**Реляционная база данных** - совокупность отношений (таблиц), содержащих всю информацию, которая должна храниться в БД. Основана на реляционной модели данных.

Реляционная модель данных включает следующие компоненты:

* Структурный аспект - данные в базе данных представляют собой набор отношений (таблиц).
* Аспект целостности — отношения (таблицы) отвечают определенным условиям целостности.

1. *Структурная целостность –* Отсутствие дубликатов кортежей, наличие первичного ключа, отсутствие упорядоченности кортежей.
2. *Ссылочная целостность –* Выполняется одно из двух правил:

* При удалении кортежа из родительского отношения, удаляются все связанные кортежи подчиненного отношения.
* При удалении кортежа из родительского отношения, на месте ключа в связанных отношениях ставится NULL.

1. *Семантическая целостность –* обеспечивается разработчиком с помощью средств СУБД.

Constraints: Тип данных, диапазон, уникальность.

NOT NULL, UNIQUE, PRIMARY KEY, CHECK, FOREYGN KEY.

* Аспект обработки (манипулирования) — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление).

определение множества операций над данными.

**Нормальная форма –** требование, предъявляемое к структуре таблиц, для устранения из базы избыточных функциональных зависимостей.

Нормализация – процесс преобразования отношений базы данных к виду, отвечающему нормальным формам (переход от одной нормальной формы к следующей). Позволяет привести БД к минимальной избыточности.

**1НФ –** все значения атомарны(отсутствуют массивы и списки).

**2НФ –** 1НФ + каждый неключевой атрибут функционально полно зависит от первичного ключа.

**3НФ –** 2НФ + отсутствие транзитивных зависимостей, когда один неключевой аттрибут зависит от другого неключевого.

**НФ Бойса-Кодда –** 3НФ + ключевые аттрибуты не зависят от неключевых.

**Денормализация –** намеренное преведение структуры БД в состояние, не соответствующее критериям нормализации, обычно с целью ускорения операций чтения.

**Потенциальный ключ –** подиножество аттрибутов отношения, которое уникально идентифицирует каждую запись в таблице. Удовлетворяет след. Требованиям – уникальность, минимальность (если из ключа убрать аттрибут, он утратит уникальность).

Из множетсва всех потенциальных ключей выбирается **первичный ключ** в качестве основного (используемого по умолчанию). Остальные ключи называют альтернативными.

Первичный ключ накладывает на поле ограничения NOT NULL, и UNIQUE. Также для каждого первичного ключа в таблице создается индекс.

**Индекс –** объект базы данных создаваемый с целью улучшения поиска записей в бд.

Индекс содержит ключи, построенные из одного или нескольких столбцов в таблице или представлении. Для индексов на диске эти ключи хранятся в виде структуры сбалансированного дерева, которая поддерживает быстрый поиск строк по значениям ключей.

Существует еще два способа хранения индекса:

* Hash – используется в основном для точного соответствия ( = ) и не работает со сравнениями (≥, ≤)
* B-tree – это наиболее распространенный способ хранения данных, обеспечивающий эффективную работу с дисковой памятью.

**Кластерные индексы -** При кластеризованном индексе строки физически хранятся на диске в том же порядке, что и индекс.Таблица может иметь только 1 кластерный индекс.

**Некластерные индексы** – они не перестраивают физическую структуру таблицы, а лишь организуют ссылки на соответствующие строки.

**Естественный ключ –** первичный ключ, состоящих из информационных полей таблицы.

**Суррогатный ключ –** дополнительное поле, добавленное в таблицу, чтобы служить первичным ключом. Генерируется исскуственно.(Гарантированная уникальность).

**Внешний ключ –** определяет связь поля с другой таблицей, ссылается на первичный ключ другой таблицы.

**Транзакция –** последовательность действий, выполняющаяся как единое целое, и переводящая базу данных из одного согласованного состояния в другое.

**ACID** описывает требования к транзакционной системе:

**Atomicity (Атомарность) –** транзакция либо выполняется полностью, либо не выполняется совсем.

**Consistency (Согласованность) –** данные до транзакции должны быть согласованы с данными после выполнения транзакции.

**Isolation (Изолированность) –** параллельно выполняющиеся транзакции не должны оказывать влияния друг на друга.

**Durability (долговечность) –** данные должны быть сохранены независимо от проблем на нижних уровнях.

Уровни изоляции транзакции

**Чтение неподтверждённых данных (read uncommitted)**

Самый низкий уровень изоляции. Можно читать незафиксированные изменения других транзакций, но запись идет строго последовательно.

Исключает потерянное обновление – проблема, когда две транзакции одновременно пишут значение в одну ячейку, при этом одно из значений теряется.

**Чтение подтверждённых данных (read committed)**

Транзакция видит только зафиксированные изменения других транзакций.

Исключает грязное чтение – когда читаются данные, изменяемые другой транзакцией, но затем транзакция откатывается, и изменения исчезают.

**Повторяемое чтение (repeatable read)**

Транзакция не видит измененные или удаленные записи другой транзакции. Она блокирует изменение данных другой транзакцией. При многократной выборке транзакция получает один и тот же результат.

Исключает неповторяемое чтение – когда несколько раз читаются данные, в данный момент изменяемые другой транзакцией. В итоге каждый раз результат может быть разным.

**Сериализуемый (serializable)**

Транзакции полностью изолируются друг от друга, фактически выполняясь последовательно.

Исключает проблему фантомов – когда однна транзакция выбирает несколько строк по каким-то критериям. Другая между выборками вставляет либо удаляет сстроки или изменяет столбцы некоторых строк, используемых в критериях выборки.

В результате получается что одни и те же выборки первой транзакции дают разные множества строк.

Уровни изоляции реализуюся с помощью блокировок. Блокировкой называется временное ограничение на выполнение некоторых операций обработки данных. Происходит блокирование доступа к объекту.

Есть 2 вида блокировок:

* Разделяемая блокировка (s-блокировка  Shared lock). Резервирует ресурс только для чтения. Несколько транзакций могут одновременно накладывать эту блокировку. Другие транзакции в этот момент не могут изменять данный ресурс.
* Монопольная блокировка (x-блокировка exclusive lock). Резервирует ресурс на изменение. Может быть установлена на ресурс только при отсутствии других блокировок. На ресурс может быть установлена только одна монопольная блокировка.

HAVING применяется для фильтрации строк после группировки, фильтрует группы по значениям результата агрегатных функций.

Агрегатной функцией в языке SQL называется функция, возвращающая какое-либо одну строку по набору из входных строк.