

```
create a new database (mysql)
create table department
(
code int primary key auto_increment,
name char(20) not null unique
)Engine=InnoDB // to validate fk constraint
```

```
create table designation
(
code int primary key auto_increment,
name char(20) not null unique
)Engine=InnoDB
create table employee
(
code int primary key auto_increment,
name char(20),
department_code int references department,
designation_code int references designation
)Engine=InnoDB
```

```
insert into department (name) values('Sales');
insert into department (name) values('Stores');
insert into department (name) values('Purchase');
insert into department (name) values('House Keeping');
```

```
insert into designation (name) values('Manager')
insert into designation (name) values('Clerk')
insert into designation (name) values('Inspector')
insert into designation (name) values('Cashier')
insert into designation (name) values('Officer')
```

create a java application that will feed 16000 dummy records in Employee, some records should contain department_code as null and some records should contain designation_code as null.

the records which contains department_code should contain as 1,2,3 or 4 and the records which contain designation code should contain as 1,2,3,4 or 5

code to feed dummy records

```
import java.sql.*;
class eg1
{
public static void main(String gg[])
{
try
```

```
{
Class.forName("com.mysql.jdbc.Driver");
Connection c;
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb","testdb","kelkar");
PreparedStatement ps;
int departmentCode=1;
int designationCode=1;
int x;
x=1;
while(x<=16000)
{

if(departmentCode<5 && designationCode<6)
{
ps=c.prepareStatement("insert into employee (name,department_code,designation_code)
values(?,?,?)");
ps.setString(1,"Employee - "+x);
ps.setInt(2,departmentCode);
ps.setInt(3,designationCode);
ps.executeUpdate();
ps.close();
}
if(departmentCode==5 && designationCode!=6)
{
ps=c.prepareStatement("insert into employee (name,designation_code) values(?,?)");
ps.setString(1,"Employee - "+x);
ps.setInt(2,designationCode);
ps.executeUpdate();
ps.close();

}
if(designationCode==6 && departmentCode!=5)
{
ps=c.prepareStatement("insert into employee (name,department_code) values(?,?)");
ps.setString(1,"Employee - "+x);
ps.setInt(2,departmentCode);
ps.executeUpdate();
ps.close();
}
if(departmentCode==5 && designationCode==6)
{
ps=c.prepareStatement("insert into employee (name) values(?)");
ps.setString(1,"Employee - "+x);
ps.executeUpdate();
ps.close();
}
departmentCode++;
}
```

```

designationCode++;
if(departmentCode==6)
{
departmentCode=1;
}
if(designationCode==7)
{
designationCode=1;
}
x++;
}
c.close();
} catch(Exception e)
{
System.out.println(e);
}
}
}
}

```

code to get time diff in milliseconds

```

class eg1
{
public static void main(String gg[])
{
long t1=System.nanoTime();
try
{
Thread.sleep(1000);
} catch(Exception e)
{
}
long t2=System.nanoTime();
double d=(t2-t1)/1e6;
System.out.println(d+" millisecs");
}
}
}

```

select * from employee,department,designation

write first 10 records

go through all records

count the number of records

note down the number of records in output, in employee, department and designation table

or to count you can use the following sql statement

```
select count(*) from employee,department,designation.
```

Now write a code in java which should fire the following statement and note down the time taken.
Don't print the records

```
Select * from employee,department,designation
```

```
select * from employee,department,designation where employee.department_code=department.code  
and employee.designation_code=designation.code
```

do the above in mysql query browser and note down the data and number of records, or you can fire the
sql statement with count function to count records

then do the same from java and note down the time taken

```
select employee.code,employee.name,department.code,department.name from employee  
inner join department on employee.department_code=department.code
```

do whatever we did earlier

for the following sql statement also, do whatever we did earlier

```
select  
employee.code,employee.name,department.code,department.name,designation.code,designation.name  
from employee  
inner join department on employee.department_code=department.code  
inner join designation on employee.designation_code=designation.code
```

in the above statement wherever we have written
inner join

replace it with
left join

and then

replace it with

right join

```
create table Country  
(  
code int primary key auto_increment,  
name char(20)  
)
```

```
create table State
```

```
(  
code int primary key auto_increment,  
name char(20),  
country_code int references country,  
)
```

```
create table City  
(  
code int primary key auto_increment,  
name char(20),  
state_code int references state  
)
```

```
create table department  
(  
code int primary key auto_increment,  
name char(20)  
)
```

```
create table employee  
(  
code int primary key auto_increment,  
name char(20),  
department_code int references department,  
city_code int references city  
)
```

```
create table Country  
(  
code int primary key auto_increment,  
name char(20)  
)
```

```
create table State  
(  
code int primary key auto_increment,  
name char(20),  
country_code int references country,  
)
```

```
create table City  
(  
code int primary key auto_increment,  
name char(20),  
state_code int references state  
)
```

```
create table department
(
code int primary key auto_increment,
name char(20)
)
```

```
create table employee
(
code int primary key auto_increment,
name char(20),
department_code int references department,
city_code int references city,
salary int
)
```

```
-----
select code from city where name='Ujjain'
```

```
select * from employee where city_code=3
```

Sub Query

```
select * from employee where city_code=(select code from city where name='Ujjain')
```

```
-----
select code from state where name='M.P.'
output is (17)
select code from city where state_code=17
output is (1 5 4 3 8)
```

```
select * from employee where city_code in (1,5,4,3,8)
```

sub query

```
select * From employee where city_code
in (select code from city where state_code=(
select code from state where name='M.P.'
))
```

```
-----
select employee.code,employee.name from employee,city,state where
employee.city_code=city.code and
city.state_code=state.code
and state.name='M.P'
```

```
-----
select count(*) from employee
```

assuming that the following are state table records

- 1 MP 3
- 2 Maharashtra 3
- 3 Karachi 4
- 4 Lahore 4
- 5 Rawalpindi 4

select count(*) from state

output will be 5

select name,count(*) from state

The above SQL does not make any sense

select state_code,count(*) from state group by state_code

select name,state_code,count(*) from state group by state_code

the above sql statement is senseless

select state_code,count(*) from state group by
state_code having state_code in (3,5)

select count(department) from employee

those employees who don't belong to any department won't
be considered in the counting process

select max(salary) from employee

Co-related sub query

Employee

c n	d_c	salary
1	Sameer 1	1000
2	Rakesh 1	2000
3	Suresh 2	3000
4	Mahesh 1	1500
5	Rohit 2	2500

Output required

Department wise max salary distributed

```
select department_code,max(salary) from  
employee group by department_code
```

output required

Department wise max salary distributed and to whom

```
select name,department_code,max(salary) from  
employee group by department_code
```

the above sql statement is senseless

solution : co-related sub query

```
select name,department_code,salary from employee e1 where salary=  
(select max(salary) from employee e2 where e2.department_code=e1.department_code)
```
