IT Department

Union and Joins

Soma Soleiman Zadeh

Database Systems I (IT 215)

Fall 2025 - 2026

Week 9

December 1, 2025

Outline

- **Union**
- Types of **Join**
  - **Cross Join**
  - **Inner Join**
  - **Outer Join**
    - ➢ Left Outer Join
    - ➢ Right Outer Join
    - ➢ Full Outer Join

Union

- The **UNION** operator is used to combine the data from the result of two or more SELECT command queries into a single distinct result set.
- The **SELECT** statements that **<u>are combined by Union operator</u>** must:
  - have the same number of fields,
  - have the same data types for each field,
  - fields are in the same order.

UNION vs. UNION ALL

- Union combines results of two or more SELECT queries and **<u>removes duplicated records</u>**.

```
SELECT    column_name(s)
FROM      table_name_1
UNION
SELECT    column_name(s)
FROM      table_name_2;
```

- Union All combines results of two or more SELECT queries, **<u>including duplicated records</u>**.

```
SELECT    column_name(s)
FROM      table_name_1
UNION ALL
SELECT    column_name(s)
FROM      table_name_2;
```

Example – Union Operator

```
SELECT *
FROM facebook

UNION

SELECT *
FROM linkedin;
```

facebook

Name	Location
Alex	San Francisco
Matt	San Francisco
Zeke	Los Angeles

linkedin

Name	Location
Matt	San Francisco
Ruby	San Francisco
Zeke	Los Angeles

Concatenated

Name	Location
Alex	San Francisco
Matt	San Francisco
Matt	San Francisco
Ruby	San Francisco
Zeke	Los Angeles
Zeke	Los Angeles

Duplicates Removed

Name	Location
Alex	San Francisco
Matt	San Francisco
Ruby	San Francisco
Zeke	Los Angeles

Example – Union All Operator

```
SELECT *
FROM facebook

UNION ALL

SELECT *
FROM linkedin;
```

facebook

Name	Location
Alex	San Francisco
Matt	San Francisco
Zeke	Los Angeles

linkedin

Name	Location
Matt	San Francisco
Ruby	San Francisco
Zeke	Los Angeles

Result

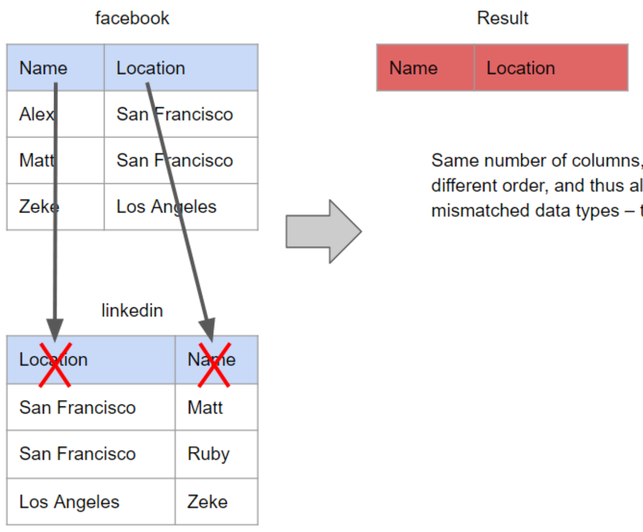
Name	Location
Alex	San Francisco
Matt	San Francisco
Matt	San Francisco
Ruby	San Francisco
Zeke	Los Angeles
Zeke	Los Angeles

Union Requirements Example

```
SELECT
Name,
Location
FROM facebook

UNION

SELECT
Location,
Name
FROM linkedin;
```



facebook

Name	Location
Alex	San Francisco
Matt	San Francisco
Zeke	Los Angeles

linkedin

Location	Name
San Francisco	Matt
San Francisco	Ruby
Los Angeles	Zeke

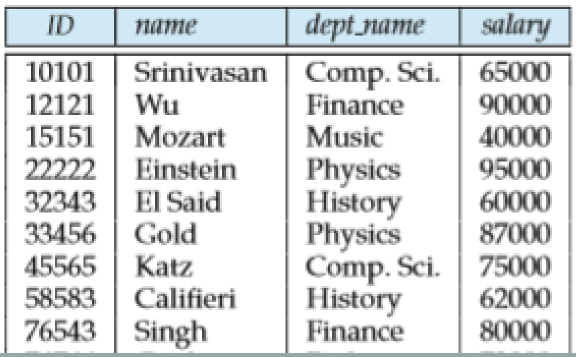
Result

Name	Location

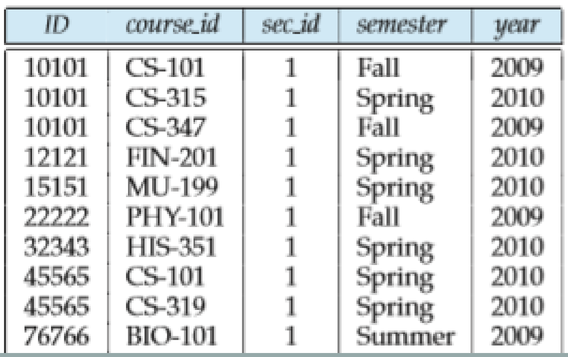
Same number of columns, *but* in a different order, and thus also having mismatched data types – this will not work!

Join of Multiple Tables

○ Sometimes the desired information can be retrieved from a single table, but usually you need to get the desired data from more than one table. In such cases, we join tables to get the desired data.



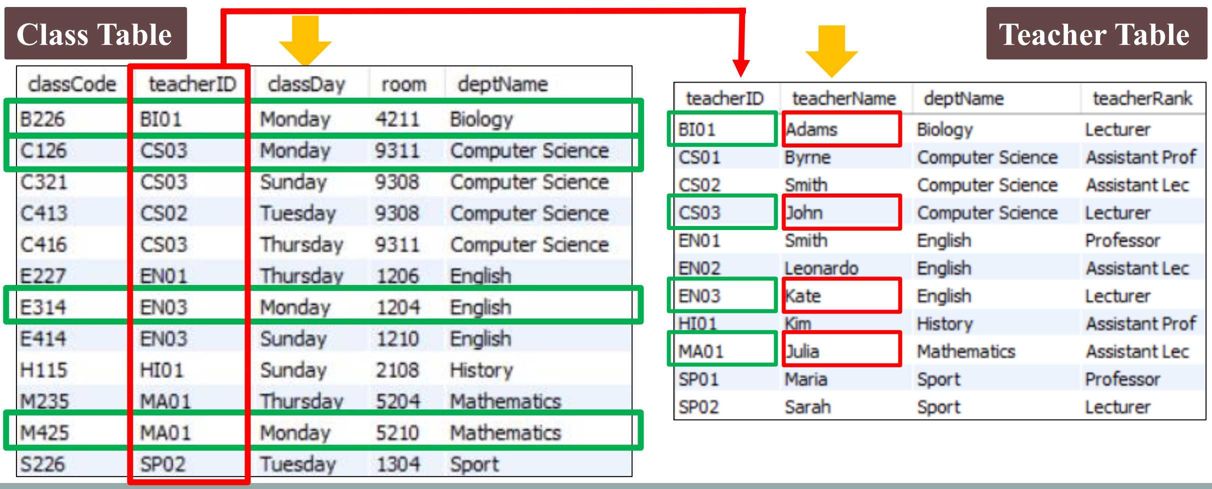
ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000



ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009

SELECT on multiple tables (Necessity of JOIN)

○ Find the **names** of teachers who teach a class on **Monday**.



Class Table

classCode	teacherID	classDay	room	deptName
B226	BI01	Monday	4211	Biology
C126	CS03	Monday	9311	Computer Science
C321	CS03	Sunday	9308	Computer Science
C413	CS02	Tuesday	9308	Computer Science
C416	CS03	Thursday	9311	Computer Science
E227	EN01	Thursday	1206	English
E314	EN03	Monday	1204	English
E414	EN03	Sunday	1210	English
H115	HI01	Sunday	2108	History
M235	MA01	Thursday	5204	Mathematics
M425	MA01	Monday	5210	Mathematics
S226	SP02	Tuesday	1304	Sport

Teacher Table

teacherID	teacherName	deptName	teacherRank
BI01	Adams	Biology	Lecturer
CS01	Byrne	Computer Science	Assistant Prof
CS02	Smith	Computer Science	Assistant Lec
CS03	John	Computer Science	Lecturer
EN01	Smith	English	Professor
EN02	Leonardo	English	Assistant Lec
EN03	Kate	English	Lecturer
HI01	Kim	History	Assistant Prof
MA01	Julia	Mathematics	Assistant Lec
SP01	Maria	Sport	Professor
SP02	Sarah	Sport	Lecturer

Joins

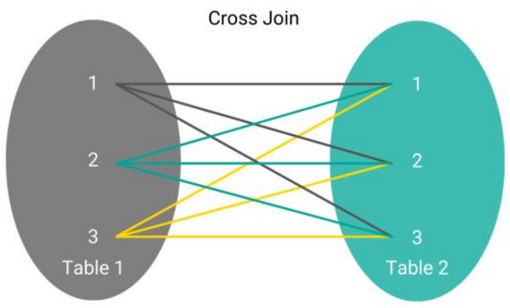
○ **JOIN** clause is used to combine records from two or more tables.

○ Three different types of Joins:

- ○ **Cross** Join (Cartesian Product)
- ○ **Inner** Join
- ○ **Outer** Join
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join

Cross Join (Cartesian Product)

- The **CROSS JOIN (Cartesian Product)** is used to generate a paired combination of each row of the first table with each row of the second table.



Cross Join (Cartesian Product)

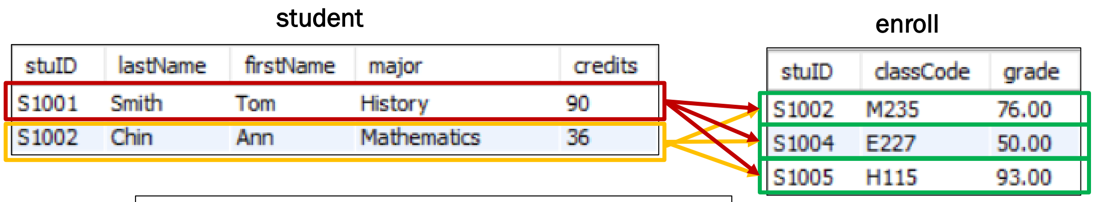
- SQL Syntax of Cross Join

```
SELECT   A1, A2, ... , An
FROM     r1, r2;
```

- **Example:** the Cartesian product of the relations **Student** and **Enroll**

(all rows of the **Student** table are joined to all rows of the **Enroll** table)

Cross Join (Cartesian Product)



student

stuID	lastName	firstName	major	credits
S1001	Smith	Tom	History	90
S1002	Chin	Ann	Mathematics	36

enroll

stuID	classCode	grade
S1002	M235	76.00
S1004	E227	50.00
S1005	H115	93.00

```
SELECT   S.* , E.*
FROM     Student AS S , Enroll AS E;
```

Student.stuId	lastName	firstName	major	credits	Enroll.stuId	classCode	grade
S1001	Smith	Tom	History	90	S1002	M235	76.00
S1001	Smith	Tom	History	90	S1004	E227	50.00
S1001	Smith	Tom	History	90	S1005	H115	93.00
S1002	Chin	Ann	Mathematics	36	S1002	M235	76.00
S1002	Chin	Ann	Mathematics	36	S1004	E227	50.00
S1002	Chin	Ann	Mathematics	36	S1005	H115	93.00

Cross Join (Cartesian Product)

- **Example –** If all the students have taken all the courses, then retrieve **full name** and **course name** of all of them.

Student

StuID	FirstName	LastName	Stage
1	Lee	Perry	2
2	Roberts	Mike	4
3	Jones	Mary	3

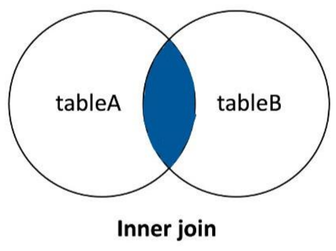
Course

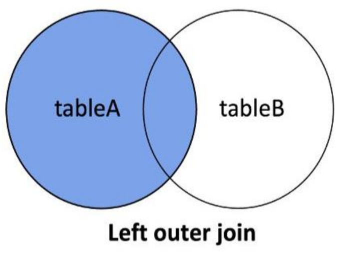
CourseCode	CName	Stage
DB1	Database	2
AI	Artificial Intelligence	4

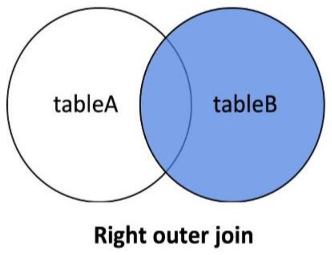
```
SELECT   FirstName, LastName, CName
FROM     Student , Course;
```

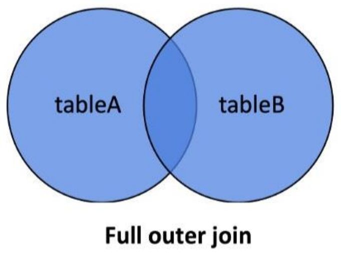
FirstName	LastName	CName
Lee	Perry	Database
Lee	Perry	Artificial Intelligence
Roberts	Mike	Database
Roberts	Mike	Artificial Intelligence
Jones	Mary	Database
Jones	Mary	Artificial Intelligence

Inner Join and Outer Join





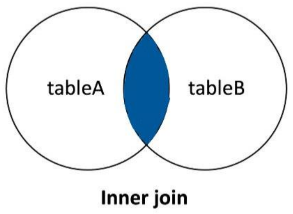




Inner Join

- **INNER JOIN** returns records that have the same value in matching columns of the two tables.
- The **INNER JOIN** is the most commonly-used SQL JOIN.
- SQL Syntax of **Inner Join:**

(Both these syntaxes are accepted; you can follow only one style.)



```
SELECT   A1, A2, … , An
FROM     r1  INNER JOIN  r2
         ON   r1.Foreign_key = r2.Primary_key;
```

OR

```
SELECT   A1, A2, … , An
FROM     r1, r2
WHERE    r1. Foreign_key = r2. Primary_key;
```

Inner Join

○ **Example –** If we want to show **name** of teachers who teach at least a course, and the **name** of courses they teach.

Teacher

TID	TName	Rank
11	Lee	Professor
45	Roberts	Lecturer
15	Jones	Lecturer

Course

CourseID	CName	TID
DB	Database	15
AI	Artificial Intelligence	15
NT	Network	11

```
SELECT   TName, CName
FROM     Teacher, Course
WHERE    Teacher.TID = Course.TID;
```

Query result

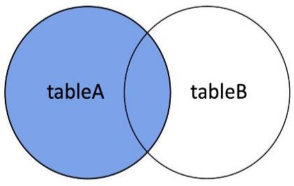
TName	CName
Lee	Network
Jones	Database
Jones	Artificial Intelligence

Outer Join

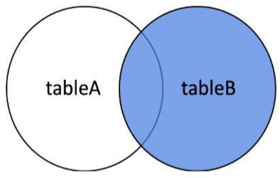
○ **OUTER JOIN** returns all rows from at least one of the tables.

○ Three types of **OUTER JOIN**:

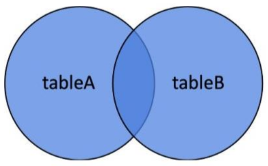
- **Left** Outer Join
- **Right** Outer Join
- **Full** Outer Join



Left outer join



Right outer join

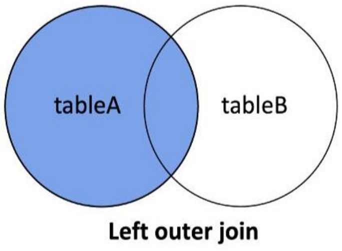


Full outer join

Left Join

- **LEFT JOIN** returns all the rows from the left table (table 1) and the matching records from the right table (table 2) are included.
- If there is no match in the right table, it fills NULL values in the columns of the right table.
- SQL Syntax of **LEFT OUTER JOIN**:

```
SELECT   A1, A2, ... , An
FROM      r1  LEFT JOIN  r2
              ON   r1.Foreign_key = r2.Primary_key;
```



Left Join

- **Example –** If we want to show **name of all teachers**, and the **name of courses** they teach, even if a teacher doesn't teach any course.

Teacher

TID	TName	Rank
11	Lee	Professor
45	Roberts	Lecturer
15	Jones	Lecturer

Course

CourseID	CName	TID
DB	Database	15
AI	Artificial Intelligence	15
NT	Network	11

```
SELECT   TName, CName
FROM      Teacher  LEFT JOIN Course
       ON     Teacher.TID = Course.TID;
```

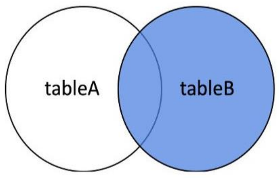
Query result

TName	CName
Lee	Network
Roberts	*NULL*
Jones	Database
Jones	Artificial Intelligence

Right Join

- **RIGHT JOIN** returns all the rows from the right table (table 2) and the matching records from the left table (table1) are included.
- If there is no match in the left table, it fills NULL values in the columns of the left table.
- SQL Syntax of **RIGHT OUTER JOIN**:

```
SELECT   A1, A2, ... , An
FROM      r1  RIGHT JOIN  r2
          ON   r1.Foreign_key = r2.Primary_key;
```



Right outer join

Right Join

- **Output of the following query?**

Teacher

TID	TName	Rank
11	Lee	Professor
45	Roberts	Lecturer
15	Jones	Lecturer

Course

CourseID	CName	TID
DB	Database	15
AI	Artificial Intelligence	15
NT	Network	11

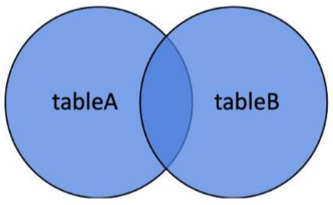
```
SELECT   Cname, Tname, Rank
FROM     Teacher  RIGHT JOIN Course
         ON    Teacher.TID = Course.TID;
```

Query result

CName	TName	Rank
Database	Jones	Lecturer
Artificial Intelligence	Jones	Lecturer
Network	Lee	Professor

Full Outer Join

- **FULL OUTER JOIN** returns all records from both tables, including matching and non-matching records.
- If no matching records exist in one or both tables, NULL values are included for those columns.



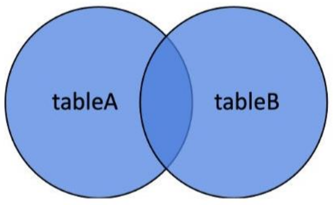
Full outer join

Full Outer Join

```
SELECT   A1, A2, … , An
FROM     r1  LEFT JOIN  r2
         ON   r1.Foreign_key = r2.Primary_key

UNION

SELECT   A1, A2, … , An
FROM     r1  RIGHT JOIN  r2
         ON   r1.Foreign_key = r2.Primary_key
;
```



Full outer join

Full Outer Join

- Output of the following query?

Teacher

TID	TName	Rank
11	Lee	Professor
45	Roberts	Lecturer
15	Jones	Lecturer

Course

CourseID	CName	TID
DB	Database	15
OS	Operating System	22
NT	Network	11

```
SELECT   TName, CName
FROM     Teacher  LEFT JOIN Course
         ON    Teacher.TID = Course.TID

UNION

SELECT   TName, CName
FROM     Teacher  RIGHT JOIN Course
         ON    Teacher.TID = Course.TID;
```

Query result

TName	CName
Lee	Network
Roberts	*NULL*
Jones	Database
NULL	Operating System