UNIVERSITI TEKNOLOGI MALAYSIA
FINAL EXAMINATION SEMESTER 1, SESSION 2024/2025

SUBJECT CODE : SECD2523

SUBJECT NAME : DATABASE

SECTION : 01 / 02 / 03 / 04 / 05 / 06 / 07 / 08 / 09 / 10

TIME : 3 HOURS

DATE/DAY :

VENUES :

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO.

INSTRUCTIONS:

1. Answer **ALL** questions in the given answer booklet.
2. This question paper consists of **THREE (3) sections**.
   Section A: **TEN (10)** multiple choice questions.
   Section B: **FOUR (4)** structured questions.
   Section C: **TWO (2)** case study questions.
3. Submit both question paper and answer booklet at the end of the examination.

(Please Write Your Name and Section in Your Answer Booklet)

Name	
IC/ISID No.	
Year / Course	
Section	
Lecturer Name	

WARNING: Disciplinary action will be taken against students who are found cheating during the examination.

This question paper consists of **SEVENTEEN (17)** printed pages including this page.

SECTION A: MULTIPLE CHOICE QUESTION (10 MARKS)

This section consists of **ten (10) questions**. Answer **all** questions in the **Answer Booklet**.

1 IT Bakery had two types of Bakers. The Full-Time Bakers worked daily and earned a monthly salary. The Part-Time Bakers were called in during busy seasons or for big orders. Which of the following is **incorrect** for matching the entities with superclass and subclass?

A Bakers (superclass)

B IT Bakery (superclass)

C Full-Time Bakers (subclass)

D Part-Time Bakers (subclass)

2 Which of the following is/are **true** regarding the constraints of superclass/subclass relationship?

I. The constraint of "AND" indicates that an instance can only belongs to one subclass.
II. The disjoint constraint determines whether an instance would belong to multiple subclasses at the same time or not.
III. Constraint of "MANDATORY" means that not every instance is categorized into the subclasses.
IV. The participation constraints determines whether all instances or some of the instances will be categorized into the subclasses.

A I and II

B I and III

C II and IV

D III and IV

3 In the derivation of a new relation in logical database design, which of the following is **should not be considered** for participation constraints in one-to-one (1:1) binary relationship types?

A Optional participation on one side only.

B Mandatory participation on both sides.

C Optional participation on both sides.

D Mandatory participation on one side, optional on the another.

4 Which type of the functional dependency is **satisfied** for:

student_matric → program_code

program_code → program_name, faculty_code, faculty_name

student_matric → program_name, faculty_code, faculty_name

A Full Functional Dependencies

B Boyce-Codd Dependencies

C Transitive Dependencies

D Partial Dependencies

5 Which of the following tasks is **most** suitable for Online Analytical Processing (OLAP) only?

A Managing daily transactions in a retail store.

B Real-time processing of ATM transactions.

C Calculating monthly sales trends across multiple regions.

D Recording and updating inventory levels in a warehouse.

6 Which of the following characteristics is **not** related to data warehouse?

A Subject-oriented

B Integrated

C Non-volatile

D Real-time processing

7 Which of the following is **true** if the table (StudentID, CourseID, StudentName, InstructorName) is still in the **second normal form (2NF)**, with the following dependencies: StudentID → StudentName; CourseID → InstructorName; ?

A No column that is not part of the primary key is dependent on only a portion of the alternate key.

B No column that is not part of the primary key is dependent on only a portion of the foreign key.

C No column that is not part of the primary key is dependent on the candidate key.

D No column that is not part of the primary key is dependent on only a portion of the primary key.

8 Which of the following is **not** concurrency control problems?

A Inconsistent analysis

B Data redundancy

C Lost update

D Uncommitted dependency

9 **Table A.1** shows **Students** relation including StudentID, Name, Marks, and Class.

Table A.1: **Students.**

StudentID	Name	Marks	Class
113	Ahmad Tan	85	10A
156	Iman Nick	98	10B
216	Leena Aiden	78	10A
217	Jaden Rao	92	10C

Sarah wants to display all **Students** data in **Table A.1** based on their Class and count how many **Students** are in each Class. Which SQL query can perform this task?

A SELECT Class, COUNT(StudentID) AS StudentCount FROM **Students** GROUP BY Class HAVING COUNT(StudentID) > 1;

B SELECT Class, COUNT(StudentID) AS StudentCount FROM **Students** GROUP BY Class;

C SELECT Class, COUNT(StudentID) FROM **Students** GROUP BY Class HAVING COUNT(StudentID) > 1

D SELECT Class, COUNT(StudentID) FROM **Students** GROUP BY Class

10 **Table A.2** shows the **Student_Registration** relation including StudentID, Name, and Program.

Table A.2: **Student_Registration**.

StudentID	**Name**	**Program**
A24EC1	Ariana Abby	Master Artificial Intelligence
A24EC2	Aiden Rao	Master Biomedical Engineering
A24EC3	Sofea Nick	Master Civil Engineering
A24EC4	Iman Tan	Master Electrical Engineering

The following SQL statements would successfully update the student’s information from the **Student_Registration** table (**Table A.2**), **except** for:

A Update **Student_Registration** (StudentID, Program) SET ('A24EC5 ', 'Master Data Science');

B Update **Student_Registration** SET Program = 'Master Data Science' WHERE StudentID = 'A24EC1';

C Update **Student_Registration** SET Name = 'Aidan Rao' WHERE StudentID = 'A24EC2';

D Update **Student_Registration** SET Program = 'Master Mechanical Engineering' WHERE StudentID = 'A24EC4';

SECTION B: STRUCTURED QUESTIONS **(60 MARKS)**

This section consists of **four (4) questions**. Answer **all** questions in the **Answer Booklet.**

Question 1: Answer **all** questions based on **two (2)** case studies in **(a)** and **(b).**

(18 Marks)

a) **Case 1:**

In a cinema system, a **User** can be defined as a **Viewer**, **Staff**, or both, but they may also be neither. Information about the **User** includes userID, name, dateOfBirth, and address. Each **Viewer** must be classified as either a **Regular** or a **Member**. Information on **Viewer** includes viewerID and preferences. **Members** have additional details such as membershipID and membershipType. A **Staff** contains details such as staffID, role, and department. A **Viewer** can book multiple tickets, but each ticket is associated with one **Viewer**. A **Staff** member can assist multiple **Viewers**, but each **Viewer** can receive assistance from only one **Staff** member per interaction.

Based on the description, **DRAW** the **Enhanced Entity Relationship Diagram (EERD)** and include relevant attributes of each superclass, subclasses, relationships, and constraints using the **UML** notation.

(8 Marks)

b) **Case 2:**

The car rental company maintains a database to record **Customer** information for those who make **Rentals**. When a **Rental** is confirmed, a **Car** is assigned to the **Rental**. After the **Car** is returned, **Payment** details are recorded, including the payment method (*Credit Card* or *Cash*). The **Staff** tracks the processing of **Rentals** and handles all related tasks.

Figure B.1 (refer to **page 8**) shows the CONCEPTUAL Entity Relationship Diagram (ERD) of the database system for car rental.

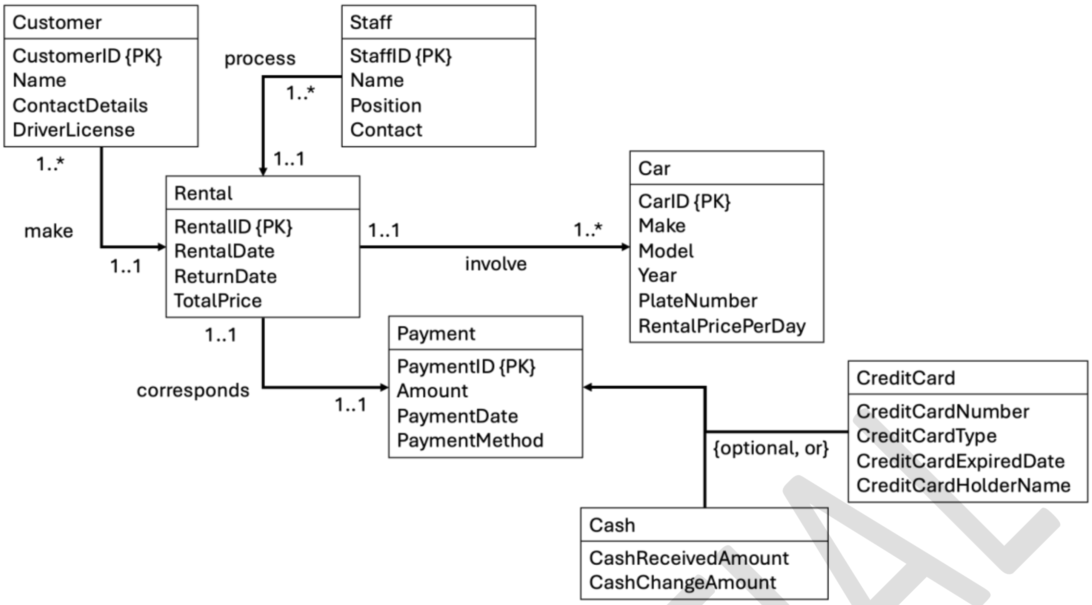

Figure B.1: Conceptual Entity Relationship Diagram (ERD) of the car rental database management system.

Based on **Figure B.1**, derive **all** relational schemas for the **LOGICAL ERD**.

(10 Marks)

Question 2: Answer **all** questions based on the case study of ACE Driving School (refer to **Page 11** and **Page 12**).

(17 Marks)

ACE Driving School offers a variety of driving lessons for students. Each lesson is conducted by a certified instructor. The school currently tracks all lesson details in a single table.

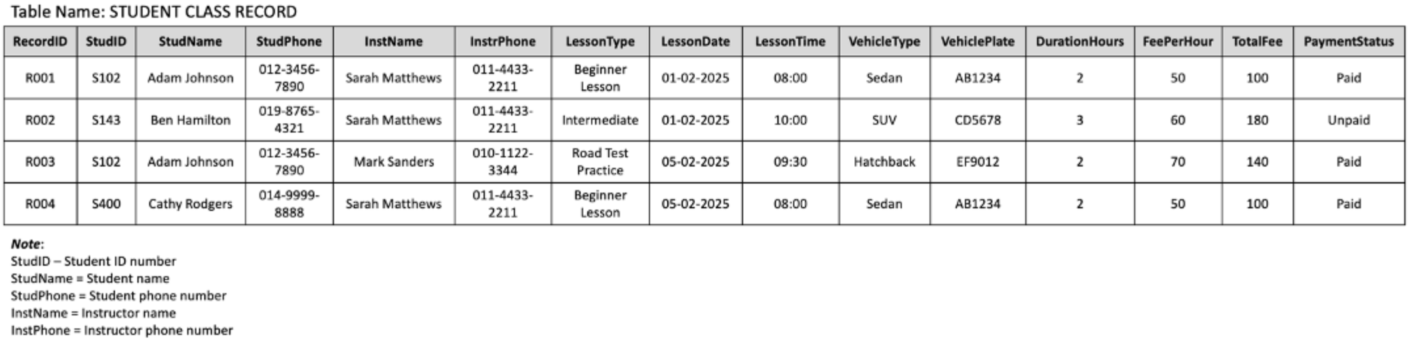

Table Name: STUDENT CLASS RECORD

RecordID	StudID	StudName	StudPhone	InstName	InstrPhone	LessonType	LessonDate	LessonTime	VehicleType	VehiclePlate	DurationHours	FeePerHour	TotalFee	PaymentStatus
R001	S102	Adam Johnson	012-3456-7890	Sarah Matthews	011-4433-2211	Beginner Lesson	01-02-2025	08:00	Sedan	AB1234	2	50	100	Paid
R002	S143	Ben Hamilton	019-8765-4321	Sarah Matthews	011-4433-2211	Intermediate	01-02-2025	10:00	SUV	CD5678	3	60	180	Unpaid
R003	S102	Adam Johnson	012-3456-7890	Mark Sanders	010-1122-3344	Road Test Practice	05-02-2025	09:30	Hatchback	EF9012	2	70	140	Paid
R004	S400	Cathy Rodgers	014-9999-8888	Sarah Matthews	011-4433-2211	Beginner Lesson	05-02-2025	08:00	Sedan	AB1234	2	50	100	Paid

Note:
StudID – Student ID number
StudName = Student name
StudPhone = Student phone number
InstName = Instructor name
InstPhone = Instructor phone number

Figure B.2: Data for Student Class Record

The agency has asked you to redesign a new system for them because they encounter problems with the current system. The first thing you do is to study the database design and you realize that database tables are not normalized, leading to redundancy and potential inconsistencies.

After discussing with the manager, you identified **RecordID** and **StudID** as the composite primary keys. Assuming that the **InstrPhone** and the **VehiclePlate** are **unique**, the dependencies of the relations are as follows:

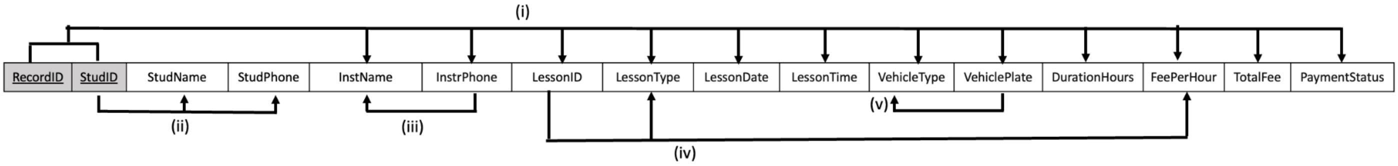

Figure B.3: Student Class Record with its dependencies

Based on the data example and dependencies diagram, answer all questions:

a) According to **Figure B.3**, **identify** the **dependencies** in **(i)**, **(ii)**, **(iii), (iv)**, and **(v)**.

(2.5 marks)

b) Based on your answer in **(a)**, **remove the partial dependencies** and **produce the relations schema** in **Second Normal Form (2NF)**.

(3.5 marks)

c) Based on your answer in **(c)**, **remove the transitive dependencies**. **Produce all relation schemas** in **Third Normal Form (3NF)**.

(5 marks)

d) **Create the new database** (**the tables and its data** as in **Figure B.2**) based on your answer in **(d)**.

(6 marks)

Question 3: Consider the transaction schedule provided in **Table B.1 and B.2.** **(15 Marks)**

3(a)

Table B.1: Transaction Schedule for T1-T2-T3

↓ **time**	**T1**	**T2**	**T3**
	begin transaction		
	read_item(y)		
	read_item(z)	begin transaction	
	commit	read_item(w)	begin transaction
		read_item(v)	read_item(v)
		w=(v+w)*1.2	read_item(w)
		write_item(w)	v=v+150
		commit	write_item(v)
			commit

Based on **Table B.1**, answer **all** the following questions:

(i) Draw the precedence graph for the transaction schedule in **Table B.1** to check for serializability.

(2 Marks)

(ii) Explain the precedence graph produced in **(i)**.

(1.5 Marks)

(iii) Apply the locking technique to the transaction schedule in **Table B.1**, and revise the transaction schedule in **Table B.1** using the following format:

T1	**T2**	**T3**

(2.5 Marks)

(iv) Explain the potential problem(s) that may arise in **(iii)** after applying the locking technique..

(2 Marks)

3(b)

Table B.2: Transaction Schedule for T4-T5-T6

↓ **time**	**T4**	**T5**	**T6**
	begin transaction		
	read_item(a)		
	read_item(b)	begin transaction	
	a=(a*1.1)+b	read_item(b)	
	write_item(a)	read_item(c)	begin transaction
	commit	b=b+c	read_item(c)
		write_item(b)	c=c-100
		commit	write_item(c)
			commit

Based on the transaction schedule in **Table B.2**, apply the two-phase locking (2PL) protocol to emphasize the positioning of lock and unlock operations in each transaction using the following format.

T4	**T5**	**T6**

(7 Marks)

Question 4: Answer **all** questions based on the case study below. **(10 Marks)**

A healthcare organization is planning to implement a data warehouse to analyze patient treatment outcomes. The organization's data sources include electronic health records (EHRs), laboratory test results, and patient satisfaction surveys. The management wants to use this data to improve treatment effectiveness and optimize resource allocation.

Based on the given case study, answer **all** the following questions.

a) Explain **two (2)** characteristics of data in a data warehouse that would be beneficial for the healthcare organization to analyze patient treatment outcomes.

(3 Marks)

b) Based on the scenario given, suggest a **suitable** approach (Top-Down or Bottom-Up) for designing the data warehouse. **Justify** your recommendation.

(3 Marks)

c) Differentiate **two (2)** key features of the ETL (Extract, Transform, Load) process and the ELT (Extract, Load, Transform) process in the context of integrating data from the given sources.

(4 Marks)

SECTION C: CASE STUDY QUESTIONS (30 MARKS)

This section consists of **two (2) questions**. Answer **all** questions in the **Answer Booklet.**

Question 1 (17 Marks)

The database of a sports shop tracks **Products, Suppliers, Orders,** and **Stock Restocking.** This system manages inventory, customer orders, and restocking activities.

Below are the tables with sample data (refer to **Page 13** and **Page 14**).

Relation 1: **Products** (*primary key: ProductID)*

ProductID	ProductName	Category	Brand	Price	Stock	ReorderLevel
101	Football	Equipment	Nike	35.00	50	10
102	Running Shoes	Footwear	Adidas	65.00	20	5
103	Tennis Racket	Equipment	Wilson	130.00	10	3
104	Sports T-shirt	Apparel	Puma	30.00	55	20
105	Basketball	Equipment	Spalding	55.00	30	5
106	Yoga Mat	Equipment	Reebok	45.00	15	5
107	Swimming Cap	Accessories	Speedo	20.00	40	10

Relation 2: **Suppliers** (*primary key: SupplierID)*

SupplierID	SupplierName	ContactNumber	Address
201	Sports Supplies	123456789	45 Jalan Canggung
202	Global Sports	987654321	87 Jalan Siakap
203	Active Wear Co	555888999	12 Jalan Lampan
204	Game Masters	444777666	60 Jalan Kaloi
205	Speedo	222333444	19 Jalan Selayang

Relation 3: **Customers** (*primary key: CustomerID)*

CustomerID	CustomerName	Email	PhoneNumber	Address
301	Hilmi Pasha	hilmi@email.com	111222333	10 Jalan A
302	Nouredin Momdouh	nouredin@email.com	222333444	25 Jalan B
303	Rio Iwasaki	rio@email.com	333444555	50 Jalan C
304	Avidian Dipesh	avidian@email.com	444555666	77 Jalan D
305	Nazmi Saifulnizam	nazmi@email.com	555666777	88 Jalan E

Relation 4: **Order** (*primary key: OrderID;* f*oreign key: CustomerID references Customers (Customers.CustomerID))*

OrderID	CustomerID	OrderDate	TotalAmount
1	301	2024-11-01	150.00
2	302	2024-11-02	360.00
3	303	2024-11-03	25.00
4	304	2024-11-04	120.00
5	305	2024-11-05	50.00

Relation 5: **OrderDetails** (*primary key: combination of OrderID and ProductID;* f*oreign key: OrderID references* ***Orders*** *(Orders.OrderID), ProductID references* ***Products*** *(Products.ProductID))*

OrderID	ProductID	Quantity	Subtotal
1	101	5	150.00
2	102	6	360.00
3	101	2	60.00
4	105	1	50.00
5	102	3	180.00
6	101	4	120.00

Relation 6: **StockRestock** (*primary key: RestockID; foreign key: ProductID references* ***Products*** *(ProductID))*

RestockID	ProductID	RestockDate	Quantity	SupplierID
401	101	2024-11-01	20	201
402	102	2024-11-02	10	202
403	105	2024-11-03	15	203
404	104	2024-11-04	50	204
405	103	2024-11-05	5	205

Write / Construct **SQL statements** for **all** the questions below (refer the relations provided on **page 13** and **page 14**):

(a) Referring to Relation 1 (Product table), new data has been inserted for ProductID = 107. Insert a new product into the **Products** table based on the given below information:

ProductName: Swimming Cap
Category: Accessories
Brand: Speedo
Price: RM20.00
Stock: 40
Reorder Level: 10

(2 Marks)

(b) List the ProductName, Price, and Stock of **all Products** in the "Footwear" or "Accessories" Category with Stock **between** 15 and 50.

(3 Marks)

(c) Draw the table output based on the question **(b)**.

(1 Mark)

(d) Decrease the Price for **Products** in the "Footwear" Category by 10% if their Stock is **less than or equal to** 20.

(2 Marks)

(d) Display the **Products** columns of ProductID, ProductName, Category, and the latest updated price from **(c)** for the "Footwear" Category only.

(3 Marks)

(e) List the CustomerID, CustomerName, ProductID, ProductName, and the total number of **Orders** (labelled as OrderCount) for each **Customer** who has ordered the same **Products more than once**.

(5 Marks)

(f) Delete all records from **StockRestock** where the RestockDate is before '2024-11-03'.

(1 Mark)

Question 2 **(13 Marks)**

The KTDI Restaurant Reservation System allows **Customers** to reserve **Tables** by reference to their CustomerID and check the availability of **Tables** using TableID. The system also enables the assignment of **Staff** to **Reservations**, improving tracking and operational efficiency. The **Reservation** process includes verifying **Table** availability, updating the **Table**'s status to "reserved," and recording the **Reservation** in the database using **Transaction Control Language (TCL)** commands. All operations are executed within a transaction to ensure atomicity, meaning if any step fails, the transaction is rolled back to maintain consistency. If all steps are successful, the transaction is committed to finalize the reservation.

The relational schemas of the database system are listed below:

Relation 1: **Tables** - Stores available **Table** information in the restaurant.

Tables (TableID, Status)

Relation 2: **Staff** - Stores **Staff** details.

Staff (StaffID, Name, Role, Contact, Email)

Relation 3: **Customers** - Stores **Customer** details.

Customers (CustomerID, Name, Contact, Email)

Relation 4: **Reservations** - Stores **Reservation** details for **Customers** and **Tables** managed by **Staff**.

Reservations (ReservationID, Date, Time, TableID, CustomerID, StaffID)

Write / Construct **SQL** statements for **all** the questions below (refer to the relational schemas provided on **page 16**):

(a) Start a transaction to process the reservation.

(1 Mark)

(b) Check the **Table** status with TableID = 1 to verify the reservation availability. If the **Table** is available, the status will show "available" and commit changes. Otherwise, it will show "reserved" and roll back the transaction. (Note: Use COMMIT statement since the status is available.)

(2 Marks)

(c) Undo the changes made within the transaction based on the answer in **(b)** before it is committed.

(2 Marks)

(d) Since the status of TableID = 1 is "available", update the **Table's** status to "reserved". Then, insert the reservation record into **Reservations** as follows: **(3004, '2025-01-13', '19:00', 1, 1001, 2002)**. Once done, commit the transaction.

(3 Marks)

(e) The transaction starts checking the availability of a **Table** (TableID = 1). A savepoint, check_table_status, is set. If the **Table** status is "available", its status is updated to "reserved," and another savepoint, update_table_status, is set. A **Reservation** record is then inserted as follows: **(3005, '2025-01-25', '19:00', 1, 1001, 2002)** into the **Reservations**, with a savepoint, insert_reservation, after the insert. Assume a duplicate **Reservation** record is attempted to insert to **Reservations**, and then the transaction rolls back to the insert_reservation savepoint. Finally, if all steps are successful, the transaction is committed to make the changes permanent.

(5 Marks)

END OF QUESTION