

Проектирование баз данных, ч.1

ФИО преподавателя: Смирнов Михаил Вячеславович

e-mail: smirnovmgupi@gmail.com

Лекция 5

Запросы Type I и Type II. Пользовательские представления

Вложенные запросы, Type I, Type II Теория.

Вложенными запросами называются комбинированные операторы выбора, вставки/удаления/добавления данных, которые содержат в себе несколько подзапросов (внешние и внутренние – outer и inner).

Являются довольно изящным и простым решением при оптимизации работы БД. Чаще всего используются при вставке (INSERT), удалении (DELETE) или же изменении (UPDATE) большого количества значений в одной таблице на основании данных другой таблицы.

Также, активно применяются при конструировании сложных операторов выборки из двух таблиц в тех случаях, когда оператор JOIN не подходит для решения задачи.

Вложенный запрос первого типа (Type I)

Особенности вложенного запроса Type I:

- в случае комбинации двух запросов оператора SELECT используется конструкция IN
- запускается на исполнение только один раз (не допускает циклической обработки)
- при исполнении первым всегда выполняется внутренний запрос (inner)

Вложенный запрос первого типа (Type I), комбинация SELECT + SELECT

Используется скрипт `typ1_selsel`.

Таблица `supplier` (поставщик)

suppno	suppname	suppemail	suppphone	suppurl	suppdiscout
34824	ColorMag	custrel@colormag.com	(456)555-5678	www.colormag.com	0,1
34983	Connex	help@connex.com	(345)666-7788	www.connex.com	0,12
56843	Ethite	help@ethite.com	(872)456-8967	www.ethite.com	0,05
98356	Intersafe	orderdesk@intersafe.com	(847)555-5738	www.intersafe.com	0,1
98347	UV Components	custserv@vpcomponents.com	(667)556-3344	www.vpcomponents.com	0,08
46871	Cybercx	orderhelp@cybercx.com	(345)556-3399	www.cybercx.com	0

Таблица `product` (продукт)

prodno	prodname	prodsup	suppno	ProdQual	ProdPrice
334	17 inch Color Monitor	ColorMag	34824	12	160,00
335	19 inch Color Monitor	ColorMag	34824	10	200,00
6789	R3000 Color Laser Printer	Connex	34983	5	750,00
478	10 foot Printer Cable	Elhite	56843	100	10,00
757	Ink Jet Color Printer	Connex	34983	8	100,00
678	36-bit Color Scanner	UV Components	98347	16	200,00
4838	Printer Cable Protector	Intersafe	98356	32	15,00
387	Battery Back-Up System	Cybercx	46871	12	90,00

Вложенный запрос первого типа (Type I), комбинация SELECT + SELECT продолжение

Необходимо показать, какие продукты поставляет в наш магазин, поставщик, название которого начинается с Color.

	prodname
1	17 inch Color Monitor
2	19 inch Color Monitor

```
SELECT prodname FROM product
WHERE suppno IN
  (SELECT suppno FROM supplier WHERE suppname like 'Color%');
```

Обязательная команда IN, связывающая внешний и внутренний запрос.

Внешний запрос (outer)

Внутренний запрос (inner)

Вложенный запрос второго типа (Type II)

Особенности вложенного запроса Type II:

- конструкция IN не используется, как правило, чаще всего используется конструкция EXISTS (включает) или NOT EXISTS
- при исполнении первым всегда выполняется последний запрос в “дереве”

Вложенный запрос второго типа (Type II), комбинация INSERT INTO + SELECT

Используется скрипт `typ2_inssel`.

Таблица А

	status	pair
1	1	first
2	2	second
3	3	third



Таблица В

	status	pair
1	1	first
2	2	second

Из таблицы А в таблицу В необходимо перенести (скопировать) данные при условии `status < 3`.

```
INSERT INTO B (status, pair)
  SELECT status, pair FROM A WHERE status < 3;
```

Второй запрос в дереве, выполняется первым

Первый запрос в дереве, выполняется вторым

Представления (VIEW). Теория. Зачем они нужны?

Таблица является **физическим** носителем информации

Вся информация, занесенная в таблицу хранится на **жесткотелем** накопителе

Информация на накопителе может добавляться и извлекаться по двум принципам:

СТЭК (stack)

- добавляем данные снизу
- читаем данные снизу

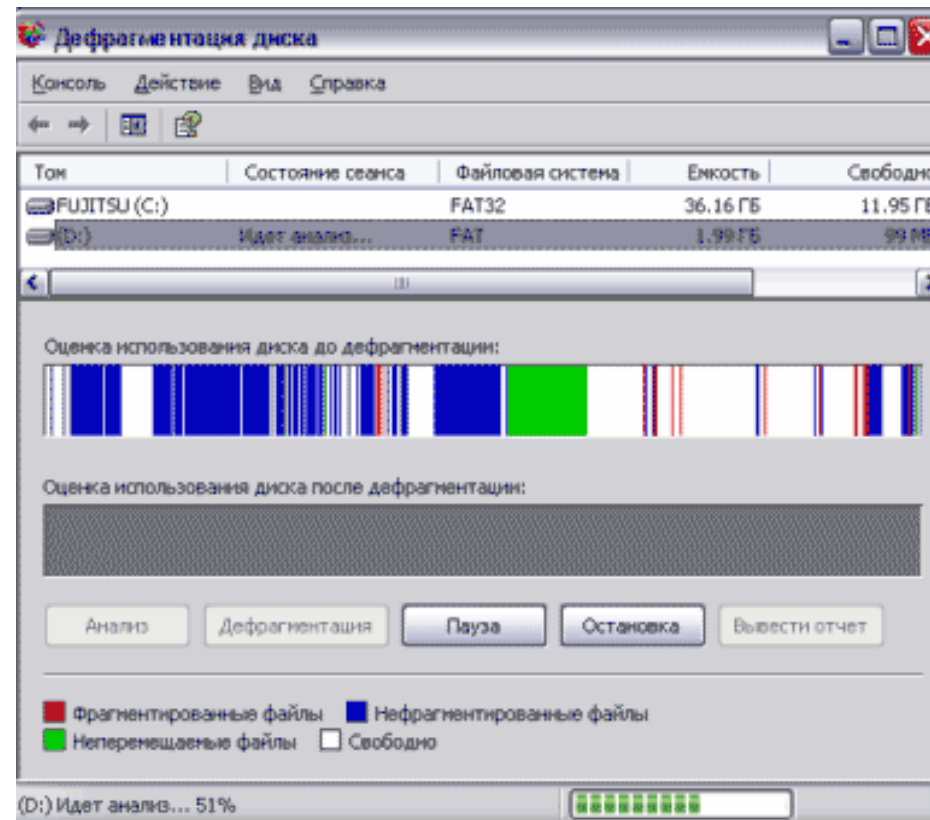
ОЧЕРЕДЬ (queue)

- добавляем данные снизу
- читаем данные сверху

Ключевой вывод: записанная в таблицы информация статична и малоподвижна.

Представления (VIEW). Теория. Зачем они нужны?

Вот как раз сочная, известная многим из нас картинка, подтверждающая предыдущие тезисы. Дефрагментация диска.



Представления (VIEW). Теория. Зачем они нужны?

Как же тогда работают запросы типа SELECT? Команды ORDER BY, NOT EXIST..

ВИРТУАЛЬНО.. В операционном режиме.

Нам действительно нужно каждый раз выполнять выборку или группировку для того чтобы видеть данные в удобном виде. А для тех, кто не знает SQL или знает настолько хорошо, что может упростить всем работу с БД придумали представления.

Итак, **представление** – это виртуальная (логическая) таблица, представляющая собой поименованный запрос, который будет подставлен как подзапрос при использовании представления.

В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью набора данных, хранящегося в базе.

Содержимое представления динамически вычисляется на основании данных, находящихся в реальных таблицах. Изменение данных в реальной таблице базы данных немедленно отражается в содержимом всех представлений, построенных на основании этой таблицы. *

* **Материал из Википедии - свободной энциклопедии**

Представления (VIEW). Теория. Зачем они нужны?

Что **могут делать** представления?

1. Представления скрывают от прикладной программы сложность запросов и саму структуру таблиц БД.
2. Использование представлений позволяет отделить прикладную схему представления данных от схемы хранения. Проще говоря, изменение программы не затрагивает физические таблицы, а изменение физической структуры таблиц не требует корректировки программы.
3. С помощью представлений обеспечивается ещё один уровень защиты данных. Пользователю могут предоставляться права только на представление, благодаря чему он не будет иметь доступа к данным, находящимся в тех же таблицах, но не предназначенных для него.
4. Поскольку SQL-запрос, выбирающий данные представления, зафиксирован на момент его создания, СУБД получает возможность применить к этому запросу оптимизацию или предварительную компиляцию

Представления (VIEW). Теория. Зачем они нужны?

Что **могут хранить** представления?

1. Подмножество записей из таблицы БД, отвечающее определенным условиям.
2. Подмножество столбцов таблицы БД, требуемое программой.
3. Результат обработки данных таблицы определенными операциями.
4. Результат объединения (join) нескольких таблиц.
5. Результат слияния нескольких таблиц с одинаковыми именами и типами полей.
6. Результат группировки записей в таблице
7. Практически любую комбинацию вышеперечисленных возможностей.

Чтение на дом

- Русский Кренке, стр. 305-324.
- Английский Кренке, 361-371.
- MIT, стр. 86-91.

Спасибо за внимание!