MODULE NAME:	MODULE CODE:
DATABASES	DBAS6211
DATABASES	DBAS6211d

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 restaurant chain with several restaurants around the country:

1. All entities should have surrogate primary keys.
2. Each dish has a dish type, and there can be lots of dishes with the same dish type.
3. The description for each dish type must be stored in the database.
4. The name and description for each dish must be stored in the database.
5. Each restaurant has exactly one owner, but an owner may own multiple restaurants in the chain.
6. The name and surname of each owner must be stored in the database.
7. The name of each restaurant must be stored in the database.
8. Each restaurant may select several dishes to put on their menu, and a dish can be served at many different restaurants.
9. Since dish selections can change over time, the start date and end date of each dish selection must also 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**

<u>Question 2 – Normalisation (Marks: 25)</u>

Answer this question in your answer booklet.

A lot of data about restaurant staff employment has already been collected 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.

Note: A waiter can work at a specific restaurant only once. But waiters are allowed to move to a different restaurant in the chain. The employment type for a waiter never changes.

<u>Restau-rant ID</u>	**Restaurant Name**	**<u>Waiter ID</u>**	**Waiter Name**	**Waiter Surname**	**Employee Type ID**	**Employee Type**	**Start Date**	**End Date**
1	Bears of Brooklyn	2	John	Smith	1	Permanent	2020-01-19	2020-09-15
1	Bears of Brooklyn	3	Sarah	Ndambi	2	Temporary	2021-09-09	2021-10-30
2	Cats of Cape Town	3	Sarah	Ndambi	2	Temporary	2021-11-02	-
2	Cats of Cape Town	4	Themba	Dlamini	1	Permanent	2022-02-02	2022-03-15
3	Doves of Durban	5	Pete	Nair	2	Temporary	2022-04-30	-

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

Question 3 – SQL (Marks: 60)

The answer to this question should be submitted digitally.

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: Restaurant

Field name	Data type	Data format	Field size	Req?	Description	Example
RestaurantID {PK}	int			Yes	Autonumber primary key	1
Name	varchar(100)		100	Yes	The name of the restaurant	Bears of Brooklyn
Occupancy	int			Yes	Number of people allowed in the restaurant	50

Table: City

Field name	Data type	Data format	Field size	Req?	Description	Example
CityID {PK}	int			Yes	Autonumber primary key	2
Name	varchar(100)		100	Yes	The name of the city.	Cape Town

Table: RestaurantCity

Field name	Data type	Data format	Field size	Req?	Description	Example
RestaurantCityID {PK}	int			Yes	Autonumber primary key	14
RestaurantID {FK1}	int			Yes	Foreign key that links to the Restaurant table	2
CityID {FK2}	int			Yes	Foreign key that links to the City table	1

OpenDate	date	YYYY-mm-dd		Yes	Date that the restaurant opened	2022-01-01
CloseDate	date	YYYY-mm-dd		No	Date that the restaurant closed	2022-01-31

Q.3.1 Create a schema called restaurantgroup_<studentNumber>, as well as the Restaurant, City, and RestaurantCity tables. (15)

Q.3.2 Insert the following data into the tables:

Table: Restaurant

RestaurantID	Name	Occupancy
1	Bears of Brooklyn	100
2	Cats of Cape Town	500

Table: City

CityID	Name
1	Pretoria
2	Cape Town

(7)

Table: RestaurantCity

RestaurantCityID	RestaurantID	CityID	OpenDate	CloseDate
1	1	1	2020-01-01	
2	2	2	2022-01-01	2022-02-03

Q.3.3 Query the names of all the restaurants in the databases, sorted from highest occupancy to lowest occupancy. Include all columns from the restaurant table. (2)

Q.3.4 Query the restaurants that closed between 2022-01-01 and 2022-12-31, both inclusive. Include the CityID, OpenDate and CloseDate in the results. (5)

Q.3.5 Query the number of restaurants in each city. Include the CityID and the number of restaurants in the results. (5)

Q.3.6 Query a list of all the restaurants in the database. Include the name of the restaurant, the name of the city where it is located, and the restaurant's occupancy in the results. (8)

Q.3.7 A restaurant is currently open if it does not have a closing date and the opening date is not in the future. Create a view that gets the list of restaurants that are currently open. Include just the name of each open restaurant in the results. (8)

Q.3.8 Create a stored procedure called count_restaurants_in_city. It should take the CityID for the city as input and determine the number of restaurants in the city. Then, only return the number of restaurants. And include a call to the newly created stored procedure to get the number of restaurants in the city 2. (10)

Question 4 – NoSQL (Marks: 15)

The answer to this question should be submitted digitally.

A restaurant wants to store recipe information in a NoSQL database. Write **MongoDB interactive shell commands** to complete the tasks below. Then, 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 recipes_<your-student-number>. The <your-student-number> part should be replaced with your student number, for example recipes _st29876543. (2)

Q.4.2 In a collection called recipeBook, create the following data:

RecipeName	Type	MainIngredient	Calories
Choc Pudding	Dessert	Chocolate	500
Toasty	Main	Bread	200

(6)

Q.4.3 Get a list of all the recipes in the collection. (3)

Q.4.4 Query all the recipes where the calories are less than or equal to 300. (4)

END OF PAPER