

The Sequel to SQL: Level 4 – Section 3

Outer Joins

Listing All Movies With Optional Reviews

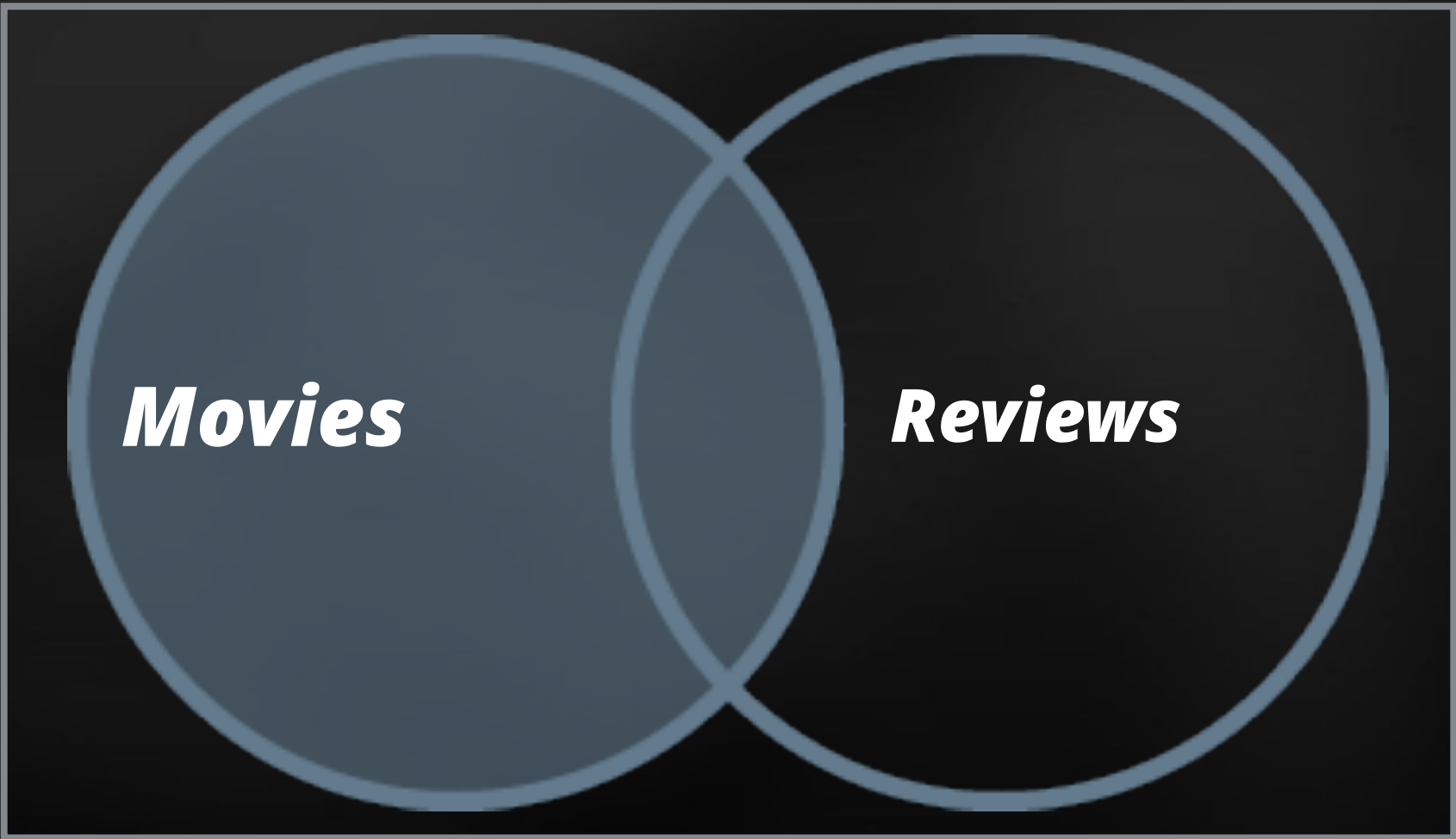
Movies

id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews

id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	1
4	It was okay	3
5	Do not see!	4

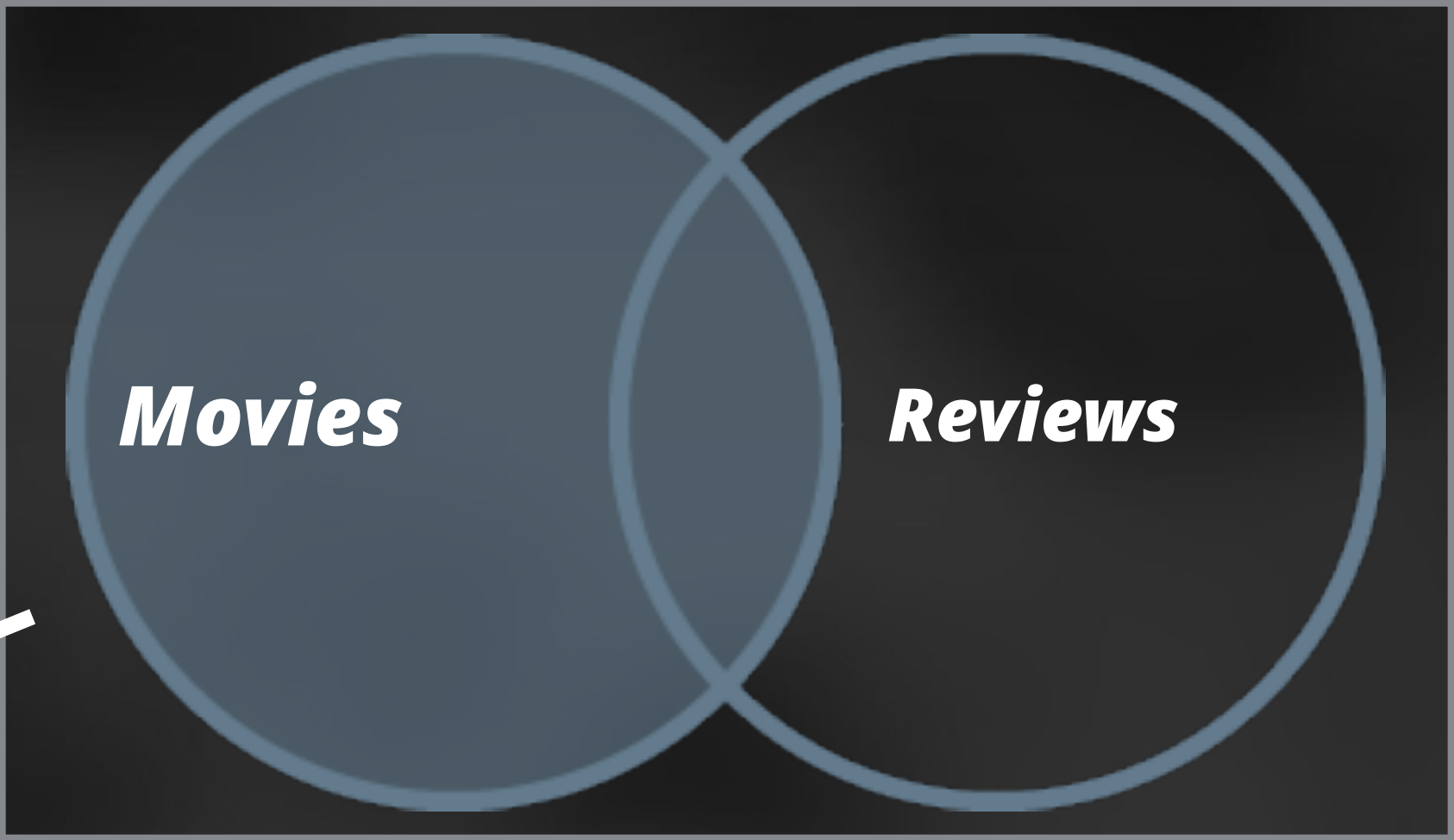
We are looking to display **all** records in the left table...



...and display matching records from the table on the right.

Where the Left Outer Join Takes Place

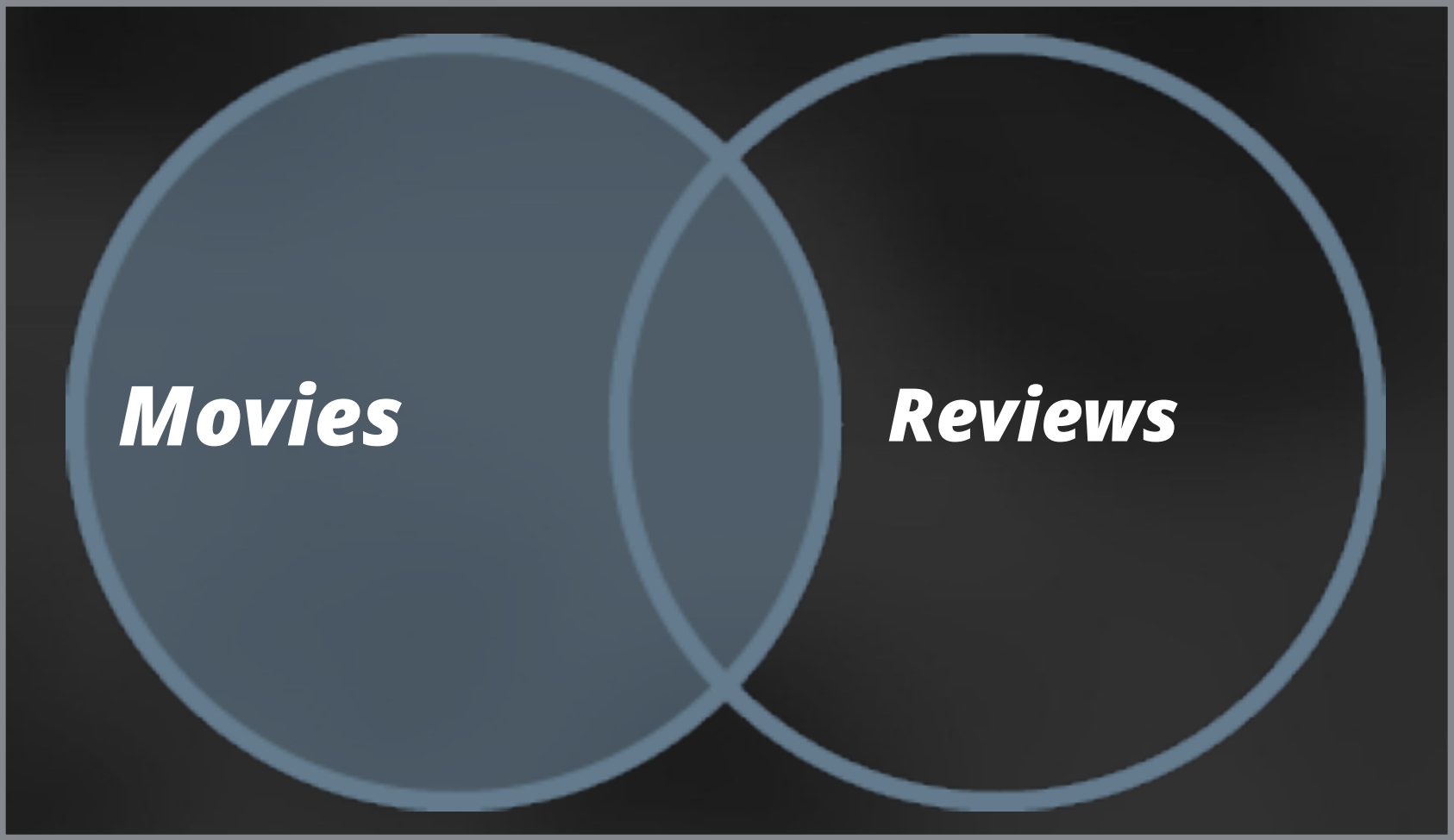
We want to take all rows in the **Movies** table...



Movies			
id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews		
id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	1
4	It was okay	3
5	Do not see!	4

The Left Outer Join



*...and join them with matching **movie_ids** in our **Reviews** table, which is our table on the right.*

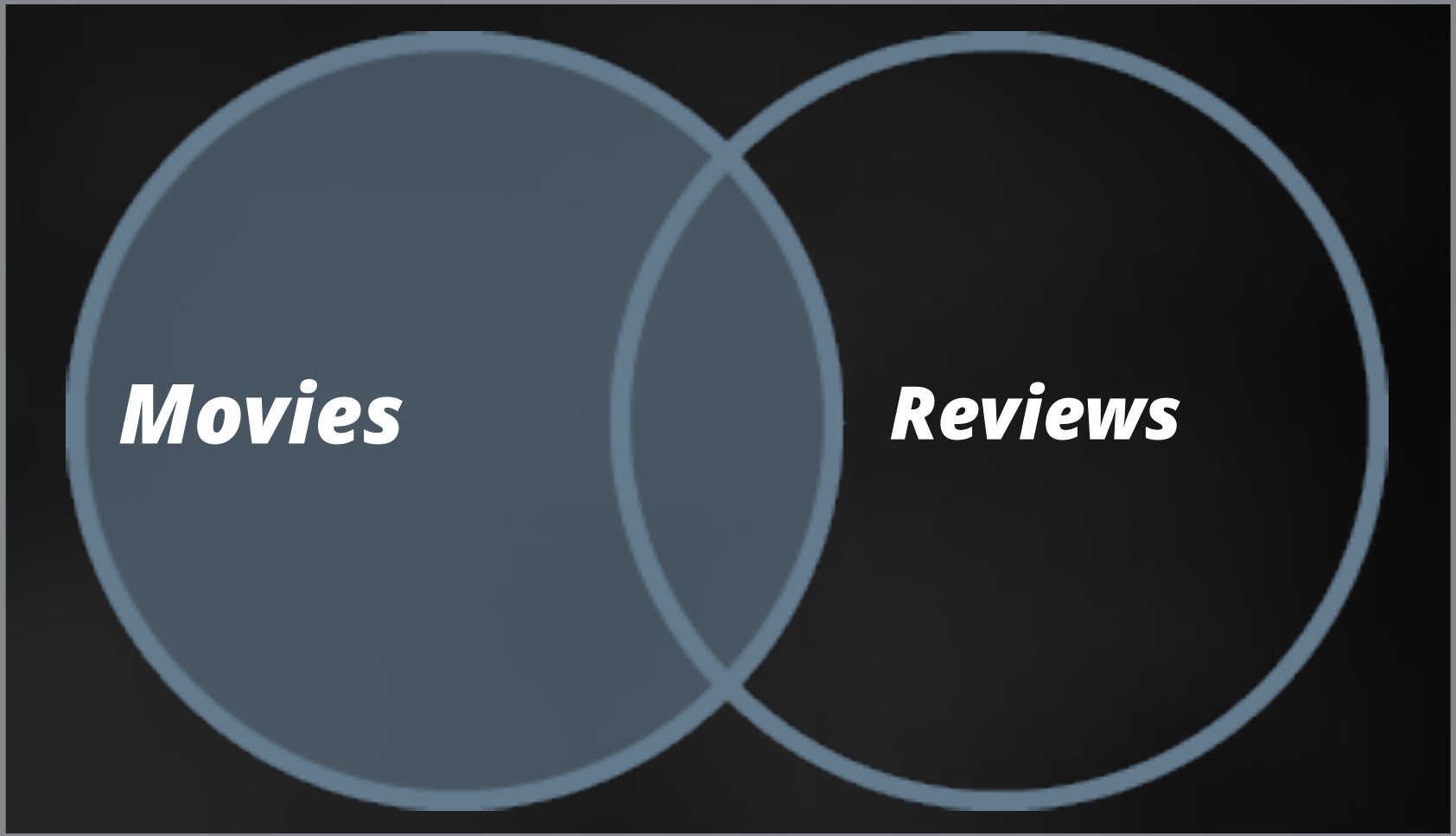
Movies			
id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews		
id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	1
4	It was okay	3
5	Do not see!	4

Matching id and movie_id Values

All film information and only the **Reviews** associated with them

```
SELECT *
FROM Movies
LEFT OUTER JOIN Reviews
ON Movies.id=Reviews.movie_id;
```



title	genre	duration	id	movie_id	review	id
Don Juan	Romance	110	1	1	Loved it!	1
Don Juan	Romance	110	1	1	A must-see!	2
Don Juan	Romance	110	1	1	Hated it	3
Peter Pan	Adventure	105	2			
The Lost World	Fantasy	106	3	3	It was okay	4
Robin Hood	Adventure	143	4	4	Do not see!	5

(6 rows)

*This row is empty because there is no matching value for this row in the **Reviews** table.*

Finding Matching Columns With LEFT OUTER JOIN

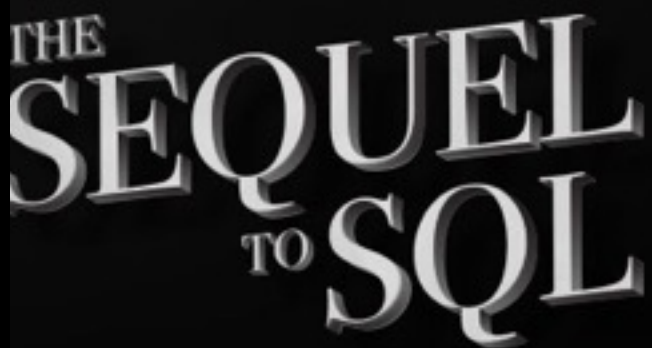
Movies			
id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews		
id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	1
4	It was okay	3
5	Do not see!	4

This time with columns and table aliases

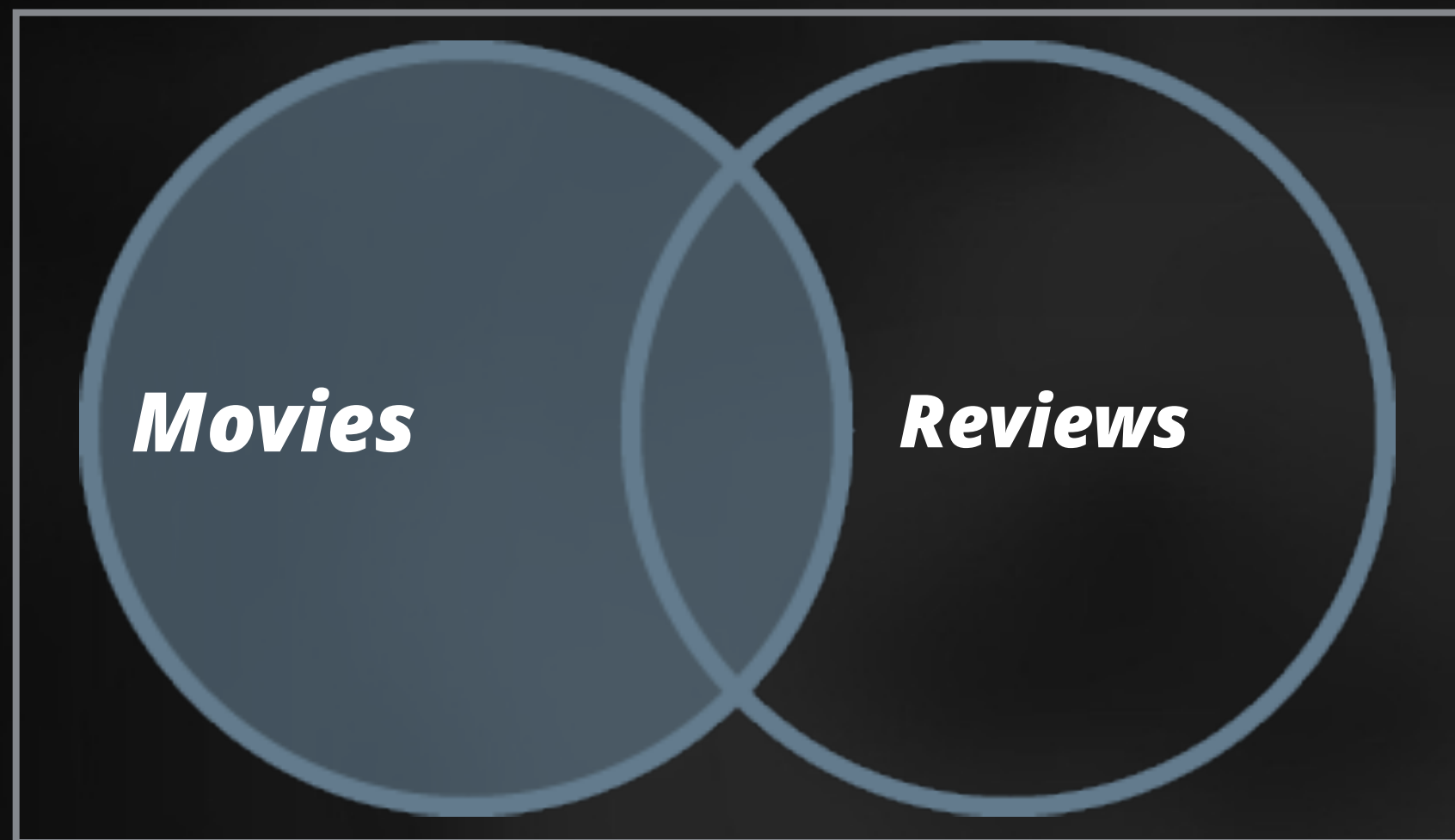
```
SELECT m.title, r.review
FROM Movies m
LEFT OUTER JOIN Reviews r
ON m.id=r.movie_id
ORDER BY r.id;
```

title	review
Don Juan	Loved it!
Don Juan	A must-see!
Don Juan	Hated it
Robin Hood	It was okay
The Lost World	Do not see!
Peter Pan	
(6 rows)	



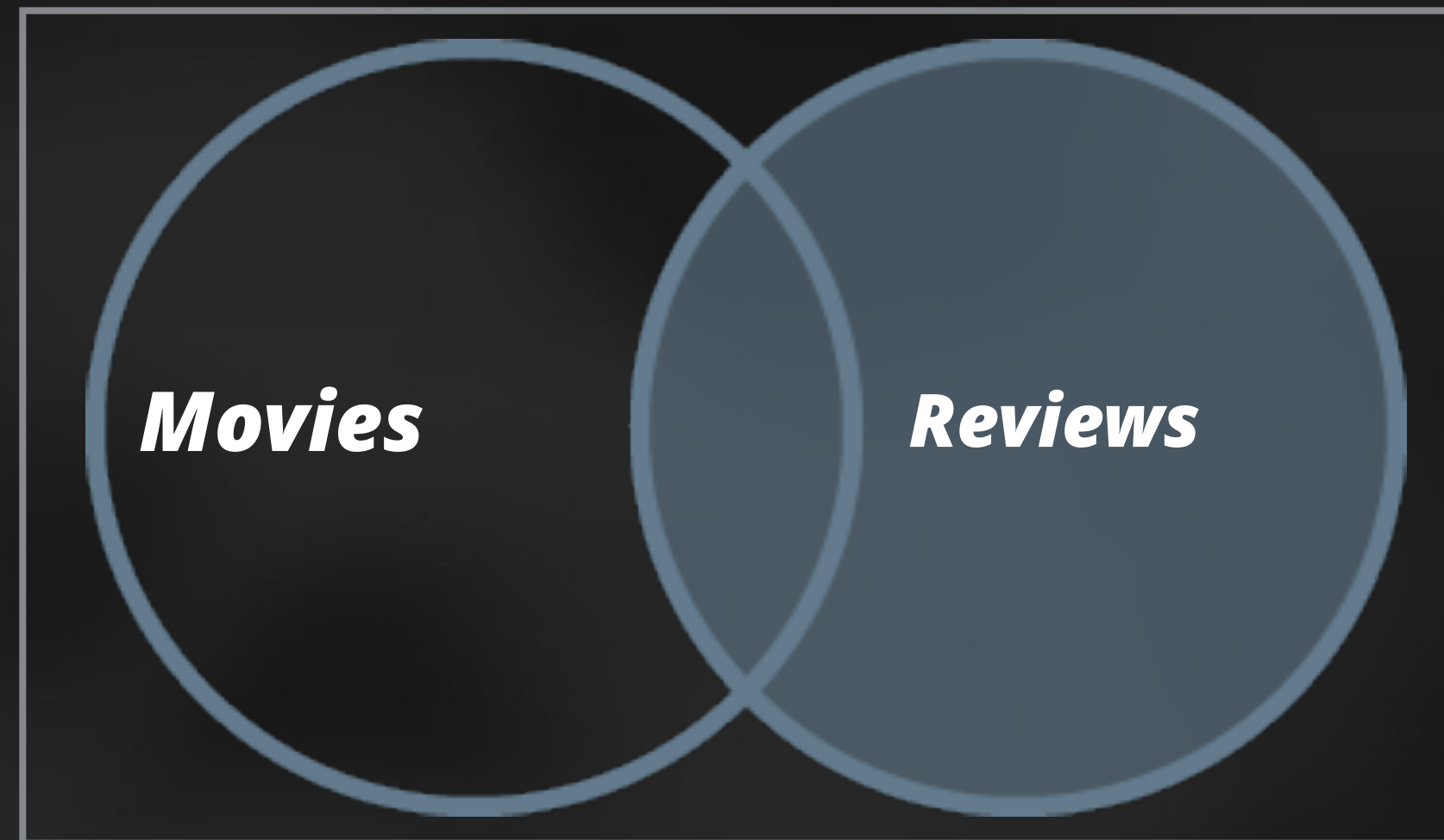
Left Outer Join vs. Right Outer Join

Left Outer Join



Display all the **Movies** and matching **Reviews** if they exist.

Right Outer Join



Display all the **Reviews** and matching **Movies** if they exist.

Using the RIGHT OUTER JOIN Clause

We will use the RIGHT OUTER JOIN clause to list all **Reviews** and only films that are associated.

Movies

id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews

id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	NULL
4	It was okay	NULL
5	Do not see!	4

```
SELECT *  
FROM Movies  
RIGHT OUTER JOIN Reviews  
ON Movies.id=Reviews.movie_id;
```

Notice we modified a few reviews to have a null **movie_id**

Result of RIGHT OUTER JOIN

Movies			
id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews		
id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	NULL
4	It was okay	NULL
5	Do not see!	4

title	genre	duration	id	movie_id	review	id
Don Juan	Romance	110	1	1	Loved it!	1
Don Juan	Romance	110	1	1	A must-see!	2
					Hated it	3
					It was okay	4
Robin Hood	Adventure	143	4	4	Do not see!	5
(5 rows)						

These rows are empty because there are no matching films for these rows in the **Movies** table.

Finding Matching Columns With RIGHT OUTER JOIN

And this time with table aliases

Movies			
id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	105
3	The Lost World	Fantasy	106
4	Robin Hood	Adventure	143

Reviews		
id	review	movie_id
1	Loved it!	1
2	A must-see!	1
3	Hated it	NULL
4	It was okay	NULL
5	Do not see!	4

```
SELECT m.title, r.review
FROM Movies m
RIGHT OUTER JOIN Reviews r
ON m.id = r.movie_id
ORDER BY r.id;
```

title	review
Don Juan	Loved it!
Don Juan	A must-see!
	Hated it
	It was okay
The Lost World	Do not see!

(6 rows)

