

Домен — множество значений, допустимых в определенном контексте. Смысл домена: если значения берутся из одного и того же домена, то они относятся к одному типу — эти значения можно сопоставить (сравнить).

```
CREATE DOMAIN имя_домена AS базовый_тип_данных
DEFAULT значение_по_умолчанию
CHECK (условие);
CREATE DOMAIN age_domain AS INTEGER
CHECK (VALUE >= 0 AND VALUE <= 120);
```

Степень отношения — это число его атрибутов (отношение степени один - унарное, степени два — бинарное, степени n — n -арное).

Кардинальное число (мощность отношения) — это число его кортежей.

Реляционная алгебра — язык для определения новых отношений на основе существующих. В реляционной алгебре определён ряд операций над отношениями. Результат операции — новое отношение.

Операция выборки — в результате операции формируется отношение на основе R , которое содержит только те строки (кортежи), которые удовлетворяют заданному предикату.

Проекция — в результате операции формируется новое отношение, содержащее только те атрибуты из R , которые были указаны в проекции.

Нормализация - формальный метод для проверки / доработки модели на основе ключей и функциональных зависимостей в отношениях.

Если в отношении атрибут A_2 функционально зависит от атрибута A_1 , то каждое значение A_1 связано с одним значением A_2 и определяет его. A_1 — детерминант функциональной зависимости. Функциональная зависимость определяется смысловыми связями, на основе которых строится отношение. Текущие данные в отношении не влияют на функциональные зависимости. Зависимость называется тривиальной, если среди предикатов зависимости есть атрибут, который есть среди зависимостей.

Аномалии:

- 1) Избыточность Данных: информация о кураторе (GrMentor) повторяется для каждого студента из одной и той же группы (Egor Kirov для группы P3100). Это ведет к неэффективному использованию памяти.
- 2) Аномалии Обновления (Update Anomalies): если куратор группы P3100 сменится, нам придется обновить поле GrMentor во всех строках, где Group = 'P3100'. Если мы обновим не все строки, данные станут противоречивыми (у одной группы окажется несколько кураторов).
- 3) Аномалия вставки: мы не можем добавить информацию о новой группе и ее кураторе, пока в этой группе нет хотя бы одного студента.
- 4) Аномалии Удаления (Deletion Anomalies): если мы удалим последнего студента из какой-либо группы (например, Василия Иванова из P3101), мы потеряем информацию о самой группе P3101 и ее кураторе Романе Ивове, даже если эта информация нам еще нужна.

Аксиомы Армстронга:

- 1) Рефлексивность: если A_2 - подмножество A_1 , то $A_1 \rightarrow A_2$
- 2) Дополнение: если $A_1 \rightarrow A_2$, то $A_1, A_3 \rightarrow A_2, A_3$.
- 3) Транзитивность

Тета-соединение (Theta Join): $R \bowtie_{\Theta} S$. Общий случай, где Θ - любое условие сравнения между атрибутами R и S . Логически эквивалентно: $\sigma_{\Theta}(R \times S)$ (выборка из декартова произведения).

Эквисоединение (Equijoin): Частный случай тета-соединения, где условие Θ содержит только операции равенства (=).

Естественное соединение (Natural Join): Эквисоединение по всем атрибутам с одинаковыми именами, причем совпадающие столбцы включаются в результат только один раз.

1НФ: отношение, на пересечении каждой строки и столбца — одно значение.

Вариант 1: сделать из групп значений отдельные строки.

Вариант 2: декомпозиция, то есть разделить таблицу на несколько, вынеся повторяющиеся группы в отдельную таблицу со связью через внешний ключ.

2НФ: если отношения в 1НФ и атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношения.

Полная функциональная зависимость: A_2 в полной функциональной зависимости от A_1 , если $A_1 \rightarrow A_2$, но нет зависимостей вида $A_3 \rightarrow A_2$, где A_3 — подмножество A_1 .

Частичная ФЗ: зависимость $A \rightarrow B$, где A — составной детерминант, и B зависит от части A .

Чтобы привести к 2НФ — убрать частичные зависимости от ключа:

Вынести частично зависимые атрибуты и ту часть ключа, от которой они зависят, в отдельную таблицу.

3НФ: Нет транзитивных зависимостей неключевых атрибутов от ключа через другие неключевые атрибуты.

Как достичь: вынести транзитивно зависимые атрибуты и их непосредственный детерминант в отдельную таблицу. Транзитивная функциональная зависимость — если для A_1, A_2, A_3 из R : $A_1 \rightarrow A_2$ и $A_2 \rightarrow A_3$, то A_3 транзитивно зависит от A_1 через A_2 .

Иногда, после приведения базы данных к высокой нормальной форме (например, ЗНФ или НФБК), оказывается, что для выполнения частых запросов требуется слишком много операций соединения (JOIN) между таблицами. Это может снижать производительность. В таких случаях иногда прибегают к денормализации — процессу осознанного нарушения некоторых правил нормализации для повышения производительности запросов. Прием: Объединение нескольких таблиц в одну, добавление избыточных данных.

Плюсы:

Уменьшение количества соединений в запросах.

Потенциальное ускорение выполнения частых запросов на чтение.

Минусы:

Увеличение избыточности данных (занимает больше места).

Повышенный риск аномалий (вставки, обновления, удаления).

SQL-функции: Тело функции состоит из одного или нескольких SQL-запросов. Выполняются быстро, но возможности ограничены самим SQL.

PL/pgSQL-функции: Тело функции написано на языке PL/pgSQL, позволяет использовать переменные, циклы, условия и т.д. Самый распространенный вариант для сложной логики.

Функции на других языках (C, Python, Perl, Tcl и др.): Требуют установки соответствующих расширений (CREATE EXTENSION plpython3u;). Позволяют использовать возможности и библиотеки этих языков внутри БД.

CREATE [OR REPLACE] FUNCTION имя_функции ([имя_arg1] тип_arg1, [имя_arg2] тип_arg2, ...)

RETURNS тип_возвращаемого_значения – Или RETURNS TABLE(...) для возврата таблицы, или VOID если ничего не возвращает

AS \$\$ – Или AS '...' - тело функции в \$\$ или одинарных кавычках

– Тело функции (SQL или PL/pgSQL код)

\$\$ LANGUAGE язык; – язык: sql, plpgsql, plpython3u и т.д.