THE INDEPENDENT INSTITUTE OF EDUCATION

MODULE NAME:	MODULE CODE:
DATABASES	**DBAS6211/d/p**

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

SETUP TIME – SPECIAL INSTRUCTIONS:

1. *For practical IT tests or exams written on campus, the usual reading time is replaced by an **additional 30-minute setup time** allocated for setup, saving and upload activities.*
2. *Students are allowed to make notes during the 30-minute setup time.*
3. *Students are allowed to **start working** on their practical solutions as soon as the 30-minute setup time starts.*

INSTRUCTIONS:

1. *Please adhere to all instructions in the assessment booklet.*
2. *Independent work is required.*
3. *Ten 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 device 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 submitting your assessment:*
   - *Use of good programming practice and comments in code is compulsory.*
   - *Save your solution/project in the designated space for this module.*
   - *Save all files (including any source code files, template files, design files, image files, text files, database files, etc.) within the designated space.*

Question 1 (Marks: 20)

Answer this question in your answer script.

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.

- All entities must have surrogate primary keys.
- Each medication contains one or more active ingredients, and an active ingredient can be used in more than one medication in different quantities.
- The name of each medication should be stored in the database.
- The name of each active ingredient must be stored in the database.
- Each medication belongs to one specific category, and many medications can belong to the same category.
- The description of each category must be stored in the database.
- An active ingredient belongs to a specific category, and many active ingredients can belong to the same category.
- The description of each category must be stored in the database.

Tip: Pay attention to the mark allocation shown below.

Marks will be awarded as follows:

Entities	5 marks
Relationships	4 marks
Multiplicities	4 marks
Primary keys	2½ marks
Foreign keys	2 marks
Other attributes	2½ marks
Total	**20 marks**

Question 2 (Marks: 20)

Answer this question in your answer script.

A spreadsheet already contains a lot of data about pharmacies and the pharmacists who work in them. An extract from the spreadsheet is shown below. The data has already been normalised to the first normal form.

Consider the table in first normal form (1NF) below, and then answer questions 2.1 and 2.2.

PharmacyID	PharmacyName	GroupID	GroupName	PharmacistID	PharmacistName
1	Dischem Sandton	1	Dischem	3	Thandi Siko
2	Dischem Soweto	1	Dischem	6	Jake Oliver
3	Fourways Klinicare	2	Klinicare	2	Steve da Costa
4	Clicks Sandton City	3	Clicks	4	John Makalima
5	Clicks Soweto	3	Clicks	8	Jane Makalima
6	Clicks Montecasino	4	Clicks	7	Steve van Rooyen
7	Randburg Pharmacare	5	Pharmacare	1	Sarah Coetzee
8	Fourways Pharmacare	5	Pharmacare	5	Bongi Mbete

Q.2.1	Normalise the table presented to the second normal form (2NF), showing all steps with explanations. All steps, as well as the final answer, must be in dependency diagram format.	(15)
Q.2.2	Once you have answered Question 2.1, extend your answer to normalise to the third normal form (3NF), showing all steps with explanations. All steps, as well as the final answer, must be in dependency diagram format.	(5)

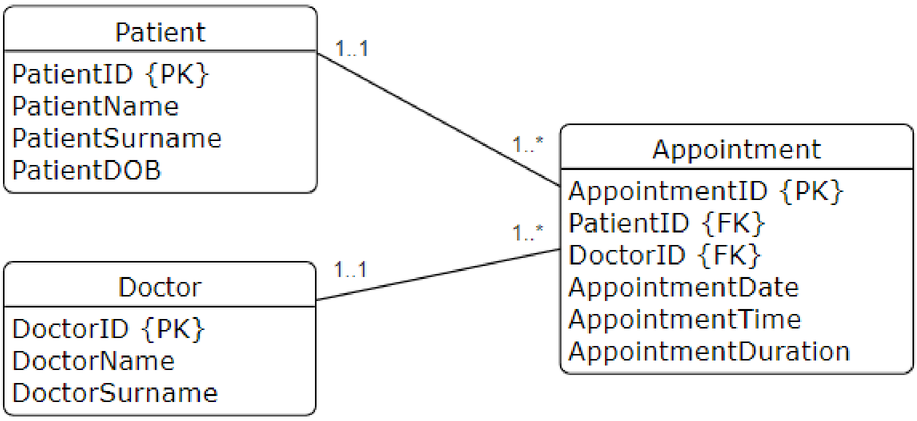
Question 3 **(Marks: 65)**

Practical computer work: The answer to this question should be submitted electronically.

Using MySQL, create a **single** Structured Query Language **(SQL) script** that answers all the questions below. Include **comments** to indicate which part of the script answers which question.

The script **must execute correctly** using MySQL to get full marks.

Consider the following extract from an Entity Relationship Diagram (ERD) before answering questions 3.1 to 3.8:



Q.3.1 Write an SQL statement to create the *Patient* table. (5)

Q.3.2 Write an SQL statement to create the *Doctor* table. (4)

Q.3.3 Write an SQL statement to create the *Appointments* table. (9)

Q.3.4 Write SQL statements to insert the following data: (11)

Table: Patient

Patient ID	**Patient Name**	**Patient Surname**	**Date of Birth**
1	Debbie	Theart	1980-03-17
2	Thomas	Duncan	1976-08-12

Table: Doctor

Doctor ID	Doctor Name	Doctor Surname
1	Zintle	Nukani
2	Ravi	Maharaj

Table: Appointments

Appointment ID	Appointment			Doctor ID	Patient ID
	Date	Time	Duration		
1	2024-01-15	9:00	15	2	1
2	2024-01-18	15:00	30	2	2
3	2024-01-20	10:00	15	1	1
4	2024-01-21	11:00	15	2	1

Q.3.5 Write an SQL statement to display all the appointments between 2024-01-16 and 2024-01-20 (inclusive). (4)

Q.3.6 Write a SQL statement to display the names and surnames of patients with the total number of appointments they have. Sort the results in descending order based on the number of appointments. (6)

Q.3.7 Write an SQL statement to display all the appointments. The statement should display the appointment date (in descending order), the time, the doctor's name and surname, and the patient's name and surname. (10)

Q.3.8 Create a view that gets a list of all the patients that have appointments with the doctor with Doctor ID 2. Include just the name and surname of the patients in the result and sort results in ascending alphabetical order according to surname. (6)

Q.3.9	Create a stored procedure called get_appointments. It should take the date of an appointment as input and display all the appointment details for that date. Display the time (in ascending order), duration, doctor name and surname, as well as patient name and surname.	(10)

Question 4 (Marks: 15)

Practical computer work: The answer to this question should be submitted electronically.

A medical practice wants to store patient details in a NoSQL database. Write **MongoDB interactive shell commands** to complete the task below. Copy and paste 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 patients_<your-student-number>. The <your-student-number> part should be replaced by your student number, for example, patients_s123456.	(2)
Q.4.2	In a collection called patients, create the following data:	(6)

Patient Name	Patient Surname	Date of Birth
Debbie	Theart	1980-03-17
Thomas	Duncan	1976-08-12

Q.4.3	Get a list of all the notes in the collection.	(3)
Q.4.4	Query all the notes of patients born after (not including) 1979-01-12.	(4)

END OF PAPER