</talentlabs>

CHAPTER 8

SQL Joining

</talentlabs>

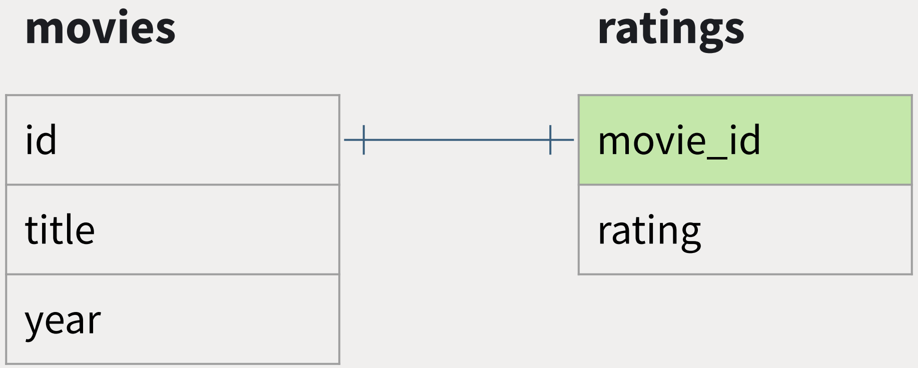
AGENDA

- What is Joining
- Types of Joins
  - Inner Join
  - Left Join & Right Join
  - Full Outer Join

What is Joining?

Joining

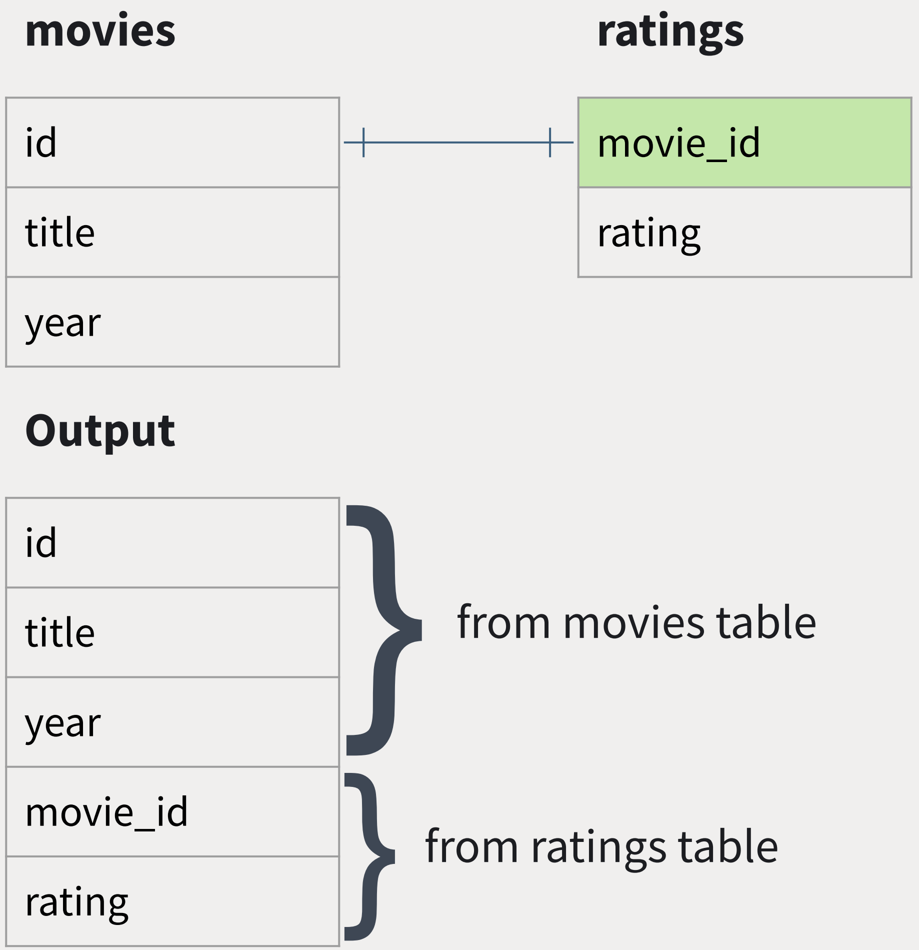
- Given there are relationship(s) between different tables, we can blend the two (or more) tables using JOIN keyword
- In general, joining can provide a more complete picture of a set of data e.g. movie data + rating data
- Let's see how we write the query to link the tables together and return the result of the joined tables



Quick Example on Joining

- Revisiting the relationship between movies and ratings, the primary key of movies table is “id”. The foreign key on ratings table is “movie_id”
- One-to-one relationships between movies and ratings

```
SELECT *
FROM
    movies JOIN ratings
        ON movies.id = ratings.movie_id
```



movies

id
title
year

ratings

movie_id
rating

Output

id
title
year
from movies table

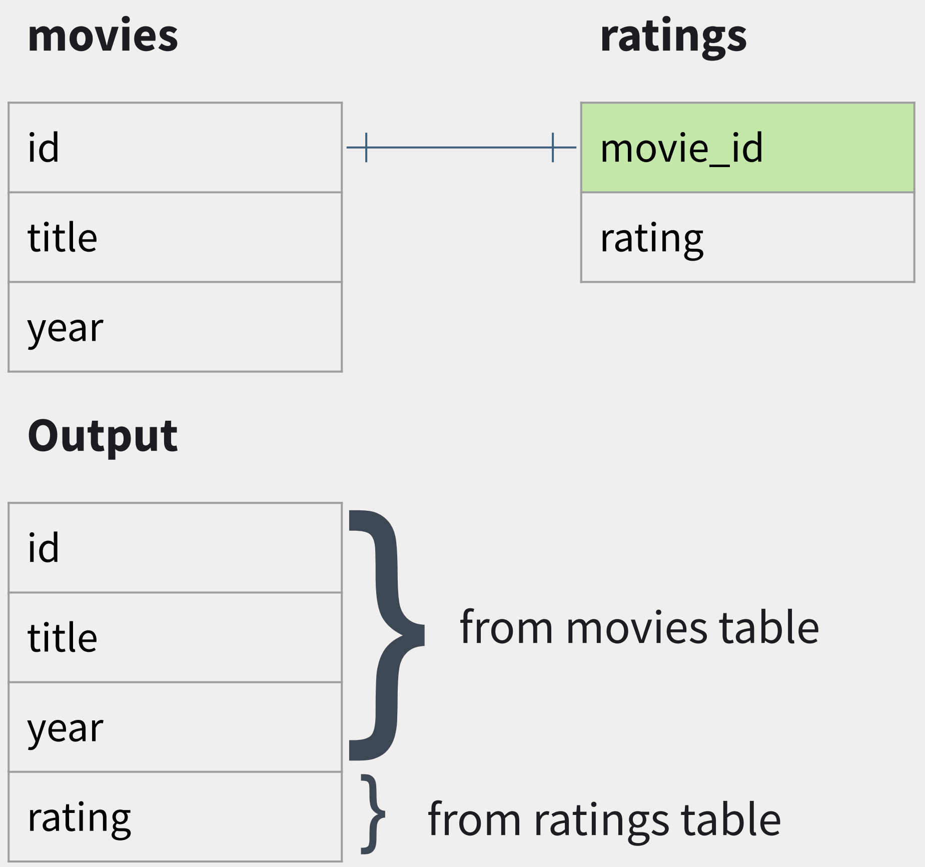
movie_id
rating
from ratings table

</talentlabs>

Quick Example on Joining

- Revisiting the relationship between movies and ratings, the primary key of movies table is “id”. The foreign key on ratings table is “movie_id”
- One-to-one relationships between movies and ratings

```
SELECT id, title, year, rating
FROM
    movies JOIN ratings
        ON movies.id = ratings.movie_id
```



movies

id

title

year

ratings

movie_id

rating

Output

id

title

year

rating

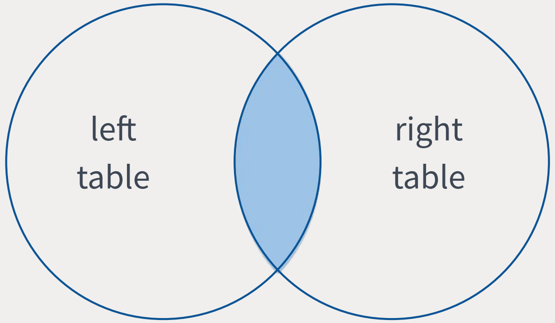
from movies table

from ratings table

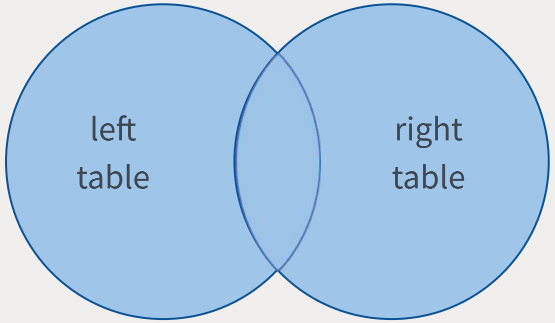
Types of Joining

Overview on Types of Joining

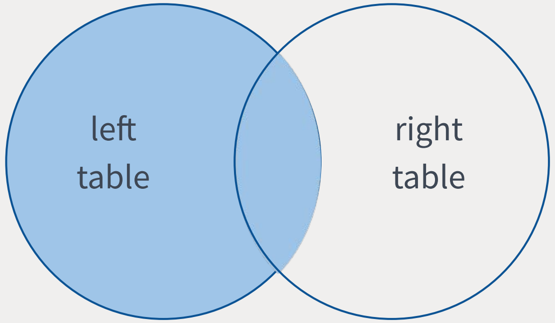
INNER JOIN



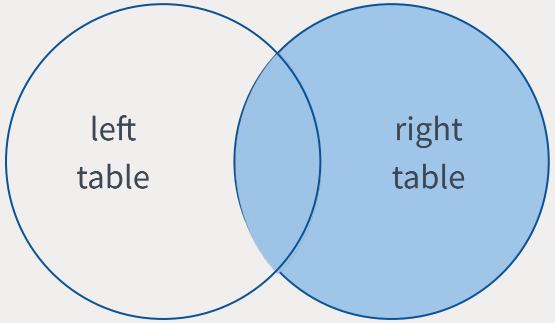
FULL JOIN / FULL OUTER JOIN



LEFT JOIN

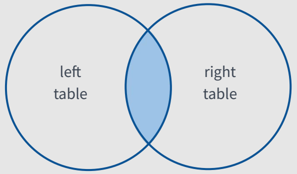


RIGHT JOIN



Inner Join

</talentlabs>



Inner Join

movies

id	title
1	Star Wars
2	Harry Potter
3	Toy Story
4	Up

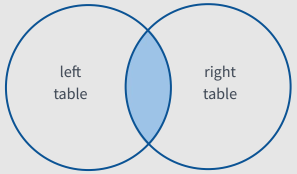
ratings

movie_id	rating
1	9.1
2	3.2
3	6.5
4	7.8

Joining Result

id	title	rating
1	Star Wars	9.1
2	Harry Potter	3.2
3	Toy Story	6.5
4	Up	7.8

```
SELECT id, title, rating
FROM
    movies JOIN ratings
        ON movies.id = ratings.movie_id
```



Inner Join

movies

id	title
1	Star Wars
2	Harry Potter
3	Toy Story
4	Up

ratings

movie_id	rating
2	3.2
3	6.5

Joining Result

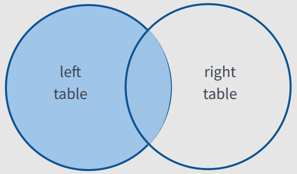
id	title	rating
2	Harry Potter	3.2
3	Toy Story	6.5

Note the joining result change when the **ratings** table (right table) has less data than **movies** table (left table)

```
SELECT id, title, rating
FROM
    movies JOIN ratings
        ON movies.id = ratings.movie_id
```

Left Join & Right Join

</talentlabs>



Left Join

movies

id	title
1	Star Wars
2	Harry Potter
3	Toy Story
4	Up

ratings

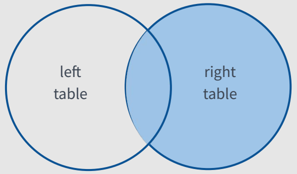
movie_id	rating
2	3.2
4	7.8

Joining Result

id	title	rating
1	Star Wars	
2	Harry Potter	3.2
3	Toy Story	
4	Up	7.8

```
SELECT id, title, rating
FROM
    movies LEFT JOIN ratings
        ON movies.id = ratings.movie_id
```

Not Available in SQLite



Right Join

movies

id	title
1	Star Wars
2	Harry Potter
3	Toy Story
4	Up

ratings

movie_id	rating
2	3.2
4	7.8
5	1.1

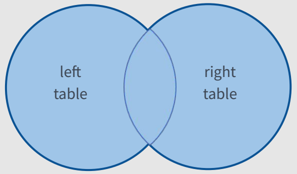
Joining Result

id	title	movie_id	rating
2	Harry Potter	2	3.2
4	Up	4	7.8
		5	1.1

```
SELECT id, title, movie_id, rating
FROM
    movies RIGHT JOIN ratings
        ON movies.id = ratings.movie_id
```

Full Outer Join

Not Available in SQLite



Full Outer Join

movies

id	title
1	Star Wars
2	Harry Potter
3	Toy Story
4	Up

ratings

movie_id	rating
2	3.2
4	7.8
5	1.1

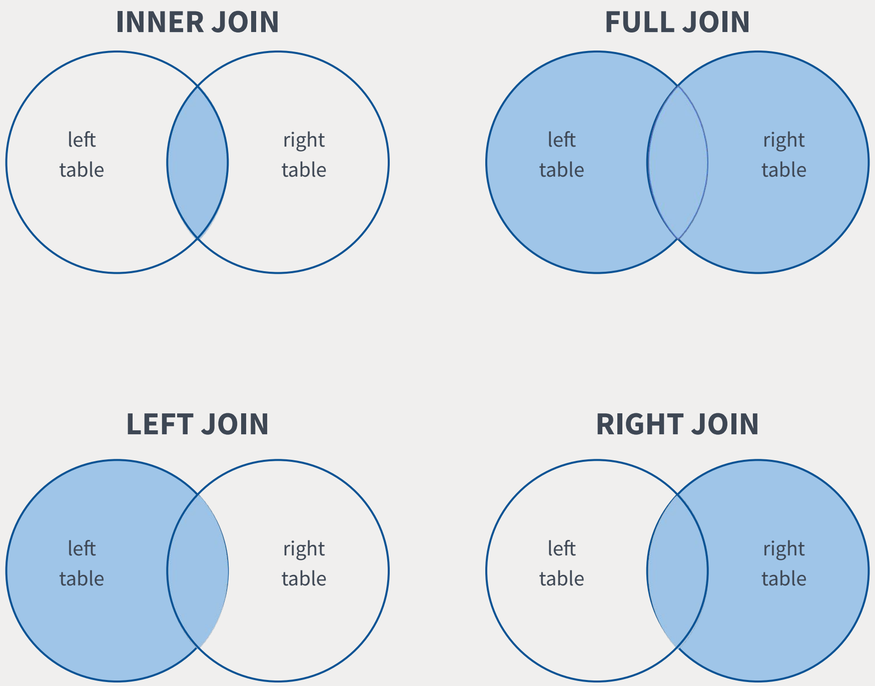
Joining Result

id	title	movie_id	rating
1	Star Wars		
2	Harry Potter	2	3.2
3	Toy Story		
4	Up	4	7.8
		5	1.1

```
SELECT id, title, movie_id, rating
FROM
    movies FULL OUTER JOIN ratings
        ON movies.id = ratings.movie_id
```

Summary

- We’ve learnt joining can blend data on multiple tables
- We’ve learnt different kinds of joining
  - Inner Join
  - Left Join
  - Right Join
  - Full Outer Join


INNER JOIN
left table
right table
FULL JOIN
left table
right table
LEFT JOIN
left table
right table
RIGHT JOIN
left table
right table