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. 43501/4 М.C. Хрусталёва

(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**

Язык SQL (Structured Query Language) -- язык структурированных запросов. Он позволяет формировать весьма сложные запросы к базам данных. В SQL определены два подмножества языка:

SQL-DDL (Data Definition Language) -- язык определения структур и ограничений целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.

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

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

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create view AnimalClassv as select \* from animalclass;

create view Animalsv as select \* from animals;

create view Animalshowv as select \* from animalshow;

create view Countryv as select \* from country;

create view Deliveriesv as select \* from deliveries;

create view Detachmentv as select \* from detachment;

create view Eatv as select \* from eat;

create view Kindv as select \* from kind;

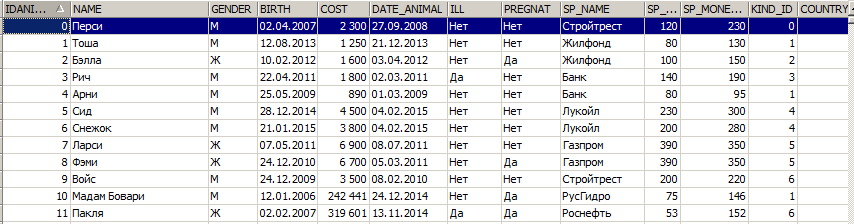
create view Linkprodv as select \* from linkprod;

create view Productsv as select \* from products;

create view Showviewv as select \* from showview;

create view Suppliersv as select \* from suppliers;

Выборка данных из таблицы Animals:



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

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create view selectShezhok as select \* from Animals where Name like '1';

create view selectPop20k as select \* from Animalclass where Popul between 0 and 20000000;

create view selectbyprice as select \* from Animals where Cost in (1600,1800);

Были созданы следующие выборки:

* Из таблицы Animals показать животных с именем Snezhok
* Из таблицы AnimalClass показать виды животных с населением от 0 до 200000000
* Из таблицы Animals показать животных с ценой 1600 или 1800.

Пример выборки с животными с ценой 1600 или 1800:

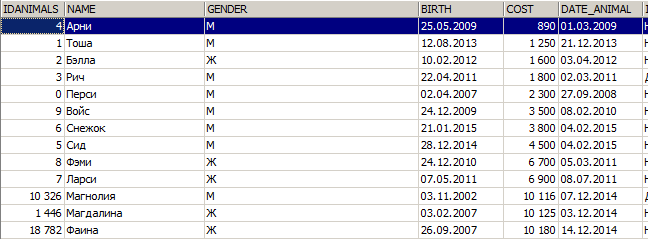


1. Выборка всех данных с сортировкой по нескольким полям

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create view sortAnimals as select \* from Animals order by cost asc, Date\_Animal asc;

Животные, отсортированные по цене и дате:



1. Выборка данных из связанных таблиц

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create view AnimalCountry as select Animals.name as Name,

Country.name as Country from Animals, Country where Animals.country\_id = Country.idcountry;

create view AnimalKind as select Animals.name as Name,

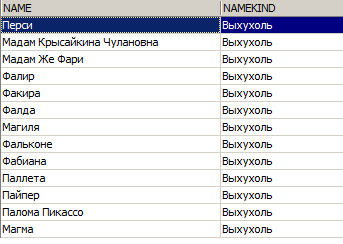
Kind.name as NameKind from Animals, Kind where Animals.kind\_id = Kind.idkind;

Выборки, отображающие:

* Имена животных и их страны:



* Имена животных и их тип:



1. Запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки

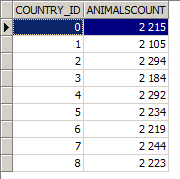
connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create view CountryCountAnimals as select Animals.country\_id as Country\_ID,

COUNT(Animals.country\_id) as AnimalsCount

from Animals group by Animals.country\_id;

Определение количества животных по странам:



1. Использование вложенного запроса

create view AnimalsNotInShow as select \* from Animals where

Animals.idanimals

not in (select ANIMALID from ANIMALSHOW);

Отображение всех животных, которые не участвую в представлениях:



1. Использование оператора INSERT для добавления по одной записи в каждую таблицу:

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create procedure insert\_animalclass (c int, cn varchar(30), p int)

as begin

insert into animalclass

values (:c, :cn, :p);

end;

create procedure insert\_animal (i int, n varchar(128), g varchar(45), b date, c numeric(9,0), da date,

il varchar(128), p varchar(128), spn varchar(128), spmi numeric(9,0), spma numeric(9,0),

ki int, ci int)

as begin

insert into animals

values (:i, :n, :g, :b, :c, :da, :il, :p, :spn, :spmi, :spma, :ki, :ci);

end;

create procedure insert\_animalshow(s int, a int, sv int, sd date)

as begin

insert into animalshow

values (:s, :a, :sv, :sd);

end;

create procedure insert\_country(i int, n varchar(128))

as begin

insert into country

values (:i, :n);

end;

create procedure insert\_deliver(i int, c numeric(9,0), d date, si int)

as begin

insert into deliveries

values (:i, :c, :d, :si);

end;

create procedure insert\_detachment(i int, n varchar(128), p int, c int)

as begin

insert into detachment

values(:i, :n, :p, :c);

end;

create procedure insert\_eat(i int, a numeric(9,0), ai int, pid int)

as begin

insert into eat

values(:i, :a, :ai, :pid);

end;

create procedure insert\_kind(i int, n varchar(128), l varchar(128), uk int, did int)

as begin

insert into kind

values(:i, :n, :l, :uk, :did);

end;

create procedure insert\_linkprod(l int, did int, pid int)

as begin

insert into linkprod

values(:l, :did, :pid);

end;

create procedure insert\_product(i int, n varchar(128), u date)

as begin

insert into products

values(:i, :n, :u);

end;

create procedure insert\_showview(i int, n varchar(100))

as begin

insert into showview

values(:i, :n);

end;

create procedure insert\_suppliers(i int, n varchar(100))

as begin

insert into suppliers

values(:i, :n);

end;

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

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create procedure update\_animal\_cost(tcost int) as

begin

update ANIMALS set COST =:tcost where ANIMALS.country\_id =0;

end;

Изменение цены животных из России.

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

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create procedure del\_expensive\_animal(i int) as

begin

delete from Animals where country\_id = :i and

cost = (select MAX(COST) from Animals where

country\_id =:i);

end;

Данная процедура удаляет самое дорогое животное из выбранной страны.

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

connect 'D:/ZOO2.fdb' user 'SYSDBA' password 'masterkey';

create procedure del\_unuseful\_animals as

begin

delete from Animals where Animals.idanimals

not in (select ANIMALID from ANIMALSHOW);

end;

Удаление животных, которые не участвуют в выступлениях.

Выполнение индивидуального задания:

Выполнить следующие запросы:

1. Вывести 5 видов, на питание которых потрачено больше всего денег за выбранный период времени.
2. Вывести 10 стран, из которых привезено наибольшее количество видов животных заданного семейства.
3. Удалить неиспользуемых поставщиков.

Для тестирования скриптов было сгенерировано по 100000 записей в соответствующие таблицы.

1. Вывести 5 видов, на питание которых потрачено больше всего денег за выбранный период времени.

connect 'D:/ZOO22.fdb' user 'SYSDBA' password 'masterkey';

create view top5expkinds as

select first 5 kind.name as KindName,

sum(deliveries.cost) as Cost from kind, eat, deliveries, animals where

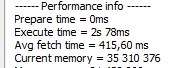
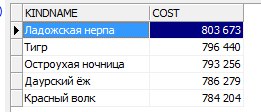
deliveries.date\_del between '2000-10-31' and '2016-10-31' and -- за выбранный период

eat.deliv\_id = deliveries.iddeliveries and

animals.idanimals = eat.animal\_id and

animals.kind\_id = kind.idkind

group by kind.name order by Cost desc;



1. Вывести 10 стран, из которых привезено наибольшее количество видов животных заданного семейства

connect 'D:/ZOO22.fdb' user 'SYSDBA' password 'masterkey';

create view top10countries as

select

first 10 country.name as CountryName,

count (kind.idkind) as KindsCount from country,animals, detachment, animalclass, kind

where

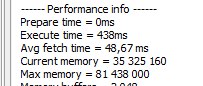
detachment.classid = 1 and -- цифрой указывается id семейства

kind.detachment\_id = detachment.iddetachment and

animals.kind\_id = kind.idkind and

animals.country\_id = country.idcountry

group by country.name order by KindsCount desc;



1. Удалить неиспользуемых поставщиков

connect 'D:/ZOO22.fdb' user 'SYSDBA' password 'masterkey';

create procedure del\_unuse\_sup as

begin

delete from suppliers where

suppliers.idsuppliers not in (select supplier\_id from deliveries);

end;

1. **Вывод**

В результате выполнения работы был изучен язык управления данными SQL-DML. Были выполнены стандартные запросы извлечения данных. Также были выполнены запросы в соответствии с индивидуальным заданием. Были изучены представления и хранимые процедуры, с помощью которых можно спокойно добавлять данные в БД. При выполнении работы проблем не было.

Использовались такие команды языка DML: insert (добавить), update (обновить), delete (удалить), select (выборка данных). Данный язык удобен для написания запросов разной сложности. При обращении к нескольким таблицам в запросе следует выбирать подходящую связь между таблицами для более быстрого выполнения SQL-запроса.