Ex.No.: 13		WORKING WITH TRIGGER TRIGGER
Date:		

DEFINITION

A trigger is a statement that is executed automatically by the system as a side effect of a modification to the database. The parts of a trigger are,

- **Trigger statement**: Specifies the DML statements and fires the trigger body. It also specifies the table to which the trigger is associated.

- **Trigger body or trigger action**: It is a PL/SQL block that is executed when the triggering statement is used.

- **Trigger restriction**: Restrictions on the trigger can be achieved

The different uses of triggers are as follows,

- *To generate data automatically*
- *To enforce complex integrity constraints*
- *To customize complex securing authorizations*
- *To maintain the replicate table*
- To audit data modifications

TYPES OF TRIGGERS

The various types of triggers are as follows,

- **Before**: It fires the trigger before executing the trigger statement.

- **After**: It fires the trigger after executing the trigger statement
- .
- **For each row**: It specifies that the trigger fires once per row
- .
- **For each statement**: This is the default trigger that is invoked. It specifies that the trigger fires once per statement.

VARIABLES USED IN TRIGGERS

- :new

- :old

Program 1

Write a code in PL/SQL to develop a trigger that enforces referential integrity by preventing the deletion of a parent record if child records exist.

```
CREATE OR REPLACE TRIGGER Prevent_parent_deletion
BEFORE DELETE ON PARENT
  FOR EACH ROW
DECLARE
    child_count NUMBER;
  BEGIN
SELECT COUNT (*) INTO Child_Count FROM childWHERE
Parent_id
  IF child_count > THEN RAISE_APPLICATION_ERROR
  END;
/
```

Program 2

Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found.

```
CREATE TABLE SampleTable (
  id NUMBER (5) primary key,
  name varchar (50) null,
  email VARCHAR 2 (100) UNIQUE );
CREATE OR REPLACE TRIGGER check_duplicate_email
BEFORE INSERT OR UPDATE ON SampleTable
FROM EACH ROW
DECLARE
      duplicate_count NUMBER
BEGIN
   SELECT COUNT (*) INTO duplicate_count
   END IF;
END;
/
```

Program 3

Write a code in PL/SQL to create a trigger that restricts the insertion of new rows if the total of a column's values exceeds a certain threshold.

```
CREATE OR REPLACE TRIGGER restrict_total_sales
BEFORE INSERT ON Sales
For each ROW
 Begin
IF (select sum (amount) from Sales) + :
New amount > 1000000
 Raise_application_Error (-20002), 'Total exceeds three
                                    hold.');
 END IF;
 END;
/
```

Program 4

Write a code in PL/SQL to design a trigger that captures changes made to specific columns and logs them in an audit table.

```
create OR Replace Trigger log_salary_changes
after update of Salary on employees
For each row
Begin
Insert Into EmployeeAudit if VALUES (audit_seq.
                                    NEXTVAL, :OLD.
emp_id, :OLD : salary, :NEW.salary, SYSDATE);
 END;
/
```

Program 5

Write a code in PL/SQL to implement a trigger that records user activity (inserts, updates, deletes) in an audit log for a given set of tables.

```
create or Replace Trigger record_user activity
after insert OR update or delete on Employees for
each row
BEGIN
Insert into Audit Log values (audit seq.NEXTVAL,
case when Inserting then 'INSERT' when updating Then 'UPDATE',
'Employees', NVL(:old.emp_id,:NEW.emp_id),
SYSDATE, USER);
END;
/
```

Program 7

Write a code in PL/SQL to implement a trigger that automatically calculates and updates a running total column for a table whenever new rows are inserted.

```
create Table Sales (
        sale_id number primary key,
amount number (10,2), running_total number (10,2)),

Create or Replace Trigger update_running total
For each row
Begin
select NVL (max (running_total,0) + :NEW.amount
                INTO :NEW.running

END;
/
```

Program 8

Write a code in PL/SQL to create a trigger that validates the availability of items before allowing an order to be placed, considering stock levels and pending orders.

```
CREATE OR Replace trigger Validate_stock_Before_order
Before Insert on order
    For each row
Begin
If :new.order_quantity > (select Stock_quantity
  From items
  Where item_id = :NEW.item_id
    END IF;
  END;
/
```

Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	

Ex.No.: 14		MONGO DB
Date:		

MongoDB is a free and open-source cross-platform document-oriented database. Classified as a NoSQL database, MongoDB avoids the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas, making the integration of data in certain types of applications easier and faster.

Create Database using mongosh

After connecting to your database using mongosh, you can see which database you are using by typing db in your terminal.

If you have used the connection string provided from the MongoDB Atlas dashboard, you should be connected to the myFirstDatabase database.

Show all databases

To see all available databases, in your terminal type show dbs.

Notice that myFirstDatabase is not listed. This is because the database is empty. An empty database is essentially non-existant.

Change or Create a Database

You can change or create a new database by typing use then the name of the database.

Create Collection using mongosh

You can create a collection using the createCollection() database method.

Insert Documents

insertOne()

```
db.posts.insertOne({
 title: "Post Title 1",
 body: "Body of post.",
 category: "News",
 likes: 1,
 tags: ["news", "events"],
```

1. db. restaurants. find (
{ $or : [{ cuisine : { $ne: 'American' } },
{ cuisine : { $ne: 'chinese' } },
{ name: { $ regex: /^Wil/i } }
};]
{ restaurant : 1 , name : 1, borough : 1,
cuisine : 1 }
);

2. db. restaurants. find (
{ grades: {
$ elemMatch : {
grade : "A",
score : 11,
date : { $ gte : ISODate ("2014-08-11T00:00:00Z"),
$ lt : ISODate ("2014-08-12T00:00:00Z") }
}}
};
{ restaurant_id : 1, name : 1, grades : 1 }
);

3. db. restaurants. find (
{ grades: {
$ elemMatch : {
$ expr : { $ eq : [{ $arrayElemAt : ["$ grades. grade", 1] }, "A"] },
$ expr : { $eq : [{ $arrayElemAt : [" $ grades. grade ", 1]} , 9 }.
}}
};
{ restaurant_id : 1, name : 1, grades : 1 }
);

4. db.restaurants.find(
{ "address.coord.1": { $gt: 42, $lte: 52 }
},
{restaurant_id: 1, name: 1, address: 1, "address.coord": 1 }
);

5. db.restaurants.find().sort({name: 1});

6. db.restaurants.find().sort({name: -1});

7. db.restaurants.find().sort({cuisine: 1, borough: -1});

8. db.restaurants.find(
{"address.street": { $exists: true, $ne: "" }}
);

9. db.restaurants.find(
{ "address.coord": { $type: "double" }
});

10.

10. db.restaurants.find(
{ grades: { $elemMatch: { score: { $mod: [7,0] } } } },
{ restaurant_id: 1, name: 1, grades: 1 }
);

11. db.restaurants.find({ name: { $regex: /mon/i } } ,
{ name: 1, borough: 1, "address.coord": 1, cuisine: 1 });

12. db.restaurants.find({ name: { $regex: /^Mad/i } } ,
{ name: 1, borough: 1, "address.coord": 1, cuisine: 1 }
);

13. db.restaurants.find(
{ grades: { $elemMatch: { score: { $lt: 5 } } } }
);

14. db.restaurants.find({
borough: "Manhattan",
grades: { $elemMatch: { score: { $lt: 5 } } }
}
);

15. db.restaurants.find(
{ $or: [
{ borough: "Manhattan", grades: { $elemMatch: { score:
{ $lt: 5 }}}},
{ borough: "Brooklyn", grades: { $elemMatch: { score:
{ $lt: 5 }}}
]
});

16. db.restaurants.find(
{ $or: [
{ borough: "Manhattan", grades: { $elemMatch:
{ score: { $lt: 5 }}}}
{ borough: "Brooklyn", grades: { $elemMatch: { score:
{ $lt: 5 }}}
],
cuisine: { $ne: "American" }
});

17. db.restaurants.find(
{ $or: [{ borough: "Manhattan", grades: { $elemMatch
: { score: { $lt: 5 }}}},
{ borough: "Brooklyn", grades: { $elemMatch: { score:
{ $lt: 5 }}}
],
cuisine: { $nin: ["American", "Chinese"] }
});

18. db.restaurants.find(
{ "grades.score": { $all: [2,4] }}
)

19. db.restaurants.find({
borough: "Manhattan",
grades: {$all: [{score: 2}, {score: 6}]}});

20. db.restaurants.find({
$or: [{borough: "Manhattan"}, {borough: "Brooklyn"}];
grades: {$all: [{score: 2}, {score: 6}]}
});

21. db.restaurants.find({
$or: [{borough: "Manhattan"}, {borough: "Brooklyn"}],
cuisine: {$ne: "American"},
grades: {$all: [{score: 2}, {score: 6}]}
});

22. db.restaurants.find ({
$or : [{borough: "Manhattan"}; {borough: "Brooklyn"}];
cuisine : {$in: ["American", "chinese"]},
grades: {$all: [{score: 2}, {score: 6}]}
});

23. db.restaurants.find ({
grades: {$elemMatch : {$or: [{score: 2},
{score: 6}]}}
});

poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
title: 'The Great Train Robbery',

fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",

languages: ['English'],
released: ISODate("1903-12-01T00:00:00.000Z"),
directors: ['Edwin S. Porter'],
rated: 'TV-G',
awards: { wins: 1, nominations: 0, text: '1 win.' },
lastupdated: '2015-08-13 00:27:59.177000000',
year: 1903,
imdb: { rating: 7.4, votes: 9847, id: 439 },
countries: ['USA'],
type: 'movie',
tomatoes: {
viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
fresh: 6,
critic: { rating: 7.6, numReviews: 6, meter: 100 },
rotten: 0,
lastUpdated: ISODate("2015-08-08T19:16:10.000Z")
}

1.Find all movies with full information from the 'movies' collection that released in the year 1893.

db.movies.find({year:1893});

2. Find all movies with full information from the 'movies' collection that have a runtime greater than 120 minutes.

db.movies.find({runtime:{$gt:120}});

3. Find all movies with full information from the 'movies' collection that have "Short" genre.

db.movies.find({genres:"Short"});

4. Retrieve all movies from the 'movies' collection that were directed by "William K.L. Dickson" and include complete information for each movie.

db.movies.find({directors:"William K.L. Dickson"});

6. Retrieve all movies from the 'movies' collection that were released in the USA and include complete information for each movie.

db.movies.find({countries:"USA"});

7. Retrieve all movies from the 'movies' collection that have complete information and are rated as "UNRATED".

db.movies.find({rated:"UNRATED"});

8. Retrieve all movies from the 'movies' collection that have complete information and have received more than 1000 votes on IMDb.

db.movies.find({"imdb.votes": {$gt: 1000}});

9. Retrieve all movies from the 'movies' collection that have complete information and have an IMDb rating higher than 7.

db.movies.find({"imdb.rating": {$gt: 7}});

10. Retrieve all movies from the 'movies' collection that have complete information and have a viewer rating higher than 4 on Tomatoes.

db.movies.find({"tomatoes.viewer.rating": {$gt: 4}});

11. Retrieve all movies from the 'movies' collection that have received an award.

db.movies.find({"award.wins": {$gt: 0}});

12. Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB that have at least one nomination.

db.movies.find({"award.nominations": {$gt: 0
{title: 1, languages: 1, released: 1, directors: 1,
writers: 1, awards: 1, year: 1, genres: 1,
runtime: 1, cast: 1, countries: 1}
);

13. Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB with cast including "Charles Kayser".

db.movies.find({ cast: "charles kayser" },
{title: 1, language: 1, released: 1, directors: 1, writers: 1,
awards: 1, year: 1, genres: 1, runtime: 1, cast: 1,
countries: 1 }
);

14. Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that released on May 9, 1893.

db.movies.find({ released: ISODate("1893-05-09T00:00:00.00Z") },
{title: 1, language: 1, released: 1, directors: 1, writers: 1,
countries: 1 }
);

14. Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that have a word "scene" in the title.

db.movies.find({ title: { $regex: /scene/i } },
{title: 1, language: 1, released: 1,
directors: 1, writers: 1,
countries: 1 }
);

Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	

Ex.No.: 15		OTHER DATABASE OBJECTS
Date:		

OTHER DATABASE OBJECTS

Objectives

After the completion of this exercise, the students will be able to do the following:
- Create, maintain, and use sequences
- Create and maintain indexes

Database Objects

Many applications require the use of unique numbers as primary key values. You can either build code into the application to handle this requirement or use a sequence to generate unique numbers.
If you want to improve the performance of some queries, you should consider creating an index. You
can also use indexes to enforce uniqueness on a column or a collection of columns.
You can provide alternative names for objects by using synonyms.

What Is a Sequence?

A sequence:
- Automatically generates unique numbers
- Is a sharable object
- Is typically used to create a primary key value
- Replaces application code
- Speeds up the efficiency of accessing sequence values when cached in memory

The CREATE SEQUENCE Statement Syntax

Define a sequence to generate sequential numbers automatically:

CREATE SEQUENCE *sequence*
[INCREMENT BY *n*]
[START WITH *n*]
[{MAXVALUE *n* | NOMAXVALUE}]
[{MINVALUE *n* | NOMINVALUE}]
[{CYCLE | NOCYCLE}]
[{CACHE *n* | NOCACHE}];

In the syntax:

sequence is the name of the sequence generator

1. create sequence Dept_id_seq
start with 200
Increment by 10
MaxValue 1000
Nocycle ;

2. Select
sequence_name,
max_value,
increment_by,
last_number
FROM
user_sequences
WHERE
sequence_name = 'DEPT_ID_SEQ';

3. INSERT into Dept (ID, Name) values (Dept_id_seq.Nextval,
'Education'),.
INSERT into Dept (ID, Name) values (Dept_id_seq.Nextval,
'Administration'),
Select * From Dept;

4. create Index IDX_DEPT_ID ON Emp (Dept_

5. SELECT
index_name,
uniqueness

FROM
user_indexes

WHERE
table_name = 'EMP';

3. Write a script to insert two rows into the DEPT table. Name your script lab12_3.sql. Be sure to use the sequence that you created for the ID column. Add two departments named Education and
Administration. Confirm your additions. Run the commands in your script.
4. Create a nonunique index on the foreign key column (DEPT_ID) in the EMP table.

5. Display the indexes and uniqueness that exist in the data dictionary for the EMP table.

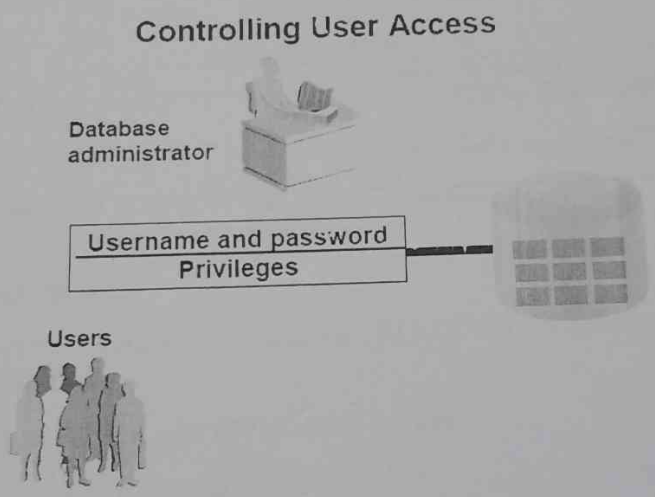
Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	

Ex.No.: 16		CONTROLLING USER ACCESS
Date:		

Objectives

After the completion of this exercise, the students will be able to do the following:

- Create users
- Create roles to ease setup and maintenance of the security model
- Use the GRANT and REVOKE statements to grant and revoke object privileges
- Create and access database links



Controlling User Access

In a multiple-user environment, you want to maintain security of the database access and use. With Oracle server database security, you can do the following:

- Control database access
- Give access to specific objects in the database
- Confirm given and received *privileges* with the Oracle data dictionary
- Create synonyms for database objects

Privileges

- Database security:
  - System security
  - Data security

Find the Solution for the following:

1. What privilege should a user be given to log on to the Oracle Server? Is this a system or an object privilege?

A user should be given the create Session privelege to log on to the oracle server. This is a System privelege.

2. What privilege should a user be given to create tables?

A user should be given the CREATE table privelege to create tables.

3. If you create a table, who can pass along privileges to other users on your table?

_____

4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?

_____

5. What command do you use to change your password?

_____

6. Grant another user access to your DEPARTMENTS table. Have the user grant you query access
to his or her DEPARTMENTS table.

7. Query all the rows in your DEPARTMENTS table.

8. Add a new row to your DEPARTMENTS table. Team 1 should add Education as department number 500. Team 2 should add Human Resources department number 510. Query the other team's table.

9. Query the USER_TABLES data dictionary to see information about the tables that you own.

10. Revoke the SELECT privilege on your table from the other team.

11. Remove the row you inserted into the DEPARTMENTS table in step 8 and save the changes.

9. Select * From User_Tables;

10. Revoke select on Departments From other_user;

11. Delete From Departments where Department_id = 500;
commit;

2. The owner of the table (the user who created) can pass along privileges to other users on that table. The owner can use GRANT statement to grant privelges.

4. As the DBA you shoud use roles to grant multiple users the same system privileges. This simplifies management by allowing you to assign a set of privelges to a role and grant that role to users.

5. Alter user username identified by new_password;

6. Grant select on Departments to other_users;
(your)

(other)
Grant select on their Departments to your username.

7. Select * from Departments;
(Team 1)

8. Insert into Departments (Department_id, Department_name) values (500, 'Education');
(Team 2)

Insert into Departments (Department_id, Deptname) values (510, 'HR');
(others team)

Select * from other_team_Departments;

Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	

Completed