1) Installing and deploying PostgreSQL database

i. Setting environmental variable.

PostgreSQL was installed and the path to bin file was given in system environmental variable.

ii. Status of database server

Syntax: pg_ctl status

```
C:\Users\Gautam>pg_ctl status
pg_ctl: no server running
```

iii. Starting database server.

Syntax: pg_ctl start

```
C:\Users\Gautam>pg_ctl start
waiting for server to start....2023-05-03 07:01:42.957 +0545 [8984] LOG:  listening on IPv6 address "::", port 5432
2023-05-03 07:01:42.957 +0545 [8984] LOG:  listening on IPv4 address "0.0.0.0", port 5432
2023-05-03 07:01:42.976 +0545 [8984] LOG:  redirecting log output to logging collector process
2023-05-03 07:01:42.976 +0545 [8984] HINT:  Future log output will appear in directory "log".
 done
server started
```

iv. Stoping the database server

Syntax: pg_ctl stop

```
C:\Users\Gautam>pg_ctl stop
waiting for server to shut down.... done
server stopped

C:\Users\Gautam>
```

v. Creating a database server

Syntax: Create database <database-name>

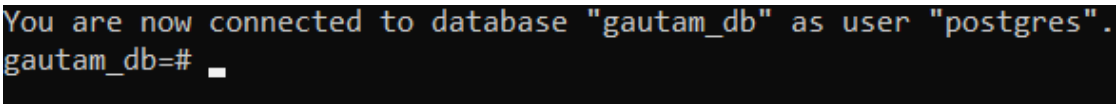
```
C:\Windows\system32\cmd.exe - psql  -U postgres
postgres=# create database gautam_db;
CREATE DATABASE
postgres=# \l
                                                 List of databases
   Name    |  Owner   | Encoding |          Collate           |           Ctype            |   Access privileges
-----------+----------+----------+----------------------------+----------------------------+-----------------------
 gautam_db | postgres | UTF8     | English_United States.1252 | English_United States.1252 |
 postgres  | postgres | UTF8     | English_United States.1252 | English_United States.1252 |
 template0 | postgres | UTF8     | English_United States.1252 | English_United States.1252 | =c/postgres          +
           |          |          |                            |                            | postgres=CTc/postgres
 template1 | postgres | UTF8     | English_United States.1252 | English_United States.1252 | =c/postgres          +
           |          |          |                            |                            | postgres=CTc/postgres
(4 rows)
```

vi. Dropping a database

Syntax: drop database <database-name>

```
postgres=# drop database gautam_db;
DROP DATABASE
postgres=# \l
                                                 List of databases
   Name    |  Owner   | Encoding |          Collate           |           Ctype            |   Access privileges
-----------+----------+----------+----------------------------+----------------------------+-----------------------
 postgres  | postgres | UTF8     | English_United States.1252 | English_United States.1252 |
 template0 | postgres | UTF8     | English_United States.1252 | English_United States.1252 | =c/postgres          +
           |          |          |                            |                            | postgres=CTc/postgres
 template1 | postgres | UTF8     | English_United States.1252 | English_United States.1252 | =c/postgres          +
           |          |          |                            |                            | postgres=CTc/postgres
(3 rows)
```

vii. Connecting to database.
Syntax: \c database_name



2) Creating a table

Syntax: create table<table_name>(

attribute datatype constraints,

$attribute_2$ datatype constraints,

$attribute_n$ datatype constraints

);

```
gautam_db=# create table accounts(
gautam_db(# user_id serial primary key,
gautam_db(# username varchar(50) unique not null,
gautam_db(# password varchar(50) not null,
gautam_db(# email varchar(50) unique not null,
gautam_db(# created_on timestamp not null,
gautam_db(# last_login timestamp
gautam_db(# );
CREATE TABLE
gautam_db=# create table roles(
gautam_db(# role_id serial primary key,
gautam_db(# role_name varchar(50) unique not null
gautam_db(# );
CREATE TABLE
gautam_db=# create table account_roles(
gautam_db(# userid int not null,
gautam_db(# roleid int not null,
gautam_db(# grant_date timestamp,
gautam_db(# primary key (userid, roleid),
gautam_db(# foreign key (roleid) references roles(role_id),
gautam_db(# foreign key (userid) references accounts(user_id)
gautam_db(# );
CREATE TABLE
```

```
gautam_db=# \d
                List of relations
 Schema |         Name          |   Type   |  Owner
--------+-----------------------+----------+----------
 public | account_roles         | table    | postgres
 public | accounts              | table    | postgres
 public | accounts_user_id_seq  | sequence | postgres
 public | roles                 | table    | postgres
 public | roles_role_id_seq     | sequence | postgres
 public | tbl_student           | table    | postgres
(6 rows)
```

3) Altering table

i. Adding column in table

Syntax: alter table <table_name> add column <attribute datatype constraint>

```
gautam_db=# alter table roles add column role_detail varchar(50);
ALTER TABLE
gautam_db=# \d roles
                                  Table "public.roles"
   Column    |         Type          | Collation | Nullable |               Default
-------------+-----------------------+-----------+----------+-------------------------------------
 role_id     | integer               |           | not null | nextval('roles_role_id_seq'::regclass)
 role_name   | character varying(50) |           | not null |
 role_detail | character varying(50) |           |          |
Indexes:
    "roles_pkey" PRIMARY KEY, btree (role_id)
    "roles_role_name_key" UNIQUE CONSTRAINT, btree (role_name)
Referenced by:
    TABLE "account_roles" CONSTRAINT "account_roles_roleid_fkey" FOREIGN KEY (roleid) REFERENCES roles(role_id)
```

ii. Adding column with default timestamp of recent of recent time

Syntax: alter table <table_name> add column <column_name> timestamp with time zone default now();

```
gautam_db=# alter table roles add column mtime timestamp with time zone default now();
ALTER TABLE
gautam_db=# \d roles
                                     Table "public.roles"
   Column    |           Type           | Collation | Nullable |               Default
-------------+--------------------------+-----------+----------+-------------------------------------
 role_id     | integer                  |           | not null | nextval('roles_role_id_seq'::regclass)
 role_name   | character varying(50)    |           | not null |
 role_detail | character varying(50)    |           |          |
 mtime       | timestamp with time zone |           |          | now()
Indexes:
    "roles_pkey" PRIMARY KEY, btree (role_id)
    "roles_role_name_key" UNIQUE CONSTRAINT, btree (role_name)
Referenced by:
    TABLE "account_roles" CONSTRAINT "account_roles_roleid_fkey" FOREIGN KEY (roleid) REFERENCES roles(role_id)


gautam_db=# insert into roles(role_name) values('Viewer');
INSERT 0 1
gautam_db=# select * from roles;
 role_id | role_name | role_detail |             mtime
---------+-----------+-------------+-------------------------------
       1 | Viewer    |             | 2023-05-11 22:35:20.700945+05:45
(1 row)
```

iii. Renaming table names.

Syntax: alter table <table_name> rename to <new_name>;

```
gautam_db=# alter table accounts rename to tbl_users;
ALTER TABLE
gautam_db=# \d
                List of relations
 Schema |          Name          |   Type   |  Owner
--------+------------------------+----------+----------
 public | account_roles          | table    | postgres
 public | accounts_user_id_seq   | sequence | postgres
 public | roles                  | table    | postgres
 public | roles_role_id_seq      | sequence | postgres
 public | tbl_student            | table    | postgres
 public | tbl_users              | table    | postgres
(6 rows)
```

iv. Adding check constraints in attribute

Syntax: alter table<table_name> add constraint<any_name> check(<attribute> in (‘<options>’);

```
gautam_db=# alter table tbl_users add column gender varchar(1);
ALTER TABLE
gautam_db=# alter table tbl_users add constraint gen_check check(gender in ('M','F','o'));
ALTER TABLE
gautam_db=# \d tbl_users
                                  Table "public.tbl_users"
   Column   |           Type           | Collation | Nullable |                 Default
------------+--------------------------+-----------+----------+----------------------------------------
 user_id    | integer                  |           | not null | nextval('accounts_user_id_seq'::regclass)
 username   | character varying(50)    |           | not null |
 password   | character varying(50)    |           | not null |
 email      | character varying(50)    |           | not null |
 created_on | timestamp without time zone |        | not null |
 last_login | timestamp without time zone |        |          |
 gender     | character varying(1)     |           |          |
Indexes:
    "accounts_pkey" PRIMARY KEY, btree (user_id)
    "accounts_email_key" UNIQUE CONSTRAINT, btree (email)
    "accounts_username_key" UNIQUE CONSTRAINT, btree (username)
Check constraints:
    "gen_check" CHECK (gender::text = ANY (ARRAY['M'::character varying, 'F'::character varying, 'o'::character varying]::text[]))
Referenced by:
    TABLE "account_roles" CONSTRAINT "account_roles_userid_fkey" FOREIGN KEY (userid) REFERENCES tbl_users(user_id)
```

v. Renaming column

Syntax: alter table <table_name> rename column<column_name> to <new_column_name>;

```
gautam_db=# alter table tbl_student rename column phone to contact_no;
ALTER TABLE
gautam_db=# \d tbl_student
                    Table "public.tbl_student"
   Column   |          Type           | Collation | Nullable | Default
------------+-------------------------+-----------+----------+---------
 sid        | integer                 |           | not null |
 fname      | character varying(50)   |           | not null |
 mname      | character varying(50)   |           |          |
 lname      | character varying(50)   |           | not null |
 gender     | character(1)            |           |          |
 contact_no | character varying(100)  |           |          |
 email_id   | character varying(100)  |           |          |
Indexes:
    "tbl_student_pkey" PRIMARY KEY, btree (sid)
    "tbl_student_fname_lname_key" UNIQUE CONSTRAINT, btree (fname, lname)
```

vi. Dropping column

Syntax: alter table <table_name> drop column<column_name>;

```
gautam_db=# alter table tbl_student drop column mname;
ALTER TABLE
gautam_db=# \d tbl_student
                    Table "public.tbl_student"
   Column   |          Type           | Collation | Nullable | Default
------------+-------------------------+-----------+----------+---------
 sid        | integer                 |           | not null |
 fname      | character varying(50)   |           | not null |
 lname      | character varying(50)   |           | not null |
 gender     | character(1)            |           |          |
 contact_no | character varying(100)  |           |          |
 email_id   | character varying(100)  |           |          |
Indexes:
    "tbl_student_pkey" PRIMARY KEY, btree (sid)
    "tbl_student_fname_lname_key" UNIQUE CONSTRAINT, btree (fname, lname)
```

4) Insert Query

i. Single insert query

Syntax: insert into <tbl_name>(<attribute>) values (<corresponding values for attribute>);

```
gautam_db=# insert into tbl_users(username, password, email, created_on, last_login, gender)
gautam_db-# values('gautam', '123321', 'gautam@gmail.com', current_date,current_date, 'M');
INSERT 0 1
gautam_db=# select * from tbl_users;
 user_id | username | password |      email       |     created_on      |     last_login      | gender
---------+----------+----------+------------------+---------------------+---------------------+--------
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
(1 row)
```

ii. Multi-line insert query

Syntax: insert into<tbl_name> (<attribute>)

values(<corresponding values for attribute >),

(<next corresponding values for attribute >);

```
gautam_db=# insert into tbl_users(username, password, email, created_on, last_login, gender)
gautam_db-# values('Rajak', '123321', 'rajak@gmail.com',current_date,current_date,'M'),
gautam_db-# ('Sumon', '123321', 'sumon@gmail.com', current_date, current_date, 'M'),
gautam_db-# ('Sita', '123321', 'sita@gmail.com',current_date,current_date,'F');
INSERT 0 3
gautam_db=# select * From tbl_users;
 user_id | username | password |      email       |     created_on      |     last_login      | gender
---------+----------+----------+------------------+---------------------+---------------------+--------
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       2 | Rajak    | 123321   | rajak@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       3 | Sumon    | 123321   | sumon@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       4 | Sita     | 123321   | sita@gmail.com   | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F
(4 rows)
```

5) Select Query

i. Simple select

Syntax: select * from<tbl_name>;

```
gautam_db=# select * from tbl_users;
 user_id | username | password |      email       |     created_on      |     last_login      | gender
---------+----------+----------+------------------+---------------------+---------------------+--------
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       2 | Rajak    | 123321   | rajak@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       3 | Sumon    | 123321   | sumon@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       4 | Sita     | 123321   | sita@gmail.com   | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F
(4 rows)
```

ii. Select spefic column.

Syntax: select<colum_name> from <tbl_name>;

```
gautam_db=# SELECT username FROM tbl_users;
 username
----------
 gautam
 Rajak
 Sumon
 Sita
(4 rows)
```

iii. Where clause.

Syntac: select * from <tbl_name> WHERE<condition>

```
gautam_db=# select * from tbl_users where user_id='4';
 user_id | username | password |      email       |     created_on      |     last_login      | gender
---------+----------+----------+------------------+---------------------+---------------------+--------
       4 | Sita     | 123321   | sita@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F
(1 row)
```

iv. Order by clause

Ascending Order:

Syntax: select * from <tbl_name> order by <column_name>;

```
gautam_db=# select * from tbl_users order by username;
 user_id | username | password |      email       |     created_on      |     last_login      | gender
---------+----------+----------+------------------+---------------------+---------------------+--------
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       2 | Rajak    | 123321   | rajak@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       4 | Sita     | 123321   | sita@gmail.com   | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F
       3 | Sumon    | 123321   | sumon@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
(4 rows)
```

Descending Order:
Syntax: select * from<tbl_name> order by<column_name> desc;

```
gautam_db=# select * from tbl_users order by username desc;
 user_id | username | password |      email       |     created_on      |     last_login      | gender
---------+----------+----------+------------------+---------------------+---------------------+--------
       3 | Sumon    | 123321   | sumon@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       4 | Sita     | 123321   | sita@gmail.com   | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F
       2 | Rajak    | 123321   | rajak@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M
(4 rows)
```

v. Upper and lower clause
For Upper:
Syntax: select upper(column_name) from <tbl_name>;
For Lower:
Syntax: select lower(column_name) from<tbl_name>;

```
gautam_db=# select upper(username), lower(full_name) from tbl_users;
 upper  |     lower
--------+---------------
 GAUTAM | gautam bibek
 RAJAK  | rajak putuwar
 SUMON  | sumon khadka
 SITA   | sita abc
(4 rows)
```

vi. Distinct clause
Syntax: select distinct <column_name> from <tbl_name>;

```
gautam_db=# select distinct full_name from tbl_Users;
   full_name
---------------
 Sita abc
 Sumon Khadka
 Rajak Putuwar
 Gautam Bibek
(4 rows)
```

vii. In clause
Syntax: select * from <tbl_name> where <column_name> in (<value>);

```
gautam_db=# select * from tbl_users where user_id in (1,3,4);
 user_id | username | password |      email       |     created_on      |     last_login      | gender |  full_name
---------+----------+----------+------------------+---------------------+---------------------+--------+--------------
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M      | Gautam Bibek
       3 | Sumon    | 123321   | sumon@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M      | Sumon Khadka
       4 | Sita     | 123321   | sita@gmail.com   | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F      | Sita abc
(3 rows)
```

viii. Like clause.

Syntax: select * from<tbl_name> where <column_name> like<%XXX%>;

```
gautam_db=# select * from tbl_users where username like '%m%';
 user_id | username | password |      email       |     created_on      |     last_login      | gender |  full_name
---------+----------+----------+------------------+---------------------+---------------------+--------+-------------
       1 | gautam   | 123321   | gautam@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M      | Gautam Bibek
       3 | Sumon    | 123321   | sumon@gmail.com  | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | M      | Sumon Khadka
(2 rows)
```

ix. Not equal to

Syntax: select * from <tbl_name> where <column_name>!=<value>;

```
gautam_db=# select * from tbl_users where gender!='M';
 user_id | username | password |     email      |     created_on      |     last_login      | gender | full_name
---------+----------+----------+----------------+---------------------+---------------------+--------+-----------
       4 | Sita     | 123321   | sita@gmail.com | 2023-05-11 00:00:00 | 2023-05-11 00:00:00 | F      | Sita abc
(1 row)
```

x. Is null

Syntax: select * from <tbl_name> where<column_name> is null;

```
gautam_db=# select * from roles where role_detail is null;
 role_id | role_name | role_detail |              mtime
---------+-----------+-------------+-----------------------------------
       1 | Viewer    |             | 2023-05-11 22:35:20.700945+05:45
(1 row)
```

xi. Aliasing

Syntax: select<column_name><value> from<tbl_name>;

```
gautam_db=# select full_name "FULLNAME" from tbl_users;
   FULLNAME
---------------
 Gautam Bibek
 Rajak Putuwar
 Sumon Khadka
 Sita abc
(4 rows)
```

Syntax: select <column_name> as <value> from <tbl_name>;

```
gautam_db=# select full_name as "FULL NAME" from tbl_users;
   FULL NAME
---------------
 Gautam Bibek
 Rajak Putuwar
 Sumon Khadka
 Sita abc
(4 rows)
```

xii. Count clause

Syntax: select count(<column)name>) from <tbl_name> ;

```
gautam_db=# select count(username) from tbl_users;
 count
-------
     4
(1 row)
```

xiii. Group by clause

Syntax: select<column_name> from <tbl_name> group by<column_name>;

```
gautam_db=# select fruits from basket group by fruits;
  fruits
-----------
 pineapple
 mango
 banana
 apple
(4 rows)
```

```
gautam_db=# select fruits, count(*) from basket group by fruits order by 2 desc;
  fruits   | count
-----------+-------
 banana    |     3
 apple     |     3
 mango     |     2
 pineapple |     1
(4 rows)
```

6) Delete Query:

Syntax: delete from <table_name> where <condition>;

```
gautam_db=# delete from basket where id = 2;
DELETE 1
gautam_db=# select * from basket;
 id | fruits
----+--------
  1 | banana
  3 | mango
  4 | orange
(3 rows)
```

7) Update Query
   Syntax: update <table> set attribute = ‘value’, set attribute = ‘value2’ where <condition>