

Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра Компьютерных Систем и Программных Технологий

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Дисциплина: **Базы данных**

Тема: **Язык SQL-DML**

Выполнил студент гр. № 43501/1

А.М. Кириллов

Преподаватель

А.В. Мяснов

Санкт-Петербург
2016

1. Цели работы

Познакомить студентов с языком создания запросов управления данными SQL-DML.

2. Программа работы

1. Изучите SQL-DML

2. Выполните все запросы из списка стандартных запросов.

Продемонстрируйте результаты преподавателю.

3. Получите у преподавателя и реализуйте SQL-запросы в соответствии с индивидуальным заданием. Продемонстрируйте результаты

4. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложите скрипт в Subversion.

3. Ход работы

3.1 выборка всех данных из каждой таблицы

```
create view v1 as select * from CLIENT;
create view v2 as select * from HOTEL;
create view v3 as select * from KURORT;
create view v4 as select * from PLACE_OF_INTEREST;
create view v5 as select * from TOUR;
create view v6 as select * from TRIP_PLACE;
create view v7 as select * from GROUP_TOUR;
create view v8 as select * from INSURANCE;
create view v9 as select * from KURORT_PLACE;
create view v10 as select * from PRICE;
create view v11 as select * from TRIP;
create view v12 as select * from TYPE;
```

```
SQL> SELECT * FROM v12;
```

ID	TYPENAME
1	Castle
2	Museum
3	Park
4	Palace
5	Tower

3.2 Выборка данных из одной таблицы при нескольких условиях

Выведем из таблицы "Hotel" положение отелей и количество звезд от 4 до 5.

```
SQL> SELECT * FROM HOTEL;
```

ID	LOCATION_HOTEL	STARS	LANGUAGE
1	center	4	English
2	near the sea	3	English
3	near the sea	5	English
4	center	4	English
5	near the village	4	English
6	near the village	2	Franch
7	center	4	Franch
8	center	5	English

```
create view v20 as select location_hotel,stars from HOTEL
WHERE location_hotel in ('center','near the village')
AND stars BETWEEN 4 AND 5;
```

```
SQL> SELECT * FROM v20;
```

LOCATION_HOTEL	STARS
center	4
center	4
center	4
center	5

3.3 Создание вычисляемого поля

```
create view v21 as select id, insurance_price,(insurance_price/70) as Dollar from INSURANCE;
```

Результат:

```
SQL> SELECT * FROM v21;
```

ID	INSURANCE_PRICE	DOLLAR
1	500000	7142
2	100000	1428

3.4 Выборка сортировкой

Полученные результаты: Отели отсортированы по количеству звезд

```
create view v22 as select * from hotel order by stars desc;
```

```
SQL> SELECT * FROM v22;
```

ID	LOCATION_HOTEL	STARS	LANGUAGE
8	center	5	English
3	near the sea	5	English
1	center	4	English
4	center	4	English
5	near the village	4	English
7	center	4	Franch
2	near the sea	3	English
6	near the village	2	Franch

3.5 Запрос, вычисляющий несколько совокупных характеристик таблиц

```
create view v23 as select max(adult) as "Max_price_for_adult" , min(child) as "Min_price_for_child" from price;
```

Полученные результаты:

```
SQL> select * from v23;
```

Max_price_for_adult	Min_price_for_child
60000	17000

3.6 Выборка данных из связанных таблиц (не менее двух примеров)

```
create view v24 as select client.pasport_id, client.name, kurort.location_kurort from client, trip, tour, kurort where client.pasport_id=trip.id_client and trip.id_tour=tour.id and tour.id_kurort=kurort.kurort_id;
```

Была осуществлена выборка клиента и страны, в которую он едет:

```
SQL> select * from v24;
```

PASPORT_ID	NAME	LOCATION_KURORT
1	Ivan	Sidnei
6	Sasha	Sidnei
11	Katya	Sidnei
2	Sasha	Kair
3	Andrei	Kair
7	Lena	Kair
12	Sasha	Kair
8	Ksenia	Paris
13	Ivan	Paris
4	Ulad	Pekin
9	Nina	Pekin
14	Sasha	Pekin
5	Seva	Rome
15	Ulada	Rome
16	Olya	Rome
17	Seva	Rome
18	Sasha	Rome

```
create view v25 as select client.pasport_id, client.name, price.adult
from client, trip, tour, price where client.pasport_id=trip.id_client
and trip.id_tour=tour.id and tour.id_price=price.id;
```

Выборка клиента и цены тура:

PASPORT_ID	NAME	ADULT
1	Ivan	50000
6	Sasha	50000
11	Katya	50000
5	Seva	40000
15	Ulada	40000
16	Olya	40000
17	Seva	40000
18	Sasha	40000
842345	Uladislava	40000
2	Sasha	60000
3	Andrei	60000
7	Lena	60000
12	Sasha	60000
8	Ksenia	60000
13	Ivan	60000
4	Ulad	35000
9	Nina	35000
14	Sasha	35000

3.7 Запрос, рассчитывающий совокупную характеристику с использованием группировки.

```
create view v26 as select name as "Name" , sum(number_phone) as "Phone"
from client group by name having sum(number_phone)>7700000;
```

Клиенты, имеющие номер телефона больше задаваемого значения(7700000):

```
SQL> select * from v26;
```

Name	Phone
Ivan	15454749
Lena	7727312
Sasha	32089793
Seva	14246197

3.8

```
create view v885 as select * from trip_place where id_trip
in(select type_id from place_of_interest);
```

ID	ID_TRIP	ID_PLACE_OF_INTEREST
1	2	1
2	3	5
3	4	2
4	1	3
10	5	8

3.9

```
create procedure insert_client (client_ID integer, name varchar(10), number_phone
integer)

as

begin

    insert into client(client_ID, name, number_phone) values
(:client_ID, :name, :number_phone);

end;

create procedure insert_trip (id_client integer, id_tour integer, id_insurance
integer, id integer)

as

begin

    insert into trip(id_client, id_insurance, id_tour, id) values (:id_client,
:id_insurance, :id_tour, :id);

end;

create procedure insert_tour(tour_id integer, id_kurort integer, id_hotel integer, price
integer)

as

begin

    insert into tour(tour_id, id_kurort, id_hotel, price) values (:tour_id, :id_kurort,
:id_hotel, :price);

end;

create procedure insert_hotel(hotel_ID integer, location varchar(15), stars integer)

as

begin

    insert into hotel(hotel_ID, location, stars) values (:hotel_ID, :location, :stars);

end;

create procedure insert_kurort(kurort_ID integer, country varchar(10), climate
varchar(10), weather varchar(10))

as

begin

    insert into kurort(kurort_ID, country, climate, weather) values(:kurort_ID,
:country, :climate, :weather);

end;
```

```

create procedure insert_kurort_place(ID integer, ID_kurort integer,
ID_place_of_interest integer)

as

begin

    insert into kurort_place(ID, id_kurort, id_place_of_interest) values(:ID,
:id_kurort, :id_place_of_interest);

end;

create procedure insert_place_of_interest(id_place_of_interest integer, id_type
integer, name_place varchar(15))

as

begin

    insert into place_of_interest(id_place_of_interest, id_type,
name_place) values(:id_place_of_interest, :id_type, :name_place);

end;

eate procedure insert_type_place(type_id integer, name_type varchar(15))

as

begin

    insert into type_place(type_id, name_type) values(:type_id, :name_type);

end;

```

3.10 Обновление записей

```

create procedure update_name(buy_t date) as

begin

    update client set name='Kristina' where client.client_id in (select

    client_id from contracts where contracts.buy_time<:buy_t);

end;

```

Была выбрана дата: 16.01.2014

Name	Type	Null	Value	De
BUY_T	DATE	<input type="checkbox"/>	16.01.2014	

Результат работы скрипта:

CLIENT_ID	NAME	NUMBER_PHONE
1	Dariya	7 412 582
2	Artem	7 125 852
3	Andrey	7 415 852
4	Nikita	7 412 585
5	Vlad	7 458 632
6	Ruslan	7 258 632
7	Anastasiya	7 455 896
8	Kristina	7 582 369
9	Katya	7 258 932
10	ElenUlyaa	7 458 962
11	Igor	7 412 586
12	Pavel	7 258 314
13	Alex	7 951 364
14	Sasha	7 562 344
15	Masha	7 856 314
16	Marina	7 215 634
17	Svatlana	7 852 146
18	Nataly	7 896 235

3.11С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

```
create procedure del_hotel_star_location(location_hotel varchar(15), star_hotel integer)
as begin
    delete from hotel where location = :location_hotel and stars =
:star_hotel; end;
```

Были заданы следующие данные:

Name	Type	Null	Value
LOCATION_HOTEL	VARCHAR(...)	<input type="checkbox"/>	near_the_sea
STAR_HOTEL	INTEGER	<input type="checkbox"/>	5

В результате была удалена запись из таблицы HOTEL в соответствии с заданными данными

1.12 С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

```
create procedure del_hotel as
begin
    delete from hotel where hotel_id not in (select id_tour from tour);
```

end;

До:

TOUR_ID	ID_KURORT	ID_HOTEL	PRICE
1	14	4	60 000
2	15	3	32 000
3	10	2	54 000
4	9	1	84 000
5	11	5	24 000
6	8	5	65 000
7	6	4	8 000
8	4	3	35 000
9	5	2	60 000
10	1	1	25 000
11	14	4	34 000

После выполнения скрипта:

TOUR_ID	ID_KURORT	ID_HOTEL	PRICE
1	14	4	60 000
2	15	3	32 000
3	10	2	54 000
4	9	1	84 000
5	11	5	24 000
6	8	5	65 000
7	6	4	8 000
8	4	3	35 000
9	5	2	60 000

Индивидуальное задание

1. Вывести 10 достопримечательностей, которые чаще всего посещают в поездках.
2. Вывести 5 наиболее интересных туров по критерию наилучшего отношения количества достопримечательностей курорта к стоимости тура.
3. Вывести 10 клиентов, которые чаще всего ездили куда-то за заданный период.

1.

```
create view top10place as select first 10 id_place_of_interest, typename,
nameplace, count(trip_place.id_trip) as countp
from trip, trip_place, place_of_interest, type
where(trip.id=trip_place.id_trip)and(trip_place.id_place_of_interest=
place_of_interest.id)and(place_of_interest.type_id=type.id)
group by id_place_of_interest, typename, nameplace order by countp desc;
```

Результат работы скрипта:

ID_PLACE_OF_INTEREST	TYPENAME	NAMEPLACE	COUNTP
1	Museum	Powerhouse Museum	1
2	Park	Al-Azhar Park	1
3	Palace	forbidden city	1
5	Tower	Eiffel Tower	1
6	Museum	National Museum	1
7	Park	Independent Park	1
8	Palace	Mysore Palace	1
9	Castle	Castle Knights	1
10	Tower	Leaning Tower	1
12	Tower	Montrond	1

2.

```
CREATE OR ALTER VIEW TOP5PLACE2(
    TOURID,
    KURORT_ID,
    id_price,
    PriceVouch,
    countp
)
AS
select first 5 tour.id as tourID, kurort.kurort_id as kurort_id, id_price,
adult as PriceVouch, (adult/count(place_of_interest.kurort_id)) as countp
from tour, place_of_interest, price, kurort
where
(price.id=tour.id_price)and(tour.id_kurort=kurort.kurort_id)and(kurort.kurort_id=place
_of_interest.kurort_id)
group by tourID, PriceVouch, id_price, kurort_id order by countp desc
```

TOURID	KURORT_ID	PRICEVOUCH	COUNTP	ID_PRICE
1	1	50 000	25 000	1
6	1	50 000	25 000	1
2	2	60 000	20 000	3
3	3	60 000	20 000	3
5	5	40 000	20 000	2

3.

```
create procedure top10client(start_watch date, end_watch date)

returns(
    Client_id integer, Tour_id integer, Count_Tour integer
)as
begin
    for select first 10 id_client, id_tour, count(id) as count_s from trip
    where (start_time>=:start_watch)and(end_time<=:end_watch)
    group by id_client, id_tour order by count_s desc into:Client_id, :Tour_id, :Count_Tour
    do
        begin
            /* Procedure Text */
            suspend;
        end
    end
end
```

Результат:

Parameters		SQL	
Name	Type	Null	Value
START_WATCH	DATE	<input type="checkbox"/>	16.12.2016
END_WATCH	DATE	<input type="checkbox"/>	07.01.2017

CLIENT_ID	TOUR_ID
1	1
8	3
9	4
11	1
14	5
18	5
19	5
20	5
21	5
22	5

Выводы:

В ходе лабораторной работы мы научились эффективно использовать конструкции языка sql-DML. Так же были изучены хранимые процедуры.