

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 Дьячков В.В.

Работу принял преподаватель

_____ Мясов А.В.

Санкт-Петербург

2018

Содержание

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. 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.

3. 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»).

4. Выполнение работы

4.1. Структура базы данных

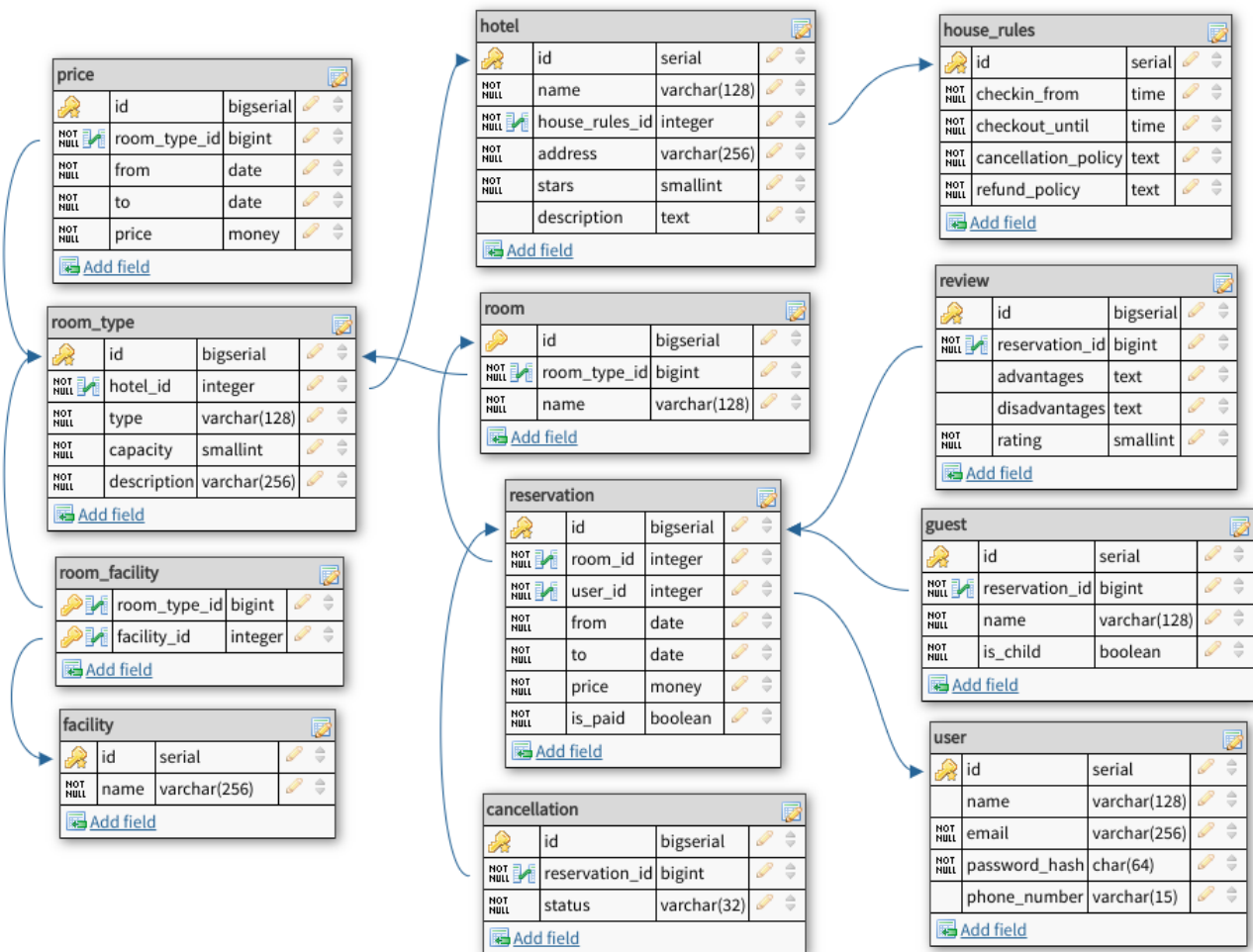


Рис. 4.1: Структура базы данных

4.2. Скрипт создания структуры базы данных

```

1 CREATE TABLE IF NOT EXISTS house_rules (
2     id SERIAL PRIMARY KEY,
3     checkin_from TIME NOT NULL,
4     checkout_until TIME NOT NULL,
5     cancellation_policy TEXT NOT NULL
6 );
7
8 CREATE TABLE IF NOT EXISTS hotel (
9     id SERIAL PRIMARY KEY,
10    name VARCHAR(128) NOT NULL,
11    house_rules_id INTEGER NOT NULL,
12    address VARCHAR(256) NOT NULL,
13    stars SMALLINT NOT NULL,
14    description TEXT,
15    FOREIGN KEY (house_rules_id) REFERENCES house_rules (id)

```

```

16 );
17
18 CREATE TABLE IF NOT EXISTS "user" (
19     id                SERIAL PRIMARY KEY,
20     name              VARCHAR(128),
21     email             VARCHAR(256) NOT NULL,
22     password_hash     CHAR(64)     NOT NULL,
23     phone_number      VARCHAR(15)
24 );
25
26 CREATE TABLE IF NOT EXISTS room_type (
27     id                BIGSERIAL PRIMARY KEY,
28     hotel_id          INTEGER      NOT NULL,
29     type              VARCHAR(128) NOT NULL,
30     capacity          SMALLINT     NOT NULL,
31     description       VARCHAR(256) NOT NULL,
32     FOREIGN KEY (hotel_id) REFERENCES hotel (id)
33 );
34
35 CREATE TABLE IF NOT EXISTS room (
36     id                BIGSERIAL PRIMARY KEY,
37     room_type_id      BIGINT       NOT NULL,
38     name              VARCHAR(128) NOT NULL,
39     FOREIGN KEY (room_type_id) REFERENCES room_type (id)
40 );
41
42 CREATE TABLE IF NOT EXISTS reservation (
43     id                BIGSERIAL PRIMARY KEY,
44     room_id           BIGINT       NOT NULL,
45     user_id           INTEGER      NOT NULL,
46     "from"            DATE         NOT NULL,
47     "to"              DATE         NOT NULL,
48     price             MONEY        NOT NULL,
49     is_paid           BOOLEAN      NOT NULL,
50     FOREIGN KEY (room_id) REFERENCES room (id),
51     FOREIGN KEY (user_id) REFERENCES "user" (id)
52 );
53
54 CREATE TABLE IF NOT EXISTS guest (
55     id                BIGSERIAL PRIMARY KEY,
56     reservation_id    BIGINT       NOT NULL,
57     name              VARCHAR(128) NOT NULL,
58     is_child          BOOLEAN      NOT NULL,
59     FOREIGN KEY (reservation_id) REFERENCES reservation (id)
60 );
61
62 CREATE TABLE IF NOT EXISTS review (
63     id                BIGSERIAL PRIMARY KEY,
64     reservation_id    BIGINT       NOT NULL,
65     advantages        TEXT,
66     disadvantages     TEXT,
67     rating            SMALLINT NOT NULL,

```

```

68     FOREIGN KEY (reservation_id) REFERENCES reservation (id)
69 );
70
71 CREATE TABLE IF NOT EXISTS cancellation (
72     id BIGSERIAL PRIMARY KEY,
73     reservation_id BIGINT NOT NULL,
74     status VARCHAR(32) NOT NULL,
75     FOREIGN KEY (reservation_id) REFERENCES reservation (id)
76 );
77
78 CREATE TABLE IF NOT EXISTS price (
79     id BIGSERIAL PRIMARY KEY,
80     room_type_id BIGINT NOT NULL,
81     "from" DATE NOT NULL,
82     "to" DATE NOT NULL,
83     price MONEY NOT NULL,
84     FOREIGN KEY (room_type_id) REFERENCES room_type (id)
85 );
86
87 CREATE TABLE IF NOT EXISTS facility (
88     id SERIAL PRIMARY KEY,
89     name VARCHAR(256) NOT NULL
90 );
91
92 CREATE TABLE IF NOT EXISTS room_facility (
93     room_type_id BIGINT NOT NULL,
94     facility_id INTEGER NOT NULL,
95     CONSTRAINT room_facility_pk PRIMARY KEY (room_type_id,
96     facility_id),
97     FOREIGN KEY (room_type_id) REFERENCES room_type (id),
98     FOREIGN KEY (facility_id) REFERENCES facility (id)
99 );

```

4.3. Скрипт заполнения таблиц тестовыми данными

```

1 INSERT INTO house_rules
2 VALUES (DEFAULT, '15:00:00', '12:00:00', 'Предоплата не возвращается
   при отмене бронирования менее чем за сутки'),
3         (DEFAULT, '16:00:00', '14:00:00', 'Предоплата возвращается');
4
5 INSERT INTO hotel
6 VALUES (DEFAULT, 'Гранд Будапешт', 1, 'Республика Зубровка', 5, '
   Комфортабельный отель Гранд Будапешт');
7
8 INSERT INTO "user"
9 VALUES (DEFAULT,
10         'traveller',
11         'traveller@travel.com',
12         '5
   E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8
   ');

```

```

13
14 INSERT INTO room_type
15 VALUES (DEFAULT, 1, 'double king-size', 2, 'Номер для некурящих с 2
    кроватями размера king-size'),
16         (DEFAULT, 1, 'single', 1, 'Номер для некурящих с 1 кроватью');
17
18 INSERT INTO room
19 VALUES (DEFAULT, 1, '№123'),
20         (DEFAULT, 2, '№456'),
21         (DEFAULT, 1, '№789');
22
23 INSERT INTO reservation
24 VALUES (DEFAULT, 1, 1, '2018-10-08', '2018-10-10', 1234, TRUE),
25         (DEFAULT, 2, 1, '2018-10-10', '2018-10-15', 4321, FALSE);
26
27 INSERT INTO guest
28 VALUES (DEFAULT, 1, 'M. Gustave', FALSE),
29         (DEFAULT, 1, 'Mr. Moustafa', FALSE),
30         (DEFAULT, 2, 'Serge X.', FALSE);
31
32 INSERT INTO review
33 VALUES (DEFAULT, 1, 'Все хорошо', 'Все плохо', 3),
34         (DEFAULT, 2, 'Все плохо', 'Все хорошо', 5);
35
36 INSERT INTO price
37 VALUES (DEFAULT, 1, '2018-10-01', '2018-10-31', 1234),
38         (DEFAULT, 1, '2018-09-01', '2018-09-30', 1000),
39         (DEFAULT, 2, '2018-10-01', '2018-10-31', 4321);
40
41 INSERT INTO facility
42 VALUES (DEFAULT, 'Wi-Fi'),
43         (DEFAULT, 'Холодильник'),
44         (DEFAULT, 'TV'),
45         (DEFAULT, 'Душ');
46
47 INSERT INTO room_facility
48 VALUES (1, 2),
49         (1, 3),
50         (1, 4),
51         (2, 1),
52         (2, 4);

```

4.4. Изменение структуры базы данных

По заданию преподавателя схема БД была изменена для удовлетворения следующим требованиям:

- Добавить связь со странами и городами
- Добавить учет бонусов и штрафов с учетом условий бронирования

Были добавлены следующие таблицы:

- **country** – хранит 3-символьный код **id** и название **name** страны.
- **city** – хранит название **name** и 3-символьный код страны **country_id**.
- **bonus_penalty** – хранит условие предоставления бонуса или причину штрафа **condition** и соответствующий денежный бонус/штраф **price**.
- **bonus_penalty_reservation** – соотносит бронирование **reservation_id** с бонусами/штрафами **bonus_penalty_id**.

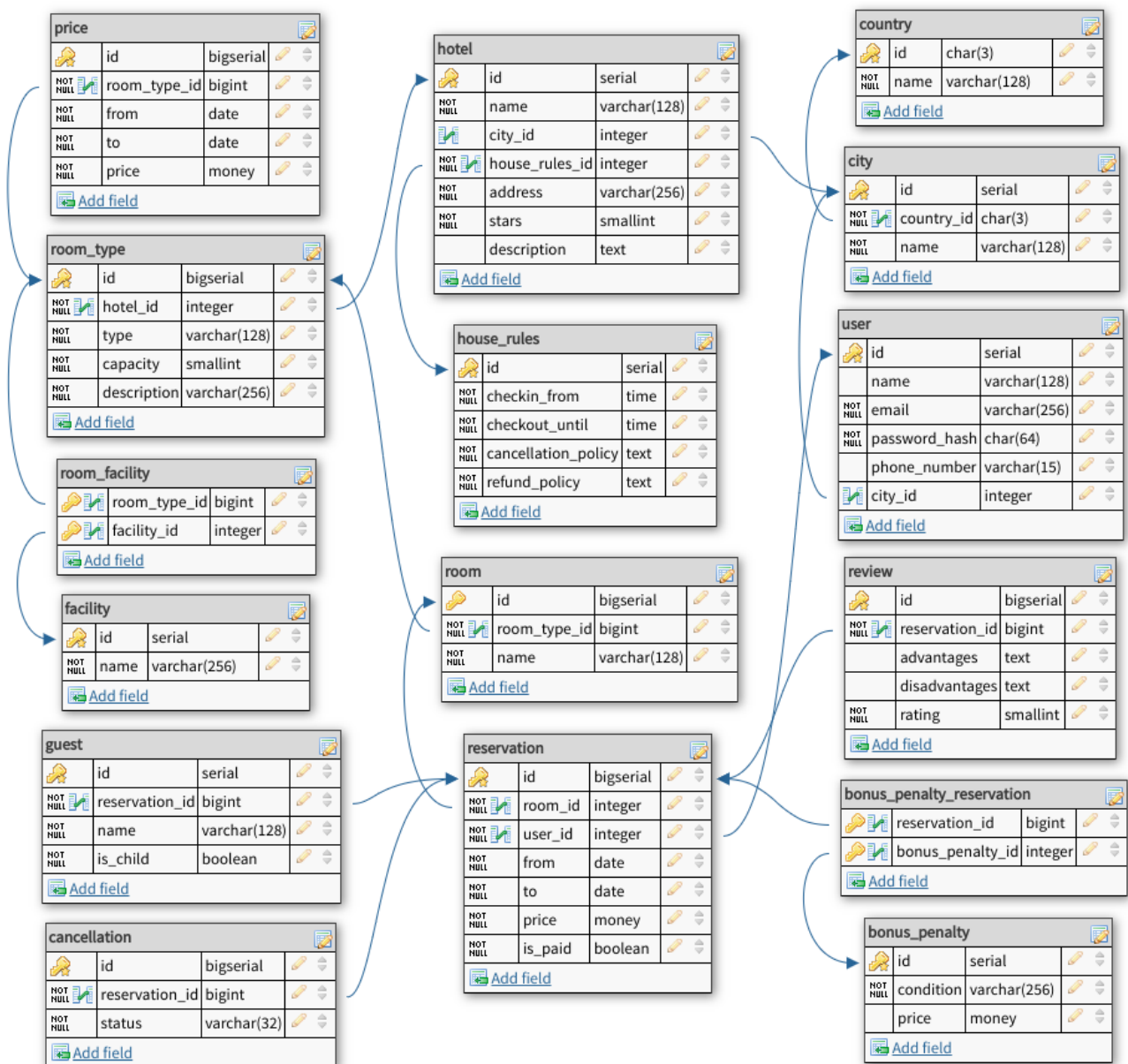


Рис. 4.2: Структура базы данных

Скрипт создания дополнительных таблиц, внесения в них тестовых данных, и изменения существующих:

```

1 CREATE TABLE IF NOT EXISTS country (
2     id    CHAR(3) PRIMARY KEY,
3     name  VARCHAR(128)

```



```

4 );
5
6 CREATE TABLE IF NOT EXISTS city (
7     id          SERIAL PRIMARY KEY,
8     country_id  CHAR(3),
9     name        VARCHAR(128),
10    FOREIGN KEY (country_id) REFERENCES country (id)
11 );
12
13 CREATE TABLE IF NOT EXISTS bonus_penalty (
14     id          SERIAL PRIMARY KEY,
15     condition   VARCHAR(256),
16     price       MONEY
17 );
18
19 CREATE TABLE IF NOT EXISTS bonus_penalty_reservation (
20     bonus_penalty_id INTEGER,
21     reservation_id   BIGINT,
22     FOREIGN KEY (bonus_penalty_id) REFERENCES bonus_penalty (id),
23     FOREIGN KEY (reservation_id) REFERENCES reservation (id)
24 );
25
26 ALTER TABLE IF EXISTS "user"
27     ADD COLUMN city_id INTEGER;
28 ALTER TABLE IF EXISTS "user"
29     ADD FOREIGN KEY (city_id) REFERENCES city (id);
30
31 ALTER TABLE IF EXISTS hotel
32     ADD COLUMN city_id INTEGER;
33 ALTER TABLE IF EXISTS hotel
34     ADD FOREIGN KEY (city_id) REFERENCES city (id);

```

5. Выводы

В ходе выполнения данной работы были изучены основы создания скриптов на языке SQL. С помощью SQL-DDL описаны структуры разрабатываемой схемы базы данных. С использованием SQL-DML созданные структуры заполнены тестовыми данными. Изучен синтаксис обновления структуры существующей таблицы.