

Евдокимова Дарья

Подготовка к БД 2024г

ОТ ДЕДОВ

- Что-то от дедов [номер раз](#), [номер два](#).
- [Тесты от Гриши](#)

ТЕСТЫ

- Тесты 1 и 2:
<https://docs.google.com/forms/d/e/1FAIpQLSftTzNRV76ZmXtdrQcRH9CDDsvL3789nO-YcFQZ0MhlaWmeig/viewform>
- Расширение вопросов на тесты 1 и 2:
<https://docs.google.com/forms/d/e/1FAIpQLSdOIDxjT-jBs8ms8jPF2t5TVnluvgrA137GieY4cQuSgU6HQ/viewform>
- Тест 3:
https://docs.google.com/forms/d/e/1FAIpQLSfrfL_1Pk5vMY-Xk0UJlyaph_xV5lvN--E5IXnkQEJVAEoE6w/viewform
- Тест 4:
https://docs.google.com/forms/d/e/1FAIpQLSfvjvK4Eeg-Cxr-HtwIL0t3g58F2wXI7zBabbPTFi_bkudU2g/viewform
- Тест 5:
https://docs.google.com/forms/d/e/1FAIpQLSf0JYrE9lZy7EQw51Slfq96A59VI_ruyvLPFE0Sc0n8dX_TQ/viewform
- Тест 6:
https://docs.google.com/forms/d/e/1FAIpQLSeiuzk8FTSroiivdO3dwtNPw4KQ9_1C9aRqJ4DVL3-RqhixOg/viewform

ЧЁТ ПОЛЕЗНОЕ

- [DDL, DML, DCL и TCL в языке SQL](#)
- Почитать про [домены](#)
- Почитать про [реляционную алгебру](#)
- Какой-то [онлайн учебник](#)
- Еще крутой [онлайн-учебник](#)
- Ещё [полезный учебник](#)
- Почитать про функциональные зависимости и нормализацию: [раз](#), [два](#), [три](#)
- Сборная [солянка](#) про нормализацию и нормальные формы
- Почитать про функции и процедуры [здесь](#)
- Про RAID-ы смотреть [тут](#), читать [тут](#), [тут](#)
- Очень крутой [теоретический курс по БД](#) (тут про реляционное исчисление на БД вроде намана так рассказывает), вот [тут](#) его лекции в виде презенташек
- Хороший файл про [проектирование БД](#), ооочень хорошо написано про проектирование БД (физическое, логическое, концептуальное) [вот тут](#)
- [Глоссарий](#) какой-то (полезный кста) - кратенько написано всё
- [Основы sql](#) видео курс
- [Симулятор sql от karpov courses](#)

ОПРЕДЕЛЕНИЯ

Предметная область

Часть реального мира, для которой создается информационная система

Данные

Представление фактов или идей в формализованном виде, пригодном для обработки в некотором информационном процессе

База данных

Набор связанных данных

Система управления базой данных (СУБД-Database Management System)

Пакет программ, обеспечивающих средства создания и манипулирования базой данных на ЭВМ

Система базы данных (Database System)

Пакет программ СУБД вместе с данными. Иногда в систему базы данных включают и приложения.

Определения термина «База данных»

- **База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
- **База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения.
- **База данных** — организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.
- **База данных** — некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия.
- **База данных** — совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации

Уровни “модели данных”

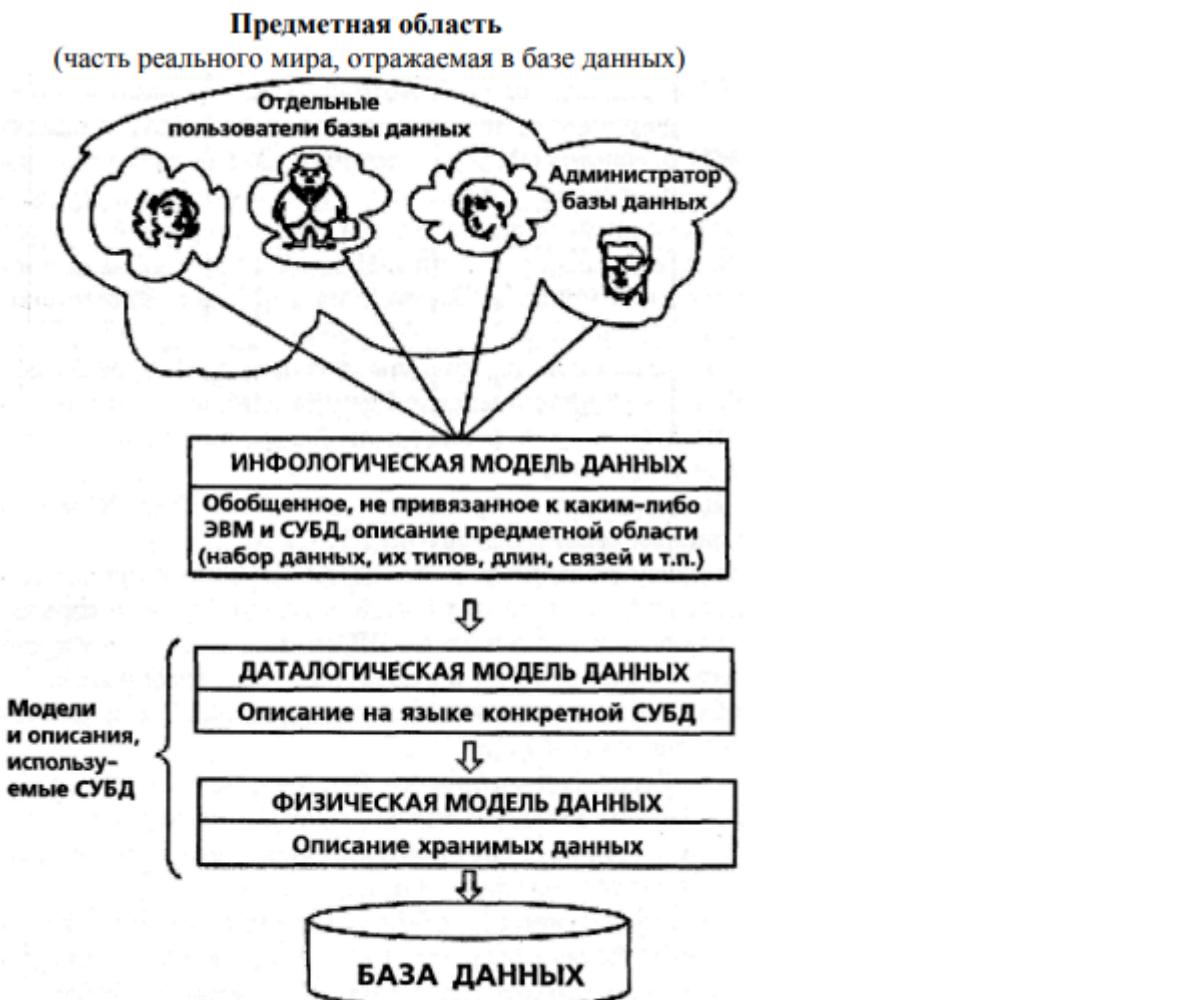


Рис. 1.3. Уровни моделей данных

Модель данных

Модель данных – фиксированная система понятий и правил для представления структуры данных, состояния и динамики проблемной области в базах данных [12]. Как правило, задается языком определения данных и языком манипулирования данными. Примерами модели данных, получившими широкое распространение, являются модели данных сетевая, иерархическая, реляционная и др.

Модель данных состоит из трех компонент [11].

1. Структура данных для представления точки зрения пользователя на базу данных.
2. Допустимые операции, выполняемые на структуре данных. Они составляют основу языка данных рассматриваемой модели данных. Одной лишь хорошей структуры данных недостаточно. Необходимо иметь возможность работать с этой структурой при помощи различных операций языка определения данных и языка манипулирования данными. Богатая структура данных ничего не стоит, если нет возможности оперировать ее содержимым.
3. Ограничения для контроля целостности. Модель данных должна быть обеспечена средствами, позволяющими сохранять ее целостность и защищать ее.

Схема

Схема – это средство, с помощью которого определяется модель данных приложения. В действительности схема содержит не только модель данных: в ней присутствует также некоторая семантическая информация, относящаяся к конкретному приложению. В модели данных можно определить, например, что база данных будет хранить информацию об организациях и служащих. Однако, тот факт, что данный служащий не может работать более чем в одной организации, отражает семантику приложения. Это семантическое ограничение должно выполняться для каждого отдельного экземпляра записи базы данных об этом служащем. Поддержка ограничений заданной модели данных в базе данных также является частью функций СУБД по обеспечению защиты и целостности.

Отношение 1 опр

Основой структур данных в модели является теоретико-множественное понятие
ОТНОШЕНИЯ

Определение 1

- Пусть даны D_1, D_2, \dots, D_n – множества (домены).
- Декартово произведение $D_1 \times D_2 \times \dots \times D_n$ состоит из множества кортежей $\{v_1, v_2, \dots, v_n\}$, длины n , таких что $v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$
- Отношением называется КОНЕЧНОЕ подмножество декартового произведения доменов $D_1 \times D_2 \times \dots \times D_n$
- Запись вида $R(A_1:D_1, A_2:D_2, \dots, A_k:D_n)$
или $R(A_1, A_2, \dots, A_k)$ называют **схемой отношения**,
где $k \geq n$,
 R – имя отношения,
 A_i – имена атрибутов,
 D_j - имена доменов
- Все атрибуты A_i в схеме отношения различны.

Отношение 2 опр

Определение 2

- Пусть A_1, A_2, \dots, A_n – имена атрибутов, D_1, D_2, \dots, D_m – имена доменов.
- Конечное множество пар вида $\{A_i, D_j\}$ называется схемой отношения, если $A_t \neq A_s$, при $t \neq s$ где $t, s \in \{1, 2, \dots, n\}$. Имена доменов для разных атрибутов могут совпадать.
- Множество пар $\{A_i, d_j\}$ называется кортежем отношения. где A_i – имя атрибута из A_1, A_2, \dots, A_n , а d_j из D_j
- Множество кортежей, соответствующих одной схеме отношения называется отношением
- Число атрибутов в отношении – **степень отношения**
- Число кортежей в отношении – **кардинальное число**

Примеры. Пусть $n=2$, $D_1=\{0,1\}$, $D_2=\{a,b,c\}$.

Тогда отношение $R (A_1, A_2)$ есть $\{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$

Отношение. Неформальные определения. Атрибут

Неформальные определения

- **Отношение** – формальная структура, используемая для моделирования СУЩНОСТЕЙ и СВЯЗЕЙ
- Неформально, **отношение** - это **двумерная таблица** данных, состоящая из **строк, столбцов**.
- Каждая строка содержит **набор элементов данных** о некоторых **сущностях или связях**
 - В формальной модели строки называются кортежами
- Каждый столбец содержит данные о свойствах сущностей или связи, то есть некоторого атрибута.
- Каждый столбец имеет **заголовок**, что позволяет судить о значении элементов данных в этом столбце
 - В формальной модели заголовок столбца, называется **именем атрибута** (или просто **атрибут**)
- Схема отношения – имя отношения с набором атрибутов

Отношения и таблицы

Отношения и таблицы

■ Столбцы и атрибуты

- Столбцы – список
- Атрибуты – множество

■ Строки и кортежи

- Строки – элемент расширенного декартова произведения
- Кортежи – элемент декартова произведения¹

■ Таблицы и отношения

- Таблица – список строк
- Отношение – множество кортежей

Функциональная зависимость

Функциональная зависимость (ФЗ)

■ $X \rightarrow Y$

- Каждому X соответствует ровно один Y
- X и Y – множества атрибутов

■ Все атрибуты функционально зависят от (над)ключа

■ Функциональные зависимости нельзя «вычислить»

Примеры взято [отсюда](#):

Примеры ФЗ (1)

<u>CId</u>	Name	Lecturer	Phone
1	СУБД	Корнеев Г.А.	111 1111
2	Пр. ПО	Киракозов А.Х.	222 2222
3	Ан. дан.	Муравьёв С.В.	333 3333
4	Ассемблер	Скаков П.С.	444 4444

■ Функциональные зависимости?

- CId → Name Lecturer Phone
- Name → CId Lecturer Phone
- Name → Lecturer Phone



Примеры ФЗ (2)

<u>CId</u>	<u>SId</u>	RMark	EMark
1	1	5	A
2	2	4	B
3	3	4	C
4	4	3	D

■ Функциональные зависимости

- CId SId → RMark EMark
- EMark → RMark

Домен

- Домен – понятие специфичное для баз данных
- Домен задается **базовым типом** и логическим выражением, определяющим элементы домена

Пример.

Домен **ИМЕНА** может базироваться на строковом типе, но содержать только те строки, которые НЕ начинаются на Ъ или Ъ знак

Из вики: тип данных, множество допустимых значений

Ключ отношения

Атрибут или множество атрибутов, однозначно определяющих кортеж в отношении, называется ключом отношения.

Пр: В таблице Студент ключ - № студ. Билета

Ключ суррогатный

Иногда идентификатор строки или порядковое число используются как ключи для идентификации строки в таблице.

Пр: Такой ключ называют искусственным или суррогатным ключ.

Степень отношения, Кардинальное число

- Число атрибутов в отношении – **степень отношения**
- Число кортежей в отношении – **кардинальное число**

КЛЮЧИ

Пусть $R(A_1, A_2, \dots, A_n)$ - отношение, $F - \Phi 3$,
а X подмножество $\{A_1, A_2, \dots, A_n\}$.

X называется **ключом**, если

1. $X \rightarrow A_1, A_2, \dots, A_n$ принадлежит F^+ ,
2. Ни для какого собственного подмножества $Y \subset X$,
 $Y \rightarrow A_1, A_2, \dots, A_n$ не принадлежит F^+ .

Множество атрибутов, содержащее ключ в качестве подмножества, называется суперключем.

Иногда используется термин возможный ключ

Операции в реляционной алгебре

Основные восемь операций реляционной алгебры были предложены Э.Коддом.

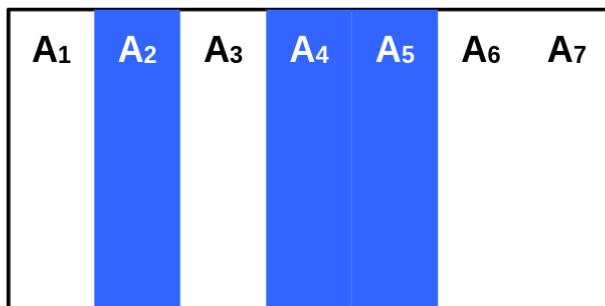
- Объединение
- Пересечение
- Вычитание
- Декартово произведение
- Выборка
- Проекция
- Соединение
- Деление

Дальше будут примеры, взяты [отсюда](#)

Проекция

Проекция

- $\pi_{A_1, A_2, \dots, A_n}$ – проекция на атрибуты A_1, A_2, \dots, A_n
- Заголовок
 - Пересечение с $\{A_1, A_2, \dots, A_n\}$
- Тело
 - Пересечение кортежей с $\{A_1, A_2, \dots, A_n\}$



Пример проекции (1)

■ $\pi_{FirstName, LastName}$

	Id	LastName	FirstName	
1		Иванов	Иван	
2		Петров	Петр	
3		Сидоров	Сидор	
4		Кулебякин	Иван	

→

LastName	FirstName
Иванов	Иван
Петров	Петр
Сидоров	Сидор
Кулебякин	Иван

Пример проекции (2)

■ $\pi_{FirstName}$

	Id	LastName	FirstName	
1		Иванов	Иван	
2		Петров	Петр	
3		Сидоров	Сидор	
4		Кулебякин	Иван	

→

FirstName
Иван
Петр
Сидор

Выборка (фильтрация, сечение)

Фильтрация (сечение, выборка)

■ $\sigma_{condition}$ – фильтрация по условию

■ Заголовок

- Сохраняется

■ Тело

- Кортежи, удовлетворяющие условию

A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇

Пример фильтрации (1)

■ $\sigma_{Id>2}$

Id	LastName	FirstName
1	Иванов	Иван
2	Петров	Петр
3	Сидоров	Сидор
4	Кулебякин	Иван

→

Id	LastName	FirstName
3	Сидоров	Сидор
4	Кулебякин	Иван

Пример фильтрации (2)

■ $\sigma_{Id>2 \wedge FirstName=Иван}$

Id	LastName	FirstName
1	Иванов	Иван
2	Петров	Петр
3	Сидоров	Сидор
4	Кулебякин	Иван

→

Id	LastName	FirstName
4	Кулебякин	Иван

Пример фильтрации (3)

■ $\sigma_{length.FirstName)+2 \geq length.LastName}$

Id	LastName	FirstName
1	Иванов	Иван
2	Петров	Петр
3	Сидоров	Сидор
4	Кулебякин	Иван

→

Id	LastName	FirstName
1	Иванов	Иван
2	Петров	Петр
3	Сидоров	Сидор

Объединение

Пример объединения

- $R_1 \cup R_2$

Id	FirstName
1	Иван
2	Петр
3	Сидор

Id	FirstName
1	Иван
1	
4	Сидор



Id	FirstName
1	Иван
1	
2	Петр
3	Сидор
4	Сидор

Пересечение

Пример пересечения

- $R_1 \cap R_2$

Id	FirstName
1	Иван
2	Петр
3	Сидор

Id	FirstName
1	Иван
1	
4	Сидор

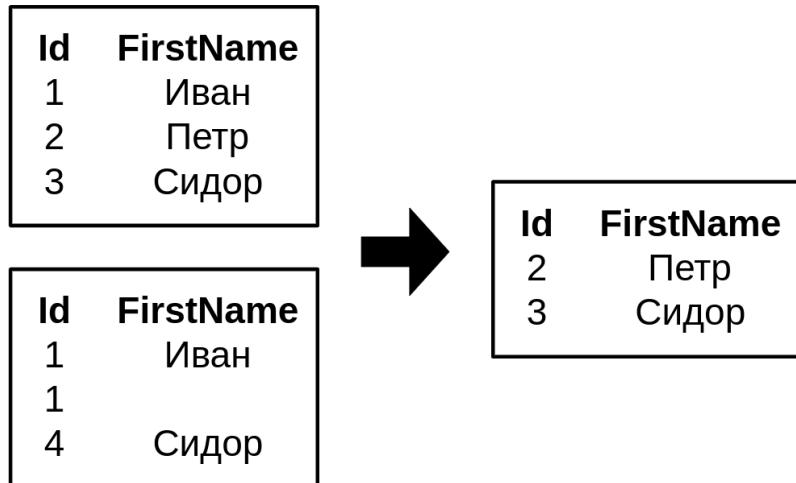


Id	FirstName
1	Иван

Разность (вычитание)

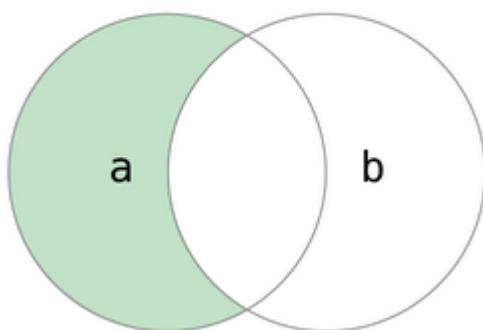
Пример разности

- $R_1 \setminus R_2$



Разность

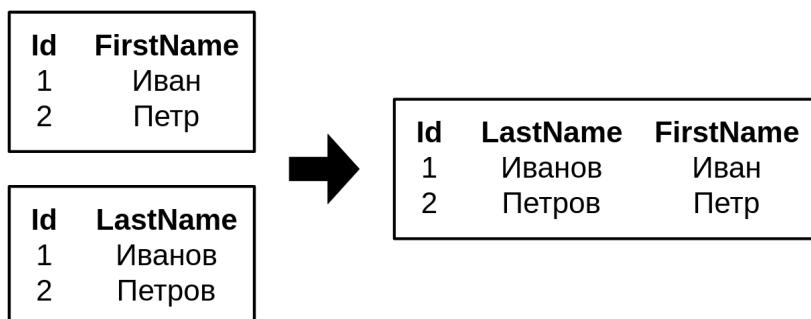
a.subtract(b)



Соединение

Соединения

- Соединение – объединяет данные двух отношений



Деление

Деление

■ $Q(XY) \div S(Y)$ – деление

- Найти максимальное X , такое, что $X \times S \subseteq Q$
- $Q \div S \equiv \{x | x \in \pi_X(Q), \{x\} \times S \subseteq Q\}$
- $Q \div S \equiv \pi_X(Q) \setminus \pi_X(\pi_X(Q) \times S \setminus Q)$

■ Интуиция

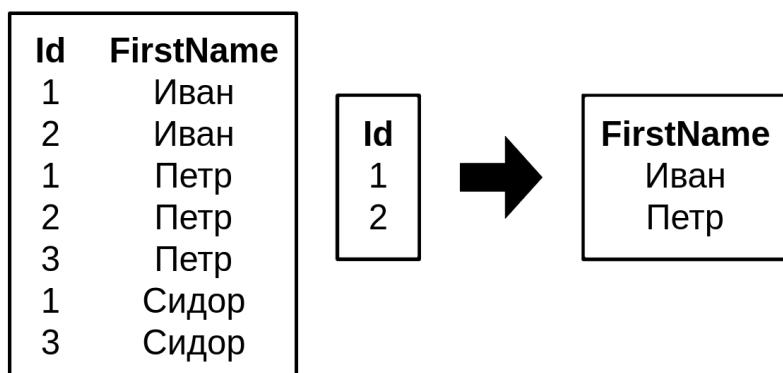
- Запрос «для всех»
- x , для которых есть пара для каждого y
 - ◆ $x \in \pi_X(Q) : \forall y \in S : (x, y) \in Q$

■ Заголовок – X

- $S \subseteq Q$

Пример деления

■ $Q \div S$



Здесь делитель S – это $\{1, 2\}$. И надо выбрать такие имена, которые встречаются И с $id = 1$ И с $id = 2$. Это Иван и Пётр. Пётр с $id = 3$ никак тут ничему не мешает. А вот Сидора (ммм, персиковый сидр) тут нет, тк нет Сидора с $id = 2$.

Декартово произведение (полное соединение)

Полное соединение

- $R_1 \times R_2$ – декартово произведение двух отношений
- Заголовок
 - Объединение заголовков
 - Различные имена атрибутов
- Тело
 - Декартово произведение тел отношений

Пример полного соединения

- $R_1 \times R_2$

The diagram illustrates the Cartesian product of two relations, R_1 and R_2 . On the left, there are two separate tables: R_1 (containing columns **Id1** and **FirstName**) and R_2 (containing columns **Id2** and **LastName**). An arrow points from these two tables to a larger table on the right, which represents their Cartesian product. This resulting table has four columns: **Id1**, **FirstName**, **Id2**, and **LastName**. It contains 12 rows, showing every possible combination of names from both relations.

Id1	FirstName	Id2	LastName
1	Иван	1	Иванов
1	Иван	3	Петров
1	Иван	4	Сидоров
2	Петр	1	Иванов
2	Петр	3	Петров
2	Петр	4	Сидоров
3	Сидор	1	Иванов
3	Сидор	3	Петров
3	Сидор	4	Сидоров

Натуральное соединение (естественное)

Естественное соединение

- $R_1 \bowtie R_2$ – соединение по одноимённым атрибутам
- Заголовок
 - Объединение заголовков
- Тело
 - «Сплющенные» пары кортежей с равными одноимёнными атрибутами

Пример естественного соединения

- $R_1 \bowtie R_2$

The diagram illustrates the natural join operation (\bowtie) between two tables, R_1 and R_2 , producing a third table.

Table R_1 :

Id	FirstName
1	Иван
2	Петр
3	Сидор

Table R_2 :

Id	LastName
1	Иванов
1	Петров
2	Сидоров

Resulting Table:

Id	FirstName	LastName
1	Иван	Иванов
1	Иван	Петров
2	Петр	Сидоров

Размер естественного соединения

- Пусть $n = |R_1|$, $m = |R_2|$
- Максимальный $|R_1 \bowtie R_2|$
 - $n \times m$ – нет общих атрибутов
- Минимальный $|R_1 \bowtie R_2|$
 - 0 – нет равных атрибутов

Примеры левых и правых соединений left join right join

Примеры косых соединений

- $R_1 \bowtie R_2$ и $R_1 \bowtie R_2$

The diagram illustrates the semi-join and anti-join operations between two tables, R_1 and R_2 .

Table R_1 :

Id	FirstName
1	Иван
2	Петр
3	Сидор

Table R_2 :

Id	FirstName	LastName
1	Иван	Иванов
1	Иван	Петров
2	Петр	
3	Сидор	Сидоров

Resulting Table (Left Join):

Id	FirstName	LastName
1	Иван	Иванов
1	Иван	Петров

Resulting Table (Right Join):

Id	FirstName	LastName
1	Иван	Иванов
1	Иван	Петров
3	Сидор	Сидоров
4		Плюшкин

Нормализация

(не из презентах)

Это удаление избыточности данных.

Избыточность - когда мы храним больше данных, чем необходимо.

Избыточность ведет к

- увеличению занимаемого места на диске
- аномалиям включения, обновления, удаления

Устранение избыточности производится, как правило, за счет декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (т.е. факты, невыводимые из других хранимых фактов).

Хорошо про нормальные формы + базовые определения: <https://habr.com/ru/articles/254773/>

1 НФ

Из презы:

Отношение R находится в 1 НФ, если все атрибуты R определены на простых доменах.
Выполняется по определению.

Или так можно:

Переменная отношения находится в 1 НФ тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

ПОНЯТНО:

- 1) В каждой клеточке таблицы должно быть только одно значение.
- 2) Не должно быть повторяющихся строк.

ну или так еще можно

Первая нормальная форма

■ Первая нормальная форма (1НФ)

- В отношении нет повторяющихся групп
- Все атрибуты атомарны
- У отношения есть ключ

■ 1НФ (Дейт)

- $1\text{НФ} \equiv \text{отношение}$

2 НФ

Из презы:

- **Отношение R находится в 2 НФ**, если оно находится в 1 НФ и все неключевые атрибуты R функционально полно зависят от ключа.

Множество атрибутов X функционально полно зависит от ключа K, если

- ❖ имеется зависимость $K \rightarrow X$
- ❖ $\neg \exists Z \subset K$, что $Z \rightarrow X$

Проект(N проекта, N сотр, N эт,

Дата_нач, Дата_кон, Назв_проекта, N_отдела, Назв_отдела)

Ф3:

1. $N_{\text{проекта}}, N_{\text{сотр}}, N_{\text{эт}} \rightarrow \text{Дата_нач}, \text{Дата_кон}, \text{Назв_проекта}, N_{\text{отдела}}, \text{Назв_отдела}$
2. $N_{\text{проекта}}$ — Назв_проекта %нарушает 2 НФ%

Нарушение 2 НФ **НЕ ВОЗМОЖНО**, если **ключ** состоит из **одного атрибута**

15

Или:

Переменная отношения находится в 2 НФ тогда и только тогда, когда она находится в 1й НФ и каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа.

ПОНЯТНО:

- 1) Таблица в 1й НФ
- 2) Есть первичный ключ
- 3) Все атрибуты зависят от первичного ключа целиком, а не от какой-то его части.

Пример:

2 НФ - до нормализации

name	project	project_client
Суворов Алексей	КайзерДом	Сергей И.
Суворов Алексей	Юнитраст	Иван С.
Окулов Ярослав	Доска почета	Аркадий Г.
Созинов Илья	КайзерДом	Сергей И.
Збуров Роман	КайзерДом	Сергей И.
Збуров Роман	Юнитраст	Иван С.

2 НФ - добавим первичный ключ

worker_id	project_id	name	project	project_client
1	1	Суворов Алексей	КайзерДом	Сергей И.
1	2	Суворов Алексей	Юнитраст	Иван С.
2	3	Окулов Ярослав	Доска почета	Аркадий Г.
3	1	Созинов Илья	КайзерДом	Сергей И.
4	1	Збруев Роман	КайзерДом	Сергей И.
4	2	Збруев Роман	Юнитраст	Иван С.

Теперь у нас составной ключ.

Важно! Все атрибуты зависят от первичного ключа целиком, а не от какой-то его части.

У нас получается, что имя зависит только от части первичного ключа - а именно - worker_id, проект зависит только от части первичного ключа - а именно - project_id. То есть некоторые атрибуты зависят только от части первичного ключа, а должна быть целиком зависима.

Решение - декомпозиция. И вот итоговая таблица

2 НФ - Декомпозиция		
project_id	project	project_client
1	КайзерДом	Сергей И.
2	Юнитраст	Иван С.
3	Доска почета	Аркадий Г.
worker_id	project_id	name
1	1	Суворов Алексей
1	2	Окулов Ярослав
2	3	Созинов Илья
3	1	Збруев Роман
4	1	
4	2	

3 НФ

Из презы:

- **Отношение R находится в 3 НФ, если оно находится в 2 НФ и все неключевые атрибуты R нетранзитивно зависят от ключа.**

Или:

Переменная отношения находится в 3й НФ тогда и только тогда, когда она находится во 2й НФ и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

ПОНЯТНО:

- 1) Таблица во 2й НФ
- 2) Все атрибуты зависят только от первичного ключа, но не от других атрибутов

Пример:

tyre_id	tyre_name	supplier	supplier_phone
1	Nokian Hakka	ИП Шлепаков Л.	+79233213123
2	Nokian Nordman	ООО Русский Север	+79242343423
3	Continental Contilce	ООО Автомакс	+79434234234
4	Matador	ИП Кузнецов П.	+79989493248
5	Nordmaster	ИП Иванов П.	+79646564345

Здесь проблема в том, что номер зависит от поставщика, а от идентификатора шины он никак не зависит.

Вот здесь таблицы в 3й НФ

tyre_id	tyre_name	supplier_id
1	Nokian Hakka	1
2	Nokian Nordman	2
3	Continental Contilce	3
4	Matador	4
5	Nordmaster	5

id	supplier	supplier_phone
1	ИП Шлепаков Л.	+79233213123
2	ООО Русский Север	+79242343423
3	ООО Автомакс	+79434234234
4	ИП Кузнецов П.	+79989493248
5	ИП Иванов П.	+79646564345

Здесь поставщиков отделяем от шин. Делаем суррогатный первичный ключ.

НФ Бойса-Кодда

Из презы:

■ Нормальная форма Бойса-Кодда (НФБК)

- Тривиальные ФЗ. Говорят, что функциональная зависимость $A_1 A_2 \dots A_n \rightarrow B$ является тривиальной, если атрибут B совпадает с любым из атрибутов A_i , $i = 1, \dots, n$
- Нетривиальные ФЗ. $A_1 A_2 \dots A_n \rightarrow B_1, B_2, \dots, B_k$ называется нетривиальной, если по меньшей мере один из атрибутов B_i не является элементом множества $\{A_1, A_2, \dots, A_n\}$
- Отношение R удовлетворяет BCNF, т. и т. т., когда для любой нетривиальной зависимости в R вида $A_1 A_2 \dots A_n \rightarrow B$ множество $\{A_1, A_2, \dots, A_n\}$ образует суперключ для R .

Или

Переменная отношения находится в НФ Бойса-Кодда (иначе - в усиленной 3 НФ) тогда и только тогда, когда каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

ПОНЯТНО:

- 1) Таблица в 3 НФ
- 2) Ключевые атрибуты не должны зависеть от неключевых

Пример:

НФБК - Пример

Дано:

- Каждый сотрудник может курировать только ту работу для которой он квалифицирован
 - Максим - курирует маркетинг,
 - Рома - программирование,
 - Илья - дизайн
 - ...
- Есть куча проектов над которыми они работают
- Для каждого из проектов могут быть выполнены и Разработка и дизайн и маркетинг
- Куратор по каждому из направлений у проекта может быть только один (дабы не было неразберихи)

НФБК - Пример

project_id	task	responsible
1	Разработка	Рома
2	Маркетинг	Максим
2	Дизайн	Илья
1	Дизайн	Илья
3	Маркетинг	Диана
3	Разработка	Миша

Так быть не должно:

НФБК - Пример

- Первичный ключ - составной Проект + Задача
 - Но проявляется зависимость части первичного ключа (задачи) от ответственного
 - Зная кем является ответственный можно четко сказать какую задачу он выполняет в проекте

Решение - декомпозиция

НФБК - Пример

id	name	skill
1	Рома	Разработка
2	Максим	Маркетинг
3	Илья	Дизайн
4	Илья	Дизайн
5	Диана	Маркетинг
6	Миша	Разработка

project_id	responsible_id
1	1
2	2
2	3
1	4
3	5
3	6

Или так можно:

НФ Бойса-Кодда

- Нормальная форма Бойса-Кодда (НФБК)
 - В каждой нетривиальной функциональной зависимости $X \rightarrow Y$, X является надключом
 - НФБК сильнее чем ЗНФ
 - ЗНФ и неперекрывающиеся ключи \Rightarrow НФБК
 - Приведение к НФБК
 - Декомпозиция по «неправильным» ФЗ

4 HΦ

Переменная отношения находится в 4 НФ, если она находится в нормальной форме Бойса-Кодда и не содержит нетривиальных зависимостей.

ПОНЯТНО:

- 1) Таблица находится в НФ БК
- 2) Устраняются многозначные зависимости

Пример:

4 НФ - Пример

Атрибуты Проект и Увлечение напрямую зависят от первого столбца, но друг от друга они независимы

worker_id	project	hobbie
1	Сантехник	Радиотехника
1	КайзерДом	Гитара
2	FabioRoss	Футбол
3	КайзерДом	Хоккей
3	Доска Почета	Гитара

Решение: отделить хобби от проектов:

4 НФ - Пример

worker_id	project	worker_id	hobbie
1	Сантехник	1	Радиотехника
1	КайзерДом	1	Гитара
2	FabioRoss	2	Футбол
3	КайзерДом	3	Хоккей
3	Доска Почета	3	Гитара

workers_projects

workers_hobbies

5 НФ

ПОНЯТНО:

- 1) Таблица находится в 4 НФ
- 2) Устраняются нетривиальные зависимости
- 3) Декомпозиция без потерь

Курсор

Курсор (cursor) – объект, связанный с запросом

Курсыры

Курсор (cursor) – объект, связанный с запросом

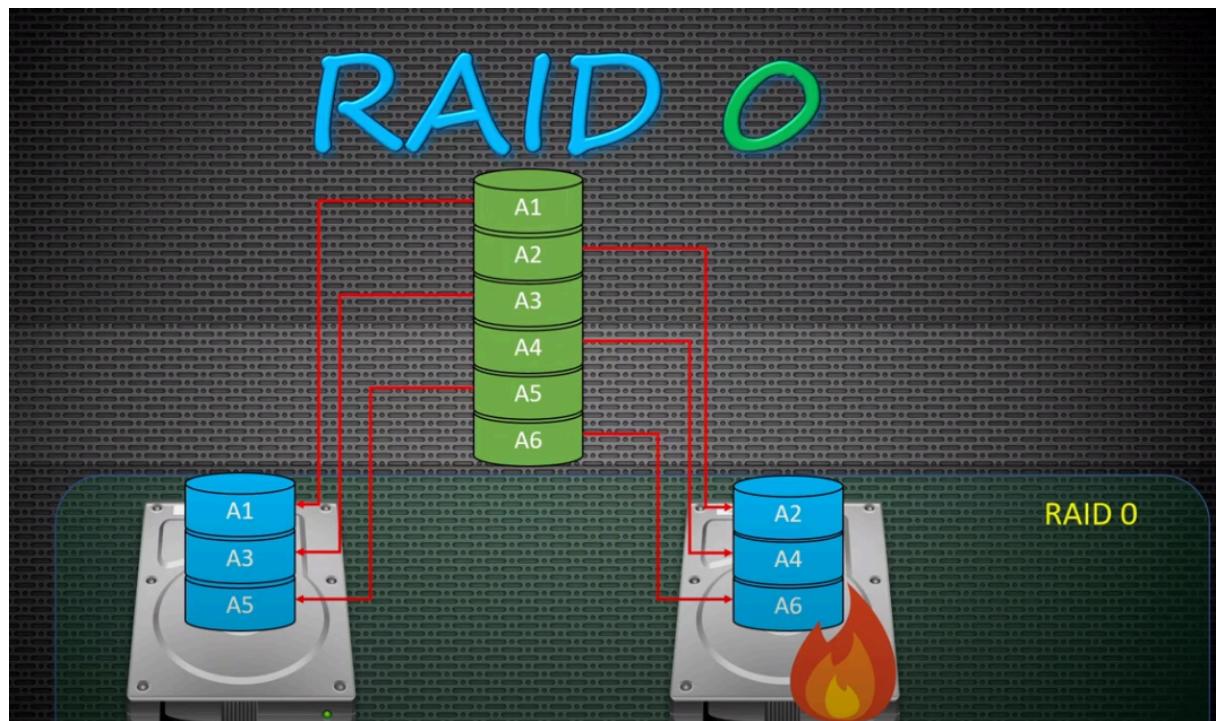
[Объявление курсора](#)

```
EXEC SQL DECLARE CURSOR Томск_продавцы  
FOR SELECT snum, sname, city, comm  
      FROM Salespeople  
      WHERE city = 'Томск';
```

Чем то похожи на VIEW, но в отличие от базовых таблиц и
VIEW у курсоров строки [упорядочены](#).

Про RAID-ы

RAID0: Чередование (Striping)



Описание: Данные распределены по всем дискам массива равномерно. В массиве участвуют два или более дисков

Производительность: Одновременно может быть записан и прочитан бит данных

Плюсы: Быстродействие чтения/записи

Минусы: Нет резервирования. Любой диск вышедший из строя приведет к разрушению массива и как следствие потере всех данных

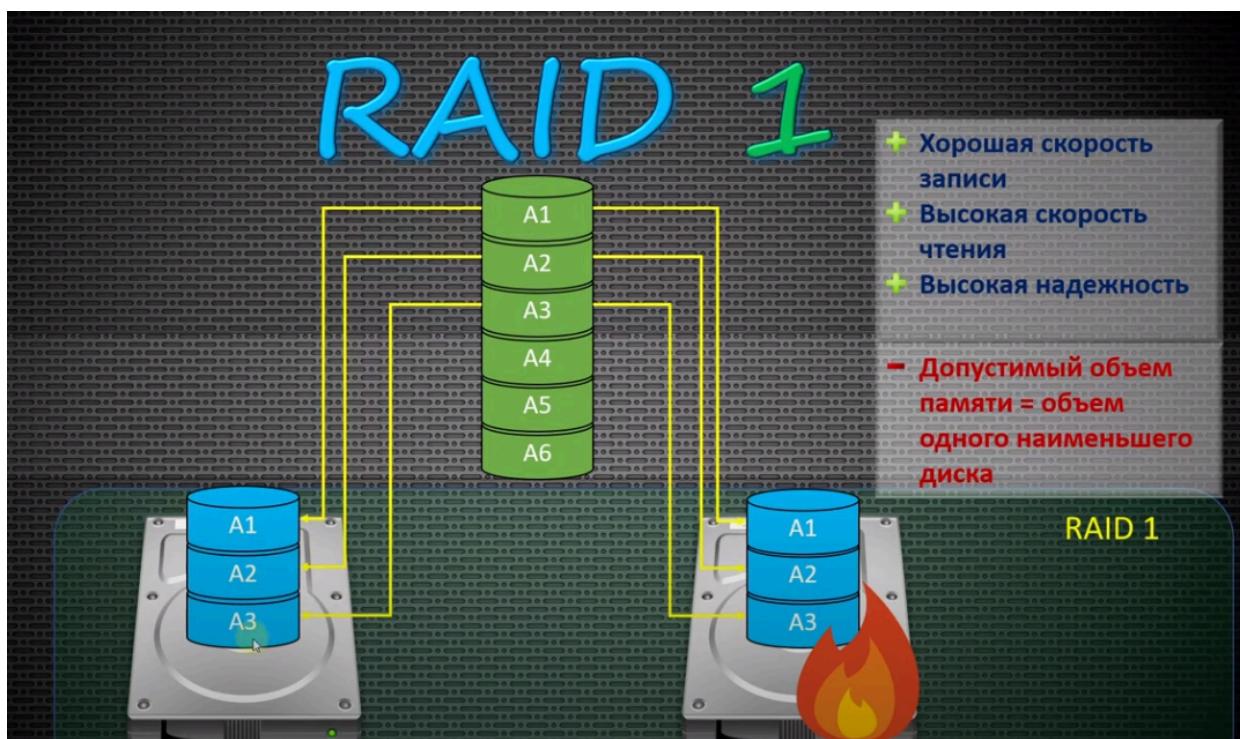
Использование: Приложения, которым необходим скоросной обмен данными, хранилище временных файлов, некритичные данные.

=====

Резюме:

- высокая производительность
- высокая пропускная способность
- низкая стоимость
- нет контроля четности (при отказе диска данные на нем становятся недоступными до тех пор, пока их не удастся перезаписать с другого диска)

RAID1: Зеркалирование (Mirroring)



Описание: Запись/чтение данных происходит одновременно на два или более дисков массива

Производительность: Операции чтения выполняются быстрее т.к. данныечитываются со всех дисков массива одновременно. Операции записи медленнее т.к. запись выполняется дважды или более раз (зависит от количества дисков в массиве)

Плюсы: Выход из строя любого количества дисков массива кроме последнего не приводит к потере данных

Минусы: Стоимость. Пропорциональна количеству дисков в массиве

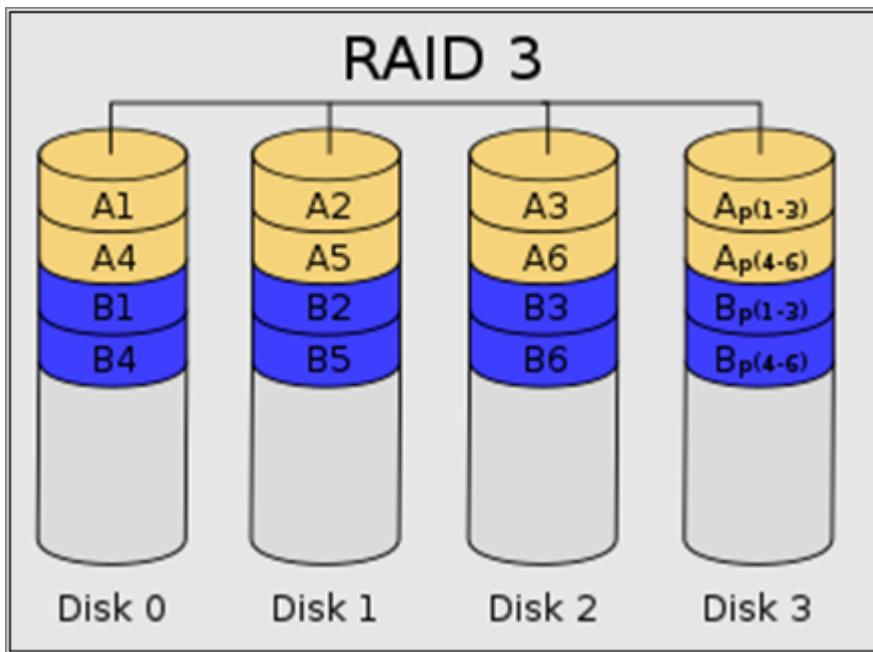
Использование: Системные разделы, разделы с важными данными, приложения использующие транзакции.

=====

Резюме:

- быстрое чтение (за счет зеркалирования)
- мгновенное восстановление после сбоя (если первичный диск выходит из строя, вторичный, зеркальный диск может взять на себя его функции, поскольку данные, операционная система и прикладное программное обеспечение реплицируются на нем)
- низкая скорость записи (поскольку данные должны быть записаны на диски дважды)
- удвоение объема дискового пространства (тк все данные дублируются)

RAID3: Чередование с выделенным диском чётности (Virtual disk blocks)



Описание: Данные чередуются по дискам массива на уровне байтов. Необходим дополнительный диск на котором хранится информация о четности. Минимально три диска в массиве

Производительность: Низкая на операциях записи

Плюсы: Данные остаются полностью доступными при выходе из строя одного диска

Минусы: Производительность

Использование: Редко меняющиеся, часто считываемые данные.

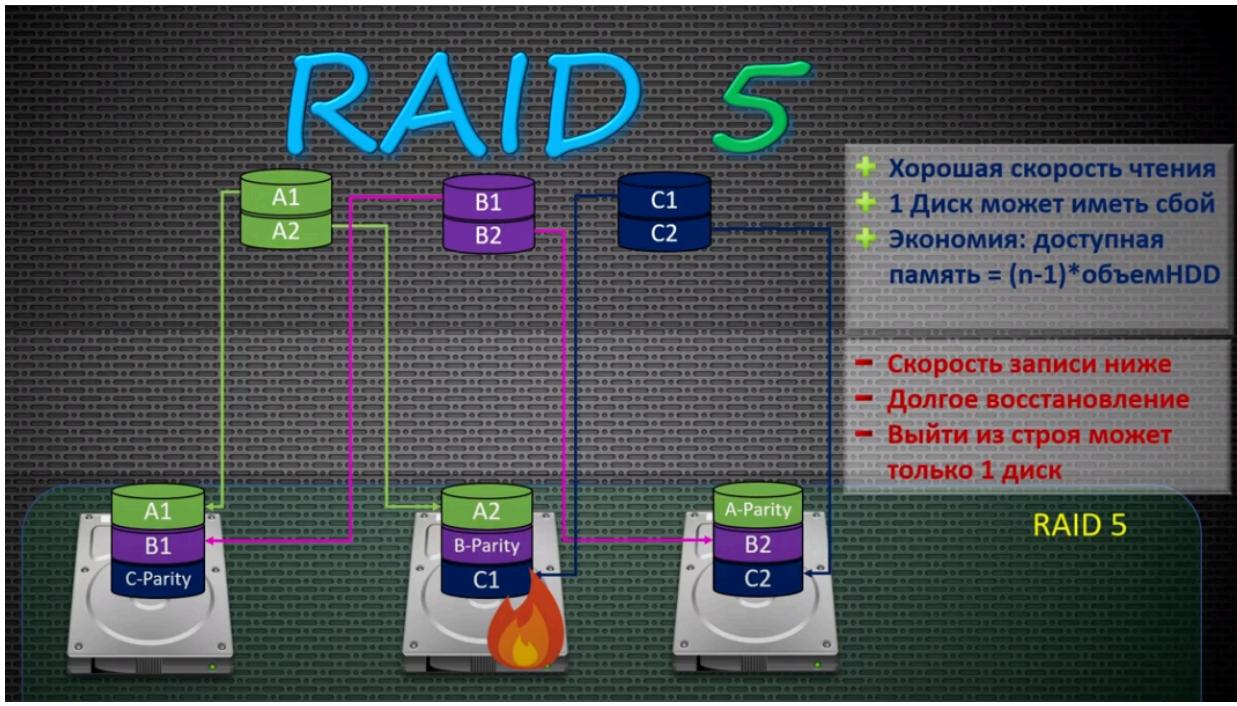
=====

Резюме:

- минимум три физических диска
- высокая пропускная способность
- медленный на операции записи
- восстанавливается при выходе из строя 1 диска (по контрольной сумме)

восстанавливаем я так пон)

RAID5: Чередование чётности (Striped parity)



Описание: В отличии от RAID4 данные и четность чередуются по всем дискам массива. Очень хорошо иметь дополнительный вакантный диск (hot spare disk) на случай если один из дисков массива выйдет из строя. Тогда контроллер подхватит вакантный диск и массив будет перестроен. Минимально три диска в массиве

Производительность: Лучше, чем в RAID4 т.к. решена проблема выделенного диска четности

Плюсы: Достигнут баланс чтения/записи/резервирования

Минусы: Просадка производительности во время перестройки массива. Если не используется кеш записи (рейд-контроллер не оборудован батарейкой и не настроен), то просадка будет особенно чувствительна

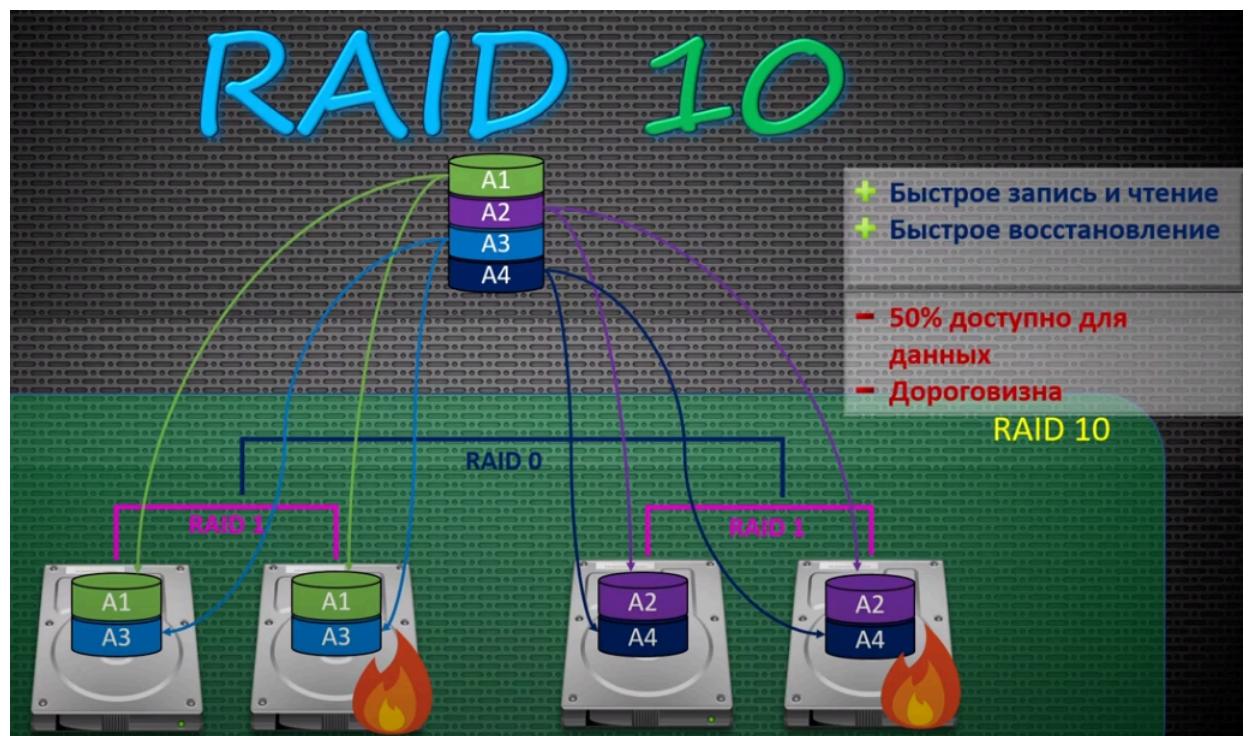
Использование: Веб-сервера, файловые сервера где используется интенсивное чтение данных.

=====

Резюме:

- хорошая пропускная способность и производительность
- восстанавливается при потере 1 диска
- распределенный контроль четности
- низкая скорость записи и чтения
- длительное восстановление
-

RAID10



Пока лень писать, но на картинке норм ахаха (не хаха)

Типы моделей

Типы моделей

- Физическая модель
 - **Как** представлены данные
 - Императивное описание
- Концептуальная модель
 - **Какие** данные
 - Декларативное описание

Физическая модель, концептуальная модель

Физическое и логическое проектирование

ОО Очень хорошо написано про [проектирование](#) БД (физическое, логическое, концептуальное).
Вот [здесь](#) кайфово написано.

Этапы физического проектирования:

Этапы физического проектирования баз данных [6]:

1. Перенос логической модели данных в среду целевой СУБД.
2. Проектирование основных отношений.
3. Разработка способов получения производных данных.
4. Реализация ограничений предметной области.
5. Проектирование физического представления базы данных.
6. Анализ транзакций.
7. Выбор файловой структуры.
8. Определение индексов.
9. Определение требований к дисковой памяти.
10. Проектирование пользовательских представлений.
11. Разработка механизмов защиты.
12. Обоснование необходимости введения контролируемой избыточности.
13. Текущий контроль и настройка операционной системы.

Индекс

Одним из способов внесения отношения порядка в значения колонок без нарушения физического расположения строк таблицы является создание объекта реляционной СУБД — индекса (index). Индекс — это объект в реляционной БД, который предназначен для организации быстрого доступа к строкам таблицы по значениям одной или более колонок этих строк.

Чем данные отличаются от информации?

Данные - это совокупность сведений (информация), представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека или автоматическими средствами.

Информация – любой вид знаний о предметах, фактах, понятиях и т. д. проблемной области, которыми обмениваются пользователи информационной системы.

Информация - это результат преобразования и анализа данных.

Отличие информации от данных состоит в том, что

- данные - это фиксированные сведения о событиях и явлениях, которые хранятся на определенных носителях,
- информация появляется в результате обработки данных при решении конкретных задач. Например, в базах данных хранятся различные данные, а по определенному запросу системы управления базой данных выдает требуемую информацию.

Распределённая бд

<http://www.kgau.ru/istiki/umk/ituman/textbox/bdras.htm>

БАЗА

Отношение

Основой структур данных в модели является теоретико-множественное понятие
ОТНОШЕНИЯ

Определение 1

- Пусть даны D_1, D_2, \dots, D_n – множества (домены).
- Декартово произведение $D_1 \times D_2 \times \dots \times D_n$ состоит из множества кортежей $\{v_1, v_2, \dots, v_n\}$, длины n , таких что $v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$
- Отношением называется КОНЕЧНОЕ подмножество декартового произведения доменов $D_1 \times D_2 \times \dots \times D_n$
- Запись вида $R(A_1:D_1, A_2:D_2, \dots, A_k:D_n)$ или $R(A_1, A_2, \dots, A_k)$ называют **схемой отношения**, где $k \geq n$, R – имя отношения, A_i – имена атрибутов, D_j – имена доменов
- Все атрибуты A_i в схеме отношения различны.

Определение 2

- Пусть A_1, A_2, \dots, A_n – имена атрибутов, D_1, D_2, \dots, D_m – имена доменов.
- Конечное множество пар вида $\{A_t, D_s\}$ называется схемой отношения, если $A_t \neq A_s$, при $t \neq s$ где $t, s \in \{1, 2, \dots, n\}$. Имена доменов для разных атрибутов могут совпадать.
- Множество пар $\{A_i, d_j\}$ называется кортежем отношения. где A_i – имя атрибута из A_1, A_2, \dots, A_n , а d_j из D_j
- Множество кортежей, соответствующих одной схеме отношения называется отношением
- Число атрибутов в отношении – **степень отношения**
- Число кортежей в отношении – **кардинальное число**

Примеры. Пусть $n=2$, $D_1=\{0,1\}$, $D_2=\{a,b,c\}$.

Тогда отношение $R(A_1, A_2)$ есть $\{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$

Фундаментальные свойства отношений

- Отсутствие кортежей дубликатов
- Отсутствие упорядоченности кортежей

- Отсутствие упорядоченности атрибутов
- Атомарность значений атрибутов

Атрибут

Атрибут – элементарное данное, описывающее свойство сущности. В записи данных представлен типом элемента данных и может использоваться в качестве первичного ключа (элемента данных, который однозначно идентифицирует запись), вторичного ключа (неоднозначно идентифицирует запись) или их составного элемента. Атрибут функционально зависит от группы других атрибутов, если его значение однозначно определяется совокупностью значений атрибутов этой группы. Существуют понятия первичного атрибута – атрибута, который входит в состав некоторого ключа, составляя весь первичный ключ или его часть, и вторичного атрибута – атрибута, используемого для индексирования записей в составе вторичного ключа.

Кортеж

Кортеж – элемент прямого (декартова) произведения множеств. В отличие от вектора компоненты кортежа на обязательно числа. Ими могут быть также ранги, символы, имена и т.п. В модели данных реляционной кортеж представляет собой строку таблицы (отношения).

Домен

Домен (от франц. *domaine* – владение) 1) область значений некоторого данного; 2) область значений атрибута в модели данных реляционной.

Функциональная зависимость

Функциональная зависимость (ФЗ)

- $X \rightarrow Y$
 - Каждому X соответствует ровно один Y
 - X и Y – множества атрибутов
- Все атрибуты функционально зависят от (над)ключа
- Функциональные зависимости нельзя «вычислить»

Функциональные зависимости

Пусть $R(A_1, A_2, \dots, A_n)$ - отношение,

X, Y являются подмножествами $\{A_1, A_2, \dots, A_n\}$

Определение

Множество атрибутов Y функционально зависит от X на R ,
 $(X \rightarrow Y)$

если в отношении R не могут содержаться

2 кортежа, компоненты которых

- ❖ совпадают по всем атрибутам из X ,
- ❖ но не совпадают по одному или более атрибутам из Y

То есть $\neg \exists r_1, r_2 (r_1, r_2 \in R, \pi_X(r_1) = \pi_X(r_2) \ \& \ \pi_Y(r_1) \neq \pi_Y(r_2))$ или

$$\forall r_1, r_2 (r_1, r_2 \in R, \pi_X(r_1) = \pi_X(r_2) \Rightarrow \pi_Y(r_1) = \pi_Y(r_2))$$

Функциональные зависимости отражают

- а) зависимость неключевых атрибутов от КЛЮЧА,
- б) отношение N:1 разных групп объектов

Этапы разработки БД

Этапы разработки базы данных в информационной системе

- Проектирование базы данных
 - Логическое проектирование
 - Физическое проектирование
- Планирование базы данных
 - Аппаратных средств
 - Программных средств
- Реализация базы данных
- Сопровождение
- Рейнжиниринг

Принципы проектирования ERD

Принципы проектирования

- Достоверность
 - ❖ Привлечение экспертов
 - ❖ Детальное изучение предметной области
- Отсутствие избыточности
- Простота
 - ❖ Включайте в проект только те структурные элементы, без которых нельзя обойтись
- Выбор подходящих связей
 - ❖ Множества сущностей можно соединить разными связями. Если брать любые – возможна избыточность.
- Использование элементов адекватных типов

Этапы разработки ERD

Этапы разработки ER модели

- Разработка локальных представлений (частей информационной системы)
 - ❖ Формулирование сущностей (имена, содержание)
 - ❖ Выбор идентифицирующего атрибута
 - ❖ Спецификация связей (имена, типы)
 - ❖ Добавление описательных атрибутов
- Объединение представлений пользователей
 - ❖ Идентичность
 - ❖ Агрегация
 - ❖ Обобщение

Первичный ключ

Ключ

- Атрибут или множество атрибутов, **однозначно определяющих кортеж в отношении**, называется **ключом отношения**.
 - В таблице Студент ключ - № студ. Билета
- Иногда **идентификатор строки** или **порядковое число** используются как **ключи** для идентификации строки в таблице
 - Такой ключ называют **искусственным или суррогатным** ключ

Внешний ключ

Неформально:

Неформально выражаясь, **внешний ключ** представляет собой **подмножество атрибутов** некоторой **переменной отношения** R_2 , значения которых должны совпадать со значениями некоторого **потенциального ключа** некоторой переменной отношения R_1 .

Формально:

Формальное определение. Пусть R_1 и R_2 — две переменные отношения, не обязательно различные. Внешним ключом FK в R_2 является **подмножество** атрибутов переменной R_2 такое, что выполняются следующие требования:

1. В переменной отношения R_1 имеется потенциальный ключ CK такой, что FK и CK совпадают с точностью до переименования атрибутов (то есть переименованием некоторого подмножества атрибутов FK можно получить такое подмножество атрибутов FK' , что FK' и CK совпадают как по именам, так и по типам атрибутов).
2. В любой момент времени каждое значение FK в текущем значении R_2 идентично значению CK в некотором кортеже в текущем значении R_1 . Иными словами, в каждый момент времени множество всех значений FK в R_2 является (нестрогим) подмножеством значений CK в R_1 .

Внешним ключом поддерживается ссылочная целостность.

Взято [отсюда](#)

UPDATE/DELETE CASCADE/RESTRICT/SET NULL

Вопрос 1. Какие действия должны предприниматься для поддержания ссылочной целостности?

Для ответа на этот вопрос используются переключатели **Update Action** и **Delete Action**. Первый из них, **Update Action**, определяет действие, выполняемое при модификации значений полей родительской таблицы, на которые ссылаются поля внешнего ключа дочерней таблицы. Назовем эти поля родительской таблицы адресуемыми полями. Второй переключатель **Delete Action** предназначен для выбора действия, выполняемого при удалении записей родительской таблицы.

Положения переключателя **Update Action** регламентируют выполнение следующих действий при модификации адресуемых полей родительской таблицы:

- **Restrict Update** - запрещение модификацию адресуемых полей;
- **Cascade** - (каскадное обновление) изменение значений полей внешнего ключа в записях, ссылающихся на модифицируемую запись родительской таблицы, на новое значение адресуемых полей;
- **Set NULL**- изменение значений полей внешнего ключа в записях, ссылающихся на модифицируемую запись родительской таблицы, на значение **NULL**;
- **Set Default** - изменение значений полей внешнего ключа в записях, ссылающихся на модифицируемую запись родительской таблицы, на значение присваиваемое полю по умолчанию.

Для указания действий, которые надлежит выполнить при удалении записей родительской таблицы используется переключатель **Delete Action**. Он следующие положения:

- **Restrict Delete**- запрещение удаления;
- **Cascade** - (каскадное удаление) удаление записей дочерней таблицы, ссылающихся на удаляемую запись родительской таблицы;
- **Set NULL**- изменение значений полей внешнего ключа в записях, ссылающихся на удаляемую запись родительской таблицы, на значение **NULL**;
- **Set Default** - изменение значений полей внешнего ключа в записях, ссылающихся на удаляемую запись родительской таблицы, на значение присваиваемое полю по умолчанию.

Вторичный ключ

Ограничения целостности, которые могут использоваться в определении столбца

- CHECK ()
- NOT NULL
- REFERENCES ...
- UNIQUE

Фраза FROM

Использует

- Операцию ОБЪЕДИНЕНИЯ ОТНОШЕНИЙ
- Операцию ПЕРЕСЕЧЕНИЯ ОТНОШЕНИЙ
- Подзапросы
- Имя таблицы

Свойства триггера

- действие может выполняться после события
- событием является операция INSERT
- действие может выполняться до события
- "срабатывает" только при наступлении определенного события
- определяется на таблицах
- перед выполнением операции может проверить условие выполнения

ТРИГГЕРЫ

- Триггеры еще называют **правилами**
"событие — условие — действие"
event — condition — action rules или ECA rules
- Отличаются от других ограничений, рассмотренных выше, в трех аспектах.
 - ❖ Триггер "срабатывает" только при наступлении определенного события (event), описание которого содержится в тексте триггера.
 - ❖ Вместо того чтобы сразу отвергнуть операцию, вызвавшую его срабатывание, триггер выполняет проверку условия (condition). Если результат проверки равен значению FALSE, работа триггера завершается.
 - ❖ Если же условие, сформулированное в тексте триггера, справедливо (TRUE) выполняет действие (action), поставленное в соответствие событию.

События, действия

- К числу **допустимых событий (Event)** обычно относят операции INSERT, DELETE и UPDATE **указанного отношения**. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.
- Под **действием (Action)** понимается последовательность операций с базой данных, отвечающая логике определенного события. Оформляется как программный блок, хранящийся в базе данных
- Триггеры определяются **ТОЛЬКО на таблицах, не на VIEW**
- Тем не менее триггеры срабатывают, когда операции модификации исполняются из под VIEW

Ограничения секции "действия"

- В секции "действия" текста триггера разрешено использовать далеко не все команды SQL.
- Например, нельзя применять команды
 - управления транзакциями (подобные COMMIT и ROLLBACK),
 - организации соединения между клиентом и сервером базы данных (скажем, CONNECT и DISCONNECT),
 - создания (CREATE), изменения (ALTER) и удаления (DROP) элементов схемы,
 - определения параметров сеанса (SET) и пр

Условие, время выполнения

- Действие может выполняться **либо до, либо после события** триггера.
- При выполнении действия триггер способен ссылаться как на прежние (old), так и на новые (new) значения компонентов кортежа, операция модификации которого рассматривается как событие, повлекшее срабатывание триггера.
- События обновления данных могут быть ограничены операциями над отдельным компонентом или множеством компонентов.
- Условие может задаваться с помощью предложения WHEN; действие выполняется при наступлении события и только в том случае, если в этот момент условие оставалось справедливым.
- Действие может выполняться в одном из двух возможных режимов:
 - уровень кортежа: один раз для каждого модифицируемого кортежа;
 - уровень команды: единожды для всех кортежей, которые подверглись изменению при выполнении одной операции.

События в триггере

К числу **допустимых событий (Event)** обычно относят операции **INSERT, DELETE и UPDATE** указанного **отношения**. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.

Транзакция

Транзакция – единица работы в СУБД. Формируется так, чтобы, начав работать с целостной БД, оставить ее после своего завершения также целостной. Указанное свойство обеспечивается правильным программированием транзакций программистом, а также системой управления транзакциями, обеспечивающей атомарность транзакции, т.е. либо доведение транзакции до завершения, либо аннулирование всех действий начавшейся транзакции. Последнее необходимо для повышения отказоустойчивости информационной вычислительной системы.

Транзакция – это последовательность операторов SQL, которая принимается или отменяется как единое целое.

Свойства транзакции ACID

Свойства транзакции

■ Автономность (Atomicity). Все или ничего!

Либо фиксируются все операции транзакции (commit), либо откат (rollback)

■ Согласованность (Consistency). Шире понятия целостности.

Согласованность может быть реализована на уровне бизнес логики. Например, операция « списание » должна соответствовать некоторой операции « зачисление »

■ Изолированность (Isolation).

При выполнении транзакций параллельно **одни** транзакции не должны оказывать влияние на результат выполнения **других** транзакций.

■ Долговечность (Durability).

Если пользователь получил подтверждение от системы, что транзакция выполнена, он может быть уверен, что сделанные им изменения не будут отменены из-за какого-либо сбоя.

Хранимая процедура

Хранимые процедуры

- Хранимая процедура - это программа на процедурном расширении языка SQL, сохраняемая в базе данных в откомпилированном виде.
- Хранение процедур в **том же месте, где они исполняются**, обеспечивает уменьшение трафика по сети данных и повышает общую производительность системы.

Вызывается явно.

Решение проблемы бесконечных ожиданий транзакций

- Выполнять транзакции по очереди

Бесконечные ожидания

- Пусть T_1 , находится в ожидании блокировки элемента **A** и еще T_2 запрашивает блокировку **A**.
- Рассмотрим ситуацию, когда T_2 получает блокировку раньше T_1 .
- Далее, в то время когда установила T_2 блокировку **A**, еще T_3 запросила блокировку **A**.
- Если T_3 получил блокировку **A** сразу за разблокированием **A** тран. T_2 , то T_1 все еще будет находиться в ожидании блокировки.

И, таким образом, некоторые транзакции могут находиться в **бесконечном ожидании**

Решение – очередь транзакций

29

Решение проблемы тупиков

- Одновременная блокировка элементов БД, необходимых транзакции
- Упорядоченная блокировка элементов БД, необходимых транзакции
- Рестарт транзакции из цикла графа ожиданий

Тупики – решение конфликта (1)

■ Вариант 1 (предотвращение).

Одновременная блокировка всех необходимых для транзакции элементов БД

Транзакция T_1

Lock (A, B);

...

UnLock (B);

UnLock (A);

Транзакция T_2

Lock (B, A);

...

UnLock (A);

UnLock (B);

Тупики: варианты решения проблемы (2)

2. Ввести **ЛИНЕЙНОЕ упорядочение** элементов БД, требовать запрос блокировок делать в соответствии с ЭТИМ **порядком**

Пусть на множестве элементов БД введен **линейный** порядок :
 $A < B < C \dots < Z$

Тогда ТР1 и ТР2 должны блокировать эл.БД в этом порядке

TP1

Lock A;

Lock B;

...

UnLock A;

UnLock B;

TP2

Lock A;

Lock B;

...

UnLock B;

UnLock A;

Тупики – решение конфликта (3)

■ Вариант 3 (устранение).

- Строим граф ожиданий.
- Вершины графа – транзакции.
- Дуга из T_1 в T_2 ($T_1 \rightarrow T_2$) означает, что T_1 ожидает выполнения запроса на блокировку элемента БД, заблокированного в данный момент T_2
- Каждый цикл указывает тупик

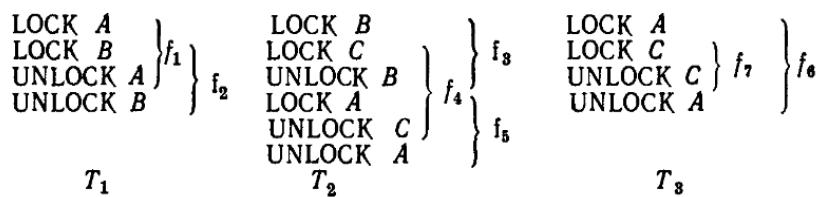
Анализируется построенный граф.

В случае обнаружения цикла **делается рестарт (Rollback)** одной из транзакций попавших в цикл.

Двухфазная транзакция

Двухфазные транзакции

- Транзакция называется **двуухфазной**, если все операции блокировки предшествуют всем операциям снятия блокировки



- **Теорема 2.** Любое расписание двухфазных транзакций – сериализуемо.

Сериализуемость транзакций

Множество транзакций называется сериализуемым, если результат выполнения расписания эквивалентен результату некоторого последовательного расписания этого множества транзакций.

Сериализуемость

Параллельное исполнение транзакций **КОРРЕКТНО** т. и т. т., когда их **совместный результат будет тем же самым**, что и при исполнении этих транзакций **в некотором последовательном порядке**.

- **Расписание** совокупности транзакций – **порядок** в котором выполняются **ЭЛЕМЕНТАРНЫЕ** шаги этих транзакций (блокировка, чтение, и т.д.).
- Расписание называется **последовательным**, если все шаги каждой транзакции выполняются вслед или перед всеми шагами других транзакций.
- Расписание называется **сериализуемым**, если его результат **ЭКВИВАЛЕНТЕН** результату некоторого последовательного расписания.

Проблемы параллельного исполнения транзакций

Типовые проблемы параллелизма

- Независимое обновление одной и той же таблицы.
Продажа без блокировки таблицы товаров.
- Отмена изменений уже после их использования
- Влияние частичного (незафиксированного) результата одной операции на результат другой операции
- Возможны «тупики» вследствие мешающих друг другу действий пользователей

Стандартные термины для проблем параллелизма (документация СУБД)

- Потерянное обновление (Lost UPDATE)
- Преждевременное чтение (Dirty READ)
- Неповторяющееся чтение (Non-repeatable READ)
- Фантомные вставки (Phantom INSERT)

14

Элемент блокировки

- Физическая страница базы данных
- Стока таблицы базы данных

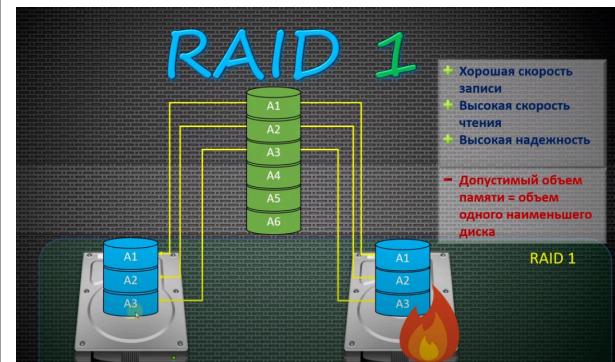
- Таблица базы данных
- Элемент строки таблицы базы данных

ЗАПОМНИ

Про RAID-ы

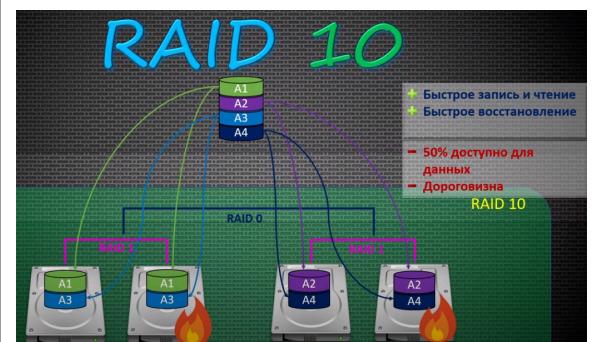
RAID1

- Для каждой операции записи нового блока нужно 2 обращения к дискам
- Минимальное количество дисков = 2
- Нет диска четности
- Массив восстанавливается при потере 1 диска



RAID1+0

- Для каждой операции записи нового блока нужно 4 обращения к дискам
- Массив восстанавливается при потере 1 диска
- Массив восстанавливается при потере 2 дисков (из одной половины зеркала)
- Нет диска четности
- Минимальное количество дисков = 4



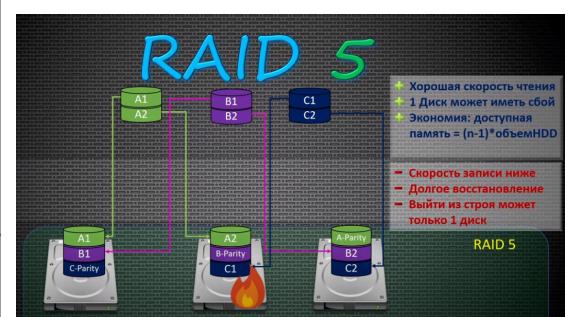
НЕ СООТВЕТСТВУЮТ

RAID5

- Имеется выделенный диск четности
- Массив восстанавливается при потере 2 дисков
- Для каждой операции записи нового блока нужно 2 обращения к дискам
- Минимальное количество дисков = 4
- Минимальное количество дисков = 2

Соответствуют RAID5:

- Минимальное количество дисков = 3
- Массив восстанавливается при потере 1 диска
- Блоки четности распределены по дискам
- Для каждой операции записи нового блока нужно 4 обращения к дискам



Базовые понятия модели внешней памяти

- файл
- блок
- запись
- поле

- ключ

Названия функций СУБД

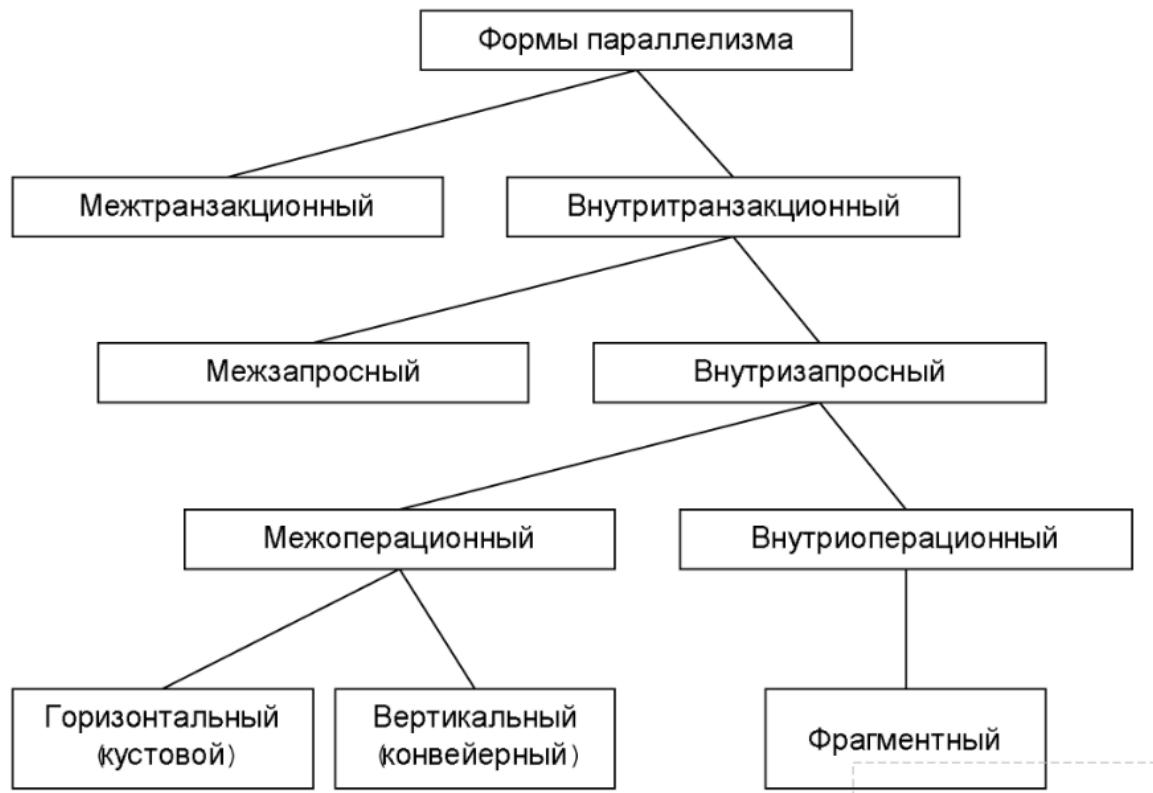
- Обеспечение одновременного доступа
- Определение структуры базы данных
- Манипулирование базой данных
- Предотвращение несанкционированного доступа

Способы управления КЭШ памятью

- Замещение последних использованных страниц
- Замещение давно неиспользованных страниц
- Фоновый сброс
- Сброс на верхнем уровне
- Принудительный сброс

Формы параллелизма

Формы параллелизма транзакций



Этапы оптимизации запросов

Стадии оптимизации

1. Преобразование запроса во внутреннюю форму
2. Преобразование в каноническую форму
3. Выбор потенциальных низкоуровневых процедур
4. Генерация планов вычисления запроса и выбор плана с наименьшими затратами

Стратегии оптимизации запросов

- Сортировка файлов перед операцией соединения
- Индексирование файлов перед операцией соединения
- Выполнять операции селекции по возможности раньше
- Собирать в каскады селекции и проекции

Типовые проблемы параллелизма

- Потерянное обновление
- Преждевременное чтение
- Фантомные вставки
- Неповторяющееся чтение

Этапы разработки ERD

- Спецификация связей
- Идентичность
- Агрегация
- Обобщение

Элементы блокировки

- Физическая страница базы данных
- Таблица базы данных
- Элемент строки таблицы базы данных
- Стока таблицы базы данных

Утверждения про триггер

- Триггер может выполняться над каждой строчкой таблицы
- Триггер срабатывает по событию
- Триггер срабатывает до или после события
- Хранимая процедура вызывается явно

Решения проблемы тупиков

- Одновременная блокировка элементов БД, необходимых транзакции
- Упорядоченная блокировка элементов БД, необходимых транзакции

- Рестарт транзакции из цикла графа ожиданий

ТЕСТЫ

ТЕСТ1

Вопрос 1

Вопрос: Пусть даны отношения R (A: int, B:char(30), C :float) и S (A: int, B:char(30)). Укажите, какие операции реляционной алгебры можно сделать над этими отношениями

- Декартово произведение
- Объединение
- Пересечение
- Натуральное соединение
- Деление
- Разность

Ответ

- Декартово произведение
- Натуральное соединение
- Деление

Пояснение

Преза 3

Объединение, пересечение и разность можно делать только если количество атрибутов одинаковое. И если соответствующие атрибуты определены на одних и тех же доменах.

Вопрос 2

Отметьте пункты, **не относящиеся** к фундаментальным свойствам отношений.
Выберите один или несколько ответов:

- Атомарность значений атрибутов
- Упорядоченность кортежей
- Отсутствие кортежей дубликатов
- Упорядоченность атрибутов
- Отсутствие дубликатов кортежей

Ответ

- Упорядоченность кортежей
- Упорядоченность атрибутов

Пояснение

преза 3, сл. 12

Фундаментальные свойства отношений:

- Отсутствие кортежей дубликатов
- Отсутствие упорядоченности кортежей
- Отсутствие упорядоченности атрибутов
- Атомарность значений атрибутов

Вопрос 3

Словарь данных – это:

- централизованное хранилище языковых средств БД
- централизованное хранилище данных
- централизованное хранилище полей БД
- централизованное хранилище метаинформации

Ответ

- централизованное хранилище метаинформации

Пояснение:

Метаданные — информация о другой информации, или данные, относящиеся к дополнительной информации о содержимом или объекте. Метаданные раскрывают сведения о признаках и свойствах, характеризующих какие-либо сущности, позволяющие автоматически искать и управлять ими в больших информационных потоках.

През 2 слайд 13

Подход баз данных

- СУБД реализует подход к обработке и хранению данных – подход баз данных
- Самоописание систем баз данных
 - ❖ Каталог баз содержит описание конкретных баз данных (структур данных, типов и ограничений целостности и т.д.)
 - ❖ Описание называется **метаданными** или **словарем базы данных**
 - ❖ Посредством **метаданных** СУБД обеспечивает доступ приложений к БД

13

Можно [тут](#) еще почитать.

СТРУКТУРА СЛОВАРЯ ДАННЫХ

1. БАЗА ДАННЫХ – полное название базы и имя файла

Описание назначения и общего содержания БД, а также лиц, которые могут ею пользоваться. Список приложений, работающих с базой, информация о других БД, использующих данные из этой базы. Если имеются, то сюда же включаются диаграммы БД.

А. ОБЛАСТЬ ДАННЫХ – название группы, к которой принадлежат таблицы

Если таблицы классифицируются по группам, то включается описание каждой группы.

1. ТАБЛИЦА – таблицы, входящие в область данных

а) ДОСТУП – права пользователей на доступ к таблице

б) ЗАПИСЬ – общее определение элементов данных

(1) ПЕРВИЧНЫЙ КЛЮЧ – поле (поля) первичного ключа

(а) ИНДЕКС – описание индекса первичного ключа

(2) ВНЕШНИЕ КЛЮЧИ – внешние ключевые поля

(а) ИНДЕКС – индексы внешних ключей

Метаданные – вспомогательные данные, представляющие характеристики, размещение, режимы использования, семантику и т.п. сведения об основных данных, относящихся непосредственно к объектам и связям предметной области процесса. Фиксируются в описании схем баз данных, а также в форме поддерживающих их словарей-справочников. Примерами метаданных могут быть описания логических структур данных, типы и длины значений данных и др.

Вопрос 4

В реляционной модели данных непроцедурное манипулирование данными описывает _____

Ответ

- реляционное исчисление на кортежах

Вопрос 5

Продолжите утверждение

"Описание структуры единиц информации, хранящихся в БД, представляет собой ... "

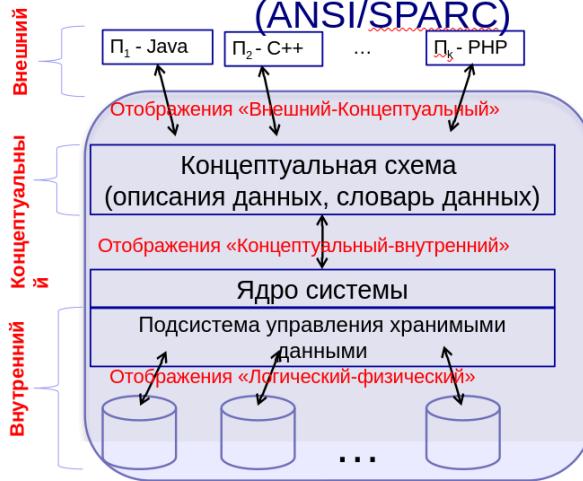
- модель .сущность . связь.
- логическую схему базы данных
- концептуальную схему
- ER-диаграмму

Ответ

- концептуальную схему

Пояснение
[Тут](#) почитать.

Три уровня архитектуры системы БД (ANSI/SPARC)



С
У
Б
Д

Типы моделей

- **Физическая модель**
 - Как представлены данные
 - Императивное описание
- **Концептуальная модель**
 - Какие данные
 - Декларативное описание

Дает Введение в системы баз данных. Глава 2

- Отображение "концептуальный-внутренний" устанавливает соответствие между концептуальным представлением и хранимой базой данных, т.е. описывает, как концептуальные записи и поля представлены на внутреннем уровне. При изменении структуры хранимой базы данных (т.е. при внесении изменений в определение структуры хранения) отображение "концептуальный—внутренний" также изменяется, с учетом того, что концептуальная схема остается неизменной. (Осуществление подобных изменений входит в обязанности администратора базы данных и может обеспечиваться самой СУБД.) Иначе говоря, чтобы обеспечивалась независимость от данных, результаты внесения любых изменений в схему хранения не должны обнаруживаться на концептуальном уровне.
- Отображение "внешний—концептуальный" определяет соответствие между некоторым внешним представлением и концептуальным представлением. В целом, различия, которые могут существовать между этими двумя уровнями, подобны различиям между концептуальным представлением и хранимой базой данных. Например, данные полей могут относиться к разным типам, названия полей и записей могут быть изменены, несколько концептуальных полей могут быть объединены в одно (виртуальное) внешнее поле и т.д. В одно и то же время допустимо существование любого количества внешних представлений, причем одно и то же внешнее представление может принадлежать нескольким пользователям, а разные внешние представления — перекрываться. Очевидно, что отображение "концептуальный—внутренний" служит основой физической независимости от данных, а отображения "внешний—концептуальный" являются ключом к логической независимости от данных. Как было показано в главе 1, система обеспечивает физическую независимость от данных [1.3], если пользователи и пользовательские программы обладают невосприимчивостью к изменениям в физической структуре хранимой базы данных. Аналогично, система обеспечивает логическую независимость от данных [1.4], если пользователи и пользовательские программы обладают невосприимчивостью к изменениям в логической структуре базы данных (подразумеваются изменения на концептуальном или "общем логическом" уровне). Этот важный вопрос будет обсуждаться в главах 3 и 10.
- Следует отметить, что большинство систем позволяет выражать одно определение внешнего представления через другое (по существу, с помощью отображения "внешний-внешний"), не требуя обязательного явного определения отображения Глава 2. Архитектура системы баз данных 85 каждого внешнего представления на концептуальный уровень. Эта возможность очень полезна, если несколько представлений подобны друг другу. В частности, аналогичная возможность предусмотрена во многих реляционных СУБД.

Вопрос 6

Отметьте предложения, содержащие название функций СУБД
Выберите один или несколько ответов:

- Обеспечение одновременного доступа
- Определение структуры базы данных
- Манипулирование базой данных
- Предотвращение несанкционированного доступа
- Логическое проектирование
- Физическое проектирование

Ответ

- Обеспечение одновременного доступа
- Определение структуры базы данных
- Манипулирование базой данных
- Предотвращение несанкционированного доступа

Презентация 2, сл.3-4

- **Определение конкретной базы данных:**
типов данных, структур данных, ограничений
- **Создание или Загрузка начального содержимого базы данных.**
- **Манипулирование базой данных:**
 - ❖ Модификация данных: вставка, удаление и обновление содержимого баз данных
 - ❖ Извлечение данных: поиск по запросам, создание отчетов
- **Управление многопользовательским доступом**
- **Управление внешней памятью**
- **Управление целостностью данных**
 - ❖ Физическая целостность
 - ❖ Логическая целостность
- **Управление защитой от несанкционированного доступа к базе данных**

Вопрос 7

Выберите варианты, **не соответствующие** этапам разработки базы данных в информационной системе

Выберите один или несколько ответов:

- Разработка структуры системы
- Реализация базы данных
- Проектирование базы данных
- Разработка UML диаграмм
- Планирование базы данных

Ответ

- Разработка структуры системы
- Разработка UML диаграмм

Пояснение

През 1, сл 13

Этапы разработки базы данных в информационной системе

- Проектирование базы данных
 - Логическое проектирование
 - Физическое проектирование
- Планирование базы данных
 - Аппаратных средств
 - Программных средств
- Реализация базы данных
- Сопровождение
- Рейнжиниринг

Вопрос 8

Отметьте термины, относящиеся к принципам проектирования ER модели.

Выберите один или несколько ответов:

- Достоверность
- Нормализация
- Отсутствие избыточности
- Выбор подходящих связей
- Детализация
- Простота

Ответ

- Достоверность
- Отсутствие избыточности
- Выбор подходящих связей
- Простота

Пояснение

Преза 1, сл 40

■ Принципы проектирования

- Достоверность
 - ❖ Привлечение экспертов
 - ❖ Детальное изучение предметной области
- Отсутствие избыточности
- Простота
 - ❖ Включайте в проект только те структурные элементы, без которых нельзя обойтись
- Выбор подходящих связей
 - ❖ Множества сущностей можно соединить разными связями. Если брать любые – возможна избыточность.
- Использование элементов адекватных типов

Вопрос 9

Определите тип связи между объектами `.Преподаватель.` и `.Дисциплина.`, если один преподаватель может вести занятия по нескольким дисциплинам.

- .один . к . одному.
- .многие . ко . многим.
- .один . ко . многим.
- .многие . к . одному.

Ответ

- .один . ко . многим.

Пояснение

Очевидно.

Вопрос 10

Выберите ответ, наиболее точно завершающий определение

"Первичный ключ (Primary key) в реляционной таблице - это:... "

Выберите один ответ:

- один или несколько столбцов, содержащих уникальные значения; используются для идентификации строк этой таблицы
- столбец таблицы, содержащий уникальные значения
- столбец или группа столбцов, используемая для организации ссылок на строки этой таблицы из других таблиц по Foreign Key
- совокупность столбцов, комбинация значений которых уникальна

Ответ

- один или несколько столбцов, содержащих уникальные значения; используются для идентификации строк этой таблицы

Пояснение

Преза 3, сл 10

Ключ

- Атрибут или множество атрибутов, **однозначно определяющих кортеж в отношении**, называется **ключом отношения**.
 - В таблице Студент ключ - № студ. Билета
- Иногда **идентификатор строки** или **порядковое число** используются как **ключи** для идентификации строки в таблице
 - Такой ключ называют **искусственным или суррогатным** ключ

Вопрос 11

Пусть отношение R содержит следующий набор кортежей

R (A,B,C,D)

2 3 5 6

3 4 7 8

2 3 5 7

3 6 7 8

Отметьте функциональные зависимости атрибутов в этом отношении

Выберите один или несколько ответов:

- B, C \rightarrow D
- A \rightarrow C
- A, C \rightarrow D
- A \rightarrow B
- A, B \rightarrow C

Ответ

- A \rightarrow C
- A, B \rightarrow C

Пояснение

- B, C \rightarrow D // одному и тому же набору должно соответствовать одно значение. А здесь (3,5 \rightarrow 6) и (3,5 \rightarrow 7) \rightarrow нет ФЗ
- A \rightarrow C // видна ФЗ, А-доминант, С-зависим
- A, C \rightarrow D // одному и тому же набору должно соответствовать одно значение. А здесь (2,5 \rightarrow 6) и (2,5 \rightarrow 7) \rightarrow нет ФЗ
- A \rightarrow B // нет ФЗ, тк есть (34) и (36) \rightarrow нет здесь ФЗ
- A, B \rightarrow C // видна ФЗ, А,В-доминант, С-зависим

Вопрос 12

Какая операция должна восстанавливать исходное отношение после декомпозиции

Выберите один ответ:

- Декартово произведение
- Эквисоединение
- Натуральное соединение
- Объединение

Ответ

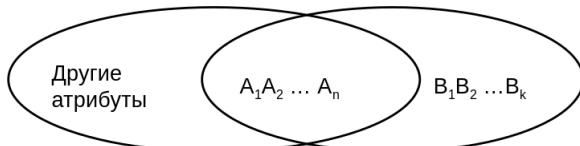
- Натуральное соединение

Пояснение

Преза 5 слайд 21

Приведение к нормальным формам

- Поиск в исходном отношении R нетривиальных функциональных зависимостей $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_k$, нарушающих нормальную форму.
- Разбиение отношения R на 2 отношения, например, T и S, таких, что



- T будет состоять из атрибутов, образующих нетривиальную ФЗ,
- S – из атрибутов отношения R, входящих в ключ и не вошедших в отношение T.
- Исходное отношение должно восстанавливаться из 2-ух получившихся отношений с помощью **НАТУРАЛЬНОГО СОЕДИНЕНИЯ**

Декомпозиция		
$R(A, B, C)$		
1 2 3		
3 2 6		
5 4 2		
$R_1(A, B)$		$R_2(B, C)$
1 2		2 3
3 2		2 6
5 4		4 2
Соединение		
$R'(A, B, C)$		
1 2 3		
I 2 6		
3 2 3		
3 2 6		
5 4 2		

Вопрос 13

Отметьте термины, используемые в описании ER модели

Выберите один или несколько ответов:

- множество элементов
- множество записей
- множество связей
- множество атрибутов
- множество сущностей

Ответ

- множество связей
- множество атрибутов
- множество сущностей

Пояснение

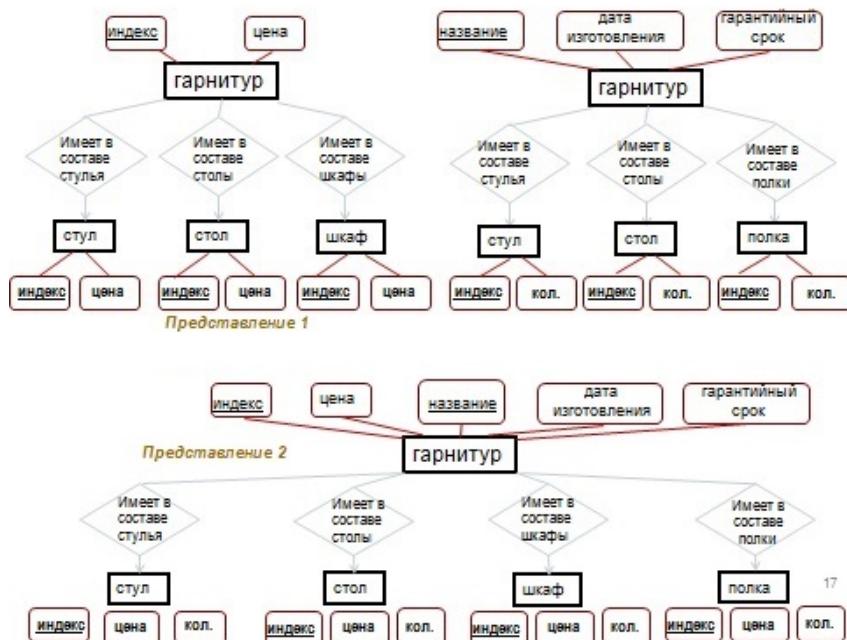
Преза 1, сл 19

Модель «Сущность-связь»

- Основные компоненты ER модели
 - ❖ Множество сущностей
 - ❖ Атрибуты сущностей
 - ❖ Связи
- Этапы проектирования ER модели
 - ❖ Выделение сущностей-объектов
 - Определение атрибутов сущностей
 - Выделение идентифицирующих атрибутов-ключей
 - ❖ Определение связей
 - Вид связи: 1:1, 1:m, n:m
 - Обязательность связей
 - ❖ Фиксация модели в виде ER-диаграммы

Вопрос 14

Какой операцией из представления 1 получено представление 2



Ответ

- агрегация

Пояснение

Преза 1, сл 36

Агрегация

- Агрегация – декартово произведение сущностей
- Новая сущность формируется на основе данных о частях объектах
 - Имя, Должность, Ном_отдела = **Сотрудник**
 - Имя, Паспорт, Дата приема = **Сотрудник**
 - Имя, Номер-страхового полиса, Адрес = **Сотрудник**

Имя, Номер-страхового полиса, Паспорт, Должность,
Ном_отдела, Дата приема, Адрес = **СОТРУДНИК**

Вопрос 15

Выберите правильный вариант декомпозиции отношения R (A, B, C, D, E) при приведении его к 3-ей нормальной форме

A, B - ключ отношения

Имеются функциональные зависимости:

B → C

D → E

Выберите один ответ:

- R (A, B, C, D, E) разбивается на R1 (A, B, C), R2 (C, D, E)
- R (A, B, C, D, E) разбивается на R1 (B, C), R2 (A, B, D) и R3 (D, E)
- R (A, B, C, D, E) разбивается на R1 (A, B), R2 (C, D) и R3 (D, E)

Ответ

- R (A, B, C, D, E) разбивается на R1 (B, C), R2 (A, B, D) и R3 (D, E)

Пояснение

Здесь функциональная зависимость между B и C (где B - детерминант отношения, C - зависимая часть) и D и E (где D - детерминант отношения, E - зависимая часть).

Отношения находятся в 3й НФ, если

1. Табл во 2 НФ
2. Все атрибуты зависят только от первичного ключа, а не от других атрибутов.

A	B	C	D	E
b1	c1			
b2	c2			
b3	c1			
		d1	e1	
		d2	e1	

Получается, что тк есть зависимость между B & C , D & E, то что надо сделать с таблицей? Декомпозировать. А как?

Выделяем отдельно таблицу с B&C (тк есть функц зав), D&E (тк есть функц зав), остается A&B&D (A&B был первичным ключом - им и остается, + добавляется D, тк в изначальной таблице D был. ДОПИСАТЬ МБ ЧЕТ ЕЩЕ НАДО?)

Вопрос 16

Верно ли, что СУБД обеспечивает независимость данных на нижнем (физическом) уровне ?

Выберите один ответ:

- Верно
- Неверно

Ответ

- Верно

Пояснение

Преза 1, сл 21



Независимость данных

■ Независимость данных от программ

позволяет изменять структуру данных и организацию данных во внешней памяти без изменения текста программ. Требуется перекомпиляция.

■ Реализуется на 2-ух уровнях: логическом и физическом

■ Физический уровень: невосприимчивость приложений к изменениям в физическом представлении данных и в методах доступа к данным

- Изменение реализации на внутреннем уровне – последовательный, индексный, В-дерево

■ Логический уровень: отсутствие влияния изменений в логической структуре базы данных на работу пользователей и пользовательских программ

- Изменение состава полей в таблице БД,
- Изменение типов данных,
- Формирование представлений (виртуальных таблиц)

Вопрос 17

Выделите ограничения целостности, которые могут использоваться в определении столбца

Выберите один или несколько ответов:

- a. FOREIGN KEY (столбец1, столбец2, ..)
- b. CHECK ()
- c. NOT NULL
- d. REFERENCES ...
- e. PRIMARY KEY (столбец1, столбец2, ...)
- f. UNIQUE

Ответ

- CHECK ()
- NOT NULL
- REFERENCES ...
- UNIQUE

Пояснение

Преза 6, сл 20

Ограничение столбца

<ограничение столбца> ::=

```
[CONSTRAINT имя ограничения ]  
| {NOT NULL}  
| { PRIMARY KEY }  
| {UNIQUE}  
| REFERENCES имя таблицы [ (имя столбца) ]  
    [<ссылочная спецификация>]  
| {CHECK предикат}  
[ [ INITIALLY DEFERRED ] | INITIALLY IMMEDIATE]  
[[NOT] DEFERRABLE ]
```

Вопрос 18

В какой наибольшей нормальной форме находится отношение R (A, B, C, D, E), если A, B - ключ отношения и

имеются следующие функциональные зависимости:

A, B → C

A, B → D

A, B → E

Выберите один ответ:

- в первой НФ
- в третьей НФ
- в нормальной форме Бойса-Кодда
- во второй НФ

Ответ

- в нормальной форме Бойса-Кодда

Пояснение. По опр.

НФБК:

- таблица в 3 НФ – да
- ключевые атрибуты не зависят от неключевых – да

Вопрос 19

В какой нормальной форме находится отношение R (A, B, C, D, E),

если A, B - ключ отношения

и имеется функциональная зависимость

$B \rightarrow D$

Выберите один ответ:

- в BCNF
- во второй НФ
- в первой НФ
- в третьей НФ

Ответ

- в первой НФ

Пояснение

По определению =))

1 НФ:

- в каждой клетке по 1 значению - да
- нет повторяющихся строк - да

2 НФ:

- табл в 1 НФ - да
- есть первичный ключ - да
- каждый атрибут зависит от первичного ключа ЦЕЛИКОМ, а не от какой-то его части - нет

Вопрос 20

В какой наибольшей нормальной форме находится отношение R (A, B, C, D, E),

если A, B - ключ отношения и

имеются следующие функциональные зависимости:

$C \rightarrow D$

$D \rightarrow E$

Выберите один ответ:

- в нормальной форме Бойса-Кодда
- в первой НФ
- во второй НФ
- в третьей НФ

Ответ

- во второй НФ

Пояснение

По определению

1 НФ:

- в каждой клетке по 1 значению - да
- нет повторяющихся строк - да

2 НФ:

- табл в 1 НФ - да
- есть первичный ключ - да
- каждый атрибут зависит от первичного ключа ЦЕЛИКОМ, а не от какой-то его части - да

3 НФ:

- табл во 2й НФ - да
- все атрибуты зависят только от первичного ключа, но не от других атрибутов - нет (тк тут зависимости С → D, D → E - зависимость между атрибутами не состоящих в первичном ключе)

Вопрос 21

Для разработки и эксплуатации баз данных используются:

Выберите один ответ:

- системы управления данных
- системы программирования
- системы управления базами данных
- системы автоматизированного проектирования

Ответ

- системы управления базами данных

Пояснение

Преза 1, сл 10 хоть чот

Система управления базой данных (СУБД-Database Management System)

❖ Пакет программ, обеспечивающих средства создания и манипулирования базой данных на ЭВМ

Вопрос 22

Продолжите определение

Отношением называется

- собственное подмножество декартового произведения доменов
- подмножество декартового произведения конечных доменов
- конечное подмножество декартового произведения доменов
- подмножество декартового произведения доменов

- Конечное подмножество декартового произведения конечных доменов

Ответ

- конечное подмножество декартового произведения доменов

Пояснение

Преза 3, сл 4

Отношением называется КОНЕЧНОЕ подмножество декартового произведения доменов $D_1 \times D_2 \times \dots \times D_n$

Вопрос 23

СУБД – это:

Выберите один ответ:

- специальный программный комплекс для обеспечения доступа к данным и управления ими
- система средств администрирования банка данных
- система средств архивирования и резервного копирования банка данных
- система средств управления транзакциями

Ответ

- специальный программный комплекс для обеспечения доступа к данным и управления ими

Преза 1, сл 10 хоть чот

Система управления базой данных (СУБД-Database Management System)

❖ Пакет программ, обеспечивающих средства создания и манипулирования базой данных на ЭВМ

Вопрос 24

Какой операцией схема 2 получена из схемы 1 ?

Схема 1

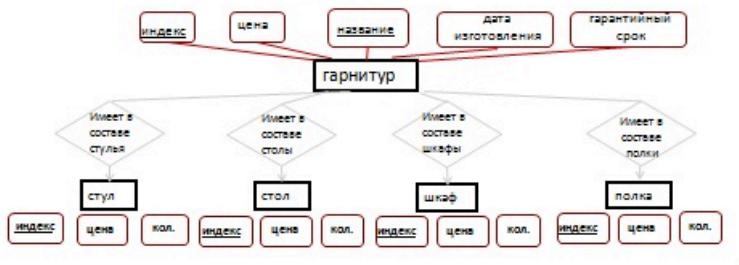


Схема 2



Ответ

- обобщение

Пояснение

Преза 1, сл 38

Обобщение

- Обобщение – это абстракция данных, позволяющая трактовать **класс объектов как ОДИН объект**
- При агрегации **части соединяются в целое**
- Обобщение фиксирует **РОДО-ВИДОВЫЕ отношения**
- **ВИД** есть **РОД** в совокупности с видовым отличием
- Ваз 2109, Ваз 2101, Ваз 2107 = **Автомобили марки ВАЗ**
- Автомобили марки ВАЗ, автомобили марки Хонда, ... = **Автомобили**
- Автомобиль, велосипед, мотоцикл, трактор = **Колесное средство передвижения**

Вопрос 25

Выберите продолжение для предложения
"Предметная область . это ..."

Выберите один ответ:

- ER-диаграмма, отражающая заданную область внешнего мира
- БД, разработанная для решения конкретной задачи
- модель .сущность . связь., отражающая заданную область внешнего мира
- часть реального мира, представляющая интерес для данного исследования

Ответ

- часть реального мира, представляющая интерес для данного исследования

Пояснение

Преза 1, сл 10

Предметная область

- ❖ Часть реального мира, для которой создается информационная система

Вопрос 26

Степень отношения . это:

Выберите один ответ:

- количество первичных ключей в таблице
- количество столбцов в таблице
- размер индекса по ключу таблицы
- количество строк в таблице
- количество значений в таблице

Ответ

- количество столбцов в таблице

Пояснение

Преза 3, сл 5

Число атрибутов в отношении – степень отношения

неформально, атрибут = столбец

Вопрос 27

Выделите термины, относящиеся к этапам разработки ER модели
Выберите один или несколько ответов:

- Моделирование
- Идентичность
- Агрегация
- Абстракция
- Спецификация связей
- Детализация
- Обобщение

Ответ

- Идентичность
- Агрегация
- Спецификация связей
- Обобщение

Пояснение

Преза 1, сл 32

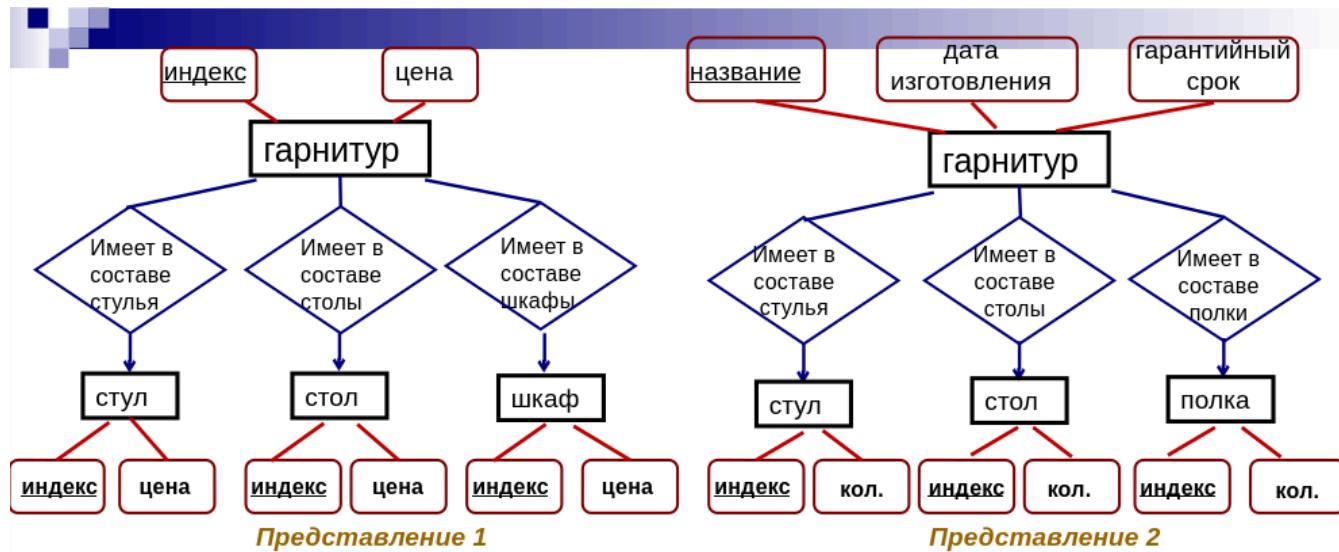
Этапы разработки ER модели

- Разработка локальных представлений (частей информационной системы)
 - ❖ Формулирование сущностей (имена, содержание)
 - ❖ Выбор идентифицирующего атрибута
 - ❖ Спецификация связей (имена, типы)
 - ❖ Добавление описательных атрибутов
- Объединение представлений пользователей
 - ❖ Идентичность
 - ❖ Агрегация
 - ❖ Обобщение

Преза 1, сл 34 - 39

Идентичность

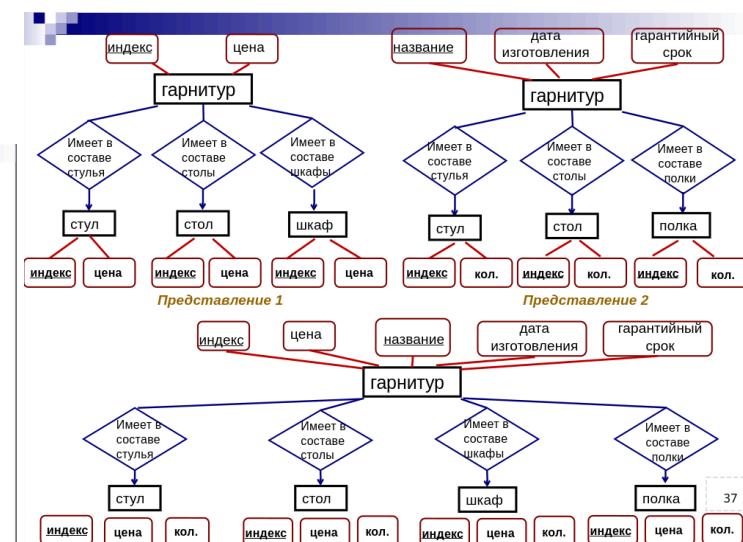
- Два или более элементов являются идентичными, если они имеют **одинаковое семантическое значение**
- В силу абстрагирования при выявлении сущностей идентичность элементов **устанавливается экспертом**
- Идентичность и подобие
- Персона, Служащий, Сотрудник, Работник, Персонал



Агрегация

- Агрегация – декартово произведение сущностей
- Новая сущность формируется на основе данных о частях объектах
 - Имя, Должность, Ном_отдела = **Сотрудник**
 - Имя, Паспорт, Дата приема = **Сотрудник**
 - Имя, Номер-страхового полиса, Адрес = **Сотрудник**

Имя, Номер-страхового полиса, Паспорт, Должность, Ном_отдела, Дата приема, Адрес = **СОТРУДНИК**



Итоговая схема

Обобщение

- Обобщение – это абстракция данных, позволяющая трактовать **класс объектов** как **ОДИН объект**
- При агрегации части **соединяются в целое**
- Обобщение фиксирует **РОДО-ВИДОВЫЕ отношения**
- **ВИД** есть **РОД** в совокупности с видовым отличием
 - Ваз 2109, Ваз 2101, Ваз 2107 = Автомобили марки ВАЗ
 - Автомобили марки ВАЗ, автомобили марки Хонда, ... = Автомобили
 - Автомобиль, велосипед, мотоцикл, трактор = Колесное средство передвижения



Вопрос 28

Пусть даны отношения

$R (A, B, C) S (B, D)$

1 3 4 2 1
2 5 7 3 7
4 2 2

Результатом какой операции является отношение T , если имеет вид

$T (A, B, C, D)$

1 3 4 7
4 2 2 1

Выберите один ответ:

- RIGHT JOIN соединение
- Натуральное соединение
- Деление
- Декартово произведение
- LEFT JOIN соединение

Ответ

- Натуральное соединение

Пояснение

Преза 3, сл 20

Натуральное соединение строится для отношений, имеющих одинаковые атрибуты. В R, S есть одинаковый атрибут B . Дальше дб понятно.

- **Натуральное соединение** (Natural JOIN, Inner JOIN). Для R и S имеющих одинаковые атрибуты
- $R \bowtie_f S = \pi_{\text{атр } R \cup \text{атр } S} (\sigma_f (R \times S))$, где f функция вида
- $$A_{r_1} = A_{s_1} \& A_{r_2} = A_{s_2} \& \dots \& A_{r_l} = A_{s_l},$$
- а $A_{r_i} = A_{s_i}$ – одинаковые атрибуты из R и S

Естественное соединение

- $R_1 \bowtie R_2$ – соединение по одноимённым атрибутам
- Заголовок
 - Объединение заголовков
- Тело
 - «Сплющенные» пары кортежей с равными одноимёнными атрибутами

Пример естественного соединения

- $R_1 \bowtie R_2$

Id	FirstName
1	Иван
2	Петр
3	Сидор

Id	LastName
1	Иванов
1	Петров
2	Сидоров



Id	FirstName	LastName
1	Иван	Иванов
1	Иван	Петров
2	Петр	Сидоров

Вопрос 29

В модели .сущность-связь. бинарной связью называется:

Выберите один ответ:

- связь .один-ко-многим.
- в модели .сущность-связь. нет такого термина
- связь, соединяющая две сущности
- связь, представленная в базе данных бинарным числом

Ответ

- связь, соединяющая две сущности

Пояснение

Преза 1, сл 23

Связи

- Связи – **соединения (соотношения)** между двумя или большим числом сущностей
- ER модель допускает наличие связей, включающей **произвольное количество** сущностей.
- Наиболее распространена **бинарная связь** между 2-мя множествами сущностей

Вопрос 30

Какая целостность данных реализуется внешним ключом?

Выберите один ответ:

- целостность доменов
- ссылочная целостность
- реляционная целостность
- целостность сущностей
- функциональная целостность

Ответ

- ссылочная целостность

Пояснение

Преза 3, сл 23

Целостность

- Целостность сущностей
 - ❖ Ключи
- Сылочная целостность
- Способы поддержания целостности:
 - ❖ Только сообщение о нарушении целостности
 - ❖ Метод значений NULL
 - ❖ Каскадный метод
 - ❖ Метод значений по умолчанию

Украла:

Почитать про целостность и виды: [тык](#)

Тоже хорошо написано: [тык](#)

Еще [тут](#) посмотреть

5. Определение требований поддержки целостности данных и их документирование. Эти требования представляют собой ограничения, которые вводятся с целью предотвратить помещение в базу данных противоречивых данных. На этом шаге вопросы целостности данных освещаются безотносительно к конкретным аспектам ее реализации. Должны быть рассмотрены следующие типы ограничений:

- обязательные данные. Выясняется, есть ли атрибуты, которые не могут иметь Null-значений;
- ограничения для значений атрибутов. Определяются допустимые значения для атрибутов;
- целостность сущностей. Она достигается, если первичный ключ сущности не содержит Null-значений;
- ссылочная целостность. Она понимается так, что значение внешнего ключа должно обязательно присутствовать в первичном ключе одной из строк таблицы для родительской сущности;
- ограничения, накладываемые бизнес-правилами. Например, в случае с проектом БАНК может быть принято правило, запрещающее клиенту распоряжаться, скажем, более чем тремя счетами.

Сведения обо всех установленных ограничениях целостности данных помещаются в словарь данных.

ТЕСТ2

Вопрос 1

Транзакция называется двухфазной, если _____

Выберите один ответ:

- а. все операции блокировки предшествуют всем операциям снятия блокировки
- б. операции блокировки предшествуют операциям снятия блокировки
- в. за всеми операциями блокировки следуют операции снятия блокировки
- г. все операции блокировки следуют за операциям снятия блокировки

Ответ

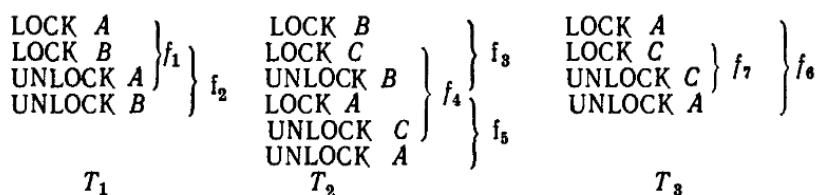
- все операции блокировки предшествуют всем операциям снятия блокировки

Пояснение

Преза 12 , сл 41

Двухфазные транзакции

- Транзакция называется **двуфазной**, если все операции блокировки предшествуют всем операциям снятия блокировки



- **Теорема 2.** Любое расписание двухфазных транзакций – сериализуемо.

Вопрос 2

Выделите термины, **не** относящиеся к этапам разработки ER модели

Выберите один или несколько ответов:

- Абстракция
- Обобщение
- Идентичность
- Детализация
- Агрегация
- Моделирование
- Спецификация связей

Ответ

- Абстракция
- Детализация
- Моделирование

Пояснение

Преза 1, сл 40

Этапы разработки ER модели

- Разработка локальных представлений (частей информационной системы)
 - ❖ Формулирование сущностей (имена, содержание)
 - ❖ Выбор идентифицирующего атрибута
 - ❖ Спецификация связей (имена, типы)
 - ❖ Добавление описательных атрибутов
- Объединение представлений пользователей
 - ❖ Идентичность
 - ❖ Агрегация
 - ❖ Обобщение

Вопрос 3

Отметьте свойства триггера

Выберите один или несколько ответов:

- действие может выполняться после события
- событием является операция SELECT
- событием является операция INSERT
- действие может выполнятся до события
- "срабатывает" только при наступлении определенного события
- определяется на таблицах
- перед выполнением операции может проверить условие выполнения
- определяется для VIEW

Ответ

- действие может выполняться после события
- событием является операция INSERT
- действие может выполнятся до события
- "срабатывает" только при наступлении определенного события
- определяется на таблицах
- перед выполнением операции может проверить условие выполнения

Пояснение

Преза 11, сл 13-16

Вот сначала просто всё по каждомуциальному ответу:

Действие может выполняться **либо до, либо после события** триггера.

К числу **допустимых событий (Event)** обычно относят операции **INSERT, DELETE и UPDATE** указанного **отношения**. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.

Триггер "срабатывает" только при наступлении определенного **события (event)**, описание которого содержится в тексте триггера.

■ Триггеры определяются **ТОЛЬКО на таблицах, не на VIEW**

Вместо того чтобы сразу отвергнуть операцию, вызвавшую его срабатывание, триггер выполняет проверку **условия (condition)**. Если результат проверки равен значению **FALSE**, работа триггера завершается.

Просто скрины слайдов:

ТРИГГЕРЫ

- Триггеры еще называют **правилами** "событие — условие — действие"
event — condition — action rules или ECA rules
- Отличаются от других ограничений, рассмотренных выше, в трех аспектах.
 - ❖ Триггер "срабатывает" только при наступлении определенного события (event), описание которого содержится в тексте триггера.
 - ❖ Вместо того чтобы сразу отвергнуть операцию, вызвавшую его срабатывание, триггер выполняет проверку условия (condition). Если результат проверки равен значению FALSE, работа триггера завершается.
 - ❖ Если же условие, сформулированное в тексте триггера, справедливо (TRUE) и выполняет действие (action), поставленное в соответствие событию.

Ограничения секции "действия"

- В секции "действия" текста триггера разрешено использовать далеко не все команды SQL.
- Например, нельзя применять команды
 - управления транзакциями (подобные COMMIT и ROLLBACK),
 - организации соединения между клиентом и сервером базы данных (скажем, CONNECT и DISCONNECT),
 - создания (CREATE), изменения (ALTER) и удаления (DROP) элементов схемы,
 - определения параметров сеанса (SET) и пр

События, действия

- К числу допустимых событий (Event) обычно относят операции INSERT, DELETE и UPDATE указанного отношения. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.
- Под действием (Action) понимается последовательность операций с базой данных, отвечающая логике определенного события. Оформляется как программный блок, хранящийся в базе данных
- Триггеры определяются ТОЛЬКО на таблицах, не на VIEW
- Тем не менее триггеры срабатывают, когда операции модификации исполняются из под VIEW

Условие, время выполнения

- Действие может выполняться либо до, либо после события триггера.
- При выполнении действия триггер способен ссылаться как на прежние (old), так и на новые (new) значения компонентов кортежа, операция модификации которого рассматривается как событие, повлекшее срабатывание триггера.
- События обновления данных могут быть ограничены операциями над отдельным компонентом или множеством компонентов.
- Условие может задаваться с помощью предложения WHEN; действие выполняется при наступлении события и только в том случае, если в этот момент условие оставалось справедливым.
- Действие может выполняться в одном из двух возможных режимов:
 - уровень кортежа: один раз для каждого модифицируемого кортежа;
 - уровень команды: единожды для всех кортежей, которые подверглись изменению при выполнении одной операции.

Вопрос 4

Продолжите предложение:

"В реляционной модели данных процедурное манипулирование данными описывает
„_____”

•

Ответ

- реляционная алгебра

Пояснение ПОТОМ ВСТАВИТЬ
Преза 1, сл

Вопрос 5

Пусть даны отношения R (A:float, B:char(20), C :float) и S (A:float, B:char(20))

Укажите, какие операции реляционной алгебры можно сделать над этими отношениями

Выберите один или несколько ответов:

- Объединение
- Деление
- Натуральное соединение
- Разность
- Пересечение
- Декартово произведение

Ответ

- Деление
- Натуральное соединение
- Декартово произведение

Пояснение

Преза 3

Объединение, пересечение и разность можно делать только если количество атрибутов одинаковое. И если соответствующие атрибуты определены на одних и тех же доменах.

Вопрос 6

Отметьте недопустимые варианты предложений для база данных, содержащей

R1		
ФИО	Дисциплина	Оценка
Петров Ф.И.	Базы данных	5
Сидоров К.А.	Базы данных	4
Миронов А.В.	Базы данных	2
Петров Ф.И.	Моделирование	5
Сидоров К.А.	Моделирование	4
Миронов А.В.	Моделирование	Null
Трофимов П.А.	Сети ЭВМ	4
Иванова Е.А.	Сети ЭВМ	5
Уткина Н.В.	Сети ЭВМ	5

R2	
ФИО	Группа
Петров Ф.И.	АИ21
Сидоров К.А.	АИ21
Миронов А.В.	АИ21
Трофимов П.А.	АИ22
Иванова Е.А.	АИ22
Уткина Н.В.	АИ22

таблицы

R3	
Группа	Дисциплина
АИ21	Базы данных
АИ21	Моделирование
АИ22	Сети ЭВМ

Выберите один или несколько ответов:

- SELECT COUNT(ФИО) FROM R1
- SELECT COUNT(*) FROM R1
- SELECT ФИО, ОЦЕНКА
FROM R1
WHERE ОЦЕНКА > AVG(ОЦЕНКА)

- SELECT MAX FROM R1
- SELECT MAX(12, 13)

Ответ

- SELECT ФИО, ОЦЕНКА
FROM R1
WHERE ОЦЕНКА > AVG(ОЦЕНКА)
- SELECT MAX FROM R1
- SELECT MAX(12, 13)

Пояснение:

С последними двумя и так понятно.

А вот с первым что не так?

Ключевое слово WHERE предназначено для работы с отдельными записями, и поэтому оно не подходит для **использования** с такими агрегирующими **функциями** как COUNT(), SUM() или AVG(), которые применяются ко всей выборке данных одновременно.

Вот [тут](#) можно почитать

Вопрос 7

Какая операция должна восстанавливать исходное отношение после декомпозиции

Выберите один ответ:

- Натуральное соединение
- Эквисоединение
- Декартово произведение
- Объединение

Ответ

- Натуральное соединение

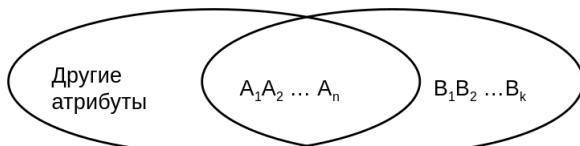
Пояснение

Пояснение

Преза 5 слайд 21

Приведение к нормальным формам

- Поиск в исходном отношении R нетривиальных функциональных зависимостей $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_k$, нарушающих нормальную форму.
- Разбиение отношения R на 2 отношения, например, T и S, таких, что



- T будет состоять из атрибутов, образующих нетривиальную ФЗ,
- S – из атрибутов отношения R, входящих в ключ и не вошедших в отношение T.
- Исходное отношение должно восстанавливаться из 2-ух получившихся отношений с помощью **НАТУРАЛЬНОГО СОЕДИНЕНИЯ**

Декомпозиция		
$R(A, B, C)$		
1 2 3		
3 2 6		
5 4 2		
$R_1(A, B)$	$R_2(B, C)$	
1 2	2 3	
3 2	2 6	
5 4	4 2	
Соединение		
$R'(A, B, C)$		
1 2 3		
I 2 6		
3 2 3		
3 2 6		
5 4 2		

Вопрос 8

Отметьте, что можно использовать во фразе FROM
Выберите один или несколько ответов:

- Операцию ПРОЕКЦИЯ
- Операцию ОБЪЕДИНЕНИЯ ОТНОШЕНИЙ
- Агрегатную функцию
- Операцию ПЕРЕСЕЧЕНИЯ ОТНОШЕНИЙ
- Подзапросы
- Имя таблицы

Ответ

- Операцию ОБЪЕДИНЕНИЯ ОТНОШЕНИЙ
- Операцию ПЕРЕСЕЧЕНИЯ ОТНОШЕНИЙ
- Подзапросы
- Имя таблицы

Пояснение
Преза 7, сл 15

Фраза **FROM**

Фраза **FROM** определяет одну или несколько таблиц или подзапросов, используемых для извлечения данных

```
FROM {{ имя_таблицы [AS] [table_alias] [(field .,:)]}
      | {subquery [AS] subquery_alas [(field .,:)]}
      | union_table
      | constructor_of_table_value
    }
```

Вопрос 9

Отметьте этапы разработки базы данных в информационной системе

Выберите один или несколько ответов:

- Реализация базы данных
- Разработка UML диаграмм
- Проектирование базы данных
- Разработка структуры системы
- Планирование базы данных

Ответ

- Реализация базы данных
- Проектирование базы данных
- Планирование базы данных

Пояснение

Преза 1, сл 13

- Этапы разработки базы данных в информационной системе
 - Проектирование базы данных
 - Логическое проектирование
 - Физическое проектирование
 - Планирование базы данных
 - Аппаратных средств
 - Программных средств
 - Реализация базы данных
 - Сопровождение
 - Рейнжиниринг

Хорошо написано про [проектирование](#) БД

Вопрос 10

Отметьте типы данных введенные в SQL-99, которых не было в SQL-92

Выберите один или несколько ответов:

- BLOB
- Rows
- Columns
- Date
- Reference
- Times
- Boolean
- CLOB

Ответ

- BLOB
- Boolean
- CLOB
- Reference
- Rows

Пояснение

Преза 13, сл 21, 24, 25

Новые типы данных. LOB

LOB (большие объекты) :

- **BLOB** {двоичные большие объекты} : это большие объемы двоичной информации, которая может быть чем угодно, но изначально предполагалось использовать BLOB для данных мультимедиа, таких, как аудио, видео и графическая информация
- **CLOB** (большие символьные объекты) : это объекты, похожие на BLOB, но предназначенные специально для хранения больших объемов текста; CLOB предполагается использовать в текстовых базах данных

Новые типы данных. Reference

Ссылка (Reference).

- Программистам более известна как *handle*. Ссылки — это переменные, в которых хранится местоположение некоторого значения, а не оно само.
- Под местоположением не обязательно понимается физический адрес в памяти. Эти адреса должны иметь смысл для разных компьютеров в сети.
- Это сделано чтобы
 - ❖ разрешить программисту создавать **ссылки на объекты, не существующие на этапе программирования** (это одно из традиционных применений указателей),
 - ❖ обеспечить **ссылки на большие объекты (БО)** как альтернативу их прямого представления.
- При таком подходе БО можно не передавать по сети и не загружать в память, до тех пор пока это действительно не потребуется.

Новые типы данных. Boolean. Rows. Tables

Булевский тип данных (Boolean)

Этот тип данных теперь поддерживается. Он отличается от однобитового типа данных BINARY, потому что **это булевские переменные в смысле SQL**. Они могут принимать три значения — UNKNOWN, TRUE и FALSE.

- **Строки (Rows)** Теперь со строками можно обращаться как с типами данных.
- **Таблицы (Tables)** Таблица может быть типом данных. Это сделано для того, чтобы **таблицы были больше похожи на классы** и реляционная модель лучше согласовывалась с объектно-ориентированным подходом.
- Такие таблицы называются *тилизированными таблицами* (typed tables). Они основаны на типах данных, определяемых пользователем (UDT).

Вопрос 11

Укажите способы решения проблемы тупиков

Выберите один или несколько ответов:

- Одновременная блокировка элементов БД, необходимых транзакции
- Требование двух-фазности транзакций
- Упорядоченная блокировка элементов БД, необходимых транзакции
- Рестарт транзакции из цикла графа ожиданий
- Упорядочение запуска транзакций
- Блокировка только изменяемых элементов БД, необходимых транзакции
- Завершение транзакции, попавшей в цикл графа ожиданий

Ответ

- Одновременная блокировка элементов БД, необходимых транзакции

- Упорядоченная блокировка элементов БД, необходимых транзакции
- Рестарт транзакции из цикла графа ожиданий

Пояснение

Преза 12, сл 31-34

Тупики – решение конфликта (1)

■ Вариант 1 (предотвращение).

Одновременная блокировка всех необходимых для транзакции элементов БД

Транзакция T_1 Транзакция T_2

Lock (A, B); Lock (B, A);

...

UnLock (B); UnLock (A);

UnLock (A); UnLock (B);

Тупики: варианты решения проблемы (2)

2. Ввести ЛИНЕЙНОЕ упорядочение элементов БД, требовать запрос блокировок делать в соответствии с ЭТИМ порядком

Пусть на множестве элементов БД введен линейный порядок :
 $A < B < C \dots < Z$

Тогда TP1 и TP2 должны блокировать эл.БД в этом порядке

TP1

Lock A;

Lock B;

...

UnLock A;

UnLock B;

TP2

Lock A;

Lock B;

...

UnLock B;

UnLock A;

Тупики – решение конфликта (3)

■ Вариант 3 (устранение).

- Строим граф ожиданий.
- Вершины графа – транзакции.
- Дуга из T_1 в T_2 ($T_1 \rightarrow T_2$) означает, что T_1 ожидает выполнения запроса на блокировку элемента БД, заблокированного в данный момент T_2
- Каждый цикл указывает тупик

Анализируется построенный граф.

В случае обнаружения цикла **делается рестарт (Rollback)** одной из транзакций попавших в цикл.

Вопрос 12

В какой наибольшей нормальной форме находится отношение R (A, B, C, D, E),

если A, B - ключ отношения и есть следующие функциональные зависимости

C → D

D → E

Выберите один ответ:

- во второй НФ
- в первой НФ

- в третьей НФ
- В БНФ

Ответ

- во второй НФ

Пояснение

По определению

1 НФ:

- в каждой клеточке таблицы только одно значение - да
- нет повторяющихся строк - да

2 НФ:

- таблица в 1 НФ - да
- есть первичный ключ - да
- все атрибуты зависят от первичного ключа целиком, а не только от какой-то его части - да

3 НФ:

- таблица в 2 НФ - да
- все атрибуты зависят только от первичного ключа, а не от неключевых атрибутов - НЕТ

Вопрос 13

Какое значение может принимать переменная SQLCODE в случае возникновения ошибки при выполнении оператора

Выберите один или несколько ответов:

- 2
- 1
- -2
- -1
- -4
- 0

Ответ

- -2
- -1
- -4

Пояснение

Преза 10, сл 21

SQLCODE (3)

3. Возникновение ошибки при выполнении оператора. В этом случае все изменения, выполненные в ходе текущей транзакции, могут быть отменены в зависимости от того, считается ли эта ошибка исправимой в данной реализации.

- ✓ В документации должны быть перечислены ошибки, вызывающие автоматический откат.
- ✓ Независимо от того, был ли выполнен автоматический откат, **SQLCODE** принимает **некоторое отрицательное значение**, определяемое реализацией.
- ✓ Оно должно идентифицировать возникшую проблему максимально точно.
- ✓ Как правило, система содержит подпрограмму для получения информации о значении отрицательных кодов, определенных разработчиком

Вопрос 14

Пусть в базе данных имеются следующие таблицы.

R1		
ФИО	Дисциплина	Оценка
Петров Ф.И.	Базы данных	5
Сидоров К.А.	Базы данных	4
Миронов А.В.	Базы данных	2
Петров Ф.И.	Моделирование	5
Сидоров К.А.	Моделирование	4
Миронов А.В.	Моделирование	Null
Трофимов П.А.	Сети ЭВМ	4
Иванова Е.А.	Сети ЭВМ	5
Уткина Н.В.	Сети ЭВМ	5

R2	
ФИО	Группа
Петров Ф.И.	АИ21
Сидоров К.А.	АИ21
Миронов А.В.	АИ21
Трофимов П.А.	АИ22
Иванова Е.А.	АИ22
Уткина Н.В.	АИ22

R3	
Группа	Дисциплина
АИ21	Базы данных
АИ21	Моделирование
АИ22	Сети ЭВМ

Какое количество строк будет получено в результате выполнения следующего запроса:
SELECT ФИО FROM R1 WHERE Оценка >= 4 GROUP BY ФИО HAVING COUNT(Оценка) > 1 ?

Выберите один ответ:

- 1

- 0
- 3
- 2

Ответ

- 2

Пояснение

Выбираем студентов, у которых оценка больше либо равна 4 и группируем по

фамилиям тех, у кого количество оценок больше одной.

Т.е. это будет Петров ФИ и Сидоров КА.

Вопрос 15

Можно ли во фразе WHERE оператора SELECT использовать другой оператор SELECT ?

Выберите один ответ:

- Верно
- Неверно

Ответ

- Верно

Пояснение

Преза 7, сл 33

Подзапросы (пример)

Найти ПОКУПАТЕЛЕЙ, которых обслуживают продавцы из Новосибирска

```
SELECT *
FROM ПОКУПАТЕЛИ
WHERE НомПР = (SELECT НомПР
                 FROM ПРОДАВЦЫ
                           WHERE Город ='Новосибирск');
```

- В данном операторе подзапрос всегда должен возвращать единственное значение, которое будет проверяться в предикате.

Вопрос 16

Укажите правильную последовательность выполнения операторов

для выполнения запроса

```
. SELECT snum, sname, city, comm
      INTO :snum, :sname,:city, :comm
      FROM Salespeople
      WHERE snum = :snum .
```

Выберите один ответ:

```
My_st := ' SELECT snum, sname, city, comm  
INTO :snum, :sname,:city, :comm  
FROM Salespeople  
WHERE snum = ?';  
EXEC SQL PREPARE LOCAL :SQL_st  
FROM :My_st  
EXEC SQL EXECUTE LOCAL: SQL_st  
USING :snum;
```

```
My_st := ' SELECT snum, sname, city, comm  
INTO :snum, :sname,:city, :comm  
FROM Salespeople  
WHERE snum = ?';  
EXEC SQL PREPARE LOCAL :SQL_st  
FROM :My_st  
USING :snum;
```

```
My_st := ' SELECT snum, sname, city, comm  
INTO :snum, :sname,:city, :comm  
FROM Salespeople  
WHERE snum = ?';  
EXEC SQL EXECUTE LOCAL: My_st  
USING :snum;
```

Ответ

```
My_st := ' SELECT snum, sname, city, comm  
INTO :snum, :sname,:city, :comm  
FROM Salespeople  
WHERE snum = ?';  
EXEC SQL PREPARE LOCAL :SQL_st  
FROM :My_st  
EXEC SQL EXECUTE LOCAL: SQL_st  
USING :snum;
```

Пояснение

Преза 10, сл 7-11

Выполнение подготовленных операторов

EXECUTE [GLOBAL | LOCAL] подготовленный оператор

[INTO {переменная, ... }
| {SQL DESCRIPTOR [GLOBAL | LOCAL] имя деск.}]

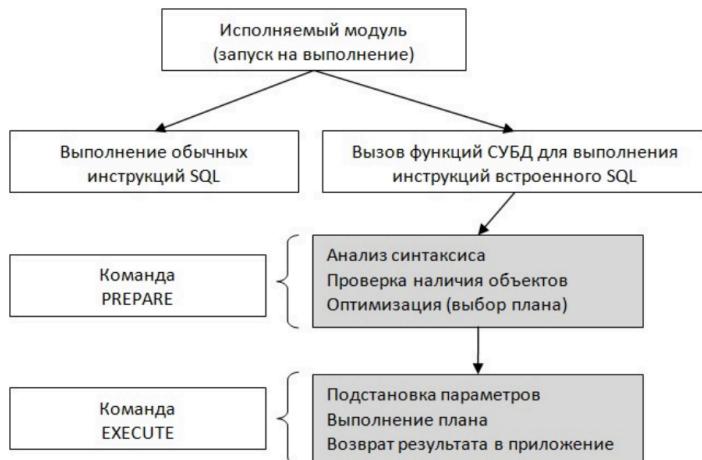
[USING {переменная, ... }
| {SQL DESCRIPTOR [GLOBAL | LOCAL] имя деск.}]

GLOBAL – оператор может быть выполнен отовсюду в пределах сеанса с СУБД

LOCAL – оператор может быть выполнен только из текущей программы

- Имеем 2 способа получения значений входных параметров и хранения значений выходных параметров
- Входные параметры – USING. Используется при наличии параметров.
- Выходные параметры – INTO (однострочный запрос), т.е. можно использовать в SELECT операторах, которые сами содержат INTO.
- Входные параметры – из переменных включающего языка -> оператор SQL.
- Выходные параметры – наоборот, значения выведенные оператором SQL и помещаемые в переменные программы на включающем языке.
- Итого EXECUTE может иметь одно из INTO, USING, оба или ни одного.

Общая схема выполнения операторов



Многократное выполнение

Синтаксис

EXEC SQL PREPARE [GLOBAL | LOCAL] <переменная объект-исполнения>
FROM <переменная объект-формирования>

- LOCAL – для выполнения только в текущей программе, модуле компиляции,
- GLOBAL – для выполнения на протяжении всего сеанса с СУБД

Пример

My_statement := 'INSERT INTO ЗАКАЗЫ VALUES (?, ?, ?, ?, ?, ?);'
EXEC SQL PREPARE GLOBAL :This_SQL_statement
FROM :My_statement;

- СУБД разберет оператор в My_statement, построит для него план выполнения и обработанный оператор SQL сохранится в виде объекта This_SQL_statement.
- EXEC SQL EXECUTE :This_SQL_statement; / выполнение
- Для запросов это возможно в случае, если результатом будет одна строка.
 - В противном случае используются курсоры.
 - После выполнения устанавливаются значения статусной переменной SQLSTATE или SQLCODE (>0 – успешное выполнение).
 - Если работа с оператором закончена, то чтобы освободить ресурс выполняется команда DEALLOCATE PREPARE.

DEALLOCATE PREPARE :This_SQL_statement;

Примеры

EXEC SQL BEGIN DECLARE SECTION

var

onum: integer;
amt: integer;
odate: integer;
cnum: integer;
snum: integer;

EXEC SQL END DECLARE SECTION;

My_statement := 'INSERT INTO ЗАКАЗЫ VALUES (?, ?, ?, ?, ?, ?);'

EXEC SQL PREPARE GLOBAL :This_SQL_statement
FROM :My_statement;

EXEC SQL EXECUTE :This_SQL_statement
USING :onum, :amt, :odate, :cnum, :snum;

Приведет к выполнению

INSERT INTO ЗАКАЗЫ VALUES (:onum, :amt, :odate, :cnum, :snum);

Динамические параметры

- Используются при многократном выполнении базового оператора, каждый раз с небольшими отличиями
- Можно использовать и во встроенным SQL (с помощью переменных включающего языка), но так как в динамическом SQL сами операторы создаются динамически – имеем 2 степени свободы:
 - ✓ можно динамически формировать базовый оператор
 - ✓ затем получать несколько вариантов с помощью параметров
- Удобно использовать в INSERT

My_statement := 'INSERT INTO ЗАКАЗЫ VALUES (?, ?, ?, ?, ?, ?);'

EXEC SQL PREPARE GLOBAL :This_SQL_statement
FROM :My_statement;

В операторе EXECUTE вместо знаков вопросов будут подставлены реальные значения.

Значения присваиваются в том порядке в каком они указаны в операторе EXECUTE.

Вопрос 17

Сколько триггеров можно определить на одну таблицу

Выберите один ответ:

- 9
- 15
- 6
- 12
- 10

Ответ

- 12

Пояснение [тык](#)

Когда	Событие	На уровне строк	На уровне оператора
BEFORE	INSERT/UPDATE/DELETE	Таблицы и сторонние таблицы	Таблицы, представления и сторонние таблицы
	TRUNCATE	—	Таблицы
AFTER	INSERT/UPDATE/DELETE	Таблицы и сторонние таблицы	Таблицы, представления и сторонние таблицы
	TRUNCATE	—	Таблицы
INSTEAD OF	INSERT/UPDATE/DELETE	Представления	—
	TRUNCATE	—	—

Вопрос 18

Выделите ограничения целостности, которые могут использоваться в определении столбца

Выберите один или несколько ответов:

- Выберите один или несколько ответов:
- PRIMARY KEY (столбец1, столбец2, ...)
- CHECK ()
- FOREING KEY (столбец1, столбец2, ...)
- NOT NULL
- REFERENCES ...
- UNIQUE

Ответ

- CHECK ()
- NOT NULL
- REFERENCES ...
- UNIQUE

Пояснение
Преза 6, сл 20

Ограничение столбца

<ограничение столбца> ::=

- [CONSTRAINT имя ограничения]
- | {NOT NULL}
- | { PRIMARY KEY }
- | {UNIQUE}
- | REFERENCES имя таблицы [(имя столбца)]
[<ссылочная спецификация>]
- | {CHECK предикат}
- [[INITIALLY DEFERRED] | INITIALLY IMMEDIATE]
- [[NOT] DEFERRABLE]

Вопрос 19

Выделите пункты, которые **не** содержат свойства ТРАНЗАКЦИИ

Выберите один или несколько ответов:

- Согласованность
- Уникальность
- Долговечность
- Надежность
- Завершенность
- Единственность
- Изолированность
- Автономность

Ответ

- Уникальность
- Надежность
- Завершенность
- Единственность

Пояснение

Преза 12, сл 13

Свойства транзакции

■ **Автономность (Atomicity). Все или ничего!**

Либо фиксируются все операции транзакции (commit), либо откат (rollback)

■ **Согласованность (Consistency). Шире понятия целостности.**

Согласованность может быть реализована на уровне бизнес логики. Например, операция « списание » должна соответствовать некоторой операции « зачисление »

■ **Изолированность (Isolation).**

При выполнении транзакций параллельно **одни** транзакции не должны оказывать влияние на результат выполнения **других** транзакций.

■ **Долговечность (Durability).**

Если пользователь получил подтверждение от системы, что транзакция выполнена, он может быть уверен, что сделанные им изменения не будут отменены из-за какого-либо сбоя.

Вопрос 20

Отметьте верные утверждения

Выберите один или несколько ответов:

- Хранимая процедура вызывается после события
- Триггер может выполняться над каждой строчкой таблицы
- хранимая процедура может содержать только операторы SQL
- триггер может содержать оператор ALTER
- Триггер срабатывает по событию
- Триггер срабатывает до или после события
- Хранимая процедура вызывается явно

Ответ

- Триггер может выполнять над каждой строчкой таблицы
- Триггер срабатывает по событию
- Триггер срабатывает до или после события
- Хранимая процедура вызывается явно

Пояснение

Преза 11, сл 13-16

Действие может выполняться [либо до, либо после события](#) триггера.

К числу [допустимых событий \(Event\)](#) обычно относят операции **INSERT, DELETE и UPDATE** [указанного отношения](#). Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.

Триггер "срабатывает" только при наступлении определенного [события \(event\)](#), описание которого содержится в тексте триггера.

Преза 11, сл 22

Хранимые процедуры

- Хранимая процедура - это программа на процедурном расширении языка SQL, сохраняемая в базе данных в откомпилированном виде.
- Хранение процедур в [том же месте, где они исполняются](#), обеспечивает [уменьшение трафика](#) по сети данных и повышает общую производительность системы.

Вопрос 21

Способ решения проблемы бесконечных ожиданий транзакций?
Выберите один ответ:

- Выполнить рестарт одной транзакции
- Выполнять транзакции по очереди
- Организовать выполнение транзакций через СТЭК
- Сортировать транзакции
- Выполнить откат одной транзакции

Ответ

- Выполнять транзакции по очереди

Пояснение

Преза 12 , сл 29

Бесконечные ожидания

- Пусть T_1 , находится в ожидании блокировки элемента **A** и еще T_2 запрашивает блокировку **A**.
- Рассмотрим ситуацию, когда T_2 получает блокировку раньше T_1 .
- Далее, в то время когда установила T_2 блокировку **A**, еще T_3 запросила блокировку **A**.
- Если T_3 получил блокировку **A** сразу за разблокированием **A** тран. T_2 , то T_1 все еще будет находиться в ожидании блокировки.

И, таким образом, некоторые транзакции могут находиться в **бесконечном ожидании**

Решение – очередь транзакций

29

Вопрос 22

Какое расписание множества транзакций называется сериализуемым ?

Выберите один ответ:

- Если все шаги каждой транзакции из данного множества выполняются вслед

- или перед всеми шагами других транзакций
- Если результат выполнения расписания эквивалентен результату любого последовательного расписания этого множества транзакций
 - Если шаги каждой транзакции из данного множества выполняются в линейном порядке
 - Если результат выполнения расписания эквивалентен результату некоторого последовательного расписания этого множества транзакций

Ответ

- Если результат выполнения расписания эквивалентен результату некоторого последовательного расписания этого множества транзакций

Пояснение

Преза 13, сл 35

Сериализуемость

Параллельное исполнение транзакций **КОРРЕКТНО** т. и т. т., когда их **совместный результат будет тем же самым**, что и при исполнении этих транзакций **в некотором последовательном порядке**.

- **Расписание** совокупности транзакций – **порядок** в котором выполняются **элементарные** шаги этих транзакций (блокировка, чтение, и т.д.).
- Расписание называется **последовательным**, если все шаги каждой транзакции выполняются вслед или перед всеми шагами других транзакций.
- Расписание называется **сериализуемым**, если его результат **эквивалентен** результату некоторого последовательного расписания.

Или вот [здесь](#) еще можно почитать

“Понятно, что для того, чтобы добиться изолированности транзакций, в СУБД должны использоваться какие-либо методы регулирования совместного выполнения транзакций.”

План (способ) выполнения набора транзакций называется **сериальным**, если результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций.

Сериализация транзакций - это механизм их выполнения по некоторому сериальному плану. Обеспечение такого механизма является основной функцией компонента СУБД, ответственного за управление транзакциями. Система, в которой поддерживается сериализация транзакций обеспечивает реальную изолированность пользователей.”

Вопрос 23

Что может быть элементом блокировки?

Выберите один или несколько ответов:

- Физическая страница базы данных
- База данных
- Таблица базы данных
- Элемент строки таблицы базы данных
- Стока таблицы базы данных
- Файл данных
- Буфер СУБД

Ответ

- Физическая страница базы данных
- Таблица базы данных
- Элемент строки таблицы базы данных
- Стока таблицы базы данных

Пояснение

АККУРАТНО

Преза 12, сл 27 (но тогда какого фига База данных - не ответ хмххмх)

Грануляции блокировок (Locking Level)

- В логических единицах БД:
 - Базы данных** (database-level locking).
 - Таблицы** (table-level locking). Блокировки быстро устанавливаются и снимаются, но максимально ограничивают доступность данных
 - Строки** (row-level locking). Применяется к отдельной строке таблицы. Наиболее распространены.
 - Элементы** (item-level locking). Блокируется элемент (конкретное значение) строки или столбца. Вариант идеален с точки зрения параллелизма, но работает довольно медленно. Не имеет широкого распространения.
- В физических единицах БД:
 - Пространства БД** или таблицы (dbspace-level locking, tablespace-level locking,)
 - Страницы** (dbspace-level locking). Единица хранения, настраиваемый размер, 1К. Очень эффективна по производительности, но...

27

Еще тут чот Преза 12, сл 22

Элементы блокировки

Что может быть элементом блокировки?

- Отношение
- Кортеж
- Компоненты кортежа
- Возможно некоторая совокупность кортежей (блок)

Размер элемента блокировки определяется АБД исходя из размера элемента, которым манипулирует средняя транзакция.

Вопрос 24

Укажите типовые проблемы параллельного исполнения транзакций

Выберите один или несколько ответов:

- Незавершенное чтение
- Параллельное чтение
- Потерянное обновление
- Преждевременное чтение
- Совместное обновление
- Совместное чтение
- Фантомные вставки
- Неповторяющееся чтение

Ответ

- Потерянное обновление
- Преждевременное чтение
- Фантомные вставки
- Неповторяющееся чтение

Пояснение

Преза 12, сл 14

Типовые проблемы параллелизма

- Независимое обновление одной и той же таблицы.
Продажа без блокировки таблицы товаров.
- Отмена изменений уже после их использования
- Влияние частичного (незафиксированного) результата одной операции на результат другой операции
- Возможны «тупики» вследствие мешающих друг другу действий пользователей

Стандартные термины для проблем параллелизма (документация СУБД)

- Потерянное обновление (Lost UPDATE)
- Преждевременное чтение (Dirty READ)
- Неповторяющееся чтение (Non-repeatable READ)
- Фантомные вставки (Phantom INSERT)

14

Вопрос 25

Укажите операторы, используемые для работы с курсорами

Выберите один или несколько ответов:

- a. CLOSE CURSOR
- b. UPDATE CURSOR
- c. SELECT
- d. DECLARE CURSOR
- e. FETCH
- f. INSERT
- g. OPEN CURSOR
- h. DELETE CURSOR

Ответ

- CLOSE CURSOR
- DECLARE CURSOR
- FETCH
- OPEN CURSOR

Пояснение

Преза 9, сл 9, 11

Курсы

Манипулирование данными

- Для начала работы с курсором нужно выполнить оператор **OPEN**

EXEC SQL OPEN CURSOR Томск_продавцы;

- Для построчного извлечения данных - **FETCH**

EXEC SQL FETCH Томск_продавцы

INTO :id_num, :salesperson, :loc, :comm;

- В конце работы нужно выполнить оператор **CLOSE**

EXEC SQL CLOSE CURSOR Томск_продавцы;

Позиционированные операторы

- Для обновления курсора в операторах **DELETE** и **UPDATE** может использоваться фраза **WHERE CURRENT OF**, определяющая, что действие относится к текущей строке курсора. Такой оператор называется позиционированным, и к нему предъявляются следующие требования:
 - ✓ и курсор, и оператор должны использовать только одну и ту же таблицу;
 - ✓ в запросе, используемом для создания курсора, не должно быть фраз **UNION** и **ORDER BY**;
 - ✓ курсор должен удовлетворять критериям обновляемого курсора (например, не применять агрегирующие функции).
- Пример:

```
EXEC SQL DECLARE CURSOR комиссионные
      FOR SELECT sname, comm
        FROM Salespeople;
EXEC SQL OPEN CURSOR комиссионные;
EXEC SQL FETCH комиссионные INTO :salesperson, :comm;
EXEC SQL
      UPDATE Salespeople
      SET comm = :comm *1.3
      WHERE CURRENT OF комиссионные;
```
- Позиционированный оператор **DELETE** удобно использовать для удаления из таблицы группы строк, предварительно выбранных в курсор.

Вопрос 26

Отметьте пункты, не относящиеся к фундаментальным свойствам отношений

Выберите один или несколько ответов:

- Упорядоченность кортежей
- Упорядоченность атрибутов
- Отсутствие дубликатов кортежей
- Отсутствие кортежей дубликатов
- Атомарность значений атрибутов

Ответ

- Упорядоченность кортежей
- Упорядоченность атрибутов

Пояснение

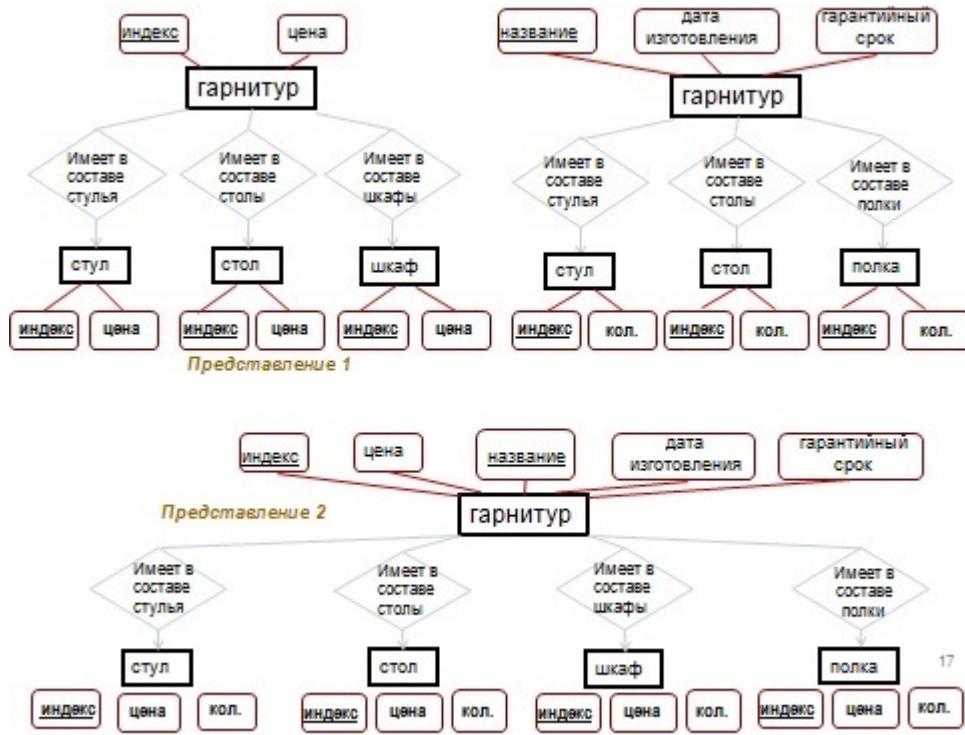
преза 3, сл. 12

Фундаментальные свойства отношений:

- Отсутствие кортежей дубликатов
- Отсутствие упорядоченности кортежей
- Отсутствие упорядоченности атрибутов
- Атомарность значений атрибутов

Вопрос 27

Какой операцией из представления 1 получено представление 2



Ответ

- агрегация

Пояснение

Преза 1, сл 36

Агрегация

- Агрегация – декартово произведение сущностей
- Новая сущность формируется на основе данных о частях объектах
 - Имя, Должность, Ном_отдела = Сотрудник
 - Имя, Паспорт, Дата приема = Сотрудник
 - Имя, Номер-страхового полиса, Адрес = Сотрудник

Имя, Номер-страхового полиса, Паспорт, Должность,
Ном_отдела, Дата приема, Адрес = СОТРУДНИК

Вопрос 28

Пусть отношение R содержит следующий набор кортежей

R (A, B, C, D)
2 3 5 6
3 4 7 8
2 3 5 7
3 6 7 8

Отметьте функциональные зависимости атрибутов в этом отношении
Выберите один или несколько ответов:

- C → D
- A → C
- A → D
- B → C
- A, B → C
- A, C → D

Ответ

- A → C
- B → C
- A, B → C

Пояснение

Посмотреть можно [тут](#).

Вопрос 29

Что означает фраза WITH GRANT OPTION в операторе GRANT

Выберите один ответ:

- передача полномочий на оператор с потерей собственных полномочий
- передача полномочий на оператор с добавлением себе полномочий
- передача полномочий на оператор с изменением собственных полномочий
- передача полномочий на оператор с сохранением собственных полномочий

Ответ

- передача полномочий на оператор с сохранением собственных полномочий

Пояснение

Преза 9, сл 15

Привилегии в SQL

- После фразы GRANT через запятую можно перечислить список всех назначаемых привилегий.
- Фраза ON определяет объект, для которого устанавливается привилегия.
- Фраза TO указывает пользователя или пользователей, для которых устанавливается привилегия
- Оператор GRANT SELECT ON tbl1 TO PUBLIC; предоставляет доступ к выполнению оператора SELECT для таблицы tbl1 не только всем существующим пользователям, но и тем, которые позднее будут добавлены в базу данных.
- Оператор GRANT UPDATE ON tbl1 TO user1; предоставляет пользователю user1 привилегию UPDATE на всю таблицу, а оператор GRANT UPDATE (f1,f2) ON tbl1 TO user1; предоставляет пользователю user1 привилегию UPDATE для изменения только столбцов f1 и f2.
- Фраза WITH GRANT OPTION предоставляет получающему привилегию пользователю дополнительную привилегию GRANT OPTION, позволяющую выполнять передачу полученных привилегий.

Украда:

“Благодаря параметру WITH GRANT OPTION, указанные в операторе GRANT пользователи имеют право передавать все предоставленные им в отношении указанного объекта привилегии другим пользователям, которые, в свою очередь, будут наделены точно таким же правом передачи своих полномочий. Если данный параметр не будет указан, получатель привилегии не сможет передать свои права другим пользователям. Таким образом, владелец объекта может четко контролировать, кто получил право доступа к объекту и какие полномочия ему предоставлены.”

Откуда-то взято:

“Команда GRANT используется для назначения привилегий пользователям.

Предоставляет разрешения на таблицу, представление, функцию с табличным значением, хранимую процедуру, расширенную хранимую процедуру, скалярную функцию, агрегатную функцию, очередь обслуживания или синоним."

Вопрос 30

Укажите операторы, используемые для описания ограничений целостности
Выберите один или несколько ответов:

- a. CREATE ASSERTION
- b. GRANT
- c. CHECK()
- d. PRIMARY KEY
- e. CREATE VIEW
- f. NON NULL

Ответ

- CREATE ASSERTION
- CHECK()
- PRIMARY KEY
- NON NULL

Пояснение

Преза 6, сл 20-21

Ограничение столбца

<ограничение столбца> ::=

[CONSTRAINT имя ограничения]
| {NOT NULL}
| {PRIMARY KEY }
| {UNIQUE}
| REFERENCES имя таблицы [(имя столбца)]
 [<ссыпочная спецификация>]
| {CHECK предикат}
[[INITIALLY DEFERRED] | INITIALLY IMMEDIATE]
|[NOT] DEFERRABLE]

Пр 11 сл 2 хмхмхм

Ограничение таблицы

<ограничение таблицы> ::=

[CONSTRAINT имя ограничения]
{ PRIMARY KEY (имя столбца, ...)}
| {UNIQUE (имя столбца, ...)}
| FOREIGN KEY (имя столбца, ...)
REFERENCES имя таблицы [(имя столбца, ...)]
 [<ссыпочная спецификация>]
| {CHECK предикат} [[NOT] DEFERRABLE]

Ограничения целостности

- на уровне колонки в операторе CREATE TABLE:
 - ❖ PRIMARY KEY,
 - ❖ REFERENCES,
 - ❖ NOT NULL,
 - ❖ UNIQUE,
 - ❖ CHECK (...).
- на уровне таблицы в операторе CREATE TABLE
 - ❖ PRIMARY KEY (...),
 - ❖ FOREIGN KEY (...),
 - ❖ UNIQUE (...),
 - ❖ CHECK (...).
- НА УРОВНЕ БАЗЫ ДАННЫХ
 - Assertion (ограничение общего вида, утверждения).
 - Trigger – последовательность команд, поставленная в соответствие определенному событию (операции(ям) модификации содержимого таблиц базы данных) и выполняемая в ответ на это событие.

ТЕСТ 3

Вопрос 1

Выделите базовые объекты сетевой модели данных

Выберите один или несколько ответов:

- Запись
- Набор
- Массив
- Список
- Элемент данных
- Поле
- Агрегат данных
- Стока

Ответ

- Запись
- Набор
- Элемент данных
- Агрегат данных

Пояснение

Преза 15 , сл 3, 5, 7

Модель данных

- Базовыми объектами модели являются:
 - ❖ Элемент данных
 - ❖ Агрегат данных
 - ❖ Запись
 - ❖ Набор
- **Элемент данных** — то же, что и в иерархической модели — минимальная информационная единица, доступная пользователю СУБД.
- **Агрегат данных** соответствует следующему уровню обобщения в модели.
- В модели определены **агрегаты двух типов**:
 - ❖ вектор,
 - ❖ повторяющаяся группа.

3

Запись данных

- Записью называется **совокупность агрегатов** или **элементов данных**, моделирующая объект из некоторого класса объектов реального мира.
- Понятие записи соответствует понятию "сегмент" в иерархической модели.
- Для записи, так же как и для сегмента, вводятся понятия **типа записи** и **экземпляра записи**.
- При моделировании реального мира могут быть варианты, что определить как тип записи

5

Набор данных

- Набором называется отношение «**один-ко-многим**», связывающее записи разных типов.
- **Тип набора (тип связи)** отражает иерархическую связь между **двумя** или более **тиปами записей**.
- **Родительский тип записи (предок)** в данном наборе называется **владельцем набора**
- **Дочерний(ие) тип(ы) записи (потомок(и))** — **членом (нами)** того же **набора**.



Вопрос 2

Отметьте типы данных добавленные SQL-99

Выберите один или несколько ответов:

- Double float (вещественный с улучшенной точностью)
- Boolean (Булевский тип данных)
- Tables (Таблицы)
- Class (Класс)
- BLOB (двоичные большие объекты)
- CLOB (большие символьные объекты)
- Multimedia(Мультимедийный)
- Rows (Строки)

Ответ

- Boolean (Булевский тип данных)
- Tables (Таблицы)
- BLOB (двоичные большие объекты)
- CLOB (большие символьные объекты)
- Rows (Строки)

Пояснение

Преза 13, сл 21, 24, 25

Новые типы данных. Reference

Новые типы данных. LOB

LOB (большие объекты) :

- **BLOB** {двоичные большие объекты}: это большие объемы двоичной информации, которая может быть чем угодно, но изначально предполагалось использовать BLOB для данных мультимедиа, таких, как аудио, видео и графическая информация
- **CLOB** (большие символьные объекты): это объекты, похожие на BLOB, но предназначенные специально для хранения больших объемов текста; CLOB предполагается использовать в текстовых базах данных

Ссылка (Reference).

- Программистам более известна как *handle*. Ссылки — это переменные, в которых хранится местоположение некоторого значения, а не оно само.
- Под местоположением не обязательно понимается физический адрес в памяти. Эти адреса должны иметь смысл для разных компьютеров в сети.
- Это сделано чтобы
 - ❖ разрешить программисту создавать **ссылки на объекты, не существующие на этапе программирования** (это одно из традиционных применений указателей),
 - ❖ обеспечить **ссылки на большие объекты (БО)** как альтернативу их прямого представления.
- При таком подходе БО можно не передавать по сети и не загружать в память, до тех пор пока это действительно не потребуется.

Новые типы данных. Boolean. Rows. Tables

■ **Булевский тип данных (Boolean)**

Этот тип данных теперь поддерживается. Он отличается от однобитового типа данных BINARY, потому что **это булевские переменные в смысле SQL**. Они могут принимать три значения — UNKNOWN, TRUE и FALSE.

■ **Строки (Rows)** Теперь со строками можно обращаться как с типами данных.

■ **Таблицы (Tables)** Таблица может быть типом данных. Это сделано для того, чтобы **таблицы были больше похожи на классы** и реляционная модель лучше согласовывалась с объектно-ориентированным подходом.

■ Такие таблицы называются *типовыми таблицами* (typed tables). Они основаны на типах данных, определяемых пользователем (UDT).

Вопрос 3

Продолжите предложение:

"В реляционной модели данных непроцедурное манипулирование данными описывает
_____"

Ответ

- реляционное исчисление

Пояснение

Преза , сл

Вопрос 4

Можно ли во фразе WHERE оператора SELECT использовать другой оператор SELECT ?

Ответ

- Верно

Пояснение

Преза , сл

Вопрос 5

Отметьте правильные утверждения об объектной модели стандарта Object Data Management Group

Выберите один или несколько ответов:

- Объекты имеют свойства и методы
- Объект имеет уникальный идентификатор
- Поведение объекта определяется свойствами
- Литерал - это экземпляр класса
- Свойства объекта - это набор атрибутов и бинарных связей

Ответ

- Объекты имеют свойства и методы
- Объект имеет уникальный идентификатор
- Свойства объекта - это набор атрибутов и бинарных связей

Пояснение

Преза 16 , сл 12

Объектная модель ODMG

- **Объектная модель** - унифицированная основа всего стандарта.
- Она расширяет объектную модель консорциума OMG за счет введения таких свойств как **связи** и понятия **транзакции** для обеспечения функциональности, требуемой при взаимодействии с базами данных.
- Модель определяет 2 вида примитивов моделирования:
объекты (изменчивые типы), **литералы** (неизменчивые типы)
- Примитивы обоих видов могут быть: **атомарные, коллекции**
- Каждый объект имеет **уникальный идентификатор**
- Литералы **не имеют уникального идентификатора**, т.е. самоидентифицируемы
- Объекты и литералы обладают **типами**
- Между **типами** могут устанавливаться **бинарные связи** (вида 1:1, 1:n и n:m)
- Состояние элементов данного типа определяется соответствующими им **значениями его свойств** (общих для всего типа) — **атрибутов или бинарных связей** с экземплярами других типов.
- Поведение экземпляров типа определяется **набором операций**, задаваемых их сигнатурами, и этот набор также является общим для всех экземпляров **данного** типа.

12

Вопрос 6

Выберите термины, не относящиеся к ER модели

Выберите один или несколько ответов:

- сущность
- связь
- набор
- запись
- атрибуты

Ответ

- набор
- запись

Пояснение

Преза 1, сл 32

Этапы разработки ER модели

- Разработка локальных представлений (частей информационной системы)
 - ❖ Формулирование сущностей (имена, содержание)
 - ❖ Выбор идентифицирующего атрибута
 - ❖ Спецификация связей (имена, типы)
 - ❖ Добавление описательных атрибутов
- Объединение представлений пользователей
 - ❖ Идентичность
 - ❖ Агрегация
 - ❖ Обобщение

Вопрос 7

Отметьте правильные утверждения о сетевой модели данных

Выберите один или несколько ответов:

- Тип записи может быть одновременно владельцем и членом нескольких типов наборов
- В базе данных обязан быть хотя один набор
- Между двумя типами записи может быть определено любое количество типов наборов
- Не допускаются циклические структуры между типами записей в наборах

Ответ

- Тип записи может быть одновременно владельцем и членом нескольких типов наборов
- Между двумя типами записи может быть определено любое количество типов наборов

Пояснение

Преза 15 , сл 9

Свойства наборов данных

- ❖ Тип записи **может быть одновременно владельцем и членом нескольких типов наборов**
- ❖ Между **двух типами записи** может быть определено **любое количество типов наборов.**
- ❖ База данных может содержать **любое число** типов набора и типов записей
- ❖ Допускаются циклические структуры.
То есть в **одном наборе** тип записи *X* может быть **владельцем** и тип записи *Y* **членом**, а в **другом наборе** **наоборот**.

Вопрос 8

Пусть даны отношения R (A:float, B:char(20), C :float) и S (A:float, B:char(20))
Укажите, какие операции реляционной алгебры можно сделать над этими отношениями
Выберите один или несколько ответов:

- Декартово произведение
- Объединение
- Пересечение
- Натуральное соединение
- Деление
- Разность

Ответ

- Декартово произведение
- Натуральное соединение
- Деление

Пояснение

Преза , сл

Вопрос 9

Отметьте свойства триггера

Выберите один или несколько ответов:

- действие может выполняться после события
- событием является операция SELECT
- событием является операция INSERT
- действие может выполняться до события
- "срабатывает" только при наступлении определенного события
- определяется на таблицах
- перед выполнением операции может проверить условие выполнения
- определяется для VIEW

Ответ

- действие может выполняться после события
- событием является операция INSERT
- действие может выполняться до события
- "срабатывает" только при наступлении определенного события
- определяется на таблицах
- перед выполнением операции может проверить условие выполнения

Пояснение

Преза 11, сл 13-16

Вот сначала просто всё по каждому правильному ответу:

Действие может выполняться **либо до, либо после события** триггера.

К числу **допустимых событий (Event)** обычно относят операции **INSERT, DELETE и UPDATE** указанного **отношения**. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.

Триггер "срабатывает" только при наступлении определенного **события (event)**, описание которого содержится в тексте триггера.

■ Триггеры определяются **ТОЛЬКО на таблицах, не на VIEW**

Вместо того чтобы сразу отвергнуть операцию, вызвавшую его срабатывание, триггер выполняет проверку **условия (condition)**. Если результат проверки равен значению **FALSE**, работа триггера завершается.

Просто скрины слайдов:

ТРИГГЕРЫ

- Триггеры еще называют **правилами** "событие — условие — действие"
event — condition — action rules или ECA rules
- Отличаются от других ограничений, рассмотренных выше, в трех аспектах.
 - ❖ Триггер "срабатывает" только при наступлении определенного **события** (event), описание которого содержится в тексте триггера.
 - ❖ Вместо того чтобы сразу отвергнуть операцию, вызвавшую его срабатывание, триггер выполняет проверку **условия** (condition). Если результат проверки равен значению FALSE, работа триггера завершается.
 - ❖ Если же условие, сформулированное в тексте триггера, справедливо (TRUE) и выполняет **действие** (action), поставленное в соответствие событию.

Ограничения секции "действия"

- В секции "действия" текста триггера разрешено использовать далеко не все команды SQL.
- Например, нельзя применять команды
 - управления транзакциями (подобные COMMIT и ROLLBACK),
 - организации соединения между клиентом и сервером базы данных (скажем, CONNECT и DISCONNECT),
 - создания (CREATE), изменения (ALTER) и удаления (DROP) элементов схемы,
 - определения параметров сеанса (SET) и пр

События, действия

- К числу **допустимых событий (Event)** обычно относят операции INSERT, DELETE и UPDATE указанного **отношения**. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.
- Под **действием (Action)** понимается последовательность операций с базой данных, отвечающая логике определенного события. Оформляется как программный блок, хранящийся в базе данных
- Триггеры определяются **ТОЛЬКО на таблицах, не на VIEW**
- Тем не менее триггеры срабатывают, когда операции модификации исполняются из под VIEW

Условие, время выполнения

- Действие может выполняться **либо до, либо после события** триггера.
- При выполнении действия триггер способен ссылаться как на прежние (old), так и на новые (new) значения компонентов кортежа, операция модификации которого рассматривается как событие, повлекшее срабатывание триггера.
- События обновления данных могут быть ограничены операциями над отдельным компонентом или множеством компонентов.
- Условие может задаваться с помощью предложения WHEN: действие выполняется при наступлении события и только в том случае, если в этот момент условие оставалось справедливым.
- Действие может выполняться в одном из двух возможных режимов:
 - уровень кортежа: один раз для каждого модифицируемого кортежа;
 - уровень команды: единожды для всех кортежей, которые подверглись изменению при выполнении одной операции.

Вопрос 10

Можно ли во фразе WHERE оператора SELECT использовать **агрегатные функции** ?

Выберите один ответ:

- Верно
- Неверно

Ответ

- Неверно

Пояснение

Преза 7, сл 16, 27

Фраза WHERE

- Фраза **WHERE** определяет условие, которому должны удовлетворять все строки, используемые для формирования результирующего набора

WHERE condition

Condition – логическая функция (предикат).

- Предикат содержит **одно или несколько выражений**, выполняющих сравнения.
- Применяется трехзначная логика: **TRUE**, **FALSE** или **UNKNOWN**.
- Значение **UNKNOWN** получается при сравнении значения **NULL** с любым другим значением, включая и само значение **NULL**.

16

Агрегатные функции

Агрегатные функции (еще называют функции множества) в качестве параметров имеют группу значений, указанных в запросе (или подзапросе) в предложении **SELECT** или **HAVING** и вычисляют единственное значение-результат

COUNT (*) функция подсчитывает количество строк, получаемых по запросу. Не учитываются значения отдельных столбцов и результат не зависит от того указан ли параметр **DISTINCT** или имеются значения **NULL**.

COUNT(имя поля) - автоматически **отбрасывает (игнорирует), как и все другие агрегатные функции значения **NULL**** и выдает кол-во строк для не-**NULL** значения полей которые выбраны запросом.

AVG(имя поля) - вычисляет среднее арифметическое (среднее) значение для указанных элементов. Можно использовать только для типов данных: **NUMERIC** (цифровой) и **INTERVAL** (интервал)

SUM (имя поля) - выдает арифметическую сумму значений. Может использоваться для тех же типов, что и **AVG**.

MAX (имя поля) - возвращает наибольшее из всех значений, вычисленных на основе правила предикатов сравнения. Может использоваться для любых типов данных.

MIN (имя поля) - возвращает наименьшее из указанных значений.

Агрегатные функции, включенные в предложение **SELECT**, не содержащее **GROUP BY** исполняются над всеми результирующими строками.

Если запрос содержит **GROUP BY**, то агрегатные функции выполняются для каждой группы (набора строк имеющих одинаковое значение по столбцам входящим в группу) отдельно .

27

Вопрос 11 хм

Что значит, что модель данных допускает таблицы в не первой нормальной форме ?

Выберите один или несколько ответов:

- Поле строки таблицы может содержать несколько значений разных типов
- Поле строки таблицы может содержать несколько значений структурного типа
- Поле строки таблицы может содержать одно значение разного типа
- Поле строки таблицы может содержать несколько значений атомарного типа

upd. от 2024г:

- Несколько атрибутов могут образовать ассоциацию нескольких значений атомарного типа

Ответ

- Поле строки таблицы может содержать несколько значений структурного типа
- Поле строки таблицы может содержать несколько значений атомарного типа

Как будто бы правильный ответ только

- Несколько атрибутов могут образовать ассоциацию нескольких значений атомарного типа

Вот что какие ответы были и скок баллов

Вопрос 12

Выполнен

Баллов: 0,00 из
1,00

Отметить
вопрос

Что значит, что модель данных допускает таблицы в **не** первой нормальной форме ?

Выберите один или несколько ответов:

- Атрибут таблицы может содержать несколько значений атомарного типа
- Несколько атрибутов таблицы могут образовать ассоциацию нескольких значений атомарного типа
- Атрибут таблицы может содержать несколько значений разных типов
- Атрибут таблицы может содержать несколько значений структурного типа

Вопрос 38

Выполнен

Баллов: 0,00 из
1,00▼ Отметить
вопрос

Что значит, что модель данных допускает таблицы в не первой нормальной форме ?

Выберите один или несколько ответов:

- Атрибут таблицы может содержать несколько значений разных типов
- Атрибут таблицы может содержать несколько значений атомарного типа
- Атрибут таблицы может содержать несколько значений структурного типа
- Несколько атрибутов таблицы могут образовать ассоциацию несколько значений атомарного типа

Вопрос 26

Выполнен

Баллов: 0,00 из 1,00

▼ Отметить вопрос

Что значит, что модель данных допускает таблицы в не первой нормальной форме ?

Выберите один или несколько ответов:

- Атрибут таблицы может содержать несколько значений разных типов
- Атрибут таблицы может содержать несколько значений структурного типа
- Атрибут таблицы может содержать несколько значений атомарного типа
- Несколько атрибутов таблицы могут образовать ассоциацию несколько значений атомарного типа

Пояснение

Преза 16 прмд, сл 4

Не первая нормальная форма NF2

- **Достоинством** постреляционной модели является возможность представления **совокупности связанных реляционных таблиц** одной постреляционной таблицей.
- Это обеспечивает высокую наглядность представления информации и повышение эффективности её хранения и обработки.
- **Недостатком** постреляционной модели является сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.
- В основе проблемы лежат три вопроса:
 - ✓ работа с полями переменной длины и группами записей;
 - ✓ управление отношениями между таблицами и полями;
 - ✓ отражение подлинно семантического содержания реальных структур, которые будут моделированы базой данных.

Структуры модели данных

- Модель данных постреляционной СУБД jBASE поддерживает **ассоциированные многозначные поля**, которые часто называют **множественными группами**.
- Вы можете связать **несколько столбцов с множественными значениями** в единое целое, называемое **ассоциацией**.
- При этом в строке **первое значение одного столбца ассоциации** соответствует **первым значениям всех других столбцов ассоциации**, в такой же связи находятся все вторые значения столбцов и т.д.
- Многозначные поля и ассоциации не могут вкладываться друг в друга.
- Расширение синтаксиса SQL позволяет осуществлять доступ к множественным полям как расширению реляционной модели.
- Можно применять также стандартный SQL или язык запросов СУБД jBASE - jQL.
- В целом многозначность полей является очень полезным свойством при создании коммерческих приложений, где информация нередко представлена в виде списков предметов.

Вопрос 12

Отметьте, что можно использовать во фразе FROM

Выберите один или несколько ответов:

- Операцию ПРОЕКЦИЯ
- Операцию ОБЪЕДИНЕНИЯ ОТНОШЕНИЙ
- Агрегатную функцию
- Операцию ПЕРЕСЕЧЕНИЯ ОТНОШЕНИЙ
- Подзапросы
- Имя таблицы

Ответ

- Операцию ОБЪЕДИНЕНИЯ ОТНОШЕНИЙ
- Операцию ПЕРЕСЕЧЕНИЯ ОТНОШЕНИЙ
- Подзапросы
- Имя таблицы

Пояснение

Преза 7, сл 15

Фраза **FROM**

Фраза **FROM** определяет одну или несколько таблиц или подзапросов, используемых для извлечения данных

```
FROM {{ имя_таблицы [AS] [table_alias] [(field ..,:)]}
      | {subquery [AS] subquery_alas [(field ..,:)]}
      | union_table
      | constructor_of_table_value
}
```

Вопрос 13

Какой тип связи описан в примере ?

```
class Star

(extent Stars key name )

{
    attribute string name;

    attribute Struct Addr

        {string street, string city} address;

    relationship Set<Movie> starredIn

        inverse Movie: :starsOf;

}
```

Выберите один ответ:

- Никакой
- отношение N:M
- отношение 1:1
- отношение 1:M

Ответ

- отношение N:M

Пояснение

Преза 16, сл 24

relationship Set<Movie> starredIn
inverse Movie: :starsOf;

Объявление связей (n:m, 1:m, m:1)

```
class Movie
(extent Movies key(title, year) )
{
    attribute string title;
    attribute integer year;
    attribute integer length;
    attribute enum Film {color, blackAndWhite} filmType;
    relationship Set<Star> starsOf      %(n:m)
        inverse Star::starredIn;
    relationship Studio ownedBy      %(m:1)
        inverse Studio::owns;
};
```

Вопрос 14

Укажите метод доказательства теоремы "эквивалентности реляционной алгебры и реляционного исчисления на кортежах" и набор операций, используемых в доказательстве

Выберите один или несколько ответов:

- Объединение
- Проекция
- Деление
- метод "от противного"
- Натуральное соединение
- Разность
- Селекция
- метод "перебором всех вариантов"
- метод "по индукции"

Ответ

- метод "по индукции"
- Селекция
- Проекция
- Объединение
- Разность

Пояснение

Преза 4, сл 13 -

Редукция РА к РИК

Теорема 1. Если E – выражение реляционной алгебры, то существует эквивалентное ему безопасное выражение в реляционном исчислении с переменными-кортежами.

Доказательство.

Индукция по числу вхождений операторов в E .

Базис. Нуль операторов.

Тогда E – либо постоянное выражение $\{t_1, t_2, \dots, t_n\}$, либо переменная R , обозначающая отношение.

В последнем случае E эквивалентно $\{t | R(t)\}$, которое безопасно.

В первом случае E эквивалентно $\{t | t=t_1 \wedge t=t_2 \wedge \dots \wedge t=t_n\}$, где $t=t_i$ – краткая запись $t[1]=t_i[1] \& t[2]=t_i[2] \& \dots \& t[k]=t_i[k]$

$1 \leq i \leq n$

Редукция РА к РИК

Случай 2. $E = E_1 \setminus E_2$.

Пусть E_1 эквивалентно безопасному выражению $\{t|\varphi_1(t)\}$

Пусть E_2 эквивалентно безопасному выражению $\{t|\varphi_2(t)\}$

Тогда для E строим эквивалентное выражение

$$\{t | \varphi_1(t) \& \neg \varphi_2(t)\}.$$

Так как $\text{DOM}(\varphi_1(t) \& \neg \varphi_2(t)) = \text{DOM}(\varphi_1(t)) \cup \text{DOM}(\varphi_2(t))$, то E эквивалентно безопасному выражению

$$\{t | \varphi_1(t) \& \neg \varphi_2(t)\}$$

Редукция РА к РИК

Случай 2. $E = E_1 \setminus E_2$.

Пусть E_1 эквивалентно безопасному выражению $\{t|\varphi_1(t)\}$

Пусть E_2 эквивалентно безопасному выражению $\{t|\varphi_2(t)\}$

Тогда для E строим эквивалентное выражение

$$\{t | \varphi_1(t) \& \neg \varphi_2(t)\}.$$

Так как $\text{DOM}(\varphi_1(t) \& \neg \varphi_2(t)) = \text{DOM}(\varphi_1(t)) \cup \text{DOM}(\varphi_2(t))$, то E эквивалентно безопасному выражению

$$\{t | \varphi_1(t) \& \neg \varphi_2(t)\}$$

Редукция РА к РИК

Случай 3. $E = E_1 \times E_2$.

Пусть E_1 – отношение степени k и эквивалентно
безопасному выражению $\{t|\phi_1(t)\}$

Пусть E_2 – отношение степени s и эквивалентно
безопасному выражению $\{t|\phi_2(t)\}$

Тогда для E эквивалентно выражению

$\phi(t) = \{t^{(k+s)}| \exists u \exists v (\phi_1(u) \& \phi_2(v) \& t[1]=u[1] \& \dots \& t[k]=u[k] \&$
 $\& t[k+1]=v[1] \& \dots \& t[k+s]=v[s])\}$.

$DOM(\phi(t))=DOM(\phi_1(t)) \cup DOM(\phi_2(t))$.

Выражение $\phi(t)$ безопасно, так как $t[i]$ ограничено значениями $u[i]$,
если $i \leq k$, и значениями $v[i]$, если $k < i \leq k + s$

Редукция РА к РИК

Случай 4. $E = \pi_{i_1, i_2, \dots, i_k}(E_1)$.

Пусть E_1 – отношение степени n и эквивалентно
безопасному выражению $\{t|\phi_1(t)\}$

Тогда для E эквивалентное выражение

$\phi(t) = \{t^{(k)}| \exists u (\phi_1(u) \& t[1]=u[i_1] \& \dots \& t[k]=u[i_k])\}$,
где $k \leq n$

$DOM(\phi(t))=DOM(\phi_1(t))$.

Выражение $\phi(t)$ безопасно, так как $t[i]$ ограничено
значениями $u[i_j]$, если $j \leq n$

Вопрос 15

Выделите пункты, в которых перечислены фундаментальные свойства отношений

Выберите один или несколько ответов:

- Упорядоченность кортежей
- Упорядоченность атрибутов
- Отсутствие дубликатов кортежей
- Отсутствие кортежей дубликатов
- Атомарность значений атрибутов

Ответ

- Отсутствие дубликатов кортежей
- Отсутствие кортежей дубликатов
- Атомарность значений атрибутов

Пояснение

преза 3, сл. 12

Фундаментальные свойства отношений:

- Отсутствие кортежей дубликатов
- Отсутствие упорядоченности кортежей
- Отсутствие упорядоченности атрибутов
- Атомарность значений атрибутов

Вопрос 16

Сопоставьте тип базы данных и основную структуру соответствующей модели данных

- База данных Ключ-значение –
- Реляционная база данных –
- Иерархическая база данных –
- Сетевая база данных –

Ответ

- База данных Ключ-значение – СЛОВАРЬ
- Реляционная база данных – ОТНОШЕНИЕ
- Иерархическая база данных – СЕГМЕНТ
- Сетевая база данных – ЗАПИСЬ

Пояснение

Преза 14, сл 3

Основными информационными единицами в иерархической модели являются:

- ❖ **База данных**
- ❖ **Сегмент**
- ❖ **Поле (элемент данных)**

Преза 15 сл 5

Запись данных

- **Записью** называется *совокупность агрегатов* или *элементов данных*, моделирующая объект из некоторого класса объектов реального мира.
- Понятие записи соответствует понятию "сегмент" в иерархической модели.
- Для записи, так же как и для сегмента, вводятся понятия *типа записи* и *экземпляра записи*.
- При моделировании реального мира могут быть варианты, что определить как тип записи

Вопрос 17

Укажите типовые проблемы параллельного исполнения транзакций

Выберите один или несколько ответов:

- Незавершенное чтение
- Параллельное чтение
- Потерянное обновление
- Преждевременное чтение
- Совместное обновление
- Совместное чтение
- Фантомные вставки
- Неповторяющееся чтение

Ответ

- Потерянное обновление
- Преждевременное чтение
- Фантомные вставки
- Неповторяющееся чтение

Пояснение

Преза 12, сл 14

Типовые проблемы параллелизма

- Независимое обновление одной и той же таблицы.
Продажа без блокировки таблицы товаров.
- Отмена изменений уже после их использования
- Влияние частичного (незафиксированного) результата одной операции на результат другой операции
- Возможны «тупики» вследствие мешающих друг другу действий пользователей

Стандартные термины для проблем параллелизма (документация СУБД)

- Потерянное обновление (Lost UPDATE)
- Преждевременное чтение (Dirty READ)
- Неповторяющееся чтение (Non-repeatable READ)
- Фантомные вставки (Phantom INSERT)

Вопрос 18

Продолжите предложение:

"В реляционной модели данных процедурное манипулирование данными описывает

“

Ответ

- реляционная алгебра

Пояснение

Преза , сл

Вопрос 19

Отметьте предложения, характеризующие свойства постреляционной модели данных

Выберите один или несколько ответов:

- Постреляционная модель всегда медленнее реляционной модели
- Форма NF2 не нарушает принципы реляционной алгебры
- Постреляционная модель не позволяет делать нормализацию данных
- Таблицы могут содержать поля в не первой нормальной форме (NF2)

Ответ

- Форма NF2 не нарушает принципы реляционной алгебры
- Таблицы могут содержать поля в не первой нормальной форме (NF2)

Пояснение

Преза 16прмд, сл 2, 7

Постреляционная модель

- Важным аспектом традиционной реляционной модели данных является тот факт, что **элементы данных**, которые хранятся на пересечении строк и столбцов таблицы, **должны быть неделимы и единственны**.
- Такое правило было заложено в основу реляционной алгебры при ее разработке как математической модели данных.
- Постреляционная модель представляет собой **расширенную реляционную модель**, снимающую ограничение неделимости данных.
- Модель допускает многозначные поля – поля, значения которых состоят из множества подзначений.
- Набор значений многозначных полей считается **самостоятельной таблицей, встроенной в основную таблицу**.
- Таким образом, модель использует **трехмерные структуры**, позволяя хранить в полях таблицы **другие таблицы**. Тем самым расширяются возможности по описанию **сложных объектов** реального мира или **связей между объектами**.

NF2

- О таблицах, содержащих многозначные поля, говорят, что они находятся в непервой нормальной форме (**Non First Normal Form -NFNF**) или **NF2**
- При условии, что **вложенная таблица** удовлетворяет общим критериям (например **имеет уникальный ключ**), естественным образом происходит **расширение операторов реляционной алгебры**.
- Так как **многозначные поля** подчиняются правилам, позволяющим обращаться с ними, как с таблицами, встроенными в другие таблицы, **форма NF2 не нарушает принципы реляционной алгебры**.
- Расширенные операторы, которые работают с таблицами NF2, позволяют извлекать встроенные таблицы и рассматривать данные как информацию, поступившую из таблиц 1NF.
- Однако во многих случаях форма 1NF будет скорее исключением, а не правилом.
- В большинстве случаев гораздо **более эффективно** осуществлять **доступ к многозначным полям одновременно с остальными данными**, зная, что их всегда можно извлечь и рассматривать как **отдельную таблицу** в тех случаях, когда это может понадобиться.

Вопрос 20

Что может быть элементом блокировки?

Выберите один или несколько ответов:

- Физическая страница базы данных
- База данных
- Таблица базы данных
- Элемент строки таблицы базы данных
- Стока таблицы базы данных
- Файл данных
- Буфер СУБД

Ответ

- Физическая страница базы данных
- Таблица базы данных
- Элемент строки таблицы базы данных
- Стока таблицы базы данных

Пояснение

АККУРАТНО

Преза 12, сл 27 (но тогда какого фига База данных - не ответ хмххмхм)

Еще тут чот Преза 12, сл 22

Грануляции блокировок (Locking Level)

- В логических единицах БД:
 - Базы данных (database-level locking).
 - Таблицы (table-level locking). Блокировки быстро устанавливаются и снимаются, но максимально ограничивают доступность данных
 - Строки (row-level locking). Применяется к отдельной строке таблицы. Наиболее распространены.
 - Элементы (item-level locking). Блокируется элемент (конкретное значение) строки или столбца. Вариант идеален с точки зрения параллелизма, но работает довольно медленно. Не имеет широкого распространения.
- В физических единицах БД:
 - **Пространства** БД или таблицы (dbspace-level locking, tablespace-level locking.)
 - **Страницы** (dbspace-level locking). Единица хранения, настраиваемый размер, 1К. Очень эффективна по производительности, но...

27

Элементы блокировки

Что может быть элементом блокировки?

- Отношение
- Кортеж
- Компоненты кортежа
- Возможно некоторая совокупность кортежей (блок)

Размер элемента блокировки определяется АБД исходя из размера элемента, которым манипулирует средняя транзакция.

Вопрос 21

Какой операцией схема 2 получена из схемы 1 ?

Схема 1

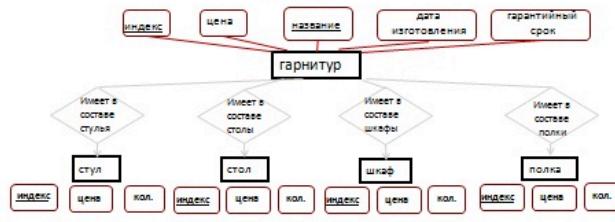


Схема 2



Ответ

- обобщение

Пояснение

Преза 1, сл 38

Обобщение

- Обобщение – это абстракция данных, позволяющая трактовать **класс объектов** как **ОДИН объект**
- При агрегации **части соединяются в целое**
- Обобщение фиксирует **РОДО-ВИДОВЫЕ отношения**
- **ВИД** есть **РОД** в совокупности с видовым отличием
- Ваз 2109, Ваз 2101, Ваз 2107 = **Автомобили марки ВАЗ**
- Автомобили марки ВАЗ, автомобили марки Хонда, ... = **Автомобили**
- Автомобиль, велосипед, мотоцикл, трактор = **Колесное средство передвижения**

Вопрос 22

Какое расписание множества транзакций называется сериализуемым ?

Выберите один ответ:

- Если все шаги каждой транзакции из данного множества выполняются вслед или перед всеми шагами других транзакций
- Если результат выполнения расписания эквивалентен результату любого последовательного расписания этого множества транзакций
- Если шаги каждой транзакции из данного множества выполняются в линейном порядке
- Если результат выполнения расписания эквивалентен результату некоторого последовательного расписания этого множества транзакций

Ответ

- Если результат выполнения расписания эквивалентен результату некоторого последовательного расписания этого множества транзакций

Пояснение

Преза 13, сл 35

Сериализуемость

Параллельное исполнение транзакций **КОРРЕКТНО** т. и т. т., когда их **совместный результат будет тем же самым**, что и при исполнении этих транзакций **в некотором последовательном порядке**.

- **Расписание совокупности транзакций – порядок** в котором выполняются **ЭЛЕМЕНТАРНЫЕ** шаги этих транзакций (блокировка, чтение, и т.д.).
- Расписание называется **последовательным**, если все шаги каждой транзакции выполняются вслед или перед всеми шагами других транзакций.
- Расписание называется **сериализуемым**, если его результат **ЭКВИВАЛЕНТЕН** результату некоторого последовательного расписания.

Вопрос 23

Выберите правильный вариант декомпозиции отношения R (A, B, C, D, E) при приведении его к 3-ей нормальной форме

A, B - ключ отношения

Имеются функциональные зависимости:

B → C

D → E

Выберите один ответ:

- R (A, B, C, D, E) разбивается на R1 (A, B, C), R2 (C, D, E)
- R (A, B, C, D, E) разбивается на R1 (B, C), R2 (A, B, D) и R3 (D, E)
- R (A, B, C, D, E) разбивается на R1 (A, B), R2 (C, D) и R3 (D, E)

Ответ

- R (A, B, C, D, E) разбивается на R1 (B, C), R2 (A, B, D) и R3 (D, E)

Пояснение

Здесь функциональная зависимость между B и C (где B - детерминант отношения, C - зависимая часть) и D и E (где D - детерминант отношения, E - зависимая часть).

A	B	C	D	E
	b1	c1		
b2	c2			
b3	c1		d1	e1

Отношения находятся в 3й НФ, если

3. Табл во 2 НФ
4. Все атрибуты зависят только от первичного ключа, а не от других атрибутов.

Получается, что тк есть зависимость между B & C , D & E, то что надо сделать с таблицей? Декомпозировать. А как?
Выделяем отдельно таблицу с B&C (тк есть функц зав), D&E (тк есть функц зав), остается A&B&D (A&B был первичным ключом - им и остается, + добавляется D, тк в изначальной таблице D был. ДОПИСАТЬ МБ ЧЕТ ЕЩЕ НАДО?

Вопрос 24

Отметьте способы встраивания SQL в языки программирования

Выберите один или несколько ответов:

- Dynamic SQL
- CRUD
- SQL/CLI
- Embedded SQL
- SOAP
- JDBC
- ODBC

Ответ

- Dynamic SQL
- SQL/CLI
- Embedded SQL
- ODBC

Пояснение

ODBC = Open Database Connectivity

Преза 13, сл 6

Разделы SQL99 (2)

- **SQL/CLI** (интерфейс уровня вызова) Определяет стандарт API (прикладной интерфейс), с помощью которого происходит вызов операций SQL.
 - ❖ Этот интерфейс разработан на основе старого стандарта SQL Access Group, на котором также основан стандарт ODBC, широко применяемый в промышленности.
 - ❖ В силу общего происхождения SQL/CLI и ODBC — это практически одно и то же. Таким образом, зная ODBC, можно с небольшими изменениями писать под интерфейс SQL/CLI.
 - ❖ Интерфейсы SQL/CLI и ODBC должны в будущем сближаться, а не расходиться.

Преза 9, сл 2-3

Создание приложений на SQL

SQL разработан исключительно для доступа к БД.

Типичная структура ПО при работе с БД – программа на универсальном языке программирования в тексте которой встроены операторы доступа к базе данных (вставка, удаление, редактирование, поиск)

Способы встраивания в ЯП

- Embedded SQL (встроенный)
- Dynamic SQL
- API (может содержать статический или динамический SQL)
 - ❖ API ориентирован на конкретную СУБД
 - ❖ API межплатформенный (cartridge)

Межплатформенные API

- SQL/CLI (SQL Call Level Interface) входит в стандарт SQL 99 – это API на языке C
- ODBC (Open Database Connectivity) – на момент создания – промышленный стандарт. API на языке C, но основан на более раннем стандарте чем SQL/CLI. Разновидность расширения SQL/CLI
- JDBC (Java Database Connectivity) – скопирован с ODBC, но для вызовов из программ на Java, а не на C. Промышленный стандарт, хотя и не утвержден ISO
- Процедурные расширения SQL. Выполняется самой СУБД, а не внешним приложением. 4GL (forth-generation language). SQL/PSM стандартный способ в SQL99
- Блоки SQL-кода (modules). Аналогичен embedded SQL, но один и тот же блок кода могут вызывать разные приложения.

Вопрос 25

Укажите способы решения проблемы тупиков

Выберите один или несколько ответов:

- Одновременная блокировка элементов БД, необходимых транзакции
- Требование двух-фазности транзакций
- Упорядоченная блокировка элементов БД, необходимых транзакции
- Рестарт транзакции из цикла графа ожиданий
- Упорядочение запуска транзакций
- Блокировка только изменяемых элементов БД, необходимых транзакции
- Завершение транзакции, попавшей в цикл графа ожиданий

Ответ

- Одновременная блокировка элементов БД, необходимых транзакции
- Упорядоченная блокировка элементов БД, необходимых транзакции
- Рестарт транзакции из цикла графа ожиданий

Пояснение

Преза 12, сл 31-34

Тупики – решение конфликта (1)

■ Вариант 1 (предотвращение).

Одновременная блокировка всех необходимых для транзакции элементов БД

Транзакция T_1

Lock (A, B);

...

UnLock (B);

UnLock (A);

Транзакция T_2

Lock (B, A);

...

UnLock (A);

UnLock (B);

Тупики: варианты решения проблемы (2)

2. Ввести ЛИНЕЙНОЕ упорядочение элементов БД, требовать запрос блокировок делать в соответствии с ЭТИМ порядком

Пусть на множестве элементов БД введен линейный порядок :
 $A < B < C \dots < Z$

Тогда ТР1 и ТР2 должны блокировать эл.БД в этом порядке

TP1	TP2
Lock A;	Lock A;
Lock B;	Lock B;
...	...
UnLock A;	UnLock B;
UnLock B;	UnLock A;

Тупики – решение конфликта (3)

■ Вариант 3 (устранение).

- Строим граф ожиданий.
- Вершины графа – транзакции.
- Дуга из T_1 в T_2 ($T_1 \rightarrow T_2$) означает, что T_1 ожидает выполнения запроса на блокировку элемента БД, заблокированного в данный момент T_2
- Каждый цикл указывает тупик

Анализируется построенный граф.

В случае обнаружения цикла **делается рестарт (Rollback)** одной из транзакций попавших в цикл.

Вопрос 26

Укажите операторы, используемые для описания ограничений целостности
Выберите один или несколько ответов:

- a. CREATE ASSERTION
- b. GRANT
- c. CHECK()
- d. PRIMARY KEY
- e. CREATE VIEW
- f. NON NULL

Ответ

- CREATE ASSERTION
- CHECK()
- PRIMARY KEY
- NON NULL (очепятка тут типа должно быть NOT NULL)

Пояснение

Преза 6, сл 20-21

Ограничение столбца

<ограничение столбца> ::=
[CONSTRAINT имя ограничения]
| {NOT NULL}
| { PRIMARY KEY }
| {UNIQUE}
| REFERENCES имя таблицы [(имя столбца)]
 [<ссылочная спецификация>]
| {CHECK предикат}
[[INITIALLY DEFERRED] | INITIALLY IMMEDIATE]
[[NOT] DEFERRABLE]

Пр 11 сл 2 хмхмхм

Ограничение таблицы

<ограничение таблицы> ::=
[CONSTRAINT имя ограничения]
{ PRIMARY KEY (имя столбца, ...)}
| {UNIQUE (имя столбца, ...)}
| FOREIGN KEY (имя столбца, ...)
REFERENCES имя таблицы [(имя столбца, ...)]
 [<ссылочная спецификация>]
| {CHECK предикат} [[NOT] DEFERRABLE]

Ограничения целостности

- на уровне колонки в операторе CREATE TABLE:
 - ❖ PRIMARY KEY,
 - ❖ REFERENCES,
 - ❖ NOT NULL,
 - ❖ UNIQUE,
 - ❖ CHECK (...).
- на уровне таблицы в операторе CREATE TABLE
 - ❖ PRIMARY KEY (...),
 - ❖ FOREIGN KEY (...),
 - ❖ UNIQUE (...),
 - ❖ CHECK (...).
- НА УРОВНЕ БАЗЫ ДАННЫХ
 - Assertion (ограничение общего вида, утверждения).
 - Trigger – последовательность команд, поставленная в соответствие определенному событию (операции(ям) модификации содержимого таблиц базы данных) и выполняемая в ответ на это событие.

Вопрос 27

Укажите операторы, используемые для работы с курсорами

Выберите один или несколько ответов:

- a. CLOSE CURSOR
- b. UPDATE CURSOR
- c. SELECT
- d. DECLARE CURSOR
- e. FETCH
- f. INSERT
- g. OPEN CURSOR
- h. DELETE CURSOR

Ответ

- CLOSE CURSOR
- DECLARE CURSOR
- FETCH
- OPEN CURSOR

Пояснение

Преза 9, сл 9, 11

Курсы

Манипулирование данными

- Для начала работы с курсором нужно выполнить оператор **OPEN**

EXEC SQL OPEN CURSOR Томск_продавцы;

- Для построчного извлечения данных - **FETCH**

EXEC SQL FETCH Томск_продавцы

INTO :id_num, :salesperson, :loc, :comm;

- В конце работы нужно выполнить оператор **CLOSE**

EXEC SQL CLOSE CURSOR Томск_продавцы;

Курсор – это особый временный объект SQL, предназначенный для использования в программах и хранимых процедурах. С его помощью можно в цикле пройти по результирующему набору строк запроса, по отдельности считывая и обрабатывая каждую его строку.

DECLARE CURSOR определяет набор данных, который может быть возвращен, при обращении к этому курсору. Это первая инструкция, из группы инструкций работы с курсором, которая должна быть использована в работе с ним.

Позиционированные операторы

- Для обновления курсора в операторах **DELETE** и **UPDATE** может использоваться фраза **WHERE CURRENT OF**, определяющая, что действие относится к текущей строке курсора. Такой оператор называется позиционированным, и к нему предъявляются следующие требования:

- ✓ курсор, и оператор должны использовать только одну и ту же таблицу;
- ✓ в запросе, используемом для создания курсора, не должно быть фраз **UNION** и **ORDER BY**;
- ✓ курсор должен удовлетворять критериям обновляемого курсора (например, не применять агрегирующие функции).

- Пример:

```
EXEC SQL DECLARE CURSOR комиссионные  
FOR SELECT sname, comm  
FROM Salespeople;  
EXEC SQL OPEN CURSOR комиссионные;  
EXEC SQL FETCH комиссионные INTO :salesperson, :comm;  
EXEC SQL
```

UPDATE Salespeople

SET comm = :comm *1.3

WHERE CURRENT OF комиссионные;

- Позиционированный оператор **DELETE** удобно использовать для удаления из таблицы группы строк, предварительно выбранных в курсор.

Вопрос 28

Отметьте предложения, формулирующие ограничения иерархической модели

Выберите один или несколько ответов:

- любой родительский сегмент может быть починенным
- каждый родительский сегмент может быть связан только с одним подчиненным сегментом
- каждый подчиненный сегмент может быть связан только с одним родительским сегментом
- каждый сегмент может быть связан с произвольным числом подчиненных сегментов
- в каждой физической БД существует один корневой сегмент
- корневой сегмент имеет связь только с одним родительским сегментом

Ответ

- каждый подчиненный сегмент может быть связан только с одним родительским сегментом
- каждый сегмент может быть связан с произвольным числом подчиненных сегментов
- в каждой физической БД существует один корневой сегмент

Пояснение

Преза 14 слайд 7

Свойства физической базы данных

Схема иерархической БД представляет собой **совокупность отдельных деревьев**

Каждое дерево в рамках модели называется **физической базой данных**.

Каждая физическая БД удовлетворяет следующим иерархическим ограничениям:

- ❖ в каждой физической БД существует один корневой сегмент, то есть сегмент, у которого нет логически исходного (родительского) типа сегмента;
- ❖ каждый логически исходный сегмент может быть связан с произвольным числом логически подчиненных сегментов;
- ❖ каждый логически подчиненный сегмент может быть связан только с одним логически исходным (родительским) сегментом.

Вопрос 29

Какая операция должна восстанавливать исходное отношение после декомпозиции.

Выберите один ответ:

- Декартово произведение
- Экви соединение
- Натуральное соединение
- Объединение

Ответ

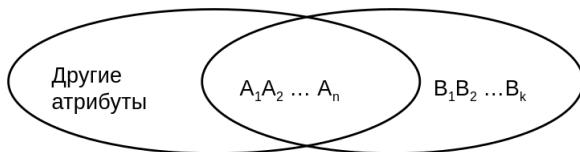
- Натуральное соединение

Пояснение

Преза 5 слайд 21

Приведение к нормальным формам

- Поиск в исходном отношении R нетривиальных функциональных зависимостей $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_k$, нарушающих нормальную форму.
- Разбиение отношения R на 2 отношения, например, T и S, таких, что



- T будет состоять из атрибутов, образующих нетривиальную ФЗ,
- S – из атрибутов отношения R, входящих в ключ и не вошедших в отношение T.
- Исходное отношение должно восстанавливаться из 2-ух получившихся отношений с помощью **НАТУРАЛЬНОГО СОЕДИНЕНИЯ**

Декомпозиция		
R(A, B)	R(A, C)	R(B, C)
1 2	1 2 3	2 3
3 2	3 2 6	2 6
5 4	5 4 2	4 2

Соединение		
R'(A, B, C)		
1 2 3	1 2 3	
1 2 6		2 6
3 2 3		4 2
3 2 6		
5 4 2		

Вопрос 31

Транзакция называется двухфазной, если _____

Выберите один ответ:

- a. все операции блокировки предшествуют всем операциям снятия блокировки
- b. операции блокировки предшествуют операциям снятия блокировки
- c. за всеми операциями блокировки следуют операции снятия блокировки
- d. все операции блокировки следуют за операциям снятия блокировки

Ответ

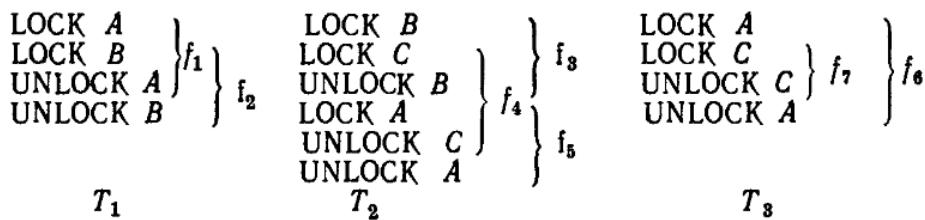
- все операции блокировки предшествуют всем операциям снятия блокировки

Пояснение

Преза 12 , сл 41

Двухфазные транзакции

- Транзакция называется **двуухфазной**, если все операции блокировки предшествуют всем операциям снятия блокировки



- **Теорема 2.** Любое расписание двухфазных транзакций – сериализуемо.

Вопрос 31

Продолжите определение

Отношением называется

- собственное подмножество декартового произведения доменов
- подмножество декартового произведения конечных доменов
- конечное подмножество декартового произведения доменов
- подмножество декартового произведения доменов
- Конечное подмножество декартового произведения конечных доменов

Ответ

- конечное подмножество декартового произведения доменов

Пояснение

Преза 3, сл 4

Отношением называется КОНЕЧНОЕ подмножество декартового произведения доменов $D_1 \times D_2 \times \dots \times D_n$

Вопрос 32

Выделите пункты, которые **не** содержат свойства ТРАНЗАКЦИИ

Выберите один или несколько ответов:

- Согласованность
- Уникальность
- Долговечность
- Надежность
- Завершенность
- Единственность
- Изолированность
- Автономность

Ответ

- Уникальность
- Надежность
- Завершенность
- Единственность

Пояснение

Преза 12, сл 13 ACID

Свойства транзакции

■ Автономность (Atomicity). Все или ничего!

Либо фиксируются все операции транзакции (commit), либо откат (rollback)

■ Согласованность (Consistency). Шире понятия целостности.

Согласованность может быть реализована на уровне бизнес логики. Например, операция « списание » должна соответствовать некоторой операции « зачисление »

■ Изолированность (Isolation).

При выполнении транзакций параллельно **одни** транзакции не должны оказывать влияние на результат выполнения **других** транзакций.

■ Долговечность (Durability).

Если пользователь получил подтверждение от системы, что транзакция выполнена, он может быть уверен, что сделанные им изменения не будут отменены из-за какого-либо сбоя.

Вопрос 33

Пусть отношение R содержит следующий набор кортежей

R (A, B, C, D)
2 3 5 6
3 4 7 8

2 3 5 7
3 6 7 8

Отметьте функциональные зависимости атрибутов в этом отношении
Выберите один или несколько ответов:

- C → D
- A → C
- A → D
- B → C
- A, B → C
- A, C → D

Ответ

- A → C
- B → C
- A, B → C

Пояснение

Посмотреть можно [тут](#).

Вопрос 34

Способ решения проблемы бесконечных ожиданий транзакций?

Выберите один ответ:

- Выполнить рестарт одной транзакции
- Выполнять транзакции по очереди
- Организовать выполнение транзакций через СТЭК
- Сортировать транзакции
- Выполнить откат одной транзакции

Ответ

- Выполнять транзакции по очереди

Пояснение

Преза 12 , сл 29

-  **Бесконечные ожидания**
- Пусть T₁, находится в ожидании блокировки элемента A и еще T₂ запрашивает блокировку A.
 - Рассмотрим ситуацию, когда T₂ получает блокировку раньше T₁.
 - Далее, в то время когда установила T₂ блокировку A, еще T₃ запросила блокировку A.
 - Если T₃ получил блокировку A сразу за разблокированием A тран. T₂, то T₁ все еще будет находиться в ожидании блокировки.
- И, таким образом, некоторые транзакции могут находиться в **бесконечном ожидании**

Решение – очередь транзакций

Вопрос 35

Отметьте транзакции, которые СУБД выполнит без сообщения об ошибке.

Отношения

Сотрудники (таблом int primary key,
ФИО char (45),
должность int reference Должности
);

Должности (ном int primary key,
назв char (25)
);

Выберите один или несколько ответов:

- ```
Insert into ДОЛЖНОСТИ values (1, 'директор');
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);
```
- ```
Insert into ДОЛЖНОСТИ values (1, 'директор');  
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');  
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);  
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);
```
- ```
Insert into ДОЛЖНОСТИ values (1, 'директор');
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);
```
- ```
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);  
Insert into ДОЛЖНОСТИ values (1, 'директор');  
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');
```
- ```
Insert into ДОЛЖНОСТИ values (1, 'директор');
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);
```

- Insert into ДОЛЖНОСТИ values (1, 'директор');  
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');  
Insert into СОТРУДНИКИ values (1003, 'Лебедев А.А', 3);  
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);

Ответ

- Insert into ДОЛЖНОСТИ values (1, 'директор');  
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');  
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);  
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);
- Insert into ДОЛЖНОСТИ values (1, 'директор');  
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');  
Insert into СОТРУДНИКИ values (1001, 'Соколов И.И', 1);  
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);
- Insert into ДОЛЖНОСТИ values (1, 'директор');  
Insert into ДОЛЖНОСТИ values (2, 'зам. директора');  
Insert into СОТРУДНИКИ values (1002, 'Сидоров В.А', 2);

Пояснение

В правильных вариантах ответов всё норм (капитан очевидность) по добавлению данных в таблицу. С последними 3я вариантами проблемы: добавляются данные, которых еще нет (но которые потом хотят добавить. Но получают по носу. Мне лень расписывать.

### Вопрос 36

Отметьте основные информационные единицы в иерархической модели

Выберите один или несколько ответов:

- Запись
- Поле
- Индекс
- Раздел
- Сегмент
- База данных
- Файл данных

Ответ

- Поле
- Сегмент
- База данных

Пояснение  
Преза 14 , сл 3

## Модель данных

- В реальном мире очень многие связи между информационными объектами соответствуют иерархии
- Один объект выступает как родительский, а с ним может быть связано множество подчиненных объектов.
- Иерархия проста и естественна в отображении взаимосвязи между классами объектов.
- Основными информационными единицами в иерархической модели являются:
  - ❖ *База данных*
  - ❖ *Сегмент*
  - ❖ *Поле (элемент данных)*

### Вопрос 37

В какой наибольшей нормальной форме находится отношение R (A, B, C, D, E),

если A, B - ключ отношения и

имеются следующие функциональные зависимости:

$C \rightarrow D$

$D \rightarrow E$

Выберите один ответ:

- в нормальной форме Бойса-Кодда
- в первой НФ
- во второй НФ
- в третьей НФ

Ответ

- во второй НФ

Пояснение

По определению

1 НФ:

- в каждой клетке по 1 значению - да
- нет повторяющихся строк - да

2 НФ:

- табл в 1 НФ - да
- есть первичный ключ - да
- каждый атрибут зависит от первичного ключа ЦЕЛИКОМ, а не от какой-то его части - да

3 НФ:

- табл во 2й НФ - да
- все атрибуты зависят только от первичного ключа, но не от других атрибутов - НЕТ (тк тут зависимости С -> D, D -> E - зависимость между атрибутами не состоящих в первичном ключе)

### Вопрос 38

Отметьте этапы разработки базы данных в информационной системе

Выберите один или несколько ответов:

- Программирование
- Планирование
- Проектирование
- Реализация
- Тестирование

Ответ

- Планирование
- Проектирование
- Реализация

Пояснение

Преза 1, сл 13

■ **Этапы разработки базы данных в информационной системе**

- **Проектирование базы данных**
  - Логическое проектирование
  - Физическое проектирование
- **Планирование базы данных**
  - Аппаратных средств
  - Программных средств
- **Реализация базы данных**
- **Сопровождение**
- **Реинжиниринг**

## Вопрос 39

Сколько видов триггеров можно определить на одну таблицу

Выберите один ответ:

- 9
- 15
- 6
- 12
- 10

Ответ

- 12

Пояснение [тык](#)

| Когда      | Событие              | На уровне строк             | На уровне оператора                        |
|------------|----------------------|-----------------------------|--------------------------------------------|
| BEFORE     | INSERT/UPDATE/DELETE | Таблицы и сторонние таблицы | Таблицы, представления и сторонние таблицы |
|            | TRUNCATE             | —                           | Таблицы                                    |
| AFTER      | INSERT/UPDATE/DELETE | Таблицы и сторонние таблицы | Таблицы, представления и сторонние таблицы |
|            | TRUNCATE             | —                           | Таблицы                                    |
| INSTEAD OF | INSERT/UPDATE/DELETE | Представления               | —                                          |
|            | TRUNCATE             | —                           | —                                          |

## Вопрос 40 ЧЕГО БЛИН

Пусть имеются отношения R, S и F.

Пусть имеются отношения R, S и F.

$\{ t \mid R(t) \}$  – множество кортежей отношения R,

$\{ t \mid S(t) \}$  – множество кортежей отношения S,

$\{ t \mid F(t) \}$  – множество кортежей отношения F.

Отметьте безопасные выражения реляционного исчисления на кортежах эквивалентные следующему выражению реляционной алгебры:  $R \cap S \setminus F$

Выберите один или несколько ответов:

- { t | S(t) v R(t) & F(t) }
- { t | S(t) & R(t) & ¬ F(t) }
- { t | S(t) v R(t) & ¬ F(t) }
- { t | R(t) & S(t) & ¬ F(t) }
- { t | S(t) & R(t) & F(t) }
- { t | R(t) & S(t) v ¬ F(t) }

Ответ

- { t | S(t) & R(t) & ¬ F(t) }
- { t | R(t) & S(t) & ¬ F(t) }

Пояснение  
Преза , сл

## ТЕСТ 4

### Вопрос 1

За какое количество обращений к диску можно найти запись в файле с организацией в виде "B-дерева" , если известно, что

1. основной файл состоит из  $n = 70\ 000$  записей,
2. блок основного файла содержит  $e = 7$  записей,
3. блок индексного файла содержит  $d = 10$  записей

Ответ

- 4

Пояснение

Преза 20, сл 23

Считаем вот так:

$$1 + \log_{10}(70\ 000 / 7) = 1 + \log_{10}(10\ 000) = 1 + 3 = 4$$

## ■ Временные характеристики B-дерева

- Рассмотрим файл с  $n$  записями, организованный в виде *B*-дерева с параметрами  $d$  – количество записей в блоке индекса и  $e$  - количество записей в блоке основного файла.
- В этом дереве не более  $n/e$  листьев, не более  $n/de$  отцов листьев  $n/d^2e$  отцов отцов листьев и т. д.
- Если на путях от корня к листьям имеется  $i$  узлов, то  $n \geq d^{i-1}e$ , так как в противном случае было бы менее одного узла на уровне корня, что невозможно.

- Тогда 
$$i \leq 1 + \frac{\log_2(\frac{n}{e})}{\log_2 d} = 1 + \log_d(\frac{n}{e})$$
- Чтобы выполнить поиск, достаточно  $i$  операций чтения.
- Для операций включения, удаления или модификации обычно необходимо записать лишь один блок — лист, содержащий участвующую в операции запись.
- Итого число операций чтения/записи =  $2 + \log_d(n/e)$

23

### Вопрос 2

Отметьте термины, относящиеся к логическим компонентам вычислительной системы

Выберите один или несколько ответов:

- Операционная система
- Процессор
- Файловая система
- Компилятор
- Виртуальная память
- Диспетчер логических томов
- Система управления базами данных

Ответ

- Операционная система
- Файловая система
- Виртуальная память
- Диспетчер логических томов

Пояснение

Преза 19 цод 3, сл 16

## Что такое вычислительная система?

- Вычислительная платформа (оборудование и системное программное обеспечение), на которой запускаются приложения
  - К физическим компонентам относятся процессор, память, внутренняя система хранения и устройства ввода-вывода.
  - К логическим компонентам относятся ОС, драйверы устройств, файловая система и диспетчер логических томов

## Вопрос 3

Отметьте набор характеристик, свойственных RAID 1+0

Выберите один или несколько ответов:

- Имеется диск четности
- Для каждой операции записи нового блока нужно 2 обращения к дискам
- Для каждой операции записи нового блока нужно 4 обращения к дискам
- Массив восстанавливается при потере 1 диска
- Минимальное количество дисков = 3
- Минимальное количество дисков = 2
- Массив восстанавливается при потере 2 дисков (из одной половины зеркала)
- Нет диска четности
- Минимальное количество дисков = 4

### Ответ

- Для каждой операции записи нового блока нужно 4 обращения к дискам
- Массив восстанавливается при потере 1 диска
- Массив восстанавливается при потере 2 дисков (из одной половины зеркала)
- Нет диска четности
- Минимальное количество дисков = 4

### Пояснение

По [видосу](#) кайф понятно

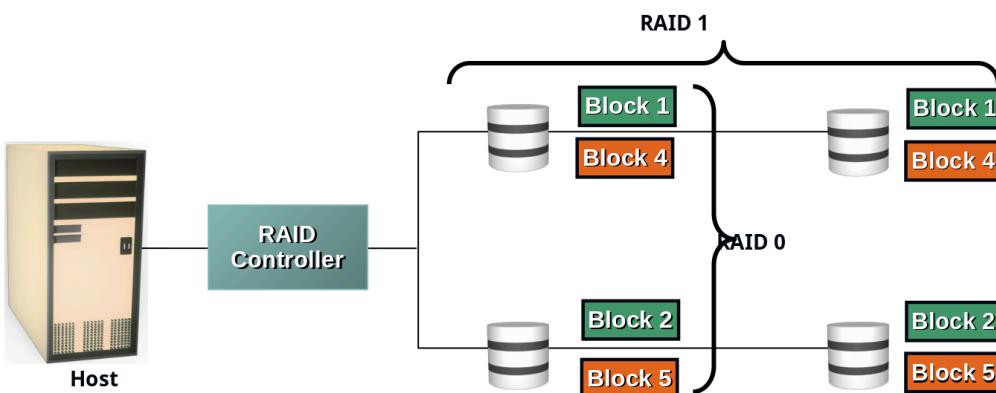
Преза 23 , сл 36

## Сравнение массивов RAID

| Уровень RAID | Мин. кол-во дисков | Доступная емкость ресурсов хранения (%) | Дополнительные издержки записи | Защита данных                                        |
|--------------|--------------------|-----------------------------------------|--------------------------------|------------------------------------------------------|
| 1            | 2                  | 50                                      | 2                              | Зеркало                                              |
| 1+0          | 4                  | 50                                      | 2                              | Зеркало                                              |
| 3            | 3                  | $[(n-1)/n]*100$                         | 4                              | Контроль четности<br>(поддерживает один сбой дисков) |
| 5            | 3                  | $[(n-1)/n]*100$                         | 4                              | Контроль четности<br>(поддерживает один сбой дисков) |
| 6            | 4                  | $[(n-2)/n]*100$                         | 6                              | Контроль четности<br>(поддерживает два сбоя дисков)  |

Преза 23, сл 17

## Блочный RAID – 0+1 (Striping and Mirroring)



Технология сочетает :

- ❖ производительность RAID 0
- ❖ избыточность RAID 1

Требуют четного количества дисков – минимум -4

Защита данных: RAID

- 17

През 23, сл 31

## Сравнение RAID'ов

| RAID | Мин дисков | Эффект хранения данных, %             | Стоимость | Скорость считывания                                                                        | Скорость записи                                                                                                   | Дополнительные расходы на запись |
|------|------------|---------------------------------------|-----------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------------|
| 0    | 2          | 100                                   | Низкая    | Отличная как при произвольном так и последовательном считывании                            | Отличная                                                                                                          | Нет                              |
| 1    | 2          | 50                                    | Высокая   | Хорошая. Лучше считывания с одного диска                                                   | Хорошая. Медленнее, чем на единичном диске, поскольку каждая запись производится на несколько дисков              | Приемлемые                       |
| 3    | 3          | (n-1)*100/n где n = количество дисков | Средняя   | Хорошая при произвольном и отличная последовательном считывании                            | Плохая или приемлемая при небольших записях в произвольном порядке. Хорошая при объемных последовательных записях | Большие                          |
| 4    | 3          | (n-1)*100/n где n = количество дисков | Средняя   | Отличная при произвольном считывании. Хорошая или отличная при последовательном считывании | Плохая или приемлемая при произвольных записях. Хорошая при последовательных записях.                             | Большие                          |

## Сравнение RAID'ов (2)

| Мин RAID дисков | Эффект хранения данных, % | Стоимость                             | Скорость считывания           | Скорость записи                                                           | Дополнительные расходы на запись                                                                                                                                |               |
|-----------------|---------------------------|---------------------------------------|-------------------------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 5               | 3                         | (n-1)*100/n где N = количеству дисков | Средняя                       | Очень хорошая для случного чтения<br>Хорошая для последовательного чтения | Приемлемая при произвольных записях. Замедление объясняется дополнительными расходами на запись четности.<br>Приемлемая или хорошая при последовательной записи | Большие       |
| 6               | 4                         | (n-2)*100/n где N = количеству дисков | Средняя, но больше чем RAID 5 | Очень хорошая для случного чтения<br>Хорошая для последовательного чтения | Хорошая для небольших произвольных записей (бывают необходимы дополнительные затраты на запись)                                                                 | Очень большие |
| 1+0 и 0+1       | 4                         | 50                                    | Высокая                       | Отличная                                                                  | Хорошая                                                                                                                                                         | Умеренные     |

#### Вопрос 4

Отметьте этапы оптимизации запросов

Выберите один или несколько ответов:

- Генерация планов вычисления запроса и выбор плана с наименьшими затратами
- Преобразование запроса во внутреннюю форму
- Сокращение операций ввода-вывода
- Сокращение объема используемой оперативной памяти
- Выбор потенциальных низкоуровневых процедур
- Сокращение времени выполнения запроса
- Преобразование в каноническую форму

Ответ

- Генерация планов вычисления запроса и выбор плана с наименьшими затратами
- Преобразование запроса во внутреннюю форму
- Выбор потенциальных низкоуровневых процедур
- Преобразование в каноническую форму

Пояснение

Преза 22, сл 7

## Стадии оптимизации

1. Преобразование запроса во внутреннюю форму
2. Преобразование в каноническую форму
3. Выбор потенциальных низкоуровневых процедур
4. Генерация планов вычисления запроса и выбор плана с наименьшими затратами

#### Вопрос 5

Отметьте базовые понятия модели внешней памяти

Выберите один или несколько ответов:

- сегмент
- участок
- запись
- поле
- сектор
- блок
- файл
- ключ

Ответ

- файл
- блок
- запись
- поле
- ключ

Пояснение

Преза 20, сл 6-7

### Модель организации внешней памяти

Физические блоки – секторы диска,  $2^9 - 2^{12}$  байт

Физический блок (блок) имеет адрес на диске.

Файл – хранится в одном или более блоках.

Файл состоит из записей.

В каждом блоке может храниться ЦЕЛОЕ количество записей.

Запись имеет адрес:

- ❖ абсолютный адрес ее первого байта,
- ❖ адрес блока + смещение (число байтов предшествующих началу записи)

### Модель организации внешней памяти

ЗАПИСЬ состоит из **полей**. Поля имеют тип и размер.

Выделенное поле – **ключ**.

Оценка скорости выполнения операции над записью – количество считанных блоков для выполнения операции

Операции – **поиск, вставка, удаление, редактирование**

## Вопрос 6

Какую структуру называют вторичным индексом по некоторому полю записи?

Выберите один ответ:

- Структура, устанавливающая связь между ключевым доменом и доменом, не являющимся ключем
- Структура, содержащая индекс по второму ключевому полю
- Структура, устанавливающая связь между значениями домена и множеством записей файла
- Структура, устанавливающая связь между полем записи и другими записями файла.

Ответ

- Структура, устанавливающая связь между значениями домена и множеством записей файла

Пояснение

Преза 21, сл 8

## Вторичные индексы

- Пусть имеется файл ОФ, записи которого содержат некоторое поле  $F$ .
- Пусть это поле принимает возможные значения из множества значений  $D$ , домена для  $F$ .
- Поле  $F$  может быть элементом множества полей, которое *образует ключ*, или оно может в ключ не входить

**Вторичный индекс** по полю  $F$  представляет собой связь между доменом  $D$  и множеством записей рассматриваемого файла.

### Вопрос 7

Отметьте способы хранения записей с переменной длинной.

Выберите один или несколько ответов:

- Метод зарезервированного пространства
- Метод списков
- Хэш таблица
- Метод списков с обратным указателем
- Комбинированный метод
- Метод указателей
- Метод выделенной памяти

Ответ

- Метод зарезервированного пространства
- Комбинированный метод
- Метод указателей

Пояснение

Преза 21, сл 4 - 7

## Хранение записей переменной длины

### Метод зарезервированного пространства

Если количество повторений известно, то можно перейти к записям с фиксированной длиной.

#### Пример

Область Население (Шоссе Длина (Терминал1 Терминал2))\*

или

Область Население счетчик

Шоссе1 Длина1 Терминал1.1 Терминал1.2

...

Шоссе10 Длина10 Терминал10.1 Терминал10.2

Хорошо применять когда **среднее количество** повторяющихся элементов в списке **близко к максимальному значению** количества таких элементов для разных объектов.

## Метод указателей

### Пример

#### Область Население (Шоссе Длина (Терминал)\*)\*

Будет представлено 3 файлами с блоками фикс. длины:

1. Область Население Указатель-Шоссе

2. Шоссе Длина Указатель-Терминалов

3. Терминал

■ Оптимальный метод по занимаемому пространству (памяти).

■ Может проигрывать по скорости доступа

## Комбинированный метод

### Пример

#### Область Население (Шоссе Длина (Терминал))\*\*

Будет представлено 2 файлами с блоками фикс. длины:

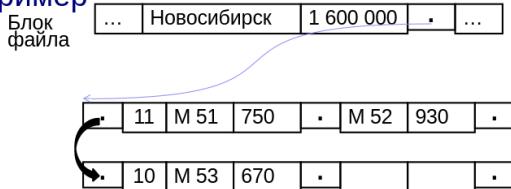
1. Область Население Указатель-Шоссе

2. Шоссе Длина Терминал 1, Терминал 2

■ Позволяет использовать оптимальные характеристики предыдущих методов

## Метод указателей

### Пример



## Вопрос 8

Отметьте методы повышения эффективности дисковых операций

Выберите один или несколько ответов:

- Использование нескольких дисковых устройств
- Использование стэка для дисковых операций
- Применение алгоритма лифта
- Создание зеркальных копий дисков
- Предварительное считывание
- Группирование данных по цилиндрям диска
- Применение алгоритмов сортировки

Ответ

- Использование нескольких дисковых устройств
- Применение алгоритма лифта
- Создание зеркальных копий дисков
- Предварительное считывание
- Группирование данных по цилиндрям диска

Пояснение

Преза 24 эдо, сл 14

## Повышение эффективности дисковых операций

1. Разместить блоки, которые должны обрабатываться совместно на одном цилиндре
2. Расположить информацию на нескольких дисках меньшего объема
3. Создать "зеркальные" копии одних и тех же данных на нескольких дисках
4. Использовать алгоритмы упорядочения запросов к блокам диска во времени
5. Применить стратегии предварительного считывания

### Вопрос 9

Какой алгоритм используется при поиске записи в файле "Разреженный индекс" ?

Ответ

- дихотомия

Пояснение

Ну чтоб хоть чот было  
Преза 20, слайд 12

## Индексный файл

- ❖ Разреженный индекс
- ❖ Плотный индекс

### Разреженный индекс - индекс

Состоит из 2 типов файлов:

- Основной файл, содержащий записи с информацией о предметной области
- Файл индекса, содержащий пары < ключ, адрес >

**Ключ** – значение ключа записи, представителя блока (старший ключ)

**Адрес** – адрес блока файла, содержащий первую запись с соответствующим значением ключа

## Вопрос 10

Чем характеризуется программное обеспечение как услуга ?

Выберите один или несколько ответов:

- Пользователь может управлять только следующими элементами базовой облачной инфраструктурой: сеть, серверы, операционные системы, системы хранения данных
- Пользователь может ограниченно влиять на пользовательские настройки конфигурации приложений
- Пользователь не может управлять базовой облачной инфраструктурой
- Приложения, предоставляемые по модели SaaS, работают в облаке и обычно не требуют установки на конечных устройствах.

Ответ

- Пользователь может ограниченно влиять на пользовательские настройки конфигурации приложений
- Пользователь не может управлять базовой облачной инфраструктурой
- Приложения, предоставляемые по модели SaaS, работают в облаке и обычно не требуют установки на конечных устройствах.

Пояснение

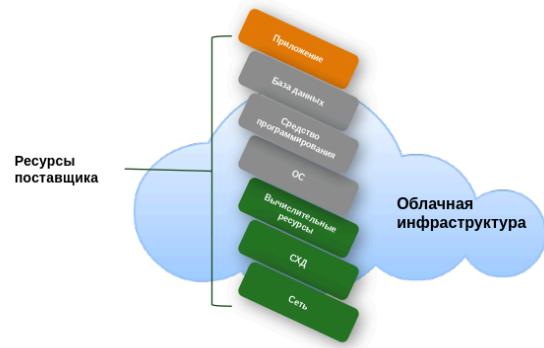
Преза 19 цод 2 , сл 10

# Программное обеспечение как услуга

## Программное обеспечение как услуга

Пользователь может использовать приложения поставщика, которые работают на базе облачной инфраструктуры. Приложения доступны с различных клиентских устройств через интерфейс «тонкого» клиента, например веб-браузера (электронная почта на основе веб-интерфейса), или через программный интерфейс.

Пользователь не может управлять базовой облачной инфраструктурой, включая сеть, серверы, операционные системы, системы хранения данных или даже отдельные функции приложений, однако может ограниченно влиять на пользовательские настройки конфигурации приложений.



Из пояснения к тому же слайду:

“Модель «программное обеспечение как услуга» «Пользователь может использовать приложения поставщика, которые работают на базе облачной инфраструктуры.

Приложения доступны с различных клиентских устройств через интерфейс «тонкого» клиента, например веб-браузер (электронная почта на основе веб-интерфейса), или через программный интерфейс. Пользователь не может управлять базовой облачной инфраструктурой, включая сеть, серверы, операционные системы, системы хранения данных или даже отдельные функции приложений, однако может ограниченно влиять на пользовательские настройки конфигурации приложений», — NIST

В модели «ПО как услуга» (SaaS) поставщик предоставляет нескольким пользователям приложение, размещенное в облаке, в качестве услуги.

Пользователи не владеют и не управляют ни одним из аспектов облачной инфраструктуры.

В модели SaaS конкретная версия приложения с определенной конфигурацией (оборудования и ПО) обычно предоставляет услуги нескольким пользователям путем разделения их отдельных сессий и данных.

Приложения, предоставляемые по модели SaaS, работают в облаке и обычно не требуют установки на конечных устройствах. Это позволяет пользователю получать доступ к приложению по требованию из любого местоположения и использовать его с помощью веб-браузера на различных конечных устройствах.

Для некоторых приложений, предоставляемых по модели SaaS, может требоваться локальная установка клиентского интерфейса на конечном устройстве. В качестве примеров приложений, предоставляемых по модели SaaS, можно привести приложения для управления отношениями с заказчиками (CRM), электронной почты и планирования ресурсов предприятия (ERP), а также офисные пакеты.”

## Вопрос 11

Выберите оптимальный вариант реализации запроса

Select B,C  
FROM R, S

WHERE A>100 AND C=D

где R и S - отношения, R(A,B), S(C,D)

Выберите один ответ:

$\pi_{B,C}(\sigma_{A>100}(R) \times \sigma_{C=D}(S))$

$\pi_{B,C}(\sigma_{A>100}(R) \times \sigma_{C=D}(S))$

$\pi_{B,C}(\sigma_{A>100}(R)) \times \pi_{B,C}(\sigma_{C=D}(S))$

$\pi_{B,C}(\sigma_{A>100 \text{ AND } C=D}(R \times S))$



Ответ

$\pi_{B,C}(\sigma_{A>100}(R) \times \sigma_{C=D}(S))$



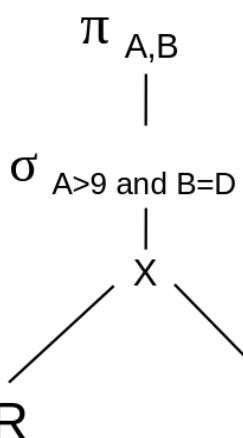
Пояснение

Преза 22, сл 9

### Пример

a.  $\pi_{A,B}(\sigma_{A>9 \text{ and } B=D}(R \times S))$

6.



### Вопрос 12

Отметьте набор характеристик, которые **не** соответствуют RAID 5.

Выберите один или несколько ответов:

- Блоки четности распределены по дискам
- Имеется выделенный диск четности
- Для каждой операции записи нового блока нужно 4 обращения к дискам
- Массив восстанавливается при потере 2 дисков
- Для каждой операции записи нового блока нужно 2 обращения к дискам
- Минимальное количество дисков = 4
- Минимальное количество дисков = 3
- Массив восстанавливается при потере 1 диска
- Минимальное количество дисков = 2

Ответ

- Имеется выделенный диск четности
- Массив восстанавливается при потере 2 дисков
- Для каждой операции записи нового блока нужно 2 обращения к дискам
- Минимальное количество дисков = 4
- Минимальное количество дисков = 2

Пояснение

По [видосу](#) кайф понятно + см что написано в "[про raid](#)"

Соответствуют RAID5:

- Минимальное количество дисков = 3
- Массив восстанавливается при потере 1 диска
- Блоки четности распределены по дискам
- Для каждой операции записи нового блока нужно 4 обращения к дискам

Преза 23, сл 36

## Сравнение массивов RAID

| Уровень RAID | Мин. кол-во дисков | Доступная емкость ресурсов хранения (%) | Дополнительные издержки записи | Защита данных                                        |
|--------------|--------------------|-----------------------------------------|--------------------------------|------------------------------------------------------|
| 1            | 2                  | 50                                      | 2                              | Зеркало                                              |
| 1+0          | 4                  | 50                                      | 2                              | Зеркало                                              |
| 3            | 3                  | $[(n-1)/n]*100$                         | 4                              | Контроль четности<br>(поддерживает один сбой дисков) |
| 5            | 3                  | $[(n-1)/n]*100$                         | 4                              | Контроль четности<br>(поддерживает один сбой дисков) |
| 6            | 4                  | $[(n-2)/n]*100$                         | 6                              | Контроль четности<br>(поддерживает два сбоя дисков)  |

През 23, сл 31

## Сравнение RAID'ов

| RAID | Мин дисков | Эффект хранения данных, %                | Стоимость | Скорость считывания                                                                           | Скорость записи                                                                                                      | Дополнительные расходы на запись |
|------|------------|------------------------------------------|-----------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------|
| 0    | 2          | 100                                      | Низкая    | Отличная как при произвольном так и последовательном считывании                               | Отличная                                                                                                             | Нет                              |
| 1    | 2          | 50                                       | Высокая   | Хорошая.<br>Лучше считывания с одного диска                                                   | Хорошая.<br>Медленнее, чем на единичном диске, поскольку каждая запись производится на несколько дисков              | Приемлемые                       |
| 3    | 3          | (n-1)*100/n<br>где n = количество дисков | Средняя   | Хорошая при произвольном и отличная последовательном считывании                               | Плохая или приемлемая при небольших записях в произвольном порядке.<br>Хорошая при объемных последовательных записях | Большие                          |
| 4    | 3          | (n-1)*100/n<br>где n = количество дисков | Средняя   | Отличная при произвольном считывании.<br>Хорошая или отличная при последовательном считывании | Плохая или приемлемая при произвольных записях.<br>Хорошая при последовательных записях.                             | Большие                          |

## Сравнение RAID'ов (2)

| Мин RAID дисков | Эффект хранения данных, %                     | Стоимость                     | Скорость считывания                                                       | Скорость записи                                                                                                                                                    | Дополнительные расходы на запись |
|-----------------|-----------------------------------------------|-------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|
| 5               | 3<br>(n-1)*100/n<br>где N = количеству дисков | Средняя                       | Очень хорошая для случного чтения<br>Хорошая для последовательного чтения | Приемлемая при произвольных записях.<br>Замедление объясняется дополнительными расходами на запись четности.<br>Приемлемая или хорошая при последовательной записи | Большие                          |
| 6               | 4<br>(n-2)*100/n<br>где N = количеству дисков | Средняя, но больше чем RAID 5 | Очень хорошая для случного чтения<br>Хорошая для последовательного чтения | Хорошая для небольших произвольных записей (бывают необходимы дополнительные затраты на запись)                                                                    | Очень большие                    |
| 1+0 и 0+1       | 4                                             | 50                            | Высокая                                                                   | Отличная                                                                                                                                                           | Хорошая                          |

## Контроль четности и производительность диска

Пусть в RAID массиве на диске  $D_4$  обновился блок  $E_4$ .

Тогда RAID контроллер выполняет следующий набор операций:



Общий расчет накладных считываний для поддержки RAID 3 и RAID 4 с выделенным диском контроля четности.

Формула расчета контроля четности в контроллере

$$E_p = E_1 + E_2 + E_3 + E_4 \text{ (операции XOR)}$$

- Чтобы сохранить четность:  $E_{p \text{ new}} = E_{p \text{ old}} - E_{4 \text{ old}} + E_{4 \text{ new}}$  (делается 4 операции)
  - Считываются блоки с 2 дисков ( $E_{4 \text{ old}}$  с  $D_4$ ,  $E_{p \text{ old}}$  с  $P_0$ )
  - Записываются блоки на 2 диска ( $E_{4 \text{ new}}$  на  $D_4$ ,  $E_{p \text{ new}}$  на  $P_0$ )
- Это присуще RAID 3, RAID 4, RAID 5, RAID 6
- При этом ввод/выводов затрагиваются 2 диска

### Вопрос 13

За какое максимальное количество обращений к диску можно найти запись в индексном файле, если известно, что файл содержит 64 блока

Ответ

- 6

Пояснение

Преза 20, сл 14, 15

Это похоже на бинарный поиск, который ищет за  $\log_2(N) = \log_2(64) = 6$  ( типа вопрос не про доступ к записи, а именно про её поиск)

## Операции. Поиск.

Пусть требуется найти запись с ключем  $k_1$ . Тогда возникает задача найти блок В с этой записью.

Мы знаем, что в индексе есть пары  $\langle k, B \rangle$ , отсортированные по  $k$ .

Следовательно, будем искать в индексе запись  $\langle k_2, B \rangle$ , такую, что  $k_2 \leq k_1$  и либо  $\langle k_2, B \rangle$  последняя запись в блоке, либо следующая запись  $\langle k_3, B \rangle$  удовлетворяет условию  $k_1 < k_3$ .

## Индексный файл. Свойства.

- Блоки файла фиксированной длины
- Записи файла индекса не закреплены
- Блоки основного файла и файла индекса упорядочены по ключу
- Доступ по сортированному индексу
- Оценка доступа –  $\log_2(N) + 3$ , где  $N$  – количество блоков в файле

## Вопрос 14

Поставьте в соответствие соединения и протокол, используемый для этого соединения

- |                                                                                    |
|------------------------------------------------------------------------------------|
| • Высокоскоростное соединение между вычислительной системой и системой хранения —— |
| • Соединение CPU -> RAM -----                                                      |
| • Соединение для подключения жестких и оптических дисков -----                     |

Ответ

- |                                                                                       |
|---------------------------------------------------------------------------------------|
| • Высокоскоростное соединение между вычислительной системой и системой хранения —— FC |
| • Соединение CPU -> RAM ----- PCI                                                     |
| • Соединение для подключения жестких и оптических дисков ----- IDE/ATA                |

Пояснение

Преза 19 цод3 , сл 48

## Протоколы подключения систем хранения

| Протокол   | Описание                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IDE/ATA    | <ul style="list-style-type: none"><li>• Популярный интерфейс, используемый для подключения жестких и оптических дисков</li><li>• Версия Ultra DMA/133 протокола ATA поддерживает пропускную способность 133 Мбайт/с</li></ul>                                                                                                                                                                                  |
| Serial ATA | <ul style="list-style-type: none"><li>• Последовательная версия спецификации IDE/ATA обычно используется для внутренних подключений</li><li>• Обеспечивает скорость передачи данных до 16 Гбит/с (стандарт 3.2)</li></ul>                                                                                                                                                                                      |
| SCSI       | <ul style="list-style-type: none"><li>• Популярный стандарт, используемый для подключения вычислительной системы к системе хранения</li><li>• Поддерживает до 16 устройств на однойшине</li><li>• Версия Ultra-640 обеспечивает скорость передачи данных до 640 Мбайт/с</li></ul>                                                                                                                              |
| SAS        | <ul style="list-style-type: none"><li>• Последовательный протокол «точка-точка», заменяющий параллельный протокол SCSI</li><li>• Поддерживает скорость передачи данных до 12 Гбит/с (SAS 3.0)</li></ul>                                                                                                                                                                                                        |
| FC         | <ul style="list-style-type: none"><li>• Широко используемый протокол для высокоскоростного обмена данными между вычислительной системой и системой хранения</li><li>• Обеспечивает последовательную передачу данных, осуществляющуюся по медному и/или волоконно-оптическому кабелю</li><li>• Последняя версия интерфейса Fibre Channel «16FC» позволяет передавать данные со скоростью до 16 Гбит/с</li></ul> |
| IP         | <ul style="list-style-type: none"><li>• Существующая сеть на основе протокола IP используется для обмена данными между системами хранения</li><li>• Примеры: протоколы iSCSI и FCIP</li></ul>                                                                                                                                                                                                                  |

С того же слайда:

“Integrated Device Electronics (IDE)/Advanced Technology Attachment (ATA) — это популярный стандартный протокол обмена данными, используемый для подключения устройств хранения данных (например, дисковых накопителей и оптических дисководов). Этот протокол поддерживает параллельную передачу данных и в связи с этим известен также как Parallel ATA (PATA), или просто ATA. Протокол IDE/ATA имеет несколько различных стандартов и названий. Ultra DMA/133-версия протокола ATA поддерживает пропускную способность 133 Мбайт/с. В конфигурации «ведущий-ведомый» интерфейс ATA поддерживает два устройства хранения данных, подключенных к одному коннектору. Тем не менее, если важно обеспечить высокую производительность дискового накопителя, не рекомендуется совместное использование одного порта двумя устройствами.

Последовательная версия этого протокола поддерживает однобитную последовательную передачу и известна под названием Serial ATA (SATA).

Высокопроизводительный и недорогой протокол SATA в большинстве случаев заменил PATA в новых системах. Версия 3.2 протокола SATA обеспечивает скорость передачи данных до 16 Гбит/с.

SCSI возник как предпочтительный протокол для подключения в вычислительных системах высшего класса. Этот протокол поддерживает параллельную передачу и предлагает улучшенную производительность, масштабируемость и совместимость по сравнению с ATA. Однако высокая стоимость систем, в которых используется SCSI, ограничивает его популярность среди домашних пользователей или пользователей настольных ПК. С годами протокол SCSI был усовершенствован и теперь включает широкий круг связанных с ним технологий и стандартов. SCSI поддерживает до 16 устройств на однойшине и обеспечивает скорость передачи данных до 640 Мбайт/с (для версии Ultra-640).

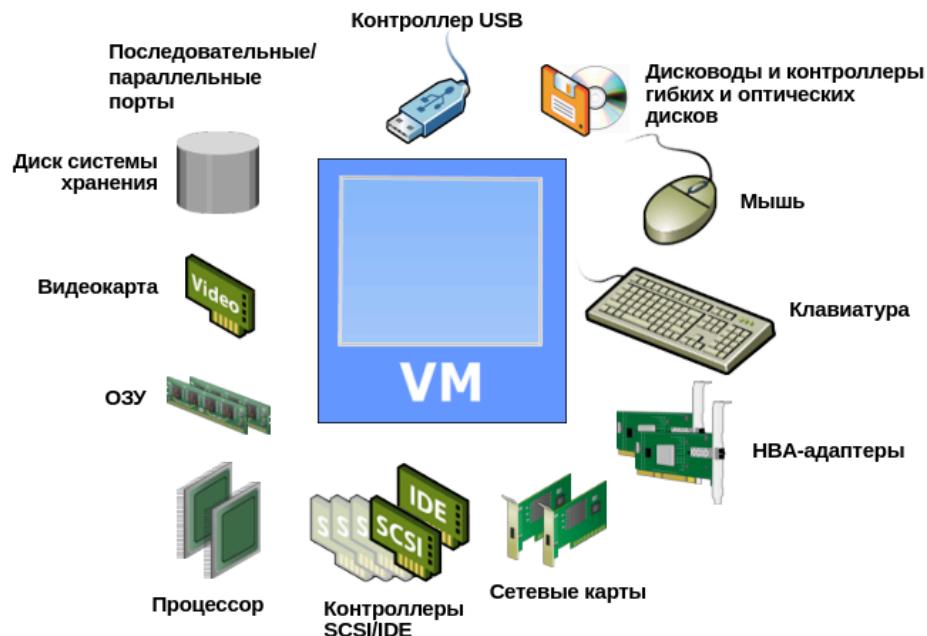
Serial attached SCSI (SAS) — это последовательный протокол «точка-точка», представляющий собой альтернативу параллельному SCSI. Новая версия последовательного протокола SCSI (SAS 3.0) поддерживает скорость передачи данных до 12 Гбит/с.

Fibre Channel — это широко используемый протокол для высокоскоростного обмена данными с устройством хранения данных. Интерфейс Fibre Channel обеспечивает скорость работы сети порядка гигабит в секунду. Он осуществляет последовательную передачу данных по медному проводу и волоконно-оптическому кабелю. Последняя версия интерфейса Fibre Channel «16FC» позволяет передавать данные со скоростью до 16 Гбит/с. Протокол Fibre Channel и его функции более подробно рассматриваются в модуле 9 («Сеть хранения данных Fibre Channel (FC)»).

IP — это сетевой протокол, который традиционно использовался для трафика между вычислительными системами. С приходом новых технологий IP-сеть превратилась в перспективный вариант для обмена данными между вычислительной системой и системой хранения. IP предлагает несколько преимуществ с точки зрения стоимости и зрелости и позволяет организациям использовать свои существующие IP-сети. Протоколы iSCSI и FCIP служат типичным примером использования IP для обмена данными между вычислительной системой и системой хранения. Эти протоколы подробно рассматриваются в модуле 10 («Internet Protocol (IP) SAN»).

преза 19 цод3, слайд 32 - про PCI

# Аппаратное обеспечение ВМ



С того же слайда:

“При создании ВМ ей предоставляются виртуальные аппаратные компоненты, которые для гостевой ОС выглядят как физические аппаратные компоненты. В рамках инфраструктуры конкретного поставщика каждая ВМ имеет стандартизованные аппаратные компоненты, что позволяет переносить их из одной физической вычислительной системы в другую. Эти виртуальные компоненты можно добавлять в ВМ и удалять из нее в зависимости от предъявляемых к ней требований. Тем не менее, не все компоненты доступны для добавления и настройки. Некоторые аппаратные устройства входят в состав виртуальной материнской платы и не могут быть изменены или удалены. Например, видеокарта и контроллеры PCI доступны по умолчанию и не могут быть удалены. На приведенном на слайде рисунке показаны типичные аппаратные компоненты ВМ. Они включают виртуальный процессор (или процессоры), виртуальную материнскую плату, виртуальное ОЗУ, виртуальный диск, виртуальный сетевой адаптер, оптические дисководы, последовательные и параллельные порты, а также периферийные устройства.

Конфигурация ВМ может включать один или несколько виртуальных процессоров. В зависимости от требований число виртуальных процессоров в ВМ можно увеличить или уменьшить. При запуске ВМ ядро гипервизора составляет график работы ее виртуальных процессоров на базе физических процессоров. Каждой ВМ назначается виртуальная материнская плата со стандартизованными устройствами, необходимыми для функционирования вычислительной системы. Виртуальное ОЗУ — это объем физической памяти, выделенный ВМ, который можно настраивать в зависимости от требований. Виртуальный диск — это физический файл большого размера или набор файлов, в котором хранятся ОС ВМ, программные файлы, данные приложений и другие данные, связанные с этой ВМ. Виртуальный адаптер сети функционирует подобно физическому сетевому адаптеру. Он обеспечивает возможность подключения между ВМ, запущенными в одной или разных вычислительных системах, а также между ВМ и физическими вычислительными системами. Виртуальные дисководы оптических и гибких дисков можно настроить для подключения либо к физическим устройствам, либо к файлам образов, например, к файлам ISO и образам гибких дисков (.flp), размещенным в системе хранения. Виртуальные контроллеры SCSI/IDE обеспечивают

возможность подключения ВМ к устройствам хранения данных. Виртуальный контроллер USB используется для подключения к физическому контроллеру USB и для получения доступа к подключенными USB-устройствам. Последовательные и параллельные порты предоставляют интерфейс для подключения к ВМ периферийных устройств.”

### Вопрос 15

Отметьте термины относящиеся к стратегиям оптимизации запросов

Выберите один или несколько ответов:

- Сортировка файлов перед операцией соединения
- Выполнять декартово произведение перед проекциями
- Индексирование файлов перед операцией соединения
- Выполнять операции селекции по возможности раньше
- Собирать в каскады селекции и проекции
- Выполнять декартово произведение до натурального соединения

Ответ

- Сортировка файлов перед операцией соединения
- Индексирование файлов перед операцией соединения
- Выполнять операции селекции по возможности раньше
- Собирать в каскады селекции и проекции

Пояснение

Преза 22 , сл 14

### Общие стратегии оптимизации

- Выполнять операции селекции по возможности раньше
- Целесообразно предварительно обрабатывать файлы перед соединением
  - ❖ Сортировка файлов
  - ❖ Создание индексов
- Искать общие подвыражения
  - ❖ Подвыражения, использующие соединения, не изменяющееся при перемещении селекции вовнутрь его.
- Собирать в каскады селекции и проекции
- Комбинировать проекции с предшествующими и последующими двуместными операциями
- Комбинировать некоторые селекции с предшествующим декартовым произведением и выполнять вместо них соединения

### Вопрос 16

Укажите основные компоненты Центров Обработки Данных

Выберите один или несколько ответов:

- Мониторинг
- Выделение ресурсов
- Отчетность
- Обслуживание
- Планирование
- Реинжиниринг
- Разработка

Ответ

- Мониторинг
- Выделение ресурсов
- Отчетность
- Обслуживание
- Планирование

Пояснение

Преза 19 1 , сл 11

## Ключевые процессы управления центром обработки данных

| Процесс управления | Описание                                                                                                                          |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Мониторинг         | Непрерывный сбор информации о ресурсах центра обработки данных                                                                    |
| Отчетность         | Предоставление подробных сведений о производительности, емкости и использовании ресурсов                                          |
| Выделение ресурсов | Конфигурирование и распределение ресурсов в соответствии с требованиями к емкости, доступности, производительности и безопасности |
| Планирование       | Оценка количества ресурсов, необходимых для поддержки бизнес-операций                                                             |
| Обслуживание       | Обеспечение надлежащего функционирования ресурсов и устранение инцидентов                                                         |

### Вопрос 17

Отметьте набор характеристик, свойственных RAID 1.

Выберите один или несколько ответов:

- а. Для каждой операции записи нового блока нужно 2 обращения к дискам
- б. Имеется выделенный диск четности
- в. Минимальное количество дисков = 3
- г. Минимальное количество дисков = 2
- е. Массив восстанавливается при потере 2 дисков

- f. Нет диска четности
- g. Для каждой операции записи нового блока нужно 4 обращения к дискам
- h. Массив восстанавливается при потере 1 диска

Ответ

- Для каждой операции записи нового блока нужно 2 обращения к дискам
- Минимальное количество дисков = 2
- Нет диска четности
- Массив восстанавливается при потере 1 диска

Пояснение

По [видосу](#) кайф понятно

Преза 23, сл 36

## Сравнение массивов RAID

| Уровень RAID | Мин. кол-во дисков | Доступная емкость ресурсов хранения (%) | Дополнительные издержки записи | Защита данных                                        |
|--------------|--------------------|-----------------------------------------|--------------------------------|------------------------------------------------------|
| 1            | 2                  | 50                                      | 2                              | Зеркало                                              |
| 1+0          | 4                  | 50                                      | 2                              | Зеркало                                              |
| 3            | 3                  | $[(n-1)/n]*100$                         | 4                              | Контроль четности<br>(поддерживает один сбой дисков) |
| 5            | 3                  | $[(n-1)/n]*100$                         | 4                              | Контроль четности<br>(поддерживает один сбой дисков) |
| 6            | 4                  | $[(n-2)/n]*100$                         | 6                              | Контроль четности<br>(поддерживает два сбоя дисков)  |

Преза 23, сл 16

# RAID 1 – зеркалирование диска

- ❖ Полная избыточность, дорогостоящая технология
- ❖ Обеспечивает защиту данных и быстрое восстановление
- ❖ Не заменяет резервное копирование, которое фиксирует образы в определенное время
- ❖ Улучшает скорость чтения
- ❖ Скорость записи снижается



RAID Controller



❖ То есть MTBF для RAID 1  
 $5 * 29200 = 146\ 000$  лет



- ❖ Вероятность того, что диск выйдет из строя в момент копирования на него  $(1/10)*(1/2920) = 1/29200$
- ❖ Тогда один из 2 дисков может отказать раз в 5 лет и только один из 29200 отказов повлечет потерю данных.

- 16

През 23, сл 31

## Сравнение RAID'ов

| RAID | Мин. дисков | Эффект. хранения данных, %              | Стоимость | Скорость считывания                                                                        | Скорость записи                                                                                                   | Дополнительные расходы на запись |
|------|-------------|-----------------------------------------|-----------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------------|
| 0    | 2           | 100                                     | Низкая    | Отличная как при произвольном так и последовательном считывании                            | Отличная                                                                                                          | Нет                              |
| 1    | 2           | 50                                      | Высокая   | Хорошая. Лучше считывания с одного диска                                                   | Хорошая. Медленнее, чем на единичном диске, поскольку каждая запись производится на несколько дисков              | Приемлемые                       |
| 3    | 3           | $(n-1)*100/n$ где n = количество дисков | Средняя   | Хорошая при произвольном и отличная последовательном считывании                            | Плохая или приемлемая при небольших записях в произвольном порядке. Хорошая при объемных последовательных записях | Большие                          |
| 4    | 3           | $(n-1)*100/n$ где n = количество дисков | Средняя   | Отличная при произвольном считывании. Хорошая или отличная при последовательном считывании | Плохая или приемлемая при произвольных записях. Хорошая при последовательных записях.                             | Большие                          |

## Сравнение RAID'ов (2)

| Мин RAID дисков | Эффект хранения данных, %                       | Стоимость                     | Скорость считывания                                                       | Скорость записи                                                                                                                                                    | Дополнительные расходы на запись |           |
|-----------------|-------------------------------------------------|-------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|-----------|
| 5               | 3<br>$(n-1)*100/n$<br>где N = количеству дисков | Средняя                       | Очень хорошая для случного чтения<br>Хорошая для последовательного чтения | Приемлемая при произвольных записях.<br>замедление объясняется дополнительными расходами на запись четности.<br>Приемлемая или хорошая при последовательной записи | Большие                          |           |
| 6               | 4<br>$(n-2)*100/n$<br>где N = количеству дисков | Средняя, но больше чем RAID 5 | Очень хорошая для случного чтения<br>Хорошая для последовательного чтения | Хорошая для небольших произвольных записей (бывают необходимы дополнительные затраты на запись)                                                                    | Очень большие                    |           |
| 1+0 и 0+1       | 4                                               | 50                            | Высокая                                                                   | Отличная                                                                                                                                                           | Хорошая                          | Умеренные |

### Вопрос 18

Отметьте ключевые характеристики центра обработки данных

Выберите один или несколько ответов:

- Отчетность
- Управляемость
- Прозрачность
- Емкость
- Безопасность
- Масштабируемость
- Достоверность
- Доступность
- Эффективность
- Производительность
- Целостность данных

Ответ

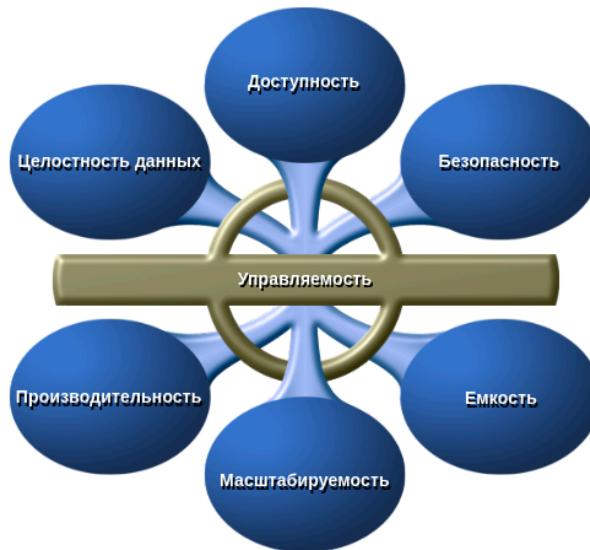
- Управляемость
- Емкость
- Безопасность
- Масштабируемость
- Доступность

- Производительность
- Целостность данных

Пояснение

Преза 19 1 , сл 10

## Ключевые характеристики центра обработки данных



Пояснение из того же слайда:

“Центры обработки данных должны иметь ключевые характеристики, приведенные на данном слайде. Хотя эти характеристики применимы почти ко всем компонентам центра обработки данных, в этом модуле речь пойдет в основном о системах хранения.

**Доступность.** Центр обработки данных должен обеспечивать доступность информации по требованию в любое время. Недоступность информации может серьезно снизить эффективность бизнес-операций, привести к значительным финансовым потерям и нанести вред репутации организации.

**Безопасность.** Необходимо создать политики и процедуры и принять контрольные меры по предотвращению несанкционированного доступа к информации и ее изменения.

**Емкость.** Для операций центра обработки данных требуется достаточный объем ресурсов, чтобы обеспечить эффективное хранение и обработку больших объемов данных, которые продолжают непрерывно расти. При увеличении требований к емкости дополнительная емкость должна предоставляться без прерывания доступности или с ее минимальным прерыванием. Емкостью можно управлять путем добавления новых ресурсов или перераспределения существующих.

**Масштабируемость.** Организациям может потребоваться развернуть дополнительные ресурсы, например вычислительные системы, новые приложения и базы данных, чтобы обеспечить соответствие растущим требованиям. Ресурсы центра обработки данных должны масштабироваться для соответствия изменяющимся требованиям без прерывания бизнес-операций.

**Производительность.** Компоненты центра обработки данных должны обеспечивать оптимальную производительность в соответствии с необходимыми уровнями обслуживания.

**Целостность данных.** Целостность данных обеспечивается различными механизмами, например кодами исправления ошибок или битами четности, которые гарантируют, что данные будут храниться и извлекаться точно в том виде, в котором они были получены. Управляемость. Центр обработки данных должен предоставлять возможность простого,

гибкого и интегрированного управления всеми своими компонентами. Эффективную управляемость можно обеспечить за счет автоматизации, которая позволяет уменьшить вмешательство сотрудников в выполнение рутинных, повторяемых задач.”

### Вопрос 19

Сопоставьте набор характеристик и наименование вычислительной платформы

- вторая вычислительная платформа –
- первая вычислительная платформа –
- третья вычислительная платформа –

Ответ

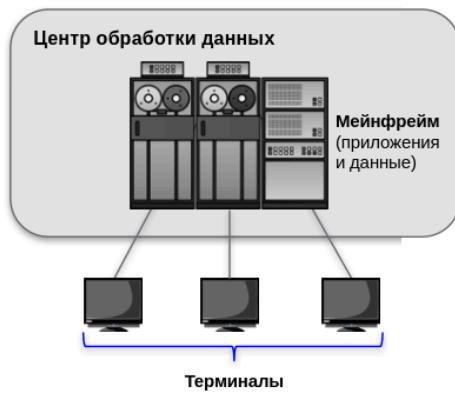
- вторая вычислительная платформа – распределенная архитектура приложений, клиент-серверное взаимодействие с вычислителем
- первая вычислительная платформа – Централизованное размещение приложений и баз данных Вычислители - мейнфреймы, подключение к вычислителям через терминалы
- третья вычислительная платформа – Облачные хранилища и вычислительные услуги, большие данные, мобильные технологии и социальные сети

Пояснение

Преза 19 всхд1 , сл 13

## Первая платформа

- Создана на базе мейнфреймов
  - Централизованное размещение приложений и баз данных
  - Пользователи подключаются к мейнфреймам через терминалы
- Сложности, связанные с мейнфреймами
  - Значительные капитальные и операционные издержки
    - Высокие затраты на приобретение
    - Значительная занимаемая площадь и высокие требования к энергопотреблению



Пояснение из этого слайда:

“Мейнфреймы — это вычислительные системы с очень высокой вычислительной мощностью, большим объемом памяти и емкостью хранения. Они главным образом используются для централизованного размещения критически важных приложений и баз данных в центре обработки данных организации. Множество пользователей одновременно подключаются к мейнфреймам через менее мощные устройства, например рабочие станции или терминалы.

Все операции по обработке данных выполняются в мейнфрейме, а терминалы только предоставляют интерфейс для использования приложений и обеспечивают представление результатов. Хотя мейнфреймы обеспечивают высокую надежность и безопасность, с ними связан ряд проблем высокой стоимости.

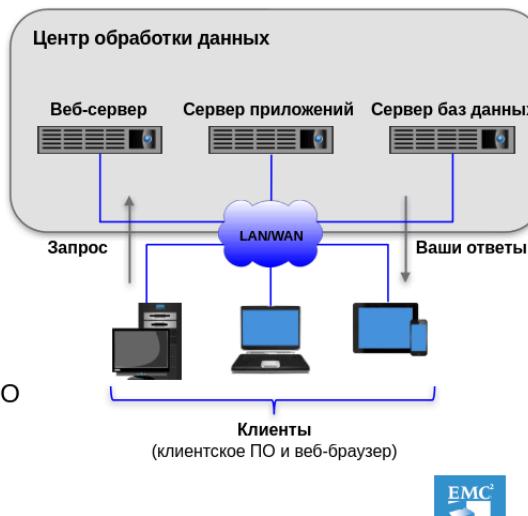
Мейнфреймы требуют высоких затрат на приобретение и большой пощади для размещения, а также

потребляют много электроэнергии. Разворачивание мейнфреймов в центре обработки данных может быть сопряжено с высокими капитальными и операционными издержками. В прошлом крупные организации, например банки, страховые агентства и государственные учреждения, использовали мейнфреймы для своих бизнес-операций.”

Преза 19 всхд1 , сл 14

## Вторая платформа

- На основе модели «клиент-сервер»
  - Распределенная архитектура приложений
  - Серверы получают и обрабатывают запросы на предоставление ресурсов от клиентов
  - Пользователи подключаются через клиентскую программу или веб-интерфейс
- Сложности, связанные с моделью «клиент-сервер»
  - Создание изолированных точек хранения в ИТ-инфраструктуре
  - Издержки на обслуживание оборудования и ПО
  - Масштабируемость, необходимая, чтобы адаптироваться к увеличению количества пользователей и рабочих нагрузок



С того же слайда:

“Модель «клиент-сервер» использует распределенную архитектуру приложений, в которой в вычислительной системе, называемой «сервером», работает программа, предоставляющая услуги по сети для других программ, которые работают на различных конечных устройствах, называемых «клиентами».

Серверные программы получают запросы на ресурсы от клиентских программ, а в ответ на эти запросы, клиенты получают доступ к ресурсам, например приложениям электронной почты, бизнес-приложениям, веб-приложениям, базам данных, файлам и принтерам.

Клиентскими устройствами могут быть настольные ПК, ноутбуки и мобильные устройства. Обычно клиенты подключаются к серверам по локальной или глобальной сети. При этом пользователи используют клиентские приложения или веб-интерфейс браузера.

В модели «клиент-сервер» и клиенты, и серверы могут выполнять четко определенные рутинные задачи по обработке данных. Например, клиент может использовать бизнес-приложение, в то время как на сервере может быть запущена система управления базой данных, которая управляет хранением, извлекает информацию из базы данных и вносит ее в эту базу. Это называется двухуровневой архитектурой.

Кроме того, клиент может использовать приложение или веб-интерфейс для принятия информации, в то время как сервер запускает другое приложение, которое обрабатывает информацию и отправляет данные второму серверу, на котором работает система управления базой данных. Это так называемая трехуровневая архитектура. Эта распределенная архитектура приложений может расширяться до любого количества уровней (n-уровневая архитектура). Поскольку и клиентская, и серверная системы представляют собой интеллектуальные устройства, модель «клиент-сервер» совершенно отлична от модели, использующей мейнфреймы.

На рисунке на данном слайде показан пример модели «клиент-сервер». В этом примере клиенты взаимодействуют с веб-сервером с помощью веб-браузера. Веб-сервер обрабатывает запросы клиентов по протоколу HTTP и выдает HTML-страницы. Бизнес-приложение находится на сервере приложений, а система управления базой данных находится на сервере баз данных. Клиенты взаимодействуют с сервером приложений через клиентское программное обеспечение. Сервер приложений связывается с сервером баз данных для извлечения информации и предоставления

результатов клиентам. В некоторых случаях приложения и базы данных могут находиться на одном и том же сервере.

Некоторые сложности, сопряженные с моделью «клиент-сервер», связаны с созданием изолированных точек хранения, издержками на обслуживание и проблемами масштабируемости.

Во многих организациях бизнес-подразделения и отделы имеют собственные серверы, на которых работают бизнес-приложения.

Это приводит к созданию изолированных приложений и точек хранения информации (отдельных, разрозненных систем).

Изолированные точки хранения усложняют эффективное использование ИТ-ресурсов и предоставление к ним общего доступа. Кроме того, часто возникают сложности с их интеграцией и управлением. Хотя стоимость серверного оборудования значительно ниже, чем стоимость мейнфреймов, обслуживание многочисленных серверов, клиентов и их ПО все равно требует значительных операционных затрат.

Более того, с помощью этой модели сложно обеспечить соответствие требованиям, связанным с быстрым ростом количества пользователей, объемов информации и рабочих нагрузок приложений. Добавление дополнительных серверов не всегда обеспечивает более эффективное управление рабочими нагрузками. Кроме того, необходимо оптимально распределить процессы обработки данных и логику приложений между серверами и экземплярами приложений.

Примечание. Обычно вычислительной системой называют устройство с операционной системой (ОС), в котором запускаются приложения. В качестве примеров вычислительных систем можно привести физические серверы, хосты, настольные ПК, ноутбуки и мобильные устройства. В рамках этого курса термин «вычислительная система» или «вычислительный ресурс» используется для обозначения физических серверов и хостов, на которых развертываются бизнес-приложения организаций.”

Преза 19 всхд1 , сл 15

## Третья платформа



- Четыре основные составляющие, которые трансформируют использование организациями технологий для бизнес-операций

С того же слайда:

“Термин «третья платформа» был введен компанией IDC. Это же понятие Gartner обозначает термином «связи сил».

Базовые элементы третьей платформы: облака, большие данные, мобильные технологии и социальные сети. Это четыре основные передовые технологии, которые в значительной мере трансформируют бизнес, экономику и общество по всему миру.

В основе третьей платформы лежат облака, которые позволяют потребителям получать ИТ-ресурсы по модели «как услуга» от поставщика облачных услуг.

Большие данные позволяют проводить аналитику, извлекая из данных больше полезных сведений для принятия более эффективных решений.

Мобильные устройства обеспечивают повсеместный доступ к приложениям и информации.

Социальные сети обеспечивают взаимодействие пользователей и обмен информацией.

В течение последних тридцати лет организациям было очень важно разумно использовать вторую платформу в своем бизнесе. По данным IDC, в течение следующих тридцати лет третья платформа станет основой для разработки решений и бизнес-инноваций.

Третья платформа используется для трансформации цифровой среды, развития и расширения всех отраслей, а также для получения новых значительных конкурентных преимуществ. Разработчики бизнес-стратегии, ИТ-руководители и разработчики решений уже создают передовые бизнес-модели и потребительские сервисы на базе технологий третьей платформы.

Технологии третьей платформы улучшают технологии второй платформы, а не заменяют их.

Ключевой аспект третьей платформы — сочетание облаков, больших данных, мобильных технологий и социальных сетей, а не использование каждой из этих технологий по отдельности.

Ключевое условие успешной работы в таких условиях — объединение двух или более технологий для создания высокопроизводительных отраслевых решений, называемых «гибридными решениями».

Например, к самым распространенным факторам, способствующим внедрению облаков, относятся решения для социальных сетей и мобильных устройств.

Это значит, что организации уже понимают значимость гибридных решений, включающих элементы из всех четырех технологий. Различные сочетания технологий третьей платформы уже трансформируют организации в сфере розничной торговли, финансовых услуг, телекоммуникаций и здравоохранения, а также государственные учреждения.

По оценкам IDC, в настоящее время свыше 80% инфраструктуры и приложений в большинстве центров обработки данных относятся ко второй платформе.

Кроме того, на технологии второй платформы в данный момент приходится 74% ИТ-затрат по всему миру. Следовательно, для организаций, которые инвестируют значительные средства в технологии второй платформы, немедленный и полный переход к третьей платформе может быть затратным и непрактичным. Это обстоятельство привело к созданию промежуточной вычислительной платформы под названием «платформа 2.5», которая сочетает в себе черты второй и третьей платформ.

Платформа 2.5 включает решения и технологии, которые позволяют организациям сократить разрыв между второй и третьей платформой. Технологии платформы 2.5 позволяют организациям использовать сочетание технологий второй и третьей платформ. С их помощью организации смогут предоставлять приложения второй платформы и получать результаты работы для третьей платформы без дубликации и перемещения данных.

Например, технологии платформы 2.5 позволяют организациям запускать приложения второй платформы, используя традиционные структуры данных и протоколы, и при этом анализировать эти же данные с применением технологий больших данных.

По прогнозам компании IDC, в будущем глобальные затраты на ИТ будут направлены в основном на следующие сегменты: беспроводная передача данных, смартфоны и планшеты, облачные услуги, аналитика больших данных и Интернет вещей.

Ожидается, что эти затраты составят несколько сотен миллиардов долларов в каждом из сегментов. Это свидетельствует о набирающей популярность отраслевой тенденции к масштабному внедрению технологий третьей платформы.

Согласно оценкам специалистов, к 2020 году на технологии третьей платформы будет направлено свыше 40% ИТ-затрат. В модуле 2 рассматриваются технологии третьей платформы.”

## Вопрос 20

Укажите цифрой число бит, которое нужно отвести под контрольную сумму, чтобы вероятность незамеченной ошибки была равна 1/128

Ответ

- 7

#### Пояснение

Считаем так:  $1 / (2^7) = 1 / 128$

Преза 23, сл 5

## Контрольная сумма

- Каждому сектору носителя соответствуют несколько **дополнительных бит – контрольная сумма (КС)**, которая зависит от содержимого сектора.
- Количество **единиц** в наборе битов данных, включая бит четности – ВСЕГДА ЧЕТНО.
- Если функция чтения обнаруживает **несоответствие между данными и КС**, то она возвращает параметр  $s = \text{«ложь»}$ , противном случае – «истина».
- Есть вероятность, что данные в секторе – некорректны, а КС соответствует корректным данным. Например, изменились сразу 2 бита – КС осталась правильной. Вероятность ошибки -  $\frac{1}{2}$
- Эту вероятность снижают увеличением битов КС.
- Например, можно предусмотреть 8 битов четности по 1 на соответствующий бит каждого байта данных в секторе.  
Тогда вероятность ошибки  $\frac{1}{2^{256}}$ .
- Если использовать  $n$  независимых битов четности, то вероятность незамеченных ошибок составляет  $1/2^n$
- И если под КС отвести **4 байта**, то ошибка **не будет обнаружена в одном из 4 миллиардов** случаев

### Вопрос 21

Отметьте типы устройств, используемых для долговременного хранения данных

Выберите один или несколько ответов:

- Магнитный ленточный накопитель
- Кэш память
- Магнитный диск
- Твердотельный диск
- ферритовая память
- Оптический диск
- Память на микросхемах

Ответ

- Магнитный ленточный накопитель
- Магнитный диск
- Твердотельный диск
- Оптический диск

Пояснение

## Типы устройств хранения данных

### Накопитель на магнитных дисках

- Данные хранятся на вращающемся диске с ферромагнитным покрытием
- Обеспечивает произвольный доступ для чтения и записи
- Самое популярное устройство хранения данных с большой емкостью хранения

### Твердотельный диск (флэш-диск)

- Данные хранятся в памяти на основе полупроводников
- Чрезвычайно малые задержки операций ввода-вывода, низкое энергопотребление и очень высокая пропускная способность

### Магнитный ленточный накопитель

- Данные хранятся на пластиковой пленке с магнитным покрытием
- Обеспечивает только последовательный доступ к данным
- Недорогое решение для длительного хранения данных

### Оптический дисковод

- Данные хранятся на поликарбонатном диске с отражающим покрытием
- Возможность однократной записи и многократного чтения: CD, DVD, BD
- Недорогое решение для длительного хранения данных

Пояснение с того же слайда:

“Магнитный диск представляет собой вращающийся носитель данных из немагнитного материала (обычно сплава), покрытый ферромагнитным материалом. Данные хранятся на обеих поверхностях магнитного диска (верхней и нижней), а их запись производится путем поляризации части поверхности диска. Дисковый накопитель представляет собой устройство с несколькими вращающимися магнитными дисками, называемыми пластинами и расположенными вертикально внутри металлического или пластикового корпуса. Каждая пластина снабжена быстро перемещающимся рычагом привода магнитных головок для чтения данных с диска и записи на диск. В настоящее время дисковые накопители — самые популярные носители информации, используемые для хранения данных и получения к ним доступа в приложениях с высокими требованиями к производительности. Диски поддерживают быстрый доступ к данным, расположенным в произвольных местах, и обеспечивают возможность быстрой записи или извлечения данных для нескольких одновременно работающих пользователей или приложений. Дисковые накопители используют стандартные протоколы, например, Advanced Technology Attachment (ATA), Serial ATA (SATA), Small Computer System Interface (SCSI), Serial Attached SCSI (SAS) и Fibre Channel (FC). Эти протоколы хранятся в контроллерах дискового интерфейса, которые обычно встроены в дисковые накопители. Каждый протокол обеспечивает уникальное сочетание характеристик производительности, стоимости и емкости дисков.

Твердотельный диск (SSD) использует для хранения данных память на основе полупроводников, например, микросхемы NAND и NOR. Твердотельные диски, известные также как «флэш-диски», обеспечивают сверхвысокое быстродействие, необходимое для приложений с высокими требованиями к производительности. В отличие от обычных механических дисковых накопителей эти устройства не содержат движущихся частей, и в них отсутствуют задержки, связанные с перемещением головок чтения-записи и вращением дисков. По сравнению с другими доступными устройствами хранения данных твердотельные диски обеспечивают относительно высокое число операций ввода-вывода в секунду (IOPS) с очень малым временем отклика. Кроме того, они потребляют меньше энергии и обычно имеют более длительный срок службы по сравнению с механическими дисковыми накопителями. Тем не менее, флэш-диски имеют самую высокую стоимость в расчете на гигабайт (\$/Гбайт).

Магнитная лента представляет собой тонкую длинную полосу пластиковой пленки, покрытую ферромагнитным материалом, например, ферритом бария. Эта лента упаковывается в пластиковые кассеты и картриджи. Ленточный накопитель — это устройство для записи данных на магнитную ленту и их извлечения. Ленточные накопители обеспечивают линейный последовательный доступ к данным для чтения и записи. Ленточный накопитель может быть автономным или входить в состав ленточной библиотеки. Ленточная библиотека включает один или несколько ленточных накопителей и ресурс хранения с рядом картриджей с магнитной лентой, вставленных в гнезда. Благодаря сравнительно

низкой стоимости и удобству переноски лента часто используется в качестве носителя для долгосрочного хранения данных. Ленточные накопители обычно используются организациями для хранения больших объемов данных, например, для резервного копирования, создания автономных архивов и аварийного восстановления. Основными ограничивающими факторами при использовании ленты служат низкая скорость доступа к данным, обусловленная последовательным механизмом доступа, отсутствие возможности одновременного доступа к ленте нескольких приложений и ухудшение свойств поверхности ленты из-за постоянного контакта с головкой чтения-записи.

Оптический диск — это плоский круглый носитель информации, изготовленный из поликарбоната, одна из поверхностей которого имеет специальное отражающее покрытие (например, алюминиевое). Оптический дисковод использует записывающий лазер для записи данных на диск в виде микроскопических светлых и темных точек. Считывающий лазер считывает эти точки и генерирует электрические сигналы, представляющие эти данные. К самым распространенным типам оптических дисков относятся компакт-диски (CD), универсальные цифровые диски (DVD) и диски Blu-ray (BD). Эти диски могут записываемыми или перезаписываемыми. Записываемые или ROM-диски используют схему «одна запись, многократное чтение» и обычно применяются в качестве распространяемого носителя для приложений или как средство передачи небольших объемов данных из одной системы в другую.

Ограниченнная емкость и скорость оптических дисков сдерживают их использование в качестве универсального решения для хранения данных предприятия. Тем не менее, оптические диски высокой плотности иногда используют в качестве решения для хранения данных фиксированного содержания и архивных данных. Некоторые поставщики, использующие модель «хранение данных как услуга», предлагают возможность копирования файлов резервных копий на зашифрованные оптические диски и последующей отправки их потребителям.”

## Вопрос 22

В каком порядке осуществляется доступ к файлу "КУЧА" при редактировании записи ?

Ответ

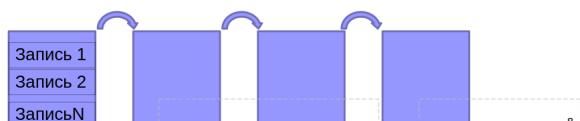
- последовательно

Пояснение

Преза 20, сл 8-9

### Куча

- Вставка. Последовательный просмотр блоков файла с целью поиска свободного места в блоке.
- Удаление. Поиск записи по значению поля (ключа) с помощью последовательного просмотра блоков файла. Удаление записи в блоке. (шкала, длина блока)
- Поиск. Последовательный просмотр блоков.
- Редактирование. Удаление и вставка.



### Куча. Свойства

- Блоки файла фиксированной длины
- Записи не закреплены
- Блоки файла неупорядочены
- Возможность плотно хранить записи
- Последовательный доступ
- Оценка доступа для кучи –  $N/2$ , где  $N$  – количество блоков в файле

## Вопрос 23

Укажите уровни ИТ инфраструктуры центра обработки данных.  
Выберите один или несколько ответов:

- Управление
- Программно-определеные компоненты
- Физические компоненты
- Оркестрация
- Безопасность
- Услуги
- Виртуальные компоненты
- Непрерывность бизнеса

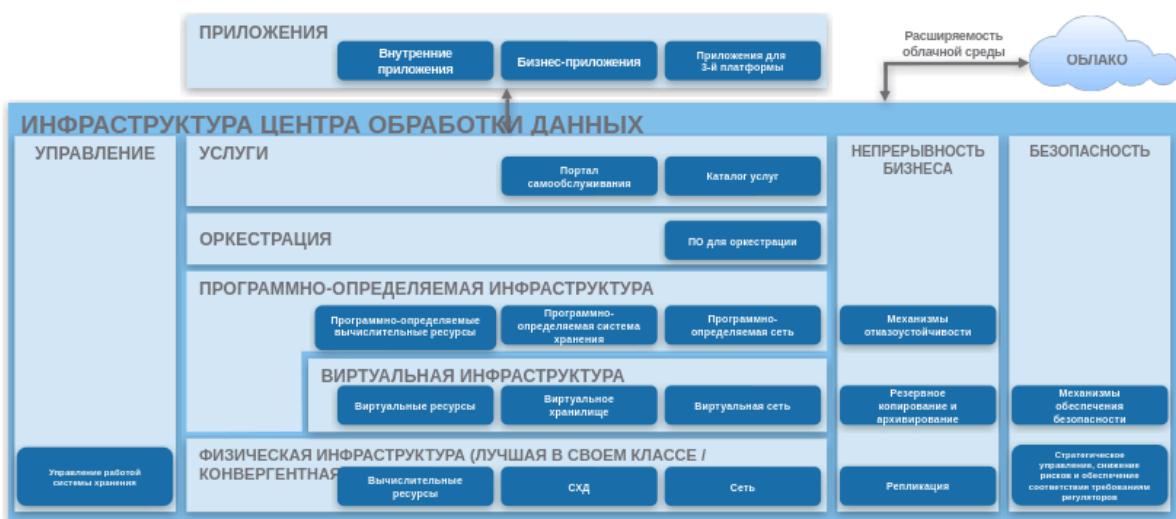
Ответ

- Оркестрация
- Виртуальные компоненты
- Услуги
- Программно-определеные компоненты
- Физические компоненты

Пояснение

Преза 19 цод3, сл 4

## Инфраструктура центра обработки данных



Пояснение с того же слайда:

“В модуле 1 «Введение в систему хранения информации» было представлено определение центра обработки данных, а также описаны ИТ-инфраструктура и инфраструктура поддержки, которые образуют центр обработки данных.

Рисунок на слайде представляет собой блок-схему, отображающую основные строительные блоки ИТ-инфраструктуры, которые образуют центр обработки данных.

Эта ИТ-инфраструктура включает пять логических уровней и три межуровневые функции.

Пять уровней представляют собой

- физическую инфраструктуру,
- виртуальную инфраструктуру,
- программно-определенную инфраструктуру,
- уровень оркестрации и
- уровень услуг.”

## Вопрос 24

Укажите методы поиска по неключевым полям

Выберите один или несколько ответов:

- Формирование множественных вторичных индексов
- Оптимизация запросов на основе регулярных выражений
- Применение функций раздельного хэширования
- Приведение запроса к каноническому виду

Ответ

- Формирование множественных вторичных индексов
- Применение функций раздельного хэширования

Пояснение

Преза 21 мвп, сл 11, 13, 15, 16

- **Поиск по частичному соответствию**
- Обобщенная формулировка задачи поиска:  
“Найти записи, в которых поле  $F_1$  имеет значение  $v_1$ , поле  $F_2 - v_2, \dots$ , поле  $F_k - v_k$ ”
- Если  $S_i$  – есть множество записей со значением  $v_i$  в поле  $F_i$ , для  $i = 1, 2, \dots, k$ , то требуемое множество записей можно получить как пересечение  $S_1 \cap S_2 \cap \dots \cap S_k$ .
- Имеется два метода решения этой задачи, использованием:
  - ❖ множественных вторичных индексов
  - ❖ специальной формы функции хэширования

## Использование множественных вторичных индексов

Задача: найти записи  $S_1 \cap S_2 \cap \dots \cap S_k$

- Не обязательно фактически искать все записи из  $S_1$  затем — из  $S_2$  и т. д. Множества  $S_i$  могут оказаться большими.
- Можно использовать **вторичные индексы** по некоторым или всем полям  $F_i$
- Когда эти множества не слишком велики, их **пересечение может быть построено в основной памяти**.
- По каждому значению индекса (указатель или ключ) в пересечении проверяем, имеет ли связываемая с этим индексом запись заданные значения в оставшихся полях, для которых не существует вторичного индекса

## Использование множественных вторичных индексов

- С целью минимизации размера списка указателей, с которыми предстоит иметь дело, следует сначала получить **наименьшее множество  $S_i$  среди всех полей  $F_i$** , имеющих вторичный индекс
- Эвристически можно выбирать поле  $F_i$  с **наибольшим множеством различных значений**, представленных в базе данных, если, конечно, такая статистика известна или может быть предсказана.
- В примере выше можно предположить, что вторичный индекс по полю «должность» будет содержать наименьший список указателей.
- **В случае очень больших файлов** метод поиска по частичному соответствию не рекомендуется.

## Функции раздельного хэширования

- Все поля принимаем за КЛЮЧ
- Хэшируем записи, используя полное множество ключей
- «Хитрость» состоит в **разбиении битов номера участка** на несколько частей таким образом, чтобы каждое поле определяло одну из частей.
- Тогда всякий раз, когда известно одно или более полей, мы уже что-либо знаем о номерах участков, в которых может находиться требуемая запись (записи).

## Вопрос 25

За какое максимальное количество обращений к диску можно найти запись в "ХЭШ файле", если

- таблица участков умещается в оперативной памяти,
- каждый участок, содержит не более 3 блоков

Ответ

- 4

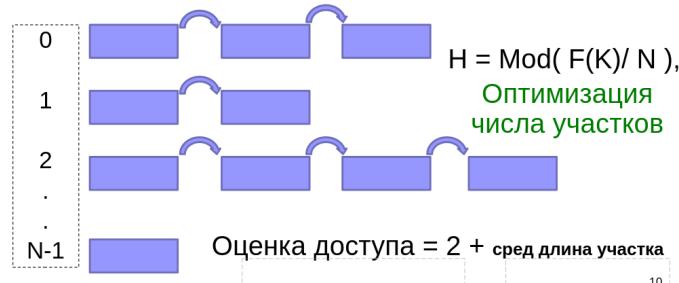
Пояснение

Преза 20, сл 10 - 14

В задании спрашивают про поиск. На слайде со свойствами написано про оценку доступа. Когда мы "доступаемся" к записи, то сначала ищем эту запись и потом берём её. И вот это взятие = 1 операция. Получаем, что оценка доступа =  $2 + 3 = 5$ . Поиск =  $5 - 1 = 4$ .

### Хэш файл. Hash file.

Идея – разбиение всех блоков файла на **участки**. Помещать записи в участки в зависимости от **значения хэш функции на ключе записи**.



### Операции. Поиск.

Пусть требуется найти запись с ключем  $k_1$ .

Тогда возникает задача найти блок В с этой записью.

Мы знаем, что в индексе есть пары  $\langle k, B \rangle$ , отсортированные по  $k$ .

Следовательно, будем искать в индексе запись  $\langle k_2, B \rangle$ , такую, что  $k_2 \leq k_1$  и либо  $\langle k_2, B \rangle$  последняя запись в блоке, либо следующая запись  $\langle k_3, B \rangle$  удовлетворяет условию  $k_1 < k_3$ .

### Общая таблица

| Вид файла          | Оценка доступа                       |
|--------------------|--------------------------------------|
| Куча               | $\frac{n}{2}$                        |
| Хэш файл           | $\geq 3$                             |
| Разреженный индекс | $\sim 2 + \log_2(n)$                 |
| B – дерево         | $\sim 2 + \log_d(\frac{n}{e})$       |
| Плотный индекс     | $\leq 2 + \text{время над индексом}$ |

### Хэш файл. Свойства.

- Блоки файла фиксированной длины
- Записи не закреплены
- Блоки файла неупорядочены
- Доступ по значению Хэш функции от ключа
- Оценка доступа = 2 + средняя длина участка
- Можно изменять Хэш функцию, улучшая скорость доступа

## ТЕСТ 5

### Вопрос 1

Отметьте правильные суждения о процессе фиксации транзакций в распределенной СУБД

Выберите один или несколько ответов:

- Транзакция завершается COMMIT, если все агенты сообщили OK
- Используется протокол 2-х фазной фиксации
- Используется протокол 2-х фазных транзакций
- Транзакция завершается ROLLBACK, если хотя бы один агент не снял блокировку
- Первая фаза протокола - фаза рассылки сообщения "готовность"
- Вторая фаза протокола - оценка ответов агентов
- Первая фаза протокола - фаза расширения блокировки
- Вторая фаза протокола - сужение блокировки

Ответ

- Транзакция завершается COMMIT, если все агенты сообщили OK
- Используется протокол 2-х фазной фиксации
- Первая фаза протокола - фаза рассылки сообщения "готовность"
- Вторая фаза протокола - оценка ответов агентов

Пояснение

Преза 29, сл 31-32

#### ■ ■ ■ Протокол 2-ух фазной фиксации

##### Первая фаза протокола

- Транзакция посылает всем своим агентам сообщение «ready» - «готовность»
- Получив сообщение «ready» агент возвращает сообщение «OK», если нет препятствий к нормальному завершению его работы - «Commit». В зависимости от протокола восстановления «REDO» или «UNDO» в системном журнале имеется информация для нормального завершения агента - «Commit».
- В противном случае агент возвращает запустившей его транзакции сообщение «NO».
- Основная транзакция собирает ответы-сообщения от всех агентов.

#### ■ ■ ■ Протокол 2-ух фазной фиксации

##### Вторая фаза протокола.

- Если все агенты ответили «OK», транзакция посылает все своим агентам сообщение «Commit» - «Завершение» и завершает выполнение с командой «Commit»
- Если хотя бы один агент ответил «NO», транзакция посылает все своим агентам сообщение «Rollback» - «Откат» и завершает выполнение с командой «Rollback»

### Вопрос 2

С какими компонентами СУБД взаимодействует "менеджер транзакций" ?  
Выберите один или несколько ответов:

- Менеджер восстановления
- Процессор запросов
- Базой данных
- Менеджер протоколирования

- Менеджер буферов
- Планировщик заданий

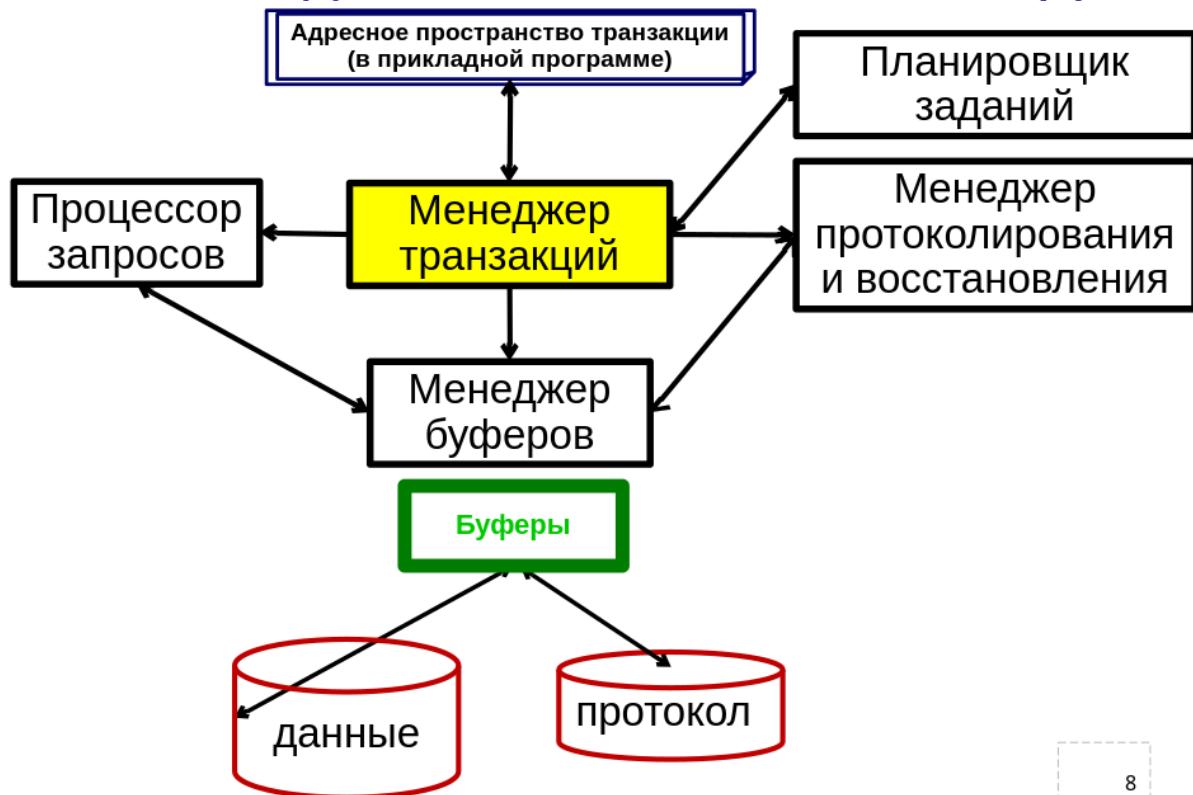
Ответ

- Процессор запросов
- Менеджер протоколирования
- Менеджер буферов
- Планировщик заданий

Пояснение

Преза 25, сл 8

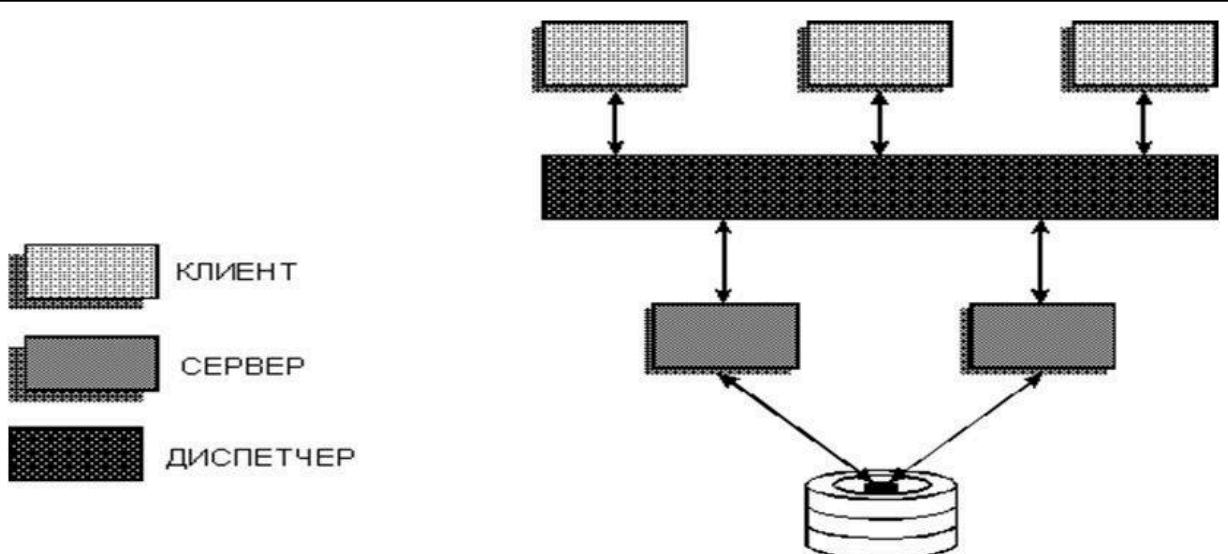
## Взаимодействие компонент СУБД



8

### Вопрос 3

Как называется клиент-серверная архитектура, изображенная на рисунке?



Выберите один ответ:

- Централизованная
- Виртуальный сервер
- Выделенный сервер
- Многопоточный сервер
- Многопотоковая мультисерверная

Ответ

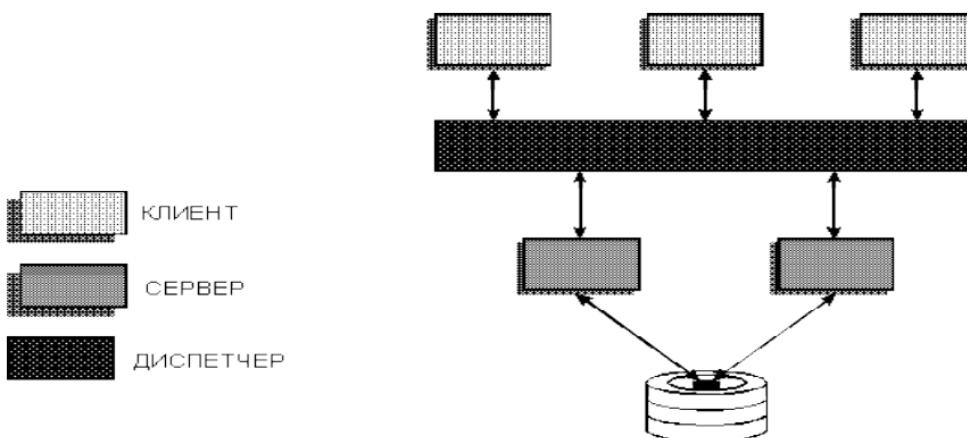
- Виртуальный сервер

Пояснение

Преза 29, сл 18

## ■ Архитектура с виртуальным сервером

- ❖ Предыдущая конфигурация никак не использует возможности многопроцессорной системы
- ❖ Проблема решается заменой выделенного сервера на диспетчер или **виртуальный сервер**



- ❖ В архитектуру системы добавляется новый слой, который размещается между клиентом и сервером, что увеличивает трату ресурсов на поддержку **баланса загрузки** (*load balancing*) , но и ограничивает возможности управления взаимодействием «клиент-сервер» – нет возможности установить приоритеты обслуживания клиентов.

## Вопрос 4

В базе данных ведется системный журнал на основе протокола REDO.

В момент отказа зафиксированы следующие записи в журнале.

```
<START T1>
<T1, A, 5>
<START T2>
<T2, B, 10>
<T1, D, 20>
<COMMIT T1>
<T2, C, 15>
<START T3>
<T3, E, 25>
<T3, F, 30>
ОТКАЗ СИСТЕМЫ
```

Какие действия будут выполнены в процессе восстановления ?

Выберите один или несколько ответов:

- В базу данных записывается значение A = 5
- В базу данных записывается значение F = 30
- В базу данных записывается значение D = 20
- В базу данных записывается значение B = 10
- В базу данных записывается значение E = 25
- В базу данных записывается значение C = 15

Ответ

- В базу данных записывается значение A = 5
- В базу данных записывается значение D = 20

Пояснение

Преза 25

Идём от начала к концу (всё противоположно алгоритму UNDO)

## Вопрос 5

Поставьте в соответствие нештатной ситуации в базе данных метод борьбы с ней

Отказ записи –

Перемежающийся отказ –

Разрушение дисковых носителей –

Сбой в работе системы –

Ошибочные элементы данных в базе данных –

Ответ

- Отказ записи – устойчивое хранилище

- Перемежающийся отказ – контрольная сумма
- Разрушение дисковых носителей – RAID массивы и копии БД
- Сбой в работе системы – системный журнал
- Ошибочные элементы данных в базе данных – ограничения целостности

## Пояснение

Про перемежающийся отказ. Преза 23, сл 4-5

### Перемежающийся отказ

- Диски обычно содержат **дополнительные избыточные** секторы, позволяющие определить корректность считанной информации
- Модель операции считывания/записи
  - ❖ Функция чтения возвращает пару  $(w, s)$ , где **w** – собственно данные, **s** – бит индикации состояния, свидетельствующий о том успешен ли результат операции: «истина», «ложь».
  - ❖ Если в результате чтения **s** возвращает «ложь», то можно повторить операцию (предел -100 раз)
  - ❖ Для проверки корректности операции записи можно повторно считывать записанный сектор и сопоставлять с содержимым того, что записывали. Если **s** – «истина», то считаем, что запись – успешна, если «ложь», то операцию записи следует повторить.
  - ❖ Нельзя полностью исключить ситуацию, когда **s** – «истина», но считанные данные не верны. Но можно **повысить вероятность** того, что **s** корректно отражает состояние считанных данных.

Про отказ записи. Преза 23, сл 6.

### Устойчивые хранилища

- Механизм контрольных сумм **позволяет обнаружить ошибки чтения/записи, но не позволяет их исправить.**
- Вполне вероятна ситуация, когда при выполнении попытки записи выясняется, что старое значение сектора уже утрачено, а **новое повторно считать не удается**
- Для преодоления проблемы на основе одного или нескольких дисков можно организовать так называемое **устойчивое хранилище**
- Общая идея – **хранить информацию каждого сектора в ПАРЕ.**  
Пара представляет содержимое одного сектора X.
- Пара  $X_L$  и  $X_R$  соответствует сектору X. Каждый член пары имеет достаточное количество битов четности и если операция чтения возвращает **значение сектора W и s** – «истина», то **w** верно отображает содержимое X.
- Алгоритм записи:
  - Сохранить X в  $X_L$ . Если бит индикации «ложь», то повторить операцию. Если после нескольких попыток не удается получить «истину», заменить  $X_L$  на другой сектор
  - Повторить действия для  $X_R$
- Алгоритм чтения:
  - Считать значение X из  $X_L$ . Если бит индикации «ложь», то повторить операцию несколько раз. Если на некоторой итерации удается получить «истину», считаем, что считанные значения из  $X_L$  содержат X
  - Если попытки чтения из  $X_L$  оказались безуспешными повторить действия для  $X_R$

С остальными и так понятно.

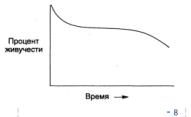
### Контрольная сумма

- Каждому сектору носителя соответствуют несколько **дополнительных бит – контрольная сумма (КС)**, которая зависит от содержимого сектора.
- Количество **единиц** в наборе битов данных, включая бит четности – **ВСЕГДА ЧЕТНО**.
- Если функция чтения обнаруживает **несоответствие между данными и КС**, то она возвращает параметр **s = «ложь»**, противном случае – «истина».
- Есть вероятность, что данные в секторе – некорректны, а КС соответствует корректным данным. Например, изменились сразу 2 бита – КС осталась правильной. Вероятность ошибки -  $\frac{1}{2^n}$
- Эту вероятность снижают увеличением битов КС.
- Например, можно предусмотреть 8 битов четности по 1 на соответствующий бит каждого байта данных в секторе.  
Тогда вероятность ошибки  $\frac{1}{2^{256}}$
- Если использовать **n** независимых битов четности, то вероятность незамеченных ошибок составляет  $1/2^n$
- И если под КС отвести **4 байта**, то ошибка **не будет обнаружена в одном из 4 миллиардов** случаев

### При разрушении носителя

### Восстановление при полном отказе диска

- Полный отказ диска может наступить из-за соприкосновения головок с поверхностью диска. Если не было копии диска, то информация утрачена.
  - Стратегии, привнесенные снизить риск потери данных при поломках основаны на модели избыточности и обозначаются общим термином RAID
  - RAID – redundant (*излишний*) array of independent disks
- Статистика отказов**
- Единичный жесткий диск имеет определенный срок службы
    - ❖ Измеряется в MTBF (*mean time between failure*, среднее время безотказной работы)
      - 50% дисков выходят из строя ~ 10 лет
    - ❖ Кривая живучести дисковых устройств
    - ❖ Пример
      - Если время наработки на отказ **одного диска** 750 000 часов, и массив состоит из 1000 дисков, то MTBF **массива становится** 750 000/1000 или 750 часов
  - Среднее время наработки на отказ – это, однако, не то же самое, что среднее время потери данных. RAID позволяет сохранять информацию даже при выходе из строя нескольких дисков.
  - RAID был введен для смягчения этой проблемы
  - RAID обеспечивает:
    - ❖ Повышение работоспособности
    - ❖ Более высокую доступность
    - ❖ Повышение производительности



## Вопрос 6

Компоненты интеллектуальной системы хранения данных EMC включают:

Выберите один или несколько ответов:

- Контроллер
- Внешний компьютер
- Средства подключения

- Система хранения

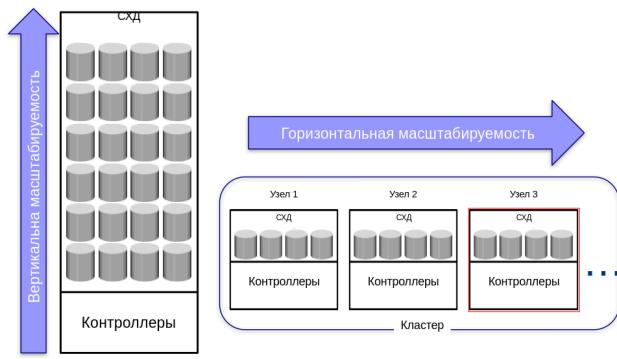
Ответ

- Контроллер
- Система хранения

Пояснение

Преза 26, сл 9-10

Сравнение вертикально и горизонтально масштабируемых архитектур



Что представляет собой блочная система хранения данных?

- Это вычислительная система с блочным доступом к ресурсам хранения
- Это система, которая может работать на базе вертикально или горизонтально масштабируемой архитектуры
- Система состоит из двух компонентов:
  - контроллер
  - СХД

## Вопрос 7

Укажите способы управления КЭШ памятью в интеллектуальных системах хранения данных

Выберите один или несколько ответов:

- Подкачка на нижнем уровне
- Замещение последних использованных страниц
- Замещение давно неиспользованных страниц
- Фоновый сброс
- Сброс на верхнем уровне
- Принудительный сброс
- Сброс на фиксированном уровне

Ответ

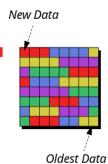
- Замещение последних использованных страниц
- Замещение давно неиспользованных страниц
- Фоновый сброс
- Сброс на верхнем уровне
- Принудительный сброс

Пояснение

Преза 26, сл 18-19

## Реализация кэш-памяти

- Кэш память может быть **выделенной** (ячейки резервированы под чтение или под запись) или **глобальной** (любые ячейки могут использованы для чтения/записи).
- Обычно под чтение отводится меньше ячеек.
- Может быть динамическое распределение в зависимости от рабочей загрузки
- **Алгоритмы управления сбросом**
  - Замещение давно неиспользованных (**LRU**)
    - Discards least recently used data  
Если страница содержит **данные для записи** на диск, то перед повторным использованием, она **предварительно сбрасывается на диск**
  - Последние использованные (**MRU**)
    - Discards most recently used data



## Управление кэш-памятью: уровни

- Управление сбросом «грязных» страниц (записанных в кэш, но не записанных на диск)
  - Фоновый сброс,
  - Сброс на верхнем уровне заполнения
  - Принудительный сброс
- Для максимальной производительности:
  - Обеспечить запас в кэш при всплесках ввода / вывода



## Вопрос 8

Отметьте типы интеллектуальных систем хранения

Выберите один или несколько ответов:

- Облачные системы
- Унифицированные системы
- Объектные системы
- Блочные системы
- Системы на виртуальных ресурсах
- Файловые системы

Ответ

- Унифицированные системы
- Объектные системы
- Блочные системы
- Файловые системы

Пояснение

Преза 26, сл 7

## Типы интеллектуальных систем хранения

- Блочные системы хранения
- Файловые системы хранения
- Объектные системы хранения
- Унифицированные системы хранения

### Вопрос 9

Укажите что из перечисленного относится к внутризапросному параллелизму в системах параллельных баз данных

Выберите один или несколько ответов:

- сегментный конвейер
- блочный конвейер
- синхронный конвейер
- горизонтальный параллелизм
- асинхронный конвейер

Ответ

- синхронный конвейер
- асинхронный конвейер
- горизонтальный параллелизм

Пояснение

Преза 29, сл 23, 21 (слайды подогнаны под ответы)

Vertикальный (конвейерный) параллелизм

Недостатком *синхронного конвейера* является блокирующий характер операций конвейерной обработки отдельных гранул

Для преодоления указанного недостатка может быть использован *асинхронный конвейер*, в котором поставщик и потребитель работают независимо друг от друга, а данные передаются через некоторый буфер.

Степень конвейерного параллелизма в любом случае ограничена количеством операций, вовлекаемых в конвейер.

При этом для реляционных систем баз данных длина конвейера редко превышает 10 операций

- ❖ горизонтальный параллелизм (кустовой)  
предполагает параллельное выполнение независимых поддеревьев дерева, представляющего план запроса.
- ❖ Очень трудно обеспечить, чтобы два подплана одного плана начали генерировать выходные данные в правильное время и в правильном темпе.
- ❖ Правильный не означает ОДИНАКОВЫЙ.
- ❖ Например для случая, когда входные потоки соединения слиянием имеют различные размеры это весьма трудно.

Кустовой параллелизм редко используется на практике.

## Вопрос 10

Укажите цифрой число бит, которое нужно отвести под контрольную сумму, чтобы вероятность незамеченной ошибки была равна 1/128

Ответ

- 7

Пояснение

Считаем так:  $1 / (2 ^ 7) = 1 / 128$

Преза 23, сл 5

## Контрольная сумма

- Каждому сектору носителя соответствуют несколько **дополнительных бит – контрольная сумма (КС)**, которая зависит от содержимого сектора.
- Количество **единиц** в наборе битов данных, включая бит четности – ВСЕГДА ЧЕТНО.
- Если функция чтения обнаруживает **несоответствие между данными и КС**, то она возвращает параметр  $s = \text{«ложь»}$ , противном случае – «истина».
- Есть вероятность, что данные в секторе – некорректны, а КС соответствует корректным данным. Например, изменились сразу 2 бита – КС осталась правильной. Вероятность ошибки -  $\frac{1}{2}$
- Эту вероятность снижают увеличением битов КС.
- Например, можно предусмотреть 8 битов четности по 1 на соответствующий бит каждого байта данных в секторе.  
Тогда вероятность ошибки  $\frac{1}{2^{256}}$ .
- Если использовать  $n$  независимых битов четности, то вероятность незамеченных ошибок составляет  $1/2^n$
- И если под КС отвести **4 байта**, то ошибка **не будет обнаружена в одном из 4 миллиардов** случаев

## Вопрос 11

Выберите предложения, характеризующие локальную автономию для распределенных СУБД.

Выберите один или несколько ответов:

- На каждом узле обеспечивается непрерывное функционирование распределенной базы данных
- На каждом узле распределенной базы данных возможен вход в систему
- Безопасность и целостность хранения данных на некотором узле X осуществляется СУБД, установленной на этом узле

- На каждом узле сети имеется репликация каталога распределенной базы данных
- Функционирование любого узла X распределенной базы данных не зависит от выполнения операций на любом другом узле Y
- Управление и владение данными на некотором узле X осуществляется СУБД, установленной на этом узле

Ответ

- Безопасность и целостность хранения данных на некотором узле X осуществляется СУБД, установленной на этом узле
- Функционирование любого узла X распределенной базы данных не зависит от выполнения операций на любом другом узле Y
- Управление и владение данными на некотором узле X осуществляется СУБД, установленной на этом узле

Пояснение

Преза 27 , сл 9

## Локальная автономия

- Локальная автономия означает, что функционирование любого узла X **НЕ зависит** от выполнения операций на любом другом узле Y.
- Управление и владение данными осуществляется локально вместе с локальным ведением учета
- Безопасность, целостность и структура хранения остаются под контролем локального узла
- Узлы следует делать автономными в **максимально возможной степени**

## Вопрос 12

Отметьте термины или операторы, которые используются при описании ограничения целостности в базе данных?

Выберите один или несколько ответов:

- Update
- Assertion
- Внешний ключ
- Вторичный ключ
- Тип данных
- Check()
- Index
- Первичный ключ
- Drop
- Trigger

Ответ

- Assertion
- Внешний ключ
- Тип данных
- Check()
- Первичный ключ
- Trigger

Пояснение

Преза 25 сл 3

- Средства СУБД для выявления ошибок – ограничения базы данных
  - ❖ Тип данных
  - ❖ Шаблон представления
  - ❖ Primary key
  - ❖ Foreign key
  - ❖ Check ()
  - ❖ Assertions
  - ❖ Triggers

### Вопрос 13

Укажите свойства протокола UNDO

Выберите один или несколько ответов:

- Менеджер восстановления сканирует журнал от начала к концу
- Менеджер восстановления сканирует журнал от конца к началу
- Запись <COMMIT T> следует помещать в протокол только после сбросывания всех измененных значений элементов базы данных на диск
- В запись вида <T, X, v> означает, что транзакция T изменила прежнее значение v элемента базы данных X
- В запись вида <T, X, v> означает, что транзакция T сохранила новое значение v элемента базы

данных X

- Записи обновления и запись <COMMIT T> следует помещать в протокол до .сбрасывания. всех измененных значений элементов базы данных на диск
- Работу протокола определяют 2 правила
- Работу протокола определяет 1 правило

Ответ

- Менеджер восстановления сканирует журнал от конца к началу
- Запись <COMMIT T> следует помещать в протокол только после .сбрасывания. всех измененных значений элементов базы данных на диск
- В запись вида <T, X, v> означает, что транзакция T изменила прежнее значение v элемента базы данных X
- Работу протокола определяют 2 правила

Пояснение

Преза 25, сл 16, 17

## Протоколирование в режиме “UNDO”

- Протокол или **журнал транзакций** (log) – файл, **открытый только для пополнения**
- Менеджер протоколирования сохраняет в журнале информацию о каждом событии при исполнении транзакций
- Блок протокола сначала создается в ОП и обрабатывается менеджером буферов, как и все другие блоки. Сохранение происходит в наиболее «подходящие» моменты времени.
- Записи протокола:  
<START T> – выполнение транзакции T начато  
<COMMIT T> – транзакция T успешно завершена  
<ABORT T> – выполнение транзакции T преждевременно остановлено
- Записи обновления <T, X, v>.

Смысл – транзакция T изменила **прежнее значение V** элемента базы данных X

### Правила протоколирования в режиме “UNDO”

- Правила регламентируют деятельность менеджера буферов
- U1:** Если транзакция T модифицирует элемент базы данных - X, запись обновления вида <T, X, v> должна быть занесена в протокол **до сохранения** нового значения элемента X **на диске**.
- U2:** При фиксации результатов транзакции T запись <COMMIT T> следует помещать в протокол **только после «сбрасывания»** всех измененных значений элементов базы данных на диск. Интервал между дисковыми операциями сохранения данных и записи <COMMIT T> должен быть **настолько КОРОТКИМ** насколько это возможно.
- Таким образом, информация, имеющая отношение к действиям в рамках одной транзакции должна **сохраняться на диске** в следующем порядке:
    - ✓ Записи протокола, свидетельствующие об изменениях, внесенных в БД
    - ✓ Новые значения элементов БД как таковых
    - ✓ Запись :<COMMIT T> протокола

16

17

## Вопрос 14

В базе данных ведется системный журнал на основе протокола UNDO.

В момент отказа зафиксированы следующие записи в журнале.

```
<START T4>
<T4, A, 1>
<START T5>
<T5, B, 2>
<START CKPT (T4 ,T5)>
<T5, C, 5>
<START T3>
<T4, D, 3>
<COMMIT T4>
<T3, E, 5>
<COMMIT T5>
<END CKPT>
```

<T3, F, 4>

## ОТКАЗ СИСТЕМЫ

Какие действия будут выполнены в процессе восстановления ?

Выберите один или несколько ответов:

- В базу данных запишется значение C=5
- В базу данных запишется значение F=4
- В базу данных запишется значение E=5
- В базу данных запишется значение B=2
- В базу данных запишется значение D=3
- В базу данных запишется значение A=1

Ответ

- В базу данных запишется значение F=4
- В базу данных запишется значение E=5

Пояснение

Преза 25 , сл 27

## Восстановление по протоколу UNDO и ДКТ

Пример 2.

<START T<sub>1</sub>>

<T<sub>1</sub>, A, 5>

<START T<sub>2</sub>>

<T<sub>2</sub>, B, 10>

<START CKPT (T<sub>1</sub>, T<sub>2</sub>)>

<T<sub>2</sub>, C, 15>

<START T<sub>3</sub>>

<T<sub>1</sub>, D, 20>

<COMMIT T<sub>1</sub>>

<T<sub>3</sub>, E, 25>

**ОТКАЗ СИСТЕМЫ**

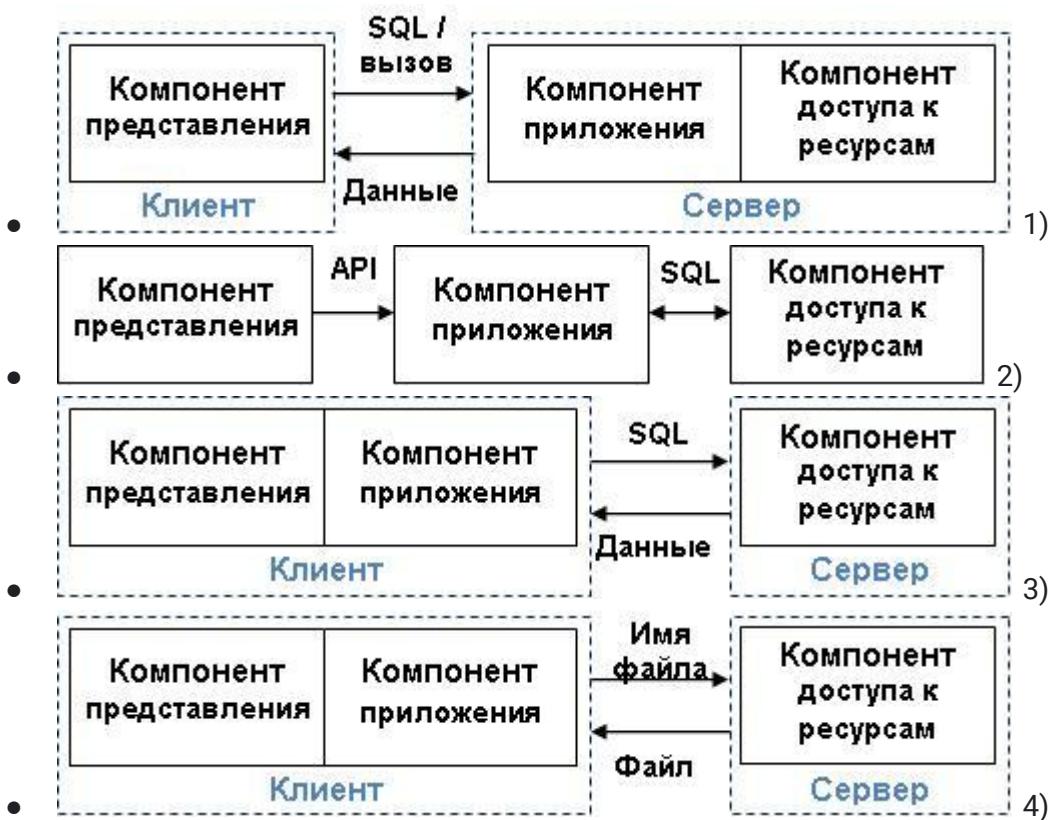
Сканируя журнал обнаруживаем, что

- ✓ Транзакция T<sub>3</sub> и T<sub>2</sub> не завершены и подлежат отмене.  
**Возвращаем E = 25 и C = 15**
- ✓ Так как первая встретившая запись ДКТ <START CKPT (T<sub>1</sub>, T<sub>2</sub>)>, то **анализируем транзакции T<sub>1</sub> и T<sub>2</sub>**
- ✓ T<sub>1</sub> завершена, так как был <COMMIT T<sub>1</sub>>
- ✓ Осталось **понять, что с T<sub>2</sub>**. Сканируем дальше
- ✓ **Возвращаем B = 10**
- ✓ **Встретили <START T<sub>2</sub>>.**
- ✓ **Дальше сканировать нет смысла, так как по условию введения КТ все транзакции до ее введения должны быть зафиксированы.**

27

## Вопрос 15

Сопоставьте архитектуру клиент-сервер с названием



Ответ

- 1) сервер баз данных
- 2) сервер приложений
- 3) удаленный доступ
- 4) файловый сервер

Пояснение

Преза 28, сл 10 про файловый сервер

### Модель файлового сервера

- Характерна для ранних СУБД



- FoxPro, Clipper, Clarion, Paradox
- Работает под сетевой ОС
- На узлах сети работают **полноценные экземпляры** СУБД
- Базы данных на разделяемых файлах
- Высокий сетевой трафик
- Узкий спектр операций манипулирования
- Отсутствие гибких средств безопасности (уровень файла)

Преза 28, сл 11 про удаленный доступ

### Модель удаленного доступа (RDA)

- Как правило SQL сервер

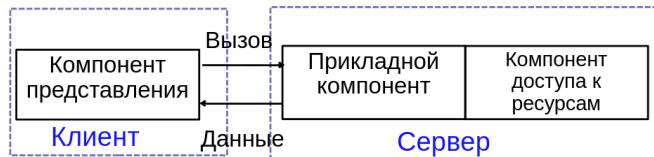


- Доступ к ресурсам: SQL оператор , либо вызовы специальной библиотеки функций (если есть API )
- Сервер возвращает блок данных по запросу (не весь файл)
- Инициатор запроса – КЛИЕНТ, Сервер - ПАССИВЕН
- Уменьшается загрузка сети
- **SQL сервер** – основное достоинство, стандартизация взаимодействия
- Но усложнено администрирование системы

Преза 28, сл 12 про сервер баз данных

### Модель сервера баз данных (DBS)

- Сервер - как правило SQL сервер

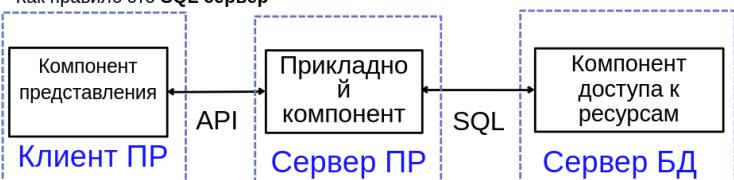


- Доступ к ресурсам: механизм *хранимых процедур (ХП)*
- Возращает сервер блок данных по запросу/после выполнения
- Инициатор запроса – КЛИЕНТ, Сервер - АКТИВЕН
- Достоинства
  - возможность централизованного администрирования прикладных функций
  - снижение сетевого трафика (вместо SQL – запросов, только вызов ХП)
  - возможность разделения процедуры между несколькими приложениями
  - экономия ресурсов компьютера за счет один раз созданного плана запроса
- Недостатки
  - ограниченность средств для написания хранимых процедур (PL/SQL)

Преза 28, сл 13 про сервер приложений

### Модель сервера приложений (AS)

- Как правило это SQL сервер



- Клиент ПР – клиент приложения
- Сервер ПР – группа процессов, выполняющих прикладные функции (бизнес процессы прикладной области)
- Сервер БД – группа процессов, реализующих операции над информационными ресурсами
- Достоинства
  - изолированность прикладного компонента
  - использование средств многозадачности ОС для прикладного компонента
  - стандартизация интерфейсов с компонентами – основа для монитора обработки транзакций

## Вопрос 16

Транзакция обязана обладать следующими свойствами:

Выберите один или несколько ответов:

- Завершенность
- Фиксированность
- Согласованность
- Долговечность
- Изолированность
- Атомарность
- Непротиворечивость

Ответ

- Согласованность
- Долговечность
- Изолированность
- Атомарность

Пояснение

Преза 12, сл 13 ACID

## Свойства транзакции

### ■ Автономность (Atomicity). Все или ничего!

Либо фиксируются все операции транзакции (commit), либо откат (rollback)

### ■ Согласованность (Consistency). Шире понятия целостности.

Согласованность может быть реализована на уровне бизнес логики. Например, операция « списание » должна соответствовать некоторой операции « зачисление »

### ■ Изолированность (Isolation).

При выполнении транзакций параллельно **одни** транзакции не должны оказывать влияние на результат выполнения **других** транзакций.

### ■ Долговечность (Durability).

Если пользователь получил подтверждение от системы, что транзакция выполнена, он может быть уверен, что сделанные им изменения не будут отменены из-за какого-либо сбоя.

### Вопрос 17

В базе данных ведется системный журнал на основе протокола UNDO.

В момент отказа зафиксированы следующие записи в журнале.

```
<START T1>
<T1, A, 5>
<START T2>
<T2, B, 10>
<START CKPT (T1 ,T2)>
<T2, C, 15>
<START T3>
<T1, D, 20>
<COMMIT T1>
<T3, E, 25>
ОТКАЗ СИСТЕМЫ
```

Какие действия будут выполнены в процессе восстановления ?

Выберите один или несколько ответов:

- В базу данных запишется значение B=10
- В базу данных запишется значение E=25
- В базу данных запишется значение A=5

- В базу данных запишется значение D=20
- В базу данных запишется значение C=15

Ответ

- В базу данных запишется значение B=10
- В базу данных запишется значение E=25
- В базу данных запишется значение C=15

Пояснение

Преза 25 , сл 27

### Восстановление по протоколу UNDO и ДКТ

Пример 2.

<START T<sub>1</sub>>

<T<sub>1</sub>, A, 5>

<START T<sub>2</sub>>

<T<sub>2</sub>, B, 10>

<START CKPT (T<sub>1</sub>, T<sub>2</sub>)>

<T<sub>2</sub>, C, 15>

<START T<sub>3</sub>>

<T<sub>1</sub>, D, 20>

<COMMIT T<sub>1</sub>>

<T<sub>3</sub>, E, 25>

**ОТКАЗ СИСТЕМЫ**

Сканируя журнал обнаруживаем, что

- ✓ Транзакция T<sub>3</sub> и T<sub>2</sub> не завершены и подлежат отмене.  
**Возвращаем E = 25 и C = 15**
- ✓ Так как первая встретившая запись ДКТ <START CKPT (T<sub>1</sub>, T<sub>2</sub>)>, то анализируем транзакции T<sub>1</sub> и T<sub>2</sub>
- ✓ T<sub>1</sub> завершена, так как был <COMMIT T<sub>1</sub>>
- ✓ Осталось понять, что с T<sub>2</sub>. Сканируем дальше
- ✓ Возвращаем B = 10
- ✓ Встретили <START T<sub>2</sub>>.
- ✓ Дальше сканировать нет смысла, так как по условию введения КТ все транзакции до ее введения должны быть зафиксированы.

27

### Вопрос 18

Отметьте ключевые проблемы, влияющие на производительность систем параллельных баз данных

Выберите один или несколько ответов:

- Количество дисковых массивов
- Балансировка загрузки
- Количество процессов модификации данных
- Доступность данных
- Организация блокировок
- Когерентность КЭШей
- Межпроцессорные коммуникации

Ответ

- Балансировка загрузки
- Организация блокировок
- Когерентность КЭШей
- Межпроцессорные коммуникации

Пояснение  
Преза 29, сл 6

## Производительность (1)

Ключевые проблемы (**накладные расходы**):

- **Межпроцессорные коммуникации;** (порождают трафик значительного объема, связанный с передачей сообщений от одного процессора – другому)
- **Когерентность КЭШей;**
  - обращение к диску примерно в  $10^5\text{-}10^6$  раз медленнее, чем обращение к оперативной памяти,
  - кэширование страниц с фрагментами базы данных,
  - согласование изменения одного и того же фрагмента в кэшах различных процессоров
- **Организация блокировок** (если различные процессоры обрабатывают одни и те же объекты базы данных (отношения, кортежи и др.), то нам необходимо поддерживать глобальную таблицу блокировок, используемую всеми процессорами)

## Производительность (2)

### ► Балансировка загрузки

- СУБД должна разбивать запрос на **параллельные агенты** и распределять их по процессорам таким образом, чтобы обеспечивалась **равномерная загрузка** всех задействованных процессоров
- величина **перекоса** способна снизить эффективность распараллеливания реляционных операций, особенно соединения и сортировки

**Перекос значений атрибута** – это когда некоторые значения для определенного атрибута встречаются значительно чаще, чем остальные.

При наличии перекоса данных **ускорение**, достигаемое при параллельном выполнении операций соединения, может быть ограничено **катастрофическим образом** по причине перегрузки одних процессоров и недогрузки других.

## Вопрос 19

Укажите методы доступа к данным, используемые в интеллектуальных системах хранения данных

Выберите один или несколько ответов:

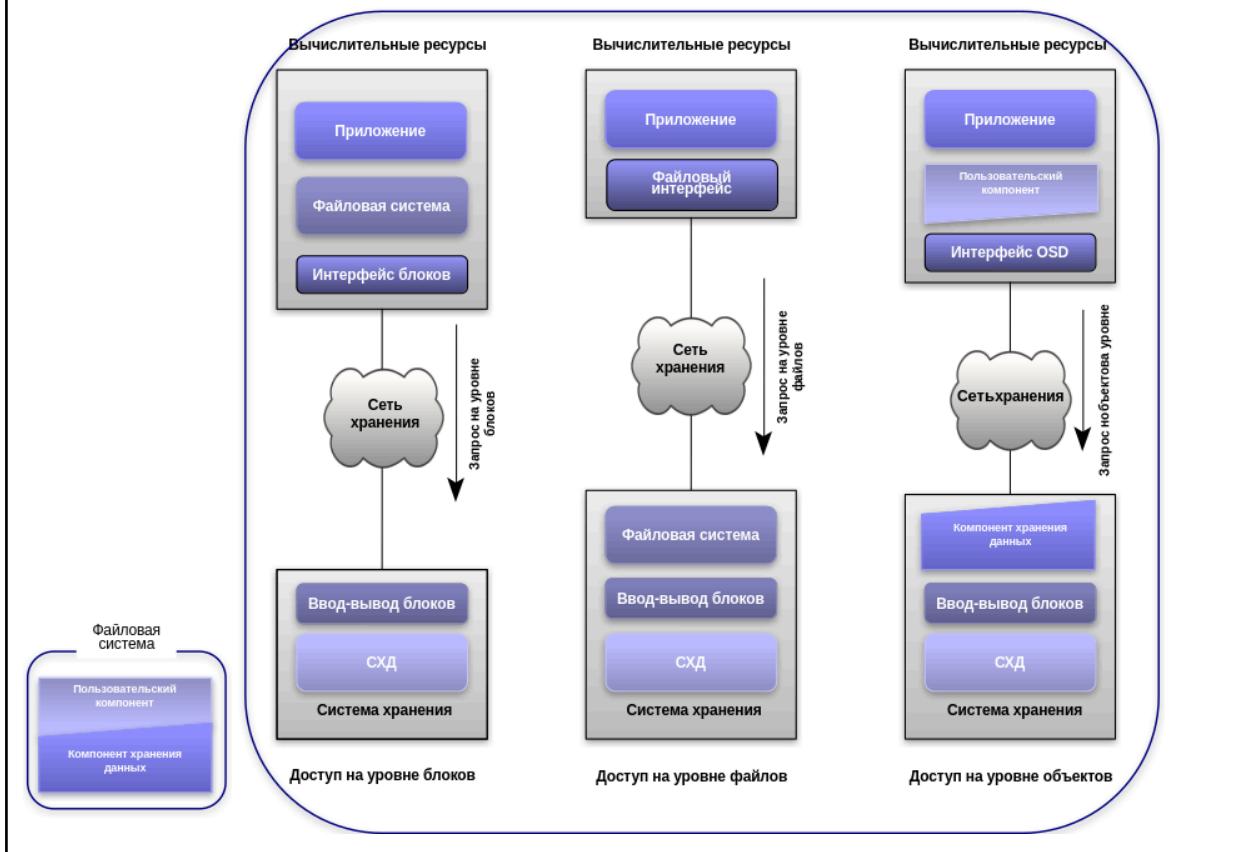
- на уровне файлов
- на уровне блоков
- на уровне записей
- на уровне секторов
- на уровне дорожек
- на уровне цилиндров
- на уровне объектов

Ответ

- на уровне файлов
- на уровне блоков
- на уровне объектов

Пояснение  
Преза 26, сл 8

## Методы доступа к данным



### Вопрос 20

Компоненты стандарта ODMG для объектной модели данных включают:

Выберите один или несколько ответов:

- Архитектура объектной СУБД
- Язык манипулирования данными (OML)
- Объектная модель
- Язык определения объектов ODL
- Объектный язык запросов OQL
- Модель связывания с языками программирования
- Модель хранения объектов

Ответ

- Язык манипулирования данными (OML)
- Объектная модель
- Язык определения объектов ODL
- Объектный язык запросов OQL

Пояснение

Преза 16, сл 11

## Компоненты стандарта ODMG

- **Объектная модель**, которую должны поддерживать ООСУБД
- **Язык определения объектов ODL**, опирающийся на эту модель
- **Объектный язык запросов OQL** (SQL&Object Model)
- **Язык манипулирования данными (OML)** – отображение объектной модели ODMG в ООЯП (Smalltalk, C++) – часто называют связыванием с ЯП

### Вопрос 21

Установите соответствие операций и видов фрагментации

- |                                                       |
|-------------------------------------------------------|
| • Вертикальная фрагментация - прямая операция ---     |
| • Горизонтальная фрагментация - прямая операция ---   |
| • Горизонтальная фрагментация - обратная операция --- |
| • Вертикальная фрагментация - обратная операция ---   |

Ответ

- |                                                                            |
|----------------------------------------------------------------------------|
| • Вертикальная фрагментация - прямая операция --- проекция                 |
| • Горизонтальная фрагментация - прямая операция --- селекция               |
| • Горизонтальная фрагментация - обратная операция --- объединение          |
| • Вертикальная фрагментация - обратная операция --- натуральное соединение |

Пояснение

Преза 27, сл 17

## Операции

- Горизонтальная фрагментация – операция реляционной алгебры «селекция».
- Обратная операция – «объединение».
- Вертикальная фрагментация – операция реляционной алгебры «проекция».
- Необходимость реконструкции исходного отношения БЕЗ потерь.
- Обратная операция – «натуральное соединение».

### Вопрос 22

Объектная модель ODMG расширяет модель Object Management Group (OMG) и вводит:

Выберите один или несколько ответов:

- Классы
- Уникальный идентификатор
- Типы "коллекции"
- Транзакции
- Свойство "relationship"

Ответ

- Типы "коллекции"
- Транзакции
- Свойство "relationship"

Пояснение

Преза 16, сл 12

## Объектная модель ODMG

- **Объектная модель** - унифицированная основа всего стандарта.
- Она расширяет объектную модель консорциума OMG за счет введения таких свойств как **связи** и понятия **транзакции** для обеспечения функциональности, требуемой при взаимодействии с базами данных.
- Модель определяет 2 вида примитивов моделирования:  
**объекты** (изменчивые типы), **литералы** (неизменчивые типы)
- Примитивы обоих видов могут быть: **атомарные, коллекции**
- Каждый объект имеет **уникальный идентификатор**
- Литералы **не имеют уникального идентификатора**, т.е. самоидентифицируемы
- Объекты и литералы обладают **типами**
- Между **типами** могут устанавливаться **бинарные связи** (вида 1:1, 1:n и n:m)
- Состояние элементов данного типа определяется соответствующими им **значениями его свойств** (общих для всего типа) — **атрибутов или бинарных связей** с экземплярами других типов.
- Поведение экземпляров типа определяется **набором операций**, задаваемых их сигнатурами, и этот набор также является общим для всех экземпляров данного <sup>12</sup> типа.

### Вопрос 23

В базе данных ведется системный журнал на основе протокола UNDO.

В момент отказа зафиксированы следующие записи в журнале.

```
<START T1>
<T1, A, 8>
<START T2>
<T2, C, 25>
<COMMIT T1>
<COMMIT T2>
<CKPT>
<START T4>
<T4, A, 1>
<START T5>
<T5, B, 2>
<T5, C, 5>
<T4, D, 3>
<COMMIT T4>
<COMMIT T5>
<CKPT>
<START T3>
```

<T3, E, 25>

<T3, F, 47>

## ОТКАЗ СИСТЕМЫ

Какие действия будут выполнены в процессе восстановления ?

Выберите один или несколько ответов:

- В базу данных запишется значение C=5
- В базу данных запишется значение F=47
- В базу данных запишется значение A=8
- В базу данных запишется значение B=2
- В базу данных запишется значение D=3
- В базу данных запишется значение E=25

Ответ

- В базу данных запишется значение F=47
- В базу данных запишется значение E=25

Пояснение

Преза 25 , сл 27

## Восстановление по протоколу UNDO и ДКТ

Пример 2.

<START T<sub>1</sub>>

<T<sub>1</sub>, A, 5>

<START T<sub>2</sub>>

<T<sub>2</sub>, B, 10>

<START CKPT (T<sub>1</sub>, T<sub>2</sub>)>

<T<sub>2</sub>, C, 15>

<START T<sub>3</sub>>

<T<sub>1</sub>, D, 20>

<COMMIT T<sub>1</sub>>

<T<sub>3</sub>, E, 25>

ОТКАЗ СИСТЕМЫ

Сканируя журнал обнаруживаем, что

- ✓ Транзакция T<sub>3</sub> и T<sub>2</sub> не завершены и подлежат отмене.  
Возвращаем E = 25 и C = 15
- ✓ Так как первая встретившая запись ДКТ <START CKPT (T<sub>1</sub>, T<sub>2</sub>)>, то анализируем транзакции T<sub>1</sub> и T<sub>2</sub>
- ✓ T<sub>1</sub> завершена, так как был <COMMIT T<sub>1</sub>>
- ✓ Осталось понять, что с T<sub>2</sub>. Сканируем дальше
- ✓ Возвращаем B = 10
- ✓ Встретили <START T<sub>2</sub>>.
- ✓ Дальше сканировать нет смысла, так как по условию введения КТ все транзакции до ее введения должны быть зафиксированы.

27

Вопрос 23 из 2021 я хз где искать

Отметьте этапы сортировки отношения БД, если все отношение не помещается целиком в оперативную память.

Выберите один или несколько ответов:

- Параллельная поблочная обработка всех отсортированных ПОДСПИСКОВ
- Последовательное слияние каждой пары отсортированных ПОДСПИСКОВ
- Отсортировать записи в ОП

- |                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Заполнить свободную ОП блоками отношения подлежащего сортировке</li><li>• Сохранить отсортированный ПОДСПИСОК в новом участке диска</li></ul> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Ответ

- Параллельная поблочная обработка всех отсортированных ПОДСПИСКОВ
- Последовательное слияние каждой пары отсортированных ПОДСПИСКОВ
- Отсортировать записи в ОП
- Заполнить свободную ОП блоками отношения подлежащего сортировке
- Сохранить отсортированный ПОДСПИСОК в новом участке диска

Пояснение

Преза , сл

## Вопрос 24

Отметьте свойства хранилищ данных

Выберите один или несколько ответов:

- Поддержка хронологии
- Неизменяемость данных
- Интегрированность
- Ориентация на сложные структуры
- Поддержка крупноблочной модификации
- Поддержка линейного упорядочения
- Ориентация на предметную область

Ответ

- Поддержка хронологии
- Ориентация на предметную область
- Неизменяемость данных
- Интегрированность

Пояснение

Преза 30, сл 15

### Свойства хранилищ

- Ориентация на предметную область.  
Представление для аналитика
  - ❖ OLTP – сделка. Заявки
  - ❖ OLAP – клиенты, товары, производители
- Интегрированность:
  - ❖ разные домены для одинаковых атрибутов («январь», «янв», 01)
  - ❖ нужно проверять на непротиворечивость и целостность
- Неизменяемость данных
  - ❖ неактуальны схемы блокировок,
  - ❖ нет понятия транзакции
- Поддержка хронологии
  - ❖ Ключевые атрибуты: «дата» или «время»

## Вопрос 25

В базе данных ведется системный журнал на основе протокола UNDO.

В момент отказа зафиксированы следующие записи в журнале.

```
<START T1>
<T1, A, 5>
<START T2>
<T2, B, 10>
<T1, D, 20>
<COMMIT T1>
<T2, C, 15>
<START T3>
<T3, E, 25>
<T3, F, 30>
<COMMIT T3>
ОТКАЗ СИСТЕМЫ
```

Какие действия будут выполнены в процессе восстановления ?

Выберите один или несколько ответов:

- В базу данных записывается значение Е = 25
- В базу данных записывается значение В = 10
- В базу данных записывается значение С = 15
- В базу данных записывается значение D = 20
- В базу данных записывается значение А = 5
- В базу данных записывается значение F = 30

Ответ

- В базу данных записывается значение В = 10
- В базу данных записывается значение С = 15

Пояснение

Преза 25 , сл 27

### Восстановление по протоколу UNDO и ДКТ

Пример 2.

```
<START T1>
<T1, A, 5>
<START T2>
<T2, B, 10>
<START CKPT (T1, T2)>
<T2, C, 15>
<START T3>
<T1, D, 20>
<COMMIT T1>
<T3, E, 25>
ОТКАЗ СИСТЕМЫ
```

Сканируя журнал обнаруживаем, что

- ✓ Транзакции  $T_3$  и  $T_2$  не завершены и подлежат отмене.
- ✓ Возвращаем  $E = 25$  и  $C = 15$
- ✓ Так как первая встретившая запись ДКТ  $<START CKPT (T_1, T_2)>$ , то анализируем транзакции  $T_1$  и  $T_2$
- ✓  $T_1$  завершена, так как был  $<COMMIT T_1>$
- ✓ Осталось понять, что с  $T_2$ . Сканируем дальше
- ✓ Возвращаем  $B = 10$
- ✓ Встретили  $<START T_2>$ .
- ✓ Дальше сканировать нет смысла, так как по условию введения КТ все транзакции до ее введения должны быть зафиксированы.

## Вопрос 26

Поставьте в соответствие систему и свойство

- OLAP
- OLTP

Ответ

- OLAP – аналитическая обработка, прогнозирование, моделирование
- OLTP – оперативный поиск, несложные запросы

Пояснение

Преза 30, сл 2, 5

### Системы операционной обработки - OLTP

- К середине 80-х годов завершился **первый этап** оснащения бизнеса и органов государственного управления средствами вычислительной техники
- В связи с совершенствованием технологий записи и хранения данных к началу 90-х годов на людей **обрушились колоссальные потоки информационной руды** в самых различных областях.
- Деятельность любого предприятия (комерческого, производственного, медицинского, научного и т.д.) теперь **сопровождается регистрацией и записью всех подробностей** его деятельности.
- Это делается в системах операционной обработки данных (**Online Transaction Processing - OLTP**)
- Накопились **большие объемы** информации.
- Стало ясно, что без продуктивной переработки **потоки сырых данных** образуют никому не нужную свалку.

### Системы поддержки принятия решений - СППР

- Системы поддержки принятия решений – **вычислительные системы для анализа данных и получения информации для принятия решений в сфере управления** (СППР). OnLine Analytical Processing - **OLAP**, BI (Business Intelligence, «бизнес-интеллект»)
- Основная задача СППР - предоставить аналитикам инструмент для выполнения анализа данных.
- Необходимо отметить, что для эффективного использования СППР ее пользователь - аналитик должен обладать соответствующей квалификацией.
- Система **не генерирует правильные решения**, а только предоставляет аналитику данные в соответствующем виде (отчеты, таблицы, графики и т. п.) для изучения и анализа.
- СППР решают три основные задачи: **сбор, хранение и анализ хранимой информации**.

Преза 30 сл 16

### Свойства данных в OLTP и OLAP

Свойство	OLTP	OLAP
Назначение данных	Оперативный поиск, несложные запросы	Аналитическая обработка, прогнозирование, моделирование
Уровень агрегации	Детализированные	Агрегированные
Период хранения	От нескольких месяцев до года	От нескольких лет до десятков лет
Частота обновления	Высокая, обновление мелкими порциями	Малая, обновление большими порциями

### Вопрос 12 поясн 21 год

В распределенной базе данных применяется репликация данных.

Определение условие завершения модифицирующей распределенной транзакции.

Выберите один ответ:

- Обновлены 90% копий объектов
- Обновлены главные копии объектов
- Обновлены все копии объектов
- Обновлены 50% копий объектов
- Обновлены первичные копии объектов

Ответ

- Обновлены первичные копии объектов

Пояснение

Преза , сл

### Вопрос 2024г

Отметьте виды параллелизма в системах параллельных баз данных

Выберите один или несколько ответов:

- фрагментарный
- внутритранзакционный
- межзапросный
- межпроцессорный параллелизм
- внутрипотоковый параллелизм

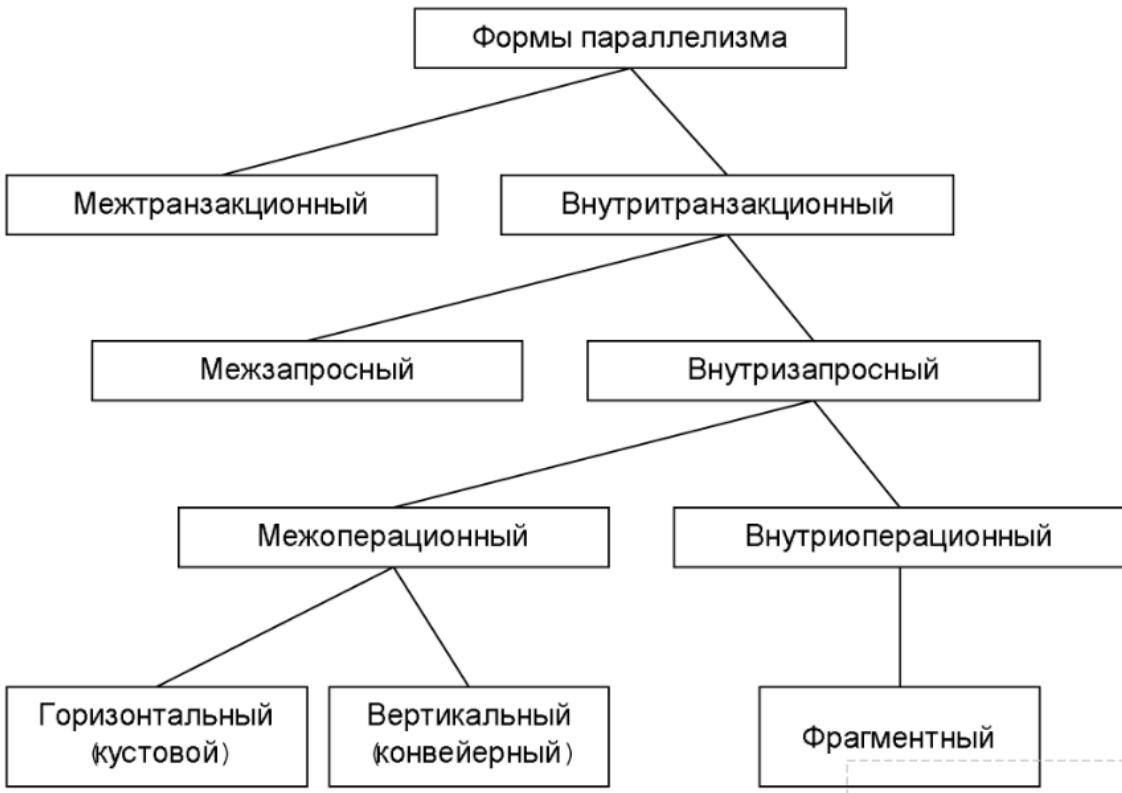
Ответ

- внутритранзакционный
- межзапросный
- фрагментарный

Пояснение

Преза 29, сл 15

# Формы параллелизма транзакций



## ТЕСТ 6

### Вопрос 4

Пусть даны следующие отношения R и S.

Отношение R ( A, B, C), где атрибут A - первичный ключ, атрибут C - внешний ключ (на атрибут С отношения S )

Отношение R содержит набор кортежей:

- 1, 201, 4
- 2, 302, 3

Отношение S (C, D), где атрибут C- первичный ключ

Отношение S содержит набор кортежей:

- 3, сторож
- 4, механик

Отметьте операции, которые можно сделать при текущем наборе данных в отношениях.  
Выберите один или несколько ответов:

- Update R  
set C=6  
where A=2

- Insert into R values( 3, 401, 5)
- Insert into S values( 2, электрик)
- Insert into R values( 3, 201, 3)
- Update R  
set B=501  
where A=1
- Insert into R values( 2, 501, 3)

Ответ

- Insert into S values( 2, электрик)
- Insert into R values( 3, 201, 3)
- Update R  
set B=501  
where A=1

Пояснение

- Update R  
set C=6 // C = 6 – где? Нет такого  
where A=2
- Insert into R values( 3, 401, 5) // C = 5 такого нет
- Insert into S values( 2, электрик) // нет id=2, ничего не воруем, норм
- Insert into R values( 3, 201, 3) // всё норм, ниче не воруем
- Update R  
set B=501  
where A=1 // всё норм, ниче не нарушаем
- Insert into R values( 2, 501, 3) // уже есть id=2, так что вставить такое не можем

## Вопрос 5

Что может являться событием в триггере ?

Выберите один или несколько ответов:

- Завершение транзакции
- Операция DELETE FROM <table>
- Операция UPDATE <table>
- Операция INSERT INTO <table>
- Операция ALTER TABLE <table>
- Операция CREATE TABLE <table>
- Операция DROP <table>

Ответ

- Завершение транзакции
- Операция DELETE FROM <table>
- Операция UPDATE <table>
- Операция INSERT INTO <table>

Пояснение

Преза , сл  
Преза 11, сл 13-16

К числу **допустимых событий (Event)** обычно относят операции **INSERT, DELETE и UPDATE** указанного **отношения**. Во многих СУБД "событием", на которое способен отреагировать триггер, разрешено считать и факт завершения транзакции.

### Вопрос 9

Выделите варианты, означающие компоненты СУБД

Выберите один или несколько ответов:

- Менеджер буферов
- Компилятор запросов
- Менеджер протоколирования
- Менеджер транзакций
- Менеджер процессов
- Менеджер схемы

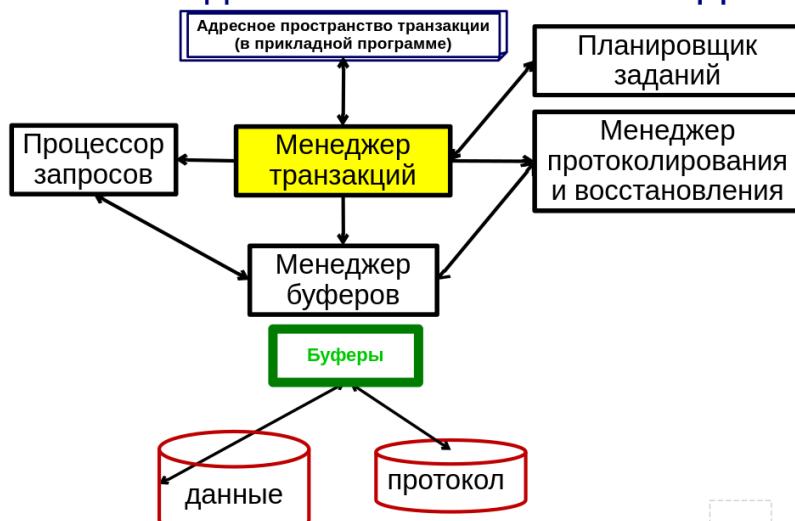
Ответ

- Менеджер буферов
- Компилятор запросов
- Менеджер протоколирования
- Менеджер транзакций

Пояснение

Преза 25, сл 8

### Взаимодействие компонент СУБД



## Вопрос 10

Из ниже перечисленного выберите действия, выполняемые ОПЕРАТИВНЫМ администратором БД

Выберите один или несколько ответов:

- выбирают поставщика БД
- утверждают план действий в случае аварийной ситуации
- манипулируют дисковым пространством
- разрабатывают и внедряют структурные элементы БД
- выполняют восстановление данных
- наблюдают за текущей производительностью системы
- запускают процедуры резервного копирования данных

Ответ

- манипулируют дисковым пространством
- выполняют восстановление данных
- наблюдают за текущей производительностью системы
- запускают процедуры резервного копирования данных

Пояснение

Преза 31, сл 25

## Классификация АБД

*Оперативные (operational) АБД:*

- манипулируют дисковым пространством
- наблюдают за текущей производительностью системы
- реагируют на возникающие неисправности БД
- обновляют системное ПО и ПО базы данных
- контролируют структурные изменения БД
- запускают процедуры резервного копирования данных
- выполняют восстановление данных
- создают и управляют тестовыми конфигурациями БД

## Вопрос 12

При создании хранилищ используются следующие модели данных:  
Выберите один или несколько ответов:

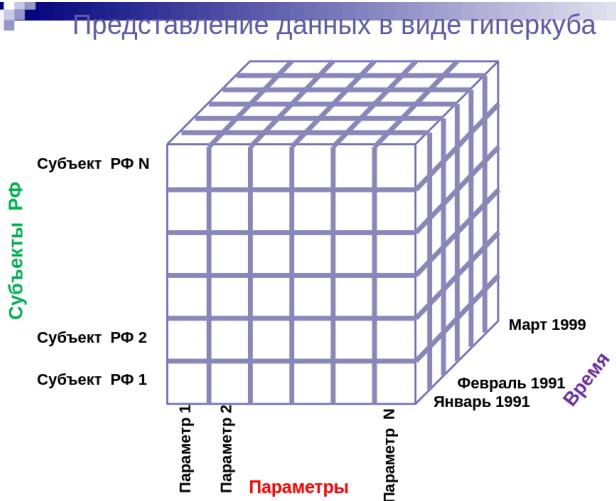
- Модель данных "Гиперкуб"
- Модель данных "Снежинка"
- Модель данных "Звезда"
- Модель данных "Гипермассив"
- Модель данных "Многомерное дерево"

Ответ

- Модель данных "Гиперкуб"
- Модель данных "Снежинка"
- Модель данных "Звезда"

Пояснение

Преза 30, сл 20-21 гиперкуб



### Многомерная модель данных

- Данные хранятся не в виде плоских таблиц, а виде упорядоченных многомерных массивов
- Многомерной хранение реализуется физически
- Среднее время ответа в 10-100 раз быстрее, чем в реляционной СУБД
- Основные понятия – измерение и значение
- Измерение – множество образующее одну из граней гиперкуба. Индексы для идентификации конкретных значений
- Операции: сечение, вращение, детализация, свертка

21

Преза 30, сл 23 звезда, Преза 30, сл 24 снежинка

### Пример: ROLAP – «Звезда»

Справочная таблица

PARAMETR_ID
Название
Единицы измерения
Тип параметра
Источник данных
Комментарий

Факториальная таблица

PARAMETR_ID
REGION_ID
PERIOD_ID
Значение параметра

Справочная таблица

REGION_ID
Название субъекта РФ
Подчиненность
Описание

Справочная таблица

PERIOD_ID
Описание
Год
Квартал
Месяц
День

23

### Пример: ROLAP – «Снежинка»

Факториальная таблица

PARAMETR_ID
REGION_ID
PERIOD_ID
Значение параметра

Справочная таблица

PARAMTYPE_ID
Описание типа параметра
Год

Справочная таблица

DATASOURCE_ID
Описание источника данных

Справочная таблица

QUARTER_ID
Название квартала
Комментарий

Справочная таблица

MONTH_ID
Название месяца

Справочная таблица

REGION_ID
Название субъекта РФ
Подчиненность

PERIOD_ID
Описание
Год
Квартал

QUARTER_ID
Название квартала
Комментарий

MONTH_ID
Название месяца

24

### Вопрос 14

К параллельным базам данных предъявляются следующие требования:

Выберите один или несколько ответов:

- высокая адаптивность
- высокая мобильность
- высокая масштабируемость
- высокая доступность данных
- высокая производительность
- высокая параллельность

Ответ

- высокая масштабируемость
- высокая доступность данных
- высокая производительность

Пояснение

Преза 29, сл 3

### Требования к параллельной системе баз данных

Параллельная СУБД должна быть  
*системой высокой готовности:*

- Способной хранить большой объем данных
- Обеспечивать эффективную обработку параллельных транзакций в режиме “**24 часа в сутки, 7 дней в неделю**”.

**Основные требования**

- **высокая масштабируемость;**
- **высокая производительность;**
- **высокая доступность данных.**

### Вопрос 26

Пусть даны отношения

R (A, B, C)   S (C, D)  
1   5   3        2   1

2	7	7	
4	2	2	
			3 7
			5 6

Выберите отношение T, являющееся натуральным соединением R и S

Выберите один ответ:

T (A, B, C, D)

1 5 3 7

4 2 2 1

2 7 5 6

T (A, B, C, D)

1 5 3 7

2 7 5 6

T (A, B, C, D)

4 2 2 1

2 7 5 6

T (A, B, C, D)

1 5 3 7

4 2 2 1

T (A, B, C, D)

1 5 3 7

2 7 3 7

4 2 2 1



Ответ



T (A, B, C, D)

1 5 3 7



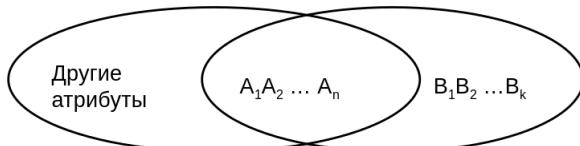
4 2 2 1

Пояснение

Преза 5 слайд 21

## Приведение к нормальным формам

- Поиск в исходном отношении R нетривиальных функциональных зависимостей  $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_k$ , нарушающих нормальную форму.
- Разбиение отношения R на 2 отношения, например, T и S, таких, что



- T будет состоять из атрибутов, образующих нетривиальную ФЗ,
- S – из атрибутов отношения R, входящих в ключ и не вошедших в отношение T.
- Исходное отношение должно восстанавливаться из 2-ух получившихся отношений с помощью **НАТУРАЛЬНОГО СОЕДИНЕНИЯ**

Декомпозиция		
$R(A, B, C)$		
1 2 3		
3 2 6		
5 4 2		
$R_1(A, B)$	$R_2(B, C)$	
1 2	2 3	
3 2	2 6	
5 4	4 2	
Соединение		
$R'(A, B, C)$		
1 2 3		
I 2 6		
3 2 3		
3 2 6		
5 4 2		

## Вопрос 34

Какое расписание 2-х фазных транзакций сериализуемо?

Выберите один ответ:

- Никакое
- Без циклов
- Линейно упорядоченное
- Любое
- Частично упорядоченное

Ответ

- Любое

Пояснение

Преза 12 , сл 41

## Двухфазные транзакции

- Транзакция называется **двуухфазной**, если все операции блокировки предшествуют всем операциям снятия блокировки

$$\begin{array}{lll} \left. \begin{array}{l} \text{LOCK } A \\ \text{LOCK } B \end{array} \right\} f_1 & \left. \begin{array}{l} \text{LOCK } B \\ \text{LOCK } C \\ \text{UNLOCK } B \end{array} \right\} f_3 & \left. \begin{array}{l} \text{LOCK } A \\ \text{LOCK } C \\ \text{UNLOCK } C \\ \text{UNLOCK } A \end{array} \right\} f_7 \\ \left. \begin{array}{l} \text{UNLOCK } A \\ \text{UNLOCK } B \end{array} \right\} f_2 & \left. \begin{array}{l} \text{LOCK } A \\ \text{UNLOCK } C \\ \text{UNLOCK } A \end{array} \right\} f_5 & \left. \begin{array}{l} \text{UNLOCK } A \end{array} \right\} f_6 \\ T_1 & T_2 & T_3 \end{array}$$

- **Теорема 2.** Любое расписание двухфазных транзакций – сериализуемо.