МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)   
  
  
Факультет информатики  
Кафедра программных систем  
  
Дисциплина  
**Базы данных  
  
  
  
ОТЧЕТ**по лабораторной работе №2  
  
Вариант №14

Студент: Михайлова К.В  
Группа: 6314-020302D  
  
Преподаватель: Повова-Коварцева Д. А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2024

**Вариант № 14**

Предметная область: Бухгалтерия (расчет стипендии).

Основные предметно-значимые сущности: Студенты, Группы студентов, Результаты сдачи сессии.

Основные предметно-значимые атрибуты сущностей:

* студенты – фамилия, имя, отчество;
* группы студентов – название или номер группы;
* результаты сдачи сессии – студент, семестр, название категории (не сдал, сдал на 3, сдал на 4-5, сдал на 5).

Основные требования к функциям системы:

* назначить размер стипендии студентов за последнюю сессию в соответствие с действующими правилами;
* вывести группы, в которых студенты не получают стипендии;
* выбрать всех студентов, обучающихся по целевым договорам, сдавших на 3 или вообще не сдавших последнюю сессию, по группам;
* выбрать всех студентов, сдавших последнюю сессию на 5, по группам;
* подсчитать сумму стипендий студентов по курсам;
* вывести размер назначенной стипендии студентов, обучающихся по целевым договорам, по предприятиям;
* подсчитать сумму стипендий по группам.

Тема работы: логическое и физическое моделирование базы данных, реализация реляционной схемы БД в выбранной СУБД.

Цель работы: приобретение навыков создания логической и физической модели базы данных с помощью Case-средства. Приобретение навыков реализации базы данных в рамках реляционной модели с помощью СУБД.

Содержание работы:

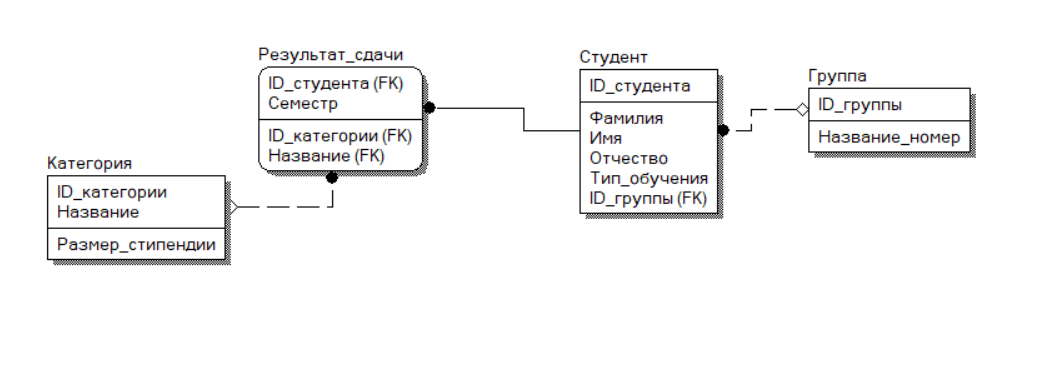
1. Изучение основ создания логической и физической модели данных с помощью Case-средства.
2. Изучение особенностей проектирования реляционных баз данных.
3. Подготовка ответов на вопросы к лабораторной работе.
4. Подготовка отчета о проделанной работе.

Последовательность выполнения работы:

1. С помощью Case-средства создайте логическую модель базы данных в соответствии с вариантом задания.
2. Укажите домены для всех атрибутов сущностей.
3. Для созданной модели опишите физический уровень представления данных, пользуясь выбранным в п.1 Case-средством.
4. Проанализируйте какие функциональные зависимости имеются в созданных отношениях.
5. Примените шаги по нормализации схем созданных отношений БД.
6. Выберите необходимую СУБД, для которой нужно создать схему данных, и сгенерируйте DDL-скрипт для выбранной СУБД.
7. Внесите необходимые изменения в имена таблиц, полей, связей и их свойства.
8. Внесите необходимые изменения в созданный DDL-скрипт.
9. Выполните DDL-скрипт и заполните созданные таблицы данными командой SQL Insert (не менее 12-15 записей на таблицу).
10. Оформите отчет о выполнении лабораторной работы.

1.

На рисунке 1 изображена логическая модель базы данных, созданная для базы данных «Бухгалтерия».

  
Рисунок 1 – Логическая модель базы данных

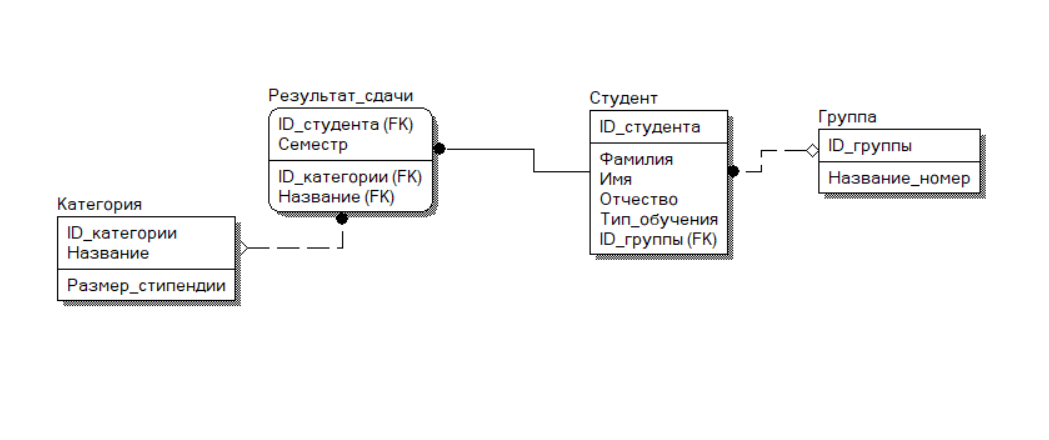
2.

ID\_студента, ID\_группы, ID\_категории, размер стипендии, семестр- int.

Фамилия, Имя, Отчество, Тип обучения, Название и номер, Название, Размер стипендии- varchar(40).

3.

На рисунке 2 изображена физическая модель базы данных, созданная для базы данных «Бухгалтерия».

  
Рисунок 2 – Физическая модель базы данных

4.

Все связи между сущностями являются «один ко многим», поэтому для нормализации отношений есть связующая сущность.

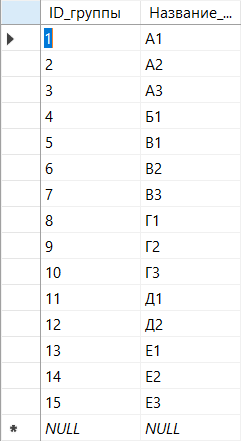
5.

На рисунке 3 мы видим пример заполнения таблицы базы данных через команду INSERT. По такому же принципу заполнены все остальные таблицы.

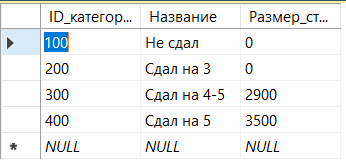
  
Рисунок 3 – Пример заполнения

6. Здесь приведены результаты заполнения таблиц базы данных.

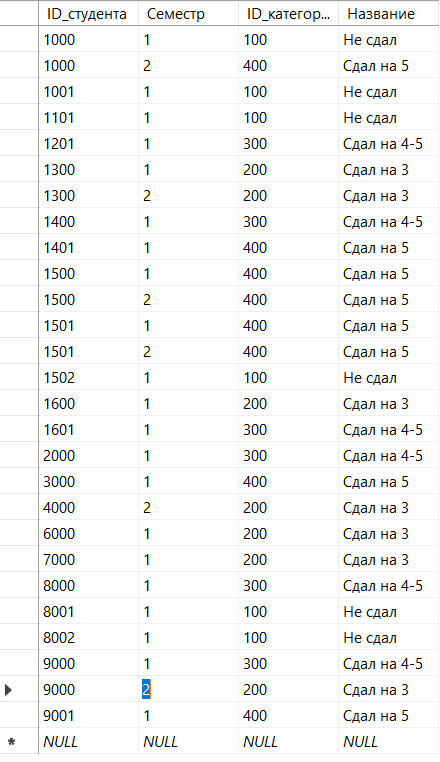
На рисунке 1 мы видим заполненную таблицу «Группа» из базы данных «Бухгалтерия». Были заполнены 2 столбца с помощью команды INSERT.

  
Рисунок 6.1 – Заполненная таблица «Группа»

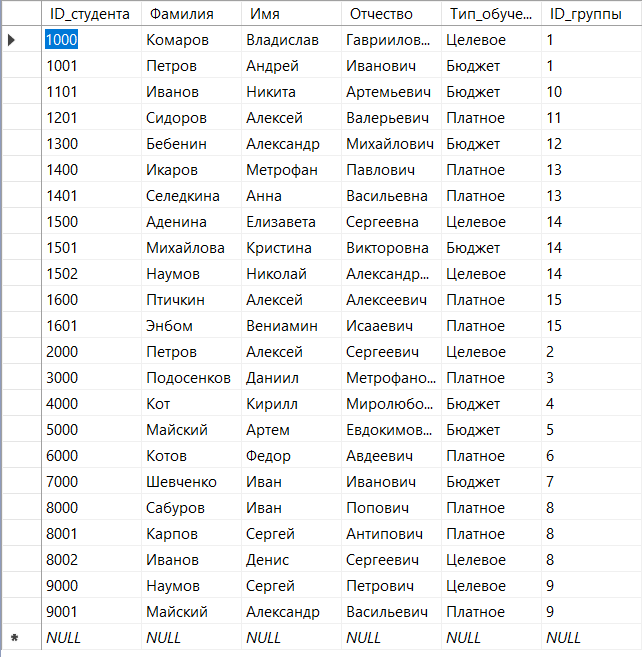
На рисунке 2 мы видим заполненную таблицу «Категория» из базы данных «Бухгалтерия». Были заполнены 3 столбца с помощью команды INSERT.

  
Рисунок 6.2 – Заполненная таблица «Категория»

На рисунке 3 мы видим заполненную таблицу «Результат сдачи» из базы данных «Бухгалтерия». Были заполнены 4 столбца с помощью команды INSERT.

  
Рисунок 6.3 – Заполненная таблица «Результат сдачи»

На рисунке 4 мы видим заполненную таблицу «Студент» из базы данных «Бухгалтерия». Были заполнены 6 столбцов с помощью команды INSERT.

  
Рисунок 6.4 – Заполненная таблица «Студент»

DDL код

CREATE TABLE Группа

(

ID\_группы integer NOT NULL ,

Название\_номер varchar(20) NULL

)

go

ALTER TABLE Группа

ADD CONSTRAINT XPKГруппа PRIMARY KEY CLUSTERED (ID\_группы ASC)

go

CREATE TABLE Категория

(

ID\_категории integer NOT NULL ,

Название varchar(20) NOT NULL ,

Размер\_стипендии integer NULL

)

go

ALTER TABLE Категория

ADD CONSTRAINT XPKКатегория PRIMARY KEY CLUSTERED (ID\_категории ASC,Название ASC)

go

CREATE TABLE Результат\_сдачи

(

ID\_студента varchar(20) NOT NULL ,

Семестр integer NOT NULL ,

ID\_категории integer NULL ,

Название varchar(20) NULL

)

go

ALTER TABLE Результат\_сдачи

ADD CONSTRAINT XPKРезультат\_сдачи PRIMARY KEY CLUSTERED (ID\_студента ASC,Семестр ASC)

go

CREATE TABLE Студент

(

ID\_студента varchar(20) NOT NULL ,

Фамилия varchar(20) NULL ,

Имя varchar(20) NULL ,

Отчество varchar(20) NULL ,

Тип\_обучения varchar(20) NULL ,

ID\_группы integer NULL

)

go

ALTER TABLE Студент

ADD CONSTRAINT XPKСтудент PRIMARY KEY CLUSTERED (ID\_студента ASC)

go

ALTER TABLE Результат\_сдачи

ADD CONSTRAINT R\_2 FOREIGN KEY (ID\_студента) REFERENCES Студент(ID\_студента)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Результат\_сдачи

ADD CONSTRAINT R\_5 FOREIGN KEY (ID\_категории,Название) REFERENCES Категория(ID\_категории,Название)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Студент

ADD CONSTRAINT R\_1 FOREIGN KEY (ID\_группы) REFERENCES Группа(ID\_группы)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go