DBMS

Lab-6

Things we will complete today

- SQL Join Expressions
- User management and Authorization






SQL Join operations

The Natural Join

- To make the life of an SQL programmer easier for this common case, SQL supports an operation called the **natural join**

```
select name, course_id
from instructor, teaches
where instructor.ID= teaches.ID;
```

```
select name, course_id
from instructor natural join teaches;
```

```
select A1, A2, . . . , An
from r1 natural join r2 natural join . . . natural join rm
where P;
```

The Natural Join

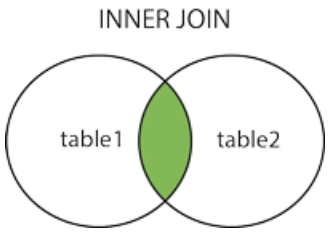
- To make the life of an SQL programmer easier for this common case, SQL supports an operation called the **natural join**

```
SELECT
    `name`, teaches.course_id, course.title, `year`
FROM
    instructor, teaches, course
WHERE
    instructor.ID = teaches.ID AND teaches.course_id = course.course_id
```

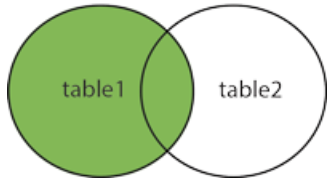
```
SELECT
    `name`, teaches.course_id, course.title, `year`
FROM
    FROM instructor NATURAL JOIN  teaches NATURAL JOIN course
```

Other Join expressions

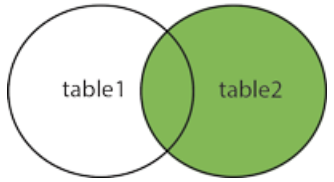
- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table



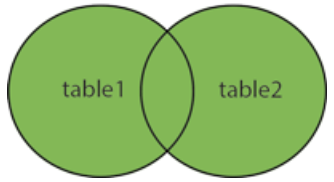
INNER JOIN



LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN

04-Dec-21 10:42 AM

The INNER JOIN Keyword

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

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

```sql
SELECT
    column_name(s)
FROM
    table1 , table2
WHERE
    table1.column_name = table2.column_name;
```

04-Dec-21 10:42 AM

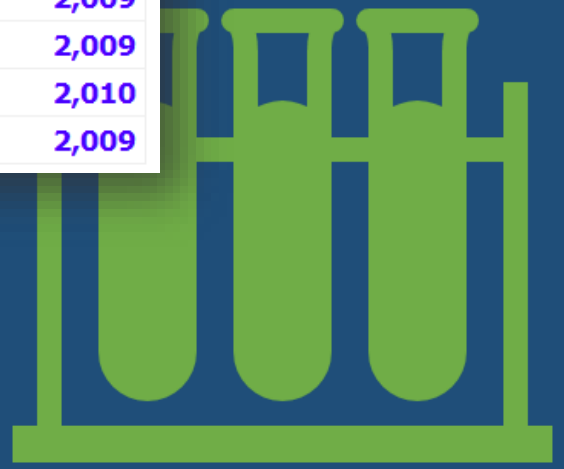
The INNER JOIN Keyword

university_db.instructor: 12 rows total

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	72,000.00
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00

university_db.teaches: 15 rows total

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2,009
10101	CS-315	1	Spring	2,010
10101	CS-347	1	Fall	2,009
12121	FIN-201	1	Spring	2,010
15151	MU-199	1	Spring	2,010
22222	PHY-101	1	Fall	2,009
32343	HIS-351	1	Spring	2,010
45565	CS-101	1	Spring	2,010
45565	CS-319	1	Spring	2,010
76766	BIO-101	1	Summer	2,009
76766	BIO-301	1	Summer	2,010
83821	CS-190	1	Spring	2,009
83821	CS-190	2	Spring	2,009
83821	CS-319	2	Spring	2,010
98345	EE-181	1	Spring	2,009



The INNER JOIN Keyword

```sql
SELECT
   instructor.id, instructor.name, teaches.*
FROM
   instructor INNER JOIN teaches
ON
   instructor.ID= teaches.ID
```

Result #1 (7×15)

id	name	ID	course_id	sec_id	semester	year
10101	Srinivasan	10101	CS-101	1	Fall	2,009
10101	Srinivasan	10101	CS-315	1	Spring	2,010
10101	Srinivasan	10101	CS-347	1	Fall	2,009
12121	Wu	12121	FIN-201	1	Spring	2,010
15151	Mozart	15151	MU-199	1	Spring	2,010
22222	Einstein	22222	PHY-101	1	Fall	2,009
32343	El Said	32343	HIS-351	1	Spring	2,010
45565	Katz	45565	CS-101	1	Spring	2,010
45565	Katz	45565	CS-319	1	Spring	2,010
76766	Crick	76766	BIO-101	1	Summer	2,009
76766	Crick	76766	BIO-301	1	Summer	2,010
83821	Brandt	83821	CS-190	1	Spring	2,009
83821	Brandt	83821	CS-190	2	Spring	2,009
83821	Brandt	83821	CS-319	2	Spring	2,010
98345	Kim	98345	EE-181	1	Spring	2,009

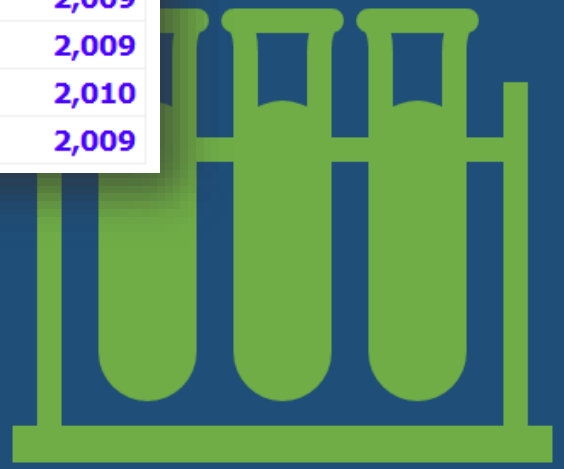
The LEFT JOIN Keyword

university_db.instructor: 12 rows total

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	72,000.00
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00

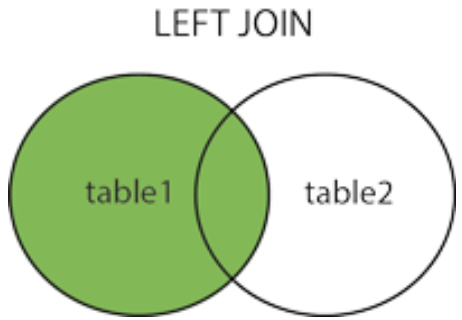
university_db.teaches: 15 rows total

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2,009
10101	CS-315	1	Spring	2,010
10101	CS-347	1	Fall	2,009
12121	FIN-201	1	Spring	2,010
15151	MU-199	1	Spring	2,010
22222	PHY-101	1	Fall	2,009
32343	HIS-351	1	Spring	2,010
45565	CS-101	1	Spring	2,010
45565	CS-319	1	Spring	2,010
76766	BIO-101	1	Summer	2,009
76766	BIO-301	1	Summer	2,010
83821	CS-190	1	Spring	2,009
83821	CS-190	2	Spring	2,009
83821	CS-319	2	Spring	2,010
98345	EE-181	1	Spring	2,009



The LEFT JOIN Keyword

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




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

The LEFT JOIN Keyword

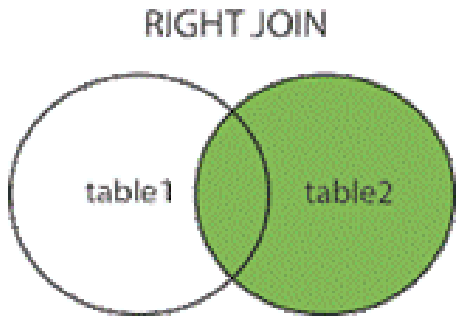
```sql
SELECT
    instructor.id, instructor.name,instructor.dept_name, teaches.*
FROM
    instructor LEFT JOIN teaches
ON
    instructor.ID= teaches.ID
```

Result #1 (8×18)

id	name	dept_name	ID	course_id	sec_id	semester	year
10101	Srinivasan	Comp. Sci.	10101	CS-101	1	Fall	2,009
10101	Srinivasan	Comp. Sci.	10101	CS-315	1	Spring	2,010
10101	Srinivasan	Comp. Sci.	10101	CS-347	1	Fall	2,009
12121	Wu	Finance	12121	FIN-201	1	Spring	2,010
15151	Mozart	Music	15151	MU-199	1	Spring	2,010
22222	Einstein	Physics	22222	PHY-101	1	Fall	2,009
32343	El Said	History	32343	HIS-351	1	Spring	2,010
33456	Gold	Physics	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
45565	Katz	Comp. Sci.	45565	CS-101	1	Spring	2,010
45565	Katz	Comp. Sci.	45565	CS-319	1	Spring	2,010
58583	Califieri	History	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
76543	Singh	Finance	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
76766	Crick	Biology	76766	BIO-101	1	Summer	2,009
76766	Crick	Biology	76766	BIO-301	1	Summer	2,010
83821	Brandt	Comp. Sci.	83821	CS-190	1	Spring	2,009
83821	Brandt	Comp. Sci.	83821	CS-190	2	Spring	2,009
83821	Brandt	Comp. Sci.	83821	CS-319	2	Spring	2,010
98345	Kim	Elec. Eng.	98345	EE-181	1	Spring	2,009

The RIGHT JOIN Keyword

- RIGHT (OUTER) JOIN Returns all records from the right table, and the matched records from the left table




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

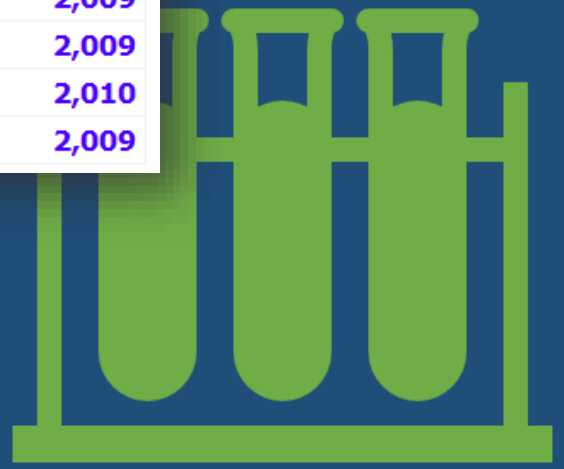
The INNER JOIN Keyword

university_db.instructor: 12 rows total

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	72,000.00
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00

university_db.teaches: 15 rows total

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2,009
10101	CS-315	1	Spring	2,010
10101	CS-347	1	Fall	2,009
12121	FIN-201	1	Spring	2,010
15151	MU-199	1	Spring	2,010
22222	PHY-101	1	Fall	2,009
32343	HIS-351	1	Spring	2,010
45565	CS-101	1	Spring	2,010
45565	CS-319	1	Spring	2,010
76766	BIO-101	1	Summer	2,009
76766	BIO-301	1	Summer	2,010
83821	CS-190	1	Spring	2,009
83821	CS-190	2	Spring	2,009
83821	CS-319	2	Spring	2,010
98345	EE-181	1	Spring	2,009



The RIGHT JOIN Keyword

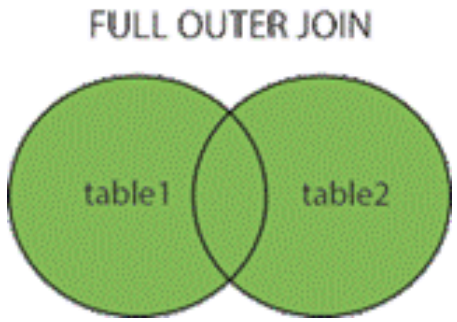
```
1 SELECT
2    instructor.id, instructor.name,instructor.dept_name, teaches.*
3 FROM
4    instructor RIGHT JOIN teaches
5 ON
6    instructor.ID= teaches.ID
7
```

Result #1 (8×15)

id	name	dept_name	ID	course_id	sec_id	semester	year
(NULL)	(NULL)	(NULL)	10101	CS-101	1	Fall	2,009
(NULL)	(NULL)	(NULL)	10101	CS-315	1	Spring	2,010
(NULL)	(NULL)	(NULL)	10101	CS-347	1	Fall	2,009
12121	Wu	Finance	12121	FIN-201	1	Spring	2,010
15151	Mozart	Music	15151	MU-199	1	Spring	2,010
22222	Einstein	Physics	22222	PHY-101	1	Fall	2,009
32343	El Said	History	32343	HIS-351	1	Spring	2,010
45565	Katz	Comp. Sci.	45565	CS-101	1	Spring	2,010
45565	Katz	Comp. Sci.	45565	CS-319	1	Spring	2,010
76766	Crick	Biology	76766	BIO-101	1	Summer	2,009
76766	Crick	Biology	76766	BIO-301	1	Summer	2,010
83821	Brandt	Comp. Sci.	83821	CS-190	1	Spring	2,009
83821	Brandt	Comp. Sci.	83821	CS-190	2	Spring	2,009
83821	Brandt	Comp. Sci.	83821	CS-319	2	Spring	2,010
98345	Kim	Elec. Eng.	98345	EE-181	1	Spring	2,009

The FULL JOIN Keyword

- FULL (OUTER) JOIN Returns all records when there is a match in either left or right table




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

Supported in MS SQL Server, Oracle
Not supported in MySQL.

The FULL JOIN Alternative in MySQL

```sql
SELECT instructor.id, instructor.name,instructor.dept_name, teaches.*
FROM  instructor LEFT JOIN teaches
ON    instructor.ID= teaches.ID

UNION

SELECT instructor.id, instructor.name,instructor.dept_name, teaches.*
FROM  instructor RIGHT JOIN teaches
ON instructor.ID= teaches.ID
```

Result #1 (8×18)

id	name	dept_name	ID	course_id	sec_id	semester	year
(NULL)	(NULL)	(NULL)	10101	CS-101	1	Fall	2,009
(NULL)	(NULL)	(NULL)	10101	CS-315	1	Spring	2,010
(NULL)	(NULL)	(NULL)	10101	CS-347	1	Fall	2,009
12121	Wu	Finance	12121	FIN-201	1	Spring	2,010
15151	Mozart	Music	15151	MU-199	1	Spring	2,010
22222	Einstein	Physics	22222	PHY-101	1	Fall	2,009
32343	El Said	History	32343	HIS-351	1	Spring	2,010
33456	Gold	Physics	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
45565	Katz	Comp. Sci.	45565	CS-101	1	Spring	2,010
45565	Katz	Comp. Sci.	45565	CS-319	1	Spring	2,010
58583	Califieri	History	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
76543	Singh	Finance	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
76766	Crick	Biology	76766	BIO-101	1	Summer	2,009
76766	Crick	Biology	76766	BIO-301	1	Summer	2,010
83821	Brandt	Comp. Sci.	83821	CS-190	1	Spring	2,009
83821	Brandt	Comp. Sci.	83821	CS-190	2	Spring	2,009
83821	Brandt	Comp. Sci.	83821	CS-319	2	Spring	2,010
98345	Kim	Elec. Eng.	98345	EE-181	1	Spring	2,009

User Management and Privileges

Create a MySQL User

```
mysql> CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'user_password';
```

04-Dec-21 10:42 AM

Grant Privileges to a MySQL User Account

- There are multiple types of privileges that can be granted to a user account. The most commonly used privileges are:

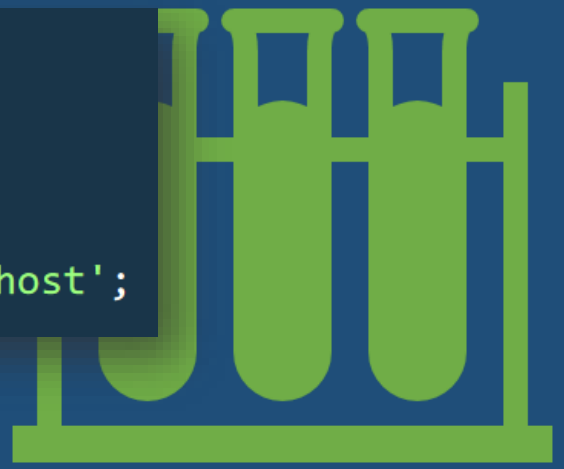
  - `ALL PRIVILEGES` – Grants all privileges to a user account.
  - `CREATE` – The user account is allowed to create databases and tables.
  - `DROP` - The user account is allowed to drop databases and tables.
  - `DELETE` - The user account is allowed to delete rows from a specific table.
  - `INSERT` - The user account is allowed to insert rows into a specific table.
  - `SELECT` – The user account is allowed to read a database.
  - `UPDATE` - The user account is allowed to update table rows.

Grant Privileges to a MySQL User Account

```
GRANT
    permission1, permission2
ON
    database_name.table_name
TO
    'database_user'@'hostname';
```

```
GRANT
    ALL PRIVILEGES
ON
    database_name.*
TO
    'database_user'@'localhost';
```

```
GRANT
    SELECT, INSERT, UPDATE
ON
    database_name.*
TO
    'database_user'@'localhost';
```



04-Dec-21 10:42 AM

Demo.

End.