

ЛАБОРАТОРНАЯ РАБОТА №4

В рамках лабораторной работы №4 разработал физическую схему данных БД «Общежитие».

Напомню, что начинал разрабатывать эту БД я ещё в Лабораторной №2 (рис. 1) Вот некоторые моменты из неё:

Цель базы данных для «общежитие» - управление информацией об учащихся, персонале и помещениях общежития.

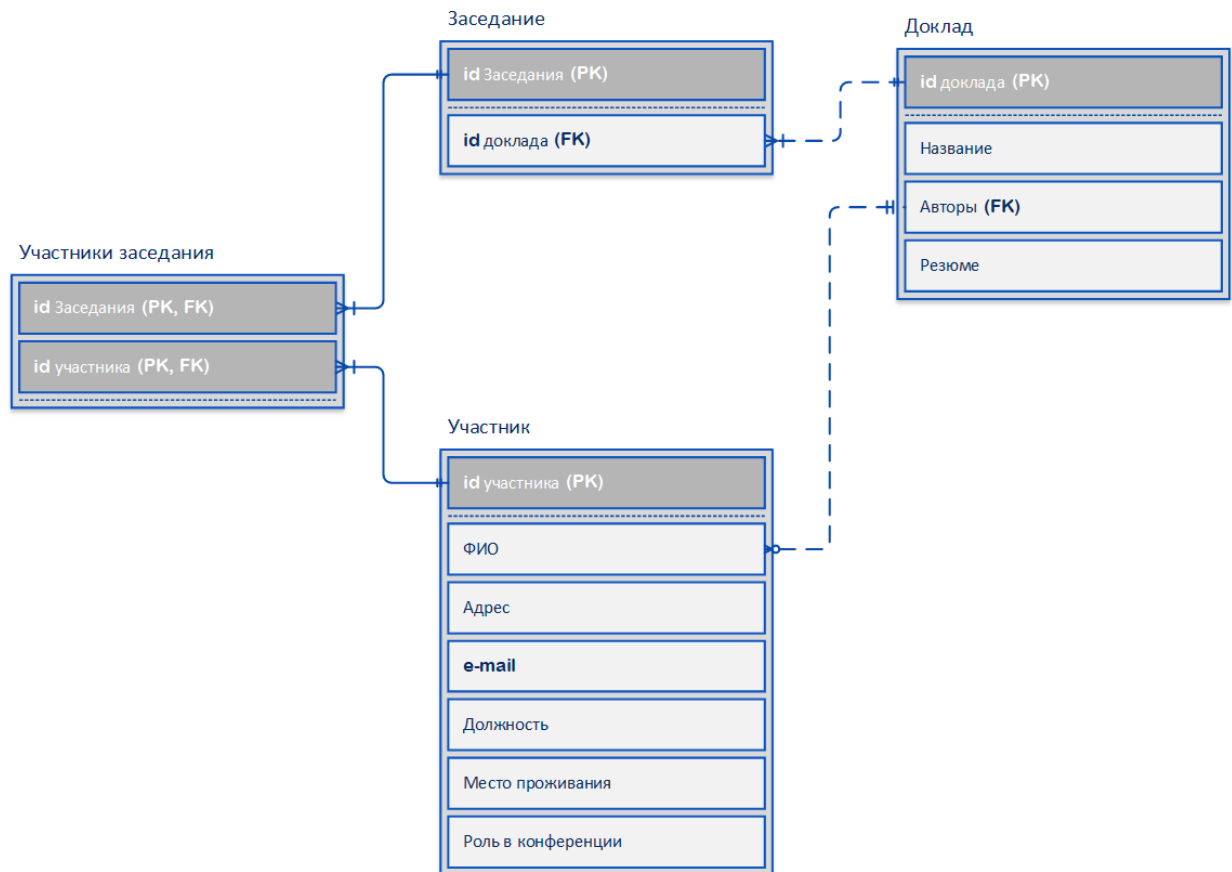


Рисунок 1. Модель данных из лабораторной работы №2

Итак, на основе данной логической модели строим физическую модель:

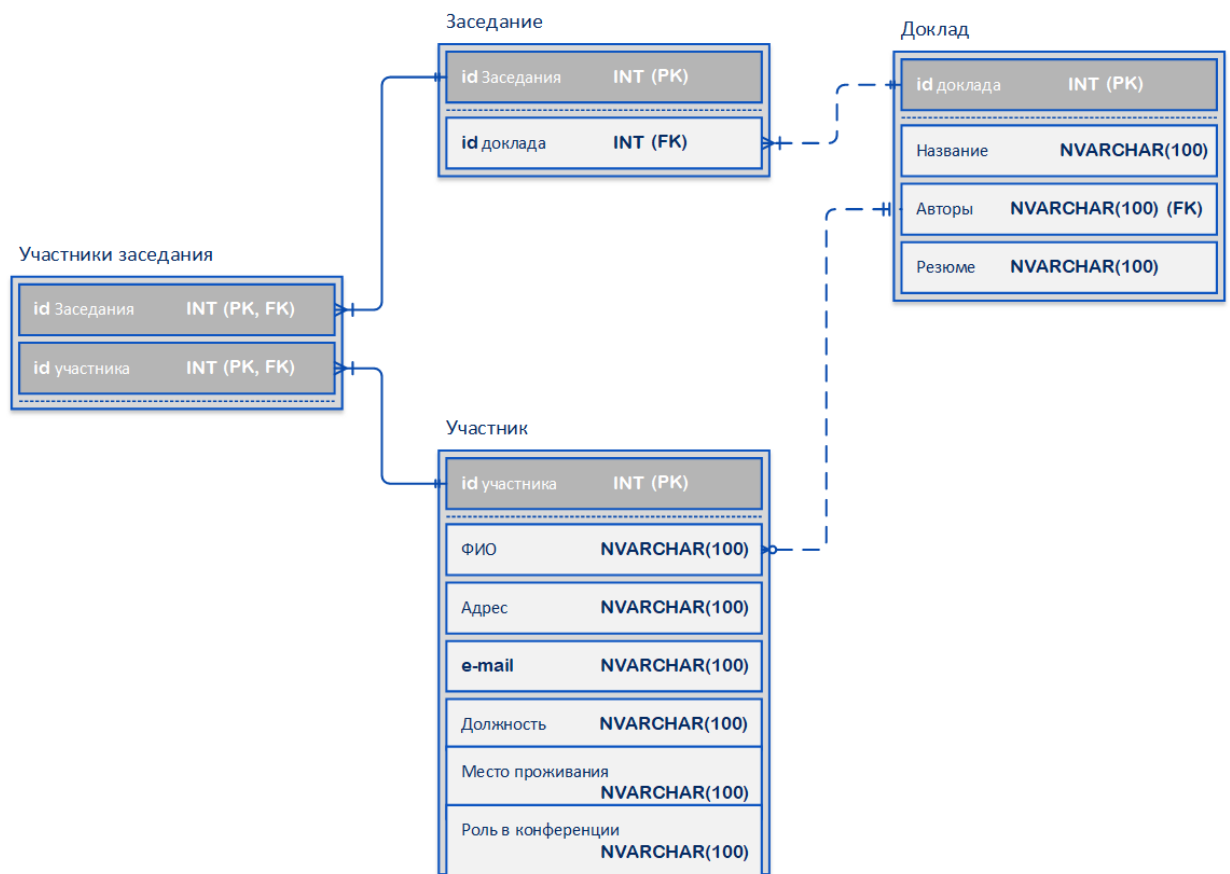


Рисунок 2. Физическая модель БД «Общежитие»

Теперь можем приступить к созданию таблиц в SSMS. Получаем такой результат (рис. 3):

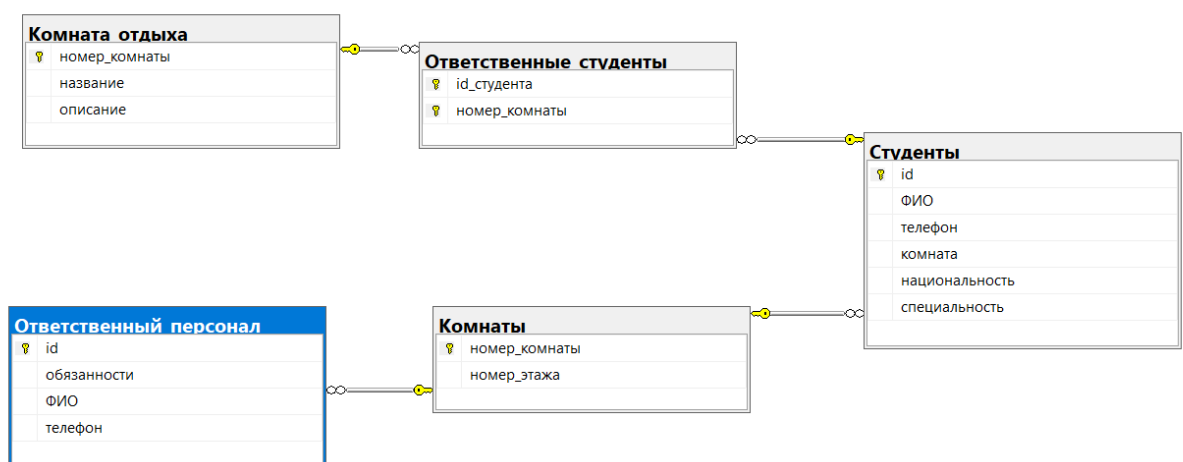


Рисунок 3. Диаграмма сгенерированная SSMS после создания всех таблиц

На SQL код запроса создания таблиц будет выглядеть примерно так:

```
USE Dormitory

-- Создание таблицы "Ответственный персонал"
```

```

CREATE TABLE Ответственный_персонал (
    id BIGINT PRIMARY KEY,
    обязанности TINYINT NOT NULL,
    ФИО VARCHAR(100) NOT NULL,
    телефон CHAR(13) NOT NULL,
);

-- Создание уникального индекса для номера этажа
CREATE UNIQUE INDEX уникальный_индекс_обязанностей
ON Ответственный_персонал (обязанности)

-- Создание таблицы "Комнаты"
CREATE TABLE Комнаты (
    номер_комнаты SMALLINT PRIMARY KEY,
    номер_этажа TINYINT NOT NULL
    FOREIGN KEY (номер_этажа) REFERENCES Ответственный_персонал(обязанности)
);

-- Создание таблицы "Студенты"
CREATE TABLE Студенты (
    id BIGINT PRIMARY KEY,
    ФИО VARCHAR(100) NOT NULL,
    телефон CHAR(13) NOT NULL,
    комната SMALLINT NOT NULL,
    национальность VARCHAR(40),
    специальность VARCHAR(40) NOT NULL,
    FOREIGN KEY (комната) REFERENCES Комнаты(номер_комнаты)
);

-- Создание таблицы "Комната отдыха"
CREATE TABLE Комната_отдыха (
    номер_комнаты SMALLINT PRIMARY KEY,
    название VARCHAR(40) NOT NULL,
    описание TEXT
);

-- Создание таблицы связи "Ответственные студенты"
CREATE TABLE Ответственные_студенты (
    id_студента BIGINT,
    номер_комнаты SMALLINT,
    FOREIGN KEY (id_студента) REFERENCES Студенты(id),
    FOREIGN KEY (номер_комнаты) REFERENCES Комната_отдыха(номер_комнаты),
    PRIMARY KEY (id_студента, номер_комнаты)
);

```

Код запроса на создание таблиц

Далее выставим все необходимые ограничения на поля:

```

-- Ограничение на поле "номер_комнаты"
ALTER TABLE Комнаты
ADD CONSTRAINT СК_Номер_Комнаты CHECK (номер_комнаты > 0 AND номер_комнаты < 1000);

-- Ограничение на поле "номер_этажа"
ALTER TABLE Ответственный_персонал
ADD CONSTRAINT СК_Номер_Этажа CHECK (обязанности > 0 AND обязанности < 10);

-- Настройка каскадного удаления для связанных записей в таблице "Комнаты"
ALTER TABLE Комнаты
ADD CONSTRAINT FK_Комнаты_Обязанности
FOREIGN KEY (номер_этажа) REFERENCES Ответственный_персонал(обязанности)
ON DELETE CASCADE;

-- Ограничение на поле "комната"

```

```

ALTER TABLE Студенты
ADD CONSTRAINT CK_Комната CHECK (комната > 0 AND комната < 1000);

-- Настройка каскадного удаления для связанных записей в таблице "Студенты"
ALTER TABLE Студенты
ADD CONSTRAINT FK_Комната_Комнаты
FOREIGN KEY (комната) REFERENCES Комнаты(номер_комнаты)
ON DELETE CASCADE;

-- Ограничение на поле "обязанности"
ALTER TABLE Ответственный_персонал
ADD CONSTRAINT CK_Обязанности CHECK (обязанности > 0 AND обязанности < 10);

-- Ограничение на поле "номер_комнаты"
ALTER TABLE Комната_отдыха
ADD CONSTRAINT CK_Номер_Комнаты_Отдыха CHECK (номер_комнаты > 0 AND номер_комнаты < 1000);

-- Настройка каскадного удаления для связанных записей в таблице "Ответственные студенты"
ALTER TABLE Ответственные_студенты
ADD CONSTRAINT FK_Студенты_Комната_отдыха
FOREIGN KEY (номер_комнаты) REFERENCES Комната_отдыха(номер_комнаты)
ON DELETE CASCADE;

ALTER TABLE Студенты
ADD CONSTRAINT CHK_Формат_Телефона
CHECK (телефон LIKE '+375[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');

```

Код запроса на создание ограничений полей таблиц

Далее вставляем данные и делаем несколько запросов SELECT. Вот код запросов и их ответы:

```

SELECT * FROM Студенты WHERE ФИО LIKE '%Иванов%';

SELECT * FROM Студенты ORDER BY ФИО;

SELECT * FROM Студенты WHERE национальность = 'белорус' AND специальность = 'Инженерия';

SELECT Студенты.*
FROM Студенты
JOIN Ответственные_студенты ON Студенты.id = Ответственные_студенты.id_студента

```

Код запроса на извлечение данных из таблиц

Результаты						Сообщения	
	id	ФИО	телефон	комната	национальность	специальность	
1	1	Иванов Иван Иванович	+375291234567	203	белорус	Инженерия программного обеспечения	
	id	ФИО	телефон	комната	национальность	специальность	
1	8	Александров Игорь Викторович	+375298901234	802	украинец	Биология	
2	6	Васильева Елена Сергеевна	+375296789012	609	белорус	Физика	
3	16	Волкова Анастасия Викторовна	+375295778899	708	белорус	Химия	
4	17	Егоров Игорь Александрович	+375296889900	803	украинец	Биология	
5	1	Иванов Иван Иванович	+375291234567	203	белорус	Инженерия программного обеспечения	
6	18	Карпова Анастасия Игоревна	+375297900011	904	белорус	Психология	
7	12	Ковалева Екатерина Петровна	+375292334455	308	белорус	Экономика	
8	4	Козлова Ольга Николаевна	+375294567890	408	белорус	Лингвистика	
	id	ФИО	телефон	комната	национальность	специальность	
1	10	Смирнова Анастасия Сергеевна	+375291112233	204	белорус	Инженерия	
2	19	Титова Мария Сергеевна	+375298011122	205	белорус	Инженерия	
	id	ФИО	телефон	комната	национальность	специальность	
1	1	Иванов Иван Иванович	+375291234567	203	белорус	Инженерия программного обеспечения	
2	2	Петров Петр Петрович	+375292345678	205	русский	Медицина	
3	3	Сидорова Анна Павловна	+375293456789	307	украинец	Экономика	
4	4	Козлова Ольга Николаевна	+375294567890	408	белорус	Лингвистика	
5	5	Михайлов Алексей Владимирович	+375295678901	510	русский	Информационные технологии	

Рисунок 4. Результаты SELECT-ов