Database Scripts

DDL

CREATE USER empdb IDENTIFIED BY empdb;  
grant dba to empdb;

CREATE TABLE department   
 (  
 department\_id NUMBER(10) NOT NULL PRIMARY KEY,  
 department\_nm VARCHAR2(25) NOT NULL   
 );

CREATE TABLE compliance   
 (   
 complianceid NUMBER(10) NOT NULL PRIMARY KEY,  
 rlType VARCHAR2(15) DEFAULT (NULL),  
 details VARCHAR(250) DEFAULT (NULL),  
 createDate DATE DEFAULT (NULL),  
 department\_id NUMBER(10) DEFAULT (NULL),   
 CONSTRAINT FK\_DEPART\_ID  
 FOREIGN KEY (department\_id)  
 REFERENCES department (department\_id )  
 );

CREATE TABLE employees  
 (   
 empid NUMBER(10) NOT NULL PRIMARY KEY,  
 firstname VARCHAR2(45) DEFAULT (NULL),  
 lastname VARCHAR2(45) DEFAULT (NULL),  
 dob DATE DEFAULT (NULL),  
 email VARCHAR2(100) DEFAULT (NULL),  
 department\_id NUMBER(10) DEFAULT (NULL),  
 CONSTRAINT FK\_DEPARTMENT\_ID  
 FOREIGN KEY (department\_id)  
 REFERENCES department (department\_id)  
 );

CREATE TABLE login\_master  
 (   
 userid NUMBER(10) NOT NULL PRIMARY KEY,  
 password VARCHAR2(30) NOT NULL,  
 role VARCHAR2(20) NOT NULL,   
 CONSTRAINT FK\_EMPID  
 FOREIGN KEY (userid)  
 REFERENCES employees(empid)  
 );

CREATE TABLE statusreport  
 (   
 complianceid NUMBER(10) NOT NULL ,  
 statusrptid NUMBER(10) NOT NULL PRIMARY KEY,  
 empid NUMBER(10) NOT NULL,  
 comments VARCHAR2(15) DEFAULT (NULL),  
 createDate DATE DEFAULT (NULL),  
 department\_id NUMBER(10) DEFAULT (NULL),   
 CONSTRAINT statusreport\_ibfk\_1  
 FOREIGN KEY (department\_id)  
 REFERENCES department (department\_id),  
 CONSTRAINT statusreport\_ibfk\_2  
 FOREIGN KEY (empid)  
 REFERENCES employees (empid),  
 CONSTRAINT statusreport\_ibfk\_3  
 FOREIGN KEY (complianceid)  
 REFERENCES compliance (complianceid)  
 );

CREATE SEQUENCE department\_seq   
start with 100   
increment by 1   
nomaxvalue;

CREATE SEQUENCE compliance\_seq   
start with 5000   
increment by 1   
nomaxvalue;

CREATE SEQUENCE employees\_seq   
start with 10000   
increment by 1   
nomaxvalue;

CREATE SEQUENCE statusreport\_seq   
start with 1   
increment by 1   
nomaxvalue;

Procedures and Functions

|  |  |
| --- | --- |
| **1** | **Stored Proc for adding Employee details into employees table** |
|  | CREATE OR REPLACE PROCEDURE addEmp\_sp  ( p\_first\_name IN VARCHAR2,  p\_last\_name IN VARCHAR2,  p\_dob IN DATE,  p\_email\_id IN VARCHAR2,  p\_dept\_id IN NUMBER   )  AS  BEGIN    INSERT INTO employees("EMPID", "FIRSTNAME", "LASTNAME", "DOB", "EMAIL","DEPARTMENT\_ID")   VALUES (employees\_seq.NEXTVAL, p\_first\_name, p\_last\_name, p\_dob, p\_email\_id, p\_dept\_id);   END addEmp\_sp; / |
| **2** | **Trigger for adding data in to Login\_Master table once after inserting the Employee details into employees table** |
|  | CREATE OR REPLACE TRIGGER employees\_after\_insert AFTER INSERT  ON employees  FOR EACH ROW   DECLARE  v\_first\_name varchar2(10);  v\_last\_name varchar2(10);   BEGIN        INSERT INTO login\_master  ( userid,  password,  role)  VALUES  ( :new.empid,'password','EMPLOYEE');   END; / |
| **3** | **Stored Proc for getting user details based on user ID** |
|  | CREATE OR REPLACE PROCEDURE getUserDetails\_sp  ( p\_user\_id IN OUT NUMBER,   p\_password OUT VARCHAR2,  p\_role OUT VARCHAR2  )  AS  BEGIN    SELECT userid,password,role   INTO p\_user\_id, p\_password, p\_role   FROM login\_master where userid =p\_user\_id;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  p\_user\_id := NULL;  p\_password := NULL;  p\_role := NULL;   END getUserDetails\_sp; / |
| **4** | **Function for getting all the rl details** |
|  | CREATE OR REPLACE Function getAllRL\_func   RETURN SYS\_REFCURSOR AS  RLCursor SYS\_REFCURSOR;   BEGIN  OPEN RLCursor FOR  SELECT com.complianceid as complianceid, com.department\_id, dept.department\_nm,rlType,details,  com.createDate, count(distinct emp.empid) AS empcount ,   count(distinct sts.empid) as statuscount   FROM department dept,employees emp,compliance com   LEFT OUTER JOIN statusreport sts   ON sts.department\_id = com.department\_id   AND sts.complianceid = com.complianceid   WHERE com.department\_id = emp.department\_id  AND com.department\_id = dept.department\_id  GROUP BY com.complianceid,com.department\_id,dept.department\_nm,rlType,details,com.createDate;  RETURN RLCursor; END; / |
| **5** | **Stored Proc for getting Employee details** |
|  | CREATE OR REPLACE PROCEDURE getAllEmp\_sp  (   cursor\_employees OUT SYS\_REFCURSOR  )  AS  BEGIN  OPEN cursor\_employees FOR  SELECT empid,firstname,lastname,dob,email,department\_nm  FROM employees emp   LEFT OUTER JOIN department depart   ON emp.department\_id = depart.department\_id order by empid;  END getAllEmp\_sp; / |