GET DATA FROM MULTIPLE TABLES QUESTIONS

http://www.tutorialspoint.com/sql_certificate/get_data_from_multiple_tables_questions.htm

1.Which of the following is not related to a Relational Database?

A. Selection
B. Projection
C. Joining
D. None of the above

Answer: D. The options A, B and C are the major capabilities of the Oracle Relational Database.

2.Which of the following methods is used for writing a query with columns from multiple tables?

A. SELECT
B. GROUP BY
C. ORDER BY
D. JOINS

Answer: D. Joins are used to connect multiple tables and project column data from multiple tables in Oracle.

3.Which of the following is one of the most common methods to join multiple tables?

A. Hash Join
B. Equijoin
C. Self Join
D. Cross Join

Answer: B. Equijoin is one of the types of joins which is the most common and simple technique for joining more than one tables. Equijoins are also called simple joins or inner joins.Equijoin involve primary key and foreign key.

4.Which of following will be used to join rows with other tables if the column values fall in a range defined by inequality operators?

A. Equijoin
B. Simple join
C. Non-equijoin
D. None of the above

Answer: C. Equijoins use equality operators to join rows, non-equijoins use inequality operators.

5.Which of the following statements is true about Oracle joins?

A. NULL values are included in the result set
B. Only the rows which have matching conditions are fetched
C. All the rows that are present in any of the one tables are fetched
D. None of the above

Answer: B. NULL values and differing entries in common join columns are excluded when joins are used.

6.Which of the following can be used to join the rows of a table with other rows of the same table?

A. Equijoin

B. Non-equijoin

C. Outer join

D. Self-join

Answer: D. The association is based on columns with logical and usually hierarchical relationships with each other.

7.What is true about a cartesian join of two tables in Oracle DB?

A. It must be avoided as it is costly and non optimized

B. It is formed when every row from one table is joined with all rows in the second table

C. Both A and B

D. None of the above

Answer: B. Cartesian join is often the result of missing or inadequate join conditions.It is simply the cross product of two tables.

8.Which of the following is one of the basic types of joins in Oracle DB ?

A. Outer join

B. Self-join

C. Equi-join

D. All of the above

Answer: C. Equi-join and non-equijoin are the two basic types of joins in Oracle DB.

9.What is the main condition for using joins between a source table and a target table in Oracle DB for getting a non-cartesian product result?

A. There is no condition

B. At least one of the columns in both the tables should be common.

C. The names of the columns in both the joining tables should be the same for using joins

D. None of the above

Answer: B. The tables must be connected through a common column relating two entities.The table joined on a common column produce non Cartesian product.

10. Which of the following can be used to fetch rows from multiple tables in a single SQL query?

A. SELECT

B. WHERE

C. FROM

D. Equi-joins

Answer: D. Equijoins are also called simple joins or inner joins. Equijoin involve primary key and foreign key.

11.What is true about the source table and the target table in terms of Oracle Joins?

A. They must have atleast one column of same name

B. All the columns should be of the same name and same data type for joining the two tables

C. The source and the target tables cannot be swapped and are position specific

D. None of the above

Answer: D. The source and the target tables can be swapped and are not fixed at their positions.Depending of the type of join used in the query, the result may differ or remain same.

12.What is true about Natural joins in Oracle DB?

A. The column names of the source and the target tables should be identical

B. If the column names of the source and the target tables are not same, Oracle implicitly does the needful

C. NATURAL JOINS, USING and ON are the keywords associated with Natural Joins

D. All of the above

Answer: C. The keyword NATURAL JOIN instruct Oracle to identify columns with identical names between source and target tables. Natural joins use all columns with matching names and data types to join the tables. The USING clause can be used to specify only those columns that should be used for an equijoin.

13.Assume the tables EMPLOYEES and DEPARTMENT have to be joined using NATURAL JOIN. What is the difference between the following two queries which follow? *Considerthetablestructuresasgiven*

```
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)
```

```
SELECT department_id
FROM employees NATURAL JOIN department
WHERE first_name = 'John'
AND last_name = 'Bacon';
```

```
SELECT department_id
FROM department NATURAL JOIN employees
WHERE first_name = 'John'
AND last_name = 'Bacon';
```

A. There is no difference

B. The result is different in both the cases

C. Both the queries will give an ORA error on execution

D. None of the above

Answer: B. The source and target tables can be swapped while using NATURAL JOIN giving relationally different result sets.

14.Which of the following options is true regarding the NATURAL JOIN in Oracle DB?

A. While using NATURAL JOIN mentioning the names of all the columns from both the tables is mandatory

B. NATURAL JOIN can be used only if the names of all the columns of both the tables are identical

C. The join in NATURAL JOIN happens only when the user specifies the columns of the source and the target tables.

D. There is no need to mention the columns when using NATURAL JOINS.

Answer: D. There's an implicit joining of the columns from the source and the target tables when a NATURAL JOIN is used. A NATURAL JOIN is a JOIN operation that creates an implicit join clause for you based on the common columns in the two tables being joined.Common columns are columns that have the same name in both tables.

15. What is the difference between a NATURAL JOIN and a join with JOIN..ON?

A. There is no difference between the both

B. JOIN..ON joins the source and target tables on specific columns with the same name

C. NATURAL JOIN implicitly joins all the matching columns from the source and target tables

D. None of the above

Answer: B, C. The join condition for the natural join is basically an equijoin of all columns with the same name.Use the ON clause to specify arbitrary conditions or specify columns to join.The join condition is separated from other search conditions. The ON clause makes code easy to understand.

16.What is true about the JOIN..ON clause in Oracle DB?

A. It does not depend on the columns in the source and target tables having identical names

B. Only those columns from the source and the target tables which have identical names can be used with this clause

C. It is a format of the NATURAL JOIN

D. All of the above

Answer: A, C. The join condition for the natural join is basically an equijoin of all columns with the same name. Use the ON clause to specify arbitrary conditions or specify columns to join. The join condition is separated from other search conditions. The ON clause makes code easy to understand.

17. The database designer has named the key *unique* columns from two tables differently.While joining these tables, which among the following will be the best practice?

A. JOIN..ON

B. Either NATURAL JOIN or JOIN...ON clauses

C. Both A and B

D. None of the above

Answer: A. Using NATURAL JOINS in this case can yield unexpected results as there is an implicit searching of columns with identical names which in this case is not present.

18.What of the following can be used to fetch non-matching rows along with the matching rows between a source and a target table in Oracle DB?

A. EQUI-JOIN

B. SELF-JOIN

C. NATURAL JOIN

D. OUTER-JOIN

Answer: D. An outer join is created when records need to be included in the results without having corresponding records in the join tables. These records are matched with NULL records so that they're included in the output.

19. What are Cartesian Joins also known as in Oracle DB?

A. Equi-join

B. Anti-join

C. Cross-Join

D. None of the above

Answer: C. A Cartesian join between two tables returns every possible combination of rows from the tables. A Cartesian join can be produced by not including a join operation in the query or by using a CROSS JOIN.

20.What will be the result of a NATURAL JOIN between two tables EMPLOYEES and DEPARTMENT as given in the query below? *Considerthetablestructuresasgiven*

```
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)
```

```
SELECT * FROM employees NATURAL JOIN department;
```

A. The common column DEPARTMENT_ID with identical name in both the tables will appear twice in the result set

B. All the columns having identical names joined with the NATURAL JOIN, will appear twice in the result set

C. The result set will have only one column for each pair of identically named columns from both tables

D. None of the above

Answer: C. The NATURAL JOIN keywords don't require a condition to establish the relationship between two tables. However, a common column must exist. Column qualifiers can't be used with the NATURAL JOIN keywords.

21.What is the difference between a NATURAL JOIN and an EQUI-JOIN in Oracle DB?

A. There is no difference

B. They are the same with respect to the result set that is obtained from both

C. Both A and B

D. None of the above

Answer: D. NATURAL JOIN joins all the columns with identical names whereas EQUI-JOIN requires the columns to be explicitly mentioned in the SQL query.

22.What is an INNER JOIN in Oracle DB?

A. The join giving the matching records between two tables is called an INNER JOIN

B. An inner join can use operators like <,>, <>

C. Both A and B

D. None of the above

Answer: C. A join can be an inner join, in which the only records returned have a matching record in all tables, or an outer join, in which records can be returned regardless of whether there's a matching record in the join.

23.What is the difference between a INNER JOIN and an EQUI-JOIN in Oracle DB?

A. They are the same in terms of syntax and result sets obtained.

B. An INNER JOIN is a subset of an EQUI-JOIN

C. An INNER JOIN can use operators like <,>, <> along with "=" while EQUI-JOIN only uses the "=" operator

D. All of the above

Answer: C. EQUI-JOIN is a type of INNER JOIN containing "=" operator in a join condition, whereas the INNER JOIN can contain both equality as well non-equality operators

24.What is true about NATURAL JOINS in terms of ANSI SQL: 1999 syntaxes in Oracle DB?

A. An Equality operator = is used

B. They fetch different results when compared to the traditional syntax

C. The ANSI SQL syntax uses words like NATURAL JOIN in the SQL queries

D. None of the above.

Answer: C. ANSI SQL syntax is different from the traditional way of using = in the traditional ways. There are keywords like NATURAL JOIN etc. in the ANSI SQL syntax to distinguish the joins used.

25.What of the following is true with respect to the query given below?

Considerthetablestructuresasgiven

```
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)
```

```
SELECT first_name, salary
FROM employees e, departments d
WHERE e.department_id  (+) = d.department_id ;
```

A. There is an outer join between the department_id from both tables which is equivalent to a Right Outer join in ANSI SQL

B. There is an outer join between the department_id from both tables which is equivalent to a Left Outer join in ANSI SQL

C. It fetches all the records of the department_id from the employees table whether they match or not

D. It fetches all the records of the department_id from the department table whether they match or not

Answer: A, D. The condition e.department_id $+$ = d.department_id means it will perform a Right Outer Join and all the department_id s from the department table will be displayed whether they match or not

26.Which of the following syntax models is used in extensively in the software systems worldwide?

A. ANSI SQL: 1999

B. Both traditional Oracle syntax and the ANSI SQL: 1999 syntax

C. Traditional Oracle syntax

D. All of the options

Answer: C. The ANSI SQL: 1999 syntax though not used as much as the traditional Oracle syntax, it still is one of the syntaxes that may be used in Oracle SQL

27.What of the following is true regarding the Cartesian product in Oracle DB?

A. If 'N' is the no of tables joined, then if no. of joins is N-1, Cartesian product is not performed

B. If 'N' is the no of tables joined, then if no. of joins is N, Cartesian product is performed

C. If 'N' is the no of tables joined, then if no. of joins is N+1, Cartesian product is performed

D. If 'N' is the no of tables joined, then if no. of joins is N-1 or less, Cartesian product is performed.

Answer: A. A Cartesian join between two tables returns every possible combination of rows from the tables. A Cartesian join can be produced by not including a join operation in the query or by using a CROSS JOIN. A query must have at least $N-1$ join conditions to prevent a cartesian product, where N is the number of tables in the query.

28.What is the reason of error in the following SQL query? *Considerthetablestructuresasgiven*

```
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)
```

```
SELECT first_name, last_name
FROM employees, departments
WHERE department_id (+) = department_id ;
```

A. There are no aliases used for the tables.

B. The word RIGHT OUTER JOIN is not used here hence it throws the error

C. The + should be on the Right side of the equality condition and not on the left side

D. Table Aliases should be used with the department_id in the condition to remove ambiguous naming

Answer: D. Without the table aliases, Oracle is unable to derive the origin of the columns being joined and hence throws an Ambiguity error on execution.

29.Which of the following is used to avoid the ambiguous column problem in Oracle DB?

A. ;

B. ,

C. .

D. /

Answer: C. The syntax for removing the Ambiguous column issue is: table_alias.column_name

30.Which of the following is the most appropriate about the following query?

Considerthetablestructuresasgiven

```
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)
```

```
SELECT employee_id , first_name, last_name
```

```
FROM employees e right outer join department d
On e.department_id  = d.department_id ;
```

A. It gives the details of those employees who are not in any department

B. It gives the details of those departments which do not have any employee

C. It gives the details of all departments irrespective of whether they've any employee or not

D. It gives the details of the employees which were hired in the company 'ABC' irrespective of the departments.

Answer: C. With the JOIN method for outer joins, you can add the LEFT, RIGHT, or FULL keywords. A left outer join includes all records from the table listed on the left side of the join, even if no match is found with the other table in the join operation. A full outer join includes all records from both tables, even if no corresponding record in the table is found.

31.What will be the outcome of the following query? *Considerthetablestructuresasgiven*

```
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)
```

```
SELECT *
FROM employees e, department d
WHERE a.department_id  = b.department_id ;
```

A. It will give all the matching values from both the tables which have the same department_id

B. It will give all the columns from the table employees and only the top 100 rows from the departments table

C. It will give an ORA error: "b.department_id " invalid identifier

D. None of the above options

Answer: C. The same aliases should in the WHERE clause as declared in the FROM clause

32.Which of the following is true regarding the USING and ON clauses in table joins? *Choosemorethanoneoptionsifapplicable*

A. The ON clause can be used to join tables on columns with the same data type but not necessarily the same name

B. The USING and ON clauses are used on only equijoins and non-equijoins

C. Not more than one condition can be used with the ON clause

D. The WHERE clause can be written after the USING..ON clause to apply additional conditions

Answer: A, D. The JOIN ... USING approach is similar to the NATURAL JOIN approach, except the common column is specified in the USING clause. A condition can't be included in the USING clause to indicate how the tables are related. In addition, column qualifiers can't be used for the common column specified in the USING clause. The JOIN... ON approach joins tables based on a specified condition. The JOIN keyword in the FROM clause indicates the tables to be joined, and the ON clause indicates how the two tables are related. This approach must be used if the tables being joined don't have a common column with the same name in each table.

33.How many tables can be joined by using the JOINS in Oracle DB?

A. 1

B. 2

C. 255

D. No limit

Answer: D. There is currently no limit on the number of tables participating in a join.

34. What is true when multiple joins are used in an SQL statement?

A. The joins are evaluated from left to right

B. The joins are evaluated from right to left

C. There is no precedence in the process of the evaluation of joins

D. None of the above

Answer: A. When multiple-joins exist in a statement, they are evaluated from left to right.

35.What is true with respect to the following query? *Considerthetablestructuresasgiven*

```
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)
```

```
SQL> desc bonus
 Name    Null?   Type
 ----------------------- -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID      VARCHAR2(10)
 SALARY      NUMBER(8,2)
 COMMISSION_PCT     NUMBER(2,2)
```

```
SELECT bonus, first_name, department_id
FROM bonus b NATURAL JOIN employees e NATURAL JOIN department d;
```

A. The use of NATURAL JOIN to join multiple tables is a better option and yields accurate results

B. The error probability is very less when NATURAL JOINs are used to join multiple tables

C. The clauses USING..JOIN..ON give more accurate results than NATURAL JOINs when joining multiple tables

D. Oracle implicitly joins multiple tables when a NATURAL JOIN is used and hence it is a good practice to use NATURAL JOINS

Answer: C. The use of NATURAL JOINS can create Cartesian products of rows and also are error prone with non-dependable result sets.

36.What is true about the clauses JOIN..ON in Oracle DB?

A. They are not very dependable as compared to NATURAL JOINS when joining multiple tables

B. The JOIN..ON clause is similar to the WHERE clause which limits rows with conditions

C. An additional WHERE clause is mandatory when the JOIN..ON clause is used

D. None of the above

Answer: B. The JOIN .... ON approach joins tables based on a specified condition. The JOIN keyword in the FROM clause indicates the tables to be joined, and the ON clause indicates how the two tables are related. This approach must be used if the tables being joined don't have a common column with the same name in each table.

Examine the table structures as given. Answer the questions 37 and 38 that follow the query 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)
```

```
SELECT e.salary, d.department_id
FROM employees e JOIN department d
On (e.department_id  = d.department_id  and e.last_name = 'Brandon');
```

37.What is true with respect to the query given above?

A. It gives an ORA error as the mandatory WHERE clause is not present

B. The JOIN..ON clause can't contain more than one condition

C. The query ignores the last condition and executes without an ORA error

D. The JOIN..ON clause can be written in the form given above for putting more conditions.

Answer: D. The WHERE clause can be omitted and the relevant conditions can be accommodated in the JOIN..ON clause itself as shown in the given query

38.With respect to the given query, if the JOIN used is replaced with NATURAL JOIN, it throws an error. What is the reason for this error?

A. When the NATURAL JOIN is used, a WHERE clause is mandatory, omitting which gives an error

B. The ON clause should be replaced with the USING clause

C. The words NATURAL, JOIN and USING are mutually exclusively in the context of the same join clause

D. A query can't combine the NATURAL JOIN and ON *orUSING* clauses while joining.

Answer: C, D.

39.What is true about Non-equijoins in Oracle DB?

A. They join based on the keyword NON-EQUI JOIN

B. They are used using the JOIN..ON clause with "=" sign

C. The results are obtained when the result of the inequality mentioned evaluates to true.

D. None of the above

Answer: C. The non-equi joins are used with the JOIN..ON clause but with inequality operators.

Examine the structures of the tables EMPLOYEES and DEPARTMENTS as given and answer the questions 40 and 41 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)
```

```
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)
```

40.What will be the outcome of the following query in Oracle DB?

```
SELECT e.first_name, e.last_name, e.employee_id
FROM employees e JOIN department d
ON (e.salary BETWEEN 1000 AND 10000);
```

A. It will throw an ORA error as the condition in the ON clause is incorrect.

B. It will throw an ORA error due to a syntax error as there is no Equality sign "=" in the ON clause

C. It will execute successfully and give the first name, last name and employee ID of employees with the condition mentioned.

D. Non-equi joins can only be used for showing in-equalities and not ranges.

Answer: C.

41.You need to find a report which lists the first names and last names of the employees whose salary is greater than 20000 and who are located in any of the departments in the location Geneva. Which of the following queries will give the required results?

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
AND d.loc = upper ('Geneva');
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000);
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000)
AND d.loc = 'Geneva';
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
WHERE d.loc = upper('Geneva');
```

Answer: A, C.

42.On which of the following conditions is a row returned when an EQUI-JOIN is used to join tables?

A. The result of the inequality match operation is true

B. The result of the inequality match operation is 0

C. The result of the inequality match operation is 1

D. The result of the inequality match operation is false

Answer: A. An equality join is created when data joining records from two different tables is an exact match *thatis, anequalityconditioncreatestherelationship*.The traditional approach uses an equal sign as the comparison operator in the WHERE clause. The JOIN approach can use the NATURAL JOIN, JOIN ... USING, or JOIN ... ON keywords.

41.You need to find a report which lists the first names and last names of the employees whose salary is greater than 20000 and who are located in any of the departments in the location Geneva. Which of the following queries will give the required results?

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
AND d.loc = upper ('Geneva');
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000);
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000)
AND d.loc = 'Geneva';
```

```
SELECT e.first_name, e.last_name
```

```
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
WHERE d.loc = upper('Geneva');
```

Answer: A, C.

42.On which of the following conditions is a row returned when an EQUI-JOIN is used to join tables?

A. The result of the inequality match operation is true

B. The result of the inequality match operation is 0

C. The result of the inequality match operation is 1

D. The result of the inequality match operation is false

Answer: A. An equality join is created when data joining records from two different tables is an exact match *thatis, anequalityconditioncreatestherelationship*.The traditional approach uses an equal sign as the comparison operator in the WHERE clause. The JOIN approach can use the NATURAL JOIN, JOIN ... USING, or JOIN ... ON keywords.

41.You need to find a report which lists the first names and last names of the employees whose salary is greater than 20000 and who are located in any of the departments in the location Geneva. Which of the following queries will give the required results?

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
AND d.loc = upper ('Geneva');
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000);
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000)
AND d.loc = 'Geneva';
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
WHERE d.loc = upper('Geneva');
```

Answer: A, C.

42.On which of the following conditions is a row returned when an EQUI-JOIN is used to join tables?

A. The result of the inequality match operation is true

B. The result of the inequality match operation is 0

C. The result of the inequality match operation is 1

D. The result of the inequality match operation is false

Answer: A. An equality join is created when data joining records from two different tables is an exact match *thatis, anequalityconditioncreatestherelationship*.The traditional approach uses an equal sign as the comparison operator in the WHERE clause. The JOIN approach can use the NATURAL JOIN, JOIN ... USING, or JOIN ... ON keywords.

41.You need to find a report which lists the first names and last names of the employees whose salary is greater than 20000 and who are located in any of the departments in the location Geneva. Which of the following queries will give the

required results?

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
AND d.loc = upper ('Geneva');
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000);
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >=20000)
AND d.loc = 'Geneva';
```

```
SELECT e.first_name, e.last_name
FROM employees e JOIN departments d
ON (e.department_id  = d.department_id  and e.salary >20000)
WHERE d.loc = upper('Geneva');
```

Answer: A, C.

42.On which of the following conditions is a row returned when an EQUI-JOIN is used to join tables?

A. The result of the inequality match operation is true

B. The result of the inequality match operation is 0

C. The result of the inequality match operation is 1

D. The result of the inequality match operation is false

Answer: A. An equality join is created when data joining records from two different tables is an exact match *thatis, anequalityconditioncreatestherelationship*.The traditional approach uses an equal sign as the comparison operator in the WHERE clause. The JOIN approach can use the NATURAL JOIN, JOIN ... USING, or JOIN ... ON keywords.

43.What is true regarding a Self-Join in Oracle DB?

A. Only two tables are required for the join to work

B. The columns in the result set are obtained from two tables but are displayed in one table

C. Conceptually, the source table duplicates itself to create the target table. *Oracledoesn'tduplicatetables*

D. All of the above

Answer: C. Self-joins are used when a table must be joined to itself to retrieve the data you need. Table aliases are required in the FROM clause to perform a self-join.

44. With respect to the query and the table structure given below,answer the question.

```
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 *
FROM employees a join employees b
WHERE a.employee_id  = b.employee_id ;
```

Which of the following tables does Oracle treat as source table and target table?

A. a is source and b is target

B. b is source and a is target

C. Oracle doesn't treat any of the tables as source or target

D. None of the above

Answer: A. The first occurrence of employees table is taken as source and the subsequent occurrences as b, c and so on.

45.In what scenarios can we use Self-Joins ideally in Oracle DB?

A. When we need to find the duplicate records in a table

B. When we need to obtain NULL values from a table

C. When we need to display a column of a table twice

D. When we need to display hierarchy of relevant entities

Answer: D.

46. What is true about NATURAL JOINS in Oracle DB?

A. The joined columns have the same name but can have different data types

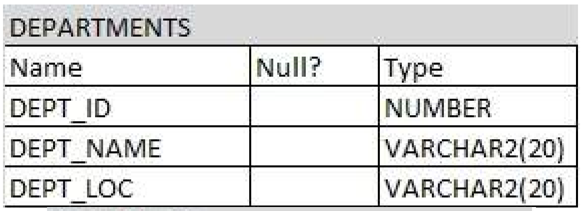
B. The joined columns can have the same data type but different names

C. The joined columns should have identical names and the same data type

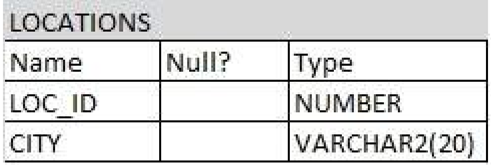
D. None of the above

Answer: C. The NATURAL JOIN keywords don't require a condition to establish the relationship between two tables. However, a common column must exist.Column qualifiers can't be used with the NATURAL JOIN keywords.

47.A report has to be extracted which gives the department name, department ID, department city and location ID only for departments 100 and 101. Using NATURAL JOINS, which of the following options will give the required results?



DEPARTMENTS		
Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)



LOCATIONS		
Name	Null?	Type
LOC_ID		NUMBER
CITY		VARCHAR2(20)

```
SELECT department_id , department_name  ,location, city
FROM departments
NATURAL JOIN locations
WHERE department_id in (100,101);
```

```
SELECT department_id , department_name  ,location, city
```

```
FROM locations
NATURAL JOIN departments
WHERE department_id BETWEEN 100 AND 101;
```

```
SELECT department_id , department_name  ,location, city
FROM departments
NATURAL JOIN locations
WHERE department_id >100
AND department_id >101;
```

```
SELECT department_id , department_name  ,location, city
FROM departments
NATURAL JOIN locations ;
```

Answer: A. The WHERE can be used for additional conditions after the NATURAL JOIN clause.

48.In which of the following scenarios shall a USING clause or a NATURAL JOIN clause be used?

A. When the names of the columns from the tables are identical then use USING clause

B. When the data types of the columns from the tables are identical then use NATURAL JOINS

C. If several columns have the same names but the data types do not match, USING can be used

D. NATURAL JOINS should be used only when the column names and their data types are the same

Answer: C, D. NATURAL JOINS and USING are mutually exclusive, the USING clause should be used to match only one column when more than one columns match.

49.Examine the table structures given. 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)
```

```
SQL> desc bonus
 Name    Null?   Type
 ----------------------- -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID      VARCHAR2(10)
 SALARY      NUMBER(8,2)
 COMMISSION_PCT     NUMBER(2,2)
```

```
SELECT e.name, b.bonus
FROM employees e
JOIN bonus b
USING (job_id)
WHERE e.job_id  like 'SA%');
```

A. It gives the names and bonus obtained by all the employees in some company

B. It gives the names and bonus obtained by all the employees in a particular job title in a company

C. It executes successfully giving all the names and the bonus obtained by all the employees in all jobs

D. It throws an ORA error.

Answer: D. The columns used in the USING clause should not have a qualifier *tablenameoralias* anywhere in the SQL statement.

50.What is true with respect to INNER JOINS and OUTER JOINS in Oracle DB?

A. INNER JOIN returns only the rows that are matched

B. OUTER JOIN returns only the rows that are not matched

C. OUTER JOIN returns the rows that are matched as well as those which do not match

D. None of the above

Answer: A, C. A join can be an inner join,in which the only records returned have a matching record in all tables,or an outer join, in which records can be returned regardless of whether there's a matching record in the join.An outer join is created when records need to be included in the results without having corresponding records in the join tables. These records are matched with NULL records so that they're included in the output.

51. What is true regarding FULL OUTER JOIN in Oracle DB?

A. When both LEFT OUTER JOIN and RIGHT OUTER JOIN appear in the same query,it is called a FULL OUTER JOIN

B. A FULL OUTER JOIN is the same as an OUTER JOIN

C. Both A and B

D. A join between two tables that returns the results of an INNER join and a LEFT and RIGHT OUTER JOIN is called a FULL OUTER JOIN

Answer: D. A full outer join includes all records from both tables, even if no corresponding record in the other table is found.

Examine the given table structures and answer the questions 52 and 53 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)
```

```
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)
```

```
SQL> desc bonus
 Name    Null?   Type
 ----------------------- -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID       VARCHAR2(10)
 SALARY       NUMBER(8,2)
 COMMISSION_PCT      NUMBER(2,2)
```

52.Consider the following query.

```
SELECT e.job_id , e.first_name, d.department_id
FROM departments D JOIN employees e JOIN BONUS b
USING (job_id );
```

This query results in an error. What is the reason of the error?

A. A JOIN..USING can happen only between two tables at a time

B. USING clause in the query doesn't have any column from the department

C. There is no WHERE clause in the query

D. None of the above

Answer: A. Table1 JOIN table2 JOIN table3 is not allowed without the ON clauses for between each JOIN

53.You need to display all the non-matching rows from the EMPLOYEES table and the non-matching rows from the DEPARTMENT table without giving a Cartesian product of rows between them. Which of the following queries will give the desired output?

```
SELECT *
FROM employees e, department d
WHERE e.department_id  = d.department_id ;
```

```
SELECT *
FROM employees e NATURAL JOIN department d;
```

```
SELECT *
FROM employees e FULL OUTER JOIN department d
ON  e.department_id  = d.department_id ;
```

```
SELECT *
FROM employees e JOIN  department d
ON ( e.department_id  > d.department_id ) ;
```

Answer: C. The FULL OUTER JOIN returns the non-matched rows from both the tables. A full outer join includes all records from both tables, even if no corresponding record in the other table is found.

54.Which of the following ANSI SQL: 1999 join syntax joins are supported by Oracle?

A. Cartesian products

B. Natural joins

C. Full OUTER join

D. Equijoins

Answer: D.

55.Which of the following is not a format for Outer Joins in Oracle DB?

A. Right

B. Left

C. Centre

D. Full

Answer: C. Except 'Centre', rest 3 types are the types of formats of the Outer Joins in Oracle DB. With the JOIN method for outer joins, you can add the LEFT, RIGHT, or FULL keywords.

Examine the given table structures. Answer the questions 56, 57 and 58 that follow by referring to 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> 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)
```

```
SELECT *
FROM employees e NATURAL JOIN department d;
```

56.You need to find the results obtained by the above query only for the departments 100 and 101. Which of the following clauses should be added / modified to the above query?

A. ON $e. department_i d = d. department_i d$ should be added

B. USING $e. department_i d$ should be added

C. WHERE e.department_id in $100, 101$ should be added

D. None of the above

Answer: C. The NATURAL JOIN clause implicitly matches all the identical named columns. To add additional conditions the WHERE clause can be used.

57.You need to find the results obtained by the above query for all those employees who have salaries greater than 20000. Which of the following clauses should be added / modified to the above query?

A. ON $e. department_i d = d. department_i d$ WHERE salary > 20000;

B. USING $e. department_i d$ WHERE salary > 20000;

C. USING $department_i d$ WHERE salary>20000;

D. WHERE salary >20000;

Answer: D.

58.If the NATURAL JOIN in the above query is replaced by only JOIN which of the following should be added / modified to the above query to give the results pertaining to Department 100?

A. ON $department_i d = 100;$

B. USING $e.\ department_id = 100;$

C. WHERE d.department_id = 100;

D. ON $e.\ department_id = d.\ department_idandd.\ department_id = 100;$

Answer: D. The equi-joins can be added for more conditions after the ON clause.

59.A report has to be extracted to get the Managers for all the employees in the departments 10 and 20 of a company 'ABC'. Which of the following queries will give the required results? ***Considerthetablestructureasgiven***

```
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 a.first_name || ' '||a.last_name "Manager", b.first_name||' '||b.last_name
"Employees"
FROM employees a join employees b
On (employee_id );
```

```
SELECT a.first_name || ' '||a.last_name "Manager", b.first_name||' '||b.last_name
"Employees"
FROM employees a join employees b
On (b.employee_id  = a.employee_id );
```

```
SELECT a.first_name || ' '||a.last_name "Manager", b.first_name||' '||b.last_name
"Employees"
FROM employees a join employees b
On (a.manager_id  = b.employee_id )
WHERE department_id  in (10,20);
```

```
SELECT a.first_name || ' '||a.last_name "Manager", b.first_name||' '||b.last_name
"Employees"
FROM employees a join employees b
On (a.manager_id  = b.employee_id )
WHERE a.department_id  in (10,20);
```

Answer: D. The option C is incorrect because the non-aliased department_id in the WHERE clause will throw an error.

60.Which of the following queries will give results without duplicate values between the two tables EMPLOYEES and DEPARTMENT? ***Considerthetablestructuresasgiven***

```
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)
```

```
SELECT *
FROM employees e NATURAL JOIN department d;
```

```
SELECT *
FROM employees e JOIN department d;
```

```
SELECT *
FROM employees e NATURAL JOIN department d
USING (e.department_id );
```

```
SELECT *
FROM employees e FULL OUTER JOIN department d
USING (department_id );
```

Answer: D. The FULL OUTER JOIN will give all the matching as well non-matching rows from both the tables excluding duplicate values.

Examine the structures for the tables as given here and answer the questions 61 to 64.

```
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 bonus
 Name    Null?   Type
 ----------------------- -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID      VARCHAR2(10)
 SALARY      NUMBER(8,2)
 COMMISSION_PCT     NUMBER(2,2)
```

```
SQL> desc locations
 Name    Null?   Type
 ----------------------- -------- ----------------
 LOCATION_ID   NOT NULL NUMBER(4)
 STREET_ADDRESS     VARCHAR2(40)
 POSTAL_CODE     VARCHAR2(12)
 CITY    NOT NULL VARCHAR2(30)
 STATE_PROVINCE     VARCHAR2(25)
 COUNTRY_ID     CHAR(2)
```

```
SQL> desc locations
 Name    Null?   Type
 ----------------------- -------- ----------------
 LOCATION_ID   NOT NULL NUMBER(4)
 STREET_ADDRESS     VARCHAR2(40)
 POSTAL_CODE     VARCHAR2(12)
 CITY    NOT NULL VARCHAR2(30)
 STATE_PROVINCE     VARCHAR2(25)
 COUNTRY_ID     CHAR(2)
```

61.What is true about the following query? *Choosethemostappropriateanswer*

```
SELECT *
FROM bonus b, employees e
WHERE b.job_id  (+) = e.job_id ;
```

A. It will display all the bonuses obtained by all the employees.

B. It will display NULL for the bonus column if a particular employee has not received any bonus

C. Both A and B

D. None of the above

Answer: B. The + is on the LHS of the equation means it is a RIGHT OUTER JOIN and vice versa.

62.You have to list all the departments who have no employees yet in a company named 'XYZ'. Which of the following queries will give you the required results?

```
SELECT department_id , department_name FROM departments d NATURAL JOIN employees e;
```

```
SELECT department_id , department_name FROM employees e JOIN departments d
ON (e.department_id  = d.department_id );
```

```
SELECT department_id , department_name FROM employees e LEFT OUTER JOIN departments
d
USING (department_id );
```

```
SELECT department_id , department_name FROM employees e RIGHT OUTER JOIN
departments d
ON (e.department_id  = d.department_id );
```

Answer: D.

63.You need to extract a report which displays 'No department yet' for all those employees who are yet to be allocated to a department. Which of the following will fulfill the purpose?

```
SELECT nvl(department_id ,'No department yet')
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_id  = d.department_id );
```

```
SELECT nvl(department_id ,'No department yet')
FROM departments d LEFT OUTER JOIN employees e
ON (e.department_id  = d.department_id );
```

```
SELECT nvl(department_id ,'No department yet')
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_id  = d.department_id );
```

```
SELECT nvl(department_id ,'No department yet')
FROM employees e FULL OUTER JOIN departments d
ON (e.department_id  = d.department_id );
```

Answer: C.

64.You need to extract a report which displays all the departments which have not been assigned to a city yet. Which of the following queries will give you the required output?

```
SELECT department_id , department_name FROM departments d NATURAL JOIN locations l;
```

```
SELECT department_id , department_name FROM departments d FULL OUTTER JOIN
locations l
ON (d.location_id = l.location_id);
```

```
SELECT  d.department_id , d.department_name FROM departments d JOIN locations l
USING (location_id);
```

```
SELECT department_id , department_name FROM departments d LEFT OUTER JOIN locations
l
ON (d.location_id = l.location_id);
```

Answer: D.

65.In which two cases an OUTER JOIN should be used?

A. If the joined tables' columns have NULL values

B. If the joined tables have NOT NULL columns

C. If the joined tables have only un-matched data

D. If the joined tables have both matching as well as non-matching data

Answer: A, D. An outer join is created when records need to be included in the results without having corresponding records in the join tables. These records are matched with NULL records so that they're included in the output.

66.You need to find the salary grade obtained by each employee. Which of the following query will you use? *Considerthetablestructuresasgiven*

```
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 grade
 Name    Null?   Type
 ----------------------- -------- ----------------
 GRADE       NUMBER
 LOSAL       NUMBER
 HISAL      NUMBER
```

```
SELECT employee_id , salary, grade
FROM employees e JOIN grade g
ON g.grade BETWEEN g.losal AND g.hisal
```

```
SELECT employee_id , salary, grade
```

```
FROM employees e FULL OUTER JOIN grade g
WHERE g.grade > g.losal AND < g.hisal;
```

```
SELECT employee_id , salary, grade
FROM employees e JOIN grade g
ON (MIN(g.grade) = g.losal
AND MAX(g.grade) = g.hisal);
```

D. None of the above

Answer: A.

67.Examine the table structures 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)
```

```
SQL> desc bonus
 Name    Null?   Type
 ------------------------ -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID      VARCHAR2(10)
 SALARY      NUMBER(8,2)
 COMMISSION_PCT     NUMBER(2,2)
```

Evaluate this SQL statement:

```
SELECT e.employee_id , (.25* e.salary) + (.5 * e.commission_pct) + (.75 * b.bonus) as
calc_val
FROM employees e, bonus b
WHERE e.employee_id  = b.employee_id ;
```

What will happen if you remove all the parentheses from the calculation?

A. The value displayed in the calc_val column will be lesser.

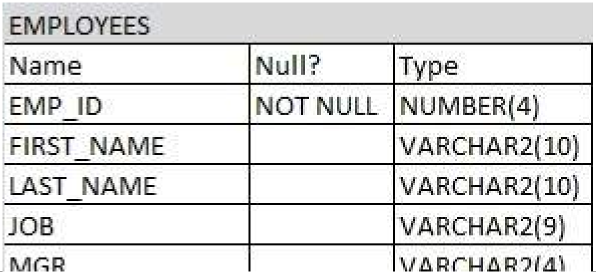
B. The value displayed in the calc_val column will be higher.

C. There will be no difference in the calc_val column.

D. An error will be reported.

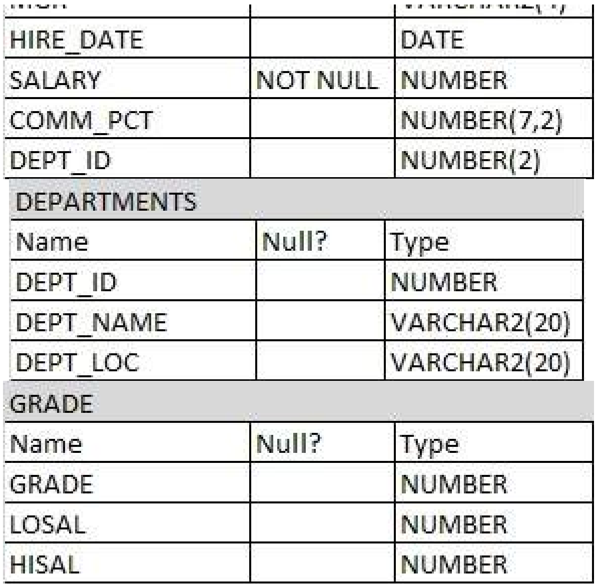
Answer: C.

68.Consider the exhibit and examine the structures of the EMPLOYEES, DEPARTMENTS, and GRADE tables. For which situation would you use a non-equijoin query?



EMPLOYEES

Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		VARCHAR2(4)



HIRE_DATE		DATE
SALARY	NOT NULL	NUMBER
COMM_PCT		NUMBER(7,2)
DEPT_ID		NUMBER(2)

DEPARTMENTS

Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)

GRADE

Name	Null?	Type
GRADE		NUMBER
LOSAL		NUMBER
HISAL		NUMBER

A. To find the grade for each of the employees

B. To list the name, job_id, and manager name for all the employees

C. To find the department name of employees.

D. To find the number of employees working for the Administrative department and earning less than 30000

Answer: A. A non-equality join establishes a relationship based on anything other than an equal condition. Range values used with non-equality joins must be mutually exclusive.

69.In which three cases would you use the USING clause? ***Choosethree.***

A. You want to create a non-equijoin.

B. The tables to be joined have multiple NULL columns.

C. The tables to be joined have columns of the same name and different data types.

D. You want to use a NATURAL join, but you want to restrict the number of columns in the join condition.

Answer: C, D. The JOIN .... USING approach is similar to the NATURAL JOIN approach, except the common column is specified in the USING clause. A condition can't be included in the USING clause to indicate how the tables are related. In addition, column qualifiers can't be used for the common column specified in the USING clause.

70.If the tables EMPLOYEES and BONUS have two columns with identical names viz: - SALARY and JOB_ID, which of the following queries are equivalent to each other? ***Considerthetablestructuresasgiven***

```
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 bonus
 Name    Null?   Type
 ----------------------- -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID       VARCHAR2(10)
 SALARY       NUMBER(8,2)
 COMMISSION_PCT     NUMBER(2,2)
```

```
SELECT * FROM employees E JOIN bonus B on (salary, job_id )
```

```
SELECT * FROM employees E NATURAL JOIN bonus B on (salary, job_id )
```

```
SELECT * FROM employees E JOIN bonus B USING (salary, job_id )
```

```
SELECT * FROM employees E JOIN bonus B on (salary, job_id )
```

Answer: B, C.

71.Examine the table structures 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)
```

```
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)
```

Examine the following two SQL statements:

Query 1

```
SELECT first_name,department_id
FROM employees E JOIN departments D
USING (department_id );
```

Query 2

```
SELECT first_name,department_id
FROM employees E NATURAL JOIN departments D
USING (department_id );
```

Which statement is true regarding the outcome?

A. Only query 1 executes successfully and gives the required result.

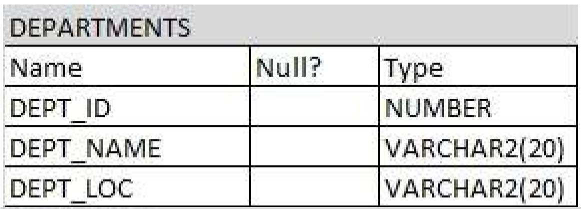
B. Only query 2 executes successfully and gives the required result.

C. Both queries 1 and 2 execute successfully and give different results.

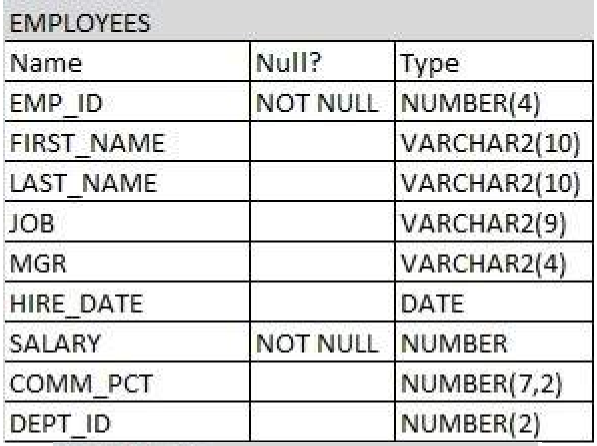
D. Both queries 1 and 2 execute successfully and give the same required result.

Answer: D.

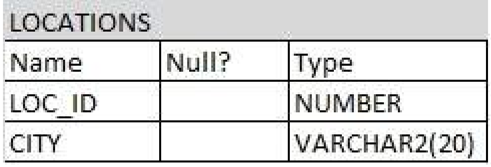
72.You need to generate a report showing the department location along with the employee name for all hires made before 20th January, 2013.



DEPARTMENTS		
Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)



EMPLOYEES		
Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		VARCHAR2(4)
HIRE_DATE		DATE
SALARY	NOT NULL	NUMBER
COMM_PCT		NUMBER(7,2)
DEPT_ID		NUMBER(2)



LOCATIONS		
Name	Null?	Type
LOC_ID		NUMBER
CITY		VARCHAR2(20)

You issue the following query:

```
SELECT department_name  , first_name||' '||last_name
FROM employees E JOIN department d
ON ( hire_date < '20-JAN-2013')
JOIN locations L
ON  (l.location_id = d.location_id) ;
```

Which statement is true regarding the above query?

A. It executes successfully and gives the required result.

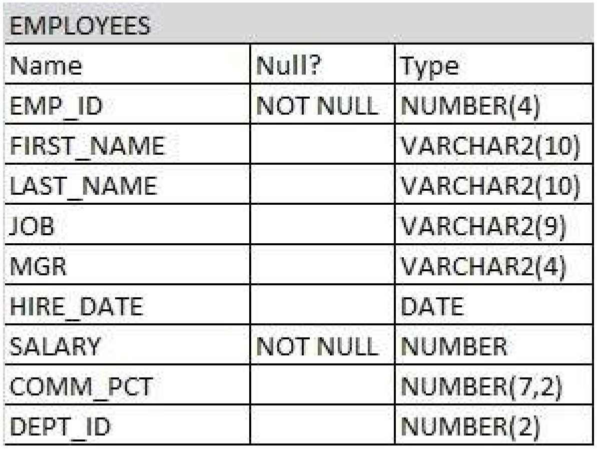
B. It executes successfully but does not give the required result.

C. It produces an error because the join order of the tables is incorrect.

D. It produces an error because equijoin and non-equijoin conditions cannot be used in the same SELECT statement.

Answer: B.

73.Examine the structure of the EMPLOYEES table:



EMPLOYEES		
Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		VARCHAR2(4)
HIRE_DATE		DATE
SALARY	NOT NULL	NUMBER
COMM_PCT		NUMBER(7,2)
DEPT_ID		NUMBER(2)

You want to find out if any employee' details have been entered more than once using different EMPLOYEE_ID , by listing all the duplicate names. Which method can you use to get the required result?

A. self-join

B. full outer-join with self-join

C. left outer-join with self-join

D. right outer-join with self-join

Answer: A. Self-joins are used when a table must be joined to itself to retrieve the data you need. Table aliases are required in the FROM clause to perform a self-join.

Examine the structure of the tables DEPARTMENTS and LOCATIONS and answer the questions 74 and 75 that follow.

```
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)
```

```
SQL> desc locations
 Name    Null?   Type
 ----------------------- -------- ----------------
 LOCATION_ID   NOT NULL NUMBER(4)
 STREET_ADDRESS     VARCHAR2(40)
 POSTAL_CODE     VARCHAR2(12)
 CITY    NOT NULL VARCHAR2(30)
 STATE_PROVINCE     VARCHAR2(25)
 COUNTRY_ID     CHAR(2)
```

74.You need to find out the departments that have not been allocated any location. Which query would give the required result?

```
SELECT d.department_id , d.department_name FROM departments d JOIN locations l
ON (d.location_id = l.location_id);
```

```
SELECT d.department_id , d.department_name FROM departments d RIGHT OUTER JOIN
locations l
ON (d.location_id = l.location_id);
```

```
SELECT d.department_id , d.department_name FROM departments d FULL JOIN locations l
ON (d.location_id = l.location_id);
```

```
SELECT d.department_id , d.department_name FROM departments d LEFT OUTER JOIN
locations l
ON (d.location_id = l.location_id);
```

Answer: B.

75.You want to list all departments that are not located in any location along with the department name. Evaluate the following query:

```
SELECT d.department_id , d.department_name  ,l.location_id, l.city
FROM departments D ___________________   location L
ON (d.location_id = l.location_id);
```

Which two JOIN options can be used in the blank in the above query to give the correct output?

A. JOIN

B. NATURAL JOIN

C. LEFT OUTER JOIN

D. RIGHT OUTER JOIN

Answer: A, C.

76. You need to generate a report that shows all department IDs, with corresponding employees *ifany* **and bonus details** *ifany*, **for all employees. Which FROM clause gives the required result?** *Considerthetablestructuresasgiven*

```
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 bonus
 Name    Null?   Type
 ----------------------- -------- ----------------
 EMPLOYEE_ID   NOT NULL NUMBER
 JOB_ID      VARCHAR2(10)
 SALARY      NUMBER(8,2)
 COMMISSION_PCT     NUMBER(2,2)
```

A. FROM departments LEFT OUTER JOIN employees USING $department_id$ FULL OUTER JOIN bonus

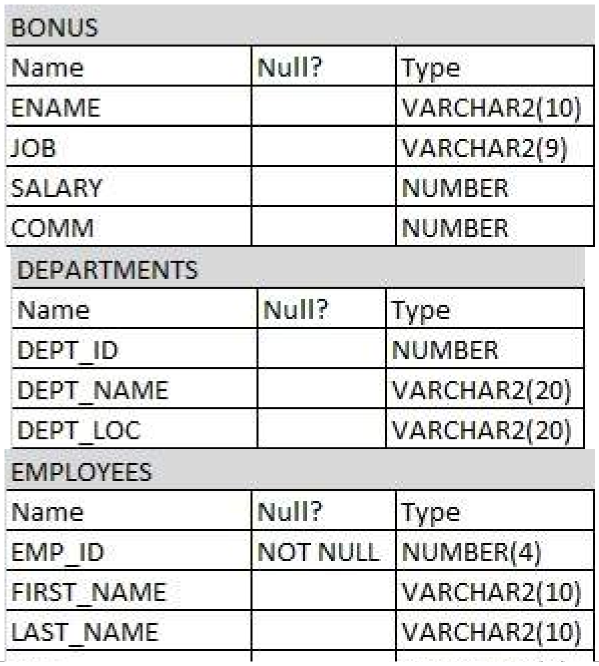
B. FULL OUTER JOIN department USING $department_id$;

C. FROM bonus JOIN employees USING job_id

D. FROM employees FULL OUTER JOIN departments FULL OUTER JOIN bonus

Answer: A.

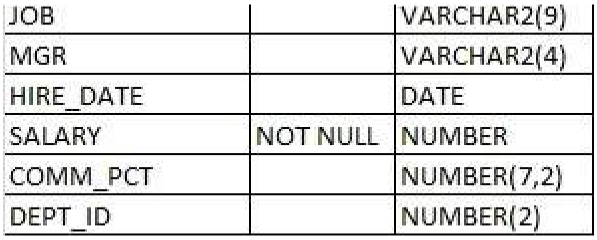
77. Examine the following exhibits:



BONUS		
Name	Null?	Type
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
SALARY		NUMBER
COMM		NUMBER

DEPARTMENTS		
Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)

EMPLOYEES		
Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)



JOB		VARCHAR2(9)
MGR		VARCHAR2(4)
HIRE_DATE		DATE
SALARY	NOT NULL	NUMBER
COMM_PCT		NUMBER(7,2)
DEPT_ID		NUMBER(2)

You want to generate a report listing the employees' IDs and their corresponding commissions and departments *ifany*, if the commissions exist or not. Evaluate the following query:

```
SELECT e.employee_id , bonus, department_name FROM bonus b_____________ employees
USING (job_id ) ____________ departments
USING (department_id )
WHERE commission_pct  IS NOT NULL;
```

Which combination of joins used in the blanks in the above query gives the correct output?

A. JOIN; LEFT OUTER JOIN

B. FULL OUTER JOIN; FULL OUTER JOIN

C. RIGHT OUTER JOIN; LEFT OUTER JOIN

D. LEFT OUTER JOIN; RIGHT OUTER JOIN

Answer: A.

78.Predict the outcome of the following query.

```
SELECT e.salary, bonus
FROM employees E JOIN bonus b
USING (salary,job_id );
```

A. It executes successfully.

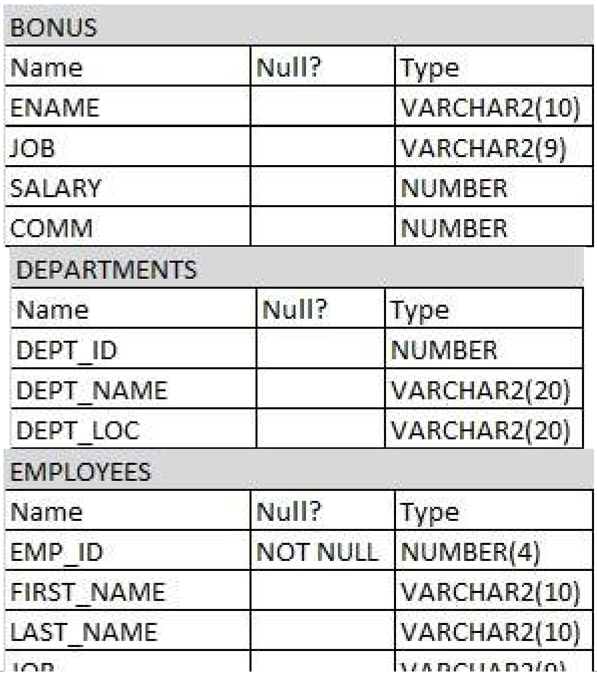
B. It throws an error because bonus in SELECT is not aliased

C. It throws an error because the USING clause cannot have more than 1 column.

D. It executes successfully but the results are not correct.

Answer: D.

View the Exhibit and examine the structure of the EMPLOYEES, DEPARTMENTS, LOCATIONS and BONUS. Answer the questions from 79 and 80 that follow:



BONUS

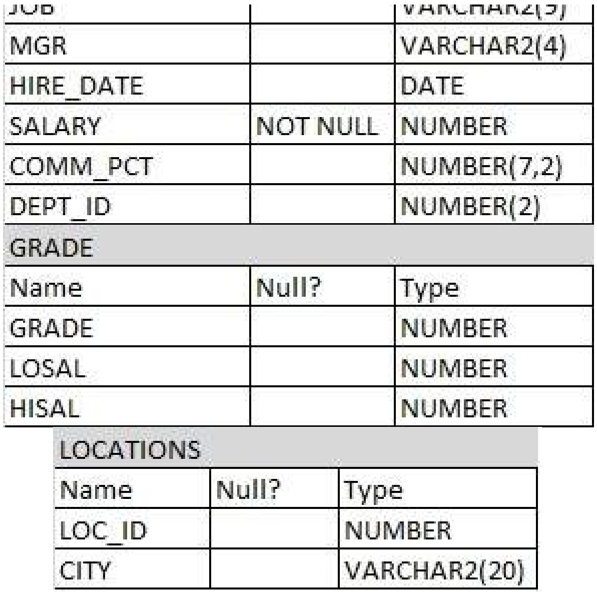
Name	Null?	Type
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
SALARY		NUMBER
COMM		NUMBER

DEPARTMENTS

Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)

EMPLOYEES

Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
JOB		VARCHAR2(9)



JOB		VARCHAR2(9)
MGR		VARCHAR2(4)
HIRE_DATE		DATE
SALARY	NOT NULL	NUMBER
COMM_PCT		NUMBER(7,2)
DEPT_ID		NUMBER(2)

GRADE

Name	Null?	Type
GRADE		NUMBER
LOSAL		NUMBER
HISAL		NUMBER

LOCATIONS

Name	Null?	Type
LOC_ID		NUMBER
CITY		VARCHAR2(20)

79.You need to list all the departments in the city of Zurich. You execute the following query:

```
SELECT D.DEPARTMENT_ID , D.DEPARTMENT_NAME  , L.CITY
FROM departments D JOIN LOCATIONS L
USING (LOC_ID,CITY)
WHERE L.CITY = UPPER('ZURICH');
```

Predict the outcome of the above query.

A. It executes successfully.

B. It gives an error because a qualifier is used for CITY in the SELECT statement.

C. It gives an error because the column names in the SELECT do not match

D. It gives an error because the USING clause has CITY which is not a matching column.

Answer: D. Only the matching column names should be used in the USING clause.

80.Answer the question that follows the query given below:

```

SELECT e.first_name, d.department_id , e.salary, b.bonus
FROM bonus b join employees e
USING (job_id )
JOIN department d
USING (department_id )
WHERE d.loc = 'Zurich';
```

You need to extract a report which gives the first name, department number, salary and bonuses of the employees of a company named 'ABC'. Which of the following queries will solve the purpose?

```
SELECT e.first_name, d.department_id , e.salary, b.bonus
FROM bonus b join employees e join departments d
on (b.job_id  = e.job_id )
on (e.department_id =d.department_id )
WHERE d.loc = 'Zurich';
```

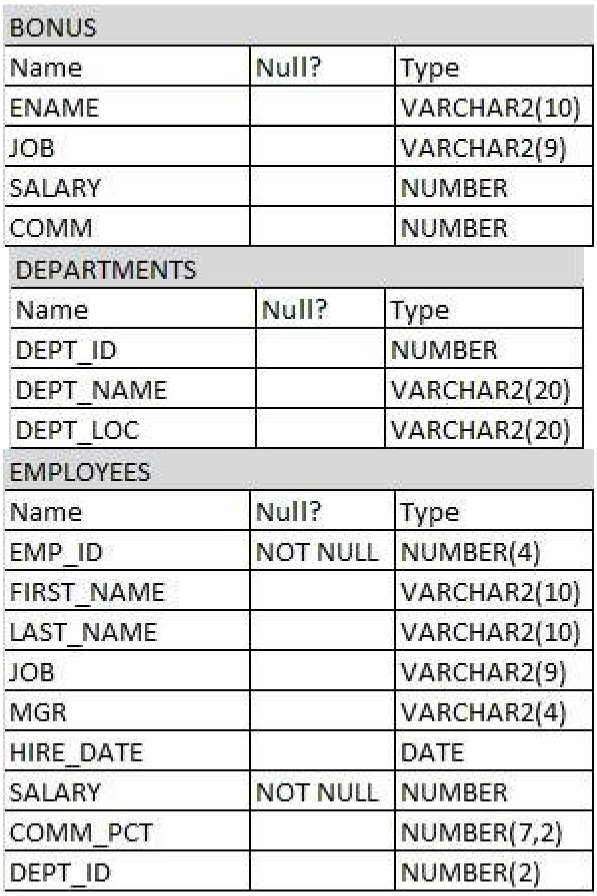
```
SELECT e.first_name, d.department_id , e.salary, b.bonus
FROM bonus b join employees e
on (b.job_id  = e.job_id )
JOIN department d
on (e.department_id =d.department_id )
WHERE d.loc = 'Zurich';
```

```
SELECT e.first_name, d.department_id , e.salary, b.bonus
FROM employees e join bonus b
USING (job_id )
JOIN department d
USING (department_id )
WHERE d.loc = 'Zurich';
```

D. None of the above

Answer: C. The query A will throw a syntactical error, query B will throw an invalid identifier error between bonus and department.

Examine the Exhibits given below and answer the questions 81 to 85 that follow.



BONUS

Name	Null?	Type
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
SALARY		NUMBER
COMM		NUMBER

DEPARTMENTS

Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)

EMPLOYEES

Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		VARCHAR2(4)
HIRE_DATE		DATE
SALARY	NOT NULL	NUMBER
COMM_PCT		NUMBER(7,2)
DEPT_ID		NUMBER(2)

81. You need to find the managers' name for those employees who earn more than 20000. Which of the following queries will work for getting the required results?

```
SELECT e.employee_id  "Employee", salary, employee_id ,
FROM employees E JOIN employees M
USING (e.manager_id  = m.employee_id )
WHERE e.salary >20000;
```

```
SELECT e.employee_id  "Employee", salary, employee_id ,
FROM employees E JOIN employees M
USING (e.manager_id)
WHERE e.salary >20000;
```

```
SELECT e.employee_id  "Employee", salary, employee_id ,
FROM employees E  NATURAL JOIN employees M
USING (e.manager_id = m.employee_id )
WHERE e.salary >20000;
```

```
SELECT e.employee_id  "Employee", salary, employee_id ,
FROM employees E JOIN employees M
ON (e.manager_id = m.employee_id )
WHERE e.salary >20000;
```

Answer: D.

82.You issue the following query:

```
SELECT e.employee_id ,d.department_id
FROM employees e NATURAL JOIN department d NATURAL JOIN bonus b
WHERE department_id  =100;
```

Which statement is true regarding the outcome of this query?

A. It executes successfully.

B. It produces an error because the NATURAL join can be used only with two tables.

C. It produces an error because a column used in the NATURAL join cannot have a qualifier.

D. It produces an error because all columns used in the NATURAL join should have a qualifier.

Answer: C.

83.You want to display all the employee names and their corresponding manager names. Evaluate the following query:

```
SELECT e.first_name "EMP NAME", m.employee_name "MGR NAME"
FROM employees e ______________ employees m
ON e.manager_id = m.employee_id ;
```

Which JOIN option can be used in the blank in the above query to get the required output?

A. Simple inner JOIN

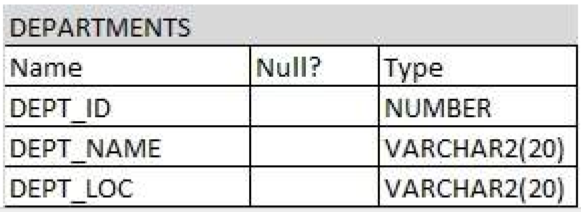
B. FULL OUTER JOIN

C. LEFT OUTER JOIN

D. RIGHT OUTER JOIN

Answer: C. A left outer join includes all records from the table listed on the left side of the join, even if no match is found with the other table in the join operation.

Consider the below exhibit and following query to answer questions 84 and 85. *Assumethetabledepartmenthasmanager$_i$danddepartment$_n$ameasitscolumns*



DEPARTMENTS		
Name	Null?	Type
DEPT_ID		NUMBER
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(20)

```

Select *
FROM employees e JOIN department d
ON (e.employee_id  = d.manager_id);
```

84. You need to display a sentence "*first$_n$ame last$_n$ame* is manager of the *department$_n$ame* department". Which of the following SELECT statements will successfully replace '*' in the above query to fulfill this requirement?

```
SELECT e.first_name||' '||e.last_name||' is manager of the '||d.department_name||'
department.' "Managers"
```

```
SELECT e.first_name, e.last_name||' is manager of the '||d.department_name||'
department.' "Managers"
```

```
SELECT e.last_name||' is manager of the '||d.department_name||' department.'
```

```
"Managers"
```

D. None of the above

Answer: A.

85.What will happen if we omit writing the braces "" after the ON clause in the above query?

A. It will give only the names of the employees and the managers' names will be excluded from the result set

B. It will give the same result as with braces ""

C. It will give an ORA error as it is mandatory to write the braces "" when using the JOIN..ON clause

D. None of the above

Answer: B. The braces are not mandatory, but using them provides a clear readability of the conditions within it.

86. Which of the following queries creates a Cartesian join?

```
SELECT title, authorid FROM books, bookauthor;
```

```
SELECT title, name FROM books CROSS JOIN publisher;
```

```
SELECT title, gift FROM books NATURAL JOIN promotion;
```

D. all of the above

Answer: A, B. A Cartesian join between two tables returns every possible combination of rows from the tables. A Cartesian join can be produced by not including a join operation in the query or by using a CROSS JOIN.

87. Which of the following operators is not allowed in an outer join?

A. AND

B. =

C. OR

D. >

Answer: C. Oracle raises the exception "ORA-01719: outer join operator + not allowed in operand of OR or IN"

88. Which of the following queries contains an equality join?

```
SELECT title, authorid FROM books, bookauthor WHERE books.isbn = bookauthor.isbn AND
retail > 20;
```

```
SELECT title, name FROM books CROSS JOIN publisher;
```

```
SELECT title, gift FROM books, promotion WHERE retail >= minretail AND retail <=
maxretail;
```

D. None of the above

Answer: A. An equality join is created when data joining records from two different tables is an exact match *thatis, anequalityconditioncreatestherelationship*.

89. Which of the following queries contains a non-equality join?

```
SELECT title, authorid FROM books, bookauthor WHERE books.isbn = bookauthor.isbn AND
retail > 20;
```

```
SELECT title, name FROM books JOIN publisher USING (pubid);
```

```
SELECT title, gift FROM books, promotion WHERE retail >= minretail AND retail <=
maxretail;
```

D. None of the above

Answer: D. Nonequijoins match column values from different tables based on an inequality expression. The value of the join column in each row in the source table is compared to the corresponding values in the target table. A match is found if the expression used in the join, based on an inequality operator, evaluates to true. When such a join is constructed, a nonequijoin is performed.A nonequijoin is specified using the JOIN..ON syntax, but the join condition contains an inequality operator instead of an equal sign.

90. The following SQL statement contains which type of join?

```
SELECT title, order#, quantity
FROM books FULL OUTER JOIN orderitems
ON books.isbn = orderitems.isbn;
```

A. equality

B. self-join

C. non-equality

D. outer join

Answer: D. A full outer join includes all records from both tables, even if no corresponding record in the other table is found.

91. Which of the following queries is valid?

```
SELECT b.title, b.retail, o.quantity FROM books b NATURAL JOIN orders od NATURAL
JOIN orderitems o WHERE od.order# = 1005;
```

```
SELECT b.title, b.retail, o.quantity FROM books b, orders od, orderitems o WHERE
orders.order# = orderitems.order# AND orderitems.isbn=books.isbn AND
od.order#=1005;
```

```
SELECT b.title, b.retail, o.quantity FROM books b, orderitems o WHERE o.isbn =
b.isbn AND o.order#=1005;
```

D. None of the above

Answer: C. If tables in the joins have alias, the selected columns must be referred with the alias and not with the actual table names.

92. Given the following query.

```
SELECT zip, order#
FROM customers NATURAL JOIN orders;
```

Which of the following queries is equivalent?

```
SELECT zip, order# FROM customers JOIN orders WHERE customers.customer# =
orders.customer#;
```

```
SELECT zip, order# FROM customers, orders WHERE customers.customer# =
orders.customer#;
```

```
SELECT zip, order# FROM customers, orders WHERE customers.customer# =
orders.customer# (+);
```

D. none of the above

Answer: B. Natural join instructs Oracle to identify columns with identical names between the source and target tables.

93. Examine the table structures as given. Which line in the following SQL statement raises an error?

```
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)
```

```
1. SELECT e.first_name, d.department_name
2. FROM employees  e, department d
3. WHERE e.department_id=d.department_id
```

A. line 1

B. line 2

C. line 3

D. No errors

Answer: A. If a query uses alias names in the join condition, their column should use the alias for reference.

94. Given the following query:

```
SELECT lastname, firstname, order#
FROM customers LEFT OUTER JOIN orders
USING (customer#)
ORDER BY customer#;
```

Which of the following queries returns the same results?

```
SELECT lastname, firstname, order# FROM customers c OUTER JOIN orders o ON
c.customer# = o.customer# ORDER BY c.customer#;
```

```
SELECT lastname, firstname, order# FROM orders o RIGHT OUTER JOIN customers c ON
c.customer# = o.customer# ORDER BY c.customer#;
```

```
SELECT lastname, firstname, order# FROM customers c, orders o WHERE c.customer# =
o.customer# (+) ORDER BY c.customer#;
```

D. None of the above

Answer: B, C.

95. Which of the below statements are true?

A. Group functions cannot be used against the data from multiple data sources.

B. If multiple tables joined in a query, contain identical columns, Oracle selects only one of them.

C. Natural join is used to join rows from two tables based on identical columns.

D. A and B

Answer: C. Group functions can be used on a query using Oracle joins. Ambiguous columns must be referenced using a qualifier.

96. Which line in the following SQL statement raises an error?

```
1. SELECT name, title
2. FROM books JOIN publisher
3. WHERE books.pubid = publisher.pubid
4. AND
5. cost < 45.95
```

A. line 1

B. line 2

C. line 3

D. line 4

Answer: C. Since the tables are joined using JOIN keyword, the equality condition should be written with the USING clause and not WHERE clause.

97. Given the following query:

```
SELECT title, gift
FROM books CROSS JOIN promotion;
```

Which of the following queries is equivalent?

```
SELECT title, gift
FROM books NATURAL JOIN promotion;
```

```
SELECT title
FROM books INTERSECT
SELECT gift
FROM promotion;
```

```
SELECT title
FROM books UNION ALL
SELECT gift
FROM promotion;
```

```
SELECT title, gift
FROM books, promotion;
```

Answer: D. Cartesian joins are same as Cross joins.

98. If the CUSTOMERS table contains seven records and the ORDERS table has eight records, how many records does the following query produce?

```
SELECT *
FROM customers CROSS JOIN orders;
```

A. 0

B. 56

C. 7

D. 15

Answer: B. Cross join is the cross product of rows contained in the two tables.

99. Which of the following SQL statements is not valid?

```
SELECT b.isbn, p.name
FROM books b NATURAL JOIN publisher p;
```

```
SELECT isbn, name
FROM books b, publisher p
WHERE b.pubid = p.pubid;
```

```
SELECT isbn, name
FROM books b JOIN publisher p
ON b.pubid = p.pubid;
```

```
SELECT isbn, name
FROM books JOIN publisher
USING (pubid);
```

Answer: A. Ambiguous columns must be referred with the table qualifiers.

100. Which of the following lists all books published by the publisher named 'Printing Is Us'?

```
SELECT title
FROM books NATURAL JOIN publisher
WHERE name = 'PRINTING IS US';
```

```
SELECT title
FROM books, publisher
WHERE pubname = 1;
```

```
SELECT *
FROM books b, publisher p
JOIN tables ON b.pubid = p.pubid
WHERE name = 'PRINTING IS US';
```

D. none of the above

Answer: A. Assuming that the column NAME is not contained in BOOKS table, query A is valid.

101. Which of the following SQL statements is not valid?

```
SELECT isbn
FROM books
MINUS
SELECT isbn
FROM orderitems;
```

```
SELECT isbn, name
FROM books, publisher
WHERE books.pubid (+) = publisher.pubid (+);
```

```
SELECT title, name
FROM books NATURAL JOIN publisher
```

D. All the above SQL statements are valid.

Answer: B. The query B raises an exception "ORA-01468: a predicate may reference only one outer-joined table".

102. Which of the following statements about an outer join between two tables is true?

A. If the relationship between the tables is established with a WHERE clause, both tables can include the outer join operator.

B. To include unmatched records in the results, the record is paired with a NULL record in the deficient table.

C. The RIGHT, LEFT, and FULL keywords are equivalent.

D. all of the above

Answer: B.

103. Which line in the following SQL statement raises an error?

```
1. SELECT name, title
2. FROM books b, publisher p
3. WHERE books.pubid = publisher.pubid
4. AND
5. (retail > 25 OR retail-cost > 18.95);
```

A. line 1

B. line 3

C. line 4

D. line 5

Answer: B. Since the tables used in the query have a qualifier, the columns must be referred using the same.

104. What is the maximum number of characters allowed in a table alias?

A. 10

B. 155

C. 255

D. 30

Answer: D. The table alias can be maximum of 30 characters.

105. Which of the following SQL statements is valid?

```
SELECT books.title, orderitems.quantity
FROM books b, orderitems o
WHERE b.isbn= o.ibsn;
```

```
SELECT title, quantity
FROM books b JOIN orderitems o;
```

```
SELECT books.title, orderitems.quantity
FROM books JOIN orderitems
ON books.isbn = orderitems.isbn;
```

D. none of the above

Answer: C.

Processing math: 100%