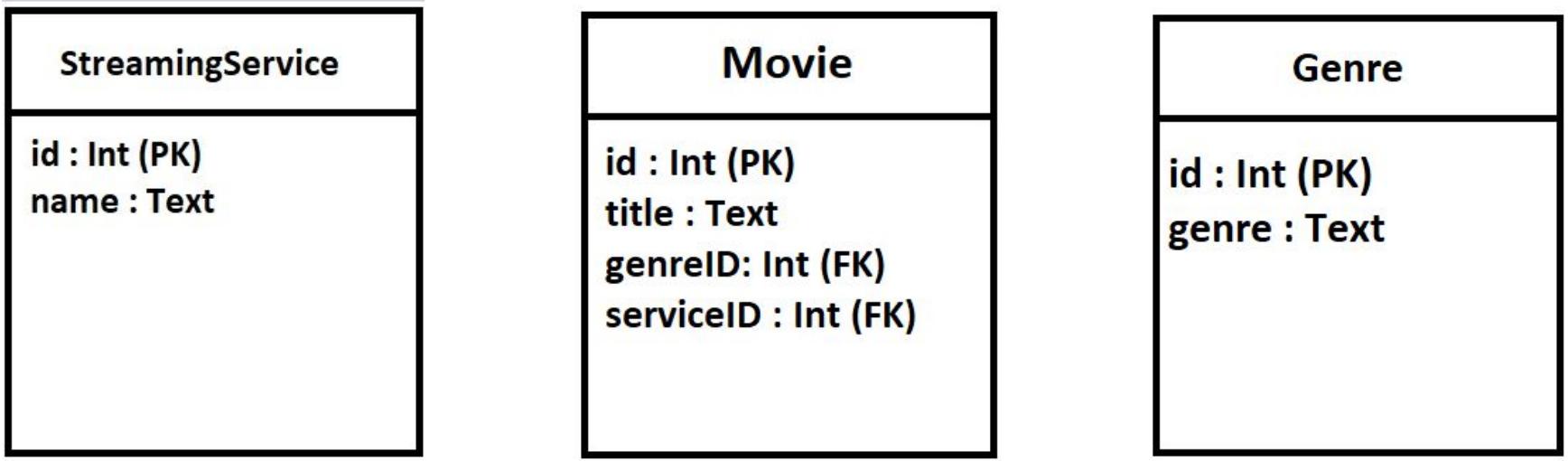
SQL STATEMENTS

SQL

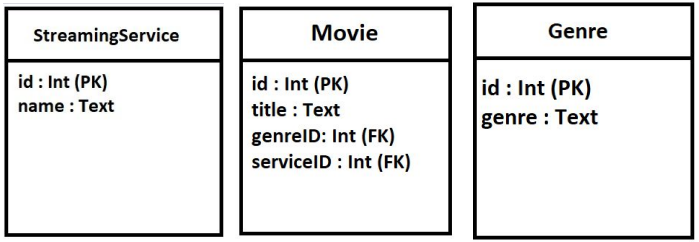
Create	Insert
Read	Select
Update	Update
Delete	Delete

SQL

Exempel baserat på “TV-spels DB”-uppgiften.



INSERT

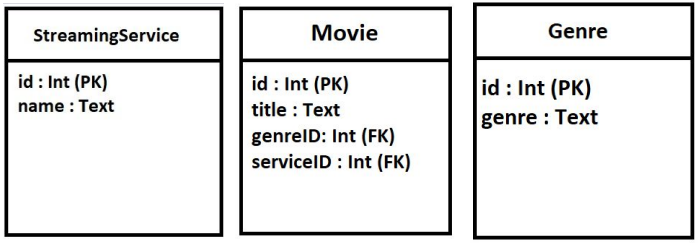


Lägg till en ny streamingtjänst

INSERT INTO streamingService **(**name**) VALUES (**‘Hulu’**)**

```
try {
    PreparedStatement statement = conn.prepareStatement( sql: "INSERT INTO streamingService (name) VALUES (?)");
    statement.setString( parameterIndex: 1, name);
    statement.executeUpdate();
```

SELECT



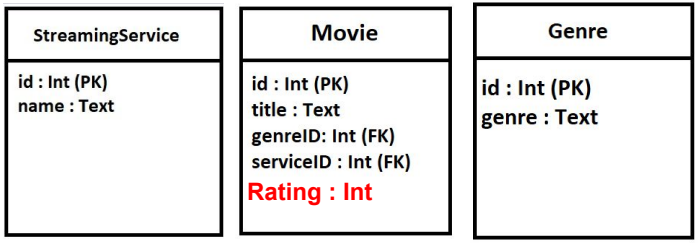
Hämta ut **alla** filmer

SELECT * FROM movie

Hämta ut **alla** filmer från en specifik streamingtjänst

SELECT * FROM streamingService **INNER JOIN** movie **ON** streamingService.id **=** movie.serviceID
WHERE streamingService.name **=** ‘Netflix’

UPDATE



Uppdatera en specifik film

UPDATE movie SET rating = 10 WHERE id = 5

DELETE

StreamingService	Movie	Genre
id : Int (PK) name : Text	id : Int (PK) title : Text genreID: Int (FK) serviceID : Int (FK)	id : Int (PK) genre : Text

Ta bort en **streamingtjänst** (namn)

DELETE FROM streamingService
WHERE streamingService.name = ‘Netflix’

```
private void deleteStreamingService(String streamingService) {
    try {
        PreparedStatement statement = conn.prepareStatement( sql: "DELETE FROM streamingService WHERE streamingService.name = ?");
        statement.setString( parameterIndex: 1, streamingService);
        statement.executeUpdate();

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

JOINs

En JOIN-sats används för att kombinera rader från två eller flera tabeller, baserat på en relaterad kolumn mellan dem.

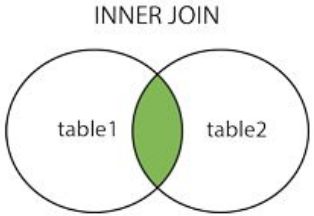
- **INNER JOIN**
- **LEFT (OUTER) JOIN**
- **RIGHT (OUTER) JOIN**
- **FULL (OUTER) JOIN**

INNER JOIN

INNER JOIN är standard när vi inte anger typen av join.

När du gör en **INNER JOIN** av två tabeller returnerar den en ny uppsättning data med alla förekomster av kopplingen där villkoret uppfylls.

Om villkoret inte var uppfyllt mellan tabellerna ignoreras raderna → leder till det minsta antalet resultat.



INNER JOIN

Exempel

Users

id	name
1	Dave
2	Jennifer
3	Ben
4	Tara
5	Justin
6	Praveen

Orders

order_id	item
1	pizza
2	soda
3	french fries
4	french fries
5	burger
6	soda
7	pizza
8	burger

INNER JOIN

Exempel

id	name
1	Dave
2	Jennifer
3	Ben
4	Tara
5	Justin
6	Praveen

order_id	item
1	pizza
2	soda
3	french fries
4	french fries
5	burger
6	soda
7	pizza
8	burger

```
SELECT * FROM users INNER JOIN orders ON users.id = orders.user_id
```

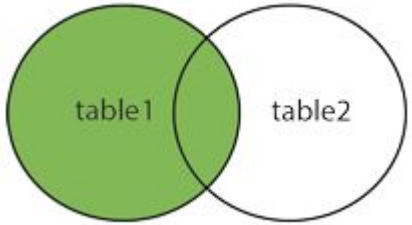
user_id	name	order_id	item
1	Dave	1	pizza
1	Dave	2	soda
1	Dave	3	french fries
2	Jennifer	4	french fries
4	Tara	5	burger
4	Tara	6	soda
5	Justin	7	pizza

LEFT (OUTER) JOIN

LEFT (OUTER) JOIN innebär att joinen gynnar tabellen till vänster.

Det innebär att alla resultat från den tabellen kommer att visas i resultatet, oavsett om de matchar den sammanfogade tabellen eller inte (på villkoret).

Om de inte matchar några rader i den sammanfogade tabellen så blir dom *null*.



LEFT (OUTER) JOIN

Exempel

id	name
1	Dave
2	Jennifer
3	Ben
4	Tara
5	Justin
6	Praveen

order_id	item
1	pizza
2	soda
3	french fries
4	french fries
5	burger
6	soda
7	pizza
8	burger

```
SELECT * FROM users LEFT OUTER JOIN orders ON users.id = orders.user_id
```

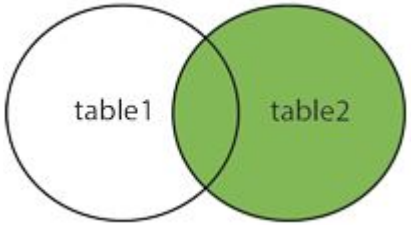
user_id	name	order_id	item
1	Dave	1	pizza
1	Dave	2	soda
1	Dave	3	french fries
2	Jennifer	4	french fries
3	Ben		
4	Tara	5	burger
4	Tara	6	soda
5	Justin	7	pizza
6	Praveen		

RIGHT (OUTER) JOIN

RIGHT (OUTER) JOIN innebär att joinen gynnar tabellen till höger.

Det innebär att alla resultat från den tabellen kommer att visas i resultatet, oavsett om de matchar den sammanfogade tabellen
eller inte (på villkoret) och resultaten från vänstra tabellen som matchar.

Om de inte matchar några rader i den sammanfogade tabellen så blir dom *null*.



RIGHT (OUTER) JOIN

id	name
1	Dave
2	Jennifer
3	Ben
4	Tara
5	Justin
6	Praveen

order_id	item
1	pizza
2	soda
3	french fries
4	french fries
5	burger
6	soda
7	pizza
8	burger

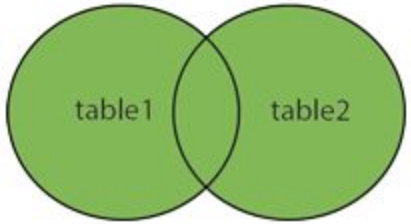
Exempel

```
SELECT * FROM orders RIGHT OUTER JOIN users ON order.id = user.order_id
```

user_id	name	order_id	item
1	Dave	1	pizza
1	Dave	2	soda
1	Dave	3	french fries
2	Jennifer	4	french fries
3		4	french fries
4	Tara	5	burger
4	Tara	6	soda
5	Justin	7	pizza
6		8	burger

FULL (OUTER) JOIN

FULL (OUTER) JOIN innebär att att alla rader från varje tabell kommer att listas i resultatet oavsett om de matchar några rader i den andra tabellen (sällsynta i praktiken).



FULL (OUTER) JOIN

Exempel

id	name
1	Dave
2	Jennifer
3	Ben
4	Tara
5	Justin
6	Praveen

order_id	item
1	pizza
2	soda
3	french fries
4	french fries
5	burger
6	soda
7	pizza
8	burger

```
SELECT * FROM users FULL OUTER JOIN orders ON users.id = orders.user_id
```

user_id	name	order_id	item
1	Dave	1	pizza
1	Dave	2	soda
1	Dave	3	french fries
2	Jennifer	4	french fries
3	Ben		
4	Tara	5	burger
4	Tara	6	soda
5	Justin	7	pizza
6	Praveen		
99		8	burger

SYSJM4

Grundläggande SQL-syntax
https://sysjm4.newton.nodehill.se/article/grundlaggande-sql-syntax

SQL Statements
https://sysjm4.newton.nodehill.se/article/sql-statements

Uppgifter

- **TV-spels databasen**
- **Recept databas**
- **The Farm**

Utöka funktionalitet. Gör om med nya klasser / scenario.