

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

Отчет по лабораторной работе №4
По дисциплине «Базы данных»
«Язык SQL-DML»

Работу выполнил студент группы №43501/1

Пузанов А.В. _____

Работу принял преподаватель _____

Мясов А.В. _____

Санкт-Петербург

2016

1. Цель работы:

Ознакомление с основами проектирования схемы БД и способами нормализации отношений.

2. 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 изменить 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.

3. Выполнение работы:

- а) Произведем выборку всех данных из каждой таблицы с помощью оператора выбора SELECT:

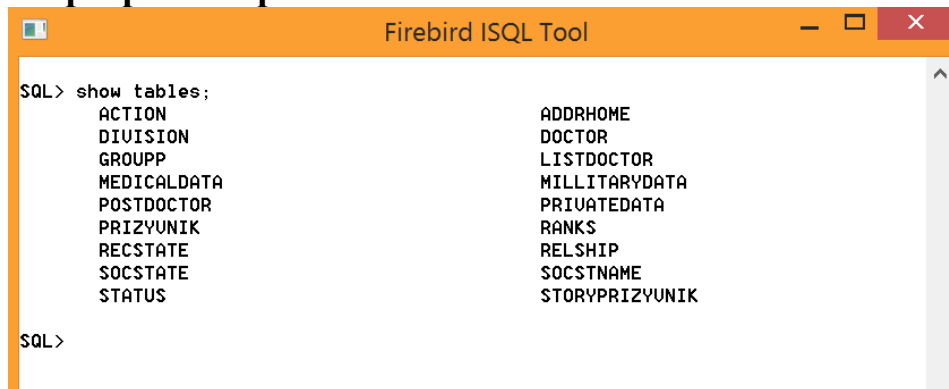
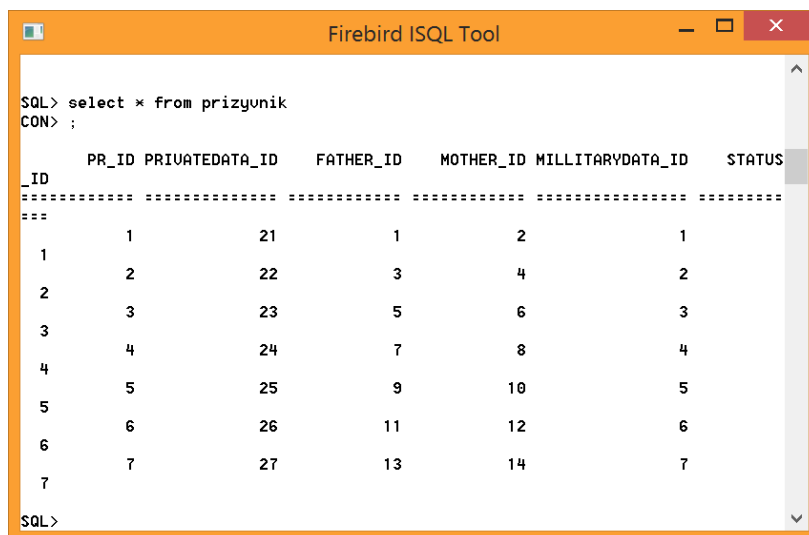


Рис.3.1. Список всех имеющихся таблиц.



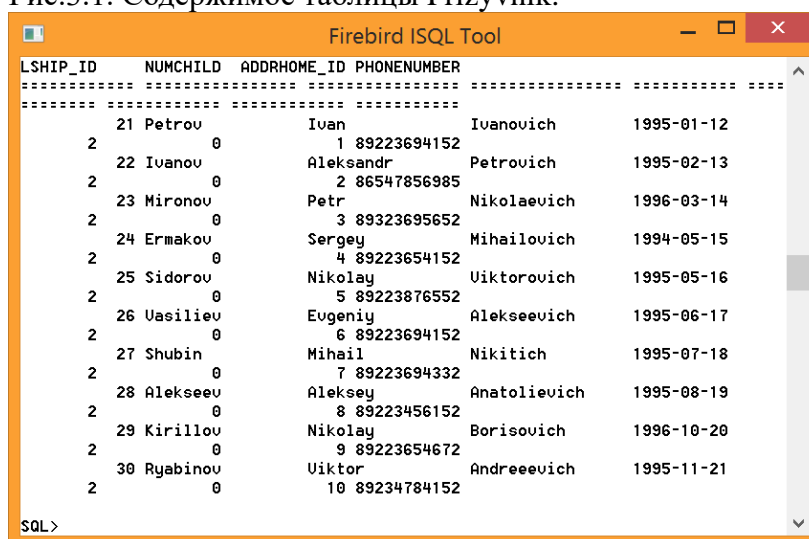
Firebird ISQL Tool

```
SQL> select * from prizyvnik
CON> ;
```

_ID	PR_ID	PRIVATEDATA_ID	FATHER_ID	MOTHER_ID	MILLITARYDATA_ID	STATUS
1	1	21	1	2	1	
2	2	22	3	4	2	
3	3	23	5	6	3	
4	4	24	7	8	4	
5	5	25	9	10	5	
6	6	26	11	12	6	
7	7	27	13	14	7	

SQL>

Рис.3.1. Содержимое таблицы Prizyvnik.

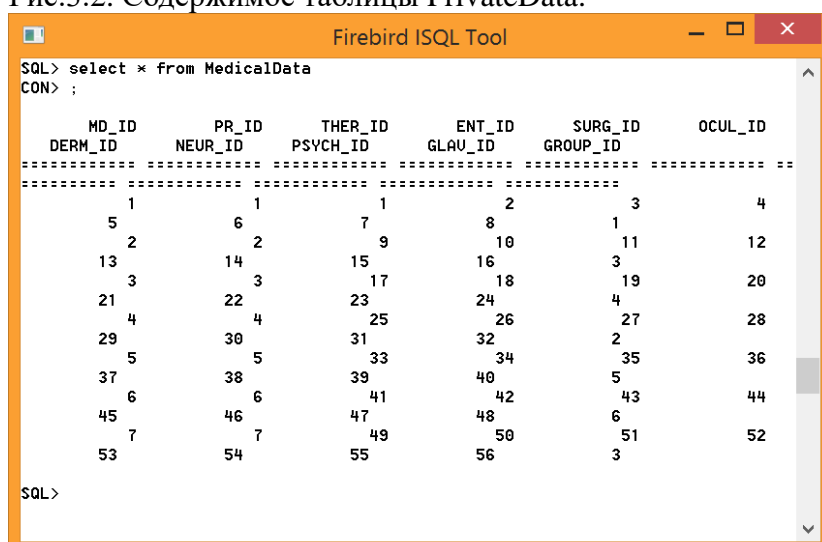


Firebird ISQL Tool

LSHIP_ID	NUMCHILD	ADDRHOME_ID	PHONENUMBER		
2	21	Petrov	Ivan	Ivanovich	1995-01-12
2	22	Ivanov	Aleksandr	Petrovich	1995-02-13
2	23	Mironov	Petr	Nikolaevich	1996-03-14
2	24	Ermakov	Sergey	Mihailovich	1994-05-15
2	25	Sidorov	Nikolay	Uiktorovich	1995-05-16
2	26	Vasiliev	Eugeny	Alekseevich	1995-06-17
2	27	Shubin	Mihail	Nikitich	1995-07-18
2	28	Alekseev	Aleksey	Anatolievich	1995-08-19
2	29	Kirillov	Nikolay	Borisovich	1996-10-20
2	30	Ryabinov	Viktor	Andreevich	1995-11-21

SQL>

Рис.3.2. Содержимое таблицы PrivateData.



Firebird ISQL Tool

```
SQL> select * from MedicalData
CON> ;
```

MD_ID	PR_ID	THER_ID	ENT_ID	SURG_ID	OCUL_ID
DERM_ID	NEUR_ID	PSVCH_ID	GLAU_ID	GROUP_ID	
5	1	1	2	3	4
2	6	7	8	1	
13	2	9	10	11	12
3	14	15	16	3	20
21	3	17	18	19	
4	22	23	24	4	28
29	4	25	26	27	
5	30	31	32	2	36
37	5	33	34	35	
6	38	39	40	5	44
45	6	41	42	43	
7	46	47	48	6	52
53	7	49	50	51	
	54	55	56	3	

SQL>

Рис.3.3. Содержимое таблицы MedicalData.

Firebird ISQL Tool

```
SQL> select * from status;
```

STATUS_ID	SOCST_ID	RECST_ID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

SQL>

Рис.3.4. Содержимое таблицы Status.

Firebird ISQL Tool

```
SQL> select * from StoryPrizyvnik;
```

ID_STOROV	PR_ID	REASON	
1	1	Shkola	200
2	2	Shkola	200
3	3	Shkola	200
4	3	Po bolezni	201
5	4	Shkola	201
6	5	Shkola	201
7	6	Shkola	201
8	7	Shkola	200
9	7	Po bolezni	201

SQL>

Рис.3.5. Содержимое таблицы StoryPrizyvnik.

Firebird ISQL Tool

```
SQL> select * from MilitaryData;
```

MILITARY_ID	ACTION	RANKS	MIL_DIVISION	DATE_FROM	DATE_TO
1	1	1	2	2011-07-01	2012-07-01
2	1	1	5	2010-06-01	2011-06-01
3	1	1	6	2014-05-01	2015-05-01
4	1	1	7	2014-06-01	2015-06-01
5	1	1	8	2010-04-01	2011-04-01
6	1	1	2	2009-09-01	2010-09-01
7	1	1	1	2009-03-01	2010-03-01

SQL>

Рис.3.5. Содержимое таблицы MilitaryData.

ID_DIVISION	NAME_DIVISION
1	25741
2	17645
3	98536
5	12312
6	23523
7	65454
8	23528
9	73468
10	24328

Рис.3.6. Содержимое таблицы Division.

LISTDOC_ID	FULLNAME
1	Uyatkin Igor Vladimirovich
2	Jmurkil Ivan Petrovich
3	Smirnov Aleksandr Kirrilovich
4	Igoraev Nikita Vladimirovich
5	Lukin Vladimir Aleksandrovich
6	Ivanov Andrey Georgievich
7	Jukov Aleksandr Stepanovich
8	Petrov Igor Nikolaevich
9	Ivanov Ivan Ivanovich
10	Petrov Petr Petrovich
11	Berkova Natalia Nikolaevna
12	Marishko Lydmila Petrovna

Рис.3.7. Содержимое таблицы-справочника ListDoctor.

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

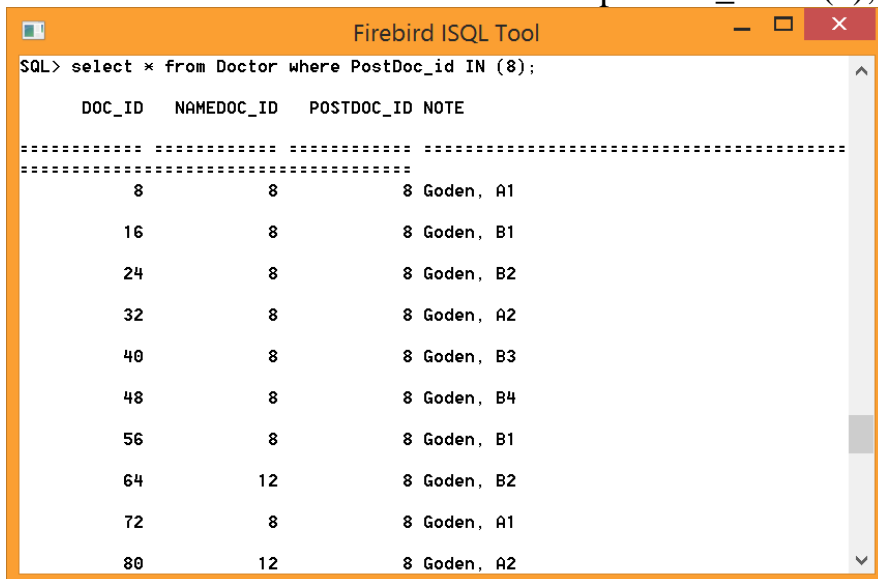
Выберем, для реализации данной задачи, таблицу Doctor, которая хранит данные о медицинских осмотрах призывников.

Пример 1. Выведем все записи терапевтов:

SELECT * FROM doctor WHERE postdoc_id IN (1);

DOC_ID	NAMEDOC_ID	POSTDOC_ID	NOTE
1	1	1	Jalob net
9	1	1	Jalob net
17	1	1	Jalob net
25	1	1	Jalob net
33	1	1	Beesonitca
41	1	1	Jalob net
49	1	1	Jalob net
57	1	1	Jalob net
65	1	1	Jalob net
73	1	1	Jalob net

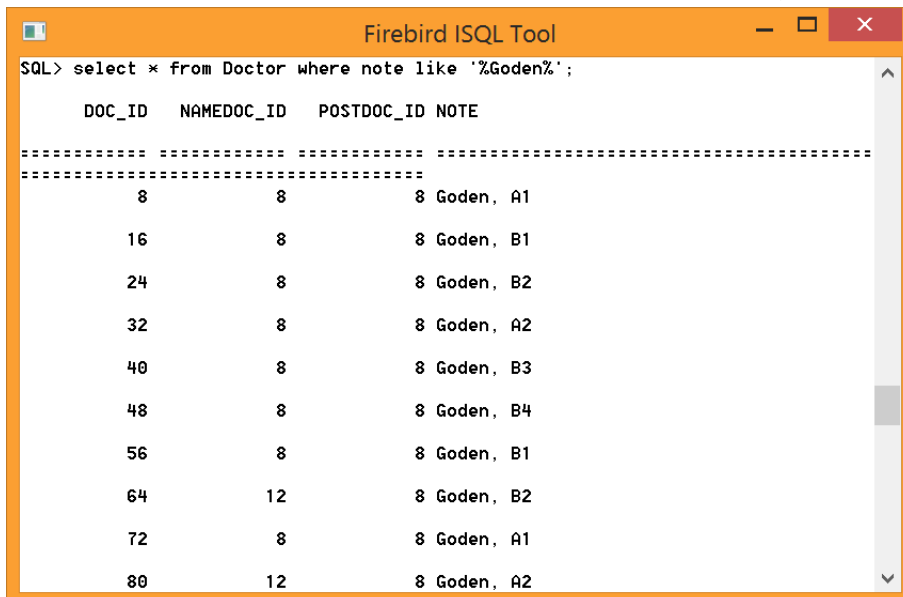
Пример 2. Выведем все записи главных врачей:
 SELECT * FROM doctor WHERE postdoc_id IN (8);



DOC_ID	NAMEDOC_ID	POSTDOC_ID	NOTE
8	8	8	Goden, A1
16	8	8	Goden, B1
24	8	8	Goden, B2
32	8	8	Goden, A2
40	8	8	Goden, B3
48	8	8	Goden, B4
56	8	8	Goden, B1
64	12	8	Goden, B2
72	8	8	Goden, A1
80	12	8	Goden, A2

Пример 3. Выведем заметки всех врачей, в которых содержатся слово Goden:

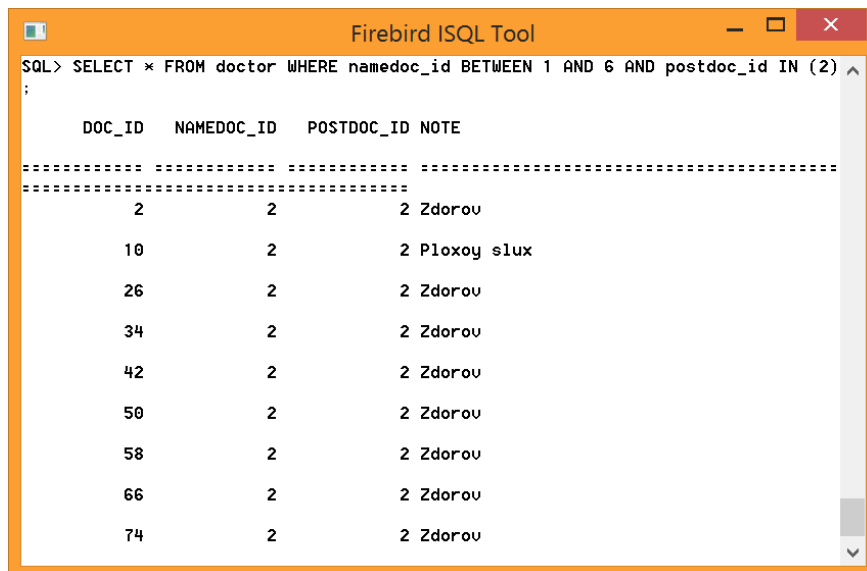
SELECT * FROM doctor WHERE note LIKE '%Goden%';



DOC_ID	NAMEDOC_ID	POSTDOC_ID	NOTE
8	8	8	Goden, A1
16	8	8	Goden, B1
24	8	8	Goden, B2
32	8	8	Goden, A2
40	8	8	Goden, B3
48	8	8	Goden, B4
56	8	8	Goden, B1
64	12	8	Goden, B2
72	8	8	Goden, A1
80	12	8	Goden, A2

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

SELECT * FROM doctor WHERE namedoc_id BETWEEN 1 AND 6 AND postdoc_id IN (2);



Firebird ISQL Tool

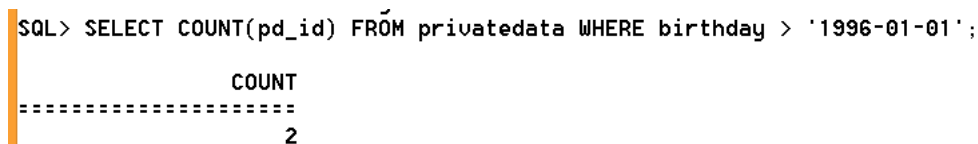
```
SQL> SELECT * FROM doctor WHERE namedoc_id BETWEEN 1 AND 6 AND postdoc_id IN (2);
```

DOC_ID	NAMEDOC_ID	POSTDOC_ID	NOTE
2	2	2	Zdorov
10	2	2	Ploxoy slux
26	2	2	Zdorov
34	2	2	Zdorov
42	2	2	Zdorov
50	2	2	Zdorov
58	2	2	Zdorov
66	2	2	Zdorov
74	2	2	Zdorov

в) Создать в запросе вычисляемое поле:

Подсчитаем количество призывников, которые должны будут пройти медкомиссию

```
SELECT COUNT(pd_id) FROM privatedata WHERE birthday > '1996-01-01'
```



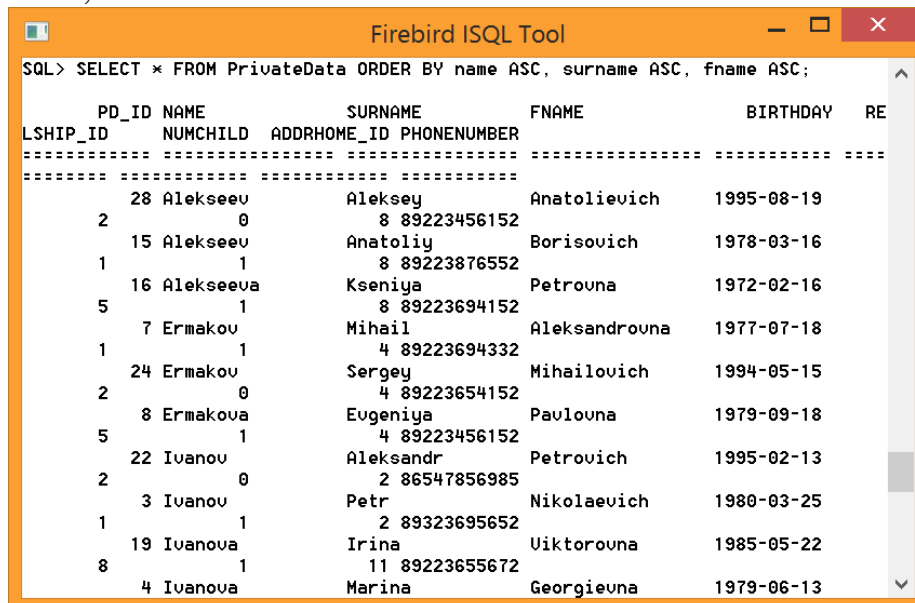
```
SQL> SELECT COUNT(pd_id) FROM privatedata WHERE birthday > '1996-01-01';
```

COUNT
2

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

Отсортируем по имени фамилии и отчеству список всех людей, числящихся в БД нашего военкомата:

```
SELECT * FROM PrivateData ORDER BY name ASC, surname ASC, fname ASC;
```



Firebird ISQL Tool

```
SQL> SELECT * FROM PrivateData ORDER BY name ASC, surname ASC, fname ASC;
```

PD_ID	NAME	SURNAME	FNAME	BIRTHDAY	RE
28	Alekseev	Aleksey	Anatolievich	1995-08-19	
2		8 89223456152			
15	Alekseev	Anatoliy	Borisovich	1978-03-16	
1		8 89223876552			
16	Alekseeva	Kseniya	Petrovna	1972-02-16	
5		8 89223694152			
7	Ermakov	Mihail	Aleksandrovna	1977-07-18	
1		4 89223694332			
24	Ermakov	Sergey	Mihailovich	1994-05-15	
2		4 89223654152			
8	Ermakova	Eugeniya	Paulovna	1979-09-18	
5		4 89223456152			
22	Ivanov	Aleksandr	Petrovich	1995-02-13	
2		2 86547856985			
3	Ivanov	Petr	Nikolaevich	1980-03-25	
1		2 89323695652			
19	Ivanova	Irina	Viktorovna	1985-05-22	
8		11 89223655672			
4	Ivanova	Marina	Georgievna	1979-06-13	

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(Prizyvnik.pr_id) AS ALL_PRIZYVNIKI_HAS_CHILD,
SUM(privatedata.numchild) AS SUM_CHILD FROM Prizyvnik, privatedata
WHERE Prizyvnik.privatedata_id = privatedata.pd_id and
privatedata.numchild>0;
```

```
SQL> SELECT COUNT(Prizyvnik.pr_id) AS ALL_PRIZYVNIKI_HAS_CHILD,
CON> SUM(privatedata.numchild) AS SUM_CHILD FROM Prizyvnik, privatedata
CON> WHERE Prizyvnik.privatedata_id = privatedata.pd_id and privatedata.numchild
>0;
```

```
ALL_PRIZYVNIKI_HAS_CHILD      SUM_CHILD
=====
0                             <null>
```

```
SQL>
```

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

Пример 1. Выведем ФИО и телефонные номера призывников:

```
SELECT name, surname, fname, phonenumber FROM Prizyvnik,
privatedata WHERE Prizyvnik.privatedata_id=privatedata.pd_id;
```

```
SQL> SELECT name, surname, fname, phonenumber FROM Prizyvnik, privatedata WHERE
Prizyvnik.privatedata_id=privatedata.pd_id;
```

NAME	SURNAME	FNAME	PHONENUMBER
Petrov	Ivan	Ivanovich	89223694152
Ivanov	Aleksandr	Petrovich	86547856985
Mironov	Petr	Nikolaevich	89323695652
Ermakov	Sergey	Mihailovich	89223654152
Sidorov	Nikolay	Viktorovich	89223876552
Uasiliev	Eugeni	Alekseevich	89223694152
Shubin	Mihail	Nikitich	89223694332

Пример 2. Выведем ФИО военнослужащих и их звания:

```
SELECT privatedata.name, privatedata.surname, privatedata.fname,
ranks.name_rank from prizyvnik,status,privatedata,militarydata,ranks where
privatedata.pd_id=prizyvnik.privatedata_id
AND prizyvnik.status_id=status.status_id
AND status.status_id IN(SELECT status.status_id FROM status,recstate
WHERE status.recst_id = recstate.recst_id AND recstate.staters LIKE
'%Slujit%')
AND prizyvnik.militarydata_id=militarydata.military_id
AND militarydata.military_id IN (SELECT militarydata.military_id from
militarydata,ranks where ranks.id_rank=militarydata.ranks)
AND ranks.name_rank IN (SELECT ranks.name_rank from militarydata,ranks
where ranks.id_rank=militarydata.ranks);
```


NAME	SURNAME	FNAME	NAME_RANK
Mironov	Petr	Nikolaevich	Ryadovoi
Shubin	Mihail	Nikitich	Ryadovoi

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;
```

```
SQL> SELECT namedoc_id, COUNT(namedoc_id) FROM doctor GROUP BY namedoc_id HAVING
COUNT(namedoc_id)>1;
```

NAMEDOC_ID	COUNT
1	10
2	9
3	10
4	10
5	10
6	10
7	10
8	8
12	2

ж) Придумать и реализовать пример использования вложенного запроса:

Для реализации вложенного запроса используем логическую операцию IN. Допустим нам требует список всех призывников 1996 года рождения. Сведения о годе рождения содержатся в таблице PrivateData, на которую с помощью внешнего ключа ссылается главная таблица, в которой содержатся списки всех призывников.

```
SELECT * FROM Prizyvnik WHERE privatedata_id IN (SELECT pd_id
FROM privatedata WHERE Birthday LIKE '1996%');
```

```
SQL> SELECT * FROM Prizyvnik WHERE privatedata_id IN (SELECT pd_id FROM privated
ata WHERE Birthday LIKE '1996%');
```

PR_ID	PRIVATEDATA_ID	FATHER_ID	MOTHER_ID	MILLITARYDATA_ID	STATUS
3	23	5	6	3	

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:

```
SQL> SELECT privatedata.fname,name_action FROM privatedata,action, militarydata
,prizyunik where militarydata.action=action.id_action
CON> AND privatedata.pd_id=prizyunik.privatedata_id AND militarydata.military
_id=prizyunik.militarydata_id;
```

FNAME	NAME_ACTION
Petrov	Prizyv
Mironov	Prizyv
Ermakov	Prizyv
Sidorov	Prizyv
Vasiliev	Prizyv
Shubin	Prizyv

В данном примере, с помощью оператора UPDATE мы переведем всех призывников на контрактную службу

```
UPDATE militarydata SET action = 2 WHERE action < 2;
```

FNAME	NAME_ACTION
Petrov	Kontrakt
Mironov	Kontrakt
Ermakov	Kontrakt
Sidorov	Kontrakt
Vasiliev	Kontrakt
Shubin	Kontrakt

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

Удалим данные о самом молодом призывнике.

```
DELETE FROM privatedata where birthday = (select max(birthday) from
privatedata);
```

```
SQL> select max(birthday) from privatedata;
```

```
MAX
=====
1996-10-20
```

21 Petrov	Ivan	Ivanovich	1995-01-12
2 0	1 89223694152	1	21
1 2	1	1	
23 Mironov	Petr	Nikolaevich	1996-03-14
2 0	3 89323695652	3	23
5 6	3	3	
24 Ermakov	Sergey	Mihailovich	1994-05-15
2 0	4 89223654152	4	24
7 8	4	4	
25 Sidorov	Nikolay	Uiktorovich	1995-05-16
2 0	5 89223876552	5	25
9 10	5	5	
26 Vasiliev	Eugeny	Alekseevich	1995-06-17
2 0	6 89223694152	6	26
11 12	6	6	
27 Shubin	Mihail	Nikitich	1995-07-18
2 0	7 89223694332	7	27
13 14	7	7	

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

Итак, удалим те семейные положения на которых никогда не ссылались.

```
DELETE FROM relship WHERE relship.relship_id NOT IN (SELECT
privatedata.relship_id FROM privatedata,relship WHERE
privatedata.relship_id=relship.relship_id);
```

```
SQL> DELETE FROM relship
CON> WHERE relship.relship_id NOT IN (SELECT privatedata.relship_id FROM private
data,relship
CON> WHERE privatedata.relship_id=relship.relship_id);
SQL> select * from relship;
```

```
RELSHIP_ID RELSHIPSTATUS
=====
1 Jenat
2 Xolost
4 Udovec
5 Zamujem
8 Udova
```

4. Индивидуальное задание:

а) Вывести призывников, у которых было более одной отсрочки, одна из которых действующая

```
SELECT
prizyvnik.pr_id,privatedata.fname,privatedata.name,privatedata.surname,storyprizyvnik.dateto from privatedata,prizyvnik,storyprizyvnik
where privatedata.pd_id=prizyvnik.privatedata_id AND
prizyvnik.pr_id=storyprizyvnik.pr_id AND
prizyvnik.pr_id IN(SELECT prizyvnik.pr_id from prizyvnik,storyprizyvnik
where storyprizyvnik.pr_id = prizyvnik.pr_id
group by prizyvnik.pr_id
having count(*)>1)
and storyprizyvnik.dateto > (SELECT CAST('NOW' AS DATE) FROM
rdb$database);
```

```
PR_ID FNAME NAME SURNAME DATETO
=====
4 Ermakov Sergey Mihailovich 2017-09-01
5 Sidorov Nikolay Viktorovich 2017-09-01
```

б) Вывести количество призывников, у которых была снижена группа здоровья после повторной медкомиссии.

```
CREATE VIEW medkomissiya1 AS
SELECT m2.pr_id, m2.group_id, m2.medical_date
from medicaldata as m1, medicaldata as m2
where m1.pr_id = m2.pr_id
AND m1.medical_date < m2.medical_date
AND m1.group_id < m2.group_id
order by m2.pr_id, m2.medical_date;
```

```
SELECT COUNT(*) FROM medkomissiya1;
```

```
SQL> SELECT COUNT(*) FROM medkomissiya1;
```

```
COUNT
=====
2
```

Проверим:

MD_ID OCUL_ID	PR_ID DERM_ID	MEDICAL_DATE NEUR_ID	THER_ID PSYCH_ID	ENT_ID GLAU_ID	SURG_ID GROUP_ID
1	1	2012-10-10	1	2	3
4	5	6	7	8	1
2	2	2013-10-10	9	10	11
12	13	14	15	16	3
3	3	2012-09-15	17	18	19
20	21	22	23	24	4
4	4	2014-10-10	25	26	27
28	29	30	31	32	2
5	5	2014-09-12	33	34	35
36	37	38	39	40	5
6	6	2008-10-10	41	42	43
44	45	46	47	48	6
7	7	2008-10-23	49	50	51
52	53	54	55	56	3
8	1	2013-09-27	57	58	59
60	61	62	63	64	4
9	2	2014-11-05	65	66	67
68	69	70	71	72	5

в) Вывести количество призывников по социальным статусам за заданный период

```
SELECT socstname.statename, count(socstate.statess_id) from socstate,socstname
where statess_id = socstname_id
and((socstate.datefrom <= '01.01.2014' and socstate.dateto >= '01.01.2017')
or ( socstate.datefrom between '01.01.2014' and '01.01.2017')
or ( socstate.dateto between '01.01.2014' and '01.01.2017'))
group by socstname.statename;
```

STATENAME	COUNT
Bezrabortnyiy	1
Prizyvnik-srochnik	2
Student	4

Проверим:

SOCST_ID	STATESS_ID	DATEFROM	DATETO
1	2	2014-09-01	2018-09-01
2	2	2014-09-01	2018-09-01
3	7	2016-05-01	2017-05-01
4	4	2013-09-01	2017-09-01
5	2	2013-09-01	2017-09-01
6	2	2014-09-01	2018-09-01
7	7	2016-06-01	2017-06-01

5. Сохранение выполненных запросов в виде хранимых процедур

Insert

```
SET TERM ^;  
Create procedure insert_value ( name_rank VARCHAR (50))  
as  
begin  
    INSERT INTO Ranks VALUES ( :name_rank );  
end  
^  
SET TERM ; ^
```

Update

```
Create procedure update_milli (action int)  
as begin  
    UPDATE millitarydata  
    SET action = 2  
    WHERE action < 2;  
end  
^
```

Delete

```
Create procedure del_unused  
as begin  
    DELETE FROM relship  
    WHERE relship.relship_id NOT IN (SELECT privatedata.relship_id FROM  
privatedata,relship  
    WHERE privatedata.relship_id=relship.relship_id);  
END  
^
```

6. Выводы:

Выполнив лабораторную работу №4, 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. 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.