

ASSIGNMENT ONE 15 marks

Due Fri Sep 13 2019 at 8pm

Released on 29/7/2019, last updated xx/08

This assignment should be done in a group of no more than 2 students.

This assignment needs to be completed in a group of no more than 2 student members. If such a group is formed, both members must come from the same campus, and also within the tutorial classes of the same tutor whenever possible. While lecturers and tutors will help as much as they can, it is essentially each student's own motivation and responsibility to form a group for this assignment. By default, each student is working in a group containing a single member of himself. Students may however make use of the [student communication board](#) to advertise their availability in seeking an assignment partnership.

In the rare case of one group member becoming seriously ill or uncontactable or not responding, the other member should consider forming a different group or working on his or her own for the assignment. As in real life, everyone should have a contingency plan, or Plan B, and failure of assignment partnership at any point of time will not be accepted as the legitimate reasons for an assignment extension. However, a student is obligated to properly notify his or her existing assignment partner in good time should it become imperative that the student need to terminate the existing partnership due to unexpected circumstances.

Students enrolled in 300941 - Database Design and Development (Advanced) must also complete the [advanced part 1](#) by its respective due date.

- For all the SQL queries in this assignment, students need to submit their SQL statements constructed directly, that is, not to generate the SQL script for you by the GUI, as this would defeat the purpose of practicing how to formulate SQL queries directly.
- All required screenshots must be clearly readable, and the relevant text in on the screenshots must be directly legible on a normal A4-sized printout of the submitted document. Otherwise the screenshots will be deemed having not been submitted.
- Students' main document submitted for their answers to this Assignment must be written in **Microsoft Word**, not in PDF.
- It is anticipated that most students will finish **forming their respective assignment group within 2 weeks**, that is, within semester week 3 and week 4. However, the earlier, the better.

Q1. Basic database design and queries (6 marks)

A small shop wants to create a simple database to keep track of its merchandise prices, buyers and purchase orders. For simplicity, a **merchandise** will be recorded for its name and price, a **buyer** will be recorded for its name, and **order** will be recorded for its date and all the items included in the order.

- i. Design a minimum database (a database of only essential attributes) to fulfil this purpose, and draw the ER diagram for your design. Indicate on the ER diagram the primary keys and the relationship multiplicities. By minimum we mean that you don't need to add anything that is not explicitly stated in the requirements, unless it's one of your artificially generated keys. For instance, you don't need to include a telephone number or email address for the entity corresponding to a buyer. (**1.5 marks**)
- ii. Draw the Global Relation Diagram (GRD) corresponding to the ER diagram in the above, indicating all the primary keys and foreign keys. The GRD should be in a form similar to Figure 17.9 (page 554 or 516 for edition 5) of the textbook, but all the attributes should be kept there too. (**1 mark**)
- iii. Write an SQL script (of statements) that generates all the tables for your designed database. (**1 mark**)
- iv. Write an SQL script to insert sufficient records into your tables. Each table should contain no less than 3 records. At least 2 orders should each contain 2 or more items of the ordered merchandise.

Screenshots are required for the records of all the tables. (**1 mark**)

- v. List all buyers whose name contains **your** own family name as a substring. Insert sufficient records into your table/s so that your query returns at least 2 records. (Screenshot required). (**0.5 marks**)
- vi. For a given order (number), write an SQL statement to list all the item names and their corresponding prices for the order (Screenshot required). (**0.5 marks**)
- vii. List all the orders by their order number, date, and the name of the buyer who places that order (Screenshot required). (**0.5 marks**)

Note that you must provide the screen shots (via ALT-PRINT-SCREEN on Windows for instance) for your results (of executing SQL statements). For each screenshot of executing an SQL script, you must keep your **username** and at least part (if long) of the SQL script in your screenshot. If your username is absent from a screenshot, then the screenshot may not be considered as your work. The screenshot

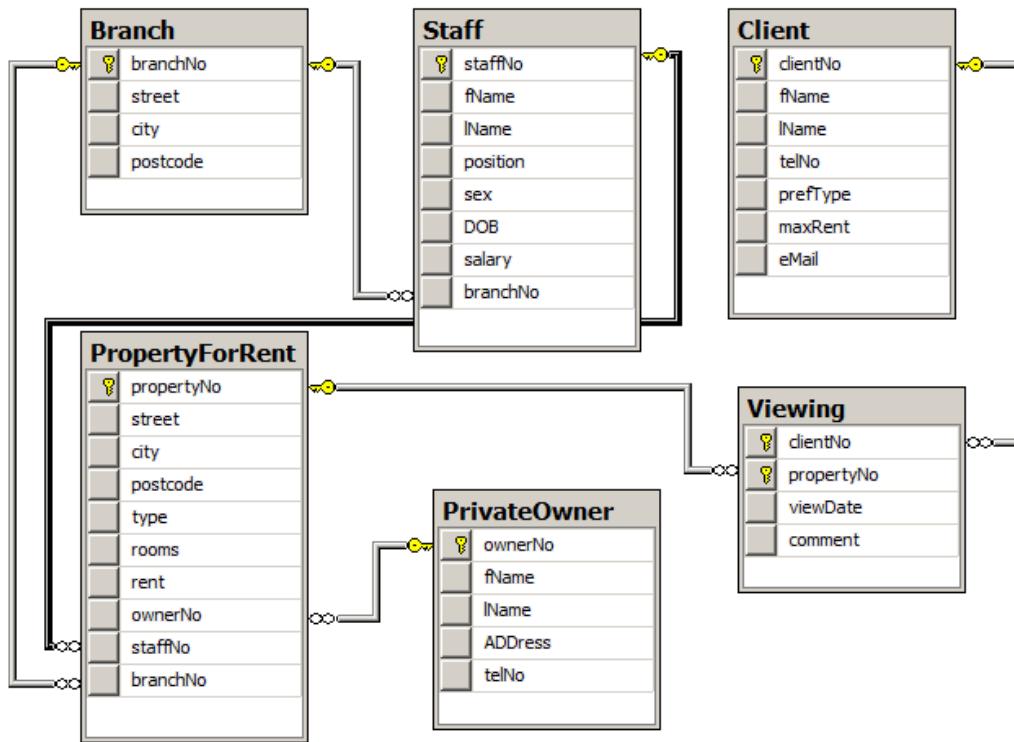
on the right is such an example in which you can see the username "zhuhan" and the full SQL script that produces the result. Not supplying the screenshots, or the screenshots are not readable, may lead to the **loss of up to 30%** of the corresponding marks.

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute MSSQL-TEACHIN... zhuhan.Hotel
Object Explorer
Connect zhuan
+ yzhou
+ zalsattouf
+ zghafari
+ zhuan
+ Database Diagrams
+ Tables
+ System Tables
+ zhuan.Hotel
+ Columns
+ Keys
+ Constraints
+ Triggers
+ Indexes
+ Statistics
+ Views
+ Synonyms
+ Programmability
+ Service Broker
+ Storage
Results Messages
SQLQuery1.sql... \zhuan (56)*
select * from Hotel;
+-----+-----+-----+
| hotelNo | hotelName | city |
+-----+-----+-----+
| 1122 | Riverside | Berlin |
| 3451 | Marriot | Paris |
| 5431 | Ritz | London |
+-----+-----+-----+
Ln 1 Col 21 Ch 21
3 rows
300104.scem.uws.edu.au (10.... VM\zhuhan 00:00 INS
```

Q2. More on SQL queries (3 marks)

A DreamHome database has been created according to a case study for a property rental business (see §11.4 at pages 381-401, or 347-367 for edition 5, of the textbook for the case study). Its (simplified) database schemas (§6.3 at page 197 or 189 for edition 5) and the relation diagram are given as

```
Branch(branchNo,street,city,postcode)
Staff(staffNo,fName,lName,position,sex,DOB,salary,branchNo)
PropertyForRent(propertyNo,street,city,postcode,type,rooms,
                rent,ownerNo,staffNo,branchNo)
Client(clientNo,fName,lName,telNo,prefType,maxRent,eMail)
PrivateOwner(ownerNo,fName,lName,address,telNo,eMail,password)
Viewing(clientNo,propertyNo,viewDate,comment)
```



- i. Draw an ER diagram to represent the above table-linking diagram (which is essentially what we would call a relation diagram). The ER diagram should bear fewer entity types than the number of tables in the above displayed diagram. That is, the table or tables that essentially represent relationships should be represented as relationships on the ER diagram, not as entities. **(0.5 marks)**
- ii. Create this set of tables and fill the records by executing this given SQL script [dreamhome.sql](#). Then use an **UPDATE** statement in SQL to modify the staff member "Julie Lee" into your own name and modify his date of birth (13/6/1975) into a date after 1990s. If you have a team member for this assignment, then also **UPDATE** the staff record for **staffNo="SL21"** by replacing the name "John White" there by that of your team partner (screenshot required for the resulting Staff table). **(0.5 marks)**
- iii. Write an SQL statement to list all the client names, the maximum rent they are willing to pay, and their telephone numbers (screenshot required). **(0.5 marks)**
- iv. Write an SQL statement to list staff name, position, and the postcode of their branch. The listing should be ordered according to postcode, and within the same postcode, ordered alphabetically according to the last name (screenshot required). **(0.5 marks)**
- v. Write an SQL statement to list all area postcode, propertyNo, and the staff name responsible for the management of the property. Sort the output according to postcode (screenshot required). **(0.5 marks)**
- vi. Write an SQL statement to list all the properties that have been viewed by one or more clients. More precisely, list the postcode, propertyNo, the street of the property, last name of the staff responsible for this property, client's last name, and the viewing date. Order the output first by the postcode, then by the street (screenshot required). **(0.5 marks)**

Note that the screenshots you provide should again conform to the requirements indicated in the previous question. For students working in a team of 2 members, the student name contained in the screenshots for this question should be different from the one in the previous question. Otherwise the person whose name is neither on the screenshots of this question nor on those of the previous one may lose the corresponding marks.

Q3. Database modelling - case study (4 marks)

In this question, you are required to construct a **Swimming Database** for a swimming club, so that the database can be used to maintain the listing of excellent performers for different swimming events. The database will record which top swimmers specialise on which events, and these events include for instance 100m Free, 200m Butterfly, 400m Medley, and 1500m Back for both men and women. The performance listing data kept in the database should include gender, performance time, swimmer's name, date of birth,

title of the swimming competition, date of the competition, venue, and the ordinal position in that competition. A typical entry of such a listing might contain the data similar to the following

Gender	M
Event	100 Free
Performance	50.24
Name	David Smith
Date of birth	10/12/2000
Date of competition	10/2/2018
Name of competition	2018 NSW Annual Swimming Games
Venue	Wagga Wagga
Position	1

- i. Design a minimum database (a database of only essential attributes) to fulfil this purpose, and draw the ER diagram for your design. Indicate on the ER diagram the primary keys and the relationship multiplicities. You must use the same notation scheme for the ER diagram as the textbook, and the ER diagram should be strictly in the sense the textbook uses. **(1.5 marks)**
- ii. Draw the corresponding Global Relation Diagram (GRD), indicating all attributes, primary keys and foreign keys where applicable. **(1 mark)**
- iii. Write a query in SQL to list all swimmers who won the 1st place in an event of a competition for the club. Each output record should contain the name of the winning swimmer, the name of the competition and the name of the winning event in that competition. Describe the output also in the form of relational algebra. **(1 mark)**
- iv. Write a query in SQL to list, for all swimmers who won (positioned 1st) at least one event for the club, the swimmer names and their corresponding winning events. The output should however exclude the case where the winning event is not registered in the club as one of the swimmer's specialised events. **(0.5 marks)**

Q4. Selected Additional Exercises (2 marks)

- i. Complete Questions α in the Additional Exercises for Practical 5. **(0.5 marks)**
- ii. Complete Questions β and γ in the Additional Exercises for Practical 6. **(0.5 marks)**
- iii. Complete Questions α in the Additional Exercises for Practical 7. **(1 mark)**

Note on Submission

- This assignment must be submitted electronically via vWSU before the due date. No email submissions will be accepted.
- It is the students' responsibility to retrieve and keep all their submission receipts, as shown in this [demonstration page](#). If in doubt, consult your tutors well before the submission due date. After each submission, a receipt will be automatically emailed back to the student's email account regardless of whether additional email address has been specified on the submission page. Hence, **if you don't receive a receipt, then your submission definitely failed.**
- Submitted files may be zipped together as a single **zip** file (but not as a **zipx** or **rar** file), if a student wishes to do so. However, no other file compression or file archiving formats will be accepted for the submission. If any students insist on submitting their files in **rar** or other non-zip compression formats, a penalty of **1 mark** may be deducted.
- A typical submission should consist of, but not limited to,
 - the **main document in Microsoft Word** that describes your solutions question by question, in the right order, including pertinent diagrams, screenshots, and SQL scripts whenever relevant.
 - a plain text file containing all SQL scripts in the order of their appearance in the main document, separated by relevant comments similar to

```
%%
%% Question 2(iv) - list staff detail, postcode, etc
%%
```

- Each group must submit exactly one copy of their assignment solution electronically by one of the team members. If the other group member really wants to submit it as well due to whatever reasons, then the name of the submitted files must start with "**please_ignore_**" (such files will not be treated

as regular submissions and will be ignored during the marking). Otherwise **1 mark** may be deducted for the duplicated electronic submission.

- Each submission must be accompanied by a declaration of the ownership of the submitted work as described in the [Learning Guide Companion](#). No signature is however required for the electronic submissions. Please note that an examiner or lecturer/tutor has the right not to mark this assignment if a pertinent declaration is not present in your submission.
- [Late submissions](#) will attract a daily incremented late penalty of 10% per day.
- Electronic submission on the due date after 8pm before 12 midnight will still be accepted without penalty. However, any submission failure in that period due to either the student faults or the fault or malfunction of the School's or WSU's servers will not be accepted as the legitimate reasons for a late submission. Beware that School's servers often need to be shut down for maintenance from late Fridays or just before public holidays.
- A statement on the work distribution in percentage (e.g. 50% for David and 50% for Louise) agreed among all the group members. If this statement is absent, then it will be assumed that all group members have made equal amount of contribution to the assignment solution. Achieving a 50%/50% work distribution is also the goal of this team work; the person who contributes less than 50% may result in having less mark than the other team member.
- The main purpose of having an assignment team is to enable students to discuss the database design with another student so as to better understand everything there, rather than splitting the actual work. Hence, regardless of whether a team member contributed 100% or just 50%, the mark remains the same. However, a team member may receive less marks if he contributes less than 50%.
- If any student is making a new assignment group, thus leaving a previous assignment group, he must first obtain a written approval from his tutor or the unit coordinator, unless he will not make use of any work jointly done in the previous team work.
- Students are welcome to leave a hardcopy of their assignment 1 with their marking tutors directly, on any agreed terms between the students and the tutors, prior to their work being already marked, so that on top of the regular feedback in the form of marking sheets additional and more concrete comments or suggestions may be written back to the assignment work on the relevant spots. However, please bear in mind that the electronic submission is the official submission, submitting a hardcopy without submitting the electronic copy within the due date will be deemed NOT having submitted the assignment.
- Any student submitting the assignment on his own must state explicitly whether he was once in a group with another student, and what part of the submitted work actually inherited from a previous joint team work. Failure to make this statement may result in this submission not being marked or a plagiarism case being lodged if the work is similar to another student's, and a late addition of such a statement may lead to the assignment being considered as a late submission.