МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

Кафедра «Информационные технологии и компьютерные системы»

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6

по дисциплине

«Проектирование баз данных»

Вариант 8

Выполнил:

Донец Н.О.

Проверил:

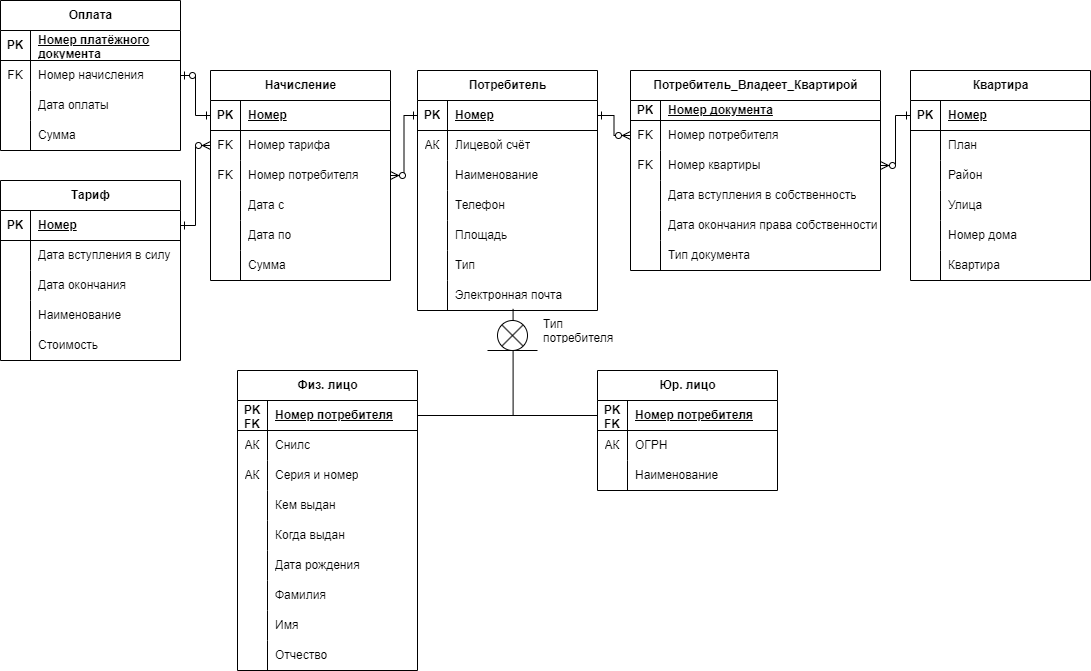
Волкова Т. В.

Севастополь

2024 г.

**Задание**

Разработать БД для ежемесячного учета оплаты услуг за теплоснабжение в теплосети района. БД должна содержать следующие данные: адрес (район, улица, номера дома и квартиры), план и площадь квартиры, месячные тарифы за потребленное тепло из расчета за 1 квадратный метр площади, общая плата за услуги, сведения об уплате с указанием дат и сумм, показатели оплаты услуг за отчетное полугодие по домам.



**4. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

* 1. **Общая схема и описание физической модели базы данных**

Для всех отношений была описана логическая структура, которая содержит последовательность расположения атрибутов в отношении и их основные характеристики.:

1. Отношение «Потребитель» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONSUMERS | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер | PK | integer | 8 | NOT NULL |
| account | лицевой счёт | АК | char | 10 | NOT NULL |
| name | наименование | - | varchar | 30 | - |
| telephone | телефон | - | integer | 11 | - |
| area | площадь | - | integer | 8 | NOT NULL |
| type | тип | - | enum | 16 | множество значений ENUM('Физ. лицо', 'Юр. лицо') |
| mail | электронная почта | - | varchar | 30 | - |

1. Отношение «Физ. лицо» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NATURAL\_PERSONS | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер потребителя | PK, FK | integer | 8 | NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE |
| snils | СНИЛС | AK | integer | 11 | NOT NULL |
| series\_and\_number | серия и номер паспорта | AK | integer | 10 | NOT NULL |
| issued\_by\_whom | кем выдан | - | varchar | 150 | - |
| when\_issued | когда выдан | - | date | - | - |
| birthday | дата рождения | - | date | - | - |
| lastname | фамилия | - | varchar | 12 | - |
| firstname | имя | - | varchar | 12 | - |
| patronomic | отчество | - | varchar | 12 | - |

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS необходимо удалить соответствующую запись данного отношения.

1. Отношение «Юр. лицо» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LEGAL\_PERSONS | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер потребителя | PK, FK | integer | 8 | NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE |
| ogrn | ОГРН | АК | integer | 13 | NOT NULL |
| name | наименование | - | varchar | 30 | - |

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS необходимо удалить соответствующую запись данного отношения.

1. Отношение «Квартира» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FLATS | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер | PK | integer | 8 | NOT NULL |
| plan | план | - | longblob | - | - |
| district | район | - | varchar | 30 | - |
| street | улица | - | varchar | 20 | - |
| house\_number | номер дома | - | integer | 4 | - |
| flat\_number | номер квартиры | - | integer | 4 | - |

1. Отношение «Потребитель\_Владеет\_Квартирой» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONSUMER\_OWN\_FLAT | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер документа | PK | integer | 8 | NOT NULL |
| consumer\_number | номер потребителя | FK | integer | 8 | NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE |
| flat\_number | номер квартиры | FK | integer | 8 | NOT NULL REFERENCES FLATS(number) ON DELETE CASCADE ON UPDATE CASCADE |
| own\_date | дата вступления в собственность | - | date | - | - |
| end\_own\_date | дата окончания права собственности | - | date | - | - |
| document\_type | тип документа | - | varchar | 30 | - |

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS или FLATS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS или FLATS необходимо удалить соответствующую запись данного отношения.

1. Отношение «Тариф» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RATES | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер | PK | integer | 8 | NOT NULL |
| start\_date | дата ступления в силу | - | date | - | - |
| end\_date | дата окончания | - | date | - | - |
| name | наименование | - | varchar | 30 | - |
| cost | стоимость | - | decimal | 8 | Точность 2 |

1. Отношение «Начисление» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ACCURALS | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер | PK | integer | 8 | NOT NULL |
| rate\_number | номер тарифа | FK | integer | 8 | NOT NULL REFERENCES RATES(number) ON DELETE CASCADE ON UPDATE CASCADE |
| consumer\_number | номер потребителя | FK | integer | 8 | NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE |
| date\_from | дата с | - | date | - | - |
| date\_to | дата по | - | date | - | - |
| amount | сумма | - | decimal | 10 | NOT NULL, Точность 2 |

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS или RATES необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS или RATES необходимо удалить соответствующую запись данного отношения.

1. Отношение «Оплата» имеет следующую структуру:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PAYMENTS | | | | | |
| Атрибут | | Признак ключа | Формат атрибута | | |
| Имя (обозначение) | Полное наименование | Тип | Длина | Ограничения |
| number | номер платёжного документа | PK | integer | 8 | NOT NULL |
| accural\_number | номер начисления | FK | integer | 8 | NOT NULL REFERENCES ACCURALS(number) ON DELETE CASCADE ON UPDATE CASCADE |
| pay\_date | дата оплаты | - | date | - | - |
| amount | сумма | - | decimal | 10 | NOT NULL Точность 2 |

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении ACCURALS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения ACCURALS необходимо удалить соответствующую запись данного отношения.

* 1. **Проектирование физической модели базы данных в выбранной СУБД**

Были созданы запросы на создание отношений базы данных. Отношения базы данных представлены на рисунках 1 – 8.

1. SQL запрос на создание отношения «Потребитель»:

CREATE TABLE CONSUMERS (

number integer(8) NOT NULL,

account char(10) NOT NULL,

name varchar(30),

telephone integer(11),

area integer(8) NOT NULL,

type ENUM('Физ. лицо', 'Юр. лицо'),

mail varchar(30),

PRIMARY KEY(number),

UNIQUE(account)

);



Рисунок 1 – Отношение «Потребитель»

1. SQL запрос на создание отношения «Физ. лицо»:

CREATE TABLE NATURAL\_PERSONS(

number integer(8) NOT NULL,

snils integer(11) NOT NULL,

series\_and\_number integer(10) NOT NULL,

issued\_by\_whom varchar(150),

when\_issued date,

birthday date,

lastname varchar(12),

firstname varchar(12),

patronomic varchar(12),

PRIMARY KEY(number),

UNIQUE(snils),

UNIQUE(series\_and\_number),

FOREIGN KEY(number) REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE

);

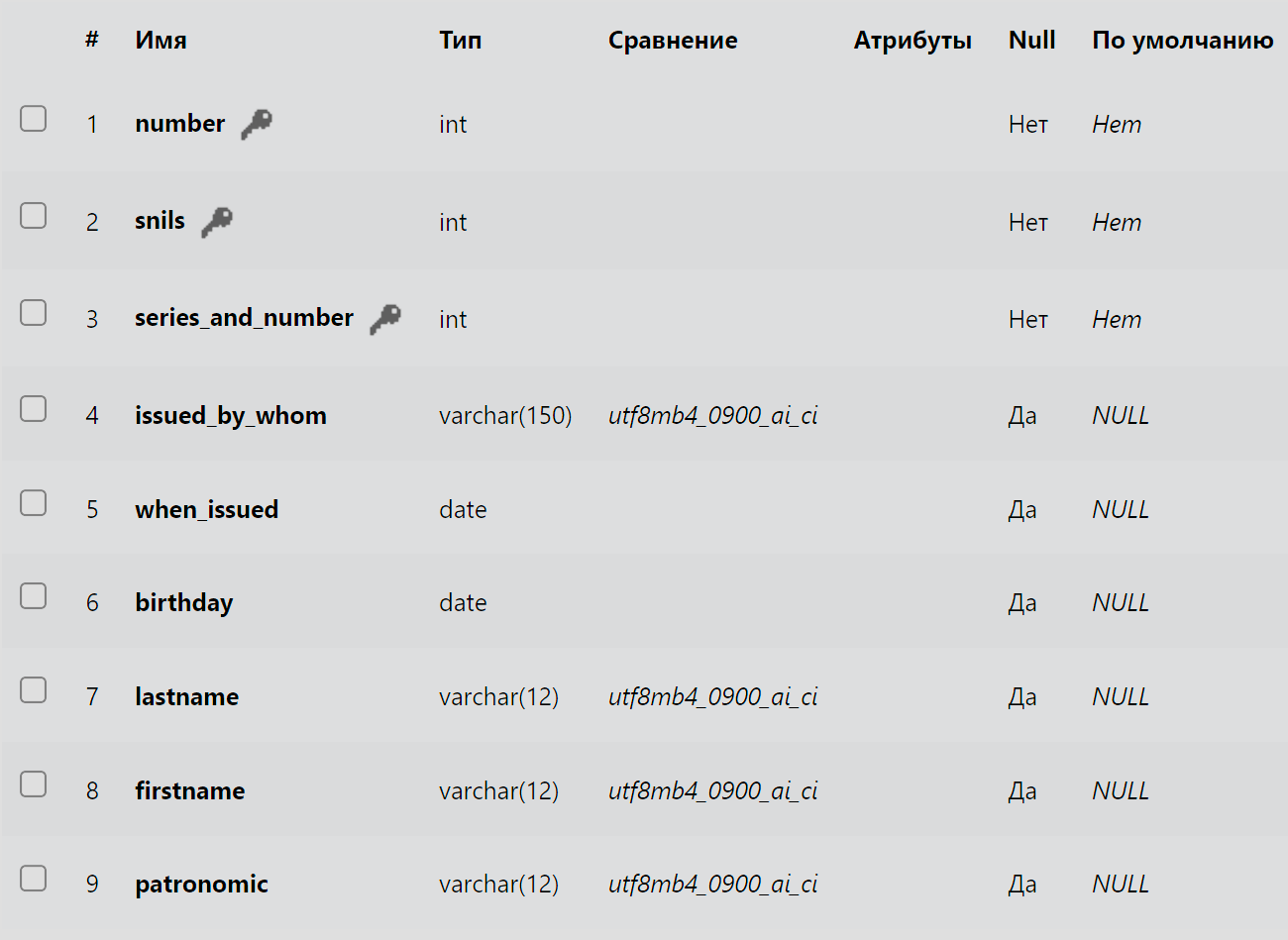


Рисунок 2 – Отношение «Физ. лицо»

1. SQL запрос на создание отношения «Юр. лицо»:

CREATE TABLE LEGAL\_PERSONS(

number integer(8) NOT NULL,

ogrn integer(13) NOT NULL,

name varchar(30),

PRIMARY KEY(number),

UNIQUE(ogrn),

FOREIGN KEY(number) REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE

);

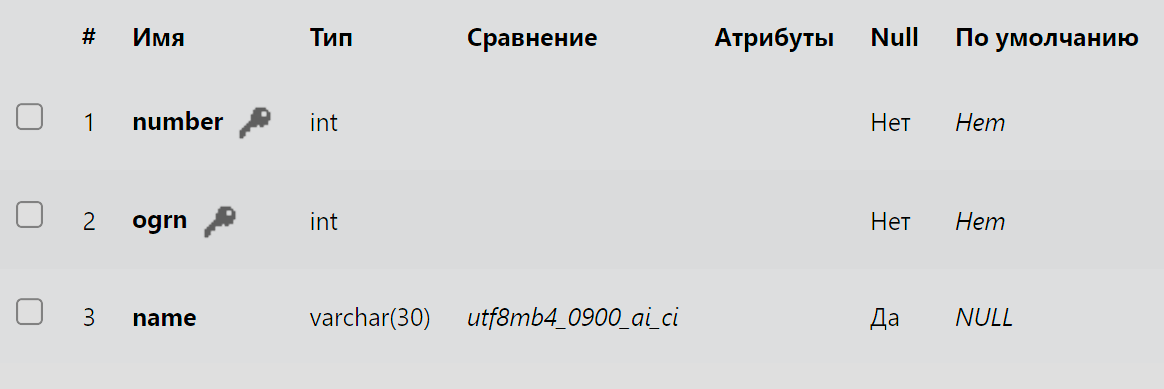


Рисунок 3 – Отношение «Юр. лицо»

1. SQL запрос на создание отношения «Квартира»:

CREATE TABLE FLATS(

number integer(8) NOT NULL PRIMARY KEY,

plan longblob,

district varchar(30),

street varchar(20),

house\_number integer(4),

flat\_number integer(4)

);

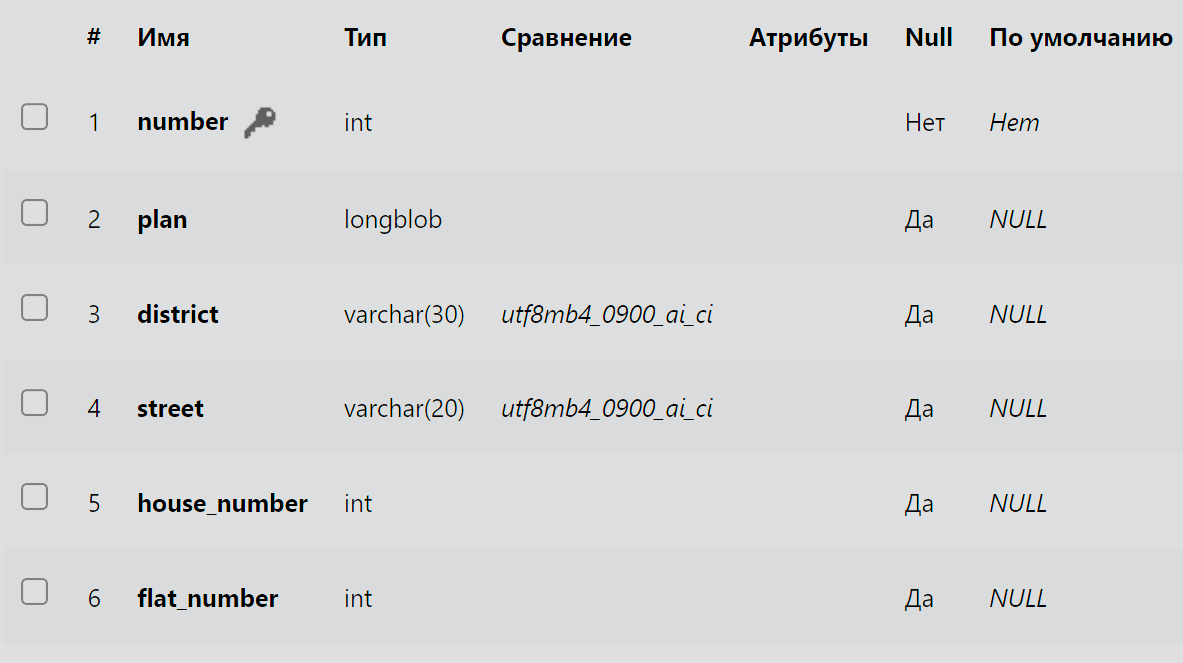


Рисунок 4 – Отношение «Квартира»

1. SQL запрос на создание отношения «Потребитель\_Владеет\_Квартирой»:

CREATE TABLE CONSUMER\_OWN\_FLAT(

number integer(8) NOT NULL,

consumer\_number integer(8) NOT NULL,

flat\_number integer(8) NOT NULL,

own\_date date,

end\_own\_date date,

document\_type varchar(30),

PRIMARY KEY(number),

FOREIGN KEY(consumer\_number) REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(flat\_number) REFERENCES FLATS(number) ON DELETE CASCADE ON UPDATE CASCADE

);

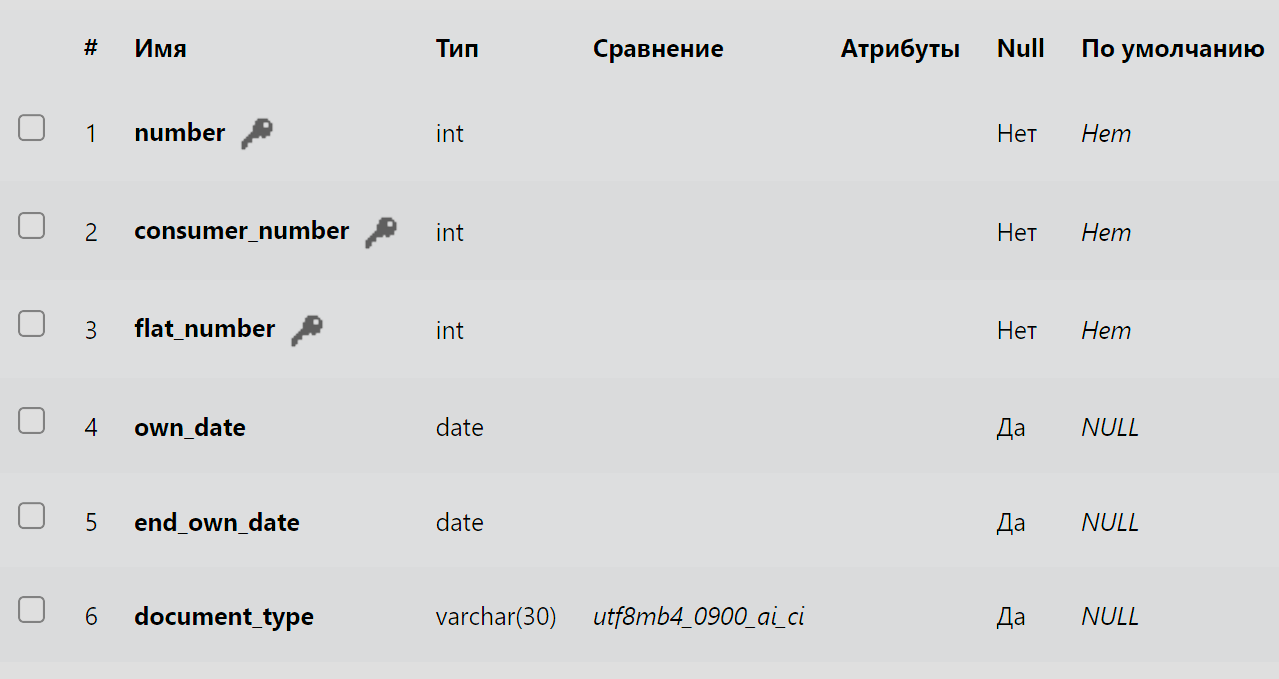


Рисунок 5 – Отношение «Потребитель\_Владеет\_Квартирой»

1. SQL запрос на создание отношения «Тариф»:

CREATE TABLE RATES(

number integer(8) NOT NULL PRIMARY KEY,

start\_date date,

end\_date date,

name varchar(30),

cost decimal(8,2)

);

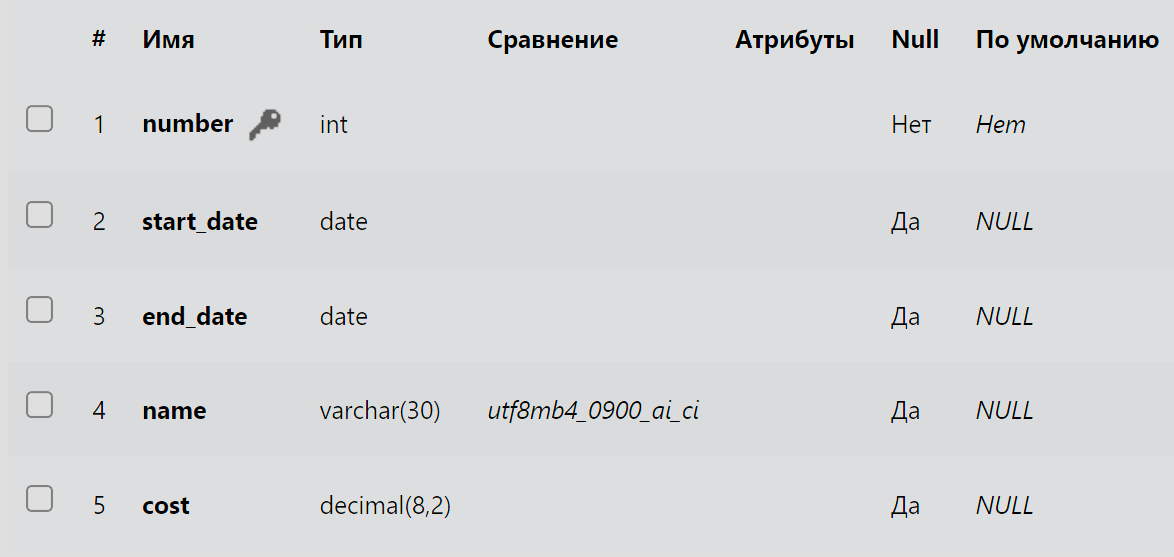


Рисунок 6 – Отношение «Тариф»

1. SQL запрос на создание отношения «Начисление»:

CREATE TABLE ACCURALS(

number integer(8) NOT NULL,

rate\_number integer(8) NOT NULL,

consumer\_number integer(8) NOT NULL,

date\_from date,

date\_to date,

amount decimal(10, 2) NOT NULL,

PRIMARY KEY(number),

FOREIGN KEY(rate\_number) REFERENCES RATES(number) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(consumer\_number) REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE

);

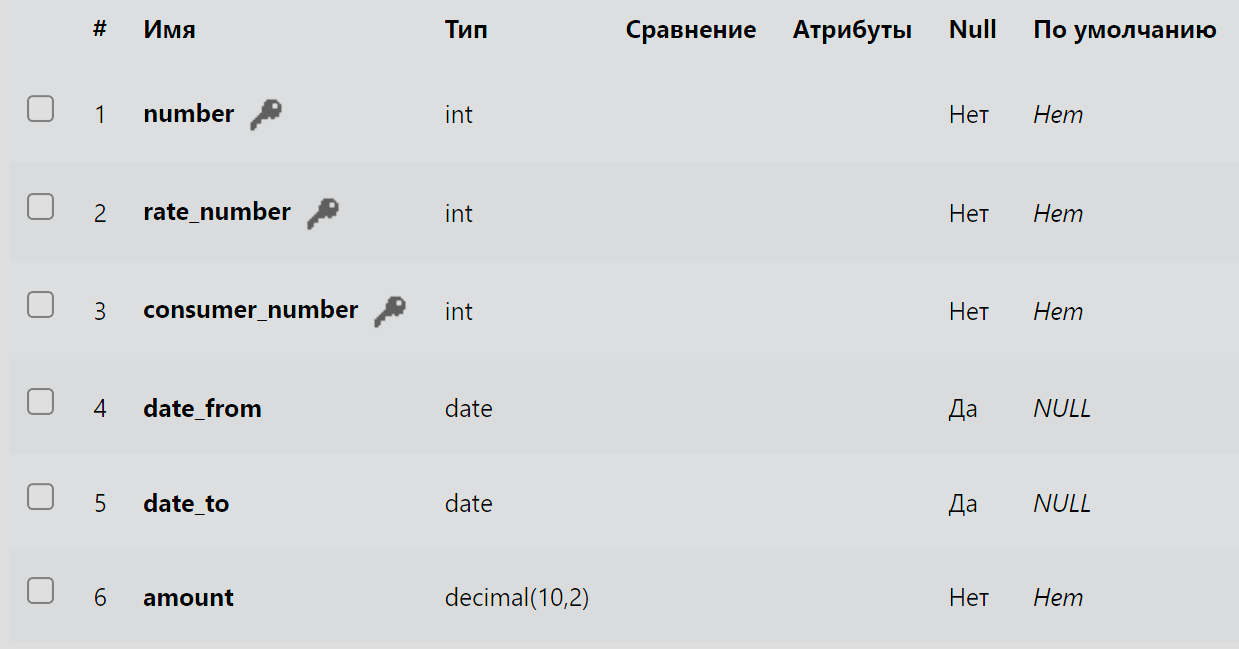


Рисунок 7 – Отношение «Начисление»

1. SQL запрос на создание отношения «Оплата»:

CREATE TABLE PAYMENTS(

number integer(8) NOT NULL PRIMARY KEY,

accural\_number integer(8) NOT NULL,

pay\_date date,

amount decimal(10, 2) NOT NULL,

FOREIGN KEY(accural\_number) REFERENCES ACCURALS(number) ON DELETE CASCADE ON UPDATE CASCADE

);

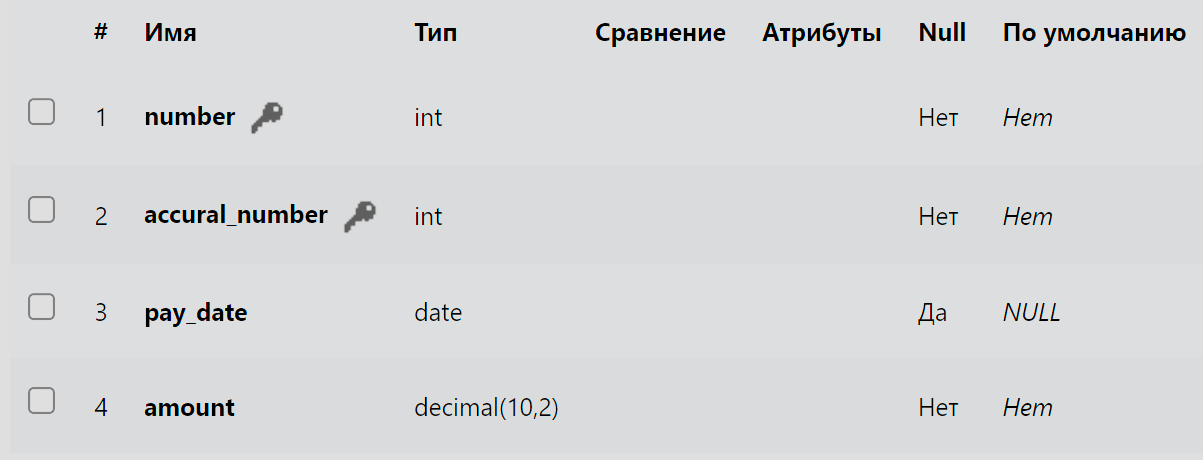


Рисунок 8 – Отношение «Начисление»

Также были созданы SQL-запросы на изменение структуры отношения, для чего была сделана временная копия таблицы RATES с названием RATES\_COPY. Было продемонстрировано умение работать с запросами ALTER TABLE: добавление нового столбца после первого имеющегося, изменение типа данных в любом столбце, добавление ограничения целостности на любой столбец, удаление столбца.

1. Копирование таблицы с новым названием (Рисунок 9):

CREATE TABLE RATES\_COPY LIKE RATES;

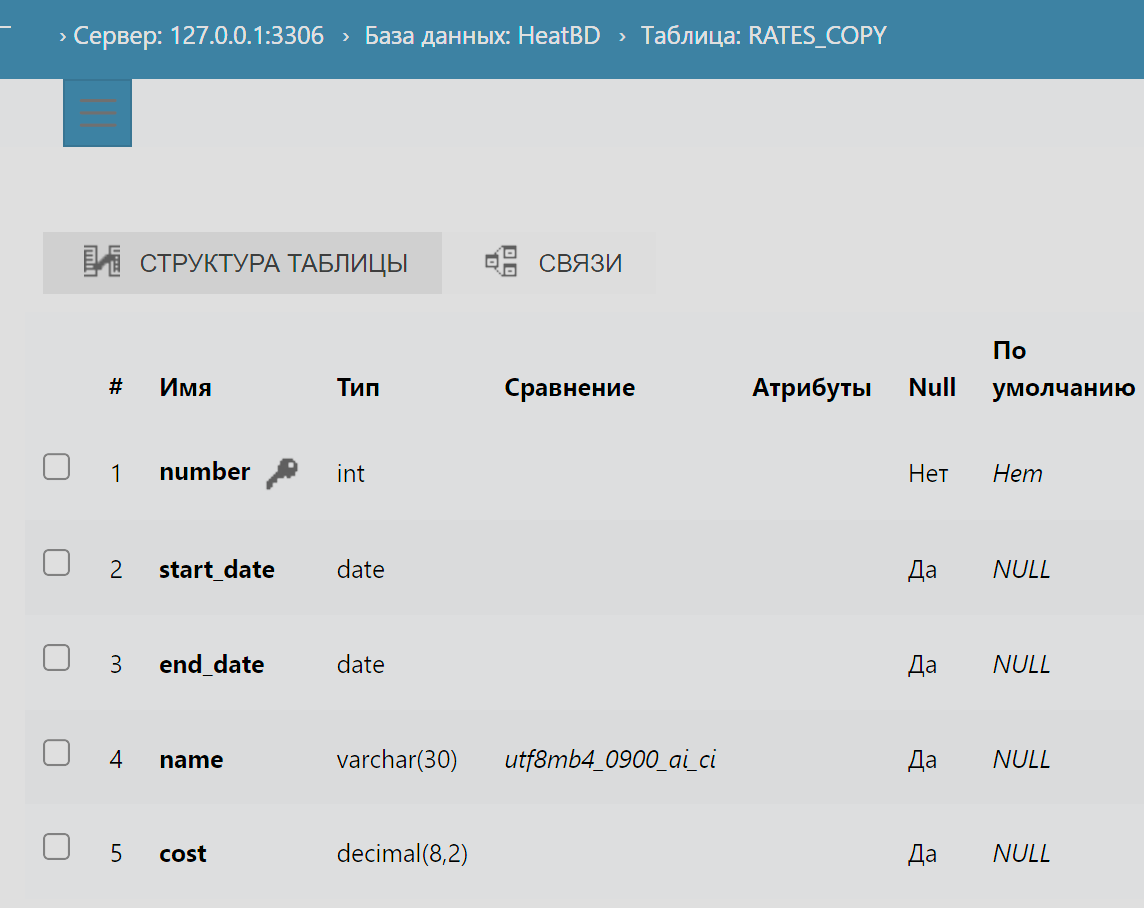


Рисунок 9 – Отношение «RATES\_COPY»

1. Добавление нового столбца после первого имеющегося (Рисунок 10):

ALTER TABLE RATES\_COPY

ADD new\_column integer NOT NULL;

ALTER TABLE RATES\_COPY

MODIFY new\_column integer AFTER number;

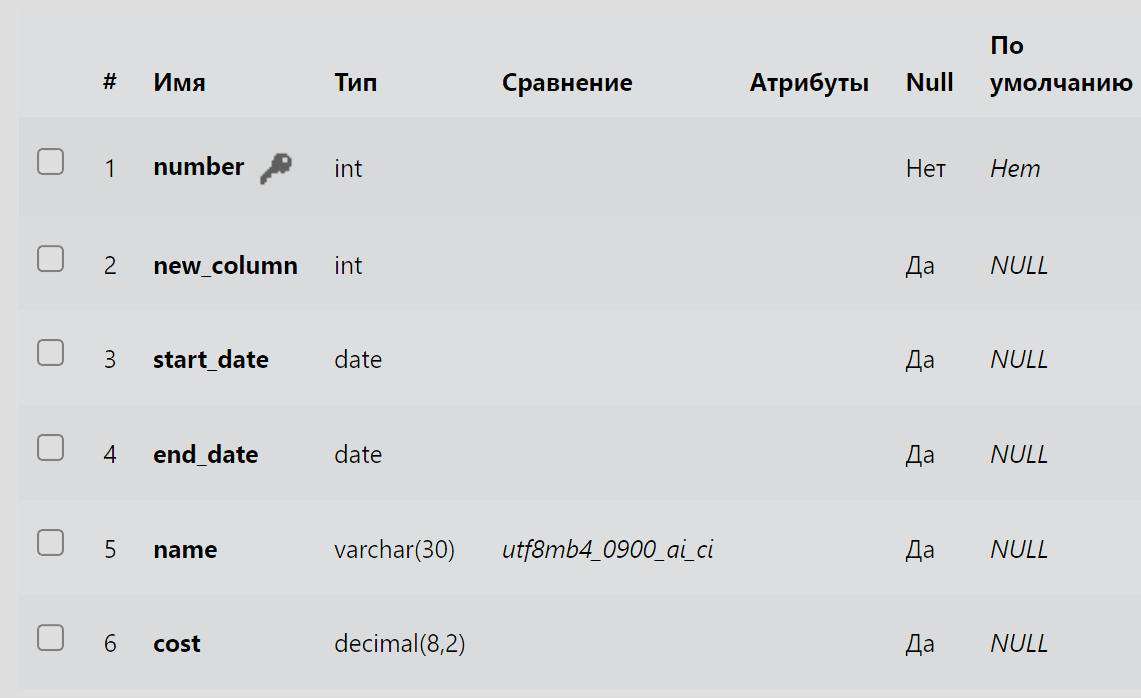


Рисунок 10 – Добавлен столбец

1. Изменения типа данных в новом столбце (Рисунок 11):

ALTER TABLE RATES\_COPY

MODIFY new\_column VARCHAR(12);



Рисунок 11 – Изменён тип данных в столбце

1. Добавление ограничения на новый столбец (Рисунок 12):

ALTER TABLE RATES\_COPY

ADD CONSTRAINT UNIQUE(new\_column),

ALTER new\_column SET DEFAULT 'dojdik';

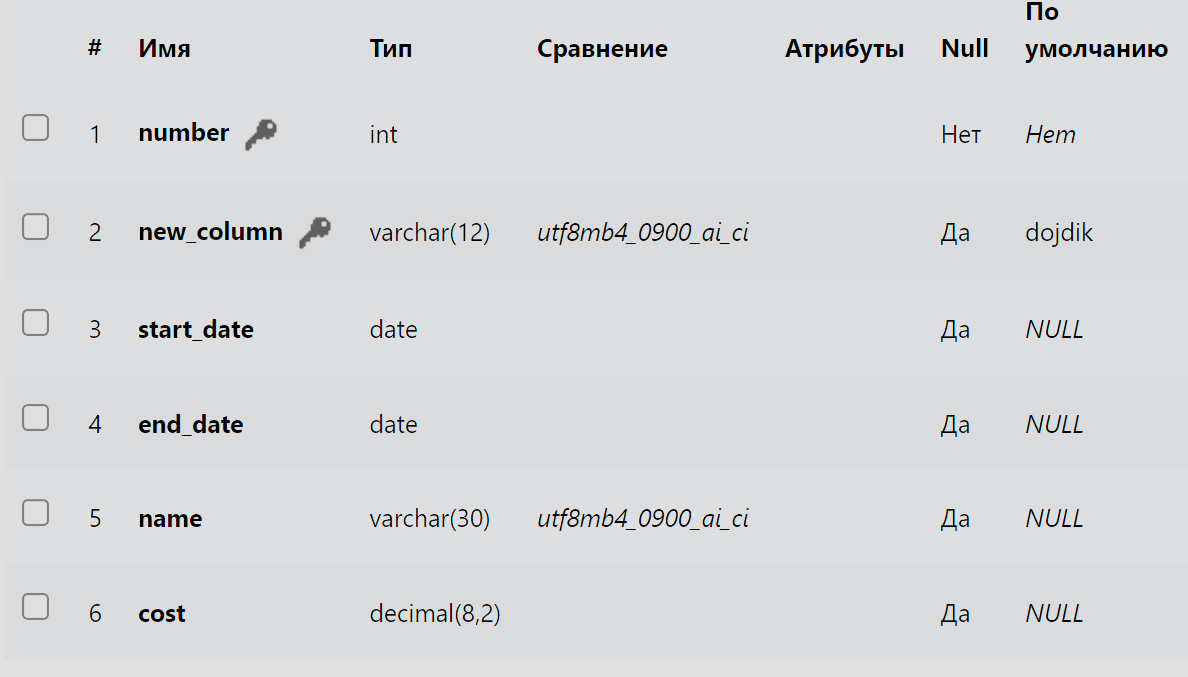


Рисунок 12 – Добавлено новое ограничение на столбец

1. Удаление столбца (Рисунок 13):

ALTER TABLE RATES\_COPY

DROP name;

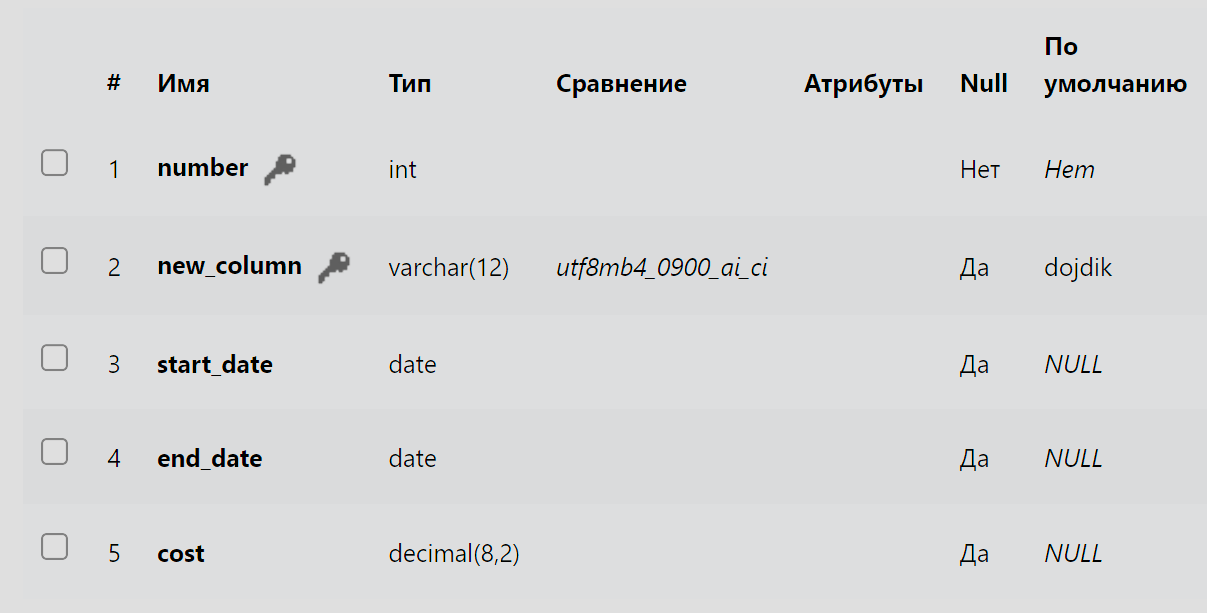


Рисунок 13 – Удалён столбец

Был создан SQL-запрос на удаление отношения RATES\_COPY:

DROP TABLE RATES\_COPY;

Также был создана схема физической модели данных на основе выбранной СУБД (Рисунок 14).

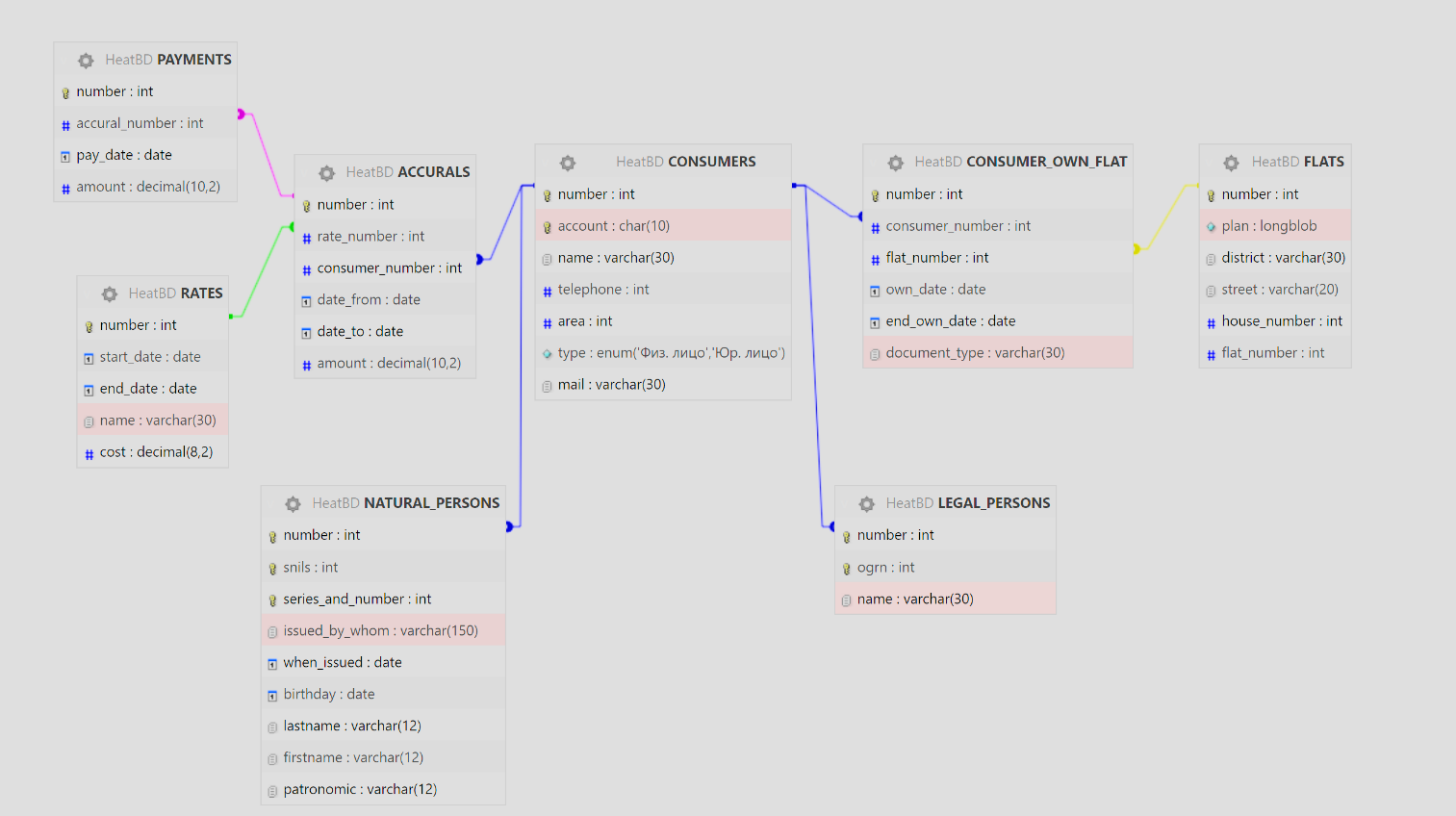


Рисунок 14 – Схема физической модели данных

**Вывод**

В ходе практической работы было проведено физическое проектирование базы данных.