ER модель данных, сущностей. Элементы ER-диаграммы.

Лектор: PhD, и.о. доцента Турарбек Әсем Турарбекқызы

Цель лекции

Изучить принципы концептуального моделирования баз данных с использованием ER-модели и правила построения ER-диаграмм.

Результаты обучения

По завершении лекции студент должен:

1. Определять сущности, атрибуты и связи предметной области.
2. Различать типы атрибутов, слабые сущности и кардинальности связей.
3. Строить ER-диаграммы и преобразовывать их в реляционные схемы.

ЧТО ТАКОЕ ER-МОДЕЛЬ?

Er-модель — это этап логического анализа предметной области, Предшествующий нормализации и построению реляционной схемы.

Модель "сущность- связь“ (entity-relationship, er) — это высокоуровневая концептуальная модель, разработанная для Отображения семантики данных в предметной области,
Не зависящей от субд.
Она описывает объекты, их свойства и взаимосвязи между ними.

Процесс построения ER-модели

1. Выявление сущностей (кто или что хранится в базе?).
2. Определение атрибутов для каждой сущности.
3. Определение связей между сущностями.
4. Указание кардинальности и обязательности.
5. Обнаружение слабых сущностей, их идентификация.
6. Уточнение атрибутов связей, если требуется.

Основные компоненты ER-модели

Компонент	Обозначение	Описание
Сущность (Entity)	Прямоугольник	Объект или предмет предметной области, о котором нужно хранить данные.
Атрибут (Attribute)	Овал	Свойство или характеристика сущности или связи.
Первичный ключ	Подчёркнутый атрибут	Атрибут, уникально идентифицирующий экземпляр сущности.
Связь (Relationship)	Ромб	Отношение между двумя или более сущностями.
Кардинальность	Числовое обозначение (1:1, 1:N, M:N)	Количество экземпляров, участвующих в связи.
Слабая сущность	Двойной прямоугольник	Сущность, не имеющая собственного ключа, зависит от другой сущности.
Связь идентификации	Двойной ромб	Связь, которая используется для идентификации слабой сущности.
Множественный атрибут	Двойной овал	Атрибут, который может иметь несколько значений.
Производный атрибут	Пунктирный овал	Атрибут, значение которого можно вычислить из других атрибутов.
Обязательный/необязательный атрибут	Зависит от обозначения (например, * для обязательного)	Определяет, обязательно ли указывать значение.

Основные компоненты ER-модели

Сущности (Entities)

Сущность — это объект или понятие из предметной области, о котором необходимо хранить информацию.

Виды:
Сильные (регулярные): имеют свой первичный ключ (например, Сотрудник(ID)).
Слабые: зависят от других сущностей и не имеют собственного уникального ключа (например, Зависимые сотрудника).

Атрибуты (Attributes)

Атрибут — это характеристика или свойство сущности или связи.

Согласно Elmasri & Navathe:

Простые (атомарные) — нельзя разделить: Имя, Номер.

Составные — включают податрибуты: Адрес → Улица, Город, Индекс.

Многозначные — могут принимать несколько значений: Телефон.

Производные — могут быть выведены из других: Возраст ← Дата рождения.

Ключевые атрибуты подчёркиваются: они формируют **идентификаторы** сущностей.

Связи (Relationships)

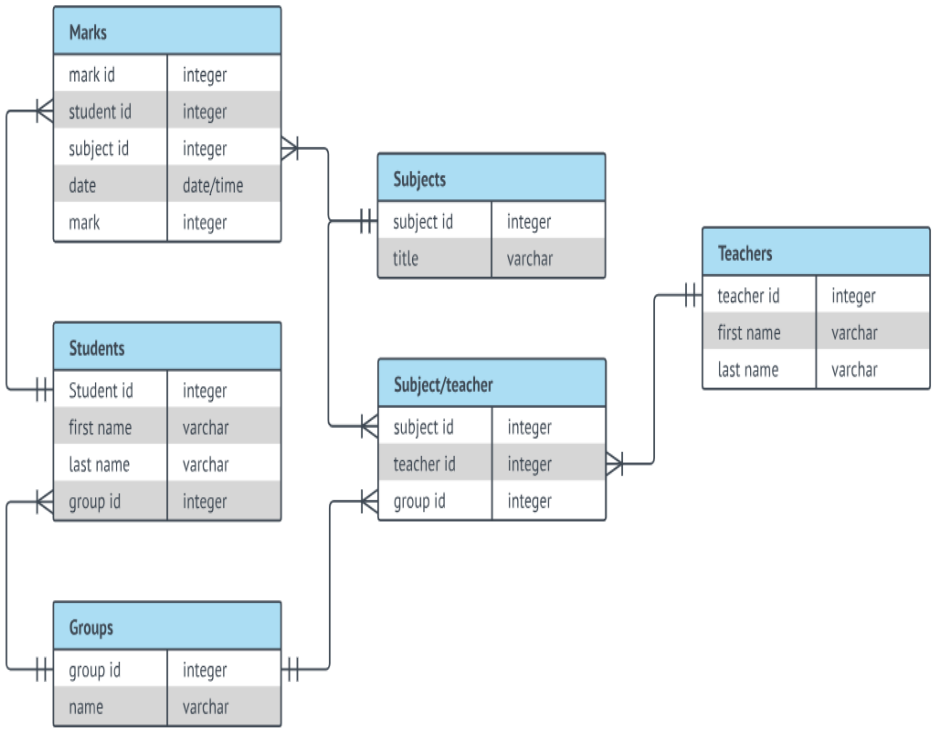
Связь (relationship) — это логическая ассоциация между двумя или более сущностями.

Классификация по мощности (кардинальности):

- 1:1 — каждый экземпляр одной сущности связан с максимум одним другим.
- 1:N — одному экземпляру одной сущности соответствует множество экземпляров другой.
- M:N — множеству экземпляров одной сущности соответствует множество другой.

Пример связи:

✓ Если у нас есть Студент и Курс, то:
✓ Студент — сущность
✓ Курс — сущность
✓ ЗаписанНа — связь
✓ НомерЗачетки, Имя — атрибуты студента
✓ КодКурса, НазваниеКурса — атрибуты курса
✓ Оценка — атрибут связи (если он важен)



Переход от ER-модели к реляционной

На основании ER-диаграммы строится **логическая модель** (реляционная схема):

- Каждая **сущность** → отдельная таблица.
- Каждый **атрибут** → столбец.
- Каждая **связь**:
  - 1:N → добавление внешнего ключа;
  - M:N → создаётся отдельная таблица для связи;
  - Слабая сущность → преобразуется с помощью внешнего ключа + частичного ключа

В ER-моделях и моделях данных обычно выделяют до трех уровней детализации:

- Концептуальная модель данных — схема наивысшего уровня с минимальным количеством подробностей. Достоинство этого подхода заключается в возможности отобразить общую структуру модели и всю архитектуру системы. Менее масштабные системы могут обойтись и без этой модели. В этом случае можно сразу переходить к логической модели.
- Логическая модель данных содержит более подробную информацию, нежели концептуальная модель. На этом уровне определяются более подробные операционные и транзакционные сущности. Логическая модель не зависит от технологии, в которой она будет применяться.
- Физическая модель данных: на основе каждой логической модели данных можно составить одну или две физических модели. В последних должно присутствовать достаточно технических подробностей для составления и внедрения самой базы данных.

Заключение

ER-модель позволяет представить структуру предметной области через сущности, их свойства и взаимосвязи. ER-диаграммы используются для концептуального проектирования базы данных и служат основой построения реляционной схемы.

Вопросы для самопроверки

1. Что такое ER-модель данных?
2. Какие элементы входят в ER-диаграмму?
3. Чем сильная сущность отличается от слабой?
4. Какие типы кардинальности связей существуют?
5. Как преобразуется связь M:N в реляционную схему?

Список использованных источников

1. Chen P. P. S. The Entity-Relationship Model—Toward a Unified View of Data // ACM Transactions on Database Systems. 1976. Vol. 1, No. 1. P. 9–36.
2. Elmasri R., Navathe S. B. *Fundamentals of Database Systems*. 7th ed. Pearson, 2016.
3. Connolly T., Begg C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Pearson, 2015.
4. Silberschatz A., Korth H. F., Sudarshan S. *Database System Concepts*. 7th ed. McGraw-Hill, 2020.
5. Garcia-Molina H., Ullman J. D., Widom J. *Database Systems: The Complete Book*. 2nd ed. Pearson, 2009.
6. Date C. J. *An Introduction to Database Systems*. 8th ed. Addison-Wesley, 2003.
7. Ramakrishnan R., Gehrke J. *Database Management Systems*. 3rd ed. McGraw-Hill, 2003.
8. PostgreSQL Global Development Group. *PostgreSQL Documentation: CREATE TABLE*. https://www.postgresql.org/docs/current/sql-createtable.html

СПАСИБО ЗА ВНИМАНИЕ!