Experiment 8

Aim: Write a PL/SQL block to create Triggers for update, delete and insert

Resources required: P-IV and above, Oracle

Theory:

View

A **view** is a virtual table, which provides access to a subset of column from one or more table. A **view** can derive its data from one or more table. An output of query can be stored as a **view**. ... A **view in oracle** is nothing but a stored sql scripts.

create view v11 as (select empid, ename ,dname from emp,dept where emp.dno=dept.dno);

Views may be created for the following reasons:

1. The DBA stores the views as a definition only. Hence there is no duplication of data.
2. Simplifies Queries.
3. Can be Queried as a base table itself.
4. Provides data security.
5. Avoids data redundancy.

Creation of Views:-

Syntax:-

CREATE VIEW viewname AS
SELECT columnname,columnname
FROM tablename
WHERE columnname=expression_list;

Renaming the columns of a view:-

Syntax:-

CREATE VIEW viewname AS
SELECT newcolumnname….
FROM tablename
WHERE columnname=expression_list;

Selecting a data set from a view-

Syntax:-

SELECT columnname, columnname

FROM viewname

WHERE search condition;

Destroying a view-

Syntax:-

DROP VIEW viewname;

Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events −

- A **database manipulation (DML)** statement (DELETE, INSERT, or UPDATE)
- A **database definition (DDL)** statement (CREATE, ALTER, or DROP).
- A **database operation** (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

Triggers can be written for the following purposes −

- Generating some derived column values automatically
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

The syntax for creating a trigger is −

CREATE [OR REPLACE] TRIGGER trigger_name

{BEFORE | AFTER | INSTEAD OF }

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF col_name]

```
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
   Declaration-statements
BEGIN
   Executable-statements
EXCEPTION
   Exception-handling-statements
END;
```

Where,

- CREATE [OR REPLACE] TRIGGER trigger_name − Creates or replaces an existing trigger with the *trigger_name*.
- {BEFORE | AFTER | INSTEAD OF} − This specifies when the trigger will be executed. The INSTEAD OF clause is used for creating trigger on a view.
- {INSERT [OR] | UPDATE [OR] | DELETE} − This specifies the DML operation.
- [OF col_name] − This specifies the column name that will be updated.
- [ON table_name] − This specifies the name of the table associated with the trigger.
- [REFERENCING OLD AS o NEW AS n] − This allows you to refer new and old values for various DML statements, such as INSERT, UPDATE, and DELETE.
- [FOR EACH ROW] − This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected. Otherwise the trigger will execute just once when the SQL statement is executed, which is called a table level trigger.
- WHEN (condition) − This provides a condition for rows for which the trigger would fire. This clause is valid only for row-level triggers.

SQL>create table customer(id number(2), name varchar2(15), age number(2), address varchar2(20), salary number(9,2));

```
SQL>create table customer_log(id number(2), name varchar2(15), age number(2), address
varchar2(20), salary number(9,2), usern varchar2(10), dt date, oper varchar2(10));
create or replace trigger cust_log
  after update or delete or insert on customer1
     for each row
        declare
        op varchar2(10);
        begin
        if updating then
        op:='update';
       insert into customer1_log(id,name,age,address,salary,dt,oper) values(:old.id, :old.name,
:old.age, :old.address, :old.salary,current_timestamp, op);
      end if;
      if deleting then
      op:='delete';
      insert into customer1_log(id,name,age,address,salary,dt,oper) values(:old.id, :old.name,
:old.age, :old.address, :old.salary,current_timestamp, op);
      end if;
      if inserting then
      op:='insert';
      insert into customer1_log(id,name,age,address,salary,dt,oper) values(:new.id, :new.name,
:new.age, :new.address, :new.salary,current_timestamp, op);
      end if;
   end;

SQL> insert into customer1(id,salary) values(9,10000);

1 row created.
```

Conclusion: We have successfully Implemented Views and Triggers in MySQL Command Line.

Code and Output:

Views:

```
MySQL 8.0 Command Line Client - Unicode

mysql> select * from invoices;
+------------+--------------+-----------+---------------+---------------+--------------+
| invoice_id | number       | client_id | invoice_total | payment_total | invoice_date |
+------------+--------------+-----------+---------------+---------------+--------------+
|          2 | 828-863-4354 |         3 |        500.97 |          0.00 | 2021-02-28   |
|          3 | 992-113-4556 |         1 |       2500.97 |          0.00 | 2021-03-01   |
|          4 | 992-113-4556 |         1 |       5500.97 |          0.00 | 2021-03-01   |
|          5 | 828-863-4354 |         3 |        600.45 |          0.00 | 2021-03-02   |
+------------+--------------+-----------+---------------+---------------+--------------+
4 rows in set (0.01 sec)

mysql>  create view invoice_v1 as (select invoice_id,number,invoice_total from invoices);
Query OK, 0 rows affected (0.03 sec)

mysql> select * from invoice_v1;
+------------+--------------+---------------+
| invoice_id | number       | invoice_total |
+------------+--------------+---------------+
|          2 | 828-863-4354 |        500.97 |
|          3 | 992-113-4556 |       2500.97 |
|          4 | 992-113-4556 |       5500.97 |
|          5 | 828-863-4354 |        600.45 |
+------------+--------------+---------------+
4 rows in set (0.01 sec)

mysql> update invoices_v1 set invoice_total = 50.97 WHERE invoice_id = 2;
ERROR 1146 (42S02): Table 'practical3.invoices_v1' doesn't exist
mysql> update invoice_v1 set invoice_total = 50.97 WHERE invoice_id = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from invoices;
+------------+--------------+-----------+---------------+---------------+--------------+
| invoice_id | number       | client_id | invoice_total | payment_total | invoice_date |
+------------+--------------+-----------+---------------+---------------+--------------+
|          2 | 828-863-4354 |         3 |         50.97 |          0.00 | 2021-02-28   |
|          3 | 992-113-4556 |         1 |       2500.97 |          0.00 | 2021-03-01   |
|          4 | 992-113-4556 |         1 |       5500.97 |          0.00 | 2021-03-01   |
|          5 | 828-863-4354 |         3 |        600.45 |          0.00 | 2021-03-02   |
+------------+--------------+-----------+---------------+---------------+--------------+
4 rows in set (0.00 sec)
```

```
MySQL 8.0 Command Line Client - Unicode

mysql> select * from invoice_v1;
+------------+--------------+----------------+
| invoice_id | number       | invoice_total  |
+------------+--------------+----------------+
|          2 | 828-863-4354 |          50.97 |
|          3 | 992-113-4556 |        2500.97 |
|          4 | 992-113-4556 |        5500.97 |
|          5 | 828-863-4354 |         600.45 |
+------------+--------------+----------------+
4 rows in set (0.00 sec)

mysql> DROP view invoice_v1;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from invoice_v1;
ERROR 1146 (42S02): Table 'practical3.invoice_v1' doesn't exist
```

Triggers:

```
mysql> select * from clients;
+-----------+----------+-------------+-------------+-------+--------------+
| client_id | name     | address     | city        | state | phone        |
+-----------+----------+-------------+-------------+-------+--------------+
|         1 | adnan    | Jamal Nagar | MUmbai      | MH    | 992-113-4556 |
|         2 | Shanmont | damo        | New Delhi   | DL    | 828-863-4354 |
|         3 | Shubham  | Khargar     | Navi-Mumbai | MH    | 828-863-4354 |
+-----------+----------+-------------+-------------+-------+--------------+
3 rows in set (0.01 sec)

mysql> create table clients_log(client_id int(10),name varchar(50),address varchar(256),city varchar(20),state char(2),phone varchar(20),operation varchar(20)
);
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> DELIMITER $$
```

```
MySQL 8.0 Command Line Client - Unicode
mysql> DELIMITER $$
mysql>  CREATE TRIGGER cl_log_insert
    -> AFTER INSERT ON clients
    -> FOR each row
    -> BEGIN
    -> INSERT INTO clients_log VALUES(NEW.client_id,NEW.name,NEW.address,NEW.city,NEW.state,NEW.phone,'inserted');
    -> END$$
Query OK, 0 rows affected (0.02 sec)

mysql>  CREATE TRIGGER cl_log_delete
    -> AFTER DELETE ON clients
    -> FOR each row
    -> BEGIN
    -> INSERT INTO clients_log VALUES(OLD.client_id,OLD.name,OLD.address,OLD.city,OLD.state,OLD.phone,'deleted');
    -> END $$
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TRIGGER cl_log_update
    -> AFTER UPDATE ON clients
    -> FOR each row
    -> BEGIN
    -> INSERT INTO clients_log VALUES(OLD.client_id,OLD.name,OLD.address,OLD.city,OLD.state,OLD.phone,'updated');
    -> END$$
Query OK, 0 rows affected (0.03 sec)

mysql> DELIMITER ;
```

MySQL 8.0 Command Line Client - Unicode

```
mysql> DELIMITER ;
mysql> INSERT into clients(4,'Safwan','Near Bank Of Plaza Chembur','Mumbai','MH','666-999-4200');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '4,'
Safwan','Near Bank Of Plaza Chembur','Mumbai','MH','666-999-4200')' at line 1
mysql> INSERT into clients values(4,'Safwan','Near Bank Of Plaza Chembur','Mumbai','MH','666-999-4200');
Query OK, 1 row affected (0.01 sec)

mysql> update clients SET phone = '568-777-3200' WHERE name='adnan';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> delet from clients where name='Shubham';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'del
et from clients where name='Shubham'' at line 1
mysql> delete from clients where name='Shubham';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`practical3`.`invoices`, CONSTRAINT `FK_client_id` FOREIGN KEY (`cli
ent_id`) REFERENCES `clients` (`client_id`) ON DELETE RESTRICT ON UPDATE CASCADE)
mysql> select * from invoices;
+------------+--------------+-----------+---------------+---------------+--------------+
| invoice_id | number       | client_id | invoice_total | payment_total | invoice_date |
+------------+--------------+-----------+---------------+---------------+--------------+
|          2 | 828-863-4354 |         3 |        500.97 |          0.00 | 2021-02-28   |
|          3 | 992-113-4556 |         1 |       2500.97 |          0.00 | 2021-03-01   |
|          4 | 992-113-4556 |         1 |       5500.97 |          0.00 | 2021-03-01   |
|          5 | 828-863-4354 |         3 |        600.45 |          0.00 | 2021-03-02   |
+------------+--------------+-----------+---------------+---------------+--------------+
4 rows in set (0.01 sec)

mysql> delete from clients where name='Shanmont';
Query OK, 1 row affected (0.01 sec)

mysql> select * from clients_log;
+-----------+----------+---------------------------+-----------+-------+--------------+-----------+
| client_id | name     | address                   | city      | state | phone        | operation |
+-----------+----------+---------------------------+-----------+-------+--------------+-----------+
|         4 | Safwan   | Near Bank Of Plaza Chembur | Mumbai    | MH    | 666-999-4200 | inserted  |
|         1 | adnan    | Jamal Nagar               | MUmbai    | MH    | 992-113-4556 | updated   |
|         2 | Shanmont | damo                      | New Delhi | DL    | 828-863-4354 | deleted   |
+-----------+----------+---------------------------+-----------+-------+--------------+-----------+
3 rows in set (0.01 sec)
```