

CSIT115/CSIT815 Data Management and Security
Laboratory 4
27 March 2017

Scope

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

The outcomes of the laboratory work are due by **Saturday, 8 April, 2017, 7.00 pm.**

This laboratory contributes to 2% of the total evaluation in the subject.

A submission procedure is explained at the end of specification.

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

Task 1 (1 mark)

Analyze the relational schemas listed below and find the schemas that are incorrectly designed. A schema is incorrectly designed when the insertions of rows into the schema creates the redundancies. Consider a method of row insertions explained in a presentation 07 Database Design Quality to find out which relational schemas are incorrectly designed.

Next, for each incorrectly designed relational schema propose its decomposition into a pair of correctly designed relational schemas and equivalent to the original one. A pair of relational schemas is equivalent to other relational schemas when the respective pair of relational tables contains exactly the same information as a relational table built on the original schema.

- (1) `PLAYER(pnum, pname, team, position, team-address)`

A relational schema `PLAYER` is a header of a relational table that contains information about the players and teams the players belong to. A player number (`pnum`) uniquely identifies each player in a team; and team name (`team`) uniquely identifies the team address (`team-address`). The attribute `position` describes the player's position in a team.

- (2) `WAREHOUSE(name, address, item, price, quantity)`

A relational schema `WAREHOUSE` is a header of a relational table that contains information about the warehouses and items kept there. An attribute `address` is an address of a warehouse and it uniquely identifies each warehouse. An attribute `name` is name of a warehouse and it also uniquely identifies each warehouse. An attribute `item` is a unique name of an item. A pair of attributes (`address, item`) uniquely identify the item's unit price (attribute `price`) and quantity (attribute `quantity`) of the item.

- (3) `FLIGHT(passenger, flightNum, seatType, departurePlace, departureTime)`

A relational schema `FLIGHT` is a header of a relational table that contains information about the flights booked by the passengers. A passenger is uniquely identified by a value of an attribute `passenger`. A passenger books seats (`seatType`) on many different flights (`flightNum`). Only one flight (`flightNum`) takes off from a given departure place (`departurePlace`) at a given departure time (`departureTime`). A flight number (`flightNum`) uniquely determines departure place (`departurePlace`) and departure time (`departureTime`).

- (4) `ENROLMENT(studentNum, subjectCode, enrolDate, IP)`

A relational schema `ENROLMENT` is a header of a relational table that contains information about the students enrolled in the subjects (`subjectCode`). A triple of attributes (`studentNum, subjectCode, enrolDate`) uniquely identifies each

row in the relational table. An attribute `IP` is an IP address of a computer from where an enrolment has been done.

(5) `CAR(regoNum, manufacturer, model, year, colour)`

A relational table `CAR` contains the descriptions of cars. A registration number (`regoNum`) uniquely identifies each car. A car is described by a manufacturer name (`manufacturer`), model name (`model`), year when manufactured (`year`), and colour (`colour`).

No new attributes can be added to the relational schemas listed above !

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

Deliverables

For each one of the cases from (1) to (5) write either a statement:

A relational schema `<schema-name>` is designed correctly

or a statement:

A relational schema `<schema-name>` is designed incorrectly

`<schema-name>` must be replaced with a name of a respective relational schema.

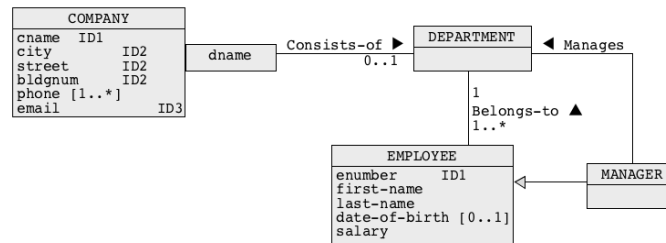
For each incorrectly designed schema, show the results from an insertion test that reveal the redundancies in a relational table that has incorrectly designed relational schema. Insert precisely 3 rows into a relational table.

Additionally, for each incorrectly designed relational schema list a correctly designed relational schemas obtained from a decomposition of the incorrect schema. The relational schemas must be listed in a format presented in the slides 43-44 in a presentation 06 Logical Design.

Save your answers in a file `solution1.pdf`. Submission of a file with a different name and/or different extension and/or different type scores no marks.

Task 2 (1 mark)

Consider a conceptual schema given below.



Implement SQL script `solution2.sql` with `CREATE TABLE` statements that implement the relational schemas obtained from transformation of the conceptual schema given above into the relational schemas. Use a superset method for implementation of generalization.

Note, that you **MUST** use only `CREATE TABLE` statements and no other statement of SQL !

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 `dbdrop2.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 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. Submission of a file with a different name and/or different extension and/or different type scores no marks.

Submission

Note, that you have only one submission. So, make it absolutely sure that you submit correct files with the correct contents. No other submission is possible !

Submit the files **solution1.pdf**, and **solution2.rpt** through Moodle in the following way:

- (1) Access Moodle at **<http://moodle.uowplatform.edu.au/>**
- (2) To login use a **Login** link located in the right upper corner the Web page or in the middle of the bottom of the Web page
- (3) When logged select a site **CSIT815/CSIT115 (S117) Data Management and Security**
- (4) Scroll down to a section **Submissions**
- (5) Click at a link **In this place you can submit the outcomes of Laboratory 4**
- (6) Click at a button **Add Submission**
- (7) Move a file **solution1.pdf** into an area **You can drag and drop files here to add them**. You can also use a link **Add...**
- (8) Repeat step (7) for a file **solution2.rpt**.
- (8) Click at a button **Save changes**
- (9) Click at a button **Submit assignment**
- (10) Click at a button **Continue**

A policy regarding late submissions is included in the subject outline.

Only one submission of the outcomes of Laboratory 4 is allowed and only one submission per student is accepted.

A submission marked by Moodle as "late" is always treated as a late submission no matter how many seconds it is late.

A submission that contains an incorrect file attached is treated as a correct submission with all consequences coming from the evaluation of the file attached.

It is expected that all tasks included within **Laboratory 4** will be solved **individually without any cooperation** with the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or office hours. Plagiarism will result in a **FAIL** grade being recorded for that assessment task.

The evaluated outcomes of will be electronically returned to the students before 11.55pm on Saturday, 22 April, 2017.

End of specification