```
================================================================
  MySQL-SQL- Correlated Sub-Query(using the EXISTS operator)
================================================================
----------------------------------------------------
   rarely used and always asked in interview
----------------------------------------------------

* This is the EXCEPTION when Sub-Query is Faster than Join.
* EXITS is a Special OPERATOR

Problems:
-----------------------------------------------
 Q.Display the Dname that contains Employee
-----------------------------------------------

SOlution:
Step 1: Select deptno from emp;
+--------+
| deptno |
+--------+
|      1 |
|      1 |
|      1 |
|      2 |
|      2 |
+--------+

Step 2: Use distinct to remove Duplicate
mysql> Select distinct deptno from emp;
+--------+
| deptno |
+--------+
|      1 |
|      2 |
+--------+
Step 3: want see deptname on which employee present...not dept name
Select dname from dept where deptno=(step 2: deptno(1,2) will error if select use any
operator)
Select dname from dept where deptno=(Select distinct deptno from emp);//error


Using Any Operator
mysql> Select dname from dept where deptno= any(Select distinct deptno from emp);
+-------+
| dname |
+-------+
| TRN   |
| EXP   |
+-------+
Using IN Operator
mysql> Select dname from dept where deptno in (Select distinct deptno from emp);
+-------+
| dname |
+-------+
| TRN   |
| EXP   |
+-------+

-----------------------------------------------------
 Q.Display the Dname that not contains  Employee
-----------------------------------------------------
Using Any Operator
mysql> Select dname from dept where deptno!= any(Select distinct deptno from emp);

Using IN Operator
mysql> Select dname from dept where deptno not in (Select distinct deptno from emp);
+-------+
| dname |
+-------+
| MKTG  |
```

```
+-------+

--------------------------------------------------------------------------------

Solution no 2:
above solution has two problems
1. more select statment
2. Distinct using searching makes system slower

Step 1: join operation
select dname from emp,dept
where dept.deptno=emp.deptno;
+-------+
| dname |
+-------+
| TRN   |
| TRN   |
| TRN   |
| EXP   |
| EXP   |
+-------+
Step2: distinct
select distinct dname from emp,dept
where dept.deptno=emp.deptno;
+-------+
| dname |
+-------+
| TRN   |
| EXP   |
+-------+
------------------------------------------------------------
* if you have  a join along with Distinct to make it work faster,
  use correalated sub-Query(use the exists operatro)

Solution no 3:
select dname from dept where exists
(select deptno from emp
where dept.deptno=emp.deptno);

+-------+
| dname |
+-------+
| TRN   |
| EXP   |
+-------+

explaination:

* first the main query is executed
* for every row returned by main query, it will run the sub-query once
* the sub-query returns a boolean TRUE or FALSE value back to main query if sub-query
returns a
  TRUE value, then main query is eventually executed for that row
* if sub-query returns a FALSE value, then main query is not executed for that row
* unlike earlier, we do not use DISTINCT here; this speeds it up unlike a traditional join,
  the number of full table scans is reduced; this further speeds it up
------------------------------------------------------------

select dname from dept where not exists
(select deptno from emp
where dept.deptno=emp.deptno);
+-------+
| dname |
+-------+
| MKTG  |
+-------+
--------------------------------------------------------------------------------
================================================================
SQL set Operatroes based on set theory
================================================================
create table emp1(empno1 int(4),ename varchar(20));
```

```
insert into emp1 values(1,'A');
insert into emp1 values(2,'B');
insert into emp1 values(3,'c');
select * from emp1;
+--------+-------+
| empno1 | ename |
+--------+-------+
|      1 | A     |
|      2 | B     |
|      3 | c     |
+--------+-------+

create table emp2(empno2 int(4),ename varchar(20));
insert into emp2 values(1,'A');
insert into emp2 values(2,'B');
insert into emp2 values(4,'D');
insert into emp2 values(5,'E');
select * from emp2;
+--------+-------+
| empno2 | ename |
+--------+-------+
|      1 | A     |
|      2 | B     |
|      4 | D     |
|      5 | E     |
+--------+-------+
```

UNION

UNION:
will combine the output of both the select statement and it will suppress the duplicate

```
mysql> select empno,ename from emp1;
+-------+-------+
|empno1 | ename |
+-------+-------+
|     1 | A     |
|     2 | B     |
|     3 | c     |
+-------+-------+
3 rows in set (0.03 sec)

mysql> select empno2,ename from emp2;
+-------+-------+
|empno2 | ename |
+-------+-------+
|     1 | A     |
|     2 | B     |
|     4 | D     |
|     5 | E     |
+-------+-------+
```

```
select empno1,ename from emp1
union
select empno2,ename from emp2;
+--------+-------+
| empno1 | ename |
+--------+-------+
|      1 | A     |
|      2 | B     |
|      3 | c     |
|      4 | D     |
|      5 | E     |
+--------+-------+
```

Union ALL

Union all->will combine the **output of both** the **Select statement and** THE duplicates **are not** suppressed

```
select empno1,ename from emp1
union all
select empno2,ename from emp2;
```

```
+--------+-------+
| empno1 | ename |
+--------+-------+
|      1 | A     |
|      2 | B     |
|      3 | c     |
|      1 | A     |
|      2 | B     |
|      4 | D     |
|      5 | E     |
+--------+-------+
```

Intersect

Intersect->will **return** what **is** common **in both** the **select statement and** the duplicates **are** suppressed.

```
select empno1,ename from emp1
intersect
select empno2,ename from emp2
order by 1;
```

Minus

Minus->will **return** what **is** present **in** the **first select statement and** **not** present **in** the **second select statement and** the duplicates **are** suppressed.

```
select empno1,ename from emp1
Minus
select empno2,ename from emp2
order by 1;
```

Order by x

MAx upto 255 **select statement** (this **limit of SQL** cab be exceeded **using** views)

```
mysql> select distinct job from emp where deptno=1;
+------+
| job  |
+------+
| M    |
| C    |
+------+
```

==========
SQL Pseudo columns
==========
* fake columns(virtual columns)
* computed colums(e.g. ANNUAL=sal*12)
* expressions(e.g. net Earning=Sal+Commission)
* Funcction based colums(e.g. Total=Sum(sal)))

RDBMS supplied Pseudo columns:

```
select ename,sal from emp;
mysql> select ename,sal from emp;
+--------+------+
| ename  | sal  |
+--------+------+
| arun   | 8000 |
| ali    | 7000 |
| kiran  | 3000 |
| jack   | 9000 |
| Thomas | 8000 |
+--------+------+
```

Rowid

---

* **row** indentifier
* rowis **is** the **row** address
* **rowid is** the actual physcial memory location **in** the DB server Hd **WHERE** that **row is** stored
* **rowid is** fixed **length** excrypted string **of** 18 characters
* **when** you **select from** a **table**,the **order of rows in** the **output** depends **on** the **row** adress**(**it will always be **in** ascending **order of rowid)(**searching **IS** sequential**)**
* **When** you **UPDATE** a **row**, **if** the **row length is** increasing, **then** the **Rowid** MAY change
* YOU CAN **USE ROWID TO UPDATE OR DELETE** THE DUPLICATE **ROWS**

```
select rowid,ename,sal from emp;
select rowid, ename, sal from emp where rowid = 'AAASSMAABAAACG5AAA';
delete from emp where rowid = 'AAASSMAABAAACG5AAA' ;
```

Rowid is used internally **by** MySQL: **-**

1. **To** distinguish **between** 2 **rows in** the database
2. **For row** locking
3. **To** manage the INDEXES
4. **To** manage the cursors
5. ro management
6. etc

* **In** oracle,feature **of** crowid **is** available **and** you can **view** it
* **In** MySQL,feature **of rowid is** available **and** you cannot **view** it

---

Alter Table(DDL command**)**

```
create table emp(empno int(4),ename varchar(20),sal int(10));
insert into emp values(1,'Scott',3000);
insert into emp values(2,'King',5000);
mysql> select * from emp;
+-------+-------+------+
| empno | ename | sal  |
+-------+-------+------+
|     1 | Scott | 3000 |
|     2 | King  | 5000 |
+-------+-------+------+
```

* **rename** a **TABLE**
* **add** a COLUMNS
* **drop** a COLUMNS
* increse width **of column**

Indirectly:

* reduce width **of column**
* chanhe datatype **of column**

* copy rows from one table to another table
* copy a table
* copy only structure of table
* rename a column
* change position of columns in table structure
(because of null values, for storage considerations)

---
rename a columns
---
* rename is DDL command(auto commit)

```
rename table emp to employees;
mysql> select * from employees;
+-------+-------+------+
| empno | ename | sal  |
+-------+-------+------+
|     1 | Scott | 3000 |
|     2 | King  | 5000 |
+-------+-------+------+
```

---
add a columns .... Alter table

```
alter table emp add
alter table emp add gst float;

mysql> alter table emp add gst float;
mysql> select * from emp;
+-------+-------+------+------+
| empno | ename | sal  | gst  |
+-------+-------+------+------+
|     1 | Scott | 3000 | NULL |
|     2 | King  | 5000 | NULL |
+-------+-------+------+------+
```

---
drop a columns .... Alter table

```
alter table emp drop
alter table emp drop gst;

mysql> alter table emp add gst;
mysql> select * from emp;
+-------+-------+------+
| empno | ename | sal  |
+-------+-------+------+
|     1 | Scott | 3000 |
|     2 | King  | 5000 |
+-------+-------+------+
```

Increse wodth of column

```
alter table emp modify ename varchar(30);
mysql> desc emp;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| empno | int         | YES  |     | NULL    |       |
| ename | varchar(20) | YES  |     | NULL    |       |
| sal   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.06 sec)
```

```
mysql> alter table emp modify ename varchar(30);
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc emp;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| empno | int         | YES  |     | NULL    |       |
| ename | varchar(30) | YES  |     | NULL    |       |
| sal   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

Reduce the length of table

```
alter table emp drop
alter table emp drop gst;
alter table emp modify ename varchar (20);<- data will get truncated


In Oracle: -
alter table emp modify ename varchar (20) ;<-ERROR in Oracle


* you can reduce the width provided the contents are null



update emp set ename = null;
alter table emp modify ename varchar (20);


mysql> alter table emp modify ename varchar (20);
mysql> desc emp;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| empno | int         | YES  |     | NULL    |       |
| ename | varchar(20) | YES  |     | NULL    |       |
| sal   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+



In Oracle: -

alter table emp modify ename varchar (20);< ERROR in Oracle
you can reduce the width provided the contents are null

alter table emp add x varchar (25) ;
update emp set x = ename, ename = null;
alter table emp modify ename varchar (20) ;

/* DATA TESTING ON X COLUMN, CHECK THE names <= 20 characters */
update emp set ename = x;
alter table emp drop column x;

*ABOVE SOLUTION WILL WORK IN MySQL ALSO AND SHOULD BE IMPLEMENTED IN MySQL ALSO
```

---

copy a table (for testing purposes)

```
create table emp_copy
as
select * from emp;
```

```
mysql> select * from emp_copy;
+-------+-------+------+
| empno | ename | sal  |
+-------+-------+------+
|     1 | Scott | 3000 |
|     2 | King  | 5000 |
+-------+-------+------+
drop table emp_copy;


CREATE table emp(empno int(4),ename varchar(20),sal int(10),deptno int(4),job
VARCHAR(20),mgr VARCHAR(20));
insert into emp values(1,'arun',8000,1,'M','4');
insert into emp values(2,'ali',7000,1,'C','1');
insert into emp values(3,'kiran',3000,1,'C','1');
insert into emp values(4,'jack',9000,2,'M',null);
insert into emp values(5,'Thomas',8000,2,'C',4);




-----------------------------------------------------
to copy certain rows only:
-----------------------------------------------------
create table emp_copy as select * from emp where deptno = 10 ;

mysql> create table emp_copy as select * from emp where deptno = 2 ;
mysql> select * from emp_copy;
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+


-----------------------------------------------------
to copy certain columns only: -
-----------------------------------------------------
create table emp_copy
as
select empno, ename from emp;
mysql> select * from emp_copy;
+-------+--------+
| empno | ename  |
+-------+--------+
|     1 | arun   |
|     2 | ali    |
|     3 | kiran  |
|     4 | jack   |
|     5 | Thomas |
+-------+--------+
```