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 Д. С. Раскин.

(подпись)

Руководитель А. В. Мяснов

(подпись)

“ ” 2015 г.

Санкт-Петербург

2015

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)

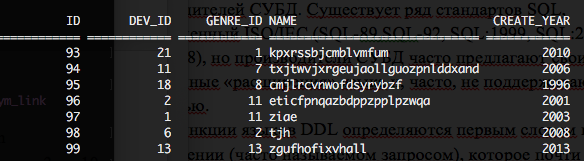
SQL-DML (Data Manipulation Language) -- язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

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

create view s1 as select \* from games;

commit;

Пример выборки данных из таблицы GAMES:



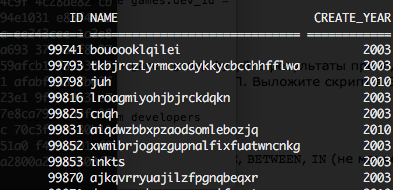
1. Выборка данных из одной таблицы при нескольких условиях, с использованием логических операций

Пример выборки для логического оператора OR для таблицы games;

create view s2(id,name,create\_year) as select id,name,create\_year

from games where create\_year = 2010 or create\_year = 2003;

commit;



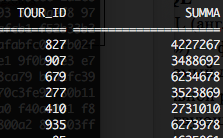
1. Вычисляемое поле:

create view s3(tour\_id, summa) as select tour\_id,first\_place+second\_place+third\_place+finals+groups

from pool;

commit;

Сумма всех призовых для каждого турнира из таблицы призовых:



1. Сортировка по нескольким полям:

create view s4 as select \* from games order by create\_year, id;

commit;

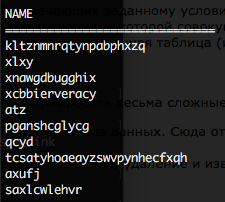


1. Запрос из нескольких таблиц:

create view s5(name) as select games.name from games, developers where games.dev\_id = developers.id

AND games.create\_year = developers.create\_year;

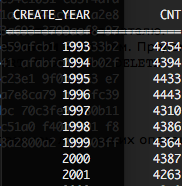
commit;



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

create view s6(create\_year, cnt) as select create\_year, count(id) from games group by create\_year having count(id) > 2;

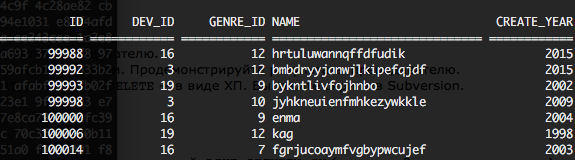
commit;



1. Вложенный запрос:

create view s7 as select \* from games where dev\_id in ( select id from developers where create\_year>2000);

commit;



1. Использование оператора INSERT:

set term ^;

CREATE PROCEDURE InsProc

AS

BEGIN

insert into Developers

values(22,5,'For delete',2015);

END^

Set term ;^

1. Использование оператора Update(с определенным условием)

set term ^;

CREATE PROCEDURE UpdProc

AS

BEGIN

update developers set create\_year=1994 where id = 21;

END^

set term ;^

1. Использование оператора DELETE:

set term ^;

CREATE PROCEDURE DelProc1

AS

BEGIN

delete from pool order by first\_place+second\_place+third\_place+finals+groups desc rows 1;

END^

Выполнение индивидуального задания:

Выполнить следующие запросы:

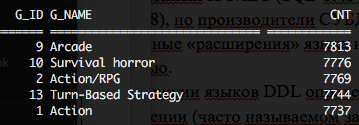
1. Отобразить 5 наиболее популярных жанров игр.
2. Вывести рейтинг игроков по призовым за турниры.
3. Удалить неиспользуемых разработчиков.
4. 5 Наиболее популярных жанров:

create view my1 as

select \* from genre where id in (

select genre\_id from games group by genre\_id order by count(genre\_id) desc rows 5);

Результат работы скрипта. 5 самых популярных жанров игр



1. Вывести рейтинг игроков по призовым за турниры:

Если необходимо узнать рейтинг игроков по призовым за конкретные места(1ое,2ое,3е, участие в финалах или групповом этапе турнира) следует вызывать одно из нижепредставленных представлений:

create view firsts(team\_id, prize) as

select tables.winner, sum(pool.first\_place) from tables, pool,

(select winner as W, tour\_id as T from tables where match\_id =5)

where pool.tour\_id = T and tables.winner = W

group by tables.winner;

create view seconds(team\_id, prize) as

select tables.winner, sum(pool.first\_place) from tables, pool,

(select winner as W, tour\_id as T from tables where match\_id =4 or match\_id=10 or match\_id=15)

where pool.tour\_id = T and tables.winner = W

group by tables.winner;

create view thirds(team\_id, prize) as

select tables.winner, sum(pool.first\_place) from tables, pool,

(select winner as W, tour\_id as T from tables where match\_id =3 or match\_id =9 or match\_id=14)

where pool.tour\_id = T and tables.winner = W

group by tables.winner;

create view finals(team\_id, prize) as

select tables.winner, sum(pool.first\_place) from tables, pool,

(select winner as W, tour\_id as T from tables

where match\_id = 6 or match\_id = 11 or match\_id=16 or match\_id =2 or match\_id = 8 or match\_id = 13)

where pool.tour\_id = T and tables.winner = W

group by tables.winner;

create view groups(team\_id, prize) as

select tables.winner, sum(pool.first\_place) from tables, pool,

(select winner as W, tour\_id as T from tables where match\_id =1)

where pool.tour\_id = T and tables.winner = W

group by tables.winner;

Если необходим полный рейтинг то используем нижепредставленное представление:

create view rating(team\_id,prize)

as

select \* from finals

union distinct

select \* from seconds

union distinct

select \* from thirds

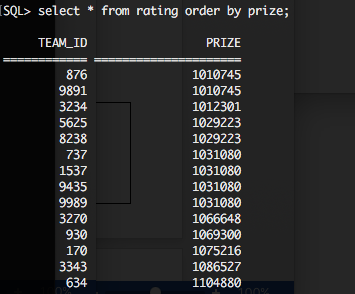
union distinct

select \* from finals

union distinct

select \* from groups;

Отрывок результата работы:



1. Удалить неиспользуемых разработчиков:

set term ^;

CREATE PROCEDURE DelProcMy

AS

BEGIN

delete from developers where id not in (

select dev\_id from games group by dev\_id having count(dev\_id)>0);

END^

set term ;^

1. **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.