

Санкт-Петербургский политехнический университет
Петра Великого
Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе

Дисциплина: «Базы данных»

Тема: «Язык SQL-DML»

Выполнил студент гр. 43501/3

(подпись) М.Ю. Попсуйко

Преподаватель

(подпись) А.В. Мяснов

“ ____ ” _____ 2016 г.

Санкт-Петербург

2016

Оглавление

Цель работы:	3
Программа работы:.....	3
Выполнение работы:	3
Подготовка базы данных	3
Выполнение запросов:	4
Создание sql-запросов.....	6
Выводы:	7

Цель работы:

Создание запросов управления данными SQL-DML.

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

Выполнение работы:

Подготовка базы данных

Для начала выполним подготовку базы данных:

Таблица order была переименована в таблицу sells, так как order уже зарезервированная команда. Так же в нее был добавлен столбец sell_date для контроля даты продажи, и изменен тип столбца price.

```
CREATE TABLE SELLS (
    ID            INTEGER NOT NULL,
    ID_CAR        INTEGER,
    ID_CLIENT     INTEGER,
    ID_EMPLOYEE   INTEGER,
    PRICE         DECIMAL(18,2),
    SELL_DATE     DATE
);

ALTER TABLE SELLS ADD CONSTRAINT PK_SELL PRIMARY KEY (ID);

ALTER TABLE SELLS ADD CONSTRAINT FK_SELL_1 FOREIGN KEY (ID_CLIENT)
REFERENCES CLIENT (ID);
ALTER TABLE SELLS ADD CONSTRAINT FK_SELL_2 FOREIGN KEY (ID_EMPLOYEE)
REFERENCES EMPLOYEE (ID);
ALTER TABLE SELLS ADD CONSTRAINT FK_SELL_3 FOREIGN KEY (ID_CAR)
REFERENCES CAR (ID);
```

Также был добавлен foreign key из таблицы additional_options:

```
ALTER TABLE additional_options ALTER COLUMN ID_ORDER TO ID_SELL;
ALTER TABLE additional_options ADD CONSTRAINT FK_ADDITIONAL_OPTIONS_2
FOREIGN KEY (ID_SELL) REFERENCES SELLS (ID);
```

Восстановлен foreign key из таблицы sells:

```
ALTER TABLE SELLS ADD CONSTRAINT FK_SELL_3 FOREIGN KEY (ID_CAR) REFERENCES CAR (ID);
```

В таблицу car_equipment добавим столбец price. В таблицу options добавляем столбец price:

```
ALTER TABLE car_equipment ADD PRICE FLOAT;  
ALTER TABLE options ADD PRICE FLOAT;
```

Заполнили таблицы необходимым количеством случайных значений.

Выполнение запросов:

1. Сделайте выборку всех данных из каждой таблицы:

```
select * from additional_options;  
select * from car;  
select * from car_body_type;  
select * from "CAR_DEFAULT-OPTIONS";  
select * from car_equipment;  
select * from car_models;  
select * from car_modification;  
select * from client;  
select * from drive_gear;  
select * from employee;  
select * from engine_type;  
select * from factory;  
select * from gear_box;  
select * from options;  
select * from sells;
```

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

```
--Выбор модели авто, название которой начинается на MAX  
select * from car_models where name like 'M%';  
  
--Выбор авто 2012-2014 года производства  
select * from car where car_year between 2012 and 2014;  
  
--Выбор авто стоимостью: 509399.39, 230216.95  
select * from sells where price in (509399.39, 230216.95);
```

3. Создайте в запросе вычисляемое поле:

```
--Скидуля на авто 25%  
select id, sells.price, (price*0.75) discount from sells;
```

4. Сделайте выборку всех данных с сортировкой по нескольким полям:

```
--Сортировка авто по названию и дате продажи  
select * from car order by id, car_year;
```

5. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц:

```
--Характеристики заказов среднее, мин, max
```

```
select avg(price), min(price), max(price) from sells;
```

6. Сделайте выборку данных из связанных таблиц (не менее двух примеров):

```
--Узнаем фамилию клиента и стоимость машины, которую он купил
select sells.id, sells.price as sells_info, client.second_name as
client_info from sells join client on sells.id_client=client.id order by
sells.id;

--Определим в какой стране какую колесницу собрали
select car.id as car_info, factory.country as factory_info from car join
factory on car.id_factory=factory.id order by car.id;
```

7. Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки:

```
--вывод 5 самых популярных стран производителей
select first 5 factory.country, count(country) as total_count from
factory
join car on car.id_factory=factory.id
where car.id_equipment=1
group by country
order by total_count desc;
```

8. Придумайте и реализуйте пример использования вложенного запроса:

```
--выберем авто, которые собраны в стране: ALIJXPIIHO
select * from car where id_factory=(select id from factory where
country='ALIJXPIIHO');
```

9. С помощью оператора INSERT добавьте в каждую таблицу по одной записи:

```
insert into additional_options (id_sell, id_options) values (100101,
100101);
insert into options (id, name, price) values (100101, 'jfkdf', 104556);
insert into client (id, first_name, second_name, address) values
(100101, 'FDDDBFFB', 'FDBDFB', 'fvdfdv', '88005553535');
insert into employee (id, first_name, second_name, address) values
(100101, 'DDDBFFB', 'DDDDFB', 'dddfdfv', '89005553535');
insert into sells (id, id_car, id_client, id_employee, price, sell_date)
values (100101, 100101, 100101, 100101, 2500000, '10.12.2016');
insert into factory (id, country, city, address, phone) values (100101,
'STRANA', 'GOROD', 'address', 88005553536);
insert into car_equipment (id, name, price) values (4, 'Дешман',
100002);
insert into car_default-options (id_options, id_equipment) values
(100101, 100101);
insert into car_modification (id, id_car_model, car_body_type,
engine_type, drive_gear, gear_box) values (100101, 100101, 100101, 100101,
100101, 100101);
insert into car_models (id, name) values (101, 'ASDDDHJK');
insert into car_body_type (id, name) values (101, 'ASDDDHJK');
insert into gear_box (id, name) values (101, 'ASDDDHJK');
insert into engine_type (id, name) values (101, 'ASDDDHJK');
insert into drive_gear (id, name) values (101, 'ASDDDHJK');
```

10. С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию:

```
--Поднимем цену на 100000 на все авто
update sells set price=price+100000;
```

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

```
insert into options (id, name, price) values (100102, 'TEST', 99999999);
delete from options where id=(select id from options where price=(select
max(price) from options));
```

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

```
delete from car where id in(
select car.id from car where car.id not in(
select id_car from sells));
```

Создание sql-запросов

Запрос 1: Вывести все автомобили, проданный в самой дорогой комплектации за последние 5 лет.

```
select car.id as car_id from car
join car_equipment on car.id_equipment=car_equipment.id
join sells on sells.id_car=car.id
where car_equipment.id=(select car_equipment.id
from(select first 1 car_equipment.id, car_equipment.price from
car_equipment order by price desc))
and sells.sell_date between (dateadd(year, -5, current_timestamp)) and
current_date;
```

Запрос 2: Вывести 5 сотрудников, которые продали дополнительных опций на наибольшую сумму за последний год.

```
select first 5 employee.id,
(employee.second_name || ' ' || employee.first_name) as Sotrudnik,
sum(options.price) as total_sum
from employee
join sells on sells.id_employee=employee.id
join additional_options on additional_options.id_sell=sells.id
join options on options.id=additional_options.id_options
where sells.sell_date between (dateadd(year, -1, current_timestamp))
and current_date
group by employee.id, Sotrudnik
order by total_sum desc;
```

Запрос 3: Вывести наиболее популярный тип трансмиссий среди наиболее дешевых комплектаций для всех моделей.

```
select gear_box.id, gear_box.name, count(gear_box.id) as total_count
from gear_box
join car_modification on car_modification.gear_box=gear_box.id
join car on car.id_modification=car_modification.id
join car_equipment on car_equipment.id=car.id_equipment
where car_equipment.id=(select car_equipment.id
```

```
from(select first 1 car_equipment.id, car_equipment.price from  
car_equipment order by price))  
group by gear_box.id,gear_box.name  
order by total_count desc;
```

Выводы:

В результате выполнения данной лабораторной работы были изучены основы создания запросов управления SQL-DML. Были изучены основные операции языка SQL_DML: select – выборка из таблицы, insert – добавление записи, join – объединение таблиц для выборки из них, where – условие для выборки, order by – группировка выбранных данных, update – загрузка новых данных в таблицу, delete – удаление данных из таблицы. Научился объединять и группировать записи, делать вложенные запросы. Реализовал SQL-запросы в соответствии с индивидуальным заданием.