Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №4**

**Дисциплина**: Базы данных

**Тема**: Язык SQL-DML

Выполнил студент гр. 43501/1 Приймак А.Г.

(подпись)

Руководитель Мяснов А.В.

(подпись)

“ ” 2016 г.

Санкт-Петербург

2016

1. **Цель работы**

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

1. **Программа работы**
2. Изучите SQL-DML
3. Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
4. Получите у преподавателя и реализуйте SQL-запросы в соответствии с **индивидуальным** заданием. Продемонстрируйте результаты преподавателю.
5. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложите скрипт в Subversion.
6. **Язык SQL**

SQL (англ. structured query language — «язык структурированных запросов») — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД, в данном случае Firebird)

SQL-DML (Data Manipulation Language) -- язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

1. **Выполнение работы**
2. Выборка всех данных из таблицы:

create view d1 as select \* from catalog;

commit;

Пример выборки данных из таблицы catalog:



1. Выборка данных из одной таблицы при нескольких условиях, с использованием логических операций

Пример выборки для логического оператора AND для таблицы games;

create view d2 as select \* from catalog where id\_class = 2 and price < 1000;

commit;



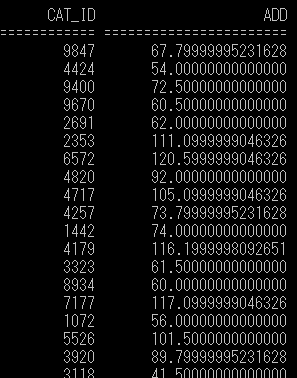
1. Вычисляемое поле:

create view d3(cat\_id, summa) as select cat\_id, mass+bullet\_mass

from ttc;

commit;

Сумма всех характеристик массы для образцов из таблицы ТТХ:



1. Сортировка по нескольким полям:

create view d4 as select \* from catalog order id\_def, price;

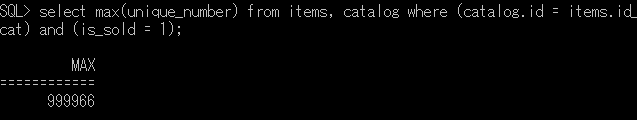
commit;



1. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц:

select max(unique\_number) from items, catalog where (catalog.id = items.id\_cat) and (is\_sold = 1);

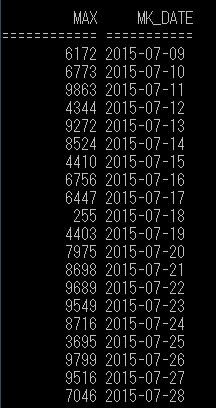
commit;



1. Использование совокупной харастеристики с группировкой, и наложением ограничения на результат:

select max( id\_client), mk\_date from orders group by mk\_date;

commit;



1. Вложенный запрос:

select \* from catalog where id\_caliber in (select id from caliber where caliber>7);

commit;



1. Использование оператора INSERT:

set term ^;

CREATE PROCEDURE InsProc

AS

BEGIN

insert into classes

values(7,'Hunt\_waepons');

END^

Set term ;^

1. Использование оператора Update (с определенным условием)

set term ^;

CREATE PROCEDURE UpdClient

AS

BEGIN

update clients set name=Alan where id = 5;

END^

set term ;^

1. Использование оператора DELETE:

set term ^;

CREATE PROCEDURE DelProc1

AS

BEGIN

delete from orders order by mk\_date;

END^

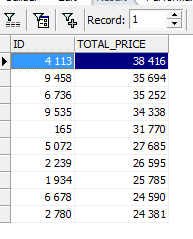
Выполнение индивидуального задания:

Выполнить следующие запросы:

1. За заданный период вывести 10 классов оружия с максимальными продажами.
2. Вывести клиентов, которые приобретали оружие более 5 различных классов.
3. Удалить неиспользуемые классы.
4. За заданный период вывести 10 классов оружия с максимальными продажами:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | SELECT first 10 id, SUM(price) as total\_price FROM catalog, | |  | (select id\_cat from items, | |  |  | |  |  | |  | (select id\_items from orders\_lists, | |  | (select id as order\_id from orders where orders.mk\_date between date '01.01.2010' | |  | and date '01.01.2015') | |  | where orders\_lists.id\_order = order\_id) | |  |  | |  | where items.id = id\_items) | |  | where catalog.id = id\_cat GROUP BY id ORDER BY total\_price DESC | |

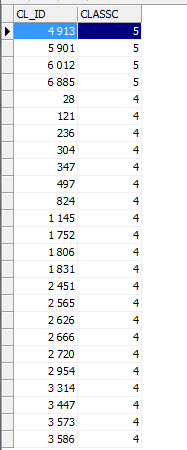
Результат работы скрипта.



1. Вывести клиентов, которые приобретали оружие более 5 различных классов:

|  |
| --- |
| select \* from (select cl\_id, COUNT(DISTINCT id\_class) as classc FROM catalog, (select cl\_id, id\_cat from items, (select cl\_id, id\_item from orders, (select clients.id as cl\_id from clients) where orders.id\_client = cl\_id) where items.id = id\_item) where catalog.id = id\_cat GROUP BY cl\_id order by classc desc) where classc > 2 |

Результат работы:



1. Удалить неиспользуемые классы:

|  |
| --- |
| delete from classes where classes.id NOT IN (select id\_class from catalog) |

**Вывод**

В результате работы было проведено знакомство с языком SQL-DML. C помощью данного языка были получены скрипты, для изучения стандартных запросов извлечения данных. Также были выполнены запросы в соответствие с индивидуальным заданием. выполнения работы был изучен язык управления данными SQL-DML. Были выполнены стандартные запросы извлечения данных. Также были выполнены запросы в соответствии с индивидуальным заданием.

Спектр команд языка DML не велик (insert (добавить), update (обновить), delete (удалить), select (выборка данных)), в связи с этим язык удобен для изучения и понятен. Главный вопрос, в котором следует разобраться это написание правильных и быстро выполняющихся запросов, ведь если нашей БД будут пользоваться множество клиентов одновременно, скорость выполнения запросов одного клиента косвенно будет влиять на производительность БД в целом. Если следовать некоторым “неписанным правилам”, таким как обращение к различным таблицам с помощью создания связей между таблицами, а не вложенные запросы, то можно реализовать производительные запросы любой сложности. С помощью языка DML создать необходимую связь между таблицами, подсчитать некоторое значение, или вывести данные соответствующие некоторому логическому выражению или ограничению достаточно просто и удобно.