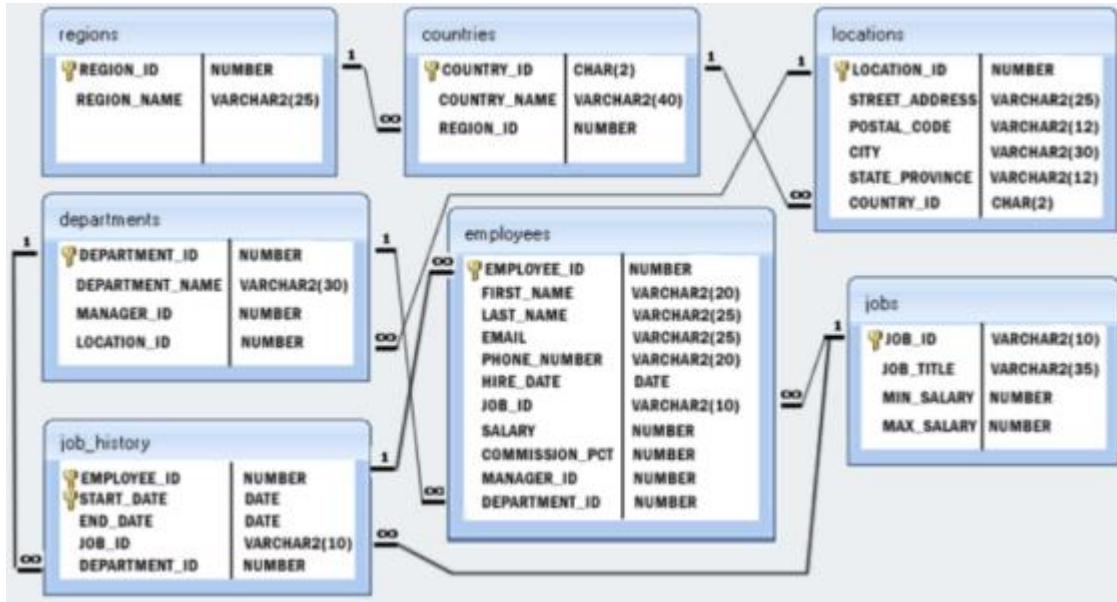


Practical – 3

Aim – Create a database and create the tables given below and execute the mentioned queries on this database.

Schema –



Tables –

- Regions –**

Query - `CREATE TABLE regions(REGION_ID INT PRIMARY KEY, REGION_NAME VARCHAR(25));`

Table schema

```
labwednesday=# \d regions
Table "public.regions"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 region_id    | integer         |           | not null |
 region_name  | character varying(25) |           |          |
Indexes:
    "regions_pkey" PRIMARY KEY, btree (region_id)
Referenced by:
    TABLE "countries" CONSTRAINT "countries_region_id_fkey" FOREIGN KEY (region_id) REFERENCES regions(region_id)
```

Table Data

```
labwednesday=# select * from regions;
 region_id | region_name
-----+-----
1 | Africa
2 | Antarctica
3 | Asia
4 | Europe
5 | North america
6 | South America
7 | Oceania
(7 rows)
```

- **Countries -**

Query – *CREATE TABLE countries (COUNTRY_ID CHAR(2) primary key, COUNTRY_NAME VARCHAR(40), REGION_ID INT REFERENCES regions(REGION_ID));*

Table schema

```
labwednesday=# \d countries;
          Table "public.countries"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
country_id    | character(2)    |           | not null |
country_name  | character varying(40) |           |          |
region_id     | integer         |           |          |
Indexes:
    "countries_pkey" PRIMARY KEY, btree (country_id)
Foreign-key constraints:
    "countries_region_id_fkey" FOREIGN KEY (region_id) REFERENCES regions(region_id)
Referenced by:
    TABLE "locations" CONSTRAINT "locations_country_id_fkey" FOREIGN KEY (country_id) REFERENCES countries(country_id)
```

Table data

```
labwednesday=# select * from countries;
 country_id | country_name | region_id
-----+-----+-----
AL          | Algeria      |         1
AN          | Angola       |         1
BE          | Benin        |         1
AF          | Afghanistan  |         3
IN          | India        |         3
GR          | Germany      |         3
JP          | Japan        |         3
GL          | Greenland    |         2
PT          | Portland     |         4
US          | United states|         4
(10 rows)
```

- **Locations –**

Query - *CREATE TABLE locations (LOCATION_ID INT PRIMARY KEY, STREET_ADDRESS VARCHAR(25), POSTAL_CODE VARCHAR(12), CITY VARCHAR(30), STATE_PROVINCE VARCHAR(12), COUNTRY_ID CHAR(2) REFERENCES countries(COUNTRY_ID));*

Table Schema

```
labwednesday=# \d locations
          Table "public.locations"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
location_id   | integer         |           | not null |
street_address | character varying(25) |           |          |
postal_code   | character varying(12) |           |          |
city          | character varying(30) |           |          |
state_province | character varying(12) |           |          |
country_id    | character(2)     |           |          |
Indexes:
    "locations_pkey" PRIMARY KEY, btree (location_id)
Foreign-key constraints:
    "locations_country_id_fkey" FOREIGN KEY (country_id) REFERENCES countries(country_id)
Referenced by:
    TABLE "departments" CONSTRAINT "departments_location_id_fkey" FOREIGN KEY (location_id) REFERENCES locations(location_id)
```

Table Data

```
labwednesday=# select * from locations;
 location_id | street_address | postal_code | city      | state_province | country_id
-----+-----+-----+-----+-----+-----
1 | Oran-algeria  | 31000      | Oran      | Or-Province    | AL
2 | Shin-angola   | 11000      | Shin      | Sp-province    | AN
3 | Delhi-India   | 110000     | Delhi     | Punjab         | IN
4 | Mysuru-India  | 570000     | Mysuru    | Karnataka      | IN
5 | Karnataka-India | 560000     | Bangaluru | Karnataka      | IN
6 | Mumbai-India  | 90000      | Mumbai    | Maharashtra    | IN
7 | Punjab-India  | 120000     | Punjab    | Punjab         | IN
8 | Jammu-India   | 150000     | Jammu     | Jammu          | IN
9 | Assam-India   | 980000     | Assam     | Assam          | IN
(9 rows)
```

- **Departments –**

Query - CREATE TABLE departments (DEPARTMENT_ID INT, DEPARTMENT_NAME VARCHAR(30), MANAGER_ID INT, LOCATION_ID INT REFERENCES locations(location_id), primary key(department_id));

Table schema

```
labwednesday=# \d departments;
          table "public.departments"
   Column |          type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 department_id | integer                |           | not null |
 department_name | character varying(30) |           |          |
 manager_id    | integer                |           |          |
 location_id   | integer                |           |          |
 (Indexes)
    "departments_pkey" PRIMARY KEY, btree (department_id)
Foreign-key constraints:
    "departments_location_id_fkey" FOREIGN KEY (location_id) REFERENCES locations(location_id)
Referenced by:
    TABLE "employees" CONSTRAINT "employees_department_id_fkey" FOREIGN KEY (department_id) REFERENCES departments(department_id)
    TABLE "job_history" CONSTRAINT "job_history_department_id_fkey" FOREIGN KEY (department_id) REFERENCES departments(department_id)
```

Table Data

```
labwednesday=# select * from departments;
```

department_id	department_name	manager_id	location_id
1	Storage domain	1	1
3	Network domain	3	1
4	Wintel domain	4	3
5	Technical domain	5	1
6	Services domain	6	3
7	Hardware domain	7	1
8	Software domain	8	3
9	Nightly domain	9	1
10	Early domain	10	3
2	IT domain	2	3
11	Sh clerk	11	3

(11 rows)

- **Employees –**

Query - CREATE TABLE employees(employee_id int, first_name varchar(20), last_name varchar(25), email varchar(25), phone_number varchar(20), hire_date date, job_id varchar(10) references jobs(job_id), salary int, commission_pct int, manager_id int, department_id int references departments(department_id), primary key(employee_id));

Table schema

```
labwednesday=# \d employees;
          table "public.employees"
   Column |          type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 employee_id | integer                |           | not null |
 first_name  | character varying(20) |           |          |
 last_name   | character varying(25) |           |          |
 email       | character varying(25) |           |          |
 phone_number | character varying(20) |           |          |
 hire_date   | date                  |           |          |
 job_id      | character varying(10) |           |          |
 salary      | integer                |           |          |
 commission_pct | integer                |           |          |
 manager_id  | integer                |           |          |
 department_id | integer                |           |          |
 (Indexes)
    "employees_pkey" PRIMARY KEY, btree (employee_id)
Foreign-key constraints:
    "employees_department_id_fkey" FOREIGN KEY (department_id) REFERENCES departments(department_id)
    "fk_employees_jobs" FOREIGN KEY (job_id) REFERENCES jobs(job_id)
Referenced by:
    TABLE "job_history" CONSTRAINT "job_history_employee_id_fkey" FOREIGN KEY (employee_id) REFERENCES employees(employee_id)
```

Table Data

```
labwednesday=# select * from employees;
```

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	Ankit	Gupta	ankit@gmail.com	1234	2020-12-23	2	10	10	2	2
2	It	Gupta	ankit@gmail.com	1234	2020-12-23	3	10	10	3	3
3	Kit	Gupta	ankit@gmail.com	1234	2020-12-23	4	10	10	4	4
4	Nkit	Gupta	ankit@gmail.com	1234	2020-12-23	2	10	10	2	2
5	Ank	Gupta	ankit@gmail.com	1234	2020-12-23	1	10	10	1	1
6	An	Gupta	ankit@gmail.com	1234	2020-12-23	5	10	10	5	5
7	A	Gupta	ankit@gmail.com	1234	2020-12-23	5	10	10	5	5
8	T	Gupta	ankit@gmail.com	1234	2020-12-23	6	10	10	6	6
9	Nki	Gupta	ankit@gmail.com	1234	2020-12-23	8	10	10	8	8
10	Kit	Ga	ankit@gmail.com	1234	2020-12-23	9	10	10	9	9
11	Sensex	Bull	bull@gmail.com	1234	2020-12-23	4	5	10	4	4
12	Puca	Pu	popo@gmail.com	1234	2020-12-23	11	6	12	11	11

(12 rows)

- Jobs –

Query- *CREATE TABLE jobs (job_id varchar(10) primary key, job_title varchar(35), min_salary int, max_salary int);*

Table schema

```
labwednesday=# \d jobs;
              Table "public.jobs"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 job_id       | character varying(10) |           | not null |
 job_title    | character varying(35) |           |          |
 min_salary   | integer          |           |          |
 max_salary   | integer          |           |          |
Indexes:
    "jobs_pkey" PRIMARY KEY, btree (job_id)
Referenced by:
    TABLE "employees" CONSTRAINT "fk_employees_jobs" FOREIGN KEY (job_id) REFERENCES jobs(job_id)
    TABLE "job_history" CONSTRAINT "job_history_job_id_fkey" FOREIGN KEY (job_id) REFERENCES jobs(job_id)
```

Table Data

```
labwednesday=# select * from jobs;
 job_id |  job_title  | min_salary | max_salary
-----+-----+-----+-----
    1   | Storage admin |          1 |         10
    3   | Network admin |          3 |         10
    4   | Wintel admin |          4 |         30
    5   | Technical admin |          5 |         10
    6   | Services admin |          6 |         30
    7   | Hardware admin |          7 |         10
    8   | Software admin |          8 |         30
    9   | Nightly admin |          9 |         10
   10   | Early admin |         10 |         30
    2   | Backup admin |         10 |         30
   11   | SH_clerk |         10 |         10
(11 rows)
```

- Jobs history –

Query - *CREATE TABLE job_history(employee_id int references employees(employee_id), start_date date, end_date date, job_id varchar(10) references jobs(job_id), department_id int references departments(department_id), primary key(employee_id, start_date));*

Table schema

```
labwednesday=# \d job_history;
              Table "public.job_history"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 employee_id   | integer         |           | not null |
 start_date    | date            |           | not null |
 end_date      | date            |           |          |
 job_id        | character varying(10) |           |          |
 department_id | integer         |           |          |
Indexes:
    "job_history_pkey" PRIMARY KEY, btree (employee_id, start_date)
Foreign-key constraints:
    "job_history_department_id_fkey" FOREIGN KEY (department_id) REFERENCES departments(department_id)
    "job_history_employee_id_fkey" FOREIGN KEY (employee_id) REFERENCES employees(employee_id)
    "job_history_job_id_fkey" FOREIGN KEY (job_id) REFERENCES jobs(job_id)
```

Table Data

```
labwednesday=# select * from job_history;
 employee_id | start_date | end_date | job_id | department_id
-----+-----+-----+-----+-----
          2 | 2020-12-30 | 2020-12-31 | 2      |            2
          3 | 2020-12-30 | 2020-12-31 | 4      |            4
          4 | 2020-12-30 | 2020-12-31 | 2      |            2
          5 | 2020-12-30 | 2020-12-31 | 1      |            1
          6 | 2020-12-30 | 2020-12-31 | 1      |            1
          7 | 2020-12-30 | 2020-12-31 | 1      |            1
          8 | 2020-12-30 | 2020-12-31 | 1      |            1
          9 | 2020-12-30 | 2020-12-31 | 1      |            1
         10 | 2020-12-30 | 2020-12-31 | 1      |            1
(9 rows)
```

Queries –

- 1) Write a query to find the first_name, last_name and salaries of the employees who have a higher salary than the employee who's last_name is Bull.

Sol.

Query - *select first_name, last_name, salary from employees where salary > (select salary from employees where last_name='Bull');*

Output :

```
labwednesday=# select first_name, last_name, salary from employees where
salary > (select salary from employees where last_name='Bull');
 first_name | last_name | salary
-----+-----+-----
Ankit      | Gupta    | 10
It         | Gupta    | 10
Kit        | Gupta    | 10
Nkit       | Gupta    | 10
Ank        | Gupta    | 10
An         | Gupta    | 10
A          | Gupta    | 10
T          | Gupta    | 10
Nki        | Gupta    | 10
Kit        | Ga       | 10
Puca       | Pu       | 6
(11 rows)
```

- 2) Write a SQL subquery to find the first_name and last_name of all employees who works in the IT department.

Sol.

Query - *select first_name, last_name from employees where department_id in (select department_id from departments where department_name LIKE 'IT%');*

Output :

```
labwednesday=# select first_name, last_name from em
ployees where department_id in (select department_i
d from departments where department_name LIKE 'IT%'
);
 first_name | last_name
-----+-----
Ankit      | Gupta
Nkit       | Gupta
(2 rows)
```

- 3) Write a SQL subquery to find the first_name and last_name of the employees under a manager who works for a department based in the United States.

Sol.

Query - *select first_name, last_name from employees where manager_id in (select manager_id from departments where location_id in (select location_id from locations where country_id='US'));*

Output :

```
labwednesday=# select first_name, last_name f
rom employees where manager_id in (select man
ager_id from departments where location_id in
(select location_id from locations where cou
ntry_id='US'));
 first_name | last_name
-----+-----
(0 rows)
```

4) Write a SQL subquery to find the first_name and last_name of the employees who are working as a manager.

Sol.

Query- *select first_name, last_name from employees where employee_id in (select manager_id from departments);*

Output :

```
labwednesday=# select first_name, last_name f
rom employees where employee_id in (select ma
nager_id from departments);
 first_name | last_name
-----+-----
Ankit      | Gupta
It         | Gupta
Kit        | Gupta
Nkit       | Gupta
Ank        | Gupta
An         | Gupta
A          | Gupta
T          | Gupta
Nki        | Gupta
Kit        | Ga
Sensex    | Bull
(11 rows)
```

5) Write a SQL subquery to find the first_name, last_name and salary, which is greater than the average salary of the employees.

Sol.

Query- *select first_name, last_name, salary from employees where salary > (select avg(salary) from employees);*

Output :

```
labwednesday=# select first_name, last_name,
salary from employees where salary > (select
avg(salary) from employees);
 first_name | last_name | salary
-----+-----+-----
Ankit      | Gupta    | 10
It         | Gupta    | 10
Kit        | Gupta    | 10
Nkit       | Gupta    | 10
Ank        | Gupta    | 10
An         | Gupta    | 10
A          | Gupta    | 10
T          | Gupta    | 10
Nki        | Gupta    | 10
Kit        | Ga       | 10
(10 rows)
```

6) Write a SQL subquery to find the first_name, last_name and salary, which is equal to the minimum salary for this post, he/she is working on.

Sol.

Query - *select first_name, salary from employees e where salary = (select min_salary from jobs where job_id=e.job_id);*

Output :

```
labwednesday=# select first_name, salary from
employees e where salary = (select min_salar
y from jobs where job_id=e.job_id);
 first_name | salary
-----+-----
Ankit      | 10
Nkit       | 10
(2 rows)
```

7) Write a SQL Subquery to find the first_name, last_name and salary of the employees who earn more than the average salary and works in any of the IT departments.

Sol.

Query- select first_name, last_name, salary from employees where salary > (select avg(salary) from employees) AND department_id = (select department_id from departments where department_name LIKE 'IT%');

Output :

```
labwednesday=# select first_name, last_name,
salary from employees where salary > (select
avg(salary) from employees) AND department_id
= (select department_id from departments whe
re department_name LIKE 'IT%');
 first_name | last_name | salary
-----+-----+-----
Ankit      | Gupta    | 10
Nkit       | Gupta    | 10
(2 rows)
```

8) Write a subquery to find the first_name, last_name, job_id and salary of the employees who draws a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = 'SH_CLERK'). Sort the results on salary from the lowest to highest.

Sol.

Query - select first_name, last_name, job_id, salary from employees where salary > (select max(salary) from employees where job_id='11');

Output :

```
labwednesday=# select first_name, last_name,
job_id, salary from employees where salary >
(select max(salary) from employees where job_
id='11');
 first_name | last_name | job_id | salary
-----+-----+-----+-----
Ankit      | Gupta    | 2      | 10
It         | Gupta    | 3      | 10
Kit        | Gupta    | 4      | 10
Nkit       | Gupta    | 2      | 10
Ank        | Gupta    | 1      | 10
An         | Gupta    | 5      | 10
A          | Gupta    | 5      | 10
T          | Gupta    | 6      | 10
Nki        | Gupta    | 8      | 10
Kit        | Ga       | 9      | 10
(10 rows)
```