School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2016**
Lecturer: **Janusz R. Getta**

CSIT115/CSIT815 Database Management and Security
Laboratory 4

Scope

This laboratory includes the tasks related to quality evaluation of relational database design and `CREATE TABLE` statement of SQL

This laboratory consists of 2 tasks and specification of each task starts from a new page.

It is strongly recommended to solve the problems included in this specification **before coming to a laboratory class** and bring the preliminary solutions to a laboratory class such that any doubts, question, problems, etc can be discussed with a tutor in a laboratory class. Such procedure allows for more effective use of time spent in a supervised laboratory class.

Tasks

Task1 (1 mark)

Analyze the following collection of relational schemas.

```
EMPLOYEE(enumber, first-name, last-name, project-title,
                                                budget, deadline)
```

A relational table `EMPLOYEE` contains information about the employees and projects the employees are working on. A project is implemented by more than one employee and each employee works on one or more projects. Employee number (`enumber`) uniquely identifies each employee and project title (`project-title`) uniquely identifies each project. The attributes `budget` and `deadline` describe the projects.

```
MANAGER(enumber, first-name, last-name, mnumber)
```

A relational table `MANAGER` contains information about the managers of employees. An attribute `enumber` uniquely identifies each employee. An attribute `mnumber` is an employee number of a direct manager of an employee. Each employee has only one manager and a manager manages one or more employees.

```
BUILDING(bnumber, bname, rnumber, area, enumber)
```

A relational table `BUILDING` contains information about the buildings such as building number (`bnumber`) and building name (`bname`). A building hosts a number of rooms. Each room in a building has a unique number. Additionally, each room is described by an area (`area`). An employee is identified by employee number (`enumber`) and he/she is located in a room. More than one employee can be located in one room.

```
EQUIPMENT(serialnum, bnumber, rnumber)
```

A relational table `EQUIPMENT` contains information about equipment items located in the rooms. An equipment item is identified by serial number (`serialnum`) and it is located in a given room in a given building. An equipment item can be located in one room only. A room contains a number of equipment items.

```
EQPTDESC(serialnum, name, colour, weight)
```

A relational table `EQPTDESC` contains the descriptions of each equipment item. A description of equipment item consists of unique serial number (`serialnum`), colour (`colour`) and weight (`weight`). Each item has one description.

No new attributes can be added to the design !

Your task is to verify the quality of relational design given above and to improve quality through decomposition/synthesis of relational schemas in all cases whenever it is necessary. The quality objectives include elimination of redundancies on one side and minimalization of the total number of relational schemas. It means that some of the relational schemas given above must be decomposed into several schemas and some of the schemas must synthesised into a smaller number of schemas.

To verify the quality of relational schemas listed above use a method of row insertions explained in a presentation 07 Database Design Quality.

List all decompositions and synthesises performed. Each time you propose a decomposition/synthesis of relational schemas list the new relational schemas obtained after decompositions and synthesis. There is no need to list all relational schemas at the end of quality improvement process.

The relational schemas must be listed in a format presented in the slides 43-44 in a presentation 06 Logical Design.

Deliverables

A file `solution1.pdf` with a list of decomposed and/or synthesised relational schemas, primary key for each relational schema, candidate keys (if any) for each relational schema, foreign keys (if any) for each relational schema.

Task 2 (1 mark)
Consider the following collection of relational schemas together with the domain constraints.

`EMPLOYEE(enum, first-name, last-name, salary)`
primary key = (`enum`)
domain constraints:
- `enum` is a positive number 7 digits long,
- `first-name`, `last-name` are variable size strings no longer than 30 characters,
- `salary` is a positive number with 6 digits before decimal do and 2 digits after decimal dot,
- the values in all columns are compulsory.

`DRIVER(enum, license, category, experience)`
primary key = (`license`)
candidate key = (`enum`)
foreign key = (`enum`) references `EMPLOYEE(enum)`
domain constraints:
- `license` is a fixed size string of 7 characters,
- `category` is a single character, only the following values are possible: A, B, C,
- `experience` is a variable size strong of maxim 2000 characters,
- the values in all columns are compulsory.

No automatic deletion of the rows is allowed when a referenced row with a respected value of primary key is deleted from the other table.

`ADMIN(enum, position)`
primary key = (`enum`)
foreign key = (`enum`) references `EMPLOYEE(enum)`
domain constraints:
- `position` is a variable size string no longer than 30 characters.
- the values in all columns are compulsory.

Automatic deletion of the rows is required when a referenced row with a respected value of primary key is deleted from the other table.

`TRUCK(rego, capacity, manufacturer, model)`
primary key = (`rego`)
domain constraints:
- `rego` is a fixed size string o 8 characters
- `capacity` is a floating point number
- `manufacturer` and `model` are variables size strings no longer than 50 characters
- the values in the columns `manufacturer` and `model` are optional.

`TRIP(origin, destination, license, rego, tdate)`
primary key = (`license, rego, tdate`)
foreign key1 = (`license`) references `DRIVER(license)`

foreign key2 = (rego) references TRUCK(rego)
domain constraints:
origin and destination are variable size strings no longer than 30 characters,
tdate is of type DATE.
No automatic deletion of the rows is allowed when a referenced row with a respected value of primary key is deleted from the other table.

Your task is to implement SQL script solution2.sql with CREATE TABLE statements that can be used to implement the relational schemas in MySQL DBMS.

You can find a lot of information about implementation of CREATE TABLE statements in a presentation 09 SQL - Data Definition Language (DDL) and in Cookbook, How to use data definition and basic data manipulation statements of SQL, Recipe 4.1 How to create and how to alter the relational tables ?

When a script file solution2.sql is ready connect to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and execute your script file.

If processing of the file returns any errors then you must eliminate the errors ! Processing of your script must return NO ERRORS ! A solution with errors is worth no marks !

It is recommended to create a script drop2.sql that drops all relational tables created during execution of a script solution2.sql and it is recommend to execute drop2.sql after each execution of solution2.sql (you can also put DROP TABLE statements at the end of a script solution2.sql). In such a way you can avoid an unpleasant syntax error messages like:

```
ERROR 1050 (42S01): Table '…' already exists
```

when you execute a script solution2.sql the next time. Please, remember that such message also count as an error in processing of the script with all consequences coming from such fact.

When execution of your script returns no errors then connect to MySQL server using command based interface mysql and create a report from processing of the script solution2.sql. Save your report in a file solution2.rpt. To create a report you must use a command tee solution2.rpt before processing the script and a command notee after processing of a script. Then, you can find a file solution2.rpt in the current folder of mysql client. You can find more information about creating reports from processing of SQL scripts in Cookbook, Recipe 3.1 How to use "mysql? Command based interface to MySQL database server ? Step 4 How to save the results of SQL processing in a file ?.

Your report must contain a listing of SQL statements processed. To achieve that, you must logon mysql client with `-v` (verbose) option in the following way:

```
mysql -u csit115 -p -v
```

You can find more information on how to display SQL statements while a script is processed in Cookbook, Recipe 3.1 How to use "mysql? Command based interface to MySQL database server ? Step 3 How to process SQL script ?.

A report that contains no listing of executed SQL statements scores no marks ! So, make sure that you connect to mysql client with an option `-v` !

If you find more problems with the implementation then consult Cookbook, your tutor, and finally your lecturer.

A report created by copying and pasting screen contents from MySQL Workbench scores no marks ! You must use mysql client to create a report !

And again, … a report from processing of SQL script must contain NO ERRORS !

And finally, … do not even think about manually changing the contents of a report file `solution2.rpt`. Manually changing the outcomes of computer generated reports is a CRIME and it causes very serious consequences up to and including the failure of entire subject !

Deliverables

A file `solution2.rpt` with a report from processing of SQL script `solution2.sql`. The report MUST NOT include any errors and the report must list all SQL statements processed.

End of specification