

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

ДОЛЖНОСТЬ

старший преподаватель

подпись, дата

Путилова Н.В.

инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

### **« Создание и модификация базы данных и таблиц базы данных»**

**по дисциплине: Проектирование баз данных**

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

4236

подпись, дата

Л. Мвале

инициалы, фамилия

Санкт-Петербург  
2024

**Цель работы:** Получение умений и навыков создания и модификации таблиц на языке SQL.

### **Задание и последовательность выполнения работы**

В соответствии с моделью, разработанной в предыдущей работе, создать базу данных. Продемонстрировать умение добавить и удалить столбец командой alter table

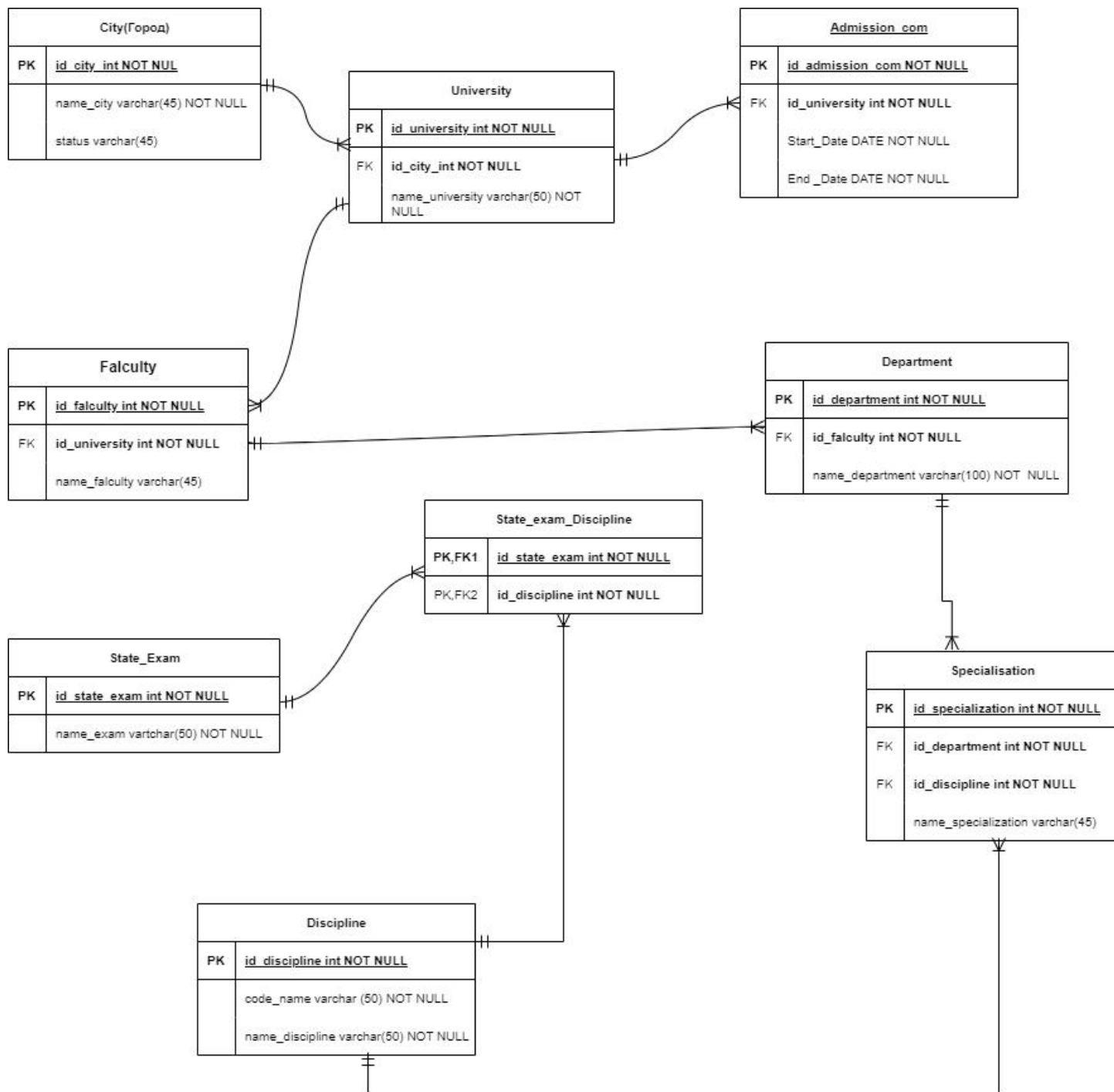
### **Текст задания Вариант 15**

15. вузы для абитуриента: город, вуз, факультеты, направления, направленности, ЕГЭ которые нужно сдать, дата начала/конца приемной кампании.

(Направление -09.03.04 «Программная инженерия», Направленность — его конкретизация «Разработка программно-информационных систем», именно направленность закреплена за кафедрой и соответственно факультетом)

- а. направленности, в которых есть слово «систем», но оно не первое
- б. Кафедра, не принимающая ни на одну направленность
- в. направление, на которое надо сдавать математику и информатику
- г. факультет, принимающий на количество направлений больше среднего
- д. город, в котором есть все укрупненные группы направлений и специальностей(УГСН) (первые 2 цифры номера специальности, т.е у 09.03.04 УГСН=09, а у 02.03.03-02)
- е. вуз, с последним по алфавиту названием
- ж. направление, на которое не надо сдавать ЕГЭ по математике, но надо по иностранному языку

## Физическую модель базы данных



## Описания ссылочной целостности БД

Дочерняя таблица	Внешний ключ	Родительская таблица	Целостность ссылок при удалении	Описание целостности ссылок при удалении	Целостность ссылок при обновлении	Описание целостности ссылок при обновлении
University	id_city	city	CASCADE	При удаление города, все связанные университет также удалятся	NO ACTION	Не даёт изменить первичный ключ из таблицы City, если с ним связаны какие-либо данные в University (проверка сразу)
Faculty	id_university	University	CASCADE	При удаление университета, все связанные факультеты также удалятся	CASCADE	При изменении первичного ключа из университетской таблицы, связанные с ним внешние ключи в дочерней Faculty изменятся на такое же значение
Admission_comp	Id_university	University	NO ACTION	Нельзя удалить университет id, если существует запись в приёмной кампании	NO ACTION	Не даёт изменить первичный ключ из таблицы University, если с ним связаны какие-либо данные в Admission_comp (проверка сразу)
Department	Id_faculty	Faculty	CASCADE	При удалении факультета все связанные кафедры удаляются	CASCADE	При изменении первичного ключа из факультетской таблицы, связанные с ним внешние ключи в Department изменятся на такое же значение
Specialization	Id_department	Department	NO ACTION	Нельзя удалить Department id, если существует запись в Specialization	NO ACTION	Не даёт изменить первичный ключ из таблицы Department, если с ним связаны какие-либо данные в Specialization (проверка отложена)
Specialization	Id_discipline	Discipline	NO ACTION	Нельзя удалить Discipline id, если	NO ACTION	Не даёт изменить первичный ключ из

				существует запись в Specialization		таблица Discipline , если с ним связаны какие-либо данные в Specialization (проверка отложена)
State_exam_Discipline	Id_state_exam	State_exam	NO ACTION	Нельзя удалить экзамен, если существуют связанные State_exam_Discipline	CASCADE	При изменении первичного ключа из State_exam таблицы, связанные с ним внешние ключи в State_exam_Discipline изменяются на такое же значение
State_exam_Discipline	Id_discipline	Discipline	NO ACTION	Нельзя удалить Discipline, если существуют связанные State_exam_Discipline	CASCADE	При изменении первичного ключа из Discipline таблицы, связанные с ним внешние ключи в State_exam_Discipline изменяются на такое же значение

## Коды базы данных

### Создание таблиц CREATE:

```
CREATE TABLE IF NOT EXISTS City (
    id_city INT NOT NULL AUTO_INCREMENT,
    name_city VARCHAR(20) NOT NULL,
    status VARCHAR(20),
    PRIMARY KEY (id_city)
);
```

```
CREATE TABLE IF NOT EXISTS University (
    id_university INT NOT NULL AUTO_INCREMENT,
    name_university VARCHAR(50) NOT NULL,
    id_city INT NOT NULL,
    PRIMARY KEY (id_university),
    FOREIGN KEY (id_city) REFERENCES City(id_city)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);
```

```
CREATE TABLE IF NOT EXISTS Admission_com(  
    id_admission INT NOT NULL AUTO_INCREMENT,  
    id_university INT NOT NULL,  
    Start_Date DATE NOT NULL,  
    End_Date DATE NOT NULL,  
    PRIMARY KEY(id_admission),  
    FOREIGN KEY(id_university) REFERENCES University(id_university)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
);
```

```
CREATE TABLE IF NOT EXISTS Falculty(  
    id_falculty INT NOT NULL AUTO_INCREMENT,  
    name_falculty VARCHAR(50) NOT NULL,  
    id_university INT NOT NULL,  
    PRIMARY KEY(id_falculty),  
    FOREIGN KEY(id_university) REFERENCES University(id_university)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Department(  
    id_department INT NOT NULL AUTO_INCREMENT,  
    name_department VARCHAR(50) NOT NULL,  
    id_falculty INT NOT NULL,  
    PRIMARY KEY(id_falculty),  
    FOREIGN KEY(id_falculty) REFERENCES Falculty(id_falculty)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Discipline(  
    id_discipline INT NOT NULL AUTO_INCREMENT,  
    code_name VARCHAR(50) NOT NULL,  
    name_discipline VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id_discipline)  
);
```

```
CREATE TABLE IF NOT EXISTS Specialization (  
    id_specialization INT NOT NULL AUTO_INCREMENT,  
    id_department INT NOT NULL,  
    id_discipline INT NOT NULL,  
    PRIMARY KEY (id_specialization),  
    FOREIGN KEY (id_department) REFERENCES Department(id_department)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    FOREIGN KEY (id_discipline) REFERENCES Discipline(id_discipline)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
);
```

```
CREATE TABLE IF NOT EXISTS State_Exam(  
    id_state_exam INT NOT NULL AUTO_INCREMENT,  
    name__state_exam VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id_state_exam)  
);
```

```
CREATE TABLE IF NOT EXISTS State_Exam_Discipline (  
    id_state_exam INT NOT NULL,  
    id_discipline INT NOT NULL,  
    PRIMARY KEY (id_state_exam, id_discipline),  
    FOREIGN KEY (id_state_exam) REFERENCES State_Exam(id_state_exam)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE,  
    FOREIGN KEY (id_discipline) REFERENCES Discipline(id_discipline)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

## **Изменение таблиц ALTER:**

### **1. Добавить новый столбец в таблицу**

Допустим, мы хотим добавить новый столбец population в таблицу City:

```
ALTER TABLE City ADD COLUMN population INT;
```

### **2. Изменить существующий столбец**

Чтобы изменить тип данных столбца name\_faculty в таблице Faculty с VARCHAR(50) на VARCHAR(100):

```
ALTER TABLE Faculty  
MODIFY name_faculty VARCHAR(100);
```

### **3. Переименовать столбец**

Чтобы переименовать столбец name\_city в city\_name в таблице City:

```
ALTER TABLE City  
CHANGE name_city city_name VARCHAR(20);
```

### **4. Удалить столбец**

Если необходимо удалить столбец status из таблицы City:

```
ALTER TABLE City DROP COLUMN status;
```

### **5. Добавить внешний ключ**

Предположим, мы хотим добавить внешний ключ в таблицу Admission\_com, связывающий её с таблицей Faculty:

```
ALTER TABLE Admission_com ADD CONSTRAINT fk_faculty_admission FOREIGN KEY  
(id_university) REFERENCES Faculty(id_university) ON DELETE CASCADE ON  
UPDATE CASCADE;
```

### **6. Удалить внешний ключ**

Чтобы удалить существующее ограничение внешнего ключа в таблице Specialization:

```
ALTER TABLE Specialization DROP FOREIGN KEY fk_specialization_department;
```

### **7. Переименовать таблицу**

Чтобы переименовать таблицу Faculty в Faculty (исправление орфографической ошибки):



```
ALTER TABLE Falculty  
RENAME TO Faculty;
```

## 8. Добавить уникальное ограничение

Чтобы сделать столбец code\_name в таблице Discipline уникальным:

```
ALTER TABLE Discipline ADD CONSTRAINT unique_code_name Unique (code_name);
```

## 9. Добавить индекс

Если вы хотите добавить индекс для столбца name\_city в таблице City:

```
ALTER TABLE CityADD INDEX idx_name_city (name_city);
```

## 10. Добавить новый внешний ключ

Допустим, вам нужно добавить внешний ключ, связывающий таблицу Specialization с другой таблицей:

```
ALTER TABLE SpecializationADD CONSTRAINT fk_new_foreign_keyFOREIGN KEY  
(id_department) REFERENCES Department(id_department)ON DELETE CASCADEON  
UPDATE NO ACTION;
```

## Удалить таблица DROP:

1: Удалить таблицу

```
DROP TABLE IF EXISTS Admission_com;
```

2: Удалить базу данных

```
DROP DATABASE IF EXISTS UniversityDB;
```

## Оператор TRUNCATE

Очистить таблицу от всех данных

```
TRUNCATE TABLE Discipline;
```

## Оператор RENAME

Переименовать таблицу

```
RENAME TABLE Falculty TO Faculty;
```

## **Выводы**

В данной работе были приобретены навыки создания физической модели базы данных в третьей нормальной форме и описания ссылочной целостности. Таблица с преференциальной целостностью помогает обосновать выбранные стратегии для поддержания целостности данных.