Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчёт о лабораторной работе №2

Дисциплина: Базы данных

Тема: Язык SQL-DDL

Выполнил студент гр. 43501/1 В.А.Чернышева

(подпись)

Руководитель А.В. Мяснов

(подпись)

“ ” 2015 г.

Санкт-Петербург

2015

1. **Цель работы**

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

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

SET SQL DIALECT 3;

SET NAMES WIN1251;

CREATE DATABASE 'C:\123\BIBLIOTEKA.FDB'

USER 'SYSDBA' PASSWORD 'masterkey'

PAGE\_SIZE 16384

DEFAULT CHARACTER SET WIN1251 COLLATION WIN1251;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\* Tables \*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TABLE AUTHOR (

ID INTEGER NOT NULL,

SURNAME VARCHAR(50) NOT NULL,

NAME VARCHAR(50) NOT NULL,

PARTRONYMIC VARCHAR(50) NOT NULL

);

CREATE TABLE BOOK (

ID INTEGER NOT NULL,

NAME VARCHAR(50) NOT NULL,

ID\_TYPE INTEGER NOT NULL,

ID\_GENRE INTEGER NOT NULL

);

CREATE TABLE BOOK\_AUTHOR (

ID\_AUTHOR INTEGER NOT NULL,

ID\_BOOK INTEGER NOT NULL

);

CREATE TABLE EXEMP (

ID INTEGER NOT NULL,

ID\_BOOK INTEGER NOT NULL,

ID\_STATUS INTEGER NOT NULL

);

CREATE TABLE GENRE (

ID INTEGER NOT NULL,

NAME VARCHAR(50) NOT NULL

);

CREATE TABLE INTAKE (

ID INTEGER NOT NULL,

DATE\_INTAKE DATE NOT NULL,

ID\_PROVAIDER INTEGER NOT NULL,

ID\_EXEMP INTEGER NOT NULL

);

CREATE TABLE OPERATION (

ID INTEGER NOT NULL,

ID\_READERSHIP INTEGER NOT NULL,

ID\_EXEMP INTEGER NOT NULL,

ID\_TYPE\_OF\_ISSUE INTEGER NOT NULL,

PERIOD INTEGER NOT NULL,

DATE\_OF\_ISSUE DATE NOT NULL,

RETERN\_DATE DATE

);

CREATE TABLE PROVAIDER (

ID INTEGER NOT NULL,

NAME VARCHAR(50)

);

CREATE TABLE READERSHIP (

ID INTEGER NOT NULL,

SURNAME VARCHAR(50) NOT NULL,

NAME VARCHAR(50) NOT NULL,

PARTRONYMIC VARCHAR(50) NOT NULL,

PHONE\_NUMBER INTEGER NOT NULL

);

CREATE TABLE RESERVE (

ID INTEGER NOT NULL,

ID\_READSHIP INTEGER NOT NULL,

ID\_EXEMP INTEGER NOT NULL,

DATE\_RESERVE DATE

);

CREATE TABLE STATUS (

ID INTEGER NOT NULL,

NAME VARCHAR(100)

);

CREATE TABLE "TYPE" (

ID INTEGER NOT NULL,

NAME VARCHAR(50) NOT NULL

);

CREATE TABLE TYPE\_OF\_ISSUE (

ID INTEGER NOT NULL,

NAME VARCHAR(50) NOT NULL

);

CREATE TABLE WRITE\_OFF (

ID INTEGER NOT NULL,

DATE\_WRITE\_OFF DATE NOT NULL,

ID\_EXEMP INTEGER NOT NULL

);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\* Primary keys \*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ALTER TABLE AUTHOR ADD PRIMARY KEY (ID);

ALTER TABLE BOOK ADD PRIMARY KEY (ID);

ALTER TABLE EXEMP ADD PRIMARY KEY (ID);

ALTER TABLE GENRE ADD PRIMARY KEY (ID);

ALTER TABLE INTAKE ADD PRIMARY KEY (ID);

ALTER TABLE OPERATION ADD PRIMARY KEY (ID);

ALTER TABLE PROVAIDER ADD PRIMARY KEY (ID);

ALTER TABLE READERSHIP ADD PRIMARY KEY (ID);

ALTER TABLE RESERVE ADD PRIMARY KEY (ID);

ALTER TABLE STATUS ADD PRIMARY KEY (ID);

ALTER TABLE "TYPE" ADD PRIMARY KEY (ID);

ALTER TABLE TYPE\_OF\_ISSUE ADD PRIMARY KEY (ID);

ALTER TABLE WRITE\_OFF ADD PRIMARY KEY (ID);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\* Foreign keys \*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ALTER TABLE BOOK ADD FOREIGN KEY (ID\_TYPE) REFERENCES "TYPE" (ID);

ALTER TABLE BOOK ADD FOREIGN KEY (ID\_GENRE) REFERENCES GENRE (ID);

ALTER TABLE BOOK\_AUTHOR ADD FOREIGN KEY (ID\_AUTHOR) REFERENCES AUTHOR (ID);

ALTER TABLE BOOK\_AUTHOR ADD FOREIGN KEY (ID\_BOOK) REFERENCES BOOK (ID);

ALTER TABLE EXEMP ADD FOREIGN KEY (ID\_BOOK) REFERENCES BOOK (ID);

ALTER TABLE EXEMP ADD FOREIGN KEY (ID\_STATUS) REFERENCES STATUS (ID);

ALTER TABLE INTAKE ADD FOREIGN KEY (ID\_PROVAIDER) REFERENCES PROVAIDER (ID);

ALTER TABLE INTAKE ADD FOREIGN KEY (ID\_EXEMP) REFERENCES EXEMP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_EXEMP) REFERENCES EXEMP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_READERSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE OPERATION ADD FOREIGN KEY (ID\_TYPE\_OF\_ISSUE) REFERENCES TYPE\_OF\_ISSUE (ID);

ALTER TABLE RESERVE ADD FOREIGN KEY (ID\_READSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE RESERVE ADD FOREIGN KEY (ID\_EXEMP) REFERENCES EXEMP (ID);

ALTER TABLE RESERVE ADD FOREIGN KEY (ID\_READSHIP) REFERENCES READERSHIP (ID);

ALTER TABLE WRITE\_OFF ADD FOREIGN KEY (ID\_EXEMP) REFERENCES EXEMP (ID);

* 1. **Заполнение БД данными**

insert into GENRE (ID, NAME) values (1, 'Detectiv');

insert into GENRE (ID, NAME) values (2, 'Outline');

insert into GENRE (ID, NAME) values (3, 'Story');

insert into GENRE (ID, NAME) values (4, 'Play');

insert into GENRE (ID, NAME) values (5, 'Novel');

insert into GENRE (ID, NAME) values (6, 'Sketch');

insert into GENRE (ID, NAME) values (7, 'Epopee');

insert into GENRE (ID, NAME) values (8, 'Epos');

insert into GENRE (ID, NAME) values (9, 'Message');

insert into GENRE (ID, NAME) values (10, 'Elegy');

insert into type (ID, NAME) values (1, 'Artist`s books');

insert into TYPE (ID, NAME) values (2, 'Booklets');

insert into TYPE (ID, NAME) values (3, 'Chapbooks');

insert into TYPE (ID, NAME) values (4, 'Diaries');

insert into TYPE (ID, NAME) values (5, 'Festival books');

insert into TYPE (ID, NAME) values (6, 'Manuscripts');

insert into TYPE (ID, NAME) values (7, 'Scrolls');

insert into TYPE (ID, NAME) values (8, 'Volksbuch');

insert into TYPE (ID, NAME) values (9, 'Journal');

insert into TYPE (ID, NAME) values (10, 'Book magazine');

insert into TYPE\_OF\_ISSUE (ID, NAME) values (1, 'Reserve');

insert into TYPE\_OF\_ISSUE (ID, NAME) values (2, 'Stock');

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (1, 'Rusakov','Mihail','Andreevich', 89316666666);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (2, 'Larionova','Anna','Alekseevna', 89117564378);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (3, 'Menshikov', 'Alexandr', 'Georgievuch', 89128765434);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (4, 'Fridman','Mihail','Anatolevich', 89568675423);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (5, 'Chernyshev','Matvey','Andreevich', 89236451290);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (6, 'Ivanov','Ivan','Ivanovich', 84673747632);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (7, 'Peters','Elizaveta','Petrovna', 89846446732);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (8, 'Nikulina','Alexsandra','Leonidovna', 89464646);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (9, 'Novokirova','Julia','Anatolievna', 89564323456);

insert into READERSHIP (ID, SURNAME, NAME, PARTRONYMIC, PHONE\_NUMBER )

values (10, 'Ivanov','Anton','Nikolaevich', 89567353423);

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (2, 'Radzinsky','Edvard','Stanislavovich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (3, 'Kuprin','Alexandr','Ivanovich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (4, 'Bulgakov','Mihail','Afanasevich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (5, 'Chehon','Anton','Pavlovich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (6, 'Pushkin','Alexandr','Sergeevich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (7, 'Fonvisin','Denis','Ivanovich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (8, 'Radischev','Alexandr','Nikolaevich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (9, 'Blok','Alexandr','Aleksandrovich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (1, 'Zaitzev','Boris','Konstantinovich');

insert into AUTHOR (ID, SURNAME, NAME, PARTRONYMIC )

values (11, 'Zamyatin','Evgeniy','Ivanovich');

insert into exemp(ID, ID\_BOOK) values (10,0);

insert into exemp(ID, ID\_BOOK) values (11,0);

insert into exemp(ID, ID\_BOOK) values (12,1);

insert into exemp(ID, ID\_BOOK) values (13,1);

insert into exemp(ID, ID\_BOOK) values (14,2);

insert into exemp(ID, ID\_BOOK) values (15,2);

insert into exemp(ID, ID\_BOOK) values (16,2);

insert into exemp(ID, ID\_BOOK) values (17,4);

insert into exemp(ID, ID\_BOOK) values (18,3);

insert into exemp(ID, ID\_BOOK) values (19,4);

insert into BOOK(ID, NAME, ID\_TYPE, ID\_GENRE) values (0,'MASTER AND MARGARITA',1,2);

insert into BOOK(ID, NAME, ID\_TYPE, ID\_GENRE) values (1,'THE WHITE GUARD',2,3);

insert into BOOK(ID, NAME, ID\_TYPE, ID\_GENRE) values (2,'COLLECTION OF POEMS',7,5);

insert into BOOK(ID, NAME, ID\_TYPE, ID\_GENRE) values (3,'EUGENE ONEGIN',8,10);

insert into BOOK(ID, NAME, ID\_TYPE, ID\_GENRE) values (4,'GARNET BRACELET. TALE.',6,9);

insert into BOOK\_AUTHOR(ID\_AUTHOR, ID\_BOOK) values (4,0);

insert into BOOK\_AUTHOR(ID\_AUTHOR, ID\_BOOK) values (4,1);

insert into BOOK\_AUTHOR(ID\_AUTHOR, ID\_BOOK) values (9,2);

insert into BOOK\_AUTHOR(ID\_AUTHOR, ID\_BOOK) values (6,3);

insert into BOOK\_AUTHOR(ID\_AUTHOR, ID\_BOOK) values (3,4);

SET NAMES WIN1251;

CONNECT 'C:\123\BIBLIOTEKA.FDB' USER 'SYSDBA' PASSWORD 'masterkey';

insert into RESERVE (ID, ID\_READSHIP, ID\_EXEMP)values (0,1,10);

insert into RESERVE (ID, ID\_READSHIP, ID\_EXEMP)values (1,1234,11);

insert into RESERVE (ID, ID\_READSHIP, ID\_EXEMP)values (2,5674,14);

insert into RESERVE (ID, ID\_READSHIP, ID\_EXEMP)values (3,56,15);

Поля BOOK, BOOK\_AUTHOR,AUTHOR,READERSHIP,EXEMP были заполнены и дозаполнены на 100000 записей с помощью генератора данных в IBExpert.

* 1. **Модификация БД соответственно заданию.**

1. Ввести учет приобретения и списания книг.

SET NAMES WIN1251;

CONNECT 'C:\123\BIBLIOTEKA.FDB' USER 'SYSDBA' PASSWORD 'masterkey';

CREATE TABLE PROVAIDER(

ID INTEGER NOT NULL,

NAME VARCHAR(50)

);

CREATE TABLE INTAKE(

ID INTEGER NOT NULL,

DATE\_INTAKE DATE NOT NULL,

ID\_PROVAIDER integer NOT NULL,

ID\_EXEMP integer NOT NULL

);

CREATE TABLE WRITE\_OFF(

ID INTEGER NOT NULL,

DATE\_WRITE\_OFF DATE NOT NULL,

ID\_EXEMP integer NOT NULL

);

ALTER TABLE INTAKE ADD PRIMARY KEY (ID);

ALTER TABLE PROVAIDER ADD PRIMARY KEY (ID);

ALTER TABLE WRITE\_OFF ADD PRIMARY KEY (ID);

ALTER TABLE INTAKE ADD FOREIGN KEY (ID\_PROVAIDER) REFERENCES "PROVAIDER" (ID);

ALTER TABLE INTAKE ADD FOREIGN KEY (ID\_EXEMP) REFERENCES "EXEMP" (ID);

ALTER TABLE WRITE\_OFF ADD FOREIGN KEY (ID\_EXEMP) REFERENCES "EXEMP" (ID);

+ добавлено поле status в таблицу EXEMP

SET NAMES WIN1251;

CONNECT 'C:\123\BIBLIOTEKA.FDB' USER 'SYSDBA' PASSWORD 'masterkey';

alter table EXEMP drop WRITTEN\_OFF

CREATE TABLE STATUS (

ID int NOT NULL PRIMARY KEY,

NAME varchar(100)

);

ALTER TABLE EXEMP ADD FOREIGN KEY (ID\_STATUS) REFERENCES "STATUS" (ID);

Заполнение:

SET NAMES WIN1251;

CONNECT 'C:\123\BIBLIOTEKA.FDB' USER 'SYSDBA' PASSWORD 'masterkey';

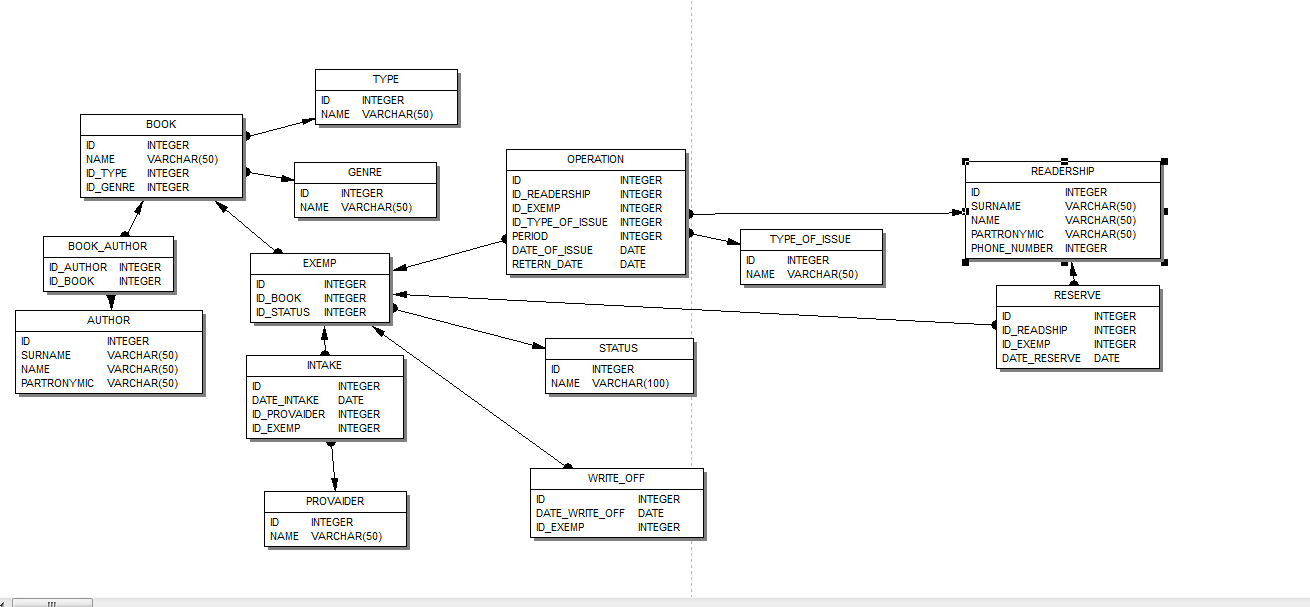
insert into STATUS (ID, NAME)values (0, 'Вналичии');

insert into STATUS (ID, NAME)values (1, 'У читателя');

insert into STATUS (ID, NAME)values (2, 'Списана');

insert into STATUS (ID, NAME)values (3, 'Резерв');

С помощью Database Designer в IBExpert была сгенерирована ER-диаграмма нашей БД:



1. **Вывод**

В результате работы с IBExpert был изучен язык SQL-DDL.

Data Definition Language (DDL) — язык описания данных. С помощью этого подмножества языка создаются, модифицируются и удаляются объекты базы данных .

Эти действия возможны с помощью инструментов языка таких как:

1)create

2)alter

3)drop и другие

После выполнения данных команд автоматически выполняется команда «commit», т.е. выполненные изменения уже нельзя отменить, поэтому следует быть предельно внимательным при работе с данным языком.

Язык очень прост в изучении, поэтому при выполнении работы затруднений не возникло. С помощью IBExperta очень легко получить ER-диаграмму, сгенерировать данные для любой таблицы, что очень удобно в случае, когда таблица очень большая. и гибкость языка позволяет легко изменить составляющие БД.