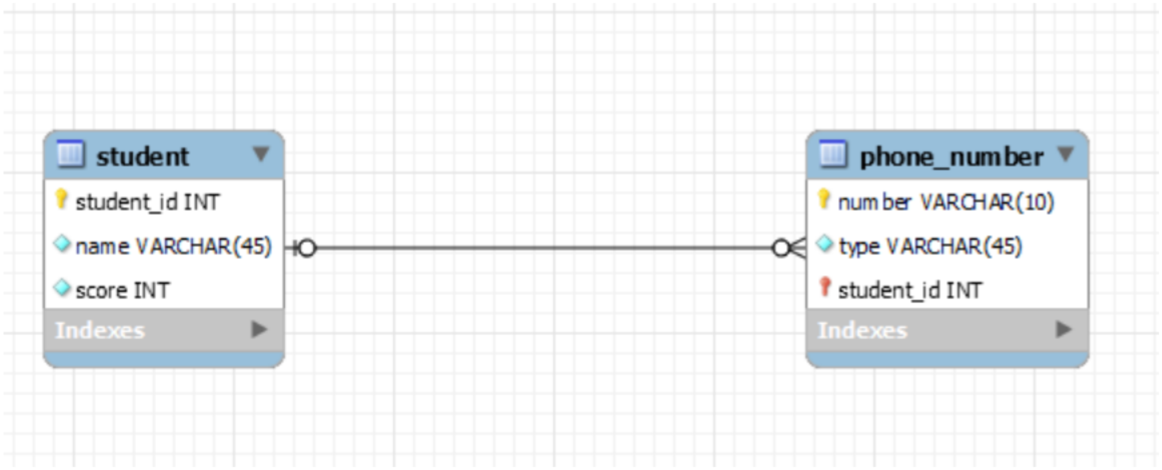
Database Homework 4

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.**

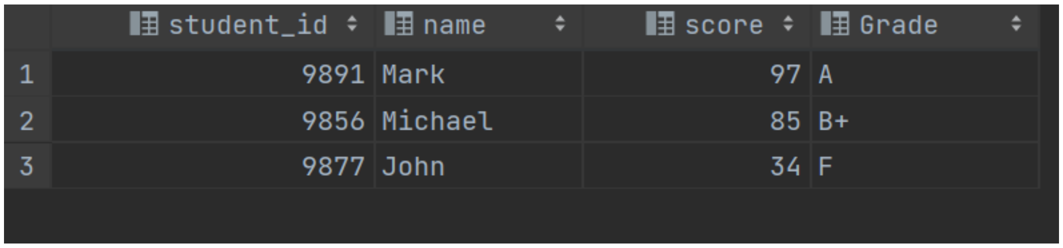


```
create table student (
   student_id int not null,
   name VARCHAR(45) not null,
   score INT NOT NULL,
   PRIMARY KEY (student_id)
);

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;
```

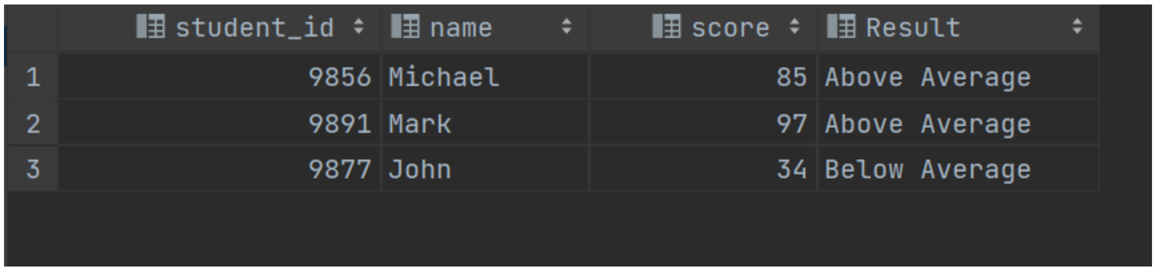


	student_id	name	score	Grade
1	9891	Mark	97	A
2	9856	Michael	85	B+
3	9877	John	34	F

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;
```

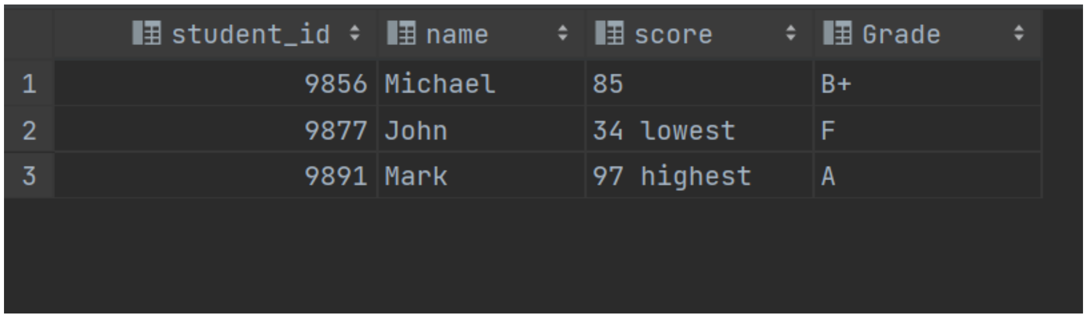


	student_id	name	score	Result
1	9856	Michael	85	Above Average
2	9891	Mark	97	Above Average
3	9877	John	34	Below Average

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;
```

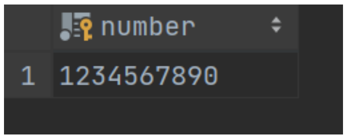


	student_id	name	score	Grade
1	9856	Michael	85	B+
2	9877	John	34 lowest	F
3	9891	Mark	97 highest	A

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;

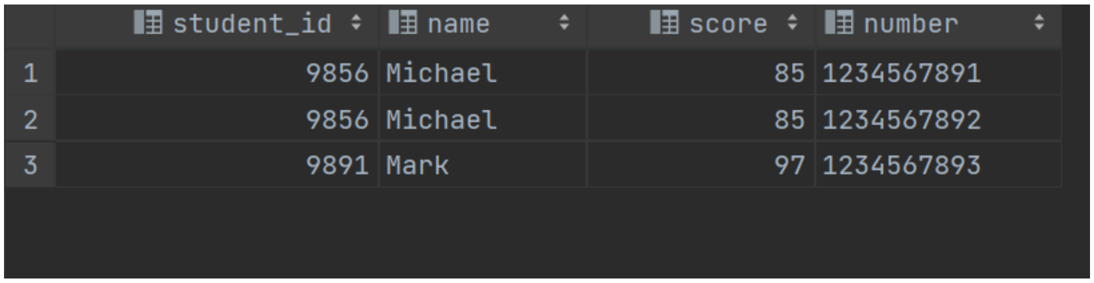


	number
1	1234567890

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;

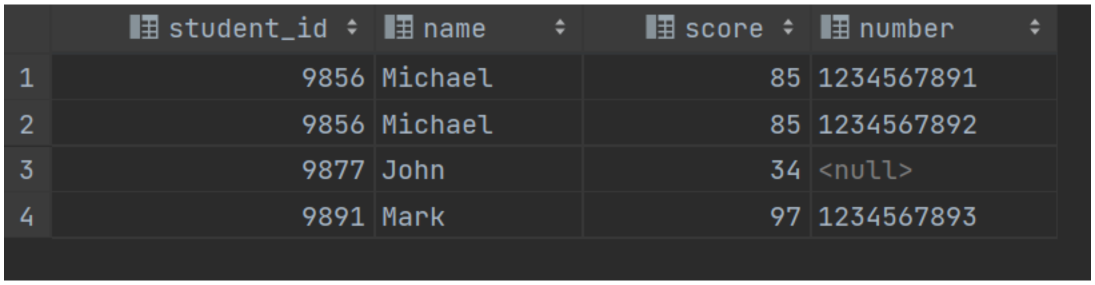


	student_id	name	score	number
1	9856	Michael	85	1234567891
2	9856	Michael	85	1234567892
3	9891	Mark	97	1234567893

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;

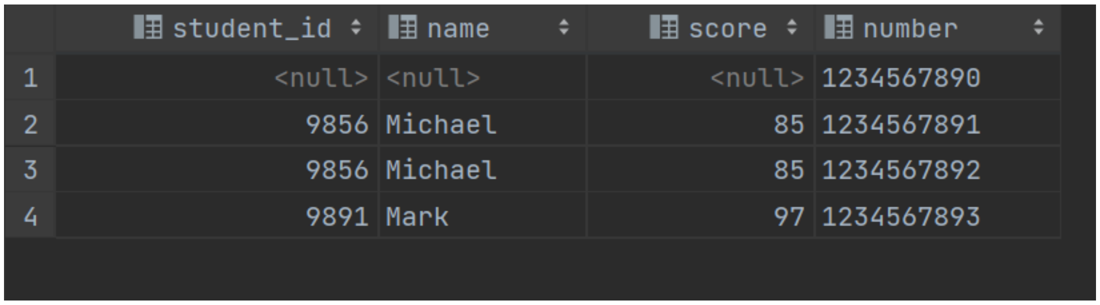


	student_id	name	score	number
1	9856	Michael	85	1234567891
2	9856	Michael	85	1234567892
3	9877	John	34	<null>
4	9891	Mark	97	1234567893

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;
```



	student_id	name	score	number
1	<null>	<null>	<null>	1234567890
2	9856	Michael	85	1234567891
3	9856	Michael	85	1234567892
4	9891	Mark	97	1234567893