

Assignment 2

Obligatory assignment. Deadline: Thursday 25.01.2018

The report should include all necessary commands to complete the tasks, printout from the system, explanation of what is done, the result and explanation of the result.

Remember to include the names of the group members on the front page of the report. The report can be in English or Norwegian. The report should be handed in via it's learning.

The assignment should be accomplished in groups of three or four students.

Lecturers:

Bjarte Kileng, Bjarte.Kileng@hib.no, room E418

Maksim Melnik Storetvedt, Maxim.Storetvedt@hvl.no, room E506

Faustin Ahishakiye, Faustin.Ahishakiye@hvl.no, room E506

Violet Ka I Pun, violet@foldr.org

Task 1: DBMS statements and utilities

Create some tables as a normal database user (i.e. not root), insert some test data into the tables, and test the following commands and/or utilities one by one. Document the input and output, and explain what you did in your words.

Use the [MariaDB Knowledge Base](#) to solve this task.

1. Convert a table to InnoDB, and to MyISAM
2. SHOW INDEX
3. ANALYZE TABLE
4. CHECK TABLE
5. REPAIR TABLE
6. OPTIMIZE TABLE
7. CHECKSUM TABLE
8. innodbchecksum
9. myisamchk

Note that some commands or tools have a requirement on the storage engine, so make sure that you have a table with the correct engine.

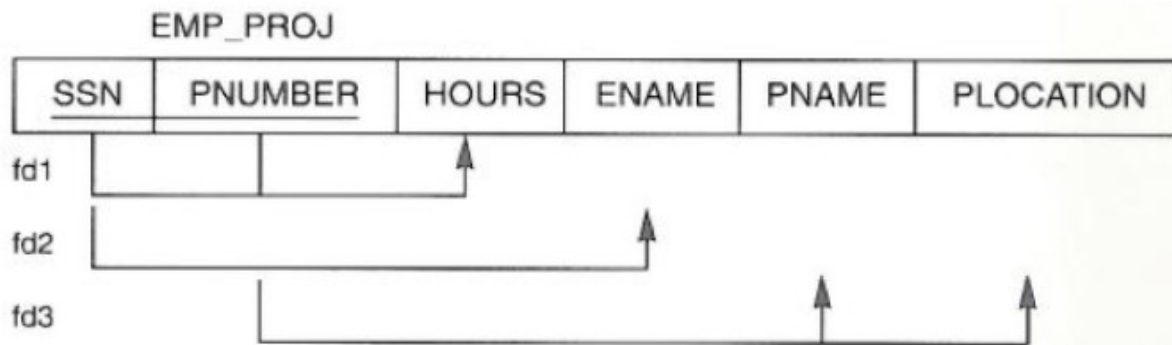
Task 2: Normalisation

This task is theory only. You do not need to implement anything on the lab computers.

Below a relation (table) is given, including example data.

You can assume that ENAME is atomic.

What normal form does it currently conform to? Normalize it to 3NF if it does not currently conform to 3NF.



EMP_PROJ

| SSN | PNUMBER | HOURS | ENAME | PNAME | PLOCATION |
|-----------|---------|-------|----------------------|-----------------|-----------|
| 123456789 | 1 | 32.5 | Smith, John B. | ProductX | Bellaire |
| 123456789 | 2 | 7.5 | Smith, John B. | ProductY | Sugarland |
| 666884444 | 3 | 40.0 | Narayan, Ramesh K. | ProductZ | Houston |
| 453453453 | 1 | 20.0 | English, Joyce A. | ProductX | Bellaire |
| 453453453 | 2 | 20.0 | English, Joyce A. | ProductY | Sugarland |
| 333445555 | 2 | 10.0 | Wong, Franklin T. | ProductY | Sugarland |
| 333445555 | 3 | 10.0 | Wong, Franklin T. | ProductZ | Houston |
| 333445555 | 10 | 10.0 | Wong, Franklin T. | Computerization | Stafford |
| 333445555 | 20 | 10.0 | Wong, Franklin T. | Reorganization | Houston |
| 999887777 | 30 | 30.0 | Zelaya, Alicia J. | Newbenefits | Stafford |
| 999887777 | 10 | 10.0 | Zelaya, Alicia J. | Computerization | Stafford |
| 987987987 | 10 | 35.0 | Jabbar, Ahmad V. | Computerization | Stafford |
| 987987987 | 30 | 5.0 | Jabbar, Ahmad V. | Newbenefits | Stafford |
| 987654321 | 30 | 20.0 | Wallace, Jennifer S. | Newbenefits | Stafford |
| 987654321 | 20 | 15.0 | Wallace, Jennifer S. | Reorganization | Houston |
| 888665555 | 20 | null | Borg, James E. | Reorganization | Houston |

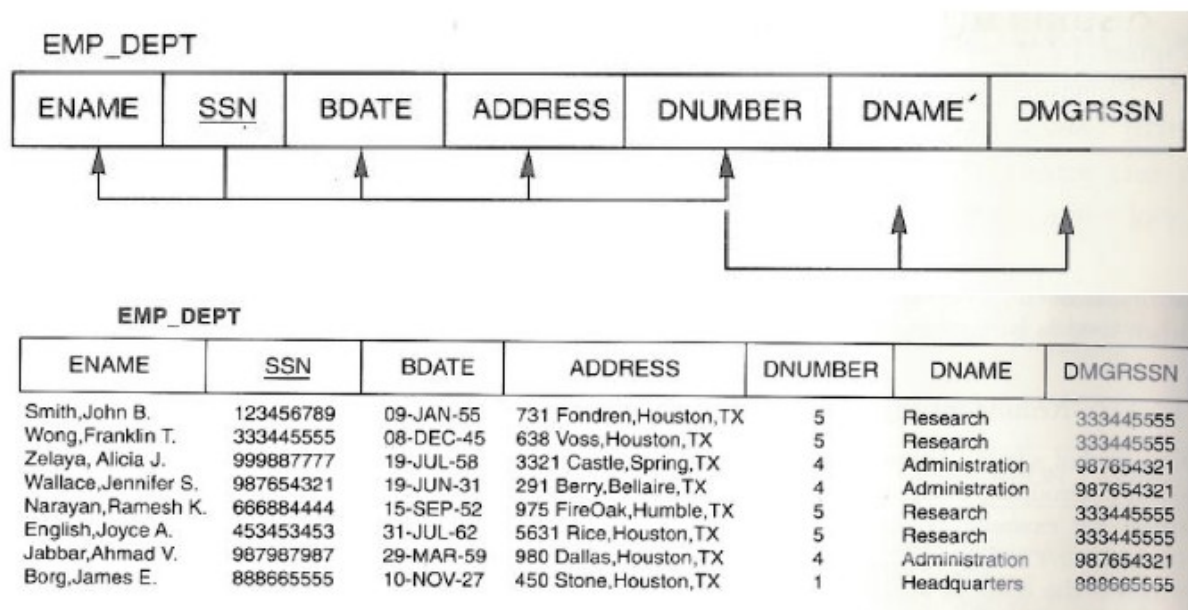
Task 3: More normalisation

This task is theory only. You do not need to implement anything on the lab computers.

Below a relation (table) is given, including example data.

You can assume that ENAME and ADDRESS are atomic.

Which normal form does it currently conform to? Normalize it to 3NF if it does not currently conform to 3NF.



Task 4: Normalisation & Denormalisation

- a) Western Norway University of Applied Sciences (*Høgskolen På Vestlandet*) needs a new system to store grades of students. You will design the database schema. These are the requirements:
- The school needs the name, student number, birth number, current address, telephone number, home address, birth date, gender, year (1st, 2nd or 3rd), faculty (FIN, FLKI, FHS, FØS), study program (Computing, Chemistry, Teaching, etc...), and study level (bachelor, master, PHD). Both student number and birth number are unique for each student.
 - Each faculty is described by a name ("Faculty of Engineering and Science"), a code (FIN), a phone number, and an address. Both name and code are unique for each faculty.
 - Each course has a name, a code (DAT151), a number of lecture hours each week, and a responsible department ("Department of Computing, Mathematics and Physics"). Each department belongs to a faculty. The code is unique for each course. (The same course is usually thought multiple years, but if the content of the course is changed significantly, it gets a new code.)
 - Each course is thought once a year, and has a teacher that is responsible the course that year.
 - It is desired to have a table where one row represents the student's grade in one course a specific year.
 - You can assume that the student name and student addresses are atomic values.
 - Use the following entity names:
 - STUDENT is the entity with student data, e.g. student name and current address.
 - FACULTY is the entity with e.g. faculty name and code.
 - COURSE_SCHEDULE is the entity that couples a course with a responsible teacher when the course is thought a specific year.

- The following attribute names are used later in this task:
 - SNAME is the name of a student
 - FNAME is the name of a faculty, e.g. “Faculty of Engineering and Science”
 - FCODE is the faculty code, e.g. “FIN”
 - CYEAR is a year a that a course is thought

Design a relational database schema for this application, in 3NF.

- b) Implement this database schema into your MariaDB database. Create the tables as a normal database user (i.e. not root), using the command line interface.

There are several GUIs that work with MariaDB, but most are either commercial, or Windows only (see [list at the MariaDB knowledge base](#)). Some open source GUIs that run on Centos7 are [phpMyAdmin](#) that is available in the *epel* repository, and [MySQL Workbench](#) available from the MySQL web site.

Optional: Also create the tables using a GUI, e.g. *mysql-workbench* or *phpMyAdmin*.

Can you implement the needed foreign keys as well? In MySQL and MariaDB, the foreign key constraint is ignored by the MyISAM (or Aria) engine. Create the tables using the InnoDB engine.

- c) For each of the following SQL-queries, discuss whether a denormalisation of the schema from a) might improve performance:

1. `SELECT s.SNAME, f.FNAME FROM STUDENT s, FACULTY f
WHERE s.FCODE = f.FCODE;`
2. `SELECT COUNT(*) FROM STUDENT WHERE FCODE = 'FIN';`
3. `SELECT DISTINCT CYEAR FROM COURSE_SCHEDULE;`

- d) Can you implement the needed foreign keys as well? In MySQL and MariaDB, the foreign key constraint is ignored by the MyISAM (or Aria) engine. Create the tables using the InnoDB engine.