SQL - UNION

Overview over this video

This video covers UNION

UNION

Basically, the same as in math with sets (if you know what that is – otherwise no problem!)

You can also view it as putting one list after another

Let us try:

Say we have an Employees table

We want a list of names i.e. both first and family name

We write as follows:

```
SELECT first_name AS name
FROM Employees
UNION
SELECT family_name
FROM Employees;
```

Column names is from first SELECT

Employees

birthday	first_name	family_name	e_id
1990-11-10	Anne	Smith	1
2000-02-05	David	Jones	2
1995-05-09	William	Taylor	3

name
Anne
David
William
Smith
Jones
Taylor

IMPORTANT: The things you UNION must have similar datatypes and there must be the same number of columns

UNION: Things work

Let us look at some queries:

```
SELECT first_name AS name, e_id
FROM Employees
UNION
SELECT family_name, e_id
FROM Employees;
```

Employees

birthday	first_name	family_name	e_id
1990-11-10	Anne	Smith	1
2000-02-05	David	Jones	2
1995-05-09	William	Taylor	3

name	e_id
Anne	1
David	2
William	3
Smith	1
Jones	2
Taylor	3

UNION: Things work

Let us look at some queries:

```
SELECT first_name AS name, e_id
FROM Employees
UNION
SELECT family_name, e_id
FROM Employees
ORDER BY name;
```

Employees

birthday	first_name	family_name	e_id
1990-11-10	Anne	Smith	1
2000-02-05	David	Jones	2
1995-05-09	William	Taylor	3

name	e_id
Anne	1
David	2
Jones	2
Smith	1
Taylor	3
William	3

UNION: Things work in interesting ways

Let us look at some queries:

```
SELECT first_name
FROM Employees
UNION
SELECT last_name, e_id
FROM Employees;
```

```
SELECT e_id, first_name AS name
FROM Employees
UNION
SELECT family_name, e_id
FROM Employees;
```

Does work in w3school and MySQL but not, e.g. SQL Server because of different definitions of what datatypes are similar

Employees

birthday	first_name	family_name	e_id
1990-11-10	Anne	Smith	1
2000-02-05	David	Jones	2
1995-05-09	William	Taylor	3

e_id	name
1	Anne
2	David
3	William
Smith	1
Jones	2
Taylor	3