1. From the EMP table show the minimum, maximum and average basic for each department.

Now, Suppose, Minimum salary of each department is to be shown:

**SELECT MIN(BASIC) FROM EMP GROUP BY DEPT\_CODE;**

But, this will display something like:

|  |
| --- |
| **MIN(BASIC)** |
| 2100000.00 |
| 1600000.00 |
| 1200000.00 |
| 1000000.00 |
| 0.00 |

If you want to see DEPT\_NAME as well, you need to make a join query.

**Select MIN(EMP.BASIC), DEPARTMENT.DEPT\_NAME from EMP,DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP. DEPT\_CODE;**

Or,

A little better:

**select MIN(EMP.BASIC) AS MIN\_BASIC, DEPARTMENT.DEPT\_NAME from EMP,DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP. DEPT\_CODE;**

**Or,**

**select MIN(EMP.BASIC) AS MIN\_BASIC, DEPARTMENT.DEPT\_NAME from EMP INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE;**

|  |  |
| --- | --- |
| **MIN\_BASIC** | **DEPT\_NAME** |
| 2100000.00 | Research |
| 1600000.00 | Finance |
| 1200000.00 | Purchase |
| 1000000.00 | Production |
| 0.00 | Personnel |

Using similar concepts, maximum and average basic for each department can be found.

2. Find the number of female employees in each department.

select COUNT(DISTINCT EMP\_CODE) FROM EMP WHERE SEX='F' GROUP BY DEPT\_CODE;

This will print something like following:

|  |
| --- |
| COUNT(DISTINCT EMP\_CODE) |
| 2 |
| 2 |

TO make this better again join query is needed to be used.

**select COUNT(DISTINCT EMP.EMP\_CODE) AS NO\_OF\_FEMALE\_EMPLOYEE, DEPARTMENT.DEPT\_NAME FROM EMP,DEPARTMENT WHERE SEX='F' AND EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE;**

Actually, this is not good. EMP\_CODE is primary key.

Or,

**SELECT COUNT(\*) AS NO\_OF\_FEMALE\_EMPLOYEE, DEPARTMENT.DEPT\_NAME FROM EMP INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE WHERE EMP.SEX='F' GROUP BY EMP.DEPT\_CODE;**

|  |  |
| --- | --- |
| **NO\_OF\_FEMALE\_EMPLOYEE** | **DEPT\_NAME** |
| 2 | Finance |
| 2 | Personnel |

3. SELECT COUNT(\*) AS NO\_OF\_EMPLOYEE from EMP WHERE CITY=(SELECT DISTINCT CITY FROM EMP) GROUP BY DEPT\_CODE;

Now, this will print the following:

|  |
| --- |
| **NO\_OF\_EMPLOYEE** |
| 3 |
| 3 |
| 3 |
| 3 |
| 4 |

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEE, DEPARTMENT.DEPT\_NAME from EMP, DEPARTMENT WHERE CITY=(SELECT DISTINCT CITY FROM EMP) AND EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE;

Now, this will print the following:

|  |  |
| --- | --- |
| **NO\_OF\_EMPLOYEE** | **DEPT\_NAME** |
| 3 | Research |
| 3 | Finance |
| 3 | Purchase |
| 3 | Production |
| 4 | Personnel |

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEE, DEPARTMENT.DEPT\_NAME FROM EMP INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE WHERE EMP.CITY=(SELECT DISTINCT CITY FROM EMP)GROUP BY EMP.DEPT\_CODE;

However, since, every employee’s city in the example dataset is New Delhi, Let’s perform these queries on ADDRESS and check whether the queries are actually right or wrong.

Now, SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES FROM EMP WHERE ADDRESS=(SELECT DISTINCT ADDRESS FROM EMP) GROUP BY DEPT\_CODE;

This won’t work.

Telling that subquery returns more than one row.

Whereas, SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES FROM EMP WHERE CITY=(SELECT DISTINCT CITY FROM EMP) GROUP BY DEPT\_CODE;

Will work.

To make it work with address, use IN CLAUSE:

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES FROM EMP WHERE ADDRESS IN (SELECT DISTINCT ADDRESS FROM EMP) GROUP BY DEPT\_CODE;

But, this generates wrong result.

So, my query would actually fail if city contains distinct value.

**The correct query is:**

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES,ADDRESS FROM EMP WHERE ADDRESS IN (SELECT DISTINCT ADDRESS FROM EMP) GROUP BY ADDRESS,DEPT\_CODE;

**(Additionally Address is printed to be sure)**

This will give the following result:

|  |  |
| --- | --- |
| **NO\_OF\_EMPLOYEES** | **ADDRESS** |
| 3 | C.R. Park, Delhi |
| 1 | Kalkaji, Delhi |
| 3 | Noida, Delhi |
| 2 | Okhla, Delhi |
| 3 | Okhla, Delhi |
| $ | Okhla, Delhi |

**To make the query result more clear:**

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES,EMP.ADDRESS,DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE AND EMP.ADDRESS IN (SELECT DISTINCT ADDRESS FROM EMP) GROUP BY EMP.ADDRESS,EMP.DEPT\_CODE;

**Or,**

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES,EMP.ADDRESS,DEPARTMENT.DEPT\_NAME FROM EMP INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE WHERE EMP.ADDRESS IN (SELECT DISTINCT ADDRESS FROM EMP) GROUP BY EMP.ADDRESS,EMP.DEPT\_CODE;

**These will print the following:**

|  |  |  |
| --- | --- | --- |
| **NO\_OF\_EMPLOYEES** | **ADDRESS** | **DEPT\_NAME** |
| 3 | C.R. Park, Delhi | Research |
| 1 | Kalkaji, Delhi | Purchase |
| 3 | Noida, Delhi | Finance |
| 2 | Okhla, Delhi | Purchase |
| 3 | Okhla, Delhi | Production |
| 4 | Okhla, Delhi | Personnel |

**It could be further improved by:**

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES,EMP.ADDRESS,DEPARTMENT.DEPT\_NAME FROM EMP INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE WHERE EMP.ADDRESS IN (SELECT DISTINCT ADDRESS FROM EMP) GROUP BY EMP.ADDRESS,EMP.DEPT\_CODE ORDER BY DEPT\_NAME;

**4.** Show the designation wise no of employees who have joined in the year 2000 in each department. The listing should app ear in the ascending order of no. of employees.

**Now, in my sample database, no people actually joined in 2000, so, let’s try 2014.**

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES, DESIGNATION.DESIG\_DESC DESIGNATION, DEPARTMENT.DEPT\_NAME DEPARTMENT\_NAME FROM EMP,DESIGNATION, DEPARTMENT WHERE EMP.DESIG\_CODE=DESIGNATION.DESIG\_CODE AND EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE AND JN\_DATE LIKE "2014%" GROUP BY EMP.DESIG\_CODE, EMP.DEPT\_CODE ORDER BY EMP.DEPT\_CODE,NO\_OF\_EMPLOYEES ASC;

|  |  |  |
| --- | --- | --- |
| **NO\_OF\_EMPLOYEES** | **DESIGNATION** | **DEPARTMENT\_NAME** |
| 1 | Manager | Finance |
| 1 | Officer | Finance |
| 1 | Executive | Purchase |
| 1 | Manager | Purchase |
| 1 | Officer | Purchase |
| 1 | Manager | Production |
| 1 | Officer | Production |
| 1 | Executive | Production |
| 1 | Officer | Personnel |
| 3 | Clerk | Personnel |

Now, here, a warning is generated telling wrong date value. However, it does not affect the query result.

Now, this could be more organised if the query is changed to the following:

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES, DESIGNATION.DESIG\_DESC DESIGNATION, DEPARTMENT.DEPT\_NAME DEPARTMENT\_NAME FROM EMP,DESIGNATION, DEPARTMENT WHERE EMP.DESIG\_CODE=DESIGNATION.DESIG\_CODE AND EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE AND JN\_DATE LIKE "2014%" GROUP BY EMP.DESIG\_CODE, EMP.DEPT\_CODE ORDER BY EMP.DEPT\_CODE ASC, EMP.DESIG\_CODE ASC, NO\_OF\_EMPLOYEES ASC;

Though the original question does not need up to this extent.

5. Find the department wise total basic of male employees only for the departments for which such total is more than 50, 000 and the listing should appear in the descending order of total basic.

Now, the following would do the task:

**SELECT SUM(BASIC) AS SUMMATION\_OF\_BASIC FROM EMP WHERE SEX='M' GROUP BY DEPT\_CODE HAVING SUMMATION\_OF\_BASIC>50000 ORDER BY SUMMATION\_OF\_BASIC DESC;**

Let’s improvise this. One improvisation could be print DEPT\_NAME as an info.

For that, one of the followings could be done:

SELECT SUM(EMP.BASIC) AS SUMMATION\_OF\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE SEX='M' AND EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING SUMMATION\_OF\_BASIC>50000;

Or,

SELECT SUM(EMP.BASIC) AS SUMMATION\_OF\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE WHERE SEX='M' GROUP BY EMP.DEPT\_CODE HAVING SUMMATION\_OF\_BASIC>50000;

6. Show the employee name , Designation description and basic for all employees.

select EMP.EMP\_NAME, DESIGNATION.DESIG\_DESC, EMP.BASIC FROM EMP, DESIGNATION WHERE EMP.DESIG\_CODE=DESIGNATION.DESIG\_CODE;

Or,

SELECT EMP.EMP\_NAME, DESIGNATION.DESIG\_DESC, EMP.BASIC FROM EMP INNER JOIN DESIGNATION ON EMP.DESIG\_CODE=DESIGNATION.DESIG\_CODE;

7. Show the employee name, Designation description, Department Name & Basic for all employees.

SELECT EMP.EMP\_NAME, DESIGNATION.DESIG\_DESC, DEPARTMENT.DEPT\_NAME, EMP.BASIC FROM EMP, DESIGNATION, DEPARTMENT WHERE EMP.DESIG\_CODE=DESIGNATION.DESIG\_CODE AND EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE;

Or,

SELECT EMP.EMP\_NAME, DESIGNATION.DESIG\_DESC, DEPARTMENT.DEPT\_NAME, EMP.BASIC FROM((EMP INNER JOIN DESIGNATION ON EMP.DESIG\_CODE=DESIGNATION.DESIG\_CODE) INNER JOIN DEPARTMENT ON EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE);

1. Find the department Codes in which no employee works.

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES FROM EMP GROUP BY DEPT\_CODE HAVING NO\_OF\_EMPLOYEES=0;

This will work.

However, we could improve the query:

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE DEPARTMENT.DEPT\_CODE=EMP.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING NO\_OF\_EMPLOYEES=0;

1. Find the department names where at least one employee works.

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE DEPARTMENT.DEPT\_CODE=EMP.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING NO\_OF\_EMPLOYEES>=1;

This is an improved version printing department information, too.

1. Find the department names where at least 10 employees work.

SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE DEPARTMENT.DEPT\_CODE=EMP.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING NO\_OF\_EMPLOYEES>=10;

1. Find the department code in which employee with highest Basic works.  
   SELECT DEPT\_CODE FROM EMP WHERE BASIC>= ALL(SELECT BASIC FROM EMP) ;
2. SELECT COUNT(\*) AS NO\_OF\_MANAGERS, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE DEPARTMENT.DEPT\_CODE=EMP.DEPT\_CODE AND EMP.DESIG\_CODE="00002" GROUP BY EMP.DEPT\_CODE;

It could be written using INNER\_JOIN, too.

1. Find the maximum basic from EMP table without using MAX()  
     
   SELECT BASIC FROM EMP WHERE BASIC>= ALL(SELECT BASIC FROM EMP);
2. Find the minimum basic from EMP table without using MIN()  
     
   SELECT BASIC FROM EMP WHERE BASIC<=ALL(SELECT BASIC FROM EMP);

16. Find the name of the department with highest total basic. Do the same for highest average basic and maximum no. of employee.

Now, the result for every query will be RESEARCH.

For Highest TOTAL\_BASIC I Initially thought of

SELECT SUM(EMP.BASIC) AS TOTAL\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP,DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING TOTAL\_BASIC>=ALL(SELECT SUM(BASIC) FROM EMP GROUP BY DEPT\_CODE);

But, this will generate wrong answer.

SELECT SUM(EMP.BASIC) AS TOTAL\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP,DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE;

This outer query is generating:

|  |  |
| --- | --- |
| **TOTAL\_BASIC** | **DEPT\_NAME** |
| 7500000.00 | Research |
| 6100000.00 | Finance |
| 4100000.00 | Purchase |
| 3300000.00 | Production |
| 1600000.00 | Personnel |

But, the total query is generating:

|  |  |
| --- | --- |
| **TOTAL\_BASIC** | **DEPT\_NAME** |
| 12000000.00 | Research |
| 9600000.00 | Finance |

SELECT SUM(EMP.BASIC) AS TOTAL\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP,DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE ORDER BY TOTAL\_BASIC DESC LIMIT 1;

This can do the job.

SELECT MAX(TOTAL\_BASIC) as MAX\_TOTAL\_BASIC, DEPT\_CODE FROM(SELECT SUM(BASIC)AS TOTAL\_BASIC, DEPT\_CODE FROM EMP GROUP BY DEPT\_CODE) EMP;

This works too.

SELECT MAX(TOTAL\_BASIC) as MAX\_TOTAL\_BASIC, DEPARTMENT.DEPT\_NAME FROM(SELECT SUM(EMP.BASIC)AS TOTAL\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE) EMP,DEPARTMENT;

This is a slightly modified version.

**Now, however,these queries are wrong.**

**If TOTAL\_BASIC OF Various Departments look like following:**

|  |  |
| --- | --- |
| **TOTAL\_BASIC** | **DEPT\_NAME** |
| 7500000.00 | Research |
| 6100000.00 | Finance |
| 4100000.00 | Purchase |
| 3300000.00 | Production |
| 1600000.00 | Personnel |

**It works.**

**But, if TOTAL\_BASIC of various departments looks like following:**

|  |  |
| --- | --- |
| **TOTAL\_BASIC** | **DEPT\_NAME** |
| 7500000.00 | Research |
| 7500000.00 | Finance |
| 5400000.00 | Purchase |
| 3000000.00 | Production |
| 3400000.00 | Personnel |

**The query fails.**

SELECT SUM(BASIC) AS MAX\_TOTAL\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING MAX\_TOTAL\_BASIC=(SELECT SUM(BASIC) AS TOTAL\_BASIC FROM EMP GROUP BY DEPT\_CODE ORDER BY TOTAL\_BASIC DESC LIMIT 1);

**For, highest average basic:**

SELECT AVG(BASIC) AS MAX\_AVG\_BASIC, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING MAX\_AVG\_BASIC=(SELECT AVG(BASIC) AS AVG\_BASIC FROM EMP GROUP BY DEPT\_CODE ORDER BY AVG\_BASIC DESC LIMIT 1);

**For maximum number of employees:**

SELECT COUNT(\*) AS MAX\_NO\_OF\_EMPLOYEES, DEPARTMENT.DEPT\_NAME FROM EMP, DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE GROUP BY EMP.DEPT\_CODE HAVING MAX\_NO\_OF\_EMPLOYEES=(SELECT COUNT(\*) AS NO\_OF\_EMPLOYEES FROM EMP GROUP BY DEPT\_CODE ORDER BY NO\_OF\_EMPLOYEES DESC LIMIT 1);

1. Insert same rows into EMP table with designation code not existing in DESIGNATION table.

INSERT INTO EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN) VALUES("00019", "Vidya Bagchi", "00005", "00006", 'F', "Gurgaon, Delhi", "New Delhi","New Delhi","122001");

INSERT INTO EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN) VALUES("00020", "Abhirup Bagchi", "00001", "00000", 'M', "Gurgaon, Delhi", "New Delhi","New Delhi","122001");

18. Delete the rows from EMP table with invalid DESIG\_CODE.

SELECT \* FROM EMP WHERE DESIG\_CODE NOT IN(SELECT DESIG\_CODE FROM DESIGNATION);

These would fetch them.

**Note: I initially thought of and tried a join query (INNER JOIN) and it cause a disaster. :)**

1. Find the name of the female employees with basic greater than the average basic of their respective department.

**Now, this can cause problem.**

**Initially, the query which is tries by me is:**

SELECT a.EMP\_NAME,a.BASIC FROM EMP a, EMP b WHERE a.SEX='F' AND a.DEPT\_CODE=b.DEPT\_CODE AND a.BASIC>(SELECT AVG(BASIC) FROM EMP GROUP BY DEPT\_CODE);

**And, it’s true. I mistook it. Obviously, inner query generates more than 1 row.**

**Now, just first focus on the initial query:**

SELECT AVG(BASIC) FROM EMP GROUP BY DEPT\_CODE;

**This will generate the following:**

|  |
| --- |
| **AVG(BASIC)** |
| 2500000.000000 |
| 2033333.333333 |
| 1366666.666667 |
| 1100000.000000 |
| 400000.000000 |

**SELECT EMP\_NAME,DEPT\_CODE,BASIC FROM EMP WHERE SEX='F';**

|  |  |  |
| --- | --- | --- |
| **EMP\_NAME** | **DEPT\_CODE** | **BASIC** |
| Neha Sen | 00002 | 1900000.00 |
| Tilottoma Majumdar | 00002 | 1600000.00 |
| Sananda Roy | 00005 | 900000.00 |
| Prerona Hazra | 00005 | 700000.00 |

Now, By inspecting these two tables, we could get that the result would be:

|  |  |
| --- | --- |
| Sananda Roy | 900000.00 |
| Prerona Hazra | 700000.00 |

SELECT a.EMP\_NAME,a.BASIC FROM EMP a, EMP b WHERE a.SEX='F' AND a.BASIC>(SELECT AVG(b.BASIC) FROM EMP b WHERE a.DEPT\_CODE=b.DEPT\_CODE GROUP BY b.DEPT\_CODE);

Now, this will generate 16 rows result. (8 pair of these result)

How,

The inner query will generate 5 rows.

The Inner query generates 4 female employees list

But, how 16?

SELECT a.EMP\_NAME, a.BASIC FROM EMP a WHERE a.SEX='F' GROUP BY a.DEPT\_CODE HAVING a.BASIC>(SELECT AVG(b.basic) FROM EMP b WHERE a.DEPT\_CODE=b.DEPT\_CODE GROUP BY b.DEPT\_CODE);

**This does not work.**

SELECT a.EMP\_NAME, a.BASIC FROM EMP a WHERE a.SEX='F' GROUP BY a.DEPT\_CODE HAVING a.BASIC>(SELECT AVG(b.basic) FROM EMP b WHERE a.DEPT\_CODE=b.DEPT\_CODE GROUP BY b.DEPT\_CODE);

**Even this does not work.**

SELECT a.EMP\_NAME, a.BASIC FROM EMP a WHERE a.SEX='F' GROUP BY a.DEPT\_CODE HAVING a.BASIC>(SELECT AVG(b.basic) FROM EMP b WHERE a.DEPT\_CODE=b.DEPT\_CODE GROUP BY b.DEPT\_CODE);

**Now, I read an example and tried this:**

SELECT a.EMP\_NAME, a.Basic from EMP a WHERE a.SEX='F' AND a.BASIC> (SELECT AVG(b.basic) FROM EMP b WHERE a.DEPT\_CODE=b.DEPT\_CODE) group by a.DEPT\_CODE;

**Even this does not work.** The last one should actually give us the result.

Then, I suddenly remember something. What I remember, cannot be expressed through words. Let’s illustrate it using an Example:

select EMP\_NAME, DEPT\_CODE,BASIC from EMP where SEX='F';

|  |  |  |
| --- | --- | --- |
| **EMP\_NAME** | **DEPT\_CODE** | **BASIC** |
| Neha Sen | 00002 | 1900000.00 |
| Tilottoma Majumdar | 00002 | 1600000.00 |
| Sananda Roy | 00005 | 900000.00 |
| Prerona Hazra | 00005 | 700000.00 |

But,

SELECT EMP\_NAME,BASIC, DEPT\_CODE FROM EMP WHERE SEX='F' GROUP BY DEPT\_CODE;

This generates the following:

|  |  |  |
| --- | --- | --- |
| **EMP\_NAME** | **DEPT\_CODE** | **BASIC** |
| Neha Sen | 00002 | 1900000.00 |
| Sananda Roy | 00005 | 900000.00 |

And I just realise, Sometimes how pathetic can I be.

The query would be:

SELECT a.EMP\_NAME,a.BASIC, a.DEPT\_CODE FROM EMP a WHERE a.SEX='F' and a.BASIC > (SELECT AVG(b.BASIC) FROM EMP b WHERE a.DEPT\_CODE=b.DEPT\_CODE);

If you additionally want to solve the problem.

1. Find the number of female managers.

That is easy. I just wrote:

SELECT COUNT(\*) FROM EMP WHERE SEX='F' AND DESIG\_CODE="00002";

To cross check it, I do the following:

SELECT EMP\_NAME, DEPT\_CODE FROM EMP WHERE SEX='F' AND DESIG\_CODE="00002";

And, It gives the following result:

|  |  |
| --- | --- |
| EMP\_NAME | DEPT\_CODE |
| Neha Sen | 00002 |

So, it’s ok.