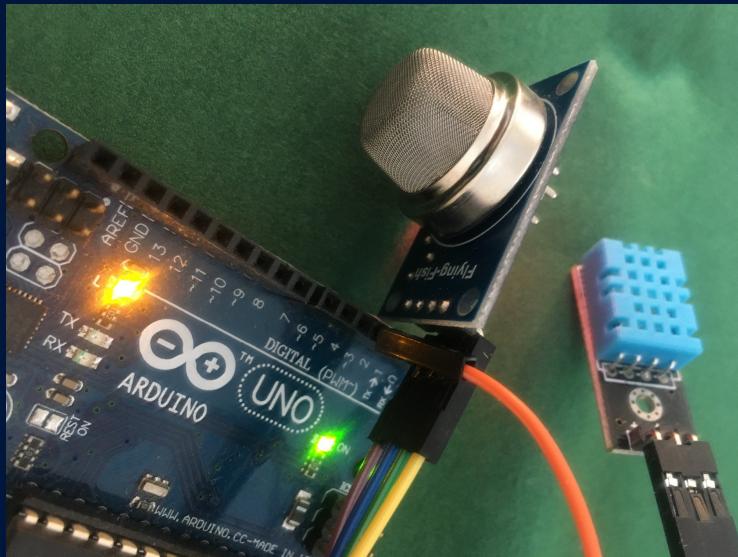


## COPENHAGEN BUSINESS ACADEMY



## DATA ENGINEERING



## FLOW 3 – Foreløbig plan

|        |            |                             |
|--------|------------|-----------------------------|
| Uge 10 | 06.11.2023 | intro til dataforespørgsler |
|        | 07.11.2023 | API                         |
|        | 08.11.2023 |                             |
|        | 09.11.2023 | API                         |
|        | 10.11.2023 | Statistik og ML             |
| Uge 11 | 13.11.2023 | Webscraping                 |
|        | 14.11.2023 | Webscraping                 |
|        | 15.11.2023 |                             |
|        | 16.11.2023 | ML og algoritmer            |
|        | 17.11.2023 | Webscraping                 |
| Uge 12 | 20.11.2023 | SQL                         |
|        | 21.11.2023 | SQL                         |
|        | 22.11.2023 |                             |
|        | 23.11.2023 | <b>SQL - Aflever OLA 4</b>  |
|        | 24.11.2023 | SQL                         |
| Uge 13 | 27.11.2023 |                             |
|        | 28.11.2023 | Cloud Computing             |
|        | 29.11.2023 |                             |
|        | 30.11.2023 | Cloud Computing             |
|        | 01.12.2023 | Statistik og ML             |
| Uge 14 | 04.12.2023 | IOT                         |
|        | 05.12.2023 | IOT                         |
|        | 06.12.2023 |                             |
|        | 07.12.2023 | <b>OLA - Aflever OLA 5</b>  |

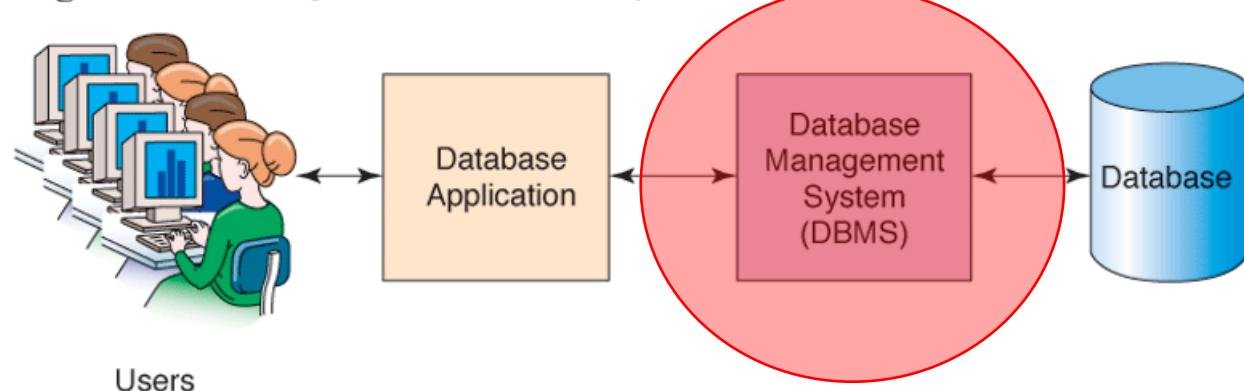
# Agenda - CRUD

- SQL
  - Intro til WorkBench
  - SELECT (conditions,join,aggregation)
    - WORLD-databasen
      - struktur
      - Øvelser i MySQL
    - Northwind
      - struktur
      - øvelser
  - UPDATE og INSERT (Northwind)
    - Demo
    - Øvelser
  - CREATE (Cars to EMP)
    - MySQL SKEMA og datatyper
    - Tilføj pris og forhandler
- SQL i R
  - SQL-queries fra R

# Database Management System (DBMS)

- Et **software system** som giver brugere mulighed for at definere, oprette og vedligeholde en database samt kontrolleret adgang til denne.

**Figure 1-15** Components of a Database System



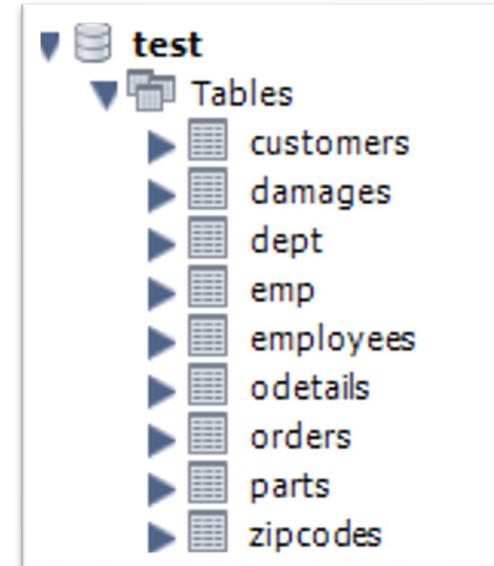
# Relationel Database

Den mest udbredte DBMS type.

- En database har et navn
- En database har en eller flere tabeller
- Hver tabel har et navn
- Hver tabel har en eller flere kolonner
- Hver kolonne har navn og datatype

**Eksempel:**

Database hedder **test**  
indeholder 9 tabeller



- A relational database consists of **tables**. Tables consist of **rows** and **columns**, which can also be referred to as records and fields.
- A table is typically modeled after a real-world entity, such as an invoice or a vendor.
- A column represents some **attribute** of the entity, such as the amount of an invoice or a vendor's address.
- A row contains a set of **values** for a **single** instance of the entity, such as one invoice or one vendor.
- The intersection of a row and a column is sometimes called a **cell**. A cell stores a **single value**.
- Most tables have a **primary key** that uniquely identifies each row in the table. The primary key is usually a single column, but it can also consist of two or more columns. If a primary key uses two or more columns, it's called a composite primary key.
- A table can also be defined with one or more indexes. An index provides an efficient way to access data from a table based on the values in specific columns. An index is automatically created for a table's primary and non-primary keys.

# Tabel eksempel Medarbejdere (emp)

Kolonner – har navn og simpel datatype

Rækker  
– indeholder  
relaterede  
værdier

| empno | ename  | job       | mgr  | hiredate   | sal  | deptno |
|-------|--------|-----------|------|------------|------|--------|
| 7369  | SMITH  | CLERK     | 7902 | 12/17/1980 | 800  | 20     |
| 7499  | ALLEN  | SALESMAN  | 7698 | 02/20/1981 | 1600 | 30     |
| 7521  | WARD   | SALESMAN  | 7698 | 02/22/1981 | 1250 | 30     |
| 7566  | JONES  | MANAGER   | 7839 | 04-02-1981 | 2975 | 20     |
| 7654  | MARTIN | SALESMAN  | 7698 | 09/28/1981 | 1250 | 30     |
| 7698  | BLAKE  | MANAGER   | 7839 | 05-01-1981 | 2850 | 30     |
| 7782  | CLARK  | MANAGER   | 7839 | 06-09-1981 | 2450 | 10     |
| 7788  | SCOTT  | ANALYST   | 7566 | 04/19/1987 | 3000 | 20     |
| 7839  | KING   | PRESIDENT |      | 11/17/1981 | 5000 | 10     |
| 7844  | TURNER | SALESMAN  | 7698 | 09-08-1981 | 1500 | 30     |
| 7876  | ADAMS  | CLERK     | 7788 | 05/23/1987 | 1100 | 20     |
| 7900  | JAMES  | CLERK     | 7698 | 12-03-1981 | 950  | 30     |
| 7902  | FORD   | ANALYST   | 7566 | 12-03-1981 | 3000 | 20     |
| 7934  | MILLER | CLERK     | 7782 | 01/23/1982 | 1300 | 10     |

# Tabel eksempel 2

## Medarbejdere (emp)

## Afdelinger (dept)

emp

| empno | ename  | job       | mgr  | hiredate   | sal  | deptno |
|-------|--------|-----------|------|------------|------|--------|
| 7369  | SMITH  | CLERK     | 7902 | 12/17/1980 | 800  | 20     |
| 7499  | ALLEN  | SALESMAN  | 7698 | 02/20/1981 | 1600 | 30     |
| 7521  | WARD   | SALESMAN  | 7698 | 02/22/1981 | 1250 | 30     |
| 7566  | JONES  | MANAGER   | 7839 | 04-02-1981 | 2975 | 20     |
| 7654  | MARTIN | SALESMAN  | 7698 | 09/28/1981 | 1250 | 30     |
| 7698  | BLAKE  | MANAGER   | 7839 | 05-01-1981 | 2850 | 30     |
| 7782  | CLARK  | MANAGER   | 7839 | 06-09-1981 | 2450 | 10     |
| 7788  | SCOTT  | ANALYST   | 7566 | 04/19/1987 | 3000 | 20     |
| 7839  | KING   | PRESIDENT |      | 11/17/1981 | 5000 | 10     |
| 7844  | TURNER | SALESMAN  | 7698 | 09-08-1981 | 1500 | 30     |
| 7876  | ADAMS  | CLERK     | 7788 | 05/23/1987 | 1100 | 20     |
| 7900  | JAMES  | CLERK     | 7698 | 12-03-1981 | 950  | 30     |
| 7902  | FORD   | ANALYST   | 7566 | 12-03-1981 | 3000 | 20     |
| 7934  | MILLER | CLERK     | 7782 | 01/23/1982 | 1300 | 10     |

dept

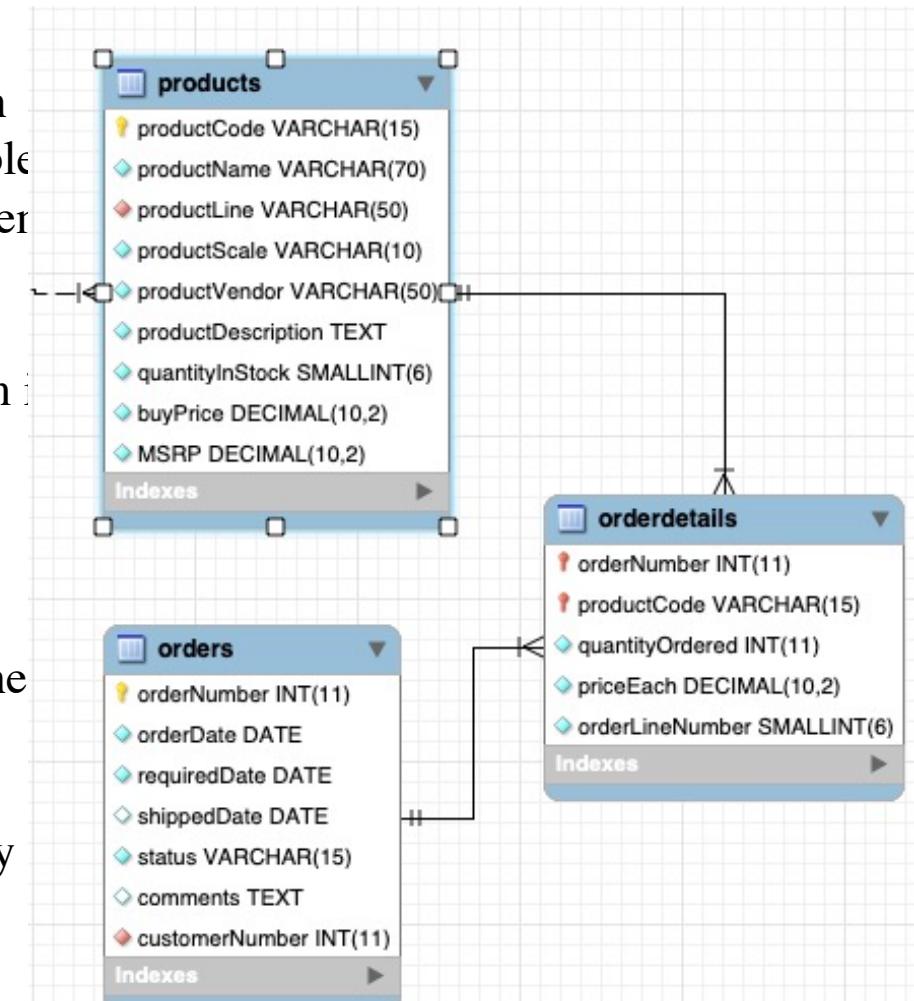
| deptno | dname      | loc      |
|--------|------------|----------|
| 10     | ACCOUNTING | NEW YORK |
| 20     | RESEARCH   | DALLAS   |
| 30     | SALES      | CHICAGO  |
| 40     | OPERATIONS | BOSTON   |

Er tabellerne  
logisk forbundne?

The tables in a database can be related to other tables by values in specific columns. The two tables shown in figure 1-5 illustrate this concept. Here, each row in the Employees table is related to one or more rows in the Customers table. This is called a **one-to-many** relationship. Typically, relationships exist between the primary key in one table and the foreign key in another table. The foreign key is simply one or more columns in a table that refer to a primary key in another table. In this figure, for example, the salesrepEmployeeNumber column is the foreign key in the Customers table and is used to create the relationship between the Employees table and the customers table.

In contrast, a **many-to-many** relationship is usually implemented by using an intermediate table that has a one-to-many relationship with the two tables in the many-to-many relationship. In other words, a many-to-many relationship can usually be broken down into two one-to-many relationships.

If you define a foreign key for a table in MySQL, you can have the foreign key enforce referential integrity. When MySQL enforces referential integrity, it makes sure that any changes to the data in the database don't create invalid relationships between tables. This helps to maintain the integrity of the data that's stored in the database.



# SQL - flere formål

## DML (Data Manipulation Language)

- Kommandoer som ændrer data i databasen

## DDL (Data Definition Language)

- Kommandoer som definerer databasen

Database forespørgsler har formatet:

```
select ...
from ...
where ...
```

Eksempel:

```
select ename
from emp
where mgr = 7698
```

# SQL

## Data Definition (DDL)

- **CREATE**
- **ALTER**
- **DROP**

## Data Manipulation (DML)

- **SELECT**
- **INSERT**
- **UPDATE**
- **DELETE**

# SQL SELECT eksempler

```
SELECT *  
FROM emp
```

```
SELECT ename, hiredate, sal  
FROM emp  
WHERE sal > 1000
```

```
SELECT avg(sal)  
FROM emp
```

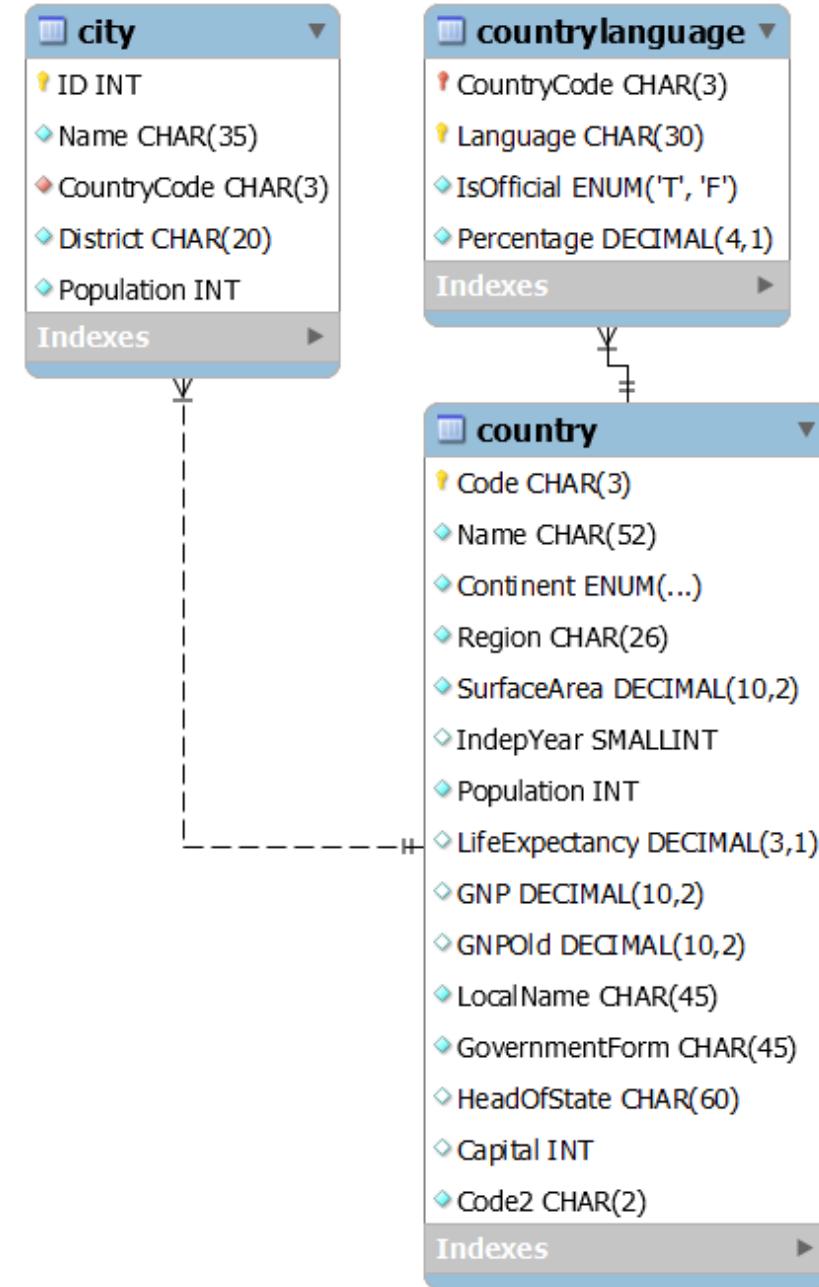
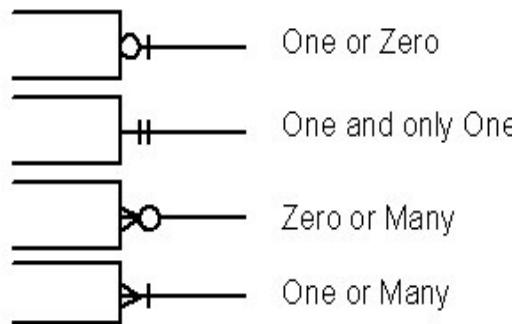
```
SELECT empno  
FROM emp  
WHERE ename = 'Smith'
```

| empno | ename  | hiredate   | sal  | deptno |
|-------|--------|------------|------|--------|
| 7369  | SMITH  | 12/17/1980 | 800  | 20     |
| 7876  | ADAMS  | 05/23/1987 | 1100 | 20     |
| 7900  | JAMES  | 12-03-1981 | 950  | 30     |
| 7934  | MILLER | 01/23/1982 | 1300 | 10     |

# World DB ER-diagram

howto: Reverse engineer in WorkBench

## Summary of Crow's Foot Notation



# World DB select

- 1) I hvilket distrikt ligger byen 'Stanley'?
- 2) Er færøsk et officielt sprog på Færøerne?
- 3) Hvad er `CountryCode` for 'Sri Lanka'
- 4) Hvilket land har det mindste areal?
- 5) Hvor mange amerikanske byer er med i DB'en?
- 6) I hvilket land taler mere end halvdelen af befolkningen 'Pashto'?
- 7) Hvad er den samlede befolkning i de danske byer der er med i DB'en?
- 8) Hvilke sprog tales i byen 'Nassau'?
- 9) Hvilket land har den højeste `LifeExpectancy`?
- 10) Hvilke lande har flere indbyggere end Russland?

# World DB - Joins

```
#sp 2
```

```
select language, name from Country  
join CountryLanguage on country.code=CountryLanguage.CountryCode  
where name like "%oe%"  
order by 2;
```

- 1) I hvilket distrikt ligger byen 'Stanley'?
- 2) Er færøsk et officielt sprog på Færøerne?
- 3) Hvad er `CountryCode` for 'Sri Lanka'
- 4) Hvilket land har det mindste areal?
- 5) Hvor mange amerikanske byer er med i DB'en?
- 6) I hvilket land taler mere end halvdelen af befolkningen 'Pashto'?
- 7) Hvad er den samlede befolkning i de danske byer der er med i DB'en?
- 8) Hvilke sprog tales i byen 'Nassau'?
- 9) Hvilket land har den højeste `LifeExpectancy`?
- 10) Hvilke lande har flere indbyggere end Russland?

# World DB - subqueries

- 1) I hvilket distrikt ligger byen 'Stanley'?
- 2) Er færøsk et officielt sprog på Færøerne?
- 3) Hvad er `CountryCode` for 'Sri Lanka'
- 4) Hvilket land har det mindste areal?

15

```
16 • select name, surfacearea from Country  
17 where SurfaceArea in (select min(surfacearea) from country);  
--
```

# World DB - Aggregation

5) Hvor mange amerikanske byer er med i DB'en?

```
20  #sp 5
21 • select count(*) as "cities", co.name from city ci, country co
22 where ci.countrycode=co.code
23 group by co.name|
24 order by 1 desc;
--
```

| Aggregate      |             |
|----------------|-------------|
| Avg            | Min         |
| BIT_AND        | STD         |
| BIT_OR         | STDDEV      |
| BIT_XOR        | STDDEV_POP  |
| COUNT          | STDDEV_SAMP |
| GROUP_CONCAT   | SUM         |
| JSON_ARRAYAGG  | VAR_POP     |
| JSON_OBJECTAGG | VAR_SAMP    |
| MAX            | VARIANCE    |

# World DB join

6) I hvilket land taler mere end halvdelen af befolkningen 'Pashto'?  
"list all land og deres sprog"

```
26  #sp 6
27 • select cl.language, cl.percentage, co.name, co.population from CountryLanguage cl, country co
28 where cl.CountryCode=co.code
29 and cl.language like "Pash%"
30 order by 2 desc;
```

# World DB Aggregation

7) Hvad er den samlede befolkning i de danske byer der er med i DB'en?

```
32 # sp 7
33 • select co.name,sum(ci.population) as "sum pop"from city ci, country co
34 where ci.countrycode=co.code
35 #and co.name like "Den%"
36 group by co.name
37 order by 1 desc;
--
```

# World DB joins

8) Hvilke sprog tales i byen 'Nassau'?

```
| #sp 8 Sprog i nassau?
| • select cl.language,ci.name,co.name from countrylanguage cl, city ci, Country Co
|   where ci.countrycode=co.Code
|     and co.code=cl.countrycode
|     and ci.name like "Nassa%"
|     order by 2;
```

# World DB subquery

9) Hvilket land har den højeste `LifeExpectancy`?

```
45
46  # sp 9 Højest life-expect
47 • Select lifeexpectancy, name from Country
48 where LifeExpectancy in (select max(lifeexpectancy) from country)
49
```

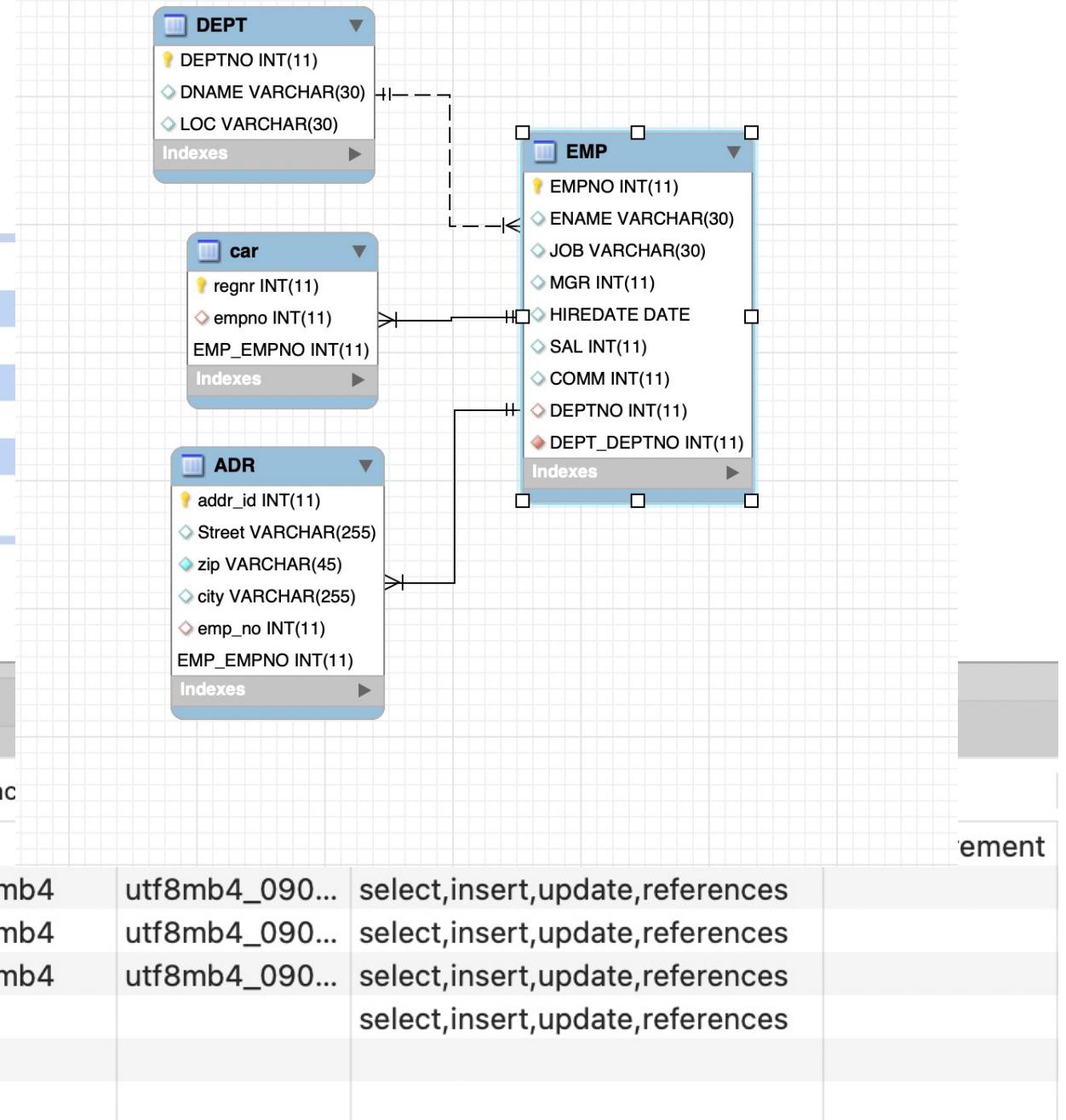
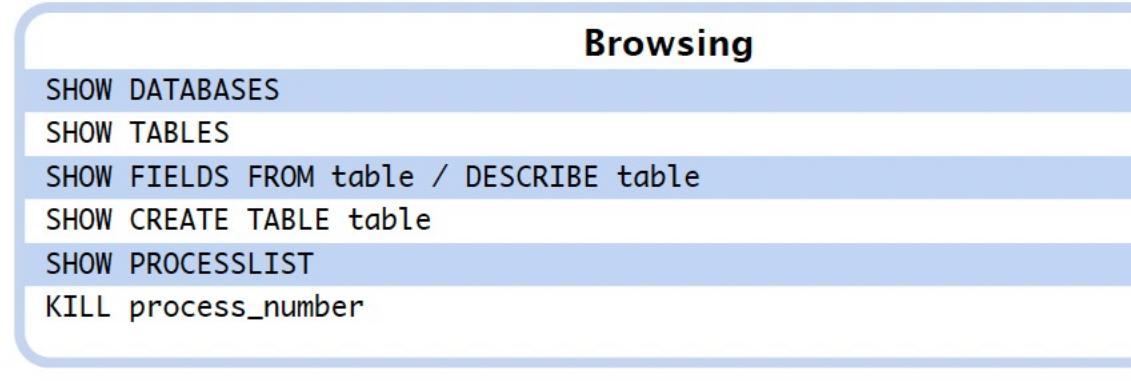
# World DB

## select

10) Hvilke lande har flere indbyggere end Rusland?

```
--  
50  # sp 10 Flere indb end Rusland?  
51 • select name, population from country  
52   where population > (select population from country  
53     where name like "Rus%")  
54   order by 2;  
55
```

# MySQL Skema



# MySQL Design

## Acme Fabrication, Inc.

|   |                 |          |
|---|-----------------|----------|
| <i>Custom Contraptions, Contrivances and Confabulations</i> | Invoice Number: | I01-1088 |
| 1234 West Industrial Way East Los Angeles California 90022  | Invoice Date:   | 10/05/18 |
| 800.555.1212 fax 562.555.1213 www.acmefabrication.com       | Terms:          | Net 30   |

| Part No. | Qty. | Description               | Unit Price | Extension |
|----------|------|---------------------------|------------|-----------|
| CUST345  | 12   | Design service, hr        | 100.00     | 1200.00   |
| 457332   | 7    | Baling wire, 25x3ft roll  | 79.90      | 559.30    |
| 50173    | 4375 | Duct tape, black, yd      | 1.09       | 4768.75   |
| 328771   | 2    | Rubber tubing, 100ft roll | 4.79       | 9.58      |
| CUST281  | 7    | Assembly, hr              | 75.00      | 525.00    |
| CUST917  | 2    | Testing, hr               | 125.00     | 250.00    |
|          |      | Sales Tax                 |            | 245.20    |

|                      |                          |
|----------------------|--------------------------|
| Your salesperson:    | Ruben Goldberg, ext 4512 |
| Accounts receivable: | Inigo Jones, ext 4901    |

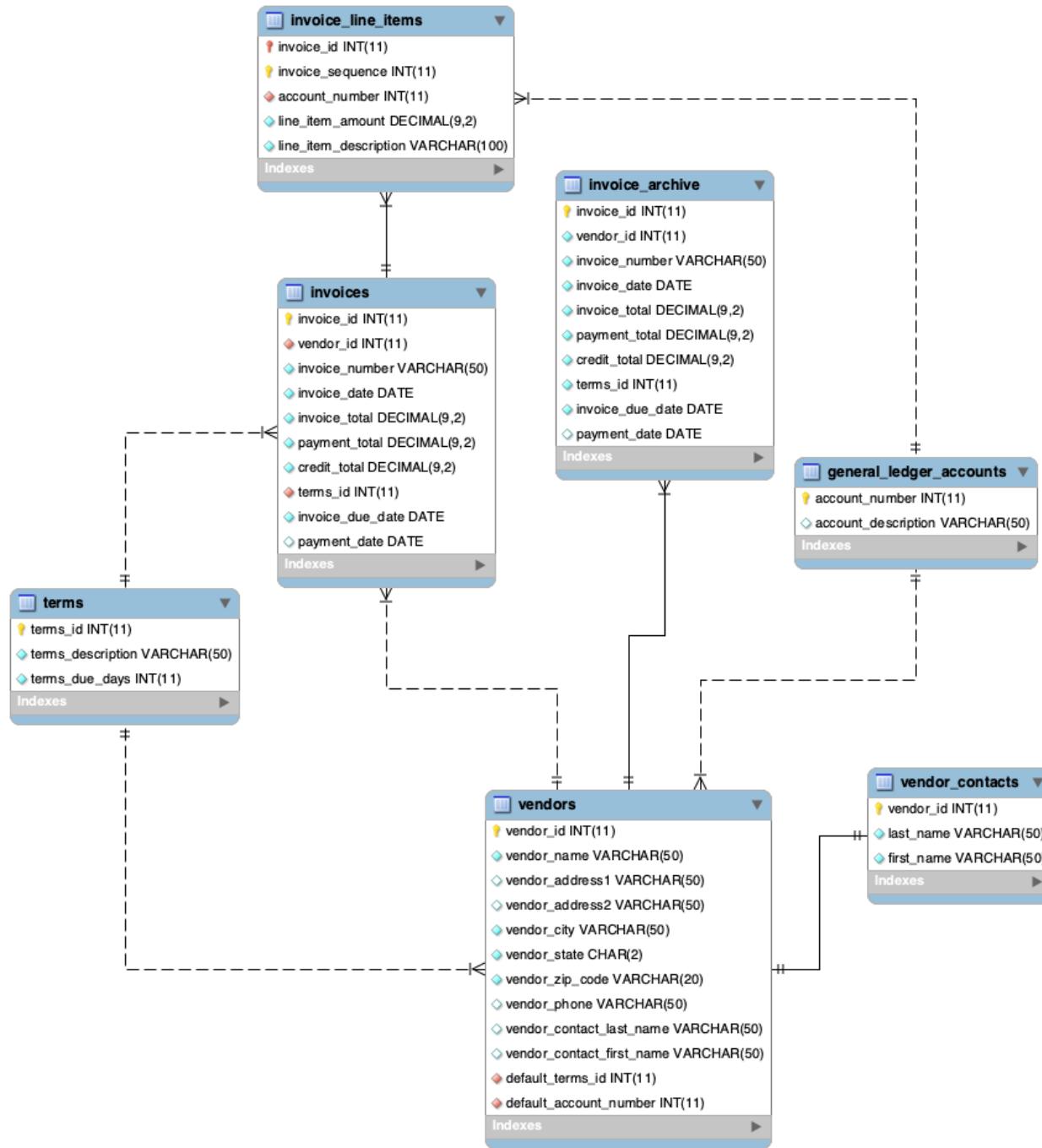
\$7,557.83  
PLEASE PAY THIS AMOUNT

Thanks for your business!

# MySQL Design - datatyper

| DATE TYPE  | SPEC                          | DATA TYPE | SPEC  |
|------------|-------------------------------|-----------|---|
| CHAR       | String (0 - 255)              | INT       | Integer (-2147483648 to 214748-3647)                  |
| VARCHAR    | String (0 - 255)              | BIGINT    | Integer (-9223372036854775808 to 9223372036854775807) |
| TINYTEXT   | String (0 - 255)              | FLOAT     | Decimal (precise to 23 digits)                        |
| TEXT       | String (0 - 65535)            | DOUBLE    | Decimal (24 to 53 digits)                             |
| BLOB       | String (0 - 65535)            | DECIMAL   | "DOUBLE" stored as string                             |
| MEDIUMTEXT | String (0 - 16777215)         | DATE      | YYYY-MM-DD  |
| MEDIUMBLOB | String (0 - 16777215)         | DATETIME  | YYYY-MM-DD HH:MM:SS                                   |
| LONGTEXT   | String (0 - 4294967295)       | TIMESTAMP | YYYYMMDDHHMMSS  |
| LONGBLOB   | String (0 - 4294967295)       | TIME      | HH:MM:SS  |
| TINYINT    | Integer (-128 to 127)         | ENUM      | One of preset options                                 |
| SMALLINT   | Integer (-32768 to 32767)     | SET       | Selection of preset options                           |
| MEDIUMINT  | Integer (-8388608 to 8388607) | BOOLEAN   | TINYINT(1)  |

# MySQL Design



# MySQL from R

- Connecting and disconnecting
  - Connecting to and disconnecting from databases
    - `dbConnect(MariaDB(), ..)`
- Tables
  - Reading and writing entire tables
    - `dbWriteTable(con, "mycarstable", mycarsdf)`
    - `mycardf <- dbReadTable(con, "mycarstable")`
- Results
  - More control for sending queries and executing statements
    - `dbGetQuery(con, "SELECT * FROM city limit 3")`
    - `req = dbSendQuery(con, "INSERT INTO city (Name,Population) VALUES ('Lviv',123123)")`
    - `rows = dbExecute (con, "UPDATE city set Name = 'Kurt' WHERE id = 123")`

# Mysql – Filtre

## Select

```
SELECT * FROM table  
SELECT * FROM table1, table2, ...  
SELECT field1, field2, ... FROM table1, table2, ...  
SELECT ... FROM ... WHERE condition  
SELECT ... FROM ... WHERE condition GROUPBY field  
SELECT ... FROM ... WHERE condition GROUPBY field HAVING condition2  
SELECT ... FROM ... WHERE condition ORDER BY field1, field2  
SELECT ... FROM ... WHERE condition ORDER BY field1, field2 DESC  
SELECT ... FROM ... WHERE condition LIMIT 10  
SELECT DISTINCT field1 FROM ...  
SELECT DISTINCT field1, field2 FROM ...
```

## Conditions

```
field1 = value1  
field1 <> value1  
field1 LIKE 'value _ %'  
field1 IS NULL  
field1 IS NOT NULL  
field1 IS IN (value1, value2)  
field1 IS NOT IN (value1, value2)  
condition1 AND condition2  
condition1 OR condition2
```

## What is Inner Join?

An Inner Join returns only the rows that have matching values in both the tables (we are considering here the join is done between the two tables).

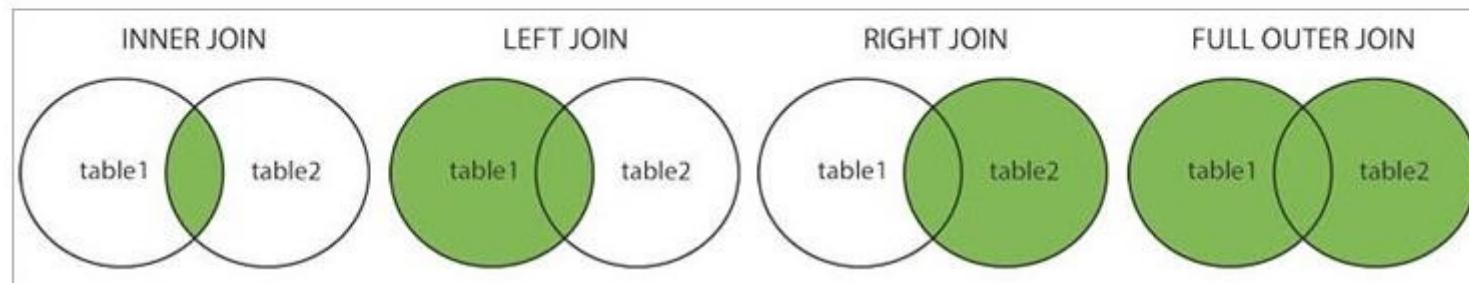
## What is Outer Join?

The Outer Join includes the matching rows as well as some of the non-matching rows between the two tables. An Outer join basically differs from the Inner join in how it handles the false match condition.

**There are 3 types of Outer Join:**

- **Left Outer Join:** Returns all the rows from the LEFT table and matching records between both the tables.
- **Right Outer Join:** Returns all the rows from the RIGHT table and matching records between both the tables.
- **Full Outer Join:** It combines the result of the Left Outer Join and Right Outer Join.

## Difference between Inner and Outer Join



| avgsal | dname      |
|--------|------------|
| ► 1567 | SALES      |
| 2175   | RESEARCH   |
| 2917   | ACCOUNTING |

| avg sal | dname      |
|---------|------------|
| ► NULL  | OPERATIONS |
| 1567    | SALES      |
| 2175    | RESEARCH   |
| 2917    | ACCOUNTING |

# WORLD Databasen

## Spørgsmål til "world" databasen

1) I hvilket distrikt ligger byen 'Stanley'?

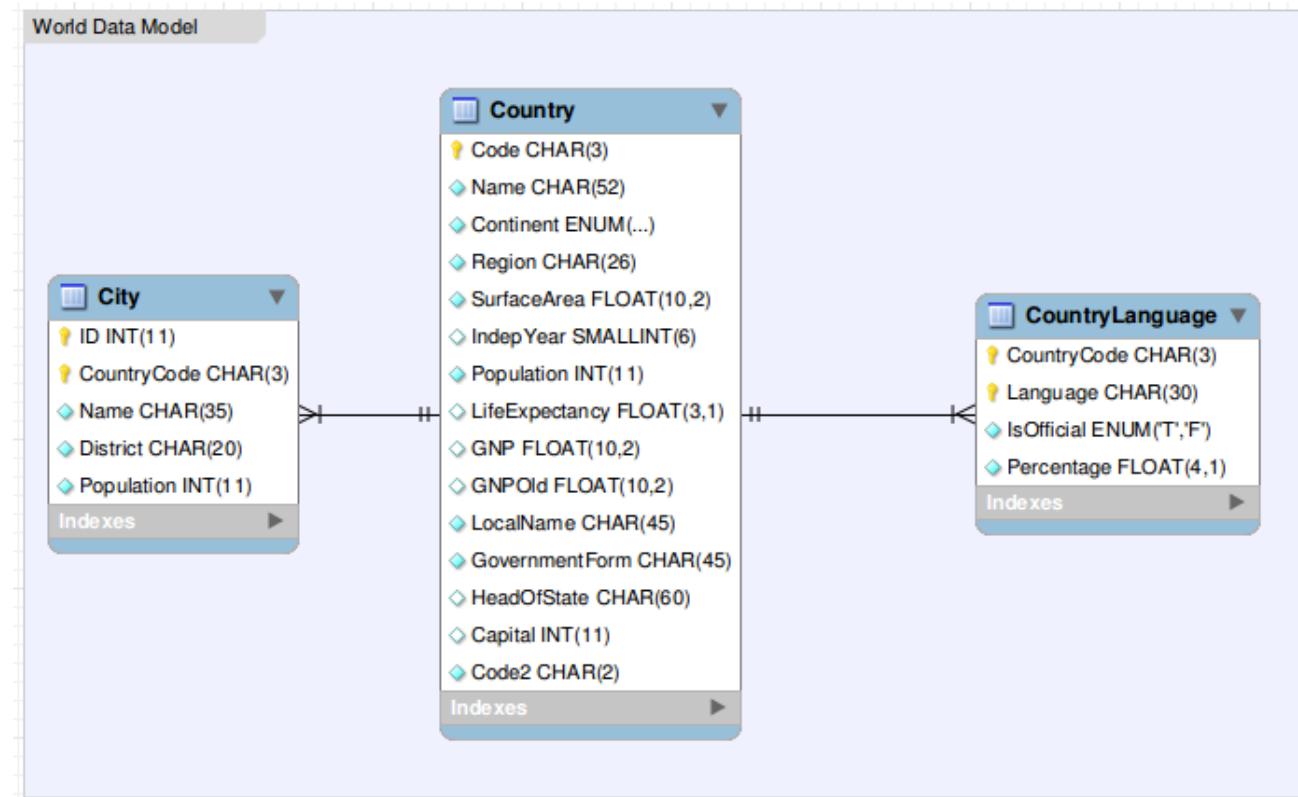
2) Er færøsk et officielt sprog på Færøerne?

3) Hvad er 'CountryCode' for 'Sri Lanka'

4) Hvilket land har det mindste areal?

5) Hvor mange amerikanske byer er med i DB'en?

6) I hvilket land taler mere end halvdelen af befolkningen 'Pashto'?



# WORLD Databasen

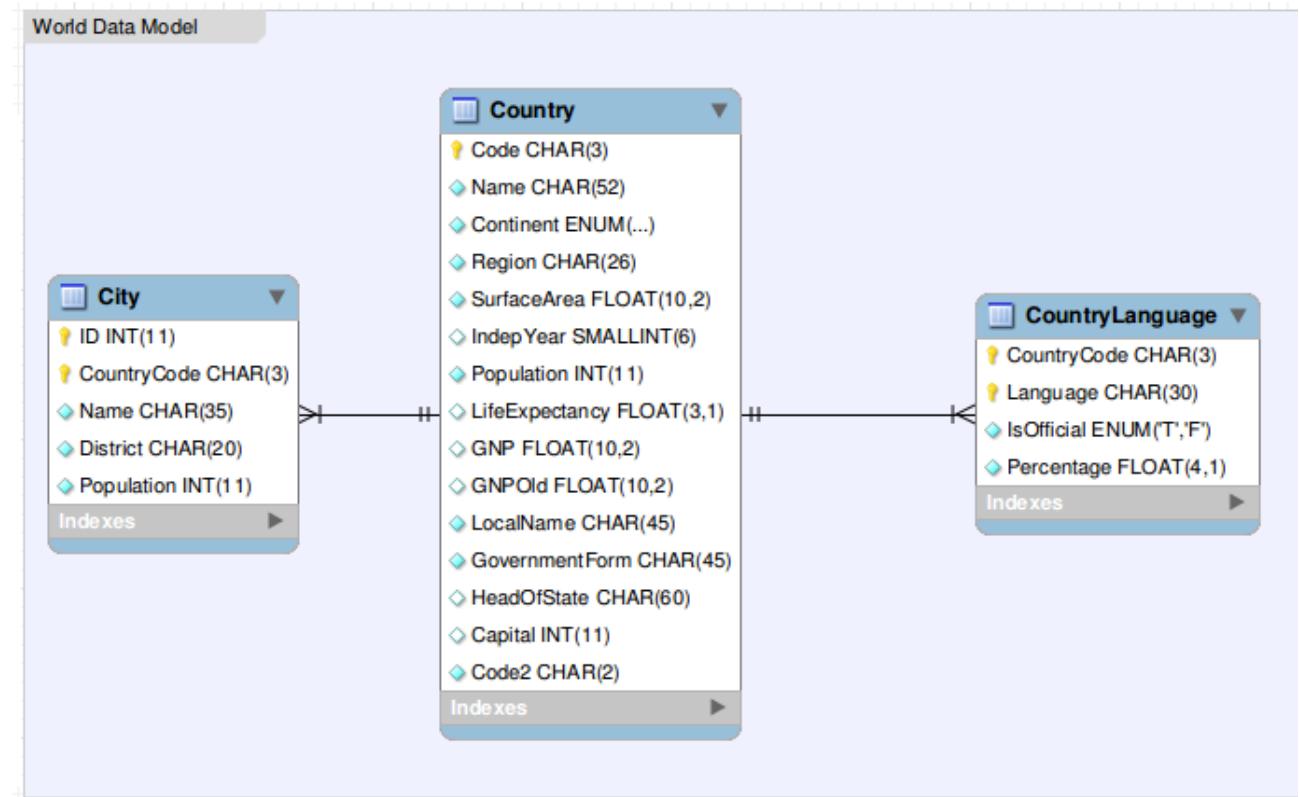
## Spørgsmål til "world" databasen

7) Hvad er den samlede befolkning i de danske byer der er med i DB'en?

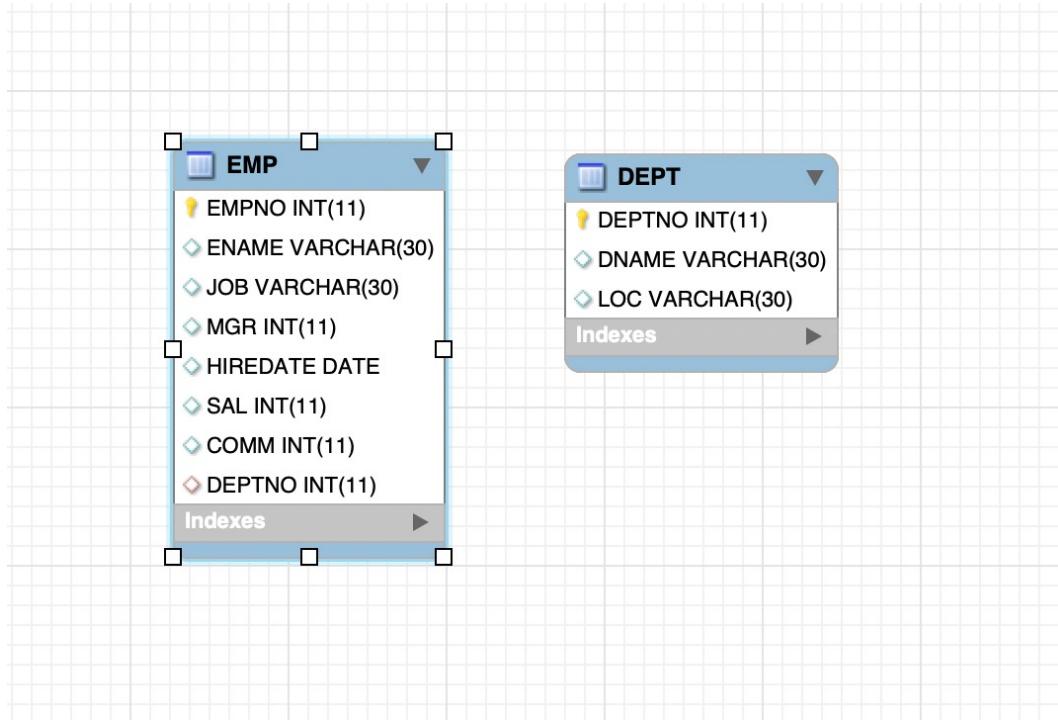
8) Hvilke sprog tales i byen 'Nassau'?

9) Hvilket land har den højeste 'LifeExpectancy'?

10) Hvilke lande har flere indbyggere end Rusland?



# EMPLOYEE Databasen



## Øvelse:

Find max-værdien af DEPTNO

Indsæt en ny afdeling, DATASCIENCE (Seattle) med passende DEPTNO

Tilføj dig selv som medarbejder med passende data (brug transaktion)

Prøv at tilføje en medarbejder til en ikke-eksisterende dept

Kan du lave en query så du får flg:

|   | gennemsnit | afdeling   |
|---|------------|------------|
| ► | 1566       | SALES      |
|   | 2175       | RESEARCH   |
|   | 2916       | ACCOUNTING |

Og samme resultat i R

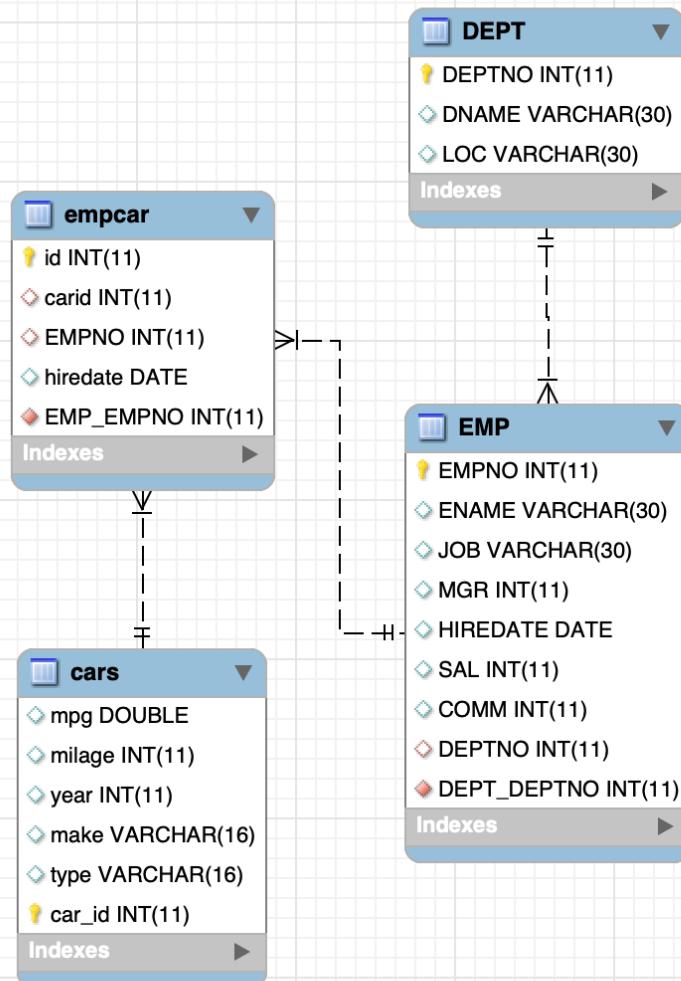
rMariaTry.R\*

Filter

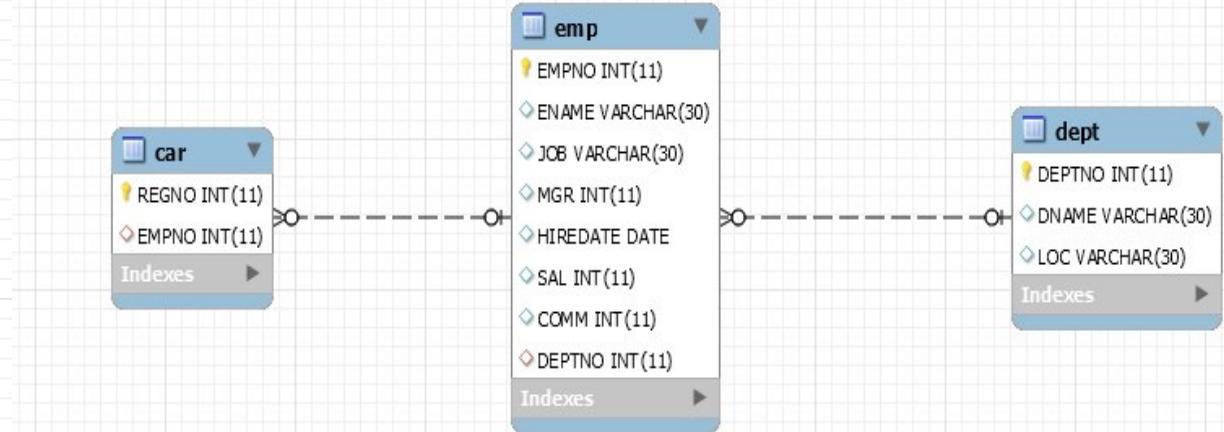
|   | DNAME      | V1   |
|---|------------|------|
| 1 | ACCOUNTING | 2916 |
| 2 | RESEARCH   | 2175 |
| 3 | SALES      | 1566 |

# EMPLOYEE Databasen – nu med biler

Den "svære"



Den "lette"



Your tasks:

- Create the new table CAR
- Insert minimum 4 registrations in the car table
- Try to make a new insert in the car table, where you use one of the same car registration numbers. What happens?  
Change owner of car with registration number 40 to employee 7698.

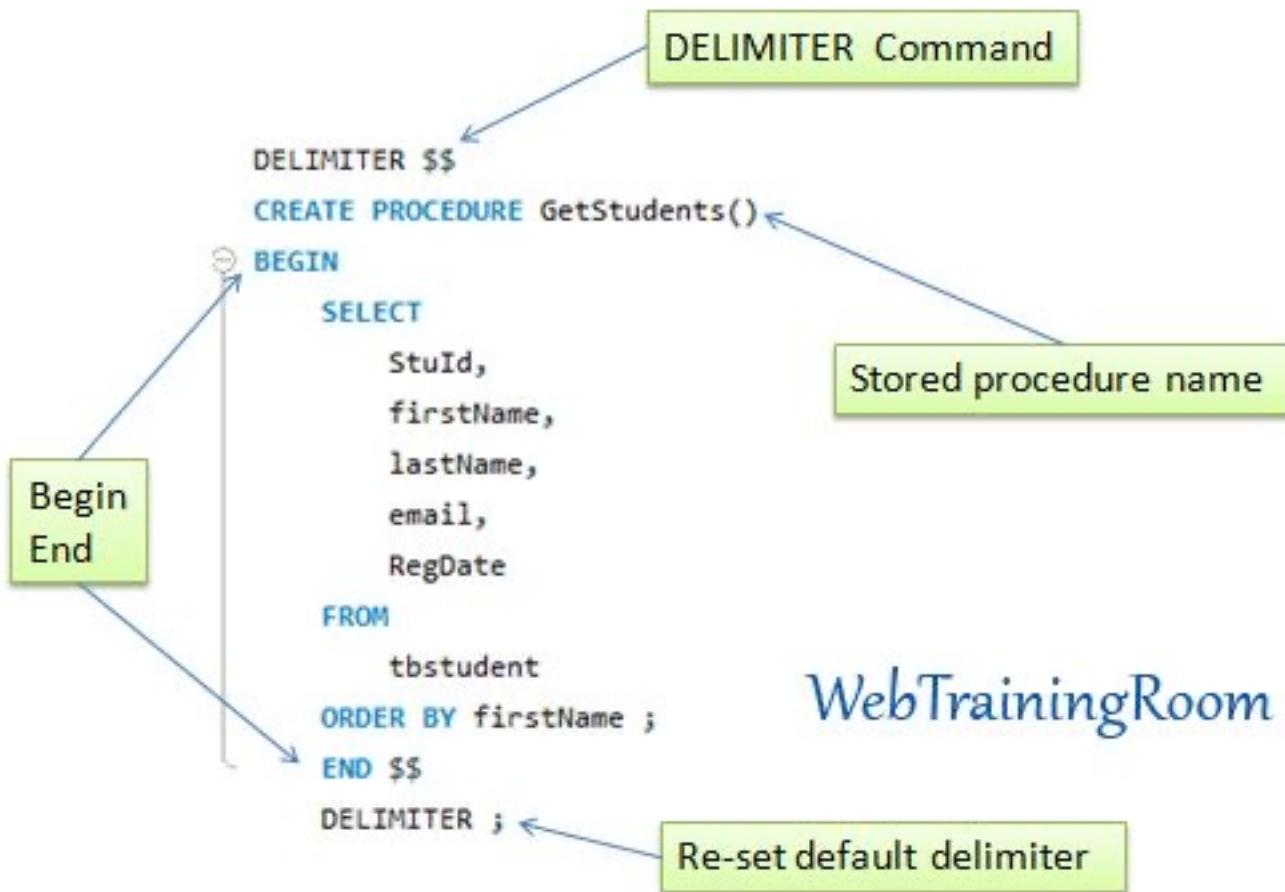
# EMPLOYEE Databasen – nu med biler & R

Your tasks:

- Indlæs biler fra bilbasen i R
- Skriv dem til mysql vha dbWriteTable()
- Ret i skemaet så det ”passer”
- Tilføj link-tabellen så medarbejdere kan leje biler
- Lav et view i MySQL som viser alle biler – lejet såvel som ledige
- Lad MARTIN leje Aygo'en i dag – tjek at den er ledig
- Modificer så man kan aflevere bilen og sætte kørt km
- Opdatér bilens kørte km.
- Lav en stored procedure hvor du kan trække en liste af lejede biler en måned tilbage

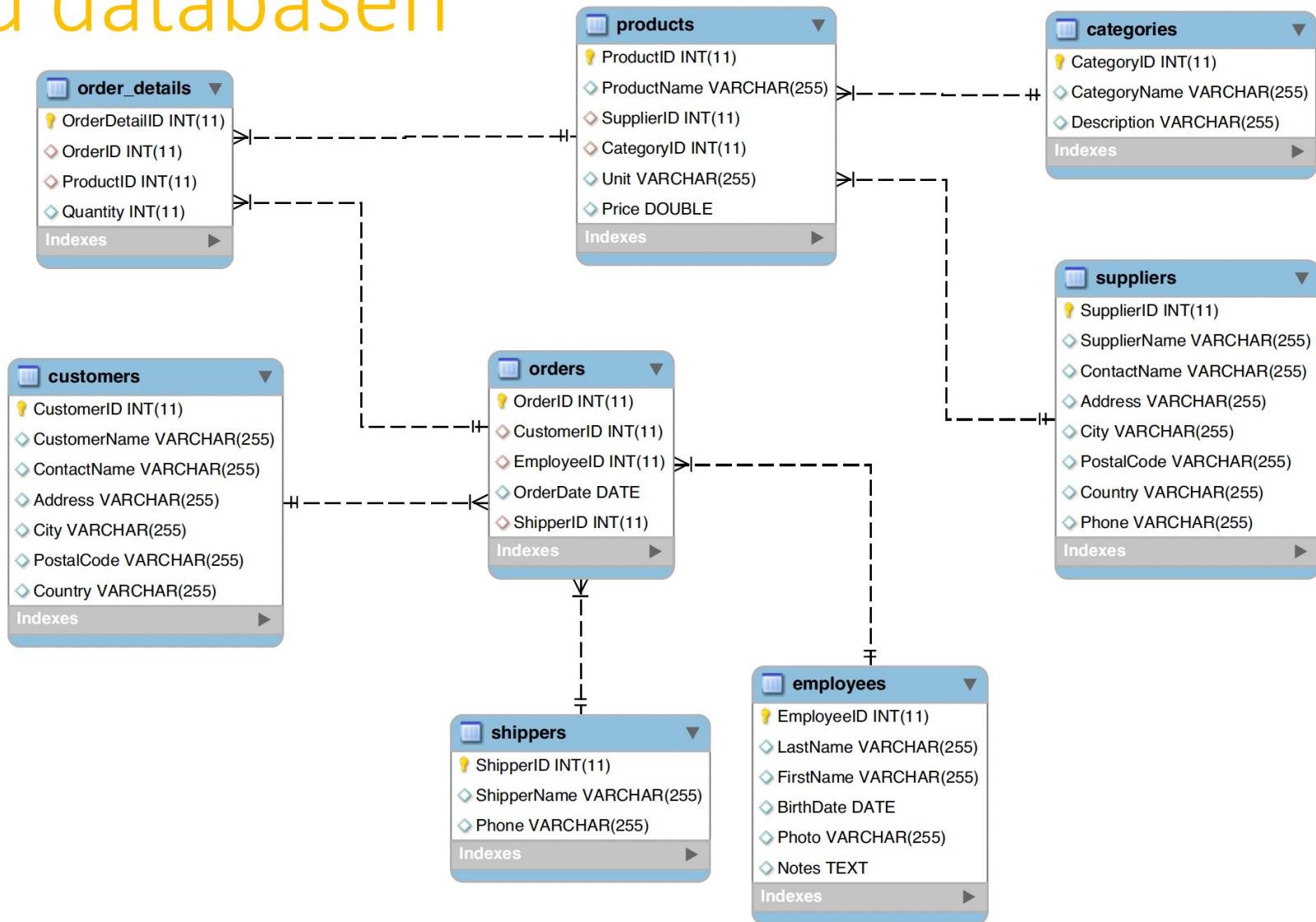
| ename  | empno | make   | car_id  | hiredate   |
|--------|-------|--------|---------|------------|
| ► NULL | NULL  | Toyota | 5260006 | NULL       |
| NULL   | NULL  | Toyota | 5569652 | NULL       |
| NULL   | NULL  | Ford   | 5575935 | NULL       |
| NULL   | NULL  | Toyota | 5578393 | NULL       |
| JONES  | 7566  | Toyota | 5599017 | 2022-11-17 |
| JAMES  | 7900  | Ford   | 5614114 | 2022-10-08 |
| NULL   | NULL  | Ford   | 5614388 | NULL       |
| NULL   | NULL  | Toyota | 5621650 | NULL       |
| NULL   | NULL  | Toyota | 5624946 | NULL       |
| NULL   | NULL  | Ford   | 5628969 | NULL       |

# EMP databasen – rapporter



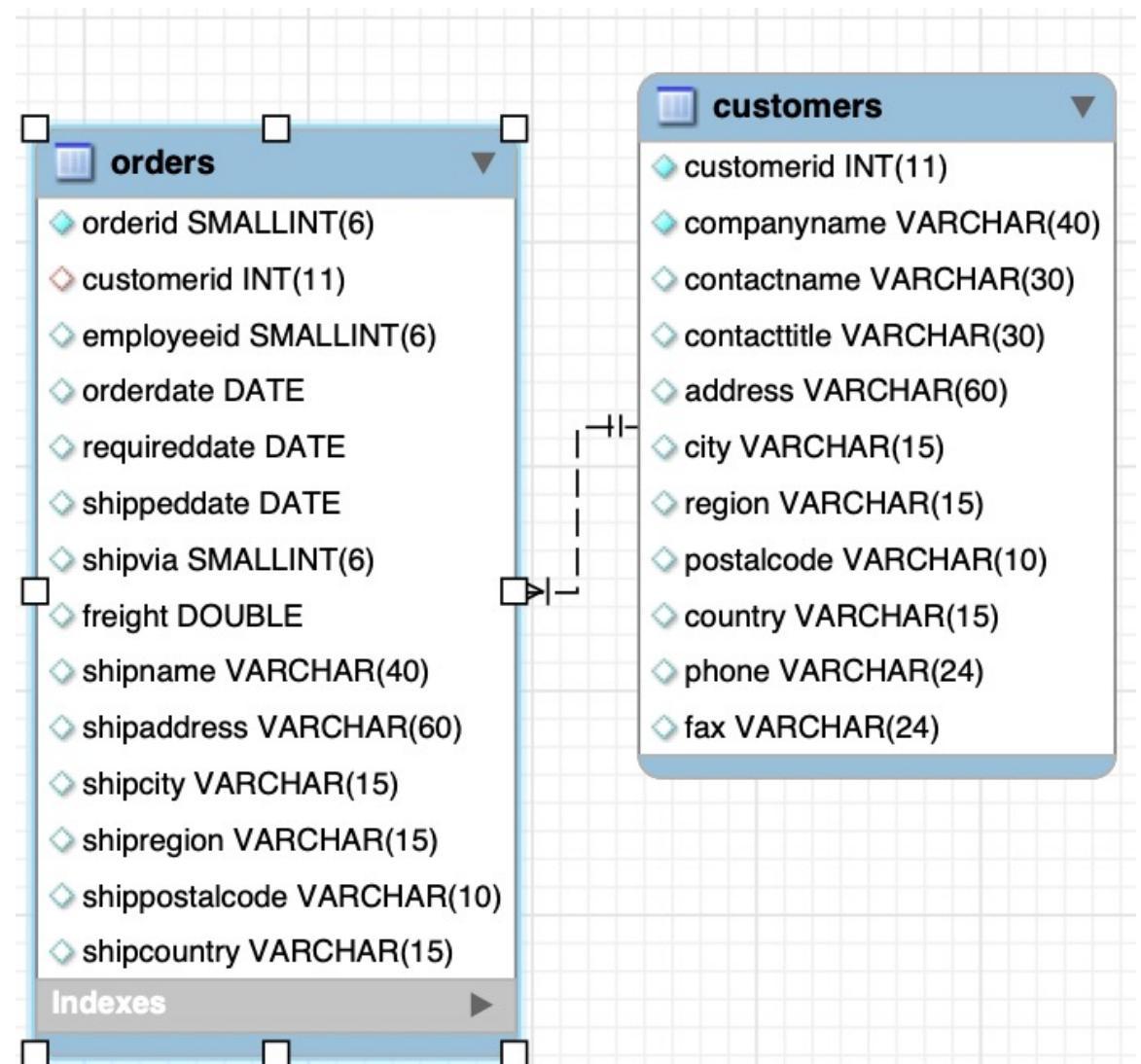
WebTrainingRoom

# Northwind databasen



# Northwind databasen - videodude

Liste over alle kunders ordrer og de medarbejdere som hjalp dem?

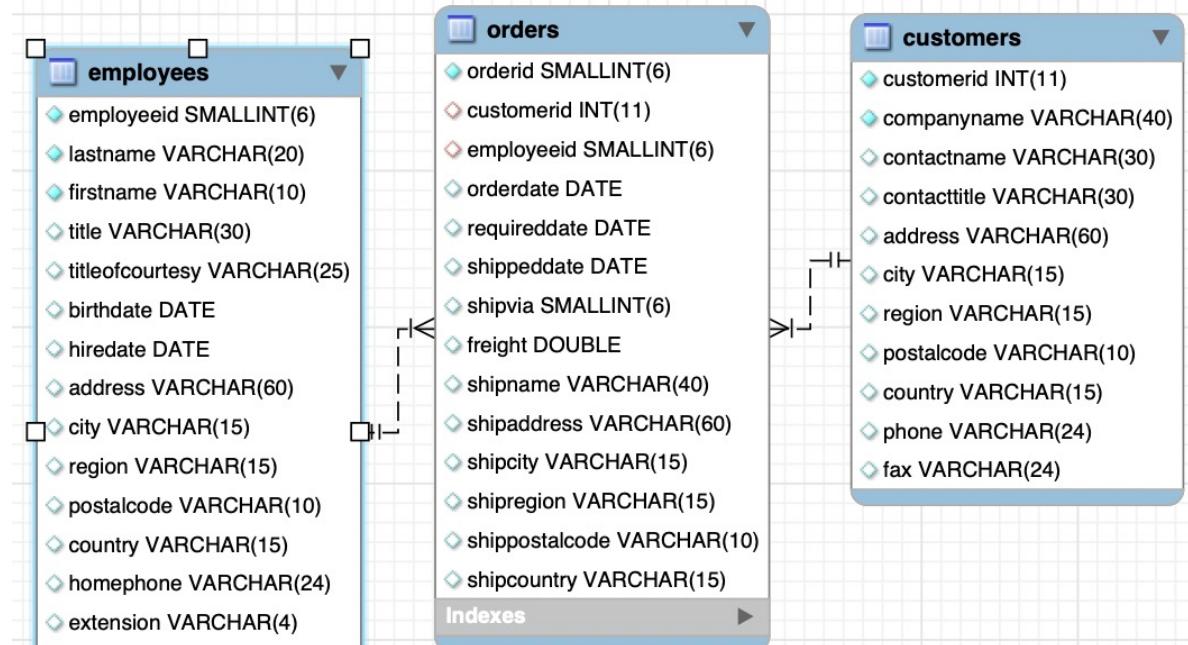


# Northwind databasen - videodude

Hvilke medarbejdere  
hjælp med handlerne?

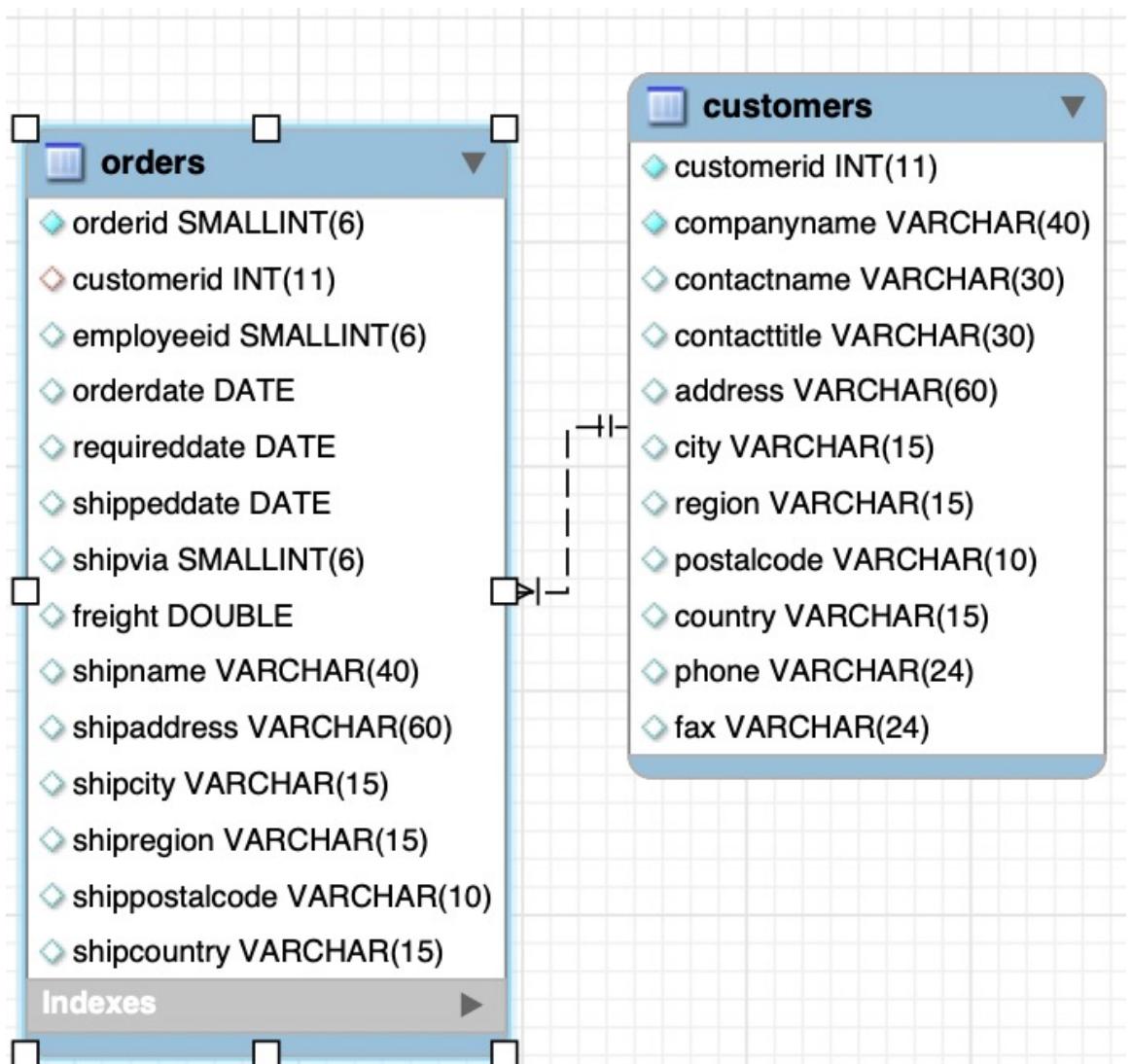
```
select companyname, orderdate, shipcountry, firstname, lastname  
from orders  
join customers on orders.customerid=customers.customerid  
join employees on orders.employeeid=employees.employeeid  
order by 2;
```

| Result Grid               | Filter Rows: | Search      | Export:   |           |  |
|---------------------------|--------------|-------------|-----------|-----------|--|
| companyname               | orderdate    | shipcountry | firstname | lastname  |  |
| Vins et alcools Chevalier | 1996-07-04   | France      | Steven    | Buchanan  |  |
| Toms Spezialitäten        | 1996-07-05   | Germany     | Michael   | Suyama    |  |
| Victuailles en stock      | 1996-07-08   | France      | Janet     | Leverling |  |
| Hanari Carnes             | 1996-07-08   | Brazil      | Margaret  | Peacock   |  |



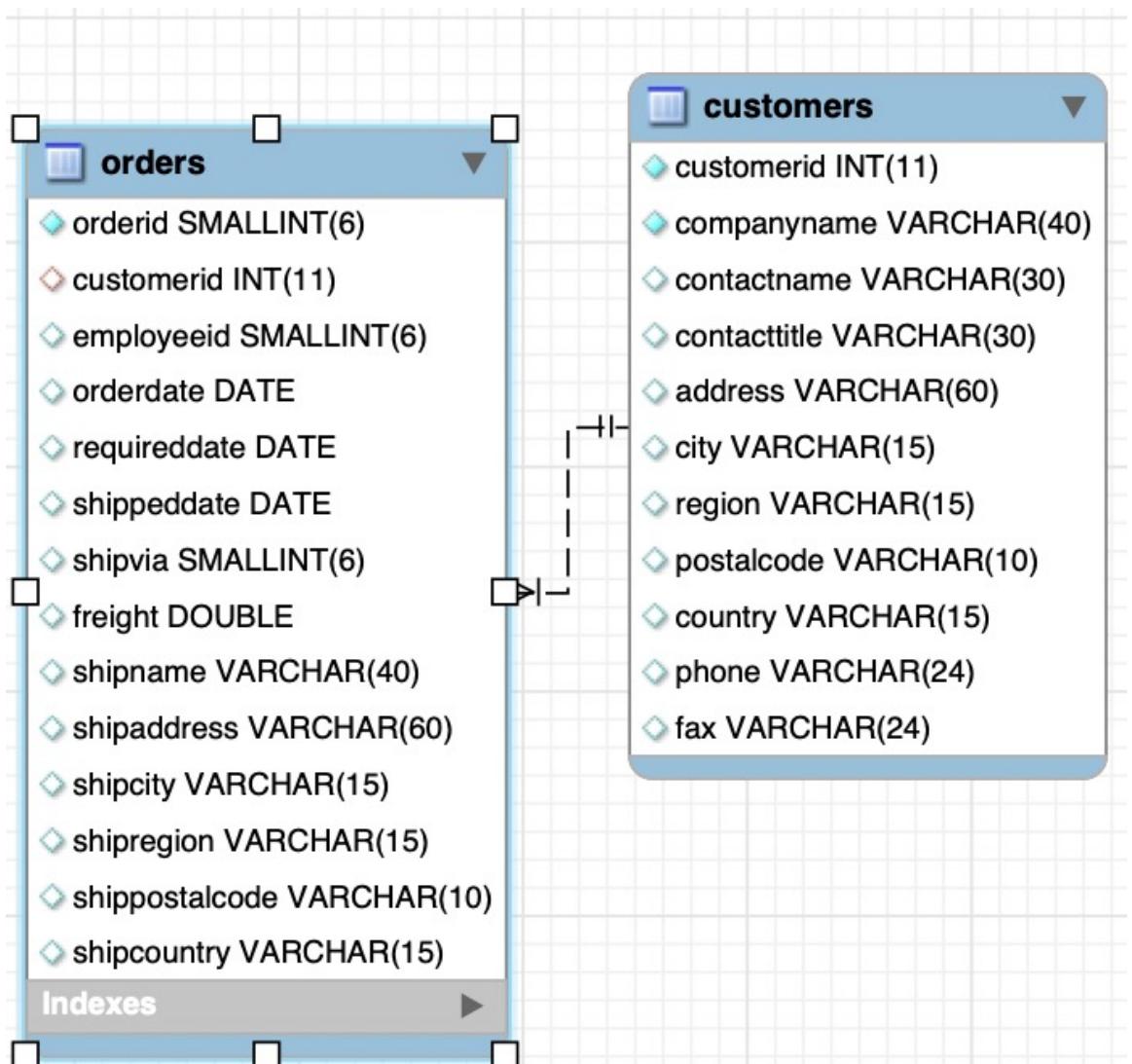
# Northwind databasen

## Liste over alle kunders ordrer og de medarbejdere som hjalp dem?



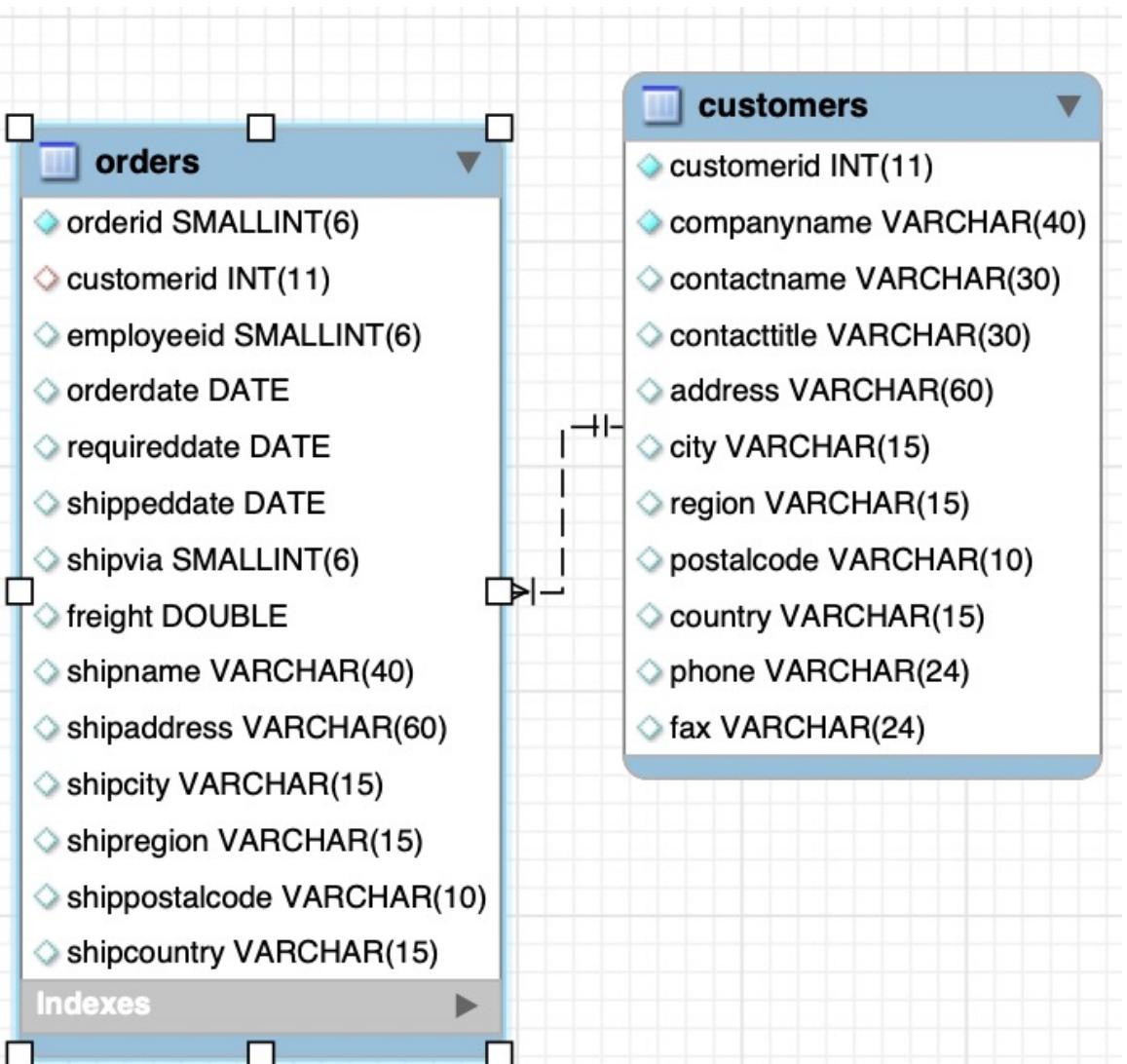
# Northwind databasen

Hvilke tre kunder placerede flest ordrer?



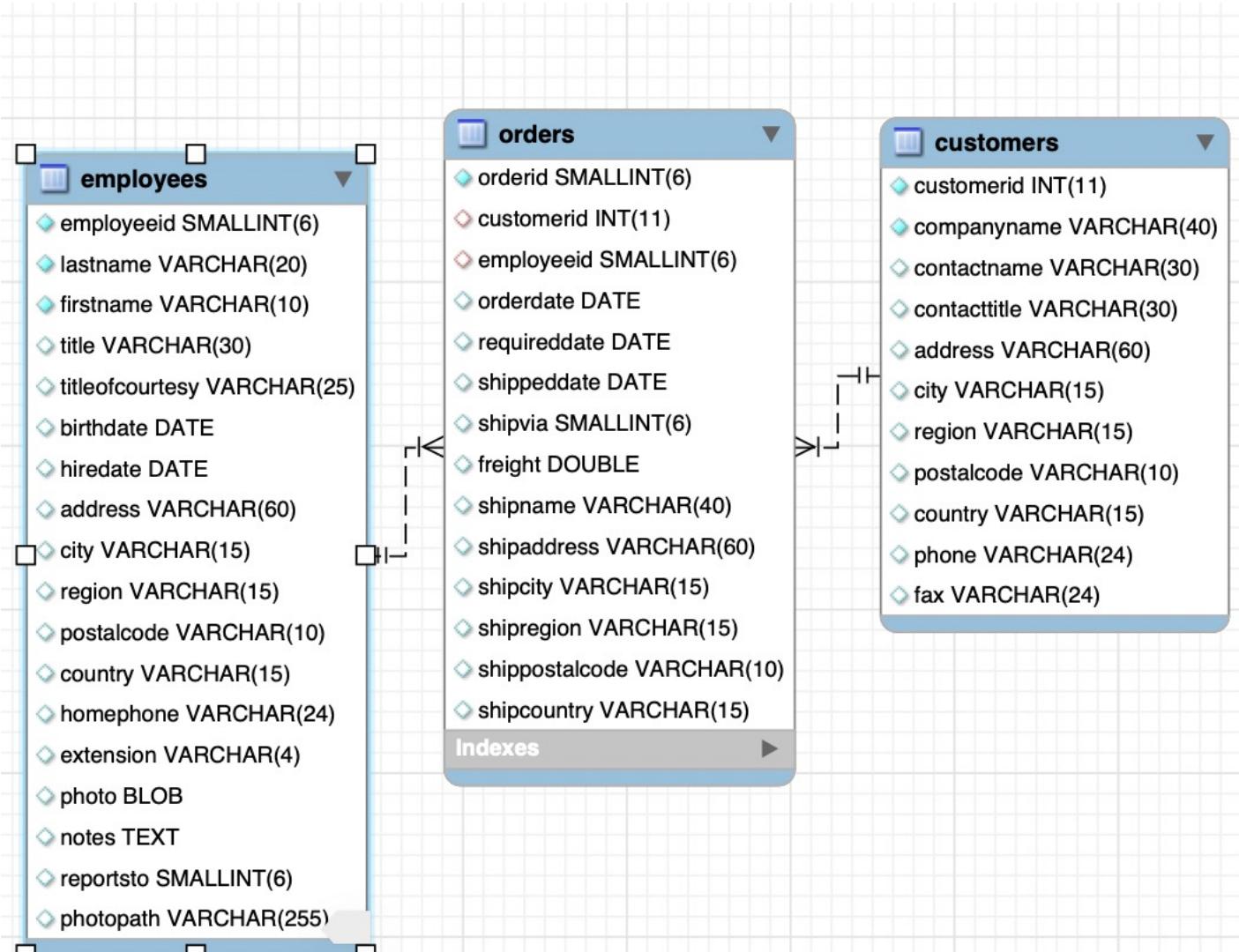
# Northwind databasen

Hvilke kunder placerede ingen ordre?



# Northwind databasen

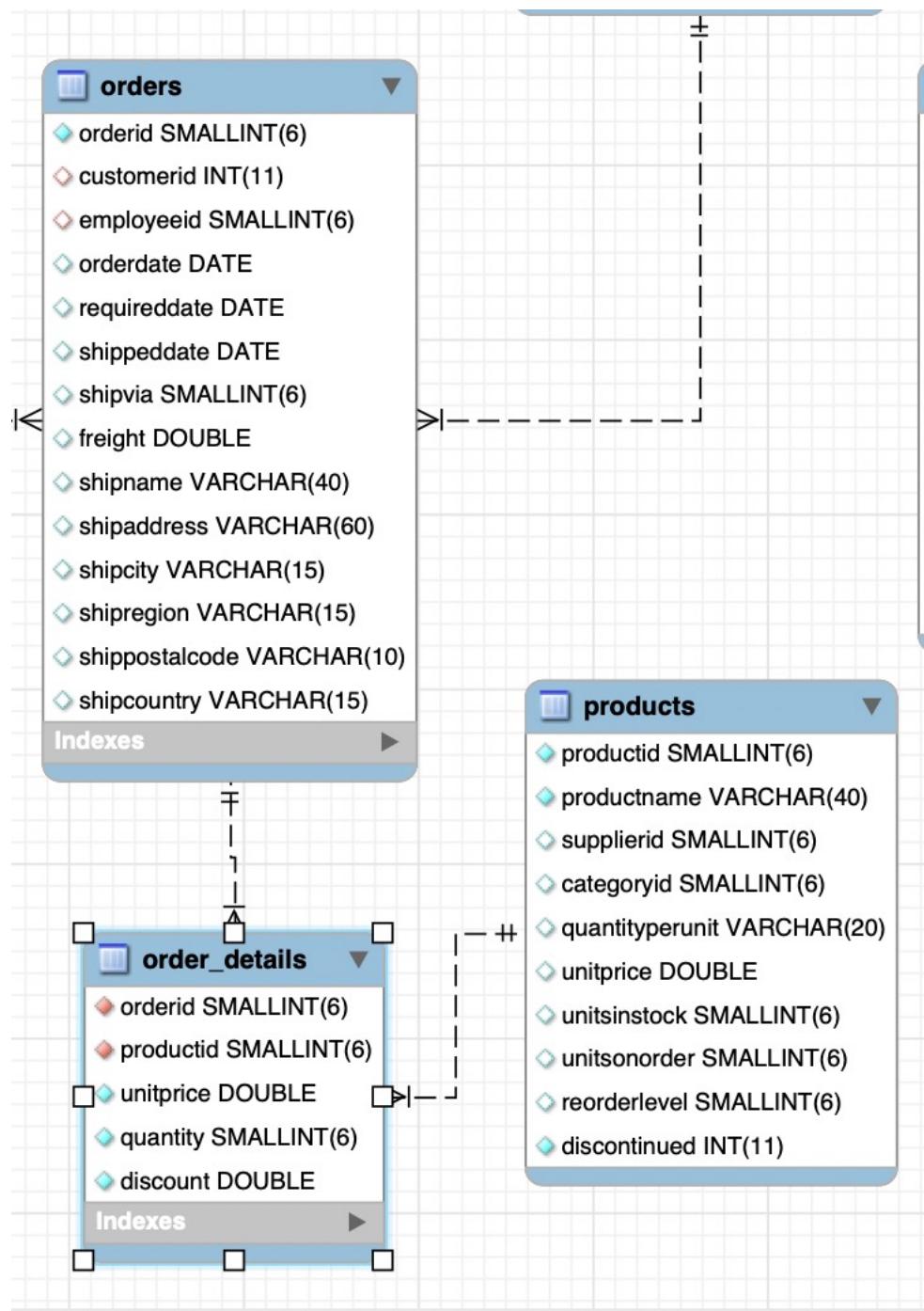
Hvilke medarbejdere hjalp "Ernst Handel"?



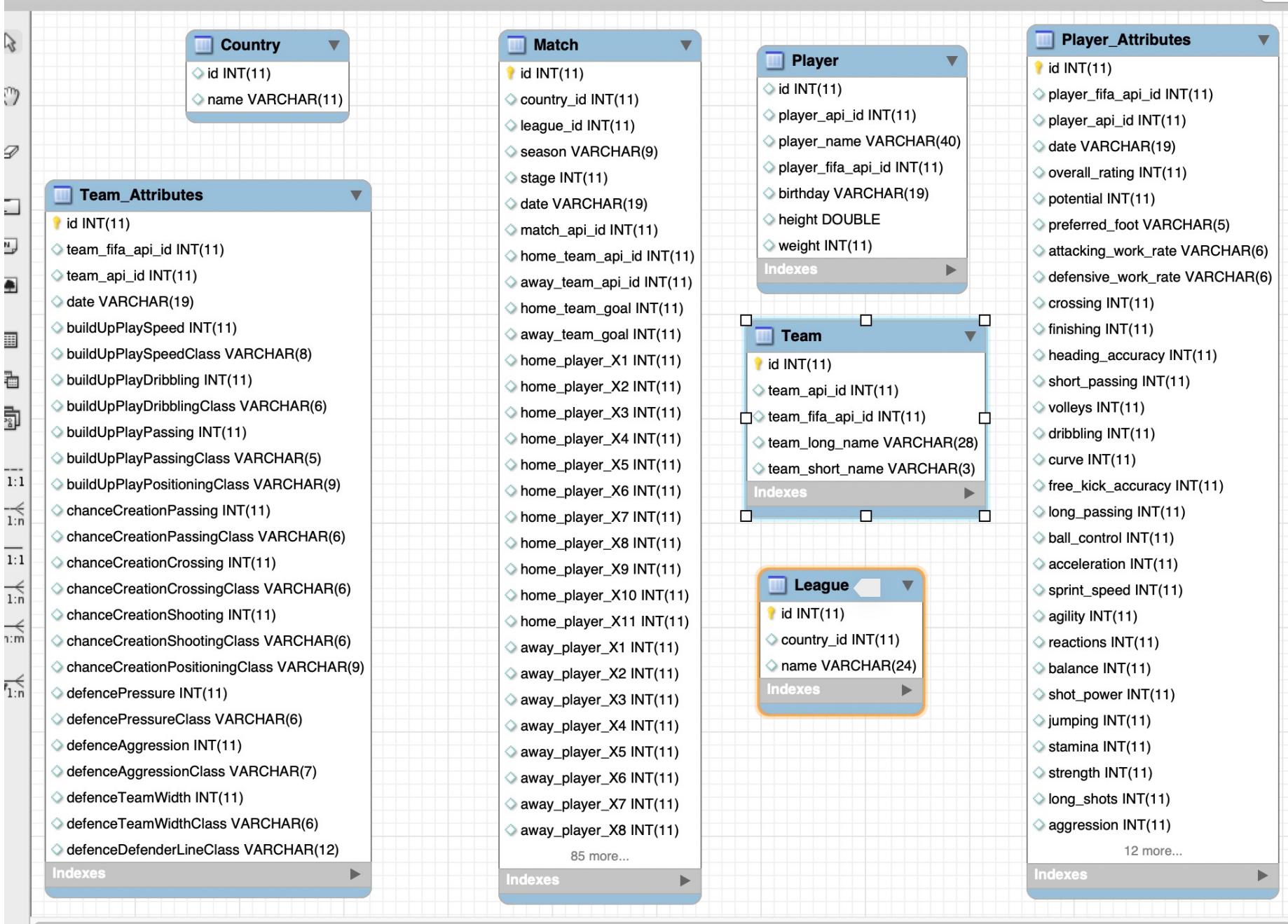
# Northwind databasen

Hvilke tre produkter solgte mest

- mængde
- kroner



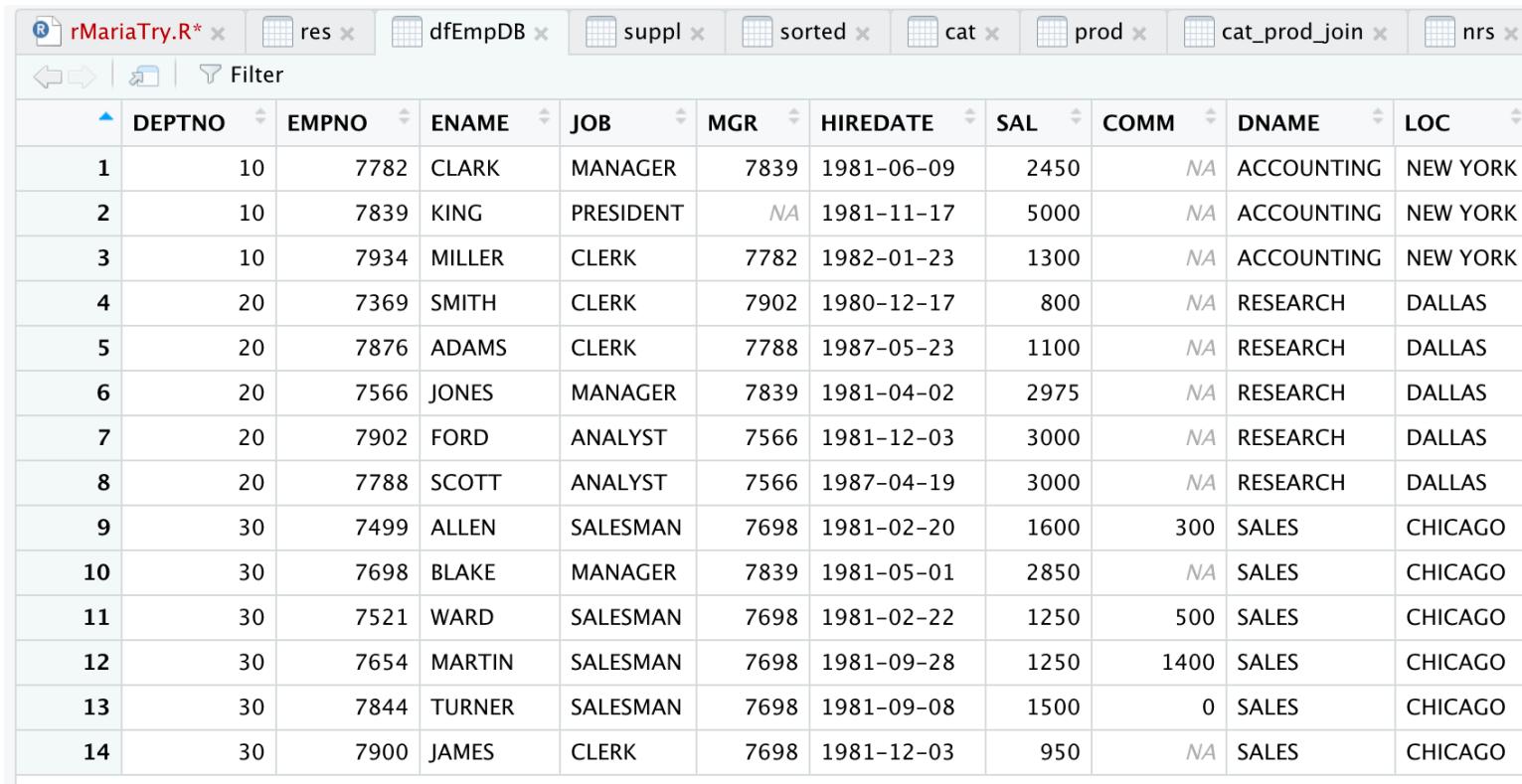
# SOCCKER databasen



# MySQL from R

- Connecting and disconnecting
  - Connecting to and disconnecting from databases
    - `dbConnect(MariaDB(), ..)`
- Tables
  - Reading and writing entire tables
    - `dbWriteTable(con, "mycarstable", mycarsdf)`
    - `mycardf <- dbReadTable(con, "mycarstable")`
- Results
  - More control for sending queries and executing statements
    - `dbGetQuery(con, "SELECT * FROM city limit 3")`
    - `dbExecute (con, "INSERT INTO city (Name,Population) VALUES ('Lviv',123123)")`

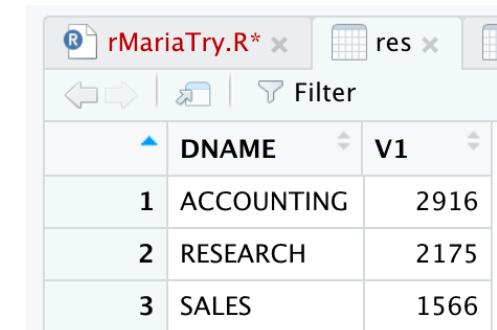
# MySQL from R



The screenshot shows the RStudio interface with the 'rMariaTry.R\*' script open. The top tab bar includes 'rMariaTry.R\*', 'res', 'dfEmpDB', 'suppl', 'sorted', 'cat', 'prod', 'cat\_prod\_join', and 'nrs'. Below the tabs is a toolbar with icons for back, forward, and filter. The main area displays the 'EMP' table from the 'dfEmpDB' database. The columns are DEPTNO, EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DNAME, and LOC. The data consists of 14 rows, each representing an employee record.

|    | DEPTNO | EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DNAME      | LOC      |
|----|--------|-------|--------|-----------|------|------------|------|------|------------|----------|
| 1  | 10     | 7782  | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NA   | ACCOUNTING | NEW YORK |
| 2  | 10     | 7839  | KING   | PRESIDENT | NA   | 1981-11-17 | 5000 | NA   | ACCOUNTING | NEW YORK |
| 3  | 10     | 7934  | MILLER | CLERK     | 7782 | 1982-01-23 | 1300 | NA   | ACCOUNTING | NEW YORK |
| 4  | 20     | 7369  | SMITH  | CLERK     | 7902 | 1980-12-17 | 800  | NA   | RESEARCH   | DALLAS   |
| 5  | 20     | 7876  | ADAMS  | CLERK     | 7788 | 1987-05-23 | 1100 | NA   | RESEARCH   | DALLAS   |
| 6  | 20     | 7566  | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NA   | RESEARCH   | DALLAS   |
| 7  | 20     | 7902  | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NA   | RESEARCH   | DALLAS   |
| 8  | 20     | 7788  | SCOTT  | ANALYST   | 7566 | 1987-04-19 | 3000 | NA   | RESEARCH   | DALLAS   |
| 9  | 30     | 7499  | ALLEN  | SALESMAN  | 7698 | 1981-02-20 | 1600 | 300  | SALES      | CHICAGO  |
| 10 | 30     | 7698  | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NA   | SALES      | CHICAGO  |
| 11 | 30     | 7521  | WARD   | SALESMAN  | 7698 | 1981-02-22 | 1250 | 500  | SALES      | CHICAGO  |
| 12 | 30     | 7654  | MARTIN | SALESMAN  | 7698 | 1981-09-28 | 1250 | 1400 | SALES      | CHICAGO  |
| 13 | 30     | 7844  | TURNER | SALESMAN  | 7698 | 1981-09-08 | 1500 | 0    | SALES      | CHICAGO  |
| 14 | 30     | 7900  | JAMES  | CLERK     | 7698 | 1981-12-03 | 950  | NA   | SALES      | CHICAGO  |

Indlæse tabellerne hver for sig.  
Udfør operationer i R  
(merge,aggregate)



The screenshot shows the RStudio interface with the 'rMariaTry.R\*' script open. The top tab bar includes 'rMariaTry.R\*', 'res', and 'dfEmpDB'. Below the tabs is a toolbar with icons for back, forward, and filter. The main area displays a summary table with columns 'DNAME' and 'V1'. The data consists of 3 rows, showing the count of employees for each department: ACCOUNTING (2916), RESEARCH (2175), and SALES (1566).

|   | DNAME      | V1   |
|---|------------|------|
| 1 | ACCOUNTING | 2916 |
| 2 | RESEARCH   | 2175 |
| 3 | SALES      | 1566 |