```
create table student(id int, name varchar(20), age int, city varchar(20));

insert into student values(1,'abi',18,'chennai'), (2, 'banu', 18, 'bangalore'), (3,'karthi', 19, 'chennai'),(4,'mano', 18, 'kerala'), (5, 'hari', 19, 'bangalore');

create table marks(id int, s1 int, s2 int, s3 int, s4 int, s5 int, avg float(3,2));

insert into marks(id,s1,s2,s3,s4,s5)values(1,95,90,94,92,90), (3,95,92,95,85,93), (5,85,99,95,96,95);
```

```
select * from student;
```

id	name	age	city
1	abi	18	chennai
2	banu	18	bangalore
3	karthi	19	chennai
4	mano	18	kerala
5	hari	19	bangalore

```
select * from marks;
```

id	s1	s2	s3	s4	s5	avg
1	95	90	94	92	90	NULL
3	95	92	95	85	93	NULL
5	85	99	95	96	95	NULL

SELECT QUERIES ON SINGLE RELATION

- SELECT -FROM-WHERE
- IN, NOT IN
- ORDER BY
- BETWEEN
- DISTINCT
- ALL

Find the details of all the students

```
select * from student;
```

id	name	age	city
1	abi	18	chennai
2	banu	18	bangalore
3	karthi	19	chennai
4	mano	18	kerala
5	hari	19	bangalore

Find the id and name of all the students

```
select id, name from student;
```

id	name
1	abi
2	banu
3	karthi
4	mano
5	hari

Find the details of all the students whose city is chennai

```
select * from student where city='chennai';
```

id	name	age	city
1	abi	18	chennai
3	karthi	19	chennai

Find the names of all the students whose city is bangalore

```
select name from student where city='bangalore';
```

name
banu
hari

Find all the details of the student whose id is 4

```
select  * from student where id=4;
```

id	name	age	city
4	mano	18	kerala

Find the details of the students who lives in chennai and with age greater then 18

```
select * from student where city='chennai' and age>18;
```

id	**name**	**age**	**city**
3	karthi	19	chennai

Find the names of the students who lives in either chennai or bangalore

```
select name from student where city='chennai' or city='bangalore';
(OR)
select name from student where city in('chennai', 'bangalore');
```

name
abi
banu
karthi
hari

Find the names of the students who neither belongs to chennai nor bangalore

```
select name from student where city<>'chennai' and city<>'bangalore';
(OR)
select name from student where city not in('chennai', 'bangalore');
```

name
mano

Increase the age of all the students by 1 and display the details

```
select id, name,city,age+1 from student;
```

id	name	city	age+1
1	abi	chennai	19
2	banu	bangalore	19
3	karthi	chennai	20
4	mano	kerala	19
5	hari	bangalore	20

```
select id, name, age+1 as newage from student;
```

id	name	newage
1	abi	19
2	banu	19
3	karthi	20
4	mano	19
5	hari	20

Find the details for the students whose id is between 2 and 4

```
select * from student where id between 2 and 4;
(OR)
select * from student where id>=2 and id<=4;
```

id	name	age	city
2	banu	18	bangalore
3	karthi	19	chennai
4	mano	18	kerala

Display the details of the students in the ascending order of name

```
select * from student order by name;
```

id	name	age	city
1	abi	18	chennai
2	banu	18	bangalore
5	hari	19	bangalore
3	karthi	19	chennai
4	mano	18	kerala

Display the details of the students in the descending order of name

```
select * from student order by name desc;
```

Id	name	age	city
4	mano	18	kerala
3	karthi	19	chennai
5	hari	19	bangalore
2	banu	18	bangalore
1	abi	18	chennai

Display the details of the students in the ascending order of age. If more than one student has the same age, display the details in descending order of name

```
select * from student order by age asc, name desc;
```

id	name	age	city
4	mano	18	kerala
2	banu	18	bangalore
1	abi	18	chennai
3	karthi	19	chennai
5	hari	19	bangalore

Display all the cities in the student table

```
select city from student;
(OR)
select all city from student;
```

city
chennai
bangalore
chennai
kerala
bangalore

```
select distinct city from student;
```

city
chennai
bangalore
kerala

SQL QUERIES ON MULTIPLE TABLES (Cartesian Product, Joins)

- **Cartesian Product, Natural Join, Inner Join, Left Outer Join, Right Outer Join, Full Outer Join**

Cartesian Product Operation:

```
select * from student,marks;
```

id	name	age	city	id	s1	s2	s3	s4	s5	avg
1	abi	18	chennai	5	85	99	95	96	95	NULL
1	abi	18	chennai	3	95	92	95	85	93	NULL
1	abi	18	chennai	1	95	90	94	92	90	NULL
2	banu	18	bangalore	5	85	99	95	96	95	NULL
2	banu	18	bangalore	3	95	92	95	85	93	NULL
2	banu	18	bangalore	1	95	90	94	92	90	NULL
3	karthi	19	chennai	5	85	99	95	96	95	NULL
3	karthi	19	chennai	3	95	92	95	85	93	NULL
3	karthi	19	chennai	1	95	90	94	92	90	NULL
4	mano	18	kerala	5	85	99	95	96	95	NULL
4	mano	18	kerala	3	95	92	95	85	93	NULL
4	mano	18	kerala	1	95	90	94	92	90	NULL
5	hari	19	bangalore	5	85	99	95	96	95	NULL
5	hari	19	bangalore	3	95	92	95	85	93	NULL
5	hari	19	bangalore	1	95	90	94	92	90	NULL

Natural Join Operation:

```
select * from student natural join marks;
```

id	name	age	city	s1	s2	s3	s4	s5	avg
1	abi	18	chennai	95	90	94	92	90	NULL
3	karthi	19	chennai	95	92	95	85	93	NULL
5	hari	19	bangalore	85	99	95	96	95	NULL

Inner Join Operation:

```
select * from student inner join marks on student.id=marks.id;
```

id	name	age	city	id	s1	s2	s3	s4	s5	avg

1	abi	18	chennai	1	95	90	94	92	90	NULL
3	karthi	19	chennai	3	95	92	95	85	93	NULL
5	hari	19	bangalore	5	85	99	95	96	95	NULL

Find the id, name and marks of the students.

Using Cartesian Product:

```
select student.id,name,s1,s2,s3,s4,s5 from student, marks where student.id=marks.id;
```

Using Natural Join

```
select id, name, s1,s2,s3,s4,s5 from student natural join marks;
```

Using Inner Join

```
select student.id, name, s1,s2,s3,s4,s5 from student inner join marks on student.id=marks.id;
```

id	name	s1	s2	s3	s4	s5
1	abi	95	90	94	92	90
3	karthi	95	92	95	85	93
5	hari	85	99	95	96	95

Find the id, name and marks of the students whose city is bangalore.

Using Cartesian Product:

```
select student.id, name, s1,s2,s3,s4,s5 from student, marks where student.id=marks.id and
city='bangalore';
```

Using Natural Join

```
select id, name, s1,s2,s3,s4,s5 from student natural join marks where city='bangalore';
```

Using Inner Join

```
select student.id, name, s1,s2,s3,s4,s5 from student inner join marks on student.id=marks.id
and city='bangalore';
```

id	name	s1	s2	s3	s4	s5
5	hari	85	99	95	96	95

Left Outer Join Operation:

```
select * from student left outer join marks on student.id=marks.id;
```

Id	name	age	city	id	s1	s2	s3	s4	s5	avg
1	abi	18	chennai	1	95	90	94	92	90	NULL
2	banu	18	bangalore	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	karthi	19	chennai	3	95	92	95	85	93	NULL
4	mano	18	kerala	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	hari	19	bangalore	5	85	99	95	96	95	NULL

Right Outer Join Operation:

```
select * from student right outer join marks on student.id=marks.id;
```

id	name	age	city	id	s1	s2	s3	s4	s5	avg
1	abi	18	chennai	1	95	90	94	92	90	NULL
3	karthi	19	chennai	3	95	92	95	85	93	NULL
5	hari	19	bangalore	5	85	99	95	96	95	NULL

STRING PATTERN MATCHING (like)

Find the details of the students whose name ends with ‘i’.

```
select * from student where name like '%i';
```

id	name	age	city
1	abi	18	chennai
3	karthi	19	chennai
5	hari	19	bangalore

Find the details of the students whose name starts with ‘a’

```
select * from student where name like 'a%';
```

id	name	age	city
1	abi	18	chennai

Find the details of the students whose name starts with ‘a’ and ends with ‘i’.

```
select * from student where name like 'a%i';
```

id	name	age	city
1	abi	18	chennai

Find the details of the students whose name contains second character as ‘a’

```
select * from student where name like '_a%';
```

id	name	age	city
2	banu	18	bangalore
3	karthi	19	chennai
4	mano	18	kerala
5	hari	19	bangalore

Find the details of the students whose name contains string ‘ar’

```
select * from student where name like '%ar%';
```

id	name	age	city

3	karthi	19	chennai
5	hari	19	bangalore

SET OPERATIONS

- Union (removes duplicates)
- Union all (allows duplicates)
- Intersect (not supported in mysql. Query to be written using **'in'** keyword)
- Except (not supported in mysql. Query to be written using **'not in'** keyword)

Union

Find the id of students from student table and marks table.

```
(select id from student) union (select id from marks);
```

```
id
1
2
3
4
5
```

```
(select id from student) union all(select id from marks);
```

```
id
1
2
3
4
5
1
3
5
```

Intersect

Find the details of the students whose details are commonly present in student table and marks table.

```
select * from student where id in (select id from marks);
```

Id	name	age	city
1	abi	18	chennai
3	karthi	19	chennai
5	hari	19	bangalore

Except

Find the details of the students whose details are present in 'student' table but not in 'marks' table.

```
select * from student where id not in (select id from marks);
```

```
id          name        age         city
2           banu        18          bangalore
4           mano        18          kerala
```

AGGREGATION AND GROUPING

- max
- min
- avg
- sum
- count
- group by
- having

Find the maximum mark in the subject 's1'

```
select max(s1) from marks;
```

```
max(s1)
95
```

Find the sum of all marks in the subject 's1'

```
select sum(s1) from marks;
```

```
sum(s1)
275
```

Find the average mark in the subject 's1'

```
select avg(s1) from marks;
```

```
avg(s1)
91.6667
```

Find the minimum marks in the subject 's4'

```
select min(s4) from marks;
```

```
min(s4)
85
```

Find the number of records in marks table.

```
select count(id) from marks;
```

```
count(id)
3
```

Find the number of records in student table.

```
select count(id) as number_of_students from student;
```

```
number_of_students
5
```

Find the number of students belong to each city

```
select city,count(id) as countofstudents from student group by city;
```

```
city        countofstudents
chennai     2
bangalore   2
kerala      1
```

Find the city and average marks of s2 in each city where the average marks of the students belong to each city is more than 8.00

```
select city,avg(s2) as s2_avg from student natural join marks group by city having s2_avg>8.00;
```

```
city        s2_avg
chennai     91.0000
bangalore   99.0000
```

REFER TO **W3SCHOOLS, SQLZOO, CODECADEMY** etc., FOR MORE ON SQL.

PRACTICE WELL. ALL THE BEST!!