# SQL Selecting Data

## The Query Framework

```
SELECT attributes,            Choose columns to select
    aggregation function      Choose aggregate features
FROM relation                 Choose table to select from
WHERE condition               Filter selection
GROUP BY attributes           Aggregate
HAVING aggregation            Filter aggregate
    function condition
ORDER BY attributes           Sort selection
LIMIT n;                      Restrict number of results
```
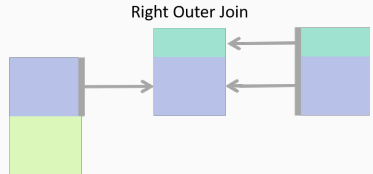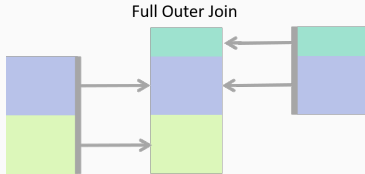
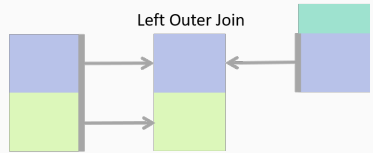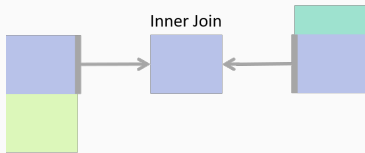# SQL Joins

## Joins

`table1` INNER JOIN `table2`           All records in both
`table1` LEFT OUTER JOIN `table2`      All records in table 1
`table1` RIGHT OUTER JOIN `table2`     All records in table 2

Syntax:

- We alias both tables and specify which column to join on

- e.g. `tableA A` inner join `tableB B` on `A.id = B.id`

# Joins



Inner Join

Left Outer Join

Full Outer Join

Right Outer Join

Note: MySQL does not support full outer joins, though it is possible to achieve this result by combining a left and right outer join with `UNION`.

# SQL Aggregation

## Aggregation

```
SELECT attr, aggregation function
FROM relation
GROUP BY attr
```

Note:

- Aggregation function applies to the group as a whole

- If aggregating, SELECT accepts only group-level
  variables, or aggregation functions

## Aggregation Functions

| | |
|---|---|
| COUNT(*) | Count number of rows in group |
| COUNT(attr) | Count non-null* rows |
| COUNT(DISTINCT attr) | Count distinct, non-null* rows |
| MAX(attr) | Maximum attribute value |
| MIN(attr) | Minimum attribute value |
| SUM(attr) | Sum of attribute values |
| AVG(attr) | Average of attribute values |
| STDDEV(attr) | Standard deviation of attr. values |

* Here, 'non-null' refers to the attribute supplied as an argument.

## Aggregation with Conditions

```
SELECT attr, aggregation function
FROM relation
GROUP BY attr
HAVING aggregation function condition;
```

Note:

- WHERE applies to rows, before computing the aggregate
- HAVING applies to aggregate values only