Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

ОТЧЁТ

по лабораторной работе №3

Вариант № 28 (Студия звукозаписи).

Студент: П. В. Сякачёв

Проверил: Л. П. Поденок

МИНСК 2022

**1 Цель работы**

В лабораторной работе выполняется реализация схемы базы данных по ранее построенной реляционной схеме данных «Студия звукозаписи». Требуется сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД. Требуется заполнить таблицы данными с помощью оператора INSERT.

**2 Выполнение работы**

Имеем готовую спроектированную реляционную схему данных модели «Студия звукозаписи». На основании уже созданной реляционной схемы данных, показанной на рисунке 1, создадим в СУБД PostgreSQL новую схему данных для хранения:

CREATE DATABASE "Recording studio"

Далее реализована структура таблиц, включающая в себя:

* 1. набор столбцов с добавлением описаний первичного ключа;
  2. ограничения для описания внешних ключей:
  3. свойства контроля целостности данных для внешних ключей (каскадное удаление и обновление);
  4. ограничения для описания бизнес-правил;

Пример реализации таблицы с описанием первичного ключа:

CREATE TABLE member (

ID SERIAL PRIMARY KEY,

first\_name VARCHAR(128) NOT NULL,

middle\_name VARCHAR(128) NOT NULL,

last\_name VARCHAR(128) NOT NULL,

...

Здесь задаётся первичный ключ типа serial (автоинкрементирующий псевдо-тип данных в PostgreSQL) и другие атрибуты таблицы;

Ограничения для описания внешних ключей и свойства контроля целостности данных для внешних ключей (каскадное удаление и обновление) выглядят следующим образом:

CREATE TABLE recording\_session (

ID SERIAL PRIMARY KEY,

r\_date DATE NOT NULL,

r\_time TIME NOT NULL,

duration FLOAT(2) NOT NULL,

session\_type VARCHAR(64) NOT NULL,

link VARCHAR(128) NULL,

room\_id INT NOT NULL REFERENCES room (ID)

ON DELETE RESTRICT,

band\_id INT NOT NULL REFERENCES band (ID)

ON DELETE CASCADE

);

Здесь задаются ограничения для описания внешних ключей, ссылающихся на таблицы room и band: ключевое слово REFERENCES указывает на то, что ограничение внешнего ключа ссылается на поле таблицы, которая указана после этого ключевого слова. В данном случае, таким ограничением устанавливается запрет на создание сессии звукозаписи, если не существует помещения, в котором она проходит, и нет группы, которую записывают.

Ключевое слово ON DELETE определяет действие при удалении данных в таблице, на которую ссылаются, ON UPDATE определяет действие при обновлении данных этой же таблицы. Но, поскольку первичные ключи везде заданы с помощью serial, то ON UPDATE не имеет смысла, т.к. эти ключи меняться не будут.

CASCADE и RESTRICT задают следующее поведение: первое ключевое слово задаёт в данном случае каскадное удаление строк, связанных с ключом, на который указывает ссылка, а второе ключевое слово запрещает удаление связанной строки. Таким образом, в данном случае, если удаляется запись, содержащая данные о процедурах, то все процедуры удаляются вместе с этой записью. И, соответственно, если существуют процедуры, ссылающиеся на конкретного врача, то данные об этом враче удалить нельзя.

Ограничения для бизнес-правил представляют собой следующее:

CREATE TABLE room (

ID SERIAL PRIMARY KEY,

room\_number INT NOT NULL

CONSTRAINT if\_above\_zero\_room\_number CHECK (room\_number > 0),

area FLOAT(2) NOT NULL,

CONSTRAINT if\_above\_zero\_area CHECK (area > 0),

purpose VARCHAR(64) NOT NULL

…

Ограничения CHECK для бизнес-правил (ограничения-проверки) могут задаваться с использованием слова CONSTRAINT, если хотим указать имя ограничения. Таким образом для room\_number задаётся ограничение, при котором номер аудитории не может быть нулевой или отрицательной. Аналогичная проверка проходит и для поля area.

Далее, заполняем таблицы данными с помощью оператора INSERT INTO. Главное требование: сначала заполняются таблицы, которые не ссылаются на другие таблицы.

Также была создана временная база данных и успешно удалена с помощью оператора DROP DATABASE.



Рисунок 1 – ER-диаграмма



Рисунок 2 – Реляционная диаграмма