Project – MySQL

Name – Anandu O A

Library Management System

-- Topic : Library Management System

-- You are going to build a project based on Library Management System.

-- It keeps track of all information about books in the library, their cost, status and total number of books available in the library.

-- Create a database named library and following TABLES in the database:

-- 1. Branch 2. Employee 3. Books 4. Customer 5. IssueStatus 6. ReturnStatus

create database library;

use library;

-- Attributes for the tables:

-- 1. Branch Branch\_no - Set as PRIMARY KEY Manager\_Id Branch\_address Contact\_no

create table Branch(

Branch\_no int Primary key,

Manager\_Id varchar(10),

Branch\_address varchar(100),

Contact\_no bigint

);

-- 2. Employee Emp\_Id – Set as PRIMARY KEY Emp\_name Position Salary Branch\_no - Set as FOREIGN KEY

-- and it refer Branch\_no in Branch table

create table Employee(

Emp\_Id int Primary Key,

Emp\_Name varchar(30),

Position varchar(30),

Salary decimal(10,2),

Branch\_no int,

foreign key (Branch\_no) references Branch(Branch\_no)

);

-- 3. Books ISBN - Set as PRIMARY KEY Book\_title Category Rental\_Price Status

-- [Give yes if book available and no if book not available] Author Publisher

create table Books(

ISBN int primary key,

Book\_Title varchar(30),

Category varchar(30),

Rental\_Price decimal(10,2),

Status enum('Yes','No') default 'Yes',

Author varchar(30),

Publisher varchar(30)

);

-- 4. Customer Customer\_Id - Set as PRIMARY KEY Customer\_name Customer\_address Reg\_date

create table customer(

Customer\_Id int primary key,

Customer\_Name varchar(30),

Customer\_address varchar(100),

Reg\_date date

);

-- 5. IssueStatus Issue\_Id - Set as PRIMARY KEY Issued\_cust –

-- Set as FOREIGN KEY and it refer customer\_id in CUSTOMER table Issued\_book\_name Issue\_date Isbn\_book –

-- Set as FOREIGN KEY and it should refer isbn in BOOKS table

create table IssueStatus (

Issue\_Id int primary key,

Issued\_Cust int,

Issued\_book\_name varchar(30),

Issue\_date date,

Isbn\_book int,

foreign key(Issued\_cust) references customer(customer\_id),

foreign key(Isbn\_book) references books(Isbn)

);

-- 6. ReturnStatus Return\_Id - Set as PRIMARY KEY Return\_cust Return\_book\_name Return\_date Isbn\_book2 -

-- Set as FOREIGN KEY and it should refer isbn in BOOKS table

create table ReturnStatus(

Return\_Id int primary key,

Return\_cust int,

Return\_book\_name varchar(30),

Return\_date date,

Isbn\_book2 int,

foreign key(return\_cust) references customer(customer\_id),

foreign key(isbn\_book2) references books(isbn)

);

-- INSERT 10 rows to the above tables each

-- 1. Branch Table

INSERT INTO Branch (Branch\_no, Manager\_Id, Branch\_address, Contact\_no) VALUES

(1, 101, '123 Library Lane, Downtown', '1234567890'),

(2, 102, '456 Book Blvd, Midtown', '2345678901'),

(3, 103, '789 Read Road, Uptown', '3456789012'),

(4, 104, '321 Shelf Street, Suburbia', '4567890123'),

(5, 105, '654 Knowledge Ave, Eastside', '5678901234'),

(6, 106, '987 Wisdom Way, Westside', '6789012345'),

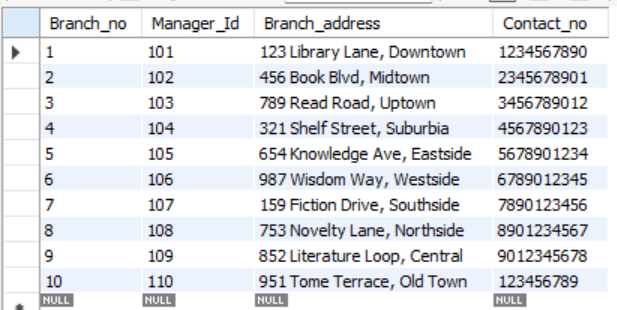
(7, 107, '159 Fiction Drive, Southside', '7890123456'),

(8, 108, '753 Novelty Lane, Northside', '8901234567'),

(9, 109, '852 Literature Loop, Central', '9012345678'),

(10, 110, '951 Tome Terrace, Old Town', '0123456789');

select \* from branch;



-- 2. Employee Table

INSERT INTO Employee (Emp\_Id, Emp\_name, Position, Salary, Branch\_no) VALUES

(201, 'Alice Smith', 'Librarian', 55000, 1),

(202, 'Bob Johnson', 'Assistant Librarian', 48000, 2),

(203, 'Charlie Lee', 'Librarian', 60000, 3),

(204, 'Diana Chang', 'Technician', 45000, 4),

(205, 'Eve Davis', 'Librarian', 53000, 5),

(206, 'Frank Wilson', 'Assistant Librarian', 49000, 6),

(207, 'Grace Kim', 'Manager', 70000, 7),

(208, 'Hank White', 'Assistant Librarian', 46000, 8),

(209, 'Ivy Brown', 'Technician', 43000, 9),

(210, 'Jack Black', 'Librarian', 58000, 10);

select \* from employee;

A screenshot of a computer

Description automatically generated

-- 3. Books Table

INSERT INTO Books (ISBN, Book\_title, Category, Rental\_Price, Status, Author, Publisher) VALUES

(1001, 'Computer Programming', 'Programming', 30.00, 'yes', 'Donald Knuth', 'Addison-Wesley'),

(1002, 'Introduction to Algorithms', 'Programming', 28.00, 'yes', 'Thomas H. Cormen', 'MIT Press'),

(1003, 'Design Patterns', 'Software Engineering', 25.00, 'no', 'Erich Gamma', 'Addison-Wesley'),

(1004, 'Clean Code', 'Programming', 27.00, 'yes', 'Robert C. Martin', 'Prentice Hall'),

(1005, 'The Pragmatic Programmer', 'Programming', 26.00, 'yes', 'Andrew Hunt', 'Addison-Wesley'),

(1006, 'Artificial Intelligence', 'AI', 35.00, 'no', 'Stuart Russell', 'Pearson'),

(1007, 'Deep Learning', 'AI', 40.00, 'yes', 'Ian Goodfellow', 'MIT Press'),

(1008, 'Digital Fortress', 'Fiction', 15.00, 'yes', 'Dan Brown', 'St. Martin\'s Press'),

(1009, '1984', 'Dystopian', 20.00, 'no', 'George Orwell', 'Secker & Warburg'),

(1010, 'Brief History of Humankind', 'History', 22.00, 'yes', 'Yuval Noah Harari', 'Harper');

select \* from books;

A screenshot of a computer screen

Description automatically generated

-- 4. Customer Table

INSERT INTO Customer (Customer\_Id, Customer\_name, Customer\_address, Reg\_date) VALUES

(301, 'John Doe', '101 Main St', '2021-03-15'),

(302, 'Jane Roe', '202 Maple Ave', '2020-07-20'),

(303, 'Mike Brown', '303 Oak St', '2022-01-10'),

(304, 'Emily White', '404 Pine Blvd', '2021-11-25'),

(305, 'David Black', '505 Cedar Rd', '2023-06-30'),

(306, 'Sarah Green', '606 Birch Ln', '2022-09-18'),

(307, 'Chris Blue', '707 Elm St', '2023-04-04'),

(308, 'Alex Gray', '808 Aspen Ct', '2020-10-12'),

(309, 'Lisa Purple', '909 Redwood Dr', '2023-05-22'),

(310, 'Paul Orange', '1010 Willow Way', '2021-02-28');

select \* from customer;

A screenshot of a computer

Description automatically generated

-- 5. IssueStatus Table

INSERT INTO IssueStatus (Issue\_Id, Issued\_cust, Issued\_book\_name, Issue\_date, Isbn\_book) VALUES

(401, 301, 'Computer Programming', '2023-06-05', 1001),

(402, 302, 'Introduction to Algorithms', '2023-05-18', 1002),

(403, 303, 'Clean Code', '2023-04-22', 1004),

(404, 304, 'The Pragmatic Programmer', '2023-06-15', 1005),

(405, 305, 'Deep Learning', '2023-07-01', 1007),

(406, 306, 'Digital Fortress', '2023-06-03', 1008),

(407, 307, 'Brief History of Humankind', '2023-06-20', 1010),

(408, 308, '1984', '2023-01-19', 1009),

(409, 309, 'Artificial Intelligence', '2023-06-14', 1006),

(410, 310, 'Design Patterns', '2023-07-07', 1003);

select \* from IssueStatus;

A screenshot of a computer

Description automatically generated

6. ReturnStatus Table

INSERT INTO ReturnStatus (Return\_Id, Return\_cust, Return\_book\_name, Return\_date, Isbn\_book2) VALUES

(501, 301, 'Computer Programming', '2023-06-25', 1001),

(502, 302, 'Introduction to Algorithms', '2023-06-30', 1002),

(503, 303, 'Clean Code', '2023-07-01', 1004),

(504, 304, 'The Pragmatic Programmer', '2023-07-05', 1005),

(505, 305, 'Deep Learning', '2023-07-10', 1007),

(506, 306, 'Digital Fortress', '2023-06-20', 1008),

(507, 307, 'Brief History of Humankind', '2023-07-14', 1010),

(508, 308, '1984', '2023-06-25', 1009),

(509, 309, 'Artificial Intelligence', '2023-06-29', 1006),

(510, 310, 'Design Patterns', '2023-07-18', 1003);

select \* from ReturnStatus;

A screenshot of a computer

Description automatically generated

-- Display all the tables and Write the queries for the following :

-- 1. Retrieve the book title, category, and rental price of all available books.

select book\_title, category, rental\_price

from books

where status = 'yes';

A screenshot of a computer

Description automatically generated

-- 2. List the employee names and their respective salaries in descending order of salary.

select Emp\_Name, Salary

from employee

order by salary asc;

A screenshot of a computer

Description automatically generated

-- 3. Retrieve the book titles and the corresponding customers who have issued those books.

select b.book\_title, c.customer\_name

from issuestatus i

join books b on i.isbn\_book = b.isbn

join customer c on i.issued\_cust = c.customer\_id;

A screenshot of a computer program

Description automatically generated

-- 4. Display the total count of books in each category.

select Category, count(\*) as Count\_of\_Books

from books

group by category;

A screenshot of a computer

Description automatically generated

-- 5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.

select Emp\_Name, Position

from employee

where Salary > 50000;

A screenshot of a computer

Description automatically generated

-- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.

select c.Customer\_Name, c.Reg\_Date

from customer c

left join issuestatus i on c.customer\_id = i.issued\_cust

where c.reg\_date < '2022-01-01';

A screenshot of a computer

Description automatically generated

-- 7. Display the branch numbers and the total count of employees in each branch.

select branch\_no, count(\*) as Count

from employee

group by branch\_no;

A screenshot of a computer

Description automatically generated

-- 8. Display the names of customers who have issued books in the month of June 2023.

select c.customer\_name

from issuestatus i

join customer c on i.issued\_cust = c.customer\_id

where month(issue\_date) = 6 and year(issue\_date) = 2023;

A screenshot of a computer

Description automatically generated

-- 9. Retrieve book\_title from book table containing history.

select Book\_title

from books

where book\_title like '%history%';

A screenshot of a computer

Description automatically generated

-- 10.Retrieve the branch numbers along with the count of employees for branches having more than 5 employees

select branch\_no, count(\*) as 'Total count of employees'

from employee

group by branch\_no

having count(\*) > 5;

-- 11. Retrieve the names of employees who manage branches and their respective branch addresses.

select e.emp\_name, b.branch\_address

from employee e

join branch b on e.emp\_id = b.manager\_id;

-- 12. Display the names of customers who have issued books with a rental price higher than Rs. 25.

SELECT C.Customer\_name

FROM IssueStatus I

JOIN Books B ON I.Isbn\_book = B.ISBN

JOIN Customer C ON I.Issued\_cust = C.Customer\_Id

WHERE B.Rental\_Price > 25;

A screenshot of a computer

Description automatically generated