PAGE:
DATE: / /

Q1 Elaborate the need of database views. Also explain situation where created views can be updated.

Ans :- A view is defined as a database object that allows us to create a virtual table in the database whose contents are defined by a query or taken from one or more table.

Need of views in database

views are generally used to focus, simplify and customize the perception each user has of the database.

Situation where created views can be updated.

(i) The views can be defined only one in table.
(ii) The select statement is used in view is not containt distinct keyword.
(iii) The views should not contain not Null value.
(iv) The views should not be created From nested or complex queries
(v) Select statement should not contain group by or order by clauses.
(vi) The view should not contain only Field which is made by aggregate function.
(vii) Any select o/p field of view must not use constant, strings or expression value.

Q2 Explain how constraints are defined in SQL - Primary key, Foreign key, Unique key.

Ans :- Primary key :-

A primary key is a unique identifier for each record (row) in a table.
It ensure that each row can be uniquely identified and accessed.
for example,

```
Create Table Tbl(
    ID int Primary Key,
    Name varchar(50),
    Age int(5),
    Address varchar(15));
```

foreign key :-

A foreign key is Field in a table that refers to the primary key in another table. It establishes a relationship between two tables on the value in these key
for Example,

```
Create Table Orders(
    OrderID INT Primary Key,
    ProductID int,
    Foreign key(ProductID) References
        Products(ProductID));
```

Unique key :-

A unique key ensures that each value in a column or group of column is unique and not

repeated in the table.

```
Create Table Employee(
    EmployeeID    int primary key,
    EmployeeName  varchar(10),
    Unique (EmployeeID),
    Unique (EmployeeName));
```

Q3 Write the PL/SQL procedure to check whether a number is even or odd.

Ans :-

```
Declare
P1 number = &num1;
Begin
If mod (n1, 2) = 0 then
DBMS_Output.PUT_Line('The number.' || n1 ||
  'is even number');

Else
DBMS_output.put_line('The number' || n1 ||
  'is odd number.');
END IF;
DBMS_Output.Put_Line(`Done Successfully`);
END;
```

Q4 Consider the Following database schema:

Emp (E_number, E_name, Dept_no)

Dept (Dept_no, Dept_name)

Address (Dept_name, Dept_location)

DATE : / /

Write SQL queries for following requirement
(i) Display the name of department for the employee having E_number 'E1011'.
(ii) Display the location of department where employee 'Ramesh' is working.
(iii) Display total no. of employee working in each department.

Ans :- (i)

```
Select Dept. Dept_name
From Emp Inner Join Dept ON
Emp. Dept_no = Dept. Dept_no where
Emp. E_number = 'E1011';
```

(ii)

```
Select Address . Dept_location
From Emp
Inner Join Dept ON Emp. Dept_no = Dept. Dept_no
Inner Join Address ON Dept. Dept_name = Address. Dept_name
Where Emp. E_name = 'Ramesh';
```

(iii)

```
Select Dept. Dept_name, COUNT (Emp. E_number)
As Total Employees
From Emp
Inner Join Dept ON Emp. Dept_no = Dept. Dept_no
Group By Dept. Dept_name;
```

PAGE:
DATE: / /

Qs Write PL/SQL block of code for following requirement:
Student_Fee(PRN, S_name, class, Fees_paid)
Accept the PRN of student from user, Check the fees paid by student, if fees paid is less than 30,000 then display the message on screen Not paid Full fees and display the total fees due. IF fees_paid is greater than or equal to 30,000 then display message no fees due.

DS
```
SET SERVEROUTPUT ON;
DECLARE
    A_PRN Varchar(20);
    A_Fees_paid Number;
    A_total_Fees Number = 30000;

Begin
    A_PRN = '&Enter_PRN';

Select fees_paid INTO A_Fees_paid From Student
_Fees Where PRN = A_PRN;

IF A_Fees_paid < A_total_Fees then
  DBms_Output.put_Line('Not paid Full Fees.');
DBMS_Output.put_Lines('Total Fees due:'||
(A_total_Fees - A_Fees_paid));

Else
```

```
        DBMS_output.put_Line('No Fees due.');
    END IF;
END;
```