

Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**ОТЧЕТ**  
**по лабораторной работе**

**«Язык SQL-DDL»**

Базы данных

**Работу выполнил студент**

группа 43501/3      Крылов И.С.

**Работу принял преподаватель**

\_\_\_\_\_ Мясов А.В.

Санкт-Петербург

2019

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Программа работы</b>	<b>3</b>
<b>3</b>	<b>Теоретическая информация</b>	<b>3</b>
<b>4</b>	<b>Выполнение работы</b>	<b>4</b>
4.1	Структура базы данных . . . . .	4
4.2	Скрипт создания структуры базы данных . . . . .	4
4.3	Скрипт заполнения таблиц тестовыми данными . . . . .	6
4.4	ИНдивидуальное задание . . . . .	7
<b>5</b>	<b>Выводы</b>	<b>7</b>

## Цель работы

Познакомиться 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. Демонстрация скрипта преподавателю.
3. Создание скрипта, заполняющего все таблицы БД данными.
4. Выполнение SQL-запросов, изменяющих схему созданной БД по заданию преподавателя. Демонстрация их работы преподавателю.

## Теоретическая информация

**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. Функции SQL-DML определяются первым словом в предложении (часто называемом запросом), которое является глаголом: **SELECT** («выбрать»), **INSERT** («вставить»), **UPDATE** («обновить»), и **DELETE** («удалить»).

# Выполнение работы

## Структура базы данных

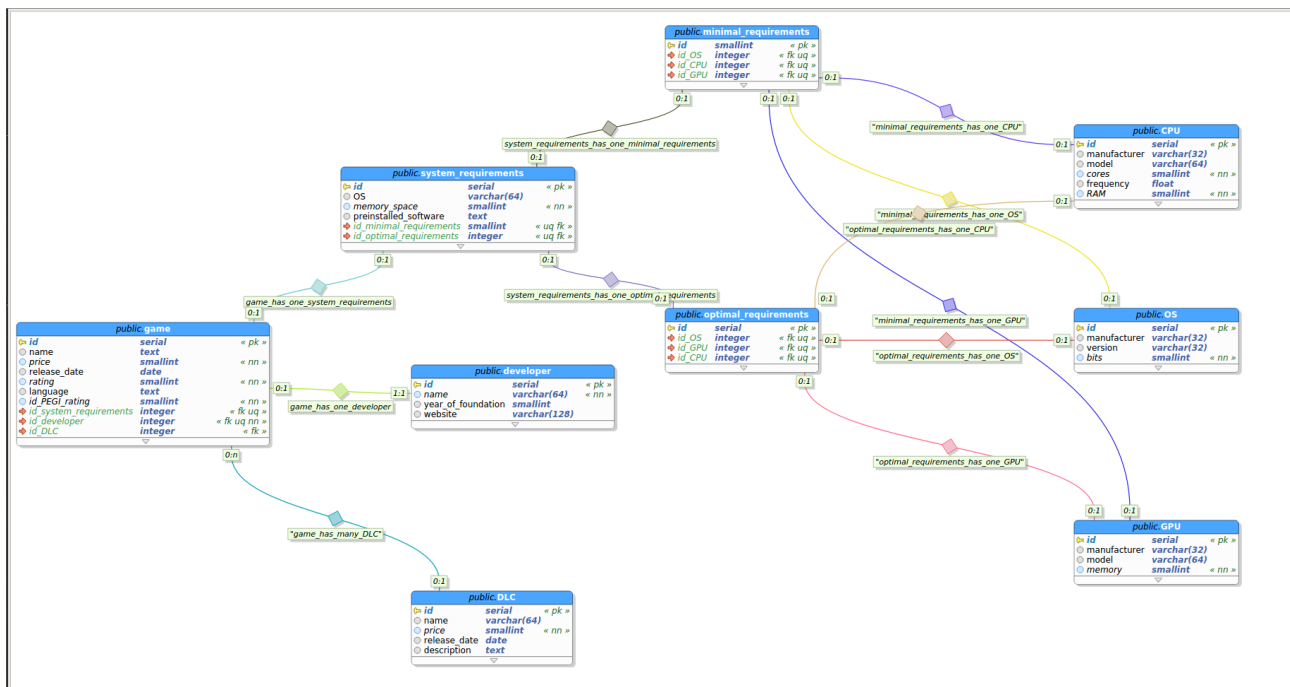


Рис. 4.1: Структура базы данных

## Скрипт создания структуры базы данных

```
1 CREATE TABLE IF NOT EXISTS OS(  
2     id_OS serial PRIMARY KEY,  
3     version varchar(32),  
4     bits smallint NOT NULL ,  
5     manufacturer varchar(32)  
6 );  
7  
8 CREATE TABLE IF NOT EXISTS CPU(  
9     id_CPU serial PRIMARY KEY,  
10    manufacturer varchar(32) NOT NULL ,  
11    model varchar(64),  
12    cores smallint NOT NULL ,  
13    frequency smallint ,  
14    RAM smallint NOT NULL  
15 );  
16 CREATE TABLE IF NOT EXISTS tournament(  
17     id_tournament serial PRIMARY KEY ,  
18     name varchar(5),  
19     rules text ,  
20     date_from date ,  
21     date_to date ,  
22     xprize smallint NOT NULL  
23 );
```

```

24 CREATE TABLE IF NOT EXISTS GPU(
25     id_GPU serial PRIMARY KEY,
26     manufacturer varchar(32) NOT NULL,
27     model varchar(64),
28     memory smallint NOT NULL
29 );
30 CREATE TABLE IF NOT EXISTS developer(
31     id_developer serial PRIMARY KEY,
32     name varchar(50),
33     year_of_foundation smallint,
34     website varchar(128)
35 );
36
37 CREATE TABLE IF NOT EXISTS minimal_requirements(
38     id_minimal_requirements serial PRIMARY KEY,
39     id_os integer,
40     id_cpu integer,
41     id_gpu integer,
42     FOREIGN KEY (id_OS) REFERENCES OS(id_OS),
43     FOREIGN KEY (id_CPU) REFERENCES CPU(id_CPU),
44     FOREIGN KEY (id_GPU) REFERENCES GPU(id_GPU)
45 );
46
47 CREATE TABLE IF NOT EXISTS optimal_requirements(
48     id_optimal_requirements serial PRIMARY KEY,
49     id_os integer,
50     id_cpu integer,
51     id_gpu integer,
52     FOREIGN KEY (id_OS) REFERENCES OS(id_OS),
53     FOREIGN KEY (id_CPU) REFERENCES CPU(id_CPU),
54     FOREIGN KEY (id_GPU) REFERENCES GPU(id_GPU)
55 );
56
57
58 CREATE TABLE IF NOT EXISTS system_requirements(
59     id_system_requirements serial PRIMARY KEY,
60     OS varchar(64) NOT NULL,
61     memory_space smallint NOT NULL,
62     id_minimal_requirements integer,
63     --id_preinstalled_software integer,-
64     id_optimal_requirements integer,
65     FOREIGN KEY (id_minimal_requirements) REFERENCES
minimal_requirements(id_minimal_requirements),
66     FOREIGN KEY (id_optimal_requirements) REFERENCES
optimal_requirements(id_optimal_requirements)
67 );
68 CREATE TABLE IF NOT EXISTS preinstalled_software(
69     id_preinstalled_software serial PRIMARY KEY,
70     name varchar(32),
71     instructions text,
72     download_link varchar(32),
73     id_system_requirements integer,

```

```

74     FOREIGN KEY (id_system_requirements) REFERENCES
      system_requirements(id_system_requirements)
75 );
76 CREATE TABLE IF NOT EXISTS game(
77     id_game serial PRIMARY KEY ,
78     name text,
79     price smallint NOT NULL ,
80     release_date date,
81     rating smallint NOT NULL,
82     language text,
83     id_developer integer,
84     id_system_requirements integer,
85     id_tournament integer,
86     FOREIGN KEY (id_developer) REFERENCES developer(id_developer)
      ,
87     FOREIGN KEY (id_system_requirements) REFERENCES
      system_requirements(id_system_requirements),
88     FOREIGN KEY (id_tournament) REFERENCES tournament(
      id_tournament),
89     PEGI_rating smallint NOT NULL
90 );
91 );
92 CREATE TABLE IF NOT EXISTS DLC(
93     id_DLC serial PRIMARY KEY,
94     name varchar(50),
95     price smallint NOT NULL ,
96     release_date date,
97     description text,
98     id_game integer,
99     FOREIGN KEY (id_game) REFERENCES game(id_game)
100 );

```

## Скрипт заполнения таблиц тестовыми данными

```

1 INSERT INTO game
2 VALUES (DEFAULT, 'Red Dead Redemption', 1999, '01-01-2018', 8, '
   english', 18, 1, 1, 1, 1);
3
4 INSERT INTO system_requirements
5 VALUES (DEFAULT, 'Windows', 73, 1, 1, 1);
6
7 INSERT INTO developer
8 VALUES (DEFAULT, 'rockstar games' , '1990', 'rockstargames.com');
9
10 INSERT INTO DLC
11 VALUES (DEFAULT, 'Rancho shooting', 395, '02-02-2018', '
   Дополнительные карты с сайджвестами-');
12
13 INSERT INTO minimal_requirements
14 VALUES (DEFAULT, 1, 1, 1);
15

```

```

16 INSERT INTO optimal_requirements
17 VALUES (DEFAULT, 1, 2, 2);
18
19 INSERT INTO CPU
20 VALUES (DEFAULT, 'Intel', 'i3', 4, 1.7, 1),
21         (DEFAULT, 'Intel', 'i7', 8, 2.4, 2);
22
23 INSERT INTO OS
24 VALUES (DEFAULT, 'Windows', '7', 64);
25
26 INSERT INTO GPU
27 VALUES (DEFAULT, 'Nvidia', 'GeForce GTX1080', 2),
28         (DEFAULT, 'Nvidia', 'RTX 2070', 8);
29
30 INSERT INTO preinstalled_software
31 VALUES (DEFAULT, 'DirectX 13', 'https://www.microsoft.com/en-us/
    download/details.aspx?id=35', 'Follow the instructions of
    install master');
32
33 INSERT INTO tournament
34 VALUES (DEFAULT, 'Cowboy ride', '[2018-12-31 22:00, 2019-03-01
    22:00]', , 0);

```

## ИНдивидуальное задание

По заданию от преподавателя база данных была дополнена двумя таблицами preinstalled\_software - для учёта необходимо ПО, и tournament - для учёта турниров.

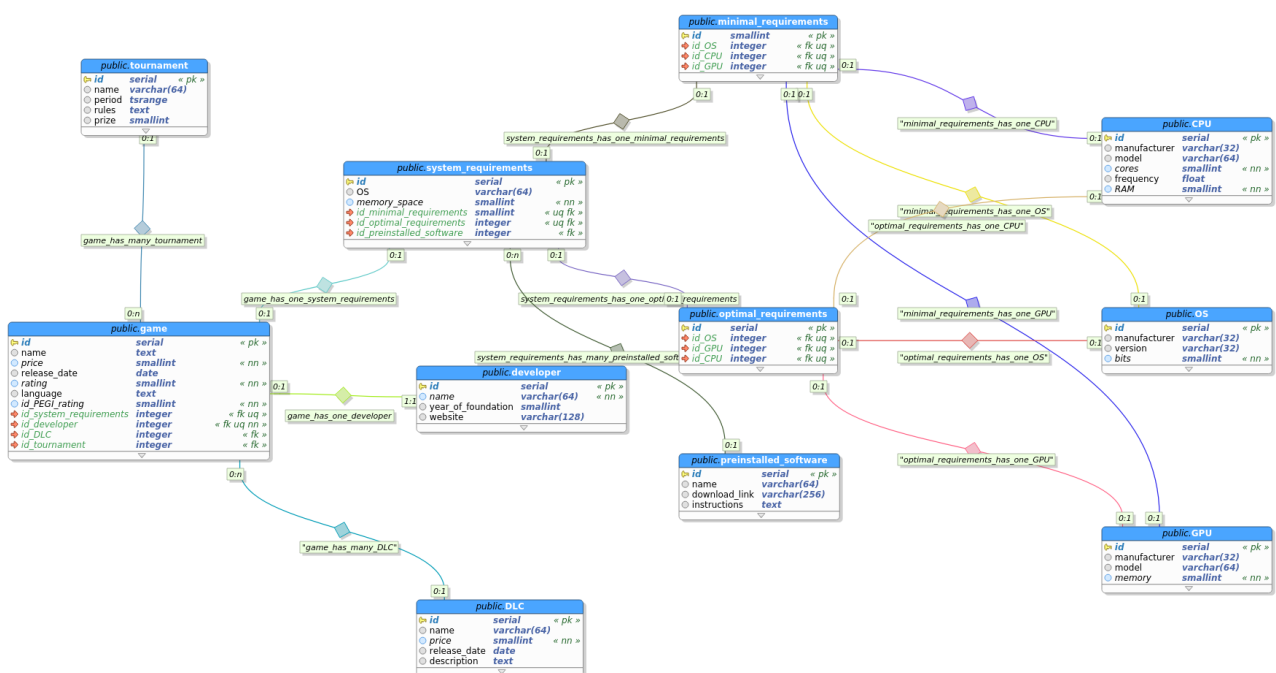


Рис. 4.2: Измененная структура базы данных

## Выводы

В ходе выполнения лабораторной работы были изучены основы создания скриптов на языке SQL. С помощью SQL-DDL были описаны структуры хранимой в баз данных информации. С использованием SQL-DML созданные структуры были заполнены конкретными данными.