**ST. FRANCIS INSTITUTE OF TECHNOLOGY** MT. POINSUR, BORIVALI (W), MUMBAI

**LAB MANUAL**

**EXPERIMENT NO. 7**

**Aim:** - Perform View and Triggers in SQL.

**Theory:-**

1. Views. Its advantages and dis advantages

2. Triggers.

**Lab Manual:**

**Views:** In database theory, a view is the result set of a stored query on the data, which the database users can query just as they would in a persistent database collection object. This pre-established query command is kept in the database dictionary.

**SQL CREATE VIEW Statement:**

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

**SQL CREATE VIEW Syntax:**

CREATE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition;

**Note:** A view always shows up-to-date data! The database engine recreates the data, using the view's SQL statement, every time a user queries a view.

**SQL CREATE VIEW Examples**

The view "Current Product List" lists all active products (products that are not discontinued) from the "Products" table. The view is created with the following SQL:

CREATE VIEW [Current Product List] AS

SELECT ProductID, ProductName

FROM Products

WHERE Discontinued=No;

We can query the view above as follows:

**ST. FRANCIS INSTITUTE OF TECHNOLOGY** MT. POINSUR, BORIVALI (W), MUMBAI

**LAB MANUAL**

SELECT \* FROM [Current Product List];

**SQL Updating a View:**

You can update a view by using the following syntax:

SQL CREATE OR REPLACE VIEW Syntax

CREATE OR REPLACE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition

**SQL Dropping a View:**

You can delete a view with the DROP VIEW command.

SQL DROP VIEW Syntax

DROP VIEW view\_name;

**Triggers:**

Triggers are similar to stored procedures. A trigger stored in the database can include SQL and PL/SQL or Java statements to run as a unit and can invoke stored procedures. However, procedures and triggers differ in the way that they are invoked. A procedure is explicitly run by a user, application, or trigger. Triggers are implicitly fired by Oracle when a triggering event occurs, no matter which user is connected or which application is being used. A trigger can also call out to a C procedure, which is useful for computationally intensive operations. The events that fire a trigger include the following:

∙ DML statements that modify data in a table (INSERT, UPDATE, or DELETE) ∙ DDL statements

∙ System events such as startup, shutdown, and error messages

∙ User events such as logon and logoff

**How Triggers Are Used**

Triggers supplement the standard capabilities of Oracle to provide a highly customized database management system. For example, a trigger can restrict DML operations against a table to those issued during regular business hours. You can also use triggers to:

**ST. FRANCIS INSTITUTE OF TECHNOLOGY** MT. POINSUR, BORIVALI (W), MUMBAI

**LAB MANUAL**

∙ Automatically generate derived column values

∙ Prevent invalid transactions

∙ Enforce complex security authorizations

∙ Enforce referential integrity across nodes in a distributed database

∙ Enforce complex business rules

∙ Provide transparent event logging

∙ Provide auditing

∙ Maintain synchronous table replicates

∙ Gather statistics on table access

∙ Modify table data when DML statements are issued against views

∙ Publish information about database events, user events, and SQL statements to subscribing applications

There are three components in trigger

Event: When this event happens, the trigger is activated

Condition (optional): If the condition is true, the trigger executes, otherwise skipped Action: The actions performed by the trigger

Semantics: When the Event occurs and Condition is true, execute the Action

**Trigger syntax:**

CREATE TRIGGER <triggerName>

BEFORE|AFTER INSERT|DELETE|UPDATE

[OF <columnList>] ON <tableName>|<viewName>

[REFERENCING [OLD AS <oldName>] [NEW AS <newName>]]

[FOR EACH ROW] (default is “FOR EACH STATEMENT”)

[WHEN (<condition>)]

< trigger body>;

In SQL\*Plus, you can also use the following shortcut to view compilation errors: SQL> **SHOW ERRORS TRIGGER** MY\_TRIGGER

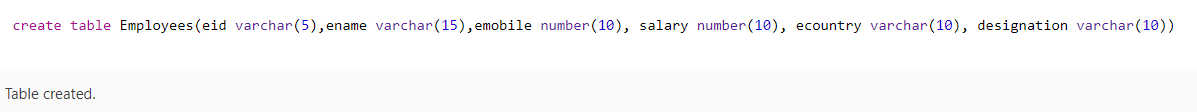
**ST. FRANCIS INSTITUTE OF TECHNOLOGY**

MT. POINSUR, BORIVALI (W),

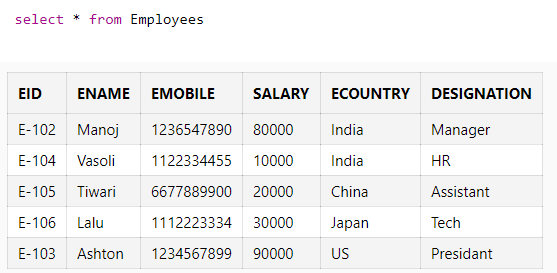
MUMBAI

**Lab Exercise on Views:-**

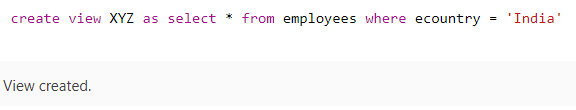
1. Create table Employee having eid varchar (5), ename varchar(15), emobile number(10), salary number(10), ecountry varchar(10), designation varchar(10).



1. Insert 5 values inside the Employee table.

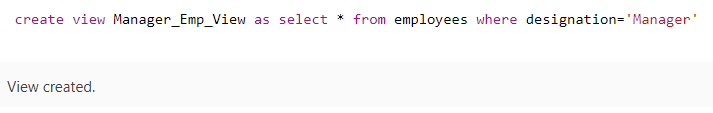


3. Create view India\_emp\_view, list all from Employee table where ecountry=’India’

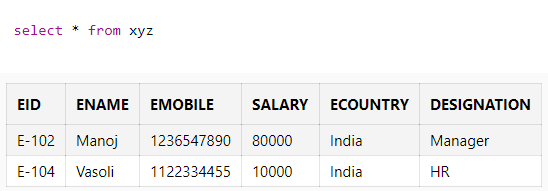


4. Create view Manager\_emp\_view, list all details for employee where designation is

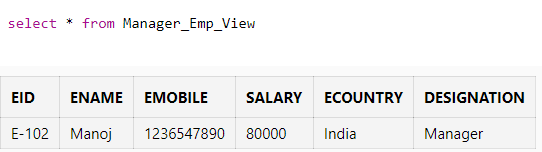
manager.



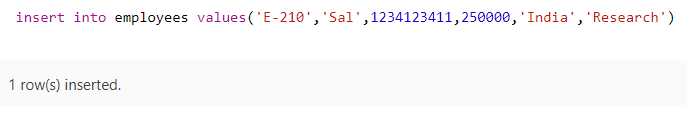
5. Display India\_emp\_view

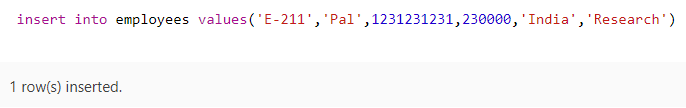


6. Display Manager\_emp\_view.

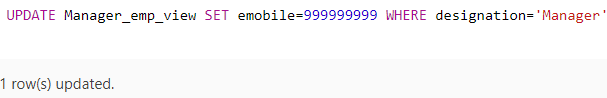


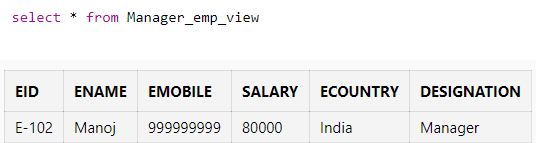
7. Insert 2 tuples in Employee where country of employee is India, and then check no. of records in India\_emp\_view.





8. Update Manager\_emp\_view, update its emobile, and then check the record in Employee table.

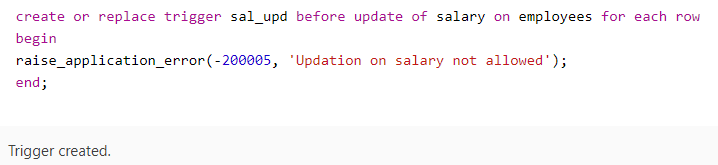




**Lab Exercise on Triggers:-**

1. For Relational Schema Employee (Eid, ename, emobile, salary, ecountry, designation),create following triggers:

a. Write a trigger to avoid updating on Salary attribute for employee relation.



b. Write a trigger to avoid insert on employee relation on Weekends.

SQL> create or replace trigger inst\_values

  2  before insert on Employees

  3  for each row

  4  begin

  5  if(TO\_CHAR(SYSDATE,'dy')IN ('sat','sun'))

  6  raise\_application\_error(-20500,'Cannot insert values today!!');

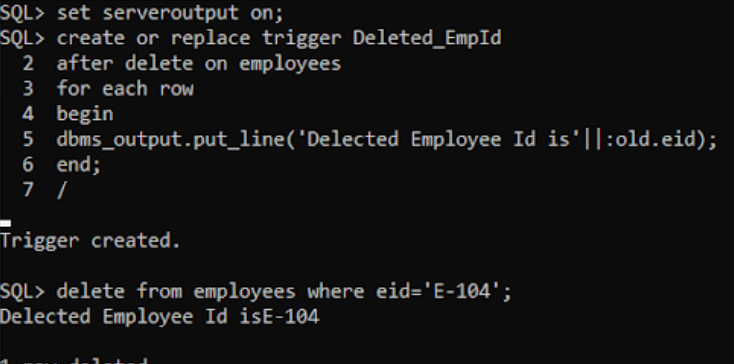
  7  endif;

  8  end;

  9  /

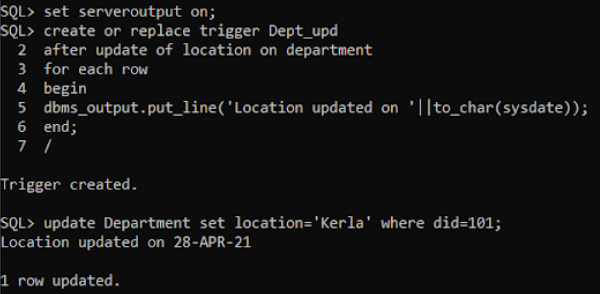
Trigger created.

c. Write a trigger that displays the employee id for the record which gets deleted.

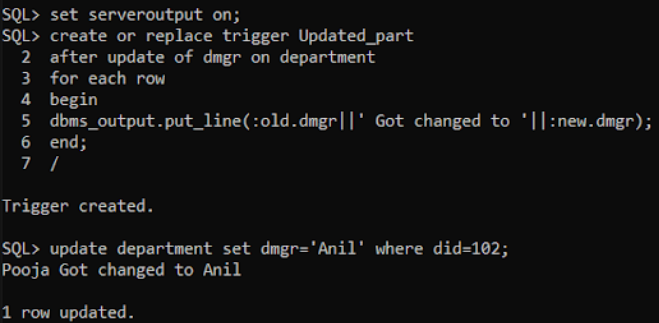


2. For Relational Schema Department (Did, Dname, Location,Dmgr),create following triggers:

a. Write a trigger that displays the system date whenever there is an update on Location attribute for department relation.



b. Write a trigger that outputs a statement stating old name which got updated by the new name whenever the Dmgr gets updated for department relation.



**Conclusion:**

Views and triggers are created in the above experiment. Views in SQL are a kind of virtual table. A View can either have all the rows of a table or specific rows based on certain conditions. Triggers are stored programs, which are automatically executed or fired when some events occur.