Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)
School of Computing
B.Tech. – Computer Science and Engineering

VTR UGE2021- (CBCS)

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211CS207

Course Name : Database Management Systems

Slot No : S4-L5

DBMS TASK - 1 REPORT

Title: Conceptual Design through FTR

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU30200	24UECS1453	GANESWARA REDDY UPPALAPALLI

STUDENT-BOOK MANAGEMENT SYSTEM

Abstract:

The **Student Book Management System** is a software application designed to efficiently manage and organize the process of issuing, returning, and tracking books within an educational institution. This system aims to replace traditional manual record-keeping methods with a digital solution that enhances accuracy, accessibility, and time efficiency. It provides functionalities for students to search and request books, while administrators or librarians can manage book inventories, monitor availability, and maintain student borrowing records. The system also helps prevent data loss and reduces the chances of human error by maintaining a secure and centralized database. Overall, the Student Book Management System improves library operations, promotes better resource management, and supports an organized learning environment.

Aim:

Using database design methodology and ER modeling, design an Entity Relationship Diagram (ERD) that satisfies the following tasks:

- Identifying entities
- Identifying attributes
- Determining relationships and cardinality
- Defining relations with keys and constraints
- Creating the ER/EER diagram

1.a Identifying the Entities

1. Student
2. Book
3. Author
4. Publisher
5. Librarian
6. IssueRecord

1.b Identifying the Attributes

1.b.1 Student

(StudentID, Name, Department, Email, Contact_No)

1.b.2 Book

(BookID, Title, AuthorID, PublisherID, ISBN, Category, Quantity)

1.b.3 Author

(AuthorID, Name, Country)

1.b.4 Publisher

(PublisherID, Name, Address, Contact_No)

1.b.5 Librarian

(LibrarianID, Name, Email, Contact_No)

1.b.6 IssueRecord

(IssueID, StudentID, BookID, Issue_Date, Return_Date, Fine)

1.c Identification of Relationships, Cardinality, and Type

Student–IssueRecord Relationship:

One student can have multiple issue records → *One-to-Many (1:N)*.

Book–IssueRecord Relationship:

One book can be issued multiple times, but each issue record refers to one book → *One-to-Many (1:N)*.

Author–Book Relationship:

One author can write multiple books, but each book has one author → *One-to-Many (1:N)*.

Publisher–Book Relationship:

One publisher can publish multiple books, but each book is published by one publisher → *One-to-Many (1:N)*.

Librarian–IssueRecord Relationship:

Each issue record is handled by one librarian, but a librarian can manage multiple issue records → *One-to-Many (1:N)*.

1.d Reframing the Relations with Keys and Constraints

1.d.1 Create Table Student

```
CREATE TABLE Student (
  StudentID VARCHAR(10) PRIMARY KEY,
  Name VARCHAR(50),
  Department VARCHAR(30),
  Email VARCHAR(50),
  Contact_No NUMBER
);
```

SQL> desc student;

Name	Null?	Type
STUDENTID	**NOT NULL**	**VARCHAR2(10)**
NAME		**VARCHAR2(50)**
DEPARTMENT		**VARCHAR2(30)**
EMAIL		**VARCHAR2(50)**
CONTACT_NO		**NUMBER**

1.d.2 Create Table Author

```
CREATE TABLE Author (
  AuthorID VARCHAR(10) PRIMARY KEY,
  Name VARCHAR(50),
  Country VARCHAR(30)
);
```

SQL> desc author;

Name	Null?	Type
AUTHORID	NOT NULL	VARCHAR2(10)
NAME		VARCHAR2(50)
COUNTRY		VARCHAR2(30)

1. d.3 Create Table Publisher

```
CREATE TABLE Publisher (
  PublisherID VARCHAR(10) PRIMARY KEY,
  Name VARCHAR(50),
  Address VARCHAR(100),
  Contact_No NUMBER
);
```

```
SQL> desc publisher;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------------
 PUBLISHERID                               NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(50)
 ADDRESS                                            VARCHAR2(100)
 CONTACT_NO                                         NUMBER
```

1. d.4 Create Table Book

```
CREATE TABLE Book (
  BookID VARCHAR(10) PRIMARY KEY,
  Title VARCHAR(100),
  AuthorID VARCHAR(10),
  PublisherID VARCHAR(10),
  ISBN VARCHAR(20),
  Category VARCHAR(30),
  Quantity NUMBER,
  FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),
  FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID)
);
```

SQL> desc book;

Name	Null?	Type
BOOKID	NOT NULL	VARCHAR2(10)
TITLE		VARCHAR2(100)
AUTHORID		VARCHAR2(10)
PUBLISHERID		VARCHAR2(10)
ISBN		VARCHAR2(20)
CATEGORY		VARCHAR2(30)
QUANTITY		NUMBER

1.d.5 Create Table Librarian

```
CREATE TABLE Librarian (
  LibrarianID VARCHAR(10) PRIMARY KEY,
  Name VARCHAR(50),
  Email VARCHAR(50),
  Contact_No NUMBER
);
```

SQL> desc librarian;

Name	Null?	Type
LIBRARIANID	NOT NULL	VARCHAR2(10)
NAME		VARCHAR2(50)
EMAIL		VARCHAR2(50)
CONTACT_NO		NUMBER

1.d.6 Create Table IssueRecord

```
CREATE TABLE IssueRecord (
  IssueID VARCHAR(10) PRIMARY KEY,
  StudentID VARCHAR(10),
  BookID VARCHAR(10),
  LibrarianID VARCHAR(10),
  Issue_Date DATE,
  Return_Date DATE,
  Fine NUMBER,
  FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
  FOREIGN KEY (BookID) REFERENCES Book(BookID),
  FOREIGN KEY (LibrarianID) REFERENCES Librarian(LibrarianID)
);
```

```
SQL> desc issuerecord;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ISSUEID                                   NOT NULL VARCHAR2(10)
 STUDENTID                                          VARCHAR2(10)
 BOOKID                                             VARCHAR2(10)
 LIBRARIANID                                        VARCHAR2(10)
 ISSUE_DATE                                         DATE
 RETURN_DATE                                        DATE
 FINE                                               NUMBER
```

1.e Using Creately, Develop ER/EER Diagram

Entities:

Student, Book, Author, Publisher, Librarian, IssueRecord

Relationships:

Student ↔ IssueRecord (1:N)

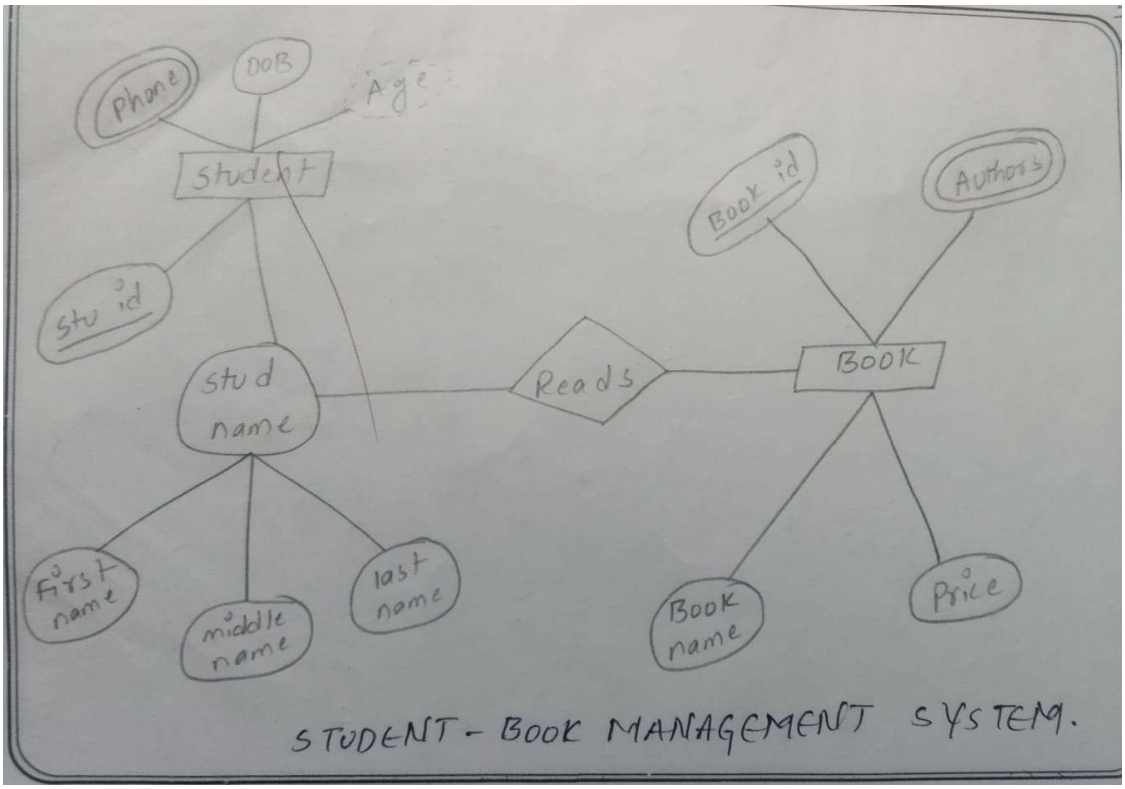
Book ↔ IssueRecord (1:N)

Author ↔ Book (1:N)

Publisher ↔ Book (1:N)

Librarian ↔ IssueRecord (1:N)

1.e. Using creately, develop ER/EER diagram



Result:

Thus, the database design methodology and ER Model for the **Student Book Management System** have been successfully developed and implemented using SQL and ERD tools.