

CSIT115/CSIT815 Data Management and Security
Final Assignment
Published on 19 June 2020

Scope

This Final Assignment is related to verification of conceptual modelling, logical design, SQL DDL, SQL DML, SQL SELECT statements, consistency constraint, implementation of a simple auditing system, and database backup and recovery techniques.

Please read very carefully information listed below.

This Final Assignment contributes to 60% of the total evaluation in a subject CSIT115 and it contributes to 60% of the total evaluation in a subject CSIT815.

The outcomes of the Final Assignment work are due by **Friday 26 JUNE 2020, 10.00 pm (sharp)**.

A submission procedure is explained at the end of specification.

This Final Assignment consists of 7 tasks in the subject CSIT115, and 8 tasks in the subject CSIT815, and specification of each task starts from a new page.

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

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, ... etc) is not allowed. The compressed files will not be evaluated.

All files left on Moodle in a state `"Draft(not submitted)"` will not be evaluated.

An implementation that does not compile due to one or more syntactical errors scores no marks and implementation that has the processing errors scores no marks.

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

Prologue

Connect to Moodle and download the files `dbcreate_f.sql`, `dbdrop_f.sql`, `dbload_f.sql`, and `dbupdate_f.sql` from **Final Assignment** on Moodle.

SQL script `dbcreate_f.sql` can be used to create the relational tables of a sample database. SQL script `dbdrop_f.sql` can be used to drop the tables of a sample database. SQL script `dbload_f.sql` can be used to load data into a sample database.

Connect to MySQL database server either through command line interface `mysql` or graphical user interface MySQL Workbench.

When connected, select a database `csit115` with a command `use csit115`.

To create the relational tables of a sample database, process SQL script `dbcreate_f.sql`.

To load data into the relational tables created in the previous step process SQL script `dbload_f.sql`.

To list the names of relational tables created, use a command `show tables`.

To list a structure of a relational table `<table-name>` use a command `describe <table-name>`.

No report is expected from the implementation of the actions listed above.

Task 1 (10 marks)

Read and analyse the following specification of a sample database domain.

A hypothetical bank is divided into several branches; each managed by exactly one branch manager. A branch is characterised by a branch number, branch name and address.

The bank employs a number of tellers, whose duties include cashing checks, accepting deposits, and handling withdrawals for the customers. Every teller has a unique identification number and he/she is described by a set of standard attributes such as: name, address, date of birth, and salary level. The tellers are rotated among the branches, but once a teller is sent to a branch he/she must stay at that branch for a whole day.

The bank opens accounts for customers. To open an account a customer provides his/her personal data, i.e. name, date of birth, address and at least three identification documents. The personal and identification information of each potential customer is recorded. Each customer obtains a unique identification number. Then he/she is allowed to open one or more accounts. Each account is described by its unique number, balance, and type. Whenever an account is opened, the current date and time is recorded. Customers are allowed to open many accounts at different branches, but each account is assigned to only one branch.

An objective of this task is to construct a conceptual schema for the specification of a database domain listed above.

It is not allowed to add any artificial identification attributes commonly known as "id" attributes to the specification listed above.

Use UMLet tool to create a drawing of a conceptual schema in a notation of UML simplified class diagrams explained to you during the lecture classes in CSIT115/CSIT815. No other notation will be accepted!

Use an option File->Export as... to export your diagram into a file solution1.bmp in BMP format. Do not delete an exported file. You will submit it as one of the deliverables from your laboratory work.

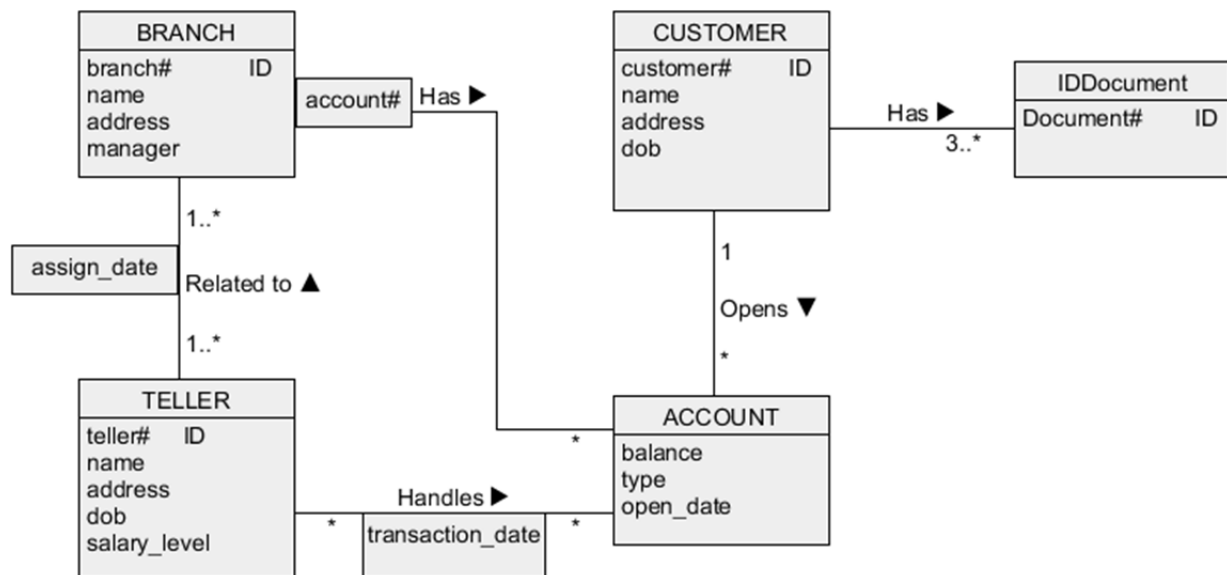
In this task there is NO NEED to provide a detailed analysis of a conceptual schema. The final conceptual schema expressed in a notation of UML simplified class is completely sufficient.

Deliverables

A file solution1.bmp with the final design of a conceptual schema. Submission of a file with a different name and/or different extension and/or different type scores no marks.

Task 2 (10 marks)

Consider a conceptual schema given below.



- (1) Perform a step of logical database design and transform a conceptual schema given above into a collection of relational schemas.

No report is expected from a step of logical design.

- (2) Next, use the relational schemas obtained in the previous step to create SQL script `solution2.sql` with CREATE TABLE statements that implement the relational schemas.

Your CREATE TABLE statements must enforce the following types of the columns in the relational tables.

- (i) All columns that contain date (e.g. dob, etc.) must be of type DATE.
- (ii) Balance of an account must be decimal values. The whole number part must be 8 digits, and the fractional part must be 2 digits. It is a positive value up to 99999999.99.
- (iii) Branch number is a fixed size decimal of 6 digits.
- (iv) Customer number is a fixed size decimal of 16 digits.
- (v) Document number is a fixed size string of 16 characters.
- (vi) The type of account can only be either SAVING, or CREDIT, or HOMELoAN, or PERSONALLoAN.
- (vii) The salary level of teller can only be either A, or B, or C, or D.
- (viii) The types of other columns in the relational tables are up to you. However, the types must make sense. For example, an address of type integer will not get a lot of appreciation from a person evaluating your solution.

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

No report is expected from a step (2).

- (3) Next, drop the relational tables created in the previous step at the end of SQL script `solution2.sql` with DROP TABLE statements.

Beware the order of relational tables that are dropped is important.

No report is expected from a step (3).

- (4) When your script is ready connect to the command line interface mysql and process the script `solution2.sql` implemented in the previous steps (2) and (3). Processing of the script must create a report. The report from processing of a script `solution2.sql` must be saved in a file `solution2.rpt`.

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

You can avoid an unpleasant error messages like:

ERROR 1050 (42S01): Table '...' already exists

Please, remember that such message also counts as an error in processing of the script and that a solution with errors is worth no marks!

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

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

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

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

Deliverables

A file `solution2.rpt` with a report from processing of SQL script `solution2.sql`. The report must be created with the command line interface mysql, 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.

QUESTION 3 (8 marks)

Download a file `solution3.sql` and insert into the file the implementations of the following modifications of the structures, consistency constraints, and/or the contents of the sample database specified in Prologue on Page 2.

Note, that you are not allowed to modify and/or to drop any consistency constraints. You may need more than one SQL statement to implement a single subtask listed below.

Your implementation must directly follow a comment with a specification of a subtask.

- (1) Modify structures of the sample database such that after modifications it is possible to record in the database information about the country of AIRLINE and PASSENGER that can store up to 50 characters.
(2 marks)
- (2) Modify consistency constraints of the sample database such that after modifications it is possible to store in the database information about arrival time is later than departure time for a flight. The capacity of a flight is a positive integer less than 1000.
(2 marks)
- (3) Modify consistency constraints of the sample database such that after modifications it is possible to store in the database information about class of a seating is either FIRST, or BUSINESS, or ECONOMY. The available number of seats is a positive integer less than 1000.
(2 marks)
- (4) Modify a structure and consistency constraint of the sample database such it is possible to store in the database information about the number of flights an airline company has. The total number of flight must be a positive integer less than 100000.
(2 marks)

It is recommended to use a script `dbdrop_f.sql` to drop all relational tables modified during the processing of a script `solution3.sql` and then to re-create the original database with a script `dbcreate_f.sql`, then load data into the database with a script `dbload_f.sql`. In such a way your script always operates on the original structures of the sample database.

To create a report from processing of a script file `solution3.sql` open a Terminal window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution3.sql` and save a report in a file `solution3.rpt`. Note, that when started with the options `-v` and `-c` the command line interface includes both listing of SELECT statements processed and the comments included in the original version of a file `solution3.sql`.

A report that contains no listing of processed SQL statements scores no marks and report that contains errors of any kind also scores no marks!

Deliverables

A file `solution3.rpt` with a report from processing of SQL script `solution3.sql`. The report must be created with the command line interface `mysql`, the report **MUST NOT** include any errors, and the report **MUST LIST ALL** SQL statements processed and **ALL** comments included in the original (downloaded) version of `solution1.sql`. **Marks will be deducted for the missing comments.** Submission of a file with a different name and/or different extension and/or different type scores no marks.

QUESTION 4 (8 marks)

Download a file `solution4.sql` and insert into the file the implementations of the following modifications of the structure, and the contents of the sample database specified in Prologue on Page 2.

Note, that you are not allowed to modify and/or to drop any consistency constraints. Also note, that to implement some of the modifications listed below you may need more than one data manipulation statement of SQL.

Your implementation must directly follow a comment with a specification of a subtask.

- (1) Assume, that a new passenger Peter, who was born on 20 April 1976, has booked an ECONOMY ticket of a flight JQ501 from SYD to MEL on 05 May 2019 at 06:05. Insert appropriate information into the sample database. The other values are up to you. Reduce the available number of seat for the economy class of the flight by one.
(2 marks)
- (2) Modify a structure of the sample database such it is possible to store in the database information about the number of seat booked by a passenger. Update the total number of seat for each passenger by using an advanced DML statement. Display the name, date of birth, total number of seat booked by each passenger.
(2 marks)
- (3) Use a single advanced DML to create a new table and to load into the table information of bookings which departure from SYD. There is no need to enforce any consistency constraints on the new table. Display the information from the new table.
(2 marks)
- (4) The flight QF497 from SYD to PEK on 26 May 2019 at 22:05 has been cancelled. Delete from the database information about the flight. Remember, that the foreign keys in all CREATE TABLE statements have no ON DELETE CASCADE clause.
(2 marks)

It is recommended to use a script `dbdrop_f.sql` to drop all relational tables modified during the processing of a script `solution4.sql` and then to re-create the original database with a script `dbcreate_f.sql`, then load data with a script `dbload_f.sql`. In such a way your script always operates on the original structures of the sample database.

To create a report from processing of a script file `solution4.sql` open a Terminal window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution4.sql` and save a report in a file `solution4.rpt`. Note, that when started with the options `-v` and `-c` the command line interface includes both listing of SELECT statements processed and the comments included in the original version of a file `solution4.sql`.

A report that contains no listing of processed SQL statements scores no marks and report that contains errors of any kind also scores no marks!

Deliverables

A file `solution4.rpt` with a report from processing of SQL script `solution4.sql`. The report must be created with the command line interface `mysql`, the report **MUST NOT** include any errors, and the report **MUST LIST ALL** SQL statements processed and **ALL** comments included in the original (downloaded) version of `solution1.sql`. **Marks will be deducted for the missing comments.** Submission of a file with a different name and/or different extension and/or different type scores no marks.

QUESTION 5 (10 marks)

Download a file `solution5.sql` and insert into the file the implementations of the following queries as SQL SELECT statements of the sample database specified in Prologue on Page 2.

Your implementation must directly follow a comment with a specification of a subtask.

- (1) Find airline name, total number of flights for each airline. Sort the results in the descending order of total number of flights, then in the ascending order of airline name. (2 marks)
- (2) Find the flight number, departure airport, departure time and class of seats that haven't been booked so far. Sort the results in the ascending order of flight number. (2 marks)
- (3) Find the name and date of birth of all passengers that have not booked flights of Qantas. (2 marks)
- (4) Find the name and date of birth of all passengers that have booked flights of an airline, of which CEO is JianJiang Cai. (2 marks)
- (5) Find the name, street, city, state, and country of all airlines that have maximum total number of flights. (2 marks)

To create a report from processing of SELECT statements above, open a Terminal window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution5.sql` and save a report in a file `solution5.rpt`. Note, that when started with the options `-v` and `-c` the command line interface includes both listing of SELECT statements processed and the comments included in the original version of a file `solution5.sql`.

A report that contains no listing of processed SQL statements scores no marks and report that contains errors of any kind also scores no marks!

Deliverables

A file `solution5.rpt` with a report from processing of SQL script `solution5.sql`. The report must be created with the command line interface `mysql`, the report **MUST NOT** include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of `solution1.sql`. **Marks will be deducted for the missing comments.** Submission of a file with a different name and/or different extension and/or different type scores no marks.

QUESTION 6 (7 marks)

Download a file `solution6.sql` and insert into the file the implementations of the following Discretionary Access Control (DAC) of the sample database specified in the questions below.

Write SQL script that performs the following operations by a user root.

Your implementation must directly follow a comment with a specification of a subtask.

- (1) Create a database with the same name as a prefix of your University email account. For example, if your University email account is `abc001@uow.edu.au` then a name of a database should be `abc001`.
Create two new user accounts. The names of user accounts are `your_email_name_1` and `your_email_name_2`. For example, if your email name is `abc001`, the two new users are `abc001_1` and `abc001_2`. The passwords are up to you.
(0.5 marks)
- (2) While connected as a user root, process the scripts `dbcreate_f.sql` and `dbload_f.sql` to create and to load data into the relational tables. All relational tables must be located in a database created in step (1). **A listing of SQL statements processed by the scripts must NOT be included in a report from processing of a script `solution6.sql`.** It means that before processing of the script you must process `notee` statement (after `use database_name` command) to turn the spooling off and after processing of the scripts you must process a statement `tee solution6.rpt` to turn the spooling on into a report file.
(0.5 marks)
- (3) Next, grant to the user `your_email_name_1` the read access right on the relational tables `AIRLINE` and `FLIGHT` in the database created in step (1). The read access rights can be propagated to other roles or users.
(1.0 mark)
- (4) Next, grant to the user `your_email_name_1` the write access rights on the relational tables `SEATING` and `BOOKING` in the database created in step (1). The write access rights cannot be propagated to other roles or users.
(1.0 mark)
- (5) Next, grant the read access right to `your_email_name_2` on all relational tables in the database create in the step (1). The privilege cannot be propagated to other roles or users.
(0.5 marks)
- (6) Next, grant the read access right to information about the total number of seats booked by each passenger to user `your_email_name_1`. The privilege can be propagated to other roles or users.
(1.5 marks)
- (7) Next, grant reference privilege to the user `your_email_name_1` on the relational table `SEATING` in the database created in step (1). The privilege can be propagated to other roles or users.
(1.0 mark)

- (8) Next, set the resource limits for the users created in step (1), and it allows for maximum 3 concurrent connections, and maximum 3 connections per hour.

(1.0 mark)

To create a report from processing of a script file `solution6.sql` open a Terminal window and start the command line interface `mysql` in the following way:

`mysql -u root -p -v -c`

Next, process SQL script `solution6.sql` and save a report in a file `solution6.rpt`. Note, that when started with the options `-v` and `-c` the command line interface includes both listing of SQL DAC statements processed and the comments included in the original version of a file `solution6.sql`.

A report that contains no listing of processed SQL statements scores no marks and report that contains errors of any kind also scores no marks!

Deliverables

A file `solution6.rpt` with a report from processing of SQL script `solution6.sql`. The report must be created with the command line interface `mysql`, the report **MUST NOT** include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of `solution6.sql`. Marks will be deducted for the missing comments. Submission of a file with a different name and/or different extension and/or different type scores no marks.

QUESTION 7 (7 marks in the subject CSIT115, 3.5 marks in the subject CSIT815)

An objective of this task is to implement SQL script that verifies the following logical consistency constraint imposed on the contents of a sample database specified in Prologue on Page 2.

"A passenger cannot book two or more different flights in the same day."

Download a file `solution7.sql` and insert into the file the implementations of the following actions.

Your implementation must directly follow a comment with a specification of a subtask.

- (1) First, write SQL statements to inserts into the sample database information about a new booking for a passenger who has already booked a flight in the same day.
Note that you can find a suitable passenger who has booked flights, and then find which flights are available in the same day with the booked flights from the sample database specified in Prologue on Page 2.

(2.0 marks)

- (2) Next, the script creates a single column relational table `AUDIT_BOOKING` to store variable size strings no longer than 500 characters.

(0.5 marks)

- (3) Next, write SQL script to insert into relational table `AUDIT_BOOKING` information about the contents of a sample database that violates the consistency constraint.

"A passenger cannot book two or more different flights in the same day."

The script must list the outcomes of verification of the consistency constraint as a single column table with the following messages as the row in the table.

A passenger <insert passenger name here>, dob is <insert passenger dob here> has booked tickets for flights <insert the first flight number here> and <insert the second flight number here>, departure at <insert the first flight departure time here> and at <insert the second flight departure time here> in the same day.

For example, if a passenger ABC who was born on 20/02/1985, has booked two flights, one flight is Q123, departure from SYD at 20/05/2020 10:30, the other flight is JS213, departure from SYD at 20/05/2020 12:30, then verification of the consistency constraint must return the following message.

A passenger ABC, dob is 1985-02-20, has booked tickets for flights Q123 and JS213, departure at 2020-05-20 10:30 and at 2020-05-20 12:30 in the same day.

Use a function `CONCAT` to create the messages like the one listed above.

(4.0 marks)

- (5) Finally, the script makes the contents of a relational table `AUDIT_BOOKING` permanent and lists the contents of the table.

(0.5 marks)

To create a report from processing of a script file `solution7.sql` open a Terminal window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution7.sql` and save a report in a file `solution7.rpt`. Note, that when started with the options `-v` and `-c` the command line interface includes both listing of SQL statements processed and the comments included in the original version of a file `solution7.sql`.

A report that contains no listing of processed SQL statements scores no marks and report that contains errors of any kind also scores no marks!

Deliverables

A file `solution7.rpt` with a report from processing of SQL script `solution7.sql`. The report must be created with the command line interface `mysql`, the report **MUST NOT** include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of `solution7.sql`. Marks will be deducted for the missing comments. Submission of a file with a different name and/or different extension and/or different type scores no marks.

QUESTION8 (No marks in the subject CSIT115, 3.5 marks in the subject CSIT815)

An objective of this task is to implement your own simple method of auditing the database activities of a sample database specified in Prologue on Page 2.

It is strongly recommended to process a script file dbdrop_f.sql and immediately after that the scripts dbcreate_f.sql and dbload_f.sql to refresh a sample database csit115.

Download a file solution8.sql and insert into the file the implementations of the following actions.

Your implementation must directly follow a comment with a specification of a subtask.

- (1) First, the script makes a relational table that contains a general log empty. (0.25 marks)
- (2) Next, the script sets the appropriate values of the variables to save a general log in a relational table and to start recording a general log from now. (0.5 marks)
- (3) Next, the script makes a database csit115 a default database, it stops recording a report, it executes a script file dbupdate_f.sql, and it resumes recording a report into a file solution8.rpt. (0.5 marks)
- (4) Next, the script sets the appropriate values of all variables to stop recording a general log from now. (0.25 marks)
- (5) Next, the script lists total number of times relational tables have been accessed by DML statements INSERT, UPDATE, and DELETE.
You have to consider the relational tables with the following names AIRLINE, FLIGHT, PASSENGER, SEATING, and BOOKING. No other relational tables need to be considered. The script must list the names of relational tables together with the DML used, and total number of times each table has been accessed by DML statements. Find a fragment of a sample output listed below.

```
+-----+-----+
| TABLE_NAME | TOTAL |
+-----+-----+
| AIRLINE DELETE | 0 |
| AIRLINE INSERT | 1 |
| AIRLINE UPDATE | 1 |
| BOOKING DELETE | 1 |
| BOOKING INSERT | 2 |
| BOOKING UPDATE | 0 |
| ... | ... |
+-----+-----+
15 rows in set (0.01 sec)
```

(2.0 marks)

When ready connect as root user, process a script file `solution8.sql`, and save a report from processing in a file `solution8.rpt`.

To create a report from processing of a file `solution8.sql`, open a Terminal window and start the command line interface `mysql` in the following way:

`mysql -u root -p -v -c`

Next, process SQL script `solution8.sql` and save a report in a file `solution8.rpt`. Note, that when started with the options `-v` and `-c` the command line interface includes both listing of `SELECT` statements processed and the comments included in the original version of a file `solution8.sql`.

A report that contains no listing of processed SQL statements scores no marks and report that contains errors of any kind also scores no marks!

Deliverables

A file `solution8.rpt` with a report from processing of SQL script `solution8.sql`. The report must be created with the command line interface `mysql`, the report **MUST NOT** include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of `solution8.sql`. **Marks will be deducted for the missing comments.** 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 the correct files with the correct contents and correct types. No other submission is possible!

Submit the files **solution1.bmp**, **solution2.rpt**, **solution3.rpt**, **solution4.rpt**, **solution5.rpt**, **solution6.rpt**, **solution7.rpt**, and/or **solution8.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 **CSIT115/CSIT815 (S120)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 Final Assignment**
- (6) Click at a button **Add Submission**
- (7) Move a file **solution1.bmp** 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 the files **solution2.rpt**, **solution3.rpt**, **solution4.rpt**, **solution5.rpt**, **solution6.rpt**, **solution7.rpt**, and/or **solution8.rpt**.
- (9) Click at a button **Save changes**
- (10) Click at a button **Submit assignment**
- (11) Click at the checkbox with a text attached: **By checking this box, I confirm that this submission is my own work, ...** in order to confirm the authorship of your submission
- (12) Click at a button **Continue**

End of specification