

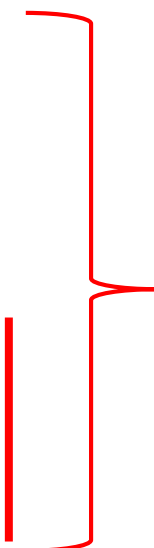
Foreign keys

Efficient and consistent databases

This table contains an overview of leased movies.

- We are storing the name "Thor: Ragnarok" multiple times. This takes up more space than necessary on the disc.
- The last entry has been typed wrong into the database, making it inconsistent.
 - Is Thor: Ragnarok the same as Thor: Ragnarök?
 - If we search for all the people having rented Thor: Ragnarok we won't get all of them

| <u>Rent ID</u> | Renter | Movie ID | Movie name |
|----------------|----------|----------|---------------------------------|
| 1 | Robert | 22 | Fight club |
| 2 | Weird Al | 42 | Hitchhikers guide to the galaxy |
| 3 | Aldrin | 11 | Thor: Ragnarok |
| 4 | Sam | 11 | Thor: Ragnarok |
| 5 | Loki | 11 | <u>Thor: Ragnarök</u> |



We want the movie names to only be written/stored once!!!

Splitting tables

| <u>Rent ID</u> | Renter | <u>Movie ID</u> | Movie name |
|----------------|----------|-----------------|---------------------------------|
| 1 | Robert | 22 | Fight club |
| 2 | Weird Al | 42 | Hitchhikers guide to the galaxy |
| 3 | Aldrin | 11 | Thor: Ragnarok |
| 4 | Sam | 11 | Thor: Ragnarok |
| 5 | Loki | 11 | Thor: Ragnarok |

Delete the row containing Movie names in the Lease table

Removing duplicate rows and making Movie ID as Primary key

Create a new Movies table containing the Movie ID and the Movie name

| <u>Movie ID</u> | Movie name |
|-----------------|---------------------------------|
| 22 | Fight club |
| 42 | Hitchhikers guide to the galaxy |
| 11 | Thor: Ragnarok |
| 11 | Thor: Ragnarok |
| 11 | Thor: Ragnarok |

Making the connection

We now have two tables,
Lease and Movies

The *Movie ID* column of the *Movies* table corresponds to the *Movie ID* column of the *Lease* table

Lease

| <u>Rent ID</u> | Renter | Movie ID |
|----------------|----------|----------|
| 1 | Robert | 22 |
| 2 | Weird Al | 42 |
| 3 | Aldrin | 11 |
| 4 | Sam | 11 |
| 5 | Loki | 11 |

Movies

| <u>Movie ID</u> | Movie name |
|-----------------|---------------------------------|
| 22 | Fight club |
| 42 | Hitchhikers guide to the galaxy |
| 11 | Thor: Ragnarok |

"Thor: Ragnarok" is only stored once, regardless of how many times we rent out the movie!!!

No chance of writing multiple names for the same movie

Foreign key

Movie ID from the *Movies* table is a **primary key** and is "sent" to the *Lease* table.

In the *Lease* table the column *Movie ID* is a **Foreign key** as it comes from another table (where it is a **primary key**). **Foreign keys** are denoted with a *.

A table with a **foreign key** is a **child table** while the table where the key comes from is a **parent** of that child.

Lease

| <u>Rent ID</u> | Renter | Movie ID* |
|----------------|----------|-----------|
| 1 | Robert | 22 |
| 2 | Weird Al | 42 |
| 3 | Aldrin | 11 |
| 4 | Sam | 11 |
| 5 | Loki | 11 |

Child

Movies

| <u>Movie ID</u> | Movie name |
|-----------------|---------------------------------|
| 22 | Fight club |
| 42 | Hitchhikers guide to the galaxy |
| 11 | Thor: Ragnarok |

Parent

Foreign key restrictions

You are not allowed to input a number in the *Movie ID* column in the *Lease* table that doesn't exist in the *Movies* tables' *Movie ID*.

| Lease | | | | |
|----------------|----------|-----------|---|--|
| <u>Rent ID</u> | Renter | Movie ID* | | |
| 1 | Robert | 22 | | |
| 2 | Weird Al | 42 | | |
| 3 | Aldrin | 33 | ✗ | |

?

?

?

| Movies | |
|-----------------|---------------------------------|
| <u>Movie ID</u> | Movie name |
| 22 | Fight club |
| 42 | Hitchhikers guide to the galaxy |
| 11 | Thor: Ragnarok |

Foreign key relations

- Foreign keys ARE relations
 - A table with a foreign key is connected with another table

| <u>Rent ID</u> | Renter | Movie ID |
|----------------|----------|----------|
| 1 | Robert | 22 |
| 2 | Weird Al | 42 |
| 3 | Aldrin | 11 |
| 4 | Jack | 22 |
| 5 | Ole | 11 |

| <u>Movie ID</u> | Movie name |
|-----------------|---------------------------------|
| 22 | Fight club |
| 42 | Hitchhikers guide to the galaxy |
| 11 | Thor: Ragnarok |

Crow foot notation

- The relation between two tables can be show with notation
 - Multiple notations exist, crow foot is popular
 - More on relationships later

A foreign key value can in this case exist multiple times in a child table hence the "many" crow foot

| <u>Rent ID</u> | Renter | Movie ID |
|----------------|----------|----------|
| 1 | Robert | 22 |
| 2 | Weird Al | 42 |
| 3 | Aldrin | 11 |
| 4 | Jack | 22 |
| 5 | Ole | 11 |



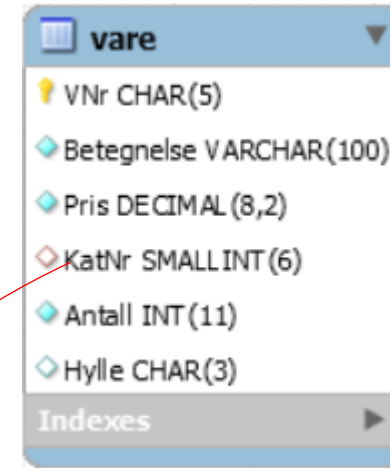
One to many

A key value can only exist once in the parent table

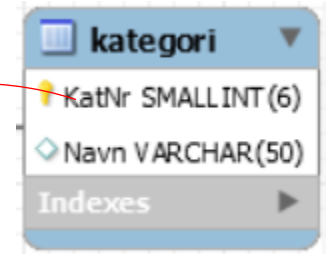
| <u>Movie ID</u> | Movie name |
|-----------------|---------------------------------|
| 22 | Fight club |
| 42 | Hitchhikers guide to the galaxy |
| 11 | Thor: Ragnarok |

Create foreign key

```
CREATE TABLE Vare
(
  VNr          CHAR(5),
  Betegnelse   VARCHAR(100) NOT NULL,
  Pris         DECIMAL(8,2) NOT NULL,
  KatNr        SMALLINT,
  Antall       INTEGER NOT NULL,
  Hylle        CHAR(3),
  CONSTRAINT VarePN PRIMARY KEY (VNr),
  CONSTRAINT VareKategoriFN FOREIGN KEY (KatNr) REFERENCES Kategori (KatNr)
);
```



| vare | |
|------------|--------------|
| VNr | CHAR(5) |
| Betegnelse | VARCHAR(100) |
| Pris | DECIMAL(8,2) |
| KatNr | SMALLINT(6) |
| Antall | INT(11) |
| Hylle | CHAR(3) |
| Indexes | |



| kategori | |
|----------|-------------|
| KatNr | SMALLINT(6) |
| Navn | VARCHAR(50) |
| Indexes | |

You can also add a foreign key later with ALTER TABLE

```
ALTER TABLE Orders
ADD CONSTRAINT FK_PersonOrder
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```