-- BASIC SELECT AND WHERE

1. https://leetcode.com/problems/big-countries/

```
SELECT name ,population,area
FROM world
WHERE area>=3000000
OR population>=25000000;
```

2. https://leetcode.com/problems/article-views-i/description/

```
SELECT distinct(author_id)
AS id
FROM views
WHERE author_id=viewer_id
ORDER BY author_id;
```

-- CONDITION FILTERING

1. https://leetcode.com/problems/not-boring-movies/

```
SELECT *
FROM Cinema
WHERE MOD(id, 2) = 1 AND description != 'boring'
ORDER BY rating DESC
```

2. https://leetcode.com/problems/find-customer-referee/

```
SELECT name
FROM Customer
WHERE referee_id!=2
OR referee_id is null;
```

3. https://leetcode.com/problems/triangle-judgement/

```
SELECT *, IF(x+y>z
AND y+z>x
AND z+x>y, "Yes", "No")
AS triangle
FROM Triangle
```

4. https://leetcode.com/problems/recyclable-and-low-fat-products/

```
SELECT product_id
FROM products
WHERE low_fats='Y'
AND recyclable='y';
```

5. https://leetcode.com/problems/calculate-special-bonus/

```
SELECT employee_id, IF(name NOT LIKE "M%"
AND employee_id%2 <> 0, salary, 0)
AS bonus
FROM Employees
ORDER BY employee_id
```

6. https://leetcode.com/problems/the-latest-login-in-2020/description/

```
SELECT user_id, MAX(time_stamp)
AS last_stamp
FROM logins
```

```
WHERE year(time_Stamp)='2020'
GROUP BY user_id ;
```

-- BASIC GROUPING

1. https://leetcode.com/problems/duplicate-emails/

```
SELECT email
FROM person
GROUP BY email
HAVING count(email)>1 ;
```

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

```
SELECT customer_number
FROM orders
GROUP BY customer_number
ORDER BY count(order_number) desc
LIMIT 1;
```

3. https://leetcode.com/problems/classes-more-than-5-students/description/

```
SELECT class
FROM courses
GROUP BY class
HAVING count(student)>=5;
```

4. https://leetcode.com/problems/actors-and-directors-who-cooperated-at-least-three-times/

```
SELECT actor_id,director_id
FROM ActorDirector
GROUP BY actor_id,director_id
HAVING count(timestamp)>=3;
```

5. https://leetcode.com/problems/game-play-analysis-i/description/

```
SELECT player_id, min(event_date)
AS first_login
FROM activity
GROUP BY player_id;
```

6. https://leetcode.com/problems/daily-leads-and-partners/description/

```
SELECT date_id, make_name , count(distinct(lead_id)) AS unique_leads,
count(distinct(partner_id)) AS unique_partners
FROM DailySales
GROUP BY date_id, make_name;
```

7. https://leetcode.com/problems/find-followers-count/

```
SELECT user_id , count(follower_id)
AS followers_count
FROM followers
GROUP BY user_id
ORDER BY user_id;
```

8. https://leetcode.com/problems/find-total-time-spent-by-each-employee/

```
SELECT event_day AS day, emp_id, sum(out_time-in_time)AS total_time
FROM employees
```

```
GROUP BY emp_id, event_day;
```

9. https://leetcode.com/problems/odd-and-even-transactions/description/

```
SELECT transaction_date,
SUM(CASE WHEN amount%2=1 THEN amount ELSE 0 END) as odd_sum,
SUM(CASE WHEN amount%2=0 THEN amount ELSE 0 END) as even_sum
FROM transactions
GROUP BY transaction_date
ORDER BY transaction_date;
```

-- BASIC JOINS

1. https://leetcode.com/problems/combine-two-tables/

```
SELECT p.firstName, p.lastName, a.city, a.state
FROM person p
LEFT JOIN address a
ON p.personId=a.personId;
```

2. https://leetcode.com/problems/product-sales-analysis-i/description/

```
SELECT p.product_name, s.year, s.price
FROM Sales s
INNER JOIN Product p
ON s.product_id=p.product_id;
```

3. https://leetcode.com/problems/employee-bonus/

```
SELECT e.name, b.bonus
FROM employee e
LEFT JOIN bonus b
ON e.empId=b.empId
WHERE b.bonus<1000
OR bonus is null;
```

4. https://leetcode.com/problems/replace-employee-id-with-the-unique-identifier/

```
SELECT u.unique_id , e.name
FROM employees e
LEFT JOIN Employeeuni u
ON e.id=u.id;
```

5. https://leetcode.com/problems/project-employees-i/description/

```
SELECT p.project_id, ROUND(AVG(e.experience_years),2)
AS average_years
FROM Project p
LEFT JOIN Employee e
ON p.employee_id = e.employee_id
GROUP BY p.project_id
```

6. https://leetcode.com/problems/replace-employee-id-with-the-unique-identifier/

```
SELECT u.unique_id , e.name
FROM employees e
LEFT JOIN employeeuni u
ON e.id=u.id;
```

7. https://leetcode.com/problems/project-employees-i/description/

```
SELECT p.project_id, ROUND(AVG(e.experience_years),2) AS average_years
FROM project p
LEFT JOIN employee e
ON p.employee_id = e.employee_id
GROUP BY p.project_id
```

-- SELF JOIN

1. https://leetcode.com/problems/employees-earning-more-than-their-managers/

```
SELECT e1.name
AS Employee
FROM Employee e1
INNER JOIN Employee e2
WHERE e1.Managerid = e2.id
AND e1.salary> e2.salary;
```

2. https://leetcode.com/problems/rising-temperature/

```
SELECT w1.id
FROM Weather w1, Weather w2
WHERE DATEDIFF(w1.recordDate, w2.recordDate) = 1
AND w1.temperature > w2.temperature;
```

3. https://leetcode.com/problems/average-time-of-process-per-machine/

```
SELECT a1.machine_id, round(avg(a2.timestamp-a1.timestamp), 3)
AS processing_time
FROM Activity a1
JOIN Activity a2
ON a1.machine_id=a2.machine_id
AND a1.process_id=a2.process_id
AND a1.activity_type='start' AND a2.activity_type='end'
GROUP BY a1.machine_id;
```