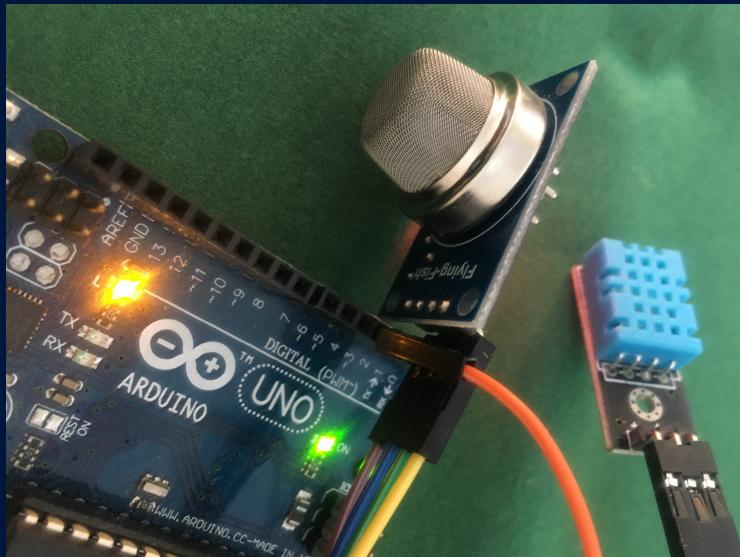


COPENHAGEN BUSINESS ACADEMY



DATA ENGINEERING



Agenda - CRUD

- SQL
 - SELECT (conditions,join,aggregation på Northwind)
 - struktur
 - øvelser
 - UPDATE og INSERT (Northwind)
 - Demo
 - Øvelser
 - CREATE (Cars to EMP)
 - MySQL SKEMA og datatyper
- SQL i R
 - SQL fra R (conditions,join,aggregation på World og Northwind)
 - SELECT
 - UPDATE & INSERT

SQL

Data Definition (DDL)

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

Data Manipulation (DML)

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

Cheat Sheet

Create / Delete Database

```
CREATE DATABASE mabase  
CREATE DATABASE mabase CHARACTER SET utf8  
DROP DATABASE mabase  
ALTER DATABASE mabase CHARACTER SET utf8
```

Browsing

```
SHOW DATABASES  
SHOW TABLES  
SHOW FIELDS FROM table / DESCRIBE table  
SHOW CREATE TABLE table  
SHOW PROCESSLIST  
KILL process_number
```

Cheat Sheet

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
```

Cheat Sheet

Select – Join

```
SELECT ... FROM t1 JOIN t2 ON t1.id1 = t2.id2 WHERE condition
```

```
SELECT ... FROM t1 LEFT JOIN t2 ON t1.id1 = t2.id2 WHERE condition
```

```
SELECT ... FROM t1 JOIN (t2 JOIN t3 ON ...) ON ...
```

Insert

```
INSERT INTO table1 (field1, field2, ...) VALUES (value1, value2, ...)
```

Update

```
UPDATE table1 SET field1=new_value1 WHERE condition
```

```
UPDATE table1, table2 SET field1=new_value1, field2=new_value2, ... WHERE  
table1.id1 = table2.id2 AND condition
```

Cheat Sheet

Backup Database to SQL File

```
mysqldump -u Username -p DatabaseName > databaseName_backup.sql
```

Restore from backup SQL File

```
mysql - u Username -p DatabaseName < databaseName_backup.sql
```

Reset Root Password

```
$ /etc/init.d/mysql stop
```

```
$ mysqld_safe --skip-grant-tables
```

```
$ mysql # on another terminal
```

```
mysql> UPDATE mysql.user SET password=PASSWORD('new_pass') WHERE user='root';
```

```
## Switch back to the mysqld_safe terminal and kill the process using Control + \
```

```
$ /etc/init.d/mysql start
```

Repair Tables After Unclean Shutdown

```
mysqlcheck --all-databases
```

```
mysqlcheck --all-databases --fast
```

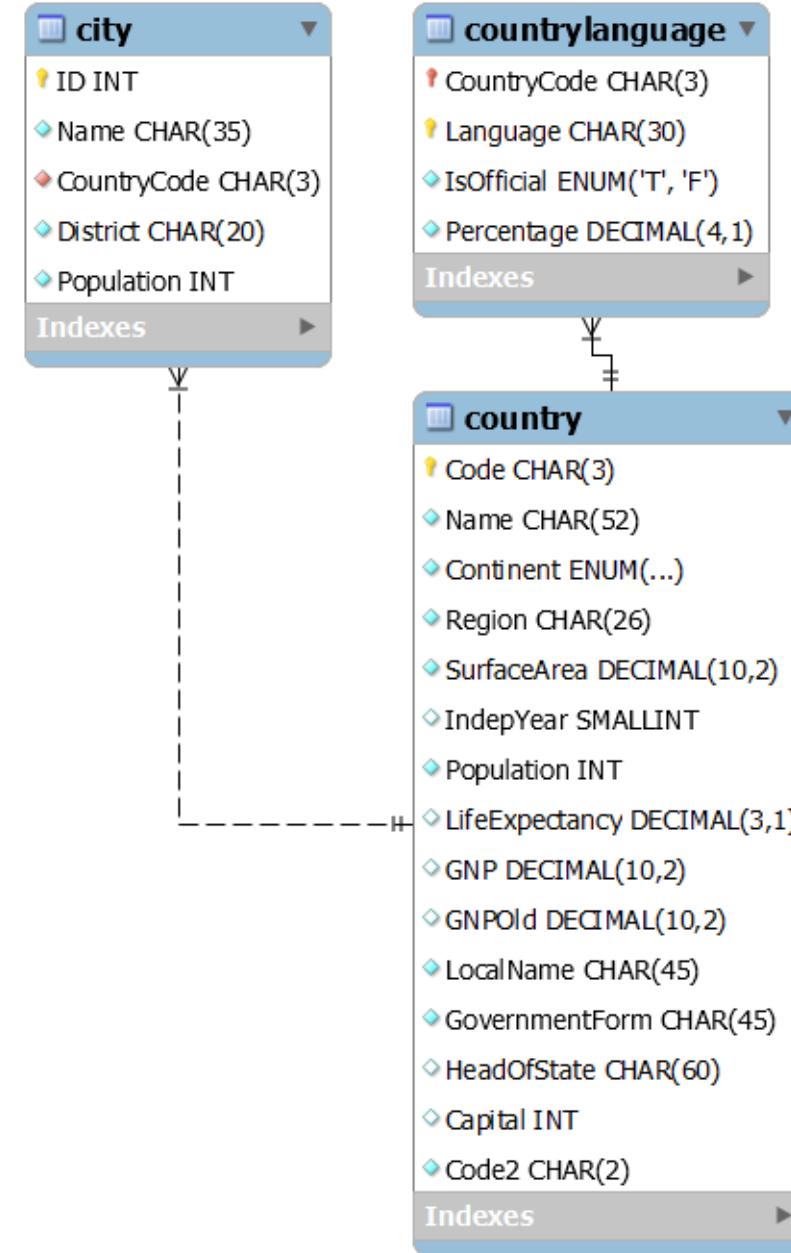
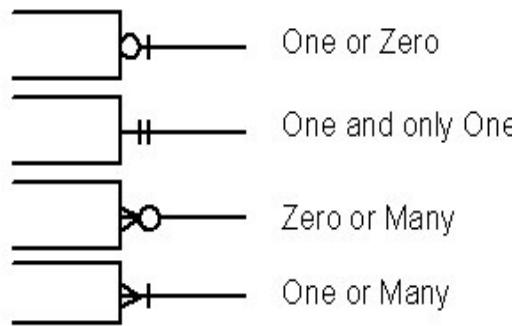
MySQL Design - datatyper

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)

World DB ER-diagram

howto: Reverse engineer in WorkBench

Summary of Crow's Foot Notation



WORLD Databasen – øvelse fra R

Hent script fra <https://github.com/cphstud/DALE23W46>

SQL

name	uc
Singapore	1.1264
Cocos (Keeling) Islands	1.1167
Gibraltar	1.0810
Macao	0.9249
Pitcairn	0.8400
South Korea	0.8325
Saint Pierre and Miquelon	0.8297
Falkland Islands	0.8180
Grenada	0.0492
Nepal	0.0473
Bermuda	0.0462
Burundi	0.0448
Uganda	0.0409
Rwanda	0.0370
Barbados	0.0225
Saint Lucia	0.0149
Netherlands Antilles	0.0108
Bhutan	0.0104

Result 19

R

▲	Name.x	urb
1	Singapore	1.12636193
2	Cocos (Keeling) Islands	1.11666667
3	Gibraltar	1.08100000
4	Macao	0.92494715
5	Pitcairn	0.84000000
6	South Korea	0.83254831
7	Saint Pierre and Miquelon	0.82971429
8	Falkland Islands	0.81800000
9	Grenada	0.04915957
10	Nepal	0.04732148
11	Bermuda	0.04615385
12	Burundi	0.04480956
13	Uganda	0.04090366
14	Rwanda	0.03698435
15	Barbados	0.02248148
16	Saint Lucia	0.01494156
17	Netherlands Antilles	0.01080645
18	Bhutan	0.01035782

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 \('Lviv',123123\)"\)`](#)
`and fetch(req)`
 - [`rows = dbExecute \(con, "UPDATE city set Name = 'Kurt' WHERE id = 123"\)`](#)

WORLD Databasen

Spørgsmål til "world" databasen

1) I hvilket distrikt ligger byen 'Stanley'?

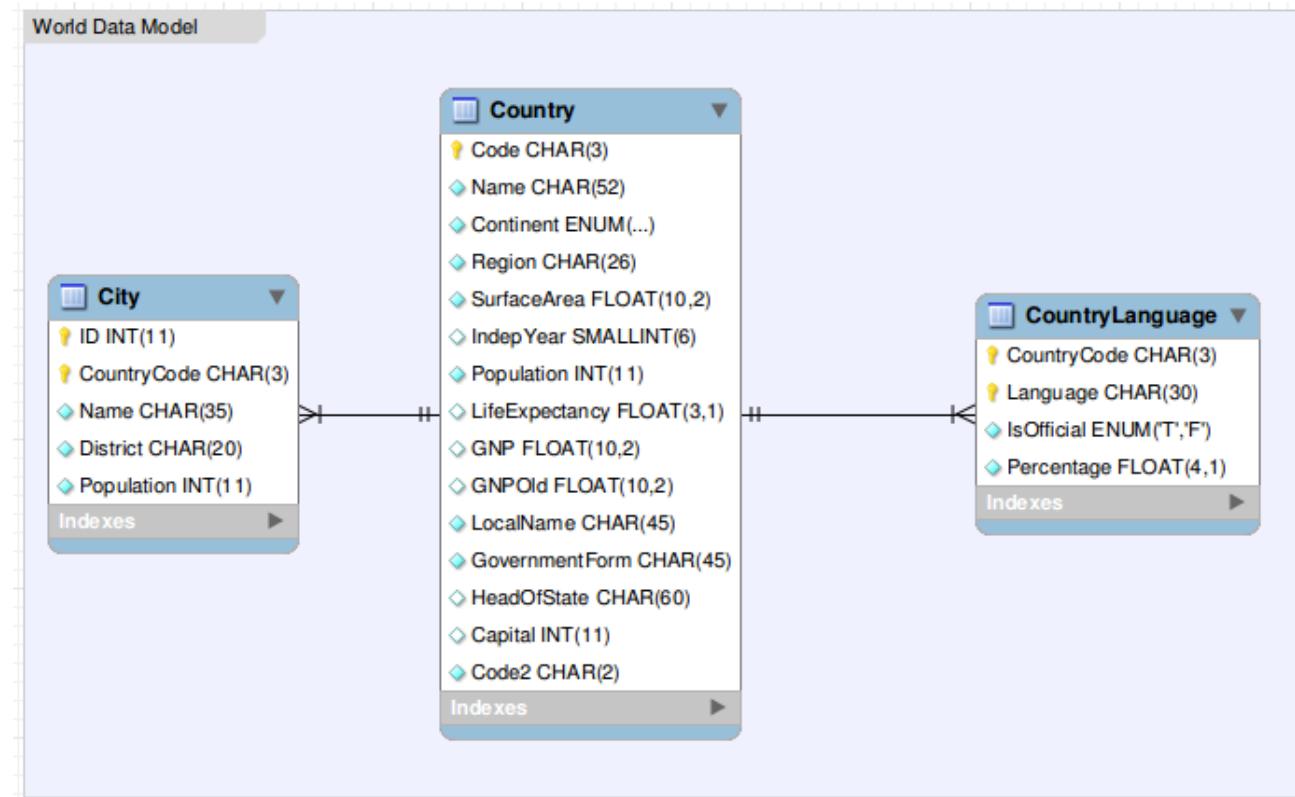
2) Er færøsk et officielt sprog på Færøerne?

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

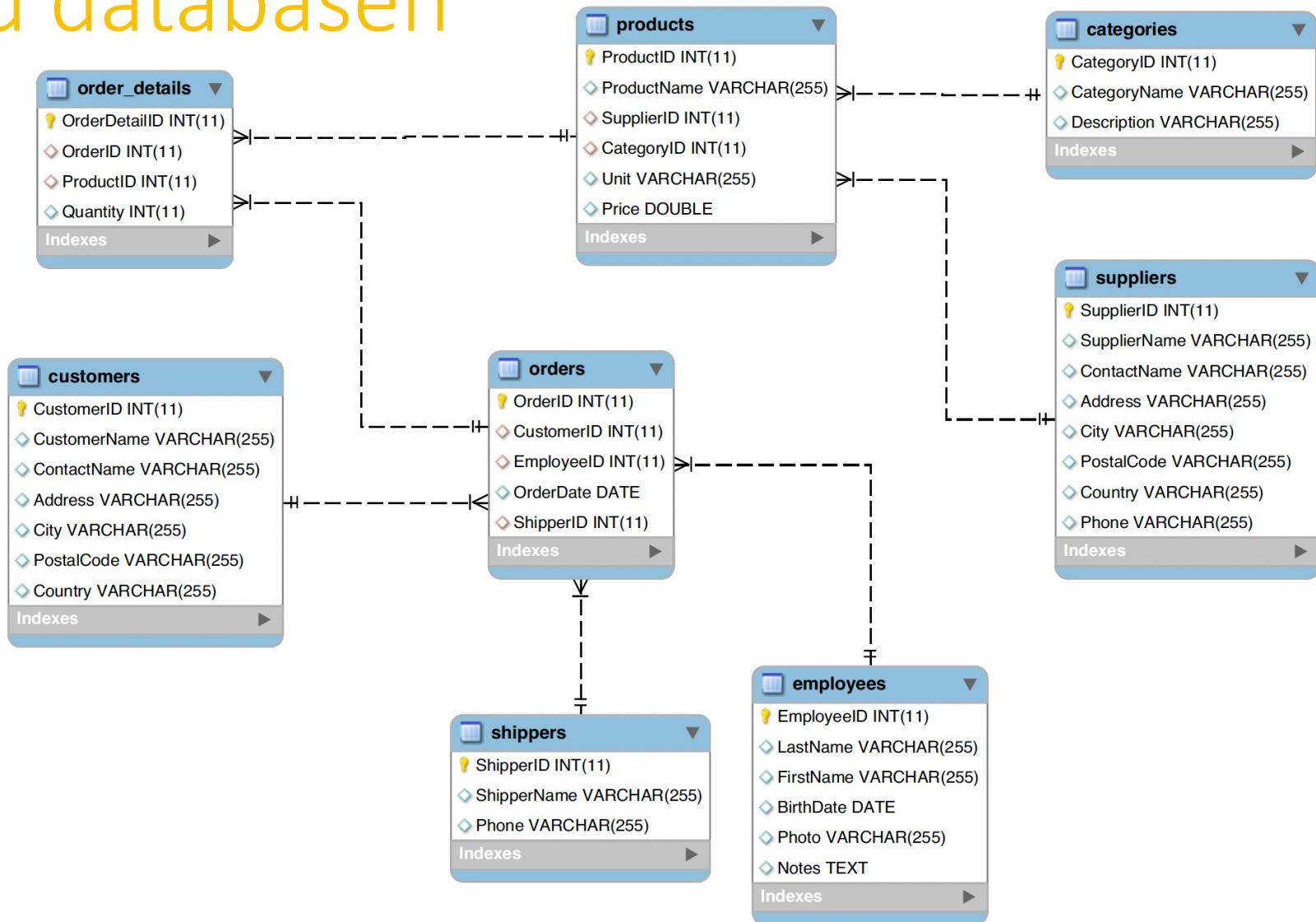
4) Hvilket land har det mindste areal?

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

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



Northwind databasen



Northwind databasen

1. Show all data for the first 10 rows in the table products.
2. Show only productid, productname and unitinstock for the first 10 rows in the table products.
3. Show all data for the first 10 rows in the table products sorted by decreasing unitsinstock.
4. Show all data for the first 10 rows in the table products where unitprice is less than 10.
5. Show all data for the first 10 rows in the table products where unitsinstock is between 0 and 10.
6. Show all data for the first 10 rows in the table products where productname is geitost.
7. Show only productname, productid and discontinued for the first 10 rows in the table products where supplierid is 7 or 9.
10. Show all data for the first 10 rows in the table products where productname contains ost.
15. Show the number of rows in the table products.
16. Show the number of rows in the table products with a column name "Number of products".
17. Show the average of unitprice with 3 decimals in the table products with a column name "Average price".
18. Show all different country in the table suppliers.
20. Show the first 10 productname and companyname (for the tables products and suppliers where supplierid is equal in the two tables).
21. Show the first 10 productname and companyname as an inner join (of the tables products and suppliers where supplierid is equal in the two tables).
22. Show the three most selling employees measured in quantity .
23. Are there companies that haven't placed any orders?

Northwind databasen - videodude

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

Northwind databasen

Hvilke tre kunder placerede flest ordrer?

companyname	most orders
Ernst Handel	30
QUICK-Stop	28
Hungry Owl All-Night Grocers	19
Folk och fä HB	19
HILARION-Abastos	18
Berglunds snabbköp	18
Rattlesnake Canyon Grocery	18
Bon app'	17

companyname	customerid	mm
Save-a-lot Markets	SAVEA	115673
Ernst Handel	ERNSH	113237
Hungry Owl All-Night Grocers	HUNGO	57317
Rattlesnake Canyon Grocery	RATTC	52246
Hanari Carnes	HANAR	34101
Folk och fä HB	FOLKO	32556
Mère Paillardé	MEREP	32204
Königlich Essen	KOENE	31746
Queen Cozinha	QUEEN	30226

Northwind databasen

Hvilke kunder placerede ingen ordre?

	orderdate	companynamne
▶	HULL	FISSA Fabrica Inter. Salchic...
	HULL	Paris spécialités
	1996-07-04	Vins et alcools Chevalier
	1996-07-05	Toms Spezialitäten
	1996-07-08	Hanari Carnes
	1996-07-08	Victuailles en stock

Northwind databasen

Hvilke medarbejdere hjalp "Ernst Handel"?

	lastname	help count
►	Davolio	5
►	Peacock	5
	Leverling	4
►	King	4
	Callahan	4
►	Dodsworth	3
	Fuller	3
►	Suyama	2

Northwind databasen

Hvilke tre produkter solgte mest

- mængde
- kroner

solgt	ProductName	categoryname
▶ 1577	Camembert Pierrot	Dairy Products
1496	Raclette Courdavault	Dairy Products
1397	Gorgonzola Telino	Dairy Products
1263	Gnocchi di nonna Alice	Grains/Cereals
1158	Pavlova	Confections
1155	Rhönbräu Klosterbier	Beverages
1125	Guaraná Fantástica	Beverages
1103	Boston Crab Meat	Seafood
1083	Tarte au sucre	Confections
1057	Chang	Beverages

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 Jeff

- Install MySQL & WorkBench
- Create database – Newspaper
- Add table – tbl_newspaper_sr
- Add columns
- Add user
- Create an R Script that connects
 - RMariaDB
 - dbConnect
 - dbListTables
 - conf-file with pw
- Storing data with SQL
 - INSERT + string-halløj
- Query data with SQL
- Storing data with R
 - dbSendQuery(storiesDb, query)
 - query <- paste() + dbSendQuery(storiesDb, query)
 - sql-errors
- Storing data with SQL and csv
 - dbWriteTable(db, df, "table", row.names=F)
- Selecting data from a table with SQL using R

```
CREATE TABLE newspaper_search_results.tbl_newspaper_search_results (
    id INT NOT NULL AUTO_INCREMENT,
    story_title VARCHAR(99) NULL,
    story_date_published DATETIME NULL,
    story_url VARCHAR(99) NULL,
    search_term_used VARCHAR(45) NULL,
    PRIMARY KEY (id));
```

Remove leading spaces	LTRIM (' MySql') = 'MySql'
Remove trailing spaces	RTRIM (' MySql ') = ' MySql'
Return the leftmost number of characters as specified	LEFT (' MySql ', 2) = 'My'

MySQL from R Jeff

```
SELECT ( COUNT( CONCAT( MONTH(story_date_published), YEAR(story_date_published) ) ) ) as 'count'  
FROM tbl_newspaper_search_results  
WHERE search_term_used='German+Submarine'  
GROUP BY YEAR(story_date_published),MONTH(story_date_published)  
ORDER BY YEAR(story_date_published),MONTH(story_date_published);
```

Return concatenated string
CONCAT ('My', 'S', 'QL') = 'MySQL'
CONCAT ('My', NULL, 'QL') = NULL
CONCAT (14.3) = '14.3'



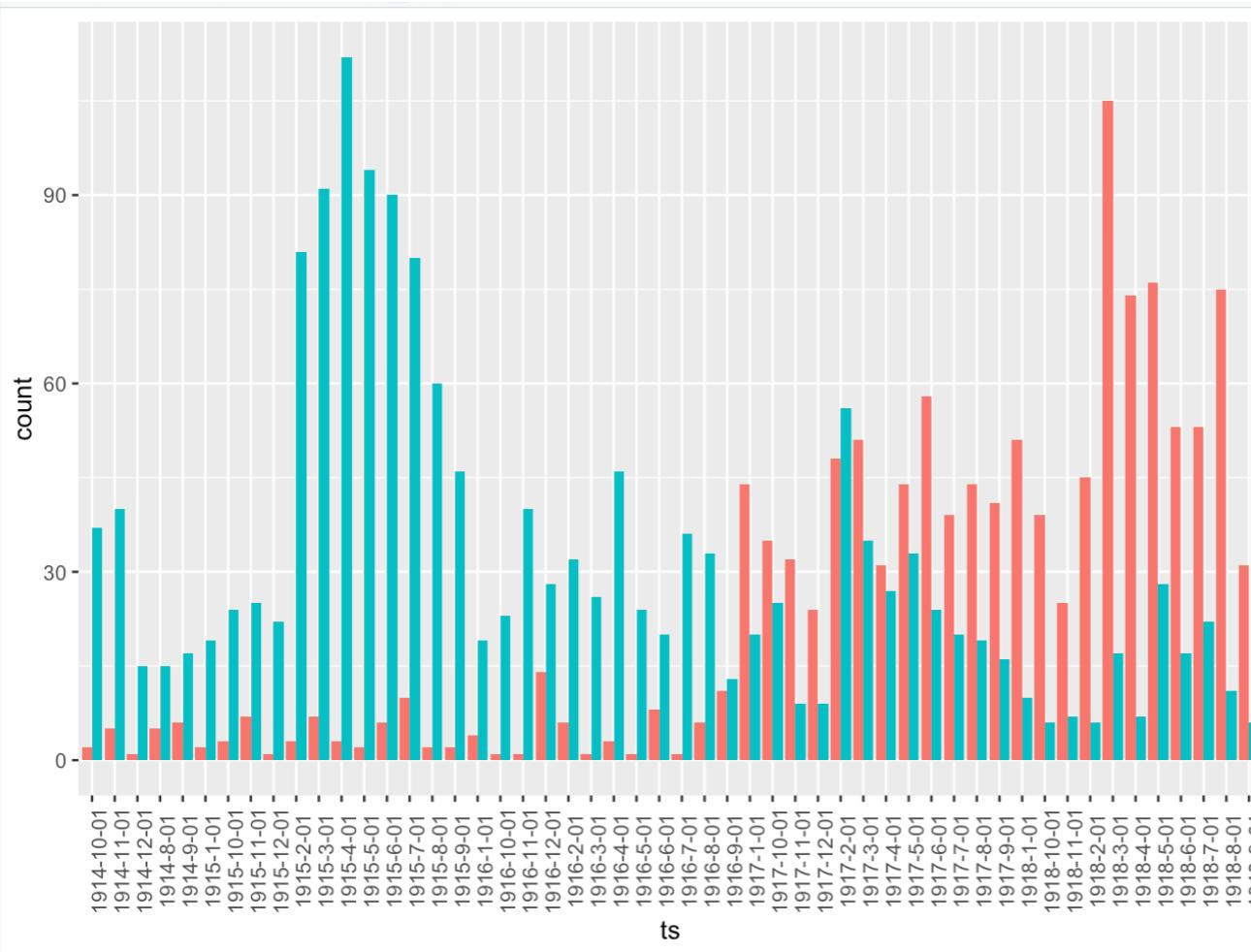
```
select Month("1915-04-03 00:00:00");  
select Year("1915-04-03 00:00:00");  
select day("1915-04-03 00:00:00");
```

😊 SELECT MONTH(story_date_published), YEAR(story_date_published), story_title
FROM tbl_newspaper_search_results
WHERE search_term_used='German+Submarine'
order by 1;

😊 SELECT * FROM tbl_newspaper_search_results

MySQL from R Jeff

Wrangling in R ..



```
dfTotal <- dbReadTable(condb,"tbl_newspaper_search_results")
dfTotal <- dfTotal %>%
  mutate(year=year(story_date_published),month=month(story_date_published))
dfTotalMC <- dfTotal %>%
  group_by(year,month,search_term_used) %>%
  summarise(count=n()) %>% ungroup()
dfTotalMC <- dfTotalMC %>% mutate(ts=(paste0(year,"-",month,"-01")))
```

search_term_used
AllotmentAndGarden
German+Submarine

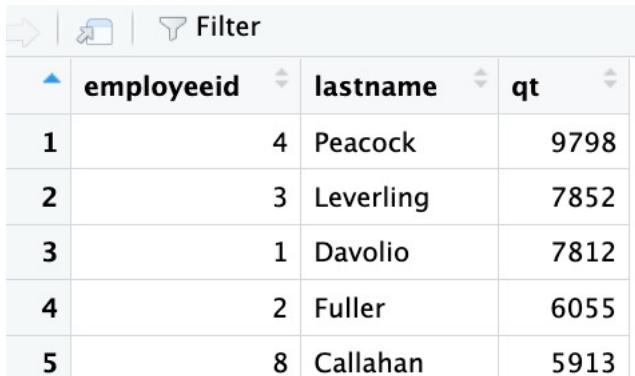
.. or in SQL?

```
SELECT ( COUNT(CONCAT(MONTH(story_date_published),
YEAR(story_date_published)))) as 'count'
FROM tbl_newspaper_search_results
WHERE search_term_used='German+Submarine'
GROUP BY YEAR(story_date_published),MONTH(story_date_published)
ORDER BY YEAR(story_date_published),MONTH(story_date_published);
```

Northwind databasen - videodude

Hvem var de fem mest sælgende ansatte (mængde og pris)?

R



	employeeid	lastname	qt
1	4	Peacock	9798
2	3	Leverling	7852
3	1	Davolio	7812
4	2	Fuller	6055
5	8	Callahan	5913

SQL

employeeid	lastname	mængde solgt
4	Peacock	9798
3	Leverling	7852
1	Davolio	7812
2	Fuller	6055
8	Callahan	5913

employeeid	lastname	mængde solgt i kroner
4	Peacock	250187
3	Leverling	213051
1	Davolio	202144
2	Fuller	177749
7	King	141296

MySQL from R NorthWind

LEFT JOIN



Everything on the left
+
anything on the right that matches

```
SELECT *  
FROM TABLE_1  
LEFT JOIN TABLE_2  
ON TABLE_1.KEY = TABLE_2.KEY
```

RIGHT JOIN



Everything on the right
+
anything on the left that matches

```
SELECT *  
FROM TABLE_1  
RIGHT JOIN TABLE_2  
ON TABLE_1.KEY = TABLE_2.KEY
```

OUTER JOIN



Everything on the right
+
Everything on the left

```
SELECT *  
FROM TABLE_1  
OUTER JOIN TABLE_2  
ON TABLE_1.KEY = TABLE_2.KEY
```

INNER JOIN



Only the things that match on the left AND the right

```
SELECT *  
FROM TABLE_1  
INNER JOIN TABLE_2  
ON TABLE_1.KEY = TABLE_2.KEY
```

Combine Data Sets

a	x1	x2
A	1	
B	2	
C	3	

b	x1	x3
A	T	
B	F	
D	T	



Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

x1	x3	x2
A	T	1
B	F	2
D	T	NA

x1	x2	x3
A	1	T
B	2	F

x1	x2	x3
A	1	T
B	2	F
C	3	NA

Filtering Joins

x1	x2
A	1
B	2

x1	x2
C	3

dplyr::left_join(a, b, by = "x1")

Join matching rows from b to a.

dplyr::right_join(a, b, by = "x1")

Join matching rows from a to b.

dplyr::inner_join(a, b, by = "x1")

Join data. Retain only rows in both sets.

dplyr::full_join(a, b, by = "x1")

Join data. Retain all values, all rows.

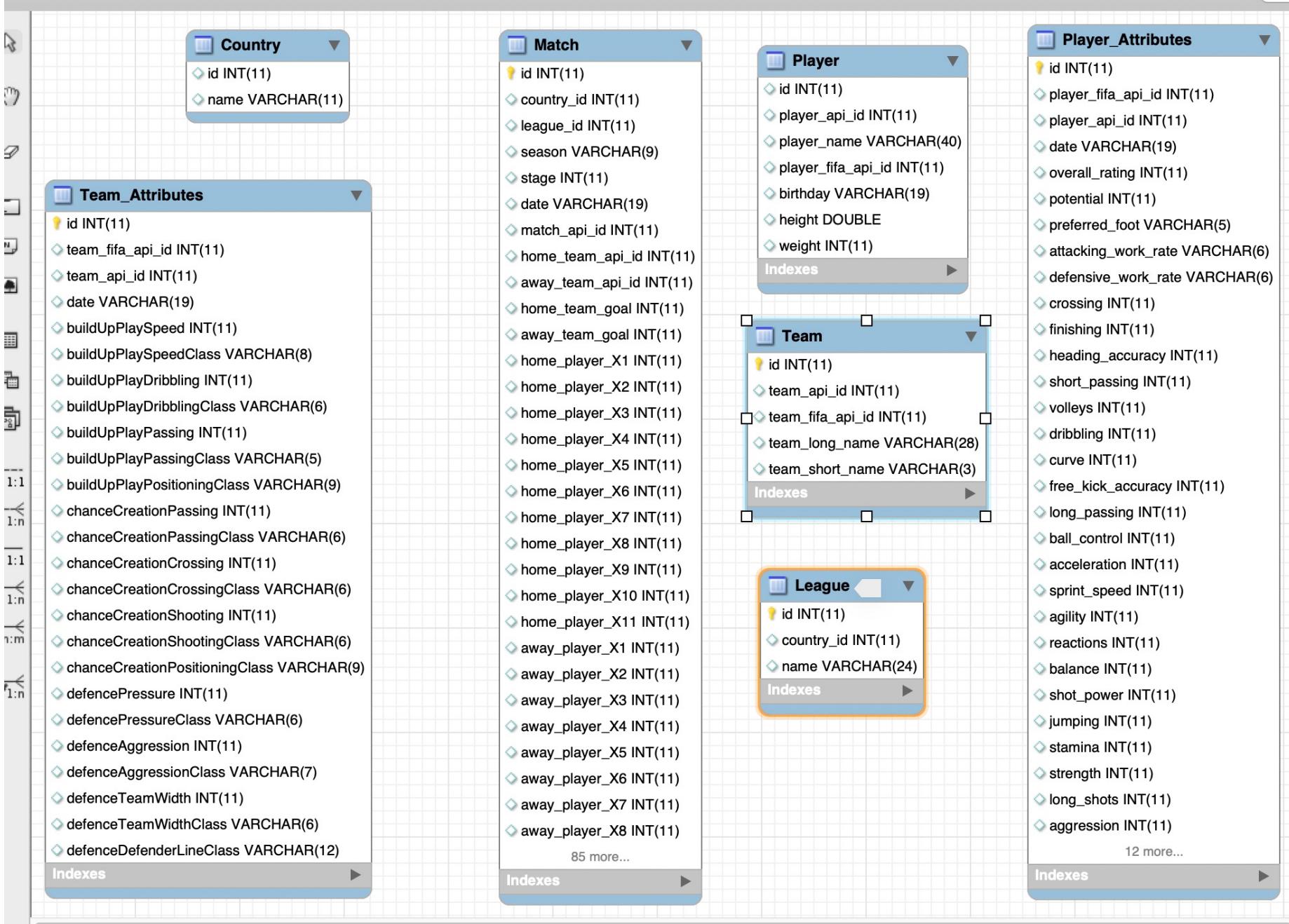
dplyr::semi_join(a, b, by = "x1")

All rows in a that have a match in b.

dplyr::anti_join(a, b, by = "x1")

All rows in a that do not have a match in b.

SOCCKER databasen

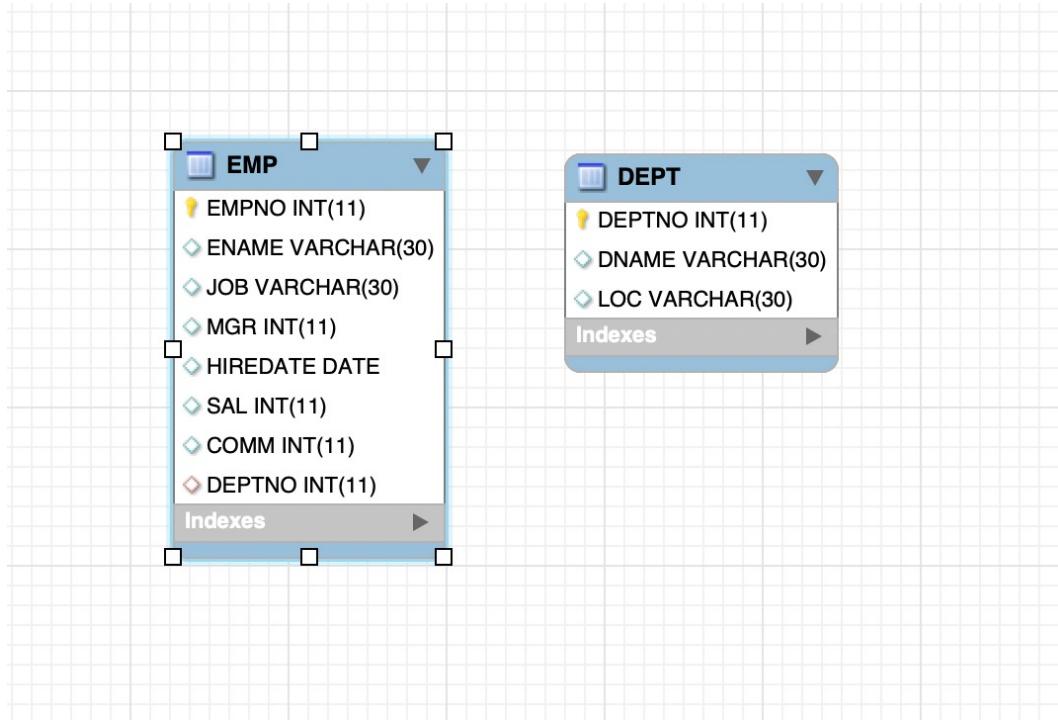


SOCCKER databasen

Result Grid Filter Rows: Search Export:

	player_api_id	player_name	rating
▶	30981	Lionel Messi	92.1923
◀	30893	Cristiano Ronaldo	91.2800
▶	30924	Franck Ribery	88.4583
◀	30955	Andres Iniesta	88.3200
▶	35724	Zlatan Ibrahimovic	88.2857
◀	30834	Arjen Robben	87.8400
▶	39854	Xavi Hernandez	87.6364
◀	30829	Wayne Rooney	87.2222

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:

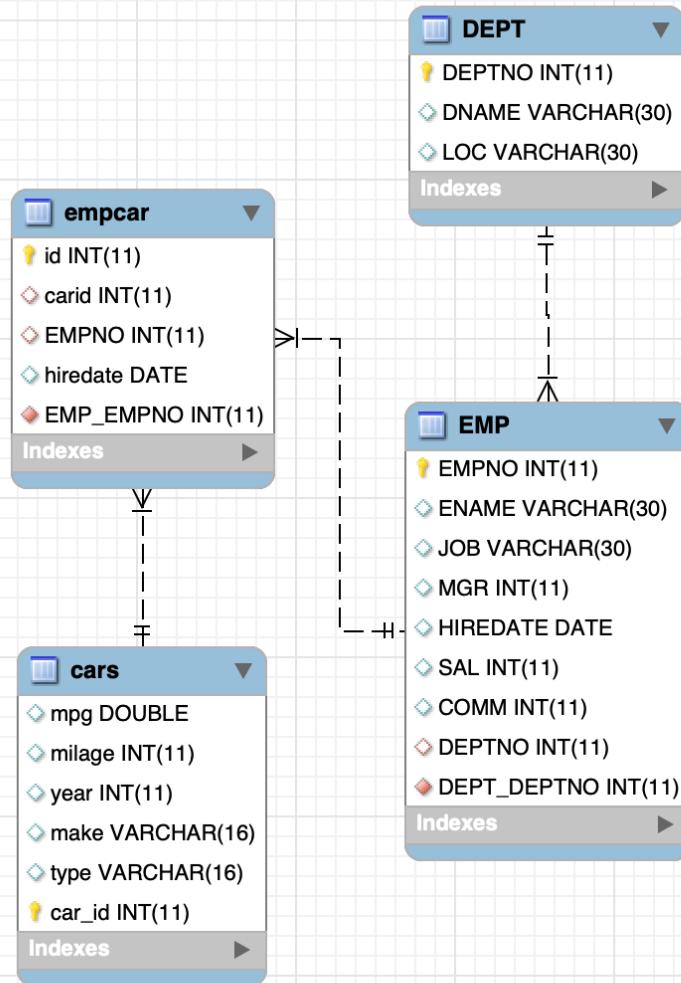
gennemsnit	afdeling
1566	SALES
2175	RESEARCH
2916	ACCOUNTING

Og samme resultat i R

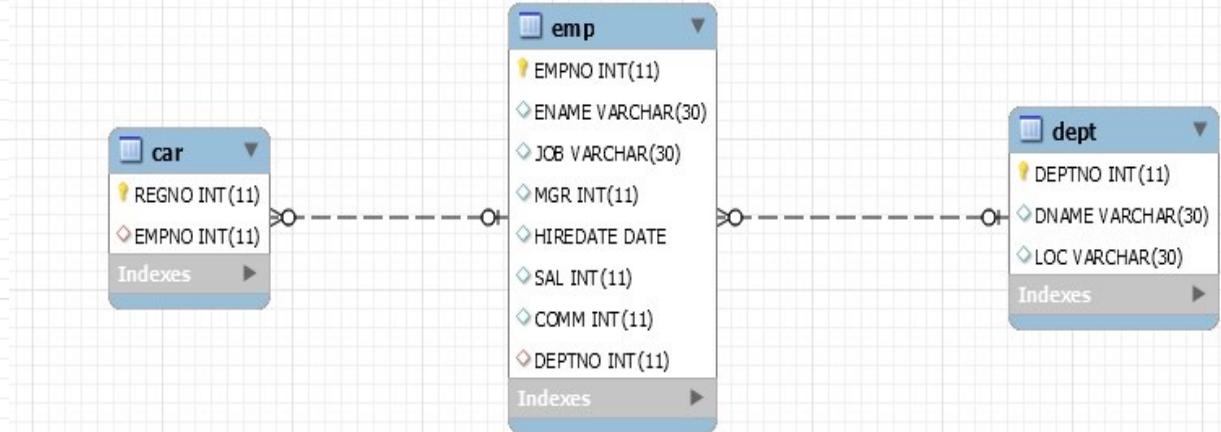
	DNAME	V1
1	ACCOUNTING	2916
2	RESEARCH	2175
3	SALES	1566

EMPLOYEE Databasen – nu med biler

Den "svære"



Den "lette"



Your tasks:

- a) Import EMP-database
- b) Create the new table CAR
- c) Create table empcar (linktable)
- d) Insert minimum 4 registrations in the car table
- e) 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.

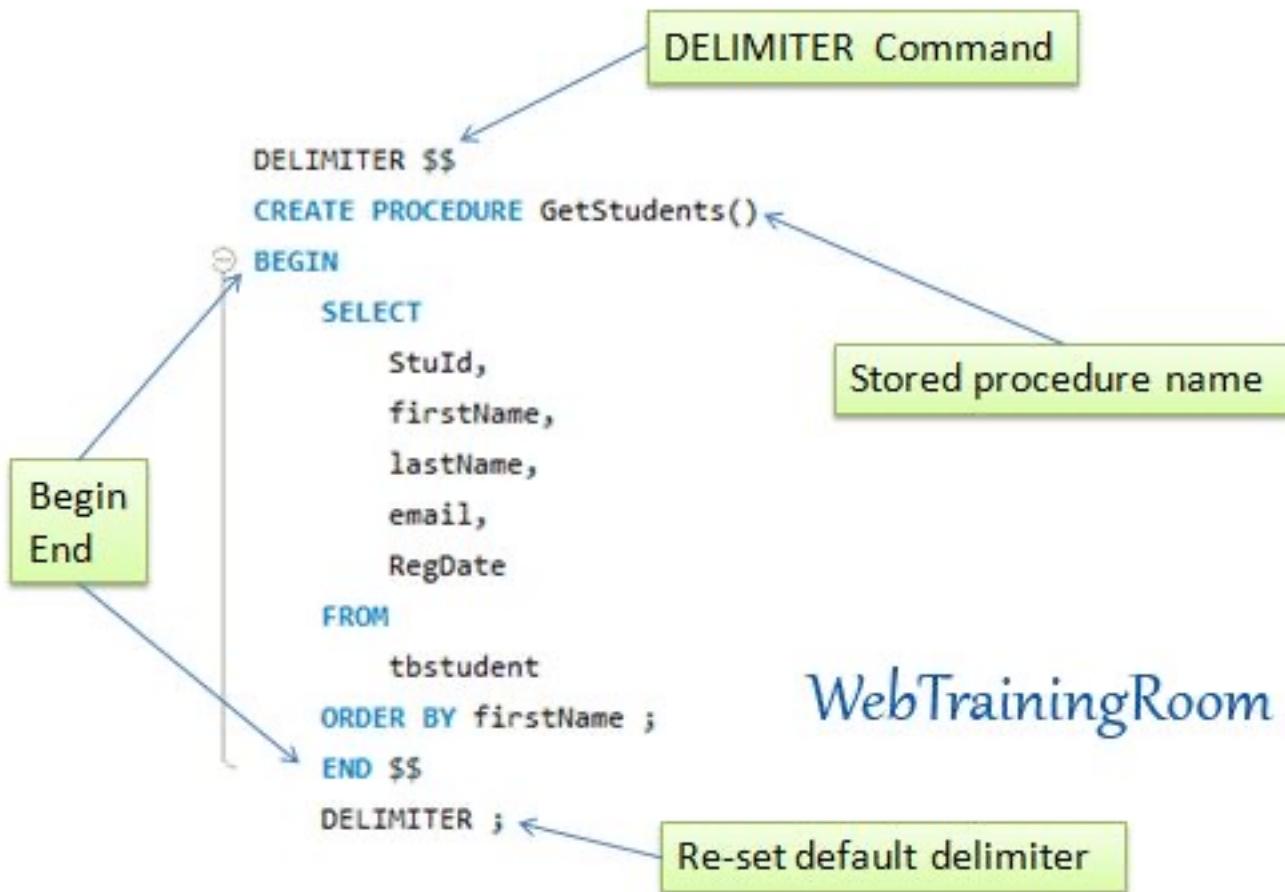
EMPLOYEE Databasen – nu med biler & R

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

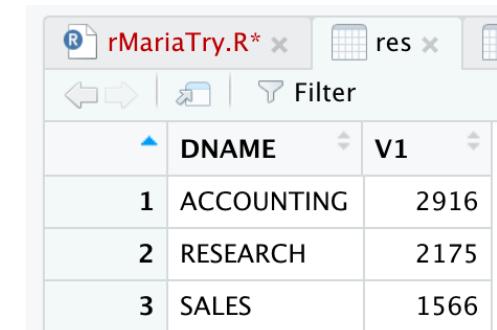
MySQL from R



The screenshot shows the RStudio interface with the 'rMariaTry.R*' file 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*' file 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