RMIT UNIVERSITY

School of Science

ISYS1055/1057 Database Concepts

Assignment 1

	Assessment Type: Individual assignment; no group work. Submit online via Canvas→Assignments→Assignment 1. Marks are awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
	Due date: Tuesday 7 April 2020, 23:59. Please check Canvas→Syllabus or via Canvas→Assignments→Assignment 1 for the most up to date information. As this is a major assignment in which you demonstrate your understanding, a late penalty of 10% of full available marks per day or part day applies for up to 5 days late. After 5 days, 0 marks will be awarded.
	Weighting: 20 marks

1. Overview

Database systems are a key technology for the storage, management, manipulation, and retrieval of structured data. They have an impact on the use of information technology in applications ranging from banking, to travel bookings, to online shopping. In this assignment you will apply the skills and concepts that you have learned about database systems in the course so far.

2. Assessment Criteria

This assessment will determine your ability to:

1. Follow coding, convention and behavioral requirements provided in this document and in the lessons.
2. Independently solve problems by using database concepts taught over the first several weeks of the course.
3. Understand the relational model.
4. Independently design a database using the ER model.
5. Write and understand SQL queries.
6. Meet deadlines.
7. Seek clarification from your instructor, when needed, via discussion forums.

This assignment is worth twenty points in total for four questions (8+6+6=20), which accounts for 20% of the overall assessment for the course. The assessment components and weights for the course are:

Assignment 1	Assignment 2	Exam
20%	30%	50%

3. Learning Outcomes

This assessment is relevant to the following Course Learning Outcomes:

- CLO 1: Describe various data modelling and database system technologies.
- CLO 3: Identify issues with and compare, justify relational database design.
- CLO 4: Apply SQL as a programming language to define database schemas and update database contents.

It also supports the following Graduate Learning Outcomes:

- Enabling Knowledge: You will gain skills as you apply data modelling knowledge effectively in diverse contexts.
- Critical Analysis: Analyse and model requirements and constraints for the purpose of designing and implementing software artefacts and IT systems.
- Problem solving: Design and implement database solutions that accommodate specified requirements and constraints, based on analysis or modelling or requirements specification.

4. Submission format

Submit your assignment via Canvas→Assignments→Assignment 1. Your submission must be a single .pdf file, with the filename being your student number (e.g., S1234567.pdf) that contains the answers for each question below. Clearly indicate each question number in your file.

It is your responsibility to correctly submit your files. Please verify that your submission is correctly submitted by downloading what you have submitted to see if your .pdf file includes the correct content.

- Never leave submission to the last minute -- you may have difficulty uploading files.
- You can submit multiple times – a new submission will override any earlier submissions. However, if your final submission is after the due time, late penalties apply.
- If unexpected circumstances affect your ability to complete the assignment, you can apply for special consideration. Special Consideration that extends beyond the release of solutions (typically 1—2 weeks) will automatically result in an equivalent assessment in the form of a test, assessing the same knowledge and skills of the assignment (location and time to be arranged by the course coordinator).
- More information on special consideration is available at https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/special-consideration

5. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.
RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the University website.

6. Assessment declaration

When you submit work electronically, you agree to the assessment declaration.

7. Rubric/assessment criteria for marking

The detailed rubric and assessment criteria are available online via Canvas→Assignments→Assignment 1.

8. Assignment questions

Question 1. The Relational Model (8 points)

A University database contains four relations: Student, Class, Staff and Enrol. A sample database instance is shown in Figure 1, where primary keys, and parent-child relations for attributes are annotated. The meaning of most attributes is self-explanatory. Some additional notes for attribute meanings are as follows:

sno: student number
cno: course number
eno: employee number

Answer the questions below based on the database instance given in Figure 1.

1.1. (2 points) Give the schema for each relation. Annotate the primary keys (underline) and any foreign keys (*).

1.2. (4 points) Give the CREATE TABLE statements for each relation, including primary key and any foreign key constraints.

1.3. (2 points) Give the INSERT INTO statements to create the sample database instance shown in Figure 1.

Student

sno	surname	givename	major
s1001	Smith	Tom	History
s1002	Chin	Ann	Maths
s1003	Lee	Perry	Arts
s1005	Smith	John	History
s1006	River	Jane	Arts

Staff

eno	surname	givename	department	rank
e123	Bowl	Alex	Maths	Lecturer
e205	Cox	Kevin	CSC	Associate Professor
e301	Jones	David	Arts	Senior Lecturer

Enrol

sno	cno	grade
s1001	isys155	HD
s1003	cosc121	
s1005	artc131	CR
s1006	cosc101	

Class

cno	lecturer	day	time	room
isys155	e123	Wed	17:30	80.01.12
cosc121	e205	Thu	08:30	12.10.02
artc131	e301	Mon	10:30	10.08.09
cosc101	e205	Tue	14:30	14.09.05

Figure 1 The University database with sample data

RMIT UNIVERSITY

Question 2. SQL (6 points, 0.6 for each item).

*In addition to the lecture notes, you should also study by yourself the SQL*Plus tutorial on Canvas (the Oracle section) and other resources for Oracle syntax and useful functions.*
The relational schema for the Academics database is as follows:

DEPARTMENT(deptnum, descrip, instname, deptname, state, postcode)
ACADEMIC(acnum, deptnum*, famname, givename, initials, title)
PAPER(panum, title)
AUTHOR(panum*, acnum*)
FIELD(fieldnum, id, title)
INTEREST(fieldnum*, acnum*, descrip)

Some notes on the Academics database:

- An academic department belongs to *one* institution (instname) and often has *many* academics. An academic only works for *one* department.
- Research papers (PAPER) are often authored by several academics, and of course an academic often writes several papers (AUTHOR).
- A research field (FIELD) often attracts many academics and an academic can have interest (INTEREST) in several research fields.

Download and run the SQL script *academics.sql* on Canvas (the Oracle section) to define and populate the Academics database in your Oracle account.

Write ONE SQL query for each of **questions 2.3--2.10**. Each component of an SQL statement must be on a separate line. Your query should not produce duplicates in output, but use DISTINCT only if necessary. **Include any explanation as comments starting with "--" so that your SQL queries could be directly executed (in SQL Developer and other SQL client programs).**

2.1. Explain the following query in English. Literal explanations will receive zero marks.

```
SELECT fieldnum, title
FROM field
where (fieldnum>=500 and fieldnum<=599)
   or (upper(title) like 'DATA %'
   or upper(title) like '% DATA %'
   or upper(title) like '% DATA');
```

2.2. The query below is meant to list the panum, title and author acnum of papers and the research interest (fieldnum) of each author, but it has errors. Give the correct SQL query.

```
Select panum, title
From author. Interest, paper
Where author.acnum=interest.acnum;
```

2.3. How many academics are there in the department where deptnum=100? Return the total number.

2.4. List the titles of all papers in the database, in alphabetical order.

2.5. Return the details of research fields which have a *title* starting with the word "Data". Note that the result should include the fields "Data" or "Data Structures" but not "Databases".

2.6. List the panum, title and author acnum of each paper.

2.7. Return the famname and givename of academics working for 'RMIT CS' (*descrip*) with acnum in the range [200..299]. The output should be in alphabetical order of famname and then givename.

2.8. List the famname, givename of academics who work for institutions in Victoria. Note that the values for "Victoria" include "VIC" or "Vic".

2.9. Are there academics who do not have any title? Print their givename, famname. The list should be in alphabetical order of famname and then givename.

2.10. How many institutions are there in the database?

Question 3. ER (6 points).
You are asked to design the ER diagram for a database to manage the data of an online multiplayer team game called "Legendary League". The requirements are:

- Players must register an account to play the game, and have a first name, last name, email address, and a chosen unique username.
- The game has a set of pre-defined characters available that a player can choose to play. Each character has a unique character name, can hold one or more items, has one or more abilities, and comes from a particular region.
- Each item has a unique name and type (where the type is one of "weapon", "shield" or "other").
- Each ability also has a unique name, and an associated rating. (For example, the ability "Fireblast" may currently be at rating 2 for a particular character.)
- Each player can use one or more characters, and go up in levels at different rates with each character. (For example, player "Joe" may currently be at level 1 with the character "Cindra", and at level 15 with character "Gargan").
- Each character comes from a single region, identified by the region's name. (For example, the character "Cindra" may be from the region "Zaund")

According to the above description, construct an Entity Relationship (ER) diagram for the database. Clearly state any assumptions that you need to make. You must represent entities, relationships and their attributes, and all applicable constraints in your ER diagram. Explain any concepts that cannot be expressed in the ER diagram in the associated description.

- Your ER diagram must only use the UML class diagram notations from the lecture notes.
- You are encouraged to use LucidChart to complete your ER diagram. You can create a free student account at: http://www.lucidchart.com/

Some common errors in ER diagrams:

- Every entity must have a primary key.
- The names of all relationships and entities must be unique.
- There are no foreign keys in ER diagrams. They come about as part of mapping an ER model into a relational schema.