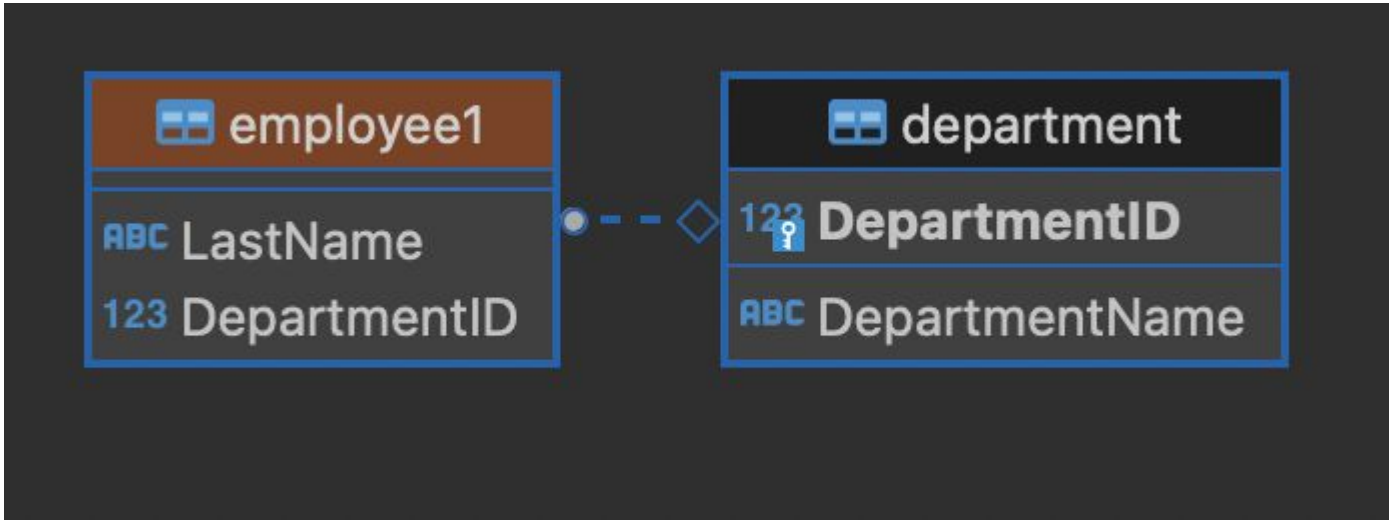
Joins

Ruirui Zhang, Tejashree Ladhake

The database we'll use



Joins

A JOIN combines rows from multiple tables

Based on a condition you specifies, which is always related columns

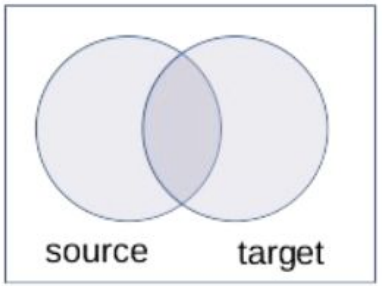
Syntax:

SELECT column1, columns2 FROM source JOIN target

ON source_col= target_col / USING(col)

Inner Join

INNER JOINS return all rows from multiple tables where the join condition is met.



INNER JOIN

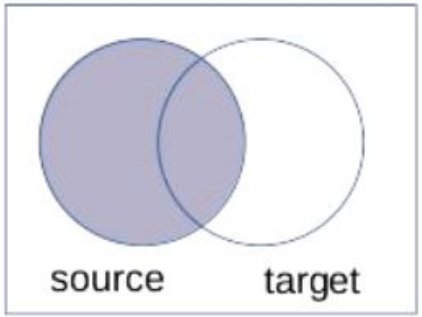
Question: Find all the employees with department.

Syntax:

```
SELECT * FROM employee1 JOIN department
ON employee1.DepartmentID = department.DepartmentID;
```

Left Join

LEFT JOINS returns all rows from the LEFT-hand table specified in the ON condition and **only** those rows from the other table where join condition is met.



LEFT JOIN

Question: Find all department for each employee.

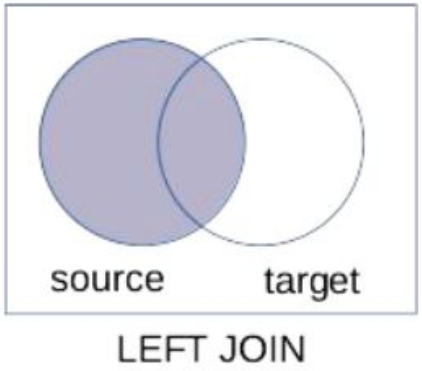
(Some employees do not have any department, but we want to see them anyway)

Syntax:

```
SELECT * FROM employee1 LEFT JOIN department
ON employee1.DepartmentID = department.DepartmentID;
```

Left Join

LEFT JOINS returns all rows from the LEFT-hand table specified in the ON condition and **only** those rows from the other table where join condition is met.

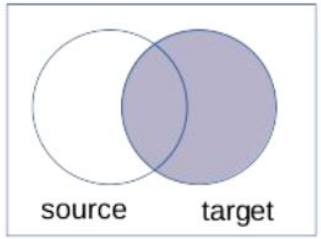


Question: Who do not have any department?

Syntax:

```
SELECT * FROM employee1 LEFT JOIN department
ON employee1.DepartmentID = department.DepartmentID
WHERE employee1.DepartmentID IS NULL;
```

Right Join



RIGHT JOIN

RIGHT JOINS returns all rows from the RIGHT-hand table specified in the ON condition and **only** those rows from the other table where join condition is met.

Right and left outer joins are functionally equivalent. Neither provides any functionality that the other does not, so right and left outer joins may replace each other as long as the table order is switched.

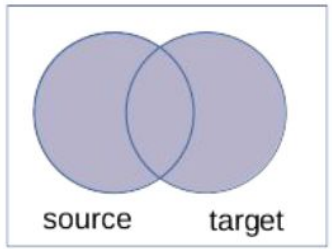
Question: Find all department for each employee..

Syntax:

```
SELECT * FROM department RIGHT JOIN employee1
ON employee1.DepartmentID = department.DepartmentID;
```

Full Join

Conceptually, a **full join** combines the effect of applying both left and right joins



FULL JOIN

Question: Show all employees (Even those without department)

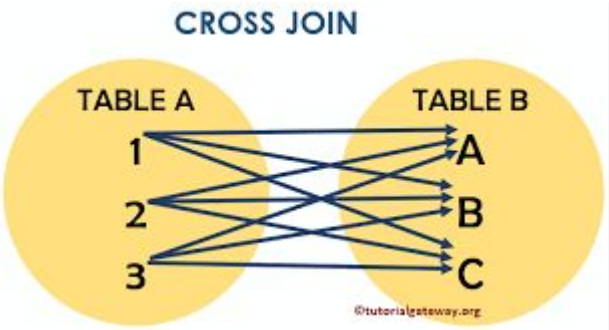
and all department (Even those without employee) .

Syntax:

```
SELECT *
FROM department FULL JOIN employee1
ON employee1.DepartmentID = department.DepartmentID;
```

```
SELECT *
FROM department LEFT JOIN employee1
ON employee1.DepartmentID =
department.DepartmentID
UNION
SELECT *
FROM department  RIGHT JOIN employee1
ON employee1.DepartmentID =
department.DepartmentID;
```

Cross Join



CROSS JOIN returns the **Cartesian Product** of rows from tables in the join.

Question: Find all possible combinations of employees and departments.

Syntax:

```
SELECT * FROM employee1 CROSS JOIN department;
```

Natural Join

NATURAL JOIN implicitly compare all columns in both tables that have the same column-names in the joined tables

Question: Find the employees those have department assigned.

Syntax:

```
SELECT * FROM employee1 NATURAL JOIN department;
```