ITM
IT MUSTAKBIL

Final Examination Evening

Full Name:

Roll No: **Section:** Circle: **A / B / C / D**

Database Systems **2 Hour 30 Minutes**

Theory & Practical Semester 2

Paper 1 Spring (2025)

Instructions:

- You must answer **all** questions. There are **no optional** questions.
- You are responsible for ensuring your answers are clear and unambiguous.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- You must give your answers in the spaces provided on the question paper.
- Write your full name, roll number and section in the boxes at the top of the page.
- **You may not use any online / offline notes, handouts, notes or resources other than those provided in this exam.**
- This exam affords **zero tolerance** to students found using dishonest or unfair means.

Information:

- The total marks for this paper are 70.
- The number of marks for each question or part question are shown in brackets [ ].
- Students may only be awarded whole number marks, with no partial marks.
- There are a total of 11 pages in this paper.
- There are **three sections** in this paper; A) Multiple Choice Questions, B) Short Questions, C) Case Study Questions.

Total Marks
/ 70

systems

Section A

Multiple Choice Questions

Examiner Use

[10]

1. **What is the purpose of the `GROUP BY` clause in SQL?**
   A) Sort data
   B) Group rows to apply aggregate functions
   C) Filter columns
   D) Remove duplicates

2. **Which of the following is a valid** set operation **in SQL Server?**
   A) JOIN
   B) UNION
   C) SELECT*
   D) CONNECT

3. **What does the following query return?**

```
SELECT COUNT(DISTINCT DepartmentID)
FROM HumanResources.EmployeeDepartmentHistory;
```

   A) Number of departments
   B) Count of employees
   C) Count of department changes
   D) Count of department heads

4. **Which command is used to grant access on a table to a user?**
   A) ALLOW
   B) GRANT
   C) PERMIT
   D) ENABLE

5. **A view in SQL Server can be used to:**
   A) Store permanent data
   B) Index data automatically
   C) Simplify complex queries
   D) Encrypt column data

Examiner Use

6. **Which clause is used in a correlated subquery?**
   A) ALL
   B) HAVING
   C) EXISTS
   D) DISTINCT

7. **Which of the following eliminates duplicates from a result set?**
   A) REMOVE
   B) DELETE
   C) DISTINCT
   D) HAVING

8. **How do you create a temporary table in SQL Server?**
   A) CREATE TABLE TEMP
   B) CREATE TABLE #TempTable
   C) MAKE TEMP TABLE
   D) CREATE TEMP TABLE

9. **A stored procedure in SQL Server:**
   A) Can only perform SELECT
   B) Is used for DDL only
   C) Automates repetitive tasks
   D) Cannot have parameters

10. **Why are indexes useful in SQL Server?**
    A) Compress tables
    B) Speed up data retrieval
    C) Help export data
    D) Avoid constraints

Section B

Short Questions

Note: **Write short answers and include relevant queries using AdventureWorks schema.**

Schema Attached at the end.

Each Question Carries 3 marks. ***Total Marks: [24]***

1- Highlight the difference between Normalization and Denormalization. Using an example, tell which one is better?

2- How are Views, Temporary Table and Indexing different from each other? Mention at least 2 differences.

3- Read the Information on the table and highlight the following fields,
a) Primary Key (b) Foreign Key (c) Candidate Key
d) Super Key (e) Surrogate Key (f) Unique Key

Table: `Students`

Field Name	Data Type	Description
StudentID	INT	System-generated identifier
RollNumber	VARCHAR(10)	Unique roll number assigned by the institution
CNIC	VARCHAR(15)	National ID number (must be unique)
FirstName	VARCHAR(50)	Student's first name
LastName	VARCHAR(50)	Student's last name
Email	VARCHAR(100)	Must be unique for communication
DepartmentID	INT	References the Department table (foreign key)

Table: `Departments`

Field Name	Data Type	Description
DepartmentID	INT	Unique department ID
DepartmentName	VARCHAR(100)	Name of the department

4- Count the number of employees in each department using GROUP BY.

5- Display the first and last names of employees along with their department names. Use proper JOIN statements between related tables.

6- Create a view that shows the SalesOrderID, CustomerID, and TotalDue for orders where the total is greater than 10,000.

7- List employees who have more VacationHours than the average VacationHours of all employees. Use only a subquery in the WHERE clause — no joins.

8- Create a stored procedure that takes a job title as input and returns a list of employees(Employee Name as well) who have that job title.

Section C

Case Study Questions

Note: After carefully reading the scenario, answer the questions that follow and implement SQL queries.

Examiner Use

[36]

AdventureWorks Sales & HR Integration

You are managing reporting for sales and HR departments. Use these tables:

1. *.Employee*
2. *Department*
3. *EmployeeDepartmentHistory*
4. *Customer*
5. *SalesOrderHeader*
6. *SalesOrderDetail*
7. *Product*
8. *Person*

Case Study: Integrated Sales and HR Reporting System Your task is to write optimized SQL queries and demonstrate advanced SQL concepts based on the AdventureWorks database.

<u>Part 1: Sales & HR Reporting Queries(2 x 1 =2 marks)</u>

1. Show each customer's total sales
only if they have placed more than
3 orders. Sort the result by total
sales in descending order.

2. List all customers who have placed more than 5 orders.
(Use GROUP BY and HAVING.)

3. Create a temporary table called #TopSellingProducts with ProductID, OrderQty, and LineTotal from SalesOrderDetail. Only include products where the quantity sold is more than 100.

4. Create a view named EmployeeDepartments that shows each employee’s ID, full name (first + last), department name, and the start date of their assignment.

5. Use a nested query to find employees who are in the same department as someone with the job title 'Engineer'.

6. Write a stored procedure usp_CustomerOrders that accepts a customer ID and returns that customer’s order details from SalesOrderHeader.

Part 2: Advanced Nested Joins & Subqueries (3 x 1 = 3 marks)

7. List employees who are either Manager or have more than 100 vacation hours (Use Set Operation)

8. Show Product Name and Quantity for the Top 3 Highest Quantity Order Lines

9. List CustomerIDs where their TotalDue in any order is greater than the average TotalDue of all orders (Use Nested Query)

10. Show CustomerIDs who placed at least one order above 15,000(Total Dues) but never below 5,000 (Use Nested Query)

11. List all customers who placed an order above 5000 and show their full name and order details.

Part 3: Roles, Security, Privileges (3 x 1 = 3 marks)

12. What SQL statement is used to give a user read-only access to the SalesOrderHeader table? Write syntax to create user ReadOnlyUser, then grant permission.

13. How would you create a role SalesAnalyst, add a user to it, and allow only SELECT on Sales.SalesOrderDetail?

Mention commands to create role, assign privileges, and add member.(Kindly create Login, User and role for first one)

14. Suppose you want to restrict UPDATE and DELETE access on the Employee table but allow SELECT. What SQL GRANT/DENY statements will you use?

SCHEMA:

Table Name	Column	Data Type	PK	FK	Description
Person	PersonID	INT	✅		Unique identifier for a person
	FirstName	NVARCHAR(50)			Person's first name
	LastName	NVARCHAR(50)			Person's last name
Department	DepartmentID	INT	✅		Unique ID for department
	Name	NVARCHAR(50)			Name of the department
Employee	EmployeeID	INT	✅		Unique ID for employee
	PersonID	INT		✅	FK to Person(PersonID)
	JobTitle	NVARCHAR(50)			Employee's job title
	VacationHours	INT			Allocated vacation hours
	Salary	DECIMAL(10,2)			Salary of the employee
EmployeeDepartmentHistory	EmployeeID	INT		✅	FK to Employee(EmployeeID)
	DepartmentID	INT		✅	FK to Department(DepartmentID)
	StartDate	DATE			Date of joining department
Customer	CustomerID	INT	✅		Unique ID for customer
	PersonID	INT		✅	FK to Person(PersonID)
Product	ProductID	INT	✅		Unique ID for product
	Name	NVARCHAR(50)			Product name
	ListPrice	DECIMAL(10,2)			Selling price of the product
SalesOrderHeader	SalesOrderID	INT	✅		Unique ID for sales order
	CustomerID	INT		✅	FK to Customer(CustomerID)
	OrderDate	DATE			Date of order
	TotalDue	DECIMAL(10,2)			Total amount due
SalesOrderDetail	SalesOrderDetailID	INT	✅		Unique ID for order detail
	SalesOrderID	INT		✅	FK to SalesOrderHeader(SalesOrderID)
	ProductID	INT		✅	FK to Product(ProductID)
	OrderQty	INT			Quantity ordered
	LineTotal	DECIMAL(10,2)			Total for this line

systems