Санкт-Петербургский государственный политехнический университет

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №2**

**Дисциплина**: Базы данных

# SQL-DDL.

Выполнил студент гр. 43501/3 Муравьев Ф.Э.

Преподаватель: Мяснов А. В.

Санкт-Петербург

2015

## Цели работы

Познакомить студентов с основами проектирования схемы БД, языком описания сущностей и ограничений БД SQL-DDL.

## Программа работы

1. Самостоятельное изучение SQL-DDL
2. Создание скрипта БД в соответствии с согласованной схемой (должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений). Продемонстрировать скрипт преподавателю.
3. Создайте скрипт, заполняющий все таблицы БД данными
4. Выполнение SQL-запросов, изменяющих схему созданной БД **по заданию преподавателя**. Продемонстрировать их работу преподавателю.
5. Изучите основные возможности IBExpert. Получите ER-диаграмму созданной БД с помощью **Database Designer**.
6. Автоматически сгенерируйте данные при помощи IBExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

**3. Выполнение работы**

**Язык SQL-DDL**

Скрипт создания БД представлен в листинге 1.

**Листинг 1.**

create database ‘locallhost:C:\DataBase\bank.fdb’ user 'SYSDBA' password 'masterkey';

commit;

connect ‘locallhost:C:\DataBase\bank.fdb’ user 'SYSDBA' password 'masterkey';

create table repository

(

repository\_ID integer not null primary key,

adress varchar(100) not null,

director varchar(32) not null

);

create table cells

(

cell\_number integer not null primary key,

repository\_ID integer references repository(repository\_ID)

);

create table clients

(

client\_ID integer not null primary key,

name varchar(50),

adress varchar(100),

born\_date date

);

create table groups

(

group\_ID integer not null primary key,

date\_create date

);

create table contracts

(

contract\_ID integer not null primary key,

start\_date date,

end\_date date,

arend\_cost integer,

cell\_number integer references cells(cell\_number)

client\_ID integer references clients(client\_ID)

group\_ID integer references groups(group\_ID)

);

create table groups\_clients

(

ID integer not null primary key,

client\_ID integer references clients(client\_ID),

group\_ID integer references groups(group\_ID)

);

create table items

(

item\_ID integer not null primary key,

contract\_ID integer references contracts(contract\_ID),

start\_date date,

end\_date date,

name varchar(50),

cost integer

);

alter table cells

add status varchar(15)

**Создадим скрипт, заполняющий все таблицы БД данными:**

**Листинг 2.**

insert into clients values (1, 'Sokolov Vasya Aleksandrovich', 'SPb, s.Borovaya, h.5, a.30', '1989-01-01');

insert into clients values (2, 'Polyakov Vasya Petrovich', 'SPb, s.Borovaya, h.10, a.68', '1984-12-03');

insert into clients values (3, 'Medvedev Dima Vasilyevich', 'SPb, s.Griboedov, h.12, a.12', '1990-11-03');

insert into clients values (4, 'Strelcov Nikolai Vladimirovich', 'SPb, s.Pyshkinskay, h.56, a.1', '1985-11-03');

insert into clients values (5, 'Prohorov Artem Denisovich', 'SPb, s.Griboedov, h.45, a.68', '1989-06-30');

insert into clients values (6, 'Gromov Airat Dmitrievich', 'SPb, s.Razejaya, h.189, a.6', '1976-07-21');

insert into clients values (7, 'Kiselev Pavel Rodionovich', 'SPb, s.Kolomenskaya, h.20, a.45', '1963-01-11');

insert into clients values (8, 'Kolobov Jhon Ibragimovich', 'SPb, s.Nevskii, h.127, a.43', '1965-05-15');

insert into groups values (1, '2012-05-15');

insert into groups values (2, '2014-01-01');

insert into groups values (3, '2014-01-01');

insert into groups values (4, '2014-01-01');

insert into groups values (5, '2014-01-01');

insert into groups values (6, '2014-01-01');

insert into groups values (7, '2014-01-01');

insert into groups values (8, '2014-01-01');

insert into groups values (9, '2014-01-01');

insert into groups values (10, '2014-01-01');

insert into groups\_clients values (1, 1, 1);

insert into groups\_clients values (2, 2, 1);

insert into groups\_clients values (3, 3, 1);

insert into groups\_clients values (4, 5, 2);

insert into groups\_clients values (5, 8, 2);

insert into groups\_clients values (6, 1, 3);

insert into groups\_clients values (7, 2, 4);

insert into groups\_clients values (8, 3, 5);

insert into groups\_clients values (9, 4, 6);

insert into groups\_clients values (10, 5, 7);

insert into groups\_clients values (11, 6, 8);

insert into groups\_clients values (12, 7, 9);

insert into groups\_clients values (13, 8, 10);

insert into repository values (1, 'SPb, s.Kolomenskaya, h.123', 'Bankov Bank Bankovich');

insert into repository values (2, 'SPb, s.Komendatskii, h.12', 'Direktorov Director Directovich');

insert into cells values (13, 1, 'arrend');

insert into cells values (1, 2, 'arrend');

insert into cells values (6, 1, 'arrend');

insert into cells values (8, 2, 'arrend');

insert into cells values (11, 1, 'arrend');

insert into cells values (21, 2, 'arrend');

insert into cells values (40, 1, 'arrend');

insert into cells values (50, 2, 'arrend');

insert into cells values (4, 2, 'free');

insert into cells values (25, 2, 'free');

insert into contracts values (1, '2014-02-01', '2015-02-01', 100, 13, 1, 3);

insert into contracts values (2, '2014-02-01', '2016-02-01', 200, 1, 2, 1);

insert into contracts values (3, '2014-02-01', '2016-06-01', 250, 6, 5, 2);

insert into contracts values (4, '2014-02-01', '2016-06-01', 250, 8, 5, 2);

insert into contracts values (5, '2014-02-01', '2015-06-01', 150, 11, 4, 6);

insert into contracts values (6, '2014-02-01', '2015-06-01', 150, 21, 8, 10);

insert into contracts values (7, '2015-02-01', '2015-06-01', 40, 40, 5, 7);

insert into contracts values (8, '2015-02-01', '2015-08-01', 70, 40, 7, 9);

insert into items values (1, 1, '2014-01-10', '2015-02-01', 'Brilliants', 100000);

insert into items values (2, 1, '2014-01-10', '2015-02-01', 'Ruby', 1000);

insert into items values (3, 2, '2014-01-10', '2015-02-01', 'Gold', 10000);

insert into items values (4, 3, '2014-01-10', '2015-03-20', 'Platina', 20000);

insert into items values (5, 7, '2015-01-02', '2015-06-01', 'Gold egg', 25000);

insert into items values (6, 8, '2015-01-02', '2015-08-01', 'Redstone', 1000);

Проверим данные с помощью IBExpert.

|  |
| --- |
| Items |
|  |
| Contracts |
|  |
| Clients |
|  |
| Groups\_Clients |
|  |
| Groups |
|  |
| Repository |
|  |
| Cells |
|  |

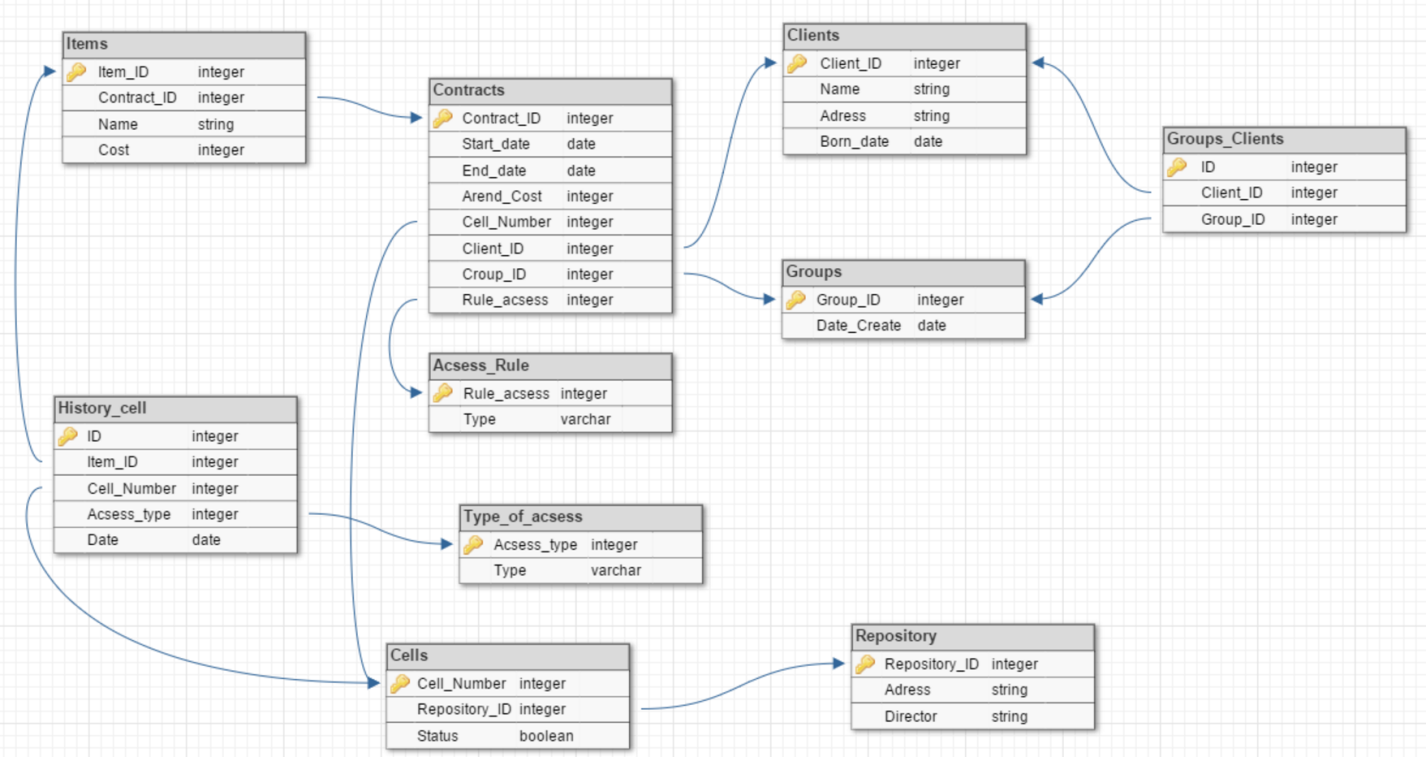
Выполнение SQL-запросов, изменяющих схему созданной БД **по заданию преподавателя**.

Задание:

1. Добавить учет правил доступа к ячейкам.
2. Добавить учет фактического доступа к ячейкам с хранением информации об изъятых и добавленных ценностях.

Удалим из таблицы Items поля Start\_date и End\_date. Добавим таблицу History\_cell для учета фактического доступа к ячейкам. Таблица History\_cell содержит следующие поля: ID, Cell\_number, Acsess\_type, Date, Item\_ID. Для учета правил доступа к ячейкам в таблицу Contracts добавим поле Rule\_acsess. Таблицы Type\_of\_acsess и Acsess\_Rule являются справочниками типов доступа.

Полученная схема БД:



Изменим схему созданной БД, выполняя SQL-запросы:

connect ‘locallhost:C:\DataBase\bank.fdb’ user 'SYSDBA' password 'masterkey';

alter table items drop start\_date;

alter table items drop end\_date;

create table type\_of\_acsess

(

acsess\_type integer primary key,

type\_a varchar(50)

);

create table history\_cell

(

ID integer not null primary key,

cell\_number integer references cells(cell\_number),

acsess\_type integer references type\_of\_acsess(acsess\_type),

date\_acsess date,

item\_ID integer references items(item\_ID)

);

create table acsess\_rule

(

rule\_acsess integer primary key,

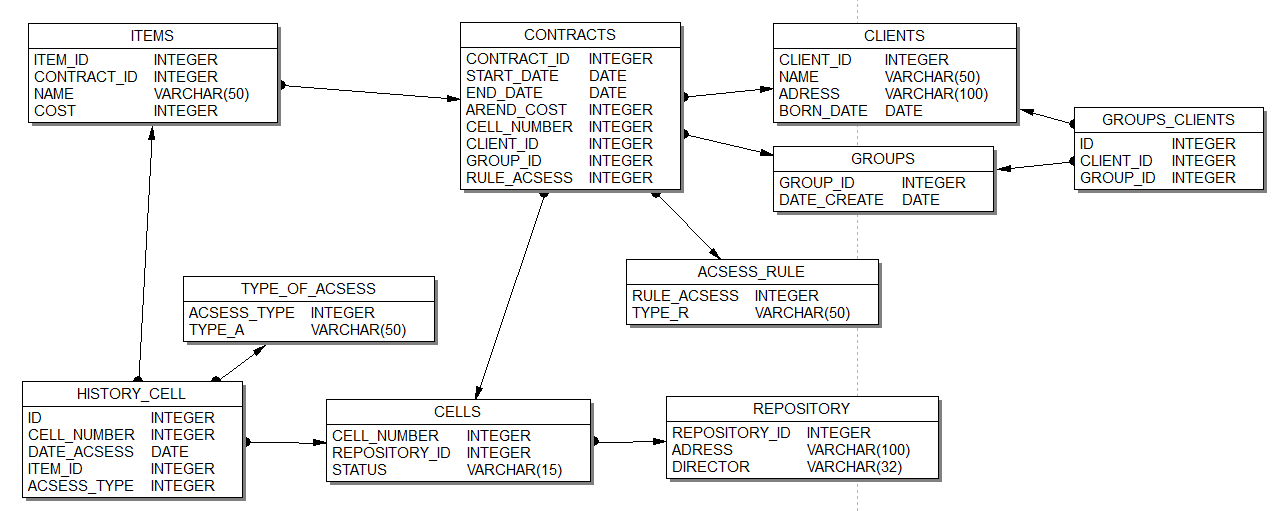
type\_r varchar(50)

);

alter table contracts

add rule\_acsess integer references acsess\_rule(rule\_acsess);

Получим ER-диаграмму созданной БД с помощью **Database Designer**.



Автоматически сгенерируем данные при помощи IBExpert (для трех таблиц, не менее 100000 записей в каждой из выбранных таблиц).

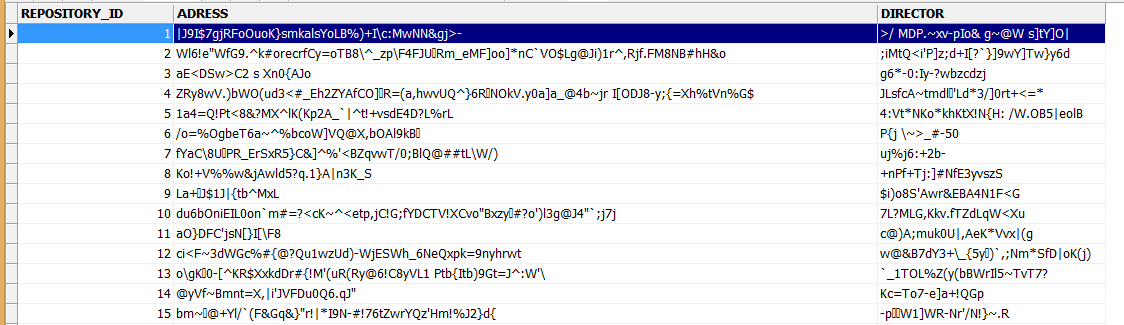
1)**Для таблицы REPOSITORY:**

Запускаем test data generator:

|  |
| --- |
| Repository\_ID |
|  |
| Adress |
|  |
| Director |
|  |

Полученные результаты:

В таблице Repository появилось 100000 записей:



…



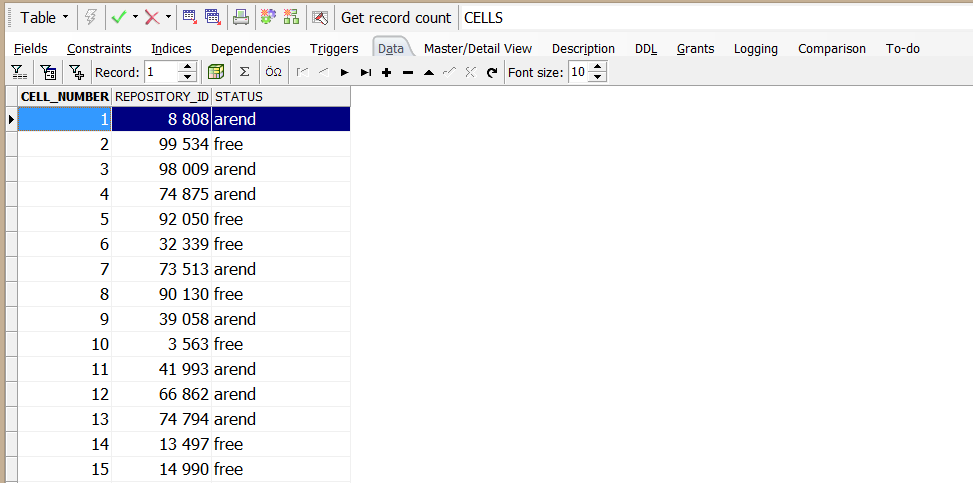
2)**Для таблицы CELLS:**

Запускаем test data generator:

|  |
| --- |
| Cell\_number |
|  |
| Repository\_ID |
|  |
| Status |
|  |

Полученные результаты:

В таблице CELLS появилось 100000 записей:



…



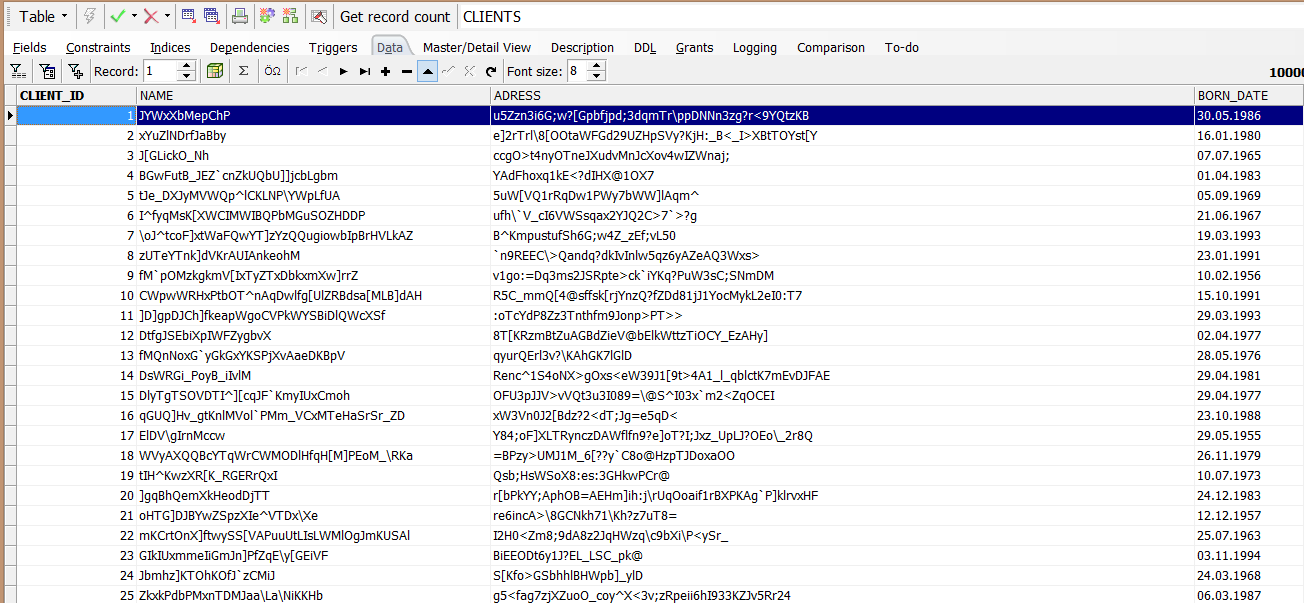
3)**Для таблицы Clients:**

Запускаем test data generator:

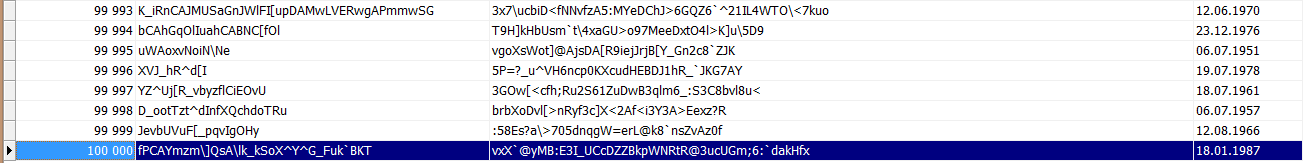
|  |
| --- |
| Client\_ID |
|  |
| Name |
|  |
| Adress |
|  |
| Born\_date |
|  |

Полученные результаты:

В таблице Clients появилось 100000 записей:



…



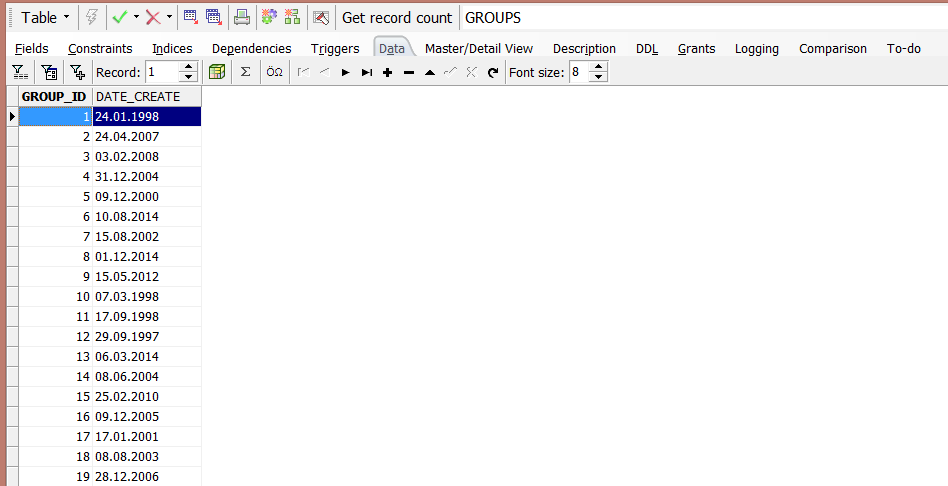
3)**Для таблицы Groups:**

Запускаем test data generator:

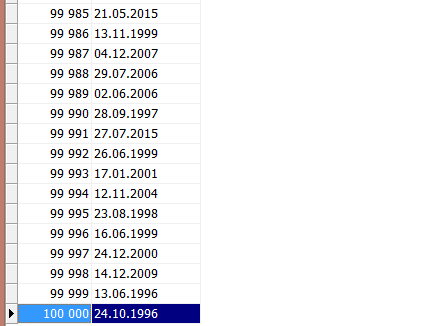
|  |
| --- |
| Group\_ID |
|  |
| Date\_Create |
|  |

Полученные результаты:

В таблице Groups появилось 100000 записей:



…



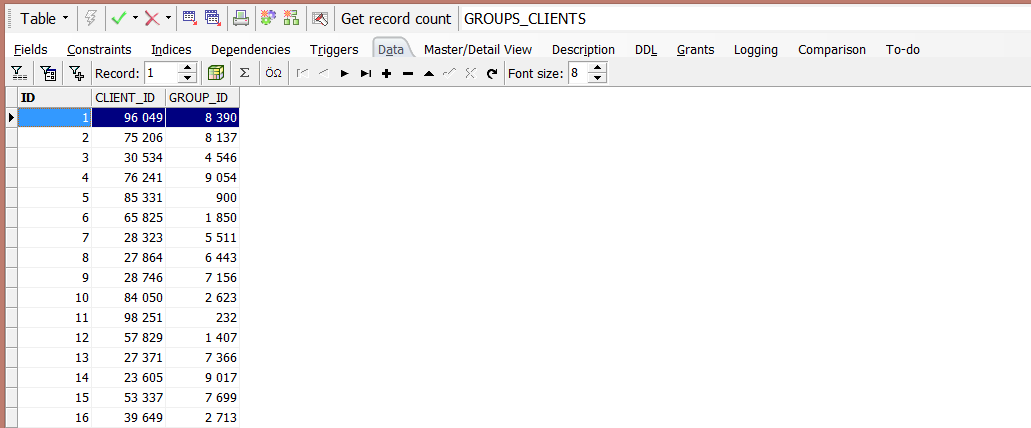
4)**Для таблицы Groups\_Clients:**

Запускаем test data generator:

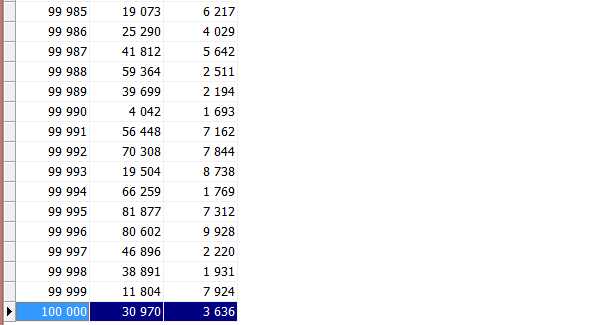
|  |
| --- |
| ID |
|  |
| Client\_ID |
|  |
| Group\_ID |
|  |

Полученные результаты:

В таблице Groups\_Clients появилось 100000 записей:



…



**Выводы:**

В ходе работы были получены навыки проектирования баз данных и изучен язык SQL-DDL. Были написаны скрипты для создания, заполнения и модификации БД. SQL-DDL (Data Definition Language) – язык определения структур и ограничений целостности баз данных. Использовались такие инструменты языка DDL как <create> (создать), <alter> (изменить), <drop> (удалить). Основой языка SQL является язык реляционных БД. Язык SQL позволяет управлять структурой БД, это обеспечивает гибкость БД к изменяющимся требованиям предметной области. Также база была заполнена некоторым количеством данных с помощью конструкции INSERT. При проектировании базы все таблицы были приведены к третьей нормальной форме. Более высокий уровень нормализации уменьшает избыточность данных, но усложняет ее структуру и замедляет поиск. Также структура базы данных становится менее понятной.

Были получены практические навыки в создании, изменении и заполнении базы данных.

Также был изучен программный продукт для работы с БД IBExpert. Он позволяет выполнять широкий набор операций с базами данных.