

## 1. Понятие физической независимости данных

**Физическая независимость** данных: **возможность изменения** способа хранения, расположения или переноса данных **без влияния** на логическую структуру БД и работу приложений, работающих с БД.

## 2. Содержимое журнала транзакций

**Журнал транзакций** хранит информацию обо всех поступающих в СУБД транзакциях и их выполнении. Подразумевается сохранение **информации о времени поступления**, времени **начала** и **завершения** выполнения транзакции.

## 3. Определение домена, семантическая нагрузка понятия домена

**Домен** – допустимое множество значений данного типа. **Семантическая нагрузка** подразумевает, что данные считаются **сравнимыми** тогда и только тогда, когда они **принадлежат одному** домену.

## 4. Требования структурной, манипуляционной частей реляционного представления

**Структурная часть**: все данные должны быть представлены в виде **отношений**. **Манипуляционная часть**: язык запросов к реляционной СУБД должен обладать **не меньшей мощностью**, чем язык реляционной алгебры или реляционного исчисления.

## 5. Операции реляционной алгебры

**Объединение**, **пересечение**, **разность**, **декартово произведение** ( $N \times M$ ), **проекция** (отображение лишь некоторых атрибутов), **ограничение** (WHERE), **соединение** (JOIN), **деление** ( $A[a,b,c] \div B[b,c] = C[a]$ , где  $a$  – только такие, для которых  $b$  и  $c$  одинаковы для  $A$  и  $B$ ).

## 6. Понятие характеристической сущности. Пример

**Характеристическая сущность** – **сущность**, которая **формализует и уточняет связь** типа «одна к одной» или «много к одной». Например, компьютер как стержневая сущность может иметь несколько характеристических сущностей – модулей памяти.

## 7. Условия нахождения отношений во второй нормальной форме

Отношение находится в **2NF**, когда оно находится в **1NF** и каждый **не ключевой атрибут функционально полно зависит** от первичного ключа.

## 8. Теорема Хеза

**Декомпозиция** функционально зависимых **отношений** произойдёт **без потери** информации.

## 9. Доступ с плотным индексом

Плотный индексный файл – это файл с **упорядоченными индексами**, соответствующими **неупорядоченным записям** БД в другом файле. Доступ к **записям** осуществляется **через** файл с **индексами**.

## 10. Понятие табличного пространства, экстенда

**Табличное пространство** – это логическое пространство, в котором хранятся индексы, таблицы и т.д. **Экстендой** называют 8 физически последовательных структурированных страниц файла БД.

## 11. Свойства классических транзакций

**Атомарность** (неделимость), **изолированность** (независимость от других транзакций), **согласованность** (переход БД из одного согласованного состояния в другое), **долговечность** (изменения сохраняются).

## 12. Идея метода временных меток

Метод **временных меток** – это метод **параллельного выполнения транзакций**, при котором, если транзакция **t1** поступила **раньше** транзакции **t2**, то система составляет такой **серийный план** выполнения транзакций, как если бы **t1** была целиком **выполнена до t2**.

## 13. Тест FASMI

**Принцип OLAP-систем**, подразумевающий следующие их свойства: **Fast** (результат не более 5) **Analysis** (логический и статистический анализ) of **Shared** (многопользовательский доступ с защитой) **Multidimensional** (многомерное представление данных) **Information** (доступ к нужной информации).

## 14. Проблема очистки данных на уровне ячейки

**Орфографические** ошибки, **отсутствие** данных, **фиктивные** значения, **логически** неверные значения, **закодированные** значения, **составные** значения.

## 15. Многомерная модель хранилищ данных (MOLAP)

**MOLAP** – модель данных, подразумевающая **физическую реализацию многомерного представления** данных. В системах с такой моделью данные организованы **не в виде реляционных таблиц, а в виде многомерных массивов: гиперкубов** (все ячейки БД одинаковой размерности) или **поликубов** (собственный набор измерений для каждой переменной).

## 16. Сложные типы данных ОРМД – коллекции. Привести примеры

**Коллекция** – это сложный **тип из отдельных элементов**, каждый из которых принадлежит **к одному и тому же типу данных**. Элементы могут быть **сложными, встроенными** или **пользовательскими**. Разновидности коллекций: **множество (SET, множество уникальных неупорядоченных элементов без дублирования)**, **мультимножество (MULTISET – SET с дублированиями)**, **списки (LIST – упорядоченный MULTISET)**.

## 17. Принципы построения защищенных систем баз данных (перечислить)

**Экономическая оправданность** механизма защиты, **открытое проектирование**, **распределение полномочий**, **минимально возможные привилегии**, **управляемость**, психологическая **приемлемость** работы средств защиты.

## 18. Дискреционная модель безопасности

**Дискреционная модель безопасности** – модель безопасности с **разграничением доступа** между **субъектами** и **объектами**, где **объект** защиты – **таблица**, представление, хранимая процедура, а **субъект** защиты – **пользователь** или хранимая процедура. Управление доступом может быть либо **добровольным** (привилегии доступа к объектам), либо **принудительным** (системные привилегии).

### 1. Понятие логической независимости данных

**Логическая независимость** данных: общая **структура** данных может быть **изменена без изменения** прикладных **программ**.

### 2. Протокол WAL

**WAL (Write Ahead Log – записывай сначала в журнал)**: **Запись** об изменении объекта базы данных должна оказаться **во внешней памяти журнала раньше, чем измененный объект** окажется во внешней **памяти** базы данных. **Если во внешней памяти** базы данных **находится** некоторый **объект** базы данных, по отношению к которому выполнена операция **модификации**, то **во внешней памяти журнала обязательно** находится **запись**, но **не обязательно наоборот**.

### 3. Фундаментальные свойства отношений

**Отсутствие дубликатов кортежей** (отношение по определению есть множество кортежей, а множество по определению не может содержать дубликатов), **отсутствие упорядоченности атрибутов и кортежей** (по определению отношения и отношения-экземпляры являются множествами, а множества по определению не имеют порядка), **атомарность значений атрибутов** (по определению домен – потенциальное множество значений простого типа).

### 4. Требования целостной части реляционной модели данных

**Целостность сущностей**: Любое **отношение должно обладать первичным ключом**.

**Целостность по ссылкам**: **отношение с определенным внешним ключом должно ссылаться на отношение**, где такой же атрибут является **первичным ключом, либо быть неопределенным**.

### 5. В чем выражается эквивалентность языка РА и РИ

**Мощности языков РИ и АР сопоставимы**: **любое выражение** на языке РИ **можно преобразовать** в его некоторый эквивалент РА, и наоборот.

### 6. Понятие обозначающей сущности. Пример

**Обозначающая сущность** – это **независимая от обозначаемой сущности сущность, дополняющая и уточняющая другую сущность**, со связью вида "многие-к-одной" или "одна-к-одной". Например, собака без человека собака не перестанет быть собакой (жена без мужа будет просто женщиной).

### 7. Условие нахождения отношений в 3й и 4й нормальной форме

Отношение находится в **3NF**, когда оно находится в **2NF** и каждый **неключевой атрибут нетранзитивно зависит** от **первичного** ключа.

Отношение находится в **4NF**, когда оно находится в **3NF/BCNF** и не содержит нетривиальных **MVD**.

## 8. Теорема Риссонена

Проекции **r1** и **r2** отношения **r** являются **независимыми**, если каждая **FD** в отношении **r** логически **следует из FD в r1 и r2** и **общие атрибуты r1 и r2 образуют возможный ключ хотя бы для одного из этих отношений**.

## 9. Доступ с разреженным индексом

Первый файл – с данными в отсортированном по значениям ключей порядке. Второй файл – **разреженный индекс**, который **состоит из пар (k, p)**, где **k-значение ключа**, а **p-файловый указатель блока**, в котором значение ключа первой записи равняется **k**. **РИ отсортирован** по ключу.

## 10. Фрагментация как способ повышения эффективности работы с данными

**Фрагментация баз данных – распределение фрагментов** одной базы данных **по нескольким другим** базам данных (возможно, под разными СУБД), находящимся на различных узлах. Подобное распределение **позволяет обращаться к нескольким фрагментам БД параллельно**.

## 11. Три проблемы, связанные с параллелизмом

Проблема **утраченного обновления** (В затёрла А). Проблема **зависимости от незафиксированного обновления** (А читает объект, куда прямо сейчас пишет В). Проблема **анализа на непротиворечивость**: проблема **повторного чтения** (А читает объект, В туда пишет, А читает объект снова, а он уже другой); проблема **появления фиктивных элементов** (из-за В появились новые незапланированные записи, удовлетворяющие А); проблема **анализа совместимости** (А анализирует объект, куда пишет В).

## 12. Идея метода выделения версий данных

Предоставление **транзакциям, читающим** данные, версии данных, имеющих на момент начала транзакции. **Блокировки не накладываются**. Каждой **транзакции даётся системный номер (SCN)**, тем **большой**, чем **позже** была запущена транзакция. Если **SCN страницы данных меньше SCN транзакции А**, то **транзакция А читает** эту страницу, **иначе** другая транзакция уже успела изменить данные – тогда **транзакция А ищет** в журнале **старый вариант** данных.

## 13. Операции OLAP-технологии

**Сечение** (подмножество гиперкуба с фиксированными измерениями). **Вращение** (изменение порядка представления). **Консолидация** (увеличение уровня абстракции ячейки). **Спуск** (обратна консолидации). **Разбиение с поворотом** (представление данных с разных точек зрения).

## 14. Проблемы очистки данных на уровне записи

**Противоречивость значений** в разных полях записи для одного объекта. Например, дата рождения – 2005 год, а возраст – 40 лет.

**Дублирующие записи**. Например, Дмитрий и Дима.

## 15. Сложные типы данных ОРМД – списки. Примеры

**LIST** – упорядоченное **множество** элементов, **допускающее** значения-**дубликаты**. **Пример** списка: зарплата сотрудника в течение квартала List{22000, 23000, 21000} как одно значение атрибута.

## 16. Реляционная модель хранилищ данных ROLAP

**ROLAP** – модель данных, подразумевающая **физическую реализацию реляционной БД с многомерным представлением** данных. В отличие от **MOLAP** таблица **фактов** в **ROLAP не содержит все возможные варианты комбинаций измерений**; она **содержит уникальный составной ключ**, который **объединяет первичные ключи таблиц измерений**.

## 17. Угрозы безопасности баз данных (перечислить)

**Внешние**: умышленные вредоносные **действия лиц**; **искажения в каналах** передачи информации; **сбои аппаратуры**; **вирусы** и иные вредоносные программные элементы;

**Внутренние**: **ошибки** на различных этапах **проектирования** БД; ошибки и несанкционированные **действия пользователей**; **недостаточная эффективность** используемых **методов** обеспечения ИБ.

## 18. Ролевая модель безопасности

**Роль идентифицирует** динамически образуемую **группу пользователей**, **каждый** из которых **обладает привилегией** на исполнение данной **роли** и всеми **привилегиями** данной роли для **доступа к объектам БД**.