ТИПЫ СВЯЗЕЙ МЕЖДУ ОТНОШЕНИЯМИ, ДОБАВЛЕНИЕ ВНЕШНЕГО КЛЮЧА

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

Цель лекции

Изучить основные типы связей между таблицами реляционной базы данных и способы их реализации с использованием внешних ключей.

Результаты обучения

По завершении лекции студент должен:

1. Различать связи 1:1, 1:N и M:N.
2. Реализовывать связи между таблицами с помощью FOREIGN KEY.
3. Применять связующие таблицы и правила ON DELETE и ON UPDATE.

Связи

Связи между таблицами в реляционной модели описывают, как строки одной таблицы соответствуют строкам другой, и реализуются через внешние ключи (foreign keys).

Виды связей:

1. Связь "один к одному" (1:1)
2. Связь "один ко многим" (1:N)
3. Связь "многие ко многим" (M:N)

Связь "один к одному" (1:1)

Каждой записи в одной таблице соответствует **не более одной** записи в другой и наоборот.

Пример:

Таблица Человек и таблица Паспорт
Один человек ↔ один паспорт

Реализация:

В одной из таблиц добавляется внешний ключ, с ограничением UNIQUE.

```
CREATE TABLE Паспорт (
    Паспорт_ID INT PRIMARY KEY,
    Человек_ID INT UNIQUE,
    FOREIGN KEY (Человек_ID) REFERENCES Человек(Человек_ID)
);
```

Связь "один ко многим" (1:N)

Каждой записи в таблице A соответствует **несколько** записей в таблице B, но каждой записи в B — **ровно одна** запись в A.

Пример:

Клиент — Заказ
Один клиент может сделать много заказов, каждый заказ относится к одному клиенту.

Реализация:

В таблицу «многие» добавляется внешний ключ на таблицу «один».

sql

КопироватьРедактировать

```
CREATE TABLE Заказ (
    Заказ_ID INT PRIMARY KEY,
    Клиент_ID INT,
    FOREIGN KEY (Клиент_ID) REFERENCES Клиент(Клиент_ID)
);
```

Связь "многие ко многим" (M:N)

Каждой записи в одной таблице может соответствовать **много** записей в другой и наоборот.

Пример:

Таблицы Студент и Курс
Один студент может записаться на несколько курсов, один курс может быть у многих студентов.

Реализация:

Создаётся промежуточная (связующая) таблица с двумя внешними ключами.

```
CREATE TABLE Студент_Курс (
    Студент_ID INT,
    Курс_ID INT,
    PRIMARY KEY (Студент_ID, Курс_ID),
    FOREIGN KEY (Студент_ID) REFERENCES Студент(Студент_ID),
    FOREIGN KEY (Курс_ID) REFERENCES Курс(Курс_ID)
);
```

ВНЕШНИЙ КЛЮЧ (FOREIGN KEY)

Внешний ключ FOREIGN KEY

Внешний ключ — это столбец или группа столбцов в таблице реляционной базы данных, которая обеспечивает связь между данными в двух таблицах.

Он действует как перекрестная ссылка между таблицами, поскольку ссылается на первичный ключ другой таблицы, тем самым устанавливая связь между ними.

Создание внешнего ключа

В момент создания таблицы:

```
CREATE TABLE Заказ (
    Заказ_ID INT PRIMARY KEY,
    Клиент_ID INT,
    FOREIGN KEY (Клиент_ID)
REFERENCES Клиент(Клиент_ID)
);
```

Или после, с помощью ALTER TABLE:

sql

КопироватьРедактировать

```
ALTER TABLE Заказ
ADD CONSTRAINT FK_Клиент
FOREIGN KEY (Клиент_ID) REFERENCES
Клиент(Клиент_ID);
```

```
FOREIGN KEY (Клиент_ID)
REFERENCES
Клиент(Клиент_ID)
ON DELETE CASCADE
ON UPDATE CASCADE
```

Поведение при удалении/обновлении:

С помощью параметров ON DELETE и ON UPDATE можно задать поведение внешнего ключа:

CASCADE — при удалении/изменении родительской строки изменяются и дочерние

SET NULL — дочерние значения становятся NULL

SET DEFAULT — устанавливается значение по умолчанию

NO ACTION / RESTRICT — отклоняет операцию, если есть связанные записи

Пример

Анализ:

ISBN → Название, Автор, Год — ФЗ

Потенциальные ключи: {ISBN}, возможно также {Название, Автор}, если они уникальны

Выбор: ISBN — стабильный, короткий, уникальный — **первичный ключ**

ISBN	Название	Автор	Год
978-5-17...	"БД для всех"	Иванов И.И.	2022

Заключение

Связи между таблицами обеспечивают логическую организацию взаимосвязанных данных. Внешние ключи поддерживают ссылочную целостность, а связующие таблицы позволяют реализовывать отношения многие ко многим.

Вопросы для самопроверки

1. Какие типы связей существуют между таблицами?
2. Как реализуется связь один к одному?
3. В какой таблице размещается внешний ключ при связи один ко многим?
4. Для чего создаётся связующая таблица?
5. Чем отличаются действия CASCADE, SET NULL и RESTRICT?

Список использованных источников

1. Elmasri R., Navathe S. B. *Fundamentals of Database Systems*. 7th ed. Pearson, 2016.
2. Silberschatz A., Korth H. F., Sudarshan S. *Database System Concepts*. 7th ed. McGraw-Hill, 2020.
3. Connolly T., Begg C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Pearson, 2015.
4. Garcia-Molina H., Ullman J. D., Widom J. *Database Systems: The Complete Book*. 2nd ed. Pearson, 2009.
5. Beaulieu A. *Learning SQL*. 3rd ed. O'Reilly Media, 2020.
6. PostgreSQL Global Development Group. *PostgreSQL Documentation: Constraints*. https://www.postgresql.org/docs/current/ddl-constraints.html
7. PostgreSQL Global Development Group. *PostgreSQL Documentation: Table Expressions*. https://www.postgresql.org/docs/current/queries-table-expressions.html
8. Oracle Corporation. *MySQL 8.4 Reference Manual: JOIN Clause*. https://dev.mysql.com/doc/refman/8.4/en/join.html

СПАСИБО ЗА ВНИМАНИЕ