

DATE: 9/4/21

1: LIBRARY Database

Consider the following schema for a Library Database

BOOK (Book_id, Title, Publisher_Name, Pub_Year)
 BOOK_AUTHORS (Book_id, Author_Name)
 PUBLISHER (Name, Address, Phone)
 BOOK_COPIES (Book_id, Branch_id, No-of_Copies)
 BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)
 LIBRARY_PROGRAMME (Programme_id, Programme_Name, Address)

Write SQL queries to

1. Retrieve details of all books in the library_id, title, name of publisher, authors, number of copies in each branch, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the Library.

Primary Key : It is also called as primary key used in a relational database that is unique for each record.

Foreign Key : It is defined in a second table, but it refers to the primary key or unique key in the first table.

Tuple : Tuple a single row of table, which contains a single record for that relation is called as tuple.

Schema : The term Schema refers to organization of data in a database.

Constraint : It is to maintain the data integrity during update/delete/insert in a table.

DESCRIPTION

DDL COMMANDS:

- i) Create
 - ii) Alter
 - iii) Drop
- ### DML COMMANDS:
- i) Insert into
 - ii) Select
 - iii) Update
 - iv) Delete

Data Definition Language (DDL)

The data definition language (DDL) is used by the DBA and by database designers to define both schemas when no strict separation of levels is maintained. The DBMS will have a DDL compiler whose function is to process DDL statements in order to identify descriptions of the schema constructs and to store schema description in the DBMS catalog.

DDL commands :

i) Create : Used to create tables, views, schema and others in the database.

Syntax : CREATE TABLE <table name> (column-name
(column-width) (?constraints)).

Column-name datatype (column-width)
(?constraints);

Ex : CREATE TABLE Explain
(ename varchar(10) not null,
SSN number(5)
Address varchar(20));

ii) ALTER: This command is used to alter the table by adding attributes, deleting and modifying attribute name, changing datatype of an attribute etc.

Ex: To add an attribute to the already existing

i) table

ALTER TABLE EMP ADD Phone number(6);

Ex: To change name of an attribute

ii) ALTER TABLE EMP RENAME Address To Add;

Ex: DROP: This command is used to drop the table, view or schema.

iii) DROP TABLE EMP;

This command will drop emp table along with all the table which refers to the EMP table.

Ex ii) : DROP VIEW BC;

The view BC will be dropped out.

DML Commands

i) INSERT: INSERT is used to add a single tuple (row) into a relation(table). We must specify the relation name and a list of values for the tuples. The values should be listed in the same order in which the corresponding attributes are specified in the create table command.

Ex: INSERT INTO EMP VALUES ('Rex', '100', 'xyz');

To insert attribute values only for few attributes we can use:

INSERT INTO EMP(Fname, Lname, Dno, SSN) VALUES
('Richard', 'Manjula', '45', 'DS 123');

ii) SELECT : This command is used to retrieve the data from database.

Ex: `SELECT * FROM EMP;`

Fname	Lname	Dno	SSn
Richard	Maeina	5	DS123

iii) UPDATE : This command is used to modify attributes values of one or more selected tuples. UPDATE command selects the tuples to be modified from a single relation. An additional SET clause in the update command specifies the attributes to be modified and their new values.

Ex: `UPDATE Project
SET Plocation = 'Bangalore', Dnum = 5
WHERE Pnum = 10;`

iv) DELETE : It removes tuples from a relation. It includes WHERE clause to select the tuples to be deleted. Tuples are explicitly deleted from only one table at a time.

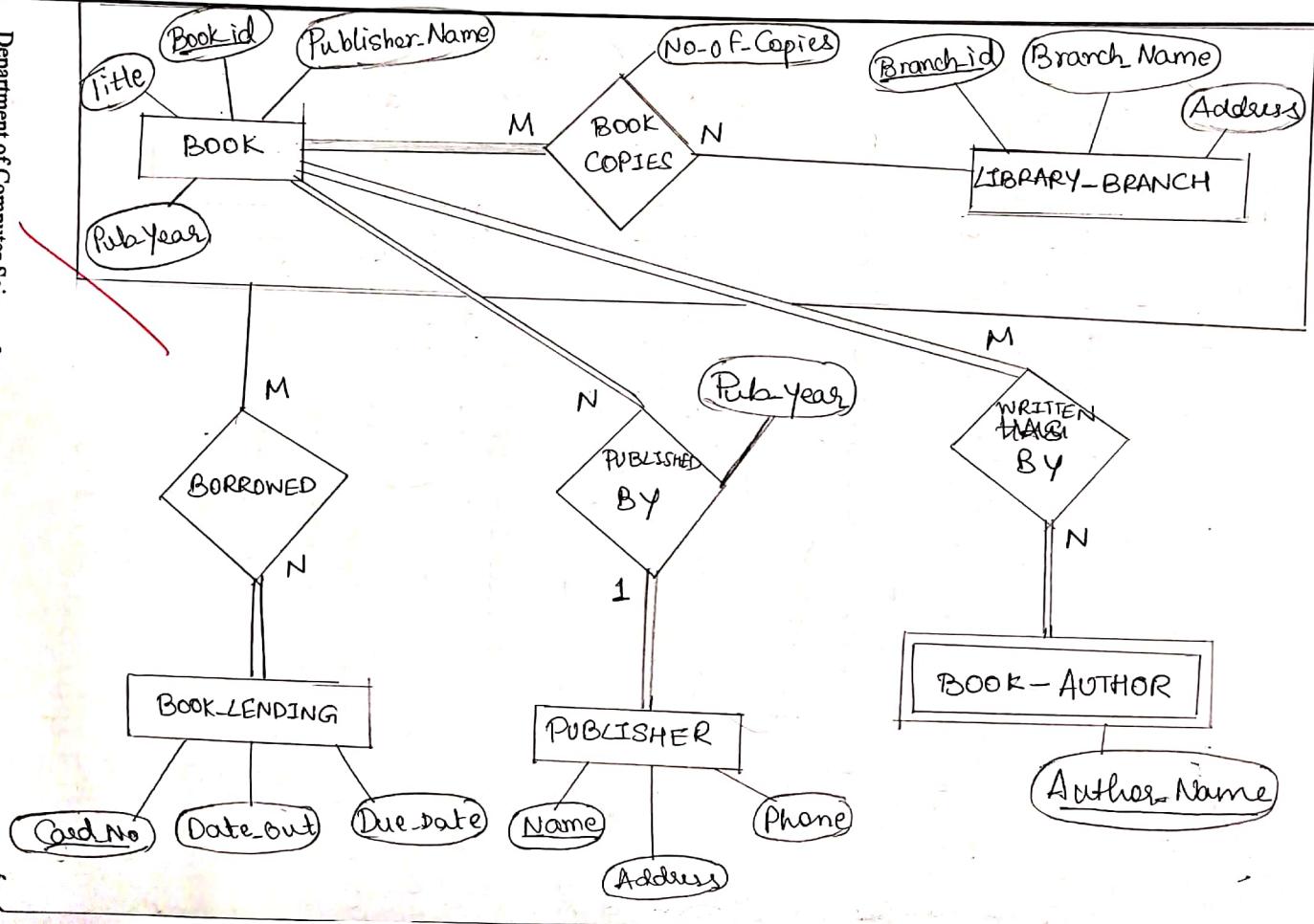
Depending on the number of tuples selected by the condition in the where clause. Zero one or several tuples can be deleted by a single delete command.

Ex: `DELETE FROM EMP WHERE Lname = "Smith";`
// Single row deletion.

~~`DELETE FROM EMP WHERE Dno = 5;`~~
// Multiple row deletion

~~`DELETE FROM EMP;`~~
// all tuples are deleted.

ER-DIAGRAM



ER - RELATIONAL MAPPING

BOOK

Book-id	Title	Publisher_Name	Pub-Year
---------	-------	----------------	----------

BOOK_COPIES

Book-id	Branch_id	No-of_copies
---------	-----------	--------------

PUBLISHER

Name	Address	Phone
------	---------	-------

LIBRARY-BRANCH

Branch_id	Branch_Name	Address
-----------	-------------	---------

BOOK_AUTHORS

Book_id	Author_Name
---------	-------------

BOOK-LENDING

Book_id	Branch_id	Card-No	Date-out	Due-Date
---------	-----------	---------	----------	----------

2: ORDER Database

DATE: 23/11/21

Consider the following schema for Order Database:

SALESMAN (Salesman_id, Name, City, Commission)

CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id)

ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

Write SQL queries to

1. Count the customers with grades above Bangalore's average.
2. Find the name and numbers of all salesman who had more than one customer.
3. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his Order must also be deleted.

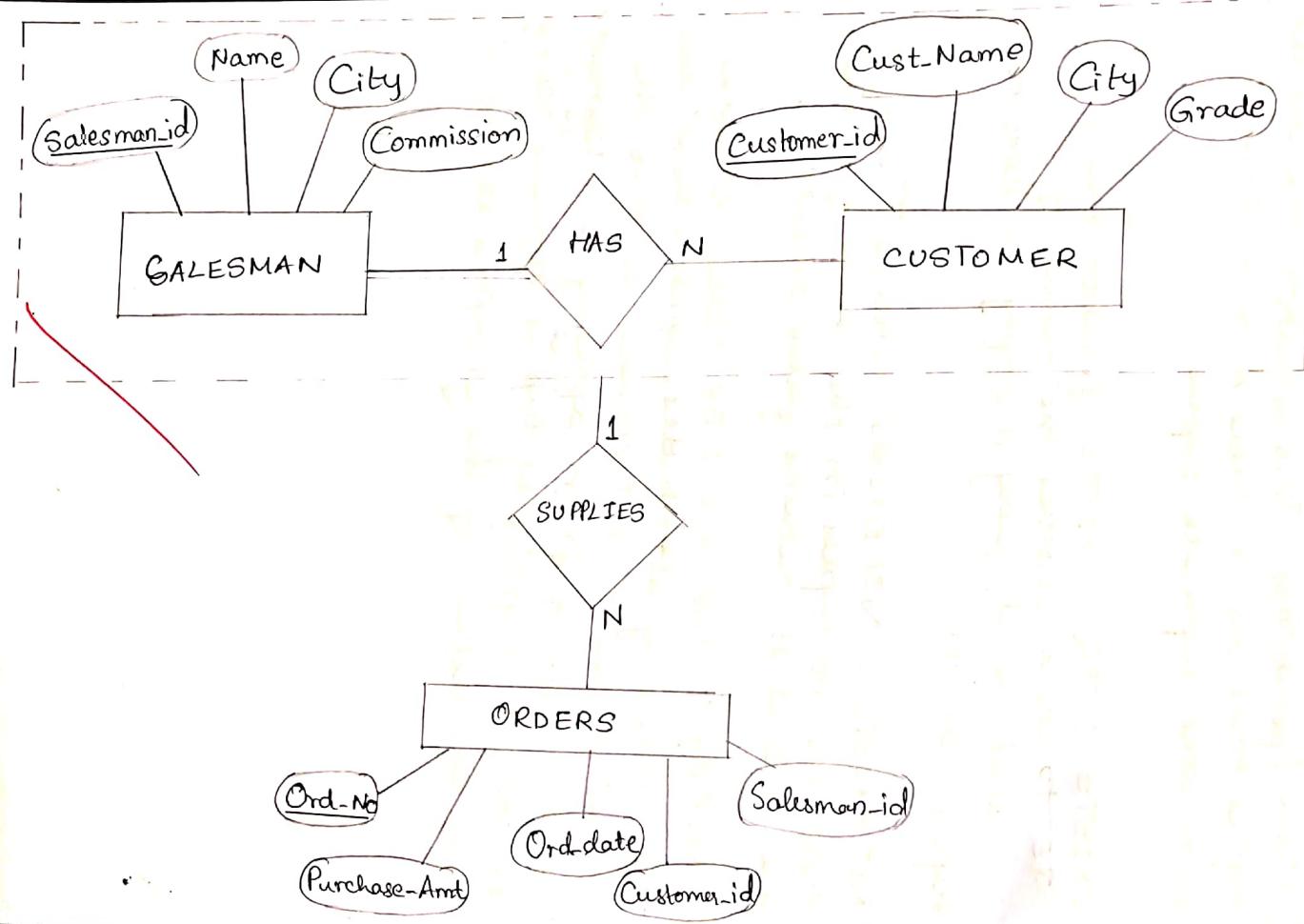
Description

- 1) COUNT : It will return total number of records.
 - 2) SUM : It returns the sum of the values present in the attribute.
 - 3) MAXIMUM : It returns maximum value present in a particular attribute.
 - 4) MINIMUM : It will return minimum value present in particular attribute .
 - 5) HAVING : Having filters records that work on summarized GROUP BY results.
- HAVING applies to summarized group records whereas WHERE applies to individual records.
- Only the group that meet the HAVING criteria will be returned.
- HAVING requires a GROUP BY clause to be present.
- WHERE and HAVING can be in the same query.

DESCRIPTION

- 6) UNION Operation(s) : This operation includes all tuples that are in table A or B. It also eliminates duplicate tuples.
- 7) EXISTS : The EXISTS function is used in SQL to check whether the result of a correlated nested query is empty (contains no tuples) or not.
- 8) NOT EXISTS : NOT EXISTS returns True if there are no tuples in the result of nested queries and it returns false otherwise.
- 9) GROUP By : The GROUP By clause specifies the grouping attributes, which should also appear in the SELECT clauses, so that the value resulting from applying each aggregate function to a group of tuples appears along with the values of the grouping attribute(s).

ER - DIAGRAM



ER - RELATIONAL MAPPING

SALESMAN

Salesman-id	Name	City	Commission

CUSTOMER

Customer-id	Cust-Name	City	Grade	Salesman-id

ORDER

Order-id	Ord-date	Purchase Amt	Customer-id	Salesman-id

DATE: 14-12-21

3: MOVIE Database

Consider the schema for Movie Database:

ACTOR (Act_id, Act_Name, Act_Gender)
 DIRECTOR (Dir_id, Dir_Name, Dir_Phone)
 MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)
 MOVIE_CAST (Act_id, Mov_id, Role)
 RATING (Mov_id, Rev_Stars)

Write SQL queries to

1. List the titles of All movies directed by 'Hitchcock'.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by 'Steven Spielberg' to 5.

Entity: An entity may be an object with a physical existence [for ex. a person, car] or it may be an object with a conceptual existence [ex. a company, job or a university course]

Entity set: An entity is an object of entity type and a set of all entities is called as an entity set. For ex, E1 is an entity having entity type student and set of all students is called an entity set.

Attributes: Attributes are the properties which define the entity type. Ex. Roll-No, Name, Age, Address are the attributes which define the entity type student.

1) Key attribute: The attribute which uniquely identifies an entity in the entity set is called key attribute.

2) Composite attributes: An attribute composed of many other attributes are called as composite attributes.

DESCRIPTION

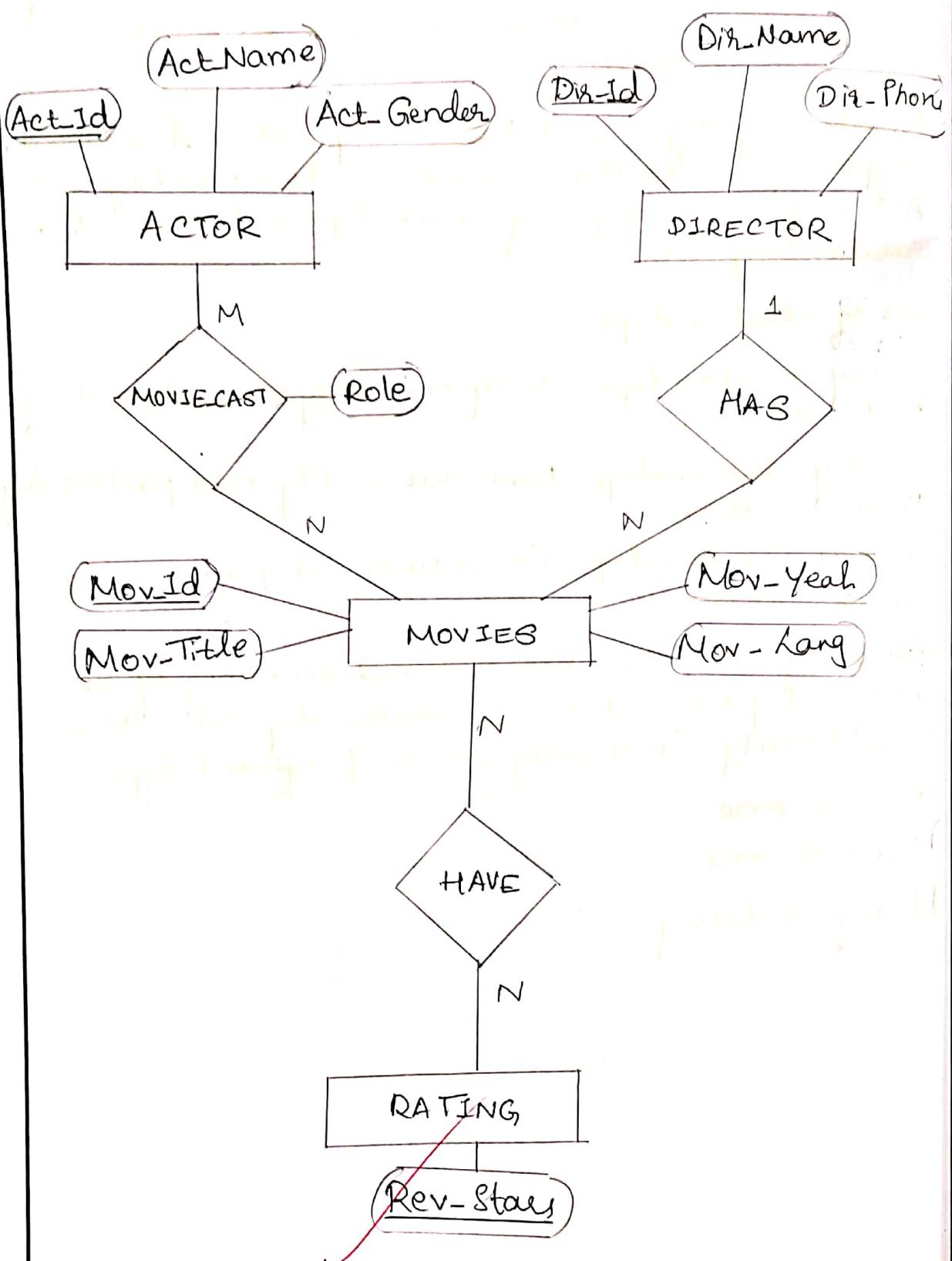
- 3) Multivalued Attributes : An attribute consisting of more than one value for a given entity.
- 4) Relationship type and Relationship set : A relationship type represents the association between entity types. A set of relationships of some type is known as a relationship set.

Types of relationships :

- 1) Unary relationships - only one entity set participating in a relation.
- 2) Binary relationship - have two entity sets participating in a relation.
- 3) n-ary relationship - n entities set participating in a relation.

Cardinality : The number of times an entity of an entity set participates in a relationship set is known as cardinality. Cardinality can be of different types !

- i) One to one
- ii) Many to one
- iii) Many to Many.

ER-DIAGRAM

ER - RELATIONAL MAPPING

ACTOR

<u>Act-id</u>	Act-Name	Act-Gender
---------------	----------	------------

DIRECTOR

<u>Dir-id</u>	Dir-Name	Dir-Phone
---------------	----------	-----------

MOVIES

<u>Mov-id</u>	Mov-Title	Mov-Year	Mov-Lang	Dir-Id
---------------	-----------	----------	----------	--------

CAST

<u>Act-id</u>	<u>Mov-id</u>	Role
---------------	---------------	------

RATING

<u>Mov-id</u>	Rev-Stars
---------------	-----------

DATE: ?

4: COLLEGE Database

Consider the schema for College Database:

STUDENT (USN, SName, Address, Phone, Gender)

SEMSEC (SSID, Sem, Sec)

CLASS (USN, SSID)

SUBJECT (Subcode, Title, Sem, Credits)

IA_MARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

1. List all the student details studying in fourth semester 'C' section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a VIEW of Test1 marks of student USN '1BI15CS101' in all subjects.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:

If FinalIA = 17 to 20 then CAT = 'Outstanding'

If FinalIA = 12 to 16 then CAT = 'Average'

If FinalIA < 12 then CAT = 'Weak'

Give these details only for 8th semester A, B, and C section students.

SQL Datatypes

SQL datatype is used to define the values that a column can contain. Every column is required to have a name and datatype in the database table.

Datatypes of SQL :1) Binary datatypes :

- a) binary : It has a maximum length of 8000 bytes. It contains fixed length binary data.
- b) varbinary : It has a maximum length of 8000 bytes. It contains variable length binary data.
- c) image : It has a maximum length of 2,147,483,647 bytes. It contains variable length binary data.

2) Approximate Datatypes :

- a) float - It is used to specify a floating point value.
Range - -1.79×10^{-308} to 1.79×10^{308}

DESCRIPTION

b) real

It specifies a single precision floating point number
Range -3.40×10^{38} to $+3.40 \times 10^{38}$

3) Exact numeric datatypes -

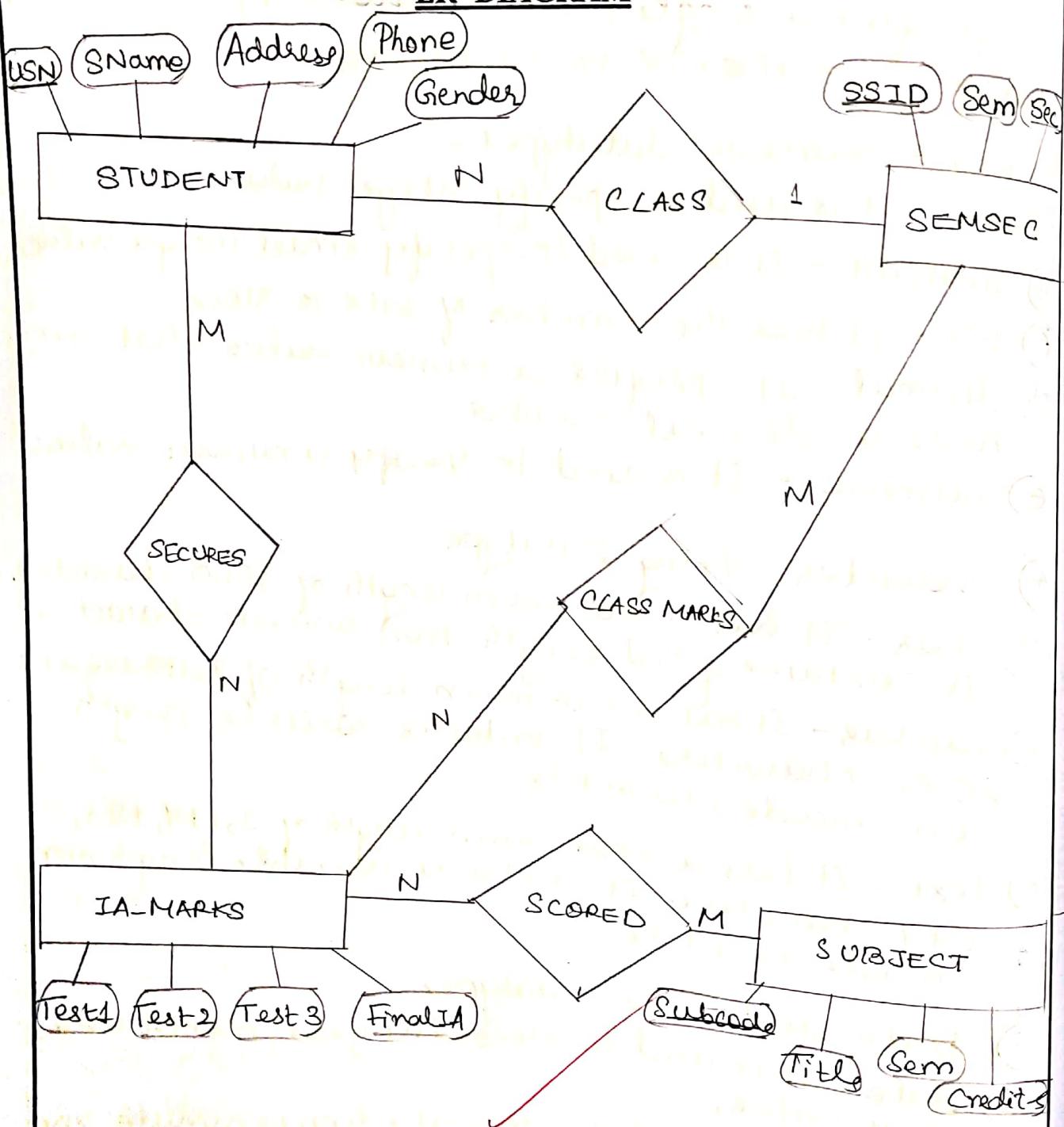
- int - It is used to specify integer values
- smallint - It is used to specify small integer value.
- bit - It has the number of bits to store.
- decimal - It specifies a numeric value that can have a decimal number.
- numeric - It is used to specify a numeric value.

4) Character String Datatype

- char - It has a maximum length of 8000 characters. It contains fixed length non unicode characters.
- varchar - It has a maximum length of 2,147,483,647 characters. It contains variable length non unicode characters.
- text - It has a maximum length of 2,147,483,647 characters. It contains variable length non unicode characters.

5) Date and Time Datatypes

- date - It is used to store the year, month and days value.
- time - It is used to store the hour, minute and second values.
- ~~timestamp - It stores the year, month, day, hour, minute and second value.~~

ER-DIAGRAM

ER - RELATIONAL MAPPING