1. Write a SQL statement to create a simple table countries including columns country\_id,country\_name and region\_id.

create table countries(country\_id integer, country\_name char(20), region\_id int);

                   Table "public.countries"

    Column   |     Type     | Collation | Nullable | Default

--------------+---------------+-----------+----------+-----------------------------------------------------

 country\_id  | integer       |           |        |

 Country\_name | character(20)   |           |          |

 region\_id   | integer   |   |         |

1. Write a SQL statement to create a simple table countries including columns country\_id, country\_name and region\_id which already exist.

create table if not exists countries(country\_id integer, country\_name char(20), region\_id int);

relation "countries" already exists, skipping

1. Write a SQL statement to create the structure of a table dup\_countries similar to countries.

create table dup\_countries as table countries with no data;

cdac1=# \d dup\_countries

                  Table "public.dup\_countries"

    Column    |     Type     | Collation | Nullable | Default

-------------- +--------------- +----------- +---------- +---------

 country\_id  | integer    |          |          |

  Country\_name | character(20) |         |         |

 region\_id   | integer       |         |          |

1. Write a SQL statement to create a duplicate copy of countries table including structure and data by name dup\_countries.

create table dup\_countries as table countries

cdac1=# select \* from dup2\_countries;

 country\_id |     country\_name     | region\_id

------------ +---------------------- +-----------

           1 | India                |         1

          2 | australia            |         2

           3 | USA                  |         3

1. Write a SQL statement to create a table named countries, including country\_id, country\_name and region\_id and make sure that no duplicate data against column country\_id will be allowed at the time of insertion.

create table if not exists countries2(country\_id integer primary key, country\_name char(20), region\_id serial);

cdac1=# \d countries2

                                      Table "public.countries2"

    Column    |     Type      | Collation | Nullable |                    Default

-------------- +--------------- +----------- +----------+-----------------------------------------------

  country\_id   | integer       |           | not null |

 country\_name | character(20) |           |          |

 region\_id    | integer       |           | not null | nextval('countries2\_region\_id\_seq'::regclass)

Indexes:

    "countries2\_pkey" PRIMARY KEY, btree (country\_id)

**Assignment 2**

1. Write a SQL statement to create a table named jobs including columns job\_id, job\_title, min\_salary and max\_salary, and make sure that, the default value for job\_title is blank and min\_salary is 8000 and max\_salary is NULL will be entered automatically at the time of insertion if no value assigned for the specified columns.

create table jobs (job\_id integer primary key , job\_title varchar(20) default '' , min\_salary integer default 8000 , max\_salary integer default null);

cdac1=# \d jobs

                                Table "public.jobs"

   Column   |         Type          | Collation | Nullable |        Default

------------+-----------------------+-----------+----------+-----------------------

 job\_id     | integer               |           | not null |

 job\_title  | character varying(20) |           |          | ''::character varying

 min\_salary | integer               |           |          | 8000

 max\_salary | integer               |           |          |

Indexes:

    "jobs\_pkey" PRIMARY KEY, btree (job\_id)

1. Write a SQL statement to create and insert a record into the table countries to ensure that, at country\_id and the region\_id combination will be entered once in the table.

cdac1=# create table countries2(country\_id integer, country\_name char(20), region\_id int , primary key (country\_id,region\_id));

CREATE TABLE

cdac1=# \d countries2

                   Table "public.countries2"

    Column    |     Type      | Collation | Nullable | Default

--------------+---------------+-----------+----------+---------

 country\_id   | integer       |           | not null |

 country\_name | character(20) |           |          |

 region\_id    | integer       |           | not null |

Indexes:

    "countries2\_pkey" PRIMARY KEY, btree (country\_id, region\_id)

1. Write a SQL statement to create and insert records into the table countries to ensure that the country\_id column will not contain any duplicate data and this will be automatically incremented and the column country\_name will be filled up by 'N/A' if no value assigned to that column.

cdac1=# create table countries3(country\_id serial primary key , country\_name char(20) default 'N/A', region\_id int );

CREATE TABLE

cdac1=# \d countries3

                                      Table "public.countries3"

    Column    |     Type      | Collation | Nullable |                    Default

--------------+---------------+-----------+----------+------------------------------------------------

 country\_id   | integer       |           | not null | nextval('countries3\_country\_id\_seq'::regclass)

 country\_name | character(20) |           |          | 'N/A'::bpchar

 region\_id    | integer       |           |          |

Indexes:

     "countries3\_pkey" PRIMARY KEY, btree (country\_id)

**Assignment 3**

1. Write a SQL statement to rename the table countries to country\_new.

alter table countries rename to country\_new ;

1. Write a SQL statement to add a column region\_id to the table locations.

alter table countries2 add column country\_code int ;

1. Write a SQL statement to change the data type of the column region\_id to text in the table locations.

alter table countries2 alter column country\_code type text ;

1. Write a SQL statement to drop the column city from the table locations.

alter table countries2 drop column country\_code ;

1. Write a SQL statement to add a primary key for the columns location\_id in the locations table.

alter table dup\_countries add primary key (country\_id);

**Assignment 4**

1. Write a SQL statement to change the salary of an employee to 8000 whose ID is 105, if the existing salary is less than 5000

update employees set salary = 8000 where employee\_id = 105 and salary < 5000 ;

1. Write a SQL statement to change the job ID of the employee which ID is 118 to SH\_CLERK if the employee belongs to a department which ID is 30 and the existing job ID does not start with SH.

update employees set job\_id = 'SH\_CLERK' where employee\_id = 118 and department\_id = 30 and not job\_id like 'SH%';

1. Write a SQL statement to increase the salary of employees under the department 40, 90 and 110 according to the company rules that, the salary will be increased by 25% of the department 40, 15% for department 90 and 10% of the department 110 and the rest of the department will remain same.

 update employees set salary = case department\_id when 40 then salary\*1.25 when 90 then salary\*1.15 when 110 then salary\*1.1 else salary end;

1. Write a SQL statement to change the email column of the employees table with 'not available' for those employees who belongs to the 'Accounting' department.

update employees set email = 'not available ' where job\_id = 'FI\_ACCOUNT';

1. Write a SQL statement to change the email column of employees table with 'not available' for those employees whose department\_id is 80 and gets a commission is less than.20%.

 update employees set email ='not available' where department\_id=80 and commission\_pct=0.2 ;

1. Write a SQL statement to change the email and commission\_pct column of the employees table with 'not available' and 0.10 for all employees.

cdac1=# update employees set email ='not available' ;

UPDATE 107

cdac1=# update employees set commission\_pct=0.10 ;

UPDATE 107

**Assignment 5.1**

SQL Exercise 1

 1. Create the table SEMP with the following structure:- EMPNO CHAR(4) EMPNAME CHAR(20) BASIC FLOAT DEPTNO CHAR(2) DEPTHEAD CHAR(4)

create table semp (EMPNO CHAR(4) , EMPNAME CHAR(20) , BASIC FLOAT , DEPTNO CHAR(2) , DEPTHEAD CHAR(4));

2. Create the table SDEPT with the following structure:- DEPTNO CHAR(2) DEPTNAME CHAR(15)

 create table sdept(DEPTNO CHAR(2) , DEPTNAME CHAR(15));

 3. Insert into the SDEPT table the following values:- 10, Development 20, Training

insert into sdept values (10, 'Development' ),(20, 'Training');

 4. Insert into the SEMP table the following values:- 0001, SUNIL, 6000, 10 0002, HIREN, 8000, 20 0003, ALI, 4000, 10, 0001 0004, GEORGE, 6000, 0002 Sameer Dehadrai Create S, P, J, SPJ tables as specified below and insert a few rows in each table:- SUPPLIER (S#, Sname, Status, City) - S PARTS (P#, Pname, Color, Weight, City) - P PROJECTS (J#, Jname, City) - J SUPPLIER-PARTS-PROJECT (S#, P#, J#, Qty) - SPJ Sample data for S# column:- ‘S1’, ‘S2’, ‘S3’, etc. Sample data for P# column:- ‘P1’, ‘P2’, ‘P3’, etc. Sample data for J# column:- ‘J1’, ‘J2’, ‘J3’, etc. Sample data for Status column:- 10, 20, 30, etc. Write the SELECT queries to do the following:-

**Joins assignment**

**1.** From the following tables, write a SQL query to find the first name, last name, department number, and department name for each employee.

Answer :

select e.first\_name , e.last\_name, d.department\_id , d.department\_name from employees e join departments d on e.department\_id = d.department\_id;

**2.** From the following tables, write a SQL query to find the first name, last name, department, city, and state province for each employee.

Answer:

select e.first\_name , e.last\_name , d.department\_name , l.city , l.state\_province from employees e join departments d on e.department\_id = d.department\_id join locations l on d.location\_id = l.location\_id;

**3.** From the following table, write a SQL query to find the first name, last name, salary, and job grade for all employees.

Answer:

 select e.first\_name , e.last\_name , j.salary , j.grade\_level from employees e join job\_grade j on e.salary = j.salary ;

**4.** From the following tables, write a SQL query to find all those employees who work in department ID 80 or 40. Return first name, last name, department number and department name.

>>  select e.first\_name,e.last\_name,d.department\_id,d.department\_name from employees e join departments d on e.department\_id=d.department\_id where d.department\_id=80 or d.department\_id=40;

**5.** From the following tables, write a SQL query to find those employees whose first name contains a letter ‘z’. Return first name, last name, department, city, and state province.

>>select e.first\_name,e.last\_name,d.department\_name,l.city,l.state\_province from employees e join departments d on e.department\_id=d.department\_id join locations l on d.location\_id=l.location\_id where first\_name like '%z%';

**6.** From the following table, write a SQL query to find all departments including those without any employee. Return first name, last name, department ID, department name.

>>select e.first\_name,e.last\_name,d.department\_id,d.department\_name from employees e right join departments d on d.department\_id=e.department\_id;

**7.** From the following table, write a SQL query to find those employees who earn less than the employee of ID 182. Return first name, last name and salary.

>>select first\_name , last\_name,salary from employees where salary < (select salary from employees where employee\_id=182);

**8.** From the following table, write a SQL query to find the employees and their managers. Return the first name of the employee and manager.

>> select e.first\_name as employee\_name , w.first\_name as manager from employees e join employees w on e.manager\_id=w.employee\_id;

**9.** From the following tables, write a SQL query to display the department name, city, and state province for each department.

>>select d.department\_name,l.city,l.state\_province from departments d join locations l on d.location\_id=l.location\_id;

**10.** From the following tables, write a SQL query to find those employees who have or not any department. Return first name, last name, department ID, department name.

>>select e.first\_name,e.last\_name,d.department\_id,d.department\_name from employees e left join departments d on e.department\_id=d.department\_id;

**11.** From the following table, write a SQL query to find the employees and their managers. These managers do not work under any manager. Return the first name of the employee and manager.

>>select e.first\_name as employee\_name , w.first\_name as manager from employees e left join employees w on e.manager\_id=w.employee\_id;

**12.** From the following tables, write a SQL query to find those employees who work in a department where the employee of last name 'Taylor' works. Return first name, last name and department ID.

>>select first\_name,last\_name,department\_id from employees where department\_id in (select department\_id from employees where last\_name='Taylor');

**13.** From the following tables, write a SQL query to find those employees who joined between 1st January 1993 and 31 August 1997. Return job title, department name, employee name, and joining date of the job.

>> select j.job\_title,d.department\_name,e.first\_name,e.last\_name,h.start\_date from employees e join job\_history h on h.employee\_id=e.employee\_id join departments d on h.department\_id=d.department\_id join jobs j on j.job\_id = h.job\_id where start\_date between '1993-01-01' and '1997-08-31';

**14.** From the following tables, write a SQL query to find the difference between maximum salary of the job and salary of the employees. Return job title, employee name, and salary difference.

>>select j.job\_title,e.first\_name||' '||e.last\_name as employee\_name, j.max\_salary-e.salary as salary\_difference from employees e join jobs j on e.job\_id=j.job\_id;

**15.** From the following table, write a SQL query to compute the average salary, number of employees received commission in that department. Return department name, average salary and number of employees.

>>select d.department\_name ,count(d.department\_name),avg(e.salary) from employees e join departments d using(department\_id) group by d.department\_name;

**16.** From the following tables, write a SQL query to compute the difference between maximum salary and salary of all the employees who works the department of ID 80. Return job title, employee name and salary difference.

>>select j.job\_title,e.first\_name||' '||e.last\_name as employee\_name, j.max\_salary-e.salary as salary\_difference from employees e join jobs j on e.job\_id=j.job\_id where e.department\_id=80;

**17.** From the following table, write a SQL query to find the name of the country, city, and departments, which are running there.

>>select c.country\_name as country,l.city,d.department\_name from countries c join locations l using(country\_id) join departments d using(location\_id);

**18.** From the following tables, write a SQL query to find the department name and the full name (first and last name) of the manager

>>