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**

А.М. Кириллов

А.В. Мяснов

Выполнил студент гр. № 43501/1

Преподаватель

Санкт-Петербург

2016

1. Цели работы

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

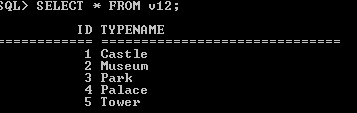
2. Программа работы

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

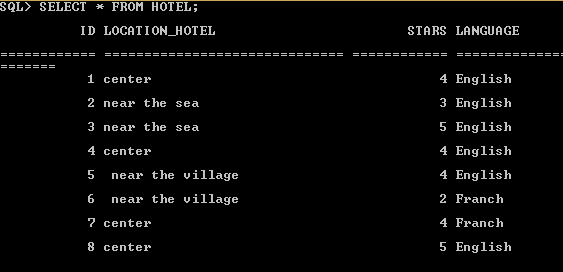
3.1 выборка всех данных из каждой таблицы

|  |
| --- |
| 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; |



3.2 Выборка данных из одной таблицы при нескольких условиях

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



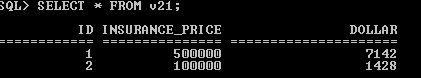
|  |
| --- |
| create view v20 as select location\_hotel,stars from HOTEL  WHERE location\_hotel in ('center','near the village')  AND stars BETWEEN 4 AND 5; |



3.3 Создание вычисляемого поля

|  |
| --- |
| create view v21 as select id, insurance\_price,(insurance\_price/70) as Dollar from INSURANCE; |

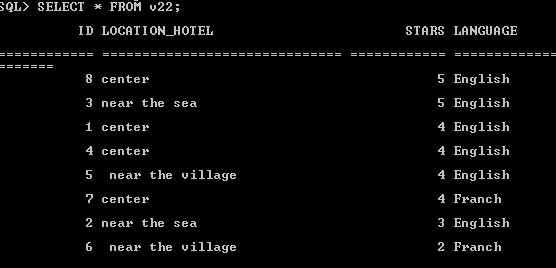
Результат:



3.4 Выборка сортировкой

Полученные результаты: Отели отсортированы по количеству звезд

|  |
| --- |
| create view v22 as select \* from hotel order by stars desc; |



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; |

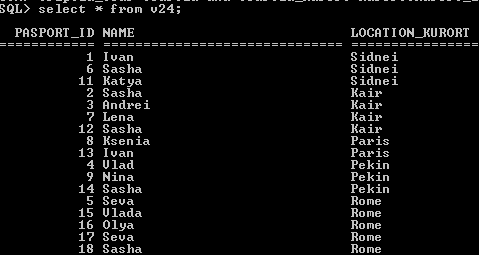
Полученные результаты:



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:



|  |
| --- |
| 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; |

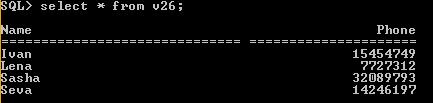
Выборка клиента и цены тура:



3.7  Запрос, рассчитывающий совокупную характеристику с использованием группировки.

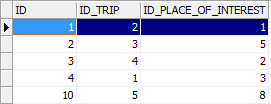
|  |
| --- |
| create view v26 as select name as "Name" , sum(number\_phone) as "Phone"  from client group by name having sum(number\_phone)>7700000; |

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



3.8

|  |
| --- |
| create view v885 as select \* from trip\_place where id\_trip  in(select type\_id from place\_of\_interest); |



3.9

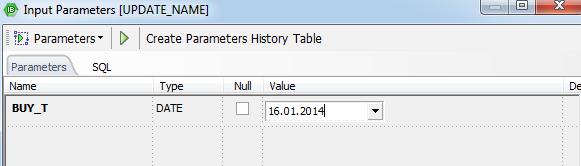
|  |
| --- |
| create procedure insert\_client (client\_ID integer, name varchar(10), number\_phone integer)  as  begin  insert into client(client\_ID, name, number\_phone) values (:client\_ID, :name, :number\_phone);  end;  create procedure insert\_trip (id\_client integer, id\_tour integer, id\_insuranc 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**

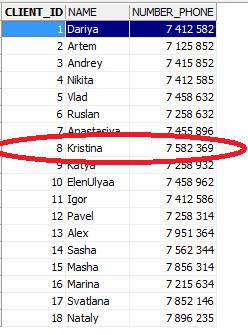
create procedure update\_name(buy\_t date) as begin

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

Результат работы скрипта:

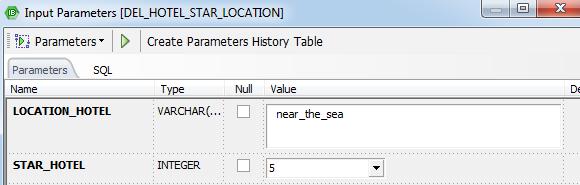


3.11С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

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:



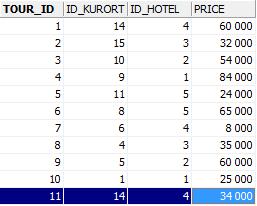
В результате была удалена запись из таблицы HOTEL в соответствии с заданными данными

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

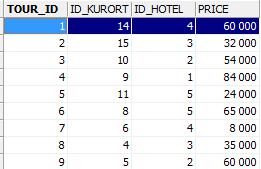
create procedure del\_hotel as begin

delete from hotel where hotel\_id not in (select id\_tour from tour); end;

До:



После выполнения скрипта:



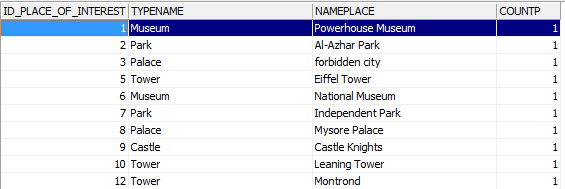
Индивидуальное задание

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.

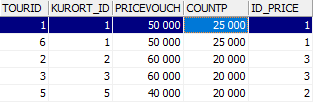
|  |
| --- |
| create view top10place as select first 10 id\_place\_of\_interest, typename,  nameplace, count(trip\_place.id\_trip) as countp  from trip, trip\_place, place\_of\_interest, type  where(trip.id=trip\_place.id\_trip)and(trip\_place.id\_place\_of\_interest=  place\_of\_interest.id)and(place\_of\_interest.type\_id=type.id)  group by id\_place\_of\_interest, typename, nameplace order by countp desc; |

Результат работы скрипта:



2.

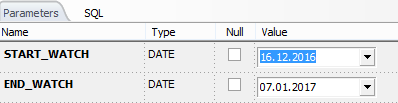
|  |
| --- |
| CREATE OR ALTER VIEW TOP5PLACE2(  TOURID,  KURORT\_ID,  id\_price,  PriceVouch,  countp  )  AS  select first 5 tour.id as tourID, kurort.kurort\_id as kurort\_id, id\_price,  adult as PriceVouch, (adult/count(place\_of\_interest.kurort\_id)) as countp  from tour, place\_of\_interest, price, 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 |

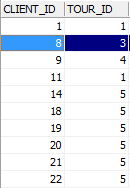


3.

|  |
| --- |
| 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 |

Результат:





**Выводы:**

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