

Просто набор приколов, тем и набросков для запоминания ответов

Основные понятия и определения

- сущность
- связь
- атрибуты
- домен
- ключ (потенциальный, искусственный, натуральный, внешний)
- БАЗА ДАННЫХ
- СУБД
- МОДЕЛЬ ДАННЫХ

Свойства

- Целостность
- Согласованность
- Восстанавливаемость
- Безопасность
- Эффективность

Характеристики связей

- Соединяете 2 сущности (или рекурсия)
- Двухнаправленная, 2 подписи
- 3 свойства: Модальность (- должен, - - может), Мощность, Трансферабельность

Дуга = xor

Супертипы и подтипы

- Аналогично классу и наследникам
- Подтип сохраняет сущность супертипа
- Общие поля у подтипов = полям супертипа, + добавляются свои
- Связь 1 к 1 + дуга

Контроль целостности

- Есть основная таблица = шмоти, и подчиненная = заказ (PRIMARY KEY = NOT NULL)

- Может быть шмотка, которую не заказали
- Не может быть заказа без шмотки
- Шмотки не повторяются

Аномалии

- Много одинаковых по смыслу данных
- Вставки - анал (если нет данных, вставляешь нихуя)
- Обновления - забыли данные
- Удаления - хватанули лишнего

Нормализация = разбиение на простые

- 1я НФ - нет повторов-столбцов, нет хуйни через запятую (неатомарных), + PRIMARY KEY
- 2я НФ - не РК столбцы зависят от всего РК, а не от части (если ПК 2 столбца, то сразу от двух)
 - 1я + РК из одного столбца
- 3я - все зависит только от РК
 - 2я без транзитивных зависимостей

Операции

- DML (Data Manipulation) - работа с данными - INSERT, SELECT, DELETE, UPDATE
- DDL (Data Definition) - создание таблиц - CREATE, ALTER, DROP
- DCL (Configuration) - пользователи - GRANT, REVOKE
- TCL (Transaction) - транзакции - COMMIT, BEGIN, ROLLBACK

Индексы = деревья

- Кластерный - хранит один столбец, быстрее, нет доп места
- Некластерный - хранит ссылки на несколько столбцов, медленнее, + доп место

SELECT

- SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY
- Порядок - FROM, WHERE, SELECT, {GROUP BY}, ORDER BY
- NULL - самый большой (при сортировке)
- ASC - возрастание, DESC - убывание

Подзапросы

- Вложенный - просто SELECT в SELECT
- Коррелирующий - второй селект использует первый селект

JOIN

- CROSS = декартово произведение (все со всем)
- INNER = CROSS + проверка условия
- LEFT = все из левой + по условию (INNER) из правой
- RIGHT = LEFT наоборот
- FULL = INNER + LEFT + RIGHT (то, что не в INNER)

Методы выполнения соединений

- NESTED (вложенные) - ходит вложенным циклом по всем строкам и сверяет условие + return if true
- HASH - строит хеш-таблицу по PK и проверяет по ней
- MERGE - если таблицы отсортированы по PK, делает алгоритм слияния таблиц

CTE - временная таблица внутри запроса (with *tablename* as (select))

```
WITH sales_data AS (  
    SELECT product_id, SUM(quantity) AS total_sales  
    FROM sales  
    WHERE date >= '2023-01-01' AND date <= '2023-06-01'  
    GROUP BY product_id  
)  
SELECT product_id, total_sales  
FROM sales_data  
WHERE total_sales > 1000;
```

- для читаемости, упрощения, повышения производительности

Оконная функция

```
sql Copy code  
  
SELECT employee_id, last_name, salary,  
       AVG(salary) OVER (PARTITION BY department_id) AS avg_department_salary  
FROM employees;
```

- Это функция, которая работает с окном (партицией) = набор строк
- Партиция - набор строк по одному/нескольким столбцам

- Порядок выполнения - FROM, JOIN, WHERE, GROUP BY, HAVING, SELECT, WINDOW FUNCTION, ORDER BY
- Типы
 - Агрегирующая - SUM(), AVG(), MAX()
 - Ранжирующая - RANK(), ROW_NUMBER() - вычисляют ранг для строк внутри окна по критерию - выполняются последней (после запроса, сортировки и тд)
 - Value - FIRST_VALUE(), LAST_VALUE(), LEAD()

Рамки в агрегирующей оконной функции

- PRECEDING - назад на опр. число (UNBOUNDED - до начала)
- FOLLOWING - вперед на опр. число (UNBOUNDED - до конца)
- CURRENT ROW - нынешняя строка
- BETWEEN ... AND ...
- По умолчанию - от начала до текущей строки
- ROWS BETWEEN ... - рамка по строкам
- RANGE BETWEEN .. - рамка по значениям строк

VIEW (Представления)

- CREATE VIEW *name* AS SELECT *что* FROM *откуда* + *условия*
- виртуальная таблица, физически не существует, хранится

Типы данных

- Скалярные и составные
- Составные - TABLE, **RECORD**, VARRAY
- RECORD = КЛАСС (TYPE *name* IS RECORD (*поля*) - явное определение)
- Неявное определение - %ROWTYPE по уже созданному

Процедуры/Функции

- Блоки PL/SQL, с именем, хранятся в БД и вызываются
- CREATE FUNCTION *имя* [*параметры*] RETURN *тип* IS/AS [*локальные переменные*] BEGIN
- IN - передается и не меняется (значение/переменная), OUT - выходной параметр (переменная = NULL), IN OUT - туда и обратно (переменная)

Курсоры

- Именованная область куда идет вывод SELECT
- Явные - делает прогер (CURSOR *имя* IS SELECT ...), Неявные - СУБД (SELECT INTO)

- Управление - явное (явное открыть, выбрать (FETCH), ...) и неявное - делает СУБД
- **Аттрибуты** - %ISOPEN, %NOTFOUND, %FOUND, %ROWCOUNT
- FOR строка IN курсор LOOP END LOOP
- В неявный курсор не более одной строки

Исключения

- блок EXCEPTION WHEN *exception* [OR *exception*] THEN
- OTHERS для неопределенной ошибки
- Если нет блока исключений в этом блоке, то он идет в блок исключений выше
- После перехода в блок выше, обратно не возвращается
- Ошибки системы, пользователя, предупреждения
- Бывают системные и определенные прогером (DECLARE, определяются, EXCEPTION)
- Выкинуть исключение - RAISE (без сообщения), RAISE_APPLICATION_ERROR(код ошибки (-20000 до -20999), сообщение)

Пакеты = библиотека

- Набор хуйни (процедур, функций, тд)
- Спецификация - то, с чем работают, можно менять не меняя тело
- Тело - реализация, нельзя менять не меняя спецификацию

Триггеры

- Хранится и запускается автоматом при системном событии или при операциях
- Если FOR EACH ROW - строковый, если нет - операторный
- INSTEAD OF - для представления
- Предикаты - если указано несколько операторов, то для определения какой оператор используется INSERTING, UPDATING, DELETING
- REFERENCING - переименование NEW и OLD
- WHEN - добавление условия на запуск триггера (только строковый)
- DDL-триггеры
 - ON DATABASE
 - ON SCHEMA
- Триггер событий БД например запуск, логин и тд
- Включение/отключение ALTER TRIGGER *name* ENABLE/DISABLE
- ALTER TABLE *name* DISABLE ALL TRIGGERS
- DROP TRIGGER
- **Мутирование таблиц** - при изменении/чтении таблицы, где должен сработать триггер

Интеграция данных

- объединение данных из нескольких источников
- Хранилище данных - для выполнения запросов и сохранения структурированных данных
- Витрина данных - вывод результатов из хранилища
- Озеро данных - структ/неструкт данные (свалка блять)
- Поставщик данных - кто дает данные
- **Методы интеграции**
 - Консолидация - хранение у себя в хранилище
 - Федерализация - данные по запросам
 - Распространение - трансфер скопированных данных

Схема на запись и чтение

- На запись - сначала делается архитектура и вся хуйня
- На чтение - БД создается после копирования данных при чтении данных

OLAP и OLTP

- OLTP - транзакции в реальном времени
- OLAP - сначала накопление + аналитика

Строковые и Колончатые СУБД

- Строковые для OLTP
- Колончатые для OLAP

Модели данных

- Звезда - центр + по одной
- Снежинка - Звезда + связи у связанных

Подходы

- Кимбалла - совокупность витрин данных, потом общее
- Инман - все вместе, потом разделение на витрины

Транзакции

- Набор операций, по принципу все или ничего
- Транзакцию надо
 - Открыть
 - Выполнить

- Закрывать - COMMIT, ROLLBACK
- **ACID** - атомарность (все или ничего), согласованность (не нарушает согласованность), изолированность (извне не трогают) - блокировки, надежность (что после завершения не удалится по ошибке)
- Блокировки - разделяемая, монополярная, блокировка обновления
- **Ошибки параллельной работы**
 - При одновременном выполнении транзакций, обращающихся к одним данным
 - Пример: таблица баланса, одна транзакция списывает 200р, а вторая получает текущий баланс, в итоге баланс получает до списания 200р => неверные данные

ETL-процесс

- Этапы- извлечение, очистка, трансформация, загрузка
- данные могут быть: структурированными (сразу можно загрузить в БД), полуструктурированными (нельзя сразу в БД, но есть какая-то организация - JSON, csv, ...), неструктурированными (вообще хаотичная хуйня - word, pdf, txt)
- Очистка данных - базовое форматирование к общему виду + дедупликация + пересмотр формата данных (единицы измерения)
- **Bad data** - то, что не подходит к общему процессу обработки, подозрительные => нахуй не нужны
- Загрузка данных - полная (сразу все), инкрементная (поток и пакетная) - загружает разницу (то, что добавилось)
- ETL позволяет - сохранять версии, консолидация данных (данные в одном месте), анализ данных, автоматизация (видимо, обработки данных)
- Инструменты визуализации - Tableau, PowerBI