

ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

БАЗА ДАННЫХ (БД) – хранилище интегрированных и коллективно используемых данных, организованное с целью:

- обеспечить независимость структур хранимых данных от обрабатывающих программ,
- оптимизировать использование памяти и время доступа.

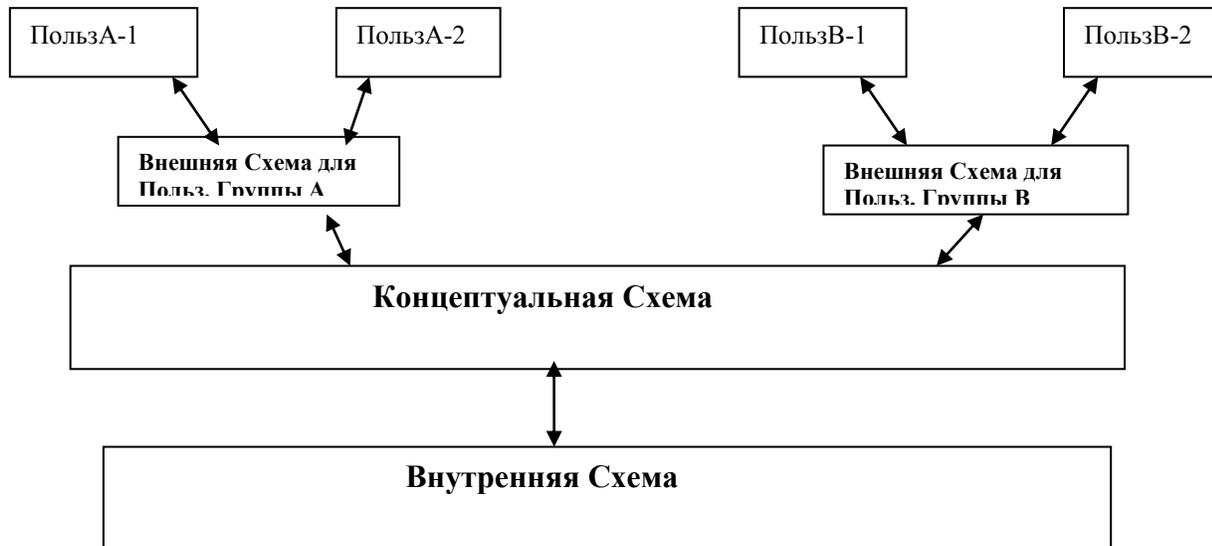
Пользователи БД: прикладные программы, непрограммирующие пользователи, прикладные программисты, администраторы БД.

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД) – программный комплекс, обеспечивающий создание, модификацию и использование БД.

СХЕМА БД	< ----- >	ЭКЗЕМПЛЯР БД
ОБЪЕКТ	-	Экземпляр, Запись (студент, деталь)
имеет ИМЯ		
имеет АТРИБУТЫ	-	Поля (ФИО-студента, номер-детали)
ОТНОШЕНИЯ между Объектами		
КЛЮЧ – атрибут (или группа атрибутов), однозначно идентифицирующая ОБЪЕКТ.		

Уровни представления данных в БД:

- Внешний** – связан с тем, как пользователи представляют себе эти данные.
- Внутренний** – связан со способом фактического хранения данных.
- Концептуальный** – промежуточное, обобщенное представление.



ФУНКЦИИ СУБД

- ОБРАБОТКА ЗАПРОСА
- КОНТРОЛЬ ПОЛНОМОЧИЙ (обеспечение БЕЗОПАСНОСТИ)
Обеспечение безопасности – защита данных в БД от несанкционированного раскрытия, изменения или разрушения.
- ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ
Обеспечение целостности – защита данных от неверных (vs. незаконных) изменений и обеспечение правильности данных в БД.
причины нарушения целостности:
сбои оборудования;
ошибки человека (пользователя, обслуживающего персонала);
программные ошибки в СУБД, ОС;
программные ошибки в прикладных программах.

Языки баз данных:

- ЯЗЫКИ ОПИСАНИЯ ДАННЫХ (ЯОД) - для описания схемы БД
- ЯЗЫКИ МАНИПУЛИРОВАНИЯ ДАННЫМИ (ЯМД) - для изменения состояния БД (включая, поиск и др. действия)
- ЯЗЫКИ ЗАПРОСОВ (ЯЗ) - для пользователя (может совпадать с ЯМД)

Транзакция – последовательность логически связанных действий, переводящая СУБД из одного состояния в другое.

ПРИМЕР (Основной пример)

ЗЧ (заказчик)

З	ФИО	АДР	ГОР
---	-----	-----	-----

ТР (товар)

Т	НАИМ	МАРКА	ЦЕНА	ГОР
---	------	-------	------	-----

ЗЗ (заказ)

$Z_i \leftrightarrow T_j$

N_{ij}

Основные операции:

ПОИСК (SEARCH): найти ЗЧ, заказавших Т2
 ВКЛЮЧЕНИЕ (INSERT): появился новый ЗЧ – З4
 УДАЛЕНИЕ (DELETE): З3 отказался от заказа на Т2
 ОБНОВЛЕНИЕ (UPDATE): изменился адрес З1

МОДЕЛИ ДАННЫХ (на концептуальном уровне):

- реляционная модель,
- иерархическая модель,
- сетевая модель.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ – модель данных, представленная набором отношений.

Реляционная база данных для Основного примера:

ЗЧ

З	ФИО	АДР	ГОР
З1	Иванов	<a1>	М
З2	Петров	<a2>	М
З3	Серов	<a3>	В

ТР

Т	НАИМ	МАРКА	ЦЕНА	ГОР
Т1	Хол	Зил	2000	М
Т2	Хол	Мор	1500	В
Т3	Тв	Рек	2500	В
Т4	Тв	Руб	3500	М

ЗЗ

З	Т	К
З1	Т1	1
З1	Т2	2
З1	Т3	4
З2	Т1	3
З2	Т2	4
З3	Т2	3

РЕЛЯЦИОННАЯ СХЕМА (для Основного примера):

DOMAIN З CHAR (2)
 DOMAIN ФИО CHAR (20)
 DOMAIN АДР CHAR (50)
 DOMAIN ГОР CHAR (3)
 DOMAIN Т CHAR (2)
 DOMAIN НАИМ CHAR (3)
 DOMAIN МАРКА CHAR (3)
 DOMAIN ЦЕНА INT
 DOMAIN К INT
 REL ЗЧ (З, ФИО, АДР, ГОР) KEY (З)
 REL ТР (Т, НАИМ, МАРКА, ЦЕНА, ГОР) KEY (Т)
 REL ЗЗ (З, Т, К) KEY (З, Т)

С помощью этого описания (на некотором формальном языке близком реальным ЯОД) задаются схема базы данных, структура таблиц, ключи.

Выполнение операций Основного примера:

ПОИСК (SEARCH):

найти ЗЧ, заказавших Т2

- 1.Обращаемся к таблице ЗЧ
- 2.Выбираем для рассмотрения атрибут (столбец) Т – *вертикальная выборка*
- 3.Отбираем строки, в которых Т = Т2
- 4.Фиксируем идентификаторы заказчиков (З) из этих строк:
 31 32 33
- 5.Обращаемся к таблице ЗЧ
- 6.Выбираем из нее строки, соответствующие найденным идентификаторам заказчиков, – *горизонтальная выборка*
- 7.Результатом поиска (то есть ответом на наш запрос) будет такая таблица

31	Иванов	<a1>	М
32	Петров	<a2>	М
33	Серов	<a3>	В

ВКЛЮЧЕНИЕ (INSERT):

появился новый ЗЧ – З4

- 1.Добавляем в таблицу ЗЧ новую строку
- 2.Заполняем значения полей (атрибутов):
 34 Козлов <a4> М
- 3.Получаем новую таблицу ЗЧ

ЗЧ

З	ФИО	АДР	ГОР
31	Иванов	<a1>	М
32	Петров	<a2>	М
33	Серов	<a3>	В
34	Козлов	<a4>	М

УДАЛЕНИЕ (DELETE):

ЗЗ отказался от заказа на Т2

- 1.Обращаемся к таблице ЗЗ
- 2.Находим строки, соответствующие заказчику ЗЗ, – *горизонтальная выборка*
- 3.Удаляем из таблицы строку, для которой Т = Т2
- 4.Получаем новую таблицу ЗЗ

ЗЗ

З	Т	К
31	Т1	1
31	Т2	2
31	Т3	4

32	T1	3
32	T2	4

ОБНОВЛЕНИЕ (UPDATE): изменился адрес 31

- 1.Обращаемся к таблице ЗЧ
- 2.Выбираем для рассмотрения строку с идентификатором заказчика 31 – *горизонтальная выборка*
- 3.Заменяем значение поля (свойства) АДР
- 4.Получаем новую таблицу ЗЧ

ЗЧ

З	ФИО	АДР	ГОР
31	Иванов	<a1>	М
32	Петров	<a2>	М
33	Серов	<a3>	В

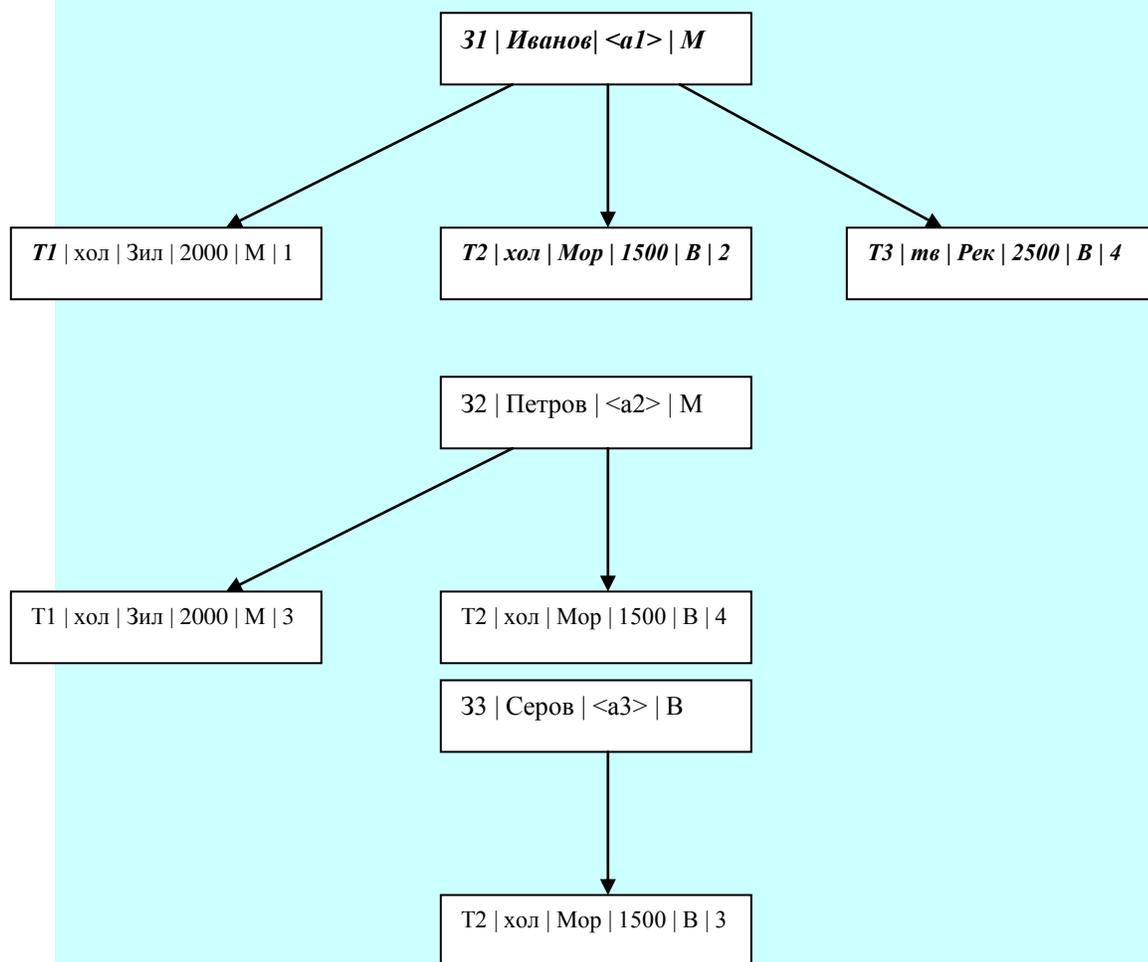
ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ – модель данных, представленная в виде дерева или группы деревьев (леса).

Иерархическая модель данных очень удобна в тех случаях, когда основным отношением между объектами является отношение иерархии.

Иерархическая база данных для Основного примера:

Случай 1.

В вершинах находятся описания объектов-заказчиков.



Выполнение операций Основного примера:

ПОИСК (SEARCH): найти ЗЧ, заказавших Т2

1. Просматриваем все корневые вершины (соответствующие заказчикам)
2. Для каждой такой вершины, если среди дочерних вершин есть вершина товара Т2, запоминаем ее
3. Результатом поиска (то есть ответом на наш запрос) будет совокупность таких вершин

ВКЛЮЧЕНИЕ (INSERT): появился новый ЗЧ – З4

1. Добавляем в качестве корневой вершины описание нового заказчика:

З4 Козлов <a4> М

2. Получаем новую изолированную вершину (заказов З4 пока не делал)
3. Результат – лес, содержащий эту новую вершину

УДАЛЕНИЕ (DELETE): З3 отказался от заказа на Т2

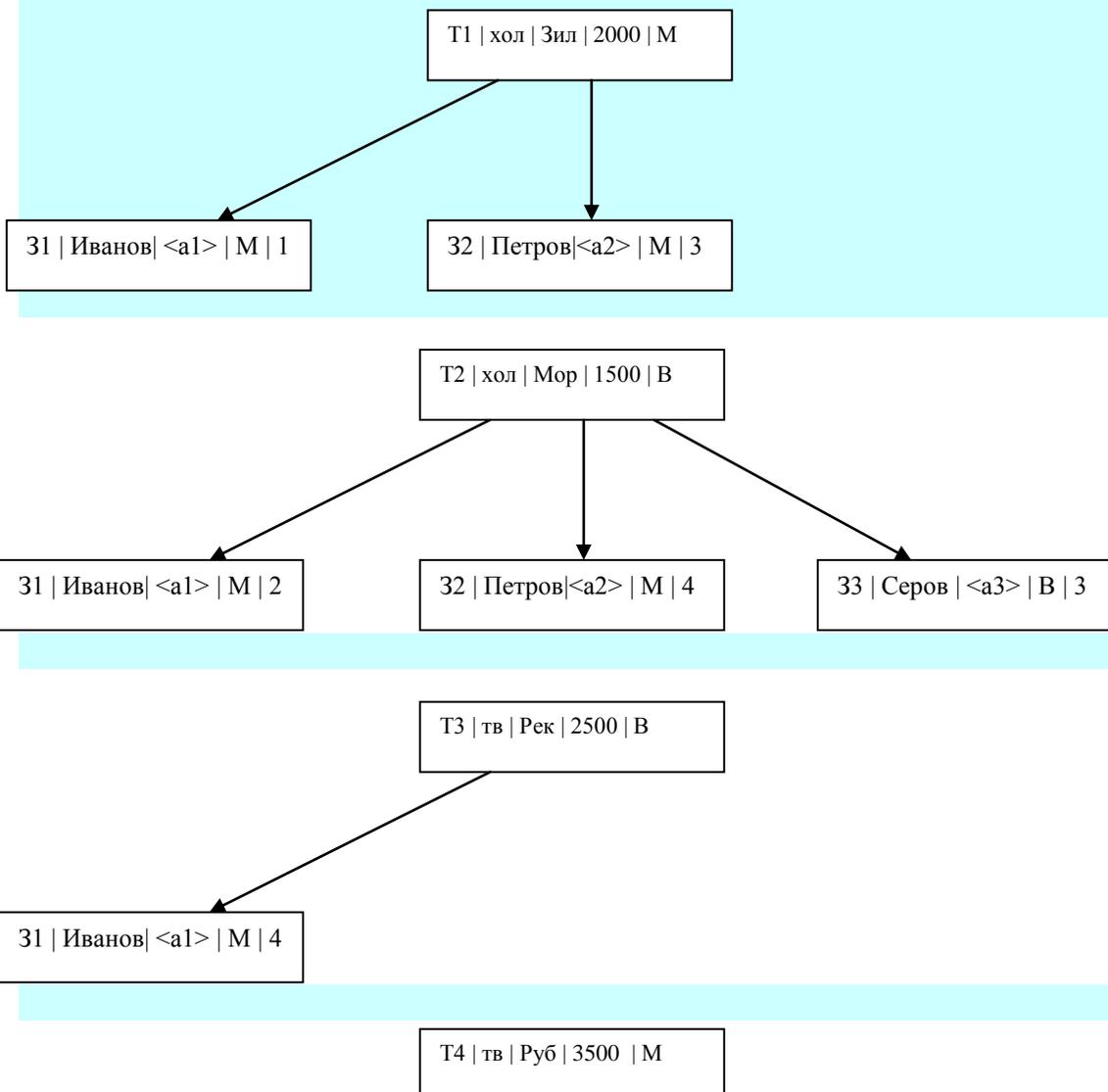
1. Находим корневую вершину, соответствующую заказчику З3
2. Удаляем дочернюю вершину, соответствующую товару Т2 (здесь она единственная)
3. Результат – измененный лес

ОБНОВЛЕНИЕ (UPDATE): изменился адрес З1

1. Находим корневую вершину, соответствующую заказчику З1
2. Заменяем значение поля (свойства) АДР
3. Результат – измененный лес

Случай 2.

В вершинах находятся описания объектов-товаров.



Выполнение операций Основного примера:

ПОИСК (SEARCH): найти ЗЧ, заказавших Т2

- 1.Находим корневую вершину, соответствующую товару Т2
- 2.Результатом поиска (то есть ответом на наш запрос) будет совокупность дочерних вершин

ВКЛЮЧЕНИЕ (INSERT): появился новый ЗЧ – 34

Ситуация отличается от ситуации для Случая 1, так как корневые вершины теперь соответствуют товарам. Решение здесь не очень естественное.

- 1.Добавляем в качестве некорневой вершины описание нового заказчика:

34 Козлов <a4> М

- 2.Подчиняем эту вершину одной из корневых вершин (любой)
- 3.Указываем, что количество заказов равно 0
- 4.Результат – измененный лес

УДАЛЕНИЕ (DELETE): 33 отказался от заказа на Т2

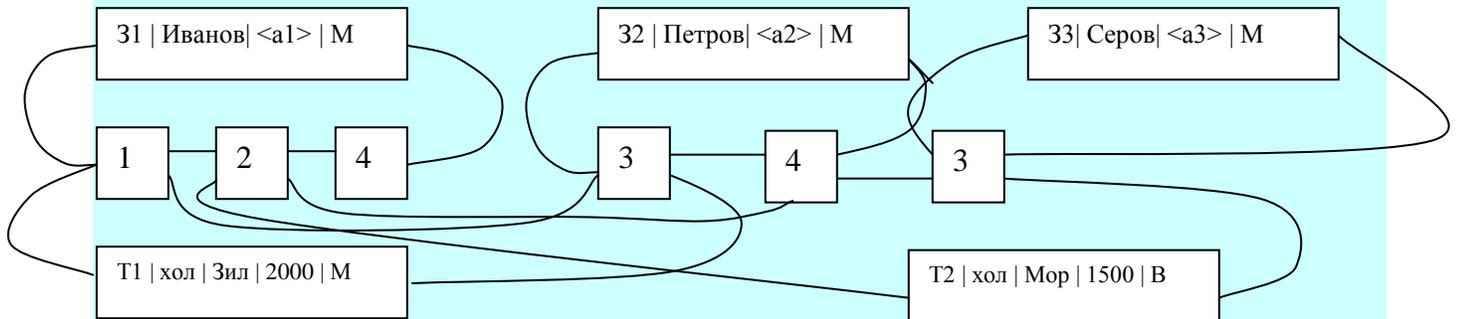
- 1.Находим корневую вершину, соответствующую товару Т2
- 2.Удаляем ее дочернюю вершину, соответствующую заказчику 33 (другое возможное решение – меняем количество заказанных товаров Т2 на 0)
- 3.Результат – измененный лес

ОБНОВЛЕНИЕ (UPDATE): изменился адрес 31

- 1.Просматриваем все корневые вершины (соответствующие товарам)
- 2.Для каждой такой вершины, если среди дочерних вершин есть вершина заказчика 31, то вносим в описание необходимые изменения (<a1> → <a11>)
- 3.Результат – измененный лес

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ – модель данных, представленная в виде произвольного графа.

Сетевая база данных для Основного примера (фрагмент):



Выполнение операций Основного примера:

ПОИСК (SEARCH): найти ЗЧ, заказавших Т2

- 1.Находим вершину, соответствующую Т2
- 2.Находим цепочку, ведущую от Т2 к одной из вершин, соответствующих количеству заказанных Т2
Это вершина, содержащая 2 («вершина-2»), в среднем ряду
- 3.Находим цепочку, ведущую от «вершины-2» к заказчику
Этим заказчиком будет 31
4. Включаем 31 в список ответа
- 5.Находим цепочку, ведущую от «вершины-2» к следующей вершине среднего ряда
Это будет пятая (слева) вершина среднего ряда, содержащая 4 («вершина-4»)
- 6.Находим цепочку, ведущую от «вершины-4» к заказчику
Этим заказчиком будет 32
7. Включаем 32 в список ответа
- 8.Находим цепочку, ведущую от «вершины-4» к следующей вершине среднего ряда
Это будет шестая (слева) вершина среднего ряда, содержащая 3 («вершина-3»)

9.Находим цепочку, ведущую от «вершины-3» к заказчику

Этим заказчиком будет 33

10. Включаем 33 в список ответа

Результат (окончательный ответ): Заказчик-31, Заказчик-32 и Заказчик-33

ВКЛЮЧЕНИЕ (INSERT): появился новый ЗЧ – 34

1.Добавляем в качестве новой вершины описание нового заказчика:

34 Козлов <a4> М

2.Получаем новую изолированную вершину (заказов 34 пока не делал)

3.Результат – несвязная сеть, содержащая эту новую вершину

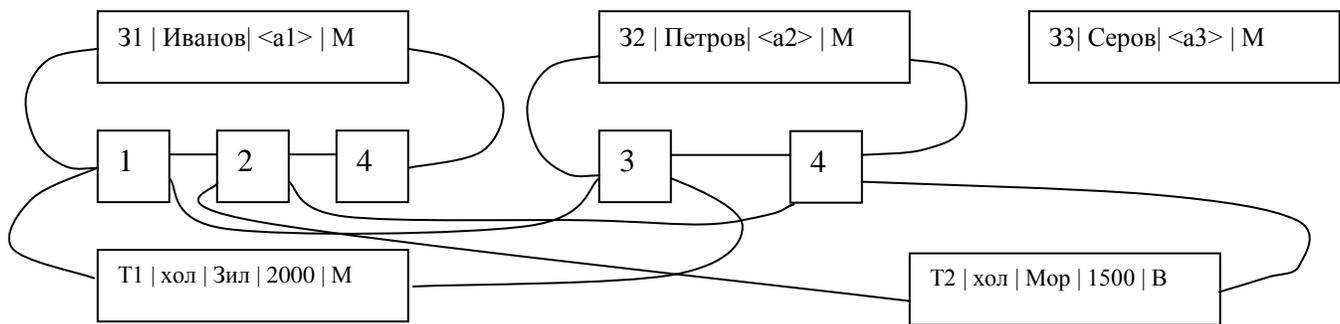
УДАЛЕНИЕ (DELETE): 33 отказался от заказа на Т2

1.Находим вершину, соответствующую заказчику 33

2.Проверяем, есть ли среди его заказов заказ на товар Т2 (в нашем случае есть – «вершина-3»)

3.Удаляем «вершину-3» и вносим необходимые изменения в сеть

4.Результат – измененная сеть



ОБНОВЛЕНИЕ (UPDATE): изменился адрес 31

1.Находим вершину, соответствующую заказчику 31

2.Заменяем значение поля (свойства) АДР

3.Результат – измененная сеть: