



**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 «Курс», которая будет состоять из названия и идентификатора. Сущность студента будет иметь в себе: ФИО, паспортные данные, адрес проживания, электронная почта, идентификатор курса, на котором учится. Также, должны быть сущности преподавателя (ФИО, паспорт, адрес, электронная почта, курс, на котором преподает) и оценка (идентификатор студента, идентификатор курса, балл).

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(# student_name varchar(50) NOT NULL,
itcourses(# student_passport int NOT NULL,
itcourses(# student_adress varchar(50) NOT NULL,
itcourses(# student_email varchar(50) NOT NULL,
itcourses(# course_id int NOT NULL,
itcourses(# constraint studentPk
itcourses(# primary key (student_id),
itcourses(# constraint courseFk
itcourses(# foreign key (course_id)
itcourses(# references course(course_id)
itcourses(# on delete cascade);
CREATE TABLE

```

Создание сущности Преподаватель:

```

itcourses=# create table teacher(
itcourses(# teacher_id int NOT NULL,
itcourses(# teacher_name varchar(50) NOT NULL,
itcourses(# teacher_passport int NOT NULL,
itcourses(# teacher_adress varchar(50) NOT NULL,
itcourses(# teacher_email varchar(50) NOT NULL,
itcourses(# course_id int NOT NULL,
itcourses(# constraint teacherPk
itcourses(# primary key (teacher_id),
itcourses(# constraint courseFk
itcourses(# foreign key (course_id)
itcourses(# references course(course_id)
itcourses(# on delete cascade);
CREATE TABLE

```

Создание сущности Оценка:

```

itcourses=# create table grade(
itcourses(# grade_id int NOT NULL,
itcourses(# course_id int NOT NULL,
itcourses(# student_id int NOT NULL,
itcourses(# grade char NULL,
itcourses(# constraint gradePk
itcourses(# primary key (grade_id),
itcourses(# constraint courseFk
itcourses(# foreign key (course_id)
itcourses(# references course(course_id)
itcourses(# on delete cascade,
itcourses(# constraint studentFk
itcourses(# foreign key (student_id)
itcourses(# references student(student_id)
itcourses(# on delete cascade);
CREATE TABLE

```

Лист созданных таблиц:

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

Создание последовательности для всех таблиц

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

```
itcourses=# create sequence seq_course
itcourses=# increment by 1
itcourses=# start with 101
itcourses=# minvalue 100;
CREATE SEQUENCE
itcourses=# create sequence seq_student
itcourses=# increment by 1
itcourses=# start with 201
itcourses=# minvalue 200;
CREATE SEQUENCE
itcourses=# create sequence seq_teacher
itcourses=# increment by 1
itcourses=# start with 301
itcourses=# minvalue 300;
CREATE SEQUENCE
itcourses=# create sequence seq_grade
itcourses=# increment by 1
itcourses=# start with 401
itcourses=# minvalue 400;
CREATE SEQUENCE
itcourses=# \ds
          Список отношений
 Схема |   Имя   | Тип           | Владелец
-----+-----+-----+-----
public | seq_course | последовательность | postgres
public | seq_grade  | последовательность | postgres
public | seq_student | последовательность | postgres
public | seq_teacher | последовательность | postgres
(4 строки)
```

Ввод данных

Названия курсов

Будут три курса: программирование, устройство компьютера и операционные системы. 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
```

Вывести список всех добавленных объектов можно через команду

`select * from *название таблицы*`

```
itcourses=# select * from course;
 course_id |      course_name
-----+-----
          101 | программирование
          102 | устройство компьютера
          103 | операционные системы
(3 строки)
```

Персональные данные студентов:

```
itcourses=# insert into student values
itcourses=# (nextval('seq_student'), 'Николаев Терентий Вячеславович', 2022140694, 'Чехова62',
'nikolai2002@oi.com', 101);
INSERT 0 1
itcourses=# insert into student values
itcourses=# (nextval('seq_student'), 'Кудрявцев Альберт Миронович', 2022940004, 'проездБухарес
тская49', 'kudmin@oi.com', 102);
INSERT 0 1
itcourses=# insert into student values
itcourses=# (nextval('seq_student'), 'Тетерин Климент Денисович', 2022516896, 'бульварКосмонав
тов70', 'tete@oi.com', 103);
INSERT 0 1
itcourses=# insert into student values
itcourses=# (nextval('seq_student'), 'Мышкина Дария Кимовна', 2022724043, 'ул1905года31', 'YAM
bISH@oi.com', 101);
INSERT 0 1
itcourses=# insert into student values
itcourses=# (nextval('seq_student'), 'Капустина Влада Мироновна', 2022806770, 'набЛомоносова2
5', 'nemiron@oi.com', 102);
INSERT 0 1
```

```
itcourses=# select * from student
itcourses-# ;
```

student_id	student_name	student_passport	student_adress	student_email	course_id
201	Николаев Терентий Вячеславович	2022140694	Чехова62	nikolai2002@oi.com	101
202	Кудрявцев Альберт Миронович	2022940004	проездБухарестская49	kudmir@oi.com	102
203	Тетерин Климент Денисович	2022516896	бульварКосмонавтов70	tete@oi.com	103
204	Мышкина Дария Кимовна	2022724043	ул1905года31	YAMbISH@oi.com	101
205	Капустина Влада Мироновна	2022806770	набЛомоносова25	nemiron@oi.com	102

(5 строк)

Преподаватели

```
itcourses=# insert into teacher values
itcourses-# (nextval('seq_teacher'), 'Андреев Валентин Аристархович', 2012182240, 'наб. Гоголя, 57', 'andrew-val@oi.com', 101);
INSERT 0 1
itcourses=# insert into teacher values
itcourses-# (nextval('seq_teacher'), 'Беляков Дмитрий Куприянович', 2002784788, 'пл. Косиора, 50', 'dmitkup@oi.com', 102);
INSERT 0 1
itcourses=# insert into teacher values
itcourses-# (nextval('seq_teacher'), 'Лебедева Марта Кирилловна', 2005450253, 'пл. Чехова, 97', 'lebmarkir@oi.com', 103);
INSERT 0 1
itcourses=# select * from teacher
itcourses-# ;
```

teacher_id	teacher_name	teacher_passport	teacher_adress	teacher_email	course_id
304	Андреев Валентин Аристархович	2012182240	наб. Гоголя, 57	andrew-val@oi.com	101
305	Беляков Дмитрий Куприянович	2002784788	пл. Косиора, 50	dmitkup@oi.com	102
306	Лебедева Марта Кирилловна	2005450253	пл. Чехова, 97	lebmarkir@oi.com	103

(3 строки)

Оценки

```
itcourses=# insert into grade values
itcourses-# (nextval('seq_grade'), 101, 201, '5');
INSERT 0 1
itcourses=# insert into grade values
itcourses-# (nextval('seq_grade'), 102, 202, '5');
INSERT 0 1
itcourses=# insert into grade values
itcourses-# (nextval('seq_grade'), 103, 203, '3');
INSERT 0 1
itcourses=# insert into grade values
itcourses-# (nextval('seq_grade'), 101, 204, '4');
INSERT 0 1
itcourses=# insert into grade values
itcourses-# (nextval('seq_grade'), 102, 205, '5');
INSERT 0 1
itcourses=# select * from grade;
```

grade_id	course_id	student_id	grade
401	101	201	5
402	102	202	5
403	103	203	3
404	101	204	4
405	102	205	5

(5 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=# create index student_fio  
itcourses=# on student using hash (student_name);  
CREATE INDEX  
itcourses=# create index teacher_fio  
itcourses=# on teacher using hash (teacher_name);  
CREATE INDEX  
itcourses=# create index course_index  
itcourses=# on course using hash (course_name);  
CREATE INDEX  
itcourses=# create index grade_symbol  
itcourses=# on grade using hash (grade);  
CREATE INDEX  
itcourses=# \di
```

Список отношений				
Схема	Имя	Тип	Владелец	Таблица
public	course_index	индекс	postgres	course
public	coursepk	индекс	postgres	course
public	grade_symbol	индекс	postgres	grade
public	gradepk	индекс	postgres	grade
public	student_fio	индекс	postgres	student
public	studentpk	индекс	postgres	student
public	teacher_fio	индекс	postgres	teacher
public	teacherpk	индекс	postgres	teacher

(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.