

Лабораторная работа № 17. Создание базы данных

Цель работы – получение навыков описания таблиц БД Oracle APEX и заполнения их данными.

Задачи:

- 1) Описать структуры простых таблиц БД.
- 2) Заполнить таблицы данными.
- 3) Создать связи между таблицами.
- 4) **Оформить отчет о выполнении ЛР в среде MS Word и представить преподавателю.**

Для знакомства с технологией APEX в последующих работах решается ряд прикладных задач на БД, описывающей структуру некоторой торговой корпорации. Первоначальная схема БД, полученная с помощью СУБД Access, представлена на рис. 1.1 (некоторые таблицы будут добавляться по ходу выполнения лабораторных работ):

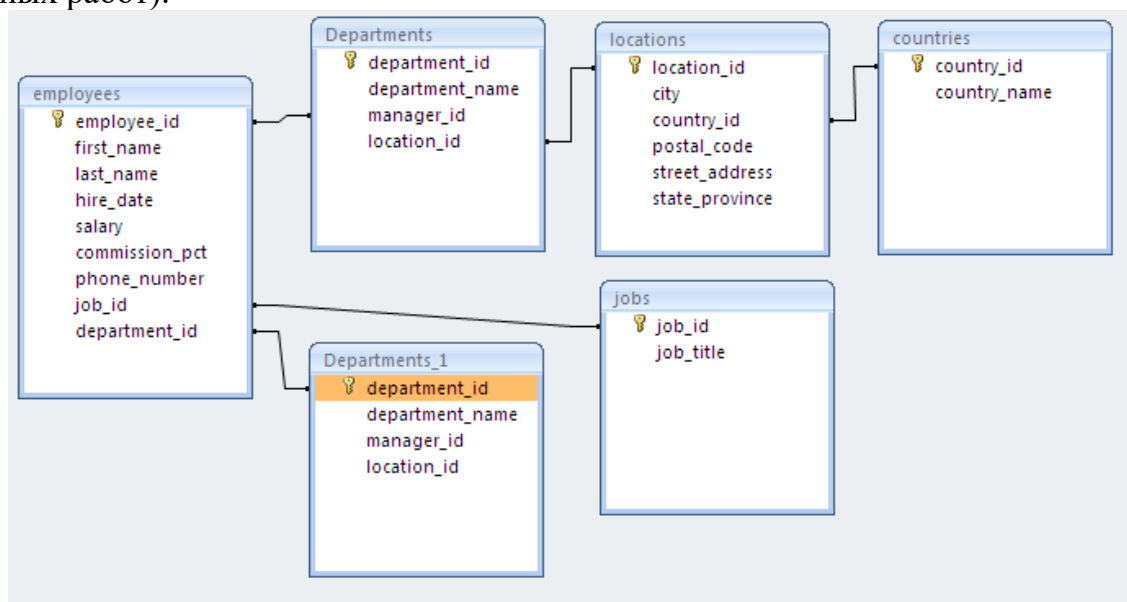


Рисунок 1.1. Логическая схема БД, представленная средствами СУБД Access

БД включает 5 основных таблиц (далее приведены описания этих таблиц в конструкторе СУБД Access):

- 1) Countries:

Countries		
	Имя поля	Тип данных
	Country_Id	Числовой
	Country_Name	Текстовый

2) Departments:

Departments		
	Имя поля	Тип данных
🔑	Department_Id	Числовой
	Department_Name	Текстовый
	Manager_Id	Числовой
	Location_Id	Числовой <input type="text"/>

3) Locations:

Locations		
	Имя поля	Тип данных
🔑	Location_Id	Числовой
	City	Текстовый
	Country_Id	Числовой
	Postal_Code	Числовой
	Street_Address	Текстовый
	State_Province	Текстовый <input type="text"/>

4) Employees :

Employees		
	Имя поля	Тип данных
🔑	Employee_Id	Числовой
	First_Name	Текстовый
	Last_Name	Текстовый
	Hire_Date	Дата/время
	Salary	Числовой
	Commission_Pct	Числовой
	Phone_Number	Текстовый
	Job_Id	Текстовый
	Department_Id	Числовой <input type="text"/>

5) Jobs :

jobs		
	Имя поля	Тип данных
🔑	job_id	Числовой
	job_title	Текстовый

Фрагменты самих таблиц, созданных в СУБД Access, показаны ниже. Они демонстрируют примеры данных, заполняющих БД:

COUNTRIES	
country_id	country_name
11	Россия
...	...

LOCATIONS					
location_id	city	country_id	postal_code	street_address	state_province
123	Калининград	11	123456	Советский пр.	Калининградская обл.
...

DEPARTMENTS			
department_id	department_name	manager_id	location_id
111	Виктория	1	123
...

EMPLOYEES								
employee_id	first_name	last_name	hire_date	salary	commission_pct	phone_number	job_id	department_id
1	Иванов	Иван	01.12.1989	12000	15	23-34-56	1	111
...

JOBS	
job_id	job_title
1	управляющий
...	...

Между таблицами сформированы связи (показаны на рис. 1.1 стрелками), позволяющие соединить между собой данные в единую систему. Наличие таких связей обеспечивает целостность данных в системе. Так, при попытке удалить родительскую запись выводится диагностическое сообщение о невозможности данной операции, поскольку существуют связанные записи. Таким образом, для удаления родительской записи сначала нужно удалить все подчиненные ей.

Кроме основных таблиц, в схеме представлена вспомогательная таблица `Departments_1`, которая сформирована самой СУБД для представления связи между основными таблицами (формирование связей рассматривается далее в данной работе).


1. Создание простых таблиц

Рассмотрим для начала технологию создания простых, не связанных между собой, таблиц в АРЕХ на примере таблиц `Countries` и `Locations`:

1. Запуск мастера создания таблиц:

- Откройте `SQL Workshop` и запустите `Object Browser`.
- Щелкните по кнопке **Create** в правом верхнем углу страницы. Отобразится список типов объектов БД, которые можно создать.



- в. Щелкните по ссылке **Table**. Запустится мастер создания таблицы. Вверху в виде линейного графика отражаются шаги создания таблицы. Активный шаг – **Columns**.
- г. В поле **Table Name** введите название таблицы - **Countries**:
- д. Поля **Column Name** (Имя столбца), **Type** (Тип Данных), **Precision** (Точность, Максимальный размер), **Scale** (Размер, Количество знаков после запятой), **Not Null** (Обязательное) предназначены для описания столбцов создаваемой таблицы. Добавьте описания следующих столбцов:
- Столбец с названием `country_id`, тип данных – `Number(4)`, **обязательный**. В этом столбце будет храниться уникальный числовой идентификатор страны.
 - Столбец с названием `country_name`, тип данных – `Varchar2(20)`, обязательный. В этом столбце будет храниться название страны.
 - Нажмите **Next>**.
- е. На шаге **Primary Key**:
- В поле **Primary Key** выберите `Populated from a new sequence` (значения столбца первичного ключа будут браться из нового объекта-последовательности). Последовательность (Sequence) – это объект БД, который используется для генерации уникальных числовых значений.
 - В полях **Primary Key Constraint Name** (Название ограничения целостности первичного ключа) и **Sequence Name** (Название последовательности) оставьте значения по умолчанию, сгенерированные APEX.
 - В поле **Primary Key** (Первичный ключ) выберите столбец `country_id`.
 - Нажмите **Next>**.
- ж. Шаг **Foreign Key** (Внешний ключ) пропустите (т.е. нажмите **Next>**), т.к. в создаваемой таблице пока нет ссылок на другие таблицы.
- з. На шаге **Constraints** (Ограничения целостности) добавьте уникальный ключ, определенный на столбце `country_name`:
- Выберите радиокнопку **Unique** (Уникальный ключ).
 - В поле-списке **Key Column(s)** (Столбцы ключа) перенесите столбец `country_name`.
 - В поле **Name** должно быть указано название создаваемого ограничения целостности. Это имя должно быть уникальным в БД. Оставьте значение, предлагаемое по умолчанию.
 - Нажмите кнопку **Add**, чтобы добавить ограничение в список ограничений целостности создаваемой таблицы.
 - Нажмите **Next>**.
- и. На шаге **Confirm** можно просмотреть сгенерированный мастером SQL-скрипт по созданию таблицы, щелкнув по ссылке .

```
CREATE table "COUNTRIES" (  
  "COUNTRY_ID" NUMBER(4) NOT NULL,  
  "COUNTRY_NAME" VARCHAR2(20) NOT NULL,  
  constraint "COUNTRIES_PK" primary key ("COUNTRY_ID")  
)  
/
```

```

CREATE sequence "COUNTRIES_SEQ"
/
CREATE trigger "BI_COUNTRIES"
before insert on "COUNTRIES"
for each row
begin
if :NEW."COUNTRY_ID" is null then
select "COUNTRIES_SEQ".nextval into :NEW."COUNTRY_ID" from sys.dual;
end if;
end;
/
alter table "COUNTRIES" add
constraint "COUNTRIES_UK1"
unique ("COUNTRY_NAME")
/

```

Подтвердите создание таблицы, нажав на кнопку **Create Table**.

Созданная таблица появится в списке таблиц. Для выбранной таблицы в центральной части **Object Browser** отображается ее детальное описание, организованное в виде вкладок (**Table, Data, Indexes, Model, Constraints, ...**). На каждой вкладке представлены кнопки с операциями, которые можно совершать над таблицей (например, на вкладке **Table: Add Column, Modify Column** и т.д.).

2. Изучите содержимое вкладок описания созданной таблицы.
3. Аналогичным образом создайте новую таблицу, предназначенную для хранения сведений о местах размещения отделений торговой корпорации:
 - а. Название таблицы - **Locations**.
 - б. В таблице должны быть определены столбцы следующим образом:

location_id - Number (4) , обязательный,
city - Varchar2 (40) , обязательный,
country_id - Number (4) ,
postal_code - Number (6) ,
street_address - Varchar2 (40) ,
state_province - Varchar2 (40)

Обратите внимание на столбец **country_id**: при формировании связей между таблицами он будет содержать ссылку на страну (на запись в соответствующей таблице). Поэтому он должен иметь тот же тип данных и размер, что и столбец первичного ключа в созданной ранее таблице **countries**.

- в. В первичный ключ должен входить столбец **location_id**, значения для него должны браться из новой последовательности, которая должна быть создана.
 - г. Шаг **Foreign Key** (Внешний ключ) пропустите (т.е. нажмите **Next>**).
4. Изучите содержимое вкладок описания созданной таблицы. Обратите внимание на вкладку **Model**. На ней отображаются таблицы, на которые ссылается данная таблица, и которые ссылаются на данную таблицу. Поскольку мы создали простые, не связанные таблицы, в окне показана отдельная таблица.

Ниже в качестве справочного материала приведены некоторые соответствия между типами полей в Access и APEX:

Тип поля Access	Тип поля Oracle
Числовой, все форматы, в том числе: Счетчик, длинное целое, Действительное Денежный	NUMBER — числовые данные

Тип поля Access	Тип поля Oracle
Текстовое	VARCHAR 2 — текстовые строки переменной длины до 4000байт
Поле MEMO	LONG - текстовые строки длиной до 2 Гб CLOB – тексты длиной до 4 Гб
Дата/время	DATE - даты TIME - время

2. Заполнение таблиц

Заполнение таблиц Oracle возможно двумя способами: путем импортирования данных из готовых Excel-таблиц (при этом возможно также и создание таблиц); путем добавления данных в таблицы в среде APEX.

Если данными заполняются простые таблицы, проблем нет. При заполнении связанных таблиц надо сначала заполнить родительскую таблицу, а затем - подчиненную.

2.1. Импортирование данных из Excel-таблиц

Для применения данной технологии структура таблицы в БД Oracle должна совпадать со структурой таблицы в Excel.

Для импортирования следует выполнить команды :

SQL Workshop→Utilities→Data Workshop.

Потом выбирается, откуда загружать данные (**Load Data**). Существуют две возможности:

- из файла (**Upload a File**),
- копировать через буфер обмена Windows (**Copy and paste**) (рекомендуется).

Рассмотрим случай, когда данные копируются через буфер обмена:

1. на следующем шаге импорта мастер предлагает вставить данные из буфера в окошко, расположенное в центре страницы:
 - открывается таблица в Excel, ее строки копируются в буфер обмена,
 - мышью щелкается в окошке на странице APEX и нажимается комбинация клавиш CTRL+V (вставить) **Next>**,
2. в новом окне (**Load Data**) устанавливается нужное имя таблицы. **Рекомендуется проверить самую первую строчку - там должны быть названия столбцов.** Если имена полей совпадают в обеих таблицах, нажимается **Save Changer>**. Если предполагается переименовать поля, то исправляются их названия в окне со вставленными данными, чтобы они совпадали с именами полей в БД APEX, на вкладке **Columns to Load** нужно проверить и исправить тип, длину и имена полей. Если какие-то столбцы не нужны, можно их убрать
3. нажимается кнопка **Load Data**.

Появляется список всех загруженных таблиц, где можно просмотреть результаты импорта, например, возможные ошибки. Если загрузка данных прошла успешно, то щелкнув по имени таблицы, можно перейти на страницу **Object Browser** с открытой структурой таблицы. Чтобы увидеть свои данные, надо выбрать **Data** в списке команд над таблицей.

Теперь можно редактировать данные, щелкнув по значку в столбце **Edit** в строке, которую нужно изменить. APEX не позволяет изменять данные прямо в

табличном формате, как это принято в Access или Excel. Вместо этого он открывает анкетную форму для редактируемой записи.

После внесения изменений в анкетную форму нажать кнопку **Apply Changes**.

2.2. Добавление данных в среде APEX

В среде APEX данные можно вводить через опции меню. Для этого сразу после входа в среду APEX выполнить команды SQL Workshop → Object Browser.

Выбрать нужную таблицу в левом столбце, выполнить команду **Data**. Нажимают кнопку **Insert Row** и в анкете вводят новые данные, после чего нажимают кнопку **Create**.

Следует отметить, что APEX-приложение, как правило, имеет специальные формы для пользователя, где последний сможет редактировать данные в таблицах, не обращаясь к среде разработки. Как и в Access, приложение более удобно для пользователя, чем средства, предназначенные для разработчиков.

3. Создание связей между таблицами

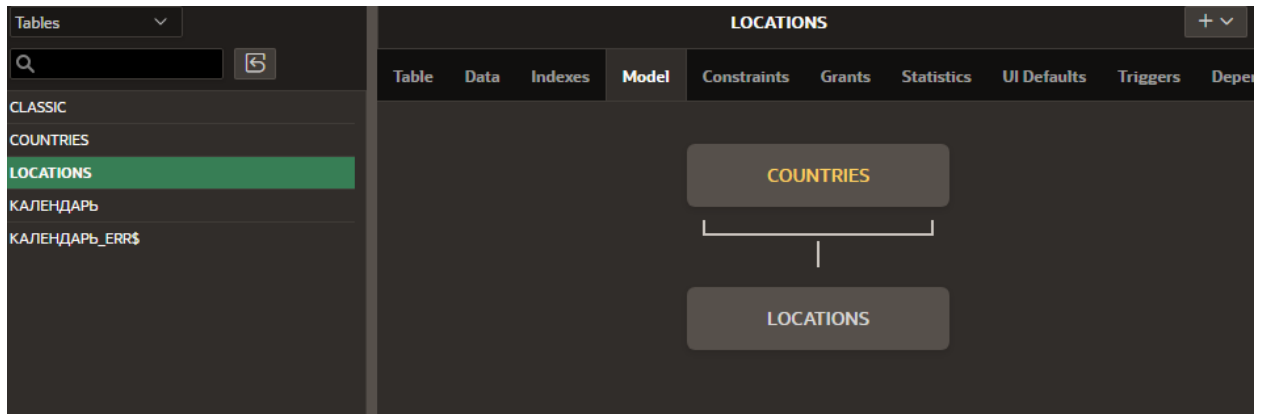
Из схемы БД видно, что между таблицами существуют связи, которые реализуются через соответствующие поля. Эти связи можно формировать как во время создания таблиц, так и после их создания и, возможно, заполнения данными.

3.1. Создание связей между простыми таблицами

Рассмотрим формирование связей после создания простых, не связанных между собой, таблиц. Для этого образуем связь подчинения Countries (страны) → Locations (местонахождение) (связующими являются поля этих таблиц country_id, причем одноименное поле в таблице Locations ссылается на такое же поле в таблице Countries, показывая тем самым, в какой стране находится отделение корпорации):

1. Откройте вкладку **Constraints** описания созданной таблицы Locations.
2. Щелкните по кнопке **Create**. Отобразится мастер добавления ограничения целостности:
 - В поле **Constraint Name** впишите название создаваемого ограничения (оно должно быть уникальным в БД) например, location_con.
 - В поле **Constraint Type** выберите Foreign Key и Disallow Delete.
 - В списке **Foreign Key Column(s)** выделите столбец country_id.
 - В поле **Reference Table Name** (название родительской таблицы) выберите название созданной таблицы Countries.
 - В списке **Reference Table Column** (столбцы родительской таблицы, на которые будут ссылаться столбцы внешнего ключа) выберите столбец первичного ключа родительской таблицы – country_id.
 - Нажмите **NEXT>**.
3. Подтвердите создание внешнего ключа, нажав на кнопку **Finish**. Созданное ограничение целостности отобразится в списке ограничений целостности таблицы.

4. Откройте вкладку **Model** для таблицы `Locations`:



Видно, что теперь таблицы связаны: таблица `Locations` подчиняется таблице `Countries`.

3.2. Создание связанных таблиц

Можно «закладывать» связь между таблицами уже во время их создания с помощью `Object Browser`. Рассмотрим эту технологию на примере тех же таблиц в предположении, что мы их создаем заново как связанные таблицы:

1. Создается таблица `Countries` по описанной в разделе «Создание простых таблиц» технологии.
2. Аналогично создается таблица `Locations` до шага **Foreign Key**:
 - а. На шаге **Foreign Key** (Внешний ключ) добавляется ограничение ссылочной целостности (внешний ключ) для столбца `country_id`:
 - В секции **Add Foreign Key** (Добавить внешний ключ) в поле **Name** вписывается название создаваемого ограничения целостности, например, `country_con`. Необходимо убедиться, что длина названия не превышает 30 символов (ограничение СУБД Oracle на длину идентификаторов объектов). При необходимости надо сократить название.
 - Из группы радиокнопок {`Disallow Delete`, `Cascade Delete`, `Set Null on Delete`} оставить выбранным `Disallow Delete` (запрещать удаление родительской записи, если у нее есть дочерние).
 - В качестве столбца, на который накладывается ограничение целостности (поле со списком **Key Column(s)**), выбрать столбец `country_id`.
 - В качестве родительской таблицы (на которую ссылается столбец `country_id`) выбрать созданную ранее таблицу `Countries`.
 - В появившееся ниже поле со списком **Referenced Column(s)** добавить столбец `country_id`, на который будет ссылаться столбец `country_id`.
 - Нажать кнопку **Add**, чтобы добавить ограничение целостности в список внешних ключей.
 - б. Закончить создание таблицы по описанной ранее технологии.

Задание к работе:

1. В соответствии с вариантом (табл. 1.1) разработать логическую схему БД, аналогичную рис. 1.1. *Вариант соответствует последней цифре номера зачетной книжки.*

Состав полей таблиц можно уточнить с преподавателем.

Таблица 1.1

Вариант	Предметная область, сущности и их атрибуты
1	Обучение в вузе: <i>Студент</i> (№ зачётки; фамилия; имя; отчество; группа; факультет; специальность; год поступления; дата рождения; адрес проживания; ...). <i>Дисциплина учебного плана</i> (код; название; группа дисциплин {гуманитарные и социально-экономические, математические и естественно-научные, общепрофессиональные, специальные}; число часов аудиторных занятий; число часов самостоятельной работы; семестр, в котором изучается дисциплина; ...). <i>Зачётная книжка</i> (№ зачётки; дисциплина; семестр; вид занятий {лекции, практические, лабораторные}; количество часов; дата сдачи; оценка; ФИО преподавателя; ...). <i>Преподаватель</i> (ФИО, ученая степень {кандидат наук, доктор наук}, научное звание {доцент, профессор}, должность {ассистент, старший преподаватель, доцент, профессор}, контактные данные,...).
2	Автомастерская: <i>Автомобиль</i> (марка; год выпуска; номер; ФИО хозяина; объём двигателя; цвет; ...). <i>Работник</i> (код; фамилия; имя; отчество; специализация {маляр, слесарь, сварщик, жестянщик, механик}; дата приёма на работу; ...). <i>Ремонт</i> (дата; автомобиль; характер неисправности/повреждения; вид ремонта; работник; стоимость; ...). <i>Автозапчасть</i> (наименование, марка автомобиля, количество на складе, стоимость единицы,...)
3	Гостиница: <i>Номер</i> (номер; этаж; кол-во мест; класс {стандарт, люкс}; стоимость проживания в сутки; ...). <i>Постоялец</i> (регистрационный номер; фамилия; имя; отчество; номер паспорта; кем выдан паспорт; дата выдачи паспорта; дата рождения; ...). <i>Регистрация</i> (номер постояльца; дата регистрации; предоставленный номер; дата выбытия; ФИО портье; ...). <i>Обслуга</i> (ФИО, контактные данные, должность...).
4	Чемпионат по футболу: <i>Команда</i> (код; название; город; ФИО тренера; годовой бюджет; ...). <i>Игрок</i> (команда; фамилия; имя; отчество; специализация {вратарь, защитник, полузащитник, нападающий}; дата рождения; гражданство; дата окончания контракта; ...). <i>Встреча</i> (принимающая команда; гостевая команда; дата игры; итоговый счет; главный судья встречи; ...). <i>Стадион</i> (размещение, число мест, категория {закрытый, открытый}...)

Вариант	Предметная область, сущности и их атрибуты
5	<p>Спортивный магазин:</p> <p><i>Товар</i> (код; наименование; вид спорта; цена; отдел {спорт. инвентарь, одежда, тренажёры, спортивное питание}); количество на складе, ...).</p> <p><i>Работник</i> (личный номер; фамилия; имя; отчество; дата рождения; должность; отдел; дата приема на работу; ...).</p> <p><i>Продажа</i> (личный номер работника; дата продажи; товар; количество; итоговая сумма без скидки; % скидки; ...).</p> <p><i>Поставщик</i> (название предприятия, размещение, ФИО директора, контактные данные...).</p>
6	<p>Склад:</p> <p><i>Продукция</i> (наименование продукции, шифр продукции, единица измерения, цена единицы измерения, код поставщика...).</p> <p><i>Поставщик</i> (наименование, код, адрес,...).</p> <p><i>Получатель</i> (наименование, код, адрес,...) .</p> <p><i>Поставка</i> (наименование продукции, шифр продукции, количество в поставке, дата заказа, плановая дата поставки, фактическая дата поставки, размер штрафа за просрочку поставки,...).</p>
7	<p>Штат кафедры:</p> <p><i>Сотрудник</i> (ФИО, табельный номер, стаж, должность, оклад, кафедра).</p> <p><i>Кафедра</i> (название, месторасположение, ФИО заведующего, телефон, факультет,...).</p> <p><i>Должность</i> (название, требуемое образование, оклад...)</p> <p><i>Факультет</i> (название, аудитория, ФИО декана,...)</p>
8	<p>Перевозки:</p> <p><i>Водитель</i> (ФИО водителя, табельный номер водителя, номерной знак его автомобиля,...).</p> <p><i>Автопарк</i> (номерной знак автомобиля, марка автомобиля, год выпуска, местонахождение автомобиля {ремонт, рейс, база},...)</p> <p><i>Автомобиль</i> (марка автомобиля, грузоподъемность данной марки, средняя скорость данной марки км/час,...)</p> <p><i>Поставка</i> (наименование груза, вес груза, отправитель груза, получатель груза, номерной знак автомобиля-перевозчика груза,...)</p>
9	<p>Библиотека:</p> <p><i>Книга</i> (название, автор, инвентарный номер, год выпуска,...).</p> <p><i>Читатель</i> (ФИО, адрес, контактные данные, номер читательского билета,...).</p> <p><i>Библиотекарь</i> (ФИО, контактные данные,...).</p> <p><i>Выдача книги</i> (ФИО библиотекаря, инвентарный номер книги, номер читательского билета, дата выдачи, плановая дата возврата, фактическая дата возврата, штраф за задержку книги,...).</p>
10	<p>Фитнес-клуб:</p> <p><i>Клиент</i> (ФИО, контактные данные, возраст, программа занятий,...).</p> <p><i>Типовая программа занятий</i> (название, вид упражнения, продолжительность упражнения, стоимость курса занятий,...).</p> <p><i>Тренер</i> (ФИО, квалификация, специализация, контактные данные,...).</p> <p><i>Занятия клиента</i> (ФИО клиента, ФИО тренера, программа занятий, дата начала занятий, дата окончания занятий,...)</p>

2. Средствами APEX, следуя описанной технологии, описать таблицы и заполнить их данными. Состав полей может быть изменен по согласованию с преподавателем. В каждую таблицу включить идентификатор экземпляра сущности (поля типа *_id в примере). Связи подчинения между сущностями указать в таблицах с помощью идентификатора (так, например, в таблице EMPLOYEES показаны связи между подразделением и должностью через поля job_id и department_id).
3. Создать связи между таблицами для поддержания целостности данных.