Санкт-Петербургский политехнический университет им. Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №3**

**Дисциплина**: Базы данных

# SQL-DDL.

Выполнил студент гр. 43501/3 Анисимов А.А.

Преподаватель: Мяснов А. В.

Санкт-Петербург

2016

## 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 'D:\datbase\bank.fdb' user 'SYSDBA' password 'masterkey';  commit;  CONNECT 'D:\Datbase\bank.fdb' user 'SYSDBA' password 'masterkey';  CREATE TABLE access\_type (  accesstype\_id int not null primary key,  accesstype\_type varchar(128) not null unique);    commit;  CREATE TABLE groups (group\_id int not null primary key);  commit;  CREATE TABLE client (  client\_id int not null primary key,  client\_name varchar(128) not null,  client\_lastname varchar(128) not null,  client\_dob timestamp not null,  client\_passport int not null unique);  commit;  CREATE TABLE deposit\_box (  depb\_id int not null primary key,  depb\_num int not null unique,  depb\_value float);  commit;  CREATE TABLE group\_member (  gm\_id int not null primary key,  group\_id int references groups,  client\_id int references client);  commit;  CREATE TABLE contract (  contract\_id int not null primary key,  contract\_start timestamp not null,  contract\_end timestamp not null,  contract\_cost float not null,  depb\_id int references deposit\_box,  accesstype\_id int references access\_type,  group\_id int references groups);  commit;  CREATE TABLE box\_history (  history\_id int not null primary key,  client\_id int references client,  history\_previousValue float,  history\_currentValue float,  depb\_id int references deposit\_box,  contract\_id int references contract);  commit;  SHOW TABLES; |

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

**Листинг 2.**

|  |
| --- |
| CONNECT 'D:\Datbase\bank.fdb' user 'SYSDBA' password 'masterkey';  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO client (client\_id, client\_name, client\_lastname, client\_dob, client\_passport) VALUES (1, 'Greg', 'Vachovsky', '23.3.1967', 666777);  INSERT INTO client (client\_id, client\_name, client\_lastname, client\_dob, client\_passport) VALUES (2, 'Andry', 'Blaskovic', '13.6.1973', 666888);  INSERT INTO client (client\_id, client\_name, client\_lastname, client\_dob, client\_passport) VALUES (3, 'Pinguin', 'Kavalsky', '14.12.1956', 666999);  INSERT INTO client (client\_id, client\_name, client\_lastname, client\_dob, client\_passport) VALUES (4, 'Lovely', 'Caezar', '1.10.1999', 666111);  INSERT INTO client (client\_id, client\_name, client\_lastname, client\_dob, client\_passport) VALUES (5, 'Dude', 'Some', '22.2.1993', 666222);  END ^  set term ; ^  SELECT \* FROM client;  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO groups (group\_id) VALUES (1);  INSERT INTO groups (group\_id) VALUES (2);  INSERT INTO groups (group\_id) VALUES (3);  INSERT INTO groups (group\_id) VALUES (4);  INSERT INTO groups (group\_id) VALUES (5);  END ^  set term ; ^  SELECT \* FROM groups;  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO access\_type (accesstype\_id, accesstype\_type) VALUES (1, 'only\_client');  INSERT INTO access\_type (accesstype\_id, accesstype\_type) VALUES (2, 'only\_group');  INSERT INTO access\_type (accesstype\_id, accesstype\_type) VALUES (3, 'client\_and\_other\_client');  INSERT INTO access\_type (accesstype\_id, accesstype\_type) VALUES (4, 'client\_and\_other\_group');  INSERT INTO access\_type (accesstype\_id, accesstype\_type) VALUES (5, 'list\_of\_clients');  END ^  set term ; ^  SELECT \* FROM access\_type;  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO deposit\_box (depb\_id, depb\_num, depb\_value) VALUES (1, 1, 123.456);  INSERT INTO deposit\_box (depb\_id, depb\_num, depb\_value) VALUES (2, 2, 5.123);  INSERT INTO deposit\_box (depb\_id, depb\_num, depb\_value) VALUES (3, 3, 4444);  INSERT INTO deposit\_box (depb\_id, depb\_num, depb\_value) VALUES (4, 4, 0);  INSERT INTO deposit\_box (depb\_id, depb\_num, depb\_value) VALUES (5, 5, 14);  END ^  set term ; ^  SELECT \* FROM deposit\_box;  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO group\_member (gm\_id, group\_id, client\_id) VALUES (1, 1, 1);  INSERT INTO group\_member (gm\_id, group\_id, client\_id) VALUES (2, 1, 2);  INSERT INTO group\_member (gm\_id, group\_id, client\_id) VALUES (3, 3, 3);  INSERT INTO group\_member (gm\_id, group\_id, client\_id) VALUES (4, 4, 4);  INSERT INTO group\_member (gm\_id, group\_id, client\_id) VALUES (5, 5, 5);  END ^  set term ; ^  SELECT \* FROM group\_member;  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO contract (contract\_id, contract\_start, contract\_end, contract\_cost, depb\_id, accesstype\_id, group\_id) VALUES (1, '24.11.2017', '1.12.2022', 7000.45, 1, 1, 1);  INSERT INTO contract (contract\_id, contract\_start, contract\_end, contract\_cost, depb\_id, accesstype\_id, group\_id) VALUES (2, '2.1.2000', '12.3.2028', 12, 2, 2, 2);  INSERT INTO contract (contract\_id, contract\_start, contract\_end, contract\_cost, depb\_id, accesstype\_id, group\_id) VALUES (3, '4.12.2003', '22.1.2019', 366, 3, 3, 3);  INSERT INTO contract (contract\_id, contract\_start, contract\_end, contract\_cost, depb\_id, accesstype\_id, group\_id) VALUES (4, '11.11.2006', '1.12.2067', 566, 4, 4, 4);  INSERT INTO contract (contract\_id, contract\_start, contract\_end, contract\_cost, depb\_id, accesstype\_id, group\_id) VALUES (5, '24.11.2007', '1.12.2032', 70, 5, 5, 5);  END ^  set term ; ^  SELECT \* FROM contract;  set term ^ ;  EXECUTE BLOCK AS BEGIN  INSERT INTO box\_history (history\_id, client\_id, history\_previousValue, history\_currentValue, depb\_id, contract\_id) VALUES (1, 1, 56.17, 3333, 1, 1);  INSERT INTO box\_history (history\_id, client\_id, history\_previousValue, history\_currentValue, depb\_id, contract\_id) VALUES (2, 2, 56.17, 3333, 2, 2);  INSERT INTO box\_history (history\_id, client\_id, history\_previousValue, history\_currentValue, depb\_id, contract\_id) VALUES (3, 3, 56.17, 3333, 3, 3);  INSERT INTO box\_history (history\_id, client\_id, history\_previousValue, history\_currentValue, depb\_id, contract\_id) VALUES (4, 4, 56.17, 3333, 4, 4);  INSERT INTO box\_history (history\_id, client\_id, history\_previousValue, history\_currentValue, depb\_id, contract\_id) VALUES (5, 5, 56.17, 3333, 5, 5);  END ^  set term ; ^  SELECT \* FROM box\_history;  commit; |

Проверим данные с помощью СУБД:

|  |
| --- |
| ACCESS\_TYPE |
|  |
| BOX\_HISTORY |
|  |
| CLIENT |
|  |
| CONTRACT |
|  |
| DEPOSIT\_BOX |
|  |
| GROUP |
|  |
| GROUP\_MEMBER |
|  |

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

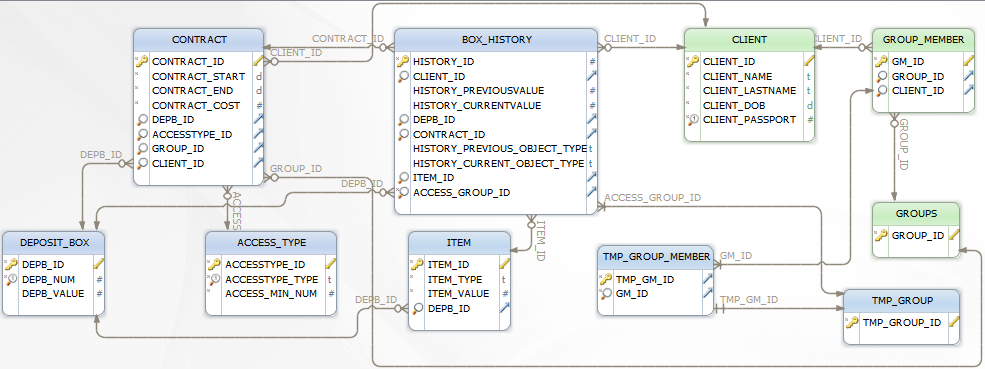
Задание:

1. Реализовать расширенные требования к доступу к ячейкам: минимальное количество одновременно получающих доступ лиц, обязательность для доступа определенного члена группы доступа.
2. Ввести учет хранения каждого предмета в ячейке с категориями и ценностью.

Изменим схему созданной БД, выполняя SQL-запросы:

|  |
| --- |
| ALTER TABLE ACCESS\_TYPE ADD access\_min\_num int DEFAULT 1 not null;  ALTER TABLE contract ADD client\_id int REFERENCES client;  INSERT INTO access\_type (accesstype\_id, accesstype\_type)  VALUES (6, 'with\_specific\_client\_only');    INSERT INTO access\_type (accesstype\_id, accesstype\_type)  VALUES (7, 'min\_amount\_of\_clients');    CREATE TABLE item (  item\_id int not null PRIMARY KEY,  item\_type varchar(128) not null,  item\_value float not null,  depb\_id int references DEPOSIT\_BOX);    ALTER TABLE BOX\_HISTORY ADD item\_id int references ITEM;  CREATE TABLE tmp\_group (tmp\_group\_id int not null primary key);  CREATE TABLE tmp\_group\_member (tmp\_gm\_id int not null primary key references tmp\_group, gm\_id int not null references group\_member);  ALTER TABLE BOX\_HISTORY ADD access\_group\_id int not null references tmp\_group; |

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



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

|  |
| --- |
| BOX\_HISTORY |
|  |
| CLIENT |
|  |
| CONTRACT |
|  |
| DEPOSIT\_BOX |
|  |
| GROUP\_MEMBER |
|  |

**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 СУБД Database Workbench. Он позволяет выполнять широкий набор операций с базами данных.