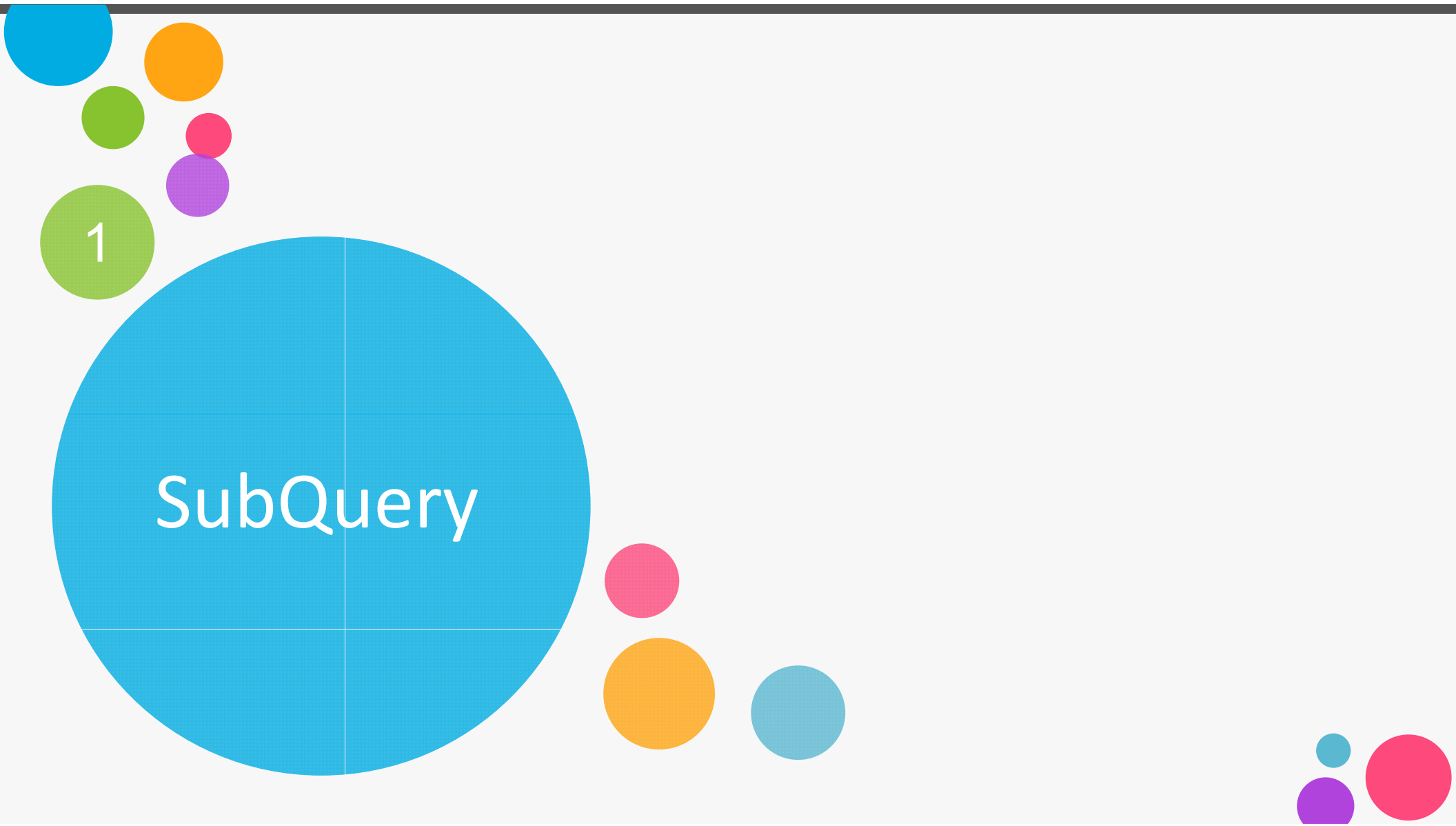
Sub queries and compound queries in SQL


1
SubQuery

Subqueries

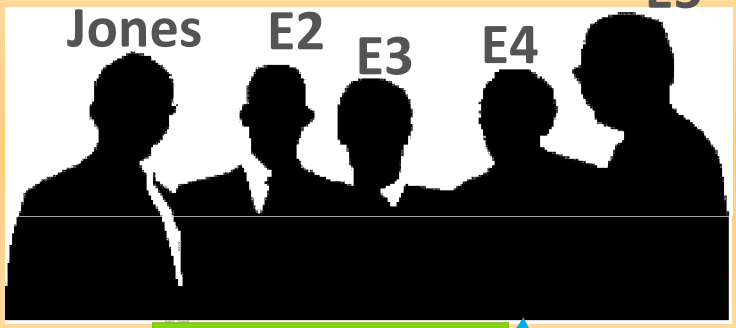
Why we use subqueries?

Example : display which employee have a salary greater than Jones salary.

Give me the record

of ALL Employees

Whose salary is greater than Jones salary



Main Query

Give me the record of jones salary

Sub Query

Subqueries

Can the output of above query be given without subquery?

Yes. We have database experts in our class who's prior knowledge tells us

```
SQL> ed
Wrote file afiedt.buf

  1  Select E2.ename from emp E1,emp E2
  2* where E2.sal > E1.sal and E1.empno=7566
SQL> /

ENAME
----------
SCOTT
KING
FORD
```

Select E2.ename from emp E1,emp E2

where E2.sal > E1.sal and E1.empno=7566

We have Printed all those employees of E2 whose salary is greater than E1's Employee Mr. Jones's salary via above self join query

But students this is a simple Scott's employee table. what if we will be retrieving records from a huge table or huge number of tables like we do in real time?




Subqueries

That is for what subquery has been designed for. we can write above query as:

```
SELECT ename FROM emp
WHERE sal > (SELECT sal FROM emp WHERE empno = 7566);
```

Main Query

subquery

```
  1  SELECT ename
  2  FROM emp
  3  WHERE sal >
  4  (SELECT sal
  5  FROM emp
  6* WHERE empno = 7566)
SQL> /

ENAME
----------
SCOTT
KING
FORD
```

Let see few more information regarding subquery:

The inner query or the subquery returns a value that is used by the outer query or the main query.

Using a subquery is equivalent to performing two sequential queries means using the result of the first query as the search value in the second query.

Subqueries

The subquery can be placed in a number of SQL clauses:

- ✓ WHERE clause // as we have seen above
- ✓ HAVING clause
- ✓ FROM clause // as we have seen in Q2 solution of lab 04 exercise solutions

Syntax of Select statement using Subquery

```
SELECT select_list
FROM table
WHERE expr operator
(SELECT select_list
FROM table);
```

Comparison operator

Single Row Operators (>, =, >=, <, <>, <=)

Comparison operator(multiple Row) IN, ANY,ALL

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery (OR)
ALL	Compare value to every value returned by the subquery (AND)

Subqueries

Example : To display the employee number whose salary is greater than clark,scott & Ford

```
SELECT empno, sal
FROM   emp
WHERE  sal > ALL(select SAL from emp
where ename='CLARK' or ename= 'SCOTT'
or ename = 'FORD')
```

All(A,B,C) = will have AND between comparator

Example : To display the employee number whose salary is greater than any of the Clark's, Scott's & Ford's salary

```
SELECT empno, sal
FROM   emp
WHERE  sal > ANY(select SAL from emp
where ename='CLARK' or ename= 'SCOTT'
or ename = 'FORD')
```

ANY(A,B,C) = will have ***OR*** between comparator
just like IN but in ***ANY &***
ALL must be preceded by ***=, !=, >, <, <=, >=***

Subqueries

Types of Subqueries

- ***Single-row subquery:*** Query that returns only one row from the inner SELECT statement.
- ***Multiple-row subquery:*** Query that returns more than one row form the inner SELECT statement.
- ***Multiple-column subquery:*** Query that returns more than one column from the inner
SELECT statement.

Subqueries

1. Single-row subquery Examples

Example 1: To display the employees whose job title is the same as that of employee 7369.

```
SELECT ename, job
FROM emp
WHERE job = (SELECT job FROM emp WHERE empno = 7369);
```

Smith is also part of result set

```
  1  SELECT ename, job
  2  FROM emp
  3* WHERE job = (SELECT job FROM emp WHERE empno = 7369)
  4  /

ENAME      JOB
---------- ---------
SMITH      CLERK
ADAMS      CLERK
JAMES      CLERK
MILLER     CLERK
```

Example 2: To display employees whose job title is the same as that of employee 7369 and whose salary is greater than that of employee 7876.

```
SELECT ename, job
FROM emp
WHERE job = (SELECT job FROM emp WHERE empno = 7369)
and sal > (SELECT sal FROM emp WHERE empno = 7876)
```

```
  1  SELECT ename, job
  2  FROM emp
  3  WHERE job = (SELECT job FROM emp WHERE empno = 7369)
  4* and sal > (SELECT sal FROM emp WHERE empno = 7876)
SQL> /

ENAME      JOB
---------- ---------
MILLER     CLERK
```

Subqueries

1. *Single-row subquery Examples (Cont)*

Example 3: display the employee name, job title and salary of all employees whose salary is equal to the minimum salary.

Hint: Use Group functions

```
SELECT ename, job, sal
FROM emp
WHERE sal = (SELECT MIN(sal) FROM emp);
```

```
  1  SELECT ename, job, sal
  2  FROM emp
  3* WHERE sal = (SELECT MIN(sal) FROM emp)
SQL> /

ENAME      JOB              SAL
---------- --------- ----------
SMITH      CLERK            800
```

Example 4 : display all departments that have a minimum salary greater than that of department 20's Minimum salary.

```
SELECT deptno, MIN(sal) FROM emp
GROUP BY deptno
HAVING MIN(sal) >(SELECT MIN(sal) FROM emp WHERE deptno = 20);
```

```
  1  SELECT deptno, MIN(sal) FROM emp
  2  GROUP BY deptno
  3* HAVING MIN(sal) >(SELECT MIN(sal) FROM emp WHERE deptno = 20)
SQL> /

    DEPTNO   MIN(SAL)
---------- ----------
        30        950
        10       1300
```

Subqueries

2. *Multiple Row Subquery Examples*

As we have already seen Multiple row operators in this session as:

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery (OR)
ALL	Compare value to every value returned by the subquery (AND)

Note: The NOT operator can be used with IN, ANY, and ALL operators.

Example 1: find out the department wise minimum salaries of all employees and please mention the name of those salary with drawers

Let see step by step answers of this query then u will came to knw why we proceed for subquery

```
SELECT empno,deptno,min(sal)
FROM emp
Group by deptno
```

```
  1  SELECT ename, MIN(sal), deptno
  2  FROM emp
  3* GROUP BY deptno
SQL> /
SELECT ename, MIN(sal), deptno
       *
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

Subqueries

2. Multiple Row Subquery (cont)

It means I can only print the columns with the aggregate function on which my data is grouped. But I need to print the ename its department and department's minimum salary not employee ,department & employee salary.

```
SELECT  ename,deptno,MIN(sal)
FROM emp
GROUP BY deptno,ename
```

ENAME	DEPTNO	MIN(SAL)
JONES	20	2975
WARD	30	1250
SCOTT	20	3000
KING	10	5000
JAMES	30	950
ALLEN	30	1600
MARTIN	30	1250
BLAKE	30	2850
FORD	20	3000
SMITH	20	800
ADAMS	20	1100
MILLER	10	1300
CLARK	10	2450
TURNER	30	1500

14 rows selected.

=

```
SELECT  ename,deptno,sal
FROM emp
```

```
  1  SELECT  ename,deptno,sal
  2* FROM emp
SQL> /
```

ENAME	DEPTNO	SAL
SMITH	20	800
ALLEN	30	1600
WARD	30	1250
JONES	20	2975
MARTIN	30	1250
BLAKE	30	2850
CLARK	10	2450
SCOTT	20	3000
KING	10	5000
TURNER	30	1500
ADAMS	20	1100
JAMES	30	950
FORD	20	3000
MILLER	10	1300

14 rows selected.

How will I going to Do that?

ANS: This task can be Performed by the Help of subquery

Subqueries

2. *Multiple Row Subquery (cont)*

```
  1  SELECT ename, deptno,sal
  2  FROM emp
  3* WHERE sal IN (SELECT MIN(sal) FROM emp GROUP BY deptno)
SQL> /

ENAME          DEPTNO        SAL
---------- ---------- ----------
SMITH              20        800
JAMES              30        950
MILLER             10       1300
```

```
SELECT ename,deptno, sal
FROM emp
WHERE sal IN (SELECT MIN(sal) FROM emp GROUP BY deptno)
```

TARGET ACHIEVED I have printed every department's min salary holder Ename

Example 2: is our company contain any such employees(rather than clerks) whose salary is less than any clerk?

```
SELECT empno, ename, job
FROM emp
WHERE sal < ANY (SELECT sal FROM emp WHERE job = 'CLERK')
AND JOB <> 'CLERK';
```

```
  1  SELECT empno, ename, job
  2  FROM emp
  3  WHERE sal < ANY (SELECT sal FROM emp WHERE job = 'CLERK')
  4* AND JOB <> 'CLERK'
SQL> /

     EMPNO ENAME      JOB
---------- ---------- ---------
      7521 WARD       SALESMAN
      7654 MARTIN     SALESMAN
```

Subqueries

2. *Multiple Row Subquery (cont)*

Example 3: To display employees whose salary is greater than the average salaries of all the departments.

```
SELECT empno, ename, job,sal
FROM emp
WHERE sal > ALL (SELECT avg(sal) FROM emp GROUP BY deptno)
```

```
  1  SELECT empno, ename, job,sal
  2  FROM emp
  3* WHERE sal > ALL (SELECT avg(sal) FROM emp GROUP BY deptno)
  4  /

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7566 JONES      MANAGER         2975
      7788 SCOTT      ANALYST         3000
      7902 FORD       ANALYST         3000
      7839 KING       PRESIDENT       5000
```

```
  1* SELECT deptno,avg(sal) FROM emp GROUP BY deptno
SQL> /

    DEPTNO   AVG(SAL)
---------- ----------
        30 1566.66667
        20       2175
        10 2916.66667
```

Subqueries

More than one condition in it

3. *Multiple Column Subquery*

If we want to compare two or more columns, we must write a compound WHERE clause using multiple logical operators. Multiple column sub queries enable us to combine duplicate WHERE conditions into a single WHERE clause.

Example 3: To display the name of all employees who have done their present job somewhere before in their career.

Any past experience of employee is present in Job History Table

```
SELECT ENAME
FROM EMP E,JOB_HISTORY H
WHERE E.empno = H.empno and E.job = H.job
```

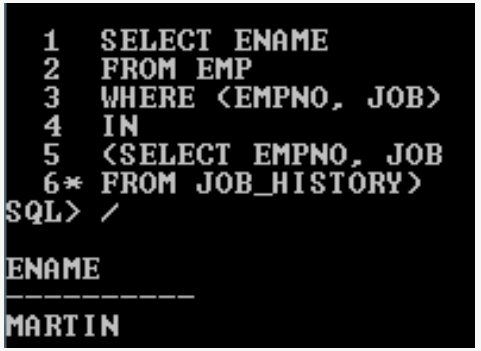
```
SQL> ed
Wrote file afiedt.buf

  1  SELECT ENAME
  2  FROM EMP E,JOB_HISTORY H
  3* WHERE E.empno = H.empno and E.job = H.job
SQL> /

ENAME
----------
MARTIN
```

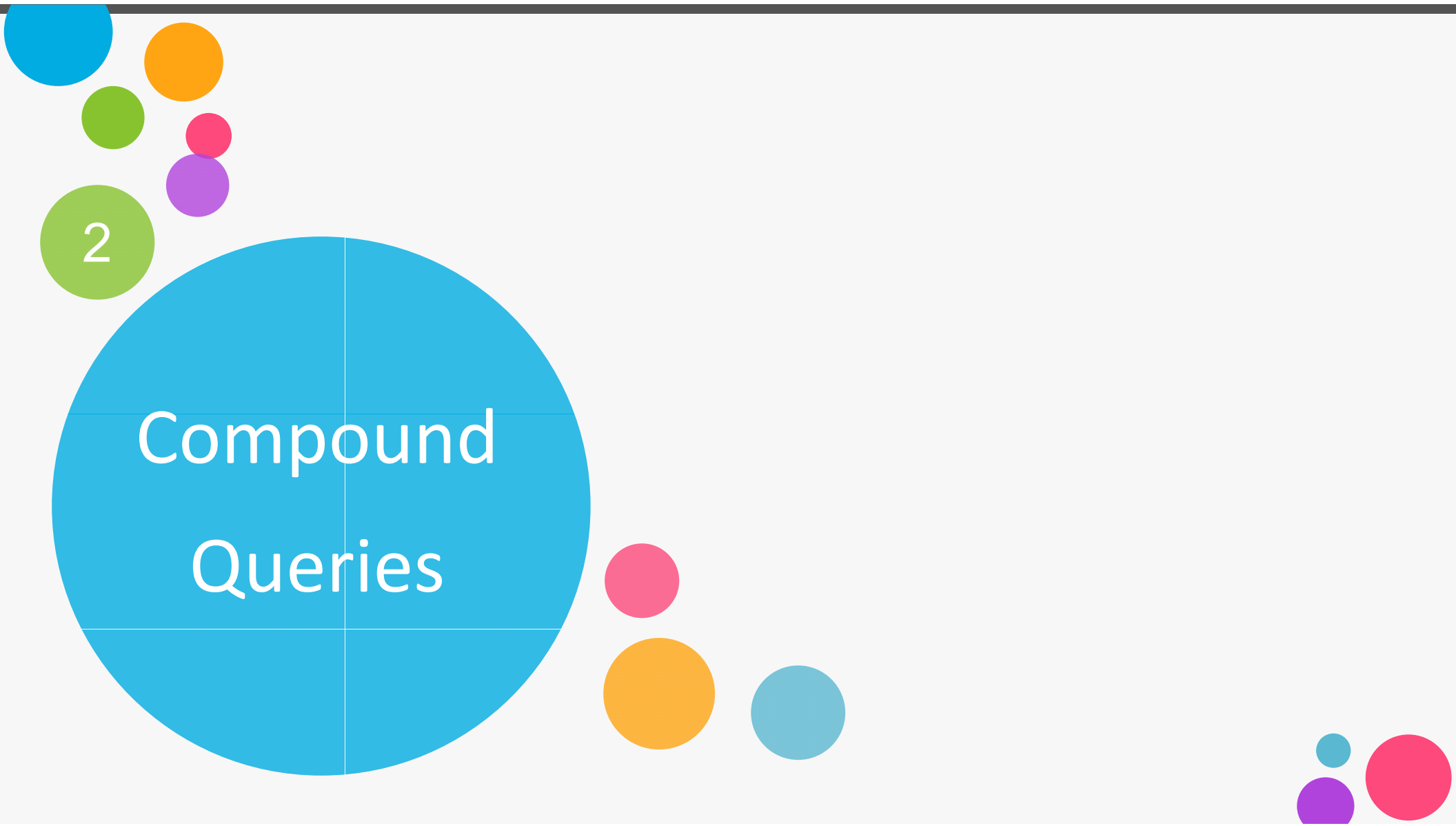
By using Multiple column subquery we can write above query as:

Subqueries

3. *Multiple Column Subquery*

```
SELECT ENAME
FROM EMP
WHERE (EMPNO, JOB) IN (SELECT EMPNO, JOB FROM JOB_HISTORY);
```




2
Compound
Queries

Compound Queries

We studied sets in mathematics so we have knowledge of Union, Intersection and Set Difference(-). Queries containing set operators known as compound queries. Let see different set operators provided in oracle SQL.

Operator	Returns
UNION	All distinct rows selected by either query
UNION ALL	All rows selected by either query including all duplicates
INTERSECT	All distinct rows selected by both queries
MINUS	All distinct rows that are selected by the first SELECT statement and

Restrictions on using Set operators?

If we are applying set operators then:

Both queries must have same sets of column & their corresponding column Contain the same datatype and lengths.

User must ensure values in the corresponding columns come from same domain

Compound Queries

1. Union Operators
    1. Union operator returns rows from both queries after eliminating duplicate rows.
    2. By default the output is sorted in the ascending order of the first column of the ***SELECT*** clause

Example 1: display all the jobs that each employee has performed in the company. (NOTE: If an employee has performed a job with a same designation multiple times, it will be shown only once)

```
SELECT EMPNO, JOB
FROM JOB_HISTORY
UNION
SELECT EMPNO, JOB
FROM EMP;
```

Compound Queries

1. Union Operators (cont.)

There are total 20 rows in output 14 from employee and 6 from job History one repeated row of ***MR Martin*** as ***Salesman*** has been eliminated in the output.

EMPNO	JOB
7369	CLERK
7499	SALESMAN
7521	SALESMAN
7566	MANAGER
7654	RECEPTIONIST
7654	SALESMAN
7698	ASSISTANT
7698	MANAGER
7782	MANAGER
7788	ANALYST
7788	PROGRAMMER
7839	ANALYST
7839	PRESIDENT
7844	SALESMAN
7876	CLERK
7876	OPERATOR
7876	TYPIST
7900	CLERK
7902	ANALYST
7934	CLERK

20 rows selected.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```
select * from job_history;
```

EMPNO	JOB	START DAT	END DATE
7698	ASSISTANT	04-MAR-80	30-APR-81
7654	RECEPTIONIST	13-JAN-80	09-SEP-80
7654	SALESMAN	10-SEP-80	20-SEP-81
7876	TYPIST	12-APR-80	13-NOV-81
7876	OPERATOR	15-NOV-81	11-JAN-83
7839	ANALYST	13-JUN-78	10-OCT-81
7788	PROGRAMMER	13-FEB-80	03-DEC-82

2. Union ALL Operators (cont.)

The UNION operator returns rows from both queries including all duplicates. So the above rejected row will be the part of our output with total 21 rows in the output

Compound Queries

2. Union ALL Operators (cont.)

```
  1  SELECT EMPNO, JOB
  2  FROM JOB_HISTORY
  3  UNION ALL
  4  SELECT EMPNO, JOB
  5* FROM EMP
SQL> /

     EMPNO JOB
---------- -----------------
      7698 ASSISTANT
      7654 RECEPTIONIST
      7654 SALESMAN
      7876 TYPIST
      7876 OPERATOR
      7839 ANALYST
      7788 PROGRAMMER
      7369 CLERK
      7499 SALESMAN
      7521 SALESMAN
      7566 MANAGER
      7654 SALESMAN
      7698 MANAGER
      7782 MANAGER
      7788 ANALYST
      7839 PRESIDENT
      7844 SALESMAN
      7876 CLERK
      7900 CLERK
      7902 ANALYST
      7934 CLERK

21 rows selected.
```

```
SELECT EMPNO, JOB
FROM JOB_HISTORY
UNION ALL
SELECT EMPNO, JOB
FROM EMP
```

It is not in order by default means first query all rows Then second query all rows unless we specify any order.

Compound Queries

3. INTERSECT Operator (cont.)

The INTERSECT operator returns all rows that are common to both queries.

Example 1: Display all employees and their jobs who have already performed their present job somewhere else in the past.

```
SELECT EMPNO, JOB
FROM JOB_HISTORY
INTERSECT
SELECT EMPNO, JOB
FROM EMP;
```

```
  1  SELECT EMPNO, JOB
  2  FROM EMP
  3  intersect
  4  SELECT EMPNO, JOB
  5* FROM JOB_HISTORY
SQL> /

     EMPNO JOB
---------- ------------------
      7654 SALESMAN
```

4. MINUS Operator(cont.)

The MINUS operator returns rows from the first query that is not present in the second query.

Compound Queries

Example 1: display the ID's of those employees whose present job is the first job in our company record

```
SELECT EMPNO, JOB
FROM EMP
MINUS
SELECT EMPNO, JOB
FROM JOB_HISTORY;
```

```
  1  SELECT EMPNO, JOB
  2  FROM EMP
  3  MINUS
  4  SELECT EMPNO, JOB
  5* FROM JOB_HISTORY
SQL> /

     EMPNO JOB
---------- ------------------
      7369 CLERK
      7499 SALESMAN
      7521 SALESMAN
      7566 MANAGER
      7698 MANAGER
      7782 MANAGER
      7788 ANALYST
      7839 PRESIDENT
      7844 SALESMAN
      7876 CLERK
      7900 CLERK
      7902 ANALYST
      7934 CLERK

13 rows selected.
```

Example 2: display the ID's of those employees who were past employees and are not the part of our company

```
SELECT EMPNO, JOB
FROM JOB_HISTORY
MINUS
SELECT EMPNO, JOB
FROM EMP;
```

```
6 rows selected.

SQL> ed
Wrote file afiedt.buf

  1  SELECT EMPNO, JOB
  2  FROM JOB_HISTORY
  3  MINUS
  4  SELECT EMPNO, JOB
  5* FROM EMP
SQL>
```

Note: We have seen that **First table-Second Table** And **Second Table – first table** don't have Same output.