при щелчке по кнопке  запрос выполняется. Если это был запрос на выборку (**SELECT**), в следующем окне будут показаны выбранные данные в виде таблицы.

Если текущий запрос — это не запрос на выборку (а, например, на удаление или добавление строк), в таблице отображаются обновленные результаты предыдущего запроса на выборку. Например, сначала можно выбрать все данные из какой-то таблицы запросом **SELECT \* FROM имя\_таблицы**. Затем мы с помощью запросов выполняем какие-то операции с таблицей (изменяем, добавляем или удаляем данные), и сразу (без дополнительных запросов) видим результат этих изменений в таблице.

Кнопка  позволяет прочитать метаданные, то есть, получить информацию о структуре базы данных (таблицах, индексах и т.п.). При этом выполняется запрос

**SELECT \* FROM SQLITE\_MASTER;**

Если при выполнении запроса произошла ошибка, в нижней части окна появляется дополнительное поле с сообщением об ошибке (на английском языке).

В архиве, который можно скачать с этой страницы, находится инструкция для выполнения практической работы с тренажёром (11 класс, тема «Базы данных»).

## ПРАКТИЧЕСКАЯ РАБОТА

**ПРИ РАБОТЕ ПРОШУ ЗАКРЕПЛЯТЬ СКРИНШОТЫ КАЖДОГО ВЫПОЛНЕННОГО ЗАДАНИЯ В ДАННЫЙ ФАЙЛ**

### Язык SQL (Microsoft Access)

Язык *SQL* (*Structured Query Language*) – это специальный язык для управления данными. С его помощью можно полностью контролировать базу данных и выполнять все операции, посылая запросы (команды) в текстовом виде. Таким образом, например, можно управлять базой данных с удаленного компьютера.

В этой работе вы познакомитесь с основными командами языка *SQL*:

|                     |                 |
|---------------------|-----------------|
| <b>CREATE TABLE</b> | создать таблицу |
| <b>SELECT</b>       | выбрать данные  |
| <b>UPDATE</b>       | изменить данные |
| <b>DELETE</b>       | удалить данные  |
| <b>DROP</b>         | удалить таблицу |

1. Создайте новую пустую базу данных *SQLbase.accdb*.

Создание и заполнение таблиц

2. Создайте запрос в режиме конструктора и перейдите в режим SQL. Введите следующую команду для создания таблицы:

```
CREATE TABLE [Туры] (  
[Код] INTEGER NOT NULL PRIMARY KEY,  
[Страна] VARCHAR(50) NOT NULL,  
[Транспорт] VARCHAR(20) NOT NULL,  
[Цена] MONEY NOT NULL )
```

В этой команде требуется создать таблицу (**CREATE TABLE**) с именем «Туры». В таблице должно быть 4 поля:

*Код* – целое число (**INTEGER**), непустое (**NOT NULL**), первичный ключ таблицы (**PRIMARY KEY**)

*Страна* – строка длиной до 50 символов, непустое

*Транспорт* – строка длиной до 20 символов, непустое

*Цена* – поле для хранения денежной суммы (**MONEY**)

**Названия таблиц и полей заключаются в квадратные скобки! Если эти названия состоят из одного слова, скобки можно не ставить:**

```
CREATE TABLE Туры (  
Код INTEGER NOT NULL PRIMARY KEY,  
Страна VARCHAR(50) NOT NULL,  
Транспорт VARCHAR(20) NOT NULL,  
Цена MONEY NOT NULL)
```

3. Выполните эту команду (вкладка *Конструктор – Выполнить*). Проверьте, что таблица действительно создана.
4. Выполните команду для добавления в базу новой записи:

```
INSERT INTO Туры  
VALUES (1, 'Финляндия', 'автобус', 1200)
```

Эта команда вставляет (**INSERT**) в таблицу «Туры» одну запись. После ключевого слова **VALUES** в скобках перечислены через запятую значения полей в

том порядке, в котором они задавались при создании таблицы.

**Символьные строки в значениях полей заключаются в апострофы или двойные кавычки!**

5. Выполните ещё одно добавление записи:

**INSERT INTO Туры  
VALUES (1, 'Норвегия', 'самолёт', 15000)**

Какая ошибка произошла? В чём её причина?

Ответ:

Исправьте ошибку и добавьте новую запись правильно. В ответе запишите SQL-запрос.

Ответ:

6. Аналогично добавьте в таблицу ещё несколько записей:

| Страна   | Транспорт | Цена     |
|----------|-----------|----------|
| Швеция   | паром     | 9000 р.  |
| Германия | автобус   | 15700 р. |
| Греция   | самолёт   | 23000 р. |
| Норвегия | автобус   | 8000 р.  |
| Германия | самолёт   | 19000 р. |

Выбор и сортировка данных

7. Выполните запрос на выборку данных:

**SELECT \* FROM Туры**

Посмотрите на результат. Этот оператор выберет все поля (\*) всех записей из таблицы «Туры».

8. Вместо \* можно указать через запятую список нужных полей:

**SELECT Страна, Цена FROM Туры**

Проверьте результат выполнения этого запроса.

9. Чаще всего нужно выбрать только записи, удовлетворяющие некоторому условию отбора. Для этого используется ключевое слово **WHERE**, после которого стоит условие:

**SELECT \* FROM Туры WHERE Страна = 'Норвегия'**

Проверьте работу этого оператора.

10. Составьте запрос, который выбирает из таблицы «Туры» значения полей «Страна», «Транспорт» и «Цена» для всех автобусных туров:

Ответ:

11. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой меньше 10000 руб:

Ответ:

12. Для того, чтобы отсортировать данные по некоторому полю, в запросе после ключевых слов **ORDER BY** (англ. «упорядочить по») указывают название этого поля:

**SELECT \* FROM Туры ORDER BY Цена**

Проверьте работу этого запроса.

Если в конце предыдущего запроса добавить слово **DESC** (англ. «descending» – нисходящий), сортировка выполняется в обратном порядке.

13. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой больше 10000 руб. и сортирует результаты по убыванию цены:

Ответ:

14. В запросах можно использовать стандартные функции. Например, функция **MIN** определяет минимальное значение заданного поля среди всех записей:

**SELECT MIN(Цена) FROM Туры**

Результат этого запроса – одно число.

15. Составьте запрос, который находит минимальную цену для туров в Норвегию:

Ответ:

16. Результаты запросов можно использовать в других запросах – получается вложенный запрос. Например, запрос

**SELECT \* FROM Туры WHERE Цена =  
(SELECT MIN(Цена) FROM Туры WHERE Страна = 'Норвегия')**

вернет данные о самом дешевом туре.

17. Составьте запрос, который находит тур минимальной цены на самолёте:

Ответ:

Изменение и удаление данных

18. Для изменения записей используется оператор **UPDATE**. Запрос, приведенный ниже, увеличивает цены всех туров на 10%:

**UPDATE Туры SET Цена = Цена\*1.1**

Проверьте, что данные в таблице «Туры» действительно изменились.

19. Авиакомпании в данный момент представляют скидку на билеты, так что цены всех туров на самолётах составляют 80% от исходных. Составьте и выполните соответствующий запрос.

Ответ:

Какая стоимость получилась у тура в Грецию?

Ответ:

20. Скопируйте таблицу «Туры», назвав копию «Туры2». Удалите все туры в Германию с помощью запроса

**DELETE FROM Туры2 WHERE Страна = 'Германия'**

Проверьте, что данные в таблице «Туры2» действительно изменились.

21. Удалите таблицу «Туры2», которая больше не нужна, с помощью запроса

**DROP TABLE Туры2**

Проверьте, что таблица «Туры2» удалена из списка таблиц.

### *Практическая работа №8. Связи между таблицами в SQL*

**ПРИ РАБОТЕ ПРОШУ ЗАКРЕПЛЯТЬ СКРИНШОТЫ КАЖДОГО ВЫПОЛНЕННОГО ЗАДАНИЯ В ДАННЫЙ ФАЙЛ**

#### *Создание связей между таблицами*

В этой работе вы познакомитесь с новой командой **ALTER TABLE** (изменить таблицу) языка *SQL* и научитесь составлять запросы к многотабличной реляционной базе данных.

Создание и заполнение таблиц

1. Используя только SQL-запросы, постройте три таблицы для реляционной базы данных кафе (значок  обозначает первичный ключ таблицы):

| Заказы                                                                                  | Заказано     | Блюда                                                                                   |
|-----------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------|
|  Номер | Код          | Код  |
| Дата                                                                                    | Номер заказа | Название                                                                                |
|                                                                                         | Код блюда    | Цена                                                                                    |

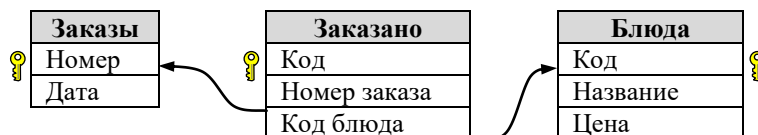
```
CREATE TABLE Заказы (  
[Номер] VARCHAR(3) NOT NULL,  
[Дата] VARCHAR(20) NOT NULL UNIQUE,  
PRIMARY KEY (Номер)  
);
```

```
CREATE TABLE Блюда (  
[Код] SERIAL PRIMARY KEY,  
[Название] VARCHAR(255) NOT NULL UNIQUE,  
[Цена] VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Заказано (  
[Код] SERIAL PRIMARY KEY,  
[Номерзаказа] VARCHAR(20) NOT NULL UNIQUE,  
[Кодблюда] INTEGER NOT NULL,  
FOREIGN KEY (Номерзаказа) REFERENCES Заказы (Номер),  
FOREIGN KEY (Кодблюда) REFERENCES Блюда (Код)  
);
```



2. Заодно мы построим связи между таблицами связи между таблицами:



3. С помощью SQL-запросов заполните базу следующими данными

| Заказы |          |
|--------|----------|
| Номер  | Дата     |
| 1      | 11.04.13 |
| 2      | 12.04.13 |

| Заказано |              |           |
|----------|--------------|-----------|
| Код      | Номер заказа | Код блюда |
| 1        | 1            | 1         |
| 2        | 1            | 3         |
| 3        | 1            | 4         |
| 4        | 2            | 1         |
| 5        | 2            | 2         |
| 6        | 2            | 2         |
| 7        | 2            | 5         |

| Блюда |          |        |
|-------|----------|--------|
| Код   | Название | Цена   |
| 1     | борщ     | 80 р.  |
| 2     | бифштекс | 110 р. |
| 3     | гуляш    | 70 р.  |
| 4     | чай      | 10 р.  |
| 5     | кофе     | 50 р.  |

Выбор данных и сортировка

4. Построим в режиме SQL запрос *СоставЗаказа*, который выводит номер заказа и название заказанных блюд. Эти данные находятся в двух таблицах – *Заказано* и *Блюда*, поэтому их нужно как-то объединить. Для этого используется связь 1:N между таблицами, которую мы недавно установили. Действительно, для каждой записи в таблице *Заказано* нужно выбрать название блюда из таблицы *Блюда*, код которого совпадает с полем *Заказано.Код блюда*.

Это запрос на выборку данных, поэтому используем оператор **SELECT**:

**SELECT "Заказано"."Номер заказа", "Блюда"."Название"**

**FROM "Заказано", "Блюда"**

**WHERE "Заказано"."Код блюда" = "Блюда"."Код"**

Здесь из таблиц *Заказано* и *Блюда* выбираются поля *Номер заказа* и *Название*; условие в последней строке связывает таблицы.

Поскольку названия полей в таблицах, из которых идет выбор, не совпадают, можно было записать запрос в сокращенной форме, указав после оператора **SELECT** только названия нужных полей:

**SELECT "Номер заказа", "Название"**

**FROM "Заказано", "Блюда"**

**WHERE "Заказано"."Код блюда" = "Блюда"."Код"**

5. Теперь добавим в запрос дату заказа. Она находится в таблице *Заказы*, которая пока в запросе не участвует. Таким образом, нам нужно объединить три таблицы. Условие отбора получается сложным, два условия (связи по коду блюда между таблицами *Заказано* и *Блюда* и по номеру заказа между таблицами *Заказы* и *Заказано*) объединяются с помощью логической операции **AND** (И):

**SELECT "Номер заказа", "Дата", "Название"**

**FROM "Заказано", "Блюда", "Заказы"**

**WHERE "Заказано"."Код блюда" = "Блюда"."Код"**

**AND "Заказано"."Номер заказа" = "Заказы"."Номер"**

Проверьте результат выполнения этого запроса.

6. Измените запрос так, чтобы он выбирал только блюда из состава заказа № 1. Запишите этот запрос:

*Ответ:*



7. Построим еще один запрос *Итоги*, в котором для каждого заказа выводится его номер, дата и общая сумма (с помощью функции SUM).

```
SELECT "Номер заказа", "Дата", SUM("Цена")
FROM "Заказано", "Блюда", "Заказы"
WHERE "Заказано"."Код блюда" = "Блюда"."Код"
AND "Заказано"."Номер заказа" = "Заказы"."Номер"
GROUP BY "Номер заказа", "Дата"
```

В последней строке указано, что по полям *Номер заказа* и *Дата* выполняется группировка, то есть сумма цен считается для каждой уникальной пары «*Номер заказа – Дата*».

В таблице с результатами запроса заголовки столбца с суммой выглядят не совсем понятно для пользователя:

«SUM("Блюда"."Цена")»

Для того, чтобы сделать у этого столбца заголовок «*Сумма*», нужно добавить в первую строку запроса после SUM("Цена") так называемый псевдоним (подпись) с ключевым словом AS:

```
SELECT "Номер заказа", "Дата", SUM("Цена") AS "Сумма"
```

...

Проверьте результат выполнения этого запроса. Псевдонимы можно задавать для всех значений, которые выводятся в запросе.

8. Измените запрос так, чтобы заказы были отсортированы в порядке убывания суммы (используйте ключевые слова ORDER BY).

Ответ:

Вложенные запросы

9. Построим запрос *МинСумма*, который выводит минимальную сумму заказа. Для этого будем использовать уже готовый запрос *Итоги*. Таким образом, источником данных для запроса *МинСумма* будет не таблица, а другой запрос. Отметим, что предварительно в запросе *Итоги* нужно отменить сортировку. Запрос получается очень простым

```
SELECT MIN("Сумма") AS "Сумма" FROM "Итоги"
```

10. Наконец, можно вывести информацию о заказе с минимальной суммой:

```
SELECT "Номер заказа", "Дата", "Сумма"
FROM "Итоги", "МинСумма"
WHERE "Итоги"."Сумма" = "МинСумма"."Сумма"
```

Обратите внимание, что этот запрос использует результаты выполнения двух ранее построенных запросов – *Итоги* и *МинСумма*. Запрос *Итоги* можно было и не составлять, а вместо этого использовать вложенный запрос (запрос в запросе):

```
SELECT "Номер заказа", "Дата", "Сумма"
FROM "Итоги"
WHERE "Сумма" =
(SELECT MIN("Сумма") FROM "Итоги")
```

Заметим, что если в базе данных есть информация о нескольких заказах с такой же (минимальной) суммой, будет показана информация обо всех этих заказах.

11. Измените запрос так, чтобы получить список всех заказов, сумма которых больше средней.

Ответ:

**Цель работы:** научиться создавать новую базу данных и освоить способы создания и редактирования таблиц данных.

Прежде всего надо запустить сервер MySQL. Идем в системное меню Пуск — Программы — MySQL — PostgreSQL 10 64bit — SQL Shell. Откроется окно, предлагающее ввести пароль.

Нажимаем Enter на клавиатуре, вводим пароль, в нашем случае 123. Ждем приглашения postgres>.

```
Текущая кодовая страница: 1251
Пароль пользователя postgres:
psql (10.20)
Введите "help", чтобы получить справку.

postgres=#
```

Нам надо создать базу данных, которую мы назовем airshow. Для этого в SQL существует оператор *create database*. Создание базы данных имеет следующий синтаксис:

*create database* имя\_базы\_данных;

Максимальная длина имени БД составляет 64 знака и может включать буквы, цифры, символ «\_» и символ «\$». Имя может начинаться с цифры, но не должно полностью состоять из цифр. Любой запрос к БД заканчивается точкой с запятой (этот символ называется разделителем — delimiter). Получив запрос, сервер выполняет его и в случае успеха выдает сообщение «Query OK ...»

Итак, создадим БД:

```
postgres=# create database airshow;
CREATE DATABASE
```

Вот так все просто. Теперь в этой базе данных нам надо создать трёх пользователей с паролями: admin, operator, reader.

*Create user* имя\_пользователя;

```
postgres=# create user admin with password 'admin';
CREATE ROLE
```

То есть, мы создали пользователя admin с паролем admin.

Создайте самостоятельно еще двоих пользователей: operator с паролем operator, reader с паролем reader.

Следующее, что нам необходимо сделать, это задать этим пользователям права доступа. Для упрощения задачи, сначала пользователю выдаются все права, затем накладываются ограничения. Начнем с администратора.

```
postgres=# grant all privileges on all tables in schema public to "admin";
GRANT
```

Теперь для оператора. У оператора нужно настроить ограничение по праву доступа: он не может удалять и создавать новые таблицы.

```
postgres=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO "operator";
GRANT
postgres=# REVOKE CREATE ON SCHEMA public FROM "operator";
REVOKE
postgres=# REVOKE DELETE, TRUNCATE ON ALL TABLES IN SCHEMA public FROM "operator";
REVOKE
```

Теперь настроим права доступа пользователя:

```
postgres=# GRANT CONNECT ON DATABASE airshow TO "reader";
GRANT
postgres=# GRANT USAGE ON SCHEMA public TO "reader";
GRANT
postgres=# GRANT SELECT ON ALL TABLES IN SCHEMA public TO "reader";
GRANT
```

Дальнейшую работу с базой данных выполняет сам администратор.

Давайте создадим таблицу:

```
Дальнейшие действия выполняются пользователем Администратор БД
-- Создание таблиц
CREATE TABLE countries (
    country_code VARCHAR(3) NOT NULL,
    country_name VARCHAR(255) NOT NULL UNIQUE,
    PRIMARY KEY (country_code)
);
COMMENT ON COLUMN countries.country_code IS 'Код названия страны в стандарте ISO 3166-1 alpha-3';
```

PRIMARY KEY - первичный ключ - это отдельное поле или комбинация полей, которые однозначно определяют запись. Ни одно из полей, являющихся частью первичного ключа, не может содержать значение NULL. Таблица может иметь только один первичный ключ. COMMENT - заметка, в каком стандарте пользователь должен вводить код страны. С помощью этой команды можно задать или изменить комментарий любого объекта. Так же создадим еще таблицы: виды самолетов, типы самолетов, самолеты, виды ремонтных работ.

```
CREATE TABLE aircraft_kinds (
    kind_id SERIAL PRIMARY KEY,
    kind_name VARCHAR(255) NOT NULL UNIQUE
);
COMMENT ON TABLE aircraft_kinds IS 'Виды самолетов: пассажирский, грузовой и т.д.';

CREATE TABLE aircraft_types (
    type_id SERIAL PRIMARY KEY,
    type_name VARCHAR(255) NOT NULL UNIQUE
);
COMMENT ON TABLE aircraft_types IS 'Типы самолетов: Ту-204, Boeing-777';

CREATE TABLE aircrafts (
    id SERIAL PRIMARY KEY,
    kind_id INTEGER NOT NULL,
    type_id INTEGER NOT NULL,
    country_code VARCHAR(3),
    year_of_issue INTEGER NOT NULL,
    flight_hours INTEGER NOT NULL,
    price INTEGER NOT NULL,
    speed INTEGER NOT NULL,
    max_range INTEGER NOT NULL,
    max_payload INTEGER NOT NULL,
```

Для связи между таблицами применяются внешние ключи. Внешний ключ FOREIGN KEY устанавливается для столбца из зависимой, подчиненной таблицы (referencing table), и указывает на один из столбцов из главной таблицы. Как правило, внешний ключ указывает на первичный ключ из связанной главной таблицы.

Установим внешние ключи:

```

FOREIGN KEY (kind_id) REFERENCES aircraft_types (type_id),
FOREIGN KEY (kind_id) REFERENCES aircraft_kinds (kind_id),
FOREIGN KEY (country_code) REFERENCES countries (country_code)
);

```

CREATE INDEX создаёт индексы по указанному столбцу(ам) заданного отношения, которым может быть таблица или материализованное представление. Индексы применяются в первую очередь для оптимизации производительности базы данных (хотя при неправильном использовании возможен и противоположный эффект).

Доработаем таблицу:

```

COMMENT ON TABLE aircrafts IS 'Самолеты и характеристики';
CREATE INDEX aircrafts_year_of_issue_idx
  ON aircrafts (year_of_issue);
CREATE INDEX aircrafts_flight_hours_idx
  ON aircrafts (flight_hours);
CREATE INDEX aircrafts_country_code_idx
  ON aircrafts (country_code);
CREATE INDEX aircrafts_country_payload_idx
  ON aircrafts (country_code, max_payload);
CREATE INDEX aircrafts_price_idx
  ON aircrafts (price);
CREATE INDEX aircrafts_max_payload_idx
  ON aircrafts (max_payload);

```

Создадим таблицу виды ремонтных работ:

```

CREATE TABLE repair_types (
  repair_type_id SERIAL PRIMARY KEY,
  repair_name    VARCHAR(200) NOT NULL UNIQUE
);
COMMENT ON TABLE repair_types IS 'Виды ремонтных работ';

CREATE TABLE repair_works (
  id            SERIAL PRIMARY KEY,
  aircraft_id   INTEGER,
  work_type_id  INTEGER,
  cost          NUMERIC(10, 2) NOT NULL,
  "date"        DATE           NOT NULL,

  FOREIGN KEY (aircraft_id) REFERENCES aircrafts (id),
  FOREIGN KEY (work_type_id) REFERENCES repair_types (repair_type_id)
);
COMMENT ON TABLE repair_works IS 'Ремонтные работы';
CREATE INDEX repair_works_work_type_idx
  ON repair_works (work_type_id);
CREATE INDEX repair_works_cost_idx
  ON repair_works (cost);
CREATE INDEX repair_works_date_idx
  ON repair_works (date);

```

Мы создали таблицы, следующий этап нашей работы – заполнение таблиц.

### ***Практическая работа №10. Запросы ввода данных в таблицы в PostgreSQL***

**Цель работы:** научиться вводить данные в таблицы базы данных.

Для вставки записей в таблицы есть команда INSERT INTO название\_таблицы (код, название столбца), например:

Добавить новую запись в таблицу самолетов aircrafts:

INSERT INTO aircrafts

(kind\_id, type\_id, country\_code, year\_of\_issue, flight\_hours, price, speed, max\_range, max\_payload)

VALUES

(идентификатор вида, идентификатор типа, код страны, год, часы налета, стоимость в \$USA, скорость км/ч, максимальная дистанция, полезная нагрузка).

Таблица Страны:

```
-- Заполнение справочных данных
INSERT INTO countries (country_code, country_name) VALUES
  ('CAN', 'Канада'),
  ('RUS', 'Россия'),
  ('USA', 'США'),
  ('FRA', 'Франция'),
  ('UKR', 'Украина');
```

Таблица Виды самолетов:

```
INSERT INTO aircraft_kinds (kind_id, kind_name) VALUES
  (1, 'Пассажирский'),
  (2, 'Транспортный'),
  (3, 'Сельскохозяйственный');
```

Таблица Типы самолетов:

```
INSERT INTO aircraft_types (type_id, type_name) VALUES
  (1, 'Airbus A220'),
  (2, 'Туполев Ту-134'),
  (3, 'Туполев Ту-154'),
  (4, 'Boeing-777'),
  (5, 'Airbus A330-200F'),
  (6, 'ИЛ-76МД'),
  (7, 'Ан-2'),
  (8, 'Ан-3');
```

Таблица Ремонтных работ:

```

INSERT INTO repair_types (repair_type_id, repair_name)
VALUES
  (1, 'Фюзеляж'),
  (2, 'Крыло'),
  (3, 'Оперение'),
  (4, 'Шасси'),
  (5, 'Силовая установка'),
  (6, 'Тормозная система'),
  (7, 'Навигационный комплекс'),
  (8, 'Система автоматического управления (САУ)'),
  (9, 'Системы оборудования силовых установок (СУ)'),
  (10, 'Система предупреждения о столкновении'),
  (11, 'Система бортового электроснабжения (БЭС)'),
  (12, 'Приборное оборудование'),
  (13, 'Радионавигационное оборудование (РНО)'),
  (14, 'Система кондиционирования (СКВ) и жизнеобеспечения'),
  (15, 'Высотное и кислородное оборудование'),
  (16, 'Аварийно-спасательное оборудование'),
  (17, 'Бытовое оборудование');

```

Заполним информацию о самолетах:

```

-- Заполнение информации о самолетах
INSERT INTO aircrafts (kind_id, type_id, country_code, year_of_issue,
flight_hours, price, speed, max_range, max_payload)
VALUES
  (1, 1, 'CAN', 2016, 10000, 60000000, 850, 4900, 13000),
  (1, 1, 'CAN', 2017, 6000, 76500000, 871, 5741, 15127),
  (1, 2, 'RUS', 1997, 45000, 10000000, 850, 2100, 8500),
  (1, 3, 'RUS', 2000, 10000, 12000000, 900, 2650, 18000),
  (1, 3, 'RUS', 2007, 5000, 20000000, 950, 3000, 18000),
  (2, 5, 'FRA', 2016, 3000, 150000000, 871, 7400, 68600),
  (2, 6, 'RUS', 2011, 32000, 25000000, 800, 4200, 60000),
  (3, 7, 'UKR', 1976, 4000, 200000, 258, 845, 2140),
  (3, 8, 'RUS', 2009, 10000, 350000, 255, 770, 1800);

```

Заполним информацию о ремонтных работах:

```

INSERT INTO repair_works (aircraft_id, work_type_id, cost, date)
VALUES
  (2, 3, 10000, '2017-05-01'),
  (2, 4, 2500, '2017-02-03'),
  (2, 13, 270285, '2018-09-14'),
  (8, 12, 6000, '2018-10-17'),
  (3, 16, 291085, '2017-11-09'),
  (8, 17, 6140, '2017-11-18'),
  (4, 12, 521498, '2018-11-18'),
  (3, 5, 466318, '2018-02-19'),
  (6, 16, 212918, '2017-07-31'),
  (6, 15, 172562, '2017-04-25'),
  (4, 4, 15523, '2018-08-13'),
  (6, 15, 466249, '2017-03-19'),
  (5, 5, 580398, '2018-06-17'),
  (2, 13, 525048, '2017-11-03'),
  (3, 7, 69291, '2017-11-22'),
  (2, 4, 413332, '2018-09-13'),
  (2, 15, 409000, '2018-01-29'),
  (5, 13, 318737, '2018-04-25'),
  (7, 10, 446520, '2017-03-11'),
  (7, 11, 372521, '2017-12-14'),
  (4, 16, 604561, '2018-01-10'),
  (7, 17, 548845, '2017-03-27');

```

### ***Практическая работа №11. Запросы выборки данных в PostgreSQL***

**Цель работы:** научиться выводить только нужные данные путем создания запросов в базе данных.

Для каждого вида самолета выдать список, отсортированный:

- по году выпуска,
- по стоимости ремонтных работ,
- по цене производителя,
- по сроку эксплуатации.



```

SELECT
    a.id,
    k.kind_name,
    t.type_name,
    a.country_code,
    a.year_of_issue,
    a.flight_hours,
    a.price,
    a.speed,
    a.max_range,
    a.max_payload,
    coalesce(sum(rw.cost), 0) AS sum_repair_cost
FROM aircrafts a
    JOIN aircraft_kinds k ON k.kind_id = a.kind_id
    JOIN aircraft_types t ON t.type_id = a.type_id
    LEFT JOIN repair_works rw ON a.id = rw.aircraft_id
WHERE a.kind_id = 1 -- вид "Пассажирский"
GROUP BY a.id, k.kind_name, t.type_name, a.country_code, a.year_of_issue,
a.flight_hours, a.price, a.speed, a.max_range,
a.max_payload
ORDER BY sum_repair_cost; -- по стоимости ремонтных работ

```

По запросу должно вывестись:

| id | kind_name    | type_name      | country_code | year_of_issue | flight_hours | price    | speed | max_range | max_payload | sum_repair_cost |
|----|--------------|----------------|--------------|---------------|--------------|----------|-------|-----------|-------------|-----------------|
| 1  | Пассажирский | Airbus A220    | CAN          | 2016          | 10000        | 60000000 | 850   | 4900      | 13000       | 0               |
| 3  | Пассажирский | Туполев Ту-134 | RUS          | 1997          | 45000        | 10000000 | 850   | 2100      | 8500        | 826694.00       |
| 5  | Пассажирский | Туполев Ту-154 | RUS          | 2007          | 5000         | 20000000 | 950   | 3000      | 18000       | 899135.00       |
| 4  | Пассажирский | Туполев Ту-154 | RUS          | 2000          | 10000        | 12000000 | 900   | 2650      | 18000       | 1141582.00      |
| 2  | Пассажирский | Airbus A220    | CAN          | 2017          | 6000         | 76500000 | 871   | 5741      | 15127       | 1630165.00      |

5 строк)

Найти долю «отремонтированных» самолетов от общего числа самолетов.

```

SELECT round(repaired_aircrafts :: DECIMAL / total_aircrafts, 2) AS
repaired_rate
FROM
    (SELECT count(DISTINCT aircraft_id) AS repaired_aircrafts FROM
    repair_works) AS rw,
    (SELECT count(id) AS total_aircrafts FROM aircrafts) AS a;

```

По запросу должно вывестись:

```

repaired_rate
-----
0.78

```

Найти самый дорогой самолет, самый дешевый, среднюю стоимость.



```

WITH filters AS (SELECT max(price) AS max_price, min(price) AS min_price,
round(avg(price),2) AS average_price FROM aircrafts)
SELECT aircraft_data.id, t.type_name, aircraft_data.price,
aircraft_data.filter FROM
(SELECT a_max.id, a_max.type_id, a_max.price, 'max' as filter FROM
aircrafts a_max
WHERE a_max.price = (SELECT max_price FROM filters)
UNION
SELECT a_min.id, a_min.type_id, a_min.price, 'min' as filter FROM
aircrafts a_min
WHERE a_min.price = (SELECT min_price FROM filters)
UNION
SELECT * FROM (VALUES(-1, -1, (SELECT average_price FROM filters),
'average')) as val
) AS aircraft_data LEFT JOIN aircraft_types t on t.type_id =
aircraft_data.type_id;

```

На экран должно вывестись:

| id | type_name        | price       | filter  |
|----|------------------|-------------|---------|
| 6  | Airbus A330-200F | 150000000   | max     |
| 8  | Ан-2             | 200000      | min     |
| -1 |                  | 39338888.89 | average |

(3 строки)

Найти самолеты с ценой свыше 8000 т.р. (и любая другая сумма, т.е. предусмотреть ввод цены).

```

SELECT
a.id,
k.kind_name,
t.type_name,
a.country_code,
a.year_of_issue,
a.flight_hours,
a.price,
a.speed,
a.max_range,
a.max_payload
FROM aircrafts a
JOIN aircraft_kinds k ON k.kind_id = a.kind_id
JOIN aircraft_types t ON t.type_id = a.type_id
WHERE a.price > 15000000 -- цена
ORDER BY a.price;

```

На экран должно вывестись:

| id | kind_name    | type_name        | country_code | year_of_issue | flight_hours | price     | speed | max_range | max_payload |
|----|--------------|------------------|--------------|---------------|--------------|-----------|-------|-----------|-------------|
| 5  | Пассажирский | Туполев Ту-154   | RUS          | 2007          | 5000         | 20000000  | 950   | 3000      | 18000       |
| 7  | Транспортный | ИЛ-76МД          | RUS          | 2011          | 32000        | 25000000  | 800   | 4200      | 60000       |
| 1  | Пассажирский | Airbus A220      | CAN          | 2016          | 10000        | 60000000  | 850   | 4900      | 13000       |
| 2  | Пассажирский | Airbus A220      | CAN          | 2017          | 6000         | 76500000  | 871   | 5741      | 15127       |
| 6  | Транспортный | Airbus A330-200F | FRA          | 2016          | 3000         | 150000000 | 871   | 7400      | 68600       |

(5 строк)

Найти все самолеты, поступившие из заданной страны (выбор страны), чья

вместительность (ввод) больше заданной:

```
SELECT
    a.id,
    k.kind_name,
    t.type_name,
    a.country_code,
    a.year_of_issue,
    a.flight_hours,
    a.price,
    a.speed,
    a.max_range,
    a.max_payload
FROM aircrafts a
    JOIN aircraft_kinds k ON k.kind_id = a.kind_id
    JOIN aircraft_types t ON t.type_id = a.type_id
WHERE a.country_code = 'RUS' -- страна
    and a.max_payload > 15000 -- вместимость
ORDER BY t.type_name;
```

На экран должно вывестись:

| id | kind_name    | type_name      | country_code | year_of_issue | flight_hours | price    | speed | max_range | max_payload |
|----|--------------|----------------|--------------|---------------|--------------|----------|-------|-----------|-------------|
| 7  | Транспортный | ИЛ-76МД        | RUS          | 2011          | 32000        | 25000000 | 800   | 4200      | 60000       |
| 4  | Пассажирский | Туполев Ту-154 | RUS          | 2000          | 10000        | 12000000 | 900   | 2650      | 18000       |
| 5  | Пассажирский | Туполев Ту-154 | RUS          | 2007          | 5000         | 20000000 | 950   | 3000      | 18000       |

(3 строки)

Найти количество самолетов на авиасалоне (по каждому типу и в целом) с заданным сроком эксплуатации (ввод срока эксплуатации):

```
SELECT t.type_name, a.flight_hours, count(a.type_id)
FROM aircrafts a JOIN aircraft_types t on t.type_id = a.type_id
WHERE a.flight_hours = 10000
GROUP BY a.type_id, t.type_name, a.flight_hours;

SELECT count(*)
FROM aircrafts a
WHERE a.flight_hours = 10000;
```

На экран должно вывестись:

| type_name      | flight_hours | count |
|----------------|--------------|-------|
| Airbus A220    | 10000        | 1     |
| Туполев Ту-154 | 10000        | 1     |
| Ан-3           | 10000        | 1     |

(3 строки)

Для заданного типа самолета (выбор типа самолета) найти все, выпущенные за определенный период (ввод), чья стоимость находится в заданных пределах (ввод интервала):

```
SELECT a.id, t.type_name, a.year_of_issue, a.price
FROM aircrafts a JOIN aircraft_types t on t.type_id = a.type_id
WHERE a.type_id = 3
      AND a.year_of_issue BETWEEN 1990 AND 2018
      AND a.price BETWEEN 1000000 AND 50000000;
```

На экран должно вывестись:

| id | type_name      | year_of_issue | price    |
|----|----------------|---------------|----------|
| 4  | Туполев Ту-154 | 2000          | 12000000 |
| 5  | Туполев Ту-154 | 2007          | 20000000 |

(2 строки)

Количество отремонтированных машин за месяц, квартал; средняя стоимость ремонтных работ за этот же период:

```
SELECT
  EXTRACT(YEAR FROM rw.date) AS repair_year,
  EXTRACT(QUARTER FROM rw.date) AS repair_quarter,
  count(DISTINCT aircraft_id) AS repair_aircrafts_count,
  round(avg(rw.cost), 2) AS repair_avg_cost
FROM repair_works rw
GROUP BY repair_year, repair_quarter
ORDER BY repair_year, repair_quarter;

-- По месяцам
SELECT
  EXTRACT(YEAR FROM rw.date) AS repair_year,
  EXTRACT(MONTH FROM rw.date) AS repair_month,
  count(DISTINCT aircraft_id) AS repair_aircrafts_count,
  round(avg(rw.cost), 2) AS repair_avg_cost
FROM repair_works rw
GROUP BY repair_year, repair_month
ORDER BY repair_year, repair_month;
```

На экран должно вывестись:

| repair_year | repair_month | repair_aircrafts_count | repair_avg_cost |
|-------------|--------------|------------------------|-----------------|
| 2017        | 2            | 1                      | 2500.00         |
| 2017        | 3            | 2                      | 487204.67       |
| 2017        | 4            | 1                      | 172562.00       |
| 2017        | 5            | 1                      | 10000.00        |
| 2017        | 7            | 1                      | 212918.00       |
| 2017        | 11           | 3                      | 222891.00       |
| 2017        | 12           | 1                      | 372521.00       |
| 2018        | 1            | 2                      | 506780.50       |
| 2018        | 2            | 1                      | 466318.00       |
| 2018        | 4            | 1                      | 318737.00       |
| 2018        | 6            | 1                      | 580398.00       |
| 2018        | 8            | 1                      | 15523.00        |
| 2018        | 9            | 1                      | 341808.50       |
| 2018        | 10           | 1                      | 6000.00         |
| 2018        | 11           | 1                      | 521498.00       |

(15 строк)

Найти долю дешевых (по стоимости ремонтных работ) самолетов (меньше заданного, ввод ограничения) от общего числа, подлежащих ремонту:

```

SELECT
    count(DISTINCT rw.aircraft_id)
AS repair_aircrafts_filtered,
    (SELECT count(DISTINCT aircraft_id)
     FROM repair_works)
AS repair_aircraft_total,
    round(count(DISTINCT rw.aircraft_id) :: DECIMAL / (SELECT count(DISTINCT
aircraft_id)
                                                    FROM repair_works), 2)
AS repair_aircraft_rate
FROM repair_works rw
WHERE rw.cost < 100000;

```

На экран должно вывестись:

```

repair_aircrafts_filtered | repair_aircraft_total | repair_aircraft_rate
-----+-----+-----
                        4 |                      7 |                   0.57
(1 строка)

```

### ***Практическая работа №12. Анализ плана запроса - EXPLAIN в PostgreSQL***

В PostgreSQL предусмотрена команда для анализа плана запроса.

Команда EXPLAIN [ANALYZE] показывает, каким образом PostgreSQL собирается выполнять запрос. Команда EXPLAIN ANALYZE запрос[] выполняет запрос (например EXPLAIN ANALYZE DELETE ... будет выполнен) и показывает как изначальный план, так и реальный процесс его выполнения. Уверенное чтение вывода этих команд требует детального понимания принципов работы СУБД.

Внимание нужно обратить на следующее:

Использование полного просмотра таблицы (seq scan);

Использование наиболее примитивного способа объединения таблиц (nested loop);

Для EXPLAIN ANALYZE: нет ли больших отличий в предполагаемом количестве записей и реально выбранном? Если оптимизатор использует устаревшую статистику, то он может выбирать не самый быстрый план выполнения запроса;

Полный просмотр таблицы (seq scan) далеко не всегда медленнее просмотра по индексу. Если, например, в таблице-справочнике несколько сотен записей, уместяющихся в одном-двух блоках на диске, то использование индекса приведёт лишь к тому, что придётся читать ещё и пару лишних блоков индекса. Если в запросе придётся выбрать 80% записей из большой таблицы, то полный просмотр опять же получится быстрее.

При тестировании запросов с использованием EXPLAIN ANALYZE можно воспользоваться настройками, запрещающими оптимизатору использовать определённые планы выполнения. Например,

SET enable\_seqscan=false;

запретит использование полного просмотра таблицы.

Пример анализа запроса:

## EXPLAIN ANALYSE

### SELECT

```
a.id, k.kind_name, t.type_name, a.country_code, a.year_of_issue,
a.flight_hours, a.price, a.speed, a.max_range, a.max_payload
FROM aircrafts a
JOIN aircraft_kinds k ON k.kind_id = a.kind_id
JOIN aircraft_types t ON t.type_id = a.type_id
WHERE a.country_code = 'RUS' -- страна
      and a.max_payload > 15000 -- вместимость
ORDER BY t.type_name;
```

Результат:

```
QUERY PLAN
-----
Sort (cost=13.84..13.84 rows=1 width=1076) (actual time=0.056..0.057 rows=3 loops=1)
  Sort Key: t.type_name
  Sort Method: quicksort  Memory: 25kB
  -> Nested Loop (cost=1.96..13.83 rows=1 width=1076) (actual time=0.035..0.042 rows=3 loops=1)
    -> Nested Loop (cost=1.82..10.62 rows=1 width=564) (actual time=0.024..0.028 rows=3 loops=1)
      -> Bitmap Heap Scan on aircrafts a (cost=1.67..4.25 rows=2 width=52) (actual time=0.014..0.015 rows=3 loops=1)
        Recheck Cond: (((country_code)::text = 'RUS'::text) AND (max_payload > 15000))
        Heap Blocks: exact=1
        -> Bitmap Index Scan on aircrafts_country_payload_idx (cost=0.00..1.67 rows=2 width=0) (actual time=0.011..0.011 rows=3 loops=1)
          Index Cond: (((country_code)::text = 'RUS'::text) AND (max_payload > 15000))
      -> Index Scan using aircraft_types_pkey on aircraft_types t (cost=0.14..3.16 rows=1 width=520) (actual time=0.003..0.003 rows=1 loops=3)
        Index Cond: (type_id = a.type_id)
    -> Index Scan using aircraft_kinds_pkey on aircraft_kinds k (cost=0.14..3.16 rows=1 width=520) (actual time=0.004..0.004 rows=1 loops=3)
      Index Cond: (kind_id = a.kind_id)
Planning time: 1.740 ms
Execution time: 0.104 ms
(16 строк)
```