SUBQUERIES TO SOLVE QUERIES QUESTIONS

1. Which of the following are the types of sub-queries?
   A. Ordered sub-queries
   B. Grouped sub-queries
   C. Single row sub-queries
   D. None of the above
   
   **Answer:** C. A subquery is a complete query nested in the SELECT, FROM, HAVING, or WHERE clause of another query. The subquery must be enclosed in parentheses and have a SELECT and a FROM clause, at a minimum. Single row sub-queries and multi-row sub-queries are the main types of sub-queries.

2. Which of the following is true about sub-queries?
   A. They execute after the main query executes
   B. They execute in parallel to the main query
   C. The user can execute the main query and then, if wanted, execute the sub-query
   D. They execute before the main query executes.
   
   **Answer:** D. The sub-query always executes before the execution of the main query. Subqueries are completed first. The result of the subquery is used as input for the outer query.

3. Which of the following is true about the result of a sub-query?
   A. The result of a sub-query is generally ignored when executed.
   B. The result of a sub-query doesn't give a result, it is just helpful in speeding up the main query execution
   C. The result of a sub-query is used by the main query.
   D. The result of a sub-query is always NULL
   
   **Answer:** C. Subqueries are completed first. The result of the subquery is used as input for the outer query.

4. Which of the following clause is mandatorily used in a sub-query?
   A. SELECT
   B. WHERE
   C. ORDER BY
   D. GROUP BY
   
   **Answer:** A. A sub-query is just like any other query which has to start with a SELECT clause. They are contained within an outer query.

5. Which of the following is a method for writing a sub-query in a main query?
   A. By using JOINS
   B. By using WHERE clause
   C. By using the GROUP BY clause
   D. By writing a SELECT statement embedded in the clause of another SELECT statement
   
   **Answer:** D. A subquery is a complete query nested in the SELECT, FROM, HAVING, or WHERE
clause of another query. The subquery must be enclosed in parentheses and have a SELECT and a FROM clause, at a minimum.

6. In the given scenarios, which one would appropriately justify the usage of sub-query?
   A. When we need to sum up values
   B. When we need to convert character values into date or number values
   C. When we need to select rows from a table with a condition that depends on the data from the same or different table.
   D. None of the above
   Answer: C.

7. In which of the following clauses can a sub-query be used?
   A. HAVING
   B. WHERE
   C. FROM
   D. All of the above
   Answer: D. A sub-query is not different from a normal query. It can make use of all the primary clauses of a SELECT statement.

8. Which of the following single-row operators can be used for writing a sub-query?
   A. >=
   B. <
   C. =
   D. All of the above
   Answer: D. Single-row operators include =, >, <, >=, <=, and <>.

9. Which of the following multi-row operators can be used with a sub-query?
   A. IN
   B. ANY
   C. ALL
   D. All of the above
   Answer: D. Multiple-row subqueries return more than one row of results. Operators that can be used with multiple-row subqueries include IN, ALL, ANY, and EXISTS.

10. What is true about the output obtained from a sub-query?
    A. It remains in the buffer cache
    B. It remains inside the sub-query and can be used later when needed
    C. It is used to complete the outer main query
    D. Both A and C
    Answer: C. Subqueries are completed first. The result of the subquery is used as input for the outer query.

11. You need to find the salaries for all the employees who have a higher salary than the Vice President of a company 'ABC'. Which of the following queries will give you the required result? Consider the table structure as given

    SQL> DESC employees
    Name       Null?    Type
    -------------- ------- --------------
**A.** In the option 'A', the inner sub-query gives the VP's salary as a result to the outer query.

**12. What among the following is true about sub-queries?**

A. Sub-queries can be written on either side of a comparison operator
B. Parenthesis is not mandatory for sub-queries
C. Single-row sub-queries can use multi-row operators but vice versa is not possible
D. All of the above

**Answer: A.** Sub queries can be placed on left or right hand side of the comparison operator depending on the query indentation and usability.

**13. What will be the outcome of the following query?** Consider the given table structure

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary ANY (SELECT salary FROM employees);
```
A. It executes successfully giving the desired results
B. It executes successfully but does not give the desired results
C. It throws an ORA error
D. It executes successfully and gives two values for each row obtained in the result set

**Answer: C.** Multi-row operators cannot be used in single-row sub-queries and vice versa.

14. **Which of the following is true about single-row sub-queries?**

A. They give one result from the main query
B. They give only one row in the result set
C. They return only one row from the inner SELECT statement
D. They give multiple rows from the main outer query

**Answer: C.** A single-row subquery can return a maximum of one value.

15. **What is true about multi-row sub-queries?**

A. They can return more than one column as the result of the inner query
B. They return multiple rows in the main query but only a single result set in the inner query
C. They return single row in the main query but multiple rows in the inner sub-query
D. They return more than one row from the inner SELECT statement

**Answer: D.** Multi-column sub-queries return more than one column in their result set, multi-row sub-queries return more than one row from the inner query.

16. **What among the following is true about single-row sub-queries?**

A. They return only one row
B. They use single-row operators
C. Both A and B
D. None of the above

**Answer: C.**

17. **Which of the following operators cannot be used in a sub-query?**

A. AND
B. <
C. >
D. <>

**Answer: A.** Single-row operators include =, >, <, >=, <=, and <>. Multi-row operators that can be used with multiple-row subqueries include IN, ALL, ANY, and EXISTS.

**Examine the exhibit and answer the questions 18 to 21 that follow.**

<table>
<thead>
<tr>
<th>EMPLOYEES</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP_ID</td>
<td></td>
<td>NOT NULL</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td></td>
<td>NUMBER(4)</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td>VARCHAR2(9)</td>
</tr>
<tr>
<td>MGR</td>
<td></td>
<td>VARCHAR2(4)</td>
</tr>
<tr>
<td>HIRE_DATE</td>
<td></td>
<td>DATE</td>
</tr>
</tbody>
</table>
18. You need to find out the names of all employees who belong to the same department as the employee 'Jessica Butcher' who is in department 100 and has an employee ID 40. Which of the following queries will be correct?

A. ```
SELECT first_name, last_name
FROM employees
WHERE last_name = 'Butcher'
And first_name = 'Jessica';
```

B. ```
SELECT first_name, last_name
FROM employees
WHERE department = 100;
```

C. ```
SELECT first_name, last_name
FROM employees
WHERE department = (SELECT department
FROM employees
WHERE first_name = 'Jessica'
AND last_name = 'Butcher');
```

D. ```
SELECT first_name, last_name
FROM employees
WHERE department = (SELECT department
FROM employees
WHERE first_name = 'Jessica'
AND last_name = 'Butcher'
AND department = 100
AND employee_id = 40);
```

**Answer: D.** 'D' is more appropriate than 'C' because it filters on employee id which is unique and ensures that the sub-query will return single row only. 'C' can fail if there are more than one employee with the same first and last name.

19. You need to find out the employees which belong to the department of 'Jessica Butcher' and have salary greater than the salary of 'Jessica Butcher' who has an employee ID of 40. Which of the following queries will work?

A. ```
SELECT first_name, last_name
FROM employees
WHERE last_name = 'Butcher'
AND first_name = 'Jessica'
AND salary > 10000;
```

B. ```
SELECT first_name, last_name
FROM employees
WHERE department = 100;
```

C. ```
SELECT first_name, last_name
FROM employees
WHERE department = (SELECT department
FROM employees
WHERE first_name = 'Jessica'
AND last_name = 'Butcher');
```

**Answer: D.**
AND employee_id = 40)
AND salary > (SELECT salary
FROM employees
WHERE first_name = 'Jessica'
AND last_name = 'Butcher'
AND employee_id = 40);

SELECT first_name, last_name
FROM employees
WHERE department = (SELECT department
FROM employees
WHERE first_name = 'Jessica'
AND last_name = 'Butcher'
AND department = 100);

**Answer:** C. More than one sub-query can be written in one SQL statement to add more than one condition.

**20. Based on the answers for questions 18th and 19th, what type of sub-queries is used by them?**

A. Single row sub-query  
B. Multiple row sub-query  
C. Both A and B  
D. Inline sub-query

**Answer:** A. The questions 18th and 19th given above demonstrate the usage sub-queries in a SELECT statement.

**21. Consider two statements about outer and inner queries in context of SQL sub-queries?**

i. The inner queries can get data from only one table  
ii. The inner queries can get data from more than one table

Which of the above statements are true?

A. i  
B. ii  
C. Both i and ii  
D. Neither i nor ii

**Answer:** B. Sub-queries can fetch data from more than one table.

**Examine the table structure as follows and answer the questions 22 to 27 that follow:**

```
SQL> DESC employees
Name   Null?  Type
--------  --------  ------------------------
EMPLOYEE_ID NOT NULL NUMBER(6)  
FIRST_NAME  VARCHAR2(20)  
LAST_NAME  NOT NULL VARCHAR2(25)  
EMAIL NOT NULL VARCHAR2(25)  
PHONE_NUMBER VARCHAR2(20)  
HIRE_DATE NOT NULL DATE  
JOB_ID  NOT NULL VARCHAR2(10)  
SALARY  NUMBER(8,2)  
COMMISSION_PCT  NUMBER(2,2)  
MANAGER_ID  NUMBER(6)  
DEPARTMENT_ID  NUMBER(4)  
```

**22. What will be the outcome of the following query?** *Choosethemostappropriateanswer*
SQL> DESC employees
Name      Null?  Type
----------  -------  -------------
EMPLOYEE_ID  NOT NULL NUMBER(6)
FIRST_NAME   VARCHAR2(20)
LAST_NAME    NOT NULL VARCHAR2(25)
EMAIL        NOT NULL VARCHAR2(25)
PHONE_NUMBER  VARCHAR2(20)
HIRE_DATE    NOT NULL DATE
JOB_ID       NOT NULL VARCHAR2(10)
SALARY       NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID   NUMBER(6)
DEPARTMENT_ID NUMBER(4)

SELECT last_name, job_id, salary
FROM employees
WHERE salary = (SELECT max(salary)
FROM employees);

A. It executes successfully and gives the employees who have salaries equal to the max salary.
B. It executes successfully but does not give the required results
C. It throws an error as a group function is used in the sub-query
D. It throws an error as a single row sub-query should contain a multi-row operator

Answer: A. A group function can be used within a sub-query.

23. What will be the outcome of the query that follows?

SELECT first_name, last_name, min(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) >
(SELECT min(salary)
FROM employees
WHERE department_id = 100);

A. It executes successfully and gives the names and minimum salary greater than department 100 of all employees
B. It executes successfully and gives the salaries of the employees in department 100
C. It executes successfully and gives the names and minimum salaries of all the employees.
D. It throws an error.

Answer: A. HAVING clause can be used in sub-queries as shown.

24. You need to find the job which has a maximum average salary. Which of the following queries will give you the required results?

SELECT job_id, avg(salary)
FROM employees
GROUP BY job_id;

SELECT job_id, avg(salary)
FROM employees
GROUP BY job_id
HAVING job_id in (SELECT max(avg(salary)) FROM employees);

SELECT job_id, avg(salary)
FROM employees
GROUP BY job_id
HAVING max(avg(salary)) in (SELECT max(avg(salary)) FROM employees);
SELECT job_id, avg(salary) 
FROM employees 
GROUP BY job_id 
HAVING avg(salary) in (SELECT max(avg(salary)) FROM employees GROUP BY job_id);

**Answer: D.** Sub-queries can make use of group functions and HAVING clause to restrict the groups.

25. The following query throws an error. Choose the correct reason for the error as given in the options.

```sql
SELECT first_name, last_name 
FROM employees 
WHERE commission_pct = (SELECT min(commission_pct) 
                        FROM employees 
                        GROUP BY department_id);
```

A. The GROUP BY clause is not required in the sub-query  
B. A function cannot be used in a sub-query SELECT statement  
C. The single row sub-query gives multiple records  
D. The use of "=" operator is invalid; an IN operator will work correctly

**Answer: C, D.** The GROUP BY clause gives the minimum commission_pct for each department and hence multiple results are fetched to the main query giving an error.

26. Consider the query given below. How many records will be returned as a result of the above query? Assuming thenoemployeeewithjobidXXexistsinthecompany

```sql
SELECT first_name, last_name 
FROM employees 
WHERE salary = (SELECT salary 
                FROM employees 
                WHERE job_id = 'XX');
```

A. 1  
B. NULL  
C. 0  
D. The query raises ORA error because sub-query is invalid.

**Answer: C.** Since there is no employee with job_id "XX" in the company, the sub-query returns no result, which when equated to job_id in the main query gives a 0.

27. What happens if the WHERE condition in the query given in question 26 is replaced with a new one `WHERE job_id IS NOT NULL`? Assuming the number of records in 'employees' table is 14.

A. 1  
B. 14  
C. 0  
D. ORA error

**Answer: D.** The query execution raises the exception "ORA-01427: single-row subquery returns more than one row".

28. Which of the following are valid multi row operators used for sub-queries?

A. <=  
B. ANY >=
Answer: B. Multiple-row subqueries return more than one row of results. Operators that can be used with multiple-row subqueries include IN, ALL, ANY, and EXISTS. The multi row operators IN, ANY, ALL must be used with single row operators as shown in the option B.

Examine the table structure as given. Consider the query given below and answer the questions 29 to 33 that follow

```sql
SQL> DESCR employees
Name Null? Type
----------------- -------------
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME  VARCHAR2(20)
LAST_NAME  NOT NULL VARCHAR2(25)
EMAIL      NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE  NOT NULL DATE
JOB_ID      NOT NULL VARCHAR2(10)
SALARY     NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)

SELECT first_name, last_name, salary, commission_pct
FROM employees
WHERE salary < ANY (SELECT salary
    FROM employees
    WHERE department_id = 100)
AND department_id <> 101;
```

29. What does the ANY operator evaluates to in the above query?

A. TRUE
B. FALSE
C. NULL
D. 0

Answer: A. The multi row operators return Boolean results. As there are results of salary in the department 100, it returns TRUE. If there are 0 results, it evaluates to FALSE.

30. What will be the outcome of the query if we assume that the department 100 has only one employee?

A. It executes successfully giving the one result
B. It executes successfully giving salaries of all the employees
C. NULL
D. It throws an ORA error

Answer: D. If the department 100 has one result `singlerowsub – query`, the < ANY operator gives the error as it is a multi-row operator.

31. What will be the outcome of the query given above if the < ANY operator is replaced with = ANY operator?

A. Oracle will treat each value of the salary returned from the sub-query as it does with IN operator
B. There will be no difference in the results
C. The results will differ

Answer: C. The results will differ.
The execution will throw an ORA error

**Answer:** **A.** = ANY operator is equivalent to IN operator.

32. What can be said about the < ANY operator in the query given above?

A. It gives the maximum value of salary  
B. It gives the minimum value of salary  
C. It means it gives the values that are lesser than the highest  
D. None of the above  

**Answer:** **C.** The multi row operator < ANY evaluates to the statements "Less than the maximum" of the subquery. ' > ALL' More than the highest value returned by the subquery. '< ALL' Less than the lowest value returned by the subquery. '< ANY' Less than the highest value returned by the subquery. '< ANY' More than the lowest value returned by the subquery. '= ANY' Equal to any value returned by the subquery such as IN. '[NOT] EXISTS' Row must match a value in the subquery

33. Assume that the < ANY operator is replaced with the > ANY. What is true about this operator?

A. It gives the maximum salary  
B. It finds only the maximum salary from the sub-query  
C. It gives more than the minimum salary  
D. It gives the minimum salary  

**Answer:** **C.** The multi row operator > ANY evaluates to the statements "Greater than the minimum" of the subquery. ' > ALL' More than the highest value returned by the subquery. '< ALL' Less than the lowest value returned by the subquery. '< ANY' Less than the highest value returned by the subquery. '>' ANY' More than the lowest value returned by the subquery. '=' ANY' Equal to any value returned by the subquery such as IN. '[NOT] EXISTS' Row must match a value in the subquery

34. Examine the given table structure and consider the following query:

```
SQL> DESC employees
Name Null? Type
--------- ---------- 
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME   VARCHAR2(20)
LAST_NAME    NOT NULL VARCHAR2(25)
EMAIL        NOT NULL VARCHAR2(25)
PHONE_NUMBER  VARCHAR2(20)
HIRE_DATE    NOT NULL DATE
JOB_ID       NOT NULL VARCHAR2(10)
SALARY       NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID   NUMBER(6)
DEPARTMENT_ID NUMBER(4)
```

```sql
SELECT employee_id, first_name, last_name
FROM employees
WHERE salary IN (SELECT max(salary)
                  FROM employees
                  GROUP BY department_id )
```

Which WHERE clause among the following is equivalent to that given in the above query?  
Assumethatthesalariesare2500, 3000, 3500, 4000

WHERE salary < ANY (SELECT max(salary)
FROM employees
GROUP BY department_id);

WHERE salary < ALL (SELECT max(salary)
FROM employees
GROUP BY department_id);

WHERE salary = (SELECT max(salary)
FROM employees
GROUP BY department_id);

WHERE salary IN (2500,3000,3500,4000);

**Answer: D.** When the IN operator is used, Oracle treats individual results of the sub-query as shown in the option D.

**Examine the structure of the EMPLOYEES table as given below and answer the questions 35 to 37 that follow.**

```
SQL> DESC employees
Name Null? Type
------+---------+----------+
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME VARCHAR2(20)
LAST_NAME NOT NULL VARCHAR2(25)
EMAIL NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE NOT NULL DATE
JOB_ID NOT NULL VARCHAR2(10)
SALARY NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)
```

35. You need to find out which of the employees have a salary less than that of the salary for the job ID 'FIN_ACT'. Which of the following queries will give you the required output?

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE salary < ALL
 (SELECT salary
 FROM employees
 WHERE job_id = 'FIN_ACT')
AND job_id <> 'FIN_ACT';
```

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE salary > ALL
 (SELECT salary
 FROM employees
 WHERE job_id = 'FIN_ACT')
AND job_id <> 'FIN_ACT';
```

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE salary < ANY
 (SELECT salary
 FROM employees
 WHERE job_id = 'FIN_ACT')
AND job_id <> 'FIN_ACT';
```
Answer: A. < ALL means less than the minimum. ‘> ALL' More than the highest value returned by the subquery. '< ALL' Less than the lowest value returned by the subquery. '< ANY' Less than the highest value returned by the subquery. '> ANY' More than the lowest value returned by the subquery. '=' ANY' Equal to any value returned by the subquery same as IN. '[NOT] EXISTS' Row must match a value in the subquery.

36. What will be the outcome of the above query the option A in the question above, if the < ALL is replaced with the > ALL?

A. It will execute successfully giving the same result.
B. It will throw an ORA error.
C. It will execute successfully but give the employees' details who have salaries lesser than all the employees with job_id 'FI_ACCOUNTANT'.
D. None of the above.

Answer: C. > ALL means less than the minimum. ‘> ALL' More than the highest value returned by the subquery. '< ALL' Less than the lowest value returned by the subquery. '< ANY' Less than the highest value returned by the subquery. '> ANY' More than the lowest value returned by the subquery. '=' ANY' Equal to any value returned by the subquery same as IN. '[NOT] EXISTS' Row must match a value in the subquery.

37. You need to find the salaries for all employees who are not in the department 100. Which of the following queries will give you the required result?

A. `SELECT employee_id, first_name, last_name FROM employees WHERE salary != ALL (SELECT salary FROM employees WHERE department_id = 100) AND department_id <> 100;`
B. `SELECT employee_id, first_name, last_name FROM employees WHERE salary NOT IN (SELECT salary FROM employees WHERE department_id = 100) AND department_id <> 100;`
C. `SELECT employee_id, first_name, last_name FROM employees WHERE salary NOT ALL (SELECT salary FROM employees WHERE department_id = 100) AND department_id <> 100;`
D. `SELECT employee_id, first_name, last_name FROM employees WHERE salary != (SELECT salary FROM employees WHERE department_id = 100) AND department_id <> 100;`

Answer: C. NOT can be used with the multi row operators IN, ANY and ALL.

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE salary = (SELECT salary
FROM employees
WHERE job_id = 'FIN_ACT')
AND job_id <> 'FIN_ACT';
```
Examine the table structure as given. Consider the following query and answer the questions 38 and 39 that follow. You need to find the employees who do not have a sub-ordinate reporting to them. 

Assumethereare0expectedresults

SQL> DESC employees
Name Null? Type
----------------------------------
EMPLOYEE_ID   NOT NULL NUMBER(6)
FIRST_NAME     VARCHAR2(20)
LAST_NAME   NOT NULL VARCHAR2(25)
EMAIL    NOT NULL VARCHAR2(25)
PHONE_NUMBER     VARCHAR2(20)
HIRE_DATE   NOT NULL DATE
JOB_ID    NOT NULL VARCHAR2(10)
SALARY      NUMBER(8,2)
COMMISSION_PCT     NUMBER(2,2)
MANAGER_ID     NUMBER(6)
DEPARTMENT_ID     NUMBER(4)

SELECT first_name, last_name
FROM employees
WHERE employee_id NOT IN
  (SELECT manager_id FROM employees);

38. What will be the result of the query given above?

A. 10
B. NULL
C. ORA error
D. 0

Answer: D. One of the values in the inner sub-query is NULL allemployeesarenotmanagers!

39. Which of the following WHERE clauses should be added / modified to the above query to give the expected results?

WHERE employee_id != (SELECT manager_id FROM employees);
WHERE employee_id IN (SELECT manager_id FROM employees);
WHERE employee_id <> ALL (SELECT manager_id FROM employees);
WHERE employee_id NOT IN (SELECT manager_id
  FROM employees
  WHERE manager_id is NOT NULL);

Answer: B, D. If the sub-query is likely to have NULL values, do not use the NOT IN operator or if using, modify the sub-query with an additional WHERE clause optionD

40. What is true about sub-queries in general?

A. Sub-queries have to be executed separately from the main queries
B. Sub-queries can be executed at the will of the user, they are not related to the main query execution
C. Sub-queries are equal to two sequential queries where the results of inner query are used by the main query
D. All of the above

Answer: C.
41. Which of the following is true about sub-queries?

A. A sub-query can return 0 or more rows  
B. A sub-query can be used only in the SELECT clause  
C. Nesting of sub-queries is limited to 2 levels  
D. Group functions cannot be used in sub-queries

**Answer: A.** A subquery is a complete query nested in the SELECT, FROM, HAVING, or WHERE clause of another query. The subquery must be enclosed in parentheses and have a SELECT and a FROM clause, at a minimum. A single-row subquery can return a maximum of one value. Multiple-column subqueries return more than one column to the outer query.

42. Examine the table structure as given.

```
SQL> DESC employees
Name       Null?     Type
---------- ---------------
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME  VARCHAR2(20)
LAST_NAME   NOT NULL VARCHAR2(25)
EMAIL       NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE   NOT NULL DATE
JOB_ID      NOT NULL VARCHAR2(10)
SALARY      NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID  NUMBER(6)
DEPARTMENT_ID NUMBER(4)
```

Consider the following query.

```
SELECT first_name, last_name
FROM employees
WHERE employee_id NOT IN
  (SELECT manager_id, hire_date
   FROM employees
   WHERE manager_id is not null);
```

This query returns an error. What is the reason for error?

A. The NOT IN operator used is invalid  
B. The WHERE clause in the sub-query is incorrectly written  
C. The column in the sub-query SELECT clause should only be one when there's an inequality used in the main query  
D. The sub-query uses the same table as the main query

**Answer: C.** The columns selected in the sub-query should be same as on the other side of comparison operator. Any inequality of data type or number of columns would result in an ORA error.

43. A report has to be extracted which displays all the departments that have one or more employees assigned to them. Which of the following queries will give the required output?

```
SQL> DESC employees
Name       Null?     Type
---------- ---------------
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME  VARCHAR2(20)
LAST_NAME   NOT NULL VARCHAR2(25)
EMAIL       NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
```

*Considerthetablestructureasgiven*
HIRE_DATE NOT NULL DATE  
JOB_ID NOT NULL VARCHAR2(10)  
SALARY NUMBER(8,2)  
COMMISSION_PCT NUMBER(2,2)  
MANAGER_ID NUMBER(6)  
DEPARTMENT_ID NUMBER(4) 

SELECT department_name 
FROM employees 
WHERE department_id IN (SELECT distinct (department_id ) 
FROM employees); 

SELECT department_name 
FROM employees 
WHERE department_id ANY (SELECT distinct (department_id ) 
FROM employees); 

SELECT department_name 
FROM employees 
WHERE department_id < ANY (SELECT distinct (department_id ) 
FROM employees); 

SELECT department_name 
FROM employees 
WHERE department_id = ANY (SELECT distinct (department_id ) 
FROM employees); 

Answer: A, D. 

44. What is the maximum level of sub-queries allowed in Oracle in a single SQL statement? 

A. 20  
B. 50  
C. Unlimited  
D. 255  

Answer: D. Oracle supports the Nesting of queries to 255 levels. 

45. What should be the best practice to follow when we know what values we need to pass on to the main query in Oracle queries? 

A. Using GROUP BY  
B. Using sub-queries  
C. Using HAVING  
D. None of the above  

Answer: D. It might become possible that the sub-queries give a NULL result, which results in 0 rows in the main result; hence it is a good practice to use them only if we know what values we need. 

Examine the table structure as given. Consider the following query and answer the questions 46 and 47 that follow: 

SQL> DESC employees 
Name Null? Type 
------------------ ------------------ 
EMPLOYEE_ID NOT NULL NUMBER(6)  
FIRST_NAME VARCHAR2(20)  
LAST_NAME NOT NULL VARCHAR2(25)  
EMAIL NOT NULL VARCHAR2(25) 

46. What is the maximum level of sub-queries allowed in Oracle in a single SQL statement? 

A. 20  
B. 50  
C. Unlimited  
D. 255  

Answer: D. Oracle supports the Nesting of queries to 255 levels. 

47. What should be the best practice to follow when we know what values we need to pass on to the main query in Oracle queries? 

A. Using GROUP BY  
B. Using sub-queries  
C. Using HAVING  
D. None of the above  

Answer: D. It might become possible that the sub-queries give a NULL result, which results in 0 rows in the main result; hence it is a good practice to use them only if we know what values we need.
46. You need to find all the employees whose job ID is the same as that of an employee with ID as 210. Which of the following WHERE clauses would you add / modify to achieve this result? (Consider the table structure as given)

A. `WHERE job_id = (SELECT job_id FROM employees WHERE employee_id = 210);`

B. `WHERE job_id IN (SELECT job_id FROM employees WHERE employee_id = 210);`

C. `WHERE job_id > (SELECT job_id FROM employees WHERE employee_id = 210);`

D. `WHERE job_id >= (SELECT job_id FROM employees WHERE employee_id = 210);`

Answer: A.

47. Assume that you change the WHERE clause as given in the option A in question 46 as the following.

`WHERE job_id = (SELECT job_id FROM employees WHERE employee_id < 210);`

What will be the outcome of this change?

A. The results will be the same
B. ORA error thrown on execution
C. The results will differ
D. The query will execute successfully giving 0 rows.

Answer: B. The sub-query gives more than one result on the given change and hence a multi row operator should replace the "=" in the main query given above.

48. Examine the table structures as shown in the exhibit below.

<table>
<thead>
<tr>
<th>Employees</th>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EMP_ID</td>
<td>NOT NULL</td>
<td>NUMBER(4)</td>
</tr>
<tr>
<td></td>
<td>FIRST_NAME</td>
<td>VARCHAR2(10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAST_NAME</td>
<td>VARCHAR2(10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JOB</td>
<td>VARCHAR2(9)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MGR</td>
<td>VARCHAR2(4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HIRE_DATE</td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SALARY</td>
<td>NOT NULL</td>
<td>NUMBER</td>
</tr>
<tr>
<td></td>
<td>COMM_PCT</td>
<td>NUMBER(7,2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEPT_ID</td>
<td>NUMBER(2)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grade</th>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GRADE</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOSAL</td>
<td>NUMBER</td>
<td></td>
</tr>
</tbody>
</table>
You need to display the names of the employees who have the highest salary. Which of the following SQL statements will be correct?

A. ```
SELECT first_name, last_name, grade
FROM employees, grade
WHERE (SELECT max(salary) FROM employees) BETWEEN losal and hisal;
```  

B. ```
SELECT first_name, last_name, grade
FROM employees, grade
WHERE (SELECT max(salary) FROM employees) BETWEEN losal and hisal
AND salary BETWEEN losal and hisal;
```  

C. ```
SELECT first_name, last_name, grade
FROM employees, grade
WHERE salary = (SELECT max(salary) FROM employees)
AND salary BETWEEN losal and hisal;
```  

D. ```
SELECT first_name, last_name, grade
FROM employees, grade
WHERE salary IN (SELECT max(salary) FROM employees)
AND max(salary) BETWEEN losal and hisal;
```  

**Answer: B, C.** The sub-queries can be written on either side of the operator.

49. **What is the sub-query in the FROM clause of an SQL statement?**

   A. Single row sub-query  
   B. Multi row sub-query  
   C. Inline View  
   D. Co-related sub-query

**Answer: C.** If a sub-query appears in the FROM clause of the SELECT statements, it forms an Inline view. Oracle internally creates a temporary view for the query execution.

50. **What is the maximum number of nesting level allowed in an Inline View type sub-query?**

   A. 255  
   B. 300  
   C. 216  
   D. Unlimited

**Answer: D.** As there is no limit on the number of tables which can be joined, there is no limit on the number of inline view in a query.

51. **What is true about co-related sub-queries?**

   A. The tables used in the main query are also used in a co-related sub-query  
   B. The sub-queries which reference a column used in the main query are called co-related sub-queries  
   C. The sub-queries which are written without parenthesis are called co-related sub-queries  
   D. The sub-queries which mandatorily use different tables than those used in the main query are called co-related sub-queries

**Answer: B.** Correlated subquery references a column in the outer query and executes the subquery once for every row in the outer query while Uncorrelated subquery executes the
Which of the following statements cannot be parent statements for a sub-query?

A. SELECT  
B. GROUP BY  
C. UPDATE  
D. DELETE  

Answer: B. The rest of the options can be in the main query parentquery of a sub-query.

What is true about a co-related sub-query?

A. It is evaluated only once for the parent query  
B. It is evaluated only thrice for the parent query  
C. It is evaluated once for each row processed by the parent sub-query  
D. All of the above  

Answer: C. Correlated subquery references a column in the outer query and executes the subquery once for every row in the outer query;and the EXISTS operator is used to test whether the relationship or link is present.

Examine the given table structure. You need to write a query which returns the names of the employees whose salaries exceed their respective department's average salary. Which of the following will work? Choosethemostappropriateanswer

```
SQL> DESC employees
Name Null? Type
----------------------- --------
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME VARCHAR2(20)
LAST_NAME NOT NULL VARCHAR2(25)
EMAIL NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE NOT NULL DATE
JOB_ID NOT NULL VARCHAR2(10)
SALARY NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)
```

```
SELECT employee_id, first_name, last_name
FROM employees e
WHERE salary > (SELECT avg (salary)
                FROM employees
                WHERE e.department_id = department_id )
ORDER BY department_id ;
```

```
SELECT employee_id, first_name, last_name
FROM employees e
WHERE salary > ANY (SELECT avg(salary)
                    FROM employees
                    WHERE e.department_id = department_id )
ORDER BY department_id ;
```

```
SELECT employee_id, first_name, last_name
FROM employees e
WHERE salary = (SELECT avg(salary)
                FROM employees
                WHERE e.department_id = department_id )
ORDER BY department_id ;
```
**Answer: A.** Here the department ID is obtained, used to evaluate the parent query and if the salary in that row is greater than the average salary of the departments of that row, that result is returned.

55. Examine the given table structure. Which of the following queries will display duplicate records in a table EMPLOYEES?

```sql
SQL> DESC employees
Name Null? Type
----------------------- ---------
EMPLOYEE_ID   NOT NULL NUMBER(6)
FIRST_NAME     VARCHAR2(20)
LAST_NAME   NOT NULL VARCHAR2(25)
EMAIL    NOT NULL VARCHAR2(25)
PHONE_NUMBER     VARCHAR2(20)
HIRE_DATE   NOT NULL DATE
JOB_ID    NOT NULL VARCHAR2(10)
SALARY      NUMBER(8,2)
COMMISSION_PCT     NUMBER(2,2)
MANAGER_ID     NUMBER(6)
DEPARTMENT_ID     NUMBER(4)
```

- **A.**
  ```sql
  SELECT *
  FROM employees E
  WHERE exists (SELECT 1 FROM employees E1
  WHERE E.employee_id = E1.employee_id);
  ```

- **B.**
  ```sql
  SELECT *
  FROM employees E
  WHERE exists (SELECT 1 FROM employees E1
  WHERE E.employee_id = E1.employee_id
  AND E.ROWID < E1.ROWID);
  ```

- **C.**
  ```sql
  SELECT *
  FROM employees E
  WHERE exists (SELECT 1 FROM employees E1
  WHERE E.ROWID < E1.ROWID);
  ```

- **D.**
  ```sql
  SELECT *
  FROM employees E
  WHERE = ANY (SELECT 1 FROM employees E1
  WHERE E.employee_id = E1.employee_id
  AND E.ROWID < E1.ROWID);
  ```

**Answer: A.** Correlated subquery references a column in the outer query and executes the subquery once for every row in the outer query; and the EXISTS operator is used to test whether the relationship or link is present. It can be used to find the duplicate rows in a table where duplicity is subjected to a column or set of columns.

Examine the structures for the tables DEPARTMENTS and EMPLOYEES and answer the questions 56 and 57 that follow.

SQL> DESC employees
Name Null? Type
----------------------- ---------
EMPLOYEE_ID   NOT NULL NUMBER(6)
FIRST_NAME     VARCHAR2(20)
LAST_NAME   NOT NULL VARCHAR2(25)
56. Which of the following queries will display the system date and count of records in the DEPARTMENTS and EMPLOYEES table?

A. `SELECT sysdate, 
   (SELECT * FROM departments) dept_count, 
   (SELECT * FROM employees) emp_count 
FROM DUAL;`

B. `SELECT sysdate, 
   (SELECT count(*) FROM departments) dept_count, 
   (SELECT count(*) FROM employees) emp_count 
FROM DUAL GROUP BY department_id;`

C. `SELECT sysdate, 
   (SELECT * FROM departments) dept_count, 
   (SELECT * FROM employees) emp_count 
FROM DUAL GROUP BY employee_id;`

D. `SELECT sysdate, 
   (SELECT count(*) FROM departments) dept_count, 
   (SELECT count(*) FROM employees) emp_count 
FROM DUAL;`

**Answer: D.** A single-row subquery can also be nested in the outer query's SELECT clause. In this case, the value the subquery returns is available for every row of output the outer query generates. Typically, this technique is used to perform calculations with a value produced from a subquery.

57. Which of the following queries will tell whether a given employee is a manager in a Company ‘XYZ’?

A. `SELECT employee_id, manager_id 
FROM employees A 
WHERE employee_id ANY (SELECT manager_id from employees B) 
ORDER BY manager_id desc;`

B. `SELECT employee_id, manager_id 
FROM employees A 
WHERE employee_id < ALL (SELECT manager_id from employees B) 
ORDER BY manager_id desc;`

C. `SELECT employee_id, manager_id 
FROM employees A 
WHERE employee_id IN (SELECT manager_id from employees B) 
ORDER BY manager_id desc;`
SELECT employee_id, manager_id
FROM employees A
WHERE employee_id in (SELECT manager_id from employees B)
GROUP BY department_id;

Answer: C.

Examine the exhibit and answer the question 58 that follows:

58. Which of the following queries will give you maximum salary of an employee in a particular city?

A. `SELECT max(salary), city
   FROM (SELECT salary, department_id, loc, city
         FROM employees natural join departments natural join locations);`

B. `SELECT salary, city
   FROM (SELECT salary, department_id, loc, city
         FROM employees natural join departments natural join locations);`

C. `SELECT max(salary), city
   FROM (SELECT salary, department_id, loc, city
         FROM employees natural join departments natural join locations)
   GROUP BY city;`

D. `SELECT max(avg(salary)), city
   FROM (SELECT salary, department_id, loc, city
         FROM employees natural join departments natural join locations);`

Answer: C. When a multiple-column subquery is used in the outer query's FROM clause, it creates a temporary table that can be referenced by other clauses of the outer query. This temporary table is more formally called an inline view. The subquery's results are treated like any other table in the FROM clause. If the temporary table contains grouped data, the grouped subsets are treated as separate rows of data in a table.
Examine the table structures as given below.

SQL> DESC employees
Name Null? Type
--------------------
EMPLOYEE_ID  NOT NULL NUMBER(6)
FIRST_NAME   VARCHAR2(20)
LAST_NAME    NOT NULL VARCHAR2(25)
EMAIL        NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE    NOT NULL DATE
JOB_ID       NOT NULL VARCHAR2(10)
SALARY       NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID   NUMBER(6)
DEPARTMENT_ID NUMBER(4)

SQL> DESC departments
Name Null? Type
--------------------
DEPARTMENT_ID  NOT NULL NUMBER(4)
DEPARTMENT_NAME NOT NULL VARCHAR2(30)
MANAGER_ID     NUMBER(6)
LOCATION_ID    NUMBER(4)

Consider the following query and answer the questions that 59 to 62 that follow.

SELECT  department_name
FROM departments d INNER JOIN employees e
ON (d.employee_id = e.employee_id)
GROUP BY department_name;

59. Which of the following queries can replace the above query by using sub-queries giving the same result?

A

B

C

D

Answer: A, B.

60. Assume that the sub-query as shown in the query given above is modified to the following.

(SELECT distinct (department_id ) FROM employees ORDER BY department_id );

What will be the outcome as a result of this change? Choose the most appropriate answer

A. It will order the department_id fetched from the sub-query and display them in ascending order
It will throw an ORA error as the ORDER BY clause should be accompanied by the GROUP BY clause.

C. It will throw an ORA error because an ORDER BY clause cannot be used inside a sub-query.

D. It will execute successfully.

Answer: C. A subquery, except one in the FROM clause, can't have an ORDER BY clause. If you need to display output in a specific order, include an ORDER BY clause as the outer query's last clause.

61. Assume that the query given above is modified as the below one.

```sql
SELECT department_name
FROM departments
WHERE department_id = ANY (SELECT department_id FROM employees)
ORDER BY department_id desc;
```

What will be the outcome as a result of this change?

A. It will order the department_id fetched from the sub-query and display them in ascending order
B. It will order the department_id fetched from the sub-query and display them in descending order
C. It will throw an ORA error because an ORDER BY clause cannot be used inside a sub-query
D. None of the above

Answer: D. A subquery, except one in the FROM clause, can't have an ORDER BY clause. If you need to display output in a specific order, include an ORDER BY clause as the outer query's last clause.

62. Which of the following can be used to order results in a sub-query?

A. ORDER BY
B. HAVING
C. GROUP BY
D. All of the above

Answer: C. By default, the GROUP BY clause performs ordering in a sub-query.

Examine the exhibit below and answer the questions 63 to 65 that follow:

<table>
<thead>
<tr>
<th>AUDIT</th>
<th>NULL?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Null?</td>
<td>Type</td>
</tr>
<tr>
<td>AU_ID</td>
<td>NOT NULL</td>
<td>NUMBER(10)</td>
</tr>
<tr>
<td>AU_TITLE</td>
<td>VARCHAR2(20)</td>
<td></td>
</tr>
<tr>
<td>AU DETAILS</td>
<td>CLOB</td>
<td></td>
</tr>
</tbody>
</table>

Consider the following query:

```sql
SELECT au_id, au_title
FROM audit
WHERE au_details in (SELECT au_details FROM audit
                       WHERE au_title like 'S%')
ORDER BY au_title;
```

63. What will be the outcome of the query given above?

A. It gives all AU_ID and AU_TITLEs starting with the letter 'S%'
B. It gives all AU_ID and AU_TITLES starting with the letter ‘S%’ ordered by the titles in ascending order

C. It throws an ORA error

D. It returns a 0 value

**Answer: C.** A column with a CLOB, BLOB, NCLOB or an ARRAY cannot be used in a sub-query.

### 64. What will be the outcome of the following query?

```
SELECT *
FROM employees
WHERE salary BETWEEN (SELECT max(salary)
                      FROM employees
                      WHERE department_id = 100)
AND (SELECT min(salary) FROM employees where department_id = 100);
```

This query returns an error. What is the reason for the error?

A. A GROUP BY clause should be used as the function MAX is used

B. Both the sub-queries cannot use the same department ID in the same outer query

C. BETWEEN operator cannot be used with a sub-query

D. SELECT clause should have columns mentioned and not a asterix *

**Answer: C.** The BETWEEN operator can be used within a sub-query but not with a sub-query.

### 65. What is true about using NOT IN when writing queries with sub-queries in them?

A. NOT IN ignores all the NULL values and gives only the NOT NULL values

B. NOT IN puts all the NULL values at the last and gives the NOT NULL to be displayed first

C. NOT IN should not be used if a NULL value is expected in the result set

D. NOT IN is just a negation of the operator IN and can be changed without any caveat.

**Answer: C.** SQL handles NULL values in a different way and hence it is a good practice to avoid NOT IN if the result set might contain a NULL.

**Consider the following table structures and answer the questions 66 to 72 that follow:**

<table>
<thead>
<tr>
<th>EMPLOYEES</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Null?</td>
<td>Type</td>
</tr>
<tr>
<td>EMP_ID</td>
<td>NOT NULL</td>
<td>NUMBER(4)</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td>VARCHAR2(9)</td>
</tr>
<tr>
<td>MGR</td>
<td></td>
<td>VARCHAR2(4)</td>
</tr>
<tr>
<td>HIRE_DATE</td>
<td></td>
<td>DATE</td>
</tr>
<tr>
<td>SALARY</td>
<td>NOT NULL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>COMM_PCT</td>
<td></td>
<td>NUMBER(7,2)</td>
</tr>
<tr>
<td>DEPT_ID</td>
<td></td>
<td>NUMBER(2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENTS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Null?</td>
<td>Type</td>
</tr>
<tr>
<td>DEPT_ID</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>DEPT_NAME</td>
<td></td>
<td>VARCHAR2(20)</td>
</tr>
<tr>
<td>DEPT_LOC</td>
<td></td>
<td>VARCHAR2(20)</td>
</tr>
</tbody>
</table>

66. You need to find out the names and IDs of the departments in which the least salary is greater than the highest salary in the department 10. Which of the following queries will give the required result.
SELECT department_id, min(salary)
FROM employees
GROUP BY department_id
HAVING min(salary) >

( select max(salary)
  FROM employees
  where department_id = 10
)

SELECT department_id, min(salary)
FROM employees
GROUP BY department_id
HAVING min(salary) > ANY

( select max(salary)
  FROM employees
)

SELECT department_id, min(salary)
FROM employees
HAVING max(salary) < ANY

( select min(salary)
  FROM employees
  where department_id = 10
)

SELECT department_id, min(salary)
FROM employees
GROUP BY department_id
HAVING min(salary) > ALL

( select max(salary)
  FROM employees
  where department_id = 10
)

Answer: A.

67. Write a query to find the employees whose salary is equal to the salary of at least one employee in department of id 10. Choose the best answer

SELECT employee_id, Salary
FROM employees
WHERE salary in

( SELECT salary
  FROM employees
  where department_id = 10
)

SELECT employee_id, Salary
FROM employees
WHERE salary = ANY

( SELECT salary
  FROM employees
  where department_id = 10
)

SELECT employee_id, Salary
FROM employees
WHERE salary ALL
68. You need to find out all the employees who have salary greater than at least one employee in the department 10. Which of the following queries will give you the required output?

A. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} \geq\text{ ANY (SELECT salary FROM employees where department\_id = 10) } \]

B. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} >\text{ ANY (SELECT salary FROM employees where department\_id = 10) } \]

C. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} <\text{ ANY (SELECT salary FROM employees where department\_id = 10) } \]

D. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} =\text{ ALL (SELECT salary FROM employees where department\_id = 10) } \]

Answer: A, B.

69. You need to find out all the employees who have salary lesser than the salary of all the employees in the department 10. Which of the following queries will give you the required output?

A. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} \geq\text{ ANY (SELECT salary FROM employees where department\_id = 10) } \]

B. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} >\text{ ANY (SELECT salary FROM employees where department\_id = 10) } \]

C. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} <\text{ ANY (SELECT salary FROM employees where department\_id = 10) } \]

D. \[ \text{SELECT employee\_id, Salary FROM employees WHERE salary} =\text{ ALL (SELECT salary FROM employees where department\_id = 10) } \]

Answer: B.
WHERE salary > ALL
( SELECT salary
FROM employees
where department_id  = 10
)

SELECT employee_id, Salary
FROM employees
WHERE salary = ALL
( SELECT salary
FROM employees
where department_id  = 10
)

SELECT employee_id, Salary
FROM employees
WHERE salary < ALL
( SELECT salary
FROM employees
where department_id  = 10
)

SELECT employee_id, Salary
FROM employees
WHERE salary < ANY
( SELECT salary
FROM employees
where department_id  = 10
)

Answer: C. Multiple-row subqueries return more than one row of results. Operators that can be used with multiple-row subqueries include IN, ALL, ANY, and EXISTS. Multiple-column subqueries return more than one column to the outer query. The columns of data are passed to the outer query in the same order in which they're listed in the subquery's SELECT clause.

70. You need to find out all the employees who have their manager and department matching with the employee having an Employee ID of 121 or 200. Which of the following queries will give you the required output?

SELECT employee_id, manager_id, department_id
FROM employees
WHERE (manager_id, department_id ) = ANY
( select manager_id,
    department_id
FROM employees
where employee_id in (121,200)
)

SELECT employee_id, manager_id, department_id
FROM employees
WHERE (manager_id, department_id ) < ANY
( select manager_id,
    department_id
FROM employees
where employee_id in (121,200)
)

SELECT employee_id, manager_id, department_id
FROM employees
WHERE (manager_id, department_id) > ANY
(
  select manager_id,
  department_id
  FROM employees
  where employee_id in (121, 200)
)

SELECT employee_id, manager_id, department_id
FROM employees
WHERE (manager_id, department_id) in
(
  select manager_id,
  department_id
  FROM employees
  where employee_id in (121, 200)
)

Answer: A, D. Multiple-row subqueries return more than one row of results. Operators that can be used with multiple-row subqueries include IN, ALL, ANY, and EXISTS. Multiple-column subqueries return more than one column to the outer query. The columns of data are passed to the outer query in the same order in which they're listed in the subquery's SELECT clause.

71. You need to find the department name of an employee with employee ID 200. Which of the following queries will be correct? Choose the most appropriate answer

Answer: C.

72. You need to find the highest earning employee with the job ID as 'SA_REP'. Which of the following queries will be correct? Choose the most appropriate answer

Answer: C.
(SELECT  distinct salary
FROM employees E1
WHERE E.job_id = E1.job_id
AND E.salary <= E1.salary
AND job_id = 'SA_REP'

SELECT department_id, employee_id, Salary
FROM employees E
WHERE 1 =
  (SELECT count(distinct salary)
   FROM employees E1
   WHERE E.job_id = E1.job_id
   AND E.salary <= E1.salary
   AND job_id = 'SA_REP'
  )

SELECT department_id, employee_id, Salary
FROM employees E
WHERE 0 =
  (SELECT count(distinct salary)
   FROM employees E1
   WHERE E.job_id = E1.job_id
   AND E.salary = E1.salary
   AND job_id = 'SA_REP'
  )

SELECT department_id, employee_id, Salary
FROM employees E
WHERE 1 =
  (SELECT salary
   FROM employees E1
   WHERE E.job_id < E1.job_id
   AND E.salary <= E1.salary
   AND job_id = 'SA_REP'
  )

Answer: B.

Consider the EMPLOYEES table structure as shown in the exhibit and answer the questions 73 to 77 that follow:

<table>
<thead>
<tr>
<th>EMPLOYEES</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Null?</td>
<td>Type</td>
</tr>
<tr>
<td>EMP_ID</td>
<td>NOT NULL</td>
<td>NUMBER(4)</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td>VARCHAR2(9)</td>
</tr>
<tr>
<td>MGR</td>
<td></td>
<td>VARCHAR2(4)</td>
</tr>
<tr>
<td>HIRE_DATE</td>
<td></td>
<td>DATE</td>
</tr>
<tr>
<td>SALARY</td>
<td>NOT NULL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>COMM_PCT</td>
<td></td>
<td>NUMBER(7,2)</td>
</tr>
<tr>
<td>DEPT_ID</td>
<td></td>
<td>NUMBER(2)</td>
</tr>
</tbody>
</table>

73. You need to find the job which has at least one employee in it. Which of the following queries will be correct? Choose the most appropriate answer

SELECT employee_id, Job_id
FROM employees E
WHERE exists
( SELECT 1
FROM employees E1
WHERE E.job_id = E1.job_id )

SELECT employee_id, Job_id
FROM employees E
WHERE exists
  ( SELECT *
    FROM employees E1
    WHERE E.job_id = E1.job_id )

SELECT employee_id, Job_id
FROM employees E
WHERE not exists
  ( SELECT *
    FROM employees E1
    WHERE E.job_id = E1.job_id )

SELECT employee_id, Job_id
FROM employees E
WHERE exists
  ( SELECT 1
    FROM employees E1
    WHERE E.job_id < E1.job_id )

Answer: A. The EXISTS operator is used to check and match records between queries. It returns a BOOLEAN value. Correlated subquery references a column in the outer query and executes the subquery once for every row in the outer query; and the EXISTS operator is used to test whether the relationship or link is present. An Uncorrelated subquery executes the subquery first and passes the value to the outer query.

74. You need to find the job which has no employees in it. Which of the following queries will be correct? Choose the most appropriate answer

SELECT employee_id, Job_id
FROM employees E
WHERE exists
  ( SELECT *
    FROM employees E1
    WHERE E.job_id = E1.job_id )

SELECT employee_id, Job_id
FROM employees E
WHERE not exists
  ( SELECT 1
    FROM employees E1
    WHERE E.job_id = E1.job_id )

SELECT employee_id, Job_id
FROM employees E
WHERE not exists
  ( SELECT *
    FROM employees E1
    WHERE E.job_id = E1.job_id )

SELECT employee_id, Job_id
FROM employees E
WHERE exists
  ( SELECT 1
    FROM employees E1
    WHERE E.job_id < E1.job_id )
75. You need to find the 3rd maximum salary from the EMPLOYEES table. Which of the following queries will give you the required results?

**Answer: D.**

```sql
SELECT * FROM employees E WHERE 3 = (SELECT count(distinct salary ) FROM employees WHERE e.salary <= salary ) ;
```

76. You need to find the maximum salary by using the user input for getting the value of N. Which of the following queries will give you the required results?

**Answer: D.**

```sql
SELECT salary FROM ( SELECT rownum as user_sal FROM (SELECT distinct salary FROM employees ORDER BY salary desc ) ) WHERE user_sal = &N ;
```
SELECT salary FROM
(SELECT max(rownum) as user_sal, salary FROM (SELECT distinct salary FROM employees ORDER BY salary desc)
WHERE user_sal=&N ;

Answer: C. ROWNUM is a pseudo column used for finding the nth order results.

77. What will happen if a value is provided to the &N variable in the above query optionCinquestion76 does not match with any row? Choose the best answer

A. The statement would throw an ORA error
B. The statement would return all the rows in the table
C. The statement would return NULL as the output result.
D. The statement would return no rows in the result.

Answer: D.

78. What is the maximum level up to which Sub-queries can be nested?

A. 255
B. 100
C. 2
D. 16

Answer: A.

79. What is true about the EXISTS operator in SQL queries with respect to sub-queries?

A. The columns selected in the sub-queries are important
B. The inner query's should return rows, any result is what is important, not what is SELECTED
C. Both A and B
D. Neither A nor B

Answer: B.

80. What is true about the ANY operator used for sub-queries?

A. Returns rows that match all the values in a list/sub-query
B. Returns rows that match the first 5 values in a list/sub-query
C. Returns rows that match any value in a list/sub-query
D. Returns the value 0 when all the rows match in a list/sub-query

Answer: C.

81. What is true about the ALL operator used for sub-queries? Choose the most appropriate answer.

A. Returns rows that match all the values in a list/sub-query
B. Returns rows that match only some values in a list/sub-query
C. Returns rows only if all the values match in a list/sub-query
D. All of the above

Answer: C. '>' ALL' More than the highest value returned by the subquery. '< ALL' Less than the lowest value returned by the subquery. '< ANY' Less than the highest value returned by the subquery. '>' ANY' More than the lowest value returned by the subquery. '=' ANY' Equal to any value returned by the subquery sameasIN. '[NOT] EXISTS' Row must match a value in the subquery.
82. What is true about using sub-queries in INSERT statements in Oracle?

A. They can be used in the INSERT clause without any restriction
B. They can be used in the INSERT clause only for Numeric values
C. The SELECT list of a sub-query should be the same as the column list of the INSERT statement.
D. None of the above

Answer: C.

Examine the table structures as given below and answer the questions 83 to 86 that follow.

```sql
SQL> DESC employees
Name  Null?  Type
------------------------ ---------------------
EMPLOYEE_ID   NOT NULL  NUMBER(6)
FIRST_NAME    VARCHAR2(20)
LAST_NAME     NOT NULL  VARCHAR2(25)
EMAIL         NOT NULL  VARCHAR2(25)
PHONE_NUMBER   VARCHAR2(20)
HIRE_DATE     NOT NULL  DATE
JOB_ID         NOT NULL  VARCHAR2(10)
SALARY         NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID     NUMBER(6)
DEPARTMENT_ID  NUMBER(4)
```

```sql
SQL> DESC departments
Name  Null?  Type
------------------------ ---------------------
DEPARTMENT_ID   NOT NULL  NUMBER(4)
DEPARTMENT_NAME NOT NULL  VARCHAR2(30)
MANAGER_ID      NUMBER(6)
LOCATION_ID     NUMBER(4)
```

83. You need to find the details of all employees who were hired for the job ID 'SA_REP' in the month of June, 2013. Which of the following queries will give the required results?

```sql
SELECT first_name
FROM employees
WHERE employee_id =
  ( SELECT employee_id
    FROM employees
    WHERE to_char(hiredate, 'MM/YYYY') = '02/1981'
    AND job_id = 'SA_REP'
  );
```

```sql
SELECT first_name
FROM employees
WHERE employee_id = ANY
  ( SELECT employee_id
    FROM employees
    WHERE to_char(hiredate, 'MM/YYYY') = '02/1981'
    AND job_id = 'SA_REP'
  );
```

```sql
SELECT first_name
FROM employees
WHERE employee_id  ANY
  ( SELECT employee_id
    FROM employees
  );
```
WHERE to_char(hiredate, 'MM/YYYY')= '02/1981'
AND job_id = 'SA_REP';

SELECT first_name
FROM employees
WHERE employee_id  exists
   ( SELECT employee_id
     FROM employees
     WHERE to_char(hiredate, 'MM/YYYY')= '02/1981'
     AND job_id = 'SA_REP'
   );

Answer: B.

84. Which of the following statements are equivalent?

A. SELECT employee_id, salary
   FROM employees
   WHERE salary < ALL (SELECT salary FROM employees WHERE department_id=100);

B. SELECT employee_id, salary
   FROM employees WHERE salary < (SELECT min(salary) FROM employees WHERE
department_id=100);

C. SELECT employee_id
   FROM employees
   WHERE salary not >= ANY (SELECT salary FROM employees WHERE department_id=100);

D. None of the above

Answer: A, B.

85. Consider the following two queries:

Query 1:
SELECT first_name
FROM employees e join departments d
ON e.department_id = d.department_id
WHERE department_name='ACCOUNTS';

Query 2:
SELECT first_name
FROM employees e
WHERE department_id = ANY (SELECT department_id FROM departments d
WHERE department_name='ACCOUNTS');

What can be said about the two statements?

A. Both the queries should generate the same result.
B. Both the queries will throw an error.
C. If there are two departments with the same name, both the queries will fail.
D. Both the queries will run successfully even if there is more than one department named 'ACCOUNTS'.

Answer: A, D.

86. You need to display all the employees who have the highest salary in a department 100. You fire a query as below.

SELECT E.first_name, E.last_name , E.salary
FROM employees E
WHERE E.salary > ALL (SELECT E1.salary 
FROM employees E1 
WHERE E.department_id = E1.department_id 
AND E.department_id = 100);

What will be the outcome of the above query?

A. It executes successfully and gives the required results
B. It executes successfully but doesn’t give the required output
C. It throws an ORA error on execution
D. It executes successfully and gives the required result when >ALL is replaced with >ALL

Answer: B, D. >ALL will not give the required result as there may be two employees with the same salary and who are the highest earners in the department 100

Consider table structures as shown in the exhibit and answer the questions 87 to 89 that follow:

<table>
<thead>
<tr>
<th>EMPLOYEES</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP_ID</td>
<td>NOT NULL</td>
<td>NUMBER(4)</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td></td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td>VARCHAR2(9)</td>
</tr>
<tr>
<td>MGR</td>
<td></td>
<td>VARCHAR2(4)</td>
</tr>
<tr>
<td>HIRE_DATE</td>
<td></td>
<td>DATE</td>
</tr>
<tr>
<td>SALARY</td>
<td>NOT NULL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>COMM_PCT</td>
<td></td>
<td>NUMBER(7.2)</td>
</tr>
<tr>
<td>DEPT_ID</td>
<td></td>
<td>NUMBER(2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENTS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEPT_ID</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>DEPT_NAME</td>
<td></td>
<td>VARCHAR2(20)</td>
</tr>
<tr>
<td>DEPT_LOC</td>
<td></td>
<td>VARCHAR2(20)</td>
</tr>
</tbody>
</table>

87. You need to fetch the first names in reverse alphabetical order of all the employees in the department ID = 100 and who have the maximum salary in the JOB ID = ‘SA_REP’. Which of the following queries will give the required output? Choose the most appropriate output

```sql
SELECT E.first_name, job_id, salary 
FROM employees E 
WHERE salary = 
  (SELECT max(salary) 
  FROM employees E1 
  WHERE E1.department_id = 100 
  GROUP BY job_id 
  ) 
AND job_id = 'SA_REP' 
ORDER BY first_name;
```

```sql
SELECT E.first_name, job_id, salary 
FROM employees E 
WHERE salary in 
  (SELECT max(salary) 
  FROM employees E1 
  where E1.department_id = 100) 
ORDER BY first_name;
```

```sql
SELECT E.first_name, job_id, salary 
FROM employees E
```
WHERE salary IN 
(SELECT max(salary) 
FROM employees E1 
where job_id = 'SA_REP' 
GROUP BY job_id ) 
AND WHERE E.department_id = 100 
ORDER BY first_name desc;

SELECT E.first_name, job_id , salary 
FROM employees E 
WHERE salary IN 
(SELECT max(salary) 
FROM employees E1 
WHERE E1.department_id = 100 
GROUP BY job_id ) 
ORDER BY first_name ;

Answer: C.

88.In the queries given above, you need to display all the employees with the JOB ID 'SA_REP' who have the maximum salary in the department 100. Which of the following queries will give the required output?

SELECT E.first_name, job_id , salary 
FROM employees E 
WHERE salary IN 
(SELECT max(salary) 
FROM employees E1 
WHERE E1.department_id = 100 
GROUP BY job_id ) 
AND job_id = 'SA_REP' 
ORDER BY first_name dec;

SELECT E.first_name, job_id , salary 
FROM employees E 
WHERE salary in 
(SELECT max(salary) 
FROM employees E1 
WHERE E1.department_id = 100) 
ORDER BY first_name;

SELECT E.first_name, job_id , salary 
FROM employees E 
WHERE salary in 
(SELECT max(salary) 
FROM employees E1 
WHERE job_id = 'SA_REP' 
GROUP BY job_id ) 
And WHERE E.department_id = 100 
ORDER BY first_name desc;

SELECT E.first_name, job_id , salary 
FROM employees E 
WHERE salary in 
(SELECT max(salary) 
FROM employees E1 
WHERE E1.department_id = 100 
GROUP BY job_id ) 
ORDER BY first_name ;

Answer: A.

89.Select the query which will give you the maximum salary and maximum comm percentage. The query should also give the maximum comm percentage paid if the highest salaried employee gets the maximum comm percentage.
A. SELECT employee_id, max(salary), max(commission_pct) FROM employees E GROUP BY salary, commission_pct;

B. SELECT employee_id, max(salary), max(commission_pct) FROM employees E GROUP BY salary;

C. SELECT employee_id, max(salary) FROM employees E GROUP BY salary, commission_pct HAVING max(commission_pct) = 100;

D. SELECT employee_id, (SELECT max(salary) FROM employees GROUP BY department_id) * (SELECT max(commission_pct) FROM employees) FROM DUAL;

**Answer: D.** A single-row subquery can also be nested in the outer query's SELECT clause. In this case, the value the subquery returns is available for every row of output the outer query generates. Typically, this technique is used to perform calculations with a value produced from a subquery.

**90. What is true about the sub-queries used in the SELECT clause of an SQL statement?**

A. These sub-queries are the same in all aspects as those used in the FROM or WHERE clauses
B. These sub-queries have to mandatorily be single row sub-queries
C. We can use multi row operators when writing such sub-queries
D. None of the above

**Answer: B.**

**91. What will be the outcome of the following query? Consider the table structure as given**

```
SQL> DESC employees
Name         Null? Type
----------------------- --------
EMPLOYEE_ID   NOT NULL NUMBER(6)
FIRST_NAME    NOT NULL VARCHAR2(20)
LAST_NAME     NOT NULL VARCHAR2(25)
EMAIL         NOT NULL VARCHAR2(25)
PHONE_NUMBER  VARCHAR2(20)
HIRE_DATE     NOT NULL DATE
JOB_ID        NOT NULL VARCHAR2(10)
SALARY        NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID    NUMBER(6)
DEPARTMENT_ID NUMBER(4)
```

```
SELECT sysdate, 
(SELECT max(salary) FROM employees GROUP BY department_id) 
FROM DUAL;
```

A. It gives the system date and the maximum salary for each department
B. It gives the maximum salary for all the departments
C. It throws an ORA error
D. It executes successfully with 0 rows

**Answer: C.** A Multi row sub-query cannot be used in the SELECT clause of an SQL statement. Only a single-row subquery can be nested in the outer query's SELECT clause.

**Examine the given table structure. Consider the following query and answer the**
SELECT salary
FROM employees
WHERE salary > ALL (10, 20, 30);

92. Which of the following queries are equivalent to the above query?

A. SELECT salary
   FROM employees
   WHERE salary > 10 or salary > 20 and salary > 30;

B. SELECT salary
   FROM employees
   WHERE salary < 10 and salary < 20 and salary < 30;

C. SELECT salary
   FROM employees
   WHERE salary > 10 and salary > 20 and salary > 30;

D. SELECT salary
   FROM employees
   WHERE salary > 10 and salary > 20 or salary < 30;

Answer: C. The question shows the ALL clause in a simplified manner when it is followed by a list.

93. If in the above query the list 10, 20, 30 is replaced by a sub-query, which of the following queries will give the required output for the department number 100?

A. SELECT E.salary
   FROM employees E
   WHERE E.salary > (SELECT E1.salary
                      FROM employees E1
                      WHERE E1.department_id = 100);

B. SELECT E.salary
   FROM employees E
   WHERE E.salary > ALL (SELECT E1.salary
                          FROM employees E1
                          WHERE E1.department_id = 100);

C. SELECT E.salary
   FROM employees E
   WHERE E.salary = (SELECT E1.salary
                      FROM employees E1
                      WHERE E1.department_id = 100);

D. SELECT E.salary
   FROM employees E
   WHERE E.salary

Answer: A. The question involves a sub-query to replace the list with a condition based on the department number 100.
FROM employees E
WHERE E.salary >= (SELECT E1.salary
FROM employees E1
WHERE E1.department_id = 100);

**Answer: B.** The question shows the ALL clause in a simplified manner when it is followed by a sub-query.

**94.** With respect to the question 14 above, what among the following will be an equivalent query if ALL has to be replaced with ANY?

<table>
<thead>
<tr>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT E.salary</td>
</tr>
<tr>
<td>FROM employees E</td>
</tr>
<tr>
<td>WHERE NOT EXISTS (E.salary = ANY (SELECT E1.salary</td>
</tr>
<tr>
<td>FROM employees E1</td>
</tr>
<tr>
<td>WHERE E1.department_id = 100);</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SELECT E.salary</td>
</tr>
<tr>
<td>FROM employees E</td>
</tr>
<tr>
<td>WHERE E.salary &gt; ANY (SELECT E1.salary</td>
</tr>
<tr>
<td>FROM employees E1</td>
</tr>
<tr>
<td>WHERE E1.department_id = 100);</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SELECT E.salary</td>
</tr>
<tr>
<td>FROM employees E</td>
</tr>
<tr>
<td>WHERE E.salary = ANY (SELECT E1.salary</td>
</tr>
<tr>
<td>FROM employees E1</td>
</tr>
<tr>
<td>WHERE E1.department_id = 100);</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SELECT E.salary</td>
</tr>
<tr>
<td>FROM employees E</td>
</tr>
<tr>
<td>WHERE NOT ( E.salary &lt;= ANY (SELECT E1.salary</td>
</tr>
<tr>
<td>FROM employees E1</td>
</tr>
<tr>
<td>WHERE E1.department_id = 100)));</td>
</tr>
</tbody>
</table>

**Answer: D.** The NOT operator used while using '<= ANY' is used for negation of the results returned by the sub-query.

**95.** With respect to the question 94, if the operator ANY is not to be used, which of the following queries will be correct?

<table>
<thead>
<tr>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT E.salary</td>
</tr>
<tr>
<td>FROM employees E</td>
</tr>
<tr>
<td>WHERE NOT EXISTS (E.salary = ANY (SELECT E1.salary</td>
</tr>
<tr>
<td>FROM employees E1</td>
</tr>
<tr>
<td>WHERE E1.department_id = 100));</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SELECT E.salary</td>
</tr>
<tr>
<td>FROM employees E</td>
</tr>
<tr>
<td>WHERE NOT EXISTS (SELECT E1.salary</td>
</tr>
<tr>
<td>FROM employees E1</td>
</tr>
<tr>
<td>WHERE E1.department_id = 100 And E.salary &lt;= E1.salary);</td>
</tr>
</tbody>
</table>

C. Either A or B

D. None of the above

**Answer: B.** Correlated subquery references a column in the outer query and executes the subquery once for every row in the outer query; and the EXISTS operator is used to test whether the relationship or link is present. An Uncorrelated subquery executes the subquery first and passes the value to the outer query.

**Examine the given table structures. Consider the following query and answer the questions 96 to 98 that follow:**
SELECT salary
FROM employees
WHERE salary > ANY (10, 20, 30);

96. Which of the following queries are equivalent to the above query?

A. SELECT salary
   FROM employees
   WHERE salary >10 or salary > 20 and or >30;

B. SELECT salary
   FROM employees
   WHERE salary <10 and salary < 20 and salary <30;

C. SELECT salary
   FROM employees
   WHERE salary >10 and salary > 20 or salary >30;

D. SELECT salary
   FROM employees
   WHERE salary >10 and salary > 20 or salary < 30;

Answer: A. The question shows the ANY clause in a simplified manner when it is followed by a list.

97. In the above query, if the list 10, 20, 30 is replaced by a sub-query, which of the following queries will give the required output for the department number 100?

A. SELECT E.salary
   FROM employees E
   WHERE E.salary > (SELECT E1.salary
                     FROM employees E1
                     WHERE E1.department_id = 100);

B. SELECT E.salary
   FROM employees E
   WHERE E.salary >ANY (SELECT E1.salary
                         FROM employees E1
                         WHERE E1.department_id = 100);

C. SELECT E.salary
   FROM employees E
   WHERE E.salary = (SELECT E1.salary
                     FROM employees E1
                     WHERE E1.department_id = 100);

D. SELECT E.salary
   FROM employees E
   WHERE E.salary < (SELECT E1.salary
                     FROM employees E1
                     WHERE E1.department_id = 100);
WHERE E.salary >= (SELECT E1.salary
FROM employees E1
WHERE E1.department_id = 100);

**Answer: B.** The question shows the ANY clause in a simplified manner when it is followed by a sub-query.

**98.** With respect to the question 97 above, what among the following will be an equivalent query if ANY is removed?

- **A.** SELECT E.salary
  FROM employees E
  WHERE NOT EXISTS (E.salary = ANY (SELECT E1.salary
  FROM employees E1
  WHERE E1.department_id = 100));

- **B.** SELECT E.salary
  FROM employees E
  WHERE EXISTS (SELECT E1.salary
  FROM employees E1
  WHERE E1.department_id = 100
  And E.salary >E1.salary));

- **C.** SELECT E.salary
  FROM employees E
  WHERE EXISTS (SELECT E1.salary
  FROM employees E1
  WHERE E1.department_id = 100 );

- **D.** SELECT E.salary
  FROM employees E
  WHERE IN (SELECT E1.salary
  FROM employees E1
  WHERE E1.department_id = 100);

**Answer: B.** The EXISTS operator can substitute the ANY operator. Correlated subquery references a column in the outer query and executes the subquery once for every row in the outer query; and the EXISTS operator is used to test whether the relationship or link is present.

**99.** Examine the given table structure. How many rows will get generated if the sub-query mentioned returns 0 rows?

```
SQL> DESC employees
Name Null? Type
----------------------------- -------- ----------------
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME VARCHAR2(20)
LAST_NAME NOT NULL VARCHAR2(25)
EMAIL NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE NOT NULL DATE
JOB_ID NOT NULL VARCHAR2(10)
SALARY NUMBER(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)
```

SELECT E.salary
FROM employees E
WHERE E.salary > ANY ( select E1.salary FROM employees E1 where E1.department_id = 100);

**A.** 1 row
B. No rows
C. Either A or B
D. None of the above

**Answer:** B. If the sub-query returns zero rows, the ' > ANY' condition evaluates to FALSE, hence "No rows" are returned.

**100. A subquery must be placed in the outer query's HAVING clause if:**

A. The inner query needs to reference the value returned to the outer query.
B. The value returned by the inner query is to be compared to grouped data in the outer query.
C. The subquery returns more than one value to the outer query.
D. None of the above. Subqueries can't be used in the outer query's HAVING clause.

**Answer:** B. A HAVING clause is used when the group results of a query need to be restricted based on some condition. If a subquery's result must be compared with a group function, you must nest the inner query in the outer query's HAVING clause.