Project 19:

Elliott Francis Joseph 110

Hirdesh Kumar 114

Muhammad Asad ur Rehman 121

PROJECT MANAGEMENT SYSTEM

ORCALE DATABASE

Table of Contents

List down major features of your application, Entities, Relationships, and Constraints on attributes.

JOBS-> JOB_ID **Number(5)** ,JOB_TITLE **varchar2(25)** *Not Null*, MIN_SALARY **Number(10)** *check (MIN_SALARY > 0)*, MAX_SALARY **Number(10)** *check (MAX_SALARY > 0)*;

DEPARTMENTS-> DEPARTMENT_ID **Number(5)** , DEPARTMENT_NAME **varchar2(25)** *Not Null*, MANAGER_ID **Number(5)** (FK FROM EMPLOYEES),BRANCH_ID **Number(5)** (FK FROM BRANCH);

EMPLOYEES-> EMPLOYEE_ID **Number(5)**,FIRST_NAME **varchar2(25)** *Not Null*, LAST_NAME **varchar2(25)** *Not Null*, DOB **date** *Not Null*, CNIC **varchar2(25))** *unique*, HIRE_DATE **date** *NOT NULL*; RESIDENCE_NAME **varchar2(25)** , RESIDENCE_NO **varchar2(25)** , AREA **varchar2(25)**, SALARY **Number(10)** ,District **varchar2(25)** *check ('North','South','East','West')*, ZipCode **varchar2(25)**, E-MAIL **varchar2(25)** *Not Null*, DEPARTMENT_ID **Number(5)** (FK DEPARTMENT),JOB_ID **Number(5)** (FK FROM JOBS);

PHONE NUMBER-> PNUMBER **varchar2(25),** EMPLOYEE_ID **Number(5)**(FK FROM EMPLOYEES

CITIES->CITY_ID **Number(5)**,PROVINCE_ID **Number(5)**(FK FROM PROVINCE), ,CITY_NAME **varchar2(25)** *Not Null*,

---

PROVINCE -> PROVINCE_ID **Number(5)** ,PROVINCE_NAME **varchar2(25)** *Not Null,*,COUNTRY_ID **Number(5)**(FK FROM COUNTRY);

BRANCH-> BRANCH_ID **Number(5)**, BRANCH_NAME **varchar2(25)** *Not Null*, CITY_ID **Number(5)** (FK FROM CITIES), PROVINCE_ID **Number(5)** (FK FROM CITIES)

PROJECTS->PROJECT_ID **Number(5)** ,PROJECT_NAME **varchar2(25)** *Not Null*. START_DATE **date** *Not Null*, DUE_DATE **date**, BUDGET **Number(15)** *check (BUDGET > 0)* STATUS *check ('Complete','Incomplete'),*INCHARGE_ID **Number(5)**(FK FROM EMPLOYEES),CLIENT_ID **Number(5)**(FK FROM CLIENTS);

CLIENTS->CLIENT_ID **Number(5)** ,CLIENT_NAME **varchar2(25)** *Not Null*.,Organization_name **varchar2(25)**,CONTACT_NUMBER **varchar2(25)**,E-MAIL **varchar2(25)** *Not Null,* COUNTY_ID **Number(5)**(FK FROM COUNTRY)

TOOLS>TOOL_ID **Number(5)**,TOOL_NAME **varchar2(25)** *Not Null*, PURCHASED_DATE **date**, PRICE **Number(15)** *check (PRICE > 0)*

SKILLS ->SKILL_ID **Number(5)**,SKILL_NAME **varchar2(25)** *Not Null*

TASK->TASK_ID **Number(5)**, TASK_NAME **varchar2(25)** *Not Null*, ASSIGNED_DATE **date** *Not Null*, COMPLETE_DATE **date**, PROJECT_ID **Number(5)** (FK FROM PROJECT),PHASE_ID **Number(5)**(FK FROM PHASE);

SKILL_EMPLOYEES-> SKILL_ID **Number(5)**(FK FROM SKILLS),EMPLOYEE_ID **Number(5,0)** (FK FROM EMPLOYEES),

---

EMPLOYEE_PROJECT-> EMPLOYEE_ID **Number(5)**(FK FROM EMPLOYEE), PROJECT_ID **Number(5)**(FK FROM PROJECT), JOINING_DATE **date** *Not Null*, LEAVING_DATE **date**;

PROJECT_TOOLS-> PROJECT_ID **Number(5)** (FK FROM PROJECT), TOOL_ID **Number(5,0)** (FK FROM TOOLS), START_DATE **date** *Not Null*, END_DATE **date**;

PHASES-> PHASE_ID **NUMBER(5)**,PHASE_NAME **varchar2(25)** *check('initiation','planning','execution','monitoring','closure')*;

COUNTRIES-> COUNTRY_ID **NUMBER(5)** ,COUNTRY_NAME **varchar2(25)** *Not Null*;, CONTINENT_ID **NUMBER(5)** (FK FROM CONTINENT);

CONTIENTS->CONTINENT_ID **NUMBER(5)**,CONTIENT_NAME **varchar2(25)** *Not Null*

EMP_TASK -> TASK_ID **Number(5)**, EMPLOYEE_ID **Number(5),** Task_Status **varchar2(25)** *check('Done','Not Done',)*;

List Relationships between entities.

One Department has only one Manager(employees_department).(1 to 1 relation)
Every Department has only one branch(branch_Department) (1 to 1 relation)
One Department has many employees (Employees_Department) (1 to M relation)
One Employee has only one job (jobs_ Employee) (1 to 1 relation)
One Employee can have many Phone Numbers(Phone Number_Employees) (1 to M relation)
One Province can have Many Cities (City_Province) (1 to M relation)
One Country can have Many Provinces (Province_Country) (1 to M relation)
One City has One branch (Branches_City)(1 to 1 relation)
One Project only one Project Leader (Project_Employee)(1 to 1 relation)
One Project has only one client (client_Project)(1 to 1 relation)
One country may have many clients (Countries_Clients)(1 to M relation)
One project can have many tasks(Tasks_project)(1 to M relation)
Many employee has many tasks (Tasks_empoyees)(M to M relation)
One phase has many tasks (Phases_tasks)(1 to M relation)
Many Employees have many skills(Employee_Skills)(M to M relations)
Many Employees have many projects(Employees_Projects)(M to M relation)
Many Projects use many tools (Projects_tools)(M to M relation)
One Continent has many countries(Continent_Countries)(1 to M relation)

List Roles and Privileges

Role	Object/Table/Entity/Relationship	Privilege
Employee_Role	EMPLOYEES	VIEW/UPDATE
	PHONE	INSERT/VIEW/UPDATE
	SKILL_EMPLOYEE	INSERT/VIEW/UPDATE
Manager_Role	DEPARTMENT	UPDATE/VIEW
	TASK	INSERT/UPDATE/DELETE
	PROJECTS	INSERT/UPDATE/DELETE
	EMPLOYEE_PROJECTS	INSERT/UPDATE/DELETE
	TOOLS_PROJECT	INSERT/UPDATE/DELETE
	JOBS	INSERT/UPDATE/DELETE
	TOOLS	INSERT/UPDATE/DELETE
	PHASES	VIEW
Deo	ALL	ALL

List of USERS

User	Role
Asad	deo
Kumar	employee_role, manager_role
Elliot	employee_role

QUERIES for making USER and ROLES

DEO ROLE

```
create role deo not identified;
GRANT ALL PRIVILEGES TO deo with admin option;
```

MANAGER ROLE

```
create role manager_role;
grant connect to manager_role;
grant select,update on pm.DEPARTMENTS to manager_role;
grant select on pm.PHASES to manager_role;
grant insert,select,update,delete on pm.TASK to manager_role;
grant insert,select,update,delete on pm.PROJECTS to manager_role;
grant insert,select,update,delete on pm.EMPLOYEE_PROJECTS to manager_role;
grant insert,select,update,delete on pm.TOOLS_PROJECTS to manager_role;
grant insert,select,update,delete on pm.TOOLS to manager_role;
grant insert,select,update,delete on pm.JOBS to manager_role;
grant insert,select,update,delete on pm.EMP_TASK to manager_role;
grant insert,delete on pm.EMPLOYEES to manager_role;
```

EMPLOYEE ROLE

```
create role employee_role;
grant connect to employee_role;
grant insert,select,update,delete on pm.SKILLS_EMPLOYEES to employee_role;
grant insert,select,update,delete on pm.PHONE_NUMBER to employee_role;
grant select,update on pm.EMPLOYEES to employee_role;
```

USERS

---

asad (Admin/DEO)

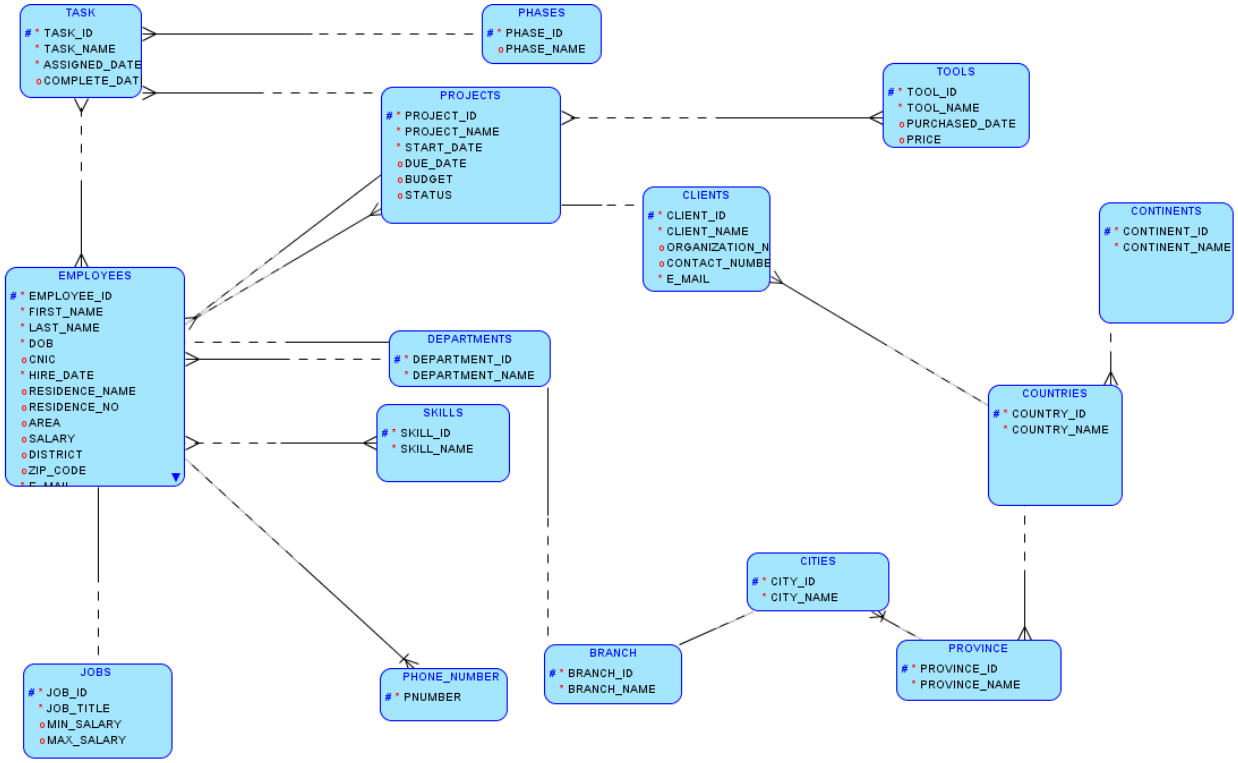
```
create user asad identified by asad;
grant deo to asad;
GRANT EXECUTE ON pm.TotalCompleteProject TO ASAD;
GRANT EXECUTE ON pm.sub_budget TO ASAD;
GRANT EXECUTE ON pm.add_budget TO ASAD;
GRANT EXECUTE ON pm.bonous TO ASAD;
GRANT EXECUTE ON pm. update_task TO ASAD;
GRANT EXECUTE ON pm. update_emp_taskTO ASAD;
GRANT EXECUTE ON pm. RAISE_SALARYTO ASAD;
GRANT EXECUTE ON pm. add_task TO ASAD;
```

kumar (Manager_role)

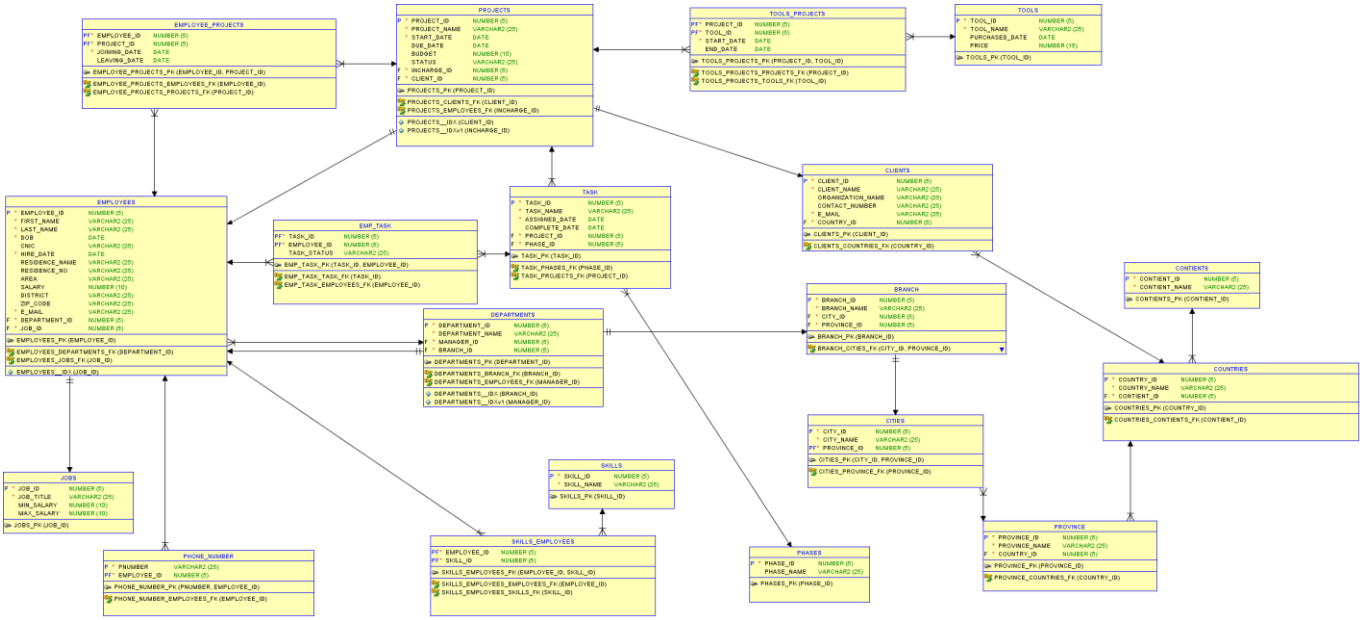
```
create user kumar identified by kumar;
grant manager_role to kumar;
grant employee_role to kumar;
```

elliot (Employee_role)
create user elliot identified by elliot;
grant employee_role to elliot;

Conceptual Schema



Relation Schema



Create Database

```
CREATE TABLE branch (
   branch_id    NUMBER(5) NOT NULL,
   branch_name  VARCHAR2(25) NOT NULL,
   city_id      NUMBER(5) NOT NULL,
   province_id  NUMBER(5) NOT NULL
);

CREATE UNIQUE INDEX branch__idx ON
   branch (
      city_id
   ASC,
      province_id
   ASC );

ALTER TABLE branch ADD CONSTRAINT branch_pk PRIMARY KEY ( branch_id );

CREATE TABLE cities (
   city_id      NUMBER(5) NOT NULL,
   city_name    VARCHAR2(25) NOT NULL,
   province_id  NUMBER(5) NOT NULL
);

ALTER TABLE cities ADD CONSTRAINT cities_pk PRIMARY KEY ( city_id,
                                                          province_id );

CREATE TABLE clients (
   client_id         NUMBER(5) NOT NULL,
   client_name       VARCHAR2(25) NOT NULL,
   organization_name  VARCHAR2(25),
   contact_number     VARCHAR2(25),
   e_mail            VARCHAR2(25) NOT NULL,
   country_id        NUMBER(5) NOT NULL
);

ALTER TABLE clients ADD CONSTRAINT clients_pk PRIMARY KEY ( client_id );

CREATE TABLE contients (
   contient_id    NUMBER(5) NOT NULL,
   contient_name  VARCHAR2(25) NOT NULL
);

ALTER TABLE contients ADD CONSTRAINT contients_pk PRIMARY KEY ( contient_id );
```

```
CREATE TABLE countries (
   country_id    NUMBER(5) NOT NULL,
   country_name  VARCHAR2(25) NOT NULL,
   contient_id   NUMBER(5) NOT NULL
);

ALTER TABLE countries ADD CONSTRAINT countries_pk PRIMARY KEY ( country_id );

CREATE TABLE departments (
   department_id    NUMBER(5) NOT NULL,
   department_name  VARCHAR2(25) NOT NULL,
   manager_id       NUMBER(5) NOT NULL,
   branch_id        NUMBER(5) NOT NULL
);

CREATE UNIQUE INDEX departments__idx ON
   departments (
      branch_id
   ASC );

CREATE UNIQUE INDEX departments__idxv1 ON
   departments (
      manager_id
   ASC );

ALTER TABLE departments ADD CONSTRAINT departments_pk PRIMARY KEY ( department_id );

CREATE TABLE emp_task (
   task_id      NUMBER(5) NOT NULL,
   employee_id  NUMBER(5) NOT NULL,
   task_status  VARCHAR2(25)
);

ALTER TABLE emp_task ADD CONSTRAINT emp_task_pk PRIMARY KEY ( task_id,
                                                              employee_id );

CREATE TABLE employee_projects (
   employee_id   NUMBER(5) NOT NULL,
   project_id    NUMBER(5) NOT NULL,
   joining_date  DATE NOT NULL,
   leaving_date  DATE
);

ALTER TABLE employee_projects ADD CONSTRAINT employee_projects_pk PRIMARY KEY ( employee_id,
                                                                                 project_id );

CREATE TABLE employees (
   employee_id     NUMBER(5) NOT NULL,
```

```
   first_name      VARCHAR2(25) NOT NULL,
   last_name       VARCHAR2(25) NOT NULL,
   dob             DATE NOT NULL,
   cnic            VARCHAR2(25),
   hire_date       DATE NOT NULL,
   residence_name  VARCHAR2(25),
   residence_no    VARCHAR2(25),
   area            VARCHAR2(25),
   salary          NUMBER(10),
   district        VARCHAR2(25),
   zip_code        VARCHAR2(25),
   e_mail          VARCHAR2(25) NOT NULL,
   department_id   NUMBER(5) NOT NULL,
   job_id          NUMBER(5) NOT NULL
);

CREATE UNIQUE INDEX employees__idx ON
   employees (
      job_id
   ASC );

ALTER TABLE employees ADD CONSTRAINT employees_pk PRIMARY KEY ( employee_id );

CREATE TABLE jobs (
   job_id      NUMBER(5) NOT NULL,
   job_title   VARCHAR2(25) NOT NULL,
   min_salary  NUMBER(10),
   max_salary  NUMBER(10)
);

ALTER TABLE jobs ADD CONSTRAINT jobs_pk PRIMARY KEY ( job_id );

CREATE TABLE phases (
   phase_id    NUMBER(5) NOT NULL,
   phase_name  VARCHAR2(25)
);

ALTER TABLE phases ADD CONSTRAINT phases_pk PRIMARY KEY ( phase_id );

CREATE TABLE phone_number (
   pnumber      VARCHAR2(25) NOT NULL,
   employee_id  NUMBER(5) NOT NULL
);

ALTER TABLE phone_number ADD CONSTRAINT phone_number_pk PRIMARY KEY ( pnumber,
                                                                      employee_id );

CREATE TABLE projects (
```

```
   project_id    NUMBER(5) NOT NULL,
   project_name  VARCHAR2(25) NOT NULL,
   start_date    DATE NOT NULL,
   due_date      DATE,
   budget        NUMBER(15),
   status        VARCHAR2(25),
   incharge_id   NUMBER(5) NOT NULL,
   client_id     NUMBER(5) NOT NULL
);

CREATE UNIQUE INDEX projects__idx ON
   projects (
      client_id
   ASC );

CREATE UNIQUE INDEX projects__idxv1 ON
   projects (
      incharge_id
   ASC );

ALTER TABLE projects ADD CONSTRAINT projects_pk PRIMARY KEY ( project_id );

CREATE TABLE province (
   province_id    NUMBER(5) NOT NULL,
   province_name  VARCHAR2(25) NOT NULL,
   country_id     NUMBER(5) NOT NULL
);

ALTER TABLE province ADD CONSTRAINT province_pk PRIMARY KEY ( province_id );

CREATE TABLE skills (
   skill_id    NUMBER(5) NOT NULL,
   skill_name  VARCHAR2(25) NOT NULL
);

ALTER TABLE skills ADD CONSTRAINT skills_pk PRIMARY KEY ( skill_id );

CREATE TABLE skills_employees (
   employee_id  NUMBER(5) NOT NULL,
   skill_id     NUMBER(5) NOT NULL
);

ALTER TABLE skills_employees ADD CONSTRAINT skills_employees_pk PRIMARY KEY ( employee_id,
                                                                              skill_id );

CREATE TABLE task (
   task_id       NUMBER(5) NOT NULL,
   task_name     VARCHAR2(25) NOT NULL,
```

```
   assigned_date  DATE NOT NULL,
   complete_date  DATE,
   project_id     NUMBER(5) NOT NULL,
   phase_id       NUMBER(5) NOT NULL
);

ALTER TABLE task ADD CONSTRAINT task_pk PRIMARY KEY ( task_id );

CREATE TABLE tools (
   tool_id        NUMBER(5) NOT NULL,
   tool_name      VARCHAR2(25) NOT NULL,
   purchased_date  DATE,
   price          NUMBER(15)
);

ALTER TABLE tools ADD CONSTRAINT tools_pk PRIMARY KEY ( tool_id );

CREATE TABLE tools_projects (
   project_id  NUMBER(5) NOT NULL,
   tool_id     NUMBER(5) NOT NULL,
   start_date  DATE NOT NULL,
   end_date    DATE
);

ALTER TABLE tools_projects ADD CONSTRAINT tools_projects_pk PRIMARY KEY ( project_id,
                                                                          tool_id );

ALTER TABLE branch
   ADD CONSTRAINT branch_cities_fk FOREIGN KEY ( city_id,
                                                 province_id )
      REFERENCES cities ( city_id,
                          province_id );

ALTER TABLE cities
   ADD CONSTRAINT cities_province_fk FOREIGN KEY ( province_id )
      REFERENCES province ( province_id );

ALTER TABLE clients
   ADD CONSTRAINT clients_countries_fk FOREIGN KEY ( country_id )
      REFERENCES countries ( country_id );

ALTER TABLE countries
   ADD CONSTRAINT countries_contients_fk FOREIGN KEY ( contient_id )
      REFERENCES contients ( contient_id );

ALTER TABLE departments
   ADD CONSTRAINT departments_branch_fk FOREIGN KEY ( branch_id )
      REFERENCES branch ( branch_id );
```

```
ALTER TABLE departments
   ADD CONSTRAINT departments_employees_fk FOREIGN KEY ( manager_id )
      REFERENCES employees ( employee_id );

ALTER TABLE emp_task
   ADD CONSTRAINT emp_task_employees_fk FOREIGN KEY ( employee_id )
      REFERENCES employees ( employee_id );

ALTER TABLE emp_task
   ADD CONSTRAINT emp_task_task_fk FOREIGN KEY ( task_id )
      REFERENCES task ( task_id );

ALTER TABLE employee_projects
   ADD CONSTRAINT employee_projects_employees_fk FOREIGN KEY ( employee_id )
      REFERENCES employees ( employee_id );

ALTER TABLE employee_projects
   ADD CONSTRAINT employee_projects_projects_fk FOREIGN KEY ( project_id )
      REFERENCES projects ( project_id );

ALTER TABLE employees
   ADD CONSTRAINT employees_departments_fk FOREIGN KEY ( department_id )
      REFERENCES departments ( department_id );

ALTER TABLE employees
   ADD CONSTRAINT employees_jobs_fk FOREIGN KEY ( job_id )
      REFERENCES jobs ( job_id );

ALTER TABLE phone_number
   ADD CONSTRAINT phone_number_employees_fk FOREIGN KEY ( employee_id )
      REFERENCES employees ( employee_id );

ALTER TABLE projects
   ADD CONSTRAINT projects_clients_fk FOREIGN KEY ( client_id )
      REFERENCES clients ( client_id );

ALTER TABLE projects
   ADD CONSTRAINT projects_employees_fk FOREIGN KEY ( incharge_id )
      REFERENCES employees ( employee_id );

ALTER TABLE province
   ADD CONSTRAINT province_countries_fk FOREIGN KEY ( country_id )
      REFERENCES countries ( country_id );

ALTER TABLE skills_employees
   ADD CONSTRAINT skills_employees_employees_fk FOREIGN KEY ( employee_id )
      REFERENCES employees ( employee_id );
```

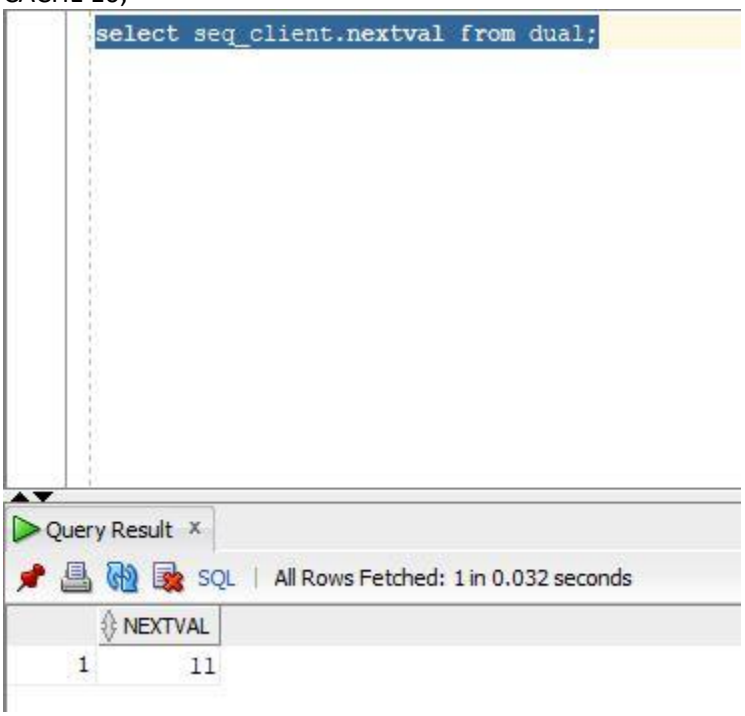
```
ALTER TABLE skills_employees
   ADD CONSTRAINT skills_employees_skills_fk FOREIGN KEY ( skill_id )
      REFERENCES skills ( skill_id );

ALTER TABLE task
   ADD CONSTRAINT task_phases_fk FOREIGN KEY ( phase_id )
      REFERENCES phases ( phase_id );

ALTER TABLE task
   ADD CONSTRAINT task_projects_fk FOREIGN KEY ( project_id )
      REFERENCES projects ( project_id );

ALTER TABLE tools_projects
   ADD CONSTRAINT tools_projects_projects_fk FOREIGN KEY ( project_id )
      REFERENCES projects ( project_id );

ALTER TABLE tools_projects
   ADD CONSTRAINT tools_projects_tools_fk FOREIGN KEY ( tool_id )
      REFERENCES tools ( tool_id );
```

```
alter table Jobs add constraint chk_Min_sal check(MIN_SALARY > 0)
alter table Jobs add constraint chk_Max_sal check(MAX_SALARY > 0)
alter table Employees add constraint chk_employees_district check(district
in(West’,’East’,’North’,’South’);
alter table projects add constraint chk_budget_project check(BUDGET > 0)
 alter table projects add constraint chk_status check(status in(‘Complete’,’Incomplete’))
alter table tools add constraint chk_budget check(price > 0);
alter table phases add constraint chk_phases check(phase_name
in(‘initiation’,’planning’,’execution’,’monitoring’,’closure’));
alter table Employees add constraint chk_sal check(SALARY > 0);
alter table emp_task add constraint chk_task_status check(TASK_STATUS in('Done','Not Done'));
ALTER TABLE pm.Employees ADD UNIQUE (CNIC);
```

Implement Sequences, Indexes, Triggers for log history, Procedures/Functions.

Sequences

```
CREATE SEQUENCE seq_client
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;
```



```
CREATE SEQUENCE seq_skills
MINVALUE 0001
START WITH 0001
INCREMENT BY 1
CACHE 10
```

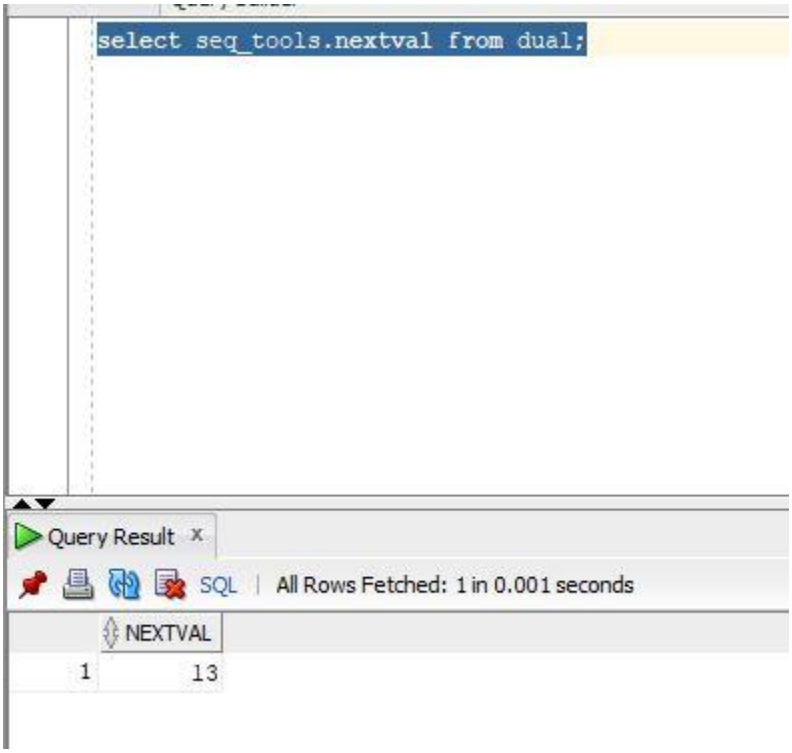
```
select seq_skills.nextval from dual;
```

Query Result

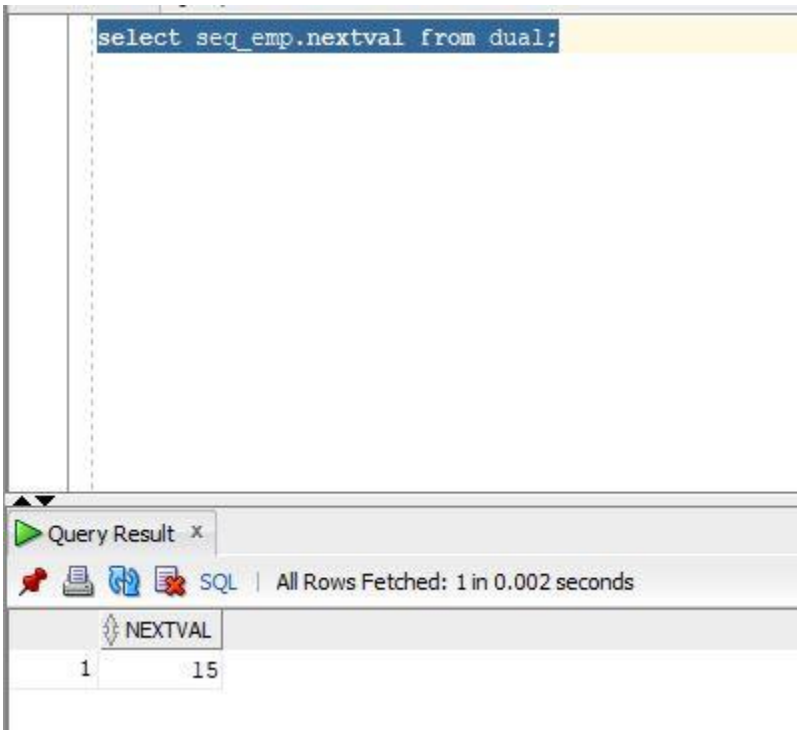
SQL | All Rows Fetched: 1 in 0.002 seconds

	NEXTVAL
1	12

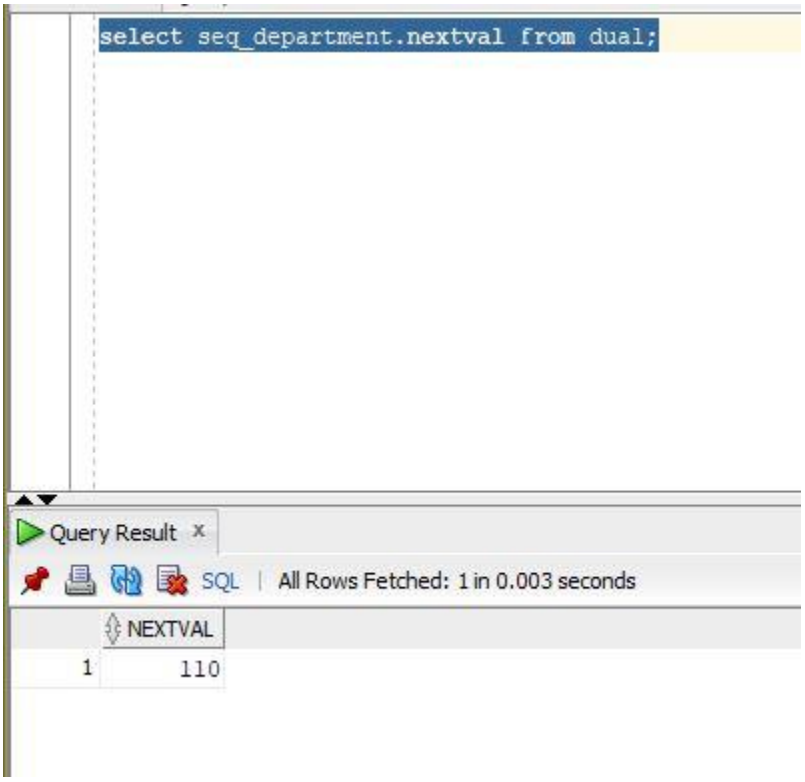
```
CREATE SEQUENCE seq_tools
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10
```



CREATE SEQUENCE seq_emp
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10



create sequence seq_department
MINVALUE 101
START WITH 101
INCREMENT BY 1
CACHE 10



CREATE SEQUENCE seq_project
MINVALUE 10001
START WITH 10001
INCREMENT BY 1
CACHE 10

```
select seq_project.nextval from dual;
```

Query Result

SQL | All Rows Fetched: 1 in 0.025 seconds

	NEXTVAL
1	10006

INDEX

```
CREATE INDEX idx_name ON  pm.Employees (First_Name,Last_Name);
CREATE INDEX idx_pnumber ON pm.phone_number (PNUMBER)
```

TRIGGER

```
create or replace TRIGGER trig_client
BEFORE INSERT OR DELETE OR UPDATE ON clients
FOR EACH ROW
DECLARE
 v_user VARCHAR2 (20);
 BEGIN
 SELECT user INTO v_user FROM dual;
 IF INSERTING THEN
 dbms_output.put_line('ONE ROW INSERTED BY  ' ||v_user);
 ELSIF DELETING THEN
 dbms_output.put_line('ONE ROW DELETED BY  ' ||v_user);
 ELSIF UPDATING THEN
  dbms_output.put_line('ONE ROW UPDATED BY  ' ||v_user);
  END IF;
 END;
/
```

```
update pm.clients
set ORGANIZATION_NAME='SZABIST KARACHI'
where client_id=3;
```

Script Output x | Query Result x

Task completed in 0.034 seconds

```
1 row updated.


Rollback complete.


Rollback complete.

ONE ROW UPDATED BY  ASAD


1 row updated.
```

```
create or replace TRIGGER trig_task
BEFORE INSERT OR DELETE OR UPDATE ON task
FOR EACH ROW
DECLARE
 v_user VARCHAR2 (20);
 BEGIN
 SELECT user INTO v_user FROM dual;
 IF INSERTING THEN
 dbms_output.put_line('ONE ROW INSERTED BY  ' ||v_user);
 ELSIF DELETING THEN
 dbms_output.put_line('ONE ROW DELETED BY  ' ||v_user);
 ELSIF UPDATING THEN
  dbms_output.put_line('ONE ROW UPDATED BY  ' ||v_user);
  END IF;
 END;
/
```

```
SET SERVEROUTPUT ON;

select * from pm.task;

insert Into pm.TASK Values (00018, 'DBA Check', '20-Jul-2021', '20-Jul-2021', 10005, 0005) ;
```

Script Output × | Query Result ×

Task completed in 0.044 seconds

```
ONE ROW UPDATED BY  KUMAR


8 rows updated.


Rollback complete.

ONE ROW INSERTED BY  KUMAR


1 row inserted.
```

```
update pm.task
set task_name='De-Budging'
where task_id=18;
```

Script Output × Query Result ×

Task completed in 0.08 seconds

```
Rollback complete.

ONE ROW INSERTED BY  KUMAR


1 row inserted.

ONE ROW UPDATED BY  KUMAR


1 row updated.
```

```
delete pm.task
where task_id=18;
```

Script Output x | Query Result x

Task completed in 0.056 seconds

```
1 row inserted.

ONE ROW UPDATED BY  KUMAR


1 row updated.

ONE ROW DELETED BY  KUMAR


1 row deleted.
```

```
CREATE TABLE PM PROJECTS_AUDIT
  (       NEW_DUE_DATE" VARCHAR2(30),
          OLD_DUE_DATE" VARCHAR2(30),
          USER_NAME" VARCHAR2(30),
          ENTRY_DATE" VARCHAR2(30),
          OPERATION" VARCHAR2(30)
  )

create or replace TRIGGER trig_pro_audit
BEFORE INSERT OR DELETE OR UPDATE ON projects
FOR EACH ROW
DECLARE
v_user VARCHAR2(30);
v_date VARCHAR2(30);
BEGIN
SELECT user, TO_CHAR(sysdate,'DD/MON/YYYY HH24;MI:SS') INTO v_user, v_date FROM dual;
IF INSERTING THEN
 INSERT INTO projects_audit(new_DUE_DATE, old_DUE_DATE, user_name, entry_date, operation)
 VALUES (:NEW.DUE_DATE,NULL, v_user,v_date, 'Insert');
ELSIF DELETING THEN
 INSERT INTO projects_audit(new_DUE_DATE, old_DUE_DATE, user_name, entry_date, operation)
 VALUES (NULL, :OLD.DUE_DATE,v_user,v_date, 'Delete');
ELSIF UPDATING THEN
 INSERT INTO projects_audit(new_DUE_DATE, old_DUE_DATE, user_name, entry_date, operation)
 VALUES (:NEW.DUE_DATE, :OLD.DUE_DATE,v_user,v_date, 'Updating');
 END IF;
 END;
/
```

```
Insert  into PM.PROJECTS  values (10006, 'UK Online App', '05-Jul-2018', '20-DEC-2019', 570000, 'Complete', 0001, 00008);
commit;

UPDATE PM.PROJECTS
set due_date='24-DEC-2019'
where project_id = 10006;
commit;

DELETE PM.PROJECTS
where project_id = 10006;
commit;
```

Worksheet | Query Builder

```
select * from PM.projects_audit;
```

Query Result

SQL | All Rows Fetched: 8 in 0.003 seconds

	NEW_DUE_DATE	OLD_DUE_DATE	USER_NAME	ENTRY_DATE	OPERATION
1	20-OCT-15	(null)	ASAD	21/JUN/2020 18;14:16	Insert
2	29-SEP-12	(null)	ASAD	21/JUN/2020 18;15:17	Insert
3	20-JUL-19	(null)	ASAD	21/JUN/2020 18;15:17	Insert
4	10-OCT-18	(null)	ASAD	21/JUN/2020 18;15:17	Insert
5	20-OCT-20	(null)	ASAD	21/JUN/2020 18;15:17	Insert
6	20-DEC-19	(null)	KUMAR	22/JUN/2020 02;19:56	Insert
7	24-DEC-19	20-DEC-19	KUMAR	22/JUN/2020 02;24:12	Updating
8	(null)	24-DEC-19	KUMAR	22/JUN/2020 02;24:47	Delete

```
CREATE TABLE EMPLOYEES_AUDIT
  (        NEW_DEPARTMENT_ID VARCHAR2(30),
           OLD_DEPARTMENT_ID VARCHAR2(30),
           USER_NAME VARCHAR2(30),
           ENTRY_DATE VARCHAR2(30),
           OPERATION" VARCHAR2(30)
  )
create or replace TRIGGER trig_emp_audit
BEFORE INSERT OR DELETE OR UPDATE ON employees
FOR EACH ROW
DECLARE
v_user VARCHAR2(30);
v_date VARCHAR2(30);
BEGIN
SELECT user, TO_CHAR(sysdate,'DD/MON/YYYY HH24;MI:SS') INTO v_user, v_date FROM dual;
IF INSERTING THEN
 INSERT INTO employees_audit(new_department_id, old_department_id, user_name, entry_date,
operation)
 VALUES (:NEW.department_id,NULL, v_user,v_date, 'Insert');
ELSIF DELETING THEN
 INSERT INTO employees_audit(new_department_id, old_department_id, user_name, entry_date,
operation)
 VALUES (NULL, :OLD.department_id,v_user,v_date, 'Delete');
ELSIF UPDATING THEN
 INSERT INTO employees_audit(new_department_id, old_department_id, user_name, entry_date,
operation)
 VALUES (:NEW.department_id, :OLD.department_id,v_user,v_date, 'Updating');
 END IF;
 END;
/
```

```
select * from pm.employees;

update PM.employees
set department_id=109
where employee_id=12;

commit;
```

```
select * from PM.employees_audit;
```

Query Result

SQL | All Rows Fetched: 160 in 0.017 seconds

	NEW_DEPARTMENT_ID	OLD_DEPARTMENT_ID	USER_NAME	ENTRY_DATE	OPERATION
150	105	(null)	ASAD	21/JUN/2020 17;40:25	Insert
151	107	(null)	ASAD	21/JUN/2020 17;40:25	Insert
152	108	(null)	ASAD	21/JUN/2020 17;40:25	Insert
153	106	(null)	ASAD	21/JUN/2020 17;40:25	Insert
154	101	(null)	ASAD	21/JUN/2020 17;40:25	Insert
155	109	(null)	ASAD	21/JUN/2020 17;40:25	Insert
156	104	(null)	ASAD	21/JUN/2020 17;40:25	Insert
157	109	(null)	ASAD	21/JUN/2020 17;40:25	Insert
158	104	(null)	ASAD	21/JUN/2020 17;40:25	Insert
159	102	(null)	ASAD	21/JUN/2020 17;40:25	Insert
160	109	104	ELLIOT	22/JUN/2020 02;32:21	Updating

FUNCTIONS

```
create or replace FUNCTION TotalCompleteProject
RETURN number IS
  total number(4) := 0;
BEGIN
  SELECT count(*) into total
  FROM projects
  WHERE STATUS = 'Complete';
  RETURN total;
END;
/
```

```
select TotalCompleteProject from dual;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.003 seconds

	TOTALCOMPLETEPROJECT
1	4

```
create or replace FUNCTION bonous(bud number, et number) RETURN NUMBER IS
BEGIN
return ( ((bud) * ((et/100))));
END;
/
```

```
SQL> select employee_id,salary,bonous(salary,15) as "bonus",salary+bonous(salary,15) as Total_Salary
  2  from pm.employees;

EMPLOYEE_ID     SALARY      bonus TOTAL_SALARY
----------- ---------- ---------- ------------
          1     120000      18000       138000
          2     225000      33750       258750
          3    1925000     288750      2213750
          4     205000      30750       235750
          5     525000      78750       603750
          6      12000       1800        13800
          7      95000      14250       109250
          8    2050000     307500      2357500
          9      95000      14250       109250
         10      12500       1875        14375
         11      95000      14250       109250

EMPLOYEE_ID     SALARY      bonus TOTAL_SALARY
----------- ---------- ---------- ------------
         12      12500       1875        14375
         13     200000      30000       230000

13 rows selected.
```

```
create or replace FUNCTION add_budget(bud number, et number) RETURN NUMBER IS
BEGIN
return (bud + et);
END;
/
```

```
SQL> select project_id,budget,pm.add_budget(budget,500) from pm.projects;

PROJECT_ID     BUDGET PM.ADD_BUDGET(BUDGET,500)
---------- ---------- --------------------------
     10001    5000000                    5000500
     10002    3000000                    3000500
     10003    5500000                    5500500
     10004    6500000                    6500500
     10005    5000000                    5000500
```

```
create or replace FUNCTION sub_budget(bud number, et number) RETURN NUMBER IS
BEGIN
return (bud - et);
END;
/
```

```
SQL> select project_id,budget,pm.sub_budget(budget,500) from pm.projects;

PROJECT_ID     BUDGET PM.SUB_BUDGET(BUDGET,500)
---------- ---------- ------------------------
     10001    5000000                  4999500
     10002    3000000                  2999500
     10003    5500000                  5499500
     10004    6500000                  6499500
     10005    5000000                  4999500
```

PROCEDURES

```
create or replace PROCEDURE add_task
 (p_TASK_ID in TASK.TASK_ID%TYPE,
  p_TASK_NAME in TASK.TASK_NAME %TYPE,
  p_ASSIGNED_DATE in TASK.ASSIGNED_DATE%TYPE,
  p_COMPLETE_DATE in TASK.COMPLETE_DATE%TYPE,
  p_PROJECT_ID  in TASK.PROJECT_ID%TYPE,
  p_PHASE_ID in TASK.PHASE_ID%TYPE
  )
  IS
  BEGIN
  INSERT INTO task VALUES (p_TASK_ID,
p_TASK_NAME,p_ASSIGNED_DATE,p_COMPLETE_DATE,p_PROJECT_ID,p_PHASE_ID);
   dbms_output.put_line('Task added.');
END;
/
```

```
execute pm.add_task(00018, 'Q/A', '25-MAR-2020', '21-OCT-2020', 10005, 0004);
select * from pm.task;
```

Script Output x Query Result x

Task completed in 0.155 seconds

```
Error starting at line : 4 in command -
excute pm.add_task(00018, 'Q/A', '25-MAR-2020', '21-OCT-2020', 10005, 0004)
Error report -
Unknown Command


PL/SQL procedure successfully completed.
```

	TASK_ID	TASK_NAME	ASSIGNED_DATE	COMPLETE_DATE	PROJECT_ID	PHASE_ID
1	11	User Interface	14-MAY-19	20-JAN-20	10005	5
2	12	Coding	20-FEB-19	20-FEB-20	10005	4
3	13	Cost	14-FEB-19	25-MAY-19	10005	5
4	14	Testing	10-OCT-19	20-OCT-20	10005	4
5	15	Designing	14-FEB-19	20-JAN-20	10005	5
6	16	Device Test	20-JUL-20	20-JUN-20	10005	4
7	17	Server Check	20-JUL-20	20-JUL-20	10005	4
8	18	Q/A	25-MAR-20	21-OCT-20	10005	4

```
create or replace PROCEDURE update_task
 (p_TASK_ID in TASK.TASK_ID%TYPE,
  p_COMPLETE_DATE in TASK.COMPLETE_DATE%TYPE
)  IS
  BEGIN
  UPDATE TASK SET TASK.COMPLETE_DATE= p_COMPLETE_DATE
   WHERE TASK_ID = p_TASK_ID;
COMMIT;
   dbms_output.put_line('Task complete date update succesfully.');
END;
/
```

```
execute pm.update_task(00018,'24-OCT-2020');
```

Script Output | Query Result

Task completed in 0.058 seconds

```
Rollback complete.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

	TASK_ID	TASK_NAME	ASSIGNED_DATE	COMPLETE_DATE	PROJECT_ID	PHASE_ID
1	11	User Interface	14-MAY-19	20-JAN-20	10005	5
2	12	Coding	20-FEB-19	20-FEB-20	10005	4
3	13	Cost	14-FEB-19	25-MAY-19	10005	5
4	14	Testing	10-OCT-19	20-OCT-20	10005	4
5	15	Designing	14-FEB-19	20-JAN-20	10005	5
6	16	Device Test	20-JUL-20	20-JUN-20	10005	4
7	17	Server Check	20-JUL-20	20-JUL-20	10005	4
8	18	Q/A	25-MAR-20	24-OCT-20	10005	4

```
create or replace PROCEDURE update_emp_task
 (p_TASK_ID in EMP_TASK.TASK_ID%TYPE,
  p_EMPLOYEE_ID in EMP_TASK.EMPLOYEE_ID%TYPE,
  p_TASK_STATUS in EMP_TASK.TASK_STATUS%TYPE
)  IS
  BEGIN
  UPDATE EMP_TASK SET EMP_TASK.EMPLOYEE_ID=p_EMPLOYEE_ID,EMP_TASK.TASK_STATUS=
p_TASK_STATUS
   WHERE EMP_TASK.TASK_ID = p_TASK_ID;
COMMIT;
   dbms_output.put_line('Task status update succesfully.');
END;
/
```

```
execute pm.update_emp_task(5, 11, 'Not Done');
```

Script Output | Query Result

Task completed in 0.116 seconds

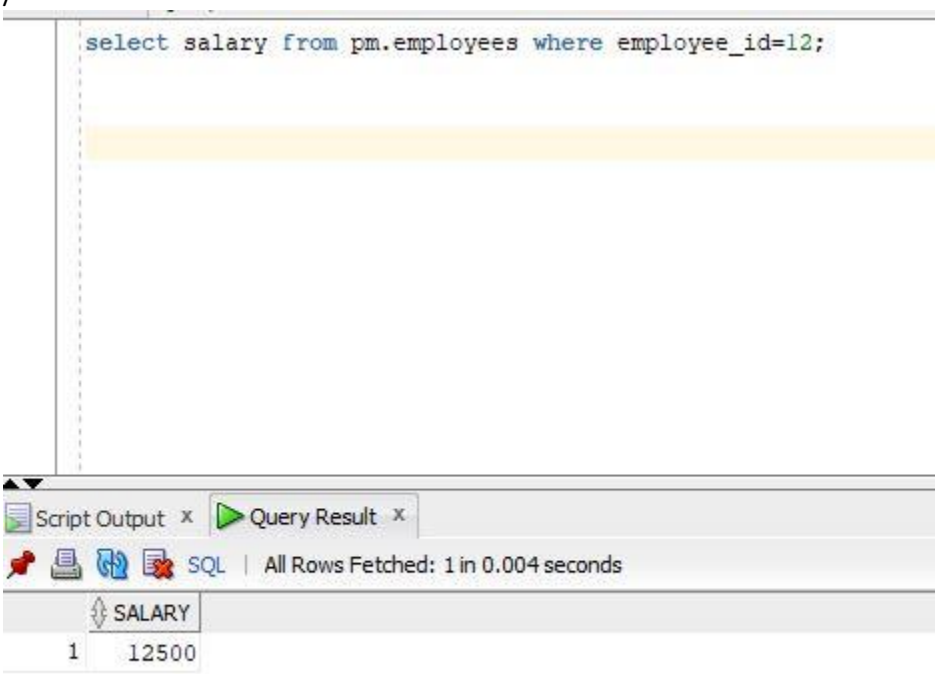
```
            For Trusted Oracle configured in DBMS MAC mode, you may see
            this message if a duplicate entry exists at a different level.
*Action:    Either remove the unique restriction or do not insert the key.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

Script Output | Query Result

SQL | All Rows Fetched: 7 in 0.004 seconds

	TASK_ID	EMPLOYEE_ID	TASK_STATUS
1	9	11	Done
2	10	11	Done
3	5	11	Not Done
4	4	14	Done
5	9	15	Done
6	10	12	Done
7	6	17	Done

```
create or replace PROCEDURE
  RAISE_SALARY (EMP IN NUMBER, AMOUNT IN NUMBER,SAL OUT NUMBER)
IS
BEGIN
UPDATE EMPLOYEES SET SALARY=SALARY+AMOUNT
   WHERE EMPLOYEE_ID = EMP;
COMMIT;
Select SALARY into SAL from EMPLOYEES where EMPLOYEE_ID = EMP;
END;
/
```



```
VARIABLE K NUMBER
EXECUTE pm.raise_salary(12,50000,:K);
print :K
```

Script Output x | Query Result x

Task completed in 0.109 seconds

```
         K
----------



PL/SQL procedure successfully completed.


         K
----------
     62500
```

INSERTION

```
Insert All
  Into CONTIENTS Values (00001, 'Asia')
  Into CONTIENTS Values (00002, 'Africa')
  Into CONTIENTS Values (00003, 'North America')
  Into CONTIENTS Values (00004, 'South America')
  Into CONTIENTS Values (00005, 'Antarctica')
  Into CONTIENTS Values (00006, 'Europe')
  Into CONTIENTS Values (00007, 'Australia')
Select * From dual;

commit;

Insert All
  Into pm.COUNTRIES Values (00101, 'Pakistan', 00001)
  Into pm.COUNTRIES Values (00102, 'UAE', 00001)
  Into pm.COUNTRIES Values (00103, 'Saudi Arabia', 00001)
  Into pm.COUNTRIES Values (00104, 'United Kingdom', 00006)
  Into pm.COUNTRIES Values (00105, 'United States Of America', 00003)
  Into pm.COUNTRIES Values (00106, 'Canada', 00003)
  Into pm.COUNTRIES Values (00107, 'Australia', 00007)
Select * From dual;

select * from pm.COUNTRIES
commit;

Insert All
  Into pm.PROVINCE Values (00001, 'Punjab', 00101)
  Into pm.PROVINCE Values (00002, 'Sindh', 00101)
  Into pm.PROVINCE Values (00003, 'KhyberPakhtunkhwa', 00101)
  Into pm.PROVINCE Values (00004, 'Balochistan', 00101)
  Into pm.PROVINCE Values (00005, 'IslamabadCapitalTerritory', 00101)
Select * From dual;

select * from pm.PROVINCE
commit;


Insert All
  Into pm.CITIES Values (0001, 'Karachi', 00002)
  Into pm.CITIES Values (0002, 'Lahore', 00001)
  Into pm.CITIES Values (0003, 'Islamabad', 00005)
  Into pm.CITIES Values (0004, 'Faislabad', 00002)
  Into pm.CITIES Values (0005, 'Quetta', 00004)
Select * From dual;
```

```
select * from pm.CITIES
commit;


Insert All
  Into pm.BRANCH Values (10001, 'Defence Branch', 1,2)
  Into pm.BRANCH Values (10002, 'Metro Branch',3,5)
  Into pm.BRANCH Values (10003, 'Fortress Branch',2,1)
  Into pm.BRANCH Values (10004, 'I.I Chadigarh Branch', 5,4)
  Into pm.BRANCH Values (10005, 'S.P Branch', 4,2)
Select * From dual;


desc pm.BRANCH;
select * from pm.BRANCH
commit;

Insert All
  Into pm.PHASES Values (0001, 'initiation')
  Into pm.PHASES Values (0002, 'planning')
  Into pm.PHASES Values (0003, 'execution')
  Into pm.PHASES Values (0004, 'monitoring')
  Into pm.PHASES Values (0005, 'closure')
Select * From dual;


desc pm.PHASES;
select * from pm.PHASES;
commit;



Insert All
  Into pm.JOBS Values (0001, 'CEO', 1500000, 5000000)
  Into pm.JOBS Values (0002, 'CTO', 1000000, 2000000)
  Into pm.JOBS Values (0003, 'CFO', 1200000, 3000000)
  Into pm.JOBS Values (0004, 'CIO', 1000000, 2000000)
  Into pm.JOBS Values (0005, 'Chief Architect', 100000, 1000000)
  Into pm.JOBS Values (0006, 'Software Architect', 65000, 500000)
  Into pm.JOBS Values (0007, 'Project Manager', 50000, 200000)
  Into pm.JOBS Values (0008, 'Engineer', 35000, 100000)
  Into pm.JOBS Values (0009, 'Junior', 20000, 50000)
  Into pm.JOBS Values (0010, 'Intern', 10000, 15000)
Select * From dual;

select * from pm.JOBS;
commit;
```

```
CREATE SEQUENCE seq_client
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;

select * from pm.CLIENTS;

Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'Habib Bank', 'HBL', '02135589874',
'admin@hbl.com', 00101);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'MCB', 'MCB', '02135698745', 'tech@mcb.com',
00101);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'SZABIST', 'SZABIST', '02135236548',
'admin@szabist.edu.pk', 00101);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'AKD', 'AKD', '02135584564',
'AKDsecurities@gmail.com', 00101);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'Nutanix', 'Nutanix', '+971600544445',
'admin@nutanix.com', 00102);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'BMA', 'BMA', '02132158936',
'admin@bma.com.pk', 00101);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'Blizzard', 'Blizzard', '5555551234.',
'tech@blizzard.com', 00105);
 Insert Into pm.CLIENTS Values (seq_client.NEXTVAL, 'JS Bank', 'JS Group', '02132451456',
'admin@JS.com.pk', 00101);

CREATE SEQUENCE seq_skills
MINVALUE 0001
START WITH 0001
INCREMENT BY 1
CACHE 10;

CREATE SEQUENCE seq_skills
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10

drop SEQUENCE seq_skills;
delete from pm.SKILLS;

Insert Into pm.SKILLS Values (seq_skills.NEXTVAL, 'JAVA');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Python');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'C Language');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Communication');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Designing');
Insert Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Problem Solving');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Teamwork');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Writting');
```

```
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Calculations');
Insert  Into pm.SKILLS Values (seq_skills.NEXTVAL, 'Analytical skills');

select * from pm.SKILLS;



CREATE SEQUENCE seq_tools
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10


Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'Server', '25-Dec-2015', 590000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'High-End Laptop', '24-Aug-2019', 480000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'Mid-Range Laptop', '25-May-2018', 200000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'MAC Book', '25-Aug-2019', 410000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'Cloud Server', '20-Nov-2015', 200000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'Adobe Sub', '25-Dec-2019', 250000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'Coding IDE', '15-Apr-2010', 150000);
Insert Into pm.TOOLS Values (SEQ_TOOLS.nextval, 'Testing Devices', '20-Feb-2017', 300000);

select * from pm.TOOLS;

CREATE SEQUENCE seq_emp
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10


select seq_emp.nextval from dual;
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Muhammad', 'Ali', '19-Jan-1980', '34603-3654456-
8', '21-Jan-2005', 'Rufi', 'B-204', 'Gulshan-e-Iqbal', 120000, 'East', '75300', 'm.ali@gmail.com' ,102 , 7);
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Ali', 'Abbas', '20-Oct-1990', '69832-2438170-2',
'22-Jul-2010', 'Marine', 'B-04', 'DHA', 225000, 'South', '75500', 'aliabbas@gmail.com' ,103 , 6);
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Faraz', 'Ahmed', '16-Aug-1960', '45985-4654346-
1', '20-Aug-2000', 'Creek Vista', 'F-0104', 'DHA', 1925000, 'South', '75500', 'fahmed9@gmail.com' ,101 ,
2);
insert into pm.EMPLOYEES Values (seq_emp.nextval, 'Umar', 'Afsar', '15-Jun-1982', '36556-3457534-6',
'12-Dec-2009', 'Creek City', '404', 'DHA', 205000, 'South', '75500', 'umar_696@gmail.com' ,00105 ,
0006);
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Alishan', 'Nadeem', '14-Jul-1985', '36974-3678322-
1', '22-Jan-2010', 'Phase-5', 'S-90/2', 'DHA', 525000, 'South', '75500', 'alinadeem078@gmail.com' ,107 ,
5);
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Omer', 'Shah', '11-Sep-1996', '14598-6789532-1',
'25-Jun-2018', 'KDA', 'E-50', 'Gulshan-e-Iqbal', 12000, 'East', '75300', 'os7895@gmail.com' ,108 , 10);
```

```
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Saba', 'Ahmad', '09-Nov-1966', '47896-4567328-3',
'02-Oct-2002', 'Rufi City', 'H-305', 'Gulshan-e-Iqbal', 95000, 'East', '75300', 'saba_a76@gmail.com' ,106 ,
8);
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Dua', 'Khan', '29-May-1981', '34604-8975432-1',
'18-May-2015', 'Phase-6', 'S-300', 'DHA', 2050000, 'South', '75500', 'duakhan7@gmail.com' ,101 , 3);
insert into pm.EMPLOYEES Values (seq_emp.nextval, 'Zarah', 'Rehman', '25-Mar-1988', '38963-2438170-
6', '08-Feb-2010', 'Phase-5', 'B-240', 'DHA', 95000, 'South', '75500', 'zarah.r786@gmail.com' ,00109 ,
0008;
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Fatimah', 'Nadeem', '27-Apr-1994', '34453-
6785432-7', '29-Apr-2018', 'Chapal', 'A-104', 'Clifton', 12500, 'South', '75600', 'fatimah_121@gmail.com'
,00104 , 0010);

insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Hirdesh','Kumar','25-Mar-1988', '38963-
24385558-6', '08-Feb-2010', 'Phase-7', 'C-40', 'DHA', 95000, 'South', '75500', 'zar6@gmail.com' ,109 ,4);
insert Into pm.EMPLOYEES Values (seq_emp.nextval, 'Elliot', 'Francis', '27-Apr-1994', '34453-6785588-7',
'29-Apr-2018', 'Chapal', 'C-105', 'Clifton', 12500, 'South', '75600', 'fath1@gmail.com' ,104 , 9);
 insert Into pm.EMPLOYEES Values(seq_emp.nextval,'Asad','Rehaman','29-June-1998','34604-8975488-
4','18-May-2', 'Phase-6', 'M-300', 'DHA', 200000, 'South', '75500', 'asad@gmail.com' ,102 , 1);



create sequence seq_department
MINVALUE 101
START WITH 101
INCREMENT BY 1
CACHE 10

insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'HR', 0003, 10004);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'PR', 0001, 10004);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Analysts', 008, 10004);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Devlopers', 0012, 10001);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Testers', 0004, 10001);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Sales', 0007, 10004);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Finance', 0005, 10004);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Marketing', 006, 10004);
insert Into pm.DEPARTMENTS Values (seq_department.nextval, 'Designers', 0009, 10001);

 commit;



CREATE SEQUENCE seq_project
MINVALUE 10001
START WITH 10001
INCREMENT BY 1
CACHE 10
```

```
Insert All
  Into pm.TASK Values (00011, 'User Interface', '14-May-2019', '20-Jan-2020', 10005, 0005)
  Into pm.TASK Values(00012, 'Coding','20-Feb-2019', '20-Feb-2020', 10005, 0004)
  Into pm.TASK Values (00013, 'Cost', '14-Feb-2019', '25-May-2019', 10005, 0005)
  Into pm.TASK Values (00014, 'Testing', '10-Oct-2019', '20-Oct-2020', 10005, 0004)
  Into pm.TASK Values (00015, 'Designing', '14-Feb-2019', '20-Jan-2020', 10005, 0005)
  Into pm.TASK Values (00016, 'Device Test', '20-Jul-2020', '20-Jun-2020', 10005, 0004)
  Into pm.TASK Values (00017, 'Server Check', '20-Jul-2020', '20-Jul-2020', 10005, 0004)
Select * From dual;


Insert  Into pm.PROJECTS Values (seq_project.nextval, 'BMA Trading Terminal', '10-Jun-2010', '20-Oct-
2015', 5000000, 'Complete', 0001,00006);
Insert  into PM.projects values (seq_project.nextval,'AKD Trading', '12-May-2011', '29-Sep-2012',
3000000, 'Complete', 0001, 00004);
  Insert  into PM.PROJECTS  values (seq_project.nextval, 'HBL Online App', '05-Jun-2015', '20-Jul-2019',
5500000, 'Complete', 0001, 00001);
  Insert  into  PM.PROJECTS values (seq_project.nextval, 'MCB Funding App', '05-Jan-2018', '10-Oct-
2018', 6500000, 'Complete', 0001, 00002);
Insert  into PM.PROJECTS values (seq_project.nextval, 'JS Online Banking', '10-Feb-2019', '20-Oct-2020',
5000000, 'Incomplete', 0001, 00008);




desc pm.task;


Insert All
  Into pm.TASK Values (00011, 'User Interface', '14-May-2019', '20-Jan-2020', 10005, 0005)
  Into pm.TASK Values(00012, 'Coding','20-Feb-2019', '20-Feb-2020', 10005, 0004)
  Into pm.TASK Values (00013, 'Cost', '14-Feb-2019', '25-May-2019', 10005, 0005)
  Into pm.TASK Values (00014, 'Testing', '10-Oct-2019', '20-Oct-2020', 10005, 0004)
  Into pm.TASK Values (00015, 'Designing', '14-Feb-2019', '20-Jan-2020', 10005, 0005)
  Into pm.TASK Values (00016, 'Device Test', '20-Jul-2020', '20-Jun-2020', 10005, 0004)
  Into pm.TASK Values (00017, 'Server Check', '20-Jul-2020', '20-Jul-2020', 10005, 0004)
Select * From dual;

select * from pm.task;

Insert All
  Into pm.PHONE_NUMBER Values ('0300-2269875', 0001)
 Into pm.PHONE_NUMBER Values  ('0345-3456785', 0002)
 Into pm.PHONE_NUMBER Values  ('0346-2353456', 0003)
 Into pm.PHONE_NUMBER Values  ('0333-2465455', 0004)
 Into pm.PHONE_NUMBER Values  ('0323-3665875', 0005)
 Into pm.PHONE_NUMBER Values  ('0345-6523264', 0006)
```

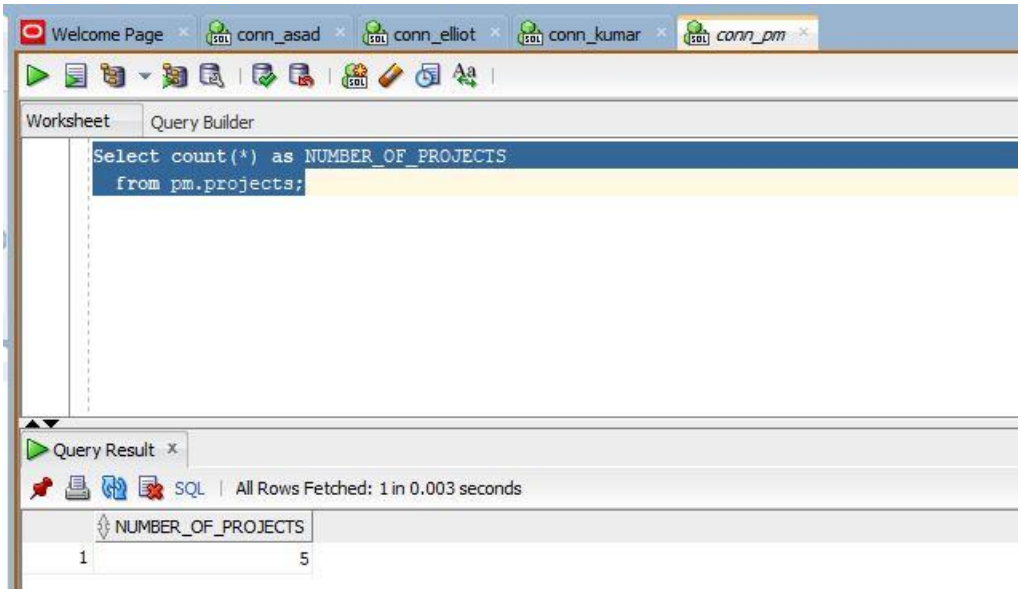
```
 Into pm.PHONE_NUMBER Values ('0322-2356555', 0007)
 Into pm.PHONE_NUMBER Values  ('0300-2456557', 0008)
 Into pm.PHONE_NUMBER Values  ('0325-2234538', 0009)
 Into pm.PHONE_NUMBER Values  ('0322-2098345', 0010)
Select * From dual;

select * from pm.PHONE_NUMBER;



Insert All
  Into pm.SKILLS_EMPLOYEES Values (0006, 0008)
   Into pm.SKILLS_EMPLOYEES Values (0003, 0003)
  Into pm.SKILLS_EMPLOYEES Values (0002, 0010)
  Into pm.SKILLS_EMPLOYEES Values (0001, 0003)
  Into pm.SKILLS_EMPLOYEES Values (0001, 0010)
    Into pm.SKILLS_EMPLOYEES Values (0007, 0010)
  Into pm.SKILLS_EMPLOYEES Values (0007, 0009)
   Into pm.SKILLS_EMPLOYEES Values (0008, 0002)
  Into pm.SKILLS_EMPLOYEES Values (0010, 0002)
  Into pm.SKILLS_EMPLOYEES Values (0009, 0006)
Select * From dual;

select * from pm.SKILLS_EMPLOYEES;



Insert All
  Into pm.EMPLOYEE_PROJECTS  Values (0001, 10005, '10-Feb-2019', '20-Oct-2020')
  Into pm.EMPLOYEE_PROJECTS Values (0010, 10005, '20-Feb-2019', '20-Feb-2020')
  Into pm.EMPLOYEE_PROJECTS   Values (0009, 10005, '14-Feb-2019', '25-May-2019')
  Into pm.EMPLOYEE_PROJECTS  Values (0002, 10005, '20-Jul-2020', '20-Jun-2020')
  Into pm.EMPLOYEE_PROJECTS   Values (0005, 10005, '14-Feb-2019', '25-May-2019')
Select * From dual;

select * from  pm.EMPLOYEE_PROJECTS;



Insert All
  Into pm.PROJECT_TOOLS Values (10005, 00002, '10-Feb-2019', '20-Oct-2020')
  Into pm.PROJECT_TOOLS Values (10005, 00007, '20-Feb-2019', '20-Feb-2020')
  Into pm.PROJECT_TOOLS Values (10005, 00004, '20-Feb-2019', '20-Feb-2020')
  Into pm.PROJECT_TOOLS Values (10005, 00005, '20-Jul-2020', '20-Jul-2020')
  Into pm.PROJECT_TOOLS Values (10005, 00001, '20-Jul-2020', '20-Jul-2020')
  Into pm.PROJECT_TOOLS Values (10005, 00008, '20-Jul-2020', '20-Jun-2020')
Select * From dual;
```

```
Insert All
  Into pm.EMP_TASK Values (00011,0009,  'Done')
  Into pm.EMP_TASK Values (00011,0010,  'Done')
  Into pm.EMP_TASK Values (00013,0005,  'Done')
  Into pm.EMP_TASK Values (00014,0004,  'Done')
  Into pm.EMP_TASK Values (00015,0009,  'Done')
  Into pm.EMP_TASK Values (00012,0010,  'Done')
  Into pm.EMP_TASK Values (00017,0006,  'Done')
Select * From dual;
```

Meaningful Queries on following topics along with results.

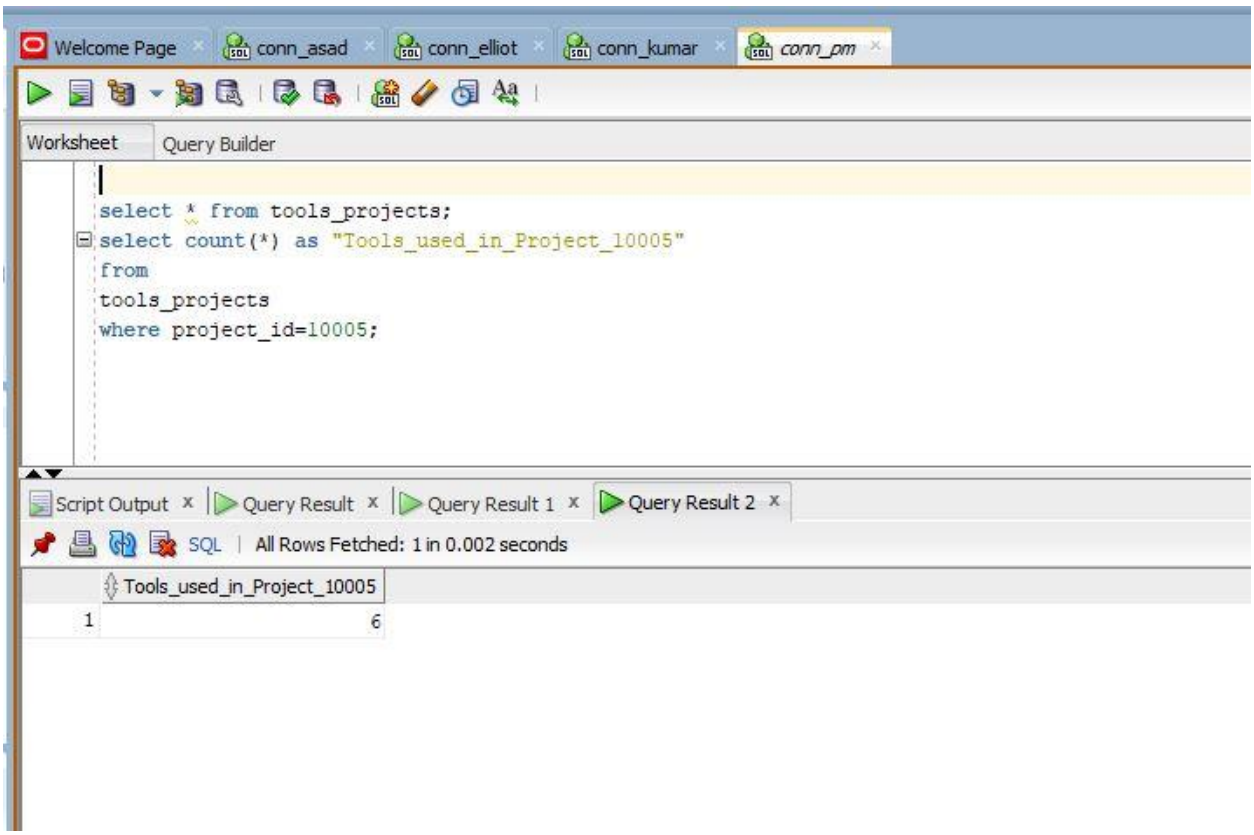
1. One or more group Functions on one table without where clause and without group by clause.

   Select count(*) as NUMBER_OF_PROJECTS
   from projects;



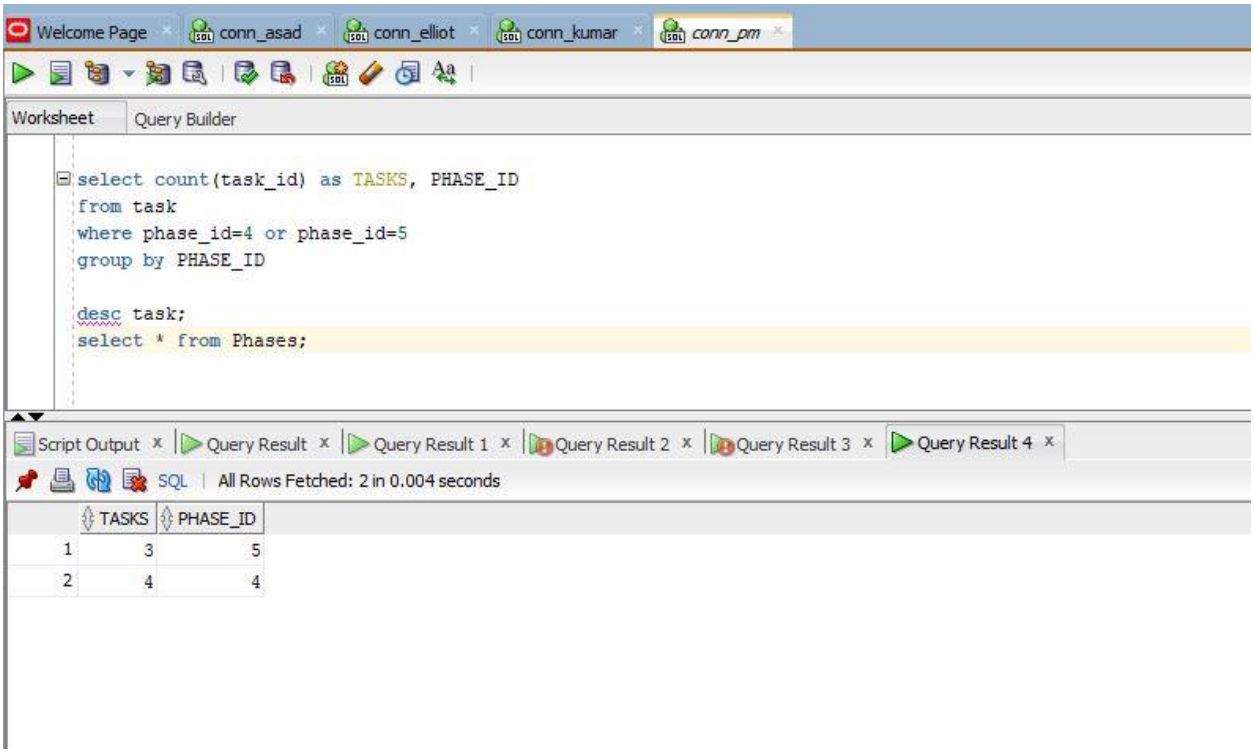
2. One or more group Functions on one table with where clause and without group by clause.

```
select count(*) as "Tools_used_in_Project_10005"
from
tools_projects
where project_id=10005;
```



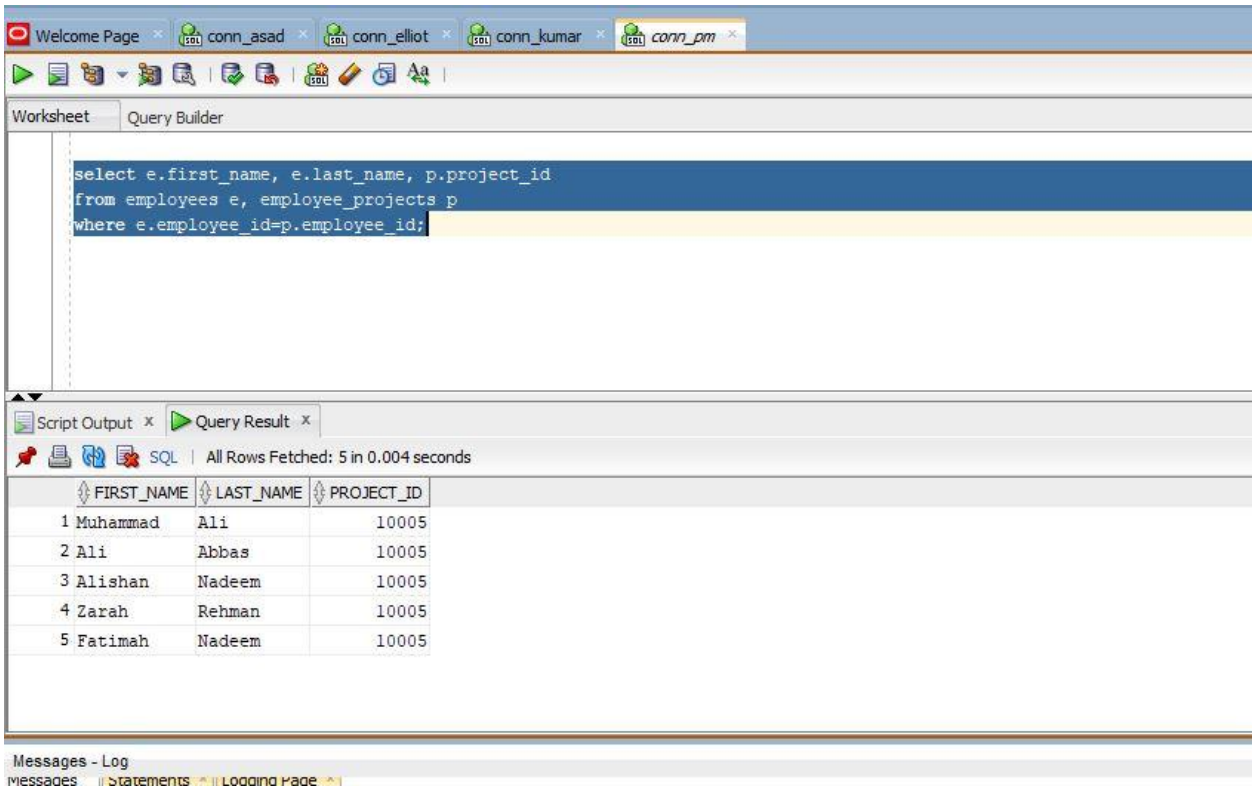
3. One or more attributes along with one or more group Functions on one table with where clause and with group by clause.

```
select count(task_id) as TASKS, PHASE_ID
from task
where phase_id=4 or phase_id=5
group by PHASE_ID
```



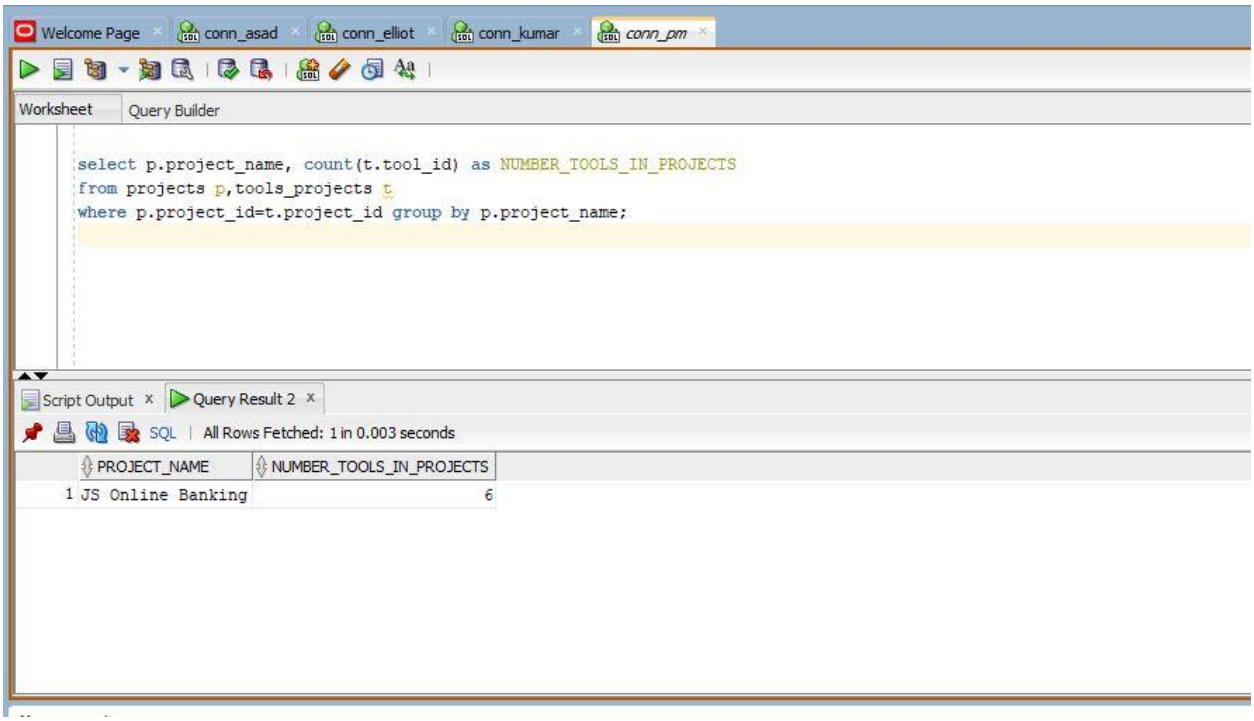
4. One or more relevant attributes from two tables with Equi-join on two tables.

```
select e.first_name, e.last_name, p.project_id
from employees e, employee_projects p
where e.employee_id=p.employee_id;
```



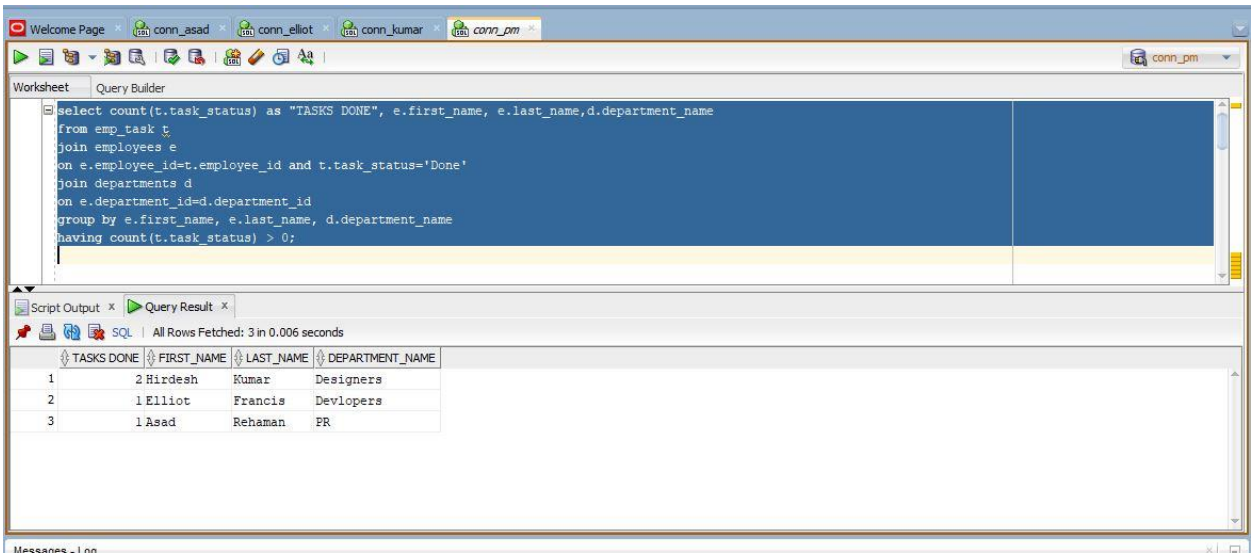
5. One or more relevant attributes from two tables with Equi-join on two tables with where clause and group-by clause.

```
select p.project_name, count(t.tool_id) as NUMBER_TOOLS_IN_PROJECTS
from projects p,tools_projects t
where p.project_id=t.project_id group by p.project_name;
```



6. One or more relevant attributes from two tables with Equi-join on three tables with group-by clause and having condition.

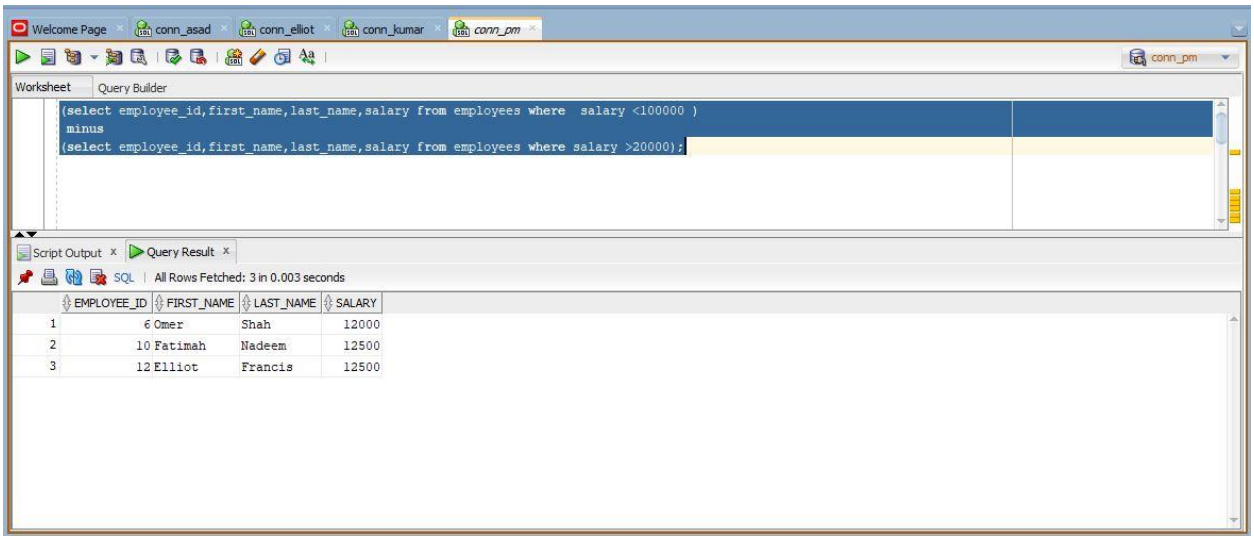
```
select count(t.task_status) as "TASKS DONE", e.first_name, e.last_name,d.department_name
from emp_task t
join employees e
on e.employee_id=t.employee_id and t.task_status='Done'
join departments d
on e.department_id=d.department_id
group by e.first_name, e.last_name, d.department_name
having count(t.task_status) > 0;
```



7. three queries on

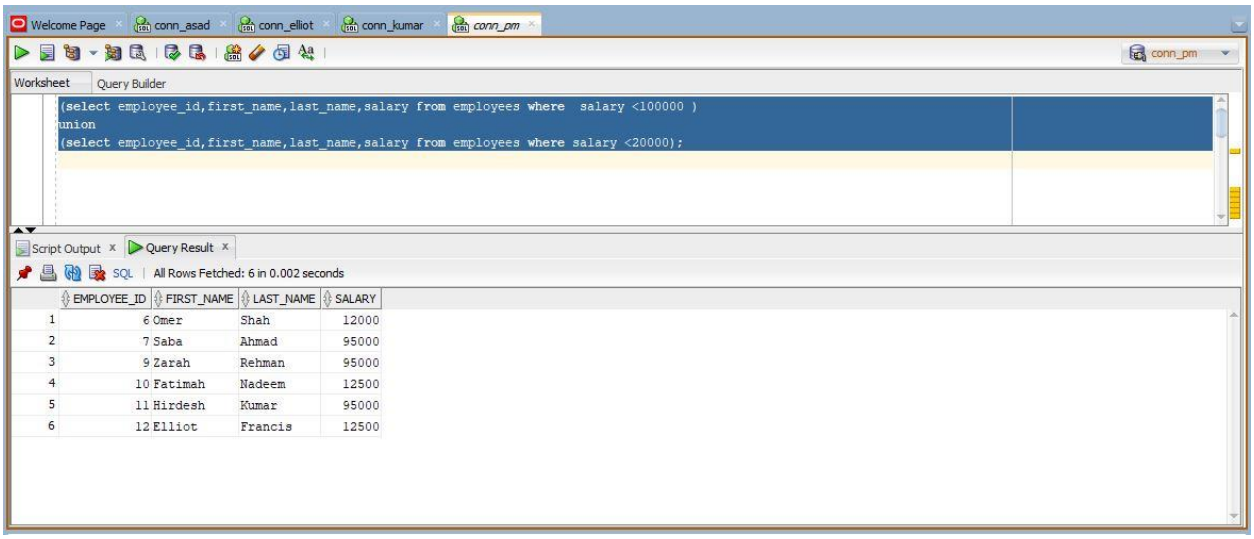
a.minus

(select employee_id,first_name,last_name,salary from employees where salary <100000)
minus
(select employee_id,first_name,last_name,salary from employees where salary >20000);



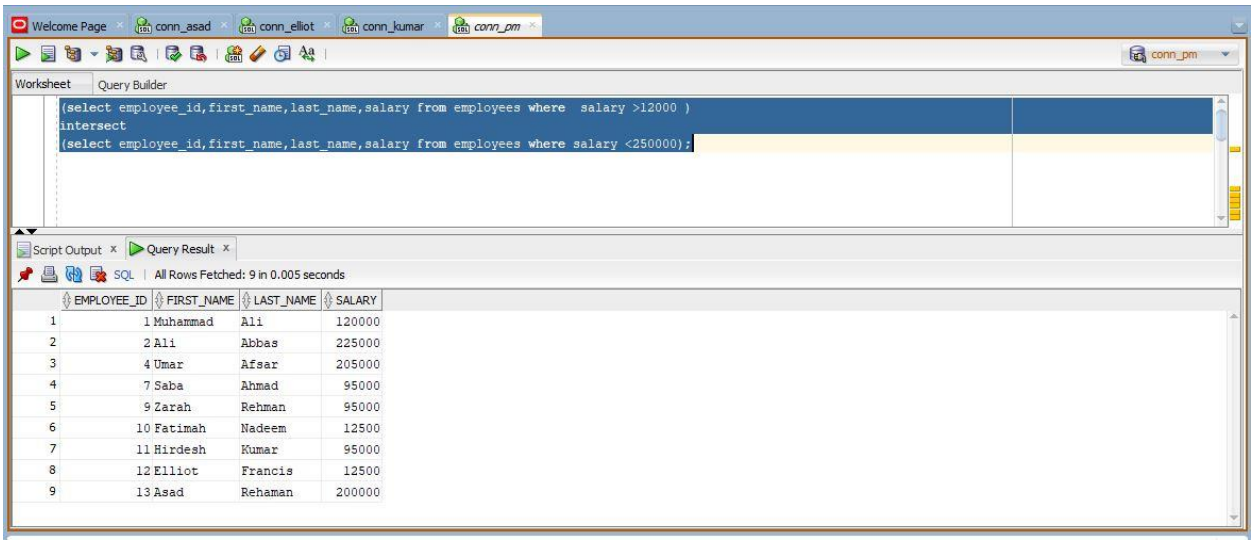
b. union

(select employee_id,first_name,last_name,salary from employees where salary <100000)
union
(select employee_id,first_name,last_name,salary from employees where salary <20000);



c. intersect

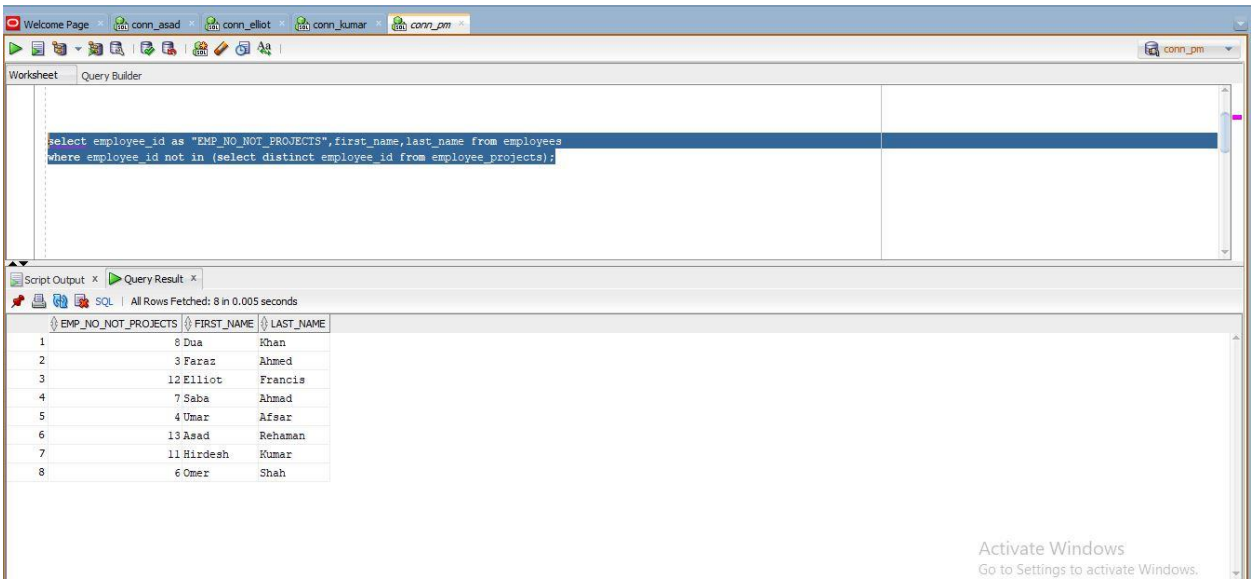
(select employee_id,first_name,last_name,salary from employees where salary >12000)
intersect
(select employee_id,first_name,last_name,salary from employees where salary <250000);



	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
1	1	Muhammad	Ali	120000
2	2	Ali	Abbas	225000
3	4	Umar	Afsar	205000
4	7	Saba	Ahmad	95000
5	9	Zarah	Rehman	95000
6	10	Fatimah	Nadeem	12500
7	11	Hirdesh	Kumar	95000
8	12	Elliot	Francis	12500
9	13	Asad	Rehaman	200000

8. Table 1 where clause containing not in condition on a column of another table through select sub-query. e.g. select .... from table1 where ... not in (select ... from table 2 and so on).

select employee_id as "EMP_NO_NOT_PROJECTS",first_name,last_name from employees
where employee_id not in (select distinct employee_id from employee_projects);



9. 3 Outer joins:

a. left,

```
select e.employee_id,e.first_name,e.last_name,p.task_id
from employees e
left join emp_task p
on e.employee_id=p.employee_id;
```

```
SQL> select e.employee_id,e.first_name,e.last_name,p.task_id
  2  from employees e
  3  left join emp_task p
  4  on e.employee_id=p.employee_id;

EMPLOYEE_ID FIRST_NAME                LAST_NAME                   TASK_ID
----------- ------------------------- ------------------------- ----------
         13 Asad                      Rehaman                            5
         11 Hirdesh                   Kumar                              9
         11 Hirdesh                   Kumar                             10
         12 Elliot                    Francis                           10
          5 Alishan                   Nadeem
          8 Dua                       Khan
          3 Faraz                     Ahmed
         10 Fatimah                   Nadeem
          7 Saba                      Ahmad
          4 Umar                      Afsar
          9 Zarah                     Rehman

EMPLOYEE_ID FIRST_NAME                LAST_NAME                   TASK_ID
----------- ------------------------- ------------------------- ----------
          1 Muhammad                  Ali
          2 Ali                       Abbas
          6 Omer                      Shah

14 rows selected.
```

b. right

select e.employee_id,e.first_name,e.last_name,p.task_id
from employees e
right join emp_task p
on e.employee_id=p.employee_id;

```
SQL> ed
Wrote file afiedt.buf

  1  select e.employee_id,e.first_name,e.last_name,p.task_id
  2  from employees e
  3  right join emp_task p
  4* on e.employee_id=p.employee_id
SQL> /

EMPLOYEE_ID FIRST_NAME                LAST_NAME                    TASK_ID
----------- ------------------------- ------------------------- ----------
         11 Hirdesh                   Kumar                             10
         11 Hirdesh                   Kumar                              9
         12 Elliot                    Francis                           10
         13 Asad                      Rehaman                            5
                                                                         9
                                                                         6
                                                                         4

7 rows selected.
```

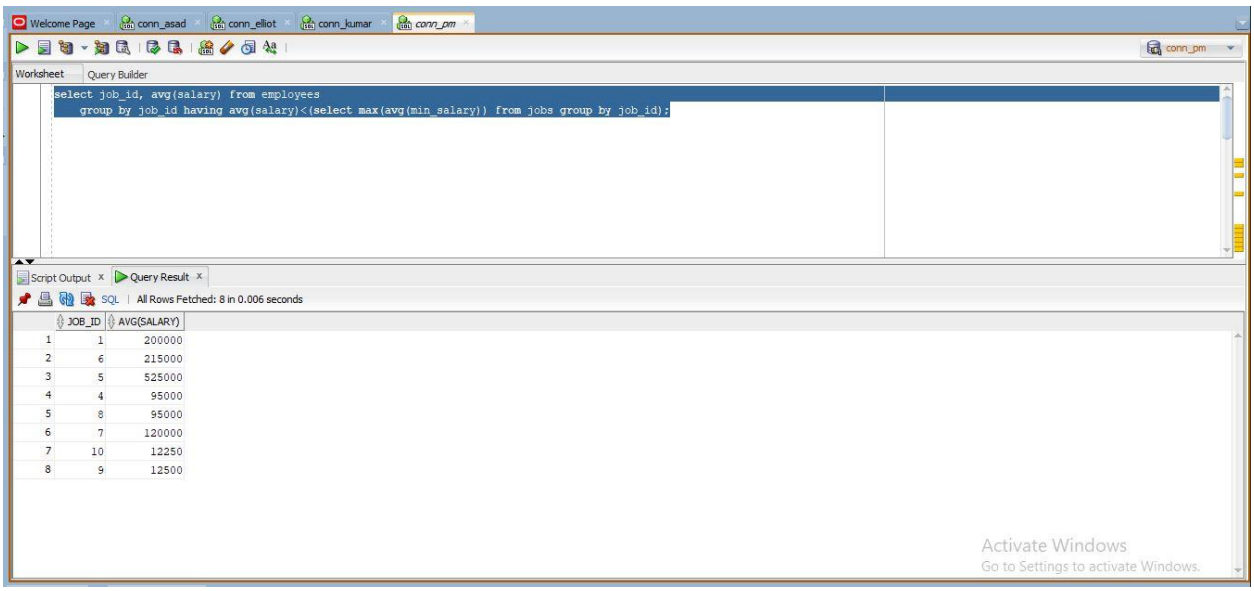
and c. full outer join

select e.employee_id,e.first_name,e.last_name,p.task_id
from employees e
full outer join emp_task p
on e.employee_id=p.employee_id

```
SQL> ed
Wrote file afiedt.buf

  1  select e.employee_id,e.first_name,e.last_name,p.task_id
  2  from employees e
  3  full outer join emp_task p
  4* on e.employee_id=p.employee_id
SQL> /

EMPLOYEE_ID FIRST_NAME                LAST_NAME                    TASK_ID
----------- ------------------------- ------------------------- ----------
          1 Muhammad                  Ali
          2 Ali                       Abbas
          3 Faraz                     Ahmed
          4 Umar                      Afsar
          5 Alishan                   Nadeem
          6 Omer                      Shah
          7 Saba                      Ahmad
          8 Dua                       Khan
          9 Zarah                     Rehman
         10 Fatimah                   Nadeem
         11 Hirdesh                   Kumar                            10

EMPLOYEE_ID FIRST_NAME                LAST_NAME                    TASK_ID
----------- ------------------------- ------------------------- ----------
         11 Hirdesh                   Kumar                             9
         12 Elliot                    Francis                          10
         13 Asad                      Rehaman                           5
                                                                        9
                                                                        6
                                                                        4

17 rows selected.
```

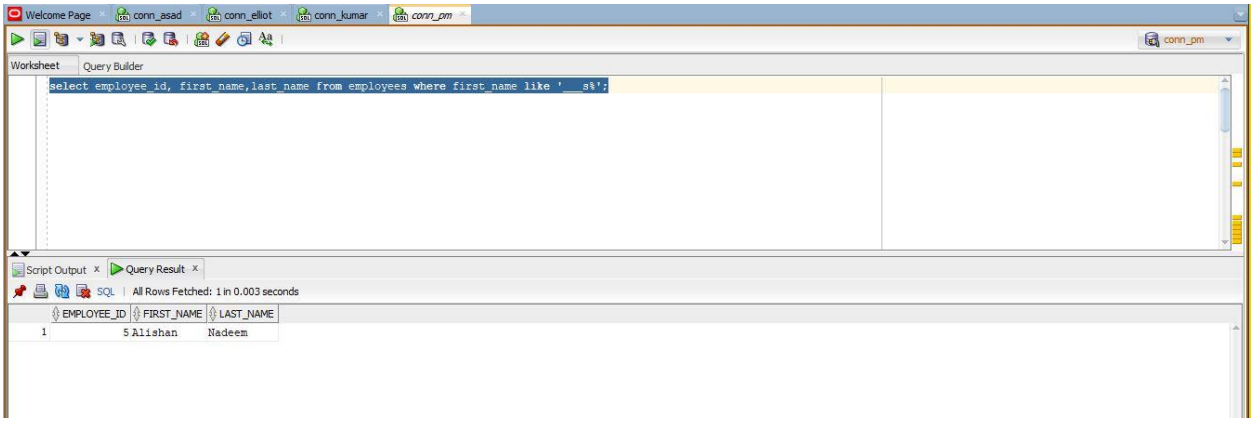
10. use of nested functions to at least two levels.

```
select job_id, avg(salary) from employees
group by job_id having avg(salary)<(select max(avg(min_salary)) from jobs group by job_id);
```



11. example of a query which uses of _ and % along with like in where clause.

select employee_id, first_name,last_name from employees where first_name like '___s%';



12. Use of sub-queries in from. One or more attributes along with one or more group Functions on one table with where clause and with group by clause.

select avg(e.salary) as "AVGSALARY ON ACTIVE_PROJECTS",e.first_name,e.last_name
from employees e
where employee_id in (select distinct employee_id from employee_projects)
group by e.first_name,e.last_name
order by e.first_name

```
SQL> ed
Wrote file afiedt.buf

  1  select avg(e.salary) as "AVGSALARY ON ACTIVE_PROJECTS",e.first_name,e.last_name
  2  from employees e
  3  where employee_id in (select distinct employee_id from employee_projects)
  4  group by e.first_name,e.last_name
  5* order by e.first_name
SQL> /

AVGSALARY ON ACTIVE_PROJECTS FIRST_NAME               LAST_NAME
---------------------------- ------------------------ ------------------------
                      225000 Ali                      Abbas
                      525000 Alishan                  Nadeem
                       12500 Fatimah                  Nadeem
                      120000 Muhammad                 Ali
                       95000 Zarah                    Rehman
```