CHAPTER 3

Module III

Relational Data Model

Syllabus

Introduction to Relational Model, Relational Model Constraints and Relational Database Schemas, Concept of Keys Primary Key, Secondary key, Foreign Key, Mapping the ER and EER Model to the Relational Model.

3.1 Relational Model

1. Introduction

- The relational model first proposed by E. F. Codd hence he is known as father of relational model.
- Relational database was an attempt to simplify database structure by making use of tables and columns.
- Tables are known as **relations**, columns are known as **attributes** and rows (or records) are known as **tuples**.

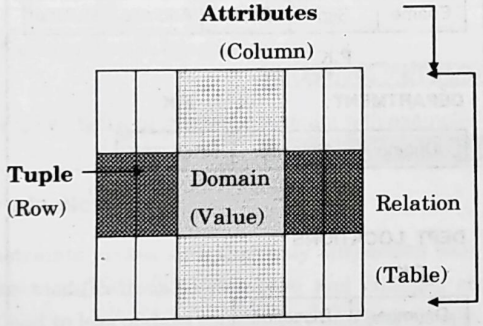

Fig. 3.1.1 : Relational Algebra Notations

2. Relation (Table)

Q. Define Relation.

- Relations are a logical structure which is a collection of tables consisting horizontal rows also called as tuples and vertical columns also called as Attributes.
- The table containing rows and columns represents entity in relational model, it is called as Relation.
- This concept doesn't represent how the data is stored in the physical memory of computer system.
- The relation can contain data about a single entity or relationship between two entities.

Characteristics of Relation

- A table composed of rows and columns.
- Each table in a database has its unique *table name.*
- Each table row (tuple) represents a single entity occurrence within the entity set.
- All values in a same column must conform to the same format of data.
- Each table must have a single attribute or set of attributes that uniquely identifies each row.

Example,

The student relation can be shown as given below,

Id	Name	Age	Class	Branch
105	Mahesh	25	BE	IT
106	Suhas	28	FE	CS
107	Jay	29	SE	CS
108	Sachin	30	TE	EXTC

3. Attributes (Column)

Q. Define Attributes

- Relation has its own properties which describes that relation (table) such properties are known as attributes.
- In relational model, the column in relation (table) or field of data is also called as Attribute.
- Every table must have at least one column in it.
- The single attribute will contains the similar type of data of all entities in relation.

Name
Mahesh
Suhas
Jay
Sachin

- It is not possible to have multiple columns with same column name in the same relation. But it is possible to have multiple columns with same column name in two different relations.
- The SQL standard does not specify any maximum number of columns in a table.

Example,

The Name attribute in above student relation will contains the name of all student entities in student relation.

4. Tuple (Row / Records)

- A single row in relational table which contains all the information about a single entity is called as Tuple.
- The single row in relation (Table) is called as Tuple.
- Each horizontal row of the student table represents a Student tuple.
- A table can have any number of rows in it.

Stud_Id	Name	Age	Std	Div
105	Mahesh	25	BE	A

Example,

The above tuple contains all data about the Id 105, student entity in student relation.

5. Domain (Data Value)

- The intersection column and row in a relational table which represents data of entity is called as Domain.
- Every column in a table has a set of data values that are allowed for that column which is called as Domain.
- In a relational table a domain can have a single value or no (Null) value.
- The single domain will contains the specific data of single entities in relation.

Name
Mahesh

Example,

The Name attribute of tuple id 105 will contains the name of all student with id 105 in student relation.

3.2 Relational Database Schema

Q. Write a note on Relational Database Schema.

Q. Compare Relational Database Schema and Relational Schema with example.

1. Introduction

- The term schema refers to the organization or structure of data in relational database.
- The relational schema describes structure of relation (i.e. table) and relational database schema explains the structure of relational database.

2. Relational Schema

- Relation schema consists of a number of attributes associated with relation.
- Example,

EMPLOYEE	(Ssn, Ename, Bdate, Address, Dnumber)
DEPARTMENT	(Dnumber, Dname, Dmgr_ssn)
DEPT_LOCATIONS	(Dnumber, Dlocation)
PROJECT	(Pnumber, Pname, Plocation, Dnum)
WORKS_FOR	(Ssn, Pnumber, Hours)

Fig. 3.2.1 Sample Relational Schema

3. Relational Database Schema

Relational database schema consists of a number of relation schemas associated with that database.

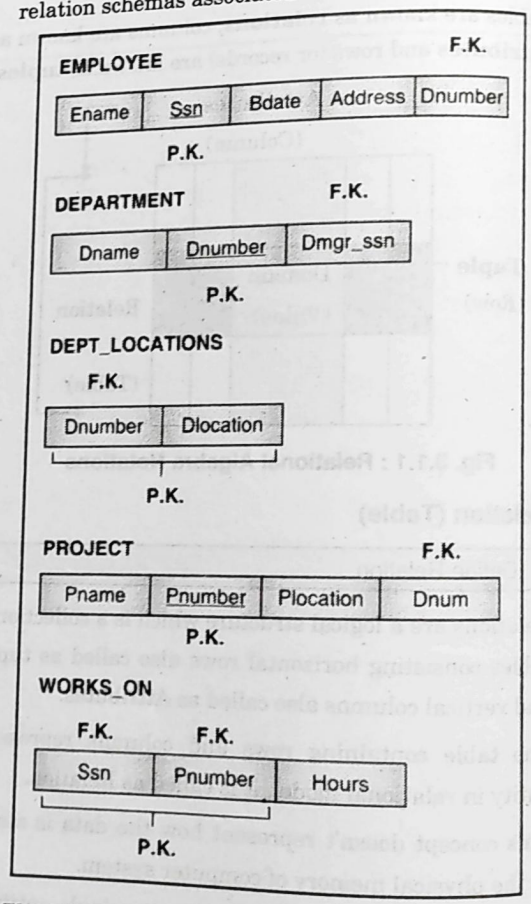

Fig. 3.2.2 : Sample Relational Database Schema

4 Comparison

- The attributes are grouped to form a relation schema by mapping a conceptual data model i.e. ER or EER data model to relational schema.
- The relational mapping will identify entity types and relationship types and their respective attributes.
- The relation schema consists of a number of attributes in relation while the relational database schema consists of a number of relation schemas in the corresponding database.

Relational Schema (Table Structure)	Relational Database Schema (Database Structure)
$R_1 (A_1, A_2, .., A_n)$ $R_2 (B_1, B_2, .., B_n)$ ... $R_m (C_1, C_2, .., C_n)$	$R_1, R_2, ..., R_n$

Where,

$R_1, R_2, ... R_n$ = Relation or Tables

A, B, C = Attributes or Columns

3.3 Relational Model Constraints

Q Explain different integrity constraints.
MU - Dec. 16, 10 Marks

Q. Explain types of integrity constraint with example.
MU - Dec. 19, 10 Marks

1. Introduction

- Constraints makes sure that only authorised user will make modifications to database and changes should not lead to loss of data consistency and correctness.
- These concepts makes sure correctness and completeness of data stored in the database.
- This objective can never be guaranteed, one cannot ensure that every entry made in database is accurate.
- Some example of incorrect data is as below,
  1. Student taking admission to a branch which is not available in the college.
  2. Employee assigned with non existing department.
  3. Sometimes inconsistency introduced due to system failures.

2. Types of Relational Constraints

- Domain Relational Constraints
- Entity Relational Constraints
- Referential Relational Constraints

3.4 Domain Relational Constraint

Q. Explain different integrity constraints.
MU - Dec. 16, 10 Marks

1. Introduction

- Domain constraints allow us to test whether the values inserted into the database are correct or not.
- The CREATE TABLE Command may also include domain constraints which can check integrity of database.
- These domain constraints are the most basic form of integrity constraint.

2. Types of Domain Constraints

- Nullness Constraint / Required Data Constraint
- CHECK Constraint / User Defined Constraint
- DEFAULT Constraint

A. Required Data Constraint / Nullness Constraint

- The database may have some attributes mandatory like user registration must have user email address.
- These attributes (columns) in a database are not allowed to contain NULL values or blanks.

Example,

In the student table, student must have an associated student name.

```
Student_Name varchar(100) NOT NULL
```

- Therefore now, the Student_Name column in the STUDENT table is a required data column. It is not possible to insert Null value in Student_Name column of Student table.
- The DBMS can prevent user from inserting NULL values in any column with help of such constraints.

B. Check Constraint

- The **check constraint** is used to ensure that attribute value satisfies specific condition as specified by data requirements or user.
- Suppose in Student Table, gender of student can be male or female only.
- The DBMS can prevent user from entering incorrect or other data in database table.

Example,

Table with student entity having gender which can be M or F.

Hence, attribute gender can take only two values either 'M' or 'F'.

```
Student_Gender varchar(1) CHECK (gender IN ('M','F'))
```

C. Default Keyword

- Default keyword is used to add a default specified value, if attribute value is not provided by user.
- It avoids the addition of NULL value to the database by inserting default value as specified by the developer while creating a table.

Example,

- Table with customer entity having name and gender.
- If name is not added for customer that will be taken as 'Unknown' if we specify DEFAULT value of NAME column to 'UNKNOWN'

```
Student_Name varchar(50) DEFAULT 'UNKNOWN'
```

3.5 Entity Integrity Constraints

Q. Explain different integrity constraints.
MU - Dec. 16, 10 Marks

1. Introduction

- Entity constraints allow us to test whether the tuple (entity) inserted into the database are correct or not.
- The create table Command may also include entity constraints which can primary key of table.

2. Types of Entity Constraints

A. Unique Constraint
B. Primary Key Constraint

A. Unique Constraint / Unique Key

- In case of unique constraint no two tuples can have equal value for same attributes.
- This constraint says that attributes forms candidates key, which allows one Null value which is unique by itself.
- This UNIQUE constraint can be applicable to user defined domain declaration also.

Example,

```
EMAIL varchar(30) UNIQUE
```

B. Primary Key Constraint

Q. Explain the term : Primary Key.
MU - May 16, 2 Marks

- A table in a relational database has one column or combination of some columns whose values uniquely identifies a single row in the table. This column or combination of columns is called the **primary key** of the table.
- Primary key attribute is same as unique key constraint with NOT NULL constraints (Unique constraint+ Not Null constraint).
- The main difference in unique constraint and primary key constraint is that one null value is allowed in unique constraint which can be treated as unique value while nulls are not allowed in primary key constraint.
- For example, each row of the STUDENT table has a unique set of values in its STUDENT_ID column, which uniquely identifies the student represented by that row.
- Duplicate values are not allowed in primary key column, because they cause problems in distinguishing one entity from another (entity may be an employee).
- The DBMS can prevent user from inserting same data values in a column again and again.

```
STUDENT_ID char(10) PRIMARY KEY
```

3.6 Referential Integrity / Foreign Key

Q. Discuss what is meant by term : Referential integrity.
MU - Dec. 15, 3 Marks

Q. Attempt the following : foreign Key.
MU - Dec. 18, 5 Marks

1. Introduction

- A value appearing in a one relation (table) for a given set of attributes also appears for another set of attributes in another relation (table). This is called referential integrity.
- The referential integrity constraint is specified between two tables to maintain the consistency among tuples in the two tables.
- The tuple in one relation refers only to an existing tuple in another relation.

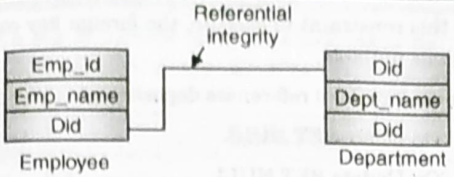

Fig. 3.6.1 : Referential integrity

Employee Table		
Emp_Id	**Emp_name**	**Did**
1	Sachin	20
2	Suhas	10
3	Jay	20
4	Om	10

Department Table	
Did	**Dept_name**
10	HR
20	TIS
30	L&D

- In the above example Employee table has Did as foreign key reference to Did column in Department table this is called as referential integrity.
- Here we are forcing the database to check the value of Did column from the department table while inserting any value in Employee table. This helps to maintain data consistency.

2. Foreign key violations in SQL

- If any row in EMP table is added with 'Did' value which is not there in department table the insert statement will give foreign key violation error.
- In above tables we will refer Department as parent table (as it is containing Primary key) and Employee table as Child table (as it is containing Foreign Key).
- There are 4 problems causes the foreign key violations,

a. Adding new tuple to Child Table (Add Child)

If we try to add an employee with Did 70 to employee table (Child Table), it will return foreign key violation error. As Did 70 is not there in Department table (Parent table).

```
INSERT INTO Employee
VALUES (11,'Devid', 70);
```

Output

ORA-02291 : integrity constraint (Employee.FK_Employee) violated - parent key not found

Adding new employee		
Emp_Id	**Emp_name**	**Did**
11	Devid	70

- This functionality helps to maintain data consistency in database.

b. Updating tuple from Child Table

If we try to update an employee Emp_Id = 2 with Did as 70 to employee table (Child Table), it will return foreign key violation error. As Did 70 is not there in Department table (Parent table).

```
UPDATE Employee
SET Did = 70
WHERE Emp_Id=2;
```

Output

- **ORA-02291 :** Integrity constraint (Employee.FK_Employee) violated - parent key not found.
- This functionality helps to maintain data consistency in database.

c. Deleting tuple from Parent Table

If we try to delete department of Did = 10 from Department table (Parent table), it will return foreign key violation error. As there are few employees working in department with Did =10.

```
DELETE Department
WHERE Did=10;
```

Output :

ORA-02292 : integrity constraint (Employee.FK_Employee) violated - child record found

This functionality will creates limitation for deletion of parent record if it has some associated child records.

d. Updating tuple from Child Table

If we try to update department of Did = 10 with Did = 70, it will return foreign key violation error. As there are few employees still working in department with Did =10.

```
UPDATE Department
SET Did = 70
WHERE Did = 10;
```

Output

ORA-02292 : integrity constraint (Employee.FK_Employee) violated - child record found

This functionality will creates limitation for updating parent record if it has some associated child records.

3. Delete-Update (DU) rules to solve problem of foreign key violation

- If any row in EMP table is added with 'Did' value which is not there in department table the insert statement will give foreign key violation error.
- This rule can be enforced as given below,

```
Create Table Employee
(
   Eid varchar (50) Primary Key,
   ....
   Did varchar (50) foreign key references department (Did)
      On delete CASCADE
      On update CASCADE
   )
```

a. NO ACTION / RESTRICT

- This clause will discards the delete or update operation on the parent table.
- In this case the database engine will not allow user to delete the row and using FK violation error.
- The RESTRICT rule will not allow you to delete a row from the parent table although as there is corresponding row present in child table.
- Foreign key (Did) references department :
  - On delete RESTRICT
  - On Update RESTRICT
- The database engine will give the error and the delete action on the row in the parent table is ignored.
- Deletion of department is not allowed as there is some employees are present in that department.

b. CASCADE

- Corresponding rows are deleted from the referencing table (Child table), if that row is deleted from the parent table.
- Foreign key (Did) references department :
  - On delete CASCADE
  - On Update CASCADE
- If a department is deleted then all the employee records that refers to the deleted department are also been deleted.

c. SET NULL

- Foreign key data value is set to NULL, if the corresponding row in the parent table is deleted.
- For this constraint to execute, the foreign key columns must be nullable.
- Foreign key (Did) references department :
  - On delete SET NULL
  - On Update SET NULL
- Insert Null value of did in the place of deleted did in employee table.

d. SET DEFAULT

- Foreign key data values refers to non-existing foreign key are set to their default values.
- For this constraint to execute, all foreign key columns must have default definitions.
- Foreign key (Did) references department :
  - On delete SET DEFAULT
  - On Update SET DEFAULT
- If a column is null able, and there is no explicit default value set, NULL becomes the implicit default value of the column.
- Insert any default value of 'did' (which exists in the departing table) in the place of deleted 'Did'.

4. Self-referential relations

- A foreign key constraint can reference some other columns in same tables.
- It form self-referential relations.

For Example,

- EMPLOYEE (employee_id, employee_name, manager_id)
- Because the manager is also employee, there is a foreign key relationship from the **manager_id** column to the **employee_id** column in same table.

3.7 Concept of Keys

Q. What is the difference between unique key and primary key? **MU - May 18, 5 Marks**

Q. Explain different keys in DBMS. **MU - Dec. 19, 10 Marks**

Q. Explain concept of keys relational Table.

- The column value that uniquely identifies a single record in a table called as key of table.
- An attribute or set of attributes whose values uniquely identify each entity in an entity set is called a key for that entity set.

- ID is a key of student table. It is possible to have only one student with a one ID (Say only one student **'Mahesh'** with ID = 1)

Key type	Definition
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row. (Explained in details in previous section Entity Integrity)
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes.
Foreign key	An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null. (Explained in details in previous section Referential Integrity)
Unique Key	In case of unique constraint no two tuples can have equal value for same attributes. This constraint says that attributes forms candidates key, which allows one Null value which is unique by itself. This UNIQUE constraint can be applicable to user defined domain declaration also. **Example : EMAILvarchar(30) UNIQUE**

3.8 Mapping Entities to Relational Model (Self Learning Topic)

Q. Explain the steps of an algorithm for ER to relational mapping. **MU – Dec. 13, 10 Marks**

Q. Explain the algorithm to map ER and EER model to relational model in detail. **MU - May 17, 10 Marks**

1. Regular entity types

- **Tables :** Regular entity sets can be represented as table in relational model.
- **Columns :** Attributes of entity set can be converted to the columns (attributes) of the tables in relational model.
- **Example :** Regular entity employee mapped as employee table in object model like 'Stud_id', 'Stud_Addr' etc. are shown as table columns.

ER model

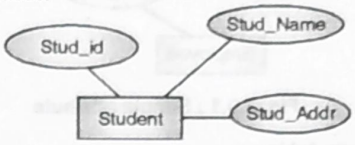

Fig. 3.8.1 : Regular entity

Stud_id	Stud_Name	Stud_Addr
1	Snehal	Mumbai
2	Pratiksha	Mumbai
3	Supriya	Mumbai
4	Tanmay	Goa

2. Weak entity types

- For each weak entity type with owner entity, create a table and include all simple attributes of weak entity type as columns of table, including foreign key attributes as the primary key attribute of the table that correspond to the owner entity type.
- **Example :** Dependents (Weak entity) in Employee (Owner entity).

ER Model

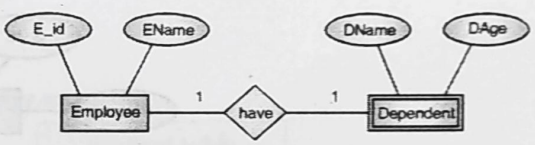

Fig. 3.8.2 : Weak entity

E_id	Ename	DName	DAge
1	Sachin	Jyoti	23
2	Suhas	Manju	22
3	Jayendra	Tanya	27

3.9 Mapping Attributes to Columns of Table

Q. Explain how to convert attribute in ER to relational Table.

a. Simple attributes

Simple attribute can be directly converted to a column (Attribute) in relational model.

Example,

Employee 'Age' can be directly converted to column.

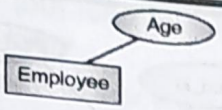

Fig. 3.9.1 : Simple attribute

Employee table

Eid	Age
1	23
2	24
3	43
4	28

b. Composite attributes

These attributes need to be stored as set of simple component attributes (Columns) in relational model by avoiding actual attribute ('name' in below example).

Example : In below example composite attribute 'Name' is converted to three columns in object model by avoiding actual attribute 'Name'.

ER

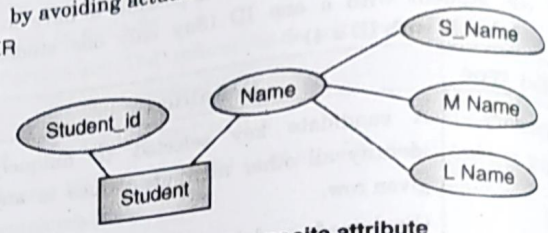

Fig. 3.9.2 : Composite attribute

Student table

Student_id	S_Name	MName	LName
1	Harshad	Rupali	Malar
2	Bipin	Anand	Shinde
3	Aanand	Ganesh	Panchal
4	Tushar	Bipin	Pimple

c. Multi valued attributes

Multi valued attributes are mapped as a relation which includes combination of the primary key of table and multi valued attribute as a composite primary key as shown in Fig. 3.9.3.

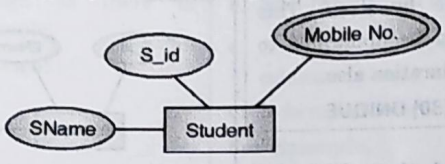

Fig. 3.9.3 : Multi valued attribute

Student table

Sid	SName
1	Jayendra
2	Suhas
3	Sachin

Mobile Table

Sid	Mobile No.
1	9891295492
1	9821959241
2	808091856
3	8095124890
3	9773112456

Mobile table	
Sid	**Mobile No.**
1	9891295492
1	9821959241
2	8080918456
3	8095124890
3	9773112456

d. Derived attributes

There is no need to store such attribute in relational model. It will be calculated from stored attribute.

e. Key attributes

Key attribute in ER Model can be directly converted to primary key attribute of relational model.

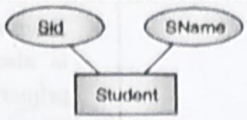

Fig. 3.9.4 : Key attributes

Student table

Sid	SName
1	Deepak
2	Vaibhav
3	Yogita
4	Bency

3.10 Mapping Relationships

Q. Explain how to convert various type of relations in ER to relational Table.

a. Foreign key approach

If binary relationship type does not possess many attributes then we can map such relation using foreign key.

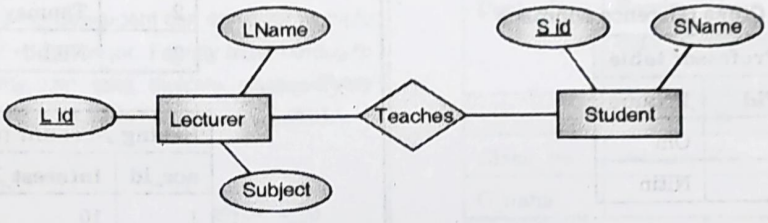

Fig. 3.10.1 : Foreign key approach

Lecturer table		
Lid	**LName**	**Subject**
1	Omprakash	IP
2	Yogesh	INS
3	Amit	PM

Student table		
Sid	**SName**	**Lid**
1	Bency	1
2	Deepak	1
3	Yogita	1
4	Snehal	2
5	Pratiksha	2

- Lid is foreign key in student table while primary key in lecturer table.

b. Merged relationship approach

When participation is total it is possible to merge relation and involved entities as a single relation and then map it to a table.

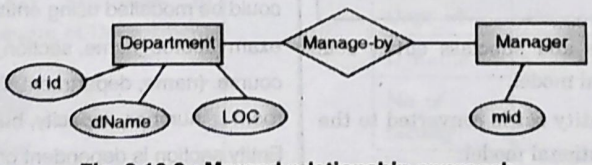

Fig. 3.10.2 : Merged relationship approach

Department table			
did	**Dname**	**Loc**	**mid**
10	IDF	Mahape	11
20	Mayban	Pune	22
30	Lax	Mumbai	33

c. Cross reference approach

- A relationship type in EER is mapped to new table in relational model.
- Column of such table is all attributes of relation and primary key attributes of all tables linked to this relation.

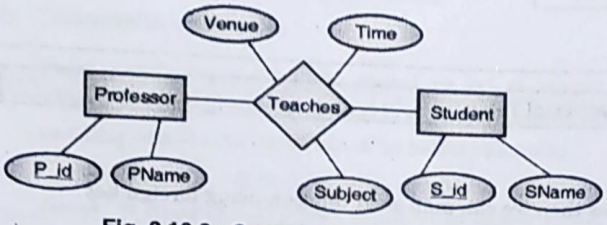

Fig. 3.10.3 : Cross reference approach

Professor table	
Pid	**PName**
1	Om
2	Nitin

Student table	
Sid	**SName**
1	Snehal
2	Tanmay

Teacher relation table				
Pid	**Sid**	**Venue**	**Time**	**Subject**
1	1	SFIT	1 pm	ADBMS
1	2	XIE	1 pm	DBMS

3.11 Mapping Inheritance constraints

Q. Explain how to convert inheritance in ER to relational Table.

- **Tables :** Each super class and subclass entity sets represents table in relational model.
- **Columns :** Attributes of entity set is converted to the columns of the tables in relational model.
- **Primary key :** The primary key column of super class is also added to all subclasses and treated as a primary key column for all tables in relational model.

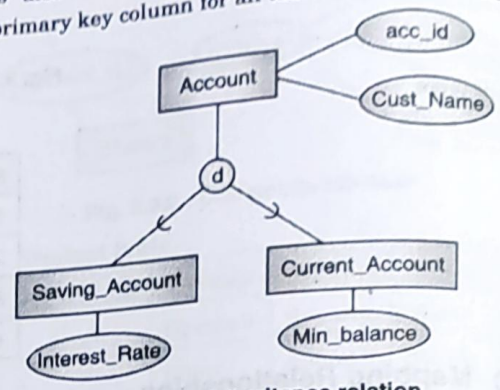

Fig. 3.11.1 : Inheritance relation

Account table	
acc_id	**Cust_Name**
1	Snehal
2	Tanmay
3	Nikhil

Saving_Account table	
acc_id	**Interest_Rate**
1	10
2	12
3	15

Current_Account	
acc_id	**Min balance**
1	1000
2	2000
3	500

3.12 Solved Examples

Example 3.12.1 : Draw an E-R diagram and reduce it to relational database model for a university database for scheduling of classrooms for final exams. This database could be modelled using entities as

exam (course_name, section_number, room_number, time);

course. (name, department, C_number),

room (r_number, capacity, building);

Entity section is dependent on course.

Solution :

Step 1 : ER diagram

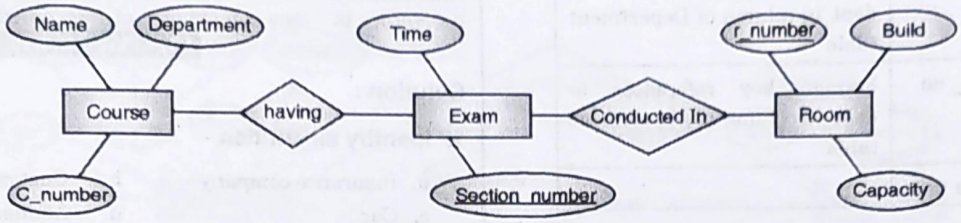

Fig. Ex. 3.12.1

Step 2 : Mapping to Tables

Course (C_number, name, department)

Exam (Section_number, time, C_number,)

Room (R_number, building, capacity, Section_number)

Example 3.12.2 : Draw an E-R diagram for a university database consisting of 4 entities,

1. Student
2. Department
3. Class
4. Faculty and convert it to tables.

A student has a unique id, the student can enroll for multiple classes and has at most one major. Faculty must belong to department and faculty can take multiple classes-Every student will get a grade for the class he/she has enrolled.

Solution :

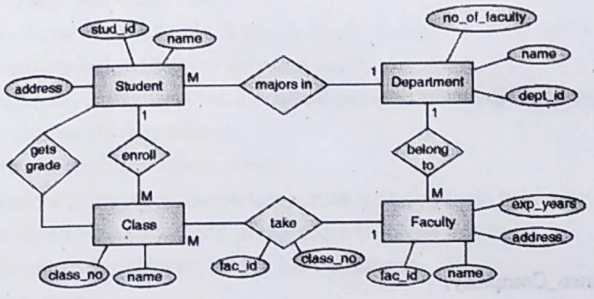

Fig. Ex. 3.12.2

Student	
Id	Primary Key
Name	
Address	
Dept_id	Foreign key references to dept_id column of Department table
Faculty	
Fac_id	Primary Key
Fac_name	
Exp_years	
Address	
Dept_id	Foreign key references to dept_id column of Department table
Class	
Class_ no	Primary Key
C_name	
Stud_class	
Class_ no	Foreign key references to dept_id column of Department table
Stud_ id	Foreign key references to dept_id column of Department table
take_class	
Fac_id	Foreign key references to fac_id column of Faculty table
Class_ no	Foreign key references to class_ no column of Class table
Department	
Dept_id	Primary Key
D_name	
No_of faculty	

Grade	
Stud_ id	Foreign key references to dept_id column of Department table
Class_ no	Foreign key references to dept_id column of Department table
Grade	

Example 3.12.3 : Construct on E-R diagram for a car-insurance company that has a set of customers each of whom owns one or more cars. Each car has associated with it zero to any number of recorded accidents. **MU - May 14, May 16, 10 Marks**

Solution :

1. Identify all entities

a. Insurance company
b. Customer
c. Car
d. Accidents

2. Construct ER diagram by merging all above relationships

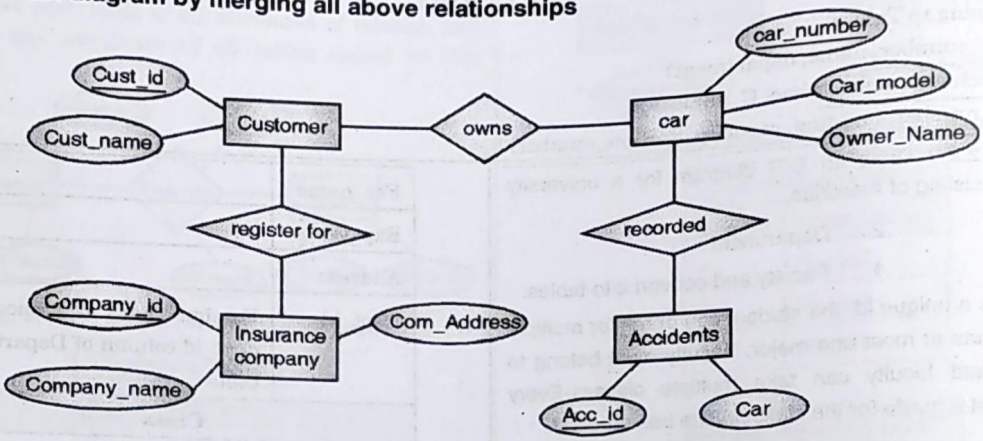

Fig. Ex. 3.12.3

A. Mapping Entities

a. Company (Company_id, Name, Address)
b. Customer (Customer_id, Name, Address, phone)
c. Car (Car_Number, Car_Model, Owner)
d. Accidents (Accident_Id, Location, date, time)

B. Mapping Relations

a. Company (Company_id, Name, Address)
b. Customer (Customer_id, Name, Address, phone, Insurance_Company)
Insurance_Company - refers to customers registered insurance company
c. Car (Car_Number, Car_Model, Owner_Id)
Owner_Id - refers to customer id owns that car.
d. Accidents (Accident_Id, Car_Number, Location, date, time)
Car_Number - refers to car involved in accident.

C. Final Relational Schema

a. Company (Company_id, Name, Address)
b. Customer (Customer_id, Name, Address, phone, Insurance_Company)
c. Car (Car_Number, Car_Model, Owner_Id)
d. Accidents (Accident_Id, Car_Number, Location, date, time)

Example 3.12.4 : Describe/list the steps/rules of ER-to-relational mapping and use the same to map the ER diagram shown in figure-Ex. 3.12.4 to relational database schema. **MU – Dec. 18, 10 Marks**

Solution :

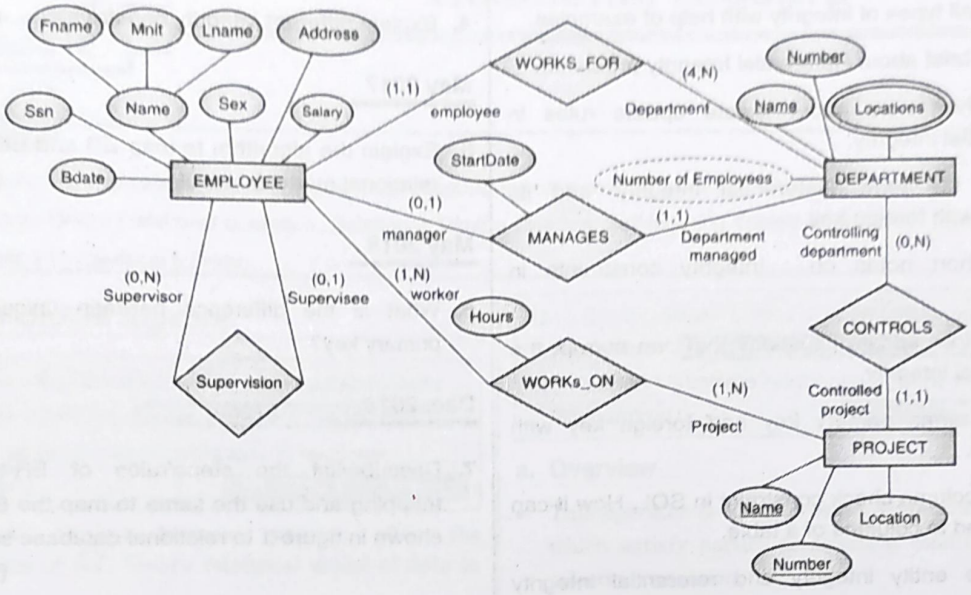

Fig. Ex. 3.12.4

Example 3.12.5 : Suppose you are given the following requirements for a simple database of the National Cricket Trophy (NCT) :

the NCT has many teams

each team has a name, a city, a coach, a captain, and a set of players,

each players belongs to only one team,

each player has a name, a position (such as left wing or goalie), a skill level,

and a set of injury records,

a team captain is also a player,

a game is played between two teams (referred to as host team and guest team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).

Construct ER diagram for the NCT database. **MU – Dec. 19, 10 Marks**

Solution :

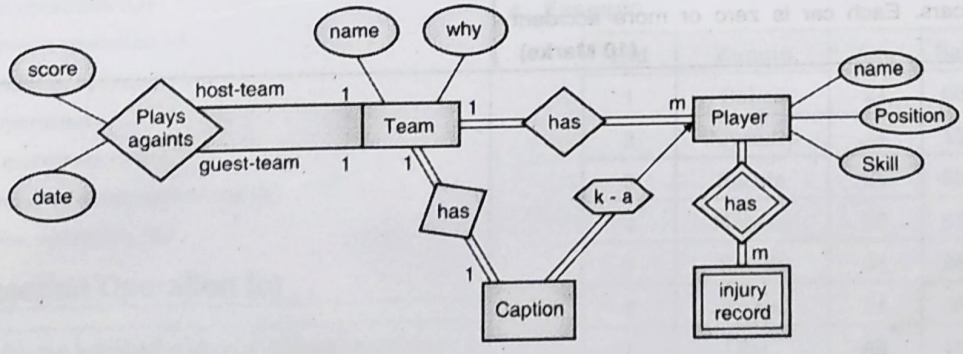

Fig. Ex. 3.12.5

Review Questions

1. Explain all types of integrity with help of examples.
2. Write in brief about Referential Integrity problems.
3. Write short note about Delete update rules in Referential integrity.
4. Explain the term 'Referential Integrity' and its relation with foreign key.
5. Write short notes on : Integrity constraints in RDBMS.
6. Explain four update situations that can corrupt the referential integrity.
7. Explain terms primary key and foreign key with example.
8. Explain column check constraint in SQL. How it can be applied to columns of a table.
9. Why are entity integrity and referential integrity important in a database ?

3.13 University Questions and Answers

Dec. 2015

1. Discuss what is meant by term : Referential integrity. **(3 Marks)**

May 2016

2. Explain the term : Primary Key. **(2 Marks)**
3. Draw E-R diagram for a car-insurance company that has a set of customers. Each of customer has one or more cars. Each car is zero or more accident record. **(10 Marks)**

Dec. 2016

4. Explain different integrity constraints. **(10 Marks)**

May 2017

5. Explain the algorithm to map ER and EER model to relational model in detail. **(10 Marks)**

May 2018

6. What is the difference between unique key and primary key? **(5 Marks)**

Dec. 2018

7. Describe/list the steps/rules of ER-to-relational mapping and use the same to map the ER diagram shown in figure-1 to relational database schema. **(10 Marks)**
8. Attempt the following : Foreign Key **(5 Marks)**

May 2019

9. Define key constraint and referential constraint. Explain the concept of foreign key with example. **(10 Marks)**

Dec. 2019

10. Explain types of integrity constraint with example. **(10 Marks)**
11. Explain different keys in DBMS. **(10 Marks)**

❑❑❑