Task #1

Create a relational database on computers in computer classroom 308, using MySQL server and any client.

Save tables using Export database (in *.sql format)

Prepare the SELECT, INSERT and UPDATE commands for each typical operation of the intended application – record the task and your solution, at least seven tasks (prepare them in a text file).

(for example – book library: add new user, add new book, record a book reservation, list of free books, list titles of books loaned by user specified by name, change user's mail address, erase book from library, return a book by user, count, how many books are free for loan).

Task #2

Create the same database, using MS Access 2013.

Task #3

Draw the E-R diagram for your database

SQL server

RDBMS separates the database from an application.

Application communicates with RDBMS using SQL.

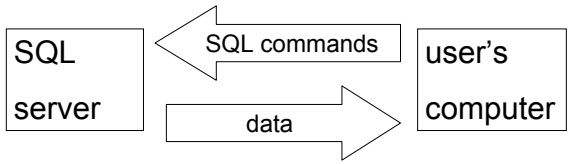
SQL = Structured Query Language

(originally not a programming language – basic standard is missing commands for cycles and variables; many extensions exist with this kind of commands)

Client - Server

Thick client

Client program assembles SQL commands and generates user interface (on the user's computer)

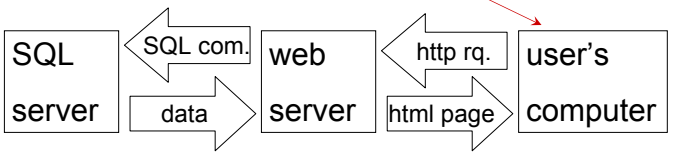


Client - Server

Thin client

SQL commands are assembled on the web server (it usually is, but don't have to be the same server – SQL server on another computer can make its maintenance easier).

As a client, we can use any web browser – in real, this is much more complicated program then thick client, but universal.



DDL, DML

Data Definition Language

Data Manipulation Language

Data Control Language

All of them are part of SQL.

Micka

-One of many SQL Clients

-Diploma thesis by Ing. Zbyněk Munzar, FEE CTU

-Can be retrieved from www.senon.cz/micka/

More common clients:

phpMyAdmin – part of MySQL installation

phpWebAdmin – can be installed on SQL server

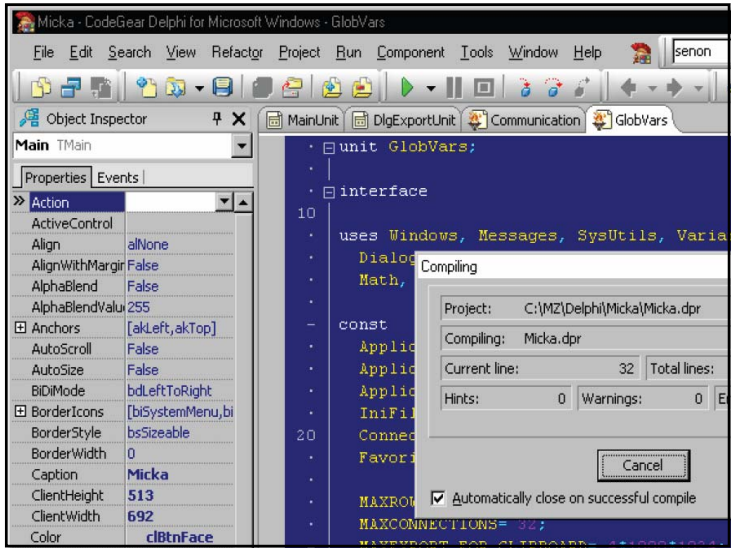
HeidiSQL (generally recommended)

Emma

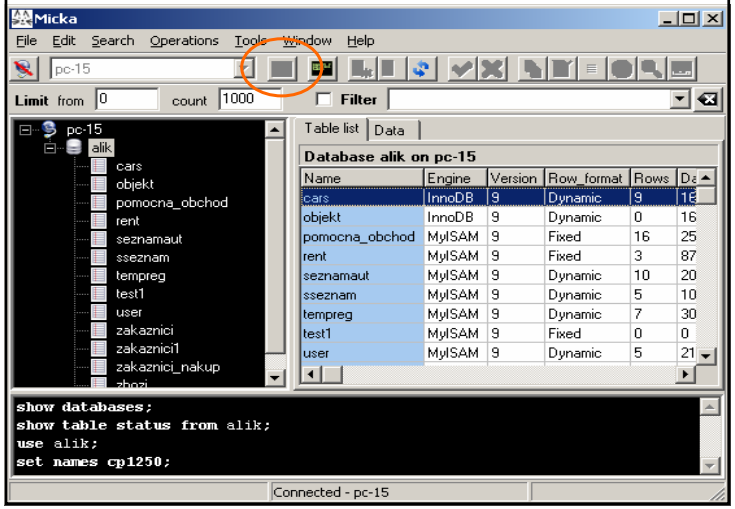
MySQL GUI Tools

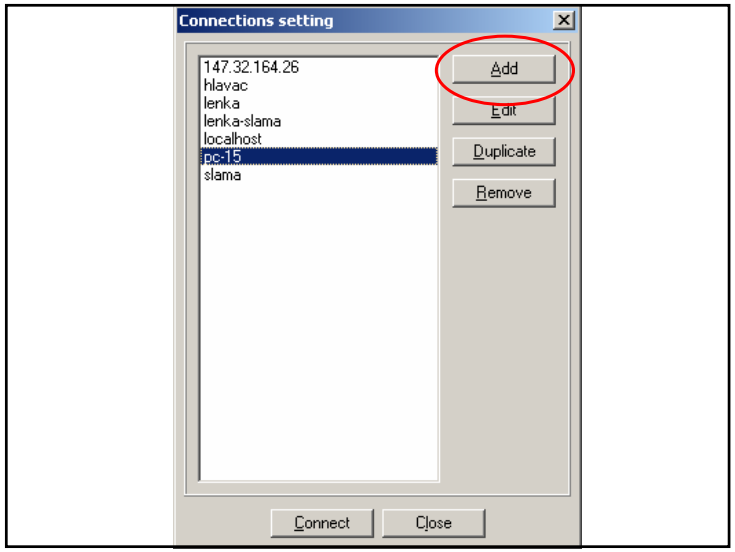
Charakteristika aplikace

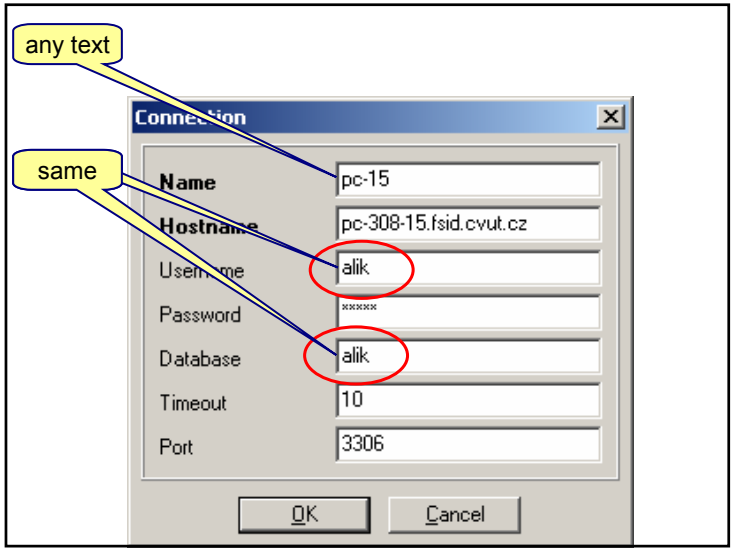
- Micka
- Desktopová aplikace
- Vyvinuta v Object Pascalu v prostředí Delphi
- V tuto chvíli pouze pro 32 bitové MS Windows
- Univerzální klient pro správu MySQL databází
- Standardní GUI
- Snadná rozšiřitelnost
- Potřeba neustálého vývoje

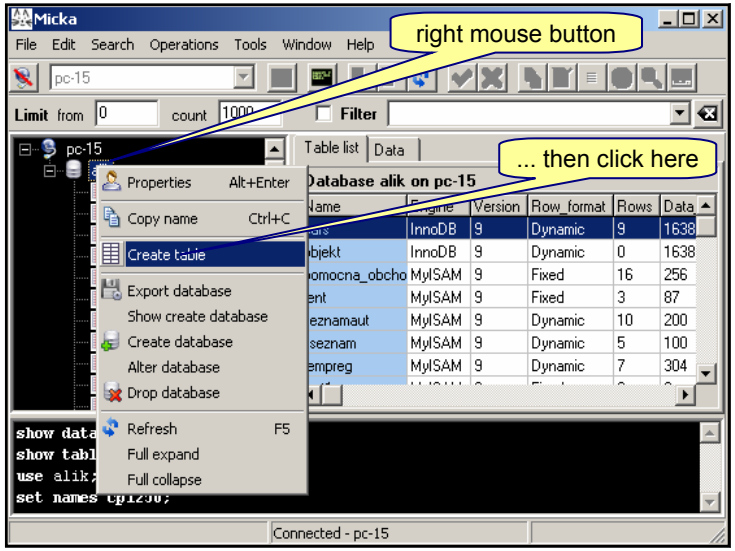


Example – client for access SQL database





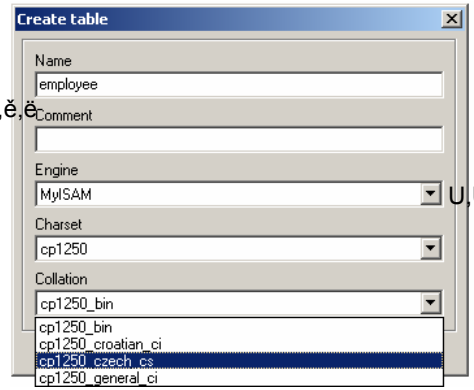




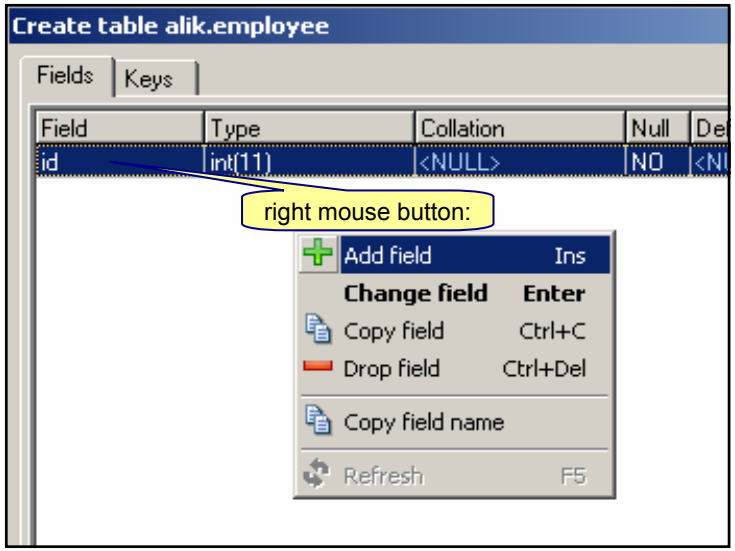
A,Á,Ä,a,á,ä
B,b
C,c
Č,č
D,d
Ď,ď
E,É,Ě,Ë,é,ě,ë
F,f
G,g
H,h
CH,Ch,ch
I, Í, i, í
J,j
K,k
L,Ľ,Ĺ,l,ľ,í
M,m
N,n
Ň,ň

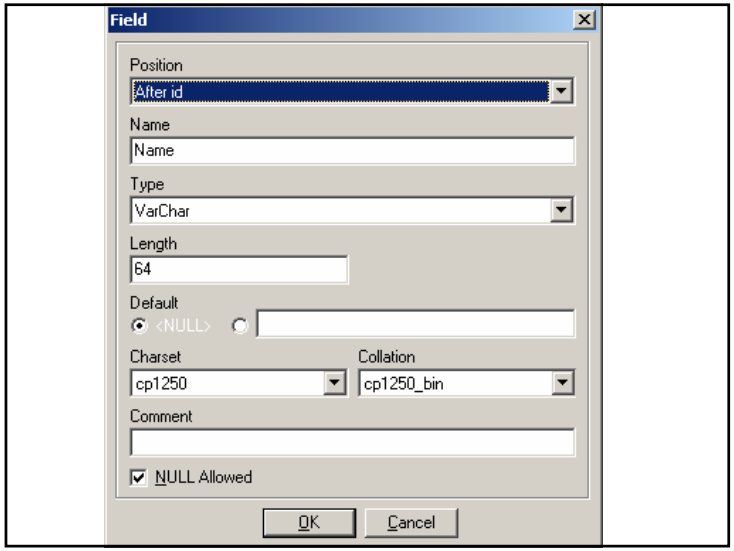
CP 1250 cs sorting

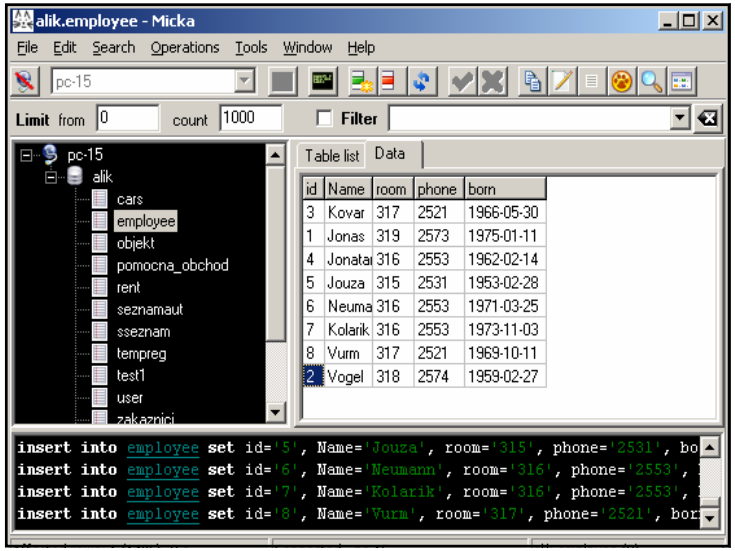
O,Ó,Ö,o,ó,ö
P,p
Q,q
R,r,Ř,ŕ
Ř,ř
S,s
Š,š
T,t
Ť,ť
U,Ú,Ů,u,ú,ů
V,v
W,w
X,x
Y,Ý,y,ý
Z,z
Ž,ž

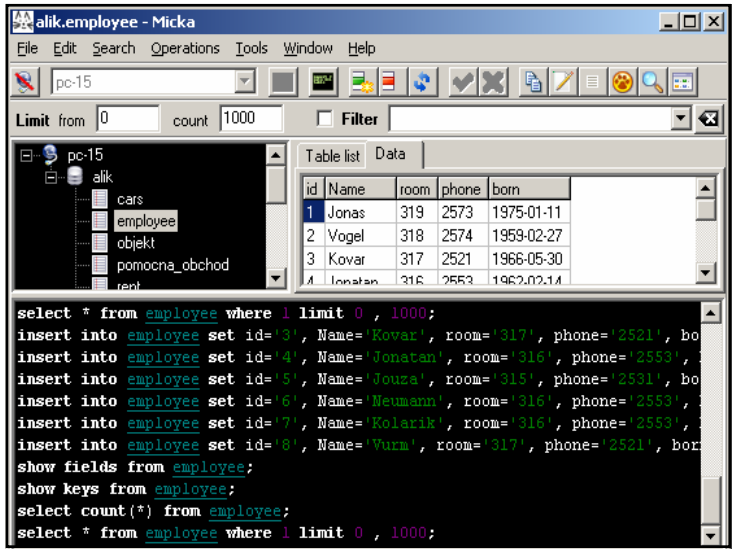


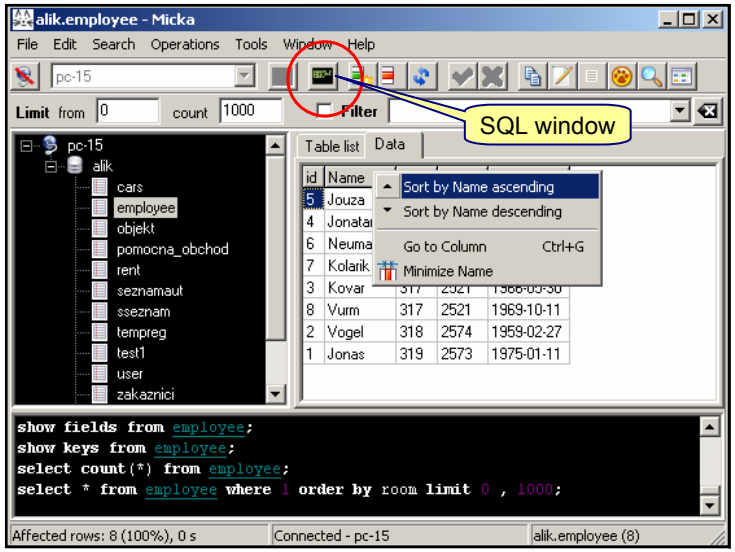
http://en.wikipedia.org/wiki/Diacritical_marks











select

SELECT * FROM tablename ;

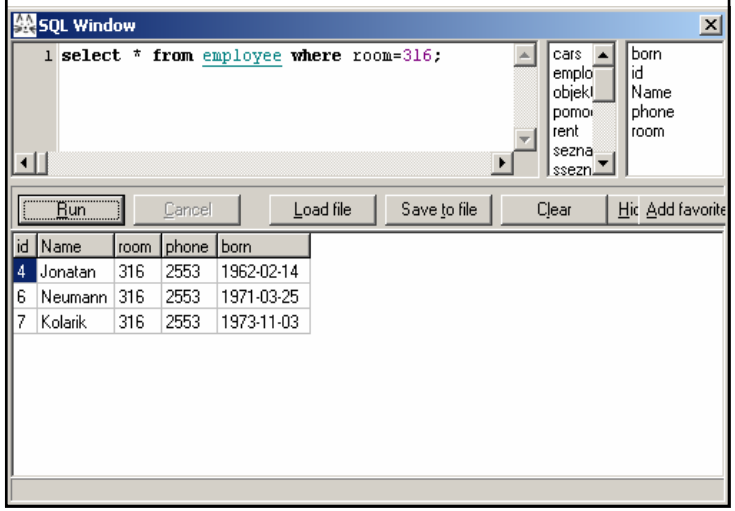
SELECT Name,room FROM employee ;

SELECT Name,room FROM employee
WHERE room = '316';

SELECT Name,room FROM employee
ORDER BY room;

SELECT Name,room FROM employee
WHERE room = '316' ORDER BY Name;

Result example:



important notice for this ppt

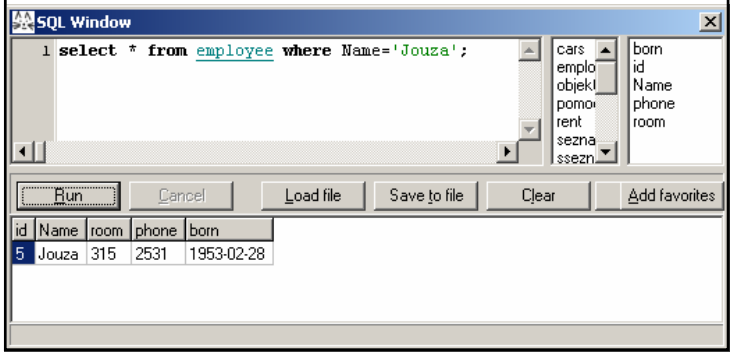
This text was written in MS Powerpoint.

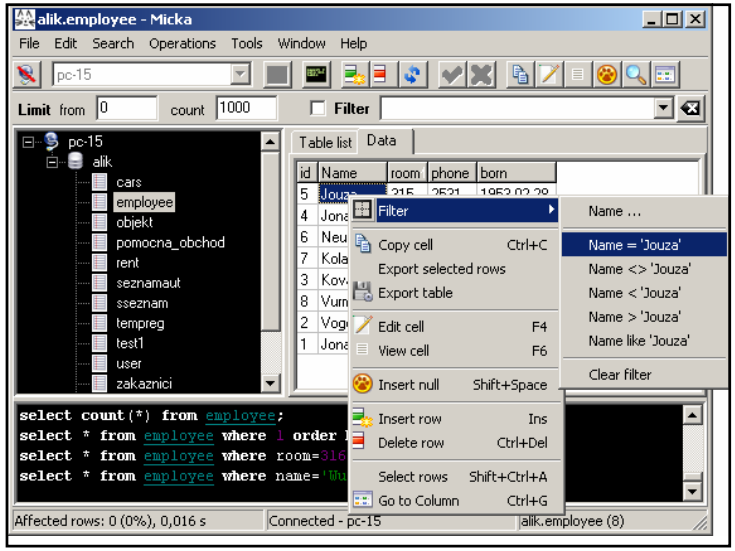
In this software, all apostrophes is changed to special characters (left and right quotation). It could be repaired or this feature switched off, but both is unreliable.

So, please, after copy to your
"SQL command window", substitute them to the correct one. Thanks.

select

SELECT Name,room FROM employee
WHERE Name = 'Jouza';





select

select * from employee order by id where
CONVERT('Name' USING utf8) = 'Jouza'

Typical character set:
USA – ASCII, 7 bit, no conversion function, DOS 437
WestEu. – ISO 8859-1, DOS 850, CP1252
CE – ISO 8859-2, DOS 852, CP1250
Greek – ISO 8859-7, CP1253
Cyrillic – ISO 8859-5, DOS 856, CP1251

UTF16
UTF8 – see http://en.wikipedia.org/wiki/UTF-8

select where

http://www.w3schools.com/sql/sql_between.asp

```
SELECT * FROM employee
 WHERE room >= '312' AND room <= '316';
```

OR, XOR

```
SELECT * FROM employee
 WHERE room BETWEEN '312' AND '316';
```

Result is dependent on the database engine; on the MySQL, both examples are the same, but on other systems it can contain only one limit value, or none of them ("between"). It is better to use the first method.

```
SELECT * FROM employee
 WHERE  room IS NULL OR room < '300';

SELECT * FROM employee
 WHERE  room IS NOT NULL ;
```

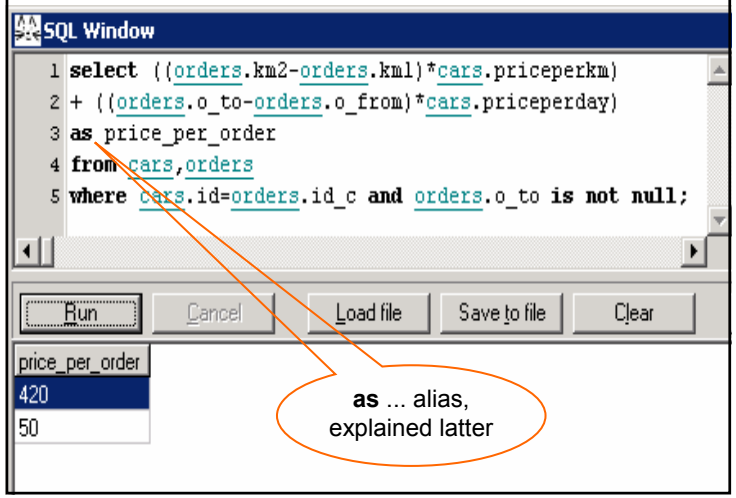
select count

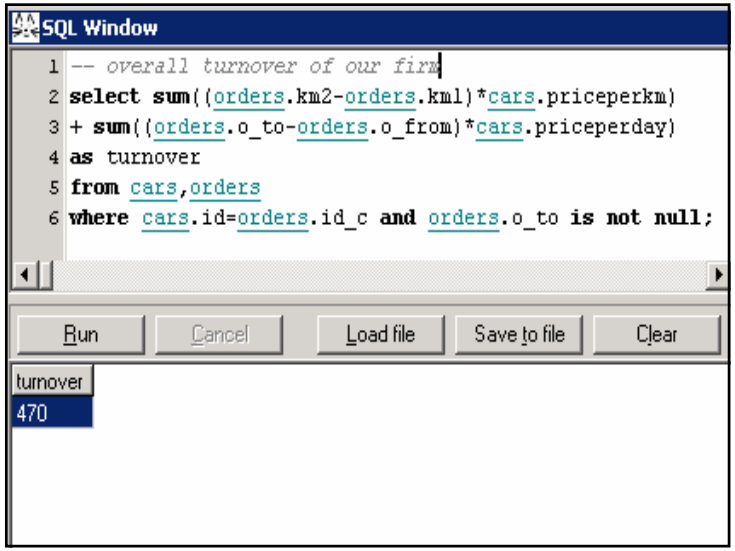
MIN(...

```
SELECT MAX(room) FROM employee;

SELECT COUNT(*) FROM employee
 WHERE room >= '312' AND room <= '317';

SELECT COUNT(DISTINCT(room))
  FROM  employee;

SELECT AVG(price) FROM stock;

SELECT SUM(price * amount) FROM stock;

SELECT MAX(price) FROM stock;
```

see GROUP BY

SQL can work with arithmetic operators (+,-,...)

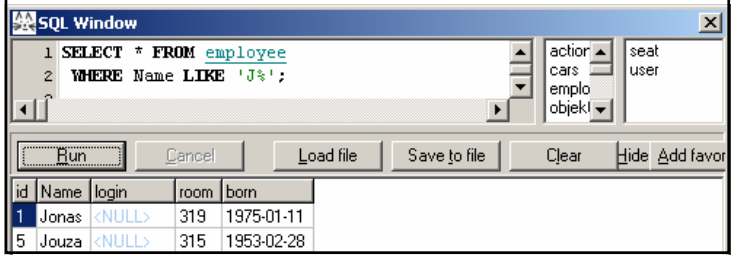




select like: wildcards

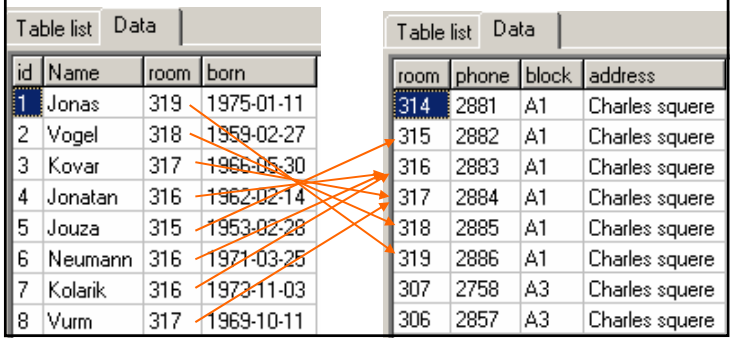
SELECT * FROM employee
WHERE Name LIKE 'J%';

Typical wildcard:
% ... any combination of letters
_ ... just one letter (exactly one, not 'none')



SELECT * FROM table1, table2 WHERE ...;

SELECT Name, room, block, phone
FROM employee, rooms
WHERE employee.room = rooms.room ;



select join

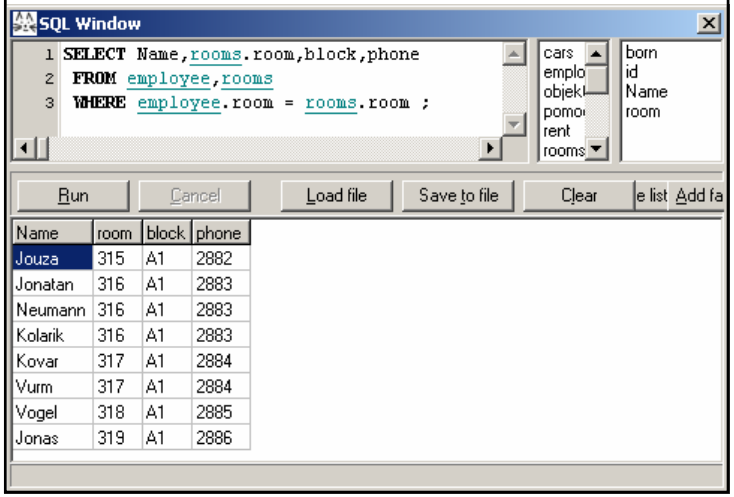
must be rooms.room

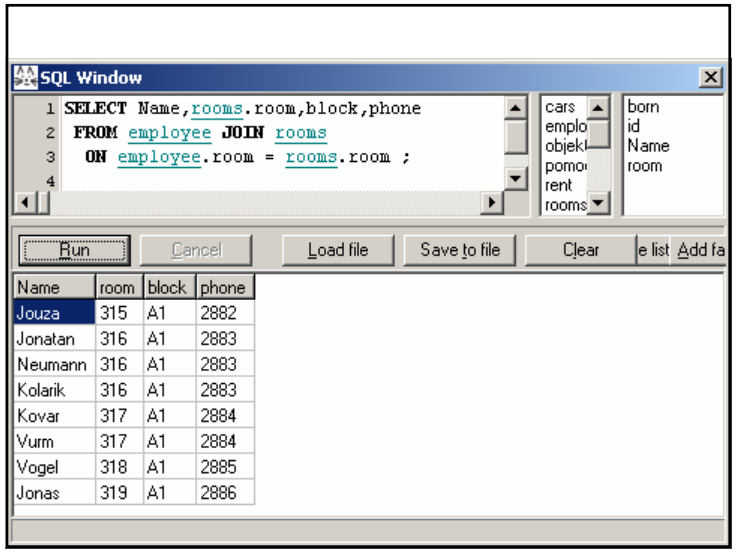
SELECT Name,room,block,phone
FROM employee,rooms
WHERE employee.room = rooms.room ;

is equivalent to

SELECT Name,room,block,phone
FROM employee JOIN rooms
ON employee.room = rooms.room ;

http://www.sql-tutorial.net/SQL-JOIN.asp





select outer join

but we have no empl. w/o room

Modified JOIN can include empty lines:

```
SELECT Name,rooms.room,block,phone
 FROM employee LEFT JOIN rooms
ON employee.room = rooms.room ;
```

is the same as

```
SELECT Name,rooms.room,block,phone
 FROM employee LEFT OUTER JOIN rooms
ON employee.room = rooms.room ;
```

select right outer join

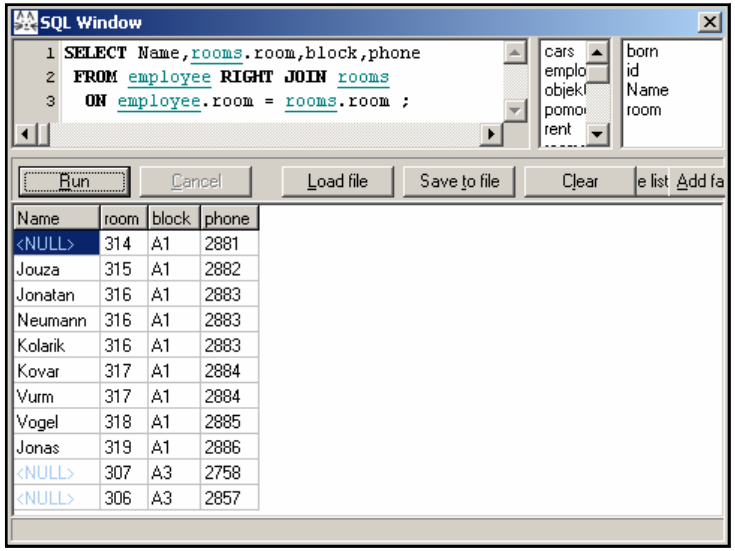
In or case, it could be shown on the „RIGHT“:

```
SELECT Name,rooms.room,block,phone
 FROM employee RIGHT JOIN rooms
ON employee.room = rooms.room ;
```

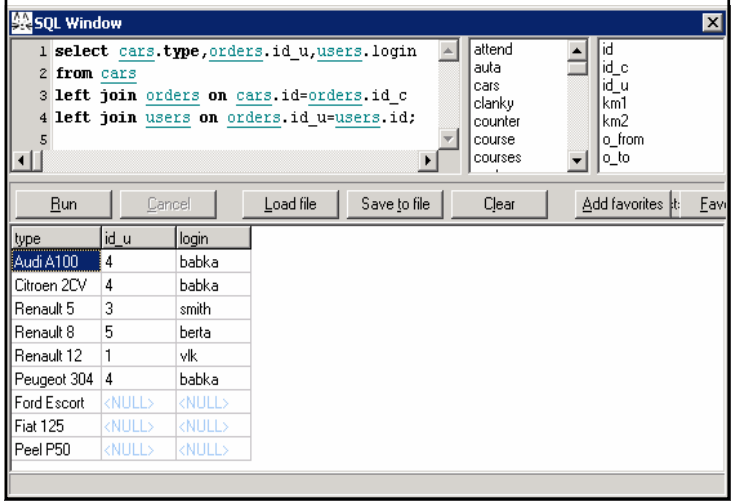
is the same as

note: FULL JOIN exists

```
SELECT Name,rooms.room,block,phone
 FROM employee RIGHT OUTER JOIN rooms
ON employee.room = rooms.room ;
```



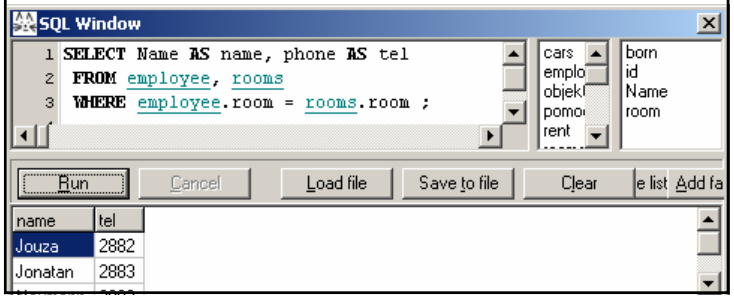
List of cars, including not in the connection table:



alias in select

Used for renaming table columns (the result):

SELECT Name AS name, phone AS tel
FROM employee, rooms
WHERE employee.room = rooms.room ;



select ... group by

Useful for sum, count or avg (average) function:

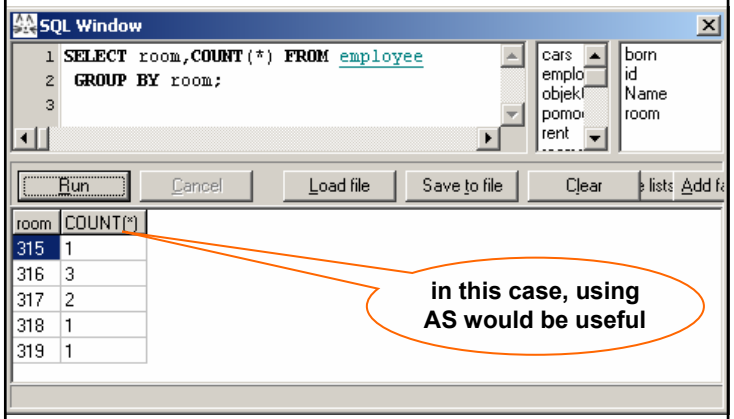
SELECT COUNT(*) FROM employee
GROUP BY room;

... will return only counts w/o room n., so ...

SELECT room, COUNT(*) FROM employee
GROUP BY room;

... will return a reasonable table:

In the rooms are so many people:



inserting data

auto increment???

Two form of the **INSERT** command:

INSERT INTO table1
VALUES (value1, value2, value3...)

... or ...

INSERT INTO table1 (column1, column2,...)
VALUES (value1, value2,...)

... the second one is safe against changing the table structure.

insert into

apostrophes

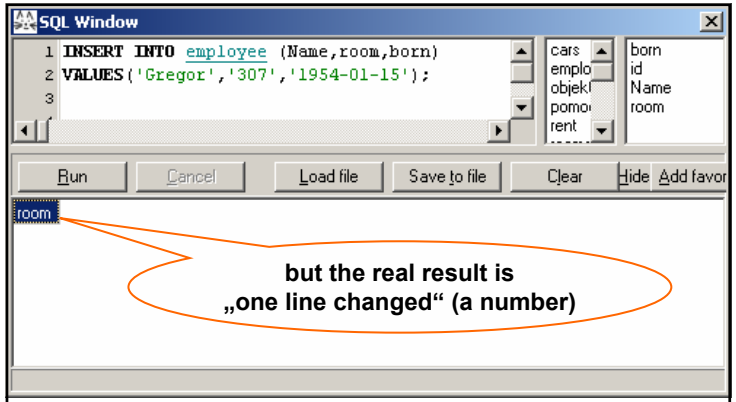
no apostrophes

INSERT INTO employee
VALUES('9','Gregor','307','1954-01-15');

... or we can use ...

INSERT INTO employee (Name,room,born)
VALUES('Gregor','307','1954-01-15');

... we can skip the ID column, because it was defined as auto increment; if we skip any other column with value, the „default“ value will be inserted; if is not defined, then the Null value.



The Micka itself use a different form:
insert into employee set id='9',Name='Gregor', room='307', born='1954-01-15';

delete

it will delete only one line
but random one (in SQL is no order)

-Deletes one or more lines

-defined by the „where“ clause:

DELETE FROM employee
WHERE room = ‘316’;

... it is useful to specify the **primary key** to prevent to delete more lines, but we can use:

DELETE FROM employee
WHERE room = ‘316’ LIMIT 1;

update

-Update rewrites one or more lines, again defined by the „where“ clause:

```
UPDATE employee
 SET room = '316'
    WHERE Name = 'Jouza';
```

... if WHERE clause is true for more lines, then the specified value (see SET) will be changed for every of this lines (be careful!).

LIMIT can be used, but there is no order in SQL.

DDL: alter table

(We will never need this, we will use Micka):

```
ALTER TABLE employee
 ADD login VarChar(33) character set cp1250
 collate cp1250_bin null default null
 AFTER Name;
```

(all rows will be filled by the default value = null)

```
ALTER TABLE employee
 DROP COLUMN login;
```

(again, only as example – not in application)

alter table

To rename table itself:

```
ALTER TABLE employee
 RENAME TO people;
```

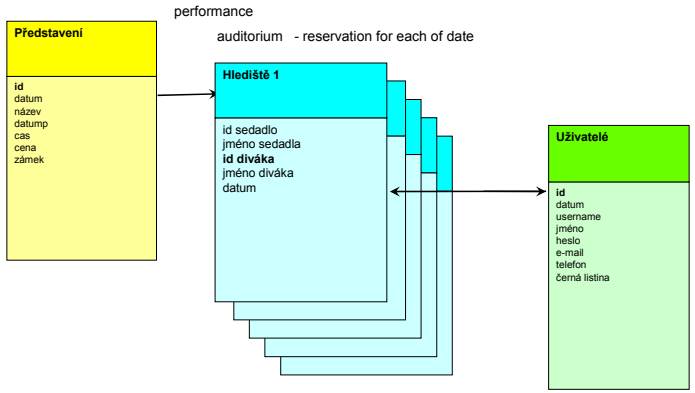
Change property of column, i.e. from 11 to 33:

```
ALTER TABLE employee
 MODIFY room VarChar(33)
   character set cp1250 collate cp1250_bin ;
```

(any use of this command is risky,
could be fatal for our application)

DDL: create table

There could be demand for create new table:



DDL: create table

CREATE TABLE action57
(seat VarChar(8) NOT null PRIMARY KEY,
  user Int);

(on the prev. slide, new “action” table is created and immediately filled by default seat numbers – rows ware marked by letters)

The application can delete table using:

DROP TABLE action17;

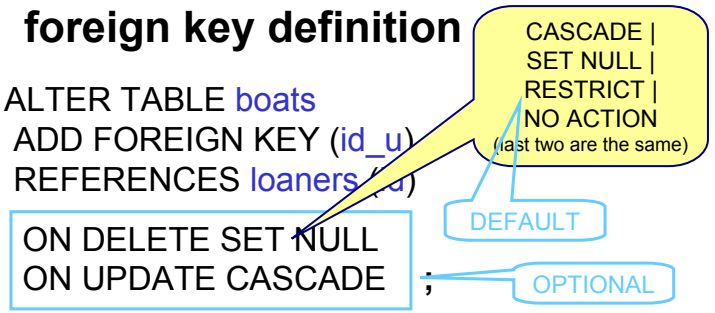
More complete on: http://www.1keydata.com/sql/sqldrop.html

create table

CREATE TABLE *<tablename>*
(*<col. name><col. type><col. properties>,*
  <col. name><col. type><col. properties>,
  ... ...
  <col. name><col. type><col. properties>);

<col. properties> (separated by space):
NULL | NOT NULL
DEFAULT *<value>*
AUTO INCREMENT
PRIMARY KEY | *<foreign key definition>*

foreign key definition



ALTER TABLE boats
ADD FOREIGN KEY (id_u)
REFERENCES loaners (id)
ON DELETE SET NULL
ON UPDATE CASCADE ;

(on the prev. slide, the part from FOREIGN can be added directly to line)

The application can delete foreign key using:

ALTER TABLE boats DROP FOREIGN KEY...

foreign key definition

More complete on: www.mysqltutorial.org/mysql-foreign-key/

Example from: www.w3schools.com/sql/sql_foreignkey.asp/ :

```
CREATE TABLE orders
(
  o_id int NOT NULL PRIMARY KEY,
  order_no int NOT NULL,
  p_id int FOREIGN KEY REFERENCES persons(id)
)
```

create pre-filled table

SELECT * FROM employee
 WHERE room >= '316' AND room <= '316'
  INTO employee316;

(will create table employee316;
  if there are no WHERE clause, complete table will be copied)

Don't forget to delete table using:

DROP TABLE employee316;

More examples on: www.w3schools.com/sql/sql_select_into.asp

CREATE VIEW

VIEW can speed up the SELECT commands:

CREATE VIEW test_v
 AS SELECT Name,room FROM employee ;

DROP VIEW test_v ;

VIEW can be used same way as a table:

SELECT * FROM test_v ;

http://www.mysqltutorial.org/create-sql-updatable-views.aspx

http://dev.mysql.com/doc/refman/5.0/en/create-index.html

For developers...

If you like to create client like Micka, you need some other commands. USE will select the database, SHOW can show the list of availaible databases, tables etc.:

USE database_name;
SHOW TABLES;

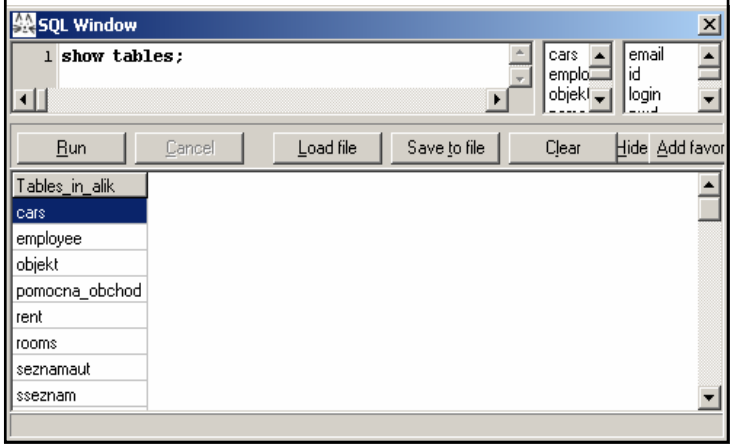
If you need to know the table structure, use:

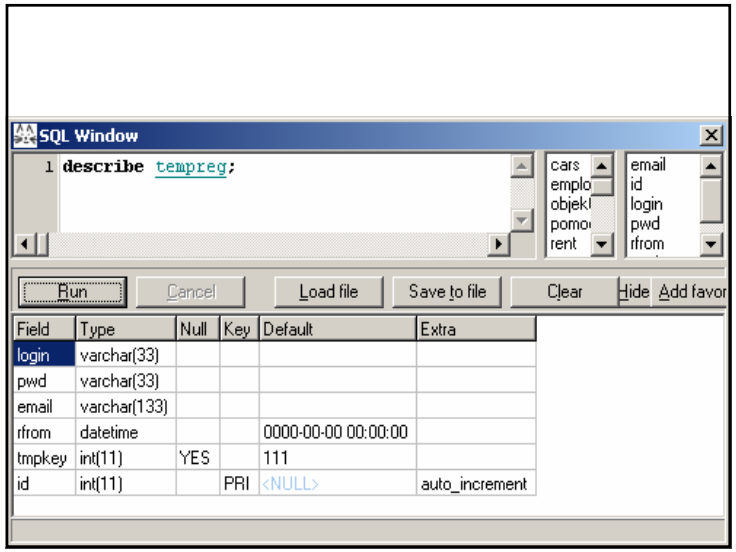
DESCRIBE table_name;

More info on the address:

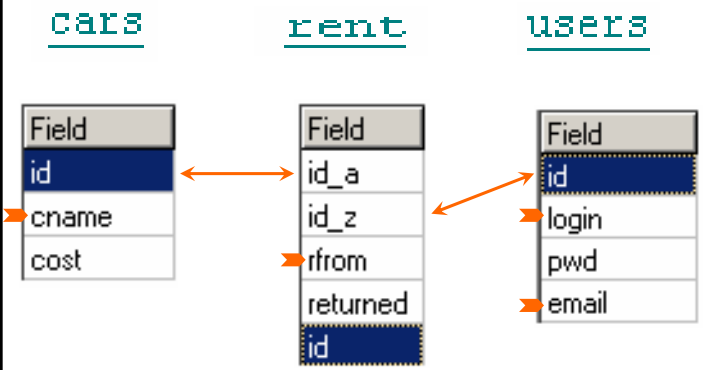
http://dev.mysql.com/doc/refman/5.0/en/show-tables.html

http://dev.mysql.com/doc/refman/5.0/en/describe.html



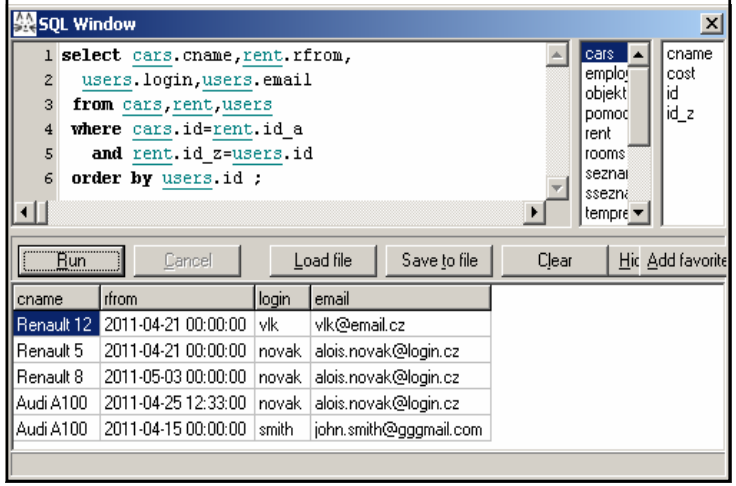


Connection table example



these elements should be displayed

Connection table – SELECT and its result:



Connection table – SELECT - note:

If you like to list only lines, where login is “novak”, you can add condition like

... and users.login = “novak” ...

complete SQL command:

select cars.cname,rent.from,users.login,users.email
from cars,rent,users
where cars.id=rent.id_a **and** rent.id_z=users.id
and users.login=“novak”**;**

subselect

http://www.tutorialspoint.com/sql/sql-sub-queries.htm

•SELECT can return table or value
•so can be used instead of table or value

Long example:

```
select cust_l from
 (select customers.login as cust_l,
  sum(orders.pcs * stock.ppp) as howmany
   from customers,orders,stock
    where customers.id=orders.customer
     and stock.id=orders.article
    group by orders.customer order by howmany desc)
as cltable;
```

task was: not list this column

alias for subselect is required by MySQL

copy from table to table

• First table with columns id, name, email
• Second table can contain more columns :-)

```
insert into students
(select
id+(select max(id) from students),
name, email
from anotherstudents
where someanothercondition) ;
```

max id as subselect

end of subselect

duplicity and alias for table

• The next simple example from somewhere on the internet will erase eventual duplicity names
(for example, if you have two students with name Smith, this looks like a good idea to kick one off)

```
delete x
 from students1 x
  join students1 z on x.name = z.name
where x.id > z.id  ;
```

(this keeps the MIN(ID); if you want MAX(ID), then just reverse the direction to x.id < z.id on the end of subselect)

http://stackoverflow.com/questions/6205640/sql-select-all-records-not-selected-by-another-query

select ... where ... in ...

The "in" condition can be used to test, if the value of property (SQL tests table row-by-row) is presented in the list of values (set, but typically column=property of another table).
The another table can be got as another select, i.e. subselect.

In the next example, select will list rows from the first table only in the case, that the property name is not in the second table. Possibly accidentally - not any of the properties is a primary key, or at least unique property of a real object (as e-mail).

```
select * from students1
where name not in
  (select name from students);
```

DCL: transaction

START TRANSACTION;

there can be any set of commands,
if we require that all of them are either successful,
or no one from them will be done.

then either:

COMMIT;

if all of the set of commands were successful, it confirms them

or:

ROLLBACK;

if any of the commands fails, it will return previous state

More complete on: http://dev.mysql.com/doc/refman/5.1/en/commit.html

Savepoint

SAVEPOINT *name-of-savepoint*;

... then we can play with the database; if we don't like new database appearance, we can go back to any savepoint, we have prepared, using command:

ROLLBACK TO *name-of-savepoint*;

... all the savepoints can be deleted,
for example by using command:

COMMIT;

or:

ROLLBACK;

More complete on: http://dev.mysql.com/doc/refman/5.1/en/savepoint.html

Literature ...

Many sources have been mentioned in text.

For the list of the commands, see

http://www.w3schools.com/sql/sql_quickref.asp

(incomplete, but nice table)

next: MS Access

then: http://www1.fs.cvut.cz/cz/u12110/prt/dks/dks5.pdf