**Student id:** 20744696

**Student name:** Quang Dong Nguyen

**Student email address:** 20744696@student.westernsydney.edu.au

**Student Class:** Database design and development (INFS2001)

**University:** Western Sydney University

**Author:** Quang Dong Nguyen

**Work contribution level:** 100%

*Statement: This work is contributed by myself (Quang Dong Nguyen) and myself only, there is no involvement from any other students in progression of the work.*

# Question 1: Basic Database Design and queries

## Question 1.i:

Entity-Relation Diagram for the Merchandise Database:

A screenshot of a graph

Description automatically generated

## Question 1.ii:

Global-Relation Diagram for the Merchandise Database: A screenshot of a graph

Description automatically generated

## Question 1.iii:

SQL Script answer for this question is contained in the text file named: Question1\_SQLQuery1

## Question 1.iv:

SQL Script answer for this question is contained in the text file named: Question1\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 1.v:

SQL Script answer for this question is contained in the text file named: Question1\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 1.vi:

SQL Script answer for this question is contained in the text file named: Question1\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 1.vii:

SQL Script answer for this question is contained in the text file named: Question1\_SQLQuery1

A screenshot of a computer

Description automatically generated

# Question 2: More on SQL queries

## Question 2.i:

ER Diagram with Relationship Multiplicities:

A computer screen shot of a computer program

Description automatically generated

ER Diagram without Relationship Multiplicities:

A diagram of a computer

Description automatically generated

## Question 2.ii:

SQL Script answer for this question is contained in the text file named: Question2\_SQLQuery1

A computer screen shot of a computer

Description automatically generated

## Question 2.iii:

SQL Script answer for this question is contained in the text file named: Question2\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 2.iv:

SQL Script answer for this question is contained in the text file named: Question2\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 2.v:

SQL Script answer for this question is contained in the text file named: Question2\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 2.vi:

SQL Script answer for this question is contained in the text file named: Question2\_SQLQuery1

A screenshot of a computer

Description automatically generated

# Question 3: Database Modelling – Case Study

Insert Records for Swimming Database:

SQL Script answer for this question is contained in the text file named: Question3\_SQLQuery1

A screenshot of a computer

Description automatically generated

## Question 3.i:

A grid with text and images

Description automatically generated with medium confidence

Explanation of each entity role:

There are 4 entities in total:

* The first entity, ‘Swimmer\_Assignment’, is used to record information about swimmers (Swimmer Name, Date of birth, Gender).
* The 2nd entity is ‘RaceEvent\_Assignment’, this entity is generally used for recording the type of events held in a Swimming competition, for instance, 100m Free, 200m Butterfly, etc.
* The ‘Competition\_Assignment’ entity is used to record name of competition, date of competition, and venue.
* Lastly, the ‘Performance\_Assignment’ entity stores records of the performance of each swimmer and the final position of swimmers after finishing the competition.

Every entity mentioned above, each will be introduced with a primary key to keep each record of the table unique and distinctive.

Explanation of relationship multiplicities between entities:

Between the ‘Swimmer\_Assignment’ entity and the ‘RaceEvent\_Assignment’ entity, a two-way one-to-many multiplicity relationship is introduced because we want to specify which swimmers can specialise in which type of events, and a swimmer can be specialized in multiple type of events. Furthermore, event types can be specialized for many swimmers, and not just for one swimmer. Hence a many-to-many cardinality relationship is established to help recording swimmers’ specialized events, (this will be represented by a relationship type named ‘Specialty\_Assignment’ later on).

## Question 3.ii:

A diagram of a graph

Description automatically generated with medium confidence

## Question 3.iii:

SQL Script answer for this question is contained in the text file named: Question3\_SQLQuery1

A screenshot of a computer

Description automatically generated

Describe in form of relational algebra:

Since the algebra notation above is too long, the numbers will represent:

|  |  |
| --- | --- |
| 1 | Gender |
| 2 | Swimmer\_name |
| 3 | Date\_of\_birth |
| 4 | Event\_name |
| 5 | Performance |
| 6 | Position |
| 7 | Date\_of\_competition |
| 8 | Name\_of\_competition |
| 9 | Venue |

## Question 3.iv:

SQL Script answer for this question is contained in the text file named: Question3\_SQLQuery1

A screenshot of a computer

Description automatically generated

# Question 4

## Question 4.i:

Supposed that we need to create **a minimum database record of students in New South Wales** so that the government can monitor students records in New South Wales and is able to view their pursues in Universities across New South Wales.

The database will mainly collect information of what Universities the NSW students wished to join in, start date and end date at their universities. Furthermore, the **student** records should contain student id, student name, Date of Birth, Nationality. And The **universities** records should contain University id and University name.

In the example above, we will divide the database into 2 entity tables and 1 relationship type table: Universities, Students, and StudentNSW (relationship type table). The section below shows the attributes in each table:

|  |
| --- |
| Universities   * Uni\_id {PK} * Uni\_name |
| Students   * Student\_id {PK} * Student\_name * Student\_DOB * Nationality |
| StudentNSW   * Student\_id {PK, FK} * Uni\_id {PK, FK} * DateJoin * DateEnd |

Entity-Relation Diagram of the above database design:

A grid with black text

Description automatically generated

Global-Relation Diagram of the above database design:

A screenshot of a graph

Description automatically generated

SQL Script for the creation of table above will be included in the file named: Question4\_i\_UniCourse

## Question 4.ii:

Describe the relation in business terms.



From the database, list all hotel names and their prices where the rooms are of type double.

**Explanation**: Inside the bracket, it produces a join between hotel and those tuples of Rooms with the type of room as ‘Double’. Effectively, together, this will produce a relation with two attributes (hotel Name and Price) containing all hotel Names and their prices with a room type as ‘Double’.

## Question 4.iii:

There are 3 phases of database design:

* Conceptual Database Design
* Logical Database Design
* Physical Database Design

The output for each of the 3 phases of database design is:

* An Entity-Relationship Diagram (for the Conceptual Database Design phase)
* A Global Relation Diagram (for the Logical Database Design phase)
* Creation of tables, constraints and indexes in SQL (for the Physical Database Design phase)

Hence, from the Additional Exercise of the practical 4, the ER Diagram of the database which contains Book, User, Borrows is shown as below:

A graph with a line and text

Description automatically generated

The Global Relation Diagram of the database which contains Book, User, Borrows is shown as below:

A screenshot of a computer

Description automatically generated

* The SQL scripts for the creation of the above tables is described in the text file named: Question4\_iii\_Borrower.