

1. Понятие трёхуровневой архитектуры

Внешний уровень, на котором пользователи воспринимают данные, где отдельные группы пользователей имеют свое представление (ПП) на базу данных; f

Внутренний уровень, на котором СУБД и операционная система воспринимают данные; f

Концептуальный уровень представления данных, предназначенный для отображения внешнего уровня на внутренний уровень, а также для обеспечения необходимой их независимости друг от друга; он связан с обобщенным представлением пользователей.

2. Основные понятия сетевой модели данных: элемент данных, агрегат данных

Элемент данных – наименьшая поименованная единица данных (аналог поля в файловой системе).

Агрегат данных – поименованная совокупность данных в записи, можно рассматривать как единую целую.

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. Правила Кода

12 правил, которым должна удовлетворять реляционная база данных (по Э. Коду):

1. Правило информации.
2. Правило гарантированного доступа.
3. Системная поддержка Null-значений.
4. Динамический оперативный каталог на основе реляционной модели.
5. Правило исчерпывающего подязыка данных.
6. Правило обновления представления (view).
7. Ввод, обновление и удаление данных высокого уровня.
8. Физическая независимость данных.
9. Логическая независимость данных.
10. Независимость целостности.
11. Независимость распределения.
12. Правило соблюдения правил.

5. 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.

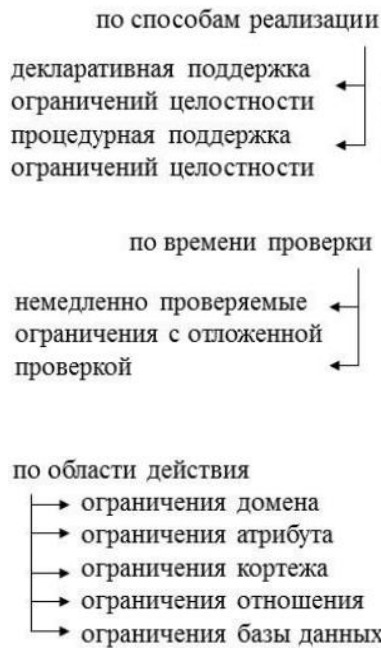
6. Функционально полная зависимость атрибута от составного ключа

Неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части ключа.

7. Теорема Фейджина

Пусть имеется переменная отношения R с атрибутами A, B, C (в общем случае, составными). Отношение R декомпозируется без потерь на проекции $\{A, B\}$ и $\{A, C\}$ тогда и только тогда, когда для него выполняется $MVD_{A \twoheadrightarrow B \mid C}$.

8. Классификации ограничений целостности



9. Правила определения набора требуемых индексов

Число строк > 200.

Столбцы, которые часто используются в WHERE, ORDER BY, GROUP BY, DISTINCT.

Столбцы, используемые для соединения таблиц или являющиеся внешними ключами.

Индексирование столбцов с высокой селективностью.

Не следует индексировать столбцы, используемые только в функциях.

Не следует индексировать часто изменяемые столбцы.

(Я хз чем отличается от вопроса «Правила индексирования»)

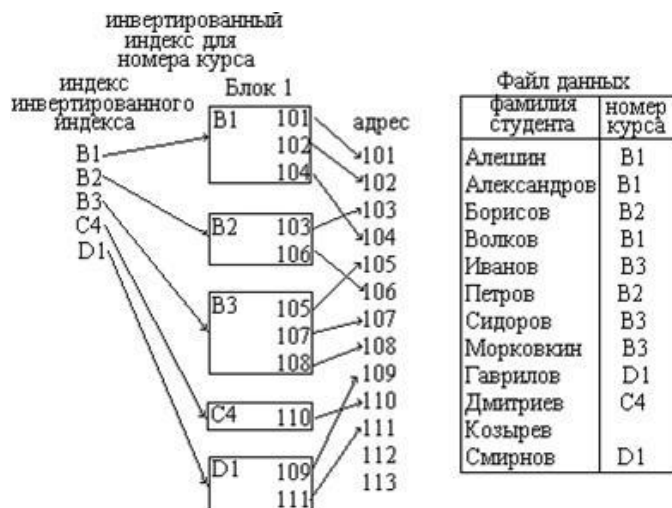
10. Плотный и разреженный индексный файл

Плотный индексный файл – это файл с упорядоченными индексами, соответствующими неупорядоченным записям БД в другом файле.

Разреженный индексный файл – индекс, который состоит из пар (k, p), где k-значение ключа, а p-файловый указатель блока, в котором значение ключа первой записи равняется k. Этот разреженный индекс также отсортирован по значениям ключей. Чтобы отыскать запись с заданным ключом k, надо сначала просмотреть индексный файл, отыскивая в нем индекс (k,p).

11. Доступ по средствам инвертированных файлов

В *инвертированном методе доступа* используется три файла – индекс вторичного ключа, инвертированный файл и основной файл базы данных. Индекс вторичного ключа содержит значения вторичных ключей, упорядоченных по возрастанию или убыванию и адрес блока в инвертированном файле. В **инвертированном файле** содержатся значения первичных ключей с данным значением вторичного ключа и адреса записей в основном файле, соответствующие значениям первичного ключа.



12. Кластеризация как способ повышения эффективности работы с данными

Кластеризация (сегментация) - задача группировки множества объектов на подмножества (кластеры) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию.

Разбиение на кластеры позволяет упростить работу с данными, так как теперь для каждого кластера возможно построить отдельную более репрезентативную модель, что позволит с большим успехом использовать методы машинного обучения для решения прикладных задач.

13. Уровни изолированности

READ UNCOMMITTED - уровень незавершенного считывания.

READ COMMITTED - уровень завершенного считывания.

REPEATABLE READ - уровень повторяемого считывания.

SERIALIZABLE - уровень способности к упорядочению.

Неаккуратное считывание ("Грязное" чтение, незафиксированная зависимость).

Неповторяемое считывание (Частный случай несовместного анализа).

Фантомы (Фиктивные элементы - частный случай несовместного анализа).

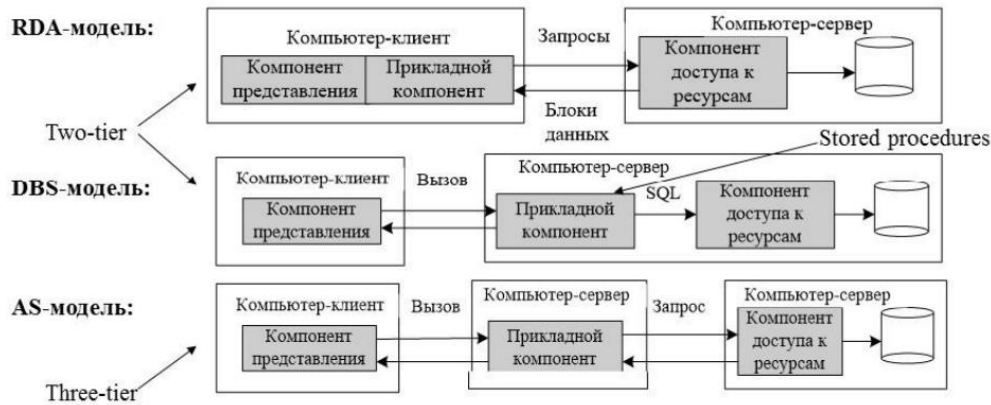
Уровень изоляции	Неаккуратное считывание	Неповторяемое считывание	Фантомы
READ UNCOMMITTED	Да	Да	Да
READ COMMITTED	Нет	Да	Да
REPEATABLE READ	Нет	Нет	Да
SERIALIZABLE	Нет	Нет	Нет

14. Три модели архитектуры «клиент-сервер»

Remote Data Access (RDA) – доступ к удалённым данным

Data Base Server (DBS) – сервер базы данных

Application Server (AS) – сервер приложений



DBS:

Достоинства:

возможность централизованного администрирования прикладных функций;
 снижение трафика (вместо SQL-запросов по сети направляются вызовы хранимых процедур);
 возможность разделения процедуры между несколькими приложениями;
 экономия ресурсов компьютера за счет использования единожды созданного плана выполнения процедуры.

Недостатки:

ограниченность средств написания хранимых процедур, представляющих собой разнообразные процедурные расширения SQL, которые уступают по изобразительным средствам и функциональным возможностям в сравнении с языками C или Pascal. Сфера их использования ограничена конкретной СУБД из-за отсутствия возможности отладки и тестирования разнообразных хранимых процедур.

RDA:

Достоинства:

большое количество готовых СУБД с SQL-интерфейсами и инструментальных средств, которые обеспечивают быстрое создание клиентских программ..

Недостатки:

достаточно высокая нагрузка системы передачи данных, т.к. вся логика сосредотачивается в приложении, а данные, которые обрабатываются, располагаются на удаленном узле.

неудобство в разработке, модификации и сопровождении

AS:

Достоинства:

универсальность и гибкость, т.к. функции приложения разделяются на 3 независимые составляющие.

Недостатки:

высокое потребление ресурсов компьютеров при обмене информацией между составляющими приложения.

15. Реляционная модель хранилищ данных (ROLAP)

ROLAP-системы позволяют представлять данные, хранимые в классической реляционной базе, в многомерной форме или в плоских локальных таблицах на файл-сервере, обеспечивая преобразование информации в многомерную модель через промежуточный слой метаданных. Агрегаты хранятся в той же БД в специально созданных служебных таблицах. В этом случае гиперкуб эмулируется СУБД на логическом уровне.

Преимущества ROLAP.

- Реляционные СУБД имеют реальный опыт работы с очень большими БД и развитые средства администрирования. При использовании ROLAP размер хранилища не является таким критичным параметром, как в случае MOLAP.
- При оперативной аналитической обработке содержимого хранилища данных инструменты ROLAP позволяют производить анализ непосредственно над хранилищем (потому что в подавляющем большинстве случаев корпоративные хранилища данных реализуются средствами реляционных СУБД).
- В случае переменной размерности задачи, когда изменения в структуру измерений приходится вносить достаточно часто, ROLAP системы с динамическим представлением размерности являются оптимальным решением, так как в них такие модификации не требуют физической реорганизации БД, как в случае MOLAP.
- Системы ROLAP могут функционировать на гораздо менее мощных клиентских станциях, чем системы MOLAP, поскольку основная вычислительная нагрузка в них ложится на сервер, где выполняются сложные аналитические SQL-запросы, формируемые системой.
- Реляционные СУБД обеспечивают значительно более высокий уровень защиты данных и хорошие возможности разграничения прав доступа.

Недостатки ROLAP.

- Ограниченные возможности с точки зрения расчета значений функционального типа.
- Меньшая производительность, чем у MOLAP. Для обеспечения сравнимой с MOLAP производительности реляционные системы требуют тщательной проработки схемы БД и специальной настройки индексов. Но в результате этих операций производительность хорошо настроенных реляционных систем при использовании схемы "звезда" сравнима с производительностью систем на основе многомерных БД.

16. Сложные типы данных ОРМД – строчные типы. Примеры

Строчные типы являются группами элементов разных типов и формируют шаблон записи, который впоследствии можно использовать при определении таблиц.

Для вставки данных в таблицу, включающую строчный тип, используется оператор «::».

Для манипуляции отдельными полями строчного типа используется точечная нотация.

При обновлении строчного типа в запросе необходимо указывать все поля, а не только те, которые обновляются. Этим подчеркивается, что данные поля не меняются.

Именованный пример:

```
create row type address_t (street char(30),
                           city char(20),
                           state char(2),
                           zip char(5));
```

Неименованный пример:

```
create table customer (customer_no integer,  
                        name char(30),  
                        address row (street char(30),  
                        city char(20),  
                        state char(2),  
                        zip char(5));  
                        balance money);
```

17. Требования к обнаруживаемым знаниям

Свойства обнаруживаемых знаний:

- Знания должны быть новые, ранее неизвестные.
- Знания должны быть нетривиальны.
- Знания должны быть практически полезны.
- Знания должны быть доступны для понимания человеку.

18. Структура метки доступа в мандатной модели

Метка объекта:

- Группа субъекта, который внес данный объект.
- Уровень доступа на чтение – RAL
- Уровень доступа на запись – WAL

Метка субъекта:

- Группа, к которой принадлежит субъект (класс субъекта);
- RAL-уровень субъекта; пользователь может получать (читать) информацию, RAL-уровень которой не выше его собственного уровня доступа;
 ↖ **принцип простого свойства секретности (simple security property) - No Read Up**
- WAL-уровень субъекта; пользователь может вносить информацию только в объекты такого же WAL-уровня доступа.
 ↖ ***-свойство - No Write Down**

19. Сложный тип данных «Multiset». Как его задать?

Multiset (мультимножество) – пример неупорядоченной (в отличие от **set**) коллекции, допускающий дублирование элементов.

```
multiset {"ха", "ха", "хи", "хо"}
```