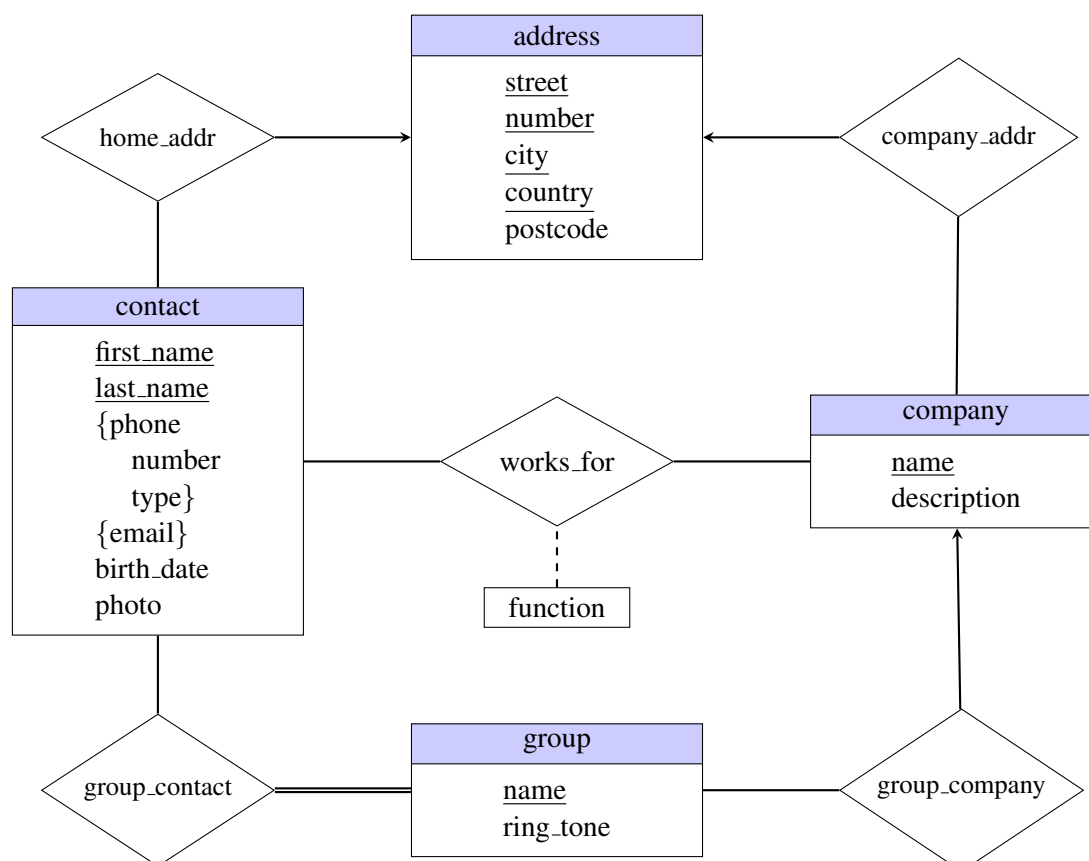


## More SQL exercises

A few practical points:

- Do not forget to write your name and student ID on your report.
- Unless explicitly stated, please do not include the output of the SQL statements.
- If anything is not clear: please ask. (TIP: start early, not just before the deadline)

This week we will work on common queries and updates to a contacts database which is the result of the following E-R design:



## Exercises

1. For which relationship sets in this E-R diagram creating a corresponding table is optional? Briefly (in no more than 3 sentences) explain your answer.
2. The commands to create the tables for the above E-R design are listed at <http://www.let.rug.nl/coltekin/courses/db2013/hw4-create-table.sql.txt> (or the file [hw4-create-table.sql](#) on Nestor). You can use one of these files with MySQL command `source` or pasting the commands into phpmyadmin SQL box.

**Some of the table names may conflict with the tables you already have in your database.** If necessary, rename your previous tables before running the new `create table` statements.

List the names of the tables created.

3. Some of the tables created in `hw4-create-table.sql` have missing columns and/or constraints required for the E-R design above. In total two columns and five constraints are missing. Find these by investigating the tables in the database or the create table statements in the file against the E-R design above, and add the missing columns and the constraints.

List the commands you used for adding these constraints.

4. Add the following contacts into your database. Note that this is slightly different than previous homework.

```
name = Donald Duck
email(s) = donald@example.com, donald@disney.com
phone(s) = 111-2222 (home), 555-5555 (work)
address = 1313 Webfoot Walk, Duckburg, Calisota, USA
company = Disney
function = actor
birth date = June 9, 1934
work address = 2100 Riverside Drive, Burbank, CA 91506, USA
group = friends, Disney
```

```
name = Sherlock Holmes
email(s) = sherlock@example.co.uk
phone(s) = 333-3333 (home)
address = 221B Baker Street, London, UK
company = none (freelancer, works from home)
birth date = January 1, 1887
group = friends
```

```
name = Clark Kent
email(s) = clark@example.com
phone(s) = 123-4567 (mobile), 555-5555 (work)
address = 1938 Sulivan Ln, Metropolis, USA
company = Daily Planet
function = journalist
birth date = June 1, 1938
group = friends
```

5. Insert a new contact without any addresses (you do not know any). Feel free to choose the values for other fields. Write down all SQL statements you need for inserting the contact information.

Do you observe a difference between not knowing the home address and the company (work) address?

6. Pick a foreign key constraint specified in `hw4-create-table.sql` and write one **insert** and one **delete** command that violates the constraint. Include the commands you used and the response from MySQL in your report.

7. Write down SQL statements for the following tasks.

- (a) List all information from the `contacts` table sorted by age such that the oldest contact is shown at the top of the list.
- (b) Show the unique list of phone numbers in the database.
- (c) List the contacts for which you do not have a photo in the database.

- (d) List all addresses in the database, combining all relevant fields in a single column labeled 'address'.
  - (e) Insert a photo for at least one of your contacts in the database. Include the SQL statement needed to insert the photo in your report. (Note: you can use `load_file` MySQL command to load a local file).
  - (f) Delete all photo(s) from the `contact` table.
8. Write down SQL queries to display the following information.
- (a) The number of contacts in the database.
  - (b) The number of mobile phone numbers.
  - (c) Average age of your contacts.
  - (d) For each phone number, the number of contacts sharing the phone.
  - (e) The phone numbers that are shared by multiple contacts.
9. Write down SQL queries listing the following information. Use appropriate `join` syntax.
- (a) A list of contacts with their occupation (`function` from the table `works_for`).
  - (b) A list of contacts with their phone numbers (note: you may have multiple rows for each contact).
  - (c) A list of contacts with phone numbers and email addresses.
  - (d) The number of email addresses for each contact.
  - (e) For each company in your database, average age of people (contacts) working for the company.
10. Write the same queries in exercise 9 without using any join keyword.

End of exercises. Good Luck!

**TIP of the DAY** (not needed for the solutions of exercise in this assignment):

The behavior differences regarding foreign key constraints are due to so-called *storage engines* that one can choose during the table creation process with statements like `engine = InnoDB`. The MySQL engine type *MyISAM* (which is the default in many configurations) silently accepts but ignores the constraints. You can see the storage engine in use for all tables in database `db_name` with this query:

```
select table_name, engine
from information_schema.tables
where table_schema='<db_name>';
```

In the setup on siegfried, `<db_name>` is your student ID. Note that `table_name` in this command should be typed literally, it is not the name of the table you are interested. You can modify the where statement to include only the table you are interested.

You can alter the storage engine of an existing table with the command,

```
alter table <table_name> engine=InnoDB;
```

sets the storage engine to 'InnoDB'. Note that you need to replace `<table_name>` with the name of the table in this command.

There are also ways to set this as a default option if you have access to the DBMS with administrator (root) privileges.