**Database Objects**

**Install PSQL client on EC2 instance**

**sudo amazon-linux-extras install postgresql14**

**psql --version**

**Connect to Aurora PostgreSQL**

**psql --host=<aurora-endpoint> --port=5432 --username=postgres --password**

**Create orders, customers and sellers tables**

**create table customers (**

**customer\_id int,**

**address text,**

**zipcode text,**

**city text,**

**state text,**

**dob date,**

**country text,**

**email text,**

**phone text,**

**CONSTRAINT customers\_pk PRIMARY KEY(customer\_id)**

**);**

**create table sellers (**

**seller\_id int,**

**seller\_address text,**

**seller\_state text,**

**seller\_zipcode text,**

**seller\_country text,**

**seller\_rating int,**

**seller\_active\_date timestamp,**

**seller\_business\_type text,**

**seller\_company\_name text,**

**CONSTRAINT sellers\_pk PRIMARY KEY(seller\_id)**

**);**

**create table orders (**

**order\_id int,**

**customer\_id int,**

**seller\_id int,**

**order\_date date,**

**delivery\_date date,**

**price decimal(8,2),**

**quantity int,**

**payment\_method text,**

**discount decimal(4,2),**

**CONSTRAINT orders\_pk PRIMARY KEY(order\_id),**

**CONSTRAINT orders\_cust\_fk FOREIGN KEY(customer\_id) REFERENCES customers (customer\_id),**

**CONSTRAINT orders\_seller\_fk FOREIGN KEY(seller\_id) REFERENCES sellers (seller\_id)**

**);**

**LOAD records from small dataset in S3 to orders, sellers & customers tables**

**create extension aws\_s3 cascade;**

**SELECT aws\_s3.table\_import\_from\_s3('customers','', '(FORMAT CSV)', aws\_commons.create\_s3\_uri('<bucket>', 'customers\_sql\_dataset.csv', 'region-code'));**

**SELECT aws\_s3.table\_import\_from\_s3('sellers','', '(FORMAT CSV)', aws\_commons.create\_s3\_uri('<bucket>', 'seller\_sql\_dataset.csv', '<region-code>'));**

**Execute the following commands before loading orders table**

**insert into customers values(10009);**

**insert into sellers values (2009);**

**insert into customers values(10005);**

**insert into sellers values (2001);**

**insert into sellers values (2017);**

**insert into sellers values (2006);**

**insert into sellers values (2010);**

**insert into customers values(10007);**

**insert into sellers values (2007);**

**insert into sellers values (2020);**

**insert into sellers values (2011);**

**insert into sellers values (2015);**

**SELECT aws\_s3.table\_import\_from\_s3('orders','', '(FORMAT CSV)', aws\_commons.create\_s3\_uri('<bucket>', 'orders\_sql\_dataset.csv', '<region-code>'));**

**Add column, remove column, change data type**

**create table orders\_duplicate (like orders);**

**alter table orders\_duplicate add column product\_id int;**

**alter table orders\_duplicate drop column payment\_method;**

**alter table orders\_duplicate rename column price to total\_price;**

**alter table orders\_duplicate alter column total\_price set data type decimal(10,2);**

**Operate on Data**

**set search\_path='db\_obj';**

**insert into customers values (22000, 'California', '29001','cxtys','CA','1983-01-23','USA','some@gmail.com', '1-400-2345');**

**SELECT**

**set search\_path='db\_obj';**

**select \* from orders;**

**select order\_id, order\_date, delivery\_date, price from orders;**

**select \* from orders limit 10;**

**select \* from orders where payment\_method = 'Visa';**

**select \* from orders where customer\_id = 10000;**

**select \* from orders where order\_date = '2024-09-21';**

**select \* from orders where quantity > 6;**

**select \* from orders where quantity > 8;**

**select \* from orders where quantity < 3;**

**select \* from orders where quantity <= 3;**

**select \* from orders where price >= 1000.00;**

**select \* from orders where price <= 100.00;**

**select \* from orders where price <= 200.00;**

**select \* from orders where price <= 500.00;**

**select \* from orders where price <= 800.00;**

**select \* from orders where payment\_method like '%isa%';**

**select \* from orders where payment\_method like '%rican%';**

**select \* from orders where payment\_method like 'rican%';**

**select \* from orders where payment\_method like '%rican';**

**select \* from orders where payment\_method like '%press';**

**select \* from orders where payment\_method like '%press' order by order\_date;**

**select \* from orders where payment\_method like '%press' order by order\_date desc;**

**select \* from orders where payment\_method like '%press' order by order\_date desc limit 6;**

**select \* from orders where payment\_method like '%press' order by order\_date desc, price asc;**

**select payment\_method, sum(price) as expense from orders;**

**select payment\_method, sum(price) as expense from orders group by payment\_method;**

**select payment\_method, sum(price) as expense from orders group by payment\_method order by expense desc limit 3;**

**select payment\_method, sum(price) as expense from orders group by payment\_method order by expense desc;**

**SELECT Operators**

**set search\_path='db\_obj';**

**select count(\*) from orders;**

**select count(\*) from customers;**

**select count(\*) from sellers;**

**select count(\*) from orders where payment\_method like 'Amer%';**

**select payment\_method,count(\*) from orders group by payment\_method;**

**select \* from orders limit 5;**

**select \* from orders where price > 1000.00 and quantity > 6;**

**select \* from orders where price > 2000.00 and quantity > 6;**

**select \* from orders where price > 2000.00 or quantity > 6;**

**select \* from orders where price > 2000.00 or quantity > 6 or payment\_method = 'American Express';**

**select \* from orders where price > 2000.00 or (quantity > 6 and payment\_method = 'American Express');**

**select \* from orders where (price < 700.00 and quantity <=5) or (quantity > 6 and payment\_method = 'American Express');**

**select \* from orders where (price < 700.00 and quantity <=5) and (quantity > 6 and payment\_method = 'American Express');**

**select \* from orders where (price < 700.00 or quantity <=5) and (payment\_method = 'American Express');**

**SELECT Arithmetic Operators**

**select order\_id, order\_date, (price/quantity) as price\_per\_item from orders limit 10;**

**select order\_id, order\_date, (price - ((price\*discount)/100)) as original\_price from orders;**

**select abs(price) from orders limit 10;**

**select order\_id, order\_date, abs(price - ((price\*discount)/100)) as original\_price from orders;**

**select abs(-100.25) ;**

**select sqrt(100) ;**

**select sqrt(102) ;**

**select power(2,3) ;**

**select power(2,4) ;**

**select 2^3;**

**select 2^5;**

**SELECT – CASE, COALESCE**

**select order\_id, price, case payment\_method when 'American Express' then 'CHARGE CARD' when 'MasterCard' then 'DEBIT CARD' when 'Visa' then 'DEBIT CARD' else 'CREDT CARD' end as card\_type from orders limit 10;**

**select order\_id, price,**

**case payment\_method**

**when 'American Express' then 'CHARGE CARD'**

**when 'MasterCard' then 'DEBIT CARD'**

**when 'Visa' then 'DEBIT CARD'**

**else 'CREDT CARD'**

**end as credit\_card\_type**

**from orders;**

**alter table orders add column product\_code int;**

**select order\_id, order\_date, price, product\_code from orders limit 10;**

**select order\_id, order\_date, price, coalesce(product\_code,'no product') from orders limit 10;**

**update orders set product\_code = 12 where order\_id=1;**

**select order\_id, order\_date, price, coalesce(product\_code,0) from orders;**

**SELECT – DATE Functions**

**select current\_time;**

**select current\_date;**

**select current\_timestamp;**

**select order\_id, order\_date, delivery\_date, price, product\_code from orders limit 10;**

**select order\_id, price, order\_date, delivery\_date from orders where extract(year from order\_date) = 2024;**

**select order\_date,extract(dow from order\_date) from orders limit 10;**

**select order\_date,case extract(dow from order\_date) when 0 then 'SUNDAY' when 6 then 'SATURDAY' else 'WEEKDAY' end as day\_of\_delivery from orders limit 10;**

**select current\_date - interval '30 days';**

**select current\_date - interval '30 hours';**

**SELECT – Upper, Lower, Concat, Cast**

**alter table orders drop column product\_code;**

**select concat('My name', ' is DE');**

**select concat(price,discount) from orders limit 10;**

**select concat('total price: ',price,'discount: ' ,discount) from orders limit 10;**

**select concat('total price: ',price,' discount: ' ,discount) from orders limit 10;**

**select concat('total price: ',price,' discount: ' ,discount, ' discounted price', (price-(price\*discount/100))) from orders limit 10;**

**select concat('total price: ',price,' discount: ' ,discount, ' discounted price: ', (price-(price\*discount/100))) from orders limit 10;**

**select order\_id, concat('total price: ',price,' discount: ' ,discount, ' discounted price: ', (price-(price\*discount/100))) from orders limit 10;**

**select order\_id, concat('total price: ',price,' discount: ' ,discount, ' discounted price: ', (price-(price\*discount/100))) as invoice\_details from orders limit 10;**

**select order\_id, price-(price\*discount/100) as discounted\_price from orders limit 10;**

**select order\_id, cast(order\_date as timestamp) from orders limit 10;**

**select order\_id, cast(price as text) from orders limit 10;**

**select order\_id, concat('total price: ',price,' discount: ' ,discount, ' discounted price: ', (price-(price\*discount/100))) as invoice\_details from orders limit 10;**

**select order\_id, cast((price-(price\*discount/100)) as decimal(10,2)) as invoice\_details from orders limit 10;**

**select order\_id, concat('total price: ',price,' discount: ' ,discount, ' discounted price: ', cast((price-(price\*discount/100)) as decimal(10,2))) as invoice\_details from orders limit 10;**

**select \* from sellers limit 10;**

**select seller\_address, length(seller\_address) from sellers;**

**select seller\_address, lower(seller\_address) from sellers;**

**select seller\_address, upper(seller\_address) from sellers;**

**select seller\_address, substr(seller\_address,1,4) from sellers;**

**select seller\_address, substr(seller\_address,1,10) from sellers;**

**INSERT INTO … CTAS**

**create table order\_details (order\_id int, order\_date date, delivery\_date date, total\_price decimal(10,2), discounted\_price decimal(10,2));**

**insert into order\_details select order\_id, order\_date, delivery\_date, price, (price-(price\*discount/100)) from orders;**

**select \* from order\_details;**

**create table order\_details\_2 as select order\_id, order\_date, delivery\_date, price, (price-(price\*discount/100)) from orders;**

**select \* from order\_details\_2 limit 10;**

**\d+ order\_details\_2;**

**create table order\_details\_3 as select order\_id, order\_date, delivery\_date, price, cast((price-(price\*discount/100)) as decimal(10,2)) as disc\_price from orders;**

**select \* from order\_details\_3 limit 10;**

**Update, Delete, Truncate**

**update orders set product\_name = 'Default';**

**select \* from orders ;**

**update orders set product\_name = 'Electronics' where payment\_method = 'Visa';**

**select \* from orders ;**

**select \* from orders where payment\_method in ('Visa', 'Chase');**

**update orders set product\_name = 'Electronics', discount=10.0 where payment\_method = 'Visa';**

**select \* from orders where payment\_method in ('Visa', 'Chase');**

**delete from order\_details where extract(year from order\_date) = 2025;**

**select \* from order\_details;**

**delete from order\_details\_2;**

**truncate table order\_details\_3;**

**Complex SQLs**

**alter table orders drop column product\_name;**

**truncate table orders;**

**truncate table customers cascade;**

**truncate table sellers cascade;**

**Load orders, sellers, customers from small dataset**

**SELECT aws\_s3.table\_import\_from\_s3('<table>','', '(FORMAT CSV)', aws\_commons.create\_s3\_uri('<bucket>', '<folder>/<file>', '<region-code>'));**

**Inner Join**

**select o.order\_id, o.customer\_id, o.order\_date, c.email, c.phone from orders o inner join customers c on o.customer\_id = c.customer\_id;**

**Left Join**

**select o.order\_id, o.customer\_id, o.order\_date, c.email, c.phone from orders o left join customers c on o.customer\_id = c.customer\_id;**

**select o.order\_id, o.customer\_id, o.order\_date, c.email, c.phone from customers c left join orders o on o.customer\_id = c.customer\_id;**

**insert into customers values (99228, 'Mumbai', 12345, 'Mumbai', 'MH', '1981-11-11','India','abc@gmail.com','+919999111100');**

**insert into customers values (99229, 'Bangalore', 12345, 'BLR', 'KA', '1980-11-12','India','def@gmail.com','+919999222200');**

**select \* from customers;**

**select o.order\_id, o.customer\_id, o.order\_date, c.email, c.phone from customers c left join orders o on o.customer\_id = c.customer\_id;**

**Right Join**

**select o.order\_id, o.customer\_id, o.order\_date, c.email, c.phone from orders o right join customers c on o.customer\_id = c.customer\_id;**

**Full Join**

**select o.order\_id, o.customer\_id, o.order\_date, c.email, c.phone from orders o full join customers c on o.customer\_id = c.customer\_id;**

**Union, Intersect, Except**

**select \* from customers;**

**select customer\_id, address, dob from customers union select customer\_id, address, phone from customers;**

**select customer\_id, address, dob from customers union select customer\_id, address, country from customers;**

**select customer\_id, address, phone from customers union select customer\_id, address, country from customers;**

**select customer\_id, address from customers union select customer\_id, address, country from customers;**

**select customer\_id, address,dob from customers union select customer\_id, address, dob from customers;**

**select customer\_id, address,dob from customers union all select customer\_id, address, dob from customers;**

**select customer\_id, address,dob from customers union select customer\_id, payment\_method, order\_date from orders;**

**select \* from customers where country = 'Nepal';**

**select \* from customers where country = 'India';**

**select \* from customers where country = 'Nepal' union select \* from customers where country = 'India';**

**select \* from customers where country in ('Nepal','Hungary');**

**select \* from customers where country in ('India','Hungary');**

**(select \* from customers where country in ('Nepal','Hungary')) intersect (select \* from customers where country in ('India','Hungary'));**

**(select \* from customers where country in ('Nepal','Hungary')) except (select \* from customers where country in ('India','Hungary'));**

**(select \* from customers where country in ('India','Hungary')) except (select \* from customers where country in ('Nepal','Hungary'));**

**Create view**

**select o.order\_id, o.order\_date, o.payment\_method, o.discount, c.dob, c.email, s.seller\_company\_name from orders o, customers c, sellers s where o.customer\_id = c.customer\_id and o.seller\_id = s.seller\_id and payment\_method like 'Ame%';**

**create view amex\_view as select o.order\_id, o.order\_date, o.payment\_method, o.discount, c.dob, c.email, s.seller\_company\_name from orders o, customers c, sellers s where o.customer\_id = c.customer\_id and o.seller\_id = s.seller\_id and payment\_method like 'Ame%';**

**select \* from amex\_view;**

**create view visa\_view as select o.order\_id, o.order\_date, o.payment\_method, o.discount, c.dob, c.email, s.seller\_company\_name from orders o, customers c, sellers s where o.customer\_id = c.customer\_id and o.seller\_id = s.seller\_id and payment\_method = 'Visa';**

**select \* from visa\_view;**

**Analytics SQLs**

**truncate table orders, customers, sellers;**

**Load orders, sellers, customers from large dataset**

**SELECT aws\_s3.table\_import\_from\_s3('<table>','', '(FORMAT CSV)', aws\_commons.create\_s3\_uri('<bucket>', '<folder>/<file>', '<region-code>'));**

**Materialized View**

**select o.order\_id, c.customer\_id, o.order\_date, o.delivery\_date, o.price, c.address, c.email, c.phone, s.seller\_address, s.seller\_state, s.seller\_country from orders o, customers c, sellers s where o.customer\_id = c.customer\_id and o.seller\_id = s.seller\_id and s.seller\_country in ('France', 'Singapore') and c.dob > '1990-01-01';**

**create materialized view millenial\_mv as select o.order\_id, c.customer\_id, o.order\_date, o.delivery\_date, o.price, c.address, c.email, c.phone, s.seller\_address, s.seller\_state, s.seller\_country from orders o, customers c, sellers s where o.customer\_id = c.customer\_id and o.seller\_id = s.seller\_id and s.seller\_country in ('France', 'Singapore') and c.dob > '1990-01-01';**

**select count(\*) from millenial\_mv;**

**select \* from millenial\_mv;**

**delete from customers where customer\_id > 100;**

**select o.order\_id, c.customer\_id, o.order\_date, o.delivery\_date, o.price, c.address, c.email, c.phone, s.seller\_address, s.seller\_state, s.seller\_country from orders o, customers c, sellers s where o.customer\_id = c.customer\_id and o.seller\_id = s.seller\_id and s.seller\_country in ('France', 'Singapore') and c.dob > '1990-01-01';**

**select \* from millenial\_mv;**

**refresh materialized view millenial\_mv;**

**select \* from millenial\_mv;**

**Common Table Expression (CTE)**

**select customer\_id, (delivery\_date - order\_date) as lead\_days, (price - (price\*discount/100)) as discounted\_price, quantity from orders;**

**with order\_details as (select customer\_id, (delivery\_date - order\_date) as lead\_days, (price - (price\*discount/100)) as discounted\_price, quantity from orders)**

**select c.state, cast(avg(od.lead\_days) as decimal(10,2)) as avg\_lead\_days, cast(avg(od.discounted\_price) as decimal(10,2)) as avg\_discounted\_price, cast(avg(od.quantity) as decimal(10,2)) as avg\_qty from order\_details od, customers c where c.customer\_id = od.customer\_id group by c.state;**

**Window Functions**

**Find the top orders per customer by price and quantity.**

**select customer\_id, count(\*) from orders group by customer\_id;**

**select customer\_id, order\_id, quantity, row\_number() over (partition by customer\_id order by quantity desc) as quantity\_rank from orders;**

**select customer\_id, order\_id, quantity, rank() over (partition by customer\_id order by quantity desc) as quantity\_rank from orders;**

**select customer\_id, order\_id, quantity, dense\_rank() over (partition by customer\_id order by quantity desc) as quantity\_rank from orders;**

**select customer\_id, order\_id, quantity, price, dense\_rank() over (partition by customer\_id order by price desc) as quantity\_rank from orders;**

**with rank\_by\_price as (select customer\_id, order\_id, quantity, price, dense\_rank() over (partition by customer\_id order by price desc) as quantity\_rank from orders) select customer\_id, order\_id, quantity, price, quantity\_rank from rank\_by\_price where quantity\_rank < 4;**

**Find highest no. of items sold for each state**

**select c.state, o.order\_id, o.quantity, dense\_rank() over (partition by c.state order by o.quantity desc) as order\_rank from customers c inner join orders o on o.customer\_id = c.customer\_id;**

**with cte as (select c.state, o.order\_id, o.quantity, dense\_rank() over (partition by c.state order by o.quantity desc) as order\_rank from customers c inner join orders o on o.customer\_id = c.customer\_id) select state, order\_id, quantity, order\_rank from cte where order\_rank <=3;**

**Find state-wise top seller**

**Do it yourself**

**Find top orders by individual sellers**

**Do it yourself**

**MERGE Statement**

**create table customers\_staging (like customers);**

**merge into customers c using customers\_staging c\_s**

**on c.customer\_id = c\_s.customer\_id**

**when matched then update set country = 'INDIA'**

**when not matched then insert values(c\_s.\*);**

**merge into customers c using customers\_staging c\_s**

**on c.customer\_id = c\_s.customer\_id**

**when matched and c.country = 'INDIA' then delete**

**when not matched then insert values(c\_s.\*);**