

▪ **Санкт- Петербургский политехнический университет Петра Великого**

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе

Курс: «Базы данных»

Тема: «Язык SQL-DDL»

Выполнил:

Воробей Е.А. группа 43501/3

Проверил:

Мяснов А.В.

Санкт – Петербург

2016

1. Цель работы

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

2. 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. Ход работы

В соответствии с диаграммой базы данных из предыдущей работы (Рис. 1), был написан SQL-скрипт, создающий базу данных.

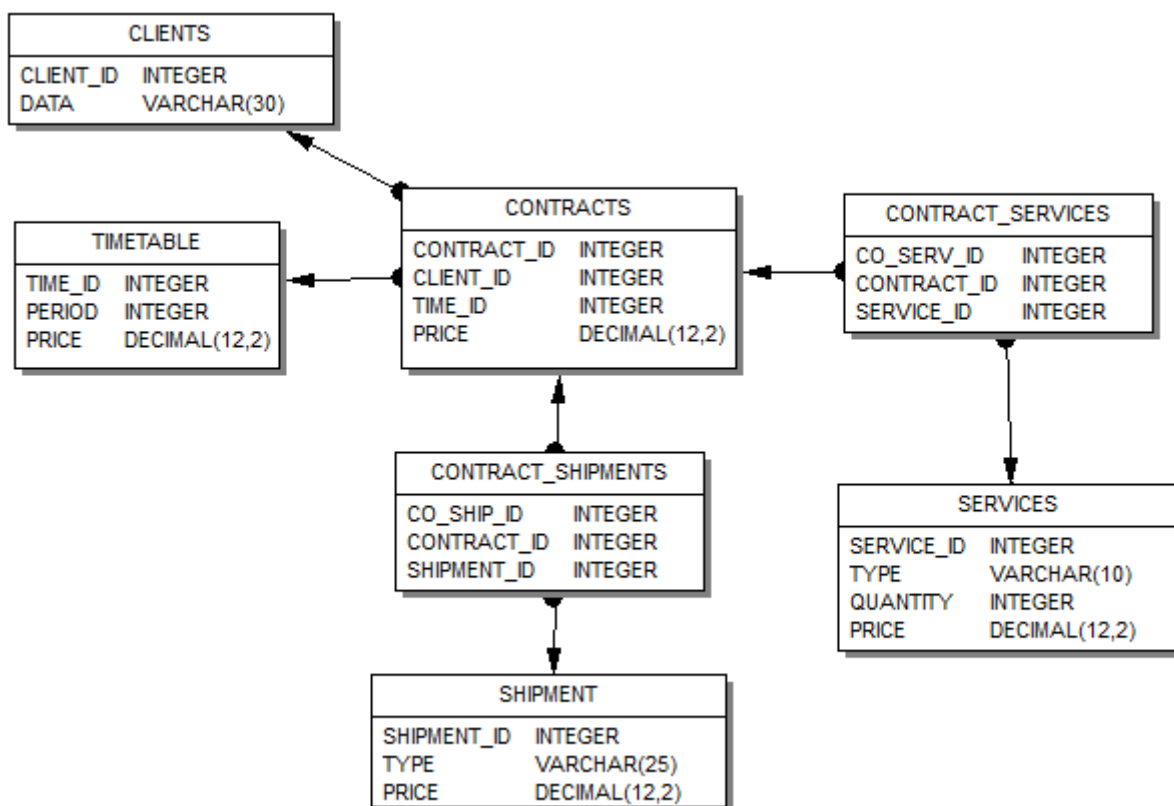


Рис. 1. SQL-диаграмма базы данных

Листинг create.sql:

```
/* If database already exists drop it. */

CONNECT 'tiger.ftk.spbstu.ru:/var/lib/firebird/435013/adagency.fdb' USER
'SYSDBA' PASSWORD 'masterkey';

DROP DATABASE;

/* Create database and connect. */

CREATE DATABASE 'tiger.ftk.spbstu.ru:/var/lib/firebird/435013/adagency.fdb' USER
'SYSDBA' PASSWORD 'masterkey';

CONNECT 'tiger.ftk.spbstu.ru:/var/lib/firebird/435013/adagency.fdb' USER
'SYSDBA' PASSWORD 'masterkey';

/* Set charset. */

SET NAMES CYRL;

/* create tables. */

CREATE TABLE CLIENTS (
    CLIENT_ID    INTEGER NOT NULL,
    DATA        VARCHAR(30) NOT NULL,
    CONSTRAINT PK_CLIENTS PRIMARY KEY (CLIENT_ID)
);
COMMIT;

CREATE TABLE TIMETABLE (
    TIME_ID    INTEGER NOT NULL,
    PERIOD     INTEGER NOT NULL,
    DECIMAL(12,2) NOT NULL,
    CONSTRAINT PK_TIMETABLE PRIMARY KEY (TIME_ID)
);
COMMIT;

CREATE TABLE CONTRACTS (
    CONTRACT_ID  INTEGER NOT NULL,
    CLIENT_ID    INTEGER NOT NULL REFERENCES CLIENTS
    TIME_ID      INTEGER NOT NULL REFERENCES TIMETABLE,
    DECIMAL(12,2) NOT NULL,
    CONSTRAINT PK_CONTRACTS PRIMARY KEY (CONTRACT_ID)
);
COMMIT;

CREATE TABLE SERVICES (
    SERVICE_ID  INTEGER NOT NULL,
```

```

        "TYPE"          VARCHAR(10) NOT NULL,
        QUANTITY        INTEGER NOT NULL,
        DECIMAL(12,2) NOT NULL,
        CONSTRAINT PK_SERVICES PRIMARY KEY (SERVICE_ID)
    );
COMMIT;

CREATE TABLE CONTRACT_SERVICES (
    CO_SERV_ID    INTEGER NOT NULL,
    CONTRACT_ID   INTEGER NOT NULL REFERENCES CONTRACTS,
    SERVICE_ID    INTEGER NOT NULL REFERENCES SERVICES,
    CONSTRAINT PK_CONTRACT_SERVICES PRIMARY KEY (CO_SERV_ID)
);
COMMIT;

CREATE TABLE SHIPMENT (
    SHIPMENT_ID   INTEGER NOT NULL,
    "TYPE"        VARCHAR(25) NOT NULL,
    DECIMAL(12,2) NOT NULL,
    CONSTRAINT PK_SHIPMENT PRIMARY KEY (SHIPMENT_ID)
);
COMMIT;

CREATE TABLE CONTRACT_SHIPMENTS (
    CO_SHIP_ID    INTEGER NOT NULL,
    CONTRACT_ID   INTEGER NOT NULL REFERENCES CONTRACTS,
    SHIPMENT_ID   INTEGER NOT NULL REFERENCES SHIPMENT,
    CONSTRAINT PK_CONTRACT_SHIPMENTS PRIMARY KEY (CO_SHIP_ID)
);
COMMIT;

```

Данный скрипт создает базу данных (если база данных уже создана, то удаляет старую), подключается к ней, устанавливает кодировку и создает таблицы со связями.

Создадим скрипт, наполняющий базу данных небольшим объемом осмысленных данных:

insert.sql:

```

INSERT INTO clients (CLIENT_ID, DATA) VALUES (01, 'Антонов Егор');
INSERT INTO clients (CLIENT_ID, DATA) VALUES (02, 'Букреев Борис');
INSERT INTO clients (CLIENT_ID, DATA) VALUES (03, 'Воргулева Анна');
INSERT INTO clients (CLIENT_ID, DATA) VALUES (04, 'Никромов Илья');
INSERT INTO clients (CLIENT_ID, DATA) VALUES (05, 'Астахов Фарид');
COMMIT;

INSERT INTO timetable (TIME_ID, PERIOD, PRICE) VALUES (1, 30, 1000);
INSERT INTO timetable (TIME_ID, PERIOD, PRICE) VALUES (2, 45, 1500);
INSERT INTO timetable (TIME_ID, PERIOD, PRICE) VALUES (3, 60, 2000);

```

```
INSERT INTO timetable (TIME_ID, PERIOD, PRICE) VALUES (4, 75, 2900);
INSERT INTO timetable (TIME_ID, PERIOD, PRICE) VALUES (5, 90, 3700);
COMMIT;
```

```
INSERT INTO contracts (CONTRACT_ID, CLIENT_ID, PRICE, TIME_ID)
VALUES (1, 2, 3000, 3);
INSERT INTO contracts (CONTRACT_ID, CLIENT_ID, PRICE, TIME_ID)
VALUES (2, 1, 2500, 1);
INSERT INTO contracts (CONTRACT_ID, CLIENT_ID, PRICE, TIME_ID)
VALUES (3, 3, 5000, 4);
INSERT INTO contracts (CONTRACT_ID, CLIENT_ID, PRICE, TIME_ID)
VALUES (4, 1, 1000, 1);
```

COMMIT;

```
INSERT INTO services (SERVICE_ID, TYPE, PRICE, QUANTITY)
VALUES (1, 'Листовки', 10, 1000);
INSERT INTO services (SERVICE_ID, TYPE, PRICE, QUANTITY)
VALUES (2, 'Рекламный щит', 2500, 5);
INSERT INTO services (SERVICE_ID, TYPE, PRICE, QUANTITY)
VALUES (3, 'Плакаты', 1000, 10);
INSERT INTO services (SERVICE_ID, TYPE, PRICE, QUANTITY)
VALUES (4, 'Интернет-реклама', 500, 5);
INSERT INTO services (SERVICE_ID, TYPE, PRICE, QUANTITY)
VALUES (5, 'ТВ-реклама', 10000, 1);
COMMIT;
```

```
INSERT INTO CONTRACTSERVICES (CO_SERV_ID, CONTRACT_ID, SERVICE_ID)
VALUES (1, 1, 3);
INSERT INTO CONTRACTSERVICES (CO_SERV_ID, CONTRACT_ID, SERVICE_ID)
VALUES (2, 1, 2);
INSERT INTO CONTRACTSERVICES (CO_SERV_ID, CONTRACT_ID, SERVICE_ID)
VALUES (3, 2, 1);
INSERT INTO CONTRACTSERVICES (CO_SERV_ID, CONTRACT_ID, SERVICE_ID)
VALUES (4, 4, 1);
INSERT INTO CONTRACTSERVICES (CO_SERV_ID, CONTRACT_ID, SERVICE_ID)
VALUES (5, 3, 5);
COMMIT;
```

```
INSERT INTO SHIPMENT (SHIPMENT_ID, TYPE, PRICE)
VALUES (1, 'Промоутеры', 1000);
INSERT INTO SHIPMENT (SHIPMENT_ID, TYPE, PRICE)
VALUES (2, 'Вебдизайнеры', 100);
INSERT INTO SHIPMENT (SHIPMENT_ID, TYPE, PRICE)
VALUES (3, 'Установщики щитов', 100);
INSERT INTO SHIPMENT (SHIPMENT_ID, TYPE, PRICE)
VALUES (4, 'ТВ-студия', 5000);
INSERT INTO SHIPMENT (SHIPMENT_ID, TYPE, PRICE)
VALUES (5, 'Расклеивающие плакатов', 100);
```

COMMIT;

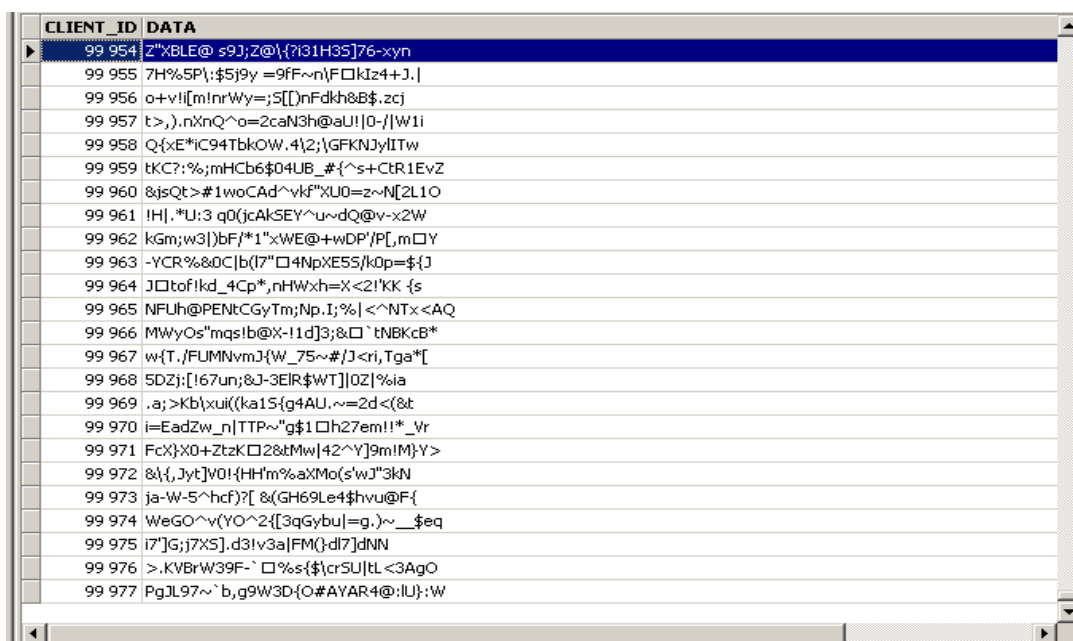
```
INSERT INTO CONTRACTSHIPMENTS (CO_SHIP_ID, CONTRACT_ID, SHIPMENT_ID)
VALUES (1, 2, 3);
INSERT INTO CONTRACTSHIPMENTS (CO_SHIP_ID, CONTRACT_ID, SHIPMENT_ID)
VALUES (1, 4, 1);
INSERT INTO CONTRACTSHIPMENTS (CO_SHIP_ID, CONTRACT_ID, SHIPMENT_ID)
VALUES (1, 1, 1);
INSERT INTO CONTRACTSHIPMENTS (CO_SHIP_ID, CONTRACT_ID, SHIPMENT_ID)
VALUES (1, 2, 5);
INSERT INTO CONTRACTSHIPMENTS (CO_SHIP_ID, CONTRACT_ID, SHIPMENT_ID)
```

```
VALUES (1, 3, 3);  
COMMIT;
```

Теперь заполним таблицы данными с помощью генератора данных среды разработки IBExpert.

Рассмотрим некоторые способы генерации данных в IBExpert: генерация случайных данных в некотором диапазоне, генерация случайных данных из списка, генерация случайных данных из другой таблицы. Для целочисленных типов существует генерация автоинкрементом, где можно также настроить величину шага. Эта опция обычно используется для генерирования первичных ключей.

На рисунке представлен пример случайно сгенерированных данных



CLIENT_ID	DATA
99 954	Z"%BLE@ s9J;Z@{\?i31H35]76-xyn
99 955	7H%5P!,:\$5j9y =9fF~n\F□kIz4+J.
99 956	o+vl[m!nrWy=;S[[]nFdkh8B\$.zc]
99 957	t>,).nXnQ^o=2caN3h@aU! 0-/ W1i
99 958	Q{xE*iC94TbkOW.4\2;\GfKNJyIITw
99 959	tKC?:%;mHCb6\$04UB_#{"^s+CtR1EvZ
99 960	&jsQt>#1woCAd^vkf"XU0=z~N[2L1O
99 961	!H . *U:3 q0(jcAkSEY^u~dQ@v-x2W
99 962	kGm;w3)bF/*1"xWE@+wDP'/P[,m□Y
99 963	-YCR%&0C b(l7"□4NpXE55/k0p=\${J
99 964	J□tofi kd_4Cp*,nHW:xh=X<2!KK {s
99 965	NFUh@PENTCGyTm;Np.I; % <^NTx<AQ
99 966	MWyoS"mqslb@X-l1d]3;&□`tNBKcB*
99 967	w{T./FUMNvmJ{W_75~#/J<ri,Tga*[
99 968	SDZ):[l67un;U-3EIR\$WT] 0Z %la
99 969	.a;>Kb\Xui((ka1S{g4AU.~≈2d<(&t
99 970	i=EadZw_n TTP~"g\$1□h27em!!*_Vr
99 971	FcX}X0+ZtzK□2&tMw 42^Y]9m!M}Y>
99 972	&\{,Jyt}V0!{HH'm%aXMo(s'wJ"3kN
99 973	ja-W-5^hcf)?[&(GH69Le4\$hvU@F{
99 974	WeGO^v{YO^2{[3qGybu =g.)~__\$eq
99 975	i7]G;j7XS].d3!v3a FM{)dl7]dNN
99 976	>.KVBrW39F~`□%s{\$\crSU tL<3AgO
99 977	PgJL97~`b,g9W3D{O#AYAR4@:lU}:vW

Рис 2. Пример сгенерированных данных

4. Вывод