National University of Computer and Emerging Sciences

FAST School of Computing **Spring-2024** **Islamabad Campus**

CS-2005: Database Systems

Sessional-II Exam

SE A-B-C-D-E-F-G

Total Time: 1 Hour

Total Marks: 66

Date: 3rd April, 2024

Course Instructor(s)

Mr. Bilal, Dr. Asif, Mr. Zeeshan

Student Name	Roll No.	Course Section	Student Signature

Do not write anything on the question paper except the information required above.

Instructions:

1. Read the question carefully, understand the question, and then attempt your answers in the provided answer booklet.
2. Verify that you have **three (3)** printed pages of the question paper including this page. There are **three (3)** questions.
3. Calculator sharing is prohibited.
4. **Return the Question Paper along with answer sheet.**
5. **You can attempt questions in any order but attempt complete question together with all parts. (else -6)**

Q1: **[7 x 3 =21 marks]**

You are working with an e-commerce database containing tables for:

Customers: (customer_id, name, email, city)
Orders: (order_id, customer_id, order_date, total_amount)
Products: (product_id, name, price, category, stock)
Order_Items: (order_item_id, order_id, product_id, quantity)

Provide SQL queries to provide following details from the above database.

A) Find all product IDs from orders where the product price is higher than the average price in that product's category.

```
SELECT p.product_id
FROM Products p
INNER JOIN Order_Items oi ON p.product_id = oi.product_id
WHERE p.price > (
  SELECT AVG(price)
  FROM Products p1
  WHERE p1.category = p.category
);
```

B) List all orders with a total amount exceeding the highest total amount from the same customer's previous orders.

```
SELECT o.order_id, o.total_amount
FROM Orders o
WHERE o.total_amount > (
SELECT MAX (total_amount)
FROM Orders pre
WHERE pre.customer_id=o.customer_id
AND pre.order_id <> o.order_id
);
```

C) Find the top customer (by total order amount) within each city for the last 6 months.

```
SELECT c.city, cu.name, SUM(oi.quantity * p.price) AS total_spent
FROM Customers c
INNER JOIN Orders o ON c.customer_id = o.customer_id
INNER JOIN Order_Items oi ON o.order_id = oi.order_id
INNER JOIN Products p ON oi.product_id = p.product_id
GROUP BY c.city, cu.name
ORDER BY c.city, total_spent DESC;
```

D) List the product not ordered.

```
SELECT p.name AS product_name
FROM Products p
LEFT JOIN Order_Items oi ON p.product_id = oi.product_id
WHERE oi.product_id IS NULL;
```

E) List all product names that have not been sold in the last 3 months but have a stock level exceeding ten units.

```
SELECT p.name
FROM Products p
WHERE p.product_id NOT IN (
SELECT product_id

FROM Order_Items oi
INNER JOIN Orders o ON oi.order_id = o.order_id
WHERE o.order_date &gt;= DATEADD(month, -3, GETDATE())
)
AND p.stock &gt; 10;
```

F) Find the total sales amount for each product category.

```
SELECT p.category, SUM(oi.quantity * p.price) AS total_sales
FROM Products p
INNER JOIN Order_Items oi ON p.product_id = oi.product_id
INNER JOIN Orders o ON oi.order_id = o.order_id
GROUP BY p.category;
```

G) Calculate the average order value (total amount divided by number of orders) for each customer.

```
SELECT c.name, AVG(o.total_amount) AS avg_order_value
FROM Customers c
INNER JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.name;
```

Q2: Part A **[6 + 14 = 20 marks]**

OrderID	ProductID	ProductName	CustomerID	CustomerName	OrderDate	Quantity	TotalPrice
1001	101	Laptop	C100	Ahmed	01/03/2024	1	$1000
1002	102	Smartphone	C201	Fahad	02/03/2024	2	$1200
1003	103	Tablet	C100	Ahmed	03/03/2024	1	$500
1004	101	Laptop	C313	Waqar	04/03/2024	1	$1100

The data in the table is susceptible to anomalies. Provide examples of how insertion, deletion, and modification anomalies could occur in this table.

The answer requires one example each of the following by keeping in mind the above table.

- **Insertion anomaly**

  If a new customer places an order for a product that hasn't been ordered by any other customer yet, a new row would need to be inserted into the table. However, if this customer hasn't purchased anything else, their information would only exist with this one product. This leads to redundancy and inefficiency.

- **Modification anomaly**

  If a customer changes their name or if there's a mistake in entering their name, modifying it in one row would necessitate updating all rows with the same customer information. Failure to do so would result in inconsistencies.

- **Deletion anomaly**

  If a customer who has only made one purchase decides to cancel their order or return the product, deleting their row from the table would also result in the loss of their information from the database entirely. This is problematic if we want to maintain customer information regardless of whether they've made a purchase.

Part B

Examine the given relation R shown below and answer the following questions.

Book_title	Author_name	Book_type	List_price	Author_affil	Publisher
The Great Gatsby	Fitzgerald	Fiction	15.99	Princeton University	Scribner
To Kill a Mockingbird	Harper	Fiction	12.50	University of Alabama	Harper Collins
1984	George	Fiction	10.99	Independent	Penguin
Pride and Prejudice	Jane	Fiction	9.99	Independent	Penguin

Book_title	Author_name	Book_type	List_price	Author_affil	Publisher
The Catcher in the Rye	Salinger	Fiction	11.25	Columbia University	Little Brown

This data represents a simplified example of a BOOK relation with various books, their authors, types, list prices, author affiliations, and publishers. Each book has a unique title, and authors may have affiliations with universities or be independent. Author affiliation and publisher are not linked or dependent on each other. Publishers are also included for each book. The Book title is the primary key for the above table.

A. Is the given relation R in 1NF? If not, explain why. (1)

It appears that all attributes in the relation R contain atomic values. There are no repeating groups or multi-valued attributes. Therefore, the given relation R is indeed in the first normal form (1NF).

B. If the answer to part (a) is NO, convert the relation R into 1NF. If the relation is already in 1NF then write "Same as above" as answer. (2)

Same as above

C. List all the functional dependencies that hold in the 1st normalized form of relation R (2)

The functional dependencies that hold in the given relation R:

1. Book_title → Author_name
   - Each book title uniquely determines its author's name.

2. Book_title → Book_type
   - Each book title uniquely determines its type or genre.

3. Book_title → List_price
   - Each book title uniquely determines its list price.

4. Book_title → Author_affil
   - Each book title uniquely determines the author's affiliation.

5. Book_title → Publisher
   - Each book title uniquely determines its publisher.

These functional dependencies are inferred based on the understanding that each book title uniquely identifies its attributes, such as author name, book type, list price, author affiliation, and publisher.

D. Are there any partial functional dependencies listed in part c? If yes, write them in the area given below, if answer is no, then mention why? (2)

In the provided functional dependencies, there are no partial functional dependencies. Each functional dependency involves the entire candidate key, which is Book_title. Therefore, there are no partial functional dependencies listed in part C.

E. Convert the 1st normalized form of relation R to 2NF. Only write the final schema. If the schema is already in 2NF, just write "Same as above." (2)

To convert the first normalized form (1NF) of relation R to the second normal form (2NF), we need to ensure that there are no partial dependencies. Since there were no partial dependencies in the 1NF schema, the schema is already in 2NF. Therefore, the final schema remains the same as above.

F. Are the above 2NF relations in 3NF? If not explain why? (2)

To determine if the 2NF relations are in the third normal form (3NF), we need to ensure that there are no transitive dependencies. A transitive dependency occurs when a non-prime attribute depends on another non-prime attribute.

Let's revisit the functional dependencies from the 2NF relations:

1. Book_title → Author_name
2. Book_title → Book_type
3. Book_title → List_price
4. Book_title → Author_affil
5. Book_title → Publisher

From these dependencies, we can observe that Book_title is the candidate key and there are no transitive dependencies present. Each non-prime attribute (Author_name, Book_type, List_price, Author_affil, Publisher) depends directly on the candidate key, Book_title.

Therefore, the above 2NF relations are also in 3NF since there are no transitive dependencies.

G. If the answer to the above question is no, convert the tables into 3NF. Only write the updated schema. If your answer is no, just write "Same as above". (2)

Same as above.

H. Are the above 3NF relations in BCNF? Explain why in both cases. (1)

The above 3NF relation is not in BCNF due to the modification anomaly which may occur due to book type. To resolve this issue and ensure BCNF, we would decompose the relation further to separate the `Book_type` attribute. If we introduce a separate table for `Book_type` and use it as a foreign key in Relation 1, we can still maintain BCNF and resolve the modification anomaly. Here's how we can structure it:

- Relation 1:

- Attributes: Book_title, Author_name, Book_type (FK), List_price, Author_affil, Publisher

- Candidate Key: Book_title

- Functional Dependencies: Book_title → Author_name, Book_type, List_price, Author_affil, Publisher

New table:

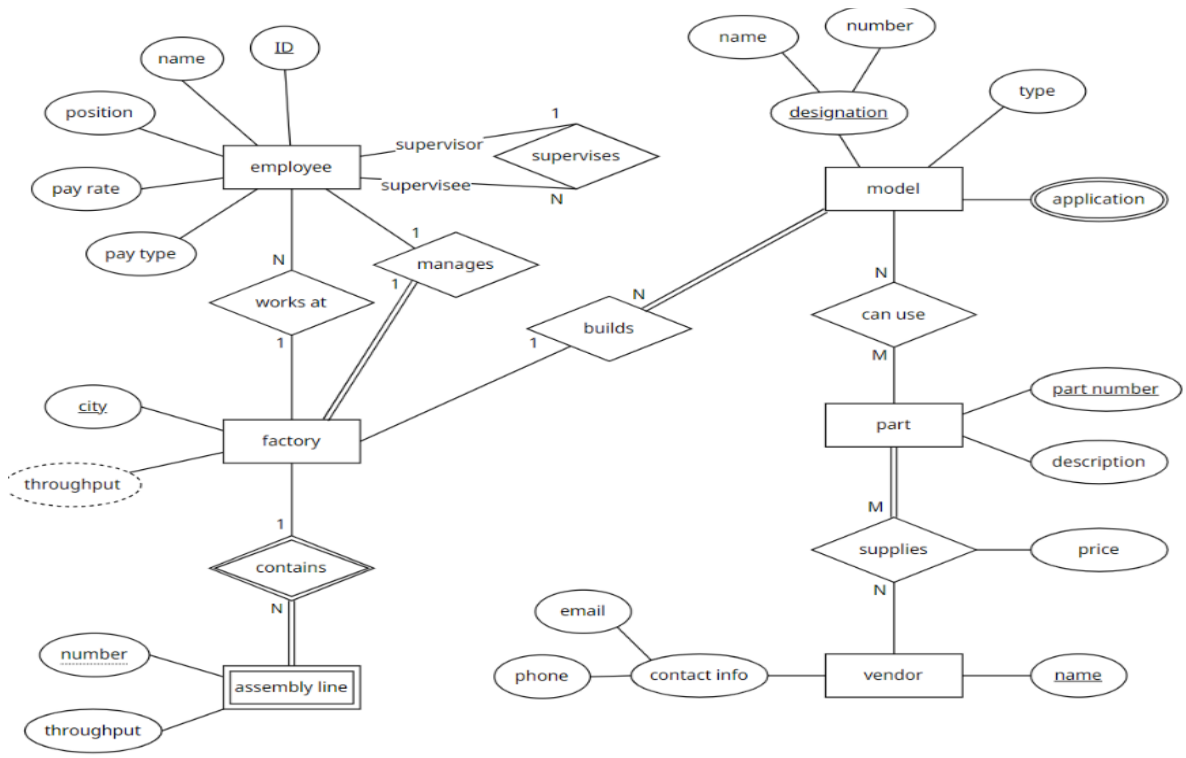
- Relation 2: Book_type

- Attributes: Book_type (PK)

- Candidate Key: Book_type

Q3: **[12+13 = 25 marks]**

Part A. Production tracking is important in many manufacturing environments (e.g., the pharmaceuticals industry, children's toys, etc.). The following ERD captures important information in the tracking of production. Specifically, the ER diagram captures relationships between production lots (or batches), individual production units, and raw materials. Please convert the ER diagram into a relational database schema (logical schema). Be certain to indicate all attributes, primary key and foreign key constraints.



All highlighted portion contains marks

Employee (ID (pk) , name, position, payrate, pay type, supervisor (FK), factory city (fk)) 2 marks

Factory (city (pk) or fid as pk, throughput, managerID (fk) 1 mark

Assembly Line (number (pk), throughput, factory (fk) 1 mark

Model (name (pk), number (pk), number, name, type, factory city(fk 2 mark

Application (ID (pk), name (fk), number (fk),) 1 mark

Model_part(partno (fK) , name (fk), number (fk),) // both make composite primary key 2 marks (1 for mapping relations and one for correct pk and fk)

Part (partno (pk), description

Vendor (name (pk), email, phone no) 1 mark

Supplier (partno (fk), (name (fk), price) // both make composite primary key 2 marks (1 for mapping relations and one for correct pk and fk)

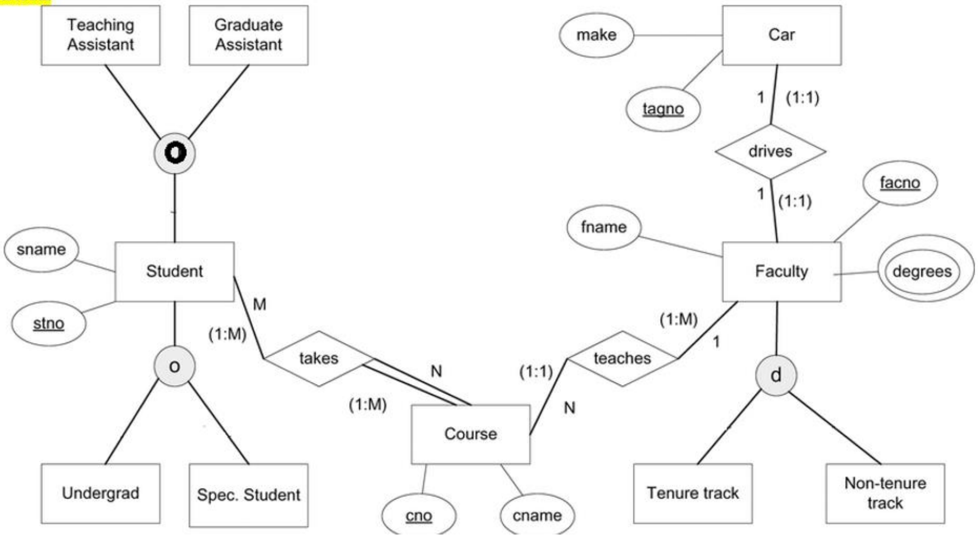
1 marks for all other tables and pk

13 marks. Incorrect - 3

Part B: Attempt the following part b on the question paper.

Complete the following enhanced ERD for a university management system by adding completeness and disjoint constraints. Fill in information in all boxes marked by →
The following EERD depicts the university management system. The university has three types of faculty members: Adjunct faculty, Tenure Track faculty (contract-based), and non-tenure track. Faculty members can choose only one of the above-mentioned roles. Faculty car data is recorded and stored in the database for security reasons. Faculty can teach courses to the students. The university has both undergraduate students and graduate students. Other than these students, there are some students who are not regular university students but are there to take some optional courses. During the semesters, students can also opt to become either teaching assistants for undergraduate students or graduate students. Students can take both roles in the same semester.

Solution



2 for each correct