

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №2

по дисциплине: Базы данных

тема: «Создание объектов базы данных в СУБД»

Выполнил: ст. группы ПВ-201
Барышникова Варвара Дмитриевна

Проверил:
Кулешова Екатерина Анатольевна

Белгород 2022 г.

Лабораторная работа №2

Создание объектов базы данных в СУБД

Цель работы: изучить основные возможности языка SQL для создания структуры базы данных. Научиться создавать базы данных, таблицы, связи, ограничения, а также создавать, изменять и удалять данные.

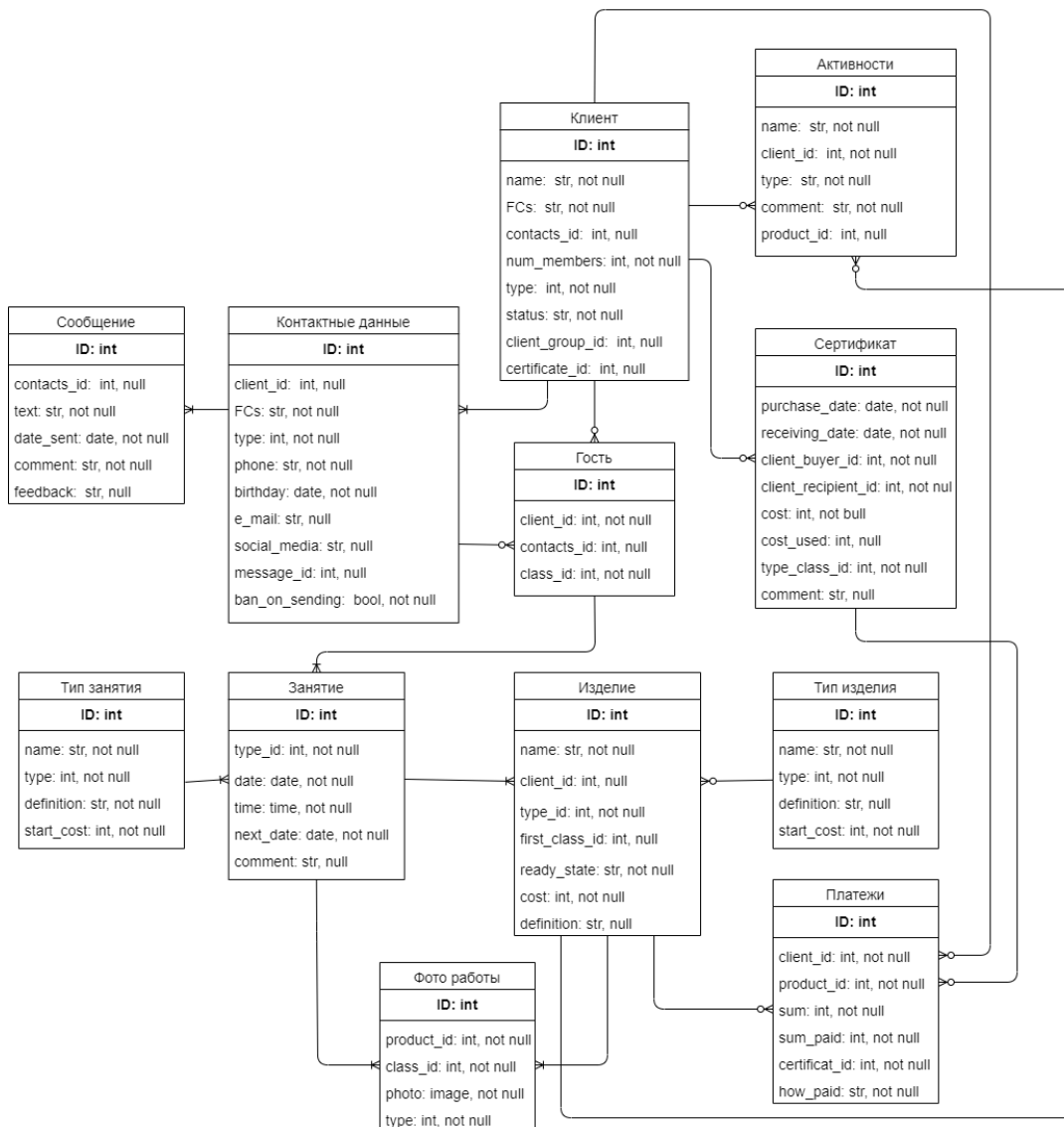
Задание к работе

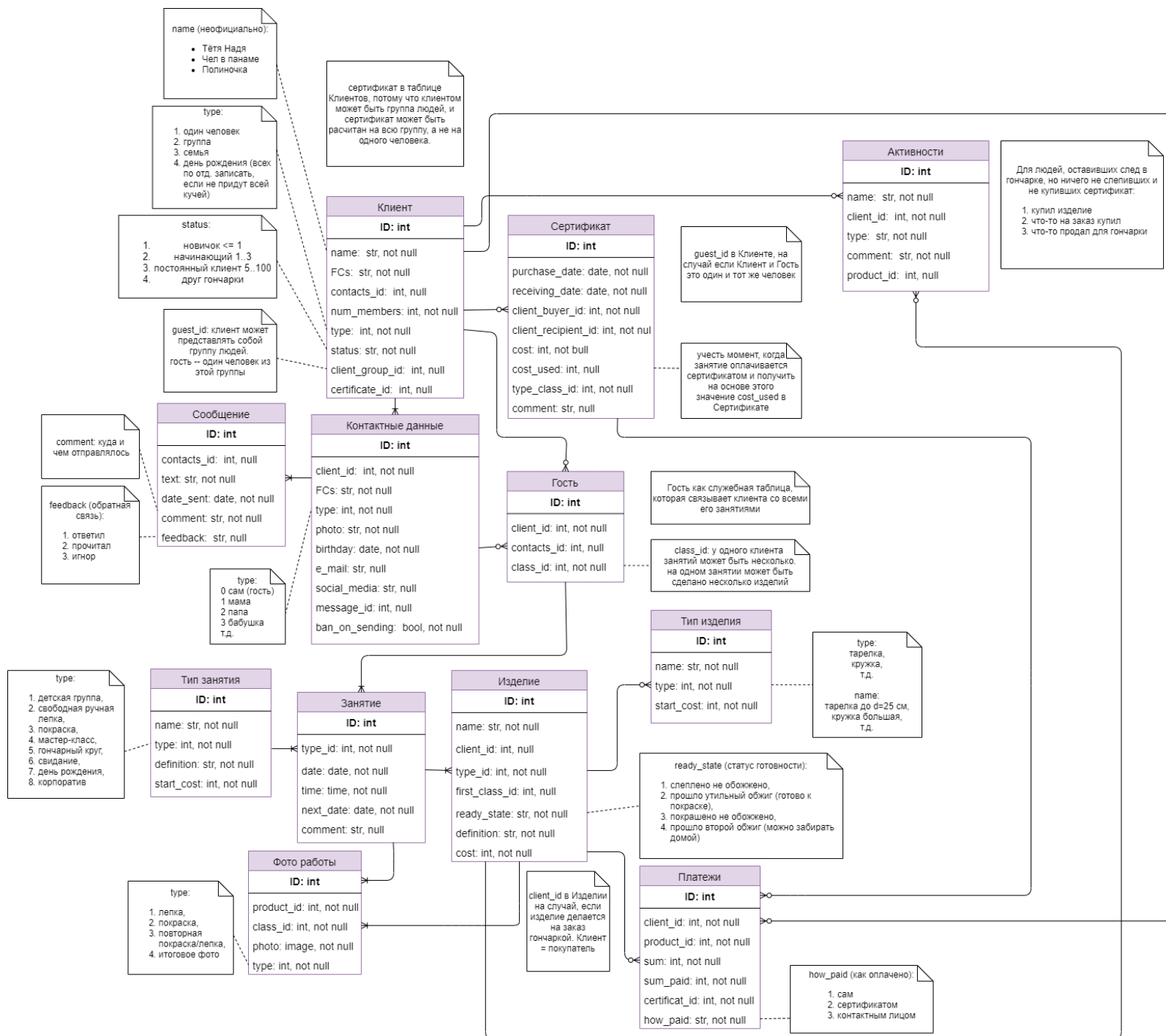
1. Составить SQL-запросы для создания структуры базы данных, полученной в результате лабораторной работы №1. Указать используемые типы данных, ограничения значений полей; для связей: действия с записями подчинённой таблицы при удалении и изменении соответствующей записи главной таблицы.

2. С помощью SQL-запросов выполнить добавление 3–4 записей в каждую таблицу, изменение и удаление нескольких записей.

Выполнение работы:

Предметная область «Гончарная студия».





1. Составить SQL-запросы для создания структуры базы данных, полученной в результате лабораторной работы №1. Указать используемые типы данных, ограничения значений полей; для связей: действия с записями подчинённой таблицы при удалении и изменении соответствующей записи главной таблицы.

Создание БД:

```
create database if not exists len_ceramic;
```

Создание таблиц:

```
create table if not exists contacts
```

```
(
    id                bigserial primary key,
    client_id         integer,
    first_name        varchar(40) not null,
    last_name         varchar(40) not null,
    middle_name       varchar(40),
    type              integer      not null,
    phone             varchar(20) not null,
    birthday          date,
    email             varchar(50),
    social_media       varchar(150),
    ban_on_sending    bool         not null,
    message_id        integer
);
```

```
create table if not exists messages
```

```
(
    id                bigserial primary key,
    contacts_id       integer      not null,
    text              varchar(200) not null,
    date_sent         date         not null,
    comment           varchar(100) not null,
    feedback          integer      not null default 0,

    foreign key (contacts_id) references contacts (id) on delete restrict on update
    restrict
);
```

```
create table if not exists class_types
```

```
(
    id                bigserial primary key,
    name              varchar(60)  not null,
    type              integer      not null,
    definition         varchar(500) not null,
    start_cost        integer      not null
);
```

```
create table if not exists classes
```

```
(
    id                bigserial primary key,
    type_id           integer not null,
    date              date     not null,
    time              time     not null,
    next_date         date     not null,
    comment           varchar(100),

    foreign key (type_id) references class_types (id) on delete restrict on update
    restrict
);
```

```
create table if not exists certificates
```

```
(
    id                bigserial primary key,
```

```

    purchase_date      date      not null,
    receiving_date      date      not null,
    client_buyer_id     integer not null,
    client_recipient_id integer not null,
    cost                integer not null,
    cost_used           integer,
    type_class_id       integer not null,
    comment              varchar(100),

    foreign key (type_class_id) references class_types (id) on delete restrict on
update restrict
);

create table if not exists clients
(
    id                bigserial primary key,
    name              varchar(100) not null,
    FCs               varchar(120),
    contacts_id       integer,
    num_members       integer      not null,
    type              integer      not null,
    client_group_id   integer,
    certificate_id     integer,
    status            integer,

    foreign key (contacts_id) references contacts (id) on delete restrict on update
restrict,
    foreign key (client_group_id) references clients (id) on delete restrict on update
restrict,
    foreign key (certificate_id) references certificates (id) on delete restrict on
update restrict
);

create table if not exists guests
(
    id                bigserial not null primary key,
    client_id         integer      not null,
    contacts_id       integer      not null,
    class_id          integer      not null,

    foreign key (client_id) references clients (id) on delete restrict on update
restrict,
    foreign key (contacts_id) references contacts (id) on delete restrict on update
restrict
);

create table if not exists product_types
(
    id                bigserial primary key,
    name              varchar(60) not null,
    type              integer      not null,
    definition         varchar(400),
    start_cost        integer      not null
);

create table if not exists products
(
    id                bigserial primary key,
    name              varchar(60) not null,
    client_id         integer,
    type_id           integer      not null,
    first_class_id    integer,
    ready_state       integer      not null,
    cost              integer      not null,
    definition         varchar(400),

    foreign key (client_id) references clients (id) on delete restrict on update
restrict,
    foreign key (type_id) references product_types (id) on delete restrict on update
restrict,
    foreign key (first_class_id) references classes (id) on delete restrict on update

```

```

restrict
);

create table if not exists classes_photos
(
    id            bigserial primary key,
    product_id integer      not null,
    class_id     integer      not null,
    photo        varchar(100) not null,
    type         int          not null,

    foreign key (product_id) references products (id) on delete restrict on update
    restrict,
    foreign key (class_id) references classes (id) on delete restrict on update
    restrict
);

create table if not exists payments
(
    id            bigserial primary key,
    client_id     integer not null,
    product_id    integer not null,
    sum           integer not null,
    sum_paid      integer not null default 0,
    certificate_id integer,
    how_paid      integer not null,

    foreign key (client_id) references clients (id) on delete restrict on update
    restrict,
    foreign key (product_id) references products (id) on delete restrict on update
    restrict,
    foreign key (certificate_id) references certificates (id) on delete restrict on
    update restrict
);

create table if not exists activities
(
    id            bigserial primary key,
    name          varchar(60)  not null,
    client_id     integer      not null,
    type          int          not null,
    comment       varchar(100) not null,
    product_id integer,

    foreign key (client_id) references clients (id) on delete restrict on update
    restrict,
    foreign key (product_id) references products (id) on delete restrict on update
    restrict
);

alter table contacts
    add foreign key (client_id) references clients (id) on delete restrict on update
    restrict,
    add foreign key (message_id) references messages (id) on delete restrict on update
    restrict;

alter table certificates
    add foreign key (client_buyer_id) references clients (id) on delete restrict on
    update restrict,
    add foreign key (client_recipient_id) references clients (id) on delete restrict
    on update restrict;

```

2. С помощью SQL-запросов выполнить добавление 3–4 записей в каждую таблицу, изменение и удаление нескольких записей.

Запросы создания записей для каждой таблицы:

Для таблицы *contacts*

```
insert into contacts
(first_name, last_name, middle_name, type, phone, email, ban_on_sending)
values ('Варвара', 'Барышникова', 'Дмитриевна', 1, '89056781243',
'barvarvaria@gmail.com', false);

insert into contacts
(first_name, last_name, middle_name, type, phone, birthday, email, ban_on_sending)
values ('Екатерина', 'Станогина', 'Владимировна', 1, '89065667091', '1975-10-07',
null, false);

insert into contacts
(first_name, last_name, middle_name, type, phone, birthday, email, ban_on_sending)
values ('Екатерина', 'Станогина', 'Владимировна', 1, '0', '1975-10-07', null, false);
```

Для таблицы *clients*

```
insert into clients
(name, fcs, contacts_id, num_members, type, status)
values ('Варя', 'Варвара Дмитриевна Барышникова', 1, 1, 1, 4);

insert into clients
(name, fcs, contacts_id, num_members, type, status)
values ('Катя', 'Екатерина Владимировна Станогина', 2, 1, 1, 4);

insert into clients
(name, contacts_id, num_members, type)
values ('Станогины', 2, 3, 3);
```

Для таблицы *class_types*

```
insert into class_types (name, type, definition, start_cost)
values ('Занятие за гончарным кругом', 5,
'Занятия рассчитаны, как на людей, которые впервые оказались за гончарным
кругом, так и для тех, кто владеет определенным опытом. ',
1500);
```

Для таблицы *certificates*

```
insert into certificates
(purchase_date, receiving_date, client_buyer_id, client_recipient_id, cost, cost_used,
type_class_id)
values ('2022-09-08', '2022-10-07', (select id from clients where name = 'Варя'),
(select id from clients where name = 'Катя'), 1000, 0, (select id from
class_types where type = 5));
```

Для таблицы *product_types*

```
insert into product_types (name, type, start_cost)
values ('Небольшая кружка', 1, 1000);

insert into product_types (name, type, start_cost)
values ('Большая кружка', 2, 1800);
```

Для таблицы *products*

```
insert into products (name, client_id, type_id, ready_state, cost, definition)
values ('Кружка с котятами', 4, (select id from product_types where name = 'Небольшая
кружка'), 1, 1200,
      'Кружка, слепленная во время ручной лепки');

insert into products (name, client_id, type_id, ready_state, cost, definition)
values ('Большая кружка', 1, (select id from product_types where name = 'Большая
кружка'), 4, 2000,
      'Кружка вытянута на круге одним из мастеров. Окрашена глазурями серой укывной
и бежевой эффектарной');
```

Для таблицы *class_types*

```
insert into class_types (name, type, definition, start_cost)
values ('Свободная ручная лепка', 2, 'Изделие выбирается из пожеланий гостя и лепится
неограниченное время', 100);

insert into class_types (name, type, definition, start_cost)
values ('Покраска', 3, 'Покраска изделия после его первого утильного обжига глазурями,
ангобами или красками', 0);
```

Для таблицы *classes*

```
insert into classes (type_id, date, time, next_date)
values ((select id from class_types where name = 'Свободная ручная лепка'), '2022-09-
25', '14:00:00', '2022-09-25');
```

Для таблицы *guests*

```
insert into guests (client_id, contacts_id, class_id)
values ((select id from clients where name = 'Варя'), (select contacts_id from clients
where name = 'Варя'), 1);
```

Для таблицы *messages*

```
insert into messages (contacts_id, text, date_sent, comment)
values ((select id from contacts where id = 2), 'Срок действия сертификата на занятие
составляет пол года!',
      '2022-10-14', 'сообщение на номер телефона');
```

Для таблицы *payments*

```
insert into payments (client_id, product_id, sum, sum_paid, how_paid)
values (1, 1, (select cost from products where id = 1), 1200, 1);
```

Для таблицы *classes_photos*

```
insert into classes_photos (product_id, class_id, photo, type)
values (1, 1, 'C:\len_ceramic\photos\product1\1.jpg', 1);
```

Для таблицы *activities*

```
insert into activities (name, client_id, type, comment, product_id)
values ('Покупка кружки', 1, 1, 'Покупка кружки со стенда гончарной студии',
      (select id from products where name = 'Большая кружка'));
```


Запросы на изменение данных в таблицах:

```
update contacts  
set client_id = 1  
where id = 1;
```

```
update contacts  
set client_id = 4  
where id = 2;
```

```
update clients  
set certificate_id = 1  
where name = 'Катя';
```

```
update clients  
set client_group_id = (select id from clients where name = 'Станогины')  
where status = 4;
```

Запрос на удаление данных в таблицах:

```
delete from contacts where phone = '0';
```

Вывод: В процессе выполнения лабораторной работы были получены навыки использования языка SQL для создания структуры базы данных. Изучены и использованы запросы для создания базы данных, таблиц, связей, ограничений, а также создания, изменения и удаления данных в таблицах.