Санкт-Петербургский Государственный Политехнический Университет

Институт Информационных Технологий и Управления

Кафедра Компьютерных Систем и Программных Технологий

**Отчет**

**о лабораторной работе №3**

**Дисциплина:** Базы данных

**Тема:** Язык SQL-DML

**Выполнил**: гр.43501/1 Саитов Илья

**Преподаватель:** Мяснов А.В.

Санкт-Петербург 2015

1. **Цель работы:**

Ознакомление с основами проектирования схемы БД и способами нормализации отношений.

1. **Программа работы:**

**а)** Сделать выборку всех данных из каждой таблицы.

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

**в)** Создать в запросе вычисляемое поле.

**г)** Сделать выборку всех данных с сортировкой по нескольким полям

**д)** Создать запрос, вычисляющий несколько совокупных характеристик таблиц.

**е)** Сделать выборку данных из связанных таблиц (не менее двух примеров).

**ё)** Создать запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки.

**ж)** Придумать и реализовать пример использования вложенного запроса

**з)** С помощью оператора INSERT добавить в каждую таблицу по одной записи.

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

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

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

**м)** Выполнить серию индивидуальных заданий.

1. **Выполнение работы:**

**а)** **Произведем выборку всех данных из каждой таблицы с помощью опертора выбора SELECT**:

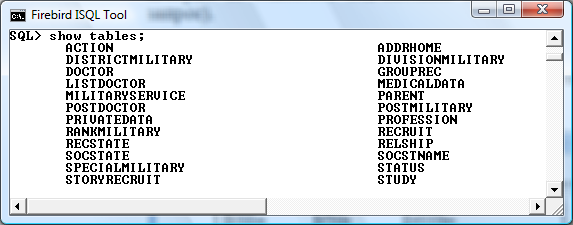


Рис.3.1. Список всех имющихся таблиц.

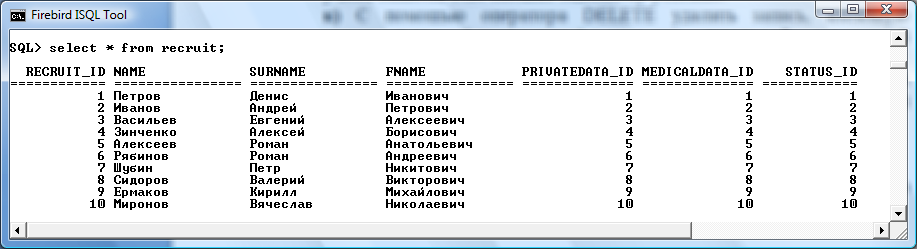


Рис.3.1. Содержимое таблицы Recruit.

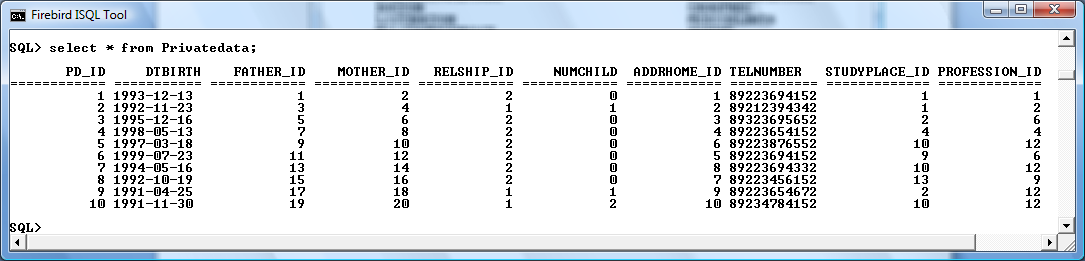


Рис.3.2. Содержимое таблицы PrivateData.

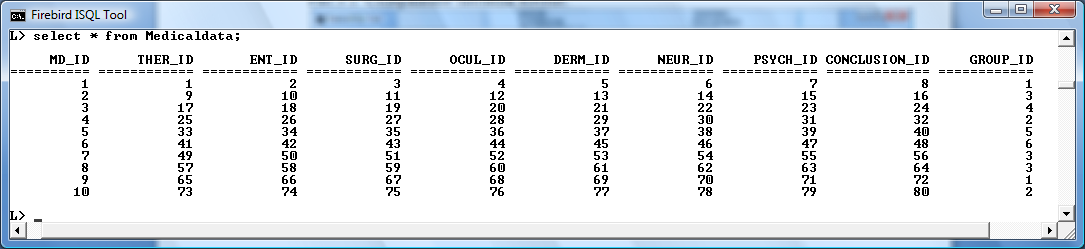


Рис.3.3. Содержимое таблицы MedicalData.

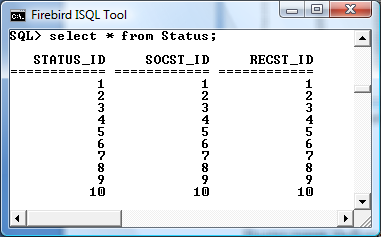


Рис.3.4. Содержимое таблицы Status.

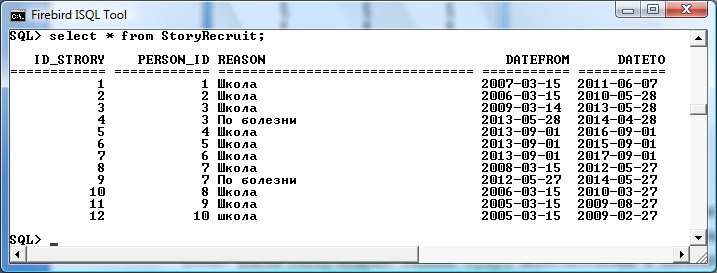


Рис.3.5. Содержимое таблицы StoryRecruit.

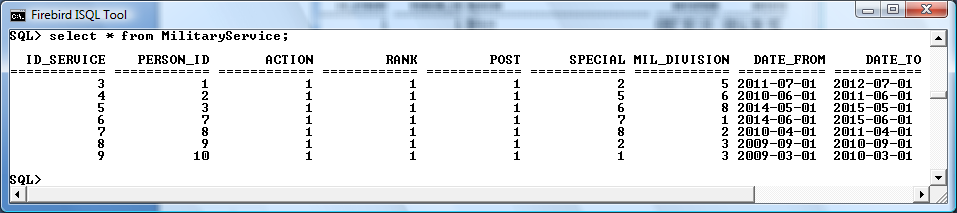


Рис.3.5. Содержимое таблицы MilitaryService.

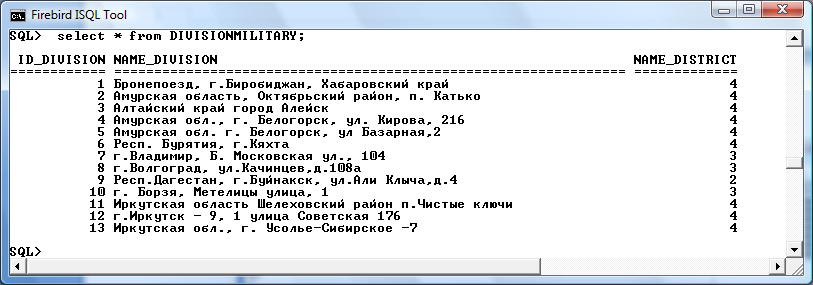


Рис.3.6. Содержимое таблицы DivisionMilitary.

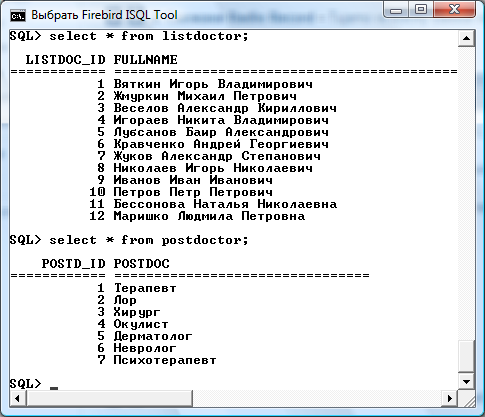


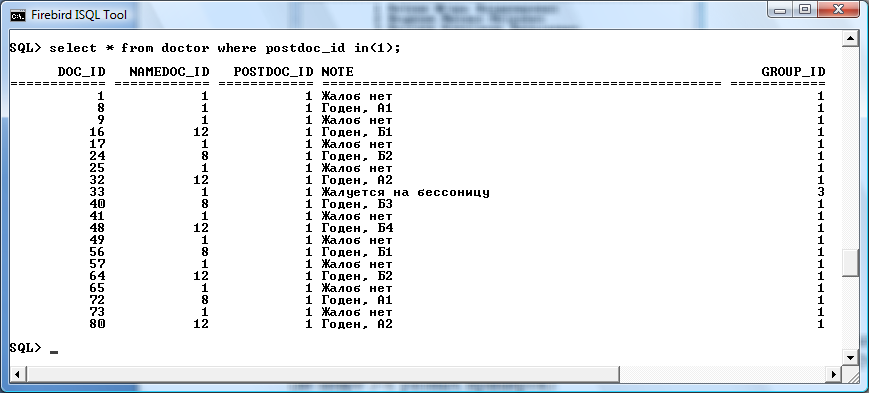
Рис.3.7. Содержимое таблицы-справочника ListDoctor и PostDoctor.

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

Выберем, для реализации данной задачи, таблицу Doctor, которая хранит данные о медицинских осмотрах призывников.

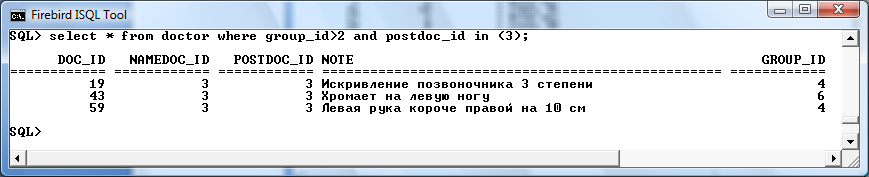
**Пример 1.** Выведем все записи терапевтов:

SELECT \* FROM doctor WHERE postdoc\_id IN (1);

****

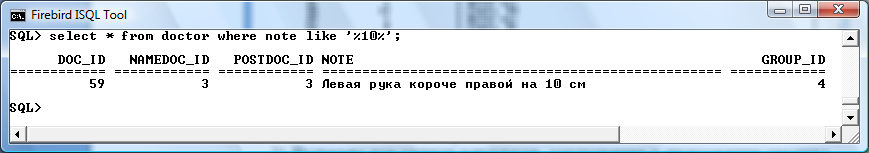
**Пример 2.** Выведем все записи хирургов, заключение о присвоении группы которых, все кроме группы «А»:

SELECT \* FROM doctor WHERE group\_id>2 AND postdoc\_id IN (3);



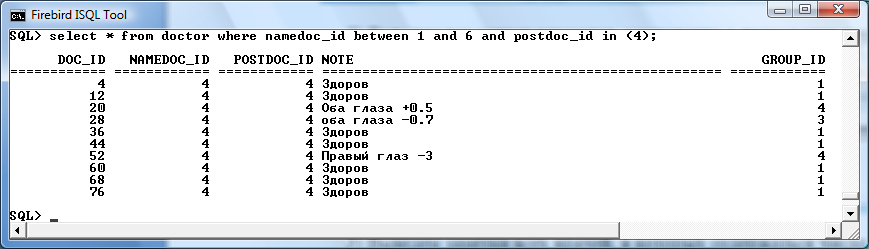
**Пример 3.** Выведем заметки всех врачей, в которых содержаться число 10:

SELECT \* FROM doctor WHERE note LIKE '%10%';



**Пример 4.** Выведем из 6-ти первых в списке всех врачей, только записи окулистов:

SELECT \* FROM doctor WHERE namedoc\_id BETWEEN 1 AND 6 AND postdoc\_id IN (4);



**в)** **Создать в запросе вычисляемое поле:**

Подсчитаем, сколько военнослужащих на должности солдат служат на данный момент и посчитаем общую сумму всех их зарплат. Создадим представление, которое будет сожержать результат выборки первой подзадачи. А далее, работая с это выборкой будем вычислять искомую сумму.

CREATE VIEW report AS

SELECT recruit.name,

recruit.surname,

recruit.fname,

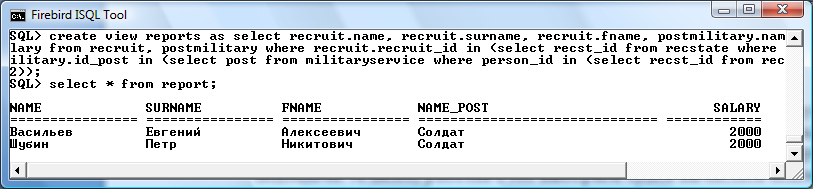
postmilitary.name\_post,

postmilitary.salary

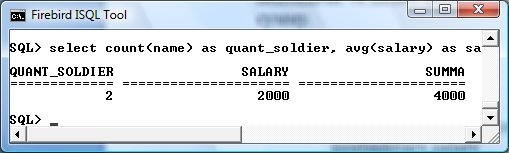
FROM recruit, postmilitary

WHERE recruit.recruit\_id IN (SELECT recst\_id FROM recstate WHERE staters\_id=2) AND

postmilitary.id\_post IN (SELECT post FROM militaryservice WHERE person\_id IN (SELECT recst\_id FROM recstate WHERE staters\_iD=2));



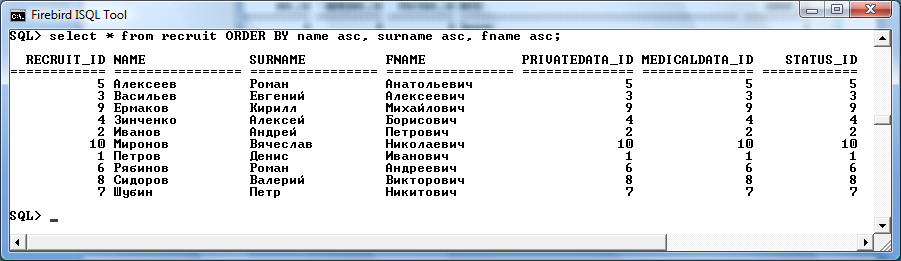
SELECT COUNT(name) AS quant\_soldier, AVG(salary) AS salary, COUNT(name)\*AVG(salary) AS summa FROM REPORT;



**г)** **Сделать выборку всех данных с сортировкой по нескольким полям:**

Отсортируем по имени фамили и отчеству список всех призывников (военнослужащих) числящихся в БД нашего военкомата:

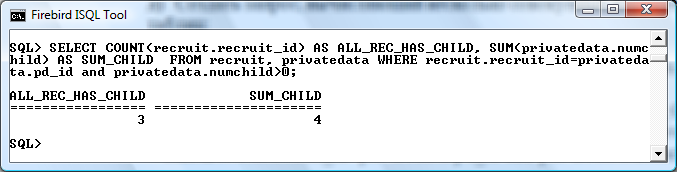
SELECT \* FROM recruit ORDER BY name ASC, surname ASC, fname ASC;



**д)** **Создать запрос, вычисляющий несколько совокупных характеристик таблиц:**

Для примера, определим, сколько всего военнослужащих/призывников в БД военкомата имеют детей и сколько всего детей у них в сумме:

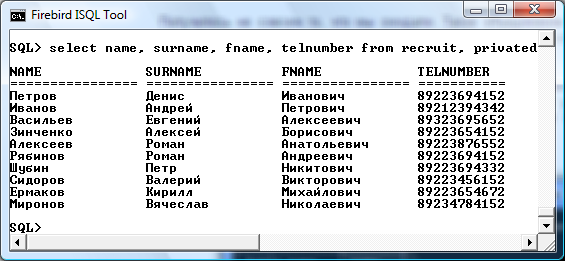
SELECT COUNT(recruit.recruit\_id) AS ALL\_REC\_HAS\_CHILD, SUM(privatedata.numchild) AS SUM\_CHILD FROM recruit, privatedata WHERE recruit.recruit\_id=privatedata.pd\_id and privatedata.numchild>0;



**е)** **Сделать выборку данных из связанных таблиц (не менее двух примеров):**

**Пример 1.** Выведем ФИО и телефонные номера:

SELECT name, surname, fname, telnumber FROM recruit, privatedata WHERE recruit.recruit\_id=privatedata.pd\_id;



**Пример 2.** Выведем ФИО военнослужащих и их звания:

SELECT recruit.name,

recruit.surname,

recruit.fname,

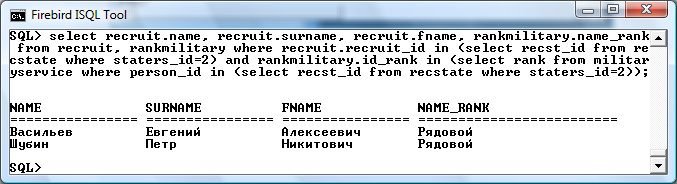
rankmilitary.name\_rank

FROM recruit,

rankmilitary

WHERE recruit.recruit\_id IN (SELECT recst\_id FROM recstate WHERE staters\_id=2) AND

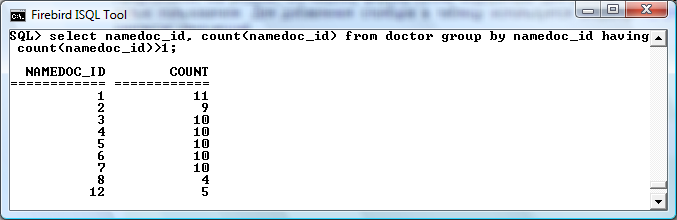
rankmilitary.id\_rank IN (SELECT rank FROM militaryservice WHERE person\_id IN (SELECT recst\_id FROM recstate WHERE staters\_id=2));



**ё)** **Создать запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки:**

Для использования группировки будем использовать GROUP BY, а для наложения ограничения на результат вместо WHERE будем использовать оператор HAVING. Допустим, нам требуется узнать сколько записей имеется у всех врачей в БД военкомата:

SELECT namedoc\_id, COUNT(namedoc\_id) FROM doctor GROUP BY namedoc\_id HAVING COUNT(namedoc\_id)>1;

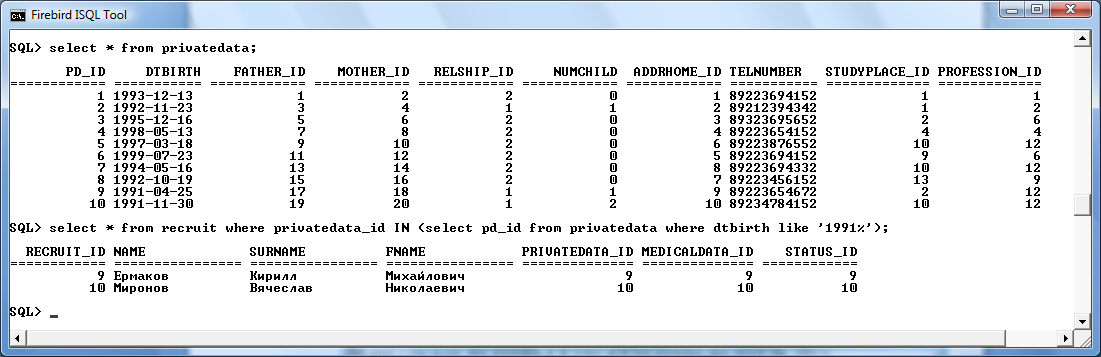


**ж)** **Придумать и реализовать пример использования вложенного запроса:**

Для реализации вложенного запроса используем логическую операцию IN. К сожалению, FireBird не понимает русских символов в логической операции LIKE, пример будем реализовывасть без использования запросов с русскими символами.

Допустим нам требует список всех призывников 1991 года рождения. Сведения о годе рожения содержаться в таблице PrivateData, на которую с помощью внещнего ключа ссылается главная таблица, в которой содержаться списки всех призывников.

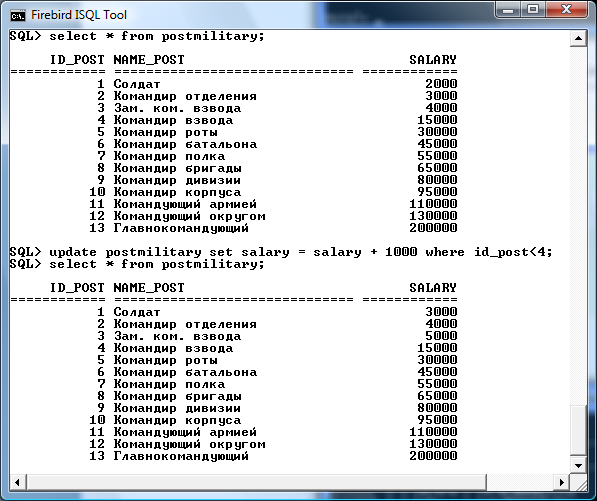
SELECT \* FROM recruit WHERE privatedata\_id IN (SELECT pd\_id FROM privatedata WHERE dtbirth LIKE '1991%');



**з) С помощью оператора INSERT добавим в каждую таблицу по одной записи:**

Данная операция многократно делается в скрипте при создании БД.

**и) С помощью оператора UPDATE изменим значения нескольких полей у всех записей, отвечающих заданному условию:**

****

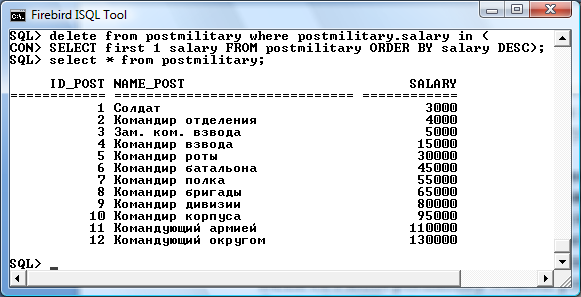
В данном примере, с помощью оператора UPDATE мы повысили зарплату всем неофицерским должностям на 1000 рублей:

UPDATE postmilitary SET salary = salary + 1000 WHERE id\_post < 4;

**к) С помощью оператора UPDATE изменим значения нескольких полей у всех записей, отвечающих заданному условию:**

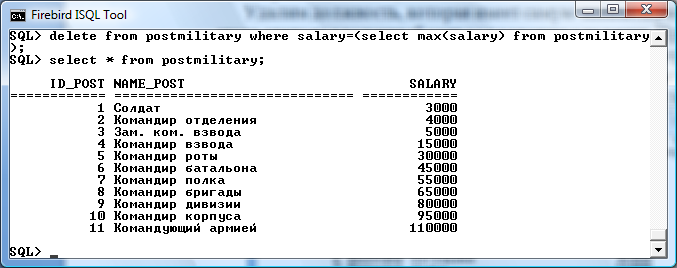
Удалим должность, которая имеет самую наивысшую зарплату. Для этого сделаем вложенный запрос к условию удаления. Просортируем по убыванию и 1 запись вложенного запроса передадим оператору DELETE:

DELETE FROM postmilitary WHERE postmilitary.salary IN (SELECT first 1 salary FROM postmilitary ORDER BY salary DESC);



Также можно было сделать проще еще так:

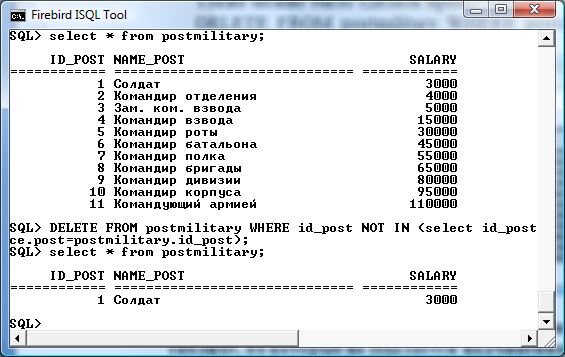
DELETE FROM postmilitary WHERE postmilitary.salary = (select max(salary) from postmilitary);



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

Итак, удалим те должности, на которых никогда не служили и не служат:

DELETE FROM postmilitary WHERE id\_post NOT IN (SELECT id\_post FROM militaryservice,postmilitary WHERE militaryservice.post=postmilitary.id\_post);



Вложенный запрос получает список номеров должностей, на которых служили и служат военнослужащие. Затем из таблицы удаляются те записи, номера которых нет во вложенном запросе.

1. **Индивидуальное задание:**

**а)** **Выведем список из десяти наиболее популярных должностей, на которые производится набор военнослужащих с количеством служащих на этих должностях за выбранный период:**

Данное задание можно просто решить с помощью оператора группировки GROUP BY и ограничения HAVING, также нужно использвовать оператор сортировки ORDER BY с ключом DESC:

SELECT FIRST 5 postmilitary.name\_post,

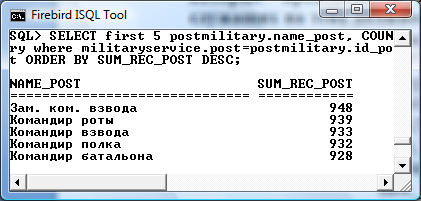
COUNT(militaryservice.post) AS sum\_rec\_post FROM militaryservice, postmilitary

WHERE militaryservice.post=postmilitary.id\_post AND date\_from > '02.02.2012' AND

date\_to < '02.02.2020'

GROUP BY name\_post

ORDER BY sum\_rec\_post DESC;



**б) Выведем пять должностей, объем выплат служащим по которым был наибольшим с суммарным объемом выплат по каждой должности (выплаты в месяц):**

Данное задание будем решать при помощи оператора группировки GROUP BY и сортировки ORDER BY с ключом DESC:

SELECT FIRST 5 postmilitary.name\_post,

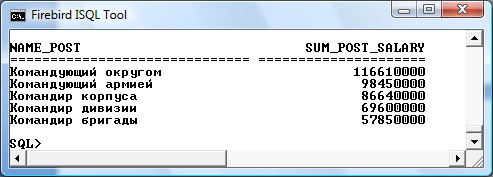
SUM(postmilitary.salary) AS sum\_ post\_salary FROM militaryservice, postmilitary

WHERE militaryservice.post=postmilitary.id\_post AND date\_from > '02.02.2012' AND

date\_to < '02.02.2020'

GROUP BY name\_post\_salary

ORDER BY sum\_ post DESC;

****

Выполним проверку - выведем результат пункта а) в полном объеме:

SELECT postmilitary.name\_post,

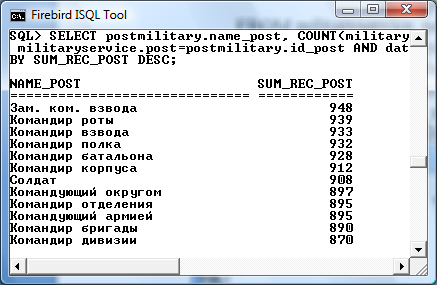
COUNT(militaryservice.post)

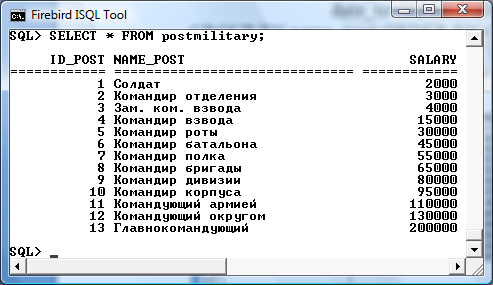
FROM militaryservice, postmilitary

WHERE militaryservice.post=postmilitary.id\_post AND date\_from>'02.02.2012' AND

date\_to<'02.02.2020'

GROUP BY name\_post ORDER BY COUNT DESC;

****

****

Теперь, умножая, например зарплату на количество людей, служащих на данной должности, получаем:

Командующий округом: 130000\*897=116610000 рублей/мес;

Все верно.

**в) Удалим должность, по которым не производился призыв военнослужащих:**

Для иммитации данной ситации, у нас не занята должность «главнокомандующего», затем, подобно пункту л) основной программы работы произведем ее удаление:

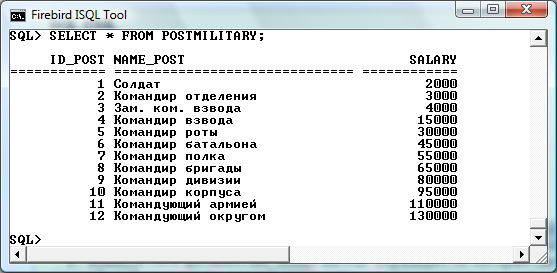
DELETE FROM postmilitary

WHERE id\_post NOT IN

(SELECT id\_post

FROM militaryservice,postmilitary

WHERE militaryservice.post=postmilitary.id\_post);



Сравнивая скрин выше и последний скрин предыдущего пункта, можно смело сказать, что запрос выполнен верно.

1. **Вывод:**

Выполнив лабораторную работу №3, были получены практические навыки программирования и модификации базы данных на языке SQL-DML. В данной работе были изучены инструменты выборки данных оператором SELECT, удаления оператором DELETE и добавления – INSERT и модификации записей - UPDATE. Оператор SELECT имеет большие возможности по созданию запросов различного характера, а именно внутреннее и внешнее объединение таблиц, группировки и агрегатные функции, вложенные запросы, представления, а также операторы условий. Однако стоит отметить, использование операторов DML очень ресурсоемко для сервера БД, что пагубно может сказаться при большом потоке сложных запросов пользователей. Поэтому, для этих целей лучше всего использовать хранимые процедуры, о чем пойдет речь в следующей лабораторной работе.

Цель работы была достигнута, полученные знания будут использованы в дальнейшем на практике при работе с БД и при проектировании более сложных информационных систем обеспечения управления.

**Приложение №1.** ER-диаграмма, полученная в IBExpert.

