

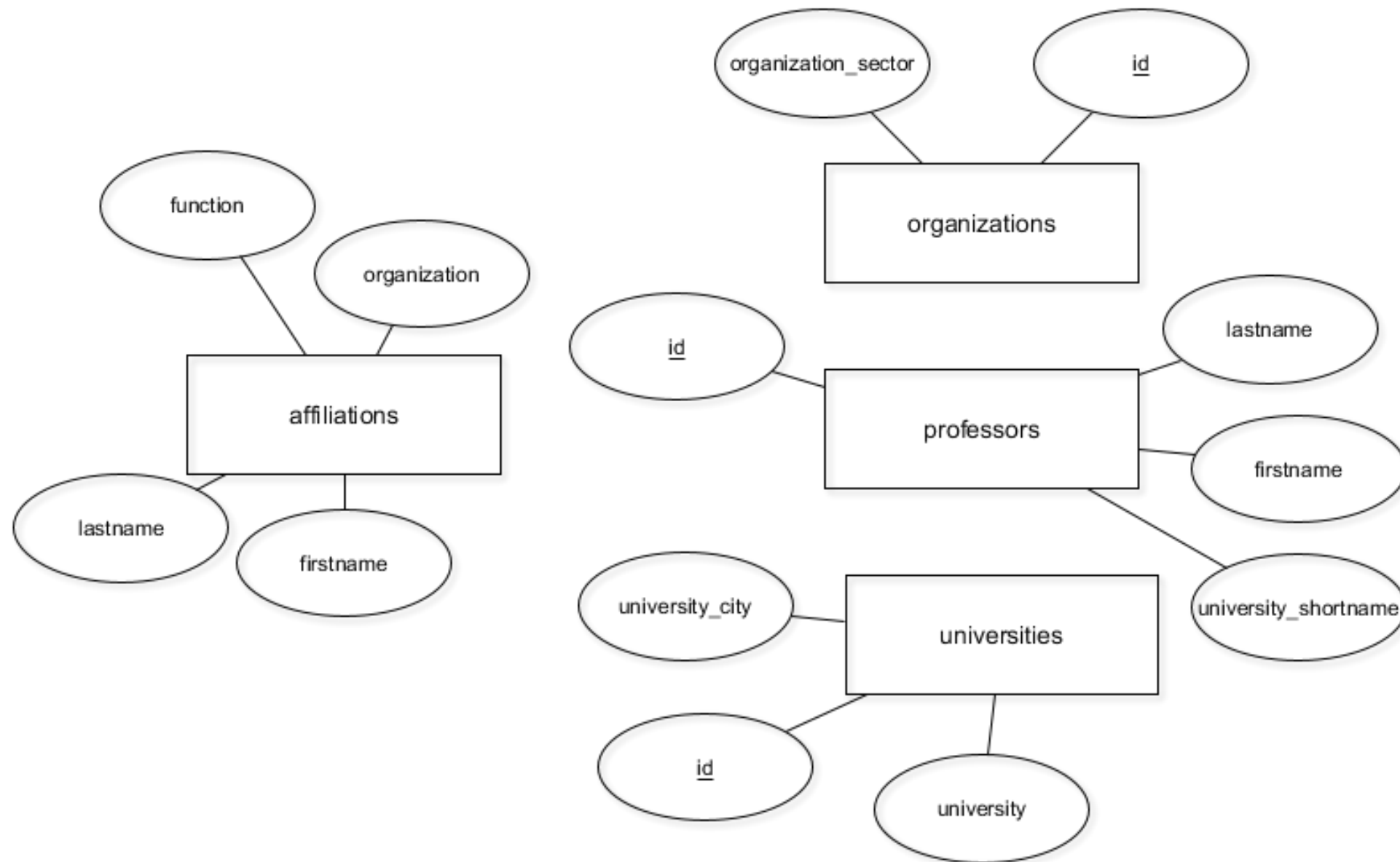
# Model 1:N relationships with foreign keys

INTRODUCTION TO RELATIONAL DATABASES IN SQL

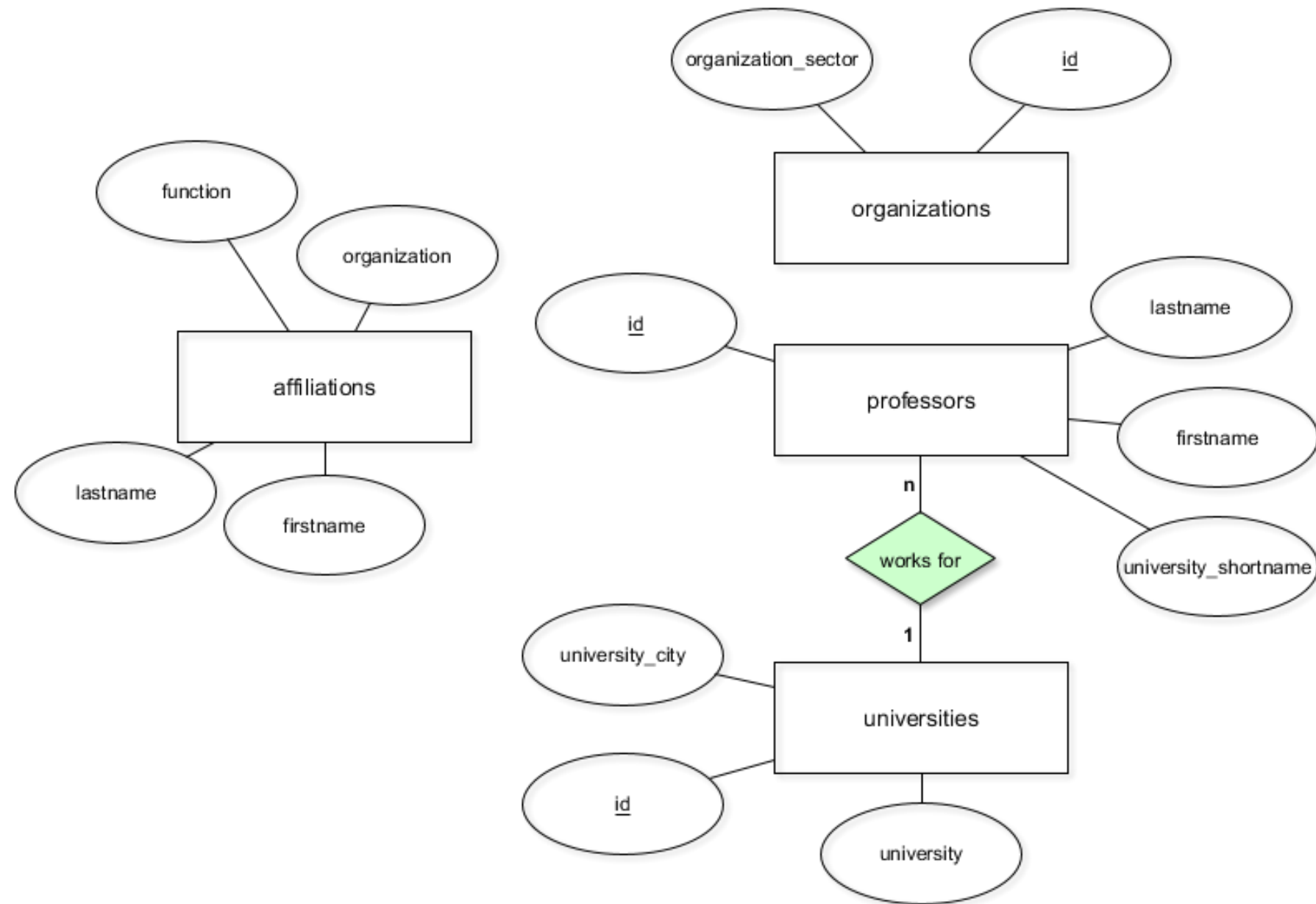
SQL

**Timo Grossenbacher**  
Data Journalist

# The current database model



# The next database model



# Implementing relationships with foreign keys

- A foreign key (FK) points to the primary key (PK) of another table
- Domain of FK must be equal to domain of PK
- Each value of FK must exist in PK of the other table (FK constraint or "referential integrity")
- FKs are not actual *keys*

# A query

```
SELECT * FROM professors LIMIT 8;
```

id	firstname	lastname	university_s..
--+-+-----+-----			
1	Karl	Aberer	EPF
2	Reza Shokrollah	Abhari	ETH
3	Georges	Abou Jaoudé	EPF
4	Hugues	Abriel	UBE
5	Daniel	Aebersold	UBE
6	Marcelo	Aebi	ULA
7	Christoph	Aebi	UBE
8	Patrick	Aebischer	EPF

```
SELECT * FROM universities;
```

id	university	university_city
<hr />--+-+-----+-----		
EPF	ETH Lausanne	Lausanne
ETH	ETH Zürich	Zurich
UBA	Uni Basel	Basel
UBE	Uni Bern	Bern
UFR	Uni Freiburg	Fribourg
UGE	Uni Genf	Geneva
ULA	Uni Lausanne	Lausanne
UNE	Uni Neuenburg	Neuchâtel
USG	Uni St. Gallen	Saint Gallen
USI	USI Lugano	Lugano
UZH	Uni Zürich	Zurich

# Specifying foreign keys

```
CREATE TABLE manufacturers (  
  name varchar(255) PRIMARY KEY);
```

```
INSERT INTO manufacturers  
VALUES ('Ford'), ('VW'), ('GM');
```

```
CREATE TABLE cars (  
  model varchar(255) PRIMARY KEY,  
  manufacturer_name varchar(255) REFERENCES manufacturers (name));
```

```
INSERT INTO cars  
VALUES ('Ranger', 'Ford'), ('Beetle', 'VW');
```

```
-- Throws an error!
```

```
INSERT INTO cars  
VALUES ('Tundra', 'Toyota');
```

# Specifying foreign keys to existing tables

```
ALTER TABLE a  
ADD CONSTRAINT a_fkey FOREIGN KEY (b_id) REFERENCES b (id);
```

# Let's implement this!

INTRODUCTION TO RELATIONAL DATABASES IN SQL



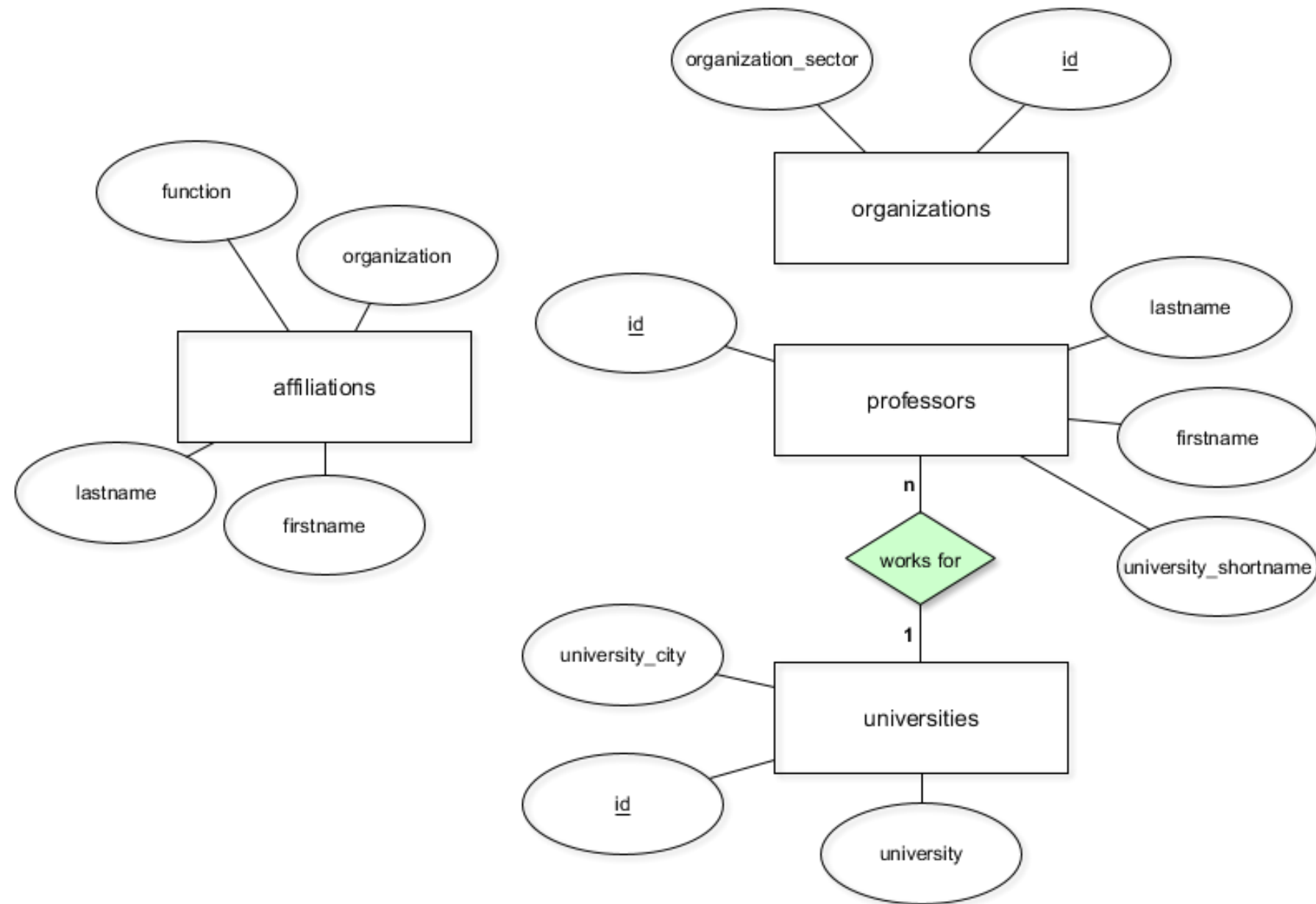
# Model more complex relationships

INTRODUCTION TO RELATIONAL DATABASES IN SQL

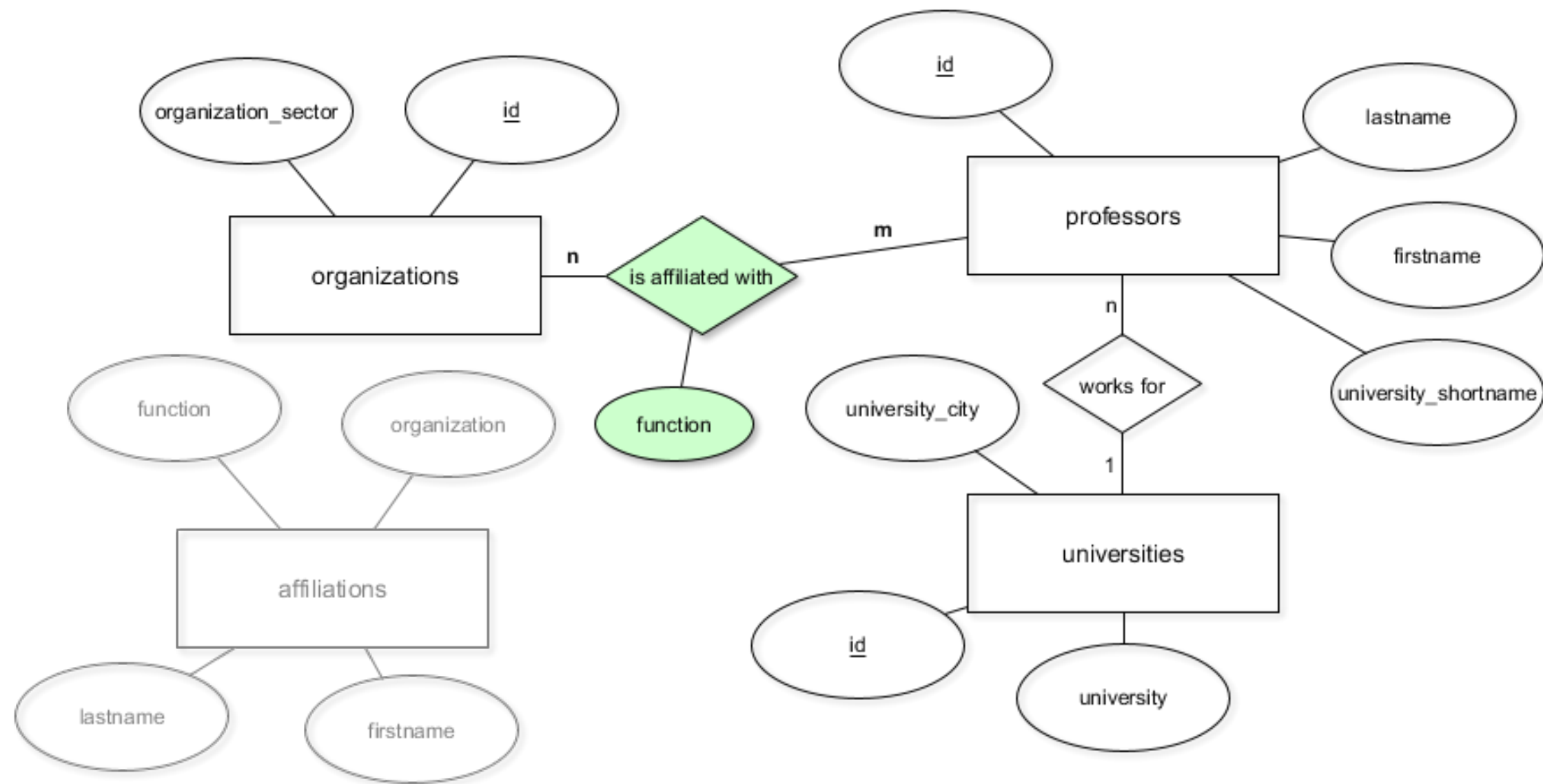


**Timo Grossenbacher**  
Data Journalist

# The current database model



# The final database model



# How to implement N:M-relationships

- Create a table
- Add foreign keys for every connected table
- Add additional attributes

```
CREATE TABLE affiliations (  
  professor_id integer REFERENCES professors (id),  
  organization_id varchar(256) REFERENCES organizations (id),  
  function varchar(256)  
);
```

- No primary key!
- Possible PK = {professor\_id, organization\_id, function}

# Time to implement this!

INTRODUCTION TO RELATIONAL DATABASES IN SQL

# Referential integrity

INTRODUCTION TO RELATIONAL DATABASES IN SQL



**Timo Grossenbacher**

Data Journalist

# Referential integrity

- *A record referencing another table must refer to an existing record in that table*
- Specified between two tables
- Enforced through foreign keys

# Referential integrity violations

Referential integrity from table A to table B is violated...

- ...if a record in table B that is referenced from a record in table A is deleted.
- ...if a record in table A referencing a non-existing record from table B is inserted.
- Foreign keys prevent violations!



# Dealing with violations

```
CREATE TABLE a (  
  id integer PRIMARY KEY,  
  column_a varchar(64),  
  ...,  
  b_id integer REFERENCES b (id) ON DELETE NO ACTION  
);
```

```
CREATE TABLE a (  
  id integer PRIMARY KEY,  
  column_a varchar(64),  
  ...,  
  b_id integer REFERENCES b (id) ON DELETE CASCADE  
);
```

# Dealing with violations, contd.

## ON DELETE...

- ...NO ACTION: Throw an error
- ...CASCADE: Delete all referencing records
- ...RESTRICT: Throw an error
- ...SET NULL: Set the referencing column to NULL
- ...SET DEFAULT: Set the referencing column to its default value

# Let's look at some examples!

INTRODUCTION TO RELATIONAL DATABASES IN SQL

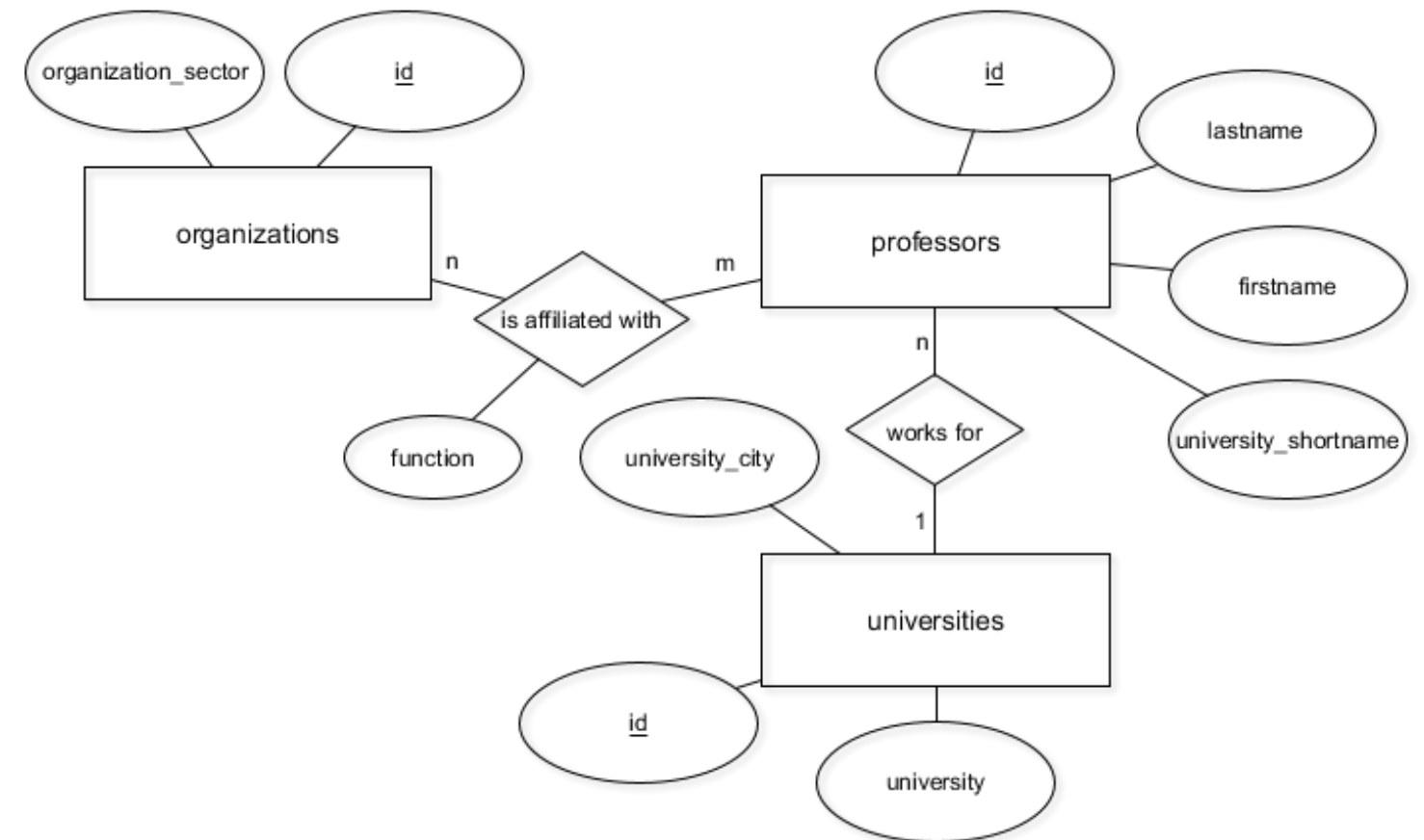
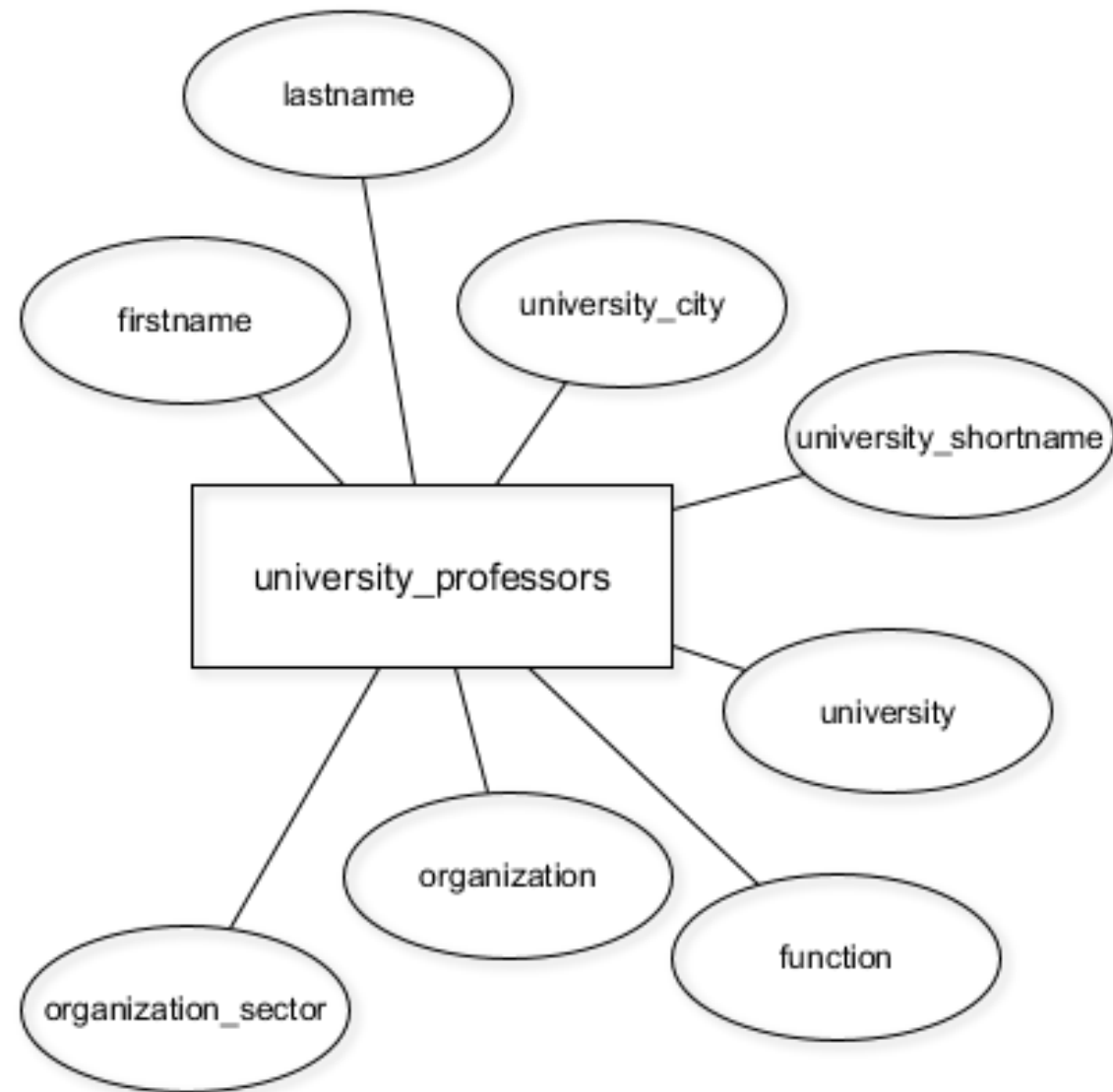
# Roundup

INTRODUCTION TO RELATIONAL DATABASES IN SQL



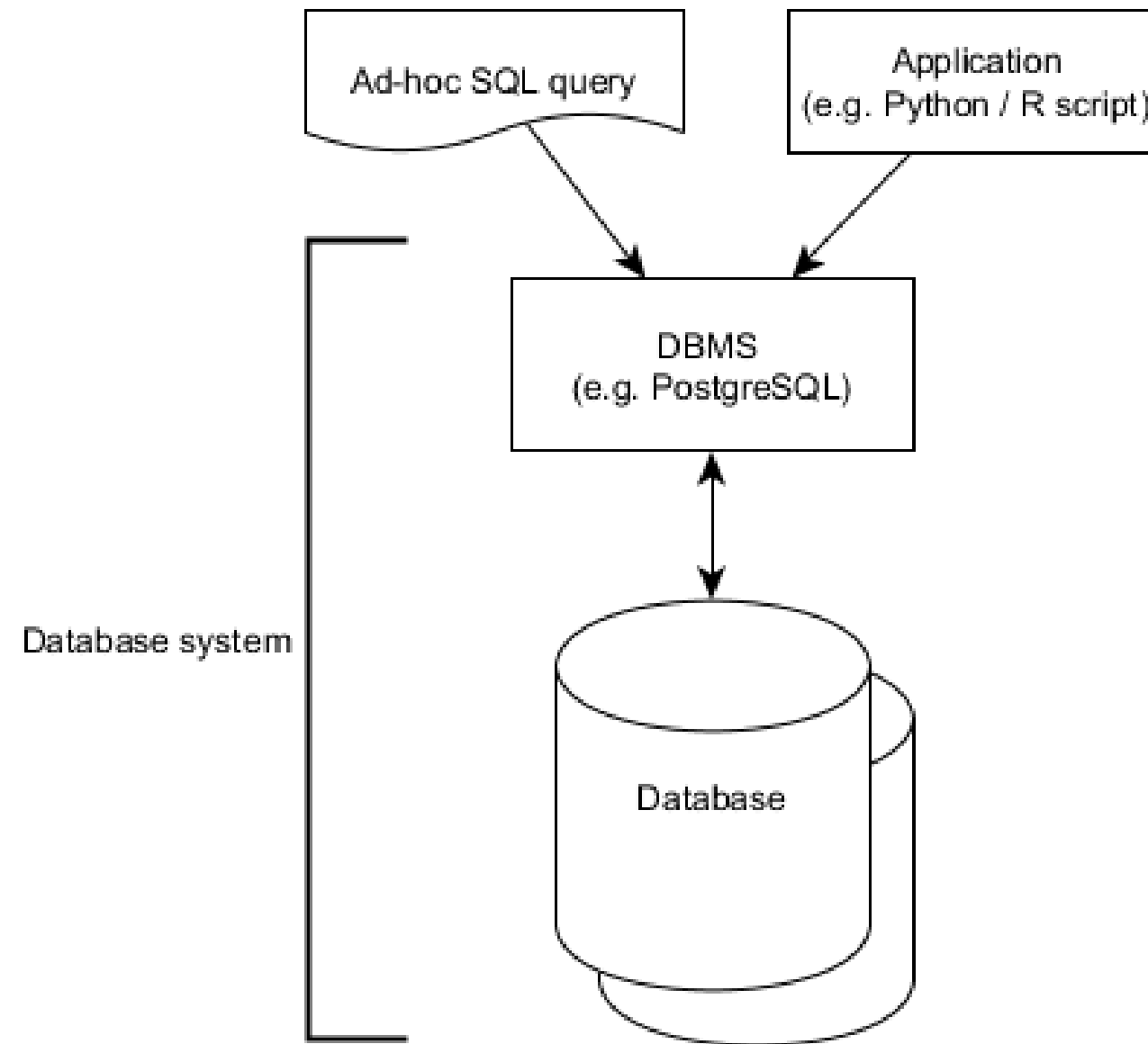
**Timo Grossenbacher**  
Data Journalist

# How you've transformed the database

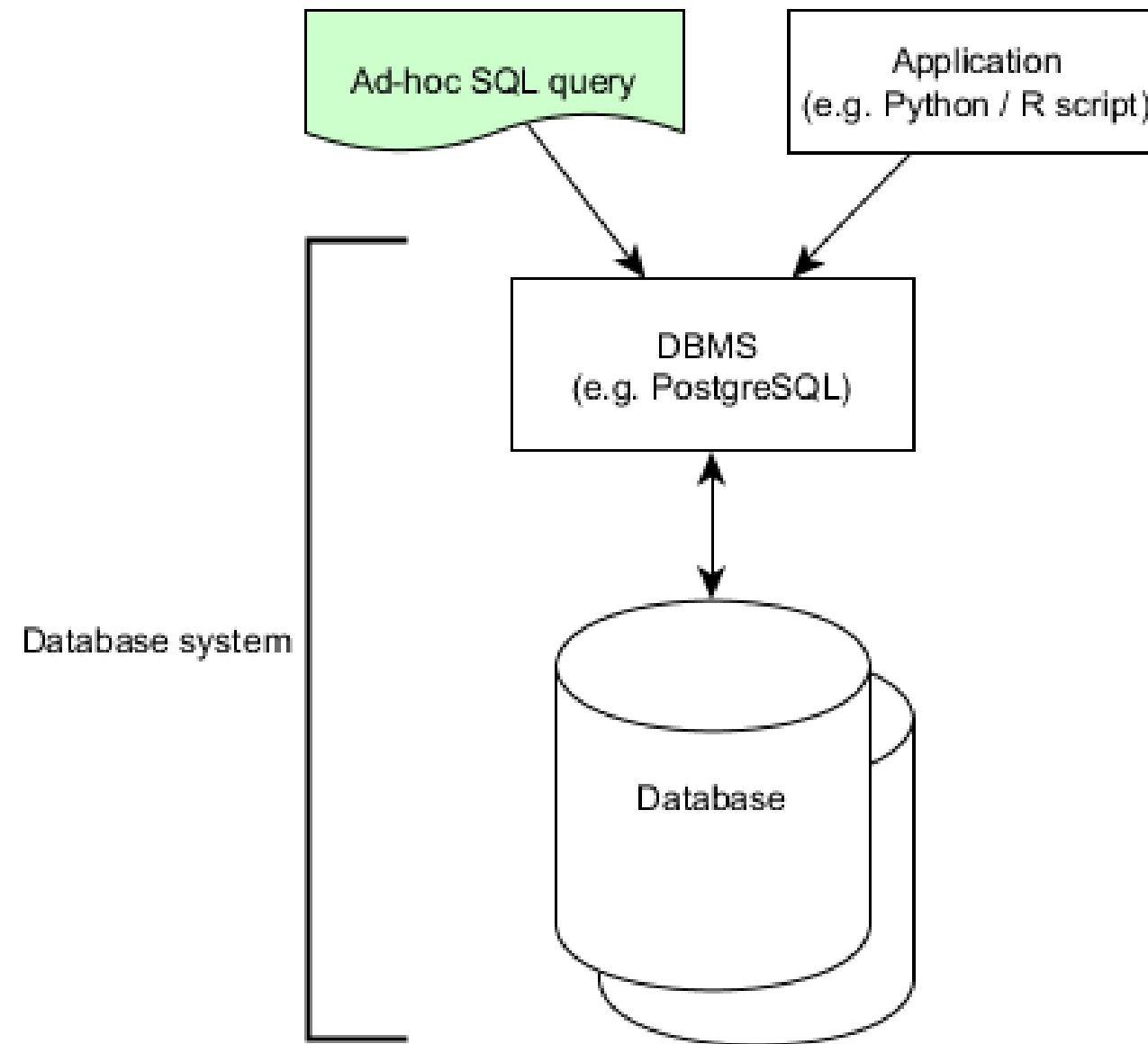


- Column data types
- Key constraints
- Relationships between tables

# The database ecosystem



# The database ecosystem



# Thank you!

INTRODUCTION TO RELATIONAL DATABASES IN SQL