1.

a) rv\_refcur is a SYS\_REFCURSOR type variable, which is a pointer or reference to the result set returned by a query. It allows the program to iterate over the rows returned by the query.

b) emp\_details is a variable of type employee%ROWTYPE. This means it is a record that can hold an entire row from the employee table, matching the structure of that table.

c) The block opens a cursor rv\_refcur for the result set returned by the SELECT statement. It enters a loop where it fetches each row from the cursor into the emp\_details record. The loop continues until all rows are fetched. For each iteration, it prints the employee’s name Finally, it closes the cursor.

2.

DECLARE

rv\_refcur SYS\_REFCURSOR;

emp\_name employee.name%TYPE;

BEGIN

rv\_refcur := GETALL;

LOOP

FETCH rv\_refcur INTO emp\_name;

EXIT WHEN rv\_refcur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(emp\_name);

END LOOP;

CLOSE rv\_refcur;

END;

3.

The Parent row in TABX may be deleted: False.

Any child of a parent row in TABX may be deleted: True.

When a parent in TABX is deleted, each child is automatically deleted: False.

When a parent in TABX is deleted, each child’s FK is set to NULL: False.

When a child in TABY is deleted, the parent in TABX is automatically deleted: False.

4.

The Parent row in TABH may be deleted: True.

Any child of a parent row in TABH may be deleted: True.

When a parent in TABH is deleted, each child is automatically deleted: False.

When a parent in TABH is deleted, each child’s FK is set to NULL: True.

When a child in TABG is deleted, the parent in TABH is automatically deleted: False.

5.

The Parent row in TABX may be deleted: True.

Any child of a parent row in TABX may be deleted: True.

When a parent in TABX is deleted, each child is automatically deleted: True.

When a parent in TABX is deleted, each child’s FK is set to NULL: False.

When a child in TABY is deleted, the parent in TABX is automatically deleted: False.

6.

The TABC row with id C8 may be deleted: True, no effect on other rows.

The TABC row with id C9 may be deleted: True, no effect on other rows.

The TABB row with id B5 may be deleted: False

The TABB row with id B6 may be deleted: True, no effect on other rows .

The TABA row with id A1 may be deleted: False.

The TABA row with id A2 may be deleted: True, no effect on other rows.

7.

a) This deletes the row from the Enrollment table where enrolment ID = 906

b) This deletes the row from the Subject table where subId = 404

c) The delete action can not be done.

d) This makes the rows from the Subject table with foreign keys ID = 1, they will be set to Null

e) This delete the row from the student table with ID = 552

f) This delete the row where stuid = 553 from the Student table, and delete the rows from the enrolment table with stuId = 553

8.

It stores metadata about the data the database, including information about database schema, tables, columns, data types, constraints, relationships, and other database objects.

9.

c. "An Oracle data dictionary is a set of tables and views".

10.

No, it cannot be directly updated by a user or programmer.

11.

It is automatically updated by the database management system when changes are made to the database structure or schema object.

12.

USER\_OBJECTS shows objects owned by the user; ALL\_OBJECTS shows objects accessible to the user, including those owned by other users.

13.

No, you won’t see the same result set for both

14.

No, you cannot query the DD for all tables across all student accounts as access is restricted by user privileges.

15.

The DBMS checks if you have SELECT privileges on the DEPT table in the s1234567 schema by looking up the data dictionary views (ALL\_TAB\_PRIVS or USER\_TAB\_PRIVS). If you don't have permission, an error is raised.

The DBMS retrieves the column names and details from the data dictionary (ALL\_TAB\_COLUMNS or USER\_TAB\_COLUMNS) and includes all columns in the DEPT table in the query result.

16.

CREATE TABLE Student (

StuId NUMBER,

StuName VARCHAR2(100),

CONSTRAINT PK\_STUDENT\_STUID PRIMARY KEY (StuId),

CONSTRAINT NN\_STUDENT\_STUNAME CHECK (StuName IS NOT NULL)

););

17.

CREATE TABLE Branch (

BranchId NUMBER,

BranchName VARCHAR2(20),

CONSTRAINT PK\_BRANCH PRIMARY KEY (BranchId)

);

CREATE TABLE Employee (

EmpId NUMBER,

Firstname VARCHAR2(50),

Surname VARCHAR2(50),

Salary NUMBER,

BranchId NUMBER,

CONSTRAINT PK\_EMPLOYEE PRIMARY KEY (EmpId),

CONSTRAINT FK\_EMPLOYEE\_BRANCHID FOREIGN KEY (BranchId) REFERENCES Branch(BranchId),

CONSTRAINT CC\_EMPLOYEE\_SALARY CHECK (Salary > 0)

);

18.

CONSTRAINTS Table: 1 new row for the unique constraint

CONS\_COLUMNS Table: 2 new rows, one for each column

19.

TABLE Object: EMPLOYEE

TAB\_COLUMNS: EMPID, NAME, GENDER

CONSTRAINT: PK\_EMPLOYEE, NN\_EMPLOYEE\_GENDER, CC\_EMPLOYEE\_GENDER

CONS\_COLUMNS: EMPID, GENDER