USN

Internal Assessment Test 1 – Sept. 2017

Sub:	DATABASE MANAGEMENT SYSTEM					Sub Code:	15CS53	Branch:	CSE / ISE
Date:	20/09/2017	Duration:	90 min’s	Max Marks:	50	Sem / Sec:	5 / CSE- A,B,C	ISE-A,B	OBE

Answer any 4 Questions from Q.no 1 to Q.no 6 questions. Answer any 1 question out of Q.no 7 and Q.no 8.

		MARKS	CO	RBT
1	Discuss concepts related to structural constraints of a relationship type with examples.	[10]	CO1	L2
2	Define an entity and an attribute. Explain the different types of attributes that occur in an ER model, with an example.	[10]	CO1	L2
3	Explain operations from set theory in relational algebra with examples.	[10]	CO2	L2
4	Explain schema-based constraints in relational model.	[10]	CO2	L2
5 (a)	Write a note on weak entity and super key	[05]	CO1	L2
(b)	Write a note on recursive relationships and role names.	[05]	CO1	L2
6 (a)	Define referential integrity constraint with suitable example.	[05]	CO2	L2
(b)	The tables ITEM and COMPANY are given below. Show the results of ITEM X COMPANY and ITEM * COMPANY	[05]	CO2	L3

ITEM		
ITEMID	**ITEMNAME**	**CID**
10	Chocolate	21
11	Cakes	22
12	Biscuit	21

COMPANY		
CID	**CNAME**	**CITY**
21	Parle	Bangalore
22	Britannia	Mysore
23	Pepsi	Bangalore

7	Write an ER diagram for hospital management considering at least four entities.	[10]	CO1	L3
8	Write an ER diagram for university data base considering at least four entities.	[10]	CO1	L3

USN

Internal Assessment Test 1 – Sept. 2017

Sub:	DATABASE MANAGEMENT SYSTEM					Sub Code:	15CS53	Branch:	CSE
Date:	20/09/2017	Duration:	90 min’s	Max Marks:	50	Sem / Sec:	5 A,B,C		OBE

Answer any 4 Questions from Q.no 1 to Q.no 6 questions. Answer any 1 question out of Q.no 7 and Q.no 8.

		MARKS	CO	RBT
1	Discuss concepts related to structural constraints of a relationship type with examples.	[10]	CO1	L2
2	Define an entity and an attribute. Explain the different types of attributes that occur in an ER model, with an example.	[10]	CO1	L2
3	Explain operations from set theory in relational algebra with examples.	[10]	CO2	L2
4	Explain schema-based constraints in relational model.	[10]	CO2	L2
5 (a)	Write a note on weak entity and super key.	[05]	CO1	L2
(b)	Write a note on recursive relationships and role names.	[05]	CO1	L2
6 (a)	Define referential integrity constraint with suitable example.	[05]	CO2	L2
(b)	The tables ITEM and COMPANY are given below. Show the results of ITEM X COMPANY and ITEM * COMPANY	[05]	CO2	L3

ITEM		
ITEMID	**ITEMNAME**	**CID**
10	Chocolate	21
11	Cakes	22
12	Biscuit	21

COMPANY		
CID	**CNAME**	**CITY**
21	Parle	Bangalore
22	Britannia	Mysore
23	Pepsi	Bangalore

7	Write an ER diagram for hospital management considering at least four entities.	[10]	CO1	L3
8	Write an ER diagram for university data base considering at least four entities.	[10]	CO1	L3

USN

SCHEME OF EVALUATION – IAT 1 – Sept. 2017

Sub:	DATABASE MANAGEMENT SYSTEM					Sub Code:	15CS53	Branch: CSE / ISE
Date:	20/09/2017	Duration:	90 min's	Max Marks:	50	Sem / Sec:	5 / CSE- A,B,C	ISE-A,B

Answer any 4 Questions from Q.no 1 to Q.no 6 questions. Answer any 1 question out of Q.no 7 and Q.no 8.

Question No.		Description	Marks Split up	Total Marks
1.		✓ Cardinality ratio definition ✓ Types of Cardinality ratio ✓ Cardinality ratio example ✓ Participation definition ✓ Types of Cardinality ratio ✓ Participation example	1 2 2 1 2 2	10M
2.		✓ Entity and Attribute definition ✓ Composite vs Simple, Multi values vs Single, Derived vs Stored with examples	2 8	10M
3		✓ List Set theoretic operations ✓ Explanation + Examples	2 2+2+2+2	10M
4.		✓ Domain Constraint ✓ Key Constraint ➢ Super key and keys ➢ Candidate key and Primary key ➢ Entity Integrity Constraint ✓ Referential Integrity Constraint ➢ Foreign key	2 3 2 3	10M
5.	a)	✓ Weak Entity, Identifying relationship ✓ Super Keys	4 1	10M
	b)	✓ Recursive Relation ship ✓ Role names ✓ ER representation	2 2 1	
6.	a)	✓ Referential integrity constraint ✓ Suitable example.	3 2	10M
	b)	✓ ITEM X COMPANY ✓ ITEM * COMPANY	2.5 2.5	
7.		✓ ER diagram for Company Database		10M
8.		✓ ER diagram either University Database		10M

SOLUTIONS

1. Discuss concepts related to structural constraints of a relationship type with examples.

Cardinality ratio and participation constraints, taken together are the structural constraints of a relationship type.

The **cardinality ratio** for a binary relationship specifies the maximum number of relationship instances that an entity can participate in.

For example, consider a binary relationship type WORKS_FOR between Department and Employee entity types, DEPARTMENT:EMPLOYEE is of cardinality ratio 1:N, meaning that each department can be related to numerous employees, but an employee can be related to (work for) only one department.

The possible cardinality ratios for binary relationship types are 1:1, 1:N, N:1, and M:N.

The binary relationship MANAGES which relates a department entity to the employee who manages that department; the cardinality ratio is 1:1. This represents the constraint that an employee can manage only one department and that a department has only one manager.

The relationship type WORKS_ON between Employee entity and the Project entity that he works for, is of cardinality ratio M:N, representing that an employee can work on several projects and a project can have several employees.

The **participation constraint** specifies whether the existence of an entity depends on its being related to another entity via the relationship type. There are two types of participation constraints—total and partial.

If a company policy states that every employee must work for a department, then an employee entity can exist only if it participates in a WORKS_FOR relationship instance. Thus, the participation of EMPLOYEE in WORKS_FOR is called total participation, meaning that every entity in "the total set" of employee entities must be related to a department entity via WORKS_FOR. Total participation is also called existence dependency.

On the other hand, we do not expect every employee to manage a department, so the participation of EMPLOYEE in the MANAGES relationship type is partial, meaning that some or "part of the set of" employee entities are related to a department entity via MANAGES, but not necessarily all.

2. Define an entity and an attribute. Explain the different types of attributes that occur in an ER model, with an example.

Entities and Attributes

The basic object that the ER model represents is an entity, which is a thing in the real world with an independent existence. An entity may be an object with a physical existence (for example, a particular person, car, house, or employee) or it may be an object with a conceptual existence (for instance, a company, a job, or a university course).

Attribute—the particular properties that describe an entity. For example, an EMPLOYEE entity may be described by the employee’s name, age, address, salary, and job.

Several types of attributes occur in the ER model: simple versus composite, singlevalued
versus multivalued, and stored versus derived.

Composite versus Simple (Atomic) Attributes: Composite attributes can be divided into smaller subparts, which represent more basic attributes with independent meanings. For example, the Address attribute of the EMPLOYEE entity can be subdivided into Street_address, City, State, and Zip,3 with the values ‘2311 Kirby’, ‘Houston’, ‘Texas’, and ‘77001.’ Attributes that are not divisible are called simple or atomic attributes. Composite attributes can form a hierarchy; for example, Street_address can be further subdivided into three simple component attributes: Number, Street, and Apartment_number,

Single-Valued versus Multivalued Attributes: Most attributes have a single value for a particular entity; such attributes are called single-valued. For example, Age is a single-valued attribute of a person.

one person may not have a college degree, another person may have one, and a third person may have two or more degrees; therefore, different people can have different numbers of values for
the College_degrees attribute. Such attributes are called multivalued. A multivalued attribute may have lower and upper bounds to constrain the number of values allowed for each individual entity.

Stored versus Derived Attributes: In some cases, two (or more) attribute values are related—for example, the Age and Birth_date attributes of a person. For a particular person entity, the value of Age can be determined from the

current (today's) date and the value of that person's Birth_date. The Age attribute is hence called a derived attribute and is said to be derivable from the Birth_date attribute,
which is called a stored attribute.

3. Explain operations from set theory in relational algebra with examples.

The following are the set theoretic operations are used to merge the elements of two sets in various ways in relational algebra,

- UNION
- INTERSECTION
- SET DIFFERENCE

When these operations are adapted to relational databases, the two relations on which any of the above three operations are applied must have the same type of tuples; this condition is called union compatibility.

UNION: The result of this operation, denoted by R ∪ S, is a relation that includes all tuples that are either in R or in S or in both R and S. Duplicate tuples are eliminated.

Example:

R:

Name
Ram
Raju
Rakesh

S:

Name
Ram
Rajesh
Ramu

R ∪ S:

Name
Ram
Raju
Rakesh
Rajesh
Ramu

INTERSECTION: The result of this operation, denoted by R ∩ S, is a relation that includes all tuples that are in both R and S.

Example:

R:

Name
Ram
Raju
Rakesh

S:

Name
Ram
Rajesh
Ramu

R ∩ S:

Name
Ram

SET DIFFERENCE: The result of this operation, denoted by R - S, is a relation that includes all tuples that are in R but not in S.

Example:

R:

Name
Ram
Raju
Rakesh

S:

Name
Ram
Rajesh
Ramu

R − S:

Name
Raju
Rakesh

4. Explain schema-based constraints in relational model.

Domain Constraints: Domain constraints specify that the value of each attribute A must be an atomic value from the domain dom(A). The data types associated with domains typically include standard numeric data types for integers (such as short-integer, integer, long-integer) and real numbers (float and double-precision float). Characters, fixed-length strings, and variable-length strings are also available, as are date, time, timestamp, and money data types.

Key Constraints and Constraints on Null: A relation is defined as a set of tuples. By definition, all elements of a set are distinct; hence, all tuples in a relation must also be distinct. This means that no two tuples can have the same combination of values for all their attributes. Usually, there are other subsets of attributes of a relation schema R with the property that no two tuples in any relation state r of R should have the same combination of values for these attributes. Suppose that we denote one such subset of attributes by SK; then for any two distinct tuples t1 and t2 in a relation state r of R, we have the constraint that

$$t_1[SK] \neq t_2[SK]$$

Any such set of attributes SK is called a super key of the relation schema R. A super key SK specifies a uniqueness constraint that no two distinct tuples in a state r of R can have the same value for SK. A super key can have redundant attributes, however, so a more useful concept is that of a key, which has no redundancy.

A key K of a relation schema R is a super key of R with the additional property that removing any attribute A from K leaves a set of attributes K' that is not a super key of R. Hence, a key is a minimal super key—that is, a super key from which we cannot remove any attributes and still have the uniqueness constraint hold.

In general, a relation schema may have more than one key. In this case, each of the keys is called a candidate key. One among of the candidate keys is designated as the primary key of the relation, which is used to uniquely identify a tuple in the relation.

Another constraint on attributes specifies whether null values are or are not permitted. For example, if every STUDENT tuple must have a valid, non-null value for the Name attribute, then Name of STUDENT is constrained to be NOT NULL.

Entity Integrity constraint: The entity integrity constraint states that no primary key value can be null. This is because the primary key value is used to identify individual tuples in a relation; having null values for the primary key implies that we cannot identify some tuples. If two or more tuples had null for their primary keys, we might not be able to distinguish them.

Referential Integrity and Foreign Keys: The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples of the two relations. The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples of the two relations.

To define referential integrity more formally, we first define the concept of a foreign key. The conditions for a foreign key, given below, specify a referential integrity constraint between the two relation schemas R1 and R2. A set of attributes FK in relation schema R1 is a foreign key of R1 that references relation R2 if it satisfies the following two rules:

1. The attributes in FK have the same domain(s) as the primary key attributes PK of R2; the attributes FK are said to reference or refer to the relation R2.

2. A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is null. In the former case, we have $t_1[FK] = t_2[PK]$, and we say that the tuple t1 references or refers to the tuple t2. R1 is called the referencing relation and R2 is the referenced relation.

If these two conditions hold, a referential integrity constraint from R1 to R2 is said to hold.

5. a) Write a note on weak entity and super key.

Weak Entity: Entity types that do not have key attributes of their own are called weak entity types. Entities belonging to a weak entity type are identified by being related to specific entities from another entity type in combination with some of their attribute values. We call this other entity type the **identifying or owner entity type** and we call the relationship type that relates a weak entity type to its owner the **identifying relationship** of the weak entity type. A weak entity type always has a total participation constraint (existence dependency) with respect to its identifying relationship, because a weak entity cannot be identified without an owner entity.

Consider the entity type DEPENDENT, related to EMPLOYEE, which is used to keep track of the dependents of each employee. The attributes of DEPENDENT are Name (the first name of the dependent), BirthDate, Sex, and Relationship (to the employee). Two dependents of two distinct employees may, by chance, have the same values for Name, BirthDate, Sex, and Relationship, but they are still distinct entities. They are identified as distinct entities only after determining the particular employee entity to which each dependent is related. Each employee entity is said to own the dependent entities that are related to it.

In ER diagrams, both a weak entity type and its identifying relationship are distinguished by surrounding their boxes and diamonds with double lines. A partial key, which is the set of attributes, is used to uniquely identify weak entities that are related to the same owner entity. The partial key attribute is underlined with a dashed or dotted line

Super Key: A relation is defined as a set of tuples. By definition, all elements of a set are distinct; hence, all tuples in a relation must also be distinct. This means that no two tuples can have the same combination of values for all their attributes. Usually, there are other subsets of attributes of a relation schema R with the property that no two tuples in any relation state r of R should have the same combination of values for these attributes. Suppose that we denote one such subset of attributes by SK; then for any two distinct tuples t1 and t2 in a relation state r of R, we have the constraint that

$$t1[SK] \neq t2[SK]$$

Any such set of attributes SK is called a super key of the relation schema R. A super key SK specifies a uniqueness constraint that no two distinct tuples in a state r of R can have the same value for SK. A super key can have redundant attributes, however, so a more useful concept is that of a key, which has no redundancy.

b) Write a note on recursive relationships and role names.

Each entity type that participates in a relationship type plays a particular role in the relationship. The role name signifies the role that a participating entity from the entity type plays in each relationship instance, and helps to explain what the relationship means. For example, in the WORKS_FOR relationship type, EMPLOYEE plays the role of employee or worker and DEPARTMENT plays the role of department or employer.

In some cases the same entity type participates more than once in a relationship type in different roles. In such cases the role name becomes essential for distinguishing the meaning of the role that each participating entity plays. Such relationship types are called recursive relationships.

The SUPERVISION relationship type relates an employee to a supervisor, where both employee and supervisor entities are members of the same EMPLOYEE entity set. Hence, the EMPLOYEE entity type participates twice in SUPERVISION: once in the role of supervisor (or boss), and once in the role of supervisee (or subordinate).

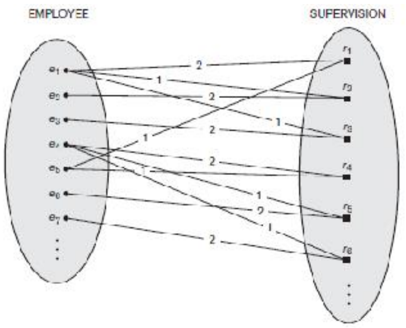


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the supervisor role (1) and EMPLOYEE in the subordinate role (2).

6. a) Define referential integrity constraint with suitable example.

The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples of the two relations. The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples of the two relations.

To define referential integrity more formally, we first define the concept of a foreign key. The conditions for a foreign key, given below, specify a referential integrity constraint between the two relation schemas R1 and R2. A set of attributes FK in relation schema R1 is a foreign key of R1 that references relation R2 if it satisfies the following two rules:

1. The attributes in FK have the same domain(s) as the primary key attributes PK of R2; the attributes FK are said to reference or refer to the relation R2.

2. A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is null. In the former case, we have t1[FK] = t2[PK], and we say that the tuple t1 references or refers to the tuple t2. R1 is called the referencing relation and R2 is the referenced relation.

If these two conditions hold, a referential integrity constraint from R1 to R2 is said to hold.

Example:

ITEM		
ITEMID	**ITEMNAME**	**CID**
10	Chocolate	21
11	Cakes	22
12	Biscuit	~~**24**~~

COMPANY		
CID	**CNAME**	**CITY**
21	Parle	Bangalore
22	Britannia	Mysore
23	Pepsi	Bangalore

In the above example, the row with ItemID 12 is not a valid row as it is referring to a row with CID 24, which does not exist in the parent table COMPANY.

b) The tables ITEM and COMPANY are given below. Show the results of

ITEM X COMPANY and ITEM * COMPANY

ITEM		
ITEMID	**ITEMNAME**	**CID**
10	Chocolate	21
11	Cakes	22
12	Biscuit	21

COMPANY		
CID	**CNAME**	**CITY**
21	Parle	Bangalore
22	Britannia	Mysore
23	Pepsi	Bangalore

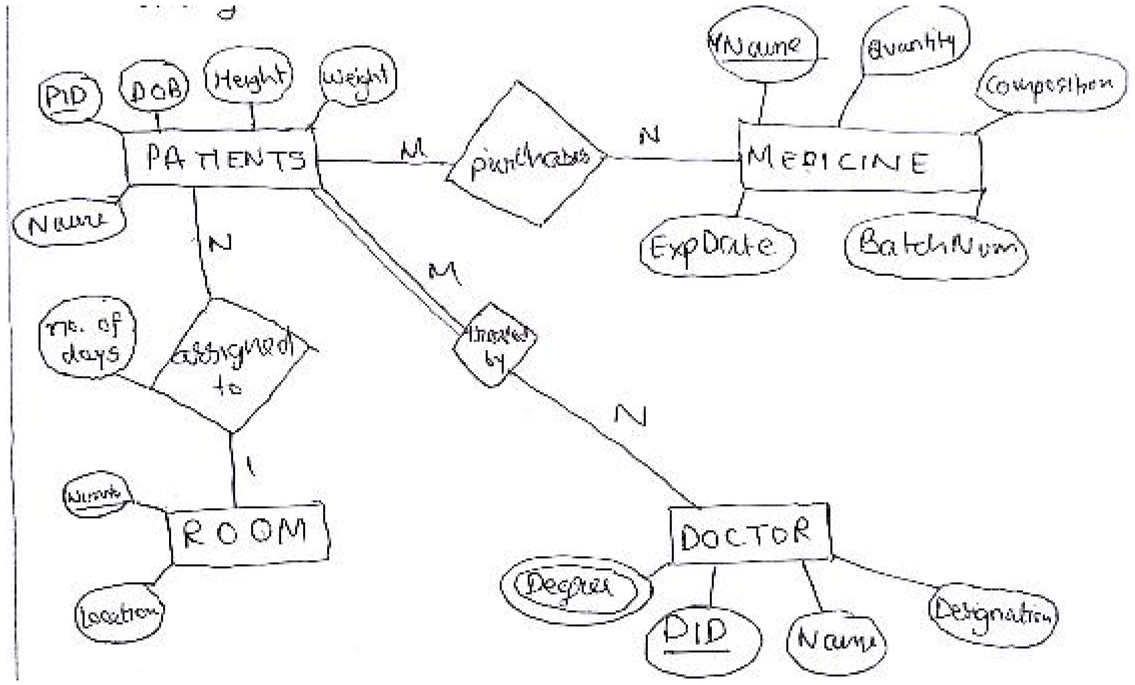
ITEM X COMPANY

ITEMID	ITEMNAME	CID	CID	CNAME	CITY
10	Chocolate	21	21	Parle	Bangalore
10	Chocolate	21	22	Britannia	Mysore
10	Chocolate	21	23	Pepsi	Bangalore
11	Cakes	22	21	Parle	Bangalore
11	Cakes	22	22	Britannia	Mysore
11	Cakes	22	23	Pepsi	Bangalore
12	Biscuit	21	21	Parle	Bangalore
12	Biscuit	21	22	Britannia	Mysore
12	Biscuit	21	23	Pepsi	Bangalore

ITEM * COMPANY

ITEMID	ITEMNAME	CID	CNAME	CITY
10	Chocolate	21	Parle	Bangalore
11	Cakes	22	Britannia	Mysore
12	Biscuit	21	Parle	Bangalore

7. **Write an ER diagram for hospital management considering at least four entities.**



Assumptions

① Patient has to purchase atleast one medicine

② Nurses, security Guards, Housekeeping staff are partime hence needn't be accounted for.

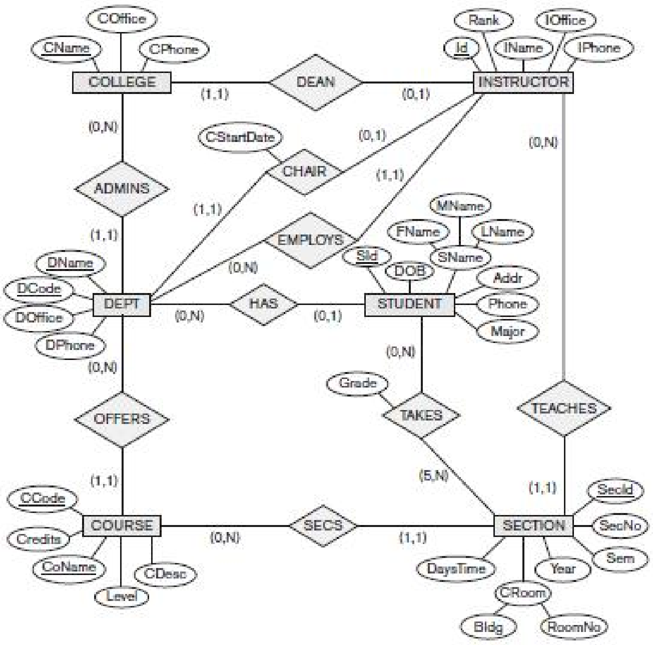
③ Patient should be assigned to atleast one doctor.

④ Any nurse can be ~~directe~~- assigned to a patient based on availability

⑤ Nurse is assigned by doctor.

⑥ Every doctor will be the boss of a ~~pixed~~ fixed no. of Nurses.

8. Write an ER diagram for university data base considering at least four entities.



COffice
CName
CPhone
COLLEGE
(1,1)
DEAN
(0,1)
Rank
IOffice
Id
IName
IPhone
INSTRUCTOR
(0,N)
CStartDate
(0,1)
CHAIR
(1,1)
(0,N)
ADMINS
(1,1)
(1,1)
EMPLOYS
MName
FName
LName
SName
SId
DOB
Addr
Phone
Major
DName
DCode
DOffice
DPhone
DEPT
(0,N)
HAS
(0,1)
STUDENT
(0,N)
(0,N)
Grade
TAKES
TEACHES
OFFERS
(5,N)
(1,1)
(1,1)
SecId
SecNo
Sem
Year
CCode
Credits
CoName
Level
CDesc
COURSE
(0,N)
SECS
(1,1)
SECTION
DaysTime
CRoom
Bldg
RoomNo