



## INTRODUCTION TO RELATIONAL DATABASES IN SQL

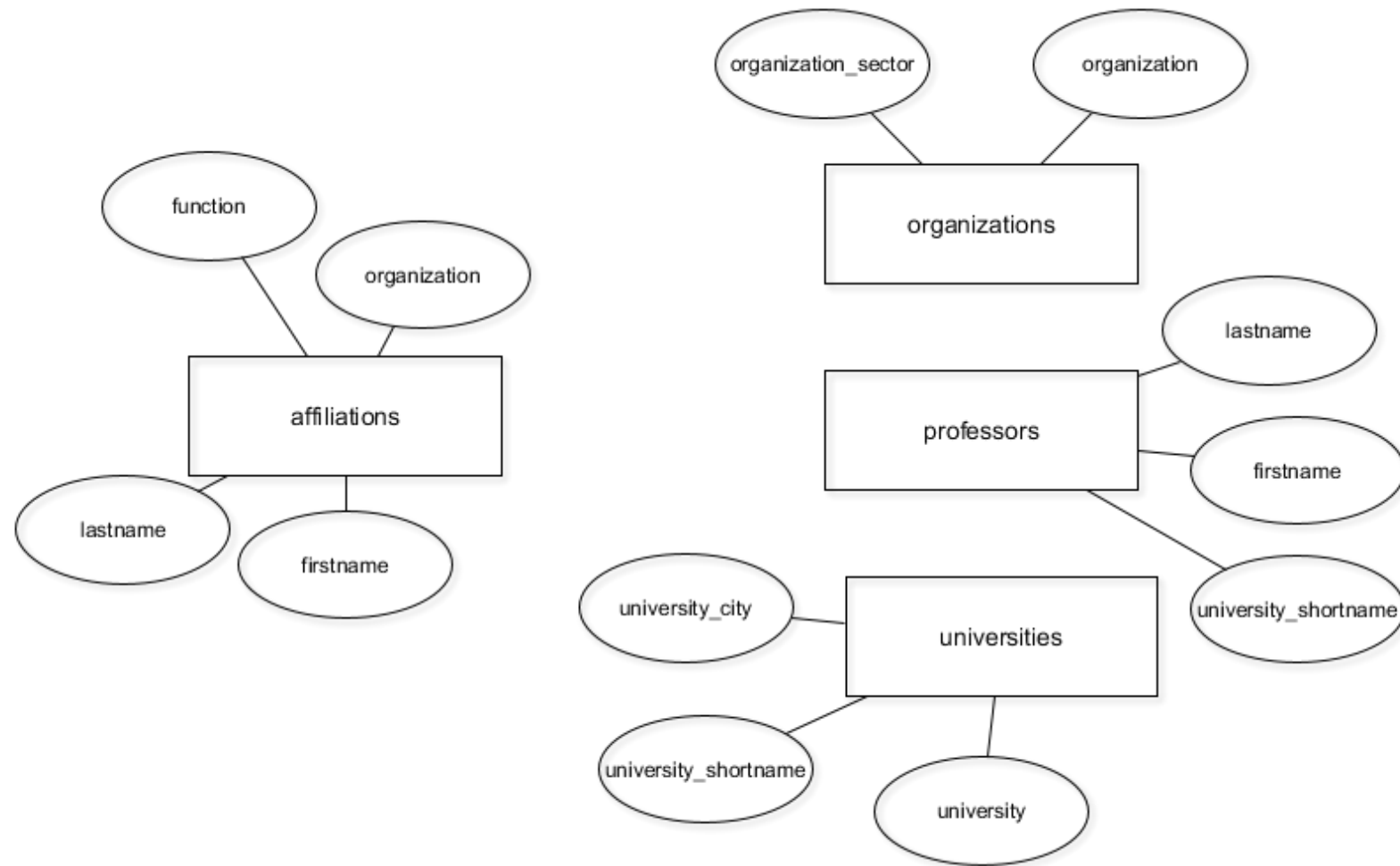
# Keys and superkeys

Timo Grossenbacher

Data Journalist

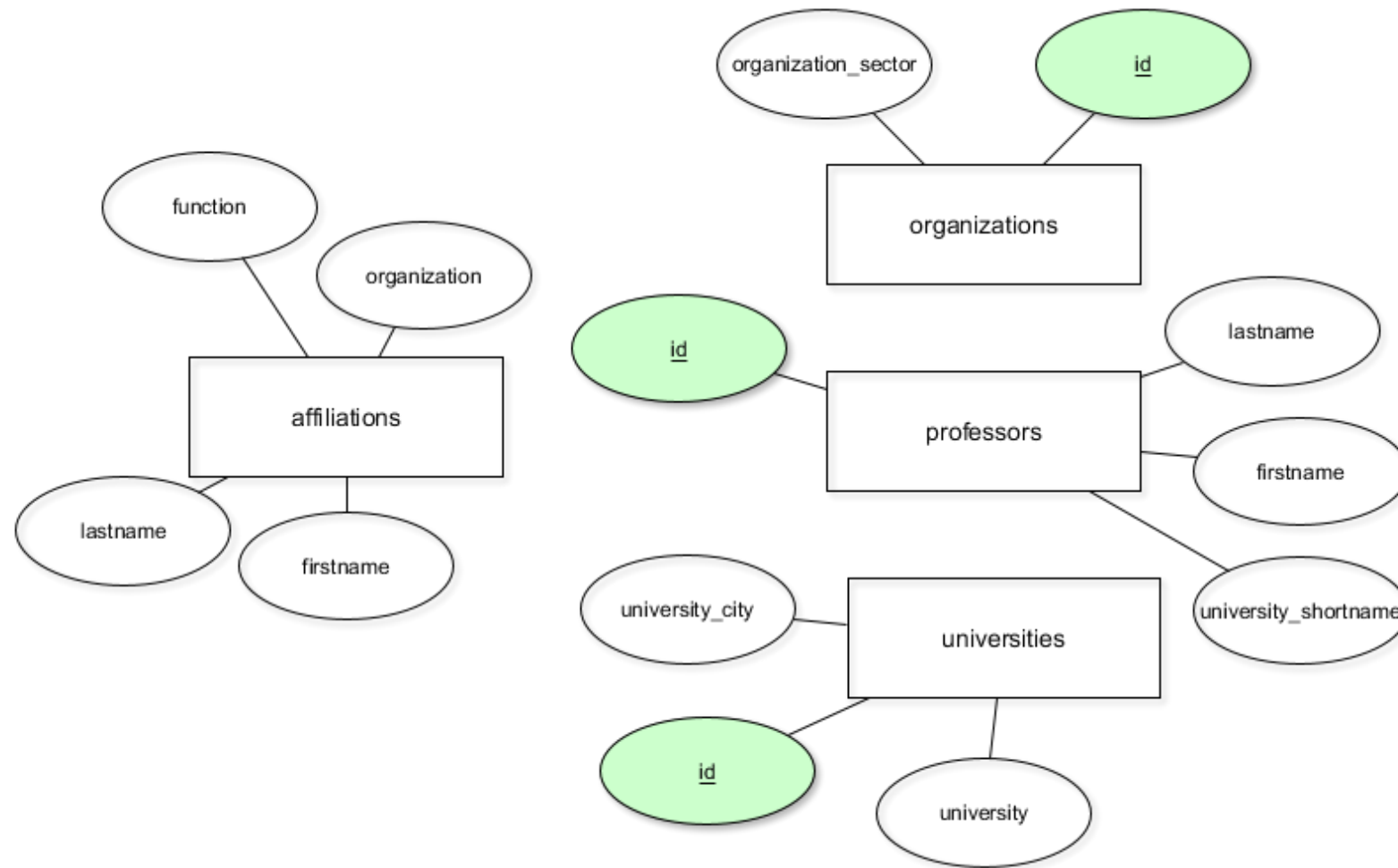


# The current database model





# The database model with primary keys





# What is a key?

- Attribute(s) that identify a record uniquely
- As long as attributes can be removed: **superkey**
- If no more attributes can be removed: minimal superkey or **key**

# An example

license_no	serial_no	make	model	year
Texas ABC-739	A69352	Ford	Mustang	2
Florida TVP-347	B43696	Oldsmobile	Cutlass	5
New York MPO-22	X83554	Oldsmobile	Delta	1
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	4
Texas RSK-629	U028365	Jaguar	XJS	4

SK1 = {license\_no, serial\_no, make, model, year}

SK2 = {license\_no, serial\_no, make, model}

SK3 = {make, model, year}, SK4 = {license\_no, serial\_no}, SK<sub>i</sub>, ..., SK<sub>n</sub>

*Adapted from Elmasri, Navathe (2011): Fundamentals of Database Systems, 6th Ed., Pearson*

## An example (contd.)

license_no	serial_no	make	model	year
Texas ABC-739	A69352	Ford	Mustang	2
Florida TVP-347	B43696	Oldsmobile	Cutlass	5
New York MPO-22	X83554	Oldsmobile	Delta	1
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	4
Texas RSK-629	U028365	Jaguar	XJS	4

$K1 = \{\text{license\_no}\}$ ;  $K2 = \{\text{serial\_no}\}$ ;  $K3 = \{\text{model}\}$ ;  $K4 = \{\text{make, year}\}$

- $K1$  to  $3$  only consist of one attribute
- Removing either "make" or "year" from  $K4$  would result in duplicates
- Only one candidate key can be the *chosen* key



## INTRODUCTION TO RELATIONAL DATABASES IN SQL

**Let's discover some keys!**



## INTRODUCTION TO RELATIONAL DATABASES IN SQL

# Primary keys

Timo Grossenbacher  
Data Journalist





# Primary keys

- One primary key per database table, chosen from candidate keys
- Uniquely identifies records, e.g. for referencing in other tables
- Unique and not-null constraints both apply
- Primary keys are time-invariant: choose columns wisely!



# Specifying primary keys

```
CREATE TABLE products (  
    product_no integer UNIQUE NOT NULL,  
    name text,  
    price numeric  
);
```

```
CREATE TABLE products (  
    product_no integer PRIMARY KEY,  
    name text,  
    price numeric  
);
```

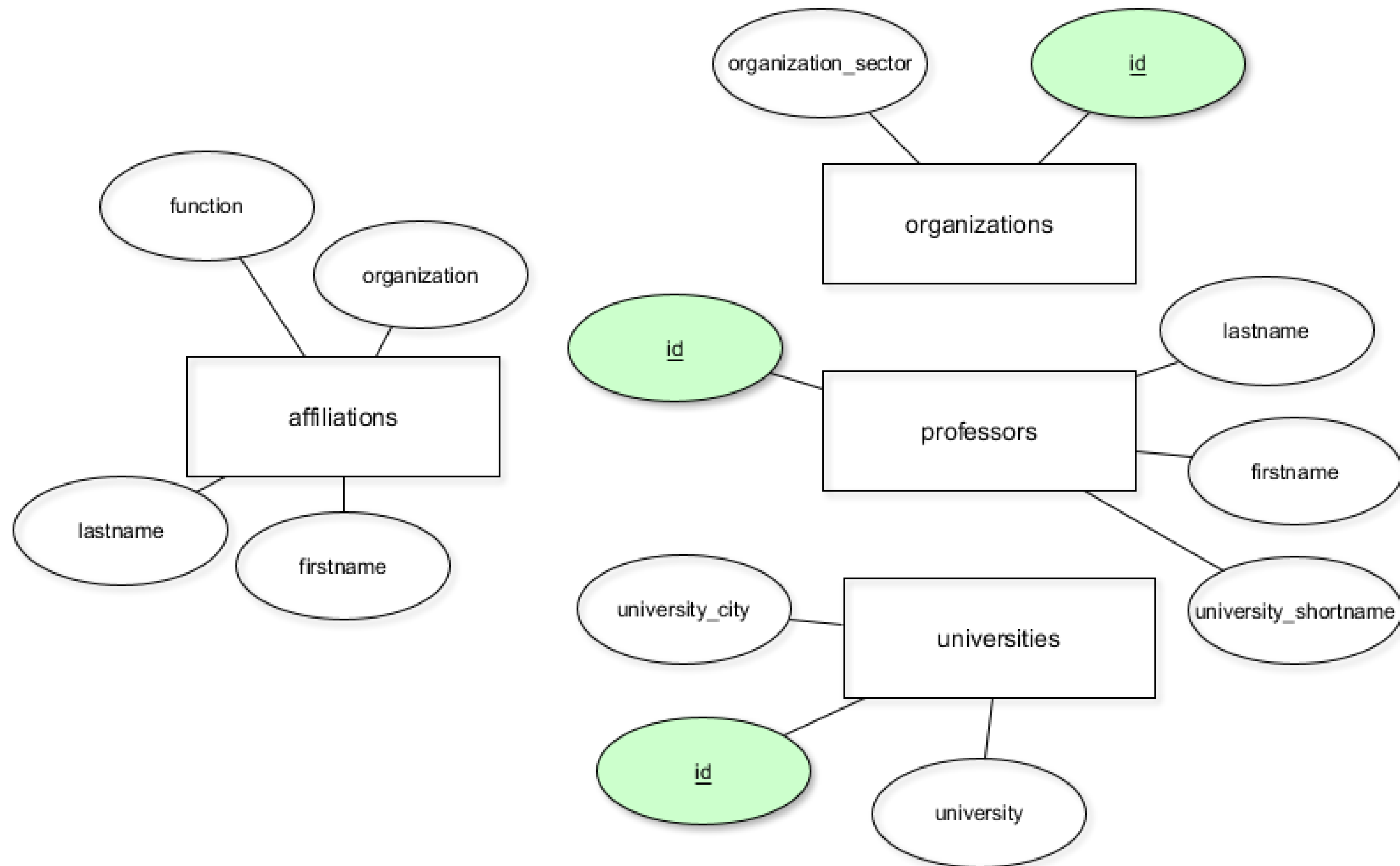
```
CREATE TABLE example (  
    a integer,  
    b integer,  
    c integer,  
    PRIMARY KEY (a, c)  
);
```

Taken from the [PostgreSQL documentation](#).



# Specifying primary keys (contd.)

```
ALTER TABLE table_name  
ADD CONSTRAINT some_name PRIMARY KEY (column_name)
```





## INTRODUCTION TO RELATIONAL DATABASES IN SQL

**Let's practice!**



## INTRODUCTION TO RELATIONAL DATABASES IN SQL

# Surrogate keys

Timo Grossenbacher  
Data Journalist



# Surrogate keys

- Primary keys should be built from as few columns as possible
- Primary keys should never change over time



# An example

license_no	serial_no	make	model	color
Texas ABC-739	A69352	Ford	Mustang	blue
Florida TVP-347	B43696	Oldsmobile	Cutlass	black
New York MPO-22	X83554	Oldsmobile	Delta	silver
California 432-TFY	C43742	Mercedes	190-D	champagne
California RSK-629	Y82935	Toyota	Camry	red
Texas RSK-629	U028365	Jaguar	XJS	blue

make	model	color
Ford	Mustang	blue
Oldsmobile	Cutlass	black
Oldsmobile	Delta	silver
Mercedes	190-D	champagne
Toyota	Camry	red
Jaguar	XJS	blue



# Adding a surrogate key with serial data type

```
ALTER TABLE cars
ADD COLUMN id serial PRIMARY KEY;
```

```
INSERT INTO cars
VALUES ('Volkswagen', 'Blitz', 'black');
```

make	model	color	id
Ford	Mustang	blue	1
Oldsmobile	Cutlass	black	2
Oldsmobile	Delta	silver	3
Mercedes	190-D	champagne	4
Toyota	Camry	red	5
Jaguar	XJS	blue	6
Volkswagen	Blitz	black	7



# Adding a surrogate key with serial data type (contd.)

```
INSERT INTO cars  
VALUES ('Opel', 'Astra', 'green', 1);
```

```
duplicate key value violates unique constraint "id_pkey"  
DETAIL:  Key (id)=(1) already exists.
```

- "id" uniquely identifies records in the table – useful for referencing!

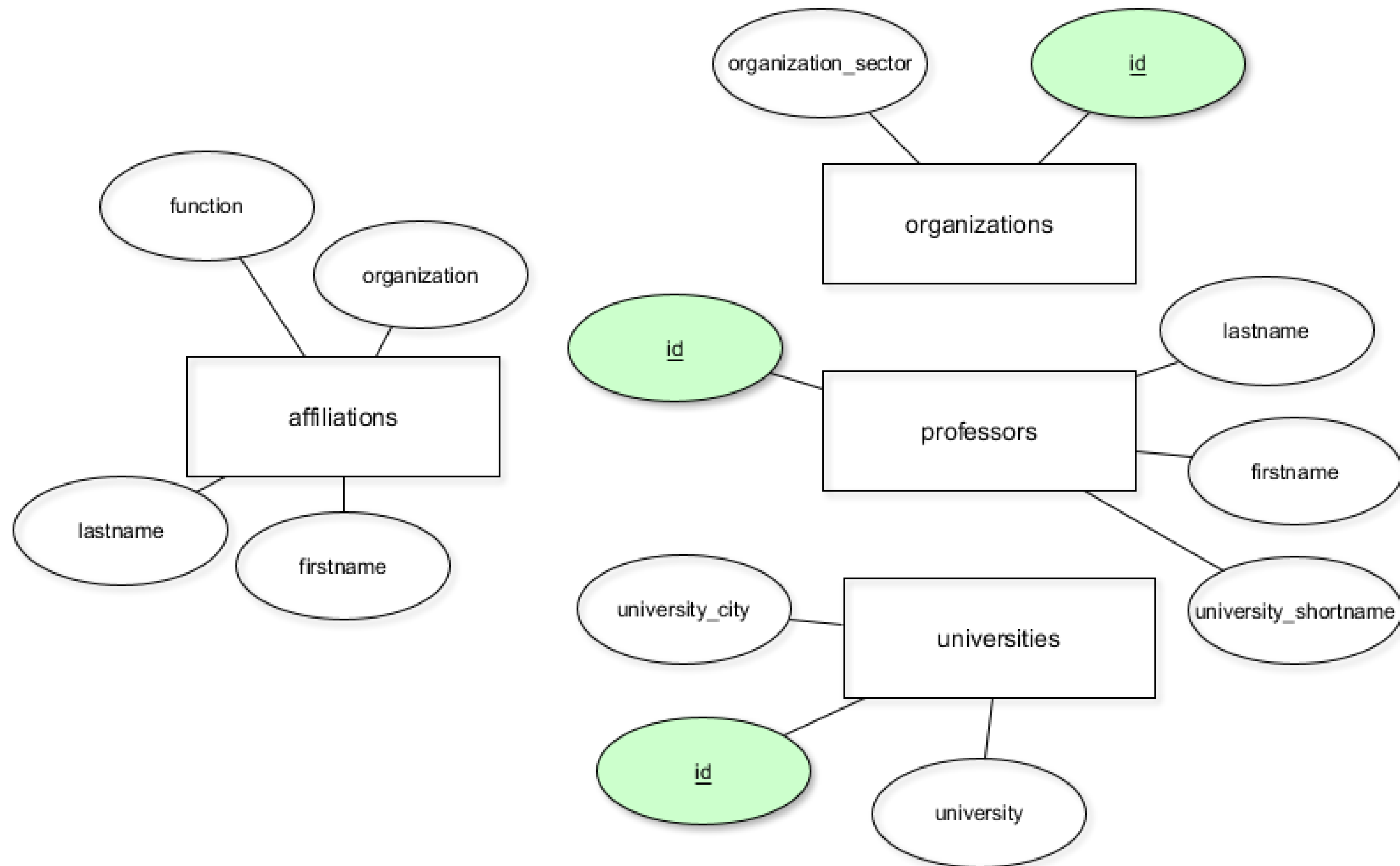


# Another type of surrogate key

```
ALTER TABLE table_name
ADD COLUMN column_c varchar(256);

UPDATE table_name
SET column_c = CONCAT(column_a, column_b);

ALTER TABLE table_name
ADD CONSTRAINT pk PRIMARY KEY (column_c);
```





## INTRODUCTION TO RELATIONAL DATABASES IN SQL

**Let's try this!**