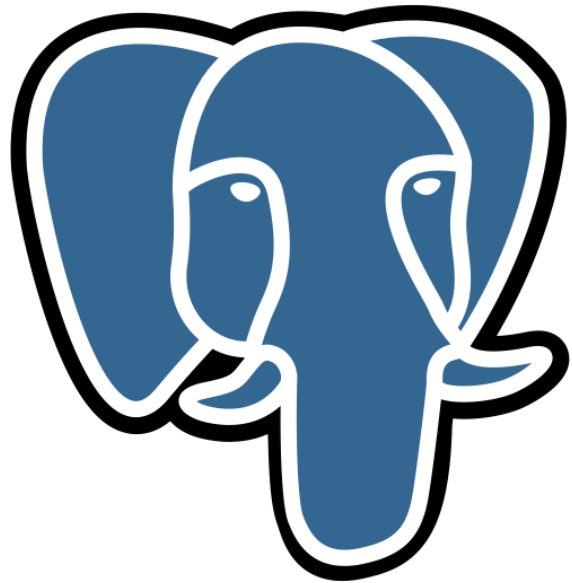




Start-TechAcademy



The Complete SQL Masterclass For Data Analytics

Exercise 1

1. Create a Database 'Classroom'
2. Create a table named 'Science_class' with the following properties
 - 3 Cloumns (Enrollment_no INT, Name VARCHAR, Science_Marks INT)

Solution :

create table science_class

(Enrollment_no INT, Name VARCHAR, Science_Marks INT);



Exercise 2

1. Insert the following data into Science_class using insert into command

1	Popeye	33
2	Olive	54
3	Brutus	98

2. Import data from csv file 'Student.csv' attached in resources to Science_class to insert data of next 8 students

Solution:

```
insert into science_class values (1,'Popeye',33);
```

```
insert into science_class values (2,'Olive',54);
```

```
insert into science_class values (3,'Brutus',98);
```

```
COPY science_class FROM 'address/student.csv' CSV HEADER;
```



Exercise 3

1. Retrieve all data from the table 'Science_Class'
2. Retrieve the name of students who have scored more than 60 marks
3. Retrieve all data of students who have scored more than 35 but less than 60 marks
4. Retrieve all other students i.e. who have scored less than or equal to 35 or more than or equal to 60.



Sol 3

1. `select * from science_class;`
2. `select name from science_class where science_marks>60;`
3. `select * from science_class where science_marks between 35 and 60;`
4. `select * from science_class where science_marks not between 35 and 60;`



Exercise 4

1. Update the marks of Popeye to 45
2. Delete the row containing details of student named 'Robb'
3. Rename column 'Name' to 'student_name'.



Sol 4

```
UPDATE science_class
```

```
SET science_marks = 45
```

```
WHERE name = 'Popeye';
```

```
delete from science_class WHERE name = 'Robb';
```

```
alter table science_class rename column name to student_name;
```



Exercise 5

1. Backup this database into a TAR file
2. Drop the 'science_class' table
3. Restore from the backup file to get back the deleted table

DIY

Post your Questions in the discussion forum



Exercise 6

1. In the database Supermart_DB, find the following
 - a. Get the list of all cities where the region is South or east without any duplicates using IN statement
 - b. Get the list of all orders where the 'sales' value is between 100 and 500 using the BETWEEN operator
 - c. Get the list of customers whose last name contains only 4 characters using LIKE



Sol 6

1. `select distinct city from customer where region in ('South','East');`
2. `select * from sales where sales between 100 and 500;`
3. `select * from customer where customer_name like '% ____';`



Exercise 7

1. Retrieve all orders where 'discount' value is greater than zero ordered in descending order basis 'discount' value
2. Limit the number of results in above query to top 10



Sol 7

1. `select * from sales where discount > 0 order by discount desc;`
2. `select * from sales where discount > 0 order by discount desc limit 10;`



Exercise 8

1. Find the sum of all 'sales' values.
2. Find count of the number of customers in north region with age between 20 and 30
3. Find the average age of East region customers
4. Find the Minimum and Maximum aged customer from Philadelphia



Sol 8

1. `select sum(sales) from sales; --2297200.86`
2. `select count(*) from customer where age between 20 and 30;--150`
3. `select avg(age) from customer where region = 'East';--44.33`
4. `select min (age) as min_age , max(age) as max_age from customer
where city = 'Philadelphia';--18,70`



Exercise 9

1. Make a dashboard showing the following figures for each product ID
 - a) Total sales (in \$) *order by this column in descending*
 - b) Total sales quantity
 - c) Number of orders
 - d) Max Sales value
 - e) Min Sales value
 - f) Average sales value
2. Get the list of product ID's where the quantity of product sold is greater than 10



Sol 9

1. `select product_id,sum(sales) as Total_sales, sum(quantity) as total_quantity, count(order_id) as total_order, min(sales) as min_sales , max(sales) as max_sales, avg(sales) as avg_sales from sales group by product_id order by total_sales desc;`
2. `select product_id, sum(quantity) as total_quantity from sales group by product_id having sum(quantity) > 10;`



Exercise 10

1. Find the total sales done in every state for customer_20_60 and sales_2015 table

Hint: Use Joins and Group By command

2. Get data containing Product_id, product name, category, total sales value of that product and total quantity sold. (Use sales and product table)



Sol 10

```
select b.state, sum(sales) as total_sales  
from sales_2015 as a left join customer_20_60 as b  
on a.customer_id = b.customer_id  
group by b.state;
```

```
select a.*, sum(b.sales) as total_sales, sum(quantity) as total_quantity  
from product as a left join sales as b  
on a.product_id = b.product_id  
group by a.product_id;
```



Exercise 11

1. Get data with all columns of sales table, and customer name, customer age, product name and category are in the same result set. (use JOIN in subquery)



Sol 11

```
select c.customer_name, c.age, sp.* from  
customer as c  
right join (select s.*, p.product_name, p.category  
            from sales as s  
            left join product as p  
            on s.product_id = p.product_id) as sp  
on c.customer_id = sp.customer_id;
```



Exercise 12

1. Create a View which contains order_line, Product_id, sales and discount value of the first order date in the sales table and name it as "Daily_Billing"
2. Delete this View



Sol 12

1. create view Daily_Billing as select order_line, product_id, sales, discount from sales where order_date in (select max(order_date) from sales);
2. drop view Daily_Billing;



Exercise 13

1. Find Maximum length of characters in the Product name string from Product table
2. Retrieve product name, sub-category and category from Product table and an additional column named “product_details” which contains a concatenated string of product name, sub-category and category
3. Analyze the product_id column and take out the three parts composing the product_id in three different columns
4. List down comma separated product name where sub-category is either Chairs or Tables



Sol 13

```
select max(length(product_name)) from product;
```

```
select product_name,sub_category, category, (product_name||' , '||sub_category||' ,  
'||category) as product_details from product;
```

```
select product_id, substring(product_id for 3) as category_short, substring(product_id from  
5 for 2) as sub_short , substring(product_id from 8) as id from product;
```

```
select string_agg(product_name,', ') from product where sub_category in ('Chairs','Tables');
```



Exercise 14

1. You are running a lottery for your customers. So, pick a list of 5 Lucky customers from customer table using random function
2. Suppose you cannot charge the customer in fraction points. So, for sales value of 1.63, you will get either 1 or 2. In such a scenario, find out
 - a) Total sales revenue if you are charging the lower integer value of sales always
 - b) Total sales revenue if you are charging the higher integer value of sales always
 - c) Total sales revenue if you are rounding-off the sales always



Sol 14

```
select customer_id, random() as rand_n from customer order by rand_n limit 5;
```

```
select sum(ceil(sales)) as higher_int_sales, sum(floor(sales)) as lower_int_sales,  
sum(round(sales)) as round_int_sales from sales;
```



Exercise 15

1. Find out the current age of “Batman” who was born on “April 6, 1939” in Years, months and days
2. Analyze and find out the monthly sales of sub-category chair. Do you observe any seasonality in sales of this sub-category



Sol 15

```
select age(current_date,'1939-04-06');
```

```
select extract(month from order_date) as month_n, sum(sales) as total_sales from sales  
where product_id in (select product_id from product where sub_category = 'Chairs')  
group by month_n  
order by month_n ;
```



Exercise 16

1. Find out all customers who have first name and last name of 5 characters each and last name starts with "a/b/c/d"
2. Create a table "zipcode" and insert the below data in it

PIN/ZIP codes
234432
23345
sdfe4
123&3
67424
7895432
12312

Find out the valid zipcodes from this table (5 or 6 Numeric characters)



Exercise 16

```
select * from customer where customer_name ~* '^[a-z]{5}\s(a|b|c|d)[a-z]{4}$';
```

```
create table zipcode (zip varchar);
```

```
insert into zipcode values ('234432');
```

```
insert into zipcode values ('23345');
```

```
insert into zipcode values ('sdfe4');
```

```
insert into zipcode values ('123&3');
```

```
insert into zipcode values ('67424');
```

```
insert into zipcode values ('7895432');
```

```
insert into zipcode values ('12312');
```

```
select * from zipcode where zip ~* '^[0-9]{5,6}$';
```

