**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине «Базы данных»

Разработка физической модели данных и создание БД

Студент Станиславчук С.М.

Группа АС-21-1

Руководитель Алексеев В.А.

Липецк 2023 г.

Цель работы

Изучить задачи физического проектирования базы данных. Освоить на практике разработку физической модели данных в среде Sybase PowerDesigner и модификацию схемы данных с использованием SQL-запросов.

Задание кафедры

Сформировать в среде PowerDesigner физическую модель данных на основе логической модели, разработанной в лабораторной работе №2. Создать базу данных средствами выбранной СУБД и отработать сценарий модификации схемы данных.

Ход работы

На рисунке 1 представлена диаграмма физической модели.

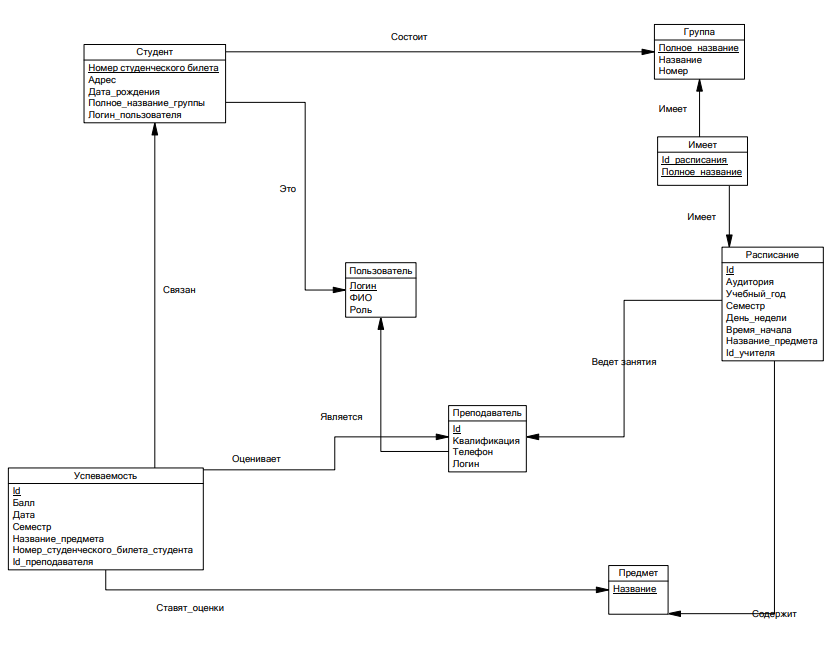


Рисунок 1 – Физическая модель

Спецификация таблиц

Спецификация таблиц физической модели составлена и заполнена в виде таблицы 1.

Таблица 1 – Спецификация сущностей физической модели

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Имя сущности** | **Код сущности** | **Имя атрибута** | **Код атрибута** | **Обязательность** | **Первичный ключ** | **Внешний ключ** | **Домен** |
| Студент | Student | Номер студенческого билета | id\_number\_student | T | T | F | Number (10) |
| Дата рождения | birth\_date\_student | T | F | F | Date |
| Адрес | address\_student | T | F | F | VARCHAR |
| **Полное\_название\_группы** | **full\_name\_group** | T | F | T | VARCHAR (256) |
| **Логин\_пользователя** | **login\_user** | T | F | T | VARCHAR (256) |
| Группа | Group | Полное название | full\_name\_group | T | T | F | VARCHAR |
| Название | name\_group | T | F | F | VARCHAR (10) |
| Номер | number\_group | T | F | F | Number |
| Преподаватель | Teacher | Id | Id\_teacher | T | T | F | Serial |
| Квалификация | qualification\_teacher | T | F | F | VARCHAR |
| Телефон | phone\_number\_teacher | F | F | F | NUMBER (11) |
| Логин | login\_user | T | F | T | VARCHAR (256) |
| Пользователь | User | Логин | login\_user | T | T | F | VARCHAR |
| ФИО | full\_name\_user | T | F | F | VARCHAR |
| Роль | role\_user | T | F | F | VARCHAR |
| Расписание | Schedule | Id | id\_schedule | T | T | F | Serial |
| Аудитория | class\_schedule | T | F | F | VARCHAR |
| Учебный год | academic\_year\_schedule | T | F | F | Number (9) |
| Семестр | semetster\_sche dule | T | F | F | Number (1) |
| День недели | weekday\_schedule | T | F | F | VARCHAR |
| Время начала | start\_time\_schedule | T | F | F | Time |
| Название\_предмета | name\_subject | T | F | T | VARCHAR (256) |
| Id\_учителя | Id\_teacher | T | F | T | INT4 |
| Предмет | Subject | Название | name\_subject | T | T | F | VARCHAR |
| Успеваемость | Perfomance | Id | Id\_perfomance | T | T | F | Serial |
| Балл | mark\_perfomance | T | F | F | Number (3) |
| Дата | date\_perfomance | T | F | F | Date |
| Семестр | semester\_perfo mance | T | F | F | VARCHAR (6) |
| Название\_предмета | name\_subject | T | F | T | VARCHAR(256) |
| Номер\_студенческого\_билета\_студента | Id\_number\_student | T | F | T | NUMERIC |
| Id\_преподавателя | Id\_teacher | T | F | T | INT4 |
| Имеет | group\_has\_schedules | Id\_расписания | Id\_schedule | T | T | T | INT4 |
| Полное\_название | full\_name\_group | T | T | T | VARCHAR(256) |

Создание базы данных

Сгенерированный SQL-скрипт:

/\*==============================================================\*/

/\* DBMS name: PostgreSQL 8 \*/

/\* Created on: 26.11.2023 20:41:34 \*/

/\*==============================================================\*/

drop index if exists Group\_PK;

drop table if exists "Group";

drop index if exists teacher\_perfomance\_FK;

drop index if exists student\_perfomance\_FK;

drop index if exists subject\_perfomance\_FK;

drop index if exists Perfomance\_PK;

drop table if exists Perfomance;

drop index if exists teacher\_schedule\_FK;

drop index if exists schedule\_subject\_FK;

drop index if exists Schedule\_PK;

drop table if exists Schedule;

drop index if exists student\_is\_user\_FK;

drop index if exists student\_group\_FK;

drop index if exists Student\_PK;

drop table if exists Student;

drop index if exists Subject\_PK;

drop table if exists Subject;

drop index if exists teacher\_is\_user\_FK;

drop index if exists Teacher\_PK;

drop table if exists Teacher;

drop index if exists User\_PK;

drop table if exists "User";

drop index if exists schedule\_has\_group\_FK;

drop index if exists group\_has\_schedules\_FK;

drop index if exists group\_has\_schedules\_PK;

drop table if exists group\_has\_schedules;

/\*==============================================================\*/

/\* Table: "Group" \*/

/\*==============================================================\*/

create table "Group" (

full\_name\_group VARCHAR(256) not null,

name\_group VARCHAR(10) not null,

number\_group NUMERIC not null,

constraint PK\_GROUP primary key (full\_name\_group)

);

/\*==============================================================\*/

/\* Index: Group\_PK \*/

/\*==============================================================\*/

create unique index Group\_PK on "Group" (

full\_name\_group

);

/\*==============================================================\*/

/\* Table: Perfomance \*/

/\*==============================================================\*/

create table Perfomance (

Id\_perfomance SERIAL not null,

mark\_perfomance NUMERIC(3) not null,

date\_perfomance DATE not null,

semester\_perfomance VARCHAR(6) not null,

name\_subject VARCHAR(256) not null,

Id\_number\_student NUMERIC not null,

Id\_teacher INT4 not null,

constraint PK\_PERFOMANCE primary key (Id\_perfomance)

);

/\*==============================================================\*/

/\* Index: Perfomance\_PK \*/

/\*==============================================================\*/

create unique index Perfomance\_PK on Perfomance (

Id\_perfomance

);

/\*==============================================================\*/

/\* Index: subject\_perfomance\_FK \*/

/\*==============================================================\*/

create index subject\_perfomance\_FK on Perfomance (

name\_subject

);

/\*==============================================================\*/

/\* Index: student\_perfomance\_FK \*/

/\*==============================================================\*/

create index student\_perfomance\_FK on Perfomance (

Id\_number\_student

);

/\*==============================================================\*/

/\* Index: teacher\_perfomance\_FK \*/

/\*==============================================================\*/

create index teacher\_perfomance\_FK on Perfomance (

Id\_teacher

);

/\*==============================================================\*/

/\* Table: Schedule \*/

/\*==============================================================\*/

create table Schedule (

Id\_schedule SERIAL not null,

class\_schedule VARCHAR(256) not null,

academic\_year\_schedule NUMERIC(9) not null,

semester\_schedule NUMERIC(1) not null,

weekday\_schedule VARCHAR(256) not null,

start\_time\_schedule TIME not null,

name\_subject VARCHAR(256) not null,

Id\_teacher INT4 null,

constraint PK\_SCHEDULE primary key (Id\_schedule)

);

/\*==============================================================\*/

/\* Index: Schedule\_PK \*/

/\*==============================================================\*/

create unique index Schedule\_PK on Schedule (

Id\_schedule

);

/\*==============================================================\*/

/\* Index: schedule\_subject\_FK \*/

/\*==============================================================\*/

create index schedule\_subject\_FK on Schedule (

name\_subject

);

/\*==============================================================\*/

/\* Index: teacher\_schedule\_FK \*/

/\*==============================================================\*/

create index teacher\_schedule\_FK on Schedule (

Id\_teacher

);

/\*==============================================================\*/

/\* Table: Student \*/

/\*==============================================================\*/

create table Student (

id\_number\_student NUMERIC not null,

address\_student VARCHAR(256) not null,

birth\_date\_student DATE not null,

full\_name\_group VARCHAR(256) not null,

login\_user VARCHAR(256) not null,

constraint PK\_STUDENT primary key (id\_number\_student)

);

/\*==============================================================\*/

/\* Index: Student\_PK \*/

/\*==============================================================\*/

create unique index Student\_PK on Student (

id\_number\_student

);

/\*==============================================================\*/

/\* Index: student\_group\_FK \*/

/\*==============================================================\*/

create index student\_group\_FK on Student (

full\_name\_group

);

/\*==============================================================\*/

/\* Index: student\_is\_user\_FK \*/

/\*==============================================================\*/

create index student\_is\_user\_FK on Student (

login\_user

);

/\*==============================================================\*/

/\* Table: Subject \*/

/\*==============================================================\*/

create table Subject (

name\_subject VARCHAR(256) not null,

constraint PK\_SUBJECT primary key (name\_subject)

);

/\*==============================================================\*/

/\* Index: Subject\_PK \*/

/\*==============================================================\*/

create unique index Subject\_PK on Subject (

name\_subject

);

/\*==============================================================\*/

/\* Table: Teacher \*/

/\*==============================================================\*/

create table Teacher (

Id\_teacher SERIAL not null,

qualification\_teacher VARCHAR(256) not null,

phone\_number\_teacher NUMERIC(11) null,

login\_user VARCHAR(256) not null,

constraint PK\_TEACHER primary key (Id\_teacher)

);

/\*==============================================================\*/

/\* Index: Teacher\_PK \*/

/\*==============================================================\*/

create unique index Teacher\_PK on Teacher (

Id\_teacher

);

/\*==============================================================\*/

/\* Index: teacher\_is\_user\_FK \*/

/\*==============================================================\*/

create index teacher\_is\_user\_FK on Teacher (

login\_user

);

/\*==============================================================\*/

/\* Table: "User" \*/

/\*==============================================================\*/

create table "User" (

login\_user VARCHAR(256) not null,

full\_name\_user VARCHAR(256) not null,

role\_user VARCHAR(256) not null,

constraint PK\_USER primary key (login\_user)

);

/\*==============================================================\*/

/\* Index: User\_PK \*/

/\*==============================================================\*/

create unique index User\_PK on "User" (

login\_user

);

/\*==============================================================\*/

/\* Table: group\_has\_schedules \*/

/\*==============================================================\*/

create table group\_has\_schedules (

Id\_schedule INT4 not null,

full\_name\_group VARCHAR(256) not null,

constraint PK\_GROUP\_HAS\_SCHEDULES primary key (Id\_schedule, full\_name\_group)

);

/\*==============================================================\*/

/\* Index: group\_has\_schedules\_PK \*/

/\*==============================================================\*/

create unique index group\_has\_schedules\_PK on group\_has\_schedules (

Id\_schedule,

full\_name\_group

);

/\*==============================================================\*/

/\* Index: group\_has\_schedules\_FK \*/

/\*==============================================================\*/

create index group\_has\_schedules\_FK on group\_has\_schedules (

full\_name\_group

);

/\*==============================================================\*/

/\* Index: schedule\_has\_group\_FK \*/

/\*==============================================================\*/

create index schedule\_has\_group\_FK on group\_has\_schedules (

Id\_schedule

);

alter table Perfomance

add constraint FK\_PERFOMAN\_STUDENT\_P\_STUDENT foreign key (Id\_number\_student)

references Student (id\_number\_student)

on delete cascade on update cascade;

alter table Perfomance

add constraint FK\_PERFOMAN\_SUBJECT\_P\_SUBJECT foreign key (name\_subject)

references Subject (name\_subject)

on delete cascade on update cascade;

alter table Perfomance

add constraint FK\_PERFOMAN\_TEACHER\_P\_TEACHER foreign key (Id\_teacher)

references Teacher (Id\_teacher)

on delete cascade on update cascade;

alter table Schedule

add constraint FK\_SCHEDULE\_SCHEDULE\_\_SUBJECT foreign key (name\_subject)

references Subject (name\_subject)

on delete set null on update restrict;

alter table Schedule

add constraint FK\_SCHEDULE\_TEACHER\_S\_TEACHER foreign key (Id\_teacher)

references Teacher (Id\_teacher)

on delete set null on update restrict;

alter table Student

add constraint FK\_STUDENT\_STUDENT\_G\_GROUP foreign key (full\_name\_group)

references "Group" (full\_name\_group)

on delete restrict on update restrict;

alter table Student

add constraint FK\_STUDENT\_STUDENT\_I\_USER foreign key (login\_user)

references "User" (login\_user)

on delete restrict on update restrict;

alter table Teacher

add constraint FK\_TEACHER\_TEACHER\_I\_USER foreign key (login\_user)

references "User" (login\_user)

on delete restrict on update restrict;

alter table group\_has\_schedules

add constraint FK\_GROUP\_HA\_GROUP\_HAS\_GROUP foreign key (full\_name\_group)

references "Group" (full\_name\_group)

on delete set null on update restrict;

alter table group\_has\_schedules

add constraint FK\_GROUP\_HA\_SCHEDULE\_\_SCHEDULE foreign key (Id\_schedule)

references Schedule (Id\_schedule)

on delete set null on update restrict;

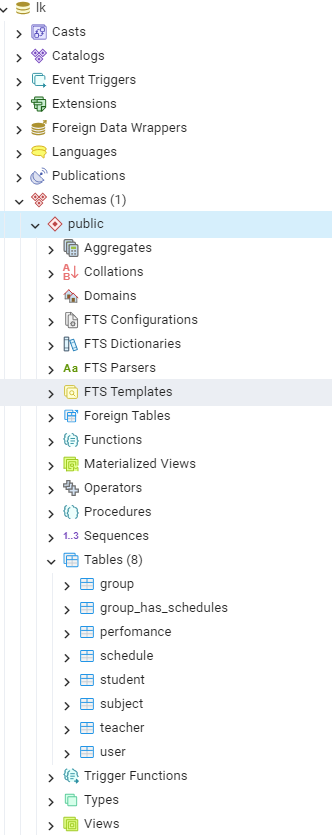


Рисунок 2 – Скриншот управления выбранной СУБД

Сценарий модификации БД

Описание сценария

Сценарий заключается в том, чтобы добавить новую таблицу с информацией о поставке книг и учебников в библиотеку ВУЗа. В ней будут атрибуты: «Название учебного материала», «Количество», «Дата доставки».

Код SQL-запросов модификации базы данных:

-- Создание таблицы

CREATE TABLE supply (

IdSupply SERIAL PRIMARY KEY,

materialName VARCHAR(100) NOT NULL,

quantity INT NOT NULL,

deliveryDate DATE NOT NULL

);

-- Добавление колонки «Название предмета» в таблицу

ALTER TABLE supply

ADD COLUMN subjectName VARCHAR (256) NOT NULL;

-- Создание ограничения внешнего ключа для связи новой таблицы с существующей (учебный материал связан с предметами)

ALTER TABLE supply

ADD CONSTRAINT subjectSupply

FOREIGN KEY (subjectName) REFERENCES subject(name\_subject);

-- Модификация колонки в таблице

ALTER TABLE supply

ALTER COLUMN materialName type varchar (90);

-- Удаление колонки в таблице

ALTER TABLE supply

DROP COLUMN deliveryDate;

-- Удаление таблицы

DROP TABLE supply;

Вывод

По итогу выполнения данной лабораторной работы я изучил задачи физического проектирования базы данных. А также освоил на практике разработку физической модели данных в среде Sybase PowerDesigner и модификацию схемы данных с использованием SQL-запросов.