East West University
Department of Computer Science and Engineering

CSE 301: LAB 10 [Assessed Lab]
Course Instructor: Dr. Mohammad Rezwanul Huq

Oracle APEX

Lab Objective

Familiarize students with a few intermediate application development options in Oracle APEX – a tool to develop oracle-based database application.

Lab Outcome

After completing this lab successfully, students will be able to:

1. Create and Use Custom Authentication.
2. Create an access control page.
3. Create a navigation bar entry.
4. Create a region in a page and insert an image in a page using shared component.
5. **Create and understand basic triggers.**

Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

Instructions

- Follow the instructor during the class.
- A simple step-by-step tutorial can be found in this link. https://goo.gl/2m2Vnr
- A series of video tutorial on the same project can be found here: https://goo.gl/BT93w4
- A tutorial for custom authentication and authorization can be found here: https://goo.gl/fAWabg

Lab Activities

1. **Log into your existing Oracle APEX account.**
2. **Create a table player and country with the following attributes.**

Player	Country
player_id (number, primary key)	country_code (number, primary key)
player_name (varchar2)	country_name (varchar2)
country_code (number, foreign key)	file_lob (blob)
file_lob (blob)	file_name (varchar2)
file_name (varchar2)	file_mimetype (varchar2)
file_mimetype (varchar2)	file_updatedate (varchar2)
file_updatedate (varchar2)	file_characterset (varchar2)
file_characterset (varchar2)	

3. **Creating appropriate sequences.**

```
Create sequence <sequence_name> start with <value> increment by <value>;
```

4. **Create appropriate authentication and an access control page**

   i) Create a my_users table. This table includes the users who can access this application.

```
create table my_users (
user_id number,
user_name varchar2(20),
user_password varchar2(20),
user_activated number default 0,
primary key(user_id));
```

ii) Creating a sequence

```
create sequence my_users_seq start with 30001 increment by 1;
```

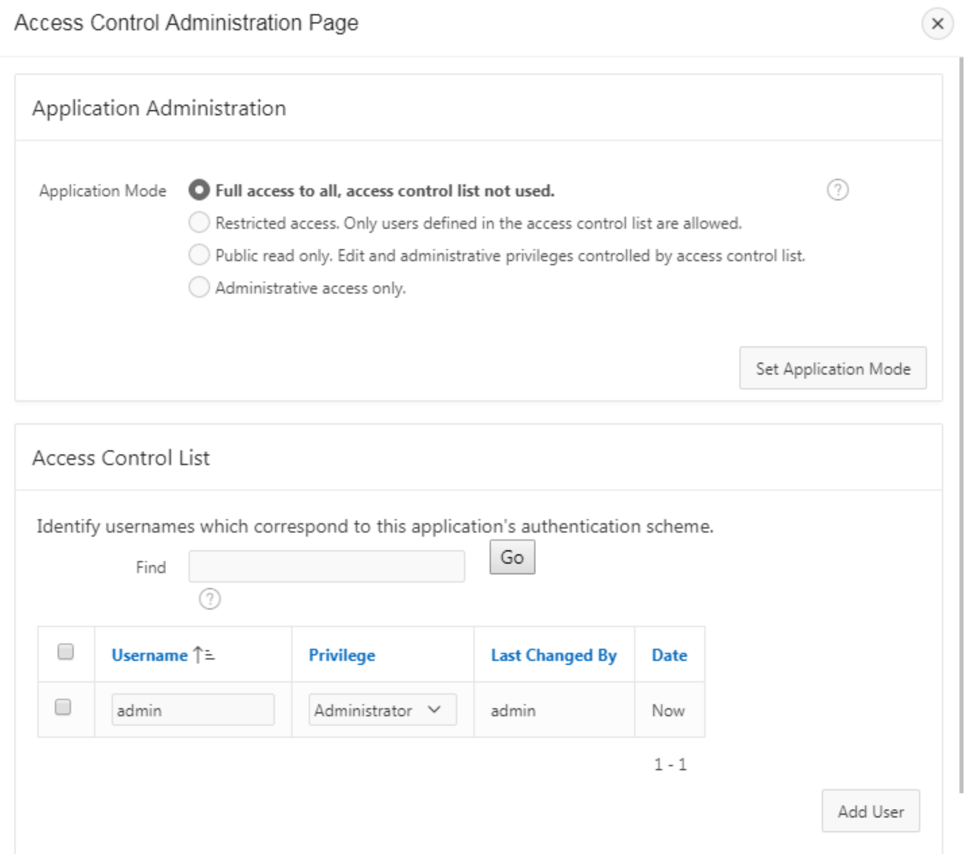
iii) Insert one record at least

```
insert into my_users values
(my_users_seq.nextval,'admin','admin123',1);
```

iv) Write this function in appropriate place (Shard Components) [Details in the class]

```
FUNCTION my_auth (
    p_username IN VARCHAR2,
    p_password IN VARCHAR2)
RETURN BOOLEAN AS
    found number := 0;
BEGIN
  SELECT 1 into found FROM MY_USERS
             WHERE upper(USER_NAME) = upper(p_username)
             AND upper(USER_PASSWORD) = upper(p_password)
             AND USER_ACTIVATED = 1;
    RETURN TRUE;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
  RETURN FALSE;
END;
```

v) Create an access control page



Access Control Administration Page

Application Administration

Application Mode

● **Full access to all, access control list not used.**

○ Restricted access. Only users defined in the access control list are allowed.

○ Public read only. Edit and administrative privileges controlled by access control list.

○ Administrative access only.

Access Control List

Identify usernames which correspond to this application's authentication scheme.

Find

☐	Username	Privilege	Last Changed By	Date
☐	admin	Administrator	admin	Now

1 - 1

Triggers in Oracle

Oracle Database automatically executes a trigger when specified conditions occur.

When you create a trigger, the database enables it automatically.

A database schema consists of –

Branch(branch_id, branch_name, branch_city, branch_budget)

Account(account_id, branch_id, balance)

Deposit(deposit_id, account_id, deposit_amount, deposit_date, deposit_user)

Withdraw(withdraw_id, account_id, withdraw_amount, withdraw_date, withdraw_user)

When you deposit an amount of money into a particular account the balance of the account will be increased.
On the other hand, when you withdraw an amount of money from a particular account the balance of the account will be decreased.

How can you make it increasing/decreasing of balance automatic? By using triggers.

Triggers can be defined for insert, update, delete operation at the time of before executing the operation or after executing the operation.

General Syntax:

```
CREATE [OR REPLACE] TRIGGER
<trigger_name>
<trigger_type>
   ON table_name
   [ FOR EACH ROW ]

DECLARE
   -- variable declarations

BEGIN
   -- trigger code

EXCEPTION
   WHEN ...
   -- exception handling

END;
```

An Example: (Before Insert)

```
CREATE OR REPLACE TRIGGER
deposit_before_insert
BEFORE INSERT
   ON deposit
   FOR EACH ROW

DECLARE
   v_username varchar2(10);

BEGIN
   -- Find username of person performing
INSERT into table
   SELECT user INTO v_username
   FROM dual;
   -- Update create_date field to current
system date
   :new.deposit_date := sysdate;
   -- Update created_by field to the username
of the person performing the INSERT
   :new.deposit_user := v_username;
END;

/
```