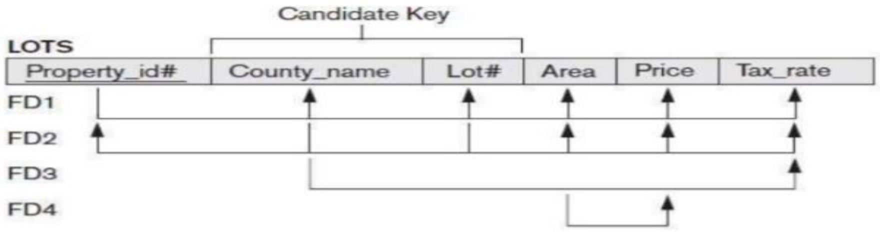
USN □□□□□□□□□□

Internal Assessment Test 2 – July 2024

Sub:	**Database Management Systems**			Sub Code:	BCS403	Branch:	ISE
Date:	**/07/2024**	Duration:	90 min's	Max Marks:	50	Sem/Sec:	IV A, B & C

	Answer any FIVE FULL Questions	MARKS	OBE CO	 RBT
1	Is this relation in 1NF, 2NF or 3NF?? Why or why not?? How would you normalize this completely?? Primary Key is Property_id#.  Candidate Key LOTS Property_id# County_name Lot# Area Price Tax_rate FD1 FD2 FD3 FD4	10	CO4	L3
2	Why normalization is required?? Explain all types of normal form with an example.	10	CO4	L2
3	a. Explain the different constraints that can be applied during table creation in SQL with an example b. Explain the Attribute datatypes and domains in SQL.	5 5	CO3	L2
4	Write the SQL queries for the following database schema: Emp (Ssn,Name,Adress,Salary,Sex,Dept.num) Dep (Dept.num,Dlocn) Proj (Pnumber,Pname,Plocation,Dnum) Workson (Ssn,pno,Hours) Dependent (Ssn,Dept name,Dept_Relationship) a) Show the resulting salaries if every employee working on the 'ProductX' project is given a 10 percent raise (2M) b) Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, first name(2M) c) Retrieve all employees in department 5 whose salary is between £30,000 and £40,000(2M) d) Retrieve all employees whose address is in Houston, Texas(2M) e) For each employee, retrieve the employee's first and last name, and the first and last name of his or her immediate supervisor.(2M)	10	CO5	L3
5	Explain the following constructs used in SQL i) Nested queries ii) Aggregate functions iii) Triggers iv) Views v) Group by and having clause	10	CO3	L2
6	Explain Join, Euijoin, Inner Join, Natural Join, Left outer join, right outer join and full outer joins with example.	10	CO3	L2

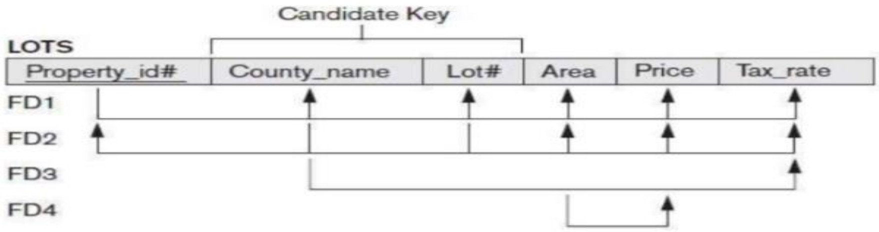
CI **CCI** **HOD**

SCHEME OF SOLUTION

Internal Assessment Test 2 – July 2024

Sub:	**Database Management Systems**				Sub Code:	BCS403	Branch:	ISE
Date:	**09/07/2024**	Duration:	90 min’s	Max Marks:	50	Sem/Sec:	IV A, B & C	OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Is this relation in 1NF, 2NF or 3NF?? Why or why not?? How would you normalize this completely?? Primary Key is Property_id#.	10	CO4	L3

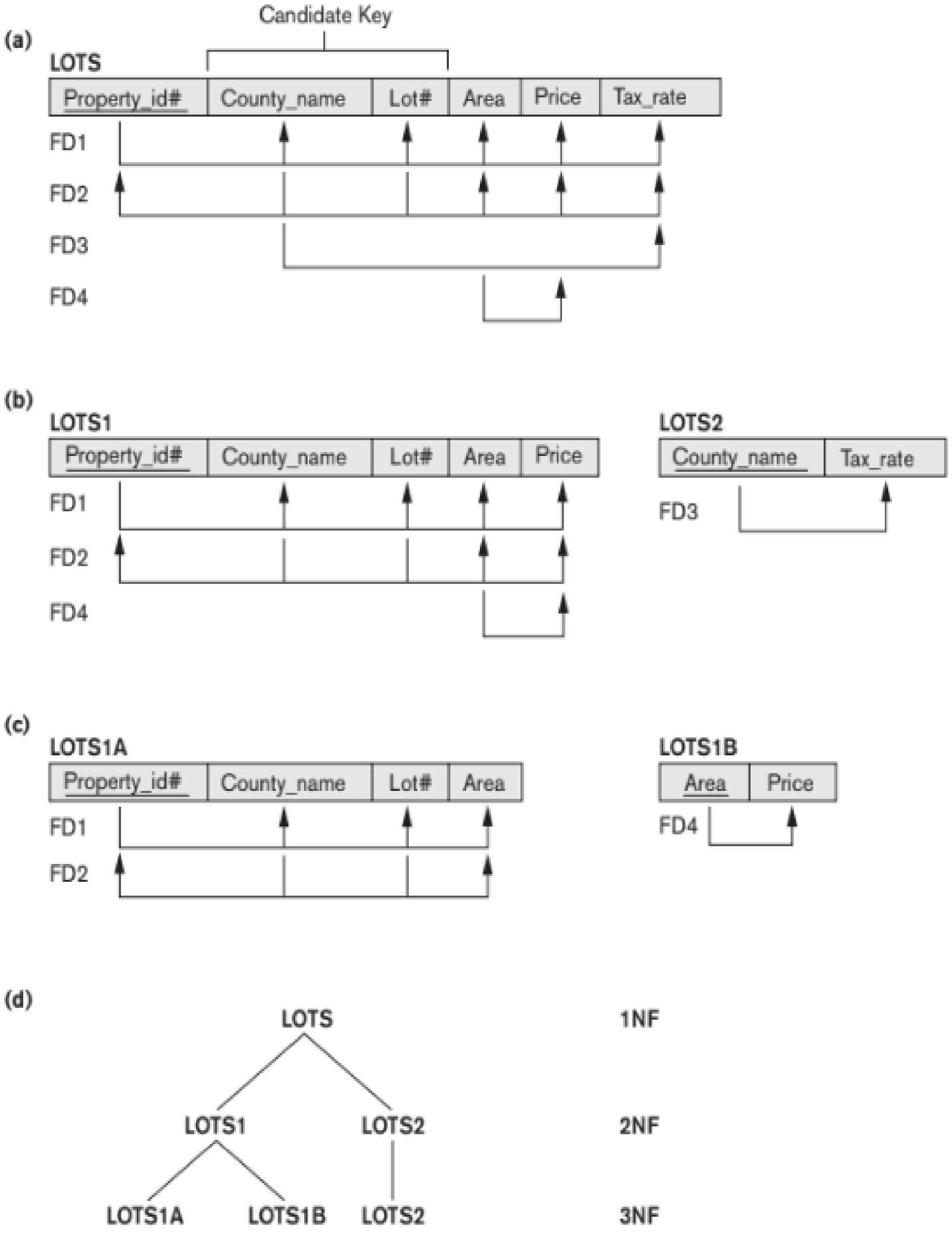


Solution:

It is in 1NF as all attributes are single valued.

It is not in 2NF as there are partial dependencies

It is not in 3NF as there are transitive dependencies



1NF:2M, 2NF: 4M, 3NF: 4M

2	Why normalization is required?? Explain alltypes of normal form with an example. **Solution:** Normalization is required to reduce redundancy and remove anomalies like insertion, deletion and updation. First (1NF) Relation should have no multivalued attributes or nested relations. Form new relations for each multivalued attribute or nested relation. Second (2NF) A relation schema R is in second normal form (2NF) if every nonprime attribute A in R is not partially dependent on any key of R. Third (3NF) A relation schema R is in third normal form (3NF) if, whenever a nontrivial functional dependency X → A holds in R, either (a) X is a superkey of R, or (b) A is a prime attribute of R.13 Definition. A relation schema R is in BCNF if whenever a nontrivial functional dependency X → A holds in R, then X is a superkey of R. BCNF A relation schema R is in BCNF if whenever a nontrivial functional dependency X → A holds in R, then X is a superkey of R. Fourth (4NF) A relation schema R is in 4NF with respect to a set of dependencies F (that includes functional dependencies and multivalued dependencies) if, for every nontrivial multivalued dependency X →→ Y in F+, Fifth (5NF) A join dependency (JD), denoted by JD(R1, R2, ... , Rn), specified on relation schema R, specifies a constraint on the states r of R. The constraint states that every legal state r of R should have a nonadditive join decomposition into R1, R2, ... , Rn. Hence, for every such r we have. Each NF: 2M	10	CO4	L2
3	c. Explain the different constraints that can be applied during table creation in SQL with anexample **Solution:** The different constraints that can be applied during table creation in SQL 1. DEFAULT Dno INT NOT NULL DEFAULT 1, 2. CHECK Dnumber INT NOT NULL CHECK (Dnumber > 0 AND Dnumber < 21); 3. PRIMARY KEY Dnumber INT PRIMARY KEY, 4. UNIQUE Dname VARCHAR(15) UNIQUE, 5. FOREIGN KEY 6. ON DELETE and ON UPDATE: SET NULL or SET DEFAULT CASCADE Any 5 constraints: 1M each b. Explain the Attribute datatypes and domains in SQL. Solution: The Attribute datatypes and domains in SQL are: 1. Numeric data types include integer numbers of various sizes (INTEGER or INT, and SMALLINT) and floating-point (real) numbers of various precision (FLOAT or REAL, and DOUBLE PRECISION). 2. Character-string data types are either fixed length—CHAR(n) or CHARACTER(n), where n is the number of characters—or varying length— VARCHAR(n) or CHAR VARYING(n) or CHARACTER VARYING(n), 3. Bit-string data types are either of fixed length n—BIT(n)—or varying length— BIT VARYING(n), 4. A Boolean data type has the traditional values of TRUE or FALSE. 5. The DATE data type has ten positions, and its components are YEAR, MONTH, and DAY in the form YYYY-MM-DD. The TIME data type has at least eight positions, with the components HOUR, MINUTE, and SECOND in the form HH:MM:SS. Any 5 datatypes: 1M each	5 5	CO3	L2

4	Write the SQL queries for the following database schema:
	Emp(Ssn,Name,Adress,Salary,Sex,Dept.num)
	Dep(Dept.num,Dlocn)
	Proj(Pnumber,Pname,Plocation,Dnum)
	Workson(Ssn,pno,Hours)
	Dependent(Ssn,Deptname,Dept_Relationship)
	f) Show the resulting salaries if every employee working on the ‘ProductX’ project is given a 10 percent raise (2M)
	g) Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, first name(2M)
	h) Retrieve all employees in department 5 whose salary is between £30,000 and £40,000(2M)
	i) Retrieve all employees whose address is in Houston, Texas(2M)
	j) For each employee, retrieve the employee’s first and last name, and the first and last name of his or her immediate supervisor.(2M)
	Solution:
	a) Show the resulting salaries if every employee working on the ‘ProductX’ project is given a 10 percent raise (2M)
	SELECT FNAME, LNAME, 1.1*SALARY
	FROM EMPLOYEE, WORKS_ON, PROJECT
	WHERE SSN = ESSN AND PNO = PNUMBER AND PNAME = ‘ProductX’;
	
	b) Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, first name(2M)
	SELECT DNAME, LNAME, FNAME, PNAME
	FROM DEPARTMENT, EMPLOYEE,WORKS_ON, PROJECT
	WHERE DNUMBER = DNO AND SSN = ESSN AND PNO = PNUMBER
	ORDER BY DNAME DESC, LNAME ASC, FNAME ASC;
	
	c) Retrieve all employees in department 5 whose salary is between £30,000 and £40,000(2M)
	SELECT *
	FROM EMPLOYEE
	WHERE (SALARY BETWEEN 30000 AND 40000) AND DNO = 5;
	
	d) Retrieve all employees whose address is in Houston, Texas(2M)
	SELECT FNAME, LNAME
	FROM EMPLOYEE
	WHERE ADDRESS LIKE ‘%Houston,TX%’;
	
	e) For each employee, retrieve the employee’s first and last name, and the first and last name of his or her immediate supervisor.(2M)
	SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME
	FROM EMPLOYEE AS E, EMPLOYEE AS S
	WHERE E.SUPERSSN = S.SSN;
	Each query: 2M
		10	CO5	L3

5	Explain the following constructs used in SQL i) Nested queries ii) Aggregate functions iii) Triggers iv) Views v) Group by and having clause **Solution:** **i) Nested queries** Whenever a condition in the WHERE clause of a nested query references some attribute of a relation declared in the outer query, the two queries are said to be correlated.We can understand a correlated query better by considering that the nested query is evaluated once for each tuple (or combination of tuples) in the outer query. For example, we can think of Q16 as follows: For each EMPLOYEE tuple, evaluate thenested query, which retrieves the Essn values for all DEPENDENT tuples with thesame sex and name as that EMPLOYEE tuple; if the Ssn value of the EMPLOYEE tupleis in the result of the nested query, then select that EMPLOYEE tuple. In general, a query written with nested select-from-where blocks and using the = or IN comparison operators can always be expressed as a single block query. Q: SELECT E.Fname, E.Lname FROM EMPLOYEE AS E, DEPENDENT AS D	10	CO3	L2

	WHERE E.Ssn = D.Essn AND E.Sex = D.Sex AND E.Fname = D.Dependent_name; **ii) Aggregate functions** Aggregate functions are used to summarize information from multiple tuples into a single-tuple summary. Grouping is used to create subgroups of tuples before summarization. Grouping and aggregation are required in many database applications, and we will introduce their use in SQL through examples. A number of built-in aggregate functions exist: COUNT, SUM, MAX, MIN, and AVG. 2 The COUNT function returns the number of tuples or values as specified in a query. The functions SUM, MAX, MIN, and AVG can be applied to a set or multiset of numeric values and return, respectively, the sum, maximum value, minimum value, and average (mean) of those values. **iii) Triggers** The trigger is given the name SALARY_VIOLATION, which can be used to remove or deactivate the trigger later. A typical trigger which is regarded as an ECA (Event, Condition, Action) rule has three components: 1. The event(s): These are usually database update operations that are explicitly applied to the database. In this example the events are: inserting a new employee record, changing an employee's salary, or changing an employee's supervisor. 2. The condition that determines whether the rule action should be executed: Once the triggering event has occurred, an optional condition may be evaluated. If no condition is specified, the action will be executed once the event occurs. If a condition is specified, it is first evaluated, and only if it evaluates to true will the rule action be executed. The condition is specified in the WHEN clause of the trigger. 3. The action to be taken: The action is usually a sequence of SQL statements, but it could also be a database transaction or an external program that will be automatically executed. In this example, the action is to execute the stored procedure INFORM_SUPERVISOR. **iv) Views** A view in SQL terminology is a single table that is derived from other tables.6 These other tables can be base tables or previously defined views. A view does not necessarily exist in physical form; it is considered to be a virtual table, in contrast to base tables, whose tuples are always physically stored in the database. In SQL, the command to specify a view is CREATE VIEW. CREATE VIEW WORKS_ON1 AS SELECT Fname, Lname, Pname, Hours FROM EMPLOYEE, PROJECT, WORKS_ON WHERE Ssn = Essn AND Pno = Pnumber; **v) Group by and having clause** The GROUP BY clause specifies the grouping attributes, which should also appear in the SELECT clause, so that the value resulting from applying each aggregate function to a group of tuples appears along with the value of the grouping attribute(s). SQL provides a HAVING clause, which can appear in conjunction with a GROUP BY clause, for this purpose. HAVING provides a condition on the summary information regarding the group of tuples associated with each value of the grouping attributes. Only the groups that satisfy the condition are retrieved in the result of the query. Each one: 2M each			

6	Explain Join, Euijoin, Inner Join, Natural Join, Left outer join, right outer join and full outer joins with example.
	
	Solution:
	In a NATURAL JOIN on two relations R and S, no join condition is specified; an implicit EQUIJOIN condition for each pair of attributes with the same name from R and S is created.
	The default type of join in a joined table is called an inner join, where a tuple is included in the result only if a matching tuple exists in the other relation. If the user requires that all employees be included, a different type of join called OUTER JOIN must be used explicitly
	LEFT OUTER JOIN (every tuple in the left table must appear in the result; if it does not have a matching tuple, it is padded with NULL values for the attributes of the right table), RIGHT OUTER JOIN (every tuple in the right table must appear in the result; if it does not have a matching tuple, it is padded with NULL values for the attributes of the left table), and FULL OUTER JOIN.
	SELECT E.Lname AS Employee_name,
	S.Lname AS Supervisor_name
	FROM (EMPLOYEE AS E LEFT OUTER JOIN EMPLOYEE AS S
	ON E.Super_ssn = S.Ssn);
	SELECT Fname, Lname, Address
	FROM (EMPLOYEE NATURAL JOIN
	(DEPARTMENT AS DEPT (Dname, Dno, Mssn, Msdate)))
	WHERE Dname = 'Research'; SELECT Fname, Lname, Address
	FROM (EMPLOYEE JOIN DEPARTMENT ON Dno = Dnumber)
	WHERE Dname = 'Research';
		10	CO3	L2

	Each one: 2M each			