**Федеральное агентство по образованию**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**По лабораторной работе №3**

**Язык SQL-DML**

**Студентка гр.43501/4 Перетятько Е.В.**

**Преподаватель Мяснов А. В.**

**Санкт-Петербург**

**2015**

1. **Цель работы**

Познакомиться с языком создания запросов управления данными SQL-DML.

1. **Программа работы**
2. Изучить SQL-DML
3. Выполнить все запросы из списка стандартных запросов. Продемонстрировать результаты преподавателю.
4. Получить у преподавателя и реализовать SQL-запросы в соответствии с индивидуальным заданием. Продемонстрировать результаты преподавателю.
5. Выполненные запросы SELECT сохранить в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложить скрипт в Subversion.
6. **Выполнение программы работы**

* Сделайте выборку всех данных из каждой таблицы

|  |
| --- |
| Use CONNECT or CREATE DATABASE to specify a database  SQL> connect 'localhost:D:\Turism.fdb' user 'SYSDBA' password 'masterkey';  Database: 'localhost:D:\Turism.fdb', User: SYSDBA  SQL> select \* from client;  NAME SURNAME NUMBER\_PHONE PAS  PORT\_ID INSURANCE\_PRICE  ============================== ============================== ============ =====  ======= ===============  Masha Petrova 4608345  789345 1  Dasha Vetrova 4567345  784567 2  Ivan Ivanov 5672341  744384 3  Pavel Sidorov 3848432  734829 1  Igor Kuznezov 7423847  785499 2  Anna Alexeeva 588  783243 3  SQL> select \* from type;  ID TYPENAME  ============ ==============================  1 Tower  2 'astle  3 Museum  4 Palace  5 Park  SQL> select \* from place\_of\_interest;  ID TYPE\_ID NAMEPLACE  ============ ============ ==============================  1 1 Tower of London  2 2 Hearst Castle  3 3 Museum of fine arts boston  4 4 Crystal Palace  5 5 Park in Paris  SQL> select \* from kurort;  ID LOCATION\_KURORT CLIMATE WEATH  ER ID\_PLACE\_OF\_INTEREST  ============ ============================== ============================== =====  ========================= ====================  1 London moderate sun  1  2 Paris moderate rain  5  3 Boston moderate snow  3  SQL> select \* from tour;  ID TIME\_TOUR ID\_KURORT ID\_HOTEL ID\_PRI  CE EXCURSIONS  ============ ============================== ============ ============ ==========  == ==============================  1 2 week 1 2  1 <null>  2 1 week 2 3  3 <null>  3 2 week 3 4  3 <null>  SQL> select \* from hotel;  ID LOCATION\_HOTEL STARS LANGUAGE  ============ ============================== ============ =======================  =======  1 center 5 English  2 center 3 English  3 near the sea 5 English  4 suburb 4 English  SQL> select \* from trip;  ID\_CLIENT ID\_INSURANCE ID\_TOUR  ============ ============ ============  789345 2 1  784567 2 2  744384 2 3  734829 2 1  785499 2 2  783243 2 3  SQL> select \* from price;  ID ADULT CHILD  ============ ============ ============  1 25000 12500  2 50000 25000  3 75000 35000  4 25000 7000  SQL> select \* from insurance;  ID INSURANCE\_CASES INSURANCE\_PRICE  ============ ============================== ===============  1 accident 500000  2 infectious disease 100000  SQL> |

* Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)

|  |
| --- |
| SQL> select \* from kurort where location\_kurort IN ('London');  ID LOCATION\_KURORT CLIMATE WEATH  ER ID\_PLACE\_OF\_INTEREST  ============ ============================== ============================== =====  ========================= ====================  1 London moderate sun  1  SQL> select \* from client where pasport\_id BETWEEN 740000 AND 785000;  NAME SURNAME NUMBER\_PHONE PAS  PORT\_ID INSURANCE\_PRICE  ============================== ============================== ============ =====  ======= ===============  Dasha Vetrova 4567345  784567 2  Ivan Ivanov 5672341  744384 3  Anna Alexeeva 588  783243 3  SQL> select \* from client WHERE Name LIKE 'Dasha';  NAME SURNAME NUMBER\_PHONE PAS  PORT\_ID INSURANCE\_PRICE  ============================== ============================== ============ =====  ======= ===============  Dasha Vetrova 4567345  784567 2 |

* Создайте в запросе вычисляемое поле

|  |
| --- |
| SQL> select id, adult, adult\*50 as price\_rub from price;  ID ADULT PRICE\_RUB  ============ ============ =====================  1 25000 1250000  2 50000 2500000  3 75000 3750000  4 25000 1250000 |

* Сделайте выборку всех данных с сортировкой по нескольким полям

|  |
| --- |
| SQL> select \* from client order by surname;  NAME SURNAME NUMBER\_PHONE PAS  PORT\_ID INSURANCE\_PRICE  ============================== ============================== ============ =====  ======= ===============  Anna Alexeeva 588  783243 3  Ivan Ivanov 5672341  744384 3  Igor Kuznezov 7423847  785499 2  Masha Petrova 4608345  789345 1  Pavel Sidorov 3848432  734829 1  Dasha Vetrova 4567345  784567 2 |

* Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

|  |
| --- |
| SQL> select avg(adult), max(adult), min(adult) from price;  AVG MAX MIN  ===================== ============ ============  43750 75000 25000 |

* Сделайте выборку данных из связанных таблиц (не менее двух примеров)

|  |
| --- |
| SQL> select 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\_kuro  rt=kurort.id;  NAME LOCATION\_KURORT  ============================== ==============================  Masha London  Pavel London  Dasha Paris  Igor Paris  Ivan Boston  Anna Boston  SQL> select 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;  NAME ADULT  ============================== ============  Masha 25000  Pavel 25000  Dasha 75000  Igor 75000  Ivan 75000  Anna 75000 |

* Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки

|  |
| --- |
| SQL> select name, sum(pasport\_id) from client group by name having sum(pasport\_i  d)>750000;  NAME SUM  ============================== =====================  Anna 783243  Dasha 784567  Igor 785499  Masha 789345 |

* Придумайте и реализуйте пример использования вложенного запроса

|  |
| --- |
| SQL> select client.name, client.surname, client.pasport\_id from client where cli  ent.pasport\_id in (select trip.id\_client from trip where trip.id\_tour like '2');  NAME SURNAME PASPORT\_ID  ============================== ============================== ============  Dasha Vetrova 784567  Igor Kuznezov 785499 |

* С помощью оператора INSERT добавьте в каждую таблицу по одной записи

|  |
| --- |
| INSERT INTO TYPE (id, typeName) VALUES (6, 'Platz');  INSERT INTO HOTEL (id, location\_hotel, stars, language) VALUES (5, 'near the sea', 2, 'English');  INSERT INTO PRICE (id, adult, child) VALUES (5, 15000, 4500);  INSERT INTO PLACE\_OF\_INTEREST (id, type\_id, namePlace) VALUES (6, 6, 'Platz');  INSERT INTO CLIENT (Name, Surname, number\_phone,pasport\_id, insurance\_price) VALUES ('Ksenia', 'Bistrova', 7388957, 785667, 5);  INSERT INTO KURORT (id, location\_kurort, climate, weather, id\_place\_of\_interest) VALUES (4, 'Moscow', 'moderate', 'snow and rain', 4);  INSERT INTO TOUR (id, time\_tour, id\_kurort, id\_hotel, id\_price) VALUES (4,'5 week', 1,4,3);  INSERT INTO TRIP (id\_client, id\_insurance, id\_tour) VALUES (785667,2,1); |

* С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию

|  |
| --- |
| SQL> update client set name='Sasha' where name='Masha';  SQL> select \* from client;  NAME SURNAME NUMBER\_PHONE PAS  PORT\_ID INSURANCE\_PRICE  ============================== ============================== ============ =====  ======= ===============  Sasha Petrova 4608345  789345 1  Dasha Vetrova 4567345  784567 2  Ivan Ivanov 5672341  744384 3  Pavel Sidorov 3848432  734829 1  Igor Kuznezov 7423847  785499 2  Anna Alexeeva 7348957  783243 3  Ksenia Bistrova 7388957  785667 5 |

* С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики

|  |
| --- |
| SQL> select \* from hotel;  ID LOCATION\_HOTEL STARS LANGUAGE  ============ ============================== ============ =======================  =======  1 center 5 English  2 center 3 English  3 near the sea 5 English  4 suburb 4 English  5 near the sea 2 English  SQL> delete from hotel where stars in (select min(stars) from hotel);  SQL> select \* from hotel;  ID LOCATION\_HOTEL STARS LANGUAGE  ============ ============================== ============ =======================  =======  1 center 5 English  2 center 3 English  3 near the sea 5 English  4 suburb 4 English |

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

|  |
| --- |
| SQL> INSERT INTO TYPE (id, typeName) VALUES (7, 'Monument');  SQL> select \* from type;  ID TYPENAME  ============ ==============================  1 Tower  2 Pastle  3 Museum  4 Palace  5 Park  6 Platz  7 Monument  SQL> select id, typename from type where not exists(select \* from place\_of\_inter  est where type.id=place\_of\_interest.type\_id);  ID TYPENAME  ============ ==============================  7 Monument  SQL> delete from type where not exists(select \* from place\_of\_interest where typ  e.id=place\_of\_interest.type\_id);  SQL> select \* from type;  ID TYPENAME  ============ ==============================  1 Tower  2 Pastle  3 Museum  4 Palace  5 Park  6 Platz |

**Вывод**

В ходе работы были изучены необходимые для работы с реляционными базами данных запросы языка SQL-DML: insert, update, delete. Изучены внутреннее и внешнее объединение таблиц, группировки и агрегатные функции, вложенные запросы, а также представления. Полученные навыки работы с SQL-запросами будут использованы в дальнейшей работе с базой данных.DML запросы имеют довольно большие возможности для манипуляции базами данных.