**Теория БД**

**Транзакция –** Логическая единица работы, которая состоит из одной или нескольких SQL-операций сгруппированных между собой. Обеспечивают свойства ACID.

**ACID –** свойство транзакций, направленные на обеспечение надежности и целостности данных.

*Atomicity* (Атомарность) – Транзакция не может быть зафиксирована частично, то есть не может иметь промежуточных состояний, она либо выполнена полностью, либо не выполнена вовсе.

*Consistency* (Согласованность) - свойство, гарантирующее, что каждая успешная транзакция зафиксирует только допустимые результаты. То есть, это гарантия того, что при успешной транзакции будут выполнены все правила, ограничения, которые предъявляет система к конкретным данным, иначе транзакция не будет выполнена и данные в системе вернутся к прежнему состоянию.

*Isolation* (Изолированность) – Параллельные транзакции (транзакции, работающие с одними и теми же данными) не должны влиять на результаты выполнения друг друга.

Durability (Надежность) – Если клиент получил подтверждение от системы подтверждение завершения транзакции, то эти изменения не будут отменены из-за сбоя.

***Уровни изоляции транзакций:***

***Read uncommitted –*** Самый низкий уровень изоляции. Изменения видны транзакциям, даже если они еще не были зафиксированы.

Возможные эффекты: грязные чтения, неповторяющиеся чтения и фантомные чтения.

***Read committed*** – Транзакции видят только подтвержденные изменения. Вторая транзакция может увидеть изменения первой во время своей транзакции, если использует те же данные. По умолчанию в PostgreSQL.

Возможные эффекты: неповторяющиеся чтения и фантомные чтения.

***Repeatable read*** – Транзакция не увидит Update и Delete других транзакций до завершения своей транзакции, но Insert возможны. По умолчанию в MySQL.

Возможные эффекты: фантомные чтения.

***Serializable*** – Максимальный уровень изоляции. Блокирует чтение. Не содержит эффектов, но ухудшает параллелизм и производительность.

Уровень изоляции устанавливается с помощью

SET<видимость> TRANSACTION ISOLATION LEVEL <Уровень> - без параметра (для 1 следующей транзакции), session (Для текущей сессии), global (на все транзакции).

[*Побочные эффекты, возникающие при параллельной работе транзакций:*](https://habr.com/ru/companies/infopulse/articles/261097/)

Потерянное обновление – Работает как race condition (при вычитывании 2мя транзакциями значения, будет видимо то изменение, которое будет наложено последним).

Грязное чтение – Возможность транзакции считать, обновить или добавить данные, изменения которых впоследствии не подтвердится.

Неповторяющееся чтение – При повторном чтении в рамках одной транзакции данные изменяются.

Фантомное чтение – При повторном чтении изменяется кол-во строк в результирующей выборке за счет изменения другой транзакцией критериев выборки или добавление/удаление строк выборки.

**Физическая организация БД**

**\***

**\***

**\***

**\***

**\***

**\***

**\***

**\***

**Версии строк и снимки**

**\***

**\***

**\***

**\***

**\***

**Очистка и vacuum**

**\***

**\***

**\***

**\***

**\***

**\***

**Буферный кэш и журналирование**

\*

\*

\*

\*

\*

**Блокировки**

**\***

**\***

**\***

**\***

**\***

**\***

**Обработка и исполнение запроса**

*Разбор (Parsing) -* Синтаксический разбор - СУБД анализирует SQL-запрос, чтобы удостовериться, что он соответствует правилам языка SQL. Семантический разбор - проверка существования таблиц и полей, а также правильность их использования. Для проверки используется системный каталог.

*Трансформация –* замена представлений на таблицы, преобразование SEARCH и CYCLE для рекурсивных запросов.

*Планирование (Query Planning) -* Планировщик запросов создает дерево выполнения запроса и выбирает вариант исполнения с наименьшей стоимостью, используя статистику из системного каталога.

*Выполнение* - СУБД выполняет запрос в соответствии с выбранным планом выполнения.

Данные этапы выполняются для каждого запроса.

Prepare - Подготовленные запросы в SQL позволяют СУБД сохранять оптимизированный план выполнения запроса и повторно использовать его при последующих выполнениях запроса с разными параметрами. Это позволяет уменьшить накладные расходы на планирование и оптимизацию запроса каждый раз при его выполнении. (отсекается часть работы – не выполняется повторно).

**База данных и схема**

База данных представляет набор связанных данных (в виде отношений), хранящиеся в структурированном виде (в виде таблиц). Помимо таблиц обычно содержит другие объекты – индексы, процедуры, представления. То есть это хранилище для всех объектов.

Схема данных логически сгруппированные объекты БД, т.е. схема обеспечивает не организацию данных в каждой отдельной таблице, а объединяет сущности БД в отдельную логическую группу – схему. Так же позволяет обеспечивать контроль доступа к объектам и безопасность.

Т.е. БД — это хранилище данных для всех объектов, а схема — это набор данных внутри БД, сгруппированных логически.

Схема – для разграничения пространства имен и ограничения прав доступа.

Проектирование БД – [1](https://www.youtube.com/watch?v=HnRXzrg3Sd4&list=PLf30vI0hEi1v435cBmZSHkr1QAJdOk9mb&index=5&ab_channel=RclassTech), [2](https://www.youtube.com/watch?v=tNQlFW3BRUc&list=PLLACapFZr4XvyQgxZaIOK-MJwblGVCL41&index=7&ab_channel=ListenIT)