

Sub-Query

- Used when we want to filter query based on another query

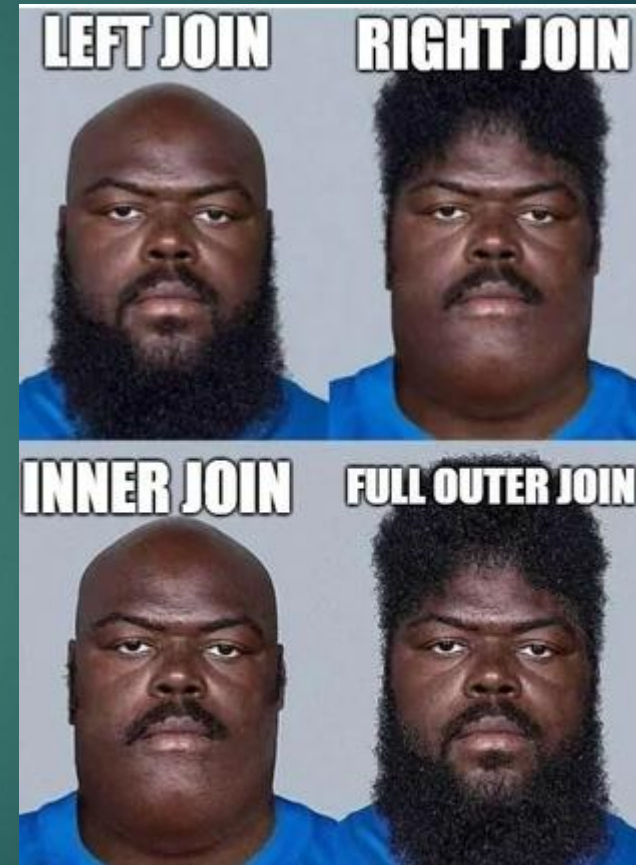
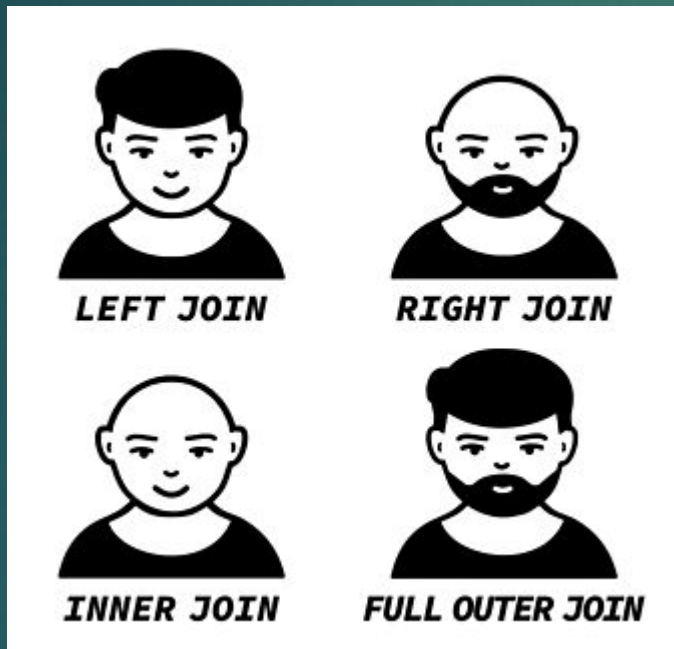
- Example:

```
SELECT *  
FROM transaction  
WHERE customer_id in (  
    SELECT id from customer WHERE address= 'KATHMANDU'  
)
```

JOIN

- Used to lookup & find information from related table
- Referential columns are used for join
- Types:
 - INNER JOIN
 - LEFT JOIN
 - RIGHT JOIN
 - CROSS JOIN
 - SELF JOIN

JOIN



Syntax

```
SELECT a.field_1, b.field_2  
FROM table_1 a  
[LEFT | INNER | CORSS] JOIN table_2 b  
    ON a.cmn_field = b.common_field;
```

Aggregate

- Similar to what we learn in NumPy, aggregate function evaluates over multiple records & returns result
- Common Functions: sum, count, min, max, avg
- Syntax:
 - `SELECT count(*) FROM student_info WHERE {condn};`
 - `SELECT sum(salary) FROM employee_info;`
 - `SELECT MIN(price) FROM product;`

GROUP BY

- Used when we want different aggregation per group
- Syntax:
 - `SELECT dept, sum(salary) FROM employee GROUP BY dept;`
 - `SELECT categ, MIN(price) FROM product GROUP BY categ;`
 - `SELECT a, b, count(1) FROM emp GROUP BY a,b;`

HAVING

- Used when we want filter on **grouped result**
- WHERE filter individual record
- HAVING filters grouped record
- Syntax:
SELECT dept, sum(salary)
FROM employee
GROUP BY dept
HAVING sum(salary) > 100;

Overall Syntax

SELECT

FROM

JOIN ON

WHERE ...

GROUP BY

HAVING

ORDER BY

LIMIT



Exercise

- 
- Set functions: union, union all, except, intersect
 - CTE
 - Window Function: rank, cumsum