Class: TE (2019) Div-A

Que No.	Question
	Assignment 1 (Unit 1)
Q1.	Consider the foreign key constraint from the dept name attribute ofinstructor to the department relation. Give examples of inserts and deletes to these relations, which can cause a violation of the foreign key constraint.
Q2.	What is the result of first performing the cross product of student and advisor, and then performing a selection operation on the result with the predicate s_id = ID? (Using the symbolic notation of relational algebra, this query can be written as σ(s_id=ID)(student × advisor).
Q3.	Consider the relational database given. Give an expression in the employee (person name, street, city) works (person name, company name, salary) company (company name, city) relational algebra to express each of the following queries: a. Find the names of all employees who live in city “Miami”. b. Find the names of all employees whose salary is greater than $100,000. c. Find the names of all employees who live in “Miami” and whose salary is greater than $100,000.
Q4.	Consider the bank database. Give an expression in the relational algebra for each of the following queries. a. Find the names of all branches located in “Chicago”. b. Find the names of all borrowers who have a loan in branch “Down-town”. branch(branch name, branch city, assets) customer (customer name, customer street, customer city) loan (loan number, branch name, amount) borrower (customer name, loan number) account (account number, branch name, balance) depositor (customer name, account number)
Q5.	Consider the relational database. Give an expression in the employee (person name, street, city) works (person name, company name, salary) company (company name, city) relational algebra to express each of the following queries: a. Find the names of all employees who work for “First Bank Corpora-tion”. b. Find the names and cities of residence of all employees who work for “First Bank Corporation”. c. Find the names, street address, and cities of residence of all employees who work for “First Bank Corporation” and earn more than $10,000.

Que No.	Question
Q6.	Consider the bank database. Give an expression in the relational algebra for each of the following queries: branch(branch name, branch city, assets) customer (customer name, customer street, customer city) loan (loan number, branch name, amount) borrower (customer name, loan number) account (account number, branch name, balance) depositor (customer name, account number) a. Find all loan numbers with a loan value greater than $10,000. b. Find the names of all depositors who have an account with a value greater than $6,000. c. Find the names of all depositors who have an account with a value greater than $6,000 at the “Uptown” branch.
Q7.	How would you apply the principles of normalization to eliminate redundancy in a database table?
Q8.	How would you implement the concept of primary and foreign keys to establish relationships between different tables in a database?
Q9.	Can you demonstrate how to write a SQL query to retrieve data from multiple related tables using JOINs?
Q10.	How would you create an ER diagram to represent the relationships between students, courses, and instructors in a university database, and ensure that all cardinalities and constraints are accurately depicted?
Assignment 1 (Unit 2)	
Q1.	Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.
Q2.	Consider a database used to record the marks that students get in different exams of different course offerings (sections). a. Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the database. b. Construct an alternative E-R diagram that uses only a binary relationship between student and section. Make sure that only one relationship exists between a particular student and section pair, yet you can represent the marks that a student gets in different exams.

Que No.	Question
Q3.	Consider the following set F of functional dependencies on the relation schema r(A, B, C, D, E, F): A → BCD BC → DE B → D D → A a. Compute B+. b. Prove (using Armstrong's axioms) that AF is a superkey. c. Compute a canonical cover for the above set of functional dependencies F; give each step of your derivation with an explanation. d. Give a 3NF decomposition of r based on the canonical cover. e. Give a BCNF decomposition of r using the original set of functional dependencies. f. Can you get the same BCNF decomposition of r as above, using the canonical cover?
Q4.	Compute the closure of the following set F of functional dependencies for relation schema r (A, B, C, D, E). A → BC CD → E B → D E → A
Q5.	An E-R diagram can be viewed as a graph. What do the following mean in terms of the structure of an enterprise schema? a. The graph is disconnected. b. The graph has a cycle.
Q6.	A weak entity set can always be made into a strong entity set by adding to its attributes the primary-key attributes of its identifying entity set. Outline what sort of redundancy will result if we do so.
Q7.	We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?
Q8.	Suppose the advisor relationship were one-to-one. What extra constraints are required on the relation advisorto ensure that the one-to-one cardinality constraint is enforced?
Q9.	Consider a many-to-one relationship R between entity sets A and B. Suppose the relation created from R is combined with the relation created from A. In SQL, attributes participating in a foreign key constraint can be null. Explain how a constraint on total participation of A in R can be enforced using not null constraints in SQL.
Q10.	How would you transform a database table from Second Normal Form (2NF) to Third Normal Form (3NF), and what steps would you follow to eliminate transitive dependencies explain with example?

Que No.	Question
	Assignment 2 (Unit 3)
Q1.	Write the following queries in SQL, using the university schema. a. Find the titles of courses in the Comp. Sci. department that have 3 credits. b. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result. c. Find the highest salary of any instructor. d. Find all instructors earning the highest salary (there may be more than one with the same salary). e. Find the enrollment of each section that was offered in Autumn 2009. f. Find the maximum enrollment, across all sections, in Autumn 2009. g. Find the sections that had the maximum enrollment in Autumn 2009.
Q2.	Suppose you are given a relation grade points(grade, points), which provides a conversion from letter grades in the takes relation to numeric scores; for example an "A" grade could be specified to correspond to 4 points, an "A−" to 3.7 points, a "B+" to 3.3 points, a "B" to 3 points, and so on. The grade points earned by a student for a course offering (section) is defined as the number of credits for the course multiplied by the numeric points for the grade that the student received. Given the above relation, and our university schema, write each of the following queries in SQL. You can assume for simplicity that no takes tuple has the null value for grade. a. Find the total grade-points earned by the student with ID 12345, across all courses taken by the student. b. Find the grade-point average (GPA) for the above student, that is, the total grade-points divided by the total credits for the associated courses. c. Find the ID and the grade-point average of every student.
Q3.	Write the following inserts, deletes or updates in SQL, using the university schema. a. Increase the salary of each instructor in the Comp. Sci. department by 10%. b. Delete all courses that have never been offered (that is, do not occur in the section relation). c. Insert every student whose tot cred attribute is greater than 100 as an instructor in the same department, with a salary of $10,000.
Q4.	Consider the insurance database, where the primary keys are underlined. Construct the following SQL queries for this relational database. person (driver id, name, address) car (license, model, year) accident (report number, date, location) owns (driver id, license) participated (report number, license, driver id, damage amount) a. Find the total number of people who owned cars that were involved in accidents in 2009. b. Add a new accident to the database; assume any values for required attributes. c. Delete the Mazda belonging to "John Smith".

Que No.	Question
Q5.	Suppose that we have a relation marks(ID, score) and we wish to assign grades to students based on the score as follows: grade F if score < 40, grade C if 40 ≤ score < 60, grade B if 60 ≤ score < 80, and grade A if 80 ≤ score. Write SQL queries to do the following: a. Display the grade for each student, based on the marks relation. b. Find the number of students with each grade.
Q6.	The SQL like operator is case sensitive, but the lower() function on strings can be used to perform case insensitive matching. To show how, write a query that finds departments whose names contain the string “sci” as a substring, regardless of the case.
Q7.	Consider the SQL query select distinct p.a1 from p, r1, r2 where p.a1 = r1.a1 or p.a1 = r2.a1 Under what conditions does the preceding query select values of p.a1 that are either in r1 or in r2? Examine carefully the cases where one of r1 or r2 may be empty.
Q8.	Consider the bank database, where the primary keys are underlined. Construct the following SQL queries for this relational database. branch(branch name, branch city, assets) customer (customer name, customer street, customer city) loan (loan number, branch name, amount) borrower (customer name, loan number) account (account number, branch name, balance) depositor (customer name, account number) a. Find all customers of the bank who have an account but not a loan. b. Find the names of all customers who live on the same street and in the same city as “Smith”. c. Find the names of all branches with customers who have an account in the bank and who live in “Harrison”.
Q9.	Write the following queries in SQL, using the university schema. a. Find the names of all students who have taken at least one Comp. Sci. course; make sure there are no duplicate names in the result. b. Find the IDs and names of all students who have not taken any course offering before Spring 2009. c. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor. d. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.
Q10.	How would you use different types of JOINs (INNER, LEFT, RIGHT, FULL) to retrieve specific data from two related tables, and in what situations would each type be most appropriate?

Que No.	Question
	Assignment 2 (Unit 4)
Q1.	Consider a file system such as the one on your favorite operating system. a. What are the steps involved in creation and deletion of files, and in writing data to a file? b. Explain how the issues of atomicity and durability are relevant to the creation and deletion of files and to writing data to files.
Q2.	Database-system implementers have paid much more attention to the ACID properties than have file-system implementers. Why might this be the case?
Q3.	Justify the following statement: Concurrent execution of transactions is more important when data must be fetched from (slow) disk or when transactions are long, and is less important when data are in memory and transactions are very short.
Q4.	Since every conflict-serializable schedule is view serializable, why do we emphasize conflict serializability rather than view serializability?
Q5.	Consider a database for an airline where the database system uses snapshot isolation. Describe a particular scenario in which a nonserializable execution occurs, but the airline may be willing to accept it in order to gain better overall performance.

Q6.	Consider the following two transactions:
	T34: read(A);
	read(B);
	if A = 0 then B := B + 1;
	write(B).
	T35: read(B);
	read(A);
	if B = 0 then A := A + 1;
	write(A).
	Add lock and unlock instructions to transactions T31 and T32, so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock?

Q7.	Consider the following graph-based locking protocol, which allows only exclusive lock modes, and which operates on data graphs that are in the form of a rooted directed acyclic graph. • A transaction can lock any vertex first. • To lock any other vertex, the transaction must be holding a lock on the majority of the parents of that vertex.
Q8.	Consider the following extension to the tree-locking protocol, which allows both shared and exclusive locks: • A transaction can be either a read-only transaction, in which case it can request only shared locks, or an update transaction, in which case it can request only exclusive locks. • Each transaction must follow the rules of the tree protocol. Read-only transactions may lock any data item first, whereas update transactions must lock the root first.
Q9.	How would you apply strict two-phase locking in a database system to ensure concurrency control, and what potential drawbacks might you encounter in its implementation?
Q10.	Most implementations of database systems use strict two-phase locking. Suggest three reasons for the popularity of this protocol.

Que No.	Question
	Assignment 3 (Unit 5)
Q1.	How would you apply range partitioning to optimize a range selection query on a partitioned attribute, and what advantages and potential limitations would you encounter if only one disk is accessed?
Q2.	What form of parallelism (interquery, interoperation, or intraoperation) is likely to be the most important for each of the following tasks? a. Increasing the throughput of a system with many small queries b. Increasing the throughput of a system with a few large queries, when the number of disks and processors is large
Q3.	Recall that histograms are used for constructing load-balanced range partitions. Suppose you have a histogram where values are between 1 and 100, and are partitioned into 10 ranges, 1–10, 11–20, ..., 91–100, with frequencies 15, 5, 20, 10, 10, 5, 5, 20, 5, and 5, respectively. Give a load-balanced range partitioning function to divide the values into 5 partitions. Write an algorithm for computing a balanced range partition with p partitions, given a histogram of frequency distributions containing n ranges.
Q4.	Large-scale parallel database systems store an extra copy of each data item on disks attached to a different processor, to avoid loss of data if one of the processors fails. a. Instead of keeping the extra copy of data items from a processor at a single backup processor, it is a good idea to partition the copies of the data items of a processor across multiple processors. Explain why. b. Explain how virtual-processor partitioning can be used to efficiently implement the partitioning of the copies as described above. c. What are the benefits and drawbacks of using RAID storage instead of storing an extra copy of each data item?
Q5.	Suppose we wish to index a large relation that is partitioned. Can the idea of partitioning (including virtual processor partitioning) be applied to indices? Explain your answer, considering the following two cases (assuming for simplicity that partitioning as well as indexing are on single attributes): a. Where the index is on the partitioning attribute of the relation. b. Where the index is on an attribute other than the partitioning attribute of the relation.
Q6.	For each of the three partitioning techniques, namely round-robin, hash partitioning, and range partitioning, give an example of a query for which that partitioning technique would provide the fastest response.
Q7.	What factors could result in skew when a relation is partitioned on one of its attributes by: a. Hash partitioning? b. Range partitioning? In each case, what can be done to reduce the skew?
Q8.	Give an example of a join that is not a simple equi-join for which partitioned parallelism can be used. What attributes should be used for partitioning?

Que No.	Question
Q9.	Describe a good way to parallelize each of the following: a. The difference operation b. Aggregation by the count operation c. Aggregation by the count distinct operation d. Aggregation by the avg operation e. Left outer join, if the join condition involves only equality f. Left outer join, if the join condition involves comparisons other than equality g. Full outer join, if the join condition involves comparisons other than equality
Q10.	How would you implement pipelined parallelism in a database system, and what are the key advantages and challenges you might encounter during its execution?
Assignment 3 (Unit 6)	
Q1.	How would you implement horizontal partitioning in a distributed database to improve performance, and what factors would influence your partitioning strategy?
Q2.	Can you explain how to implement distributed deadlock detection and resolution in a distributed database system, and what methods would you use?
Q3.	How would you apply the principles of data replication in a distributed database system to ensure data availability, and what trade-offs would you consider?
Q4.	How would you ensure fault tolerance in a distributed database system, and what recovery mechanisms would you employ in the event of site failures?
Q5.	Can you demonstrate how to manage concurrency control in a distributed database using two-phase commit protocol?
Q6.	How would you use sharding in a distributed database to manage large datasets, and what considerations would you need to take into account when designing the shard keys?
Q7.	How would you ensure data consistency across multiple sites in a distributed database, and what techniques would you use to handle network failures?
Q8.	How would you configure a distributed database for load balancing, and what techniques would you apply to distribute queries evenly across all nodes?
Q9.	How would you apply the concept of distributed query processing to optimize query performance in a multi-site database environment?
Q10.	How would you apply the CAP theorem to make trade-offs between consistency, availability, and partition tolerance in the design of a distributed database system?