* **DDL for Training and placement portal**

drop table Post;

drop table Location;

drop table Notice;

drop table Register;

drop table Status;

drop table Schedule;

drop table Company;

drop table UserInfo;

drop table Branch;

drop SEQUENCE branch\_id;

drop SEQUENCE admin\_id;

drop SEQUENCE student\_id;

drop SEQUENCE post\_id;

drop SEQUENCE company\_id;

drop SEQUENCE schedule\_id;

drop SEQUENCE notice\_id;

drop SEQUENCE register\_id;

drop SEQUENCE status\_id;

**---------------Branch Table-----------------**

CREATE SEQUENCE branch\_id INCREMENT BY 1;

CREATE TABLE Branch (

b\_id VARCHAR2(10) NOT NULL PRIMARY KEY,

bname VARCHAR2(5) NOT NULL,

hod VARCHAR2(20) NULL,

CONSTRAINT bname\_branch CHECK (REGEXP\_LIKE(bname, '^[A-Z]{2}$'))

);

**-----------------User Table-----------------**

CREATE SEQUENCE admin\_id INCREMENT BY 1 Start with 101;

CREATE SEQUENCE student\_id INCREMENT BY 1 start with 101;

CREATE TABLE UserInfo (

u\_id VARCHAR(10) NOT NULL PRIMARY KEY,

fname VARCHAR2(50) NULL,

mname VARCHAR2(50) NULL,

lname VARCHAR2(50) NULL,

imageurl VARCHAR2(20) NULL,

mobile VARCHAR2(10) NULL,

email VARCHAR2(50) NULL,

dob DATE NULL,

b\_id VARCHAR2(10) NULL,

rollno VARCHAR2(10) NULL,

password VARCHAR2(50) NULL,

cpi NUMBER(4,2) NULL,

backlog VARCHAR2(10) NULL,

address VARCHAR2(80) NULL,

sequrity\_q\_a VARCHAR2(20) NULL,

role VARCHAR2(10) NULL,

active VARCHAR2(10) DEFAULT 'false',

CONSTRAINT FK\_bid\_user FOREIGN KEY(b\_id) REFERENCES Branch(b\_id),

CONSTRAINT fname\_userinfo CHECK (REGEXP\_LIKE(fname,'^[A-Za-z]{2,10}$')),

CONSTRAINT mname\_userinfo CHECK (REGEXP\_LIKE(mname,'^[A-Za-z]{2,10}$')),

CONSTRAINT lname\_userinfo CHECK (REGEXP\_LIKE(lname,'^[A-Za-z]{2,10}$')),

CONSTRAINT mobile\_userinfo CHECK (REGEXP\_LIKE(mobile,'^([1-9]{1}[0-9]{9})$')),

CONSTRAINT password\_userinfo CHECK (REGEXP\_LIKE(password,'^([a-zA-Z0-9@\*#]{7,14})$')),

CONSTRAINT email\_userinfo CHECK

(REGEXP\_LIKE(email,'^([a-zA-Z0-9\_\-\.]+)@((gmail.com)|(yahoo.com))$'))

);

**---------------------------Post Table---------------------------**

CREATE SEQUENCE post\_id INCREMENT BY 1;

CREATE TABLE Post(

p\_id NUMBER(6) NOT NULL PRIMARY KEY,

u\_id VARCHAR2(10) NULL,

description VARCHAR2(10) NULL,

post\_date DATE NULL,

CONSTRAINT FK\_uid\_Post FOREIGN KEY(u\_id) REFERENCES UserInfo(u\_id)

);

**-----------------------------Company Table------------------------**

CREATE SEQUENCE company\_id INCREMENT BY 1 start with 1000;

CREATE TABLE Company(

c\_id VARCHAR2(10) NOT NULL PRIMARY KEY,

cname VARCHAR(50) NULL,

phone VARCHAR2(12) NULL,

email VARCHAR2(50) NULL,

website VARCHAR2(50) NULL,

CONSTRAINT cname\_company CHECK (REGEXP\_LIKE(cname,'^[A-Za-z]{2,19}$')),

CONSTRAINT email\_company CHECK

(REGEXP\_LIKE(email,'^([a-zA-Z0-9\_\-\.]+)@((gmail.com)|(yahoo.com))$'))

);

**----------------------------Location Table---------------------**

CREATE TABLE Location(

c\_id VARCHAR2(10) NOT NULL,

location VARCHAR2(100) NULL,

CONSTRAINT FK\_cid\_location FOREIGN KEY(c\_id) REFERENCES Company(c\_id)

);

**------------------------------Schedule Table-------------------**

CREATE SEQUENCE schedule\_id INCREMENT BY 1;

CREATE TABLE Schedule(

s\_id VARCHAR2(10) NOT NULL PRIMARY KEY,

c\_id VARCHAR2(10) NULL,

package NUMBER(4,2) NULL,

visit\_date DATE NULL,

deadline DATE NULL,

min\_cpi NUMBER(4,2) NULL,

backlog VARCHAR2(10) NULL,

vacancy NUMBER(4) NULL,

hrname VARCHAR2(20) NULL,

description VARCHAR(100) NULL,

CONSTRAINT FK\_cid\_schedule FOREIGN KEY(c\_id) REFERENCES Company(c\_id),

CONSTRAINT hrname\_schedule CHECK (REGEXP\_LIKE(hrname,'^[A-Za-z]{2,9}$'))

);

**-----------------------------Notice Table-----------------------------**

CREATE SEQUENCE notice\_id INCREMENT BY 1;

CREATE TABLE Notice(

n\_id VARCHAR2(10) NOT NULL PRIMARY KEY,

u\_id VARCHAR2(10) NULL,

notice\_date DATE NULL,

CONSTRAINT FK\_uid\_notice FOREIGN KEY(u\_id) REFERENCES UserInfo(u\_id)

);

**-------------------------------Register Table---------------------------**

CREATE SEQUENCE register\_id INCREMENT BY 1;

CREATE TABLE Register(

r\_id NUMBER(6) NOT NULL PRIMARY KEY,

u\_id VARCHAR2(10) NULL,

s\_id VARCHAR2(10) NULL,

active VARCHAR2(10) DEFAULT 'true',

CONSTRAINT FK\_uid\_Registers FOREIGN KEY(u\_id) REFERENCES UserInfo(u\_id),

CONSTRAINT FK\_sid\_Registers FOREIGN KEY(s\_id) REFERENCES Schedule(s\_id)

);

**------------------------------Status Table------------------------**

CREATE SEQUENCE status\_id INCREMENT BY 1;

CREATE TABLE Status(

st\_id NUMBER(6) NOT NULL PRIMARY KEY,

u\_id VARCHAR2(10) NULL,

c\_id VARCHAR2(10) NULL,

package NUMBER(4,2) NULL,

placed\_date DATE NULL,

placed VARCHAR2(10) NULL,

CONSTRAINT FK\_uid\_Status FO REIGN KEY(u\_id) REFERENCES UserInfo(u\_id),

CONSTRAINT FK\_cid\_Status FOREIGN KEY(c\_id) REFERENCES Company(c\_id)

);

**---Triggers will change active field of register table to false if placed field in status table is changed to true--**

CREATE OR REPLACE TRIGGER status\_changed

AFTER

INSERT ON Status

FOR EACH ROW

BEGIN

IF :NEW.placed = 'true' THEN

UPDATE Register SET active = 'false' WHERE u\_id = :NEW.u\_id;

end if;

end;

COMMIT;

**-------------------------------End---------------------------**

* **DML**

**--------Branch------------**

INSERT INTO Branch (b\_id,bname,hod) VALUES (branch\_id.NEXTVAL,'CE','CKB');

INSERT INTO Branch (b\_id,bname,hod) VALUES (branch\_id.NEXTVAL,'IT','ABC');

**--------------UserInfo---------------------**

INSERT INTO userinfo (u\_id,fname,mname,lname,mobile,password,email,dob,b\_id,rollno,cpi,backlog,address,sequrity\_q\_a,role) VALUES (

to\_char(sysdate,'yyyy') || 'CE' || student\_id.nextval ,

'Rohan' , 'Maheshbai' , 'Soni' ,

'9876543210','asdfghjasdfghj',

'rohan33@yahoo.com','31-MAR-1998' ,

'1','CE112',9.31,'false','asbdgcfdeg',

'1\_red','student'

);

**--------------------Comapany & location------------------------**

INSERT INTO Company (c\_id,cname,phone,email,website) VALUES (company\_id.NEXTVAL,'Amazon','9876543210','amazon@gmail.com','amazon.com');

INSERT INTO Location (c\_id,location) VALUES (company\_id.CURRVAL,'banglore');

INSERT INTO Company (c\_id,cname,phone,email,website) VALUES (company\_id.NEXTVAL,'Google','9876543210','amazon@gmail.com','amazon.com');

INSERT INTO Location (c\_id,location) VALUES (company\_id.CURRVAL,'banglore');

INSERT INTO Company (c\_id,cname,phone,email,website) VALUES (company\_id.NEXTVAL,'Oracle','9876543210','amazon@gmail.com','amazon.com');

INSERT INTO Location (c\_id,location) VALUES (company\_id.CURRVAL,'banglore');

**--------------------Schedule----------------------------**

INSERT INTO Schedule (s\_id,c\_id,package,visit\_date,deadline,min\_cpi,backlog,vacancy,hrname,description)

VALUES

(schedule\_id.nextval,'1000',9.31,'19-OCT-2017','13-OCT-2017',7.5,'false',100,'acfx','description');

INSERT INTO Schedule (s\_id,c\_id,package,visit\_date,deadline,min\_cpi,backlog,vacancy,hrname,description)

VALUES

(schedule\_id.nextval,'1001',9.31,'19-OCT-2017','13-OCT-2017',7.5,'false',100,'acfx','description');

INSERT INTO Schedule (s\_id,c\_id,package,visit\_date,deadline,min\_cpi,backlog,vacancy,hrname,description)

VALUES

(schedule\_id.nextval,'1002',9.31,'19-OCT-2017','13-OCT-2017',7.5,'false',100,'acfx','description');

**--------------------Register----------------------------**

INSERT INTO Register (r\_id,u\_id,s\_id) VALUES (register\_id.NEXTVAL,'2017CE101','1');

INSERT INTO Register (r\_id,u\_id,s\_id) VALUES (register\_id.NEXTVAL,'2017CE101','2');

INSERT INTO Register (r\_id,u\_id,s\_id) VALUES (register\_id.NEXTVAL,'2017CE101','3');

**--------------Satus-------------------**

INSERT INTO Status (st\_id,u\_id,c\_id,package,placed\_date,placed) VALUES (status\_id.NEXTVAL,'2017CE101','1001',9.31,'20-OCT-2017','true');

**------------END--------------**

**Report**

* **Query to display company name and total placed student as per given year.**

SELECT cname,AVG(package),COUNT(u\_id) AS selected\_students FROM status NATURAL JOIN company

WHERE

TO\_CHAR(placed\_date,'YYYY') =: YEAR and status.PLACED='true' GROUP BY(cname);

* **Query to display visited company as per given year.**

SELECT cname,package,min\_cpi,backlog FROM schedule NATURAL JOIN company

WHERE

TO\_CHAR(visit\_date,'YYYY') =: YEAR;

* **Query to display eligible students as per given schedule.**

SELECT u\_id,fname,mname,lname FROM userinfo JOIN schedule

ON

userinfo.cpi >= schedule.min\_cpi AND

(schedule.backlog = 'true' OR (schedule.backlog = 'false' AND userinfo.backlog = 'false'))

WHERE schedule.s\_id=:ID;

* **Query to display students who are not placed by given year.**

SELECT u\_id,fname,mname,lname,rollno FROM userinfo

WHERE

role <> 'admin' AND

u\_id NOT IN

(SELECT u\_id FROM status WHERE placed='true' AND TO\_CHAR(placed\_date,'YYYY') =: YEAR)

ORDER BY rollno;

* **Query to display registered students by given year.**

SELECT u\_id,fname,mname,lname,rollno FROM userinfo

WHERE

u\_id IN

( SELECT u\_id FROM register NATURAL JOIN schedule WHERE TO\_CHAR(visit\_date,'YYYY') =: YEAR);

* **Query to display active registered student in current year as per given schedule id.**

SELECT u\_id,fname,mname,lname,rollno FROM userinfo NATURAL JOIN register NATURAL JOIN schedule

WHERE

TO\_CHAR(visit\_date,'YYYY') = TO\_CHAR(sysdate,'YYYY') AND register.active = ‘true’;

**Deployment steps**

1. Create user named TPO.
2. Grant all permission to user TPO.
3. Drop all tables if already exists in given order Post, Location, Notice, Register, Status, Schedule, Company, UserInfo, Branch.
4. Drop all sequences if already exists in given order branch\_id, admin\_id, student\_id, post\_id, company\_id, schedule\_id, notice\_id, register\_id, status\_id.
5. Create all sequences branch\_id, admin\_id, student\_id, post\_id, company\_id, schedule\_id, notice\_id, register\_id, status\_id.
6. Create all tables in given order Branch, Userinfo, Company, Schedule, Status, Register, Notice, Location, Post.
7. Create one trigger named status\_changed.
8. Insert initial data in given order Branch, UserInfo, Company, Location, Schedule, Register, Status.
9. Commit all changes.