Register Number									

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Department of Computer Science and Engineering

Continuous Assessment Test – II

Question Paper

Degree & Branch	B.E CSE				**Semester**	IV
Subject Code & Name	UCS1404 Database Management Systems				**Regulation: 2018**	
Academic Year	2021-22	**Batch**	2020-24	**Date**	**06.05.2022**	**FN**
Time: 90 Minutes	**Answer All Questions**				**Maximum: 50 Marks**	

Part – A (6×2 = 12 Marks)

<K3>	1. Identify the cardinality ratio for the following: An employee may work in up to two departments or may not be assigned to any department. A department should have atleast five employees. Supply (min, max) constraints. EMPLOYEE ----works_in-----DEPARTMENT (0,1) works in (5,N)	<CO1>	1.4.1
<K3>	2. For a given relvar R(A, B, C, D, E, F) the given set of FDs be E: {A -> BC, B -> E, CD -> EF} Make use of the above FDs to show that the FD AD -> F holds for R. R(A,B,C,D,E,F) and the FDs :A -> BC, B -> E, CD -> EF 1. A -> BC [given] 2. A -> C [1, decomposition] 3. AD -> CD [2, augmentation] 4. CD -> EF [given] 5. AD -> EF [3,4, trasitivity] 6. AD -> F [5, decomposition]	<CO3>	1.4.1 2.1.3
<K2>	3. Infer a minimum of 2 functional dependencies that may not hold in the following instance of the relation R (A, B, C) A->B A->C t1 (10,b1,c1) t2 (10,b2,c2) t3 (11,b1,c1) t4 (12,b3,c4) A->B C->A	<CO3>	1.4.1 2.1.3
<K3>	4. Given R={A, B, C, D, E} with the set of FDs E1:{ {A,B}->{C}, A->D, D->E}. Select the FD that don't satisfy 2NF. A->D is violating 2NF	<CO3>	1.4.1 2.1.2

<K1>	5. What is normalization? State the properties to be satisfied in normalization. Process of decomposing unstatisfactory relations into relations that satisfy the normalization rules. It follows two properties non-lossy decomposition and preservation dependency	<CO3>	1.1.1 1.4.1
<K2>	6. A relation in 3NF is not necessarily in BCNF – Verify the statement. Every relation in BCNF is also in 3NF, but the reverse is not necessarily true. 3NF allows attributes to be part of a candidate key that is not the primary key; BCNF does not.This means that relations in 3NF are often in BCNF, but not always.	<CO3>	13.3.1

Part – B (3×6 = 18 Marks)

<K2>	7. Outline the algorithm for minimal cover and find the minimal cover for the given set E: {B → A, D → A, AB → D}. A set S of FDs to be irreducible if and only if it satisfies the following three properties: 1. the right side of every FD in S involves just one attribute. 2. the left side of every FD in S is irreducible – no attribute can be discarded from the left side without changing the closure S+ 3. no FD in S can be discarded from S without changing the closure S+ Let the given set of FDs be E: {B → A, D → A, AB → D}. Find the minimal cover of E. Ans: 1. B → A 2. D → A 3. AB → D All the FDs are in canonical form which satisfies property (1). {B}+={A,B,D} {B}+={B} Different so FD is needed {D}+={D,A} {D}+={D} DIFFERENT SO have FD2 {AB}+={A,B,D} {AB}+={AB} Different needed B->A D->A	<CO3>	1.4.1 2.1.2

	AB->D {ab}+={ABD} {a}+={A} {B}+={BAD} B->D Final: B->A D->A B->D (OR) AB → D has any redundant attribute on left-side? (i) B → AB (augmenting B in 1) (ii) AB → D (given) (iii) B → D (transitivity on (i) and (ii)) Thus (3)AB → D can be replaced by (iii) B → D Now E' = { B → A, D → A, B → D } Check redundant FD. B → D , D → A => B → A, hence B → A is redundant. Minimal cover E: { B → D, D → A }		

<K2>	8. Explain the algorithm for the closure set of attributes and find the candidate keys of the relation R with set of FDs F : {AB → C, CD → E, DE → B}.
	Ans: Algorithm – Closure of set of attributes. (2)
	CLOSURE [X,F] := X;
	do forever;
	for each FD Y 咯 Z in F do ;
	if Y ⊆ CLOSURE [X,F] then
	CLOSURE [X,F] := closure [X,F] ∪ Z;
	end if;
	CLOSURE[X,F] did not change on this iteration then leave the loop;
	end;
	For a given relvar R(A,B,C,D,E) that satisfies the following FDs : AB → C, CD → E, DE → B. Identify the candidate keys of the relation R.
	Ans:
	R {A, B, C, D, E}
	{A,B,C}+ = {A,B,C}
	{A,B,D}+ = {A,B,D,C,E}
	{B,C,D}+ = {B,C,D,E}
	{C,D,E}+ = {C,D,E,B}
	{A,C,D}+ = {A,C,D,E,B}
	{A,D,E}+ = {A,D,E,B,C}
	The subsets of {A,B,D}, {A,C,D}, {A,E,D} are not key.
	Hence the candidate keys of the relation R: {A,B,D} , {A,C,D}, {A,E,D}
		<CO3>	1.4.1
			2.1.2
			2.1.3

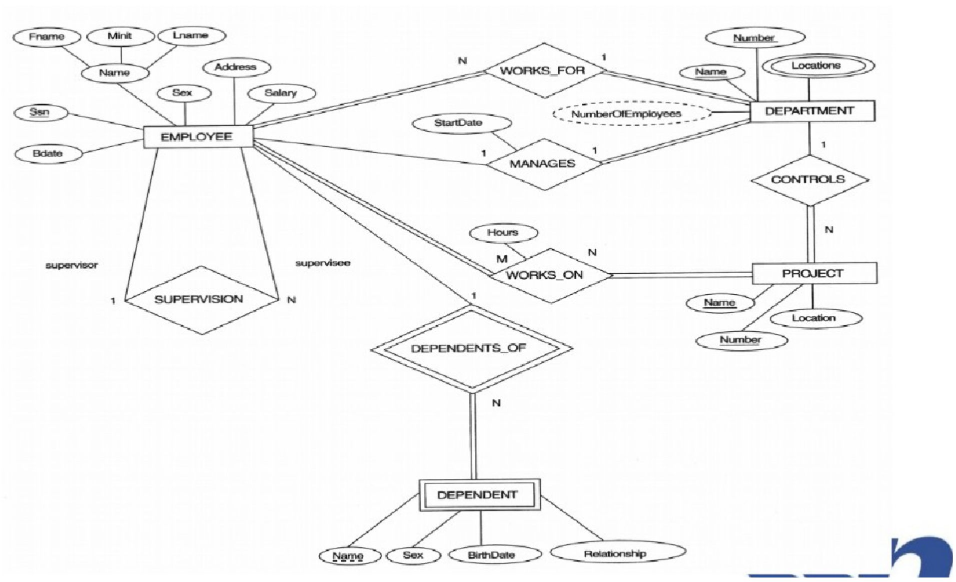
<K3>	9. Consider the following relational schema. An employee can work in more than one department; the *pct_time* of Works shows the percentage of time that a given employee works in a given department. Emp(*eid* : **integer**, *ename* : **string**, *age* : **integer**, *salary* : **real**) Works(*eid* : **integer**, *did* : **integer**, *pct_time* : **integer**) Dept(*did* : **integer**, *budget* : **real**, *managerid* : **integer**) Make use of the above schema and answer the following questions: i) Create a view SeniorEmp that shows name, age and salary of employees who are 50yrs and above. Is the view updatable? Justify your statement. CREATE VIEW Senior_Employee (empid, sname, age, salary) AS SELECT eid, ename, age, salary FROM Emp WHERE age >= 50; The view is updatable, since the primary key of the relation is included in the view. The view allows INSERT, UPDATE, DELETE through the view. ii) Give an example of a view on the above schema that would be impossible to update (automatically) and explain why your example presents the update problem. Use any group function to show it not updatable CREATE VIEW Emp_Dept AS SELECT did, count(*) FROM Emp E, Works W WHERE E.eid=W.eid GROUP BY did; The above view Emp_Dept shows the number of employees working in departmentwise. The view is nonupdatable. Because, the view contains GROUP BY clause and JOIN operations.	<CO3>	2.2.3

Part – C (2×10 = 20 Marks)

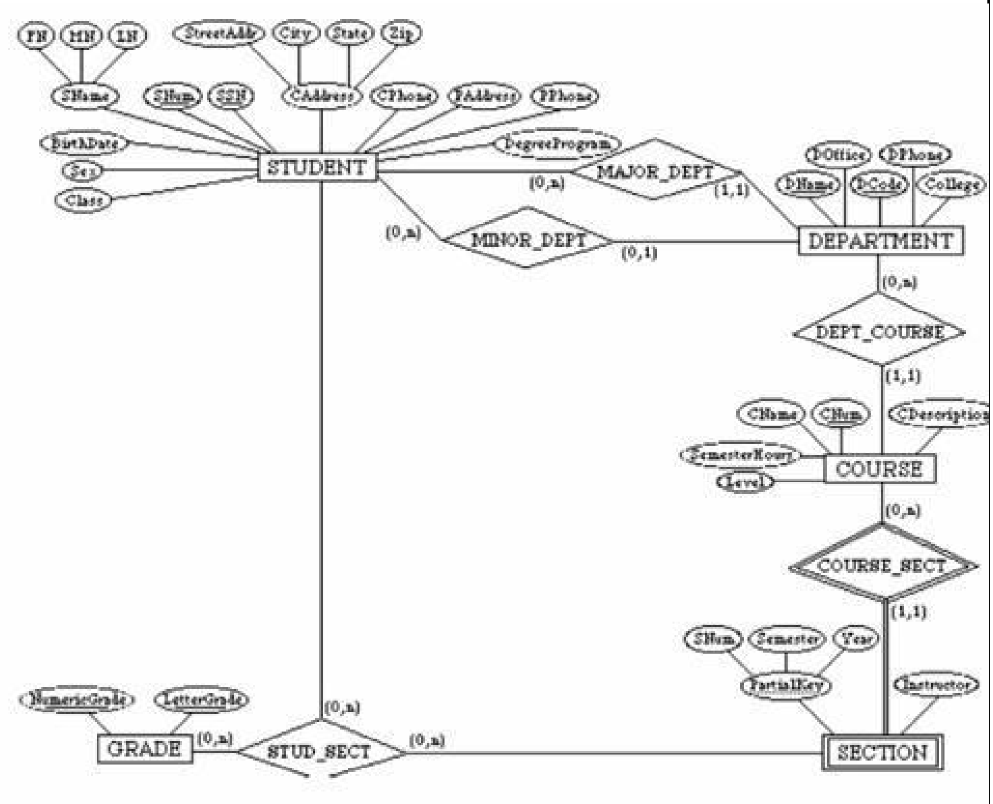
<K3>	10. The COMPANY database keeps track of company's employees, departments and projects as follows: (i) The company is organized into departments. Each department has a unique name, a unique ID, an employee who manages the department and the start date when that employee began managing the department. A department may have several locations. ii) A department controls a number of projects, each of which has a unique name, unique ID and a single location. iii) Each employee is represented by his name, social security number, address, salary, sex and birth date. An employee is assigned to one department but may work on several projects which are not necessarily controlled by same department. The number of hours per week that an employee works on each project, direct supervisor of each employee are also represented.	<CO1>	1.4.1 2.2.3 13.3.1 13.3.2

<table>
<tr>
<td></td>
<td>iv) The dependents of each employee are reprsented for insurance purposes along with dependents name, sex, birth date and relationship to the employee.

Construct ER diagram in two steps: List the entity types and its attributes. Specify key attributes of each entity type. List the relationship(s) among the entity types. Find the structural constraints on each relationship type and design an ER schema.


</td>
<td></td>
<td></td>
</tr>
<tr>
<td colspan="3">(OR)</td>
<td></td>
</tr>
<tr>
<td><K3></td>
<td>11. Consider the following set of requirements for a university database
(i) The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class, major department, minor department (if any), and degree program (B.E., M.E., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student.
ii) Each department is described by a name, department code, office number, office, phone, and college. Both name and code have unique values for each department.
iii) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.
iv) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ...; up to the number of sections taught during each semester.
v) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).

List the entity types and its attributes. Specify key attributes of each entity type. List the relationship(s) among the entity types. Find the structural constraints on each relationship type and design an ER schema for this application and construct an ER diagram.</td>
<td><CO1></td>
<td>1.4.1
2.2.3
13.3.1
13.3.2</td>
</tr>
</table>



ER Schema diagram for exercise 3.16:

<K3>	12. Consider the universal relation R = {A, B, C, D, E, F, G, H, I, J } and the set of functional dependencies F = {A,B} -> {C}, {B,D} -> {E,F}, {A,D} -> {G,H}, {A} -> {I}, {H} -> {J}. Identify the key for R? Decompose R into 2NF and then to 3NF relations. State the reasons behind each decomposition. R = {A, B, C, D, E, F, G, H, I, J } F = {A,B} -> {C}, {B,D} -> {E,F}, {A,D} -> {G,H}, {A} -> {I}, {H} -> {J} To help in solving this problem systematically, we calculate the closures of pairs of attributes that are possible keys: {A, B}+ -> {A, B, C, I}, {B, D}+ -> {B, D, E, F}, {A, D}+ -> {A, D, G, H, I, J} None of these pairs are keys either since none of the closures includes all attributes. But the union of the three closures includes all the attributes: {A, B, D}+ -> {A, B, C, D, E, F, G, H, I} Hence, {A, B, D} is a key. ---------------- (2) (Note: Algorithm for computing the closure Z+ of Z can be used to determine a key). Based on the above analysis, we decompose as follows, starting with the following relation R: R = {A, B, D, C, E, F, G, H, I, J} The first-level partial dependencies(or reducible FD) on the key (which violate 2NF) are: {A, B} -> {C, I}, {B, D} -> {E, F}, {A, D} -> {G, H, I, J} Hence, R is decomposed into R1, R2, R3, R4 (keys are underlined): R1 = {A, B, C, I}, R2 = {B, D, E, F}, R3 = {A, D, G, H, I, J}	<CO3>	1.4.1 2.1.2 2.1.3

Additional partial dependencies(or reducible FD) exist in R1 and R3 because {A} -> {I}.

Hence, we remove {I} into R4, so the following relations are the result of 2NF decomposition:

R1 = {A, B, C}, R2 = {B, D, E, F}, R3 = {A, D, G, H, J}, R4 = {A, I}

2NF Decompostion with explanation ------------------- (4)

Next, we check for transitive dependencies in each of the relations (which violate 3NF).

Only R3 has a transitive dependency {A, D} -> {H} -> {J}, so it is decomposed into R31 and R32 as follows:

R31 = {H, J}, R32 = {A, D, G, H}

The final set of 3NF relations is {R1, R2, R31, R32, R4}

3NF Decompostion with explanation ------------------- (4)

(OR)

<K3> <CO3> 1.4.1 2.1.2 2.1.3

13. Make use of the relation SJT(student, subject, teacher) and the set of FDs F:

{{student, subject} → teacher, teacher → subject}.

i) Identify the candidate keys for SJT.

ii) Define Boyce-Codd Normal Form. Is the relation in 3NF but not in BCNF ?

State the reasons behind your decomposition.

iii) Discuss the update anomalies before decomposition.

iv) After decomposition, are the resulting relations free from update anomalies

and satisfy dependency-preserving property?

SJT(student, subject, teacher)

FD1: {student, subject} -> teacher

FD2: teacher -> subject.

Student	Subject	Teacher
Ramesh	Maths	Prof. Raghav
Ramesh	Physics	Prof. Raj
Suresh	Maths	Prof. Raghav
Suresh	Physics	Prof. Krishna

i) Identify the candidate keys.

{student, teacher}+ --> {student, teacher, subject }

{student, subject}+ --> {student, subject, teacher }

Hence {student, subject} and {student, teacher} are two overlapping candidate keys.

ii) Is this relation in BCNF? If not, justify your answer and decompose if necessary.

The relation is not in BCNF. Because in FD2 Teacher is a determinant, but not a candidate key. The two BCNF projections are:

ST {student, teacher } --- No FD

TJ {teacher, subject } FD2: teacher --> subject

iii) Discuss the update anomalies before decompostion.

SJT suffers from certain update anomalies. If we wish to delete the information that student Suresh is studying Physics, we cannot do so

	without at the same time losing the information that Prof. Krishna teaches Physics. iv) After decompostion, is the resulting relations are free from update anomalies and satisfy dependency-preserving property? Although the two projections does avoid certain anomalies, they are not independent. An attempt to insert a tuple for Ramesh and Prof. Krishna into ST must be rejected, because Prof.Krishna teaches Physics and Ramesh is already being taught Physics by Prof. Raj : yet the system cannot detect this fact without examining TJ. After decompositon, the FD1 {student, subject} --> teacher cannot be deduced from the FD2 teacher --> subject. Hence the FDs are not preserved.		

Student	Subject	Teacher
Ramesh	**Maths**	**Prof. Raghav**
Ramesh	**Physics**	**Prof. Raj**
Suresh	**Maths**	**Prof. Raghav**
Suresh	**Physics**	**Prof. Krishna**
Suresh	**Chemistry**	**Prof. Hemant**

. Identify the candidate key(s) of SJT. (1)
{student, teacher}+ --> {student, teacher, subject }
{student, subject}+ --> {student, subject, teacher }
There are two candidate keys
b. Explain the update anomalies before decomposition. (1)
SJT suffers from certain update anomalies. If we wish to delete the information that student Suresh is studying Physics, we cannot do so without at the same time losing the information that Prof. Krishna teaches Physics.
c. Analyze the two cases of decomposition based on the selection of primary key from the candidate keys. (6)
Case1:
If {student,teacher} is selected as primary key, then SJT is not in 2NF.

SJT: (student, subject, teacher)
FD2 is violating 2NF – FD2 is partial FD.
Decompose SJT into
ST(student,teacher) teacher references TJ(teacher)
TJ(teacher,subject)
The above decomposition is in 3NF and also in BCNF.

Case2:
If {student,subject} is selected as primary key, then SJT is in 3NF, but not in BCNF.

SJT: (student,subject,teacher)
FD2 is violating BCNF – Because in FD2, Teacher is a determinant,
but not a candidate
key. The two BCNF projections are:
ST(student,teacher) teacher references TJ(teacher)
TJ(teacher,subject)
The above decomposition is in BCNF.

d. After decompostion, is the resulting relations are free from update anomalies and satisfy dependency-preserving property? (2)

Although the two projections does avoid certain anomalies, they are not independent. An attempt to insert a tuple for Ramesh and Prof. Krishna into ST must be rejected, because Prof.Krishna teaches Physics and Ramesh is already being taught Physics by Prof. Raj : yet the system cannot detect this fact without examining TJ. After decompositon, the FD1 {student, subject} --> teacher cannot be deduced from the FD2 teacher --> subject. Hence the FDs are not preserved.