MODULE NAME:	MODULE CODE:
DATABASES	DBAS6211

ASSESSMENT TYPE:	**EXAMINATION (PAPER ONLY)**
TOTAL MARK ALLOCATION:	**120 MARKS**
TOTAL HOURS:	**2 HOURS (+10 minutes reading time)**

INSTRUCTIONS:

1. *Please adhere to all instructions in the assessment booklet.*
2. *Independent work is required.*
3. *Five minutes per hour of the assessment to a maximum of 15 minutes is dedicated to reading time before the start of the assessment. You may make notes on your question paper, but not in your answer sheet. Calculators may not be used during reading time.*
4. *You may not leave the assessment venue during reading time, or during the first hour or during the last 15 minutes of the assessment.*
5. *Ensure that your name is on all pieces of paper or books that you will be submitting. Submit all the pages of this assessment's question paper as well as your answer script.*
6. *Answer all the questions on the answer sheets or in answer booklets provided. The phrase 'END OF PAPER' will appear after the final set question of this assessment.*
7. *Remember to work at a steady pace so that you are able to complete the assessment within the allocated time. Use the mark allocation as a guideline as to how much time to spend on each section.*

Additional instructions:

1. *This is an OPEN BOOK assessment.*
2. *For open book assessments the students may have open access to all resources inclusive of notes, books (hardcopy and e-books) and the internet. These resources may be accessed as hard copies or as electronic files on electronic devices. All electronic devices batteries must be fully charged before the assessment as no charging of devices will be permitted during the sitting of the assessment. The IIE and associated brands accept no liability for the loss or damage incurred to electronic devices used during open book assessments.*
3. *Answer All Questions.*
4. *Instructions for assessments including practical computer work:*
   - *Use of good programming practice and comments in code is compulsory.*
   - *Save your application in the location indicated by the administrator (e.g. the Z:\ drive or your local drive).*
   - *Create a folder as follows: use the module code and your own student number and create a folder with a folder name as per the format shown here:*
   - ***StudentNumber_ModuleCode_Exam**. Save all files (including any source code files, template files, design files, image files, text files, database files, etc.) within this folder.*
   - *E.g. if your student number is 12345, and you are writing an examination for the module PROG121, create a folder named **12345_Prog121_Exam** and use this throughout the session to save all of your files.*

- ***Important:*** *Upon completion of your assessment, you must save and close all your open files and double click the ExamLog application on your desktop. You must follow the instructions carefully to ensure that the information about the files that you have submitted for this assessment has been logged on the network. Specify the location of your source code on your question paper.*

Question 1 – Entity Relationship Diagram (Marks: 20)

Answer this question in your answer booklet.

Draw an Entity Relationship Diagram (ERD) using Unified Modelling Language (UML) notation according to the below business rules. Your design should be at the logical level – include primary and foreign key fields and remember to remove any many-to-many relationships.

Tip: Pay attention to the mark allocation shown below.

Business rules for a horse-riding competition:

1. All entities should have surrogate primary keys.
2. Each rider may only represent a single club, but each club may enter many riders into the competition.
3. The name of each club must be recorded in the database.
4. The name and surname of each rider must be recorded in the database.
5. Each rider may only ride one horse, and every horse may only have one rider.
6. The name of each horse must be recorded in the database.
7. A horse may compete in multiple different events, and each event will have many horses that compete in it.
8. The name of each event must be stored in the database.

Marks will be awarded as follows:

Entities	5 marks
Relationships	4 marks
Multiplicities	4 marks
Primary keys	2 marks
Foreign keys	2 marks
Other attributes	2 marks
Correct UML Notation	1 mark
Total	**20 marks**

Question 2 – Normalisation (Marks: 20)

Answer this question in your answer booklet.

Data about the horses that have been awarded top places in events has already been captured in a spreadsheet (an extract from the spreadsheet is shown below). The data has been normalised to first normal form already – underlined column names indicate composite primary key columns.

Horse ID	Horse Name	Breed ID	Breed Name	Event ID	Event Name	Place
1	Speedy	1	Thoroughbred	15	Show Jumping	1
1	Speedy	1	Thoroughbred	2	Dressage	5
2	Candy Floss	2	Appaloosa	15	Show Jumping	2
3	Getting Warmer	1	Thoroughbred	2	Dressage	4
4	Mother's Love	9	Arabian	3	Eventing	3

Normalise the above data to third normal form (3NF). Show all steps as well as the final answer in the form of dependency diagrams.

Question 3 – SQL (Marks: 60)

The answer for this question should be submitted digitally (see instructions on the cover page).

Using MySQL, create a single Structured Query Language (SQL) script that answers all the below questions. Include comments to indicate which part of the script answers which question. The script must execute correctly using MySQL to get full marks. Make use of the following data dictionary:

Table: Horse

Field name	Data type	Data format	Field size	Req?	Description	Example
Horse {PK}	int			Yes	Autonumber primary key	1
Name	varchar(50)		50	Yes	Name of the horse	Bold Ruler

Table: Race

Field name	Data type	Data format	Field size	Req?	Description	Example
RaceID {PK}	int			Yes	Autonumber primary key	2
Name	varchar(50)		50	Yes	Name of the race	2019 Vodacom Durban July
Date	date	YYYY-mm-dd		No	Date of the race	2019-07-06

Table: Entry

Field name	Data type	Data format	Field size	Req?	Description	Example
EntryID {PK}	int			Yes	Autonumber primary key	3
HorseID {FK1}	int			Yes	Foreign key that links to the Horse table	1
RaceID {FK2}	int			Yes	Foreign key that links to the Race table	2

Q.3.1 Create a schema called horseraces_<studentNumber>, as well as the Horse, Race and Entry tables. (15)

Q.3.2 Insert the following data into the tables: (7)

Table: Horse

HorseID	Name
1	Bold Ruler
2	Phantom

Table: Race

RaceID	Name	Date
1	Most Important Race 2019	2019-07-05
2	Some Other Race 2020	2020-11-19

Table: Entry

EntryID	HorseID	RaceID
1	1	1
2	1	2
3	2	2

Q.3.3	Query all the names of all the horses in the database, sorted from Z to A.	(2)
Q.3.4	Query the races where the date of the race is from 2019-01-01 to 2019-12-31 (both inclusive). Include all the columns from the Race table only.	(5)
Q.3.5	Determine how many races each horse entered. Include the Horse ID and the number of races that the horse participated in.	(5)
Q.3.6	Query the name of the horse and the name of the race for all the race entries recorded in the database.	(5)
Q.3.7	Add a unique index on the Name column in the Race table.	(5)
Q.3.8	Create a view called 'getthisyearsraces' that queries the names and dates of the races that have a date that is in the current year. Note: This query should work for any year.	(6)
Q.3.9	Create a stored procedure called 'count_horse_races'. It should take the name of the horse as input and determine the number of races it has entered. Only return the number of races, and include a call to the newly created stored procedure to get the number of races that "Phantom" entered.	(10)

Question 4 – NoSQL (Marks: 20)

The answer for this question should be submitted digitally (see instructions on the cover page).

A gym wants to store user profile information in a NoSQL database. Write MongoDB interactive shell commands to complete the tasks below. Copy and paste all the commands from the shell into a single text file for submission. The commands must execute correctly using the MongoDB shell to get full marks.

Q.4.1	Create a database called gym_<your-student-number>. The <your-student-number> part should be replaced with your student number, for example gym_29876543.	(2)

Q.4.2 In a collection called userProfiles, create the following data: (8)

Name	Surname	Date of Birth	Fitness Level
Dominic	Badeaux	1982-09-04	1
John	Dlamini	1974-05-18	2

Q.4.3 Query all the data in the userProfiles collection. (2)

Q.4.4 Query all the user profiles where the fitness level of the user is equal to 2. (3)

Q.4.5 Query all the user profiles where the date of birth of the user is from 1980-01-01 to 1982-12-31 (including both dates). (5)

END OF PAPER