```
/*  Project # 3  */

/* JOINING*/
/* 1.  Find out the name of employees whose salary is between 3000 and 5000 from each
Department and location.*/

SELECT e.ename, e.sal, d.deptno, d.dname, d.loc
  FROM scott.emp e INNER JOIN scott.dept d
  ON e.deptno = d.deptno
  where e.sal between 3000 and 5000;

/* 2.  List the number of employees whose hire date between December 1980 and 1981.*/

select  d.dname, count(e.empno) as TotalEmpNO
from scott.dept d left outer join scott.emp e
on e.deptno = d.deptno
where e.hiredate between '17-dec-80' and '03-dec-81'
group by d.dname;

/*AGGREGATING*/
/* Find the maximum salary of each dept which is greater than 2000*/

SELECT deptno, MAX(sal)
FROM scott.emp
GROUP BY deptno
HAVING MAX(sal) > 2000;
```

/*Display the information or data of most senior employee who belongs to the year 1981.*/

```
select * from scott.emp
where hiredate = (select min(hiredate)
from scott.emp where hiredate like '%81') ;
```

/*SUBQUERIES*/

/* 1. List the deptno where there are no emps*/

```
select deptno, dname from scott.dept where deptno not in
(select deptno from scott.emp) ;
```

/* 2. Find the name and job of the emps who earn Max salary and Commission.*/

```
select ename, job, sal, comm
from scott.emp
where sal=(select max(sal) from scott.emp);
```

/*FUNCTION*/

/*List the emps whose sal contain 3 digits*/

```
select * from scott.emp where length(sal) = 3 ;
```

```
/* Display only the list of  the employee first name  and position of the employee. */

select 'EmployeeFirstName: ' || ename ||'-----' || 'Position: ' || job
from scott.emp;

/*DATE FORMAT MANIPULATION*/

/*List the emps with hiredate in format June 4,1988.*/

select ename, to_char(hiredate,'Month DD,YYYY.')
from scott.emp ;



/*How to make sure the data was clean and accurate

/*    1.  Ensuring no data was lost in joins
In order to show no data was lost during combining tables, we use the ‘where not exists’ operator in the
following script. We use this operator for left outer join or right out joins, as the inner join gives the
intersection of the combined table and the full outer join give the whole combined data and the value
give no
data found as there is no missing data.*/

Select e.deptno
from scott.emp e
WHERE   NOT EXISTS ( select *
from  scott.dept d
where e.deptno = d.deptno );
```

/* 2. Accounting for null values

The SQL NULL is the term used to represent a missing value. A NULL value in a table is a value in a field that appears to be blank. A field with a NULL value is a field with no value. During table creation we use not null which signifies, that column should always accept an explicit value of the given data type.

In order to check the null values we use IS NULL operator.*/

```
select ename, job, comm
from scott.emp
where comm is null;
```

/* 3. Checking duplication

In order to show this I used the following script to see if the data is occurring more than one time in the EMP table of the ename column. Since there is no any duplication the result shows tell us that there is no data found. We can also change > 1 to =1 to check if the data is occurring exactly once.*/

```
SELECT ename,
COUNT(ename) AS NumOccurrences
FROM scott.emp
GROUP BY ename
HAVING ( COUNT(ename) > 1 ) ;  -- if = 1 exact occurence can be displayed
```

https://livesql.oracle.com/apex/livesql/s/jrknkeyyillbm79u2gff1gdps