

# Aggregates and Builtin Functions

1. Article to Read

<https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>

2. HW questions

<https://leetcode.com/problems/duplicate-emails/>

<https://leetcode.com/problems/customer-placing-the-largest-number-of-orders/>

<https://leetcode.com/problems/classes-more-than-5-students/>

<https://leetcode.com/problems/rising-temperature/>

3. Class queries

```
use sql_store;
```

```
SELECT *  
FROM customers;
```

```
-- how many customers have a phone number
```

```
SELECT COUNT(phone)  
FROM customers;
```

```
SELECT COUNT(customer_id)  
FROM customers  
WHERE phone IS NOT NULL;
```

```
-- all rows in a column
```

```
SELECT COUNT(*), COUNT(customer_id)  
FROM customers;
```

```
-- MIN, MAX, AVG, SUM, COUNT
```

```
SELECT MIN(points), MAX(points), AVG(points), SUM(points)  
FROM customers;
```

```
-- number of orders per status
```

```
SELECT status, COUNT(order_id) AS Num  
FROM orders  
GROUP BY status;
```

```
-- for each status, give the total quantity of products
```

```
SELECT o.status, SUM(p.quantity_in_stock)  
FROM orders o
```

```
JOIN order_items oi
    ON o.order_id = oi.order_id
JOIN products p
    ON oi.product_id = p.product_id
GROUP BY o.status;
```

```
-- for each status, give the total quantity of products
SELECT o.order_id, COUNT(p.product_id)
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
JOIN products p
    ON oi.product_id = p.product_id
GROUP BY o.order_id;
```

```
SELECT *
FROM order_items;
```

```
-- Give the total quantity of products with unit_price >= 2.00
SELECT SUM(quantity_in_stock)
FROM products
WHERE unit_price >= 2.00;
```

```
USE sql_invoicing;
```

```
-- total amount paid by clients (for only those clients who have atleast made 3 payments)
SELECT client_id, SUM(amount)
FROM payments
GROUP BY client_id
HAVING COUNT(*) >= 3
```

```
use sql_store;
SELECT *
FROM customers;
```

```
-- sum of points of all customers (born after 1980)
-- grouped by their birth year
-- only for the years if there are more than 3 people born in that year
SELECT YEAR(birth_date), SUM(points), COUNT(*)
FROM customers
WHERE YEAR(birth_date) > 1980
GROUP BY YEAR(birth_date)
HAVING COUNT(*) >= 3
```

```
-- get orders shipped within 1 day of ordering
SELECT shipped_date, order_date, datediff(shipped_date, order_date)
FROM orders
WHERE shipped_date IS NOT NULL AND datediff(shipped_date, order_date) <= 1;
```