

# Базы данных

01.06.2021 11:20 – 12:50 13:00 – 14:30 03.06.2021 14:40 – 16:10 16:20 – 17:50	Лабораторные работы
	Защита КП
19.06.2021 8:00	Экзамен

## Задание на 01.06.2021

**Для своего варианта выполнить:**

Инфологическое проектирование.

Анализ предметной области. (12,13,14)

Анализ информационных задач и круга пользователей системы (15).

Выбор СУБД и других программных средств (17).

Логическое проектирование базы данных.

Преобразование ER-диаграммы в схему базы данных(18,19,20).

Составление реляционных отношений (22-27).

Нормализация базы данных (28-32).

**Файл с выполненным заданием назвать так: ФамилияИО ЛР1**

## Задание на 03.06.2020

Физическое проектирование базы данных

Создание таблиц.(34)

Создание представлений (готовых запросов).(35)

Назначение прав доступа.(36)

Создание индексов.(37)

Разработка стратегий резервного копирования.(37)

**Файл с выполненным заданием назвать так: ФамилияИО ЛР2**

**Файлы выслать на почту: [Olga\\_Beznos@mail.ru](mailto:Olga_Beznos@mail.ru)**

**КП называть так:**

Фамилия И.

ПЗ к курсовой работе должна содержать 25–40 страниц текста на листах формата А4.

ПЗ должна состоять из следующих структурных элементов:

- титульный лист;
- задание на курсовую;
- реферат;
- содержание;
- введение;
- нормативные ссылки;
- термины и определения;
- сокращения;
- основная часть;
- заключение;
- список использованных источников;
- приложения.

# Титульный лист

Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО «Кубанский государственный технологический университет»  
(ФГБОУ ВО «КубГТУ»)

Институт \_\_\_\_\_

Кафедра \_\_\_\_\_

Направление подготовки/специальность \_\_\_\_\_  
(код и наименование направления подготовки/специальности)

Профиль/специализация \_\_\_\_\_  
(наименование профиля/специализации)

## КУРСОВАЯ РАБОТА

по дисциплине \_\_\_\_\_  
(наименование дисциплины)

на тему: « \_\_\_\_\_ »  
(тема курсовой работы)

Выполнил(-а) студент(-ка) \_\_\_\_\_ курса \_\_\_\_\_ группы \_\_\_\_\_  
(фамилия, имя, отчество)

Допущена к защите \_\_\_\_\_  
(дата)

Руководитель (нормоконтролер) проекта \_\_\_\_\_ Фамилия И.О.  
(должность, подпись)

Защищена \_\_\_\_\_ Оценка \_\_\_\_\_  
(дата)

Члены комиссии \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
(должность, подпись, дата, расшифровка подписи)

Краснодар  
20\_\_ г.

# Задание

Задание на курсовую работу помещается после титульного листа, выдаётся преподавателем, ведущим дисциплину и утверждается заведующим кафедрой.

Заполнить по примеру,  
поставить свою подпись

Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Остапенко Н.М.			Проектирование базы данных «Компьютерный магазин» Пояснительная записка	Лит.	Лист	Листов
Провер.		Безнос О.С.					4	30
Консультант		Безнос О.С.				Кафедра ИВТ Группа 17-ЗКБс-ИВ2		
Н. Контр.		Безнос О.С.						
Утв.		Атрощенко В.А.						

## Реферат

Реферат представляет собой краткое изложение сущности курсовой работы объемом 0,5–0,75 страницы. Текст реферата должен отражать объект исследования, цель работы, методы исследования, полученные результаты.

В начале реферата даются сведения о количестве страниц, иллюстраций, таблиц, использованных источников, приложений, например: работа 33 с., 7 рис., 13 табл., 25 источников, 4 приложения. После этого приводятся ключевые слова и словосочетания (от 5 до 15), взятые из текста курсовой работы, которые в наибольшей степени характеризуют его содержание и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже и пишутся прописными буквами в строку через запятые. Реферат выполняется в соответствии с ГОСТ 7.9.

### Пример оформления реферата

Курсовая работа: 45 с., 13 рис., 2 табл., 15 источников

БАЗА ДАННЫХ, СУБД, НОРМАЛИЗАЦИЯ, ER-МОДЕЛЬ,  
ЗАПРОС, ОТЧЕТ, ФОРМА, КНОПОЧНАЯ ФОРМА, ТАБЛИЦА, SQL.

Предмет разработки – реляционная база данных.

Цель работы – разработка программного продукта, представляющего собой базу данных для организации работы аптечного склада с использованием СУБД MSAccess.

Методы разработки – проектирование базы данных на основе ER-подхода, составление графической модели базы данных, реализация базы данных в СУБД Microsoft Access.

## **Содержание**

В содержании перечисляют заголовки всех разделов и подразделов с указанием номеров страниц, на которых помещены эти заголовки. Слово «Содержание» записывают в виде заголовка (симметрично тексту). Наименования, включенные в содержание, записывают строчными буквами, начиная с прописной буквы. Названия разделов в содержании должны быть точной копией названий в тексте.

## **Введение**

Во введении следует кратко изложить современное состояние технического вопроса, которому посвящена курсовая работа, а также цель работы. Здесь же необходимо сформулировать, в чем заключается новизна и актуальность работы, и обосновать необходимость ее проведения. Заглавием должно служить слово «Введение» (симметрично тексту, без кавычек).

Объем введения составляет 1–2 страницы.

## **Нормативные ссылки**

В нормативных ссылках указываются индекс, номер, год разработки и полное название всех стандартов, на которые имеются ссылки в пояснительной записке. Список нормативных ссылок размещают в порядке возрастания регистрационных номеров обозначений. В тексте пояснительной записки указывают только индекс и номер стандарта без года разработки.

## **Термины и определения**

Структурный элемент «Термины и определения» приводят при необходимости терминологического взаимопонимания читателей ПЗ путем определения нестандартизованных терминов или уточнения стандартизованных терминов, если они используются в более узком смысле. Термины располагаются в алфавитном порядке или в том порядке, в котором они встречаются в тексте. Каждая терминологическая статья нумеруется.

## **Сокращения**

В принятых сокращениях указываются сокращения, принятые в данной пояснительной записке.

## Основная часть

Рекомендуется придерживаться следующей структуры основной части пояснительной записки:

Инфологическое проектирование.

- Анализ предметной области.

- Анализ информационных задач и круга пользователей системы.

Определение требований к операционной обстановке.

Выбор СУБД и других программных средств.

Логическое проектирование базы данных.

- Преобразование ER-диаграммы в схему базы данных.

- Составление реляционных отношений.

- Нормализация базы данных.

- Определение дополнительных ограничений целостности.

- Описание групп пользователей и прав доступа.

Физическое проектирование базы данных

- Создание таблиц.

- Создание представлений (готовых запросов).

- Назначение прав доступа.

- Создание индексов.

- Разработка стратегий резервного копирования.

## **Заключение**

Заключение представляет обобщение проведенной работы. Выводы и предложения (при наличии) должны логично вытекать из 2 проведенных исследований. Объем раздела 1–2 страницы.

## **Список использованных источников**

Список использованных источников должен содержать литературные источники, использованные автором при выполнении курсовой работы.

Данный список оформляется в соответствии с ГОСТ 7.80, ГОСТ 7.82 и ГОСТ 7.0.5.

## **Приложения**

Приложения содержат вспомогательный материал, который нецелесообразно включать в основные разделы. Каждое приложение должно иметь заголовок, который записывают симметрично относительно текста. Приложения обозначают заглавными буквами русского алфавита, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. После слова «Приложение» следует буква, обозначающая его последовательность.

## Требования к оформлению текста пояснительной записки курсовой работы

Пояснительная записка выполняется на компьютере в текстовом редакторе Microsoft Office Word. При оформлении пояснительной записки необходимо следовать следующим требованиям:

- Times New Roman, размер 14 пт;
- шрифт – интервал между строками – одинарный;
- интервал перед названием параграфа и после него – одинарный;
- выполнение каждого задания курсовой работы начинать с нового листа.

Нижний колонтитул должен содержать номер страницы, который расположен в правом нижнем углу. На титульном листе, задании на курсовую работу, реферате номер страницы не ставят, но в общую нумерацию включают.

Абзацный отступ по всему текстовому документу должен быть равен 1,5 см.

Поля текстового набора:

- верхнее – 15 мм;
- левое – 30 мм;
- нижнее – 25 мм;
- правое – 15 мм;
- ориентация – книжная.

Рисунки, таблицы, графики и диаграммы вставляются в текст при следующих параметрах:

- размер рисунка по горизонтали – половина длины строки;
- обтекание текстом – «вокруг рамки»;
- отступ от текста – 10 мм;
- заливки – нет;
- рамки – нет.

Текст основной части ПЗ разделяется на разделы и подразделы. Разделы должны иметь порядковые номера, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и подраздела, разделённых точкой. В конце подраздела точка не ставится.

Разделы и подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов и подразделов. Заголовки разделов и подразделов следует писать с прописной буквы (для разделов полужирное начертание), не подчеркивая, с абзацного отступа, без точки в конце. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

## Пример проектирования базы данных

В качестве примера возьмем базу данных проектной организации. Основной вид деятельности такой организации – выполнение проектов по договорам с заказчиками.

## Инфологическое проектирование

### Анализ предметной области

База данных создаётся для информационного обслуживания руководства организации, руководителей проектов и участников проектов. БД должна содержать данные об отделах организации, сотрудниках и проектах.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- каждый сотрудник работает в определённом отделе, в каждом отделе могут работать несколько сотрудников;
- каждый проект относится к определённому отделу, каждый отдел может отвечать за выполнение нескольких проектов;
- каждый сотрудник может принимать участие в выполнении нескольких проектов, над каждым проектом может трудиться несколько сотрудников;
- для каждого проекта назначается руководитель из числа сотрудников того отдела, к которому относится проект;
- каждый проект должен быть выполнен в заданные сроки, каждый проект может состоять из нескольких этапов. Если проект состоит из одного этапа, то сроки его выполнения должны совпадать со сроками выполнения проекта в целом;
- оклад сотрудника зависит от занимаемой должности, за участие в проектах сотрудник получает дополнительное вознаграждение;
- виды участия сотрудников в проектах: руководитель, консультант, исполнитель;
- каждый отдел занимает одно или несколько помещений (комнат), в каждом помещении может быть один или несколько стационарных телефонов.

**Примечание** – Описание особенностей ПрО должно быть достаточно для того, чтобы создать ER-диаграмму.

Для создания ER-модели необходимо выделить сущности предметной области:

- **Отделы.** Атрибуты: название, аббревиатура, комнаты, телефоны.
- **Сотрудники.** Атрибуты: ФИО, паспортные данные, дата рождения, пол, ИНН (индивидуальный номер налогоплательщика), номер пенсионного страхового свидетельства, адреса, телефоны (рабочий, домашний, мобильный), данные об образовании (вид образования (высшее, среднеспециальное и т. д.), специальность, номер диплома, дата окончания учебного заведения), должность, оклад, логин (имя пользователя).

**Примечания:** 1 Логин потребуется нам для назначения дифференцированных прав доступа. 2 В нашем задании не предусмотрена полная информационная поддержка сотрудников отдела кадров, поэтому мы не будем отражать в БД такие сведения, как дату поступления сотрудника на работу, его переводы с одной должности на другую, уходы в отпуска и т. п.

- **Проекты.** Атрибуты: номер договора; полное название проекта; сокращённое название проекта; дата подписания договора; заказчик; контактные данные заказчика; дата начала проекта; дата завершения проекта; сумма по проекту; дата реальной сдачи проекта; сумма, полученная по проекту на текущую дату.
- **Этапы проекта.** Атрибуты: номер по порядку, название, дата начала этапа, дата завершения этапа, форма отчетности, сумма по этапу, дата реальной сдачи этапа; сумма, полученная по этапу на текущую дату.

Исходя из выявленных сущностей построим ER-диаграмму

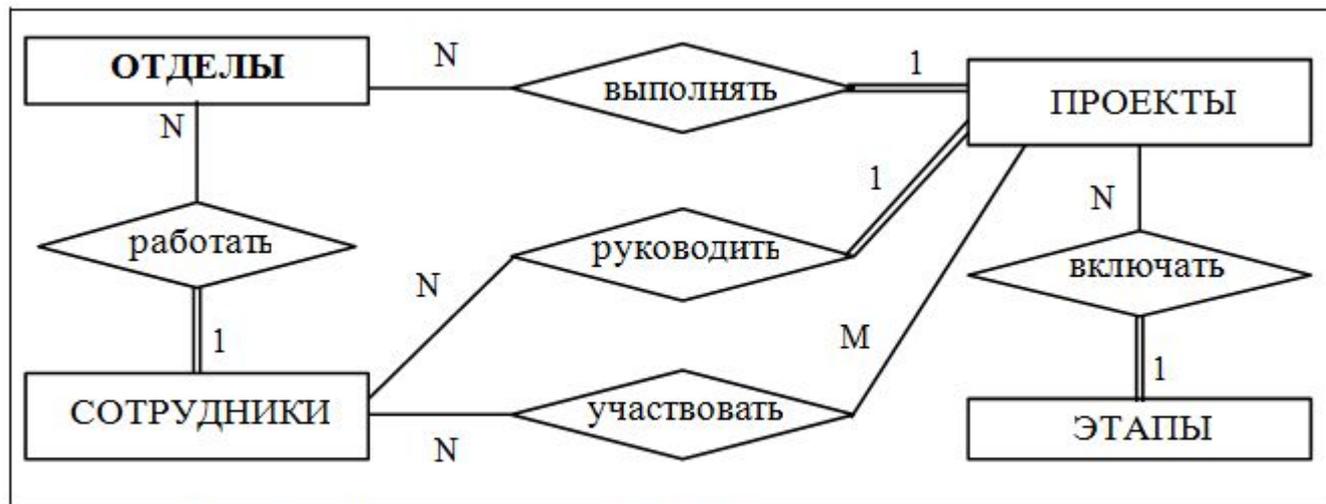


Рис. 2. ER-диаграмма ПрО «Проектная организация»

## **Анализ информационных задач и круга пользователей системы**

Определим группы пользователей, их основные задачи и запросы к БД:

### 1. Руководители организации:

- заключение новых договоров;
- назначение руководителей проектов;
- получение списка всех участников проектов;
- изменение должностных окладов и штатного расписания;
- получение полной информации о проектах;
- внесение изменений в данные о проектах;
- архивирование данных по завершённым проектам.

### 2. Руководитель проекта:

- назначение участников проекта;
- получение списка сотрудников, работающих над конкретным проектом;
- получение полной информации о проекте, руководителем которого он является;
- получение сведений о сотрудниках, которые могут стать участниками проекта;
- определение размера дополнительного вознаграждения сотрудников по конкретному проекту;
- внесение изменений в данные об этапах проекта.

### 3. Сотрудники отдела кадров:

- приём/увольнение сотрудников;
- внесение изменений в данные о сотрудниках.

### 4. Бухгалтеры: получение ведомости на выплату зарплаты.

### 5. Сотрудники – участники проектов:

- просмотр данных о других участниках проекта;
- просмотр данных о сроках сдачи проекта и форме отчётности.

## Определение требований к операционной обстановке

Для выполнения этого этапа необходимо знать (хотя бы ориентировочно) объём работы организации (т. е. количество проектов и сотрудников), а также иметь представление о характере и интенсивности запросов.

Объём внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные ( $M_D$ ). Для реальных баз данных обычно наиболее существенным является  $M_D$ .

На основе результатов анализа ПрО можно приблизительно оценить объём памяти, требуемой для хранения данных. Примем ориентировочно, что:

- одновременно осуществляется около десяти проектов, работа над проектом продолжается в среднем 1 год (по 1 К на каждый проект);
- каждый проект состоит в среднем из четырёх этапов (по 0,5 К на этап);
- в компании работают 100 сотрудников (по 0,5 К на каждого сотрудника);
- в выполнении каждого проекта в среднем участвуют 10 сотрудников (по 0,2 К);
- устаревшие данные переводятся в архив (накапливаются в архиве БД). Тогда объём памяти для хранения данных за первый год примерно составит:

$$M_D = 2(10*1+10*4*0,5+100*0,5+(10*10*0,2)) = 200 \text{ К.}$$

Коэффициент 2 необходим для того, чтобы учесть необходимость выделения памяти под дополнительные структуры (например, индексы). Объём памяти будет увеличиваться ежегодно на столько же при сохранении объёма работы.

Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных. Для нашей БД требуемый объём памяти мал, поэтому никаких специальных требований к объёму внешней и оперативной памяти компьютера не предъявляется.

## **Выбор СУБД и других программных средств**

Анализ информационных задач показывает, что для реализации требуемых функций подходят почти все СУБД для ПЭВМ (MS Access, Firebird, MySQL и др.). Все они поддерживают реляционную модель данных и предоставляют разнообразные возможности для работы с данными.

Объём внешней и оперативной памяти, требующийся для функционирования СУБД, обычно указывается в сопроводительной документации.

Для того чтобы в учебном примере не привязываться к конкретной СУБД, выполним описание логической схемы БД на SQL-92.

## Логическое проектирование реляционной БД

### Преобразование ER-диаграммы в схему базы данных

База данных создаётся на основании схемы базы данных. Для преобразования ER-диаграммы в схему БД приведём уточнённую ER-диаграмму, содержащую атрибуты сущностей (рис. 3).

Преобразование ER-диаграммы в схему БД путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. Связь типа 1:n (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь. Внешнему ключу должен соответствовать первичный или уникальный ключ основного (родительского) отношения.

Связь участвовать между ПРОЕКТАМИ и СОТРУДНИКАМИ принадлежит к типу n:m (многие-ко-многим). Этот тип связи реализуется через вспомогательное отношение Участие, которое содержит комбинации первичных ключей соответствующих исходных отношений.

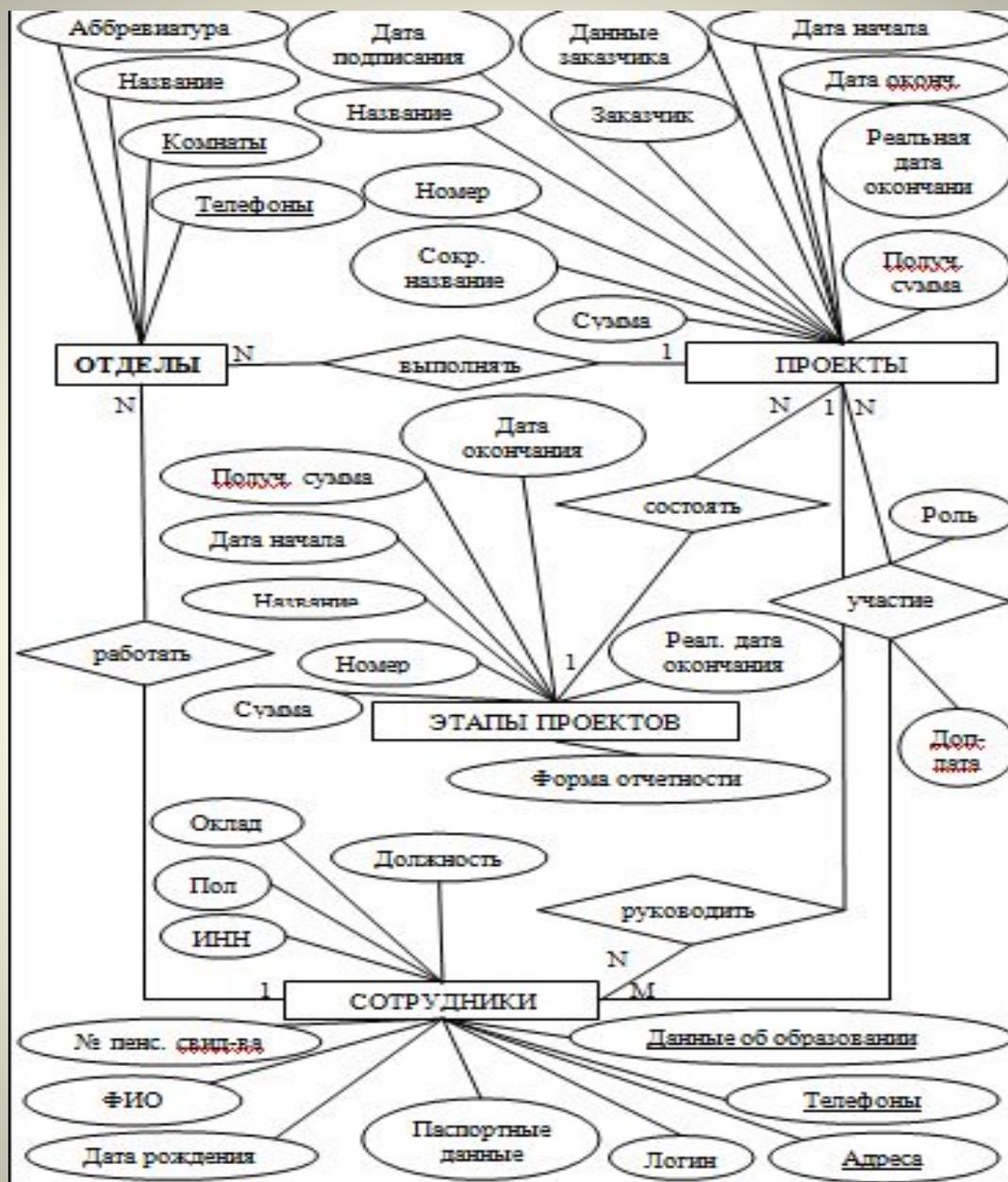


Рис.3. Уточненная ER-диаграмма проектной организации

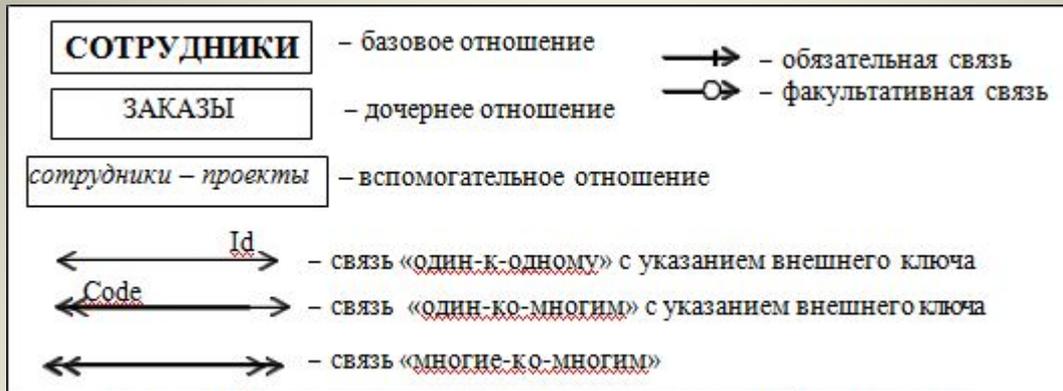


Рис. 4. Обозначения, используемые на схеме базы данных

Полученная схема реляционной базы данных (РБД) приведена на рисунке 5

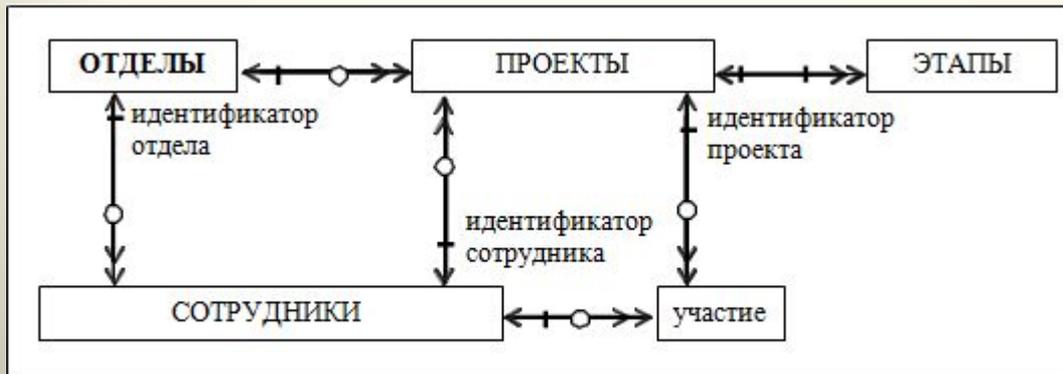


Рис. 5. Схема РБД, полученная из ER-диаграммы проектной организации

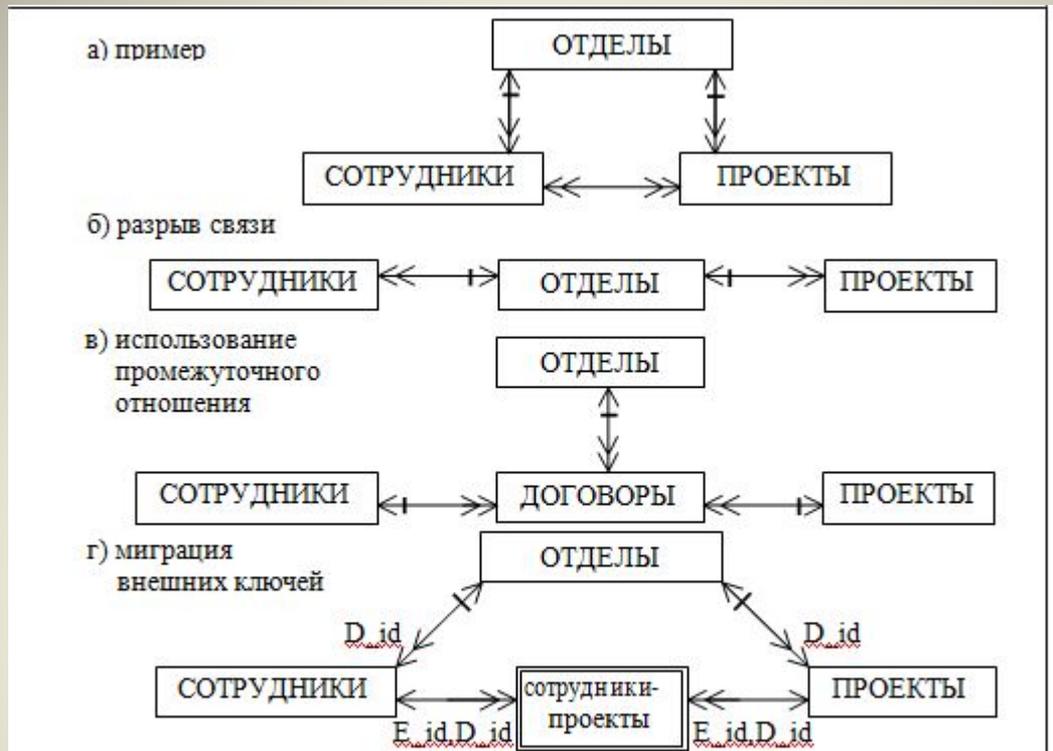


Рис.6. Некоторые способы разрешения циклов в схеме базы данных

## Составление реляционных отношений

Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так:

N – числовой,

C – символьный тип фиксированной длины,

V – символьный тип переменной длины,

D – дата (этот тип имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Потенциальными ключами отношения ОТДЕЛЫ являются атрибуты Аббревиатура и Название отдела. Первый занимает меньше места, поэтому мы выбираем его в качестве первичного ключа.

Потенциальными ключами отношения ОТДЕЛЫ являются атрибуты Аббревиатура и Название отдела. Первый занимает меньше места, поэтому мы выбираем его в качестве первичного ключа.

Таблица 1.

Таблица 1.

Схема отношения ОТДЕЛЫ (Departs)

Содержание поля	Имя поля	Тип, длина	Примечания
Аббревиатура отдела	D_ID	C(10)	<b>первичный ключ</b>
Название отдела	D_NAME	V(100)	обязательное поле
Комнаты	D_ROOMS	V(20)	обязательное многозначное поле
Телефоны	D_PHONE	V(40)	обязательное многозначное поле

Потенциальными ключами отношения СОТРУДНИКИ являются поля Паспортные данные, ИНН и Номер страхового пенсионного свидетельства. Все они занимают достаточно много места, а паспортные данные кроме того могут меняться. Введём суррогатный первичный ключ Номер сотрудника.

Таблица 2.

Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	E_ID	N(4)	<b>суррогатный первичный ключ</b>
Фамилия, имя, отчество	E_NAME	V(50)	обязательное поле
Дата рождения	E_BORN	D	обязательное поле
Пол	E_SEX	C(1)	обязательное поле, 'м' или 'ж'
Паспортные данные	E_PASP	V(50)	обязательное поле
ИНН	E_INN	C(12)	обязательное уникальное поле
Номер пенсионного страхового свидетельства	E_PENS	C(14)	обязательное уникальное поле
Отдел	E_DEPART	C(10)	внешний ключ (к <u>Departs</u> )
Должность	E_POST	V(30)	обязательное поле
Оклад	E_SAL	N(8,2)	обязательное поле, > 4500 руб.
Данные об образовании	E_EDU	V(200)	обязательное многозначное поле
Адреса	E_ADDR	V(100)	многозначное поле
Телефоны	E_PHONE	V(30)	многозначное поле
Логин	E_LOGIN	V(30)	

В отношении ПРОЕКТЫ три потенциальных ключа: Номер проекта, Название проекта и Сокращённое название. Меньше места занимает первый из них, но он малоинформативен.

Зато сокращённое название, используемое в качестве внешнего ключа в других таблицах, позволит специалисту идентифицировать проект без необходимости соединения с отношением ПРОЕКТЫ.

Таблица 3.

Схема отношения ПРОЕКТЫ (Projects)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер проекта	P_ID	N(6)	обязательное уникальное поле
Название проекта	P_TITLE	V(100)	обязательное поле
Сокращённое название	P_ABBR	C(10)	<b>первичный ключ</b>
Отдел	P_DEPART	C(10)	внешний ключ (к <u>Departs</u> )
Заказчик	P_COMPANY	V(40)	обязательное поле
Данные заказчика	P_LINKS	V(200)	обязательное поле
Руководитель	P_CHIEF	N(4)	внешний ключ (к <u>Employees</u> )
Дата начала проекта	P_BEGIN	D	обязательное поле
Дата окончания проекта	P_END	D	обязательное поле, больше даты начала проекта
Реальная дата окончания	P_FINISH	D	
Стоимость проекта	P_COST	N(10)	обязательное поле
Полученная сумма	P_SUM	N(10)	обязательное поле, значение по умолчанию – 0

Потенциальным ключом отношения ЭТАПЫ является комбинация внешнего ключа и номера этапа, а потенциальным ключом вспомогательного отношения УЧАСТИЕ является комбинация первых трёх полей этого отношения. Можно вообще не вводить первичный ключ для данных отношений, т. к. на них никто не ссылается. Но уникальность этих комбинации является в данном случае ограничением целостности данных, поэтому мы возьмём эти комбинации в качестве первичных ключей соответствующих отношений.

Таблица 4.  
Схема отношения ЭТАПЫ ПРОЕКТА (Stages)

Содержание поля	Имя поля	Тип, длина	Примечания	
Проект	S_PRO	C(10)	внешний ключ (к <u>Projects</u> )	<b>составной первичный ключ</b>
Номер этапа	S_NUM	N(2)		

Содержание поля	Имя поля	Тип, длина	Примечания
Название этапа	S_TITLE	V(200)	обязательное поле
Дата начала этапа	S_BEGIN	D	обязательное поле
Дата окончания этапа	S_END	D	обязательное поле, > даты начала
Реальная дата окончания	S_FINISH	D	больше даты начала этапа
Стоимость этапа	S_COST	N(10)	обязательное поле
Полученная сумма по этапу	S_SUM	N(10)	обязательное поле, значение по умолчанию – 0
Форма отчётности	S_FORM	V(100)	обязательное поле

Таблица 5.

Схема отношения УЧАСТИЕ\* (Job)

Содержание поля	Имя поля	Тип, длина	Примечания
Проект	J_PRO	C(10)	внешний ключ (к <u>Projects</u> )
Сотрудник	J_EMP	N(4)	внешний ключ (к <u>Employees</u> )
Роль	J_ROLE	V(20)	обязательное поле
Доплата	J_BONUS	N(2)	

\* в отношении УЧАСТИЕ первичный ключ состоит из первых 3х полей этого отношения.

## Нормализация полученных отношений (до 4НФ)

Механизм нормализации подразумевает определённую последовательность преобразования отношений к третьей нормальной форме. Мы не будем чётко придерживаться этой последовательности, т. к. она избыточна, и многозначные атрибуты сразу вынесем в отдельные отношения на первом же этапе.

**1НФ.** Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (одно значение атрибута – одна ячейка таблицы) и разбить сложные атрибуты на простые.

Разделим атрибут Фамилия, имя, отчество на два атрибута Фамилия и Имя, отчество, Паспортные данные на Номер паспорта (уникальный), Дата выдачи и Кем выдан, а Данные об образовании – на Вид образования, Специальность, Номер диплома и Год окончания учебного заведения.

Многозначные атрибуты Комнаты и Телефоны из отношения ОТДЕЛЫ вынесем в отдельное отношение КОМНАТЫ, а домашние и мобильные телефоны и адреса сотрудников – в отношение АДРЕСА = ТЕЛЕФОНЫ. Так как в комнате может не быть телефона, первичный ключ отношения КОМНАТЫ не определен (ПК не может содержать null-значения), но на этих атрибутах можно определить составной уникальный ключ. В отношении АДРЕСА – ТЕЛЕФОНЫ также нет потенциальных ключей: оставим это отношение без первичного ключа, т. к. на это отношение никто не ссылается. Данные об образовании сотрудников также вынесем в отдельное отношение.

Что касается рабочих телефонов сотрудников, то один из этих номеров – основной – определяется рабочим местом сотрудника (рассматриваются только стационарные телефоны). Будем хранить этот номер в атрибуте Рабочий телефон. Наличие других номеров зависит от того, есть ли в том же помещении (комнате) другие сотрудники, имеющие стационарные телефоны. Добавим в отношение СОТРУДНИКИ атрибут Номер комнаты, чтобы дополнительные номера телефонов сотрудника можно было вычислить из других кортежей с таким же номером комнаты.

Связь между отношениями СОТРУДНИКИ и КОМНАТЫ реализуем через составной внешний ключ (Номер комнаты, Рабочий телефон).

Мы также удалим вычисляемый атрибут Полученная сумма из отношения ПРОЕКТЫ, т. к. он является суммой значений аналогичного атрибута из отношения ЭТАПЫ ПРОЕКТОВ. Но атрибут Стоимость проекта оставим, т. к. она фигурирует в документации по проекту. А для обеспечения логической целостности данных предусмотрим в приложении проверку того, что сумма по всем этапам совпадает со стоимостью проекта.

**2НФ.** В нашем случае составные первичные ключи имеют отношения ЭТАПЫ ПРОЕКТА и УЧАСТИЕ. Неключевые атрибуты этих отношений функционально полно зависят от составных первичных ключей.

**3НФ.** В отношении ПРОЕКТЫ атрибут Данные заказчика зависит от атрибута Заказчик, а не от первичного ключа, поэтому его следует вынести в отдельное отношение ЗАКАЗЧИКИ. Но при этом первичным ключом нового отношения станет атрибут Заказчик, т.е. длинная символьная строка. Целесообразнее перенести в новое отношение атрибуты Заказчик и Данные заказчика и ввести для него суррогатный ПК. Так как с каждым заказчиком может быть связано несколько проектов, связь между отношениями ПРОЕКТЫ и ЗАКАЗЧИКИ будет 1:n и суррогатный ПК станет внешним ключом для отношения ПРОЕКТЫ.

В отношении СОТРУДНИКИ атрибут Оклад зависит от атрибута Должность. Поступим с этой транзитивной зависимостью так же, как в предыдущем случае: создадим отношение ДОЛЖНОСТИ, перенесём в него атрибуты Должность и Оклад, а первичным ключом сделаем название должности.

В отношениях СОТРУДНИКИ и ОБРАЗОВАНИЕ атрибуты (Дата выдачи и Кем выдан) и (Номер диплома и Год окончания учебного заведения) зависят не от первичного ключа, а от атрибутов соответственно Номер паспорта и Специальность. Но если мы выделим их в отдельное отношение, то получим связи типа 1:1. Следовательно, здесь декомпозиция нецелесообразна.

Таблица 6.

Схема отношения ОТДЕЛЫ (Departs)

Содержание поля	Имя поля	Тип, длина	Примечания
Аббревиатура отдела	D_ID	V(12)	<b>первичный ключ</b>
Название отдела	D_NAME	V(100)	обязательное поле

Таблица 7.

Схема отношения КОМНАТЫ (Rooms)

Содержание поля	Имя поля	Тип, длина	Примечания
Отдел	R_DEPART	V(12)	внешний ключ (к <u>Departs</u> )
Номер комнаты	R_ROOM	N(4)	<b>составной</b>
Телефон	R_PHONE	V(20)	<b>уникальный ключ</b>

Таблица 8.

Схема отношения ДОЛЖНОСТИ (Posts)

Содержание поля	Имя поля	Тип, длина	Примечания
Название должности	P_POST	V(30)	<b>первичный ключ</b>
Оклад	P_SAL	N(8,2)	обязательное поле, > 4500 руб.

Таблица 13.

Схема отношения ПРОЕКТЫ (Projects)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер проекта	P_ID	N(6)	обязательное уникальное поле
Название проекта	P_TITLE	V(100)	обязательное поле
Сокращённое название	P_ABBR	C(10)	<b>первичный ключ</b>
Отдел	P_DEPART	V(12)	внешний ключ (к <u>Departs</u> )
Заказчик	P_COMPAN NY	N(4)	внешний ключ (к <u>Clients</u> )
Руководитель	P_CHIEF	N(4)	внешний ключ (к <u>Employees</u> )
Дата начала проекта	P_BEGIN	D	обязательное поле
Дата окончания проекта	P_END	D	обязательное поле, больше даты начала проекта

Окончание табл. 13

Содержание поля	Имя поля	Тип, длина	Примечания
Реальная дата окончания	P_FINISH	D	больше даты начала проекта
Стоимость проекта	P_COST	N(10)	обязательное поле, > 0

Таблица 11.

Схема отношения АДРЕСА - ТЕЛЕФОНЫ (AdrTel)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	A_ID	N(4)	внешний ключ (к <u>Employees</u> )
Адрес	A_ADDR	V(50)	
Телефон	A_PHONE	V(30)	

Таблицы ОБРАЗОВАНИЕ и АДРЕСА-ТЕЛЕФОНЫ не имеют потенциальных ключей, но мы не будем вводить суррогатные первичные ключи, т.к. на эти таблицы никто не ссылается.

Таблица 12.

Схема отношения ЗАКАЗЧИКИ (Clients)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказчика	C_ID	N(4)	<b>суррогатный первичный ключ</b>
Заказчик	C_COMPANY	V(40)	обязательное поле
Адрес заказчика	C_ADR	V(50)	обязательное поле
Контактное лицо	C_PERSON	V(50)	обязательное поле
Телефон	C_PHONE	V(30)	обязательное поле

Таблица 14.

Схема отношения ЭТАПЫ ПРОЕКТА (Stages)

Содержание поля	Имя поля	Тип, длина	Примечания	
Проект	S_PRO	C(10)	внешний ключ (к <u>Projects</u> )	<b>составной первичный ключ</b>
Номер этапа	S_NUM	N(2)		
Название этапа	S_TITLE	V(200)	обязательное поле	
Дата начала этапа	S_BEGIN	D	обязательное поле	
Дата окончания этапа	S_END	D	обязательное поле, больше даты начала этапа	
Реальная дата окончания	S_FINISH	D	больше даты начала этапа	
Стоимость этапа	S_COST	N(10)	обязательное поле	
Полученная сумма по этапу	S_SUM	N(10)	обязательное поле, значение по умолчанию – 0	
Форма отчётности	S_FORM	V(100)	обязательное поле	

Таблица 15.

Схема отношения УЧАСТИЕ (Job)

Содержание поля	Имя поля	Тип, длина	Примечания	
Проект	J_PRO	C(10)	внешний ключ (к <u>Projects</u> )	<b>составной ПК</b>
Сотрудник	J_EMP	N(4)	внешний ключ (к <u>Employees</u> )	
Роль	J_ROLE	V(20)	обязательное поле	
Доплата	J_BONUS	N(2)		

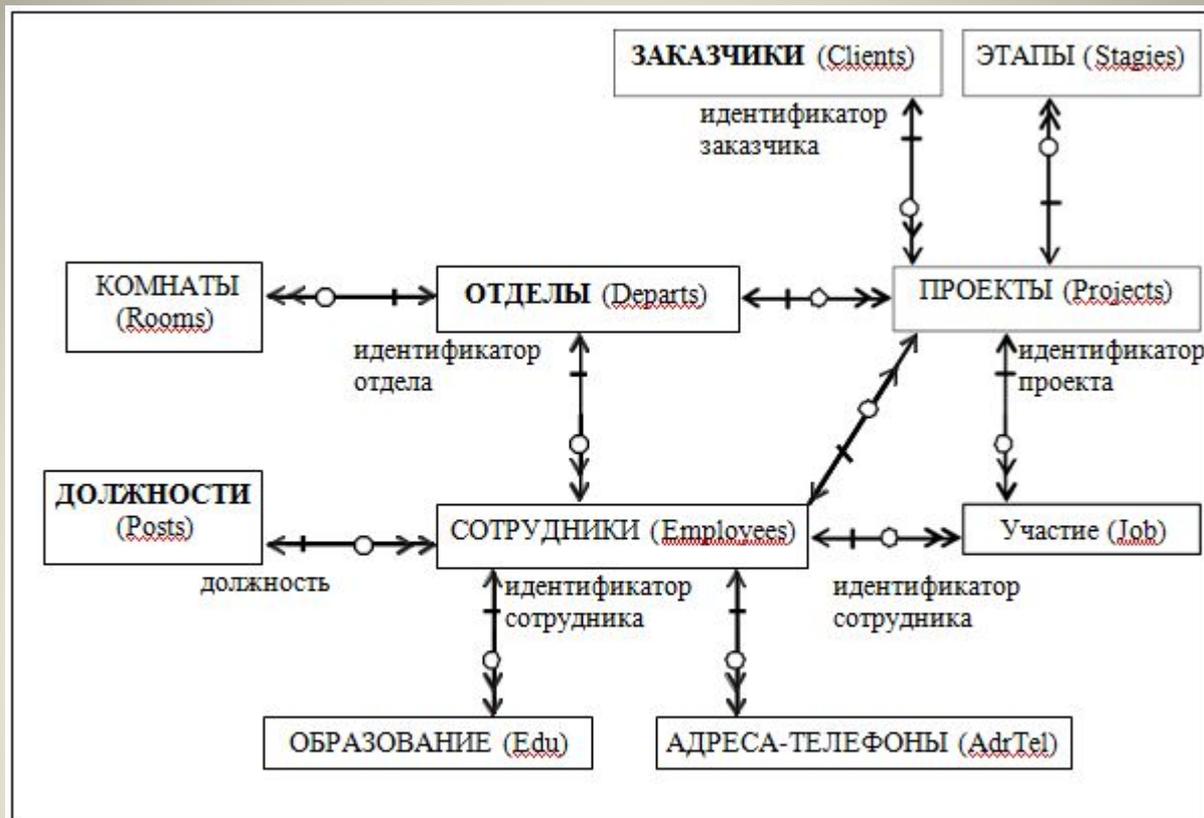


Рис. 7. Окончательная схема БД проектной организации

## Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Права приведены в таблице 16.

Таблица 16.

Права доступа к таблицам для групп пользователей

Таблицы	Группы пользователей (роли)				
	Руководители организации	Сотрудники отд. кадров	Руководители проектов	Бухгалтеры	Участники проектов
Отделы	S	SIUD	S	S	-
Комнаты	S	SUID	S	S	S
Должности	SIUD			S	-
Сотрудники	S	SUID	S	S	-
Адреса-телефоны	S	SUID	S	S	-
Образование	S	SUID	S	S	-
Заказчики	SIUD	-	S	-	-
Проекты	SIUD	-	S	-	-
Этапы проектов	SIUD	-	SUI	-	-
Участие	S	-	S	S	-

Используются следующие сокращения:

- s – чтение данных (select);
- i – добавление данных (insert);
- u – модификация данных (update);
- d – удаление данных(delete).

## Реализация проекта базы данных

Мы условились не привязываться к конкретной СУБД и выполнять описание логической схемы БД на SQL-92. Приведём описание схемы БД на DDL.

### Создание таблиц

1. Отношение Departs (отделы):

```
create table departs (  
  d_id   varchar(12)   primary key,  
  d_name varchar(100)   not null);
```

2. Отношение Rooms (комнаты):

```
create table rooms (  
  d_depart varchar(12)   references departs(d_id),  
  r_room   numeric(4)    not null,  
  r_phone  varchar(20),  
  unique(r_room, r_phone));
```

3. Отношение Posts (должности):

```
create table posts (  
  p_post  varchar(30)   primary key,  
  p_salary numeric(8,2)  not null check(p_salary >= 4500));
```

4. Отношение Employees (сотрудники):

```
create table employees (  
  e_id    numeric(4)    primary key,  
  e_fname  varchar(25)   not null,  
  e_lname  varchar(30)   not null,  
  e_born   date         not null,  
  e_sex   char(1)       check(e_sex in ('Ж', 'М')),  
  e_pasp  char(10)      not null unique,  
  e_date  date         not null,
```

---

```
  e_given varchar(50)   not null,  
  e_inn   char(12)     not null unique
```

## Создание представлений (готовых запросов)

Приведём примеры нескольких готовых запросов (представлений):

- Список всех текущих проектов (sysdate – функция, возвращающая текущую дату, определена в СУБД Oracle; в других системах аналогичная функция может называться по-другому, например, getdate() в Transact-SQL, now() в MS Access, currdate() в MySQL и т.д.):

```
create view curr_projects as
  select *
from projects
where p_begin<=sysdate and sysdate<=p_end;
```

- Определение суммы по текущим проектам, полученной на текущую дату:

```
create or replace view summ (title, cost, total) as
  select p_title, p_cost, sum(s_sum)
  from curr_projects, stages where
  p_abbr=s_pro
  group by p_title, p_cost;
```

## Назначение прав доступа

Права доступа пользователей предоставляются с помощью команды GRANT. Рассмотрим для примера права сотрудника компании **ok\_user**, который является сотрудником отдела кадров. Права доступа к отношениям Departs и Rooms могут быть описаны следующим образом:

```
grant select, insert, update, delete on departs to ok_user;  
grant select, insert, update, delete on rooms to ok_user;
```

Права доступа руководителей проектов (сотрудников, staff) к представлению my\_projects могут быть описаны следующим образом:

```
grant select, insert, update, delete on my_projects to staff;
```

Если сотрудник не является руководителем проекта, он не получит данных через этот запрос и не сможет воспользоваться правами доступа к нему.

Права доступа участников проекта (сотрудников, staff) к представлению my\_emps ии могут быть описаны следующим образом:

```
grant select on my_emps to staff;
```

Если сотрудник не является участником проекта, он не получит данных через этот запрос и не сможет воспользоваться правами доступа к нему.

## **Создание индексов**

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей. Приведём примеры создания индексов:

```
create index e_posts on employees(e_post); create index p_chief on projects(p_chief);  
create index e_tel on employees(e_room, e_phone);
```

## **Разработка стратегии резервного копирования**

Интенсивность обновления разработанной базы данных низкая, поэтому для обеспечения сохранности вполне достаточно проводить полное резервное копирование БД раз в день (перед окончанием рабочего дня). Для разработанной БД нет необходимости держать сервер включенным круглосуточно, поэтому можно создать соответствующее задание операционной системы, которое будет автоматически запускаться перед выключением сервера.

## Вопросы для подготовки к экзамену по дисциплине "Базы данных"

для студентов направления 09.03.01 «Информатика и вычислительная техника» Института компьютерных систем и информационной безопасности (ИКСИБ)

1. Информационные системы и банки данных.
2. Основные функции группы администраторов.
3. Классификация баз данных.
4. Трёхуровневая архитектура баз данных.
5. Процесс прохождения пользовательского запроса.
6. Этапы жизненного цикла базы данных.
7. Модели жизненного цикла базы данных.
8. Концептуальное проектирование базы данных. Модель сущность-связь.
9. Иерархическая модель БД.
10. Сетевая модель БД.
11. Постреляционная модель БД
12. Многомерная модель БД.
13. Объектно-ориентированная модель БД.
14. Структурная часть реляционной модели данных. Базовые понятия реляционной модели данных. Свойства отношений.
15. Целостность реляционных данных. Потенциальные, первичные и альтернативные ключи.
16. Целостность реляционных данных. Внешние ключи. Правило ссылочной целостности. Стратегии поддержания ссылочной целостности.
17. Целостность реляционных данных. Неопределённые значения.
18. Реляционная алгебра. Традиционные теоретико-множественные операции.
19. Реляционная алгебра. Специальные реляционные операции.
20. Реляционная алгебра. Дополнительные операции. Операции изменения тела отношения. Правила записи выражения реляционной алгебры.
21. Реляционное исчисление.
22. Основные категории команд языка SQL.
23. Базовый оператор структурированных запросов SELECT.
24. Язык SQL. Основные предикаты оператора SELECT (простая выборка, выборка с условием отбора, выборка с параметром, выборка с вычислениями, выборка с упорядочиванием, выборка по связанным таблицам).
25. Язык SQL. Агрегирование в операторе SELECT (группировка и итоговые функции, группировка и условие отбора, фильтрация после группировки, перекрестный запрос, ограничения на выборку).
26. Язык SQL. Нетривиальные запросы (сложные выборки с подзапросами).
27. Язык SQL. Команды манипулирования данными. Запросы-действия.
28. Язык SQL. Команды определения данных
29. Смысл нормализации схем баз данных. Вложенность нормальных форм.
30. Первая нормальная форма 1NF.

31. Вторая нормальная форма 2NF.
32. Третья нормальная форма 3NF.
33. Нормальная форма Бойса-Кодда NFBC.
34. Многозначные зависимости. Четвертая нормальная форма.
35. Пятая нормальная форма(нормальная форма проекции-соединения).
36. Достоинства и недостатки нормализации. Денормализация БД.
37. Алгоритм нормализации
38. Доступ к базам данных.
39. Методы физического проектирования для реляционных моделей по первичному ключу.
40. Методы физического проектирования для реляционных моделей по вторичному ключу.
41. Методы физического проектирования для иерархических моделей.
42. Методы физического проектирования для сетевых моделей.
43. Методы обеспечения безопасности. Избирательное управление доступом.
44. Методы обеспечения безопасности. Обязательное управление доступом.
45. Шифрование данных. Контрольный след выполняемых операций.
46. Поддержка мер обеспечения безопасности в языке SQL. Директивы GRANT и REVOKE.
47. Понятие восстановления системы. Транзакции. Свойства АСИД.
48. Алгоритм восстановления после сбоя системы.
49. Параллелизм. Проблемы параллелизма.
50. Понятие блокировки. Решение проблемы параллелизма.
51. Тупиковые ситуации.