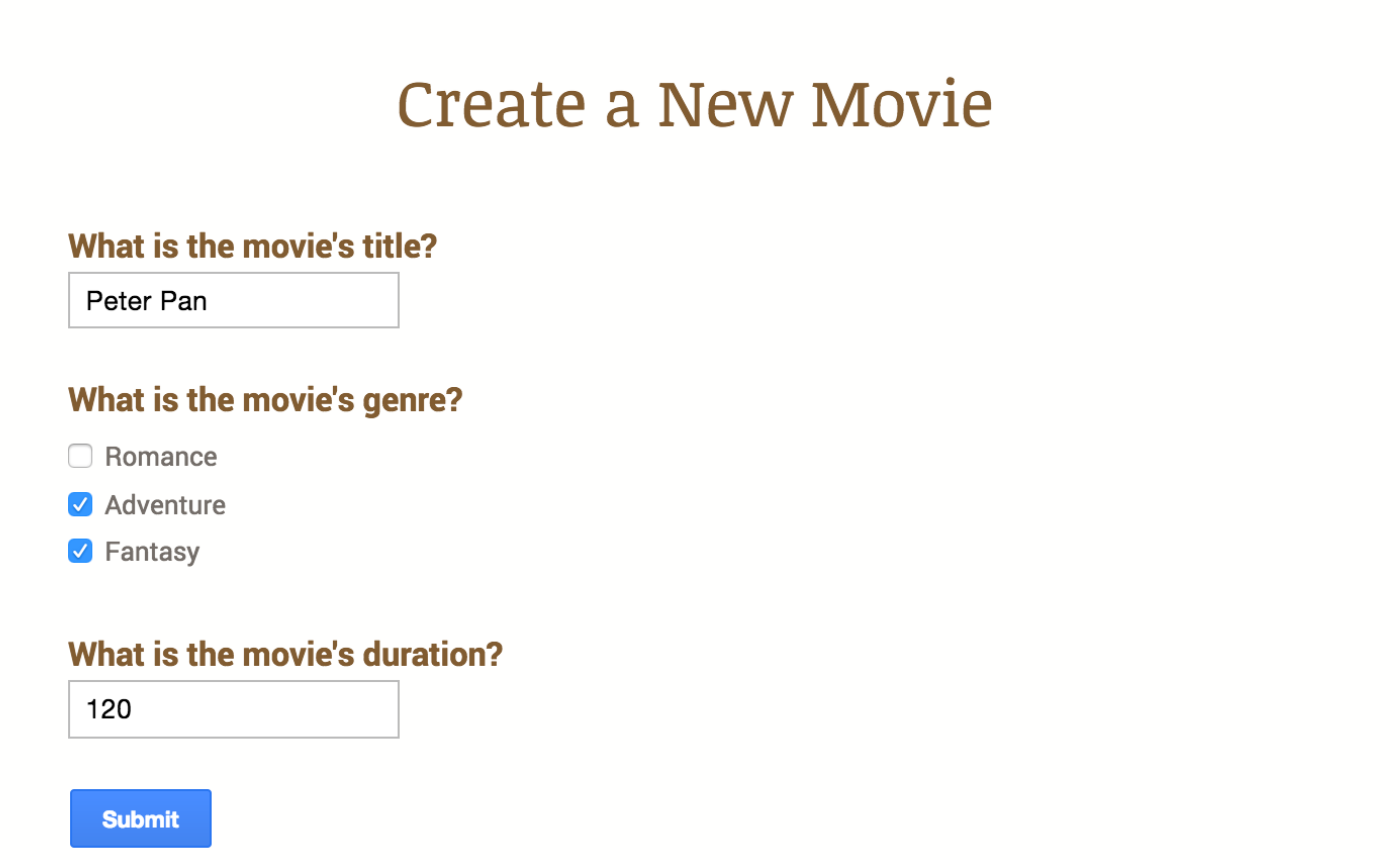
The Sequel to SQL:
Level 3 – Section 1

Normalization



Create a New Movie

What is the movie's title?

Peter Pan

What is the movie's genre?

☐ Romance

☑ Adventure

☑ Fantasy

What is the movie's duration?

120

Our Movies Table With Multiple Genres

This is the table we might generate.

id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure, Fantasy	120
3	The Lost World	Fantasy	105
4	Robin Hood	Adventure	143

✋ *We can't update these values separately.*

What if we wanted all Adventure movies?

```
SELECT *
FROM movies
WHERE genre = "Adventure";
```

✋ *This would only return "Robin Hood".*

We Need to Use Some Normalization

Normalization is the process of reducing duplication in database tables.

First Normal Form Rule:
Tables must not contain repeating groups of data in 1 column.

Second Normal Form Rule:
Tables must not contain redundancy (unnecessary repeating information).

id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure, Fantasy	120
3	The Lost World	Fantasy	105
4	Robin Hood	Adventure	143

First Normal Form: Flattening the Database

We now have no repeating groups of data.

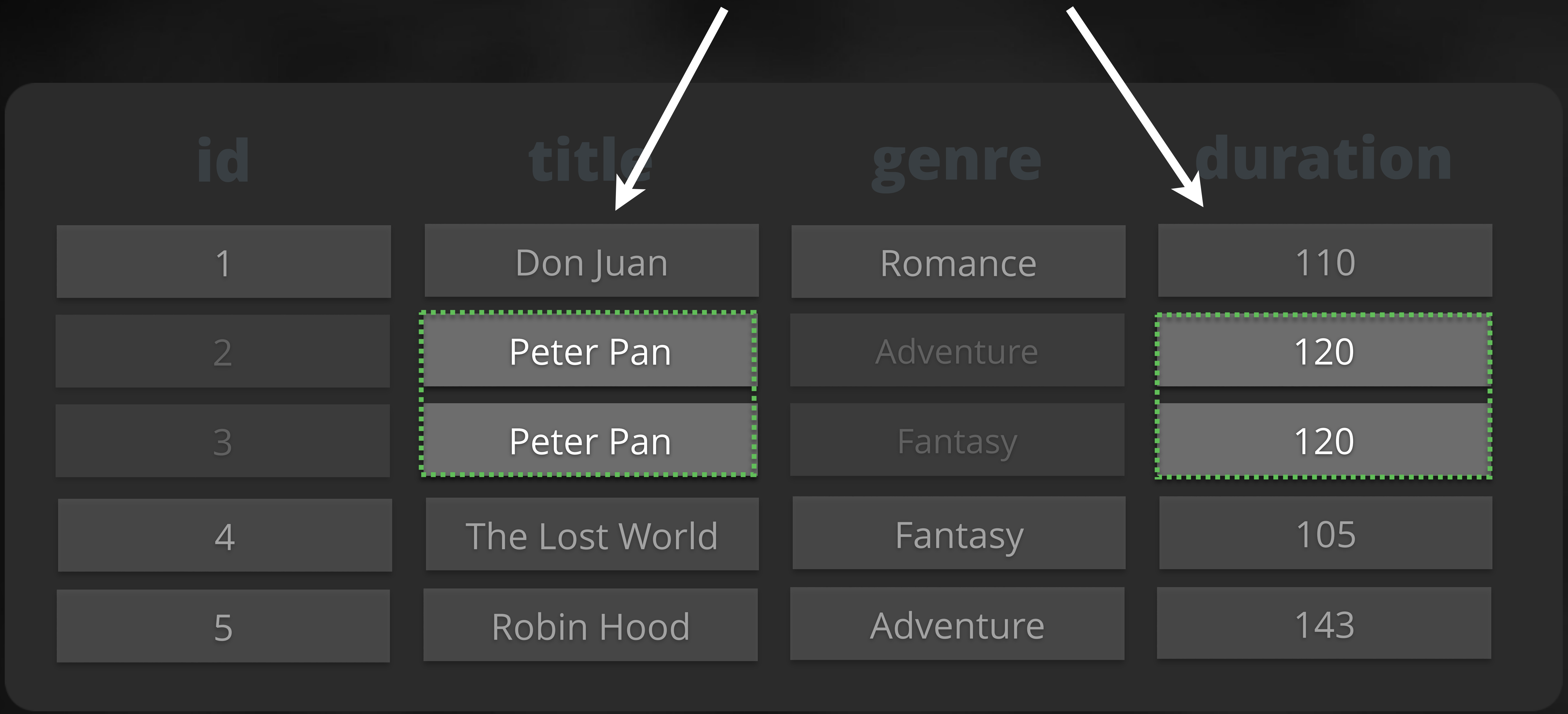
Each record has a different ***GENRE***

id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	120
3	Peter Pan	Fantasy	120
4	The Lost World	Fantasy	105
5	Robin Hood	Adventure	143

- ✋ When we update a movie's duration, we have to update every duplicate movie row.
- ✋ The title column has a unique constraint.

We Need the Second Normal Form

Tables must not contain redundancy (unnecessary repeating information).



id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	120
3	Peter Pan	Fantasy	120
4	The Lost World	Fantasy	105
5	Robin Hood	Adventure	143

Going Back to a Movies Table With Unique Titles

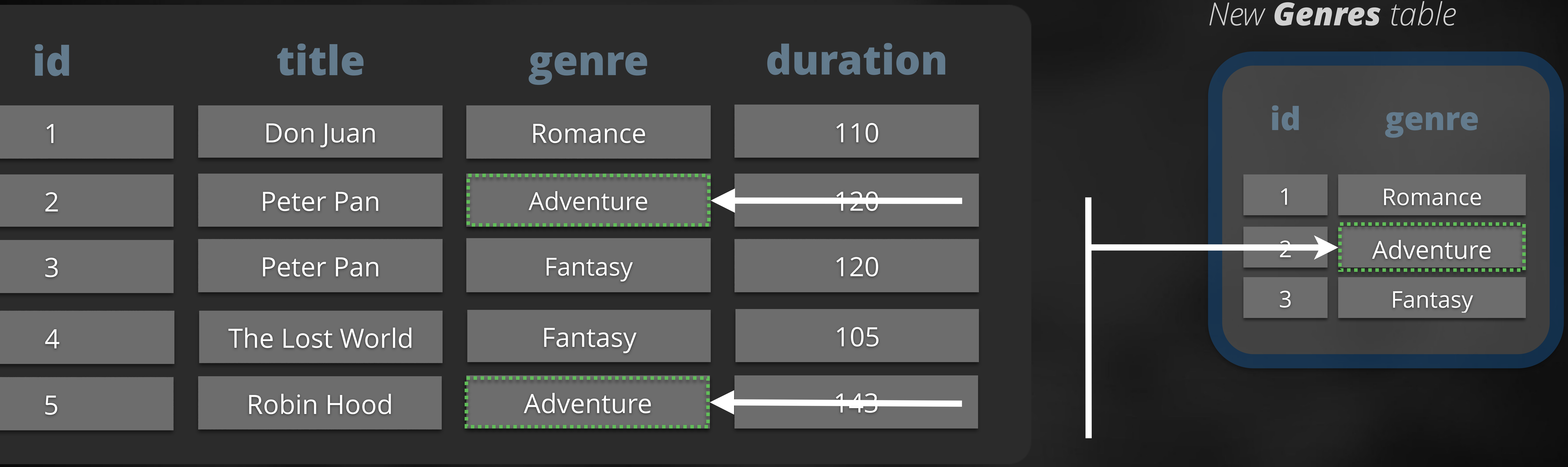
We can reduce redundancy by eliminating repeating column values within our table.

New ***Movies*** *table*

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

id	title	genre	du
1	Don Juan	Romance	
2	Peter Pan	Adventure	
3	Peter Pan	Fantasy	
4	The Lost World	Fantasy	
5	Robin Hood	Adventure	

Creating a New Genres Table



id	title	genre	duration
1	Don Juan	Romance	110
2	Peter Pan	Adventure	120
3	Peter Pan	Fantasy	120
4	The Lost World	Fantasy	105
5	Robin Hood	Adventure	143

*New **Genres** table*

id	genre
1	Romance
2	Adventure
3	Fantasy

*Repeating values on 2 different rows can be combined to make 1 row in a new **Genres** table.*

Reviewing Our New Tables

Two new tables have been created without repeating values.

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy

Both new tables contain no repeating entries.

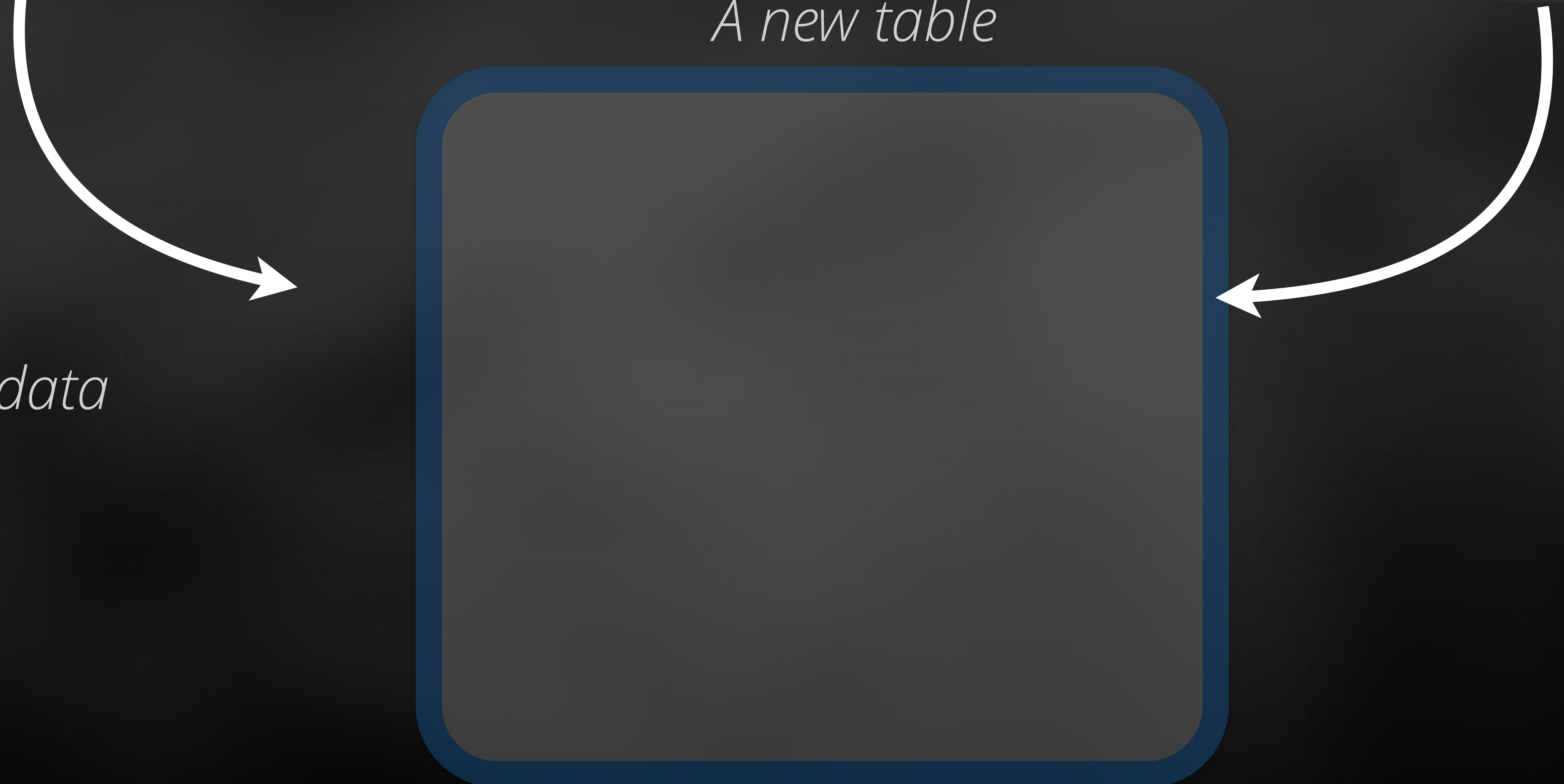
We Need a Link Between the Tables

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy



We need a way to link the data between these 2 tables.

Creating a Join Table

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy

Our join table

Join table naming convention

Movies_Genres

From the Movies table

From the Genres table

Building Our Join Table With Foreign Keys

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy

Movies_Genres

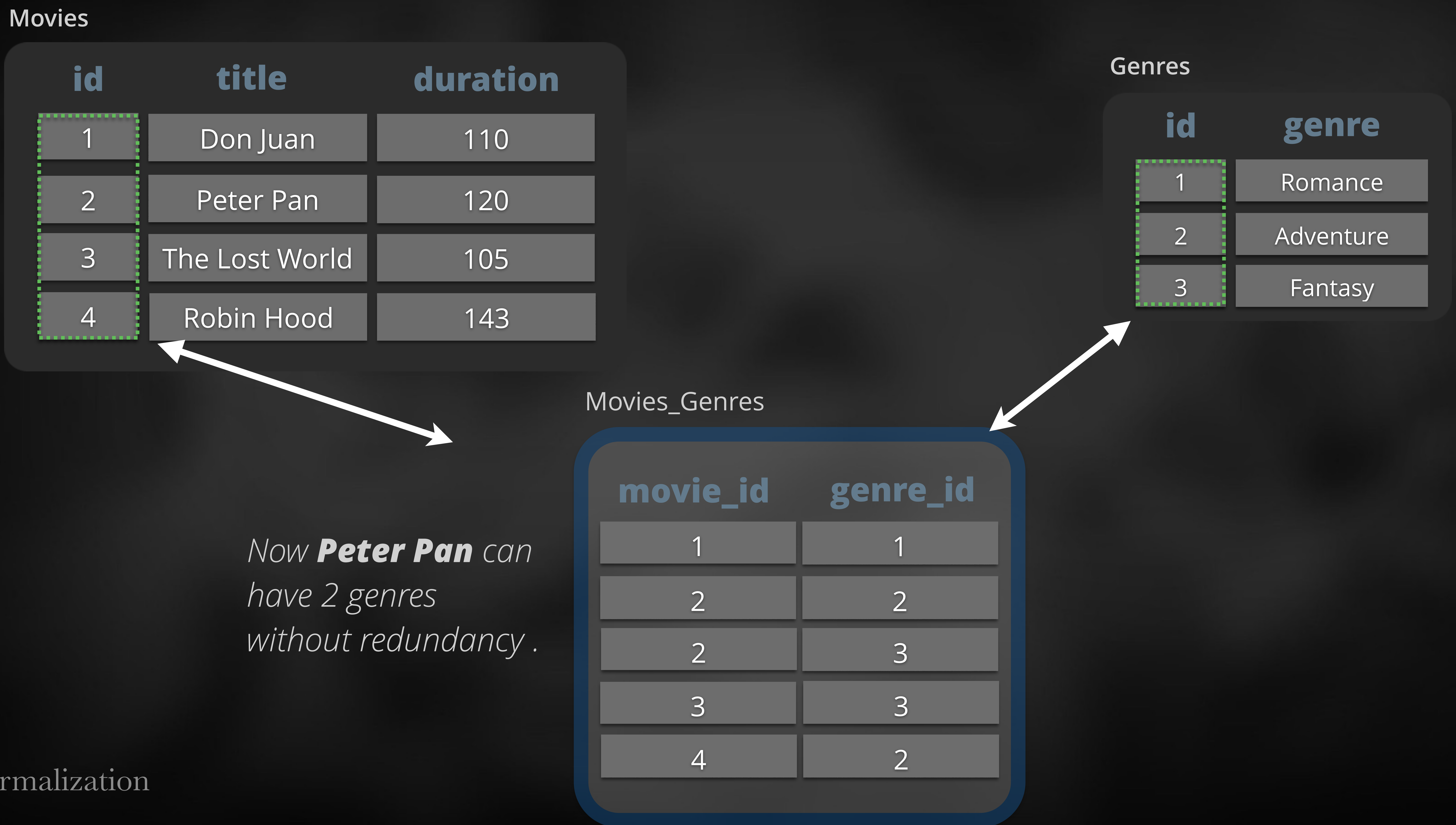
movie_id	genre_id

References the ***primary key*** *of the* ***Movies*** *table*

References the ***primary key*** *of the* ***Genres*** *table*

Both columns are ***foreign keys***

Mapping Movies to Genres



Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy

Movies_Genres

movie_id	genre_id
1	1
2	2
2	3
3	3
4	2

*Now **Peter Pan** can have 2 genres without redundancy .*

We’ve Met 2 Normalization Form Rules

First Normal Form Rule:
Tables must not contain repeating groups of data in 1 column.

Second Normal Form Rule:
Tables must not contain redundancy (unnecessary repeating information).

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	120
3	The Lost World	105
4	Robin Hood	143

Movies_Genres

movie_id	genre_id
1	1
2	2
2	3
3	3
4	2

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy

Updating Movie Information Is Easier

To update a movie duration:

```
UPDATE Movies
SET duration = 134
WHERE id = 2;
```

To add a genre to a movie:

```
INSERT INTO Movies_Genres (movie_id, genre_id)
VALUES (4, 3);
```

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	134
3	The Lost World	105
4	Robin Hood	143

Movies_Genres

movie_id	genre_id
1	1
2	2
2	3
3	3
4	2
4	3

Genres

id	genre
1	Romance
2	Adventure
3	Fantasy

Gathering Data Is a Little More Complex

How do we find the genres of *Peter Pan*?

```
SELECT id
FROM Movies
WHERE title = "Peter Pan";
```

Fetches the ***id***

```
SELECT genre_id
FROM Movies_Genres
WHERE movie_id = 2;
```

Fetches the ***genre_id****s for our movie*

```
SELECT name
FROM Genres
WHERE id = 2 or id = 3;
```

Fetches the genre ***name****s for our movie*

```
WHERE id IN (2,3);
```

Movies

id	title	duration
1	Don Juan	110
2	Peter Pan	134
3	The Lost World	105
4	Robin Hood	143

Movies_Genres

movie_id	genre_id
1	1
2	2
2	3
3	3
4	2
4	3

Genres

id	name
1	Romance
2	Adventure
3	Fantasy