Database Homework 4

Matt McCullough

1. Design an Entity Relationship Diagram for the following Schema (use Workbench):  
   ***student****(student\_id, name, score)****phone\_number****(number, type, student\_id)*A student can have 0 or multiple phone numbers. Each phone number must be assigned to one student only **or none.**

Diagram

Description automatically generated

create table student (  
 student\_id int not null,  
 name VARCHAR(45) not null,  
 score INT NOT NULL,  
 PRIMARY KEY (student\_id)  
);  
  
create table phone\_number (  
  
 number VARCHAR(10) not null,  
 type VARCHAR(45) not null,  
 student\_id INT,  
 primary key (number),  
 FOREIGN KEY (student\_id) REFERENCES student(student\_id)  
);  
  
  
insert into student values (9891, 'Mark', 97);  
insert into student values (9877, 'John', 34);  
insert into student values (9856, 'Michael', 85);

insert into phone\_number VALUES ('1234567890', 'Verizon', NULL);  
insert into phone\_number VALUES ('1234567891', 'AT&T', 9856);  
insert into phone\_number VALUES ('1234567892', 'T Mobile', 9856);  
insert into phone\_number VALUES ('1234567893', 'Sprint', 9891);

2a.

Retrieve all student I.Ds., names, scores, and assign a grade to each score. Use the grading criteria that is included on this course’s site

select \*,  
CASE  
 WHEN score >= 96 THEN 'A'  
 WHEN score >= 90 THEN 'A-'  
 WHEN score >= 85 THEN 'B+'  
 WHEN score >= 80 THEN 'B'  
 WHEN score >= 78 THEN 'B-'  
 WHEN score >= 75 THEN 'C+'  
 WHEN score >= 71 THEN 'C'  
 WHEN score >= 65 THEN 'C-'  
 WHEN score >= 61 THEN 'D'  
 ELSE 'F'  
END as Grade  
from student  
ORDER BY Grade;

Graphical user interface, application

Description automatically generated

2b.

Retrieve all student IDs, names, score, and a message that says whether the student’s score is above or below average as follows:

select \*,  
CASE  
 WHEN score > (SELECT AVG(score) FROM student) THEN 'Above Average'  
 WHEN score = (SELECT AVG(score) FROM student) THEN 'Average'  
 ELSE 'Below Average'  
END as Result  
from student  
order by Result;

A screenshot of a computer

Description automatically generated with medium confidence

2c.

Repeat query a such that, it retrieves the same thing, but if the student has achieved the highest or lowest grade, add that as a message next to the score:

select student\_id,name,  
CASE  
 WHEN score = (SELECT MAX(score) FROM student) THEN CONCAT(score, ' highest')  
 WHEN score = (SELECT MIN(score) FROM student) THEN CONCAT(score, ' lowest')  
 ELSE score  
END as score,  
CASE  
 WHEN score >= 96 THEN 'A'  
 WHEN score >= 90 THEN 'A-'  
 WHEN score >= 85 THEN 'B+'  
 WHEN score >= 80 THEN 'B'  
 WHEN score >= 78 THEN 'B-'  
 WHEN score >= 75 THEN 'C+'  
 WHEN score >= 71 THEN 'C'  
 WHEN score >= 65 THEN 'C-'  
 WHEN score >= 61 THEN 'D'  
 ELSE 'F'  
END as Grade  
from student;

A screenshot of a computer

Description automatically generated with medium confidence

2d.

Retrieve all available phone numbers (available phone number must not be assigned to a student | Null)

SELECT number from phone\_number  
WHERE student\_id IS NULL;

Graphical user interface, text, application

Description automatically generated

2e.

Retrieve all students’ details including their phone numbers. Display student with assigned number only (do not retrieve students who don’t have numbers or numbers without owners)

select student.student\_id,name,score,number  
from student  
inner join phone\_number  
on student.student\_id = phone\_number.student\_id  
WHERE number is not null;

A screenshot of a computer

Description automatically generated with medium confidence

2f.

Retrieve all students’ details, and all phone numbers stored in the database. If a phone number is assigned to a student, print all student’s details including their phone numbers, and if a phone number has no owner or student has no number, records should be displayed as well with NULL values.

select student.student\_id,name,score,number  
from student  
left join phone\_number  
on student.student\_id = phone\_number.student\_id;

A screenshot of a computer

Description automatically generated with medium confidence

2g.

Retrieve all phone numbers including their owners. If a number has no owner, still display it as well, but do not display students’ details who have no phone number assigned to them.

select student.student\_id,name,score,number  
from student  
right join phone\_number  
on student.student\_id = phone\_number.student\_id;

A screenshot of a computer

Description automatically generated with medium confidence