

1. Scenario: You have a table called "orders" that contains information about customer orders, including the customer ID, the order date, and the total order amount. Write a query to retrieve the total revenue generated by each customer.

```
SELECT CUSTOMER ID, SUM(total order amount) FROM orders GROUP BY CUSTOMER ID;
```

2. Scenario: You have a table called "employees" that contains information about company employees, including their names and the departments they work in. Write a query to retrieve the name of each department and the number of employees working in that department.

```
SELECT DEPARTMENTS, COUNT(DEPARTMENT), FROM EMPLOYEES GROUP BY DEPARTMENT;
```

3. Scenario: You have a table called "products" that contains information about products sold by your company, including the product ID, the product name, and the price. Write a query to retrieve the top 10 most expensive products.

```
Select * from products order by price desc limit 10;
```

```
Select TOP 10 * from products order by price desc;
```

4. Scenario: You have a table called "students" that contains information about students in a school, including their names and their grades in different subjects. Write a query to retrieve the average grade for each subject.

```
Select avg(grades) from students group by names;
```

5. Scenario: You have a table called "orders" that contains information about customer orders, including the customer ID, the order date, and the total order amount. Write a query to retrieve the total revenue generated by each customer for orders placed in the current year.

```
Select CustomerID,OrderDate,sum(total order amount) as revenue from orders where  
year(orderdate)=year(curdate()) group by CustomerID;
```

6. Scenario: You have a table called "sales" that contains information about the sales of a product, including the sales date, the product ID, the quantity sold, and the price. Write a query to calculate the revenue generated by the product for each month.

```
Select DATE_FORMAT(date, '%Y-%m') AS year_month ,sum(quantity_sold*price) as revenue from
sales group by DATE_FORMAT(date, '%Y-%m');
```

7. Scenario: You have a table called "customers" that contains information about customers, including their age, gender, and location. Write a query to calculate the average age of customers by gender and location.

```
Select gender, location, avg(age) as Avg_Age from customers group by gender, location;
```

8. Scenario: You have a table called "website_traffic" that contains information about website traffic, including the date, the number of visitors, and the source of the traffic. Write a query to calculate the percentage of traffic that comes from each source.

```
//Select ( ( (number_of_visitors)/sum(number_of_visitors) ) *100) as Percentage from website_traffic
group by source;
```

```
Select source, (sum(number_of_visitors)/sum(select number_of_visitors from website_traffic)) *100
as Percentage_Traffic from website_traffic group by source;
```

9. Scenario: You have a table called "sales" that contains information about the sales of a product, including the sales date, the product ID, the quantity sold, and the price. Write a query to calculate the total revenue generated by the product for each quarter of the year.

```
Select concat(year(sales_date),'-',quarter(sales_date)) ,sum(quantity_sold*price) as revenue from
sales group by concat(year(sales_date),'-',quarter(sales_date));
```

10. Scenario: You have a table called "customer_feedback" that contains information about customer feedback, including the date, the feedback score, and the comments. Write a query to calculate the average feedback score by month and year.

```
SELECT DATE_FORMAT(date, '%Y-%m') AS year_month, AVG(feedback_score) AS
average_feedback_score
FROM customer_feedback
```

GROUP BY DATE_FORMAT(date, '%Y-%m');

11.Scenario: You have a table called "product_reviews" that contains information about product reviews, including the product ID, the review date, the review text, and the reviewer's score. Write a query to calculate the average score for each product.

Select avg(score) from product_reviews GROUP BY product ID;

12.Scenario: You have a table called "website_sessions" that contains information about website sessions, including the date, the user ID, and the duration of the session. Write a query to calculate the average duration of sessions by month.

SELECT AVG(DURATION) AS AVG_DURATION FROM WEBSITE_SESSIONS GROUP BY MONTH(DATE);

or

Select DATE_FORMAT(date, '%Y-%m'),avg(Duration) as Average_duration from website_sessions group by DATE_FORMAT(date, '%Y-%m');

13.Scenario: You have a table called "product_ratings" that contains information about product ratings, including the product ID, the rating date, and the rating score. Write a query to calculate the average rating score by product for the last 30 days.

Select avg(rating_score) from product_rating datediff(Day,date,curdate())<=30 group by productid;

14.Scenario: You have two tables: "customers" and "orders". The "customers" table contains information about customers, including their ID and name. The "orders" table contains information about customer orders, including the customer ID, the order ID, and the order date. Write a query to retrieve the name of each customer and the date of their most recent order.

Select c.name, o.order_date from customers c join orders o on c.id=o.id where datediff(Week, o.order_date = (
SELECT MAX(order_date)
FROM orders o2
WHERE o2.customer_id = c.customer_id
);

15. Scenario: You have two tables: "employees" and "departments". The "employees" table contains information about company employees, including their names and the departments they work in.

The "departments" table contains information about company departments, including the department ID and the department name. Write a query to retrieve the name of each employee and the name of their department.

```
Select e.name, d.department_name from employees e join departments d on e.department_id = d.department_id;
```