SELECT version();

SELECT date\_trunc('second', current\_timestamp -

pg\_postmaster\_start\_time()) as uptime,pg\_postmaster\_start\_time(), current\_timestamp;

show data\_directory;

select \* from pg\_database;

LKL

SELECT \* FROM information\_schema.Tables ORDER BY 2 DESC;

SELECT SELECT pg\_size\_pretty(pg\_database\_size(current\_database()));

SELECT datname,pg\_size\_pretty((pg\_database\_size(datname))),pg\_database\_size(datname) from pg\_database ;

select table\_schema,table\_name FROM information\_schema.Tables

select table\_schema,table\_name,pg\_relation\_size(table\_schema||'.'||table\_name) FROM information\_schema.Tables ORDER BY 3 DESC;

SELECT pg\_size\_pretty('jass.sales')

SELECT \* FROM pg\_extension;

select \* from pg\_constraint;

select \* from information\_schema.tables;

SELECT \* FROM pg\_constraint

where conname like '%data%';

--WHERE confrelid = 'jass.edata'::regclass;"16453"

select \* from pg\_class;

select conname constraint\_name,relname table\_name from pg\_constraint pc join pg\_class pcl

on pc.conrelid=pcl.oid and relname='edata';

Select \* from pg\_settings

"C:/Program Files/PostgreSQL/16/data/postgresql.conf"

Select \* from jass.sales;

select \* from jass.product;

create table jass.test2(sr\_no int, name varchar(10));

insert into jass.test2(sr\_no,name) values (1,'hi');

select \* from jass.sales;

insert into jass.sales values(2,2009,500,400);

update jass.sales set pid=3 where yom=2009;

on Local host m :

Create USER my\_user With Password 'Goodjob1';

Grant connect on database "Baaz" to my\_user;

Select \* from pg\_database;

Grant Select on jass.sales to my\_user;

Grant All privileges on database "Baaz" to my\_user;

Grant usage on schema jass to my\_user;

Grant Select on jass.product to my\_user;

Revoke all privileges on database "Baaz" from my\_user;

Grant create on schema jass to my\_user;

Grant insert on jass.sales to my\_user;

Grant all privileges on all tables in schema jass to my\_user;

Create USER my\_user2 With Password 'Goodjob1';

create role myrole;

Grant usage on schema jass to myrole;

Grant select on jass.sales to myrole;

Grant myrole to my\_user2;

Select \* from jass.sales;

show data\_directory;

show archive\_mode; //to store files it should be on

show archive\_command;// how the files are stored

show log\_directory;//

select pg\_switch\_wal();//

Creating sample databases:

CREATE TABLE customers (

customer\_id SERIAL PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE products (

product\_id SERIAL PRIMARY KEY,

product\_name VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL,

stock INT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE orders (

order\_id SERIAL PRIMARY KEY,

customer\_id INT REFERENCES customers(customer\_id) ON DELETE CASCADE,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE order\_items (

order\_item\_id SERIAL PRIMARY KEY,

order\_id INT REFERENCES orders(order\_id) ON DELETE CASCADE,

product\_id INT REFERENCES products(product\_id),

quantity INT NOT NULL,

price DECIMAL(10, 2) NOT NULL

);

INSERT INTO customers (first\_name, last\_name, email)

VALUES

('John', 'Doe', 'john.doe@example.com'),

('Jane', 'Smith', 'jane.smith@example.com');

INSERT INTO products (product\_name, price, stock)

VALUES

('Laptop', 1200.00, 10),

('Smartphone', 800.00, 25),

('Headphones', 150.00, 100);

INSERT INTO orders (customer\_id)

VALUES

(1), (2); -- Assuming customer\_id 1 and 2 exist

INSERT INTO order\_items (order\_id, product\_id, quantity, price)

VALUES

(1, 1, 1, 1200.00), -- Order 1: 1 Laptop

(1, 3, 2, 150.00), -- Order 1: 2 Headphones

(2, 2, 1, 800.00); -- Order 2: 1 Smartphone

Joins:

select customers.first\_name, customers.customer\_id, orders.order\_id from orders full outer join customers

ON orders.customer\_id=customers.customer\_id; // just replace with join or right join or left join or cartesian product

Select \* from orders;

select \* from customers;

insert into customers values(3,'Raj','Singh','abc@gmail.com');

home work:

SELECT

o.order\_id,

i.order\_item\_id,

o.order\_date,

o.customer\_id,

cs.first\_name||' '||cs.last\_name as name,

cs.email,

i.product\_id,

pr.product\_name,

i.quantity,

i.price,

pr.product\_name

FROM

orders o right JOIN customers cs

ON o.customer\_id=cs.customer\_id

left JOIN order\_items i

ON o.order\_id=i.order\_id

left JOIN products pr

ON i.product\_id=pr.product\_id;

How to check archive files and privileges on tables :

select \* from information\_schema.role\_table\_grants

where grantee not in ('postgres','public','PUBLIC');

select pg\_current\_wal\_lsn();

select \* from pg\_ls\_dir('pg\_wal/archive\_status');

select \* from pg\_stat\_archiver;

show data\_directory;

vacuum and analyze data:

create table sample\_table (id serial primary key, data text);

insert into sample\_table (data) values ('row3'),('row4'),('row5'),('row6'),('row7');

select \* from sample\_table;

delete from sample\_table where id in (8,9,10,11);

select n\_dead\_tup from pg\_stat\_all\_tables where relname='sample\_table';

vacuum sample\_table;

vacuum full sample\_table;

vacuum;

analyze sample\_table;

analyze;

vacuum analyze sample\_table;

vacuum analyze;

select \* from information\_schema.role\_table\_grants where grantee not in ('postgres','public','PUBLIC');

archive files:

show data\_directory;

select \* from pg\_ls\_dir('pg\_wal/archive\_status');

select pg\_switch\_wal();

show archive\_command;

November 7th 2024

Functions and procedures:

CREATE OR REPLACE FUNCTION jass.add\_number(a integer, b integer) RETURNS integer AS $$

BEGIN

RETURN a + b;

END;

$$ LANGUAGE plpgsql;

select add\_number(5,4);

create table jass.employee

( salary integer , name text);

create procedure jass.insert\_employee(ename text, esalary integer)

language plpgsql

as $$

begin

insert into jass.employee(name,salary) values (ename,esalary);

end;

$$;

call jass.insert\_employee('abc',500);

select \* from jass.employee;

factorial of a function

create function jass.fact(a integer) returns integer AS $$

declare f integer:=1;

begin

loop

exit when a=1;

f:=f\*a;

a:=a-1;

end loop;

return f;

end;$$ language plpgsql;

select jass.fact(5);

**Index:**

create index my\_index on jass.edata(eid);

create index test\_index on jass.employee using hash(salary);

create index test\_index on jass.sales using gin(pid);

create table products(

id serial primary key,

tags text[]);

create index test on products using gin(tags);

create table my\_table1( id integer, name text);

insert into my\_table values( 1, 'abcd');

insert into my\_table values( 1, 'xyz');

insert into my\_table select \* from my\_table;

select count(\*) from my\_table;

select row\_number() over(order by id) from my\_table;

insert into my\_table1 select row\_number() over(order by id),

'abcd'||cast(row\_number() over(order by id)as text) from my\_table;

select \* from my\_table1 where id between 200 and 300;

create index my\_index on my\_table1(id);

create table my\_table2(id integer, name text);

insert into my\_table2 select \* from my\_table1;

select \* from my\_table2 where id between 100 and 200;

select max(id) from my\_table1;

**script writing from cmd:**

Microsoft Windows [Version 10.0.22631.4460]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd

C:\Windows\System32

C:\Windows\System32>

C:\Windows\System32>cd..

C:\Windows>cd..

C:\>cd C:\Program Files\PostgreSQL\16\bin

C:\Program Files\PostgreSQL\16\bin>psql

psql: error: connection to server at "localhost" (::1), port 5432 failed: Connection refused (0x0000274D/10061)

Is the server running on that host and accepting TCP/IP connections?

connection to server at "localhost" (127.0.0.1), port 5432 failed: Connection refused (0x0000274D/10061)

Is the server running on that host and accepting TCP/IP connections?

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -D Baaz

psql: illegal option -- D

psql: hint: Try "psql --help" for more information.

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -d Baaz

Password for user postgres:

psql (16.4)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

Baaz=# \i fi\c.sql;

fi: Permission denied

Baaz=# \i C:\Back\c.sql;

C:: Permission denied

Baaz=# \i fi\c.sql;

fi: Permission denied

Baaz=# \i C:\Back\c.sql;

C:: Permission denied

Baaz=# exit

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -d Baaz

Password for user postgres:

psql (16.4)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

Baaz=# \i C:\Back\c.sql;

C:: Permission denied

Baaz=# exit

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -d Baaz -f c:\Back\c.sql

Password for user postgres:

BEGIN

psql:c:/Back/c.sql:2: ERROR: relation "c" already exists

psql:c:/Back/c.sql:3: ERROR: current transaction is aborted, commands ignored until end of transaction block

ROLLBACK

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -d Baaz -f c:\Back\c.sql

**from writing:**

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -d baaz

Password for user postgres:

psql: error: connection to server at "localhost" (::1), port 5433 failed: FATAL: database "baaz" does not exist

C:\Program Files\PostgreSQL\16\bin>psql -U postgres -p 5433 -d Baaz

Password for user postgres:

psql (16.4)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

Baaz=# begin;

BEGIN

Baaz=\*# create table a(x int);

CREATE TABLE

Baaz=\*# begin

Baaz-\*# ;

WARNING: there is already a transaction in progress

BEGIN

Baaz=\*# rollback;

ROLLBACK

Baaz=# select \* from a

Baaz-# ;

ERROR: relation "a" does not exist

LINE 1: select \* from a

^

Baaz=# begin;

BEGIN

Baaz=\*# create table a(x int);

CREATE TABLE

Baaz=\*# commit;

COMMIT

Baaz=# select \* from a;

x

---

(0 rows)

Baaz=# begin;

BEGIN

Baaz=\*# create table b(y int);

CREATE TABLE

Baaz=\*# insert into table b values(10);

ERROR: syntax error at or near "table"

LINE 1: insert into table b values(10);

^

Baaz=!# insert into b values(10);

ERROR: current transaction is aborted, commands ignored until end of transaction block

Baaz=!# select \* from b;

ERROR: current transaction is aborted, commands ignored until end of transaction block

Baaz=!# rollback;

ROLLBACK

Baaz=# select \* from b;

ERROR: relation "b" does not exist

LINE 1: select \* from b;

^

Baaz=# begin;

BEGIN

Baaz=\*# create table b(y int);

CREATE TABLE

Baaz=\*# insert into b values(10);

INSERT 0 1

Baaz=\*# savepoint t2;

SAVEPOINT

Baaz=\*# update b set y=11;

UPDATE 1

Baaz=\*# select \* from b;

y

----

11

(1 row)

Baaz=\*# rollback to savepoint t2;

ROLLBACK

Baaz=\*# select \* from b;

y

----

10

(1 row)

Baaz=\*# commit;

COMMIT

Baaz=# @"C:\Users\guide\Desktop\c.sql"

Baaz-# ;

ERROR: syntax error at or near "@"

LINE 1: @"C:\Users\guide\Desktop\c.sql"

^

Baaz=# "C:\Users\guide\Desktop\c.sql";

ERROR: syntax error at or near ""C:\Users\guide\Desktop\c.sql""

LINE 1: "C:\Users\guide\Desktop\c.sql";

^

Baaz=# C:\Users\guide\Desktop\c.sql;

invalid command \Users

Try \? for help.

Baaz-# \i "C:\Users\guide\Desktop\c.sql";

"C:/Users/guide/Desktop/c.sql: No such file or directory

Baaz-# \i c.sql

BEGIN

CREATE TABLE

INSERT 0 1

COMMIT

Baaz-# \i C:\Users\guide\Desktop\c.sql;

C:: Permission denied

Baaz-# \i C:\Back\c.sql;

C:: Permission denied

Baaz-# \i fi\c.sql;

fi: Permission denied

**Aggregate functions:**

select \* from regions join countries on regions.region\_id=countries.region\_id

order by region\_name;

select count(\*), r.region\_name from regions as r join countries as c

on r.region\_id=c.region\_id group by region\_name having count(\*)<6;

select region\_name, array\_agg(country\_name) from regions as r join countries as c

on r.region\_id=c.region\_id group by region\_name;

select d.department\_name, array\_agg(e.first\_name) from employees as e join departments as d

on e.department\_id=d.department\_id group by d.department\_id;

select max(hire\_date), min(hire\_date) from employees;

select \* from regions as r join countries as c

on r.region\_id=c.region\_id join locations as l on

c.country\_id=l.country\_id;

select \* from pg\_catalog.pg\_proc where proname like '%agg%';