

CS F212-Database Management Systems

Project : Library Management System (DBS_PR_02)

Group No.: 58

Member1: Satyam Srivastava, 2019B1A70188P

Member2: Aditya Choraria, 2019B1A70734P

★ System Requirement Specification (SRS)->

○ Software Configuration

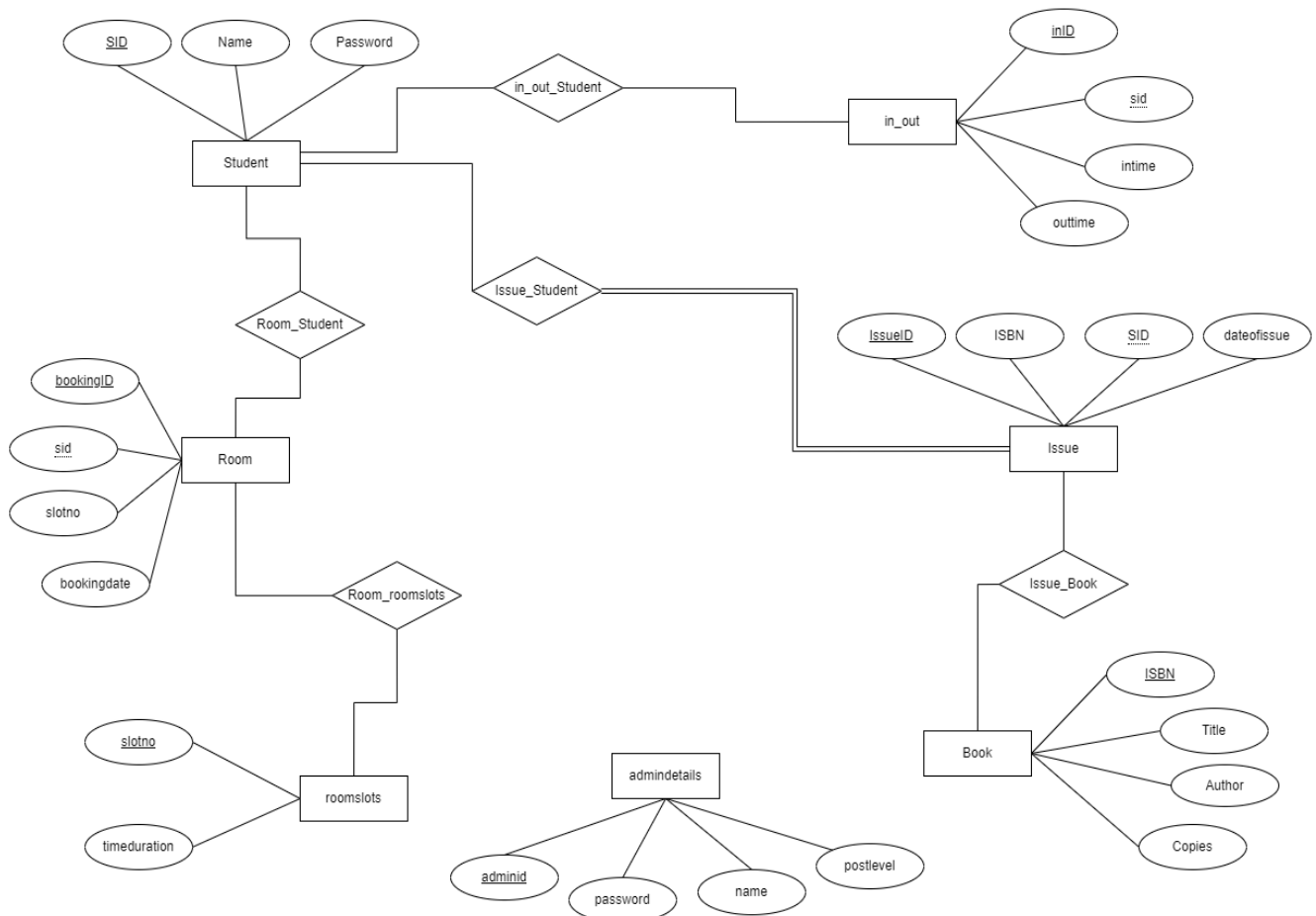
- This Library Management System is developed using MySQL 8.0 by Oracle Corporation, as the backend to store the database.
- Operating System: Linux, Solaris macOS, Windows
- Database: MySQL

○ Hardware Configuration

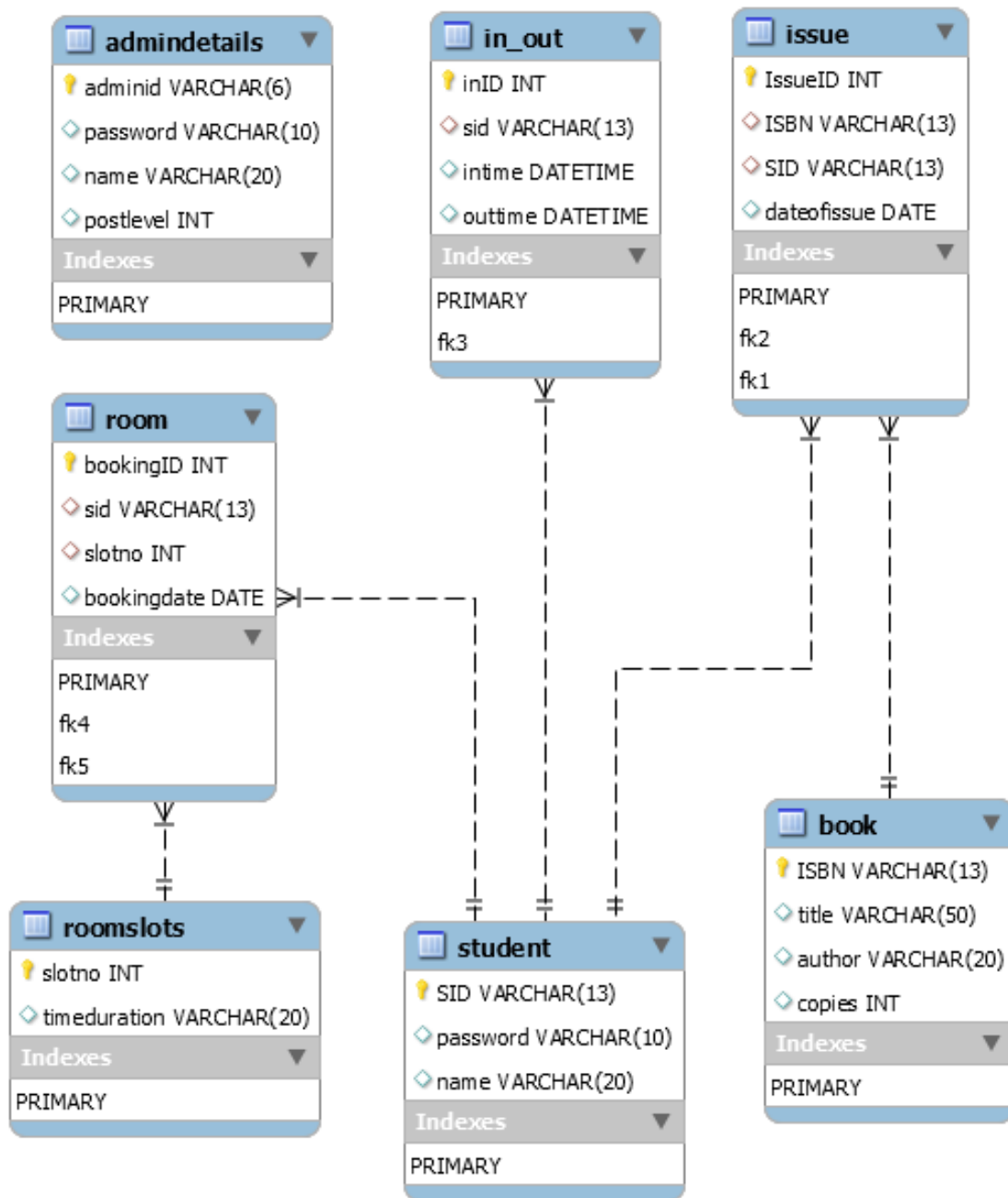
- Processor: Pentium(R)Dual-Core CPU and above
- HardDisk: 40GB or more
- RAM: 256 MB or more

★ System Modelling

○ ER Diagram



- **Schema Design:**



- **Data Normalization:**

The schema used here is in BCNF as there are no partial or transitive functional dependencies in any of the relations (3NF) and for all FDs $A \rightarrow B$, A is a superkey of the relation (BCNF).

- **List of tables required:**

- Book(ISBN, Title, Author, Copies)
- Issue(IssueID, ISBN, sid, dateofissue)
- Student(Sid, password, name)
- IN_OUT(inID, sid, intime, outtime)
- Room(BookingID, SID, SlotNo, BookingDate)
- AdminDetails()
- RoomSlots(SlotNo, TimeDuration)

- **Functions & Procedures:**

- GetAllBooks()-> Prints all the books in library.
- QueryBook()-> Searches for books in the library for a given string by title, author, and ISBN.
- GetAvailableCopies()-> Prints the no. Of copies of the book available with the library currently (excluding the ones that have been issued)
- IssueBook()-> To issue a book from the available set of books in the library.
- CalculateFine()-> To determine the fine that a student would be charged on returning the book. The fine has been calculated as INR 2 per day after 20 days of issuing the book.
- ReturnBook()-> To return a book to the library.
- Login()-> Takes in the ID and Password of the student and returns true if they match with the record.
- Entry()-> Registers the time of the entry of a student in the library to maintain a record of who all are currently in the library.\
- ExitLibrary()-> Registers the time a student leaves the library.
- StudentsInLibrary()-> Returns the number of students currently in the library.
- BrainstormingBooking() -> To book the brainstorming room of the library.
- InsertBook()-> To add a new book to the collection of books in the library or to add the no. of copies of a book already in the library.
- DeleteBook()-> To delete a book from the collection of books in the library.
- AddEmp()-> To register a new employee in bits library.
- AddStud()-> To register a new Student in bits library.

○ **Additional Functions & Procedures:**

- isIssuable(): checks if the library has copies of the book so as to issue it to someone.
- issueBookProc() and issueBookFunc() : helps IssueBook() to assemble all the data required by it.
- isReturnable(): checks if the person returning the book is the same as the one who issued the book and also that the book being returned belongs to the library.
- returnBookProc() and returnBookFunc() : helps ReturnBook() to assemble all the data required by it.
- isAlreadyIn(): checks if a person is already in the library.
- EntryProc() and EntryFunc() : helps Entry() to assemble all the data required by it.
- ExitProc() and ExitFunc() : helps ExitLibrary() to assemble all the data required by it.
- isRoomAvailable(): checks if the room is available for booking in the particular time slot.
- RoomBookingProc() and RoomBookingFunc(): helps BrainstormingBooking() to assemble all the data required by it.
- isAdmin(): checks if the corresponding ID and password belong to that of an employee of the library.
- insertBookProc() and insertBookFunc() : helps InsertBook() to assemble all the data required by it.
- deleteBookProc() and deleteBookFunc() : helps deleteBook() to assemble all the data required by it.
- addEmpProc() and addEmpFunc() : helps AddEmp() to assemble all the data required by it.
- addStudProc() and addStudFunc() : helps AddStud() to assemble all the data required by it.

★ **Operations:**

- To add an employee record to the library database.
 - SQL query
 - CALL `bitslibrary`.`AddEmp`(<{IN aid varchar(6)}>, <{IN password varchar(10)}>, <{IN addid varchar(6)}>, <{IN addpassword varchar(10)}>, <{IN addname varchar(20)}>, <{IN addpostlevel int}>);
 - Attribute description
 - Aid -> ID of an employee who is already registered as an employee of postlevel 0 in the library.

- Password-> Password of the employee of postlevel 0.
 - addid-> ID of the employee to be added.
 - addpassword-> password of the employee to be added.
 - addname -> name of the employee to be added
 - addpostlevel-> postlevel of the employee to be added
- To add a student to the library database.
 - SQL query
 - CALL `bitslibrary`.`AddStud`(<{IN aid varchar(6)}>, <{IN password varchar(10)}>, <{IN addsid varchar(13)}>, <{IN addname varchar(20)}>);
 - Attribute description
 - Aid -> ID of an employee who is already registered as an employee in the library.
 - Password-> Password of the employee.
 - addsid-> ID of the student to be added.
 - addname -> name of the student to be added.
- To add a book to the library database.
 - SQL Query
 - CALL `bitslibrary`.`InsertBook`(<{IN aid varchar(6)}>, <{IN password varchar(10)}>, <{IN ISBN varchar(13)}>, <{IN title varchar(50)}>, <{IN author varchar(50)}>, <{IN copies int}>);
 - Attribute description
 - Aid -> ID of an employee who is already registered as an employee in the library.
 - Password-> Password of the employee.
 - ISBN-> ISBN no, of the book to be added.
 - Title-> Title of the book to be added.
 - Author-> Author of the book to be added.
 - Copies-> No. of copies of the book to be added.
- To delete a book from the library database.
 - SQL Query
 - CALL `bitslibrary`.`DeleteBook`(<{IN aid varchar(6)}>, <{IN password varchar(10)}>, <{IN ISBN varchar(13)}>, <{IN title varchar(50)}>, <{IN author varchar(50)}>, <{IN copies int}>);
 - Attribute description
 - Aid -> ID of an employee who is already registered as an employee in the library.
 - Password-> Password of the employee.
 - ISBN-> ISBN no, of the book to be deleted.
 - Title-> Title of the book to be deleted

- Author-> Author of the book to be deleted.
 - Copies-> No. of copies of the book to be deleted.
- To get the record of all the books in the library database.
 - SQL query
 - CALL `bitslibrary`.`GetAllBooks`();
 - Attribute description
 - None
- To search for books using a given string
 - SQL query
 - CALL `bitslibrary`.`QueryBook`(<{IN searchstring VARCHAR(50)}>);
 - Attribute description
 - searchstring-> the string that is to be used for searching
- To issue a book from the library
 - SQL query
 - CALL `bitslibrary`.`IssueBook`(<{IN ISBNno VARCHAR(13)}>, <{IN sid varchar(13)}>, <{IN password varchar(10)}>);
 - Attribute description
 - ISBNno-> ISBN of the book to be issued
 - sid-> ID of the student who is issuing the book
 - password-> password of the student issuing the book.
- To get all the books currently in issue by a particular student
 - SQL query
 - CALL `bitslibrary`.`GetIssuedBooks`(<{IN sid varchar(13)}>);
 - Attribute description
 - sid-> ID of the student whose list of books issued is to be determined.
- To return the book previously issued by a student
 - SQL query
 - CALL `bitslibrary`.`ReturnBook`(<{IN sid varchar(13)}>, <{IN issueID int}>);
 - Attribute description
 - sid-> ID of the student returning the book.
 - issueID-> ID of the issue of the book which he/she is returning.
- To calculate the fine applicable to a student
 - SQL query
 - Select `bitslibrary`.`CalculateFine`(<{sid varchar(13)}>, <{IssueID int}>)
 - Attribute description
 - sid-> ID of the student whose fine needs to be calculated

- IssueID-> ID of the issue of the book for which the fine needs to be calculated
- To register the in-time of a student
 - SQL query
 - CALL `bitslibrary`.`Entry`(<{IN sid varchar(13)}>, <{IN password varchar(10)}>);
 - Attribute description
 - sid-> ID of the student who is coming in the library.
 - password-> password of the student who is coming in the library
- To register the ou-time of a student
 - SQL query
 - CALL `bitslibrary`.`ExitLibrary`(<{IN sid varchar(13)}>, <{IN password varchar(10)}>);
 - Attribute description
 - sid-> ID of the student who is going out of the library.
 - password-> password of the student who is going out of the library
- To know the slot number for various time durations
 - SQL query
 - CALL `bitslibrary`.`GetBrainstormingRoomSlots`();
 - Attribute description
 - None
- To book the brainstorming room
 - SQL query
 - CALL `bitslibrary`.`BrainstormingBooking`(<{IN sid varchar(13)}>, <{IN password varchar(10)}>, <{IN slotno int}>, <{IN bookingdate varchar(10)}>);
 - Attribute description
 - sid-> ID of the student who is booking the room
 - password-> password of the student who is booking the room
 - slotno-> slot no. of the time period for which the room is to be booked
 - bookingdate-> the date for which the room is to be booked.
-