Your first database

INTRODUCTION TO RELATIONAL DATABASES IN SQL

Timo Grossenbacher
Data Journalist

datacamp

Investigating universities in Switzerland

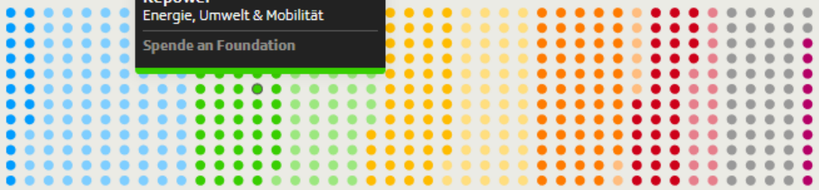
Eidgenössische Technische Hochschule Zürich

Zu dieser Hochschule gehören rund **18'600 Studierende** und **Professor/innen**. Es besteht ein jährlicher Aufwand von rund **1.6 Mrd. Fr.**, wovon **8.8 % aus privaten Drittmitteln** stammen (BFS, 2014).

Jeder Punkt in der Grafik zeigt eine von insgesamt **516 Interessenbindungen.**

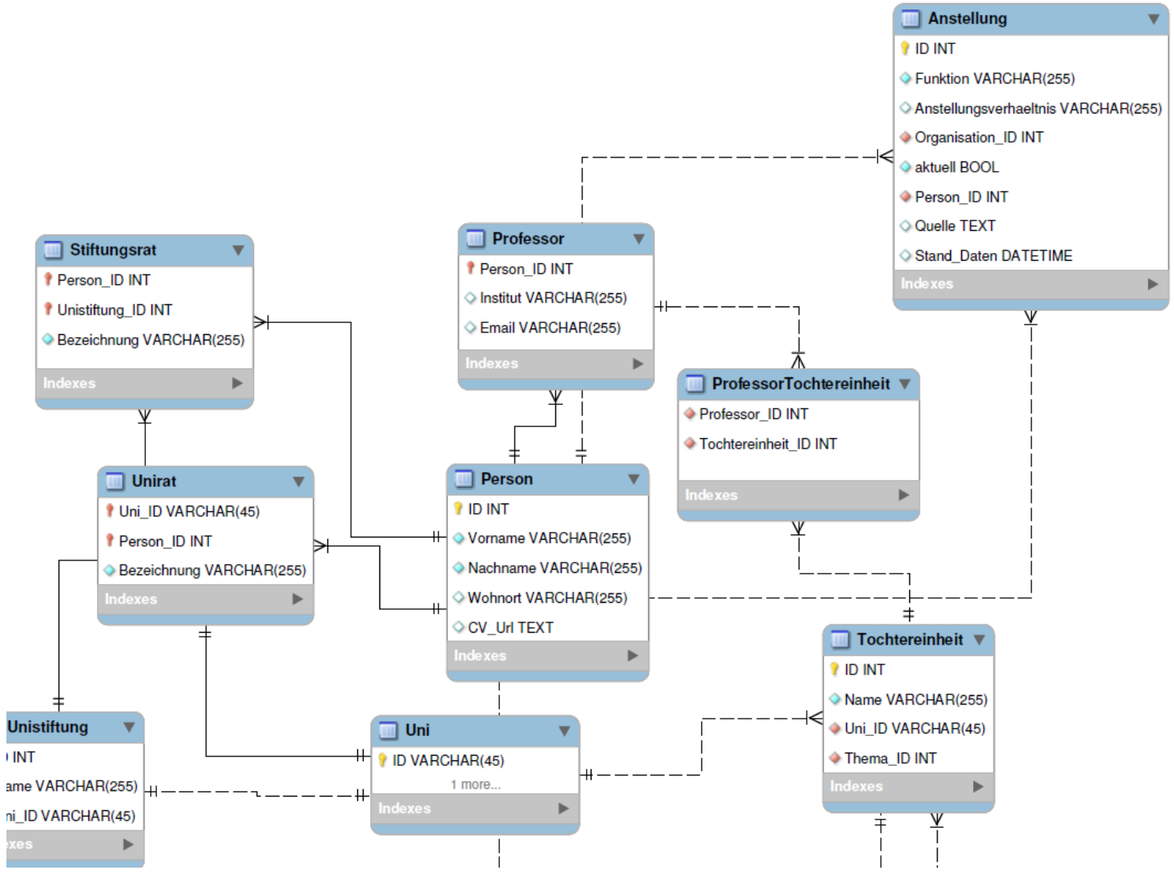
Übersicht | Detail

Repower
Energie, Umwelt & Mobilität
Spende an Foundation



Beratung, Advokatur, PR & Treuhand · Bildung & Forschung · Energie, Umwelt & Mobilität · Finanzwirtschaft & Versicherungen
Gesellschaft, Soziales, Kultur & Sport · Gewerbe, Handel & Dienstleistungen · Industrie, Bau & Landwirtschaft · Medien & Kommunikation
Pharma & Gesundheit · Politik, Verwaltung, Justiz & Sicherheit · Technologie · Nicht klassifizierbar

Stand der Daten: Juni 2015
Quelle: Pressestelle
Interessenbindungen beinhalten auch Mandate in Verwaltungs- und Stiftungsräten sowie Vereinsvorstände, die im Auftrag der ETH Zürich von Professor/innen wahrgenommen werden. ETHZ und EPFL teilen sich den Universitätsrat («ETH-Rat»).



A relational database:

- real-life *entities* become *tables*
- reduced redundancy
- data integrity by *relationships*

- e.g. `professors`, `universities`, `companies`
- e.g. only one entry in `companies` for the bank "Credit Suisse"
- e.g. a `professor` can work at multiple `universities` and `companies`, a `company` can employ multiple `professors`

Throughout this course you will:

- work with the data I used for my investigation
- create a relational database from scratch
- learn three concepts:
  - *constraints*
  - *keys*
  - *referential integrity*

You'll need: Basic understanding of SQL, as taught in **Intro to SQL for Data Science**.

Your first duty: Have a look at the PostgreSQL database

```
SELECT table_schema, table_name
FROM information_schema.tables;
```

```
   table_schema    |              table_name
 ------------------+------------------------------
 pg_catalog        | pg_statistic
 pg_catalog        | pg_type
 pg_catalog        | pg_policy
 pg_catalog        | pg_authid
 pg_catalog        | pg_shadow
 public            | university_professors
 pg_catalog        | pg_settings
...
```

Have a look at the columns of a certain table

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'pg_config';
```

```
 table_name | column_name | data_type
------------+-------------+-----------
 pg_config  | name        | text
 pg_config  | setting     | text
```

Let's do this.

INTRODUCTION TO RELATIONAL DATABASES IN SQL

Tables: At the core of every database

INTRODUCTION TO RELATIONAL DATABASES IN SQL

Timo Grossenbacher
Data Journalist

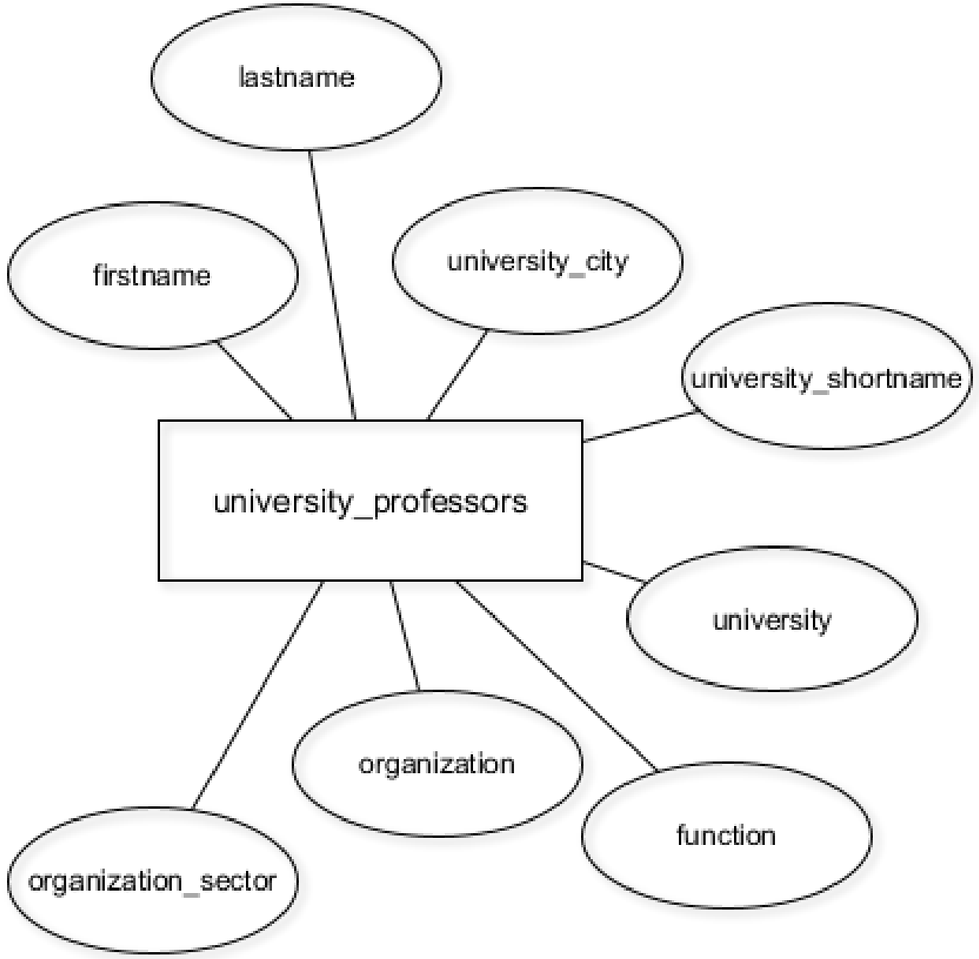
Redundancy in the university_professors table

```
SELECT * FROM
FROM university_professors
LIMIT 3;
```

```
-[ RECORD 1 ]--------+--------------------------------------------------
firstname            | Karl
lastname             | Aberer
university           | ETH Lausanne
university_shortname | EPF
university_city      | Lausanne
function             | Chairman of L3S Advisory Board
organization         | L3S Advisory Board
organization_sector  | Education & research
-[ RECORD 2 ]--------+--------------------------------------------------
firstname            | Karl
lastname             | Aberer
university           | ETH Lausanne
university_shortname | EPF
university_city      | Lausanne
function             | Member Conseil of Zeno-Karl Schindler Foundation
organization         | Zeno-Karl Schindler Foundation
organization_sector  | Education & research
-[ RECORD 3 ]--------+--------------------------------------------------
firstname            | Karl
lastname             | Aberer
(truncated)
function             | Member of Conseil Fondation IDIAP
organization         | Fondation IDIAP
(truncated)
```

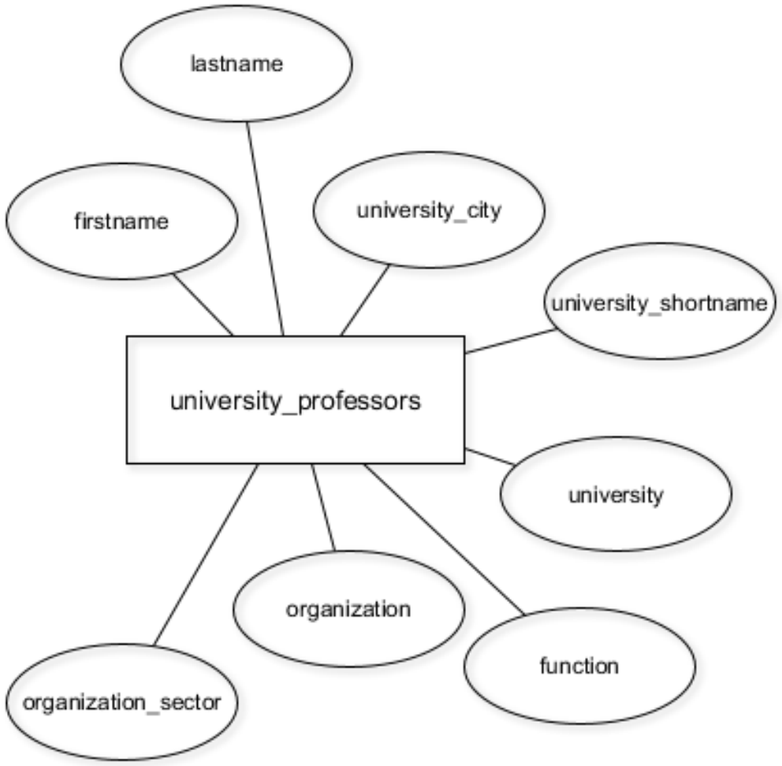
```
-[ RECORD 1 ]--------+-----------------------------------------
firstname            | Karl
lastname             | Aberer
university           | ETH Lausanne
university_shortname | EPF
university_city      | Lausanne
function             | Chairman of L3S Advisory Board
organisation         | L3S Advisory Board
organisation_sector  | Education & research
-[ RECORD 2 ]--------+-----------------------------------------
firstname            | Karl
lastname             | Aberer
university           | ETH Lausanne
university_shortname | EPF
university_city      | Lausanne
function             | Member Conseil of Zeno-Karl Schindler Foundation
organisation         | Zeno-Karl Schindler Foundation
organisation_sector  | Education & research
-[ RECORD 3 ]--------+-----------------------------------------
firstname            | Karl
lastname             | Aberer
(truncated)
function             | Member of Conseil Fondation IDIAP
organisation         | Fondation IDIAP
(truncated)
```

Currently: One "entity type" in the database

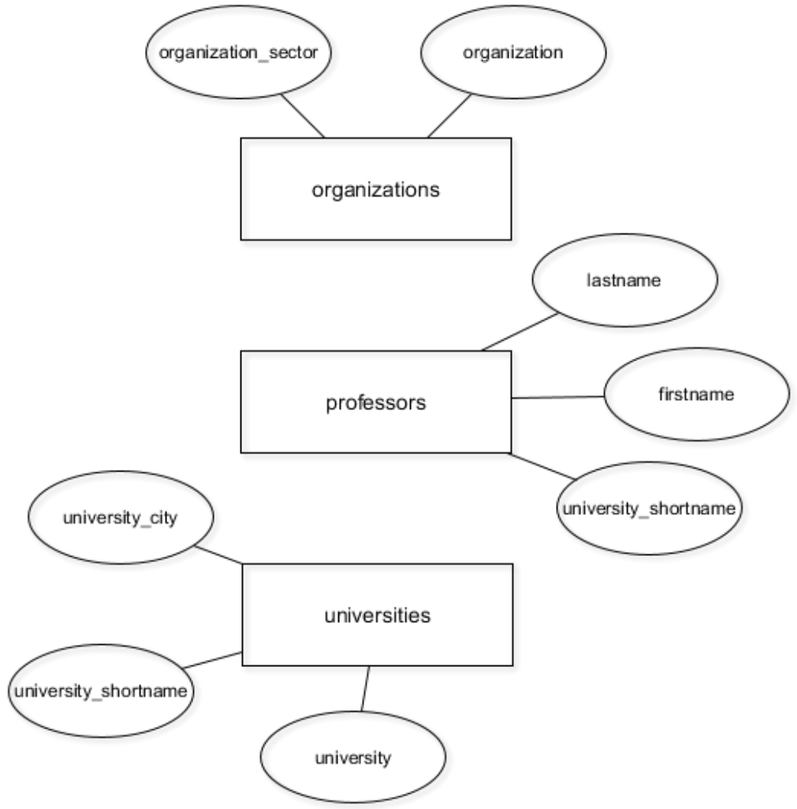


A better database model with three entity types

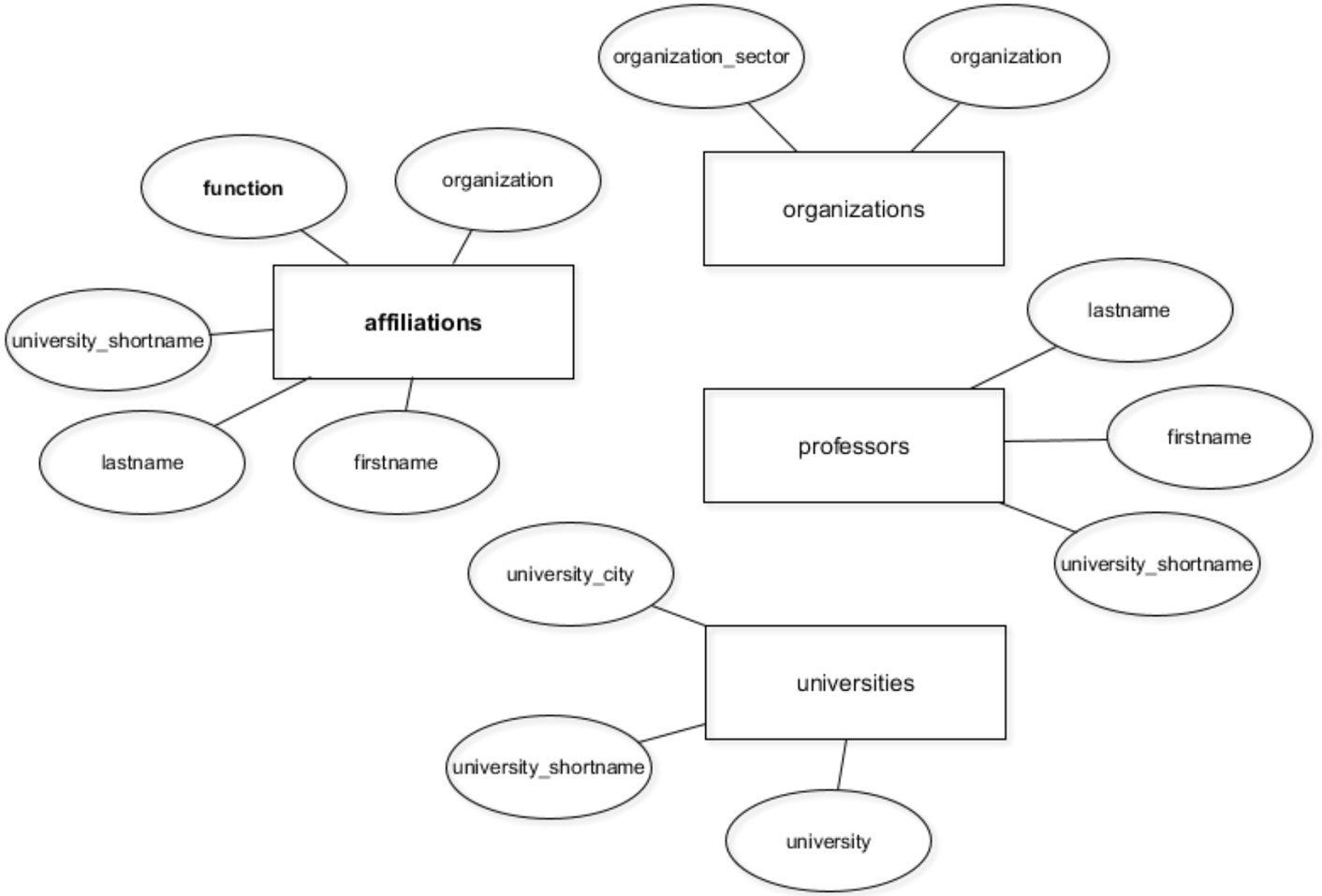
Old:



New:



A better database model with four entity types

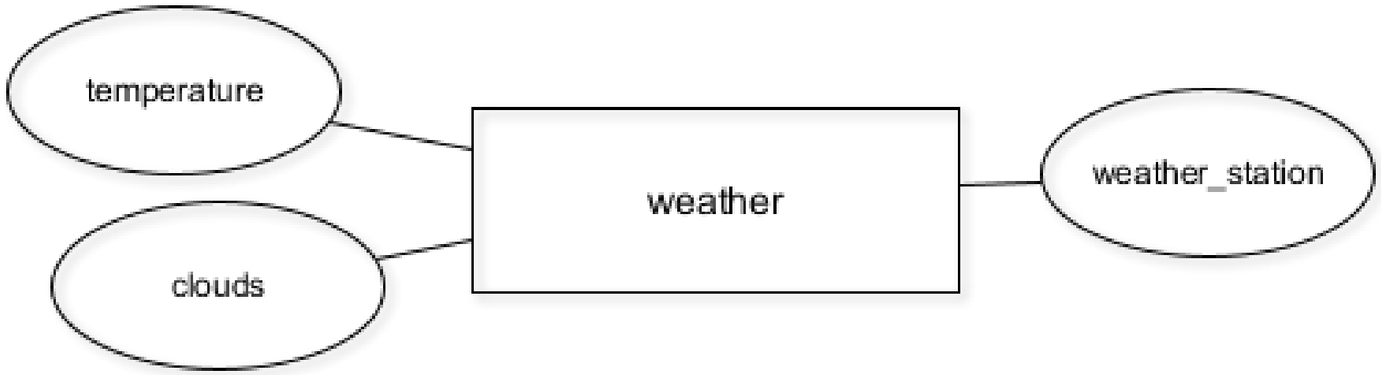


Create new tables with CREATE TABLE

```
CREATE TABLE table_name (
 column_a data_type,
 column_b data_type,
 column_c data_type
);
```

Create new tables with CREATE TABLE

```
CREATE TABLE weather (
 clouds text,
 temperature numeric,
 weather_station char(5)
);
```



Let's practice!

INTRODUCTION TO RELATIONAL DATABASES IN SQL

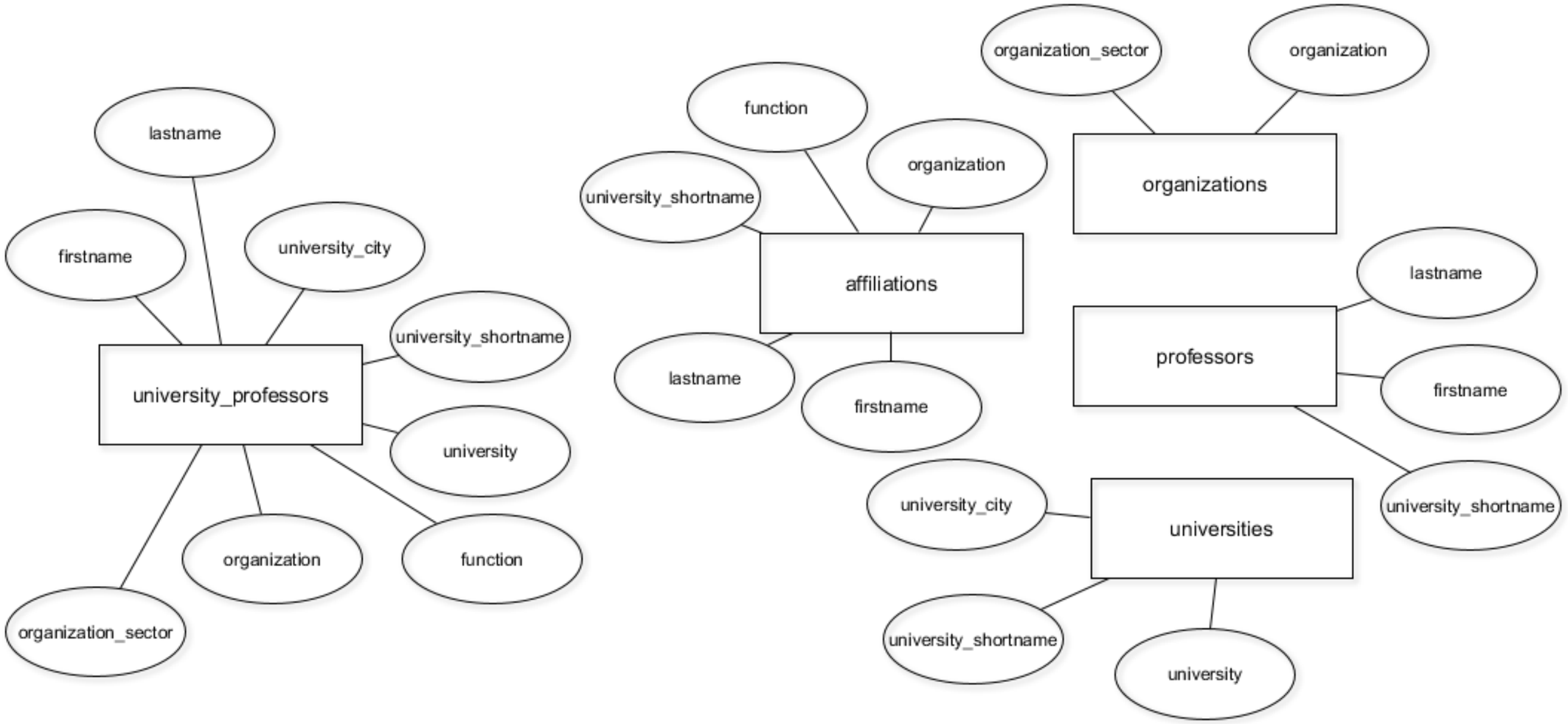
Update your database as the structure changes

INTRODUCTION TO RELATIONAL DATABASES IN SQL

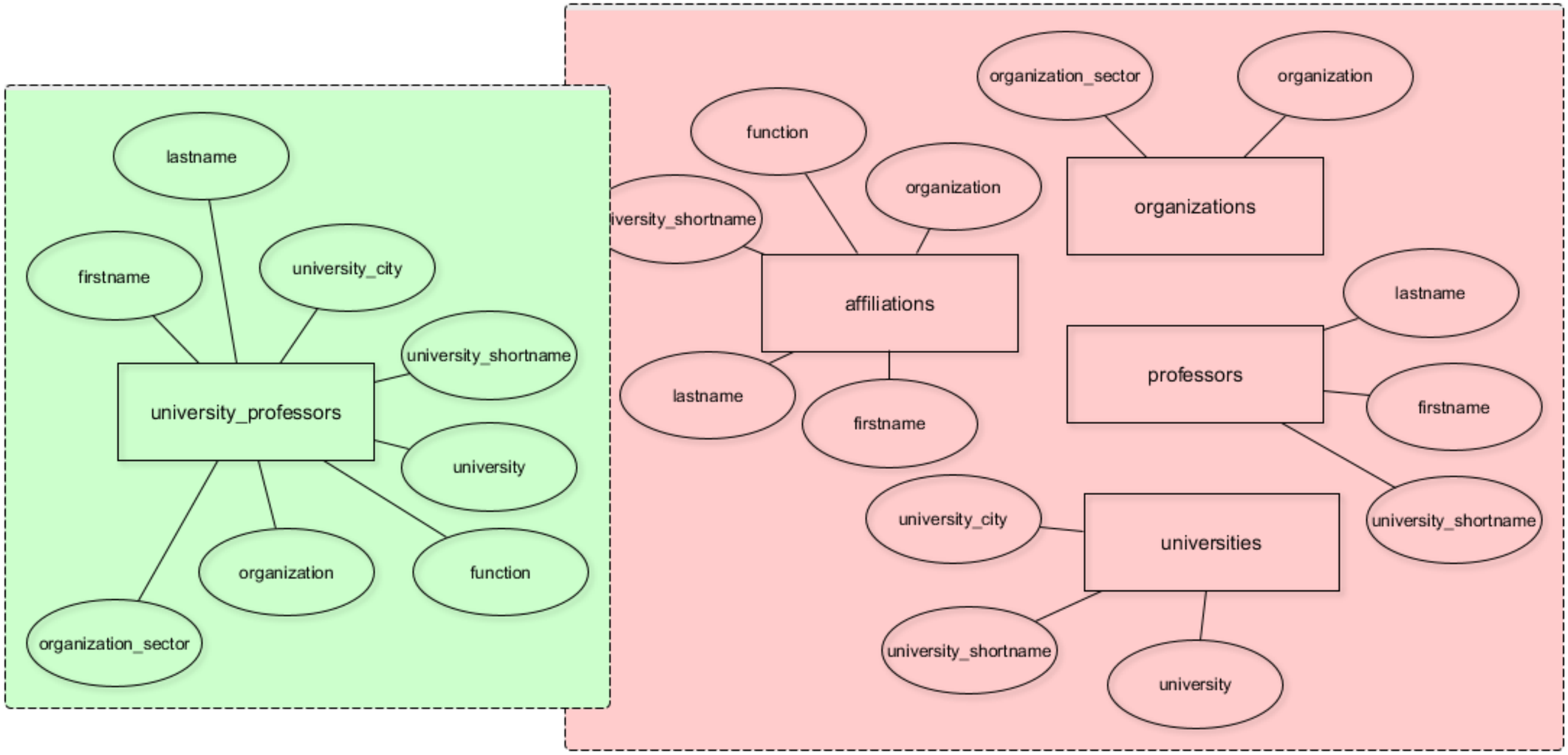
Timo Grossenbacher
Data Journalist

datacamp

The current database model



The current database model



Only store DISTINCT data in the new tables

```
SELECT COUNT(*)
FROM university_professors;
```

```
count
-----
1377
```

```
SELECT COUNT(DISTINCT organization)
FROM university_professors;
```

```
count
-----
1287
```

INSERT DISTINCT records INTO the new tables

```
INSERT INTO organizations
SELECT DISTINCT organization,
    organization_sector
FROM university_professors;
```

```
Output: INSERT 0 1287
```

```
INSERT INTO organizations
SELECT organization,
    organization_sector
FROM university_professors;
```

```
Output: INSERT 0 1377
```

The INSERT INTO statement

```
INSERT INTO table_name (column_a, column_b)
VALUES ("value_a", "value_b");
```

RENAME a COLUMN in affiliations

```
CREATE TABLE affiliations (
 firstname text,
 lastname text,
 university_shortname text,
 function text,
 organisation text
);
```

```
ALTER TABLE table_name
RENAME COLUMN old_name TO new_name;
```

DROP a COLUMN in affiliations

```
CREATE TABLE affiliations (
 firstname text,
 lastname text,
 university_shortname text,
 function text,
 organization text
);
```

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

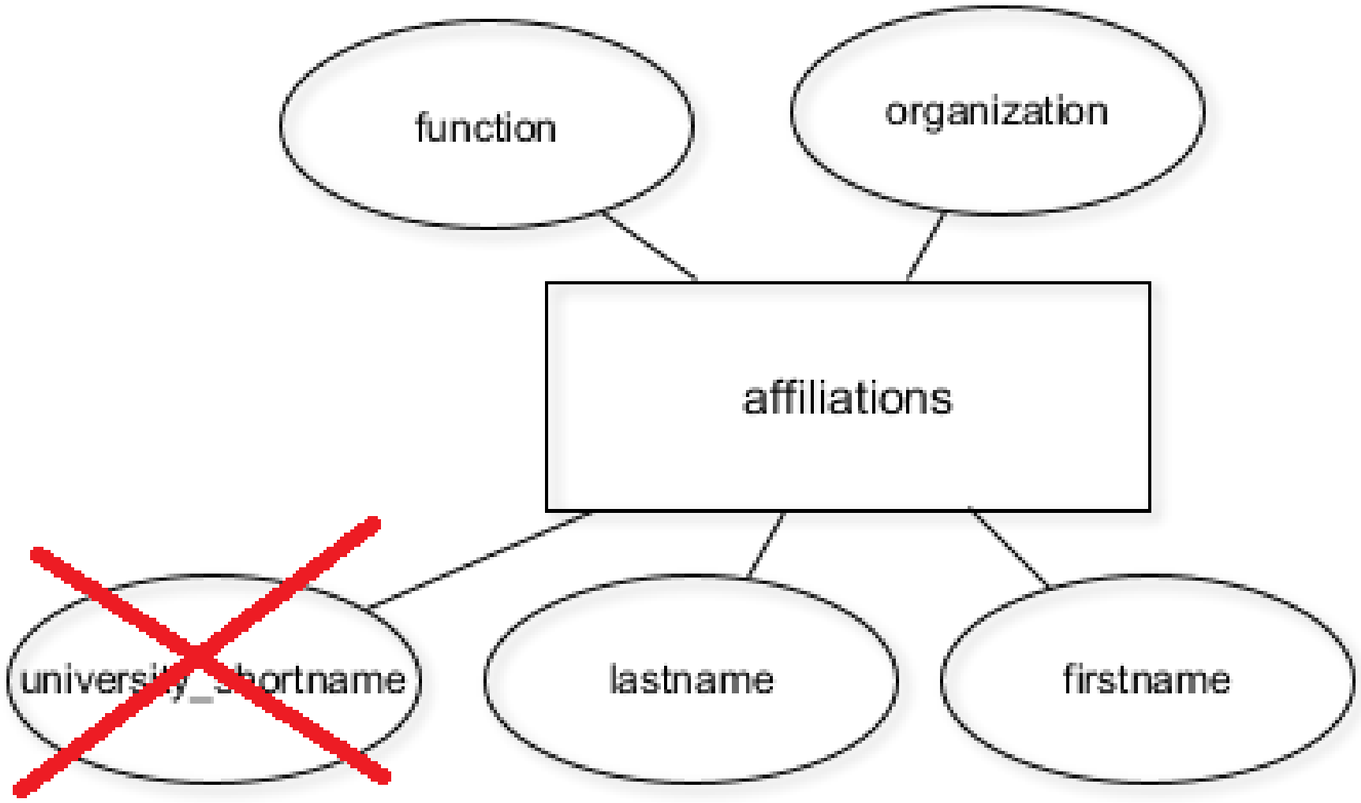
```
SELECT DISTINCT firstname, lastname,
    university_shortname
FROM university_professors
ORDER BY lastname;
```

```
-[ RECORD 1 ]--------+-------------
firstname            | Karl
lastname             | Aberer
university_shortname | EPF
-[ RECORD 2 ]--------+-------------
firstname            | Reza Shokrollah
lastname             | Abhari
university_shortname | ETH
-[ RECORD 3 ]--------+-------------
firstname            | Georges
lastname             | Abou Jaoudé
university_shortname | EPF
(truncated)

(551 records)
```

```
SELECT DISTINCT firstname, lastname
FROM university_professors
ORDER BY lastname;
```

```
-[ RECORD 1 ]----------------------
firstname | Karl
lastname  | Aberer
-[ RECORD 2 ]----------------------
firstname | Reza Shokrollah
lastname  | Abhari
-[ RECORD 3 ]----------------------
firstname | Georges
lastname  | Abou Jaoudé
(truncated)

(551 records)
```

A professor is uniquely identified by firstname, lastname only



Let's get to work!

INTRODUCTION TO RELATIONAL DATABASES IN SQL