

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

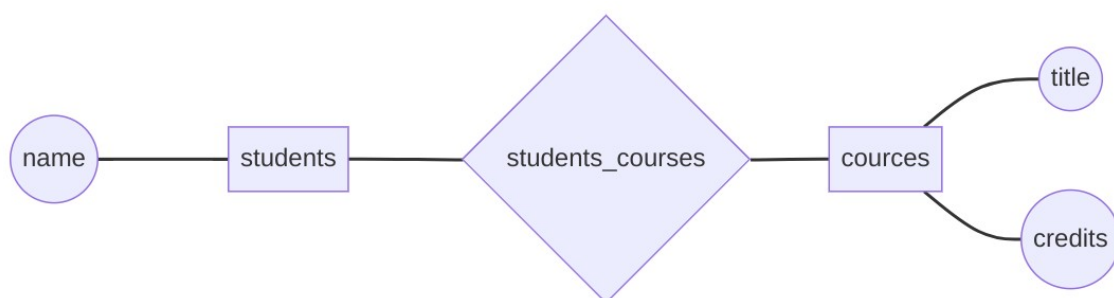
Определим связи между сущностями:

1) студент — курс → m:n

Студент может иметь несколько курсов.

По одному курсу может учиться множество студентов.

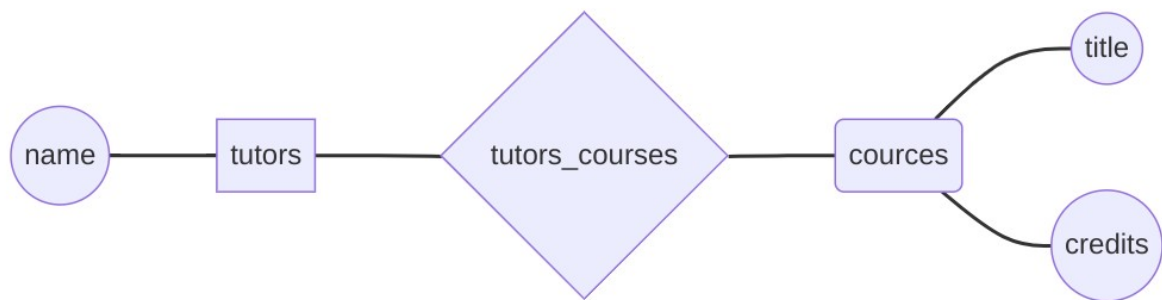
Ограничение: в одном курсе студент может быть закреплен только раз.



2) Преподаватель — курс → m:n

Преподаватель может вести несколько курсов.

Один курс может вести несколько преподавателей.



3) Студент — экзамен $\rightarrow m:n$, преподаватель — экзамен $\rightarrow m:n$

У одного студента может быть несколько экзаменов.

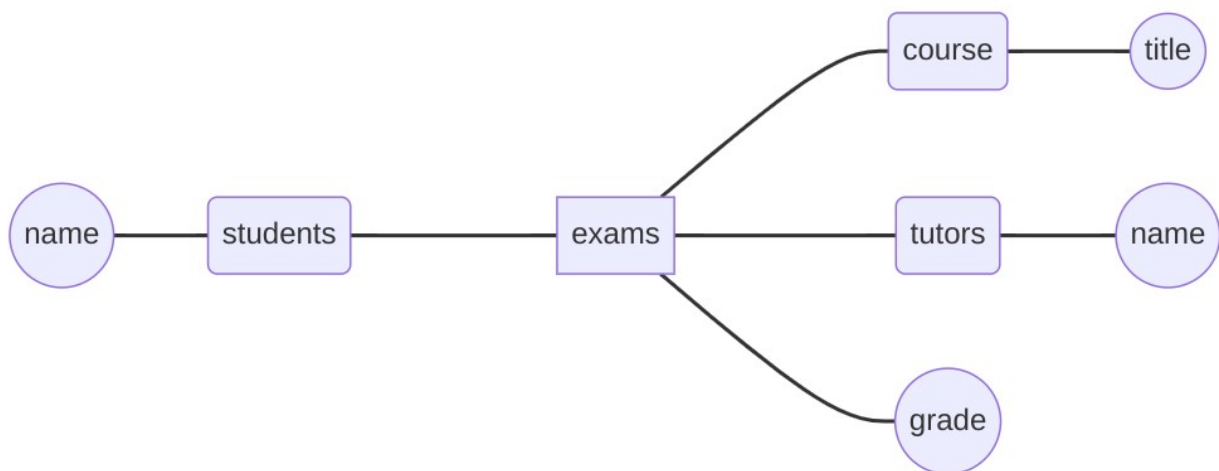
Один экзамен сдают множество студентов.

Преподаватель может принимать несколько экзаменов.

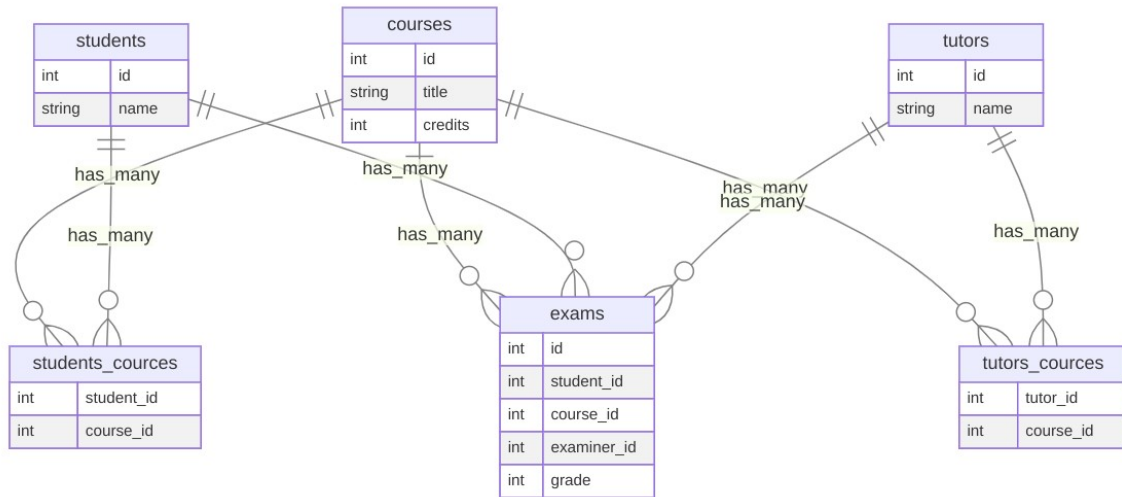
По одному курсу есть множество экзаменов.

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 отношении tickets находится в 1NF.

В отношении 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.

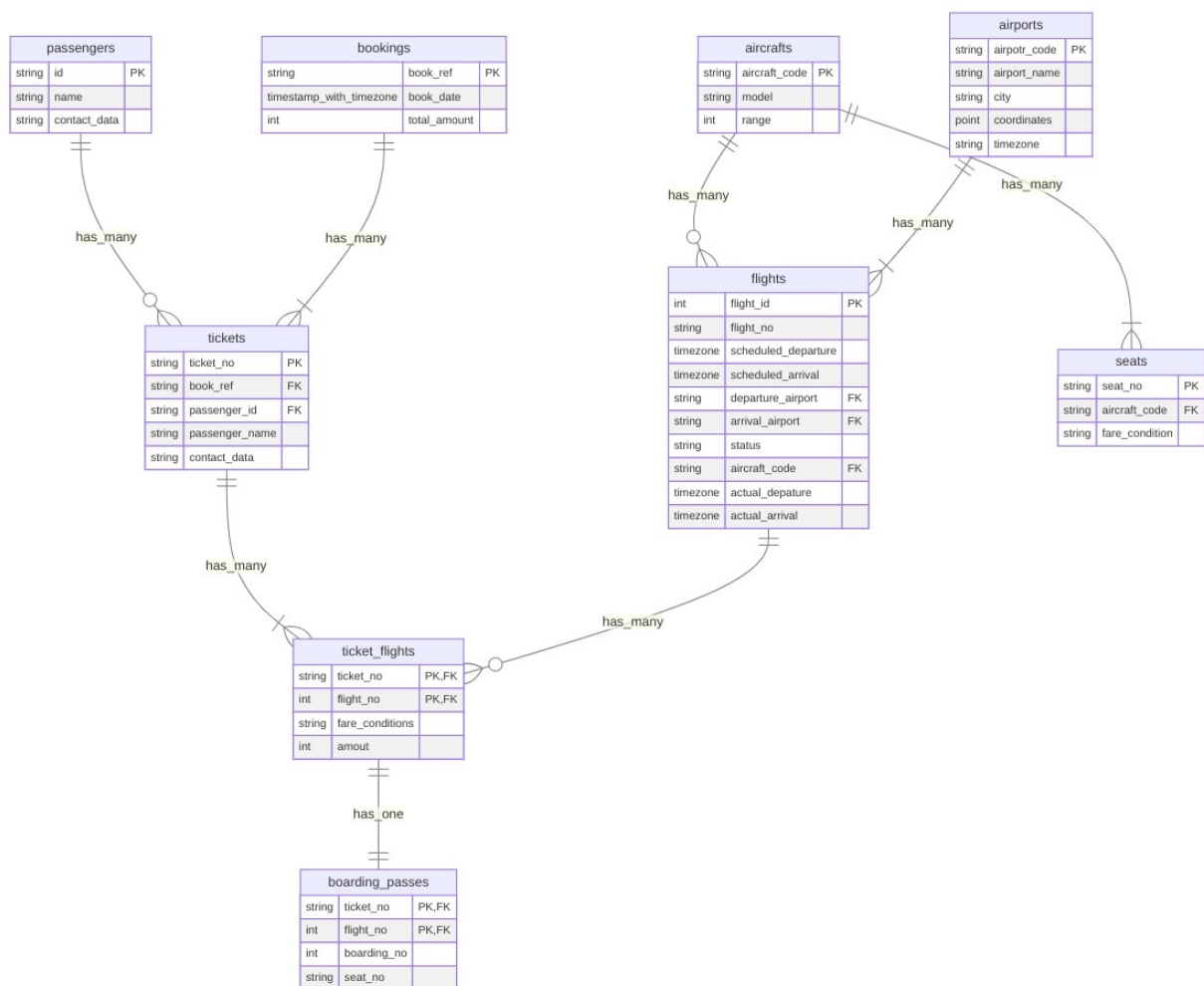
Упражнение 2.7

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 отношении passengers.

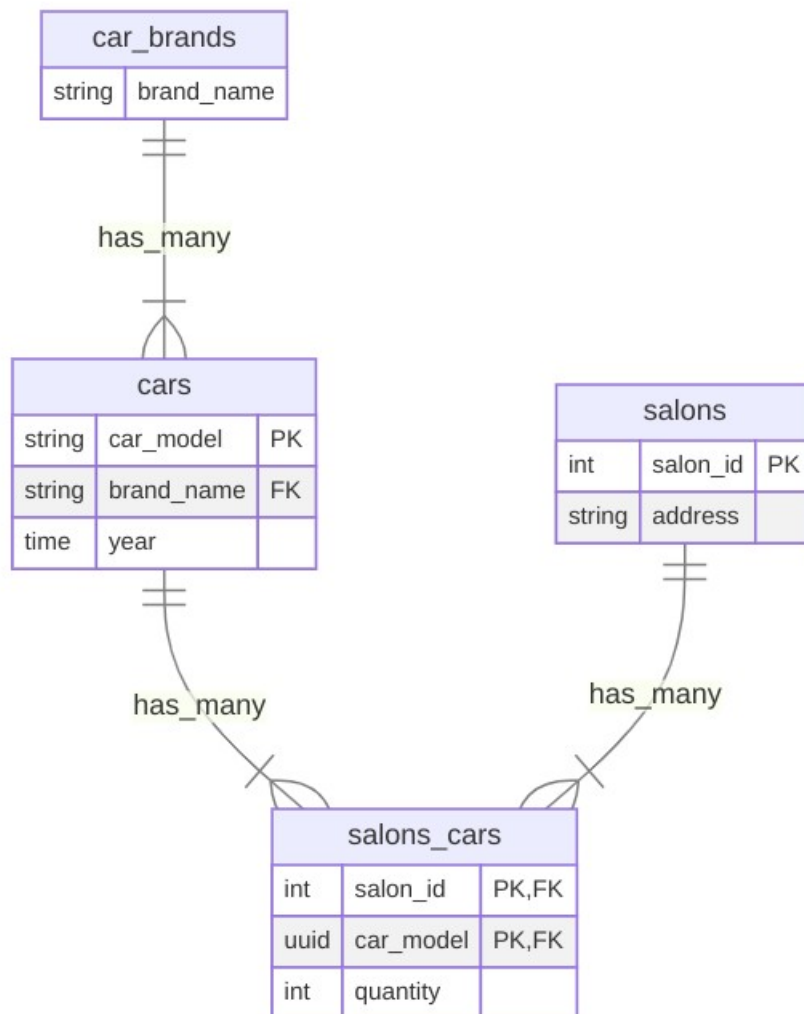
В отношении tickets есть колонки passenger_id nullable foreign_key. Есть ограничение, что если passenger_id=null, passenger_name is not null. Если клиент не постоянный, или по какой-то причине информация о постоянном клиенте должна быть изменена в билете, то такие данные хранятся в атрибутах passenger_name, 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 находиться несколько салонов), адрес салона}.

Отношение `salons_cars` сохраняет данные о том, какие виды машин и сколько есть в определенных салонах. Имеет атрибуты {идентификатор салона, идентификатор вида машины, количество таких машин в этом салоне}.