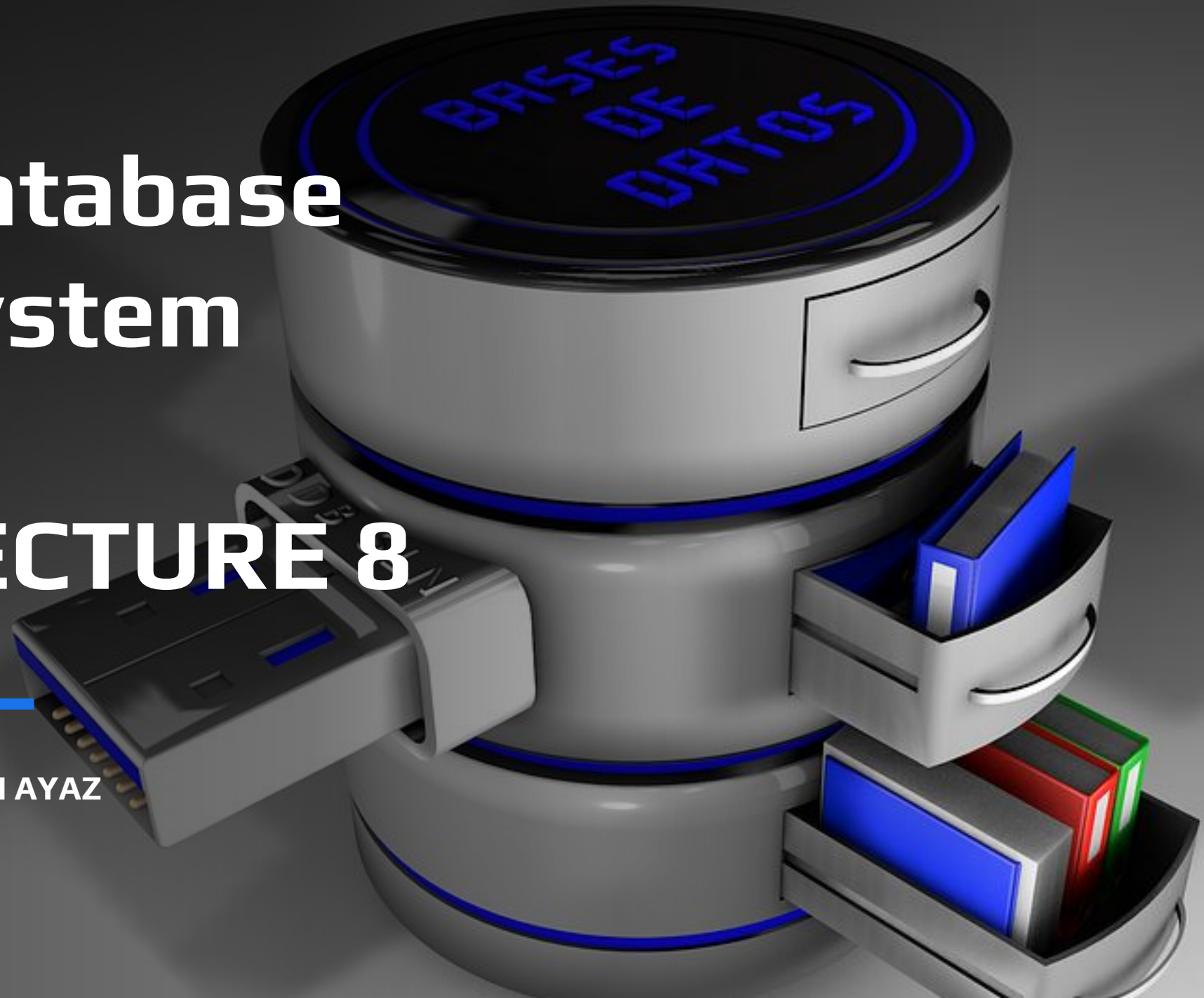
Database System

LECTURE 8

AHSAN AYAZ

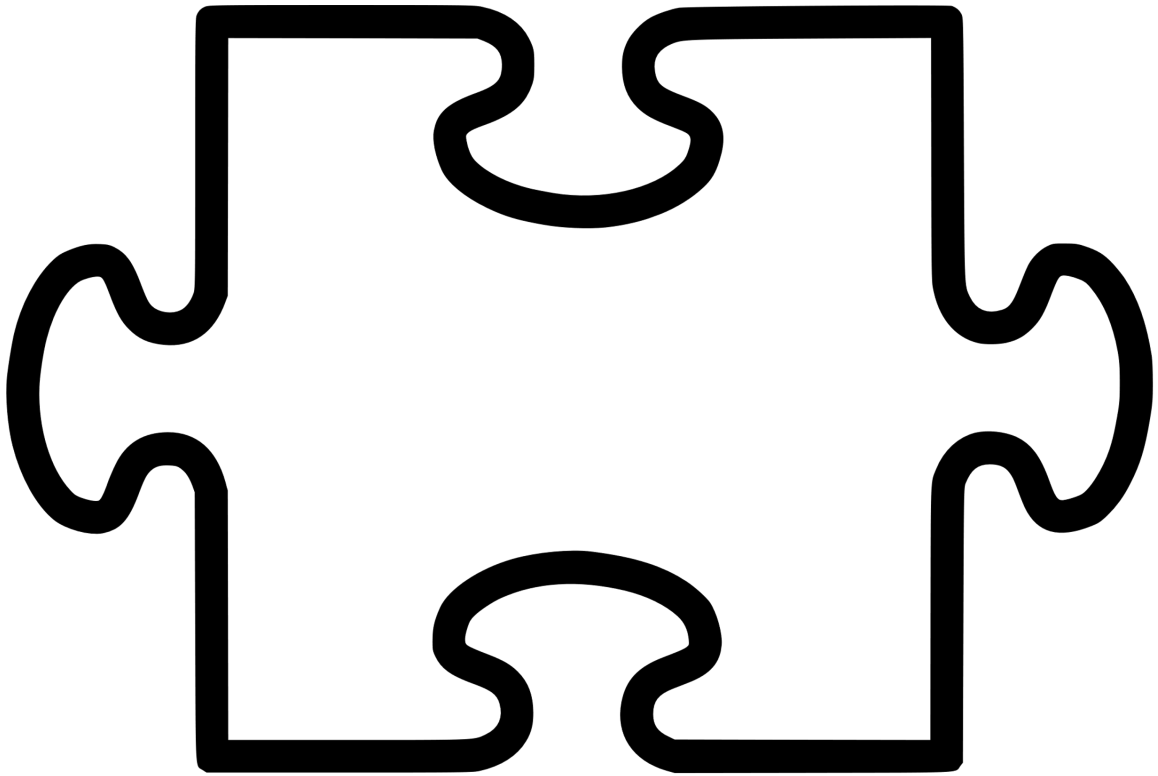


TODAY TOPIC COVER

SQL Joins

Types Of Joins

SQL JOIN



A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

TYPES OF JOINS

- INNER JOIN
- LEFT JOIN (or LEFT OUTER JOIN)
- RIGHT JOIN (or RIGHT OUTER JOIN)
- FULL JOIN (or FULL OUTER JOIN)
- CROSS JOIN

CREATE A TABLE

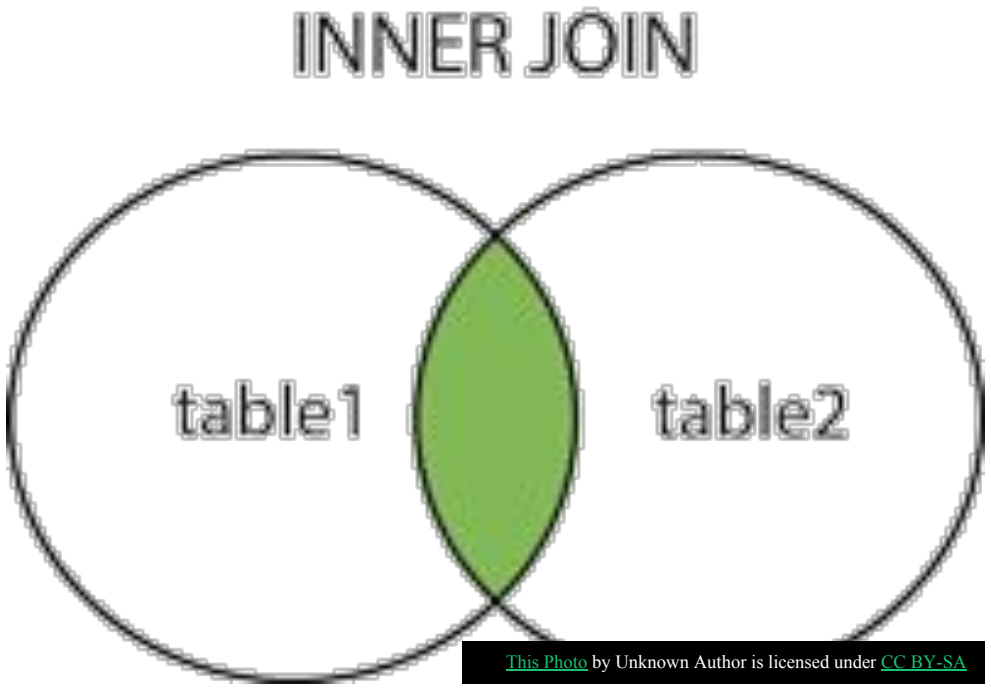
```
CREATE TABLE departments (
dept_id INT PRIMARY KEY,
dept_name VARCHAR(50)
);
CREATE TABLE employees (
emp_id INT PRIMARY KEY,
 emp_name VARCHAR(50),
dept_id INT,
FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);
```

INSERT VALUES IN TABLES

- INSERT INTO departments VALUES (1, 'HR'), (2, 'IT'), (3, 'Finance’) (4, 'HR'), (5, 'IT'), (6, 'Finance’);

- INSERT INTO employees VALUES

(101, 'Alice', 1), (102, 'Bob', 2), (103, 'Charlie', 3);



INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables.

SYNTAX:

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name =
table2.column_name;
```

INNER JOIN EXAMPLE

List all employees and their respective department names.

```
SELECT employees.emp_id, employees.emp_name, departmentS.dept_name
FROM employees
INNER JOIN departmentS ON employees.dept_id = departments.dept_id;
```

OR

```
SELECT emp_id, emp_name, dept_name
FROM employees
INNER JOIN departmentS ON employees.dept_id = departments.dept_id;
```

INNER JOIN EXAMPLES

List employee names starting with 'A' and their department:

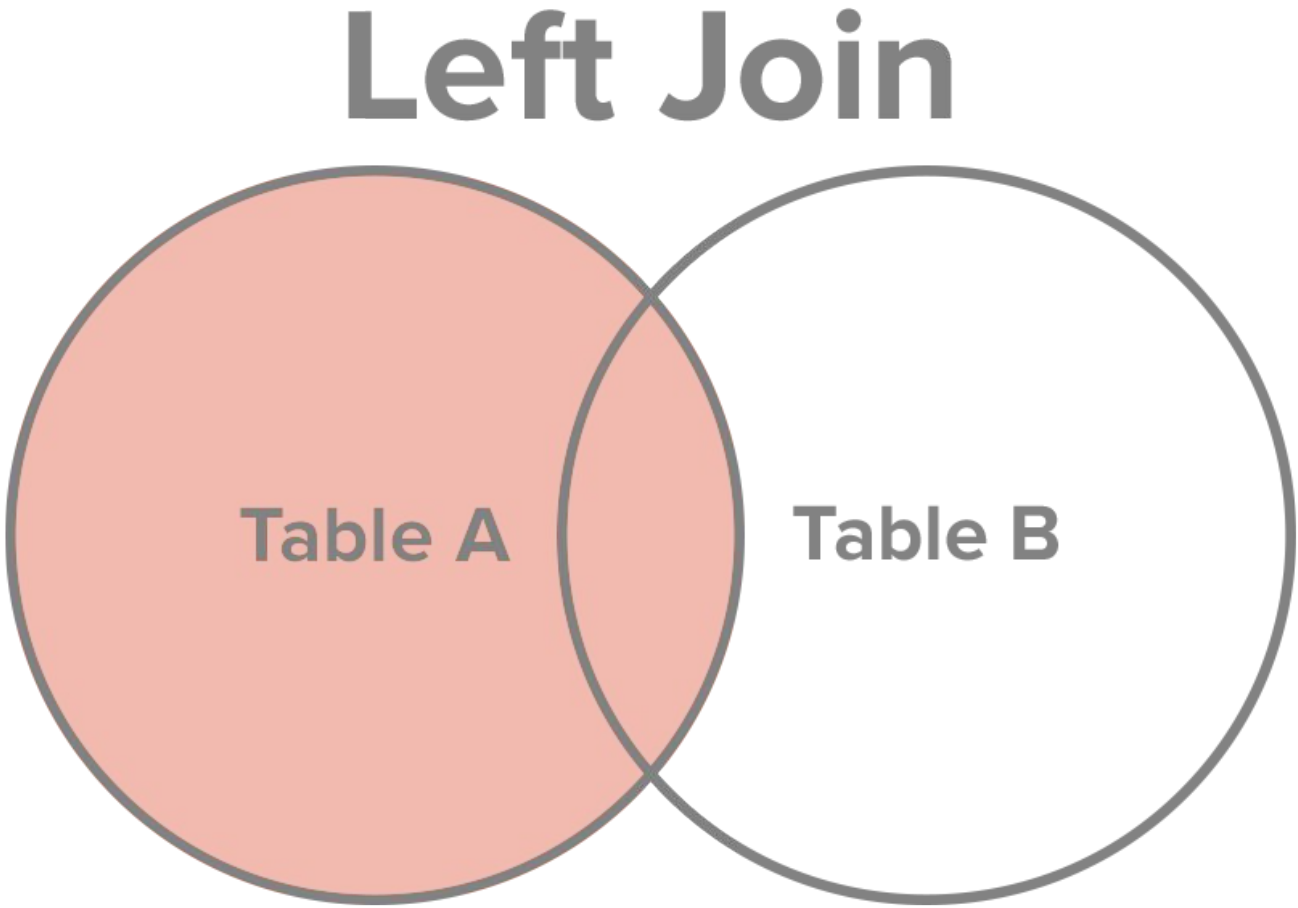
```
SELECT emp_name, dept_name

FROM Employee

INNER JOIN Department ON Employee.dept_id = Department.dept_id

WHERE emp_name LIKE 'A%';
```

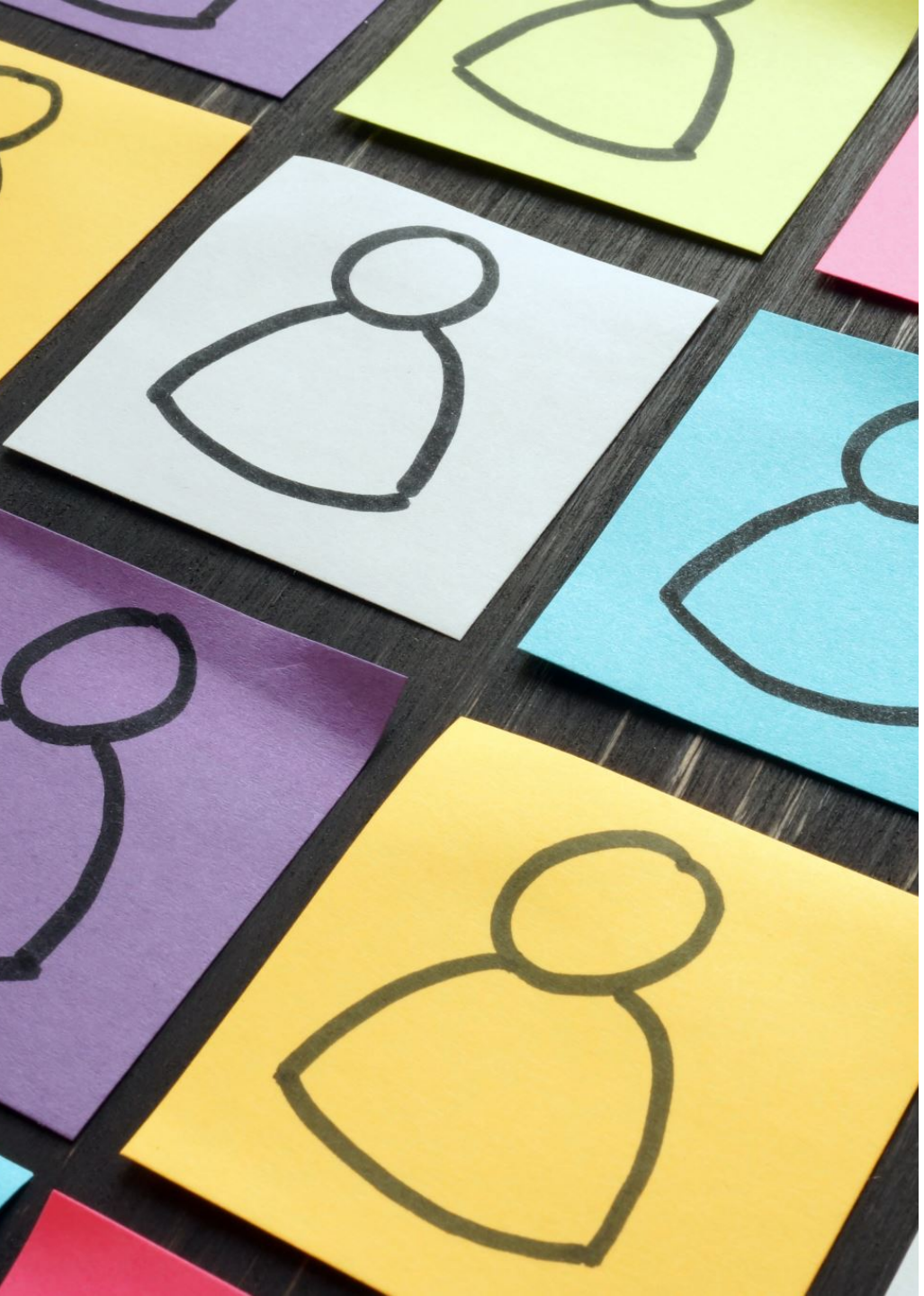
LEFT JOIN (or LEFT OUTER JOIN)

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

SYNTAX:

SELECT *column_name(s)*
FROM *table1*
LEFT JOIN *table2*
ON *table1.column_name = table2.colu*
mn_name;





LEFT JOIN (or LEFT OUTER JOIN) EXAMPLE

List all employee and department pairs

SELECT employees.emp_id, employees.emp_name, departments.dept_name

FROM employees

left JOIN departmentS ON employees.dept_id = departments.dept_id;

Employee-department pairs sorted by department name

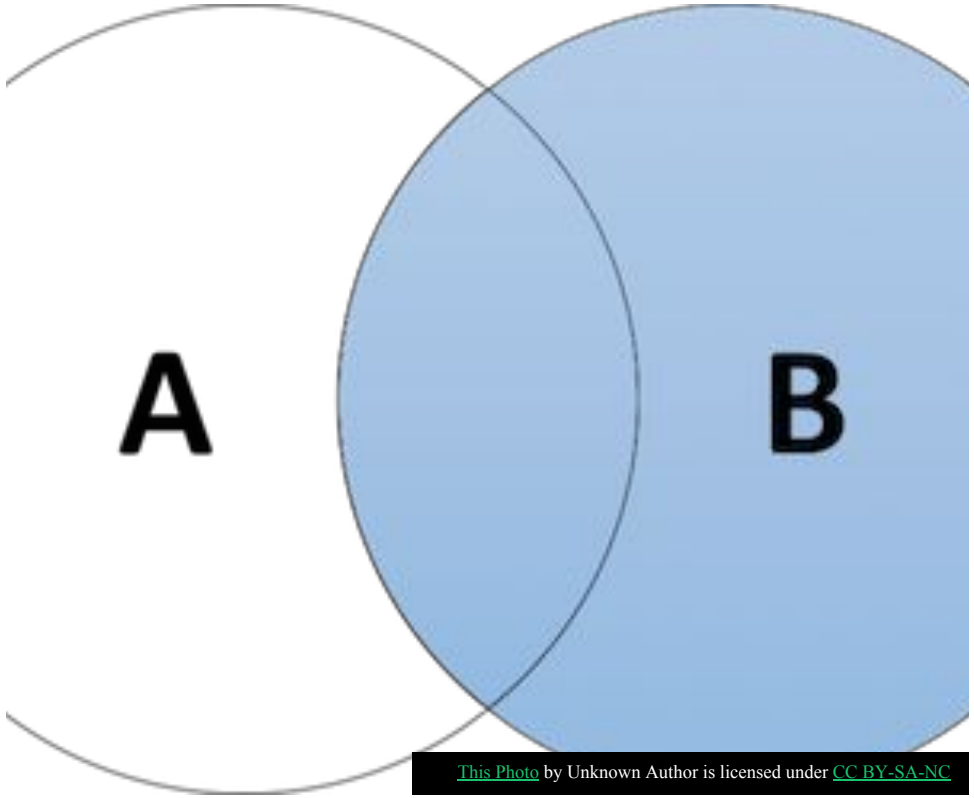
SELECT employees.emp_name, departmentS.dept_name

FROM EMPLOYEE

LEFT JOIN DEPARTMENT ON employees. dept_id = departments. dept_id

ORDER BY dept_name;

RIGHT JOIN (or RIGHT OUTER JOIN)



The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match..

SYNTAX:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

RIGHT JOIN (or RIGHT OUTER JOIN) EXAMPLE

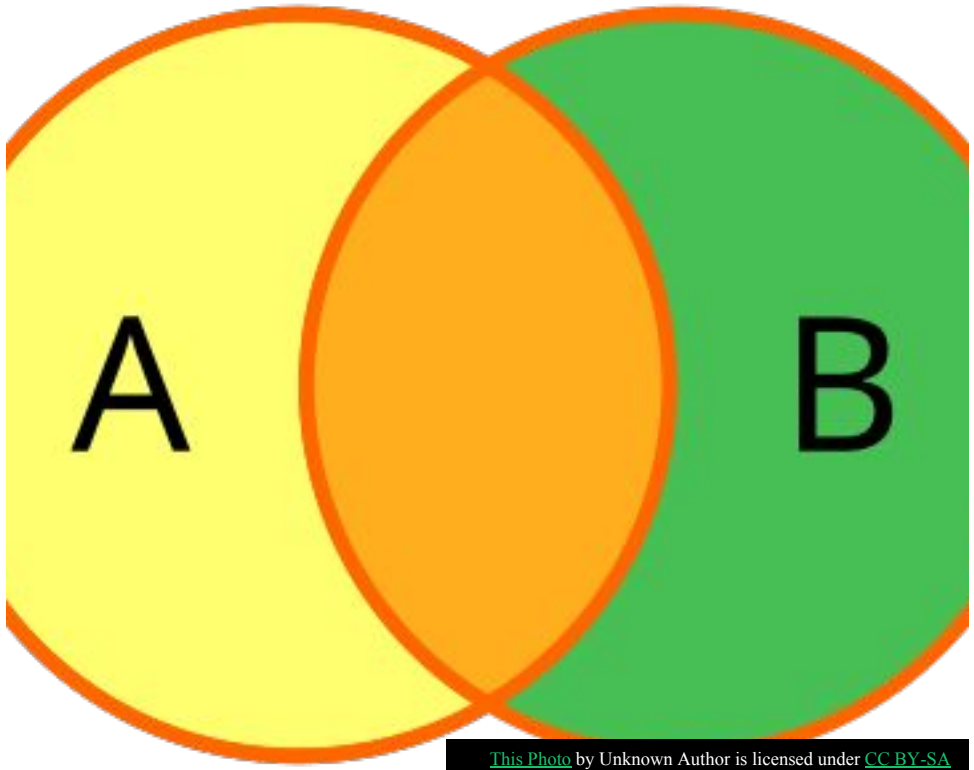
Basic RIGHT JOIN

```
SELECT emp_name, dept_name

FROM employees

RIGHT JOIN departments ON employees.dept_id
= departments.dept_id;
```

RIGHT JOIN with WHERE filter

```
SELECT emp_name, dept_name

FROM employees

RIGHT JOIN departments ON employees.dept_id
= departments.dept_id

WHERE dept_name = 'HR';
```

FULL JOIN (or FULL OUTER JOIN)



The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

SYNTAX:

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name =
table2.column_name
WHERE condition;
```

FULL JOIN (or FULL OUTER JOIN)

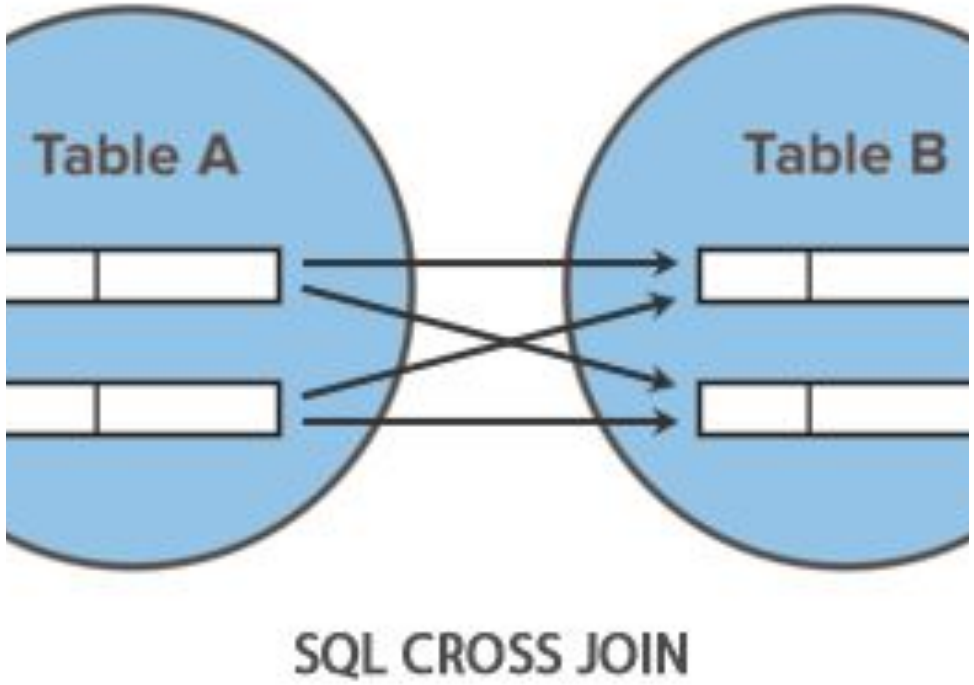
Basic FULL OUTER JOIN

```
SELECT *
FROM employees
FULL OUTER JOIN departments
ON employees.dept_id = departments. dept_id ;
```

List all employees and departments with employee names and department names:

```
SELECT emp_name, dept_name
FROM employees
FULL OUTER JOIN departments
ON employees.dept_id = departments. dept_id ;
```

CROSS JOIN



A CROSS JOIN is a type of SQL join that returns the Cartesian product of two tables.

SYNTAX:

```
SELECT column1, column2, ...
FROM table1
CROSS JOIN table2;
```

CROSS JOIN

Basic CROSS JOIN

SELECT * FROM EMPLOYEES CROSS JOIN DEPARTMENT;

Select specific columns

SELECT emp_name, dept_name FROM EMPLOYEES CROSS JOIN DEPARTMENT;

Use WHERE clause to filter specific departments

SELECT emp_name, dept_name

FROM EMPLOYEES CROSS JOIN DEPARTMENT

WHERE dept_name = 'HR';

LAB TASK

Use the SQL queries provided below to perform operations on the given table.

- Find the minimum marks for students aged between 16 and 18
- Find the minimum marks for each city where students are aged 17 or 18
- Find the minimum marks for students in Grade 10, grouped by city
- Find the maximum marks for each student in different cities
- Find the maximum marks scored by male students in Grade 12

Use the SQL queries provided below to perform operations on the given table.

- Count the number of students who scored between 80 and 90 marks
- Count the number of students from each city who scored below 75 marks
- Count the number of students who scored below 80 and group by Grades
- Count the number of students with a name starting with 'J' grouped by City
- Calculate the total marks for each gender

Use the SQL queries provided below to perform operations on the given table.

- Find the average marks of students who scored above 80
- Find the average marks of students from Seattle
- Count the number of students aged 17 who scored more than 80 marks
- Count the number of students who scored more than 90 marks and are in Grade 12