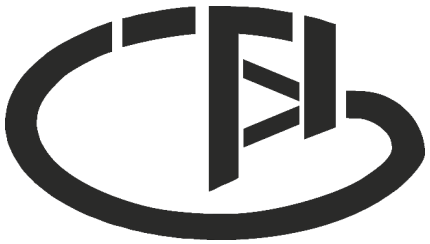


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ

БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

Отчет

по практической работе

«Создание Базы Данных»

Отчет подготовил

студент 3 курса группы УБ-01

Хомутов Константин

Воронеж 2023

Тема 26. Компьютерные занятия: список слушателей курсов, список предметов, список преподавателей, журнал учета успеваемости.

Составление модели

Студенты должны выбирать курс, на котором хотят учиться. То есть, должна быть сущность «Предмет», которая будет состоять из названия и идентификатора. Сущность студента будет иметь в себе: ФИО, паспортные данные, адрес проживания, электронная почта. Также, должны быть сущности преподавателя (ФИО, паспорт, адрес, электронная почта, курс, на котором преподает). При этом эти три сущности вступают в различные отношения:

1. студент → учится ← предмет

Один студент может обучаться множеству предметов, и на одном курсе может учиться множество студентов. Отношение «Учится» имеет связь «многие ко многим». Таким образом, нужно создать сущность, которая описывала бы принадлежность студента к определенному курсу.

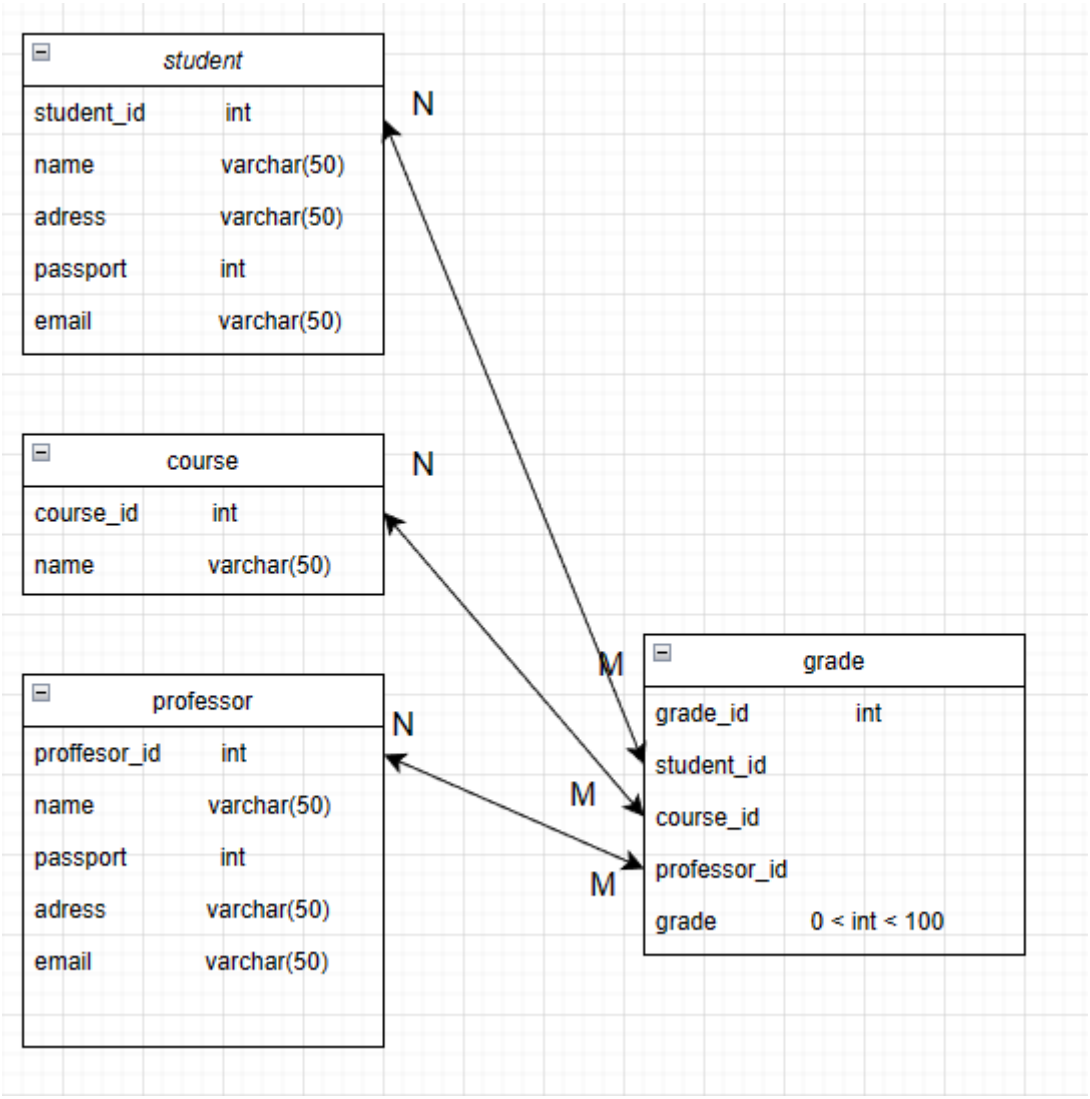
2. преподаватель → преподает ← предмет

Такая же ситуация в отношении «Преподает». Отношение имеет связь «многие ко многим».

3. преподаватель → оценивает ← студента

Аналогичная ситуация — преподаватель может ставить оценки разным студентам, студент может получать оценки от разных преподавателей.

При создании всех отношений между сущностями студент, преподаватель и курс можно заметить, что отношение «Оценка» содержит в себе идентификатор курса, преподавателя и студента. Таким образом, через отношение «Оценка» можно определить по каким предметам студент учится, какие предметы ведет преподаватель, какие баллы имеет студент по определенному предмету. Исходя из этого, само поле «оценка» может быть пустым, потому что студент может учиться на курсе, но еще не иметь рейтинг.



Создание базы данных и подключение к ней

```
postgres=# create database itcourses;
CREATE DATABASE
postgres=# \connect itcourses;
Вы подключены к базе данных "itcourses" как пользователь "postgres".
itcourses=#
```

Создание таблиц

Создание сущности Курс:

Команда - create table *название таблицы*. Ввод данных осуществляется через обозначение названия аргумента, после чего идет тип

данных. Если поле, которое нужно создать, нельзя оставлять пустым, то используется параметр NOT NULL.

```
itcourses=# create table course (
itcourses(# course_id int NOT NULL,
itcourses(# course_name varchar(50) NOT NULL,
itcourses(# constraint coursePk
itcourses(# primary key (course_id));
CREATE TABLE
itcourses=#
```

Создание сущности Студент:

```
itcourses=# CREATE TABLE student (
itcourses(# student_id int NOT NULL,
itcourses(# name varchar(50) NOT NULL,
itcourses(# adress varchar(50) NOT NULL,
itcourses(# passport int NOT NULL,
itcourses(# email varchar(50) NOT NULL,
itcourses(# constraint studentPk
itcourses(# primary key (student_id)
itcourses(# );
CREATE TABLE
itcourses=#
```

Создание сущности Преподаватель:

```
itcourses=# CREATE TABLE professor (
itcourses(# professor_id int NOT NULL,
itcourses(# name varchar(50) NOT NULL,
itcourses(# adress varchar(50) NOT NULL,
itcourses(# passport int NOT NULL,
itcourses(# email varchar(50) NOT NULL,
itcourses(# constraint professorPk
itcourses(# primary key (professor_id)
itcourses(# );
CREATE TABLE
```

Создание сущности Оценка:

```
itcourses=# CREATE TABLE grade (
itcourses(# grade_id int NOT NULL,
itcourses(# student_id int NOT NULL,
itcourses(# course_id int NOT NULL,
itcourses(# professor_id int NOT NULL,
itcourses(# grade int NOT NULL,
itcourses(# constraint lowgradestop CHECK
itcourses(# (0 <= grade),
itcourses(# constraint highgradestop CHECK
itcourses(# (grade <= 100),
itcourses(# constraint gradePk
itcourses(# primary key (grade_id),
itcourses(# constraint studentFk
itcourses(# foreign key (student_id)
itcourses(# references student(student_id)
itcourses(# on delete cascade,
itcourses(# constraint courseFk
itcourses(# foreign key (course_id)
itcourses(# references course(course_id)
itcourses(# on delete cascade,
itcourses(# constraint professorFk
itcourses(# foreign key (professor_id)
itcourses(# references professor(professor_id)
itcourses(# on delete cascade
itcourses(# );
CREATE TABLE
```

В данной сущности оценка представляет собой балл от 0 до 100 и реализуется через добавления двух ограничений:

constraint lowgradestop CHECK (0 <= grade),

constraint highgradestop CHECK (grade <= 100),

Лист созданных таблиц:

```
itcourses=# \dt
            Список отношений
 Схема  |    Имя    |   Тип   | Владелец
--------+-----------+---------+----------
 public | course    | таблица | postgres
 public | grade     | таблица | postgres
 public | professor | таблица | postgres
 public | student   | таблица | postgres
(4 строки)
```

Создание последовательностей

Последовательность — это объект, который генерирует ряд последовательных уникальных чисел, которые используются для формирования

первичных ключей. Синтаксис добавление последовательности на примере последовательности, созданной для курса:

CREATE SEQUENCE seq_course

INCREMENT BY 1

START WITH 101

MINVALUE 100;

Последовательность начинается с 101, при добавлении нового экземпляра последовательность будет увеличиваться на единицу.

Лист созданных последовательностей можно вызвать командой \ds

```
itcourses=# \ds
                    Список отношений
 Схема  |      Имя      |        Тип         | Владелец
--------+---------------+--------------------+----------
 public | seq_course    | последовательность | postgres
 public | seq_grade     | последовательность | postgres
 public | seq_professor | последовательность | postgres
 public | seq_student   | последовательность | postgres
(4 строки)
```

Ввод данных

Названия курсов

Будут четыре курса: программирование, устройство компьютера и операционные системы, компьютерные сети. Добавление данных происходит с помощью команды

insert into *название таблицы* values

(*передача параметров в соответствии с полями*);

```
itcourses=# insert into course values
itcourses-# (nextval('seq_course'), 'программирование');
INSERT 0 1
itcourses=# insert into course values
itcourses-# (nextval('seq_course'), 'устройство компьютера');
INSERT 0 1
itcourses=# insert into course values
itcourses-# (nextval('seq_course'), 'операционные системы');
INSERT 0 1
itcourses=# insert into course values
itcourses-# (nextval('seq_course'), 'компьютерные сети');
INSERT 0 1
itcourses=#
```

Вывести список всех добавленных объектов можно через команду select * from *название таблицы*

```
itcourses=# SELECT * FROM course;
 course_id |      course_name
-----------+-----------------------
       101 | программирование
       102 | устройство компьюетра
       103 | операционные системы
       104 | компьютерные сети
(4 строки)
```

Персональные данные студентов:

Студенты

имена: Николаев Терентий Вячеславович, Кудрявцев Альберт Миронович, Тетерин Климент Денисович, Мышкина Дария Кимовна, Капустина Влада Мироновна;

адреса: Чехова 62, проезд Бухарестская 49, бульвар Космонавтов 70, ул1905года31, наб Ломоносова 25;

паспорт: 2022140694, 2022940004, 2022516896, 2022724043, 2022806770;

почта: nikolai2002@oi.com, kudmir@oi.com, tete@oi.com, YAMbISH@oi.com, nemiron@oi.com.

```
itcourses=# SELECT * FROM student;
 student_id |             name             |         adress         |  passport  |       email
------------+------------------------------+------------------------+------------+--------------------
        201 | Николаев Терентий Вячеславович | Чехова 62              | 2022140694 | nikolai2002@oi.com
        202 | Кудрявцев Альберт Миронович    | проезд Бухарестская 49 | 2022940004 | kudmir@oi.com
        203 | Тетерин Климент Денисович      | бульвар Космонавтов 70 | 2022516896 | tete@oi.com
        204 | Мышкина Дария Кимовна          | ул1905года31           | 2022724043 | YAMbISH@oi.com
        205 | Капустина Влада Мироновна      | наб Ломоносова 25      | 2022806770 | nemiron@oi.com
(5 строк)
```

Преподаватели

Будет два преподавателя, которые будут преподавать по два предмета

имена: Беляков Дмитрий Куприянович, Лебедева Марта Кирилловна;

адреса: пл. Косиора, 50, пл. Чехова, 97;

паспорт: 2002784788, 2005450253;

почта: dmitkup@oi.com, lebmarkir@oi.com.

```
itcourses=# SELECT * FROM professor;
 professor_id |            name             |     adress      |  passport  |      email
--------------+-----------------------------+-----------------+------------+------------------
          301 | Беляков Дмитрий Куприянович | пл. Косиора, 50 | 2002784788 | dmitkup@oi.com
          302 | Лебедева Марта Кирилловна   | пл. Чехова, 97  | 2005450253 | lebmarkir@oi.com
(2 строки)
```

Каждый студент будет учиться на двух курсах. Для этого просто добавим 10 экземпляров с соответствующими индексами:

Заполнение таблицы grade:

```
itcourses=# select * from grade;
 grade_id | student_id | course_id | professor_id | grade
----------+------------+-----------+--------------+-------
      401 |        201 |       101 |          301 |    80
      402 |        201 |       102 |          301 |    90
      403 |        202 |       102 |          301 |    75
      404 |        202 |       103 |          302 |    60
      405 |        203 |       103 |          302 |    85
      406 |        203 |       104 |          302 |    65
      407 |        204 |       101 |          301 |    93
      408 |        204 |       104 |          302 |   100
      409 |        205 |       101 |          301 |    90
      410 |        205 |       102 |          301 |    90
(10 строк)
```

Добавление индексов

Индексы нужны для возможно поиска не по его идентификатору. Так, студента и преподавателя можно искать по имени, курс по названию, а оценку по баллу.

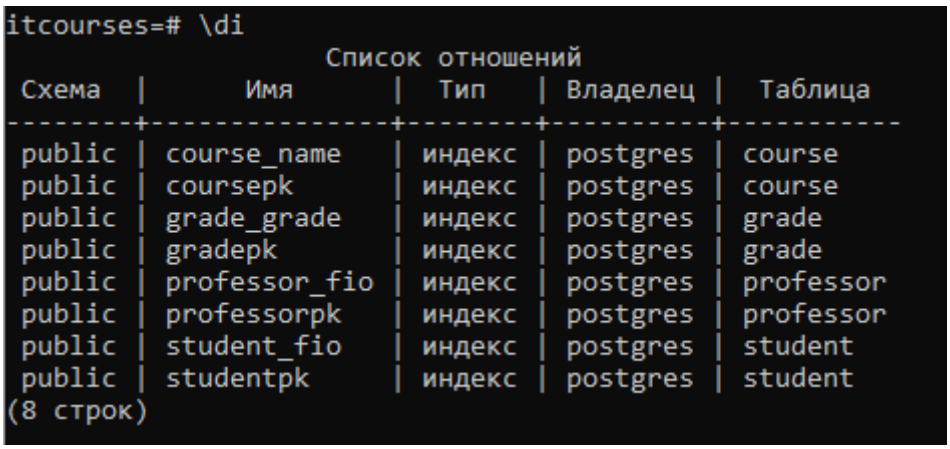
Синтаксис команд добавления индекса следующий:

create index *название индекса*

on *название таблицы* using hash (*название поля для поиска*);

В работе создавались только hash индексы, но так же имеются индексы в виде бинарного дерева, которые позволяют увеличить скорость поиска.

Список индексов выводится командой \di, помимо созданных индексов, список содержит еще и идентификаторы.



```
itcourses=# \di
                    Список отношений
 Схема  |      Имя      |  Тип   | Владелец |  Таблица
--------+---------------+--------+----------+-----------
 public | course_name   | индекс | postgres | course
 public | coursepk      | индекс | postgres | course
 public | grade_grade   | индекс | postgres | grade
 public | gradepk       | индекс | postgres | grade
 public | professor_fio | индекс | postgres | professor
 public | professorpk   | индекс | postgres | professor
 public | student_fio   | индекс | postgres | student
 public | studentpk     | индекс | postgres | student
(8 строк)
```

Вывод: в данной работе была создана база данных «Компьютерные курсы», в которую были добавлены таблицы курса, студента, преподавателя, и журнал успеваемости. К созданным таблицам были добавлены последовательности и индексы.