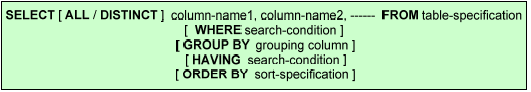
Data Retrieval Language

Select query is used to retrieve the data from the table

Select Query Syntax



Test Data

CREATE TABLE `dept` (

`DEPTNO` integer NOT NULL,

`DNAME` varchar(20) NOT NULL,

`LOC` varchar(20) NOT NULL,

PRIMARY KEY (`DEPTNO`)

);

INSERT INTO `dept` VALUES (10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO `dept` VALUES (20, 'RESEARCH', 'DALLAS');

INSERT INTO `dept` VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO `dept` VALUES (40, 'OPERATIONS', 'BOSTON');

CREATE TABLE `emp` (

`EMPNO` integer NOT NULL,

`ENAME` varchar(20) NOT NULL,

`JOB` varchar(20) NOT NULL,

`MGR` integer,

`HIREDATE` date NOT NULL,

`SAL` integer NOT NULL,

`COMM` integer,

`DEPTNO` integer NOT NULL,

PRIMARY KEY (`EMPNO`),

CONSTRAINT `fk\_DEPTNO` FOREIGN KEY (`DEPTNO`) REFERENCES `dept` (`DEPTNO`)

);

INSERT INTO `emp` VALUES (7839, 'KING', 'PRESIDENT', NULL, '1981-11-17', 5000, NULL, 10);

INSERT INTO `emp` VALUES (7698, 'BLAKE', 'MANAGER', 7839, '1981-05-01', 2850, NULL, 30);

INSERT INTO `emp` VALUES (7654, 'MARTIN', 'SALESMAN', 7698, '1981-09-28', 1250, 1400, 30);

INSERT INTO `emp` VALUES (7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20', 1600, 300, 30);

INSERT INTO `emp` VALUES (7521, 'WARD', 'SALESMAN', 7698, '1981-02-22', 1250, 500, 30);

INSERT INTO `emp` VALUES (7900, 'JAMES', 'CLERK', 7698, '1981-12-03', 950, NULL, 30);

INSERT INTO `emp` VALUES (7844, 'TURNER', 'SALESMAN', 7698, '1981-09-08', 1500, 0, 30);

INSERT INTO `emp` VALUES (7782, 'CLARK', 'MANAGER', 7839, '1981-06-09', 2450, NULL, 10);

INSERT INTO `emp` VALUES (7934, 'MILLER', 'CLERK', 7782, '1982-01-23', 1300, NULL, 10);

INSERT INTO `emp` VALUES (7566, 'JONES', 'MANAGER', 7839, '1981-04-02', 2975, NULL, 20);

INSERT INTO `emp` VALUES (7788, 'SCOTT', 'ANALYST', 7566, '1982-12-09', 3000, NULL, 20);

INSERT INTO `emp` VALUES (7876, 'ADAMS', 'CLERK', 7788, '1983-01-12', 1100, NULL, 20);

INSERT INTO `emp` VALUES (7902, 'FORD', 'ANALYST', 7566, '1981-12-03', 3000, NULL, 20);

INSERT INTO `emp` VALUES (7369, 'SMITH', 'CLERK', 7902, '1980-12-17', 800, NULL, 20);

CREATE TABLE `proj` (

`PROJID` integer NOT NULL,

`EMPNO` integer NOT NULL,

`STARTDATE` date NOT NULL,

`ENDDATE` date NOT NULL,

PRIMARY KEY (`PROJID`),

CONSTRAINT `fk\_PROJ` FOREIGN KEY (`EMPNO`) REFERENCES `emp` (`EMPNO`)

);

INSERT INTO `proj` VALUES (1, 7782, '2005-06-16', '2005-06-18');

INSERT INTO `proj` VALUES (4, 7782, '2005-06-19', '2005-06-24');

INSERT INTO `proj` VALUES (7, 7782, '2005-06-22', '2005-06-25');

INSERT INTO `proj` VALUES (10, 7782, '2005-06-25', '2005-06-28');

INSERT INTO `proj` VALUES (13, 7782, '2005-06-28', '2005-07-02');

INSERT INTO `proj` VALUES (2, 7839, '2005-06-17', '2005-06-21');

INSERT INTO `proj` VALUES (8, 7839, '2005-06-23', '2005-06-25');

INSERT INTO `proj` VALUES (14, 7839, '2005-06-29', '2005-06-30');

INSERT INTO `proj` VALUES (11, 7839, '2005-06-26', '2005-06-27');

INSERT INTO `proj` VALUES (5, 7839, '2005-06-20', '2005-06-24');

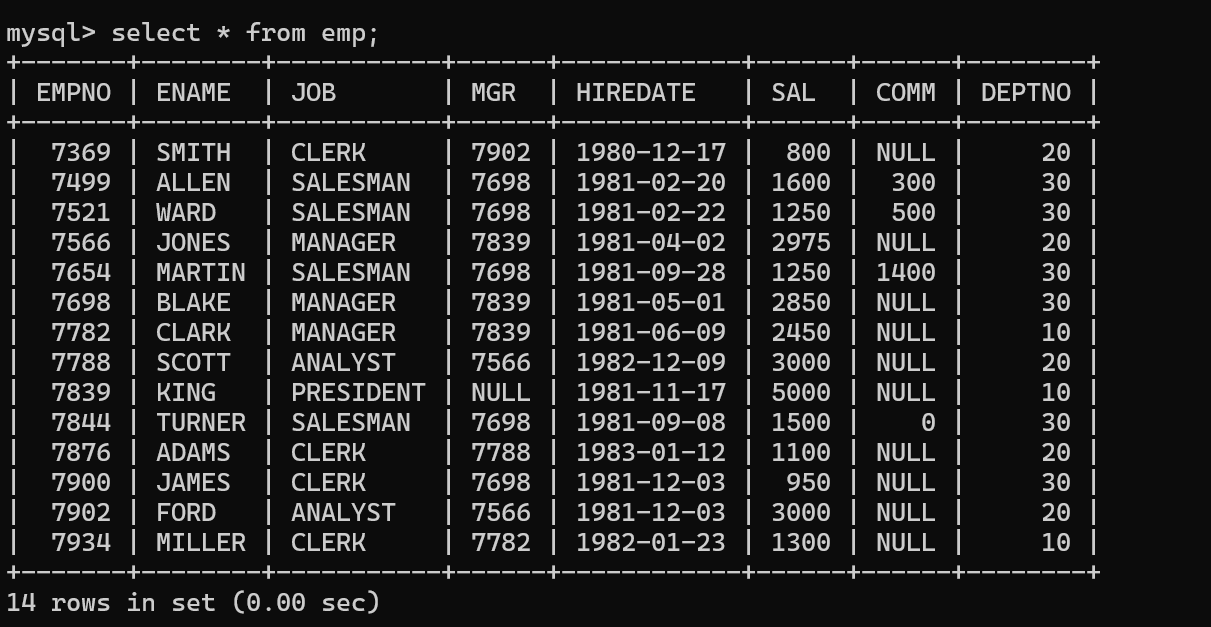
INSERT INTO `proj` VALUES (3, 7934, '2005-06-18', '2005-06-22');

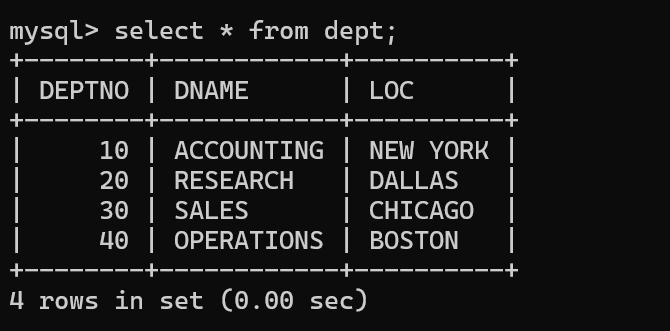
INSERT INTO `proj` VALUES (12, 7934, '2005-06-27', '2005-06-28');

INSERT INTO `proj` VALUES (15, 7934, '2005-06-30', '2005-07-03');

INSERT INTO `proj` VALUES (9, 7934, '2005-06-24', '2005-06-27');

INSERT INTO `proj` VALUES (6, 7934, '2005-06-21', '2005-06-23');





Q1) SQL query to retrieve all the information from emp table

select \* from emp;

Q2) SQL query to retrieve all the information from dept table

select \* from dept;

Q3) SQL query to display the list of Job names in emp table

select job from emp; --OR –

select all job from emp;

Q4) display the list of distinct Job names

select distinct job from emp;

distinct 🡪 retrieves the unique values present in the column

all 🡪 retrieves all the values including duplicates present in the column

\* 🡪 retrieves the values from all the columns present in the table

In order to get the data from specific columns, we must provide the list of column names between select and from keywords.

Q5) Retrieve the employee id from emp table

select empno from emp;

Q6) Retrieve employee id, employee name and job of every employee from emp table

select empno,ename,job from emp;

Q7) Retrieve ename as employee name from employee table

select ename "Employee Name" from emp;

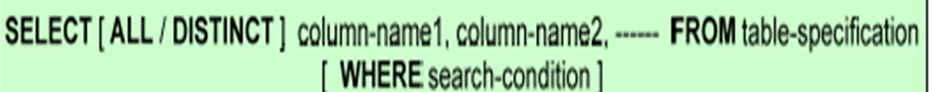
How to define an Alias name to the column in the output?

<<column-name-in-table>> “Alias-name-to-the column”

select empno "Employee Number", ename "Employee Name" from emp;

Q8) retrieve the empid, total salary (salary+comm) of every employee from emp

select empno, sal+ifnull(comm,0) "Total Salary" from emp;



Where clause in the select query is used to filter the rows from the output or result set

Q) Retrieve the employee information whose emp id is 7369

select \* from emp where empno=7369;

Note: we must use like operator when ever we are comparing varchar type of data

Q) Retrieve the information of SMITH

select \* from emp where ename like 'SMITH';

We must use Like operator even to compare date value in where clause

Q) SQL query to retrieve the list of employees whose hire date is 17 December,1980

select \* from emp where hiredate like '1980-12-17';

How to define the range of values in where clause?

Two ways:

1. Relational and logical operators
   1. >,<,>=,<=,<>,= 🡪 Relational operators
   2. AND, OR, NOT 🡪 Logical operators
2. A special operator called BETWEEN…AND…..

Q) List of employees whose empid is between 7369 and 7839

select \* from emp where empno>=7369 AND empno<=7839;

-- OR –

select \* from emp where empno BETWEEN 7369 AND 7839;

Q) List of employees who hired between 17-12-1980 and 17-11-1981

select \* from emp where hiredate>='1980-12-17' AND hiredate<='1981-11-17';

-- OR –

select \* from emp where hiredate BETWEEN '1980-12-17' AND '1981-11-17';

How to retrieve the data that satisfies the list of given values

1. Logical OR operator
2. A special operator called IN

Q) Retrieve the employee information of King, Blake and Smith

select \* from emp where ename like 'KING' OR ename like 'BLAKE' OR ename like 'SMITH';

-- OR –

select \* from emp where ename IN ('KING','BLAKE','SMITH');

Q) Retrieve the employee information of 7369,7499

select \* from emp where empno=7369 OR empno=7499;

-- OR --

select \* from emp where empno IN (7369,7499);

Q) Retrieve the employee information whose hire date is 17-12-1980 and 17-11-1981

select \* from emp where hiredate like '1980-12-17' OR hiredate like '1981-11-17';

-- OR –

select \* from emp where hiredate IN ('1980-12-17','1981-11-17');

How to retrieve the data based on some expression (regular expression)?

We can provide the format of value by using the regular expression.

Two Symbols:

* % 🡪 Zero or more symbols
* \_ (Underscore) 🡪 Exactly one Symbol

Q) Retrieve the employees information whose name starts with S

select \* from emp where ename like 'S%';

Q) Retrieve the employees information whose name ends with G

select \* from emp where ename like '%G';

Q) Retrieve the employees information whose contains I

select \* from emp where ename like '%I%';

Q) Retrieve the employees information whose contains E as a second character from Last

select \* from emp where ename like '%E\_';

Q) Retrieve the employees information whose name contains exactly 4 characters

select \* from emp where ename like '\_\_\_\_';(4 underscore symbols)

Q) Retrieve all employees who salary is Three digit number

select \* from emp where sal like '\_\_\_';(2 Underscores)

Q) Retrieve all employees who joined in Jan month.

select \* from emp where hiredate like '%-01-%';

Q) Retrieve all employees who joined in the year 1980

select \* from emp where hiredate like '1980%';

Q) Retrieve all employees who joined on 17th day of any month and any year

select \* from emp where hiredate like '%-17';

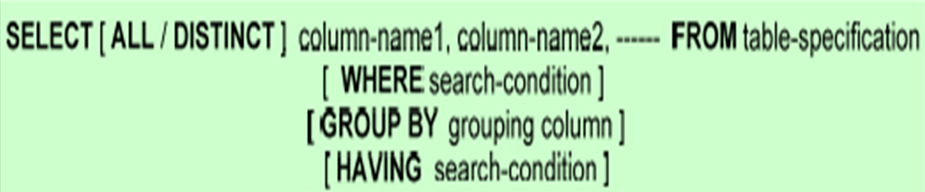
another special operator is IS NULL and IS NOT NULL

Q) Retrieve all employees who are not getting any commission

select \* from emp where comm IS NULL;

Q) Retrieve all employees who are getting commission

select \* from emp where comm IS NOT NULL;



Group by clause in Select is used to group the related rows together in order to get the consolidated information

FIVE Group By Functions

1. Sum()
2. Avg()
3. Max()
4. Min()
5. Count()

Q) SQL Query to retrieve the number of employees working in each designation

select job,count(\*) from emp group by job;

Q) SQL query to retrieve the job wise total salary paid to all the employees who are in same job

select job,sum(sal) from emp group by job;

Q) SQL query to find the max, min and average salary paid to all the employees who are working in same job

select job,max(sal),min(sal),avg(sal) from emp group by job;

Note:

we can use any number of columns in group by clause

Q) SQL query to find the max, min and average salary paid to all the employees who are working in same job and same department

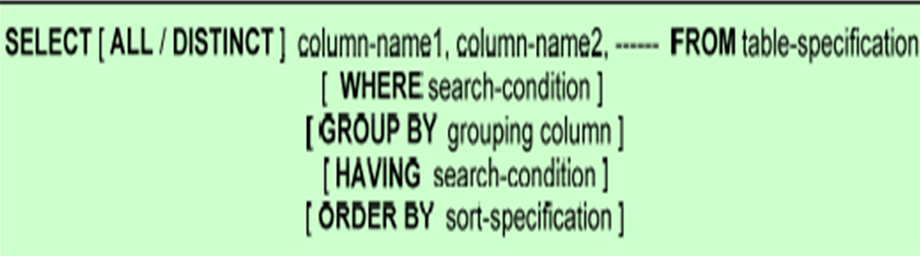
select job,deptno,max(sal),min(sal),avg(sal) from emp group by job, deptno;

Having Clause:

Having clause in group by is used to filter the groups.

Q) SQL query to find the max, min and average salary paid to all the employees who are working in same job and same department and the average salary of job,dept is > 2000

select job,deptno,max(sal),min(sal),avg(sal) from emp group by job,deptno having avg(sal)>2000;



Order by clause in Select query is used to retrieve the data in either ascending or descending order

Q) SQL query to find the max, min and average salary paid to all the employees who are working in same job and same department and the average salary of job,dept is > 2000. Arrange the result in ascending order by deptno

select job,deptno,max(sal),min(sal),avg(sal) from emp group by job,deptno having avg(sal)>2000 order by deptno asc;

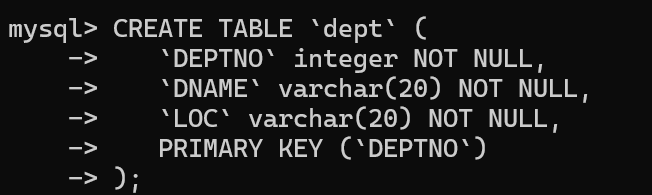
-- OR –

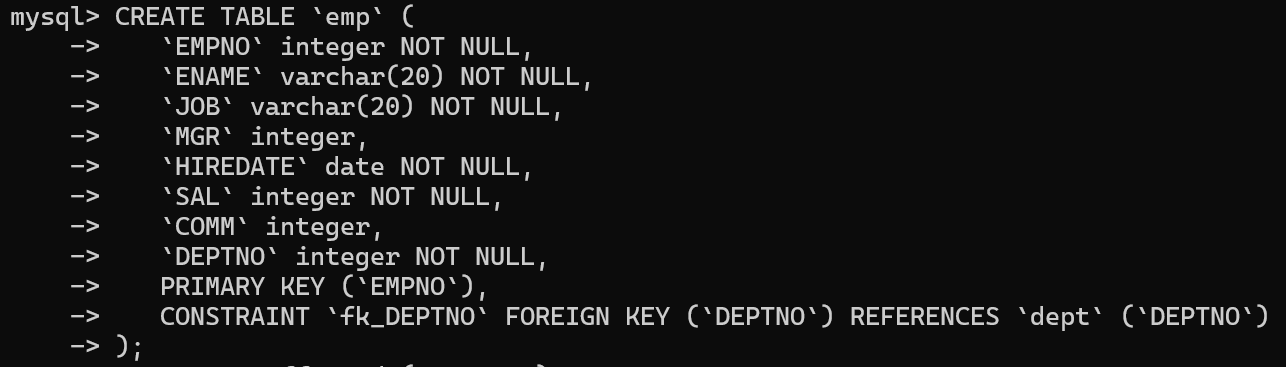
select job,deptno,max(sal),min(sal),avg(sal) from emp group by job,deptno having avg(sal)>2000 order by deptno;

Q) SQL query to find the max, min and average salary paid to all the employees who are working in same job and same department and the average salary of job,dept is > 2000. Arrange the result in descending order by deptno

select job,deptno,max(sal),min(sal),avg(sal) from emp group by job,deptno having avg(sal)>2000 order by deptno desc;

How to get the data by Joining More than one table?





Q) SQL Query to retrieve the below information:

* Empno
* Empname
* Department name
* Location

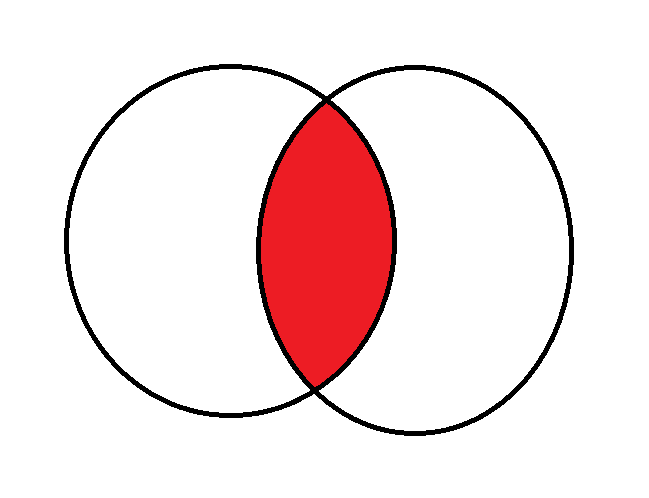
select e.empno,e.ename,d.dname,d.loc from emp e JOIN dept d on e.deptno=d.deptno;

The required information is present in employee and department table. So, in order to get the information from both the tables, we have to apply the concept called JOINS

JOINS in SQL is used to get the information by combining two or more tables

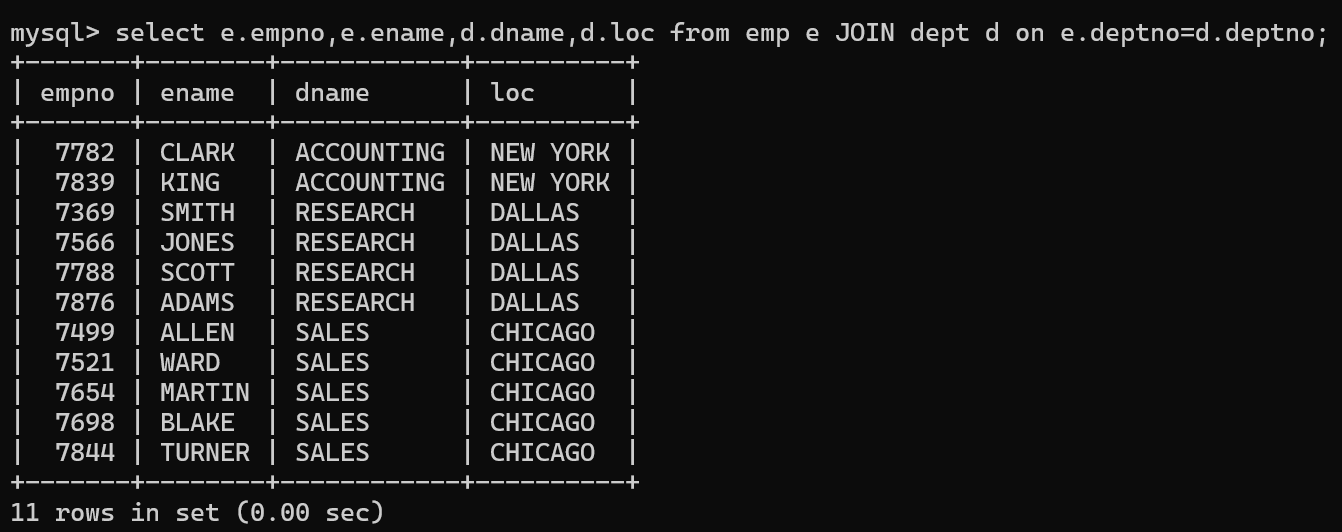
In the above scenario, we have to use a type of JOIN called EQUI JOIN or Inner join

Bcz I want to retrieve those employees info. Who are working in some department



Equi Join or Inner join retrieves the matched data between both the tables.

If any unmatched is available in any table, then equi join ignores that data



Other Type of Join is OUTER JOIN

Outer join is used to retrieve both matched data and unmatched data.

If we apply LEFT OUTER JOIN, then unmatched data from LHS table

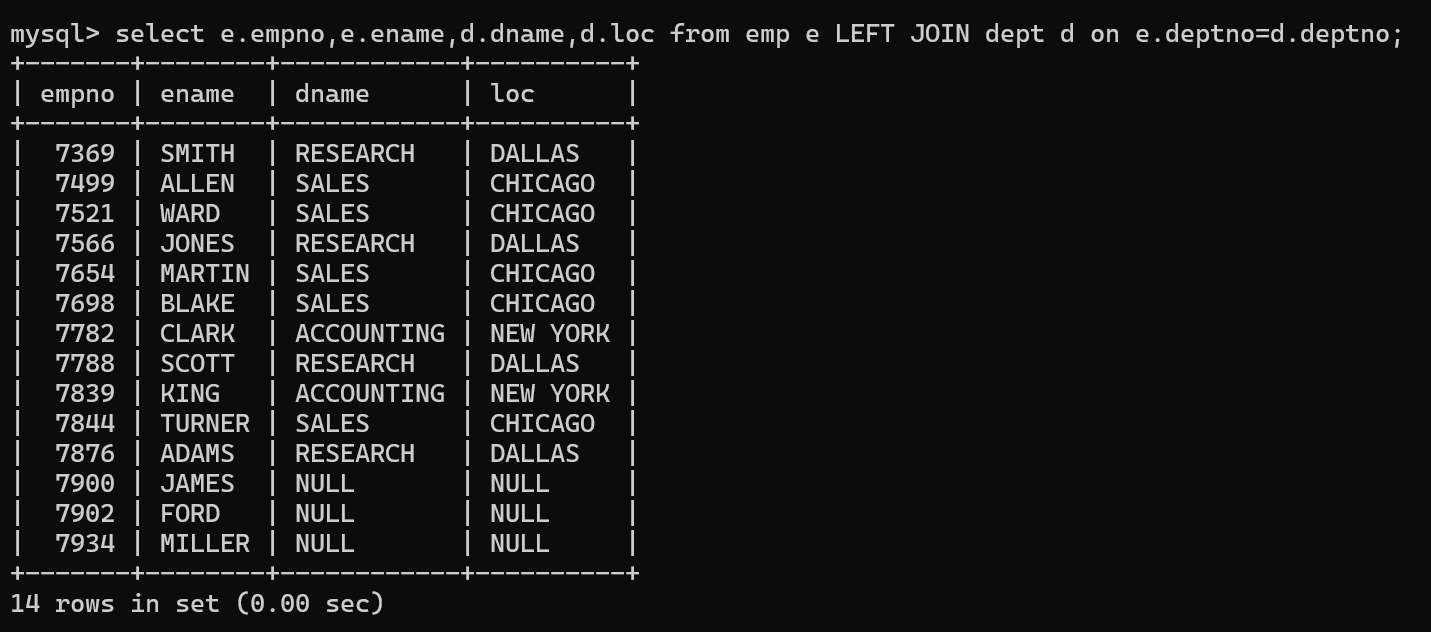
If we apply RIGHT OUTER JOIN, then unmatched data from RHS table

If we apply FULL JOIN, then unmatched data from both the tables

Query to apply Left Join

select e.empno,e.ename,d.dname,d.loc from emp e LEFT JOIN dept d on e.deptno=d.deptno;

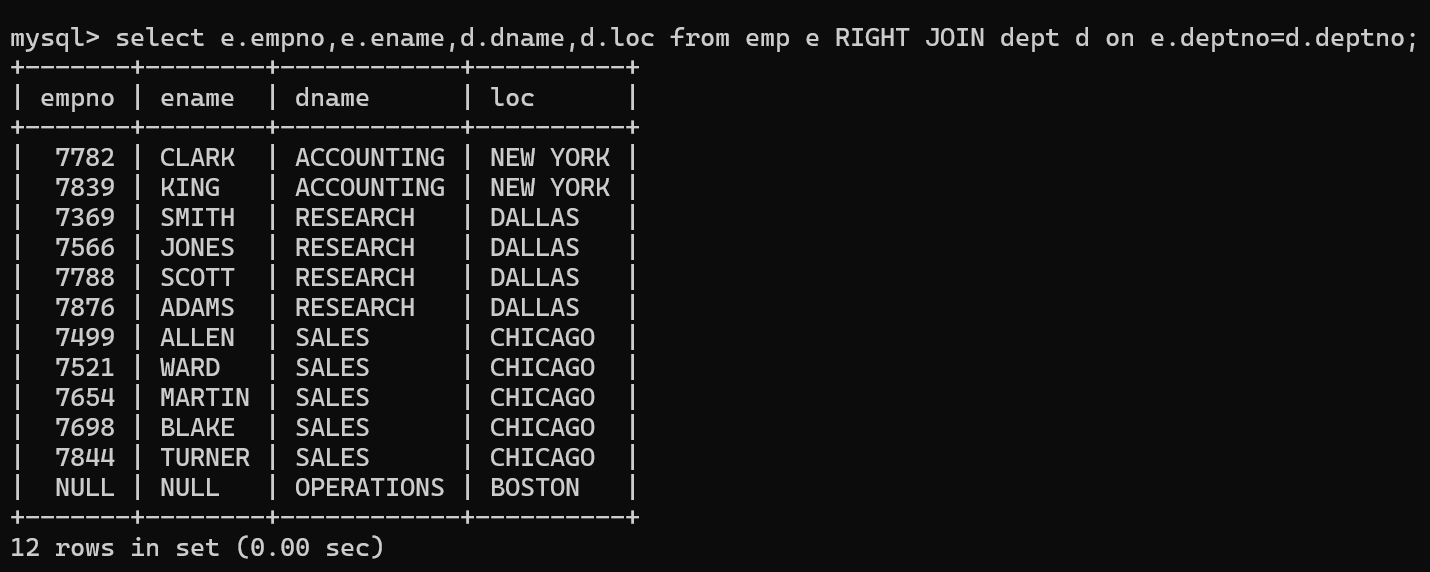
retrieves all the matched rows from both the tables and Unmatched rows from LHS



Query to apply Right Join

select e.empno,e.ename,d.dname,d.loc from emp e RIGHT JOIN dept d on e.deptno=d.deptno;

All the matched rows from both the tables and unmatched row from the RHS table



Query to apply FULL Join

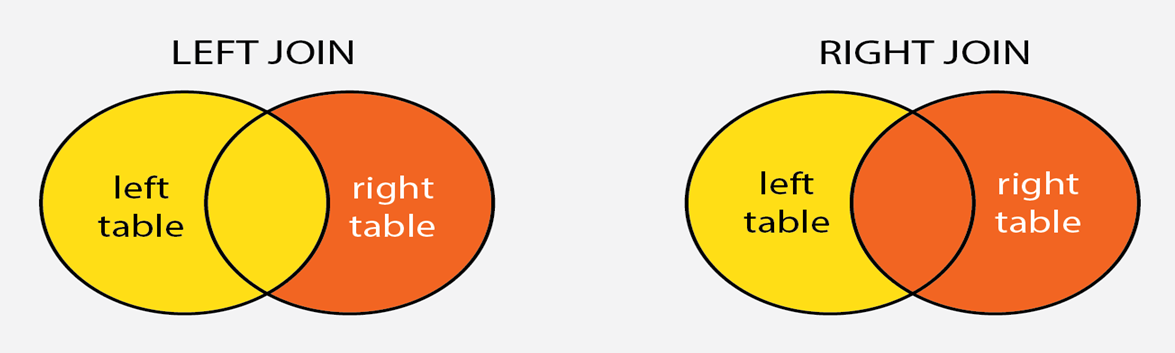
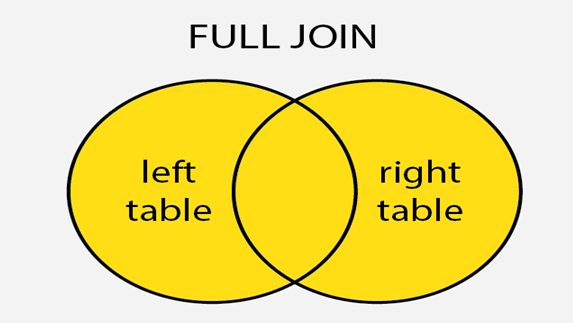
select e.empno,e.ename,d.dname,d.loc from emp e FULL JOIN dept d on e.deptno=d.deptno;

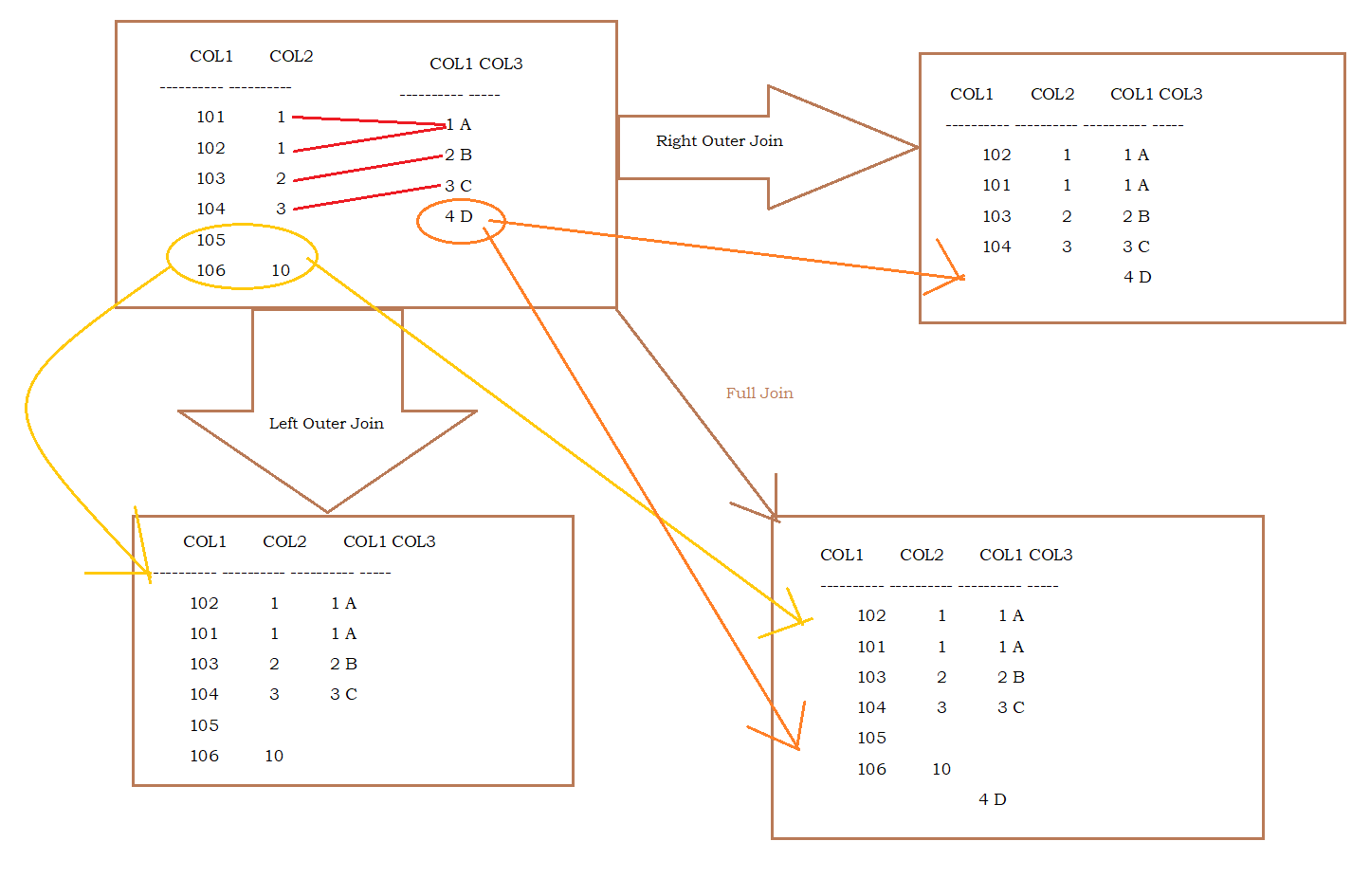
All the matched rows from both the tables

Unmatched row from LHS table

Unmatched row from RHS table

Note: the above query works only in Oracle SQL. MySQL Won’t support FULL JOIN

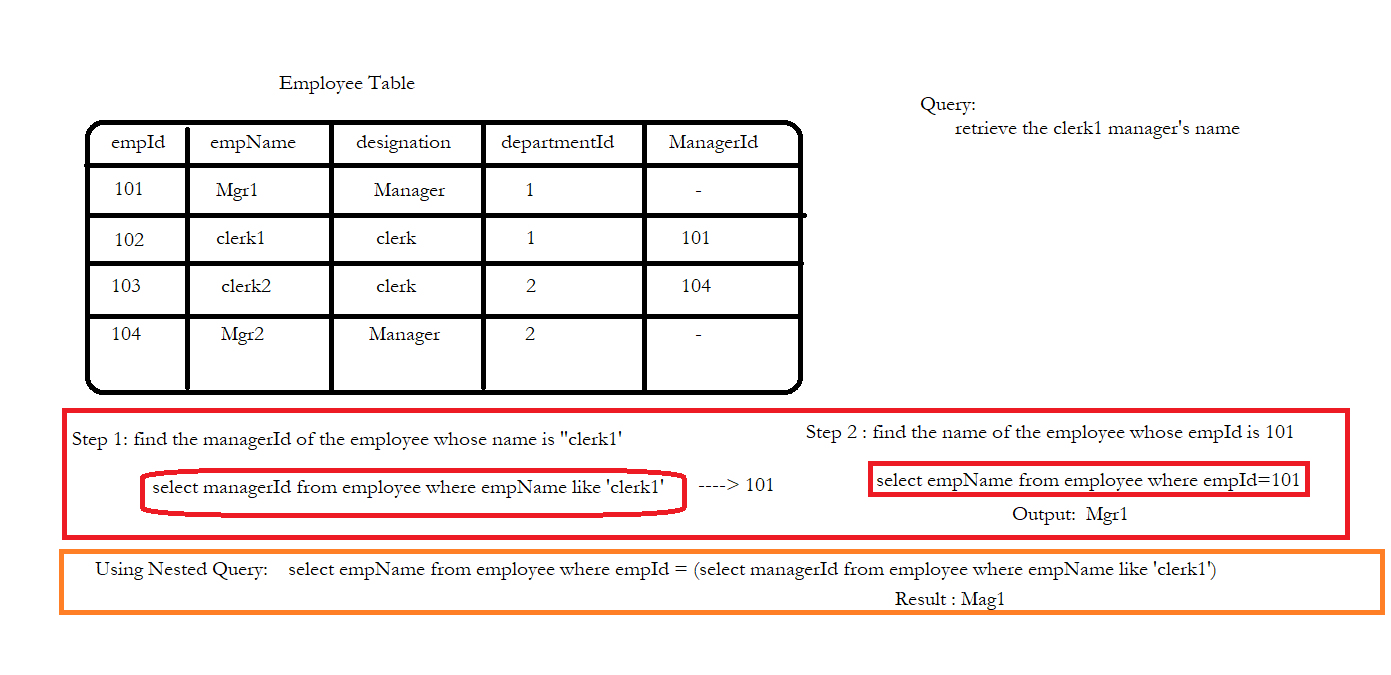




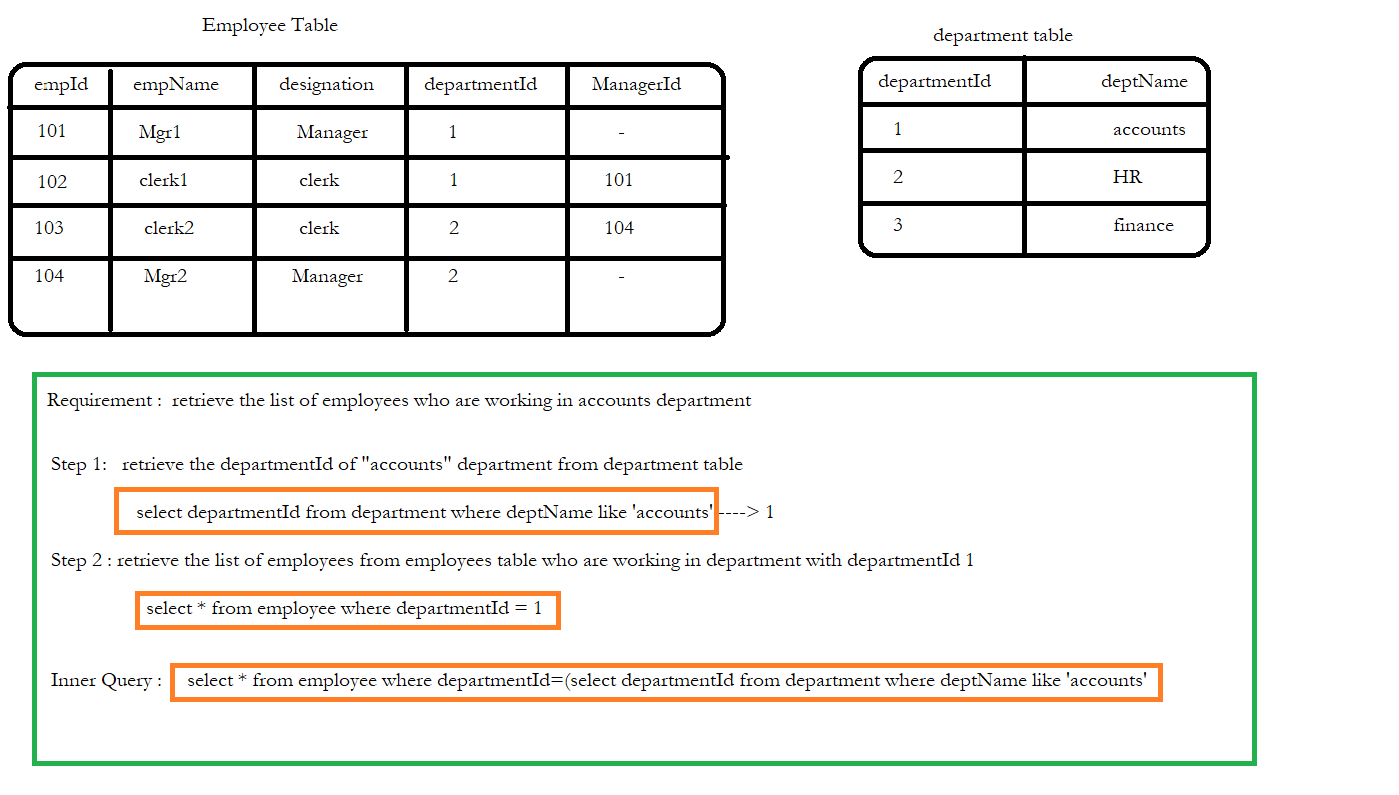
Sub Queries

Writing a query in the where of another query is called Nested Query

Use case – 1



Use Case – 2



Q) SQL query to retrieve the list of employees working in accounting department

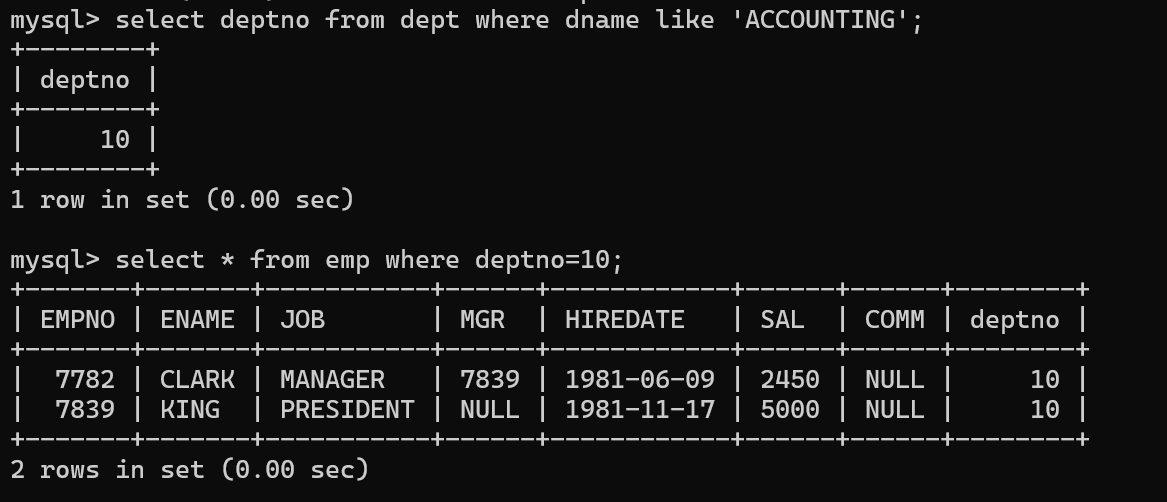
Step 1: find the department id of accounts from dept table

select deptno from dept where dname like 'ACCOUNTING';

Output: 10

Step 2: find the list of employees who are working in department number 10 from emp table

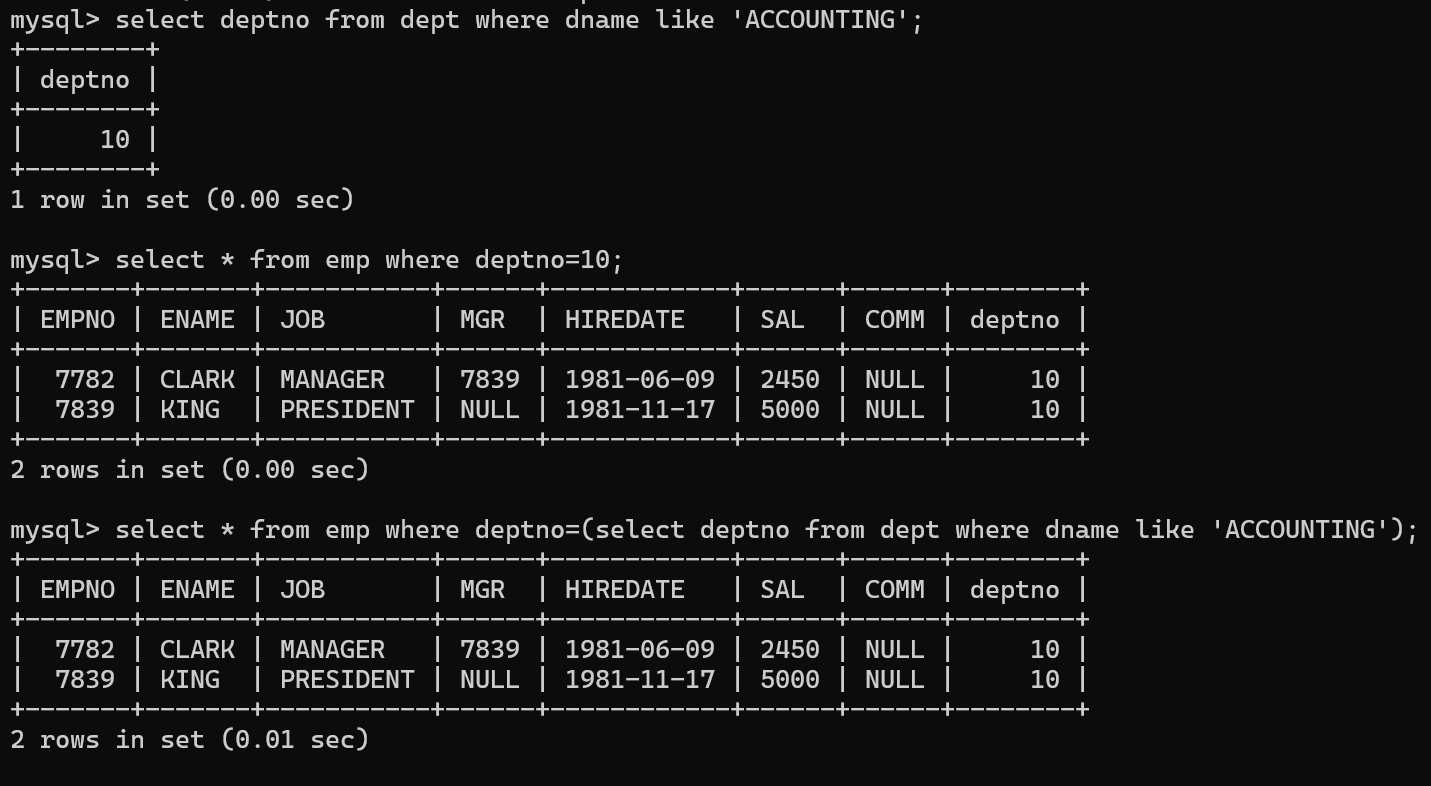
select \* from emp where deptno=10;

****

**Instead of executing the queries step by step, we can take help of sub queries concept**

**We can rewrite the above queries using sub queries:**

select \* from emp where deptno=(select deptno from dept where dname like 'ACCOUNTING');

****

**TYPES OF SUB QUERIES**

1. Single row subqueries
   1. The above query is the example for single row sub query. Bcz the sub query in the above example is returning a single value
2. Multi row subqueries

If the sub query is returning more than one value, then such a type of sub query is called Multi row sub queries

We have to use a special operators called IN or ANY or ALL to handle multi row sub query

Q) SQL query to retrieve the list of employees working in accounting and Research department

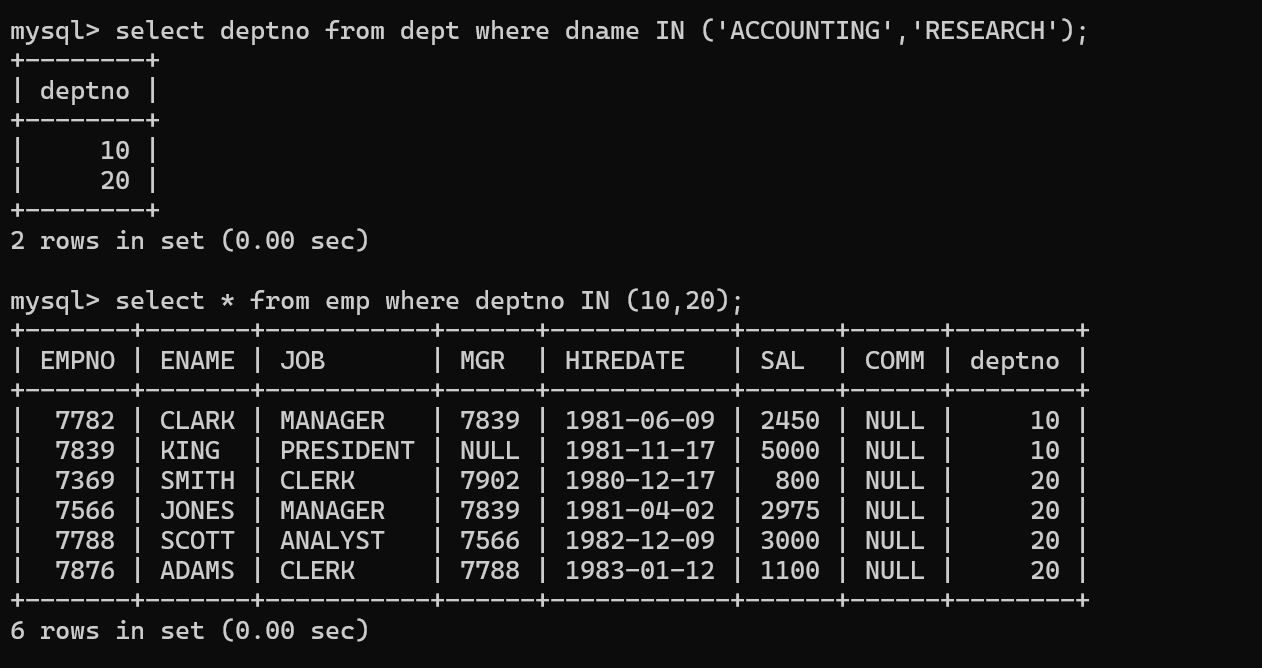
Step 1: find the department id of accounts and Research from dept table

select deptno from dept where dname IN ('ACCOUNTING','RESEARCH');

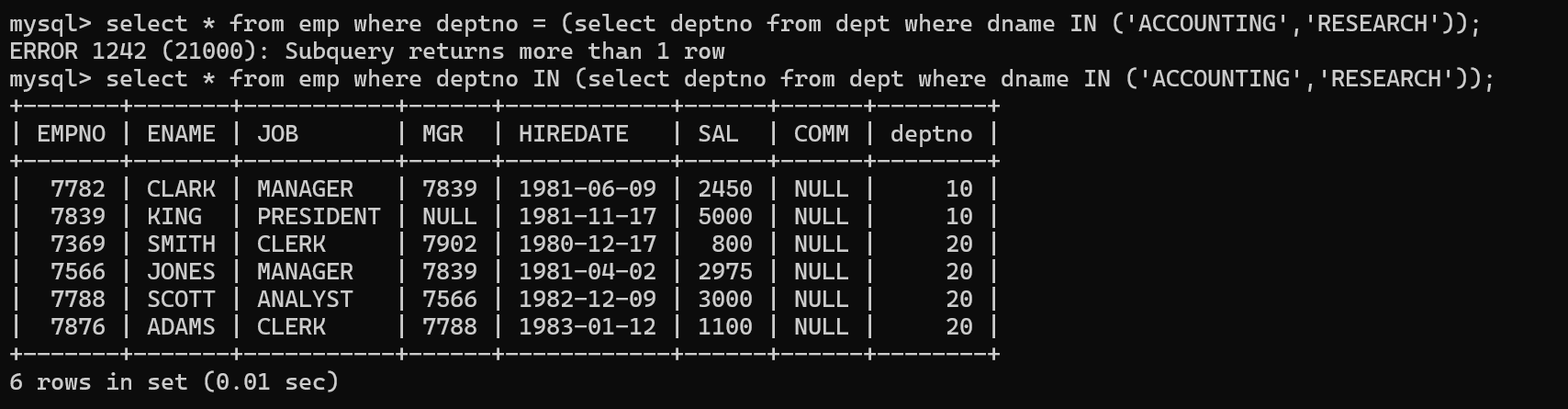
Output: 10, 20

Step 2: find the list of employees who are working in department number 10 and 20 from emp table

select \* from emp where deptno IN (10,20);

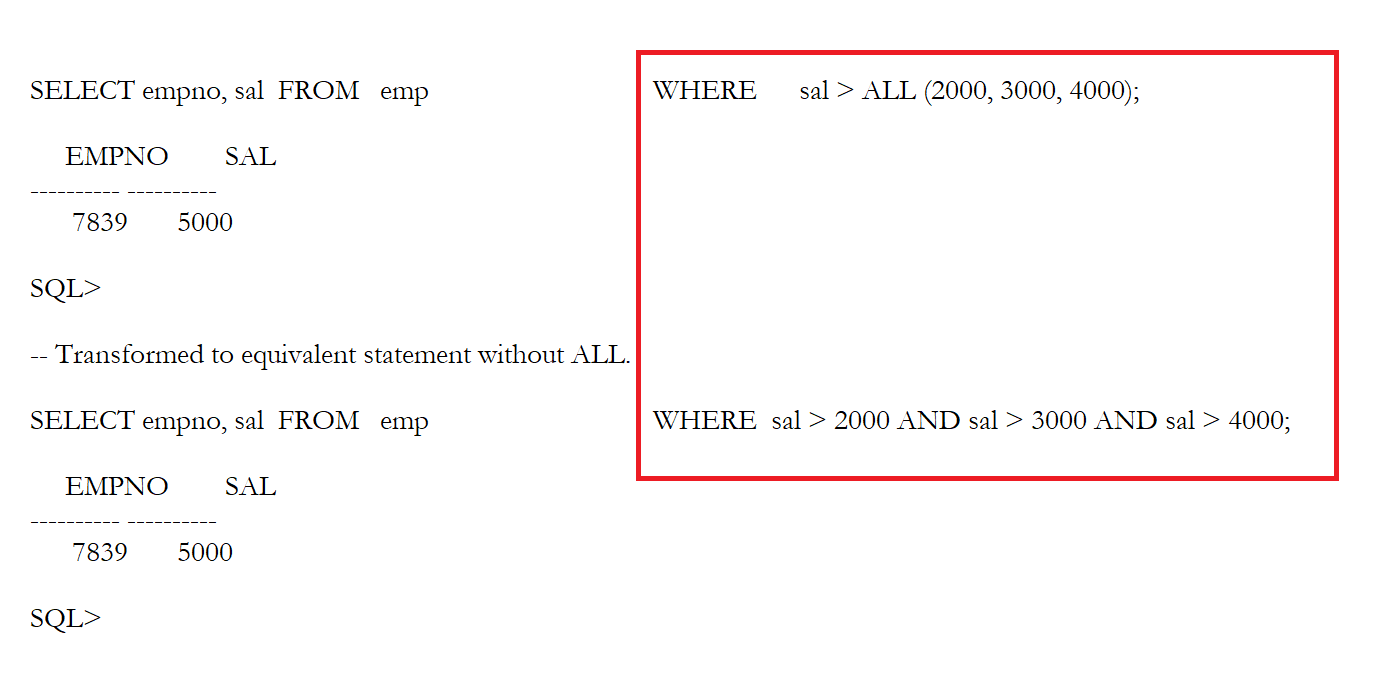


Using Sub Queries

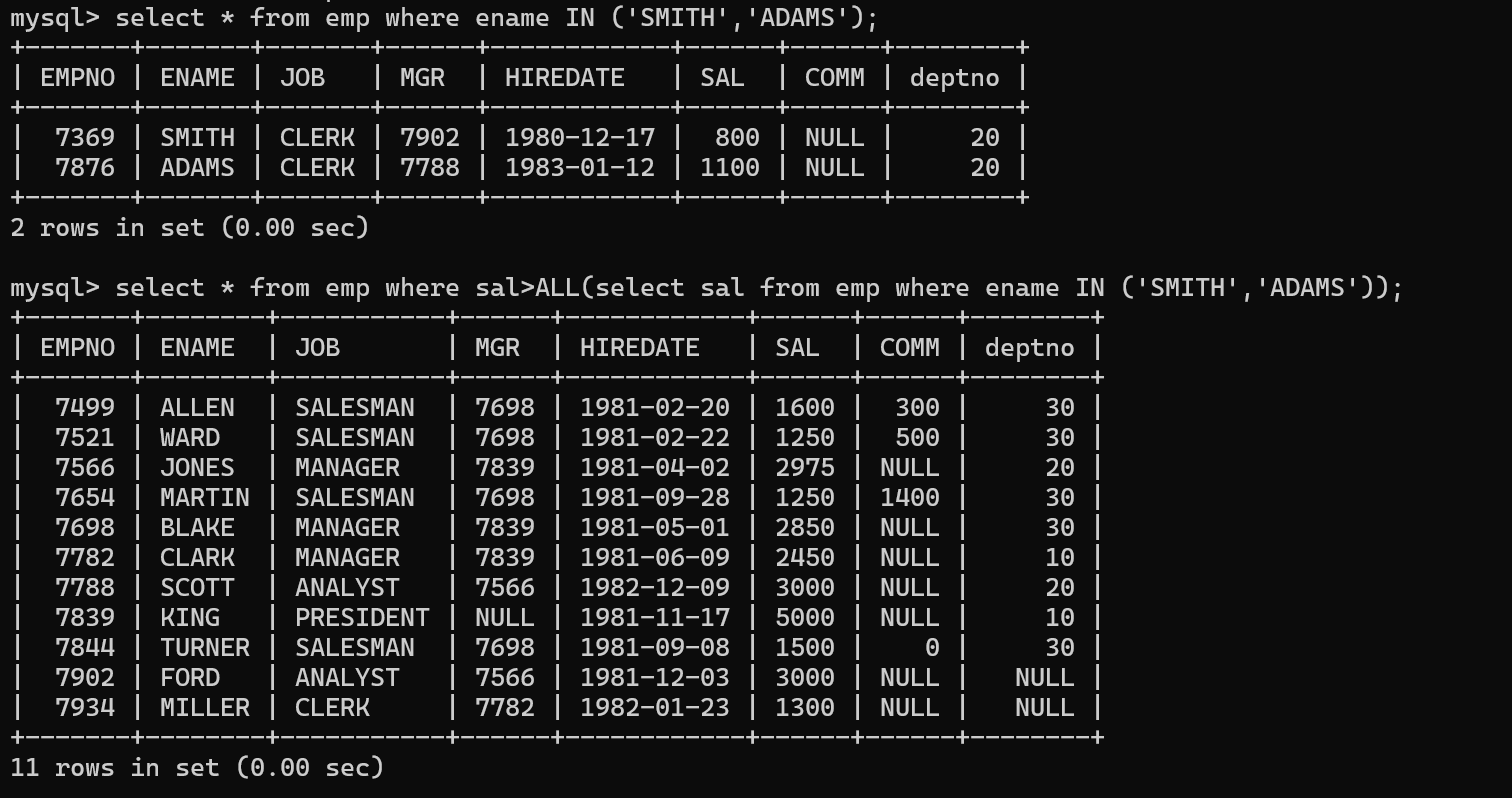


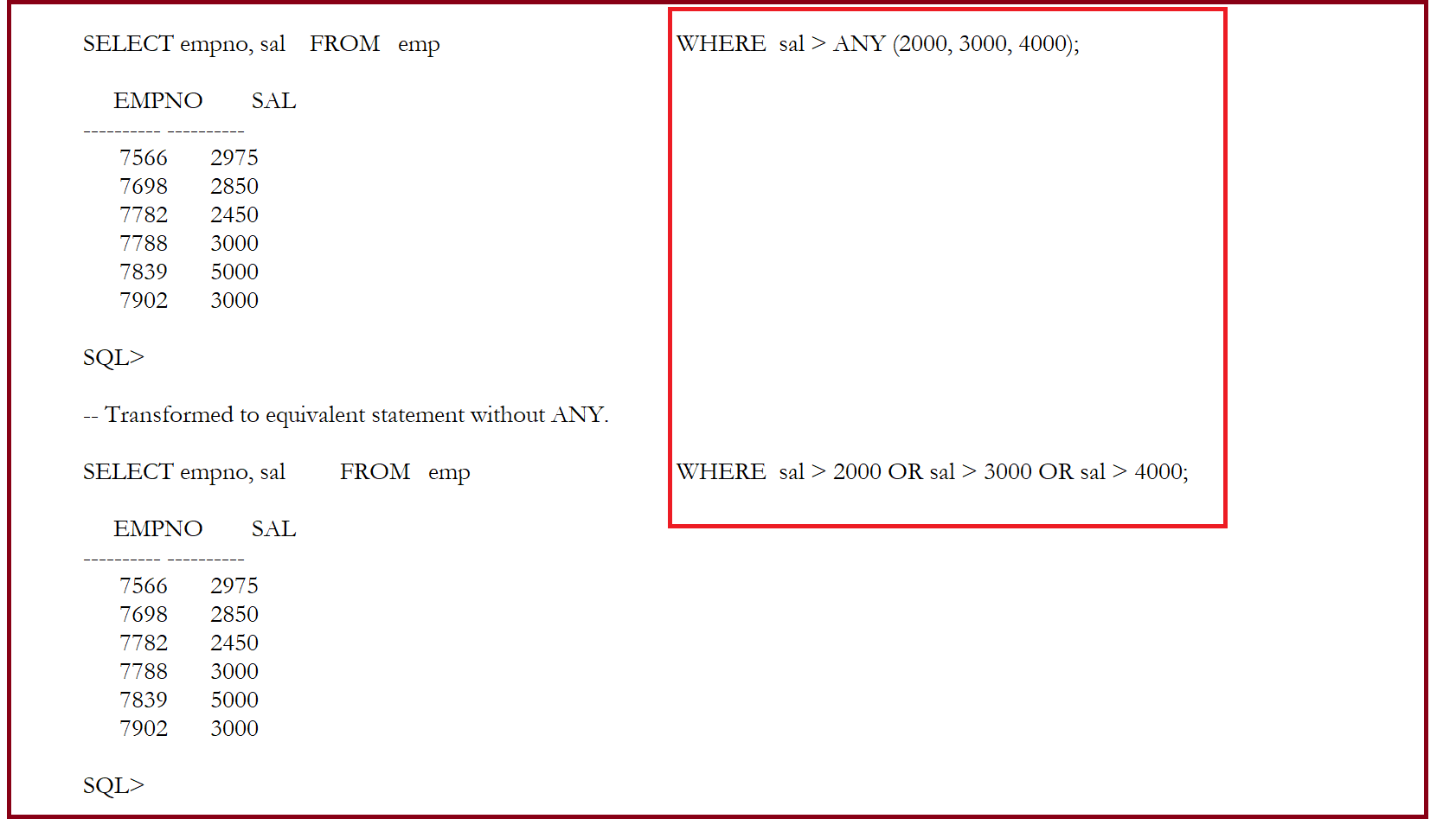
**Multi row subqueries**

* In multi row subquery, it will return more than one value.
* In such cases we should include operators like IN, ANY, ALL, EXISTS or NOT IN between the comparison operator and the subquery.
* As sub-query is returning more than one value, we can't use any relational operator alone in where clause of outer sub-query.

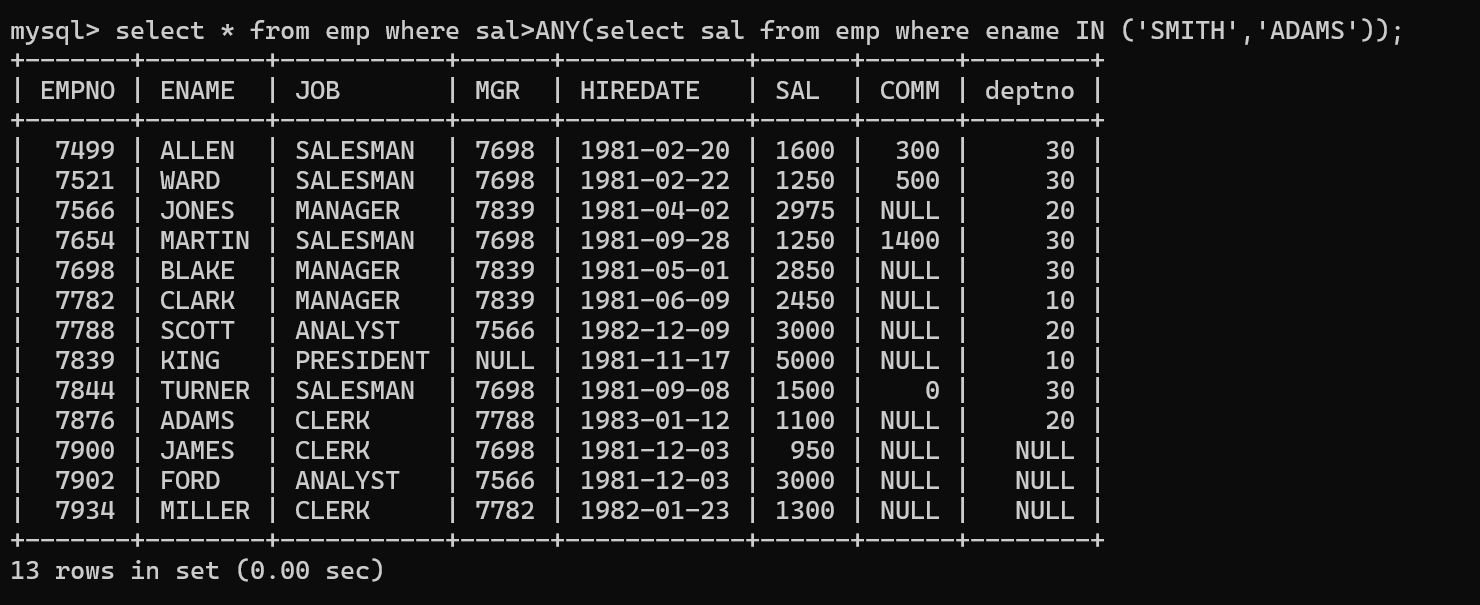


Q) SQL query to retrieve all the employees whose salary is > salary of SMITH and ADAMS



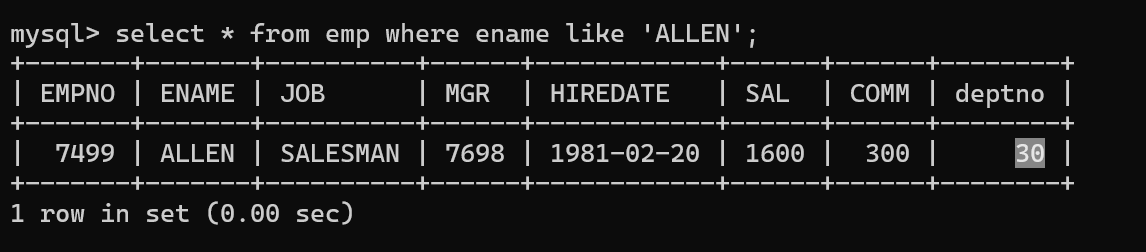


Q) SQL query to retrieve all the employees whose salary is > salary of SMITH or salary of ADAMS

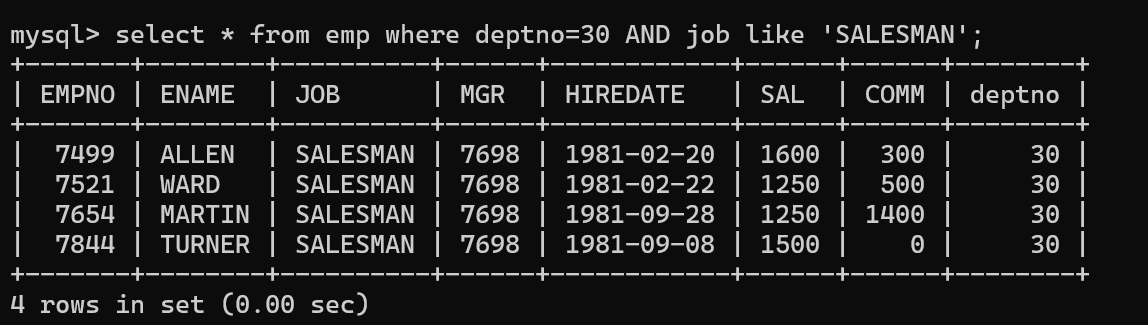


1. Correlated subqueries

SQL query to retrieve the list of employees who are working in the department where Allen is working with the designation what Allen has



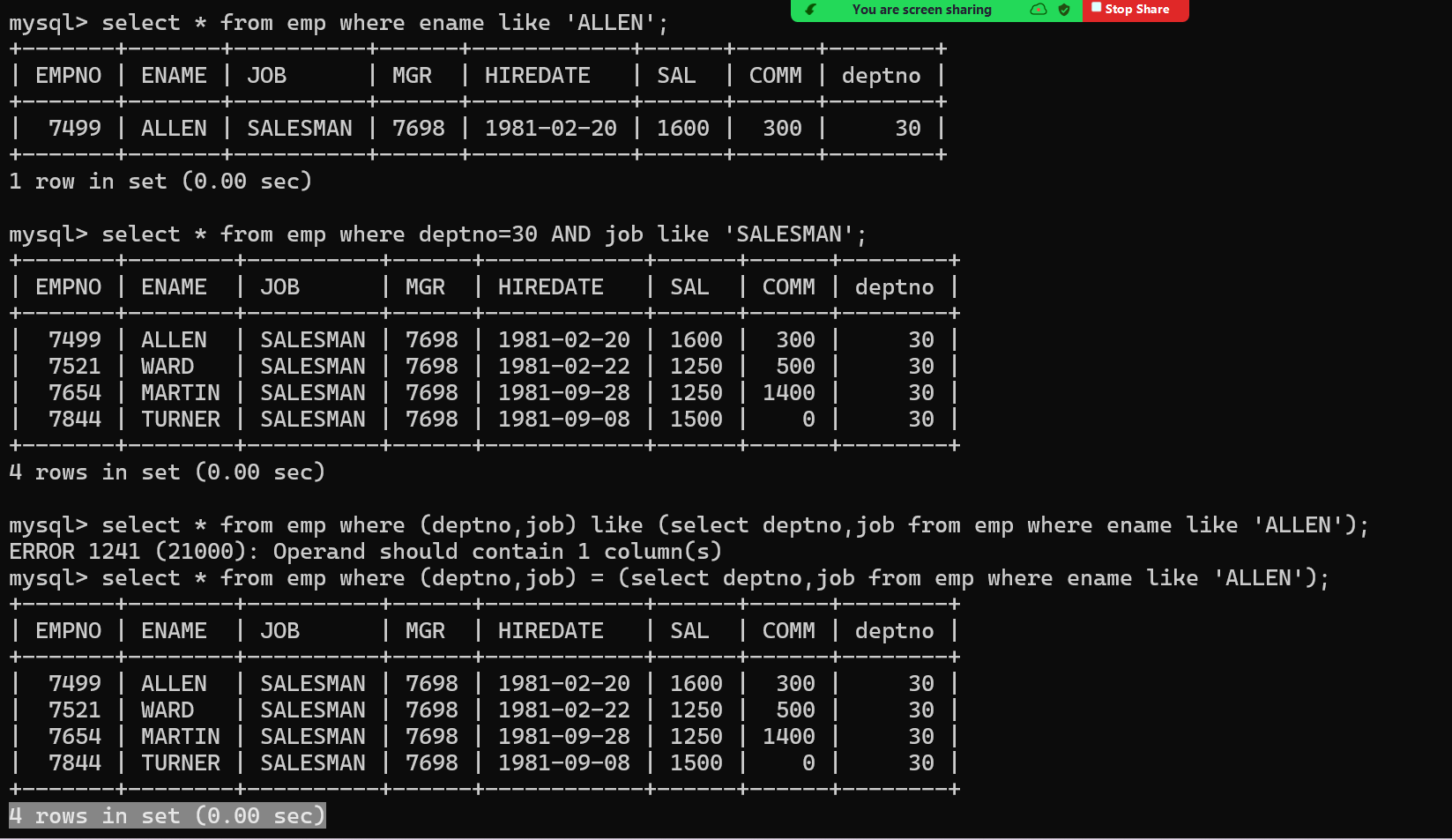
The above result showing that the Allen is working in deptno 30 as a sales man. Now our requirement is to retrieve all the salesman who are working in department number 30



We can address the above scenario using Co related sub queries.

The reason for using co related sub queries is: we have to check two conditions in order to retrieve the employees information:

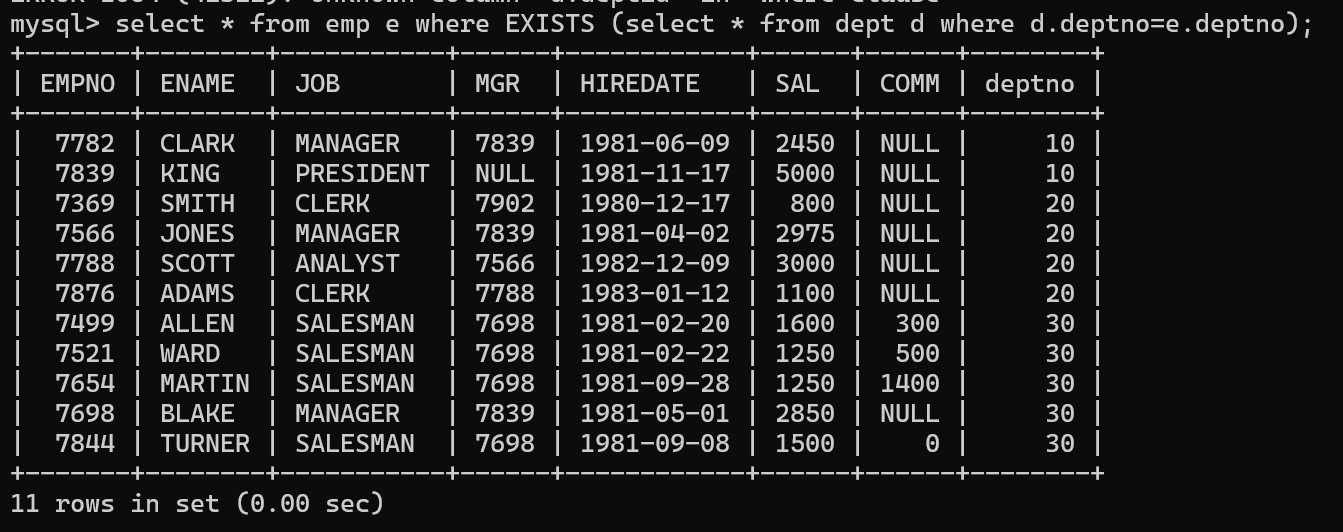
1. Deptno = 30 bcz Allen is working in dept 30
2. Job = salesman bcz Allen’s designation is Salesman



|  |  |
| --- | --- |
| **IN** | **EXISTS** |
| Works on result set list | Works on Virtual tables |
| the JOIN clause returns rows from another table | The EXISTS operator returns TRUE or FALSE |
| Doesn’t work on subqueries resulting in Virtual tables with multiple columns | Is used with co-related queries |
| Compares every value in the result list | Exists comparison when match is found |
| Performance is comparatively SLOW for larger result set of subquery | Performance is comparatively FAST for larger result set of subquery |

SQL Qeury to retrieve all the employees who are working in the dept which is present in the department table

select \* from emp e where EXISTS (select \* from dept d where d.deptno=e.deptno);



Nested Query 🡪 if we write a select in between select and from, then it is called Nested query

Sub query 🡪 if we write a select in the where clause of another query, then it is called Sub Queries

All the above are the examples of sub query bcz we wrote another query in the where clause of a select query

Example for Nested Query

Q) Retrieve empname and departname from employee and department table

select e.ename, (select dname from dept d where d.deptno=e.deptno) from emp e;

