- # 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_cpiNUMBER(4,2)          NULL,
        backlog         VARCHAR2(10)        NULL,
        vacancyNUMBER(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.