

**Федеральное государственное автономное образовательное
учреждение высшего образования национальный исследовательский
университет ИТМО**

Факультет программной инженерии и компьютерной техники

Дисциплина «базы данных»

Отчёт

По лабораторной работе №2

Вариант 1992

**Исполнитель:
Назирджонов Некруз
группа: Р33211**

Санкт-Петербург
2023

Текст задания

Для выполнения лабораторной работы №2 необходимо:

- На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
- Составить инфологическую модель.
- Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
- Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
- Заполнить созданные таблицы тестовыми данными.

Для создания объектов базы данных у каждого студента есть своя схема. Название схемы соответствует имени пользователя в базе studs (sXXXXXX). Команда для подключения к базе studs:

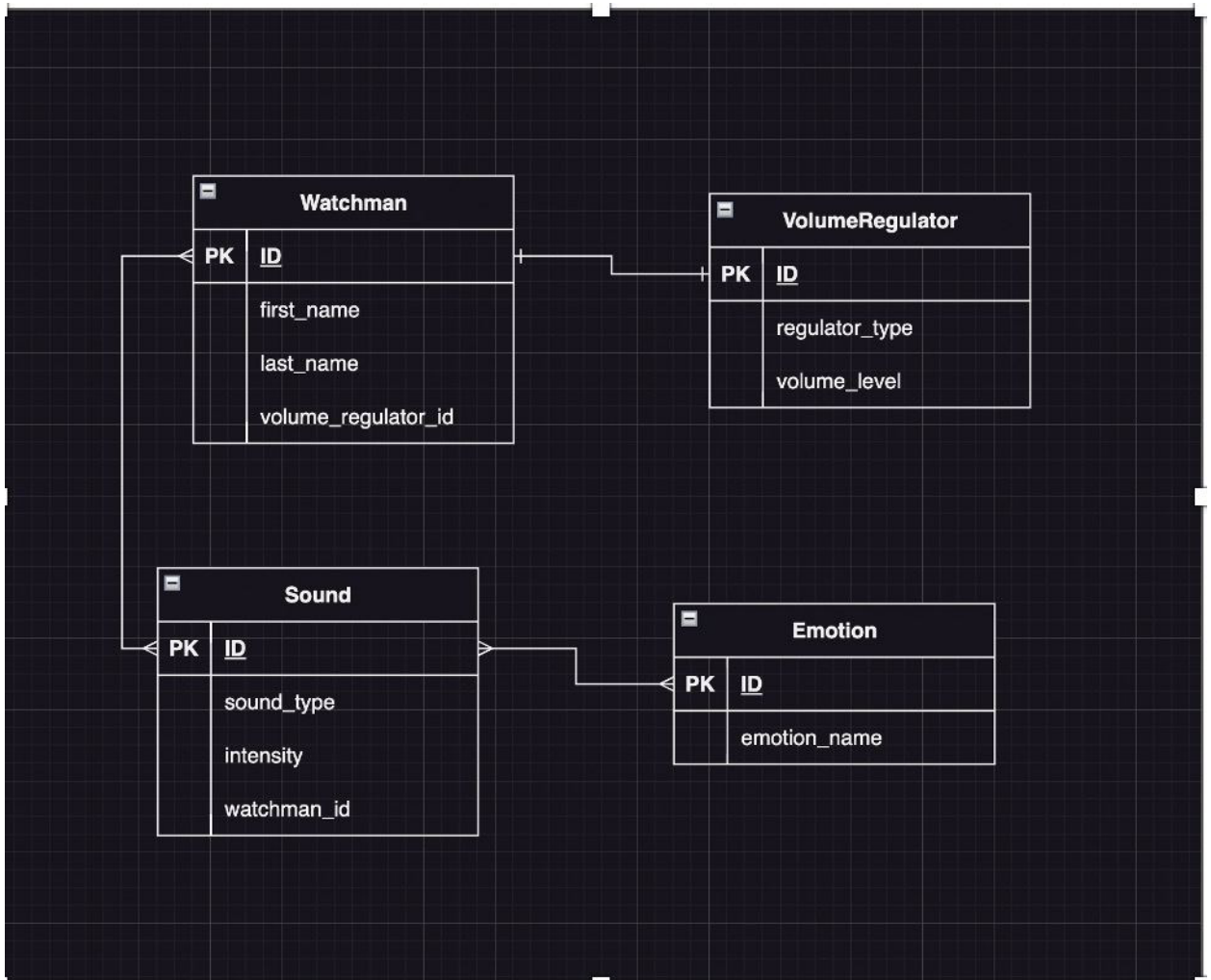
```
psql -h pg -d studs
```

Каждый студент должен использовать свою схему при работе над лабораторной работой №2 (а также в рамках выполнения лабораторных работ 3, 4, 5).

Описание предметной области, по которой должна быть построена доменная модель:

Иногда, в долгие одинокие часы вахты, Боумен слушал эти шумы. Он поворачивал регулятор громкости до тех пор, пока треск и шипение не начинали звучать на всю рубку. Порой на чатом фоне слышались свист и писк, словно кричали обезумевшие от страха птицы. Эти звуки навевали жуть, потому что исходили не от человека; они будили острое чувство одиночества - бессмысленные, как плеск волн о берег, как раскаты дальнего грома за горизонтом...

Инфологическая модель



- Watchman (Боумен):
 - ID (SERIAL PRIMARY KEY): Уникальный идентификатор Боумена.
 - first_name (VARCHAR(50)): Имя Боумена.
 - last_name (VARCHAR(50)): Фамилия Боумена.
 - volume_regulator_id (INT): Внешний ключ, связывающий Боумена с его регулятором громкости (связь 1 к 1).
- VolumeRegulator (Регулятор громкости):
 - ID (SERIAL PRIMARY KEY): Уникальный идентификатор регулятора громкости.
 - regulator_type (VARCHAR(50)): Тип регулятора громкости (например, "Ручной" или "Автоматический").
 - volume_level (INT): Уровень громкости.
- Sound (Звук):
 - ID (SERIAL PRIMARY KEY): Уникальный идентификатор звука.
 - sound_type (VARCHAR(50)): Тип звука (например, "Треск" или "Шипение").
 - intensity (INT): Интенсивность звука.
 - watchman_id (INT): Внешний ключ, связывающий звук с определенным Боуменом (связь 1 ко многим).
- Emotion (Эмоция):

-- Создание таблицы "VolumeRegulator"

-- Создание таблицы "VolumeRegulator"

```

CREATE TABLE VolumeRegulator (
  ID SERIAL PRIMARY KEY,
  regulator_type VARCHAR(50),
  volume_level INT
);

-- Создание таблицы "Watchman"
CREATE TABLE Watchman (
  ID SERIAL PRIMARY KEY,
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  volume_regulator_id INT,
  FOREIGN KEY (volume_regulator_id) REFERENCES VolumeRegulator(ID)
);

-- Создание таблицы "Sound"
CREATE TABLE Sound (
  ID SERIAL PRIMARY KEY,
  sound_type VARCHAR(50),
  intensity INT,
  watchman_id INT,
  FOREIGN KEY (watchman_id) REFERENCES Watchman(ID)
);

-- Создание таблицы "Emotion"
CREATE TABLE Emotion (
  ID SERIAL PRIMARY KEY,
  emotion_name VARCHAR(50)
);

-- Создание таблицы "WatchmanSound" (связь многие ко многим между "Боуменами" и "Звуками")
CREATE TABLE WatchmanSound (
  watchman_id INT,
  sound_id INT,
  PRIMARY KEY (watchman_id, sound_id),
  FOREIGN KEY (watchman_id) REFERENCES Watchman(ID),
  FOREIGN KEY (sound_id) REFERENCES Sound(ID)
);

-- Создание таблицы "SoundEmotion" (связь многие ко многим между "Звуками" и "Эмоциями")
CREATE TABLE SoundEmotion (
  sound_id INT,
  emotion_id INT,
  PRIMARY KEY (sound_id, emotion_id),
  FOREIGN KEY (sound_id) REFERENCES Sound(ID),
  FOREIGN KEY (emotion_id) REFERENCES Emotion(ID)
);

```

Заполнение тестовыми данными

```

-- Заполняем таблицу "VolumeRegulator" данными
INSERT INTO VolumeRegulator (regulator_type, volume_level) VALUES
('Ручной', 50),

```

```
('Автоматический', 75);
```

```
-- Заполняем таблицу "Watchman" данными
```

```
INSERT INTO Watchman (first_name, last_name, volume_regulator_id) VALUES  
('Иван', 'Иванов', 1),  
('Петр', 'Петров', 2);
```

```
-- Заполняем таблицу "Sound" данными
```

```
INSERT INTO Sound (sound_type, intensity, watchman_id) VALUES  
('Треск', 80, 1),  
('Шипение', 70, 1),  
('Свист', 60, 2);
```

```
-- Заполняем таблицу "Emotion" данными
```

```
INSERT INTO Emotion (emotion_name) VALUES  
('Страх'),  
('Удивление');
```

```
-- Заполняем таблицу "WatchmanSound" данными (связь многие ко многим)
```

```
INSERT INTO WatchmanSound (watchman_id, sound_id) VALUES  
(1, 1),  
(1, 2),  
(2, 3);
```

```
-- Заполняем таблицу "SoundEmotion" данными (связь многие ко многим)
```

```
INSERT INTO SoundEmotion (sound_id, emotion_id) VALUES  
(1, 1),  
(3, 2);
```

Пример запросов:

1. Выборка всех Боуменов (Watchman) и их регуляторов громкости:

```
SELECT Watchman.first_name, Watchman.last_name, VolumeRegulator.regulator_type,  
VolumeRegulator.volume_level  
FROM Watchman  
JOIN VolumeRegulator ON Watchman.volume_regulator_id = VolumeRegulator.ID;
```

2. Выборка звуков определенного Боумена (например, Ивана):

```
SELECT Sound.sound_type, Sound.intensity  
FROM Sound  
JOIN Watchman ON Sound.watchman_id = Watchman.ID  
WHERE Watchman.first_name = 'Иван';
```

3. Обновление уровня громкости для ручных регуляторов громкости:

```
UPDATE VolumeRegulator  
SET volume_level = 60  
WHERE regulator_type = 'Ручной';
```

4. Выборка всех Боуменов и связанных с ними звуков и эмоций:

```
SELECT Watchman.first_name, Watchman.last_name, Sound.sound_type,  
Emotion.emotion_name  
FROM Watchman
```

```
JOIN WatchmanSound ON Watchman.ID = WatchmanSound.wathman_id
JOIN Sound ON WatchmanSound.sound_id = Sound.ID
LEFT JOIN SoundEmotion ON Sound.ID = SoundEmotion.sound_id
LEFT JOIN Emotion ON SoundEmotion.emotion_id = Emotion.ID;
```

5. Выборка всех звуков и связанных с ними эмоций:

```
SELECT Sound.sound_type, Emotion.emotion_name
FROM SoundEmotion
JOIN Sound ON SoundEmotion.sound_id = Sound.ID
JOIN Emotion ON SoundEmotion.emotion_id = Emotion.ID;
```

Поиск эмоций, связанных с определенным типом звука (например, "Треск"):

6. Подсчет количества эмоций, связанных с каждым типом звука:

```
SELECT Sound.sound_type, COUNT(Emotion.ID) AS count_emotions
FROM SoundEmotion
JOIN Sound ON SoundEmotion.sound_id = Sound.ID
JOIN Emotion ON SoundEmotion.emotion_id = Emotion.ID
GROUP BY Sound.sound_type;
```