|  |  |
| --- | --- |
| Курс | Основы технологий баз данных |
| Преподаватель | Евгений Станиславович Давыдов |
| Ученик | **Шауэрман Айнур** |
| **Домашнее задание №1** | Блок №2. Модели данных, теоретическая реляционная модель данных |

# Упражнение 2.4

В схеме с курсами и студентами предусмотрите возможность ведения занятий по курсу несколькими преподавателями. Экзамен может сдаваться не тому преподавателю, который вел занятия.

**Решение**:

Имеются сущности:

* студент
* преподаватель
* курс
* экзамен

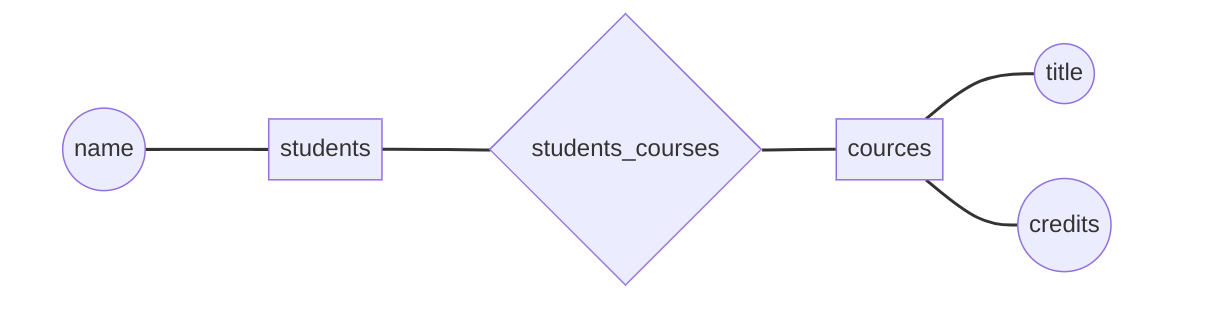
Определим связи между сущностями:

1) студент — курс → m:n

Студент может иметь несколько курсов.

По одному курсу может учиться множество студентов.

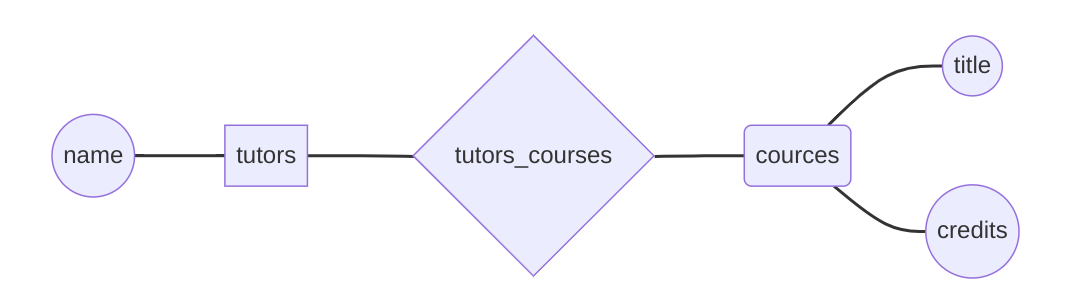
*Ограничение*: в одном курсе студент может быть закреплен только раз.



2) Преподаватель — курс → m:n

Преподаватель может вести несколько курсов.

Один курс может вести несколько преподавателей.



3) Студент — экзамен → m:n, преподаватель — экзамен → m:n

У одного студента может быть несколько экзаменов.

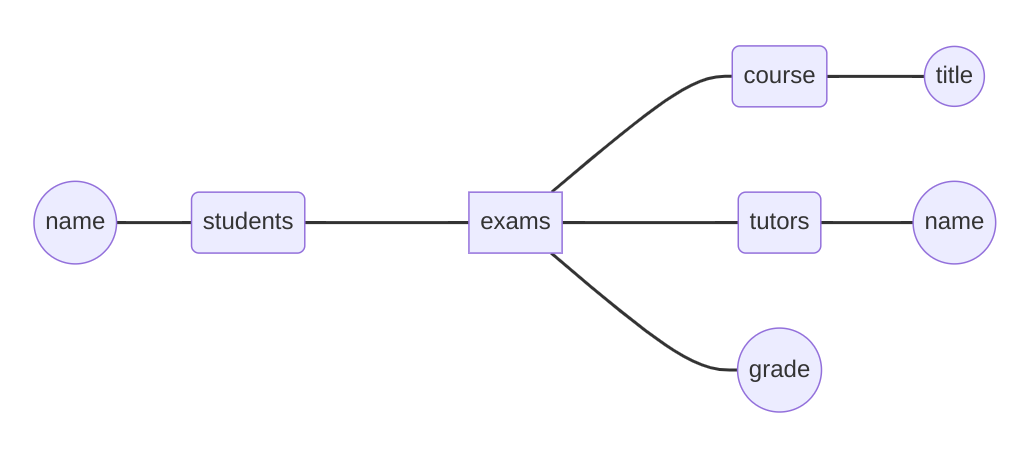
Один экзамен сдают множество студентов.

Преподаватель может принимать несколько экзаменов.

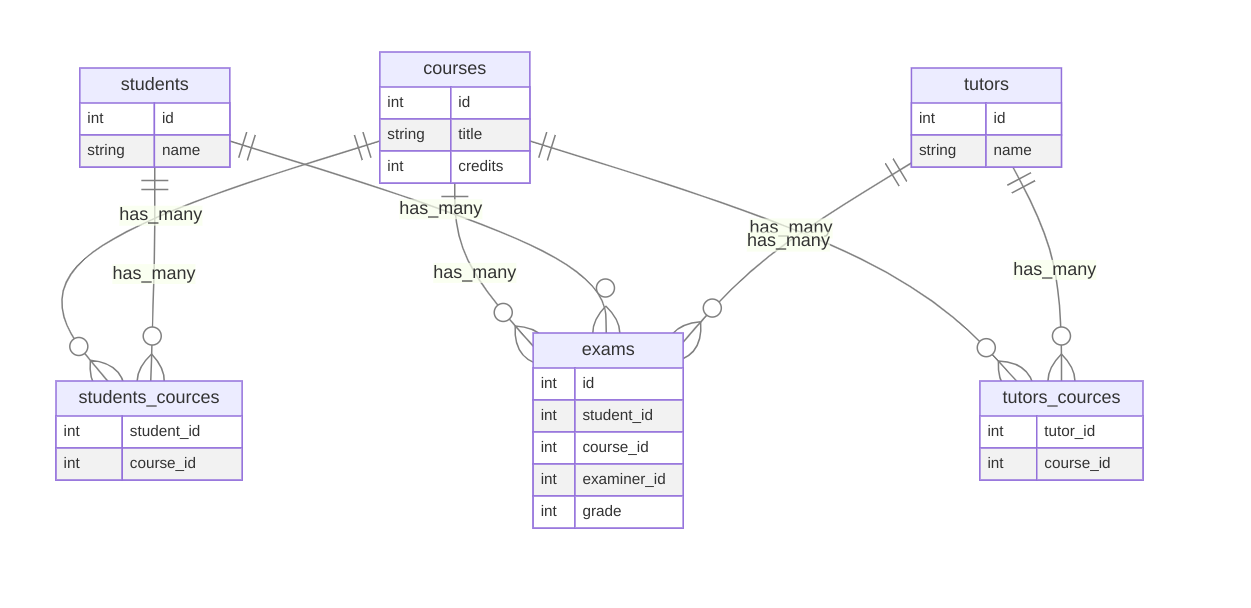
По одному курсу есть множество экзаменов.

*Ограничение*:

* Один экзамен имеет только одного экзаменатора.
* Один студент может сдавать экзамен по одному курсу не более N раз.



Итого, соберем все связи в одну картину в блок-схеме:



# Упражнение 2.5

Укажите отклонения от третьей нормальной формы в демонстрационной базе данных.

**Решение**:

В отношении tickets пассажир не выделен в отдельную сущность. Поэтому пассажир в отношений tickets находится в 1NF.

В отношений flights аэропорты отправления arrival\_airport и прибытия departure\_airport зависят только от номера рейса flight\_no, но не зависят от даты и времени вылета scheduled\_departure. Здесь присутствует зависимость от неполного ключа, и поэтому она не находится в 2NF.

В отношении seats есть функциональная зависимость fare\_conditions от места в самолете seat\_no. Потому что это закреплено на физическом уровне. Значит seat\_no тривиально определяет fare\_conditions.

В отношении ticket\_flights билет ticket\_no нетривиально (опосредованно) зависит от fare\_conditions . Вследствие этой зависимости в отношении присутствует избыточность: он повторяет атрибут значение одноименного атрибута seats.

fare\_condition является скалярной величиной, что в отношении seats, что в отношении ticket\_flights. Поэтому fare\_condition в этих отношениях находится в 1NF.

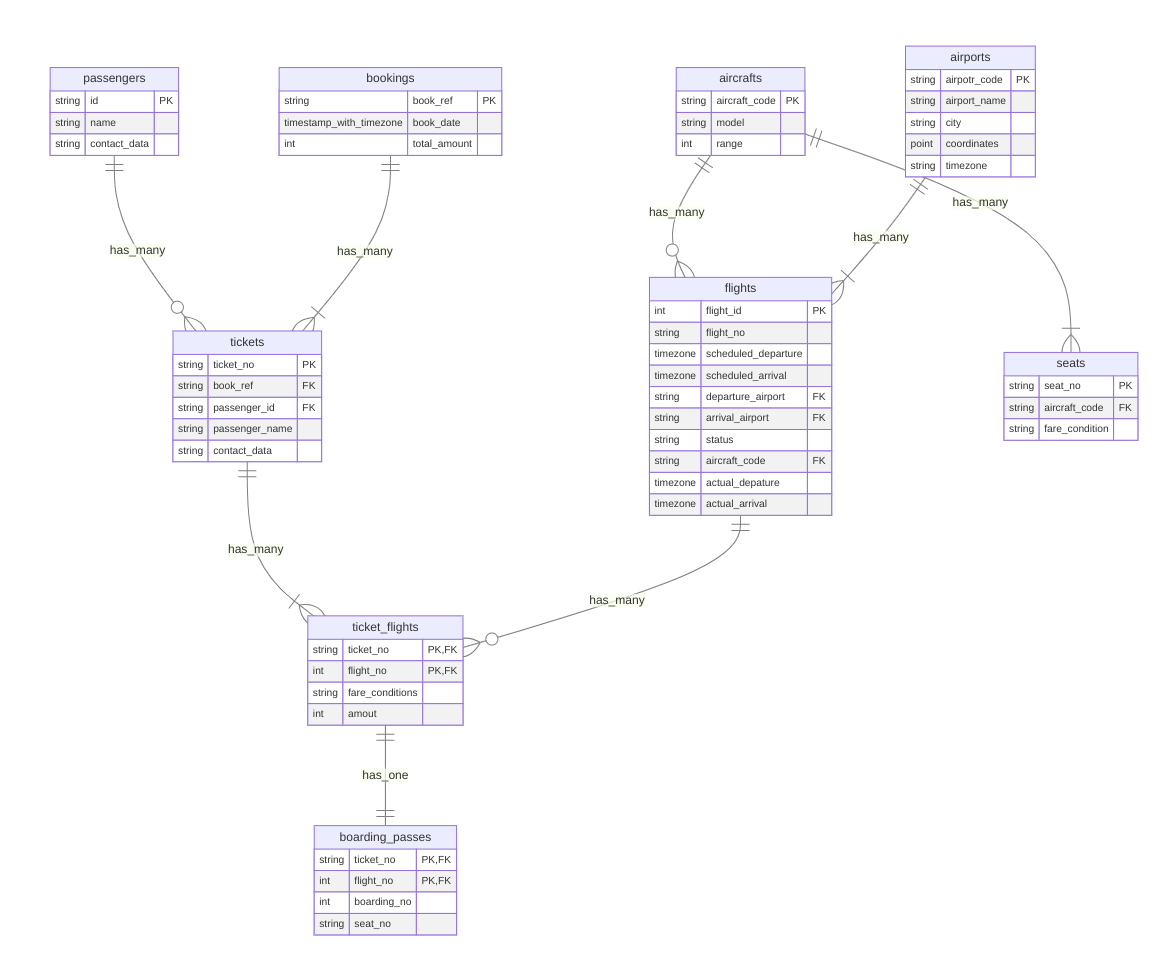
# Упражнение 2.7

Авиаперевозчики могут регистрировать постоянных клиентов. Если при бронировании указывается карточка постоянного клиента, информация о нем может быть скопирована в бронирование, но может быть и изменена. Добавьте в схему демонстрационной базы возможность хранения данных о постоянных клиентах.

**Решение**:

Данные о постоянных клиентах хранятся в отношений passengers.

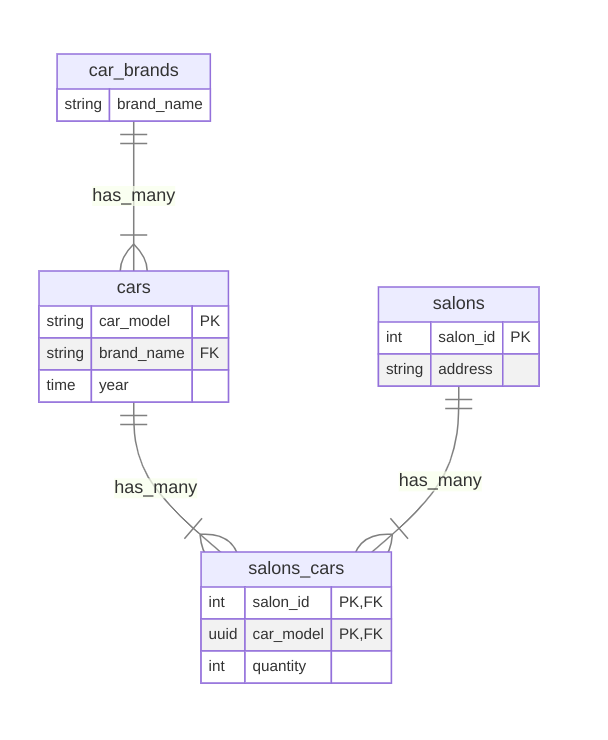
В отношений tickets есть колонки passenger\_id nullable foreign\_key. Есть ограничение, что если passenger\_id=null, passenger\_name is not null. Если клиент не постоянный, или по какой-то причине информация о постоянном клиенте должна быть изменена в билете, то такие данные хранятся в атрибутах passenger\_name, contact\_data.



# Упражнение 2.11

Создайте схему базы с информацией о наличии автомобилей в автомобильных салонах. Необходимо учитывать марку автомобиля, модель, год выпуска, адрес салона и т.п.

**Решение**:



Отношение car\_brands имеет атрибуты {названии} бренда.

Отношение cars имеет атрибуты {модель автомобиля, бренд в виде foreign\_key к отношению car\_brands , год выпуска}.

Отношение salons имеет атрибуты {идентификатор салона (даже физически допустимо, что по одному адресу может находится несколько салонов), адрес салона}.

Отношение salons\_cars сохраняет данные о том, какие виды машин и сколько есть в определнных салонах. Имеет атрибуты {идентификатор салона, идентификатор вида машны, количество таких машин в этом салоне}.