**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

A logo of a triangle

Description automatically generated

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Институт | компьютерных наук |
| Кафедра | автоматизированных систем управления |

ЛАБОРАТОРНАЯ РАБОТА №3

По дисциплине: "Базы данных"

На тему: "Разработка физической модели базы данных и создание БД"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ПИ-22-1 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Насонов Н.С. |
| Руководитель | |  |  |  |
| к.т.н., доцент  ученая степень, ученое звание | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Алексеев В.А. |

Липецк, 2024 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**

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.

1. **Физическая модель**
   1. **Диаграмма физической модели**

Диаграмма физической модели с отображением name, разработанной в среде Sybase PowerDesigner представлена на рисунке 1.



Рисунок 1 – Диаграмма физической модели с отображением name

Диаграмма физической модели с отображением code, разработанной в среде Sybase PowerDesigner представлена на рисунке 2.



Рисунок – Диаграмма физической модели с отображением code

**2 Спецификация таблиц**

Спецификация таблиц физической модели приведена в таблице 1.

Таблица 1 – Спецификация реляционных отношений физической модели

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Наименование таблицы** | **Код таблицы** | **Наименование атрибута** | **Код атрибута** | **Тип данных** | **Первичный ключ** | **Вторичный ключ** | **Обязательность** |
| Достижение | Achievement | id | id\_achivment | SERIAL | **+** | **-** | X |
| Номер студ. билета | id\_num\_student | INT4 | **-** | **+** | X |
| id2 | id\_category | INT4 | **-** | **+** | X |
| Название | name\_achivement | VARCHAR(200) | **-** | **-** | X |
| Дата получения | date\_achivment | DATE | **-** | **-** | X |
| Бюджет | Budget | Тип стипендии | type\_scholarship\_budget | VARCHAR(100) | **+** | **-** | X |
| Название семестра | name\_semester | VARCHAR(5) | **-** | **+** | X |
| Размер | size\_budget | MONEY | **-** | **-** | X |
| Категория | Category | id | id\_category | SERIAL | **+** | **-** | X |
| Тип достижения | achivments\_type\_category | VARCHAR(50) | **-** | **-** | X |
| Балл | score\_category | INT2 | **-** | **-** | X |
| Группа | Group | Название группы | name\_group | VARCHAR(10) | **+** | **-** | X |
| Направление подготовки | studies\_direction\_group | VARCHAR(8) | **-** | **-** | X |
| Профиль подготовки | studies\_profile\_group | VARCHAR(50) | **-** | **-** | X |
| Дата начала обучения | start\_date\_group | DATE | **-** | **-** | X |
| Срок обучения | studies\_period-group | INT2 | **-** | **-** | X |
| Оценка | mark | id | id\_mark | SERIAL | **+** | **-** | X |
| Номер студ. билета | id\_num\_student | INT4 | **-** | **+** | X |
| Название семестра | name\_semester | VARCHAR(5) | **-** | **+** | X |
| Название дисциплины | lesson\_name\_mark | VARCHAR(100) | **-** | **-** | X |
| Балл | score\_mark | INT2 | **-** | **-** | X |
| Тип | type\_mark | VARCHAR(40) | **-** | **-** | X |
| Стипендия | Scholarship | id | id\_scholarship | SERIAL | **+** | **-** | X |
| Номер студ. билета | id\_num\_student | INT4 | **-** | **+** | X |
| Тип стипендии | type\_scholarship\_budget | VARCHAR(100) | **-** | **+** | X |
| Название семестра | name\_semester | VARCHAR(5) | **-** | **-** | X |
| Размер | size\_scholarshp | MONEY | **-** | **-** | X |
| Тип | type\_scholarship | VARCHAR(100) | **-** | **-** | X |
| Семестр | Semester | Название семестра | name\_semester | VARCHAR(5) | **+** | **-** | X |
| Дата начала | date\_start\_semester | DATE | **-** | **-** | X |
| Дата окончания | date\_end\_semester | DATE | **-** | **-** | X |
| Студент | Student | Номер студ. билета | id\_num\_student | INT4 | **+** | **-** | X |
| Название группы | name\_group | VARCHAR(10) | **-** | **+** | X |
| Email | email\_student | VARCHAR(40) | **-** | **-** | X |
| Фамилия | second\_name\_student | VARCHAR(40) | **-** | **-** | X |
| Имя | first\_name\_student | VARCHAR(40) | **-** | **-** | X |
| Отчество | surname\_student | VARCHAR(40) | **-** | **-** |  |

Дополнительная спецификация индексов таблиц приведена в таблице 2

Таблица 2 – Спецификация индексов таблиц

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Наименование таблицы** | **Код таблицы** | **Наименование индекса** | **Unique** | **PK** | **FK** | **AL** |
| Достижение | Achivment | Achivement\_PK | + | + | - | - |
| Have\_FK | + | - | + | - |
| Refer\_FK | + | - | + | - |
| Категория | Category | Category\_PK | + | + | - | - |
| Студент | Student | Student\_PK | + | + | - | - |
| Study\_FK | + | - | + | - |
| Student\_AK | + | - | - | + |
| Группа | Group | Group\_PK | + | + | - | - |
| Оценка | Mark | mark\_PK | + | + | - | - |
| Get\_FK | + | - | + | - |
| give\_FK | + | - | + | - |
| average\_score\_IDX | - | - | - | - |
| Семестр | Semester | Semester\_PK | + | + | - | - |
| Стипендия | Scholarship | Scholarship\_PK | + | + | - | - |
| Receive\_FK | + | - | + | - |
| Calculate\_FK | + | - | + | - |
| assign\_FK | + | - | + | - |
| Бюджет | Budget | Budget\_PK | + | + | - | - |
| lay\_FK | + | - | + | - |

**3. Создание БД**

**3.1. SQL-скрипт создания схемы данных БД**

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

/\* DBMS name:      PostgreSQL 9.x                               \*/

/\* Created on:     06.11.2024 22:32:18                          \*/

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

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

/\* Table: Achievement                                           \*/

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

create table Achievement (

   id\_achivment         SERIAL               not null,

   id\_num\_student       INT4                 not null,

   id\_category          INT4                 not null,

   name\_achivement      VARCHAR(200)         not null,

   date\_achivment       DATE                 not null,

   constraint PK\_ACHIEVEMENT primary key (id\_achivment)

);

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

/\* Index: Achievement\_PK                                        \*/

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

create unique index Achievement\_PK on Achievement (

id\_achivment

);

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

/\* Index: Have\_FK                                               \*/

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

create  index Have\_FK on Achievement (

id\_num\_student

);

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

/\* Index: Refer\_FK                                              \*/

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

create  index Refer\_FK on Achievement (

id\_category

);

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

/\* Table: Budget                                                \*/

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

create table Budget (

   type\_scholarship\_budget VARCHAR(100)         not null,

   name\_semester        VARCHAR(5)           not null,

   size\_budget          MONEY                not null,

   constraint PK\_BUDGET primary key (type\_scholarship\_budget)

);

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

/\* Index: Budget\_PK                                             \*/

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

create unique index Budget\_PK on Budget (

type\_scholarship\_budget

);

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

/\* Index: lay\_FK                                                \*/

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

create  index lay\_FK on Budget (

name\_semester

);

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

/\* Table: Category                                              \*/

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

create table Category (

   id\_category          SERIAL               not null,

   achivments\_type\_category VARCHAR(50)          not null,

   score\_category       INT2                 not null,

   constraint PK\_CATEGORY primary key (id\_category)

);

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

/\* Index: Category\_PK                                           \*/

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

create unique index Category\_PK on Category (

id\_category

);

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

/\* Table: "Group"                                               \*/

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

create table "Group" (

   name\_group           VARCHAR(10)          not null,

   studies\_direction\_group VARCHAR(8)           not null,

   studies\_profile\_group VARCHAR(50)          not null,

   start\_date\_group     DATE                 not null,

   "studies\_period-group" INT2                 not null,

   constraint PK\_GROUP primary key (name\_group)

);

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

/\* Index: Group\_PK                                              \*/

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

create unique index Group\_PK on "Group" (

name\_group

);

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

/\* Table: Scholarship                                           \*/

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

create table Scholarship (

   id\_scholarship       SERIAL               not null,

   id\_num\_student       INT4                 not null,

   type\_scholarship\_budget VARCHAR(100)         not null,

   name\_semester        VARCHAR(5)           not null,

   size\_scholarshp      MONEY                not null,

   type\_scholarship     VARCHAR(100)         not null,

   constraint PK\_SCHOLARSHIP primary key (id\_scholarship)

);

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

/\* Index: Scholarship\_PK                                        \*/

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

create unique index Scholarship\_PK on Scholarship (

id\_scholarship

);

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

/\* Index: Receive\_FK                                            \*/

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

create  index Receive\_FK on Scholarship (

id\_num\_student

);

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

/\* Index: Calculate\_FK                                          \*/

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

create  index Calculate\_FK on Scholarship (

type\_scholarship\_budget

);

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

/\* Index: assign\_FK                                             \*/

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

create  index assign\_FK on Scholarship (

name\_semester

);

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

/\* Table: Semester                                              \*/

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

create table Semester (

   name\_semester        VARCHAR(5)           not null,

   date\_start\_semester  DATE                 not null,

   date\_end\_semester    DATE                 not null,

   constraint PK\_SEMESTER primary key (name\_semester)

);

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

/\* Index: Semester\_PK                                           \*/

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

create unique index Semester\_PK on Semester (

name\_semester

);

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

/\* Table: Student                                               \*/

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

create table Student (

   id\_num\_student       INT4                 not null,

   name\_group           VARCHAR(10)          not null,

   email\_student        VARCHAR(40)          not null,

   second\_name\_student  VARCHAR(40)          not null,

   first\_name\_student   VARCHAR(40)          not null,

   surname\_student      VARCHAR(40)          null,

   constraint PK\_STUDENT primary key (id\_num\_student),

   constraint AK\_EMAIL\_STUDENT\_STUDENT unique (email\_student)

);

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

/\* Index: Student\_PK                                            \*/

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

create unique index Student\_PK on Student (

id\_num\_student

);

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

/\* Index: Study\_FK                                              \*/

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

create  index Study\_FK on Student (

name\_group

);

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

/\* Table: mark                                                  \*/

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

create table mark (

   id\_mark              SERIAL               not null,

   id\_num\_student       INT4                 not null,

   name\_semester        VARCHAR(5)           not null,

   lesson\_name\_mark     VARCHAR(100)         not null,

   score\_mark           INT2                 not null,

   type\_mark            VARCHAR(40)          not null,

   constraint PK\_MARK primary key (id\_mark)

);

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

/\* Index: mark\_PK                                               \*/

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

create unique index mark\_PK on mark (

id\_mark

);

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

/\* Index: Get\_FK                                                \*/

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

create  index Get\_FK on mark (

id\_num\_student

);

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

/\* Index: give\_FK                                               \*/

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

create  index give\_FK on mark (

name\_semester

);

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

/\* Index: average\_score\_IDX                                     \*/

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

create  index average\_score\_IDX on mark (

name\_semester,

lesson\_name\_mark,

type\_mark,

score\_mark

);

alter table Achievement

   add constraint FK\_ACHIEVEM\_HAVE\_STUDENT foreign key (id\_num\_student)

      references Student (id\_num\_student)

      on delete cascade on update restrict;

alter table Achievement

   add constraint FK\_ACHIEVEM\_REFER\_CATEGORY foreign key (id\_category)

      references Category (id\_category)

      on delete restrict on update restrict;

alter table Budget

   add constraint FK\_BUDGET\_LAY\_SEMESTER foreign key (name\_semester)

      references Semester (name\_semester)

      on delete restrict on update restrict;

alter table Scholarship

   add constraint FK\_SCHOLARS\_CALCULATE\_BUDGET foreign key (type\_scholarship\_budget)

      references Budget (type\_scholarship\_budget)

      on delete restrict on update restrict;

alter table Scholarship

   add constraint FK\_SCHOLARS\_RECEIVE\_STUDENT foreign key (id\_num\_student)

      references Student (id\_num\_student)

      on delete cascade on update restrict;

alter table Scholarship

   add constraint FK\_SCHOLARS\_ASSIGN\_SEMESTER foreign key (name\_semester)

      references Semester (name\_semester)

      on delete restrict on update restrict;

alter table Student

   add constraint FK\_STUDENT\_STUDY\_GROUP foreign key (name\_group)

      references "Group" (name\_group)

      on delete restrict on update restrict;

alter table mark

   add constraint FK\_MARK\_GET\_STUDENT foreign key (id\_num\_student)

      references Student (id\_num\_student)

      on delete restrict on update restrict;

alter table mark

   add constraint FK\_MARK\_GIVE\_SEMESTER foreign key (name\_semester)

      references Semester (name\_semester)

      on delete restrict on update restrict;

**3.2. Скриншот консоли управления выбранной СУБД с представлением созданной структуры БД**

Скриншот консоли управления выбранной СУБД представлен на рисунке 3.

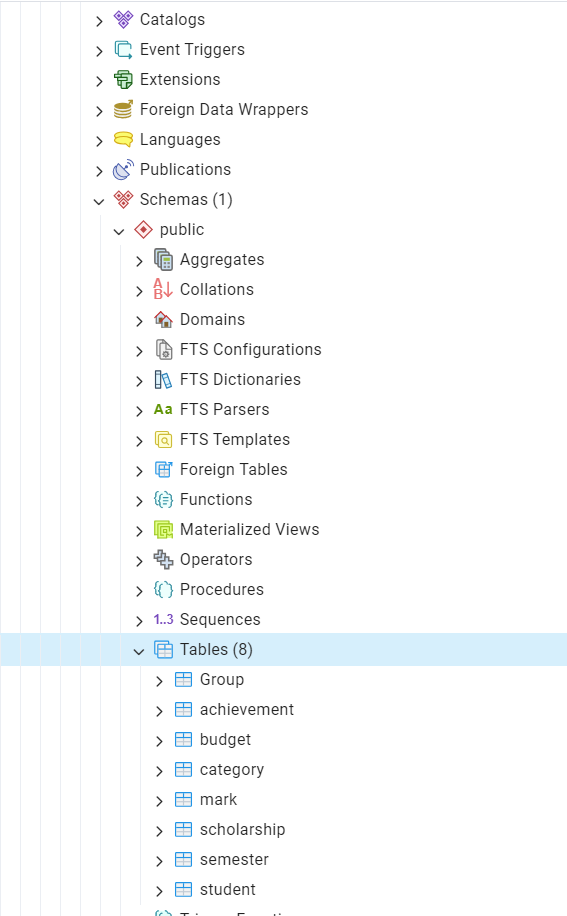


Рисунок – Структура БД

**4. Сценарий модификации БД**

**4.1. Описание сценария**

В рамках расширения функциональности базы данных в можно добавить таблицу "Дисконтные карта". Данная таблица будет хранить информацию и клиентских картах компьютерного клуба. Дисконтная карта будет обладать накопительным эффектом баллов, которыми можно расплачиваться в клубе. Также карта будет иметь особый уровень привилегий, например, "Стандартный" и "Премиум" уровни. В зависимости от уровня будет разный процент накопления бонусов.

Спецификация новой сущности представлена в таблице 2.

Таблица 2 – Спецификация сущности "Дисконтные карта"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Наименование таблицы** | **Код таблицы** | **Наименование атрибута** | **Код атрибута** | **Тип данных** | **Первичный ключ** | **Внешний ключ** | **Обязательность** |
| Льгота | benefit | id льготы | id\_benefit | SERIAK | + | - | Да |
| Название льготы | name\_benefit | VARCHAR(255) | - | - | Да |
| Описание льготы | description\_benefit | TEXT | - | - | Да |

Спецификация связей между сущностями представлена в таблице 3.

Таблица 3 – Спецификация связей для новой сущности

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Сущность 1** | **Сущность 2** | **Наименование связи** | **Код связи** | **Вид связи** | **Класс связи** | |
| **Обязательная для сущности 1** | **Обязательная для сущности 2** |
| Льгота | Стипендия | Дает | Get | 1:n | + | - |

**4.2. Текст SQL-запросов модификации схемы данных**

а) создание таблицы

create table benefit(

    id\_benefit SERIAL not null,

    name\_benefit VARCHAR(255) not null,

    description\_benefit TEXT,

    constraint PK\_BENEFIT PRIMARY KEY(id\_benefit)

);

б) добавление колонки в таблицу

alter table Scholarship

    add column id\_benefit INT4 null;

в) создание ограничение внешнего ключа для связи новой таблицы с существующими

alter table Scholarship

add constraint FK\_SCHOLARSHIP\_GET\_BENEFIT foreign key (id\_benefit)

references benefit(id\_benefit)

on delete restrict on update restrict;

г) модификация колонки в таблице

alter table Scholarship

    add column id\_benefit INT2 null;

д) удаление колонки в таблице

alter table Scholarship

    drop column id\_benefit;

е) удаление таблицы

drop table benefit;

**Вывод**

В ходе выполнения лабораторной работы изучил задачи физического проектирования базы данных. Освоил на практике разработку физической модели данных в среде Sybase PowerDesigner и модификацию схемы данных с использованием SQL-запросов.