

## Лабораторная работа №2

### Часть 1. Создание реляционной схемы данных

В лабораторной работе выполняется *логическое проектирование* БД путем построения *реляционной схемы данных* по ранее спроектированной ER-модели (см. лабораторную работу №1). Требуется преобразовать ER-диаграмму в реляционную схему данных (в виде UML-диаграммы) выполнение.

#### Порядок выполнения работы

- 1) Проверить ER-диаграмму, созданную в лабораторной работе №1.
- 2) Выполнить преобразование ER-диаграммы в реляционную модель в двух вариантах:
  - вид «бумажного» варианта преобразования приведен ниже в части 1 в пункте «Пример реляционной схемы данных»;
  - «автоматизированный» (см. часть 2 данной лабораторной работы).
- 3) Сравнить полученные диаграммы и, если есть расхождения в полученных реляционных диаграммах, найти несоответствия и устранить их.
- 4) Оформить *отчет*.

#### Пример реляционной схемы данных

Порядок перевода ER-модели в реляционную модель выполняется с помощью алгоритма, состоящего из шести шагов:

*Шаг 1.* Каждый *объект* на ER-диаграмме превращается в реляционное отношение (далее для краткости – таблицу), имя объекта становится именем таблицы (*следует указать понятное имя*).

*Шаг 2.* Каждый *атрибут объекта* становится столбцом таблицы с тем же именем (*также следует указать понятное имя*) и требуемым типом данных. *Шаг 3.* Уникальные (*ключевые*) атрибуты объекта превращаются в *первичный ключ* таблицы (при наличии нескольких возможных уникальных идентификаторов, выбирается наиболее подходящий для использования; если таковых атрибутов нет или они плохо подходят для длительного использования в БД, то желательно создать *суррогатный ключ*). *Каждая таблица в БД должна иметь первичный ключ!*

*Шаг 4.* Связи «*один-ко-многим*» (в том числе и связи «*один-к-одному*») становятся *ссылками* в уже существующих таблицах, при этом внешний ключ добавляется в виде столбца (столбцов) в таблицу, соответствующую объекту со стороны «*многие*» связи. *Внешние ключи должны ссылаться только на первичные ключи целевых таблиц!*

*Шаг 5.* Связи «*многие-ко-многим*» реализуются каждая через отдельную промежуточную таблицу:

- эта промежуточная таблица обязательно будет содержать столбцы внешних ключей, ссылающиеся на соответствующие объекты связи;

- первичный ключ промежуточной таблицы для исключения дубликатов должен быть составным и включать в себя все внешние ключи на объекты, участвующие в связи.

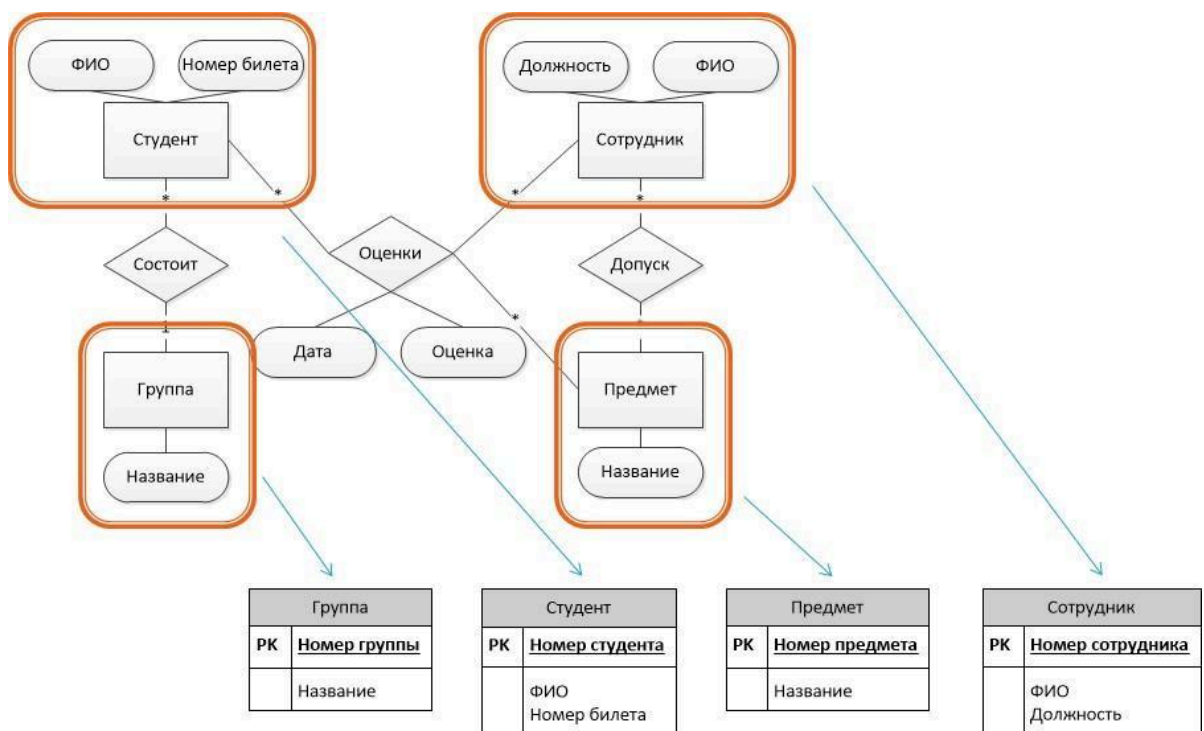
*Шаг 6.* Если связь имеет *дополнительные* атрибуты, то, как и в случае атрибутов объектов, они становятся столбцом соответствующей таблицы:

- для связей «*один-ко-многим*» (встречаются на практике редко) – в таблице со стороны «*многие*» (там, где расположен внешний ключ);

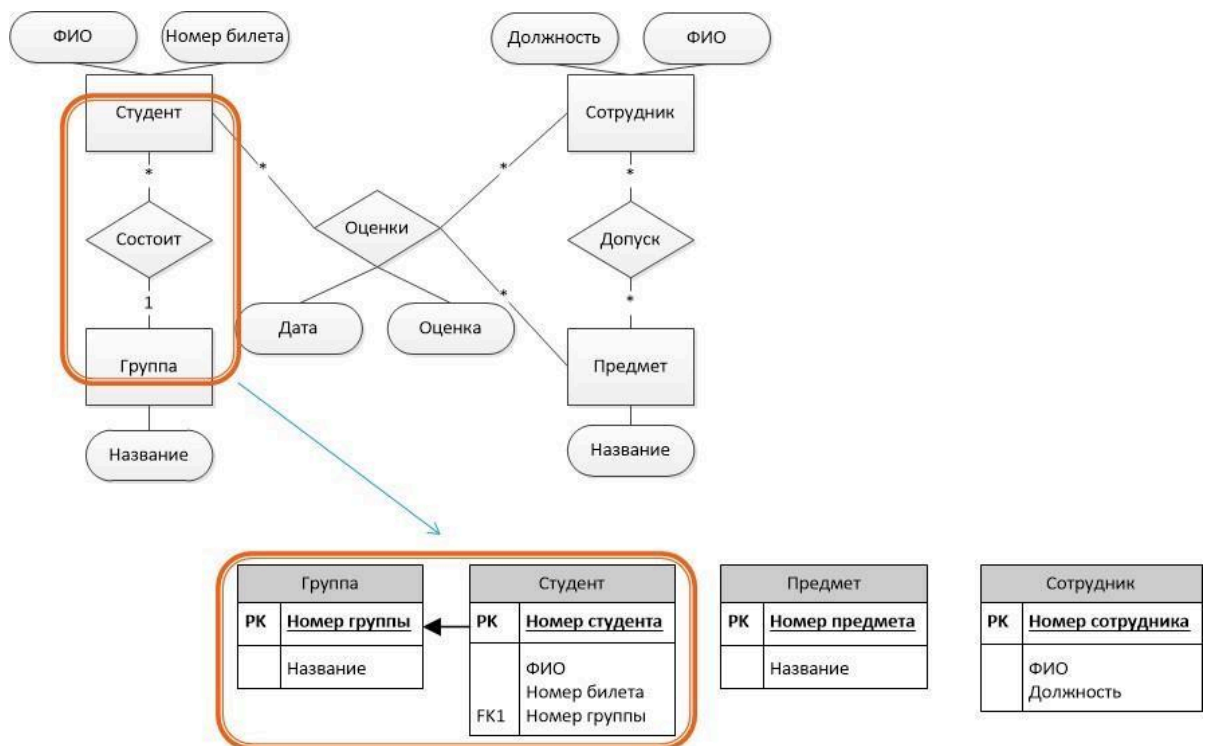
- для связей «*многие-ко-многим*» – в промежуточной таблице (при этом атрибуты, расширяющие комбинацию в связи (например, «*дата*»), также должны войти в состав *составного* первичного ключа промежуточной таблицы).

Ниже на рисунках приведены этапы выполнения шагов алгоритма для простой ER-диаграммы:

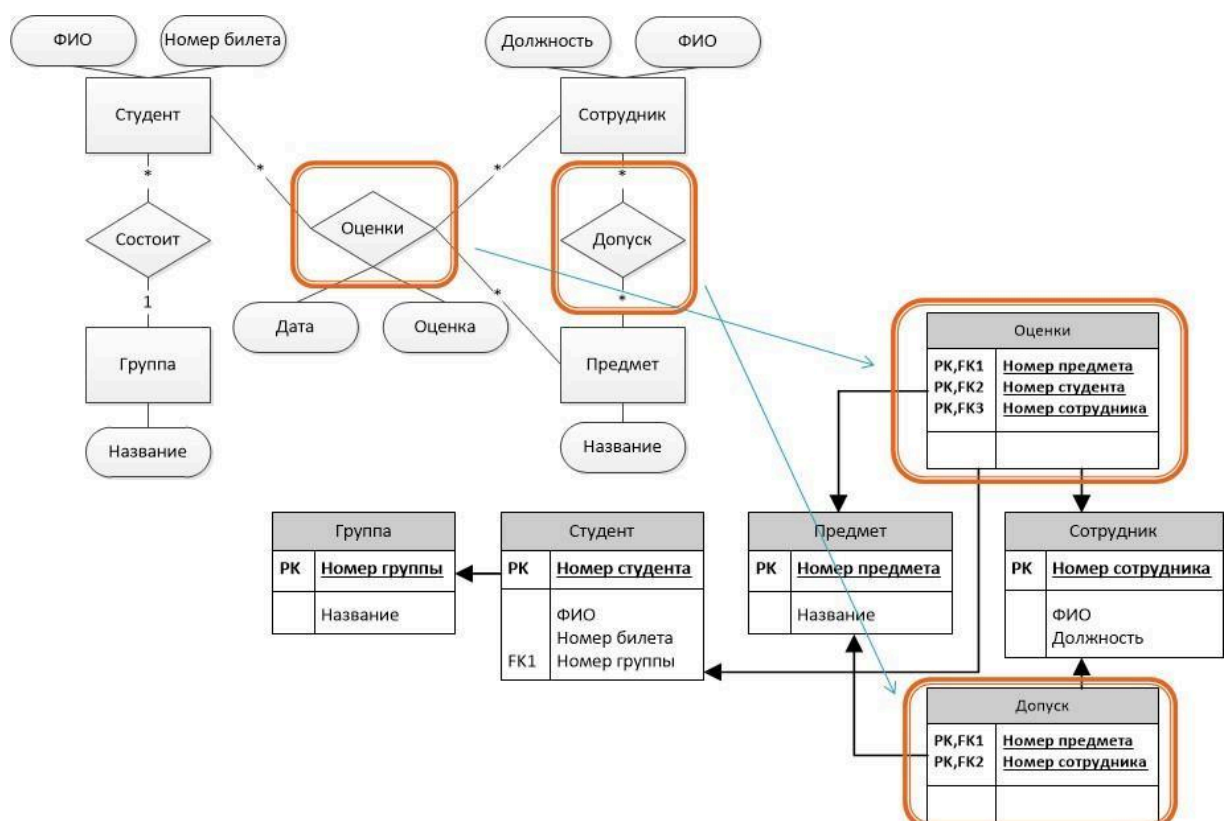
- результат выполнения шагов 1 – 3 алгоритма:



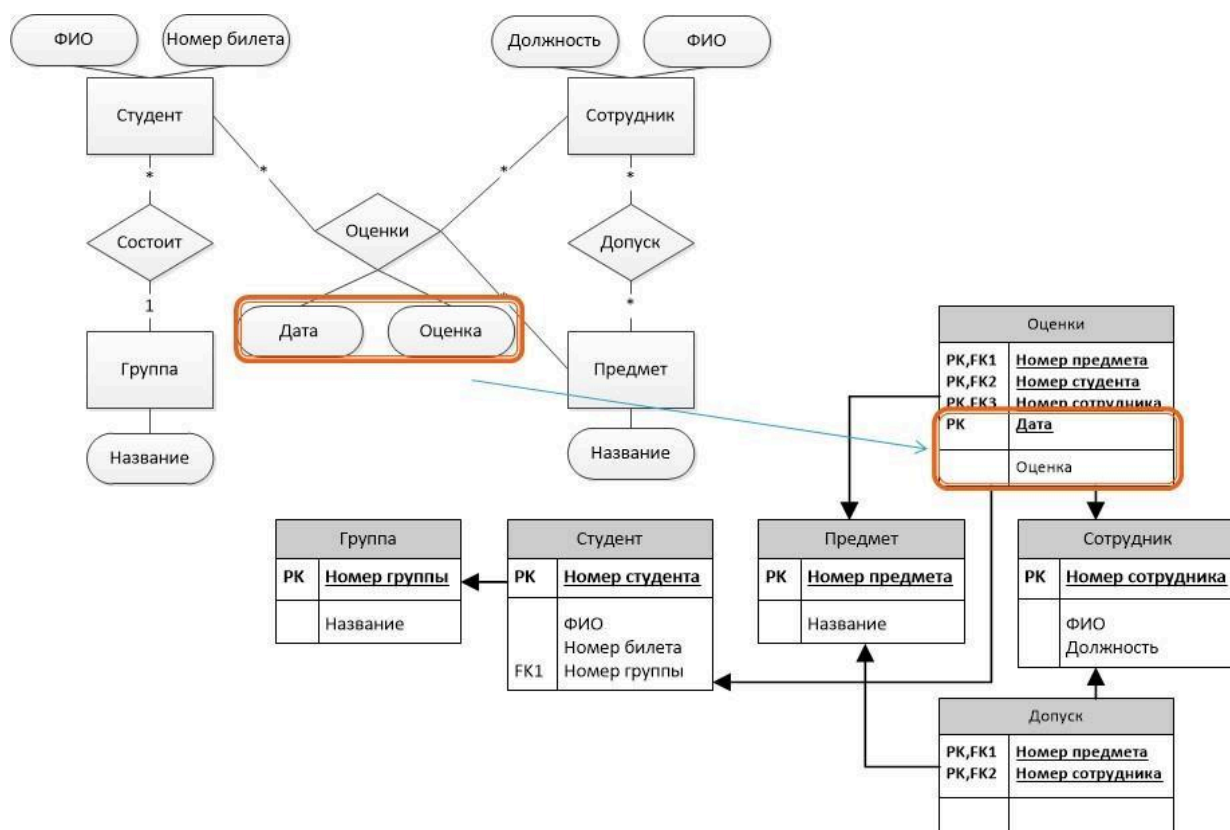
- результат выполнения шага 4 алгоритма



- результат выполнения шага 5 алгоритма:

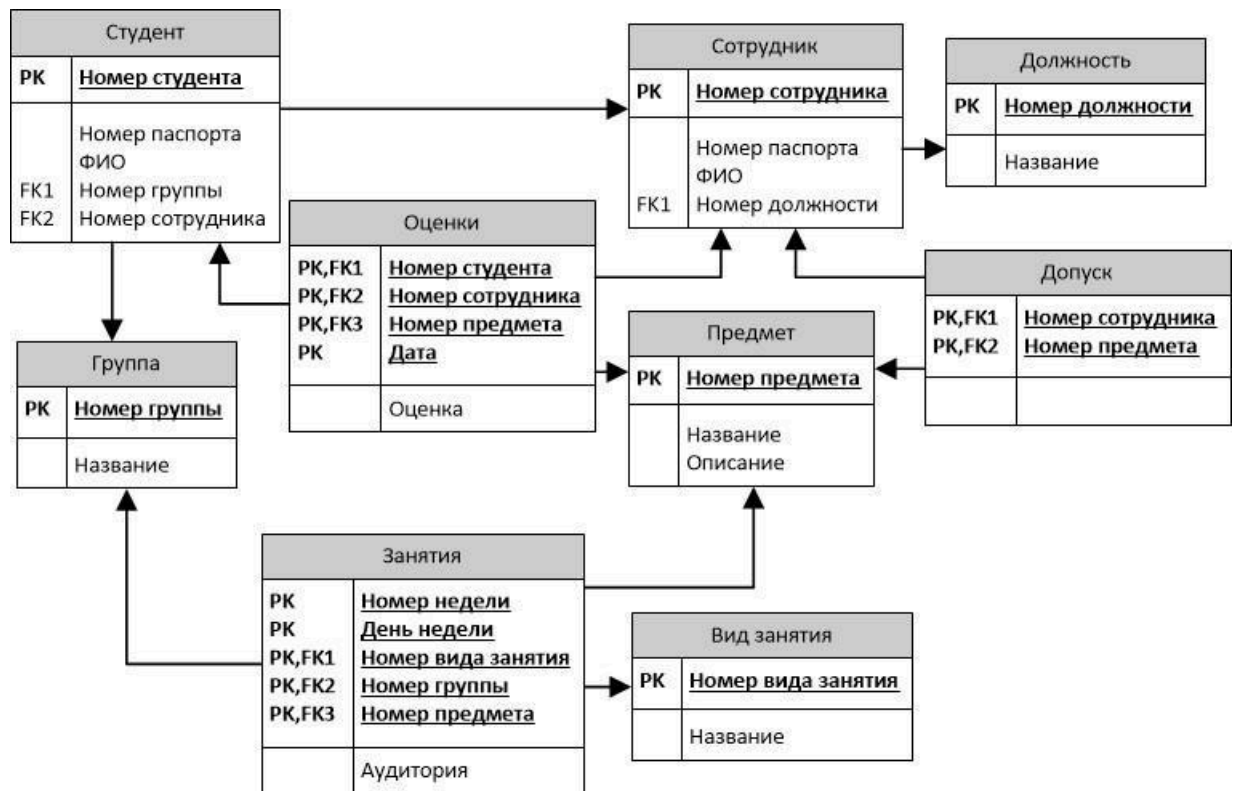


- результат выполнения шага 6 алгоритма:

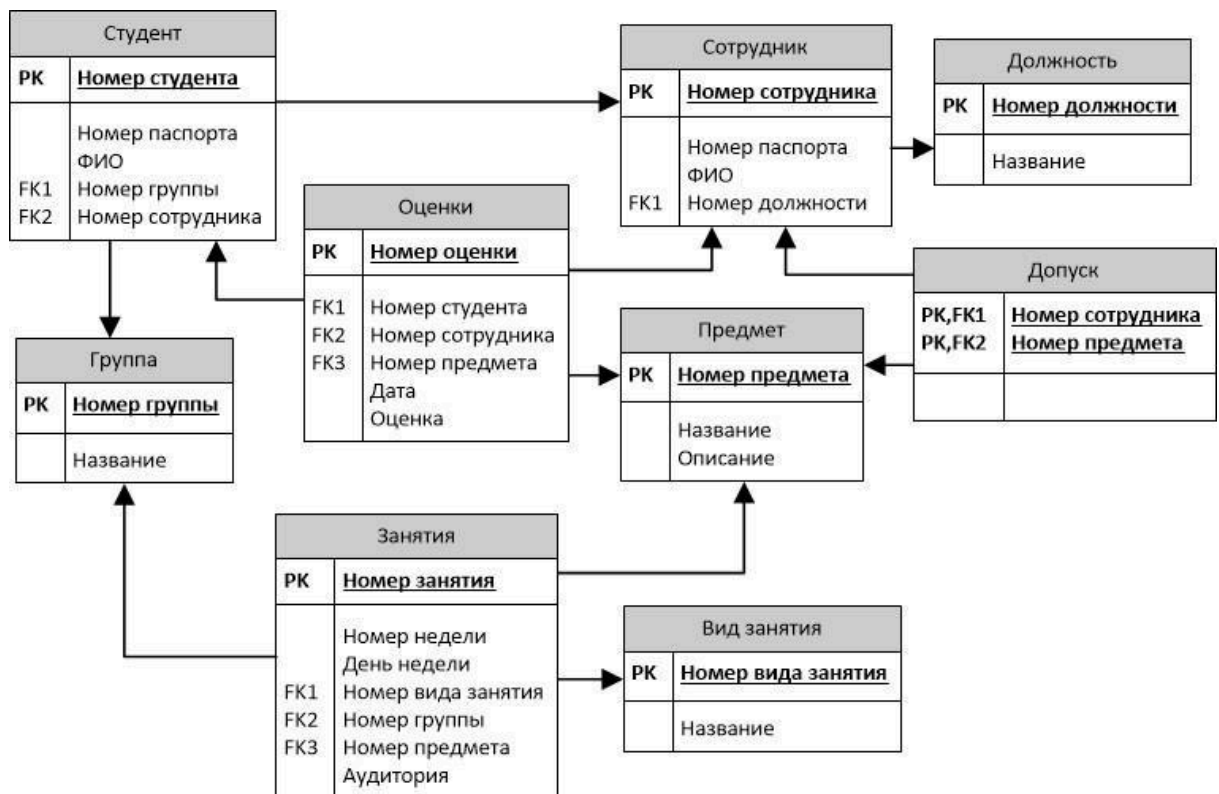


Для рассмотренной в лабораторной работе №1 концептуальной модели «Университет» ниже приведены два варианта UML-диаграмм реляционной схемы данных. Представленные варианты отличаются только принципом формирования первичного ключа в промежуточных таблицах, которые реализуют тернарные связи ER-диаграммы:

Вариант 1) первичные ключи в отношениях «Оценки» и «Занятия» построены как *составные* на основе ссылок на базовые объекты (*FK1 – FK3*) и атрибута «Дата» - такой первичный ключ позволяет контролировать уникальность комбинаций полей и блокировать добавление дубликатов (например, для отношения «Оценки» будет запрещено добавлять еще одну оценку по конкретному предмету конкретному студенту конкретным преподавателем в одну и ту же дату):



Вариант 2) первичные ключи в отношениях «Оценки» и «Занятия» построены как *простые* на основе суррогатного поля (PK) - такой первичный ключ просто реализуется и позволяет легко модернизировать БД:



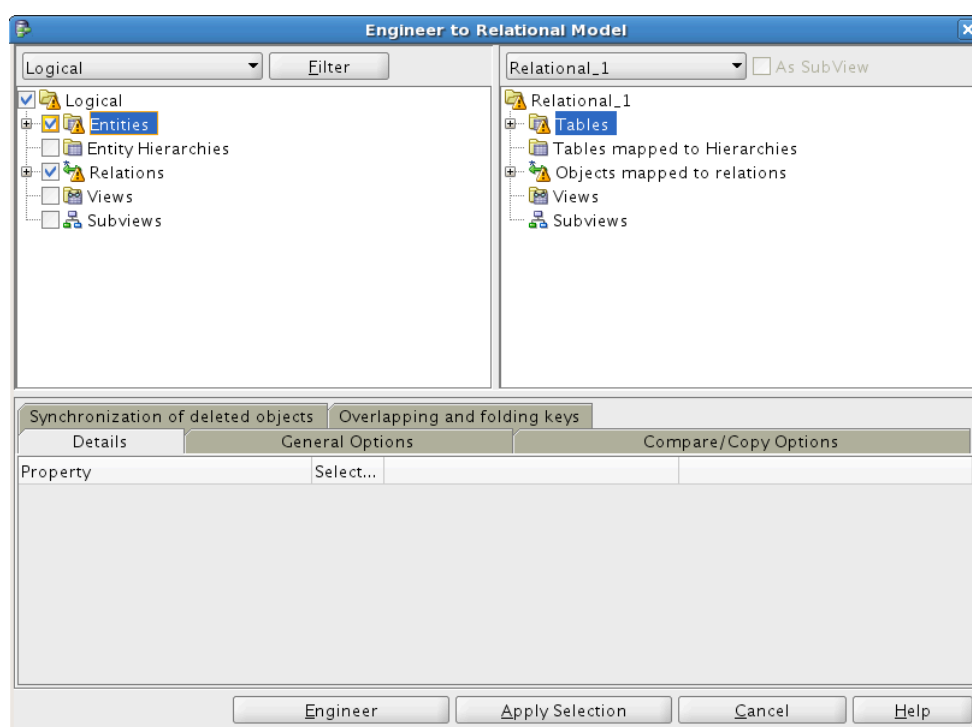
## Часть 2. Использование Oracle SQL Developer Data Modeler

В этой части кратко описаны этапы работы с CASE-средством Oracle SQL Developer Data Modeler (см. установку ПО в лабораторной работе №1) по переводу ER-диаграммы в реляционную диаграмму.

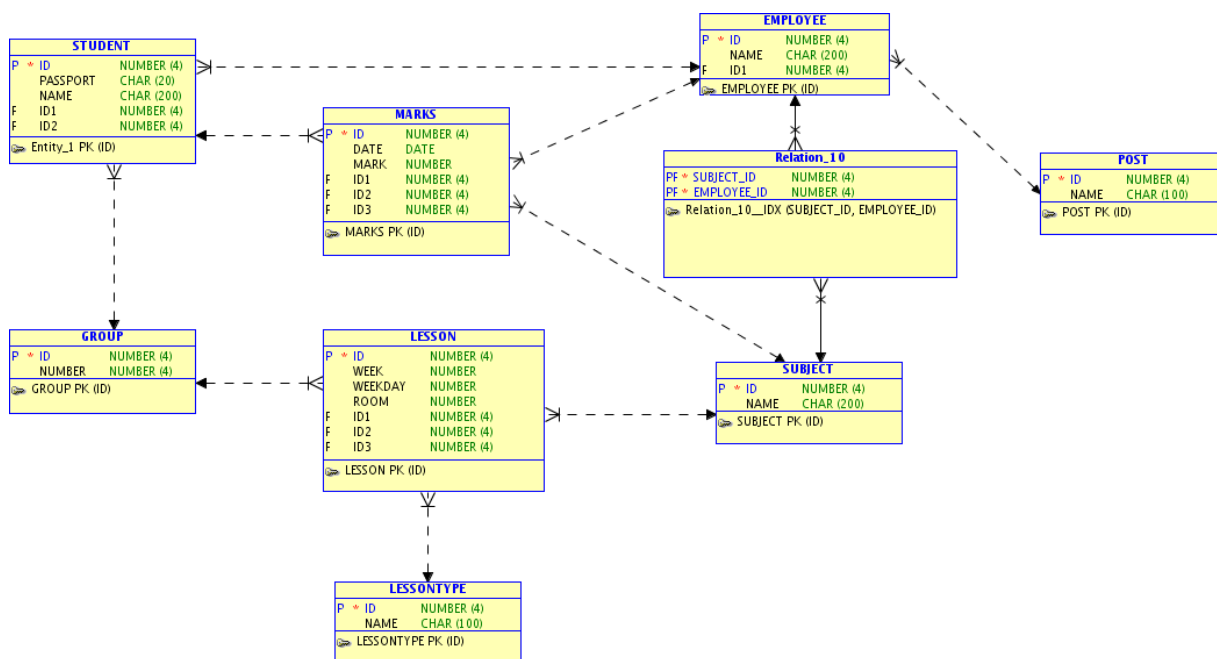
### Порядок выполнения работы

Преобразование ER-модели в реляционную модель:

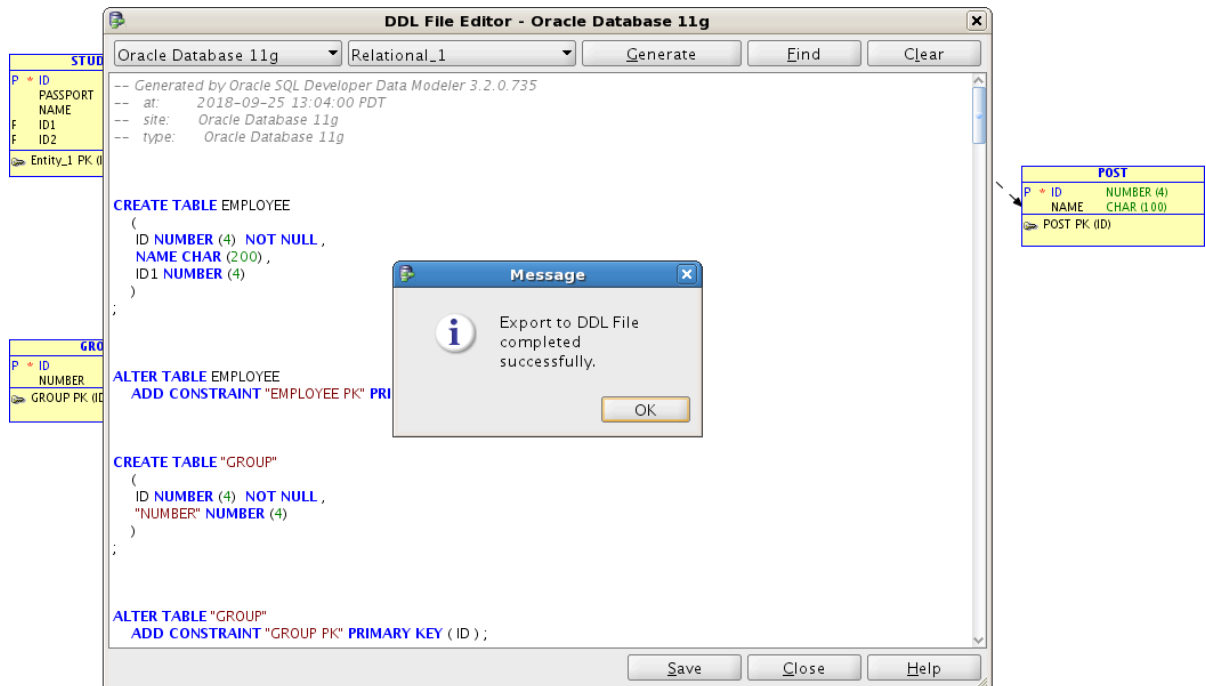
- Для выполнения преобразования требуется использовать пиктограмму «Engineer to Relational Model».



В этом окне можно настроить свойства преобразования диаграммы в табличную модель и нажать кнопку «Engineer». Получим реляционную модель (если надо переделать – выделить все и удалить в этом окне !!!):



- Созданную диаграмму можно сохранить в виде картинки (File → Print Diagram → To Image File (в формат .png), файл затем можно скопировать на флешку (FAT))).
- Полученную реляционную модель можно экспортировать (например, в DDL-файл: File → Export → DDL File, затем во всплывшем окне использовать кнопки Generate и Save. Получим текстовый файл с набором команд языка SQL для создания таблиц, который потом можно использовать в SQL Developer для создания собственной схемы данных (но обязательно провести его «ручную» доработку).



## Реляционная модель для ER-диаграммы «Университет»:

