



# Sql relationships

Database Course

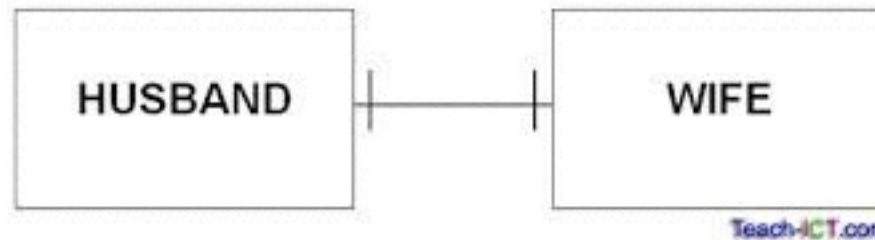
# What we'll cover

- Relationships between tables
- Primary and foreign keys
- Joins in selections

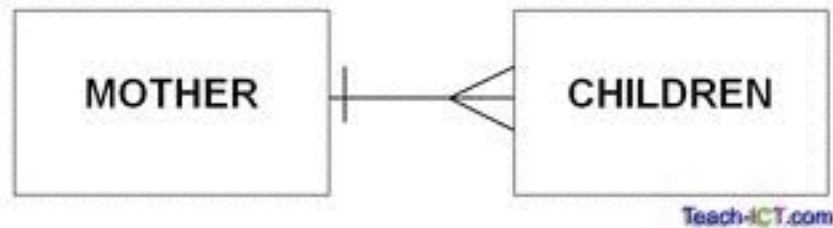
# Relationships between tables

- One to one
- One to many
- Many to many

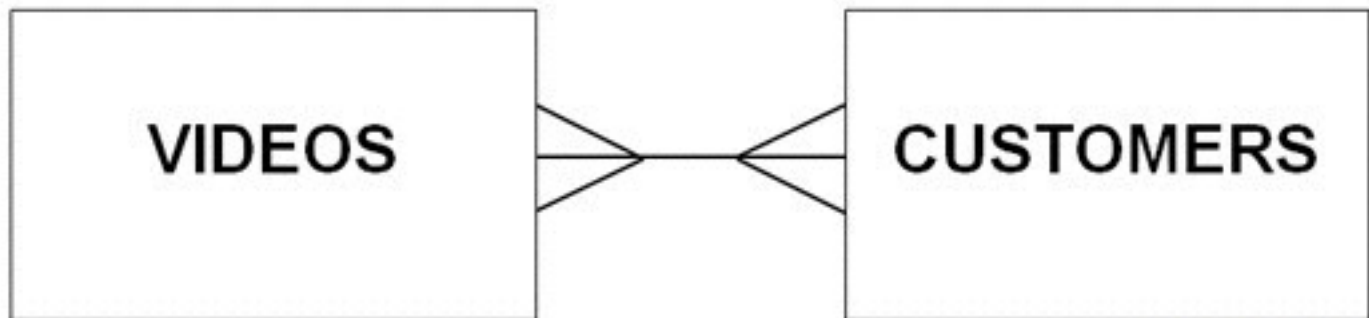
## One-to-one relationship



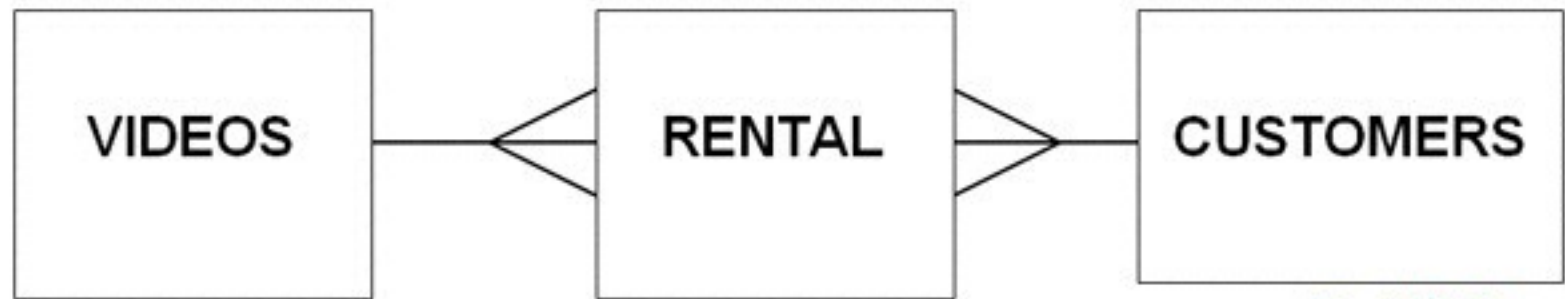
## One-to-many (or many-to-one) relationships



## Many-to-many relationships



Teach-ICT.com



Teach-ICT.com

# Keys

- Primary key
  - A primary key is the column that should be used as the unique identifier of a row in a table.
- Foreign key
  - Is a column that stores the primary key of a foreign table.



# One to One

Primary key

Foreign key

Primary key

departments

staff

id	dept_name	boss_id
1	Accounts	12
2	Research	37
3	Development	53
4	Marketing	42

Relationship

id	firstname	surname
12	Tom	Blackmore
37	Fred	Perry
53	Lotta	Svensson
42	Pia	Fredholm

# Exercise 1

# Joins

- Inner join
- Left outer join
- Right outer join
- Full outer join
- Cross join

# Inner join

- Inner join – returns only when there is match in both tables

```
SELECT firstname, surname, dept_name  
FROM staff INNER JOIN departments ON departments.id =  
staff.department_id;
```

```
SELECT firstname, surname, dept_name FROM staff,  
departments WHERE staff.id = departments.id;
```

# Left outer join

- Left outer join – returns all data in table A (departments), joining where it can with table B (staff).

```
SELECT firstname, surname, dept_name  
FROM staff LEFT OUTER JOIN departments ON departments.id  
= staff.department_id;
```

# Right outer join

- Right outer join – as left join but table B (staff) is returned in full.

```
SELECT firstname, surname, dept_name  
FROM staff RIGHT OUTER JOIN departments ON departments.id  
= staff.department_id;
```

# Full outer join

- Full outer join – returns all data. Combining data where possible, missing data will have a value of null. Not supported in MySQL :-(.

```
SELECT firstname, surname, dept_name  
FROM staff LEFT OUTER JOIN departments ON departments.id =  
staff.department_id  
UNION  
SELECT firstname, surname, dept_name  
FROM staff RIGHT OUTER JOIN departments ON departments.id =  
staff.department_id  
ORDER BY dept_name, surname;
```

# Cross

- Cross join – joins everything to everything. We do not need keys for a cross join.

```
SELECT firstname, surname, dept_name  
FROM staff CROSS JOIN departments;
```



# Joins continued

<http://www.codinghorror.com/blog/2007/10/a-visual-explanation-of-sql-joins.html>