Database Homework 3

Matt McCullough

DDL commands used to create the tables

CREATE TABLE student (

idStudent INT NOT NULL,

Name VARCHAR(45) NOT NULL,

GPA DECIMAL(3,2),

Address VARCHAR(45),

PRIMARY KEY(idStudent)

);

CREATE TABLE course (

idCourse INT NOT NULL,

title VARCHAR(45) NOT NULL,

level VARCHAR(45) NOT NULL,

PRIMARY KEY(idCourse)

);

CREATE table phoneNumber (

Phone\_number VARCHAR(10) NOT NULL,

Student\_idstudent INT NOT NULL,

PRIMARY KEY(phone\_number),

FOREIGN KEY (student\_idstudent) REFERENCES student(idStudent)

);

CREATE TABLE student\_has\_course (

student\_idstudent INT NOT NULL,

course\_idcourse INT NOT NULL,

PRIMARY KEY (student\_idstudent, course\_idcourse),

FOREIGN KEY (student\_idstudent) REFERENCES student (idstudent),

FOREIGN KEY (course\_idcourse) REFERENCES course (idcourse)

);

DML commands used to insert data into the tables

INSERT INTO student VALUES (1,'Luke Skywalker', 3.4, 'Space');

INSERT INTO student VALUES (2, 'Mario Mario', 2.4, 'Brooklyn');

INSERT INTO student VALUES (3, 'Mark Johnson', 3.2, '1723 Hawk Way');

INSERT INTO student VALUES (4,'Jack Black', 3.0, 'Valley of Peace');

INSERT INTO student VALUES (5,'Spongebob Squarepants', 2.0, '124 Conch Street');

INSERT INTO course VALUES (6,'csc351','undergraduate');

INSERT INTO course VALUES (7,'MAT325','graduate');

INSERT INTO course VALUES (8,'csc600','graduate');

INSERT INTO course VALUES (9,'PHY115','undergraduate');

INSERT INTO course VALUES (10,'HIS202','graduate');

INSERT INTO phoneNumber VALUES ('1234567890',1);

INSERT INTO phoneNumber VALUES ('2345678901',1);

INSERT INTO phoneNumber VALUES ('3456789012',3);

INSERT INTO phoneNumber VALUES ('4567890123',4);

INSERT INTO phoneNumber VALUES ('5678901234',5);

INSERT INTO student\_has\_course VALUES (1,9);

INSERT INTO student\_has\_course VALUES (2,7);

INSERT INTO student\_has\_course VALUES (5,10);

INSERT INTO student\_has\_course VALUES (4,6);

INSERT INTO student\_has\_course VALUES (2,8);

DML queries

1. Display all student names and GPA’s

**SELECT name,GPA from student;**

1. How many students are there?

**SELECT count(name) as number\_of\_students from student;**

1. How many undergraduate courses are there?

**SELECT count(title) from course**

**Where level = 'Undergraduate';**

1. How many students have a GPA greater than 3.5

**SELECT count(name) from student**

**Where GPA > 3.5;**

1. Display the GPA of the student whose name is “Mark Johnson”

**SELECT GPA from student**

**WHERE name = 'Mark Johnson';**

1. Display all Computer Science courses.

**SELECT title from course**

**WHERE title LIKE ‘csc%’;**

1. What is the average grade of all students?

**SELECT AVG(GPA) from student;**

1. What courses Mark Johnson is taking? Include course title and level as well

**SELECT title, level from course**

**inner join student\_has\_course**

**on student\_has\_course.course\_idcourse = course.idcourse**

**inner join student**

**on student\_has\_course.student\_idstudent = student.idstudent**

**Where name = 'Mark Johnson';**

1. Display all student’s information including their phone numbers

**Select idstudent, name, GPA, address, phoneNumber.phone\_number**

**From student**

**Left join phoneNumber**

**On phoneNumber.student\_idstudent = student.idStudent;**

10.Who is taking csc351? Include name, GPA, and student I.D

**SELECT name, Gpa, idStudent from student**

**inner join student\_has\_course**

**on student\_has\_course.student\_idstudent = student.idstudent**

**inner join course**

**on student\_has\_course.course\_idcourse = course.idcourse**

**WHERE title = 'csc351';**

11.Count How many courses for each level. In addition, add courses to the

counter/number of courses

**SELECT level, COUNT(level) as number\_of\_courses**

**from course**

**GROUP BY level;**

12. Count how many phone numbers each student has

**SELECT student.name, Count(phone\_number) from phoneNumber**

**Inner join student**

**On phoneNumber.student\_idstudent = student.idStudent**

**group by student.name;**

13. Display the name of students who have more than 1 phone number

**SELECT name from student**

**Inner join phoneNumber**

**On phoneNumber.student\_idstudent = student.idStudent**

**Having Count(student\_idstudent) > 1;**

14. The name of students and their I.Ds. who have not taken any courses

**Select name, idStudent from student**

**left join student\_has\_course**

**on student\_has\_course.student\_idstudent = student.idstudent**

**where student\_idstudent is null;**

Screenshots

1.

Graphical user interface, application

Description automatically generated

2.

Graphical user interface, application

Description automatically generated

3.

Graphical user interface, application

Description automatically generated

4.

Graphical user interface, text, application

Description automatically generated

5.

Graphical user interface, application

Description automatically generated

6.

Graphical user interface, application

Description automatically generated

7.

Graphical user interface, application

Description automatically generated

8.

A screenshot of a computer

Description automatically generated with low confidence

9.

Graphical user interface

Description automatically generated with medium confidence

10.

A screenshot of a computer

Description automatically generated with medium confidence

11.

Graphical user interface, application

Description automatically generated

12.

Graphical user interface, application

Description automatically generated

13.

Graphical user interface, application

Description automatically generated

14.

Graphical user interface, application

Description automatically generated