Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

Лабораторная работа №3

«Реализация SQL-запросов для создания схемы базы данных и запросов на модификацию данных»

Вариант № 3 (Автостоянка)

Студент: В.С. Шевцов

Преподаватель: Д.В. Куприянова

МИНСК 2024

СОДЕРЖАНИЕ

[1 ЦЕЛЬ РАБОТЫ 3](#_Toc158036010)

[2 ПОРЯДОК ВЫПОЛНЕНИЕ РАБОТЫ 4](#_Toc158036011)

[3 ВЫПОЛНЕНИЕ РАБОТЫ 5](#_Toc158036012)

[4 ВЫВОД 22](#_Toc158036013)

# ЦЕЛЬ РАБОТЫ

В ходе выполнения лабораторной работы необходимо выполнить реализация схемы базы данных по ранее построенной реляционной схеме данных. Требуется сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД. Требуется заполнить таблицы данными с помощью оператора INSERT. Также необходимо рассмотреть простые действия, направленные на изменения структуры таблицы (переименования столбца, добавление ограничений) и реализовать их с помощью оператора ALTER TABLE.

# ПОРЯДОК ВЫПОЛНЕНИЕ РАБОТЫ

1. Создать в СУБД новую схему данных.
2. В этой новой схеме данных с помощью скрипта с запросами на языке DDL SQL реализовать таблицы, соответствующие реляционным отношениям схемы данных, полученной в лабораторной работе №2, с помощью одного (желательно) оператора CREATE TABLE для каждой таблицы.
3. Заполнить с помощью SQL-скрипта с использованием оператора INSERT таблицы строками данных для проверки правильного выбора первичных ключей и работоспособности ссылок между таблицами. Вначале заполняются мастер-таблицы (не содержат ссылок на другие таблицы). В каждую таблицы необходимо добавить 30 строк данных.
4. Рассмотреть простые действия, направленные на изменения структуры таблицы (переименования столбца, добавление ограничений) и реализовать их с помощью оператора ALTER TABLE.

# ВЫПОЛНЕНИЕ РАБОТЫ

**3.1. Создание схемы данных и таблиц в ней**

Код SQL-запроса для создания новой схемы данных и таблиц в ней:

DROP SCHEMA PUBLIC CASCADE;

CREATE SCHEMA PUBLIC;

CREATE TABLE IF NOT EXISTS public."Client"

(

"Passport number" character varying COLLATE pg\_catalog."default" NOT NULL,

"Parktime" double precision NOT NULL,

"Surname" character varying COLLATE pg\_catalog."default",

"Name" character varying COLLATE pg\_catalog."default",

"Otchestvo" character varying COLLATE pg\_catalog."default",

CONSTRAINT "Client\_pkey" PRIMARY KEY ("Passport number")

);

CREATE TABLE IF NOT EXISTS public."Auto"

(

"Length" smallint NOT NULL,

"Width" smallint NOT NULL,

"Height" smallint NOT NULL,

"Mass" smallint NOT NULL,

"Auto number" character varying COLLATE pg\_catalog."default" NOT NULL,

"Client number" character varying COLLATE pg\_catalog."default",

CONSTRAINT "Auto\_pkey" PRIMARY KEY ("Auto number"),

CONSTRAINT "Client number" FOREIGN KEY ("Client number")

REFERENCES public."Client" ("Passport number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS public."Floor"

(

"Floor number" serial NOT NULL,

"Places amount" smallint NOT NULL,

"Height" double precision NOT NULL,

CONSTRAINT "Floor\_pkey" PRIMARY KEY ("Floor number")

);

CREATE TABLE IF NOT EXISTS public."Place"

(

"Place number" serial NOT NULL,

"Length" smallint NOT NULL,

"Width" smallint NOT NULL,

"Auto number" character varying COLLATE pg\_catalog."default",

"Floor number" smallint NOT NULL,

CONSTRAINT "Place\_pkey" PRIMARY KEY ("Place number"),

CONSTRAINT "Auto number" FOREIGN KEY ("Auto number")

REFERENCES public."Auto" ("Auto number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE,

CONSTRAINT "Floor number" FOREIGN KEY ("Floor number")

REFERENCES public."Floor" ("Floor number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS public."Elevator"

(

"Elevator number" serial,

"Length" smallint NOT NULL,

"Width" smallint NOT NULL,

"Height" smallint NOT NULL,

"lift capacity" smallint NOT NULL,

CONSTRAINT "Elevator\_pkey" PRIMARY KEY ("Elevator number")

);

CREATE TABLE IF NOT EXISTS public."Table"

(

"Table number" serial,

"Place numbers" character varying,

"Place type" character varying COLLATE pg\_catalog."default",

"Floor number" smallint NOT NULL,

CONSTRAINT "Table\_pkey" PRIMARY KEY ("Table number"),

CONSTRAINT "Floor number" FOREIGN KEY ("Floor number")

REFERENCES public."Floor" ("Floor number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS public."Client\_Table"

(

"Client number" character varying COLLATE pg\_catalog."default" NOT NULL,

"Table number" smallint NOT NULL,

CONSTRAINT "Client\_table\_pkey" PRIMARY KEY ("Client number", "Table number"),

CONSTRAINT "Client number" FOREIGN KEY ("Client number")

REFERENCES public."Client" ("Passport number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE,

CONSTRAINT "Table number" FOREIGN KEY ("Table number")

REFERENCES public."Table" ("Table number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS public."Floor\_Elevator"

(

"Floor number" smallint NOT NULL,

"Elevator number" smallint NOT NULL,

CONSTRAINT "Floor\_Elevator\_pkey" PRIMARY KEY ("Floor number", "Elevator number"),

CONSTRAINT "Elevator number" FOREIGN KEY ("Elevator number")

REFERENCES public."Elevator" ("Elevator number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE,

CONSTRAINT "Floor number" FOREIGN KEY ("Floor number")

REFERENCES public."Floor" ("Floor number") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

);

В данном запросе происходит удаление существующей схемы данных, создание новой схемы данных и создание таблиц в ней. В таблицах строковые данные в столбцах хранятся с использованием типа VARCHAR, целочисленные данные с использованием типа SMALLINT. Для первичных ключей используется тип SERIAL. Для задания ограничений использовались параметры, NOT NULL (для того, чтобы в записях таблицах не было нулевых данных). Для внешних ключей используется параметр ON DELETE со значением CASCADE. В таком случае при удалении записи, на которую есть ссылки в других таблицах, записи, содержащую эту ссылку также удаляются. Также при создании таблицы используется условие IF NOT EXISTS, чтобы, если такая таблица уже существует, она не была создана заново.

**3.2 Изменение созданных таблиц**

Код SQL-запроса, отвечающего за изменения таблиц:

ALTER TABLE IF EXISTS "Auto" RENAME TO "Autos";

ALTER TABLE IF EXISTS "Client" RENAME TO "Clients";

ALTER TABLE IF EXISTS "Elevator" RENAME TO "Elevators";

ALTER TABLE IF EXISTS "Floor" RENAME TO "Floors";

ALTER TABLE IF EXISTS "Place" RENAME TO "Places";

ALTER TABLE IF EXISTS "Table" RENAME TO "Tables";

alter table "Clients" rename column "Otchestvo" to "Patronymic";

alter table "Autos" add check("Length" > 0);

alter table "Autos" drop constraint if exists "Autos\_Length\_check1";

create table if not exists "Temporary table"

(

Id serial not null,

"MockUpInt" integer

);

drop table if exists "Temporary table";

Данный SQL-запрос изменяет имена таблиц с помощью оператора RENAME, добавляется проверка с помощью оператора ADD CHECK, изменяется имя колонки с помощью оператора RENAME COLUMN, создание таблицы с помощью CREATE TABLE, удаление ограничения и таблицы с помощью операторов DROP CONSTRAINT и DROP TABLE.

**3.3 Заполнение таблицы**

Код SQL-запроса для заполнения таблиц:

INSERT INTO "Client"("Passport number", "Parktime", "Surname", "Name", "Otchestvo")

VALUES

('MB1000041', 14.4,'Romanov', 'Egor', 'Vladimirovich'),

('AP1026500', 14.6,'Dyatlov', 'Dmitriy', 'Yanovich'),

('HB1011478', 15.0,'Kirkorov', 'Vasiliy', 'Petrovich'),

('AA1024464', 8.2,'Romanov', 'Egor', 'Vitalievich'),

('OT1023281', 5.6,'Sinitsin', 'Vasiliy', 'Petrovich'),

('KI1000491', 19.3,'Kirkorov', 'Vladislav', 'Alekseevich'),

('EO1004827', 3.1,'Kirkorov', 'Vasiliy', 'Sergeevich'),

('IC1014604', 5.7,'Kirkorov', 'Vasiliy', 'Vitalievich'),

('BC1000292', 22.4,'Kirkorov', 'Ivan', 'Petrovich'),

('IM1018716', 4.2,'Romanov', 'Aleksey', 'Alekseevich'),

('HM1005447', 18.3,'Sinitsin', 'Vasiliy', 'Vitalievich'),

('CP1011538', 24.8,'Dyatlov', 'Vladislav', 'Vladimirovich'),

('KK1025667', 3.4,'Tkachev', 'Egor', 'Petrovich'),

('MO1009894', 20.8,'Kirkorov', 'Vasiliy', 'Vitalievich'),

('CC1031322', 3.1,'Petrov', 'Ivan', 'Sergeevich'),

('CK1004664', 8.1,'Sinitsin', 'Egor', 'Vitalievich'),

('PM1028253', 17.0,'Romanov', 'Vladislav', 'Sergeevich'),

('BC1027644', 16.5,'Dyatlov', 'Ivan', 'Petrovich'),

('KM1020037', 21.6,'Dyatlov', 'Ivan', 'Yanovich'),

('TB1009741', 12.4,'Dyatlov', 'Vladislav', 'Vladimirovich'),

('MI1012316', 23.2,'Petrov', 'Vasiliy', 'Vitalievich'),

('EB1001842', 21.4,'Kirkorov', 'Vladislav', 'Yanovich'),

('IP1009040', 5.8,'Petrov', 'Aleksey', 'Vitalievich'),

('IC1022648', 8.9,'Sinitsin', 'Dmitriy', 'Vladimirovich'),

('CB1015890', 1.2,'Dyatlov', 'Aleksey', 'Yanovich'),

('HC1015350', 5.9,'Sinitsin', 'Dmitriy', 'Yanovich'),

('XC1024393', 6.9,'Dyatlov', 'Egor', 'Sergeevich'),

('EB1012623', 2.2,'Romanov', 'Vladislav', 'Alekseevich'),

('XB1018756', 21.3,'Sinitsin', 'Ivan', 'Vitalievich'),

('MP1007376', 21.2,'Kirkorov', 'Ivan', 'Petrovich') ON CONFLICT DO NOTHING;

INSERT INTO "Auto" VALUES

(2112, 706, 1095, 225, '8331 MO-1', 'MB1000041'),

(2151, 778, 1032, 219, '6362 BB-3', 'AP1026500'),

(2054, 799, 1031, 187, '6137 PE-6', 'HB1011478'),

(2086, 722, 1057, 169, '9516 OC-7', 'AA1024464'),

(5114, 1843, 1632, 7096, '2493 PO-4', 'OT1023281'),

(5114, 1843, 1632, 7096, '9064 KE-2', 'KI1000491'),

(4728, 2340, 1576, 6234, '2192 XX-7', 'EO1004827'),

(5129, 2131, 2096, 7272, '3096 BO-1', 'IC1014604'),

(4606, 1871, 1640, 5436, '3178 OX-6', 'BC1000292'),

(5079, 2258, 2710, 4091, '5971 TT-1', 'IM1018716'),

(4978, 2379, 1533, 3282, '8949 EP-4', 'HM1005447'),

(6914, 2073, 1979, 3416, '1027 HO-8', 'CP1011538'),

(5993, 1928, 2417, 4365, '3274 TA-7', 'KK1025667'),

(4115, 1707, 2515, 6287, '1029 PM-2', 'MO1009894'),

(4399, 1503, 2847, 1127, '1089 HC-3', 'CC1031322'),

(5193, 1701, 2072, 5716, '9481 AH-5', 'CK1004664'),

(4323, 2320, 2230, 6581, '3333 CM-7', 'PM1028253'),

(4500, 2002, 2204, 848, '5664 MP-4', 'BC1027644'),

(6439, 2225, 2158, 3723, '2956 KA-6', 'KM1020037'),

(5649, 1738, 2692, 3229, '5103 BO-6', 'TB1009741'),

(4667, 1954, 1806, 6284, '8178 XC-6', 'MI1012316'),

(7963, 2365, 2151, 3992, '8305 MO-7', 'EB1001842'),

(5390, 2323, 2576, 6964, '2622 HH-3', 'IP1009040'),

(5533, 2173, 2118, 6568, '1342 KB-5', 'IC1022648'),

(3824, 1950, 1889, 7383, '4138 AO-4', 'CB1015890'),

(5496, 2490, 1847, 7695, '9328 MA-3', 'HC1015350'),

(6597, 2251, 2029, 1060, '2287 EK-4', 'XC1024393'),

(5426, 1668, 1953, 6838, '5598 TX-8', 'EB1012623'),

(4442, 1894, 1889, 6127, '6277 TA-1', 'XB1018756'),

(3907, 1543, 2504, 4958, '2419 XE-1', 'MP1007376') ON CONFLICT DO NOTHING;

INSERT INTO "Elevator"("Length", "Width", "Height", "lift capacity") VALUES

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000),

(5500, 2000, 2500, 2500),

(10000, 3000, 3000, 8500),

(2000, 2000, 2500, 1000) ON CONFLICT DO NOTHING;

INSERT INTO "Floor"("Places amount", "Height") VALUES

(30, 3000),

(60, 3000),

(20, 4000),

(30, 3000),

(40, 2500),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000),

(30, 3000) ON CONFLICT DO NOTHING;

INSERT INTO "Place"("Length", "Width", "Auto number", "Floor number") VALUES

(5500, 2000, '2493 PO-4', 1),

(5500, 2000, '9064 KE-2', 1),

(5500, 2000, '2192 XX-7', 1),

(5500, 2000, '3096 BO-1', 1),

(5500, 2000, '3178 OX-6', 1),

(5500, 2000, '5971 TT-1', 1),

(5500, 2000, '8949 EP-4', 1),

(5500, 2000, '1029 PM-2', 1),

(5500, 2000, '1089 HC-3', 1),

(5500, 2000, '9481 AH-5', 1),

(5500, 2000, '3333 CM-7', 1),

(5500, 2000, '5664 MP-4', 1),

(5500, 2000, '8178 XC-6', 1),

(5500, 2000, '4138 AO-4', 1),

(5500, 2000, '9328 MA-3', 1),

(5500, 2000, '5598 TX-8', 1),

(5500, 2000, '6277 TA-1', 1),

(5500, 2000, '2419 XE-1', 1),

(5500, 2000, '2622 HH-3', 1),

(10000, 3000, '1027 HO-8', 1),

(10000, 3000, '8305 MO-7', 1),

(10000, 3000, '2956 KA-6', 1),

(10000, 3000, '5103 BO-6', 1),

(10000, 3000, '3274 TA-7', 1),

(10000, 3000, '1342 KB-5', 1),

(10000, 3000, '2287 EK-4', 1),

(2500, 1000, '8331 MO-1', 1),

(2500, 1000, '6362 BB-3', 1),

(2500, 1000, '6137 PE-6', 1),

(2500, 1000, '9516 OC-7', 1) ON CONFLICT DO NOTHING;

INSERT INTO "Table"("Place numbers", "Place type", "Floor number") VALUES

('1-30', 'Small/Basic/Large', 1),

('31-90', 'Basic', 2),

('91-110', 'Basic/Large', 3),

('111-140', 'Small/Basic/Large', 4),

('141-180', 'Small/Basic', 5),

('181-210', 'Small/Basic/Large', 6),

('211-240', 'Small/Basic/Large', 7),

('241-270', 'Small/Basic/Large', 8),

('271-300', 'Small/Basic/Large', 9),

('301-330', 'Small/Basic/Large', 10),

('331-360', 'Small/Basic/Large', 11),

('361-390', 'Small/Basic/Large', 12),

('391-420', 'Small/Basic/Large', 13),

('421-450', 'Small/Basic/Large', 14),

('451-480', 'Small/Basic/Large', 15),

('481-510', 'Small/Basic/Large', 16),

('511-540', 'Small/Basic/Large', 17),

('541-570', 'Small/Basic/Large', 18),

('571-600', 'Small/Basic/Large', 19),

('601-630', 'Small/Basic/Large', 20),

('631-660', 'Small/Basic/Large', 21),

('661-690', 'Small/Basic/Large', 22),

('691-720', 'Small/Basic/Large', 23),

('721-750', 'Small/Basic/Large', 24),

('751-780', 'Small/Basic/Large', 25),

('781-810', 'Small/Basic/Large', 26),

('811-840', 'Small/Basic/Large', 27),

('841-870', 'Small/Basic/Large', 28),

('871-900', 'Small/Basic/Large', 29),

('901-930', 'Small/Basic/Large', 30) ON CONFLICT DO NOTHING;

INSERT INTO "Client\_Table" VALUES

('MB1000041', 1),

('AP1026500', 2),

('HB1011478', 3),

('AA1024464', 4),

('OT1023281', 5),

('KI1000491', 6),

('EO1004827', 7),

('IC1014604', 8),

('BC1000292', 9),

('IM1018716', 10),

('HM1005447', 11),

('CP1011538', 12),

('KK1025667', 13),

('MO1009894', 14),

('CC1031322', 15),

('CK1004664', 16),

('PM1028253', 17),

('BC1027644', 18),

('KM1020037', 19),

('TB1009741', 20),

('MI1012316', 21),

('EB1001842', 22),

('IP1009040', 23),

('IC1022648', 24),

('CB1015890', 25),

('HC1015350', 26),

('XC1024393', 27),

('EB1012623', 28),

('XB1018756', 29),

('MP1007376', 30) ON CONFLICT DO NOTHING;

INSERT INTO "Floor\_Elevator" VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10),

(11, 11),

(12, 12),

(13, 13),

(14, 14),

(15, 15),

(16, 16),

(17, 17),

(18, 18),

(19, 19),

(20, 20),

(21, 21),

(22, 22),

(23, 23),

(24, 24),

(25, 25),

(26, 26),

(27, 27),

(28, 28),

(29, 29),

(30, 30) ON CONFLICT DO NOTHING;

Также на рисунке 3.1 приведен интерфейс pgAdmin 4 для добавления новой записи.

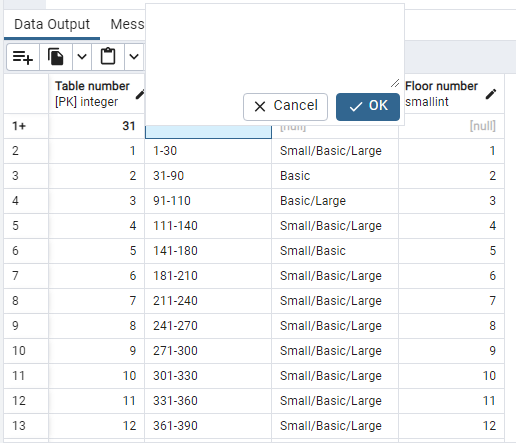
****

Рисунок 3.1 – Добавление новой записи через интерфейс pgAdmin 4

На рисунках 3.2-3.9 приведены полученные таблицы.

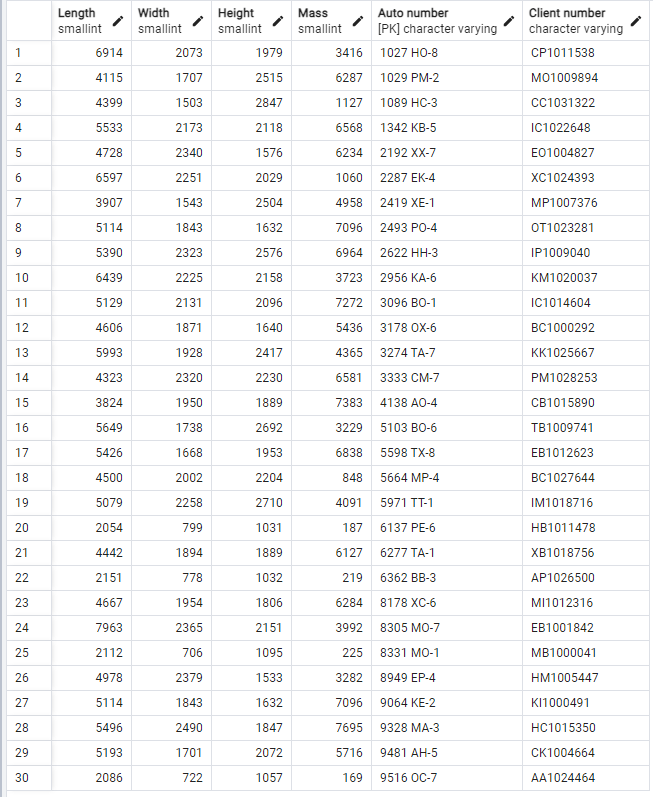
****

Рисунок 3.2 – Таблица Autos

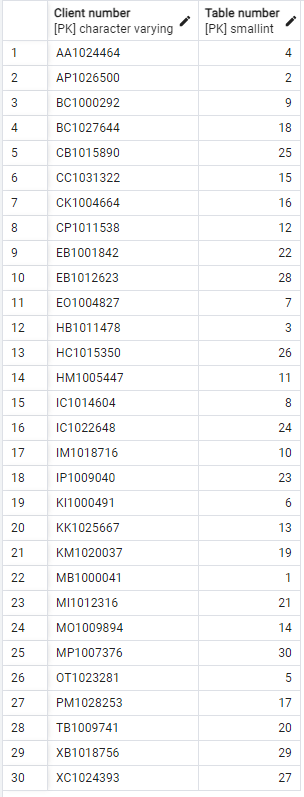
****

Рисунок 3.3 – Таблица Client\_Table

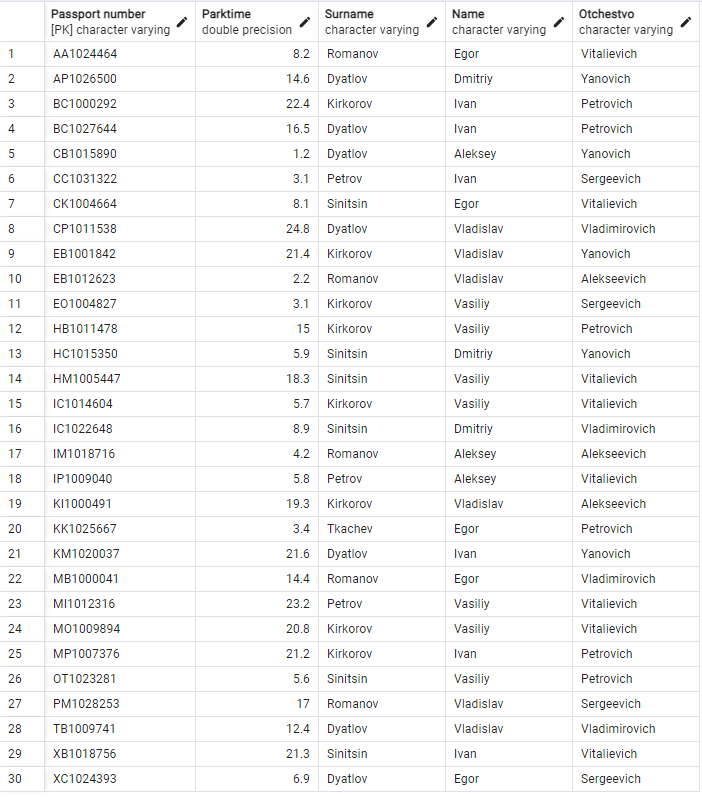


Рисунок 3.4 – Таблица Clients

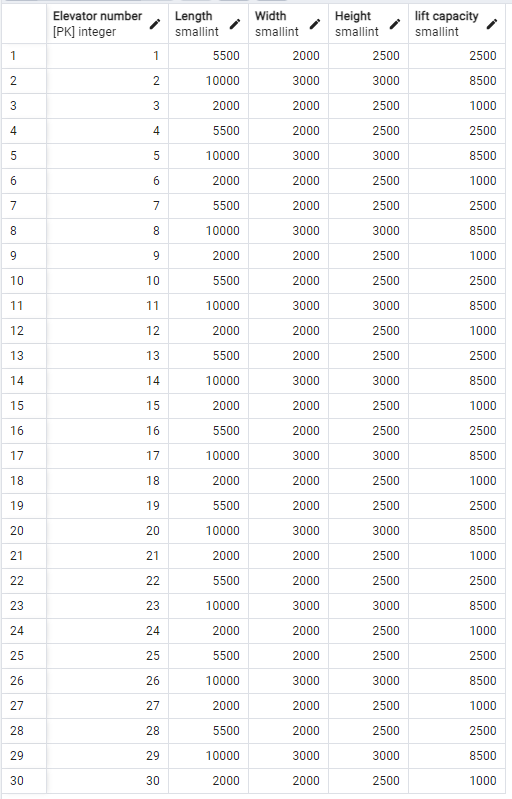


Рисунок 3.5 – Таблица Elevators

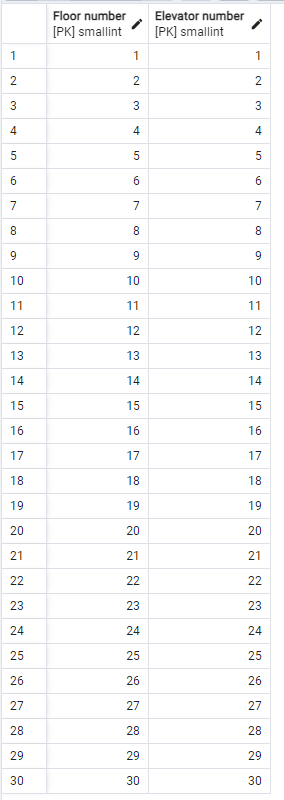


Рисунок 3.6 – Таблица Floor\_Elevator

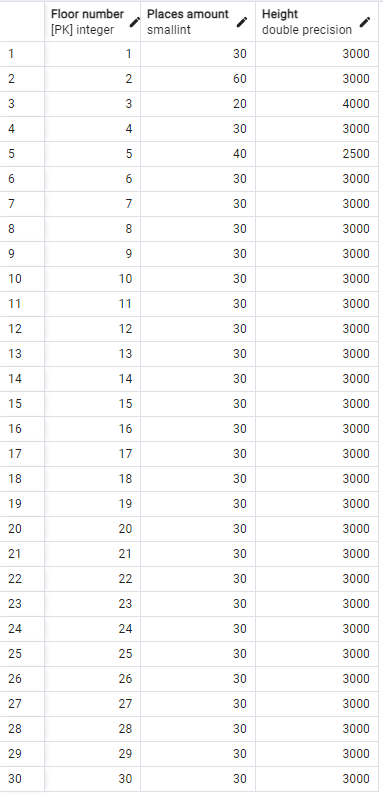


Рисунок 3.7 – Таблица Floors

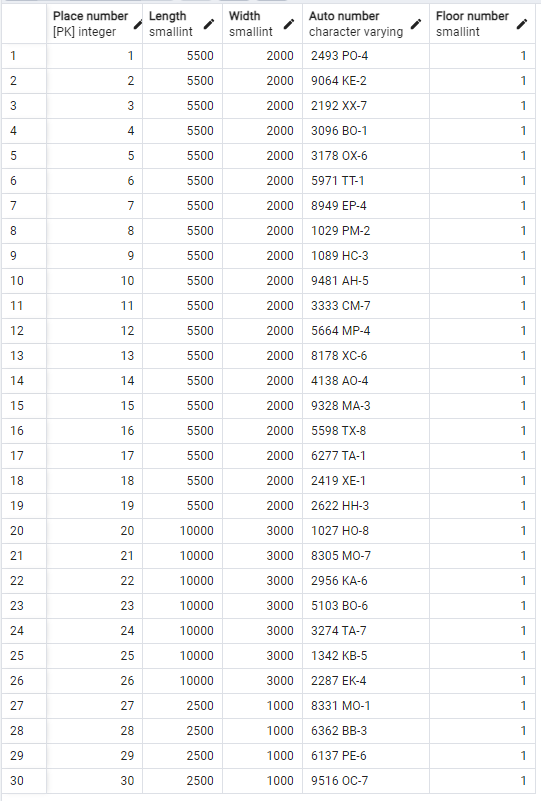


Рисунок 3.8 – Таблица Places

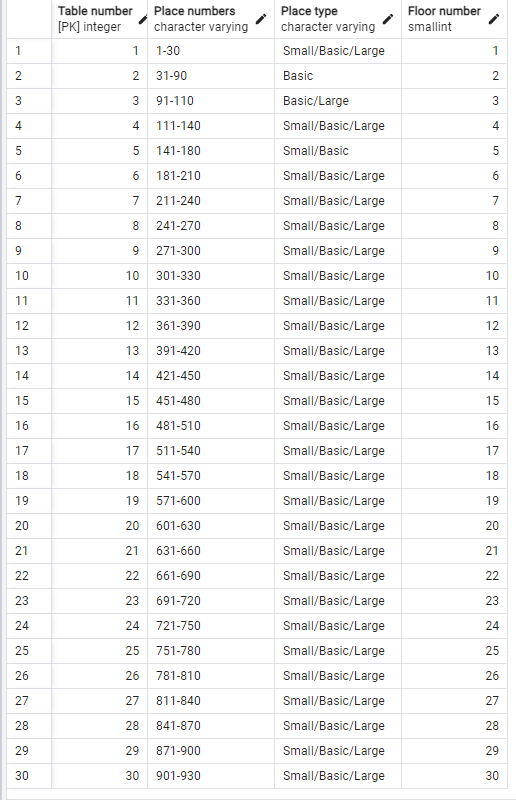


Рисунок 3.9 – Таблица Tables

# ВЫВОД

В ходе выполнения лабораторной была реализована схема базы данных по ранее построенной реляционной схеме данных. Также были сформированы SQL-запросы для создания таблиц базы данных и выполнены в СУБД. Полученные таблицы были заполнены с помощью оператора INSERT. Были рассмотрены простые действия, направленные на изменения структуры таблицы (переименования столбца, добавление ограничений) и реализованы с помощью оператора ALTER TABLE.