

Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологии

-Кафедра Компьютерных Систем и Программных Технологий-

Лабораторная работа №3

по дисциплине "Информационное управление систем управления"

«Разработка структуры и нормализация БД»

Работу выполнил студент гр. 43501/1 Нгуен Тиен Ву

Проверил преподаватель : Мяснов А.В

Санкт-Петербург

2015

Задание

Страховая компания:

Хранит информацию о клиентах, типа ущерба, типах договоров (включающих список типов ущерба), заключённых договорах, сроках действия, страховых случаях, оценках ущерба и выплатах. ИС позволяет рассчитывать стоимость страховки с учетом истории клиента.

1. Цель работы:

- ознакомление с основами проектирования схемы БД, способами нормализации отношений в БД.

2. Программа работы

- 2.1. Создание скрипта БД в соответствии с согласованной схемой (должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений). Продемонстрировать скрипт преподавателю.
- 2.2. Создайте скрипт, заполняющий все таблицы БД данными
- 2.3. Изучите основные возможности IVExpert. Получите ER-диаграмму созданной БД с помощью **Database Designer**.
- 2.4. Автоматически сгенерируйте данные при помощи IVExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

3. Выполнение программы работы

Скрипт создания БД в соответствии с согласованной схемой приведен в листинге 1.

Созданный скрипт проверен на работоспособность и продемонстрирован преподавателю.

Листинг 1

```
CONNECT 'E:\tienvu.fdb' user 'SYSDBA' password 'masterkey';
COMMIT;
DROP DATABASE;
COMMIT;
CREATE DATABASE 'E:\tienvu.fdb' user 'SYSDBA' password 'masterkey';
COMMIT;
CONNECT 'E:\tienvu.fdb' user 'SYSDBA' password 'masterkey';

CREATE TABLE ADRESS (
ID_ADRESS    INTEGER NOT NULL,
CITY         VARCHAR(30) ,
STREET       VARCHAR(30) ,
BUILDING     VARCHAR(30) ,
ROOM         VARCHAR(30)
);

CREATE TABLE "CASE" (
ID_CASE      INTEGER NOT NULL,
CASE_IN      VARCHAR(50)
);

CREATE TABLE CLIENT (
ID_CLIENT    INTEGER NOT NULL,
```

```

NAME          VARCHAR(30),
BIRTH_DAY     DATE,
ADRESS        INTEGER NOT NULL,
);

CREATE TABLE INSURANCE_CASE_CON (
ID_CONTRACT   INTEGER NOT NULL,
CASE          INTEGER,
SUMP          INTEGER
);

CREATE TABLE INSURANCE_CASE_POLIC (ID_POLIC   INTEGER   NOT   NULL,   "CASE"
INTEGER NOT NULL
);

CREATE TABLE POLIC (
TYPE          INTEGER NOT NULL,
TERM          INTEGER NOT NULL,
INSURANCE_CASE   INTEGER NOT NULL,
ESTIMATE_DAMAGES VARCHAR(50)
);

CREATE TABLE PRICE (
ID_PRICE      INTEGER NOT NULL,
PRICE_IN      VARCHAR(50)
);

CREATE TABLE SIGN_CONTRACT (
CLIENT        INTEGER NOT NULL,
POLIC         INTEGER NOT NULL,
CONTRACT      INTEGER NOT NULL,
PRICE         INTEGER NOT NULL,
TYPE          INTEGER NOT NULL,
TERM          INTEGER NOT NULL
);

CREATE TABLE "TERM" (
ID_TERM       INTEGER NOT NULL,
TERM_IN       CHAR(50)
);

CREATE TABLE "TYPE" (
ID_TYPE       INTEGER,
TYPE_IN       CHAR(50)
);

/*****PRIMARY KEYES*****/
ALTER TABLE ADRESS ADD CONSTRAINT PK_ADRESS PRIMARY KEY (ID_ADRESS);
ALTER TABLE ADRESS ADD CHECK (ID_ADRESS>0);
ALTER TABLE "CASE" ADD CONSTRAINT PK_CASE PRIMARY KEY (ID_CASE);
ALTER TABLE PK_CASE PRIMARY KEY ADD CHECK (ID_CASE>0);

ALTER TABLE CLIENT ADD CONSTRAINT PK_CLIENT PRIMARY KEY (ID_CLIENT);
ALTER TABLE CLIENT ADD CHECK (ID_CLIENT>0);
ALTER TABLE INSURANCE_CASE_CON ADD CONSTRAINT PK_INSURANCE_CASE_CON PRIMARY
KEY (ID_CONTRACT);
ALTER TABLE INSURANCE_CASE_CON ADD CHECK (ID_CONTRACT>0);
ALTER TABLE INSURANCE_CASE_POLIC ADD CONSTRAINT PK_INSURANCE_CASE_POLIC
PRIMARY KEY (ID_POLIC)

ALTER TABLE INSURANCE_CASE_POLIC ADD CHECK (ID_POLIC>0);
ALTER TABLE POLIC ADD PRIMARY KEY (ID_POLIC);
ALTER TABLE POLIC ADD CHECK (ID_POLIC>0);

```

```

ALTER TABLE PRICE ADD CONSTRAINT PK_PRICE PRIMARY KEY (ID_PRICE);
ALTER TABLE PRICE ADD CHECK (ID_PRICE>0);
ALTER TABLE SIGN_CONTRACT ADD PRIMARY KEY (ID_CONTRACT);
ALTER TABLE SIGN_CONTRACT ADD CHECK (ID_CONTRACT >0);
ALTER TABLE SUM ADD CONSTRAINT PK_SUM PRIMARY KEY (ID_SUM);
ALTER TABLE SUM ADD CHECK (ID_SUM>0);
ALTER TABLE TERM ADD CONSTRAINT PK_TERM PRIMARY KEY (ID_TERM);
ALTER TABLE TERM ADD CHECK (ID_TERM >0);
ALTER TABLE TYPE ADD CONSTRAINT PK_TYPE PRIMARY KEY (ID_TYPE);
ALTER TABLE TYPE ADD CHECK (ID_TYPE >0);

/*****FOREIGN KEYES*****/
ALTER TABLE CLIENT ADD CONSTRAINT FK_CLIENT_1 FOREIGN KEY (ADDRESS) REFERENCES
ADDRESS (ID_ADDRESS);

ALTER TABLE SIGN_CONTRACT ADD CONSTRAINT FK_SIGN_CONTRACT_1 FOREIGN KEY
(CLIENT) REFERENCES CLIENT (ID_CLIENT);

ALTER TABLE SIGN_CONTRACT ADD CONSTRAINT FK_SIGN_CONTRACT_2 FOREIGN KEY
(POLIC) REFERENCES POLIC (ID_POLIC);

ALTER TABLE SIGN_CONTRACT ADD CONSTRAINT FK_SIGN_CONTRACT_3 FOREIGN KEY
("CASE") REFERENCES INSURANCE_CASE_CON (ID_CONTRACT);

ALTER TABLE SIGN_CONTRACT ADD CONSTRAINT FK_SIGN_CONTRACT_4 FOREIGN KEY
("TERM") REFERENCES "TERM" (ID_TERM);

ALTER TABLE SIGN_CONTRACT ADD CONSTRAINT FK_SIGN_CONTRACT_5 FOREIGN KEY
(PRICE) REFERENCES PRICE (ID_PRICE);

ALTER TABLE SIGN_CONTRACT ADD CONSTRAINT FK_SIGN_CONTRACT_6 FOREIGN KEY
("TYPE") REFERENCES "TYPE" (ID_TYPE);

ALTER TABLE POLIC ADD CONSTRAINT FK_POLIC_1 FOREIGN KEY (INSURANCE_CASE)
REFERENCES INSURANCE_CASE_POLIC (ID_POLIC);

ALTER TABLE POLIC ADD CONSTRAINT FK_POLIC_2 FOREIGN KEY ("TYPE") REFERENCES
"TYPE" (ID_TYPE);

ALTER TABLE POLIC ADD CONSTRAINT FK_POLIC_3 FOREIGN KEY ("TERM") REFERENCES
"TERM" (ID_TERM);

ALTER TABLE INSURANCE_CASE_CON ADD CONSTRAINT FK_INSURANCE_CASE_CON_1 FOREIGN
KEY ("CASE") REFERENCES "CASE" (ID_CASE);

ALTER TABLE INSURANCE_CASE_POLIC ADD CONSTRAINT FK_INSURANCE_CASE_POLIC_1
FOREIGN KEY ("CASE") REFERENCES "CASE" (ID_CASE);

```

Первичный ключ - PRIMARY KEY.

Внешний ключ - FOREIGN KEY.

Скрипт, заполняющий некоторые таблицы БД осмысленными данными приведен в листинге 2.

Листинг 2

```
CONNECT 'D:\school\base\turfirma.fdb' user 'SYSDBA' password 'masterkey';

INSERT INTO ADRESS VALUES(1, "SAINT - PETERSBURG", "PROS-POPOVA UL",43,321);
INSERT INTO ADRESS VALUES(2, "SAINT - PETERSBURG", "YEVSKIY",12,211);
INSERT INTO ADRESS VALUES(3, "SAINT - PETERSBURG", "KAZANCKAYA",32,642);
INSERT INTO ADRESS VALUES(4, "SAINT - PETERSBURG", "GRAZHDANSKIY",30,123);

INSERT INTO CASE VALUES(12,"CAR");
INSERT INTO CASE VALUES(12,"PROPERTY");
INSERT INTO CASE VALUES(13,"MEDICINE");
INSERT INTO CASE VALUES(14,"TRAVELS");
INSERT INTO CASE VALUES(15,"ACCIDENT");

INSERT INTO CLIENT VALUES(1,"WAYNER ROONEY",08.02.1987,4);
INSERT INTO CLIENT VALUES(1,"NGUYEN TIEN VU",08.04.1993,3);
INSERT INTO CLIENT VALUES(1,"DAVID BECKHAM",23.12.1987,2);
INSERT INTO CLIENT VALUES(1,"TIGER GOOD",11.12.1977,2);
INSERT INTO CLIENT VALUES(1,"ANDREI ARSHAVIN",29.5.1981,1);

INSERT INTO INSURANCE_CASE_CON VALUES(1,12,32);
INSERT INTO INSURANCE_CASE_CON VALUES(2,12,32);
INSERT INTO INSURANCE_CASE_CON VALUES(3,13,33);
INSERT INTO INSURANCE_CASE_CON VALUES(4,14,34);
INSERT INTO INSURANCE_CASE_CON VALUES(5,15,35);

INSERT INTO INSURANCE_CASE_POLIC VALUES(1,11);
INSERT INTO INSURANCE_CASE_POLIC VALUES(2,12);
INSERT INTO INSURANCE_CASE_POLIC VALUES(3,13);
INSERT INTO INSURANCE_CASE_POLIC VALUES(4,14);
INSERT INTO INSURANCE_CASE_POLIC VALUES(5,15);

INSERT INTO POLIC VALUES(1,51,42,"NORMAL");
INSERT INTO POLIC VALUES(2,52,41,"INJURY");
INSERT INTO POLIC VALUES(3,52,42,"GOOD");
INSERT INTO POLIC VALUES(4,52,42,"INVALID");

INSERT INTO PRICE VALUES(1,"5000 RUB");
INSERT INTO PRICE VALUES(2,"7000 RUB");

INSERT INTO SIGN_CONTRACT VALUES(123456,1234,1,2,22,42,52);
INSERT INTO SIGN_CONTRACT VALUES(123457,1235,3,3,22,42,52);
INSERT INTO SIGN_CONTRACT VALUES(123458,1236,3,4,22,42,52);
INSERT INTO SIGN_CONTRACT VALUES(123459,1238,1,2,22,42,51);
INSERT INTO SIGN_CONTRACT VALUES(123468,1237,4,5,21,41,51);

INSERT INTO TERM VALUES (41,"SHORT TIME");
INSERT INTO TERM VALUES (42,"LONG TIME");
INSERT INTO TYPE VALUES(51,"COMPULSORY INSURANCE");
INSERT INTO TYPE VALUES(52,"VOLUNTARY INSURANCE");
```

Модифицировать схему БД для удовлетворения следующим условиям:

Добавить учет максимальных выплат по каждому типу страхового случая для контракта.

```
CREATE TABLE "SUM" (  
ID_SUM INTEGER NOT NULL,  
SUM_PAY VARCHAR(50)  
);
```

```
ALTER TABLE INSURANCE_CASE_CON ADD CONSTRAINT FK_INSURANCE_CASE_CON_2 FOREIGN  
KEY ("SUM") REFERENCES "SUM" (ID_SUM);
```

```
INSERT INTO SUM (31, "50.000 RUB");  
INSERT INTO SUM (32, "80.000 RUB");  
INSERT INTO SUM (33, "120.000 RUB");  
INSERT INTO SUM (34, "220.000 RUB");  
INSERT INTO SUM (35, "350.000 RUB");
```

С помощью **Database Designer** в среде IBExpert была получена ER-диаграмма созданной БД (Рис.1). Она полностью совпадает с диаграммой созданной в предыдущей работе.

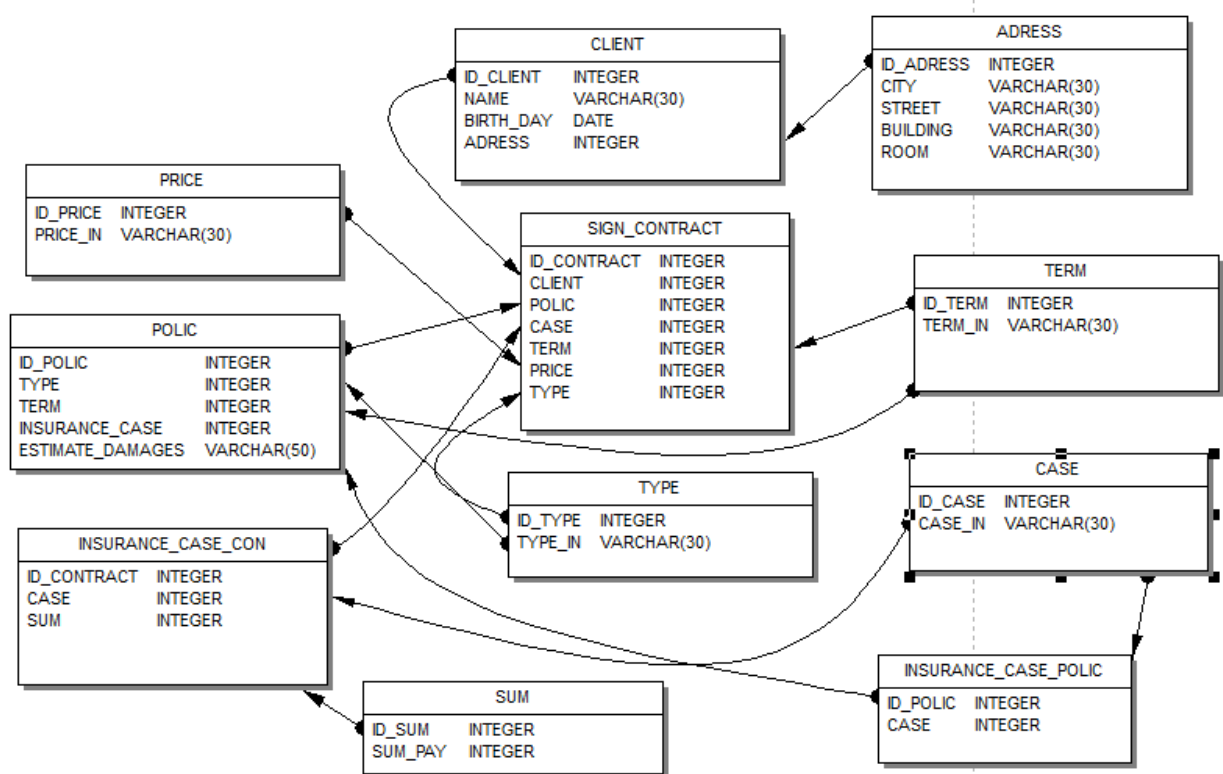


Рис. 1. ER-диаграмма, соответствующая страховой компании

4. Выводы:

SQL-DDL (Data Definition Language) - предназначен для описания данных на разных уровнях абстракции: внешнем, логическом и внутреннем. Группа команд позволяет создавать данные, создавать, изменять и уничтожать БД и записи внутри них.

Включает в себя множество операторов, основные из которых являются: **CREATE TABLE**, **DROP TABLE** и **ALTER TABLE** предназначенные для создания, удаления и изменения таблиц соответственно. 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. Также можно отметить что тексты SQL-запросов, написанные на SQL DDL могут быть достаточно легко перенесены из одной СУБД в другую.

В результате выполнения лабораторной работы было получено два скрипта:

- создающий БД со всеми связями и ограничениями
- заполняющий все таблицы БД

Также были изучены некоторые возможности среды IBExpert — автоматическое создание ER-диаграммы (**Database Designer**), автоматическое генерирование данных таблиц (**Test data generator**) и получение скриптов для автозаполнения таблиц данными (**Extract Metadata**).