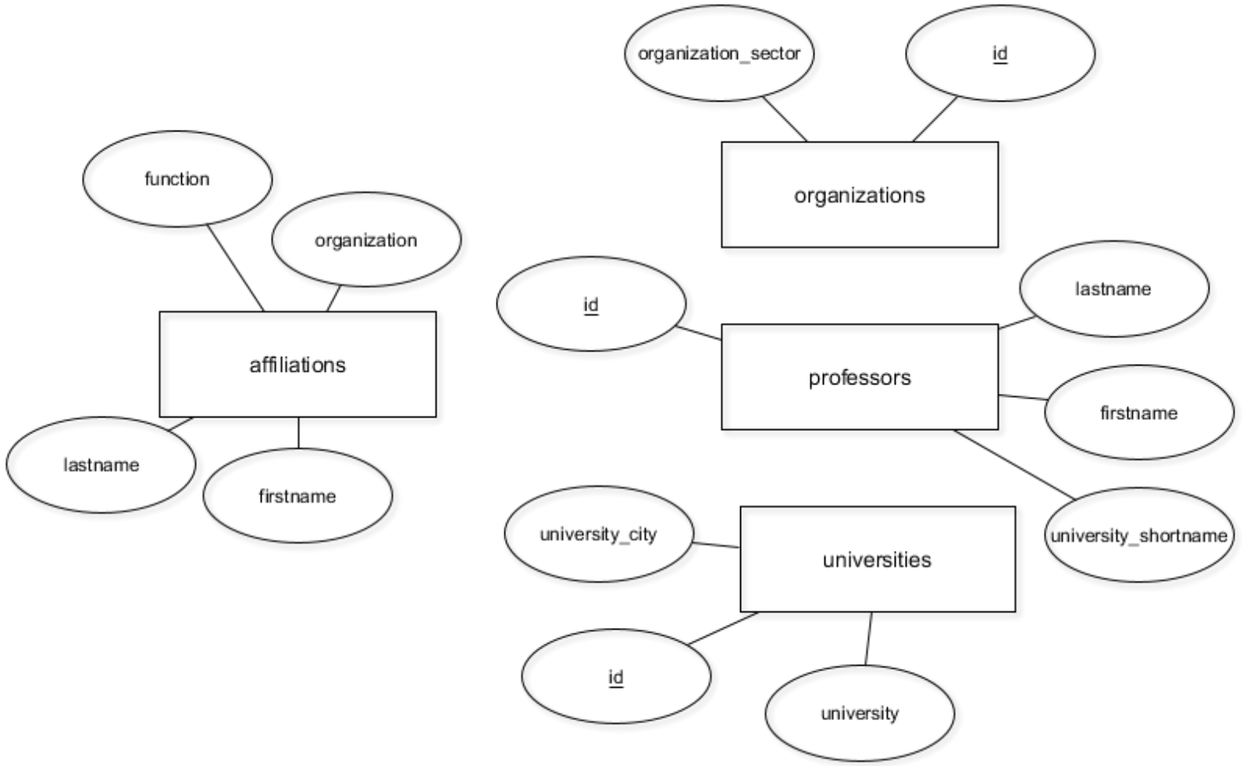
INTRODUCTION TO RELATIONAL DATABASES IN SQL

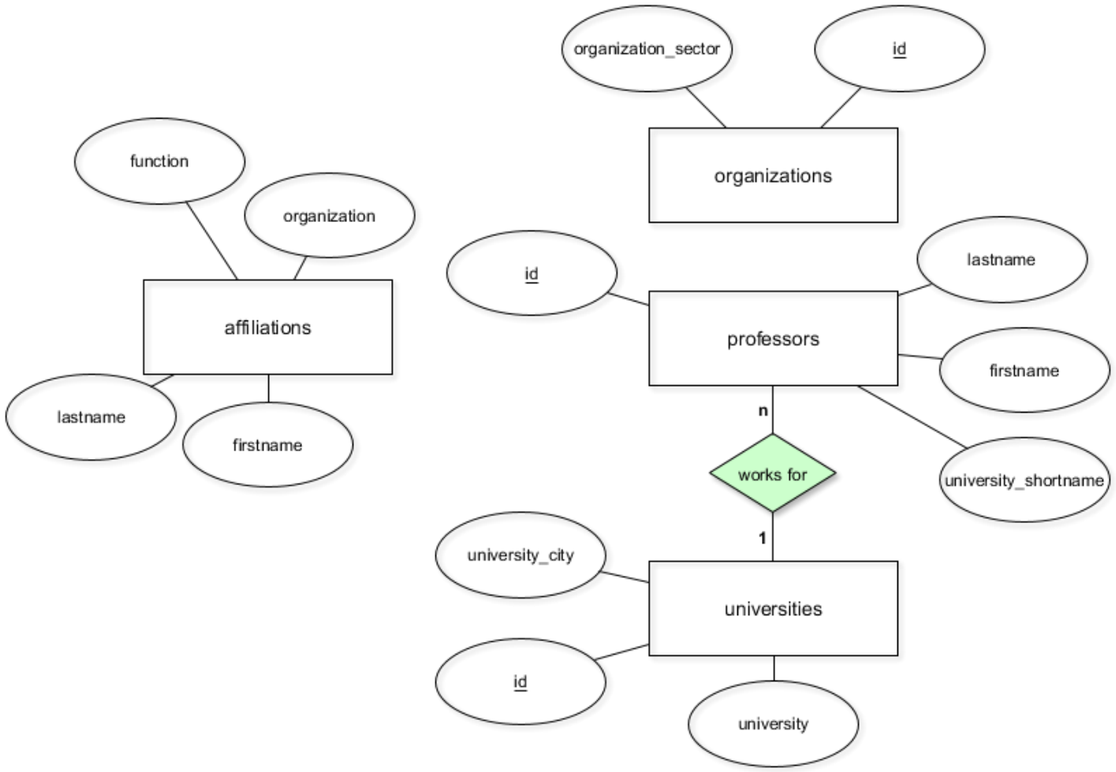
Model 1:N relationships with foreign keys

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* duplicates and NULL values are allowed

```
SELECT * FROM professors LIMIT 8;
 id |    firstname    |  lastname   | university_shortname
----+-----------------+-------------+----------------------
  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
-----+----------------+-----------------
 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 integer REFERENCES manufacturers (name)
);

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

As each car is produced by a certain manufacturer, it makes sense to also add a foreign key to this table. We do that by writing the “REFERENCES” keyword, followed by the referenced table and its primary key in brackets.
From now on, only cars with valid and existing manufacturers may be entered into that table.

```
-- 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);
```

Table a should now refer to table b, via b_id (table a), which points to id (table b).
a_fkey is a constraint name you can choose on your own.

INTRODUCTION TO RELATIONAL DATABASES IN SQL

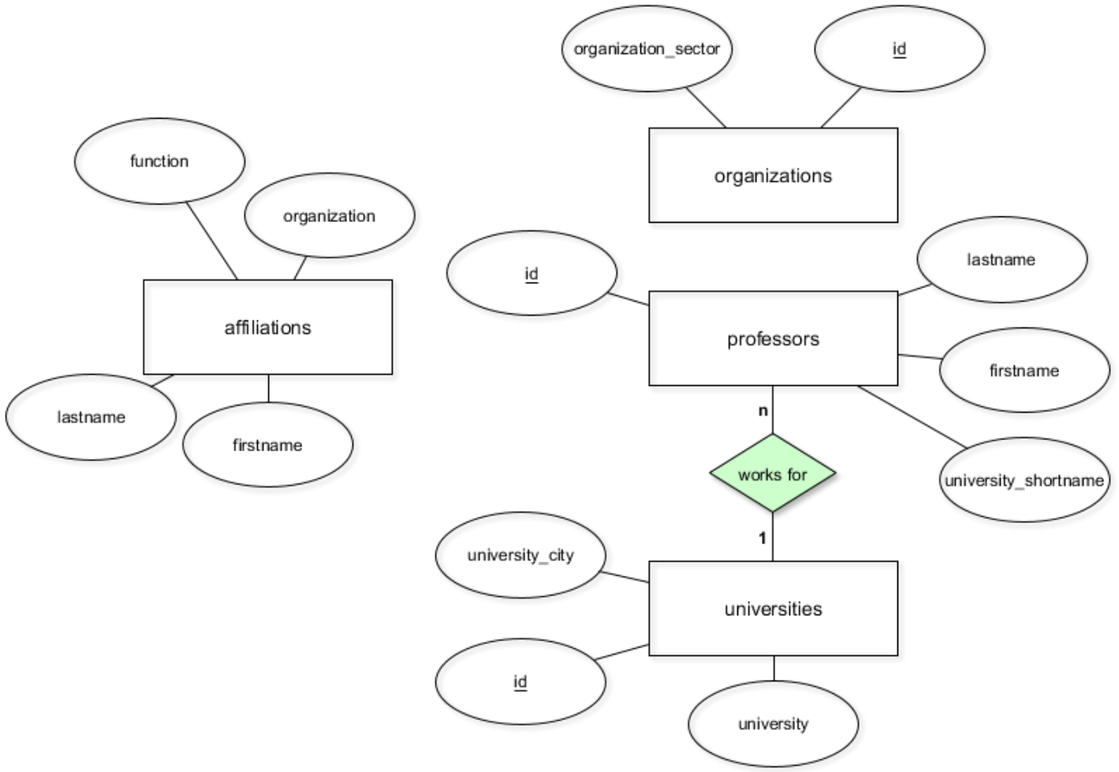
Let's implement this!

INTRODUCTION TO RELATIONAL DATABASES IN SQL

Model more complex relationships

Timo Grossenbacher
Data Journalist

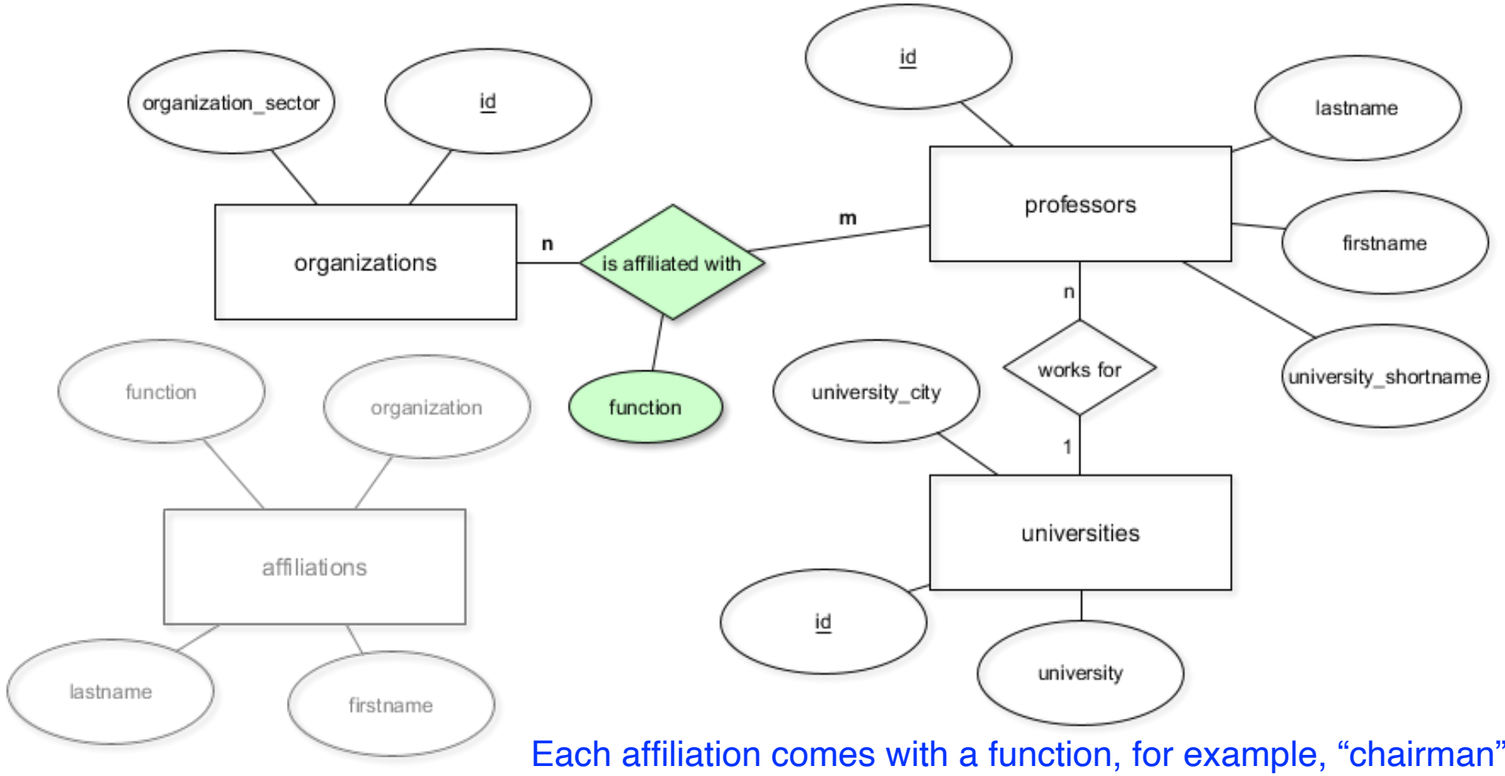
The current database model



- 1:N-relationships are implemented with one foreign key

The final database model

an N:M relationship: a professor can be affiliated with more than one organization and vice versa



Each affiliation comes with a function, for example, “chairman”.
You’ll still have four tables: three for the entities “professors”, “universities” and “organizations”, and one for the N:M-relationship between “professors” and “organizations”.

How to implement N:M-relationships

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

contains two foreign keys that point to both connected entities

function

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

- No primary key!
- Possible PK = {professor_id, organization_id, function}

INTRODUCTION TO RELATIONAL DATABASES IN SQL

Here's a way to update columns of a table based on values in another table:

```
UPDATE table_a
SET column_to_update = table_b.column_to_update_from
FROM table_b
WHERE condition1 AND condition2 AND ...;
```

This query does the following:
1. For each row in table_a, find the corresponding row in table_b where condition1, condition2, etc., are met.
2. Set the value of column_to_update to the value of column_to_update_from (from that corresponding row).
The conditions usually compare other columns of both tables, e.g. table_a.some_column = table_b.some_column. Of course, this query only makes sense if there is only one matching row in table_b.

Time to implement this!

INTRODUCTION TO RELATIONAL DATABASES IN SQL

Referential integrity

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

INTRODUCTION TO RELATIONAL DATABASES IN SQL

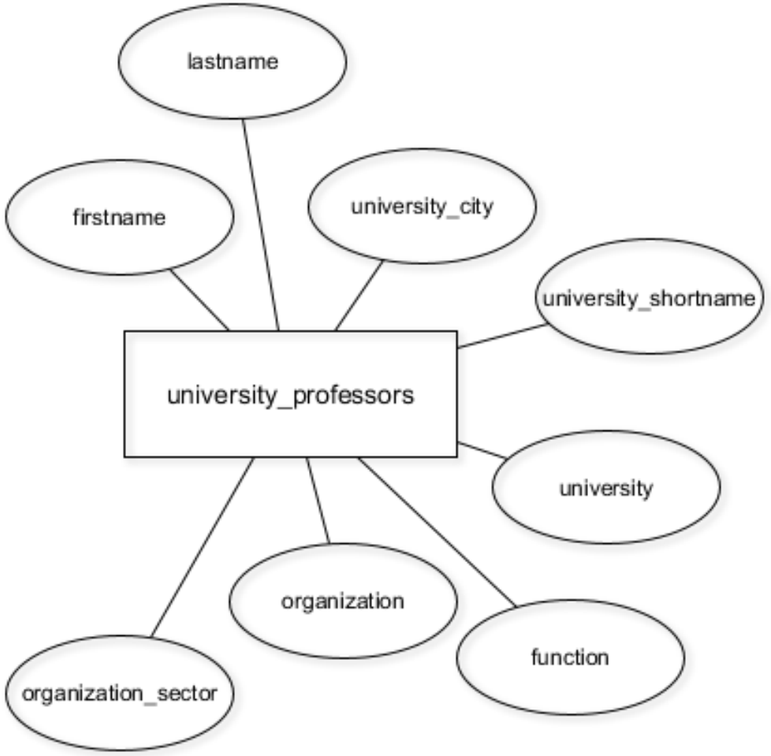
Let's look at some examples!

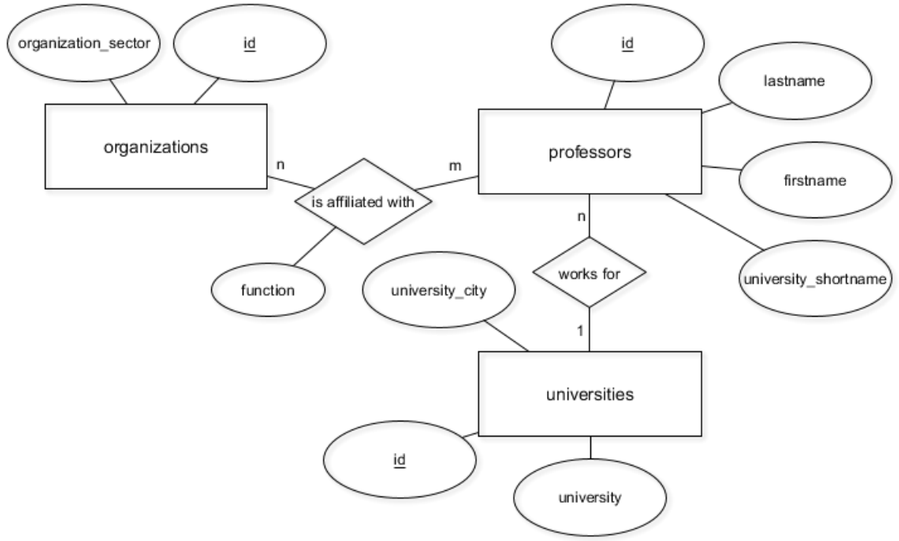
INTRODUCTION TO RELATIONAL DATABASES IN SQL

Roundup

Timo Grossenbacher
Data Journalist

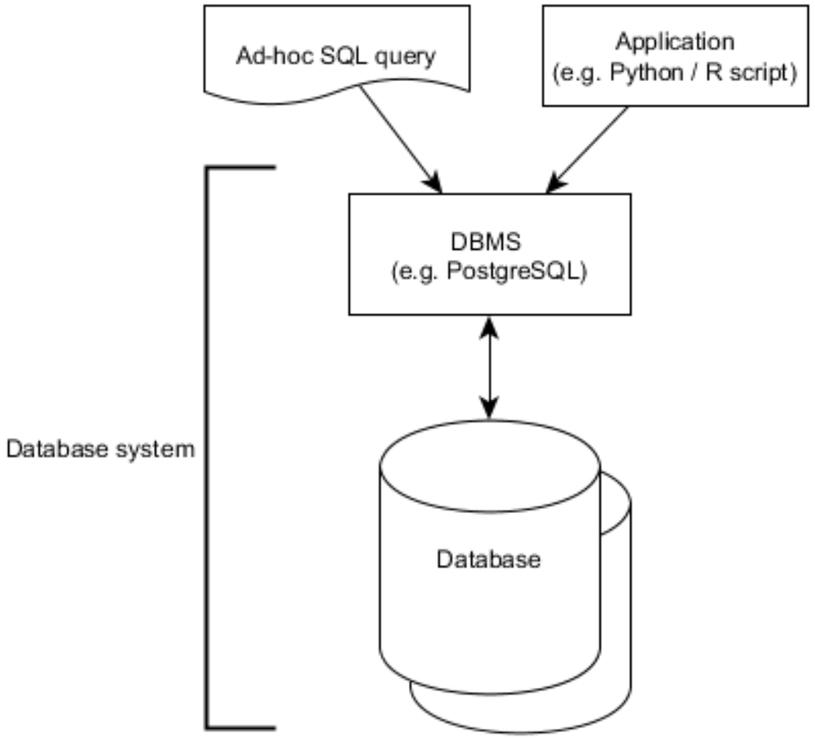
How you've transformed the database



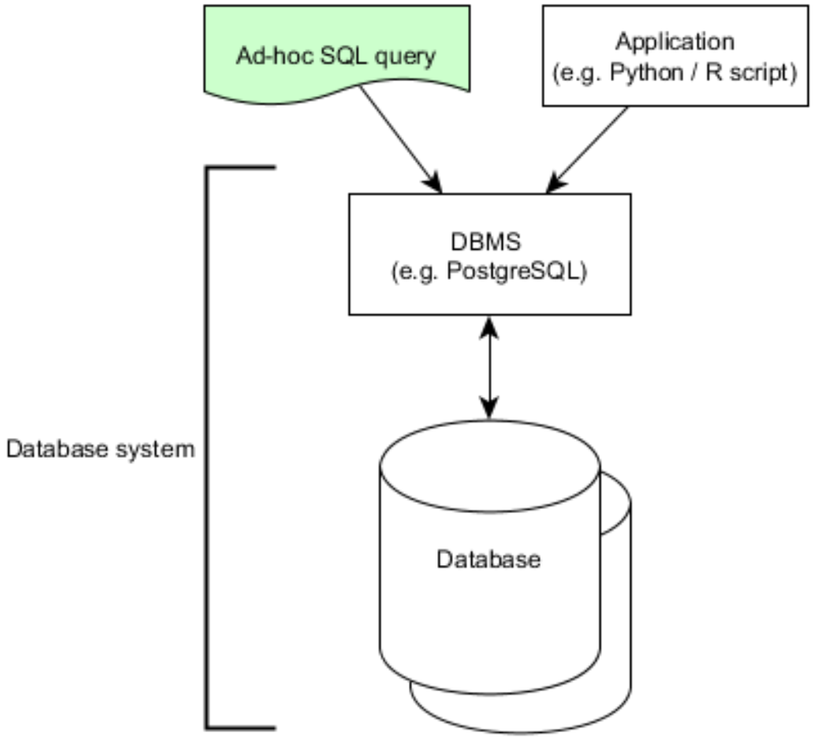


- Column data types
- Key constraints
- Relationships between tables

The database ecosystem



The database ecosystem



INTRODUCTION TO RELATIONAL DATABASES IN SQL

Thank you!