

Лабораторная работа №5

Тема: Построение нетривиальных запросов на SQL(завершение).

1. Операция объединения UNION
2. Запросы модификации данных
 - a. Добавление данных
 - b. Удаление данных
 - c. Изменение данных

1. Операция объединения

Операция объединения UNION – это еще один способ комбинирования нескольких запросов, когда запросы, участвующие в объединении, равноправны. Участвовать в объединении могут только запросы, результирующие таблицы которых имеют одинаковую структуру (свойство совместимости по объединению). Результирующая таблица содержит только уникальные строки. Если требуется сохранить дублирующие строки, используют UNION ALL. Для упорядочения строк добавляют фразу ORDER BY.

Пример 1.1. Найти поставщиков, статус которых 10 или в поставках которых количество деталей превышает 300.

```
SELECT s.sname FROM s WHERE s.status='10';
UNION;
SELECT s.sname FROM s,sp WHERE s.ids=sp.ids;
AND Sp.qty>300
```

Sname
Clark
Jones
Smith

```
SELECT s.sname FROM s WHERE s.status='10';
UNION ALL;
SELECT s.sname FROM s,sp WHERE s.ids=sp.ids;
AND Sp.qty>300 ORDER BY sname
```

Sname
Clark
Jones
Jones
Smith

2. Запросы модификации данных

Такие запросы предназначены для изменения содержания существующей базы данных и выполняются с помощью команд:

- **INSERT INTO** – запрос добавления ;
- **DELETE** – запрос удаления ;
- **UPDATE** – запрос обновления.

2а. Добавление данных

Для добавления данных в таблицу применяют оператор INSERT INTO в нескольких вариантах.

Строка добавляется своими значениями из фразы VALUES, значения не null. В этом случае каждое значение соответствует номеру атрибута в схеме отношения:

```
INSERT INTO <имя_таблицы> VALUES (значение1, ..., значениеn)
```

Если в некоторых атрибутах требуется установить значения null, используют вариант оператора, включающий помимо списка значений, список атрибутов отношения:

```
INSERT INTO <имя_таблицы> (имя_столбца1, ... имя_столбцаn)
VALUES (значение1, ... значениеn)
```

Вместо указания конкретного значения можно использовать запрос, как в следующих вариантах оператора:

```
INSERT INTO <имя_таблицы> SELECT...FROM... WHERE
```

или

```
INSERT INTO <имя_таблицы> (имя_столбца1,... имя_столбцаn)
SELECT...FROM... WHERE
```

Пример 2а.1. Добавить 5-ого поставщика с именем Ivan статусом 30 из города Moscow:

```
INSERT INTO s VALUES(5,'Ivan','30','Moscow')
```

Ids	Sname	Status	City
1	Smith	20	London
2	Jones	10	Paris
3	Blake	30	Paris
4	Clark	20	London
5	Ivan	30	Moscow

Пример 2а.2. Добавить 13-ую поставку детали с номером 6 5-ым поставщиком в количестве 250, 30 ноября 2011 года:

```
INSERT INTO sp (idsp, ids, idp, qty, end);
VALUES (13,5,6,250,{^2011/11/30})
```

Idsp	Ids	Idp	Qty	End
2	1	2	200	10/10/11
3	1	3	400	10/15/11
4	1	4	200	/ /
5	1	5	100	/ /
6	1	6	100	/ /
7	2	1	300	10/05/11
8	2	2	400	10/05/11
9	3	2	200	09/05/11
10	4	2	200	/ /
11	4	4	300	/ /
12	4	5	400	08/30/11
13	5	6	250	11/30/11

Пример 2а.3. Добавить 14-ую поставку 5-ым поставщиком детали с номером 1 в количестве 250, при условии, что дата поставки неизвестна:

```
INSERT INTO sp (idsp,ids,idp,qty);
VALUES (14,5,1,250)
```

Idsp	Ids	Idp	Qty	End
3	1	3	400	10/15/11
4	1	4	200	/ /
5	1	5	100	/ /
6	1	6	100	/ /
7	2	1	300	10/05/11
8	2	2	400	10/05/11
9	3	2	200	09/05/11
10	4	2	200	/ /
11	4	4	300	/ /
12	4	5	400	08/30/11
13	5	6	250	11/30/11
14	5	1	250	/ /

Пример 2а.4. Добавить 17-ую поставку 3-его поставщика детали под номером и в количестве, как в поставке 30 августа 2011 года, при условии, что дата 17-ой поставки неизвестна:

```
INSERT INTO sp (idsp,ids,idp,qty);
SELECT 17,3,sp.idp,sp.qty FROM sp Where sp.end={^2011/08/30})
```

Idsp	Ids	Idp	Qty	End
6	1	6	100	/ /
7	2	1	300	10/05/11
8	2	2	400	10/05/11
9	3	2	200	09/05/11
10	4	2	200	/ /
11	4	4	300	/ /
12	4	5	400	08/30/11
13	5	6	250	11/30/11
14	5	1	250	/ /
15	5	5	300	/ /
16	5	4	350	/ /
17	3	5	400	/ /

2б. Удаление данных

Удаление данных выполняют командой DELETE, которая существует в 2-х вариантах.

Очистка таблицы целиком:

```
DELETE FROM <имя_таблицы>
```

Удаление строк, удовлетворяющих условию:

```
DELETE FROM <имя_таблицы> WHERE <условие_отбора>
```

Пример 2б.1. Удалить поставки с 15 по 17-ую

```
DELETE FROM sp WHERE idsp>=15 AND idsp<=17
```

Пример 2б.2. Удалить поставщика по имени 'Ivan'

```
DELETE FROM s WHERE sname='Ivan'
```

2в. Обновление данных

Обновление данных выполняют командой UPDATE, которая также существует в 2-х вариантах.

Обновление атрибутов во всей таблице:

```
UPDATE <имя_таблицы> SET <атр1>=<знач1>, <атр2>=<знач2>, ...<атрn>=<значn>
```

Обновление атрибутов в таблице, удовлетворяющих условию:

```
UPDATE <имя_таблицы> SET <атр1>=<знач1>, <атр2>=<знач2>, ...<атрn>=<значn>  
WHERE <условие_отбора>
```

Пример 2в.1. Установить статус '20' для поставщика по имени Blake

```
UPDATE s SET status='20' WHERE sname='Blake'
```

Пример 2в.2. Увеличить на 10 вес деталей из Рима

```
UPDATE p SET weight=weight+10 WHERE p.city='Rome'
```

Пример 2в.3. Увеличить на 5 статус поставщиков, у которых имеются поставки в размере, превышающем средний

```
UPDATE s SET s.status=STR(VAL(s.status)+5);  
WHERE s.ids In (SELECT sp.ids FROM sp;  
WHERE sp.qty>(SELECT Avg(qty) FROM sp))
```

Ids	Sname	Status	City
1	Smith	25	London
2	Jones	15	Paris
3	Blake	30	Paris
0	Clark	20	London