School of Information Technology & Computer Science  
University of Wollongong

Session: **3, July 2023**  
Lecturer: **Janusz R. Getta**

# CSCI317 Database Performance Tuning
# Singapore 2023-3
# Assignment 2

Published on 5 August 2023

---

## Scope

This assignment includes the tasks related to decomposition, implementation of generalization and denormalization of conceptual and relational schemas and related to finding the smallest set of indexes that improve performance of processing of a given collection of queries.

This assignment is due by **Saturday, 19 August 2022, 9.00 pm (sharp) Singaporean Time.**

**Please read very carefully information listed below.**

This assignment contributes to 15% of the total evaluation in the subject.

A submission procedure is explained at the end of specification.

This assignment consists of 4 tasks and specification of each task starts from a new page.

It is recommended to solve the problems before attending a laboratory class in order to efficiently use supervised laboratory time.

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

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

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.

It is expected that all tasks included within **Assignment 2** 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 the assessment task.

---

Please read very carefully information included in the Prologue section below about software environment to be used in the subject.

**Prologue**

In this subject we use Oracle 19c database server running under Oracle Linux 7.4 operating system on a virtual machine hosted by VirtualBox. To start Oracle database server you have to start VirtualBox first. If you have not installed VirtualBox on your system yet then it is explained in Cookbook for CSIT115 Recipe 1.1, Step 1 "How to use VirtualBox ?" (`https://www.uow.edu.au/~jrg/115/cookbook/e1-1-frame.html`) how to install and how to start VirtualBox.

When VirtualBox is started, import an appliance included in a file `OracleLinux7.4-64bits-Oracle19c-22-JAN-2020.ova`. You can download `ova` image of the appliance using the links published on Moodle.

When ready, power on a virtual machine `OracleLinux7.4-64bits-Oracle19c-22-JAN-2020`.

A password to a Linux user `ORACLE` is `oracle` and a password to Oracle users `SYSTEM` and `SYS` (database administrators) is also `oracle`. Generally, whenever you are asked about a password then it is always `oracle`, unless you change it.

When logged as a Linux user, you can access Oracle database server either through a command line interface (CLI) `SQLcl` or through Graphical User Interface (GUI) `SQL Developer`.

You can find in Cookbook for CSCI317, Recipe 1, How to access Oracle 19c database server, how to use SQL Developer, how to use basic SQL and SQLcl, and how to create a sample database ? (`https://documents.uow.edu.au/~jrg/317sim/cookbook/e1-2-frame.html`) more information on how to use `SQLcl` and `SQL Developer`.

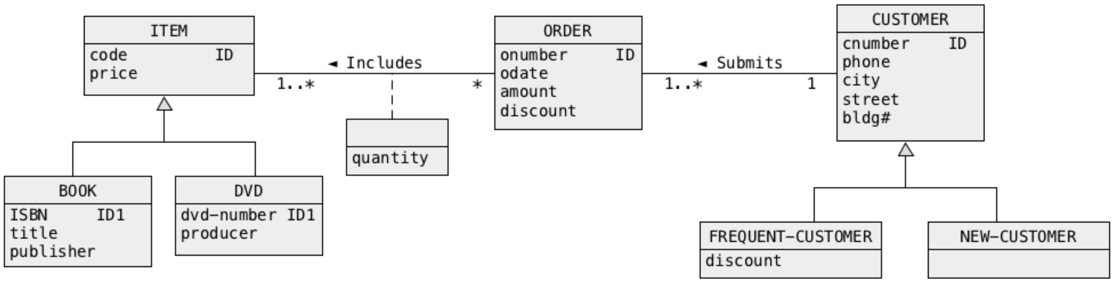
---

## Tasks

### Task 1 (3 marks)

**An objective of this task is to decompose the classes of objects and to implement a generalization in a way that speeds up processing of a given class of applications.**

Consider a conceptual schema given below.



Consider the database applications consistent with the following template.

*Find the total amounts (attribute* `amount` *in a class* `ORDER`*) of orders submitted by a frequent customer with a given number (attribute* `cnumber` *in a class* `CUSTOMER`*) and such that each order included a book with a given* `ISBN` *(attribute* `ISBN` *in a class* `BOOK`*).*

A sample application consistent with the template above is as follows.

*Find the total amounts (attribute* `amount` *in a class* `ORDER`*) of orders submitted by a frequent customer number* `007` *and such that each order included a book with* `ISBN`*: 978-1-4643-2137-3*

We would like to improve the performance of all applications consistent with the template given above through decomposition of object classes and through appropriate transformation of generalization hierarchy.

There is no need to perform denormalization.

Find a transformed conceptual schema that improves the performance of a given class of applications and redraw the entire conceptual schema after the transformation.

To draw a transformed conceptual schema you can use UMLet . A link to UMLet `UMLet 14.3 with CSIT115-815Palette (zipped)` is available at CSCI317 site on Moodle in `Resources` section.

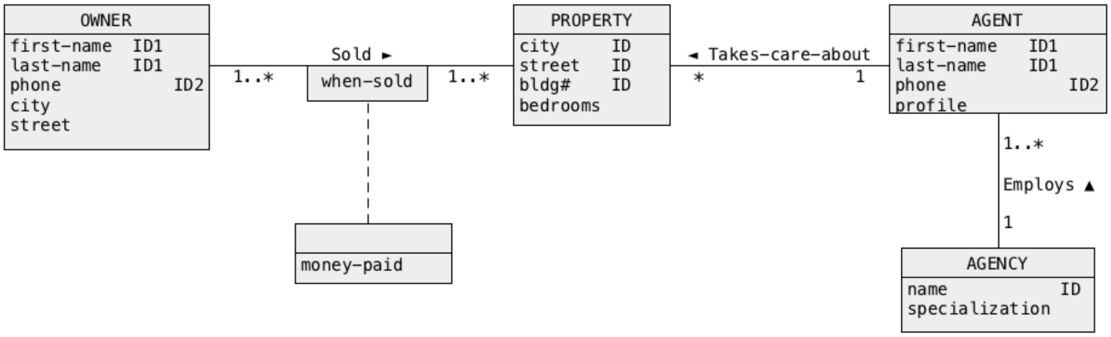
The original conceptual schema is provided in a file `task1.uxf`.

**Deliverables**

A file `solution1.pdf` with a drawing of a transformed conceptual schema. You can use UMLet to draw a transformed conceptual schema. A scanned copy of a neat hand drawing is also acceptable.

**Task 2 (4 marks)**
**An objective of this task is to denormalize a conceptual schema to speed up processing of a given class of applications.**

Consider the following conceptual schema:



(1) Perform simplification of the conceptual schema above and migration of identifiers to implement the associations. Re-draw a simplified conceptual schema with migrated identifiers.

(2)

(3) We would like to improve the performance of the following class of applications:

> *Find the specializations of real estate agencies (attribute `specialization` in a class `AGENCY`) that employ the agents who took care about selling n bedrooms real estate properties (attribute `bedrooms` in a class `PROPERTY`) owned by the owners from a given city (attribute `city` in a class `OWNER`).*

A sample application that belongs to a class described above is the following.

> *Find the specializations of real estate agencies that employ the agents who took care about selling 5 bedrooms real estate properties owned by the owners from Sydney.*

Find a denormalization of the simplified conceptual schema that improves the performance of the class of applications described above. Re-draw a simplified conceptual schema after the denormalizations.

You can use UMLet to create a simplified and denormalized conceptual schemas. A link to UMLet `UMLet 14.3 with CSIT115-815Palette (zipped)` is available at CSCI317 site on Moodle in `Resources` section.

The original conceptual schema is provided in a file `task2.uxf`.

**Deliverables**
A file `solution2.pdf` with a drawing of the simplified conceptual schema and a drawing of the denormalized conceptual schema expressed in a notation of simplified UML

object classes. You can use UMLet to draw a transformed conceptual schema. A scanned copy of a neat hand drawing is also acceptable.

**Task 3 (4 marks)**

**An objective of this task is to denormalize the relational schemas to speed up processing of a given class of applications.**

Use a TPC–HR benchmark database owned by a user tpchr to implement the following query as SELECT statement.

*Find the distinct names (C_NAME) and addresses (C_ADDRESS) of the customers who submitted orders in a given year (year from O_ORDERDATE).*

No report from implementation of SELECT statement is expected.

Implement SQL script solution3.sql that performs the following actions.

(1) First, the script connects as a user tpchr and finds a query processing plan created by a query optimizer for SELECT statement that implements a query listed above. Use EXPLAIN PLAN statement and a script showplan.sql to display the query processing plans. The statement can use any value of parameter that represents a year when the orders have been submitted. The nonempty results are returned for the years in a range 1992..1997. Use a script showplan.sql to list the query processing plan found.

(2) Next, while connected as a user tpchr the script creates a denormalized relational table TASK3. The table should be denormalized in such a way that SELECT statement implementing a query listed above accesses only a table TASK3 and no other table. The redundancies in the denormalized table must be as small as possible.

(3) Next, while connected as a user tpchr, the script fills a denormalized table TASK3 with data copied from the original TPC-H benchmark database.

(4) Next, while connected as a user tpchr the script finds a query processing plan for SELECT statement that implements a query listed above. Use EXPLAIN PLAN statement and a script showplan.sql to display the query processing plans. This time, the statement must use only the denormalized table TASK3. The statement can use any value of parameter that represents a year when the orders have been submitted. The nonempty results are returned for the years in a range 1992..1997.

(5) Next, while connected as a user tpchr the script creates an index that improves processing of a query listed above in the best possible way. Of course, an index should be created on a relational table TASK3.

(6) Next, while connected as a user tpchr the script finds a query processing plan again for SELECT statement that implements a query listed above. Use EXPLAIN PLAN statement and a script showplan.sql to display the query processing plans. This time, an index created in the previous step must be used. The statement can use any

value of parameter that represents a year when the orders have been submitted. The nonempty results are returned for the years in a range 1992..1997.

Compare the values in a column Cost of the query processing plans listed in the steps (1), (4) and (6) to make sure that processing benefited from denormalization and indexing.

When ready execute a script solution3.sql and save a report in a file solution3.lst.

The script must be executed with SQLcl options ECHO and FEEDBACK set to ON such that all SQL statements processed are included in the report !

You must put the following SQLcl statements

```
SPOOL solution3
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 300
SET PAGESIZE 300
```

at the beginning of each SQL script implemented and the following statement at the end of the script

```
SPOOL OFF
```

The report from processing of the script must have NO syntax and/or processing errors !

It is explained in "Cookbook" Recipe 1.5 how to create a report from processing of SQL script.

**Deliverables**
A file solution3.lst that contains a report from the processing of a script solution3.sql.

**Task 4 (4 marks)**
**An objective of this task is to improve performance of query processing through indexing.**

In this task you must operate on the original state of a sample benchmark `TPC-HR` database. It is explained at the end of **Prologue** section how to return to the original state of the database.

Consider the following templates of `SELECT` statements where a string of characters `...` denotes any syntactically valid constant that can be used a statement.

(1)

```
SELECT L_DISCOUNT, L_TAX, L_ORDERKEY
FROM LINEITEM JOIN ORDERS
              ON LINEITEM.L_ORDERKEY = ORDERS.O_ORDERKEY
WHERE ORDERS.O_TOTALPRICE = ...;
```

(2)

```
SELECT L_DISCOUNT, L_TAX, L_ORDERKEY
FROM LINEITEM JOIN PARTSUPP
              ON LINEITEM.L_PARTKEY = PARTSUPP.PS_PARTKEY
                 AND
                 LINEITEM.L_SUPPKEY = PARTSUPP.PS_SUPPKEY
WHERE PARTSUPP.PS_AVAILQTY = ...;
```

(3)

```
SELECT ORDERS.O_TOTALPRICE, ORDERS.O_ORDERKEY
FROM ORDERS
WHERE ORDERS.O_TOTALPRICE = ...;
```

(4)

```
EXPLAIN PLAN FOR
SELECT AVG(L_DISCOUNT), AVG(L_TAX)
FROM LINEITEM
WHERE L_DISCOUNT > ... AND
      L_TAX > ...;
```

An objective of this task is to create **<u>two</u>** indexes that improve processing of the queries listed above in the best possible way.

Implement SQL script `solution3.sql` that performs the actions listed below.

(1) First, the script finds the query processing plans for each one of the queries listed above. Use `EXPLAIN PLAN` statement and a script `showplan.sql` to display the query processing plans.

(2) Next, the script creates **two** indexes that improve the processing of the queries listed above in the best possible way.

(3) Next, the script finds again the query processing plans for each one of the queries listed above. Use `EXPLAIN PLAN` statement and a script `showplan.sql` to display the query processing plans.

(4) Finally, the script provides the calculations of the improvements achieved. To find the improvements use the Cost measures provide by `EXPLAIN PLAN` statement. Calculate the total improvements achieved through indexing. Insert your calculations as the comments at the end of the script.

The script must be processed with SQLcl options `ECHO` and `FEEDBACK` set to `ON` such that all SQL statements processed are included in the report ! To achieve that put the following statement at the beginning of your script:

When ready, process SQL script file `solution4.sql` and save a report from processing in a file `solution4.lst`.

The script must be processed with `SQLcl` options `ECHO` and `FEEDBACK` set to `ON` such that all SQL statements processed are included in the report !

You must put the following `SQLcl` statements

```
SPOOL solution4
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 300
SET PAGESIZE 300
```

at the beginning of each SQL script implemented and the following statement at the end of the script

```
SPOOL OFF
```

A report from processing of the script must have NO syntax errors !

**Deliverables**

A file `solution4.lst` that contains a report from the processing of a script `solution4.sql`.

## Submission

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

Submit the files **`solution1.pdf`**, **`solution2.pdf`**, **`solution3.lst`**, and **`solution4.lst`** 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 **`CSCI317 (SP323) Database Performance Tuning`**

(4) Scroll down to a section **`Submissions`**

(5) Click at a link **`In this place you can submit the outcomes of Assignment 2`**

(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 the files **`solution2.pdf`**, **`solution3.lst`**, and **`solution4.lst`**.

(9) Click at a button **`Submit assignment`** for the bottom of the current web page.

(10) 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.

(11) Click at a button **`Continue`**

(12) Check if **`Submission status`** is **`Submitted for grading`**.

---

*End of specification*