CSS325: Database Systems

Final Mock Exam

curated by The Peanuts

Name.....ID.....Section.....Seat No.....

Conditions: Open Book

Directions:

1. This exam has 20 pages (including this page).
2. Write your name clearly at the top.
3. When you finish, please `COMMIT`. If you would like to undo the entire exam, unfortunately `ROLLBACK` is not supported.
4. If your final answer appears without steps, you may receive partial credit.
5. If you get stuck, remember: `DELETE FROM Stress WHERE Subject = 'You';` Unfortunately, this command has no effect.
6. Do not communicate with others. If you whisper to your friend, they may whisper back: `"ERROR 1142 (42000): SELECT command denied to user 'you'@"`.

For solution, ***click here****.*

Question 1

Consider the following relation schema for a university course enrollment system:

ENROLLMENT(StudentID, StudentName, StudentEmail, CourseCode, CourseName,
InstructorName, InstructorEmail, Semester, Year, Grade, CreditHours)

Given the following assumptions:

- Each student has a unique StudentID and email
- Each course has a unique CourseCode
- A course name uniquely identifies a course
- Each instructor has a unique email
- A course is taught by only one instructor per semester
- The same course can be offered in different semesters
- A student can take the same course only once
- InstructorEmail determines InstructorName
- CourseCode determines CourseName and CreditHours

(a) Identify **all superkeys** for this relation.

(b) Identify **all candidate keys**.

(c) Select an appropriate **primary key** and explain your choice.

Question 2

For the ENROLLMENT relation above, draw a complete **dependency diagram** showing:

- All functional dependencies
- Clearly mark partial dependencies (if any)
- Clearly mark transitive dependencies (if any)
- Clearly mark full dependencies

Label each dependency appropriately.

Question 3

Given the following relation **R(A, B, C, D, E)**:

A	**B**	**C**	**D**	**E**
1	2	5	7	9
1	3	5	7	10
2	4	6	8	11
2	4	6	9	12
3	5	7	10	13

Check whether the following functional dependencies hold in this relation. For each FD, state whether it holds or not and briefly explain why.

(a) $A \rightarrow B$

(b) $A \rightarrow C$

(c) $B \rightarrow D$

(d) $AB \rightarrow DE$

(e) $C \rightarrow D$

(f) $D \rightarrow C$

(g) $BD \rightarrow A$

Question 4

You are given the following unnormalized relation for a hospital system:

PATIENT_VISIT(PatientID, PatientName, PatientPhone, DoctorID, DoctorName, DoctorSpecialization, VisitDate, Diagnosis, PrescribedMedicines, MedicineDosage, WardNumber, WardType, NurseID, NurseName)

Assumptions:

- A patient can visit multiple doctors on different dates
- PatientID uniquely identifies a patient
- PatientID determines PatientName and PatientPhone
- DoctorID uniquely identifies a doctor
- DoctorID determines DoctorName and DoctorSpecialization
- A visit is identified by PatientID, DoctorID, and VisitDate
- During a visit, a patient can be prescribed multiple medicines
- PrescribedMedicines (medicine name) determines MedicineDosage
- A patient may be admitted to a ward after a visit
- WardNumber determines WardType
- WardNumber determines NurseID and NurseName
- NurseID determines NurseName

Sample Data:

PID	PName	PPhone	DID	DName	DSpec	VDate	Diag	Medicines	Dosage
P001	John Doe	081-xxx	D01	Dr. Smith	Cardiology	2025-01-10	Heart Issue	Aspirin	100mg
P001	John Doe	081-xxx	D01	Dr. Smith	Cardiology	2025-01-10	Heart Issue	Lisinopril	10mg
P001	John Doe	081-xxx	D02	Dr. Brown	Neurology	2025-01-15	Migraine	Ibuprofen	400mg
P002	Jane Smith	082-xxx	D01	Dr. Smith	Cardiology	2025-01-12	Check-up	NULL	NULL

WardNumber	**WardType**	**NurseID**	**NurseName**
W101	ICU	N001	Alice Johnson
W101	ICU	N001	Alice Johnson
NULL	NULL	NULL	NULL
W102	General	N002	Bob Williams

(a) Identify the primary key for the unnormalized relation and explain your reasoning.

(b) List ALL functional dependencies present in this relation.

(c) Is this relation in 1NF? Explain why or why not. If not, convert it to 1NF.

(d) Convert your 1NF relation(s) to 2NF. Show all steps clearly.

(e) Convert your 2NF relation(s) to 3NF. Show all steps clearly.

Question 5

Based on your 3NF design from Question 4, write complete SQL DDL statements to create the normalized database schema.

Your `CREATE TABLE` statements must include:

- Appropriate data types for each attribute
- All primary key constraints
- All foreign key constraints with proper referencing
- NOT NULL constraints where appropriate
- Any CHECK constraints that would be useful (you may add reasonable business rules)

Notes:

- PatientPhone should support international format
- Patient names and doctor names must support international characters (e.g., Thai, Chinese, Japanese, Arabic).
- Medicine dosage should be stored as text (e.g., "100mg", "10ml")
- Visit dates should be DATE type
- Consider using appropriate VARCHAR lengths
- Assume NurseID format: N001, N002, etc. (CHAR(4))
- Assume DoctorID format: D01, D02, etc. (CHAR(3))
- Assume PatientID format: P001, P002, etc. (CHAR(4))
- Assume WardNumber format: W101, W102, etc. (CHAR(4))

Question 6

For the database you created in Question 5:

(a) Write SQL INSERT statements to populate your tables with at least 2 meaningful records per table. Ensure your data demonstrates:

- At least one patient with multiple visits
- At least one visit with multiple prescribed medicines
- At least one ward with an assigned nurse
- Proper foreign key relationships

(b) Write an UPDATE statement that changes Dr. Smith's specialization from "Cardiology" to "Cardiothoracic Surgery".

(c) Write a DELETE statement to remove all visits that occurred before January 1, 2025. Explain what happens to related records and why.

For Questions 7-11, use the following university database schema:

```
STUDENT(StudentID, Name, Email, DateOfBirth, Major, GPA)
COURSE(CourseCode, CourseName, Credits, Department)
INSTRUCTOR(InstructorID, Name, Email, Department, Salary
    )
ENROLLMENT(StudentID, CourseCode, Semester, Year, Grade)
TEACHES(InstructorID, CourseCode, Semester, Year)
```

Question 7

Write a query to find all students whose names start with 'A' or end with 'son', ordered by their GPA in descending order. Display their StudentID, Name, and GPA.

Question 8

Write a query to calculate each student's age (in years) as of today, and display students who are between 20 and 25 years old. Show StudentID, Name, and Age. Order by age descending.

Question 9

Write a query to display the course code, course name, and the number of students enrolled in each course for the Fall 2024 semester. Only show courses with more than 30 enrolled students. Order by enrollment count descending.

Question 10

Write a query to find instructors whose salary is higher than the average salary of all instructors in their department. Display InstructorID, Name, Department, and Salary.

Question 11

Write a query that displays each department and the average GPA of students majoring in that department. Only show departments where the average GPA is greater than 3.5. Round the average to 2 decimal places. Order by average GPA descending.

Continue using the university database schema

Question 12

Write a query using INNER JOIN to display:

- Student name
- Course name
- Instructor name
- Grade

For all enrollments in Spring 2025. Order results by student name, then course name.

Question 13

Write a query to find all courses that have NEVER been taken by any student (no enrollments). Use a subquery in your solution. Display CourseCode and CourseName.

Question 14

Write a query to create a view called `StudentCourseLoad` that shows:

- StudentID
- Student Name
- Total number of courses enrolled (for any semester/year)
- Total credits enrolled
- Average grade (as GPA points: A=4.0, B+=3.5, B=3.0, C+=2.5, C=2.0, D+=1.5, D=1.0, F=0)

Then, write a query using this view to find students who have enrolled in more than 10 courses with an average grade above 3.0.

Question 15

Write a query using a correlated subquery to find all students whose GPA is higher than the average GPA of students in their own major. Display StudentID, Name, Major, and GPA. Order by Major, then GPA descending.

Question 16

Write a query that shows, for each department:

- Department name
- Number of instructors
- Average instructor salary (rounded to 2 decimals)
- Number of courses offered by the department
- Total number of student enrollments across all courses in that department (for any semester/year)

Only include departments that have at least 3 instructors and at least 100 total enrollments. Order by total enrollments descending.

Hint: You will need to use multiple joins and subqueries.

Question 17

Write a stored procedure called `EnrollStudent` that:

- Takes parameters: StudentID, CourseCode, Semester, Year
- Checks if the student is already enrolled in the course for that semester/year
- If not enrolled, inserts the enrollment record with Grade = NULL
- Returns a message indicating success or failure

Question 18

Write a trigger called `UpdateGPAOnGradeChange` that:

- Fires AFTER an UPDATE on the ENROLLMENT table
- Automatically recalculates the student's GPA when their grade changes
- Updates the GPA in the STUDENT table

Assume grade points: A=4.0, B+=3.5, B=3.0, C+=2.5, C=2.0, D+=1.5, D=1.0, F=0.0

Question 19

Consider the following concurrent execution of two transactions on a banking system with accounts A and B:

Time	Transaction T1	Transaction T2
t1	BEGIN TRANSACTION	
t2	READ(Balance_A) // 1000	
t3		BEGIN TRANSACTION
t4		READ(Balance_A) // 1000
t5	Balance_A = Balance_A - 200	
t6	WRITE(Balance_A) // 800	
t7		Balance_A = Balance_A + 500
t8		WRITE(Balance_A) // 1500
t9	READ(Balance_B) // 2000	
t10		COMMIT TRANSACTION
t11	Balance_B = Balance_B + 200	
t12	WRITE(Balance_B) // 2200	
t13	COMMIT TRANSACTION	

(a) What is the final balance of Account A? Is this correct? Explain what concurrency problem occurred.

(b) What is the final balance of Account B?

(c) What ACID property is violated in this scenario? Explain.

(d) Name the specific transaction anomaly that occurred in this execution.

(e) Suggest a solution using isolation levels to prevent this problem. Which isolation level would you recommend and why?

Question 20

Given the following two transactions operating on a table **INVENTORY(ProductID, ProductName, Quantity)**:

Time	Transaction T1	Transaction T2
t1	BEGIN TRANSACTION	
t2	SELECT COUNT(*) FROM INVENTORY	
t3	WHERE Quantity < 10	
t4	Result: 5 rows	
t5		BEGIN TRANSACTION
t6		INSERT INTO INVENTORY
t7		VALUES ('P100', 'Widget', 5)
t8		COMMIT TRANSACTION
t9	SELECT COUNT(*) FROM INVENTORY	
t10	WHERE Quantity < 10	
t11	Result: 6 rows	
t12	COMMIT TRANSACTION	

(a) What transaction anomaly is demonstrated in this example? Explain in detail.

(b) Is this anomaly possible under READ COMMITTED isolation level? Explain.

(c) Is this anomaly possible under REPEATABLE READ isolation level? Explain.

(d) What is the minimum isolation level needed to prevent this anomaly?

(e) Explain one potential drawback of using that isolation level in a high-traffic e-commerce database.

Question 21 (Optional)

A developer working on an online shopping system made a critical mistake. They wrote an UPDATE statement to apply a 10% discount to all products in the "Electronics" category, but forgot to include the WHERE clause:

```
UPDATE Products
SET Price = Price * 0.9;
-- Missing: WHERE Category = 'Electronics'
```

After executing this command, they realized all 50,000 products in the database (across all categories) now have 10% reduced prices.

(a) As a Database Administrator (DBA), explain the recovery process you would use to restore the correct prices. What database feature would you use?

(b) Explain how you could have prevented this issue using database transactions. Write the corrected code using BEGIN, COMMIT, and ROLLBACK.

Question 22 (Optional)

Explain the difference between a Virtual View and a Materialized View:

(a) What are the key differences in how they are stored and updated?

(b) Give a real-world scenario where you would choose a Virtual View over a Materialized View.

(c) Give a real-world scenario where you would choose a Materialized View over a Virtual View.