

Моделирование. Концептуальный уровень

После того как вы построили контекстную диаграмму данных, следующим шагом является моделирование концептуальной модели базы данных. Концептуальная модель базы данных — это высокоуровневое представление структуры данных, которое абстрагируется от технических деталей хранения данных и сосредотачивается на **выявлении сущностей** (которые нужны на м в базе данных) и **отношениях между этими сущностями** в системе.

Концептуальная модель нужна для того, чтобы выявить и организовать основные сущности в системе и их взаимосвязи, что является важным для понимания общей информационной структуры и удовлетворения бизнес-требований. То есть это первый этап, когда мы строим бизнесовый фундамент нашей базы данных. Концептуальная модель служит связующим звеном между потребностями пользователя и физическим проектированием базы данных. (В редких случаях концептуальное проектирование можно пропустить, когда вы ясно понимаете, что у вас всего пара сущностей, одна связь между ними и будет очень маленькая база данных).

Концептуальную модель обычно проектируют бизнес-аналитики, системные аналитики, которые работают в тесном сотрудничестве с заинтересованными сторонами и пользователями системы. Эти специалисты используют концептуальную модель для обеспечения точного отражения бизнес-логики в структуре данных и для облегчения общения и понимания между техническими и не техническими участниками проекта.

Концептуальную модель базы данных, иногда называют - **модель предметной области**, то есть - абстрактное представление данных и их взаимосвязей в контексте интересующей области (например, автомобиль в контексте автолизинга будет сущностью "предмет лизинга", а в контексте автодилера будет как "товар" - поэтому аналитики выявляют нужные сущности и связи именно в нужном контексте, нужной предметной области).

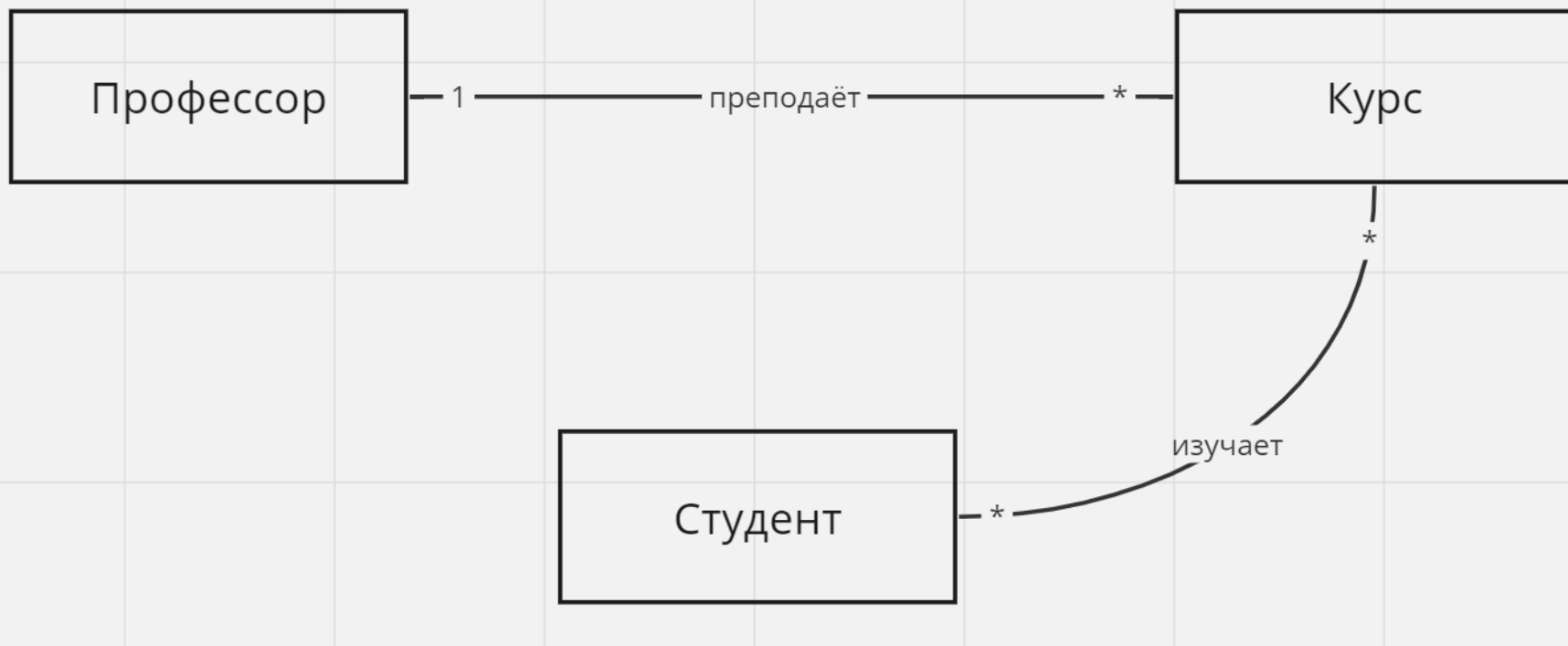
Сущность в концептуальной модели — это объект с какими-то уникальными характеристиками. Каждая сущность характеризуется набором атрибутов, которые описывают ее свойства (атрибуты описываются на логическом уровне, на концептуальном выделяются только сущности). Сущность может быть реальной (ноутбук, машина) или абстрактной (событие, услуга).

***Часто вы встретите формулировку "экземпляр сущности". Что это такое? В информационной системе объект описывается через его конкретные экземпляры (пример объекта). То есть, конкретными экземплярами объекта(то есть объект с конкретными свойствами) "Город" могут быть такие города, как "Москва"(объект Город со свойством "название"="Москва"), "Санкт-Петербург", "Новосибирск". У объекта машины экземпляры - "Лада"(объект машина со свойством "марка"="Лада"), "Мерседес" и так далее. Фишка в том, что мы мыслим сущностями (для каждой сущности будет таблица, например), но каждая строка в этой таблице - это будет конкретный экземпляр этой сущности.

Связи между сущностями отображают, как объекты связаны друг с другом. Кратность (виды) связей бывают такими:

- **"Один-к-одному" (1:1):** Каждый сотрудник компании имеет уникальный рабочий идентификатор. Здесь сотрудник и его рабочий идентификатор находятся в отношении "один-к-одному", поскольку каждому сотруднику соответствует только один уникальный идентификатор, и каждый идентификатор связан только с одним сотрудником.
- **"Один-ко-многим" (1:*):** Профессор может вести несколько разных курсов в университете. В этом случае между профессором и курсами существует связь "один-ко-многим", где один профессор связан с несколькими курсами, но каждый курс ведет только один профессор.
- **"Многие-ко-многим" (*:*):** Студенты могут записываться на разные курсы, а каждый курс может посещать множество студентов. Здесь у нас связь "многие-ко-многим", так как множество студентов могут посещать множество курсов, и на каждом курсе могут быть разные студенты.

Давайте рассмотрим очень простую концептуальную модель для университетской системы, которая включает в себя сущности "Профессор", "Курс" и "Студент".



Сущности:

1. **Профессор**
2. **Курс**
3. **Студент**

Связи:

1. **Преподает** (один-ко-многим)

- Один профессор может преподавать несколько курсов. Связь между сущностью "Профессор" и "Курс" указывает на это отношение.

2. **Изучает** (многие-ко-многим)

- Студенты могут посещать несколько курсов, и каждый курс может быть посещен многими студентами. Связь между "Студент" и "Курс" отражает эту множественность.

В этом примере, концептуальная модель помогает установить, как именно профессора связаны с курсами, которые они преподают, и как студенты взаимодействуют с этими курсами. Это высокоуровневое представление не вдаётся в подробности реализации на уровне базы данных, но даёт чёткое представление о структуре данных, которая потребуется для поддержки этих отношений в системе для университета.