-- Query the right table in information\_schema

SELECT table\_name

FROM information\_schema.tables

-- Specify the correct table\_schema value

WHERE table\_schema = 'public';

-- Query the right table in information\_schema to get columns

SELECT column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'university\_professors' AND table\_schema = 'public';

-- Query the first five rows of our table

select \*

from university\_professors

LIMIT 5;

==

-- Create a table for the professors entity type

CREATE TABLE professors (

firstname text,

lastname text

);

-- Print the contents of this table

SELECT \*

FROM professors;

-- Create a table for the universities entity type

create table universities (

university\_shortname text,

university text,

university\_city text

);

-- Print the contents of this table

SELECT \*

FROM universities

==

-- Add the university\_shortname column

alter table professors

add column university\_shortname text;

-- Print the contents of this table

SELECT \*

FROM professors

==

-- Rename the organisation column

ALTER TABLE affiliations

RENAME column organisation TO organization;

-- Delete the university\_shortname column

alter table affiliations

drop column university\_shortname;

==

-- Insert unique professors into the new table

insert into professors

SELECT DISTINCT firstname, lastname, university\_shortname

FROM university\_professors;

-- Doublecheck the contents of professors

SELECT \*

FROM professors;

-- Insert unique affiliations into the new table

insert into affiliations

select distinct firstname, lastname, function, organization

FROM university\_professors;

-- Doublecheck the contents of affiliations

SELECT \*

FROM affiliations;

-- Delete the university\_professors table

drop table university\_professors;

==

-- Let's add a record to the table

INSERT INTO transactions (transaction\_date, amount, fee)

VALUES ('2018-09-24', 5454, '30'); # as opposed to 24-09

-- Doublecheck the contents

SELECT \*

FROM transactions;

==

-- Calculate the net amount as amount + fee

SELECT transaction\_date, amount + cast(fee as integer) AS net\_amount

FROM transactions;

==

-- Select the university\_shortname column

SELECT distinct(university\_shortname)

FROM professors;

-- Specify the correct fixed-length character type

ALTER TABLE professors

ALTER COLUMN university\_shortname

TYPE char(3);

-- Change the type of firstname

alter table professors

alter column firstname

type varchar(64);

==

-- Convert the values in firstname to a max. of 16 characters

ALTER TABLE professors

ALTER COLUMN firstname

TYPE varchar(16)

using substring(firstname from 1 for 16)

==

-- Disallow NULL values in firstname

alter table professors

ALTER COLUMN firstname SET NOT NULL;

-- Disallow NULL values in lastname

alter table professors

ALTER COLUMN lastname SET NOT NULL;

==

-- Make universities.university\_shortname unique

ALTER table universities

ADD constraint university\_shortname\_unq UNIQUE(university\_shortname);

-- Make organizations.organization unique

-- Make universities.university\_shortname unique

ALTER table organizations

ADD constraint organization\_unq UNIQUE(organization);

==

-- Count the number of rows in universities

SELECT count(\*) from universities;

-- Count the number of distinct values in the university\_city column

SELECT count(distinct(university\_city))

FROM universities;

==

-- Try out different combinations

select COUNT(distinct(firstname, lastname))

FROM professors;

# allrows=551, flu=551, fl=551, f=360 lu=546, l=534, fu=479, u=11

==

-- Rename the organization column to id

alter table organizations

rename column organization TO id;

-- Make id a primary key

ALTER TABLE organizations

add constraint organization\_pk primary KEY (id);

-- Rename the university\_shortname column to id

alter table universities

rename column university\_shortname TO id;

-- Make id a primary key

ALTER TABLE universities

add constraint university\_pk primary KEY (id);

==

-- Add the new column to the table

ALTER TABLE professors

add column id serial;

-- Make id a primary key

ALTER table professors

add CONSTRAINT professors\_pkey primary key (id);

-- Have a look at the first 10 rows of professors

select \* from professors limit 10;

==

-- Count the number of distinct rows with columns make, model

select count(distinct (make,model)) FROM cars;

-- Add the id column

ALTER TABLE cars

add column id varchar(128);

-- Update id with make + model

UPDATE cars

set id = concat(make, model);

-- Make id a primary key

alter table cars

add constraint id\_pk primary key(id);

-- Have a look at the table

SELECT \* FROM cars;

==

-- Create the table

CREATE TABLE students (

last\_name varchar(128) NOT NULL,

ssn integer PRIMARY KEY,

phone\_no char(12)

);

==

-- Rename the university\_shortname column

ALTER TABLE professors

rename column university\_shortname to university\_id;

-- Add a foreign key on professors referencing universities

alter table professors

add constraint professors\_fkey FOREIGN KEY (university\_id) REFERENCES universities (id);

==

-- Try to insert a new professor

INSERT INTO professors (firstname, lastname, university\_id)

VALUES ('Albert', 'Einstein', 'UZH'); # ‘MIT’ would generate an error

==

-- Select all professors working for universities in the city of Zurich

SELECT professors.lastname, universities.id, universities.university\_city

from professors

join universities

ON professors.university\_id = universities.id

where universities.university\_city = 'Zurich';

==

-- Add a professor\_id column

alter table affiliations

add COLUMN professor\_id integer REFERENCES professors (id);

-- Rename the organization column to organization\_id

alter table affiliations

rename organization TO organization\_id;

-- Add a foreign key on organization\_id

ALTER TABLE affiliations

ADD CONSTRAINT affiliations\_organization\_fkey foreign key (organization\_id) references organizations (id);

==

-- Have a look at the 10 first rows of affiliations

select \* from affiliations limit 10;

-- Set professor\_id to professors.id where firstname, lastname correspond to rows in professors

UPDATE affiliations

SET professor\_id = professors.id

FROM professors

WHERE affiliations.firstname = professors.firstname AND affiliations.lastname = professors.lastname;

select \* from affiliations limit 10; # correct?

==

-- Drop the firstname column

alter table affiliations

DROP column firstname;

-- Drop the lastname column

alter table affiliations

DROP column lastname;

==

-- Identify the correct constraint name

SELECT constraint\_name, table\_name, constraint\_type

FROM information\_schema.table\_constraints

WHERE constraint\_type = 'FOREIGN KEY';

-- Drop the right foreign key constraint

ALTER table affiliations

drop CONSTRAINT affiliations\_organization\_id\_fkey;

-- Add a new foreign key constraint from affiliations to organizations which cascades deletion

ALTER TABLE affiliations

add constraint affiliations\_organization\_id\_fkey foreign KEY (organization\_id) references organizations (id) on delete cascade;

-- Delete an organization

DELETE FROM organizations

WHERE id = 'CUREM';

-- Check that no more affiliations with this organization exist

SELECT \* FROM affiliations

WHERE organization\_id = 'CUREM';

==

-- Count the total number of affiliations per university

SELECT count(\*), professors.university\_id

FROM affiliations

JOIN professors

ON affiliations.professor\_id = professors.id

-- Group by the ids of professors

GROUP BY professors.university\_id

order by count DESC; # 579 EPF

==

-- Join all tables

SELECT \*

FROM affiliations

JOIN professors

ON affiliations.professor\_id = professors.id

JOIN organizations

ON affiliations.organization\_id = organizations.id

JOIN universities

ON professors.university\_id = universities.id;

-- Group the table by organization sector, professor and university city

SELECT count(\*), organizations.organization\_sector,

professors.id, universities.university\_city

FROM affiliations

JOIN professors

ON affiliations.professor\_id = professors.id

JOIN organizations

ON affiliations.organization\_id = organizations.id

JOIN universities

ON professors.university\_id = universities.id

GROUP BY organizations.organization\_sector,

professors.id, universities.university\_city;

-- Filter the table and sort it

SELECT COUNT(\*), organizations.organization\_sector,

professors.id, universities.university\_city

FROM affiliations

JOIN professors

ON affiliations.professor\_id = professors.id

JOIN organizations

ON affiliations.organization\_id = organizations.id

JOIN universities

ON professors.university\_id = universities.id

where organizations.organization\_sector = 'Media & communication'

GROUP BY organizations.organization\_sector,

professors.id, universities.university\_city

order BY count DESC; # 4 Lausanne Prof #538

==