Open Library

<https://watch.screencastify.com/v/ZGE6WiXQPZruWwCNLRiz>

Business Problem:

1. Help identify books and it’s associated authors
2. Help members log in and view the current available books and reserve them
3. Help members View Past reservations
4. Help streamline the process of reservations of the books.

Key Stakeholders:

1. Members
2. Employees

Key Information:

1. Book Information
2. Book\_title, type of book, authors of different books
3. Member information and reservation information
4. Who, what, when, what type of books reserved and who reserved Employees
5. Who are the employees, how many student employees, hours worked

Why Open Library was selected as a project ?  
- Library databases offer options to quickly limit or expand your search to find the articles you need. In short, library databases helps to quickly find relevant scholarly information you can use in research papers or other course projects. Library databases help users access information more quickly and information is more stable. Library database helps to streamline the process of reservation of books etc.

Data Entities, Attributes and PK/FK relationship:

Table

Description automatically generatedTable

Description automatically generated

Conceptual Model:

Diagram

Description automatically generated

Logical Model Diagram:

Graphical user interface, table, Excel

Description automatically generated

User Stories:

* As an Member I should be able to…

1. View all the books.
2. View all the availability of books
3. View past reservations
4. Reserve new books

* As an Employee I should be able to…

1. See how much I worked within a given week
2. Also enter in how much hours you worked for a given day

Basic Layout of the application screens:

Main Screen:

1. The data entered is by both the members as well as the employees, which has been sectioned out
2. The data entered will be the unique email address that the users have

Text

Description automatically generatedGraphical user interface, application, website

Description automatically generated

Member Screen:

1. The data is entered in by the members
2. The members look for the books they want to reserve
3. The members should be able to view their past reservations
4. Data input would be the selection of books

Graphical user interface, website

Description automatically generatedGraphical user interface, table

Description automatically generatedTable

Description automatically generated with low confidenceGraphical user interface, application

Description automatically generated

Employees Screen:

1. TimeCard for the employees
2. Employees will be able to Enter in the date of their work, the hours worked and the supervisor they worked under.
3. Only two departments are available: Only two Supervisors

Graphical user interface, application

Description automatically generatedApplication, Teams

Description automatically generated

External Model Flow:

Diagram

Description automatically generated

* SQL Up/Down script to implement the internal model with initial data.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

if not exists (select \* from sys.databases where name = 'project\_659')

    create database project\_659

go

use project\_659

go

-- DOWN

drop table if exists employees

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

    where CONSTRAINT\_NAME = 'fk\_reservations\_reservation\_book\_id')

    alter table reservations drop constraint fk\_reservations\_reservation\_book\_id

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

    where CONSTRAINT\_NAME = 'fk\_reservations\_reservation\_member\_id')

    alter table reservations drop constraint fk\_reservations\_reservation\_member\_id

drop table if exists reservations

drop table if exists members

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

    where CONSTRAINT\_NAME = 'fk\_book\_authors\_book\_author\_author\_id')

    alter table book\_authors drop constraint fk\_book\_authors\_book\_author\_author\_id

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

    where CONSTRAINT\_NAME = 'fk\_book\_authors\_book\_author\_book\_id')

    alter table book\_authors drop constraint fk\_book\_authors\_book\_author\_book\_id

drop table if exists book\_authors

drop table if exists writers

drop table if exists books

GO

-- UP Metadata

create table books (

    book\_id int not null,

    book\_title varchar(50) not null,

    book\_category\_name varchar(50) not null,

    book\_status varchar(50) not null,

    constraint pk\_books\_book\_id primary key (book\_id)

)

create table writers (

    writer\_id int identity not null,

    writer\_first\_name varchar(50) not null,

    writer\_last\_name varchar(50) not null,

    constraint pk\_writers\_writer\_id primary key (writer\_id)

)

create table book\_authors (

    book\_author\_book\_id int not null,

    book\_author\_author\_id int not null,

)

alter table book\_authors

    add constraint fk\_book\_authors\_book\_author\_book\_id foreign key (book\_author\_book\_id)

    references books(book\_id)

alter table book\_authors

    add constraint fk\_book\_authors\_book\_author\_author\_id foreign key (book\_author\_author\_id)

    references writers(writer\_id)

create table members(

    member\_id int identity not null,

    member\_firstname varchar(50) not null,

    member\_lastname varchar(50) null,

    member\_joined\_date date not null,

    member\_active\_status varchar(50) not null,

    member\_email\_address varchar(50) null,

    member\_address varchar(50) null,

    constraint pk\_members\_member\_id primary key (member\_id),

    constraint u\_members\_member\_id unique (member\_email\_address),

    constraint u\_members\_memeber\_adress unique (member\_address)

)

create table reservations (

    reservation\_id int identity not null,

    reservation\_member\_id int not null,

    reservation\_book\_id int not null,

    reservation\_date date not null,

    reservation\_return\_date date not null,

    constraint pk\_reservations\_reservation\_id primary key (reservation\_id)

)

alter table reservations

    add constraint fk\_reservations\_reservation\_member\_id foreign key (reservation\_member\_id)

        references members(member\_id)

alter table reservations

    add constraint fk\_reservations\_reservation\_book\_id foreign key(reservation\_book\_id)

        references books(book\_id)

create table employees (

    employee\_id int identity not null,

    employee\_ssn int not null,

    employee\_firstname varchar(50) not null,

    employee\_lastname varchar(50) not null,

    employee\_email varchar(50) not null,

    employee\_hiredate date not null,

    employee\_termdate date null,

    employee\_jobtitle varchar(50) not null,

    employee\_department varchar(50) not null,

    employee\_payrate money null,

    employee\_department\_head varchar(50) not null,

    employee\_hours\_worked int null,

    employee\_pay\_date date not null,

    constraint pk\_employees\_employee\_id primary key (employee\_id),

    constraint u\_employees\_employee\_ssn unique (employee\_ssn)

)

GO

--UP Data

insert into books (book\_id, book\_title, book\_category\_name, book\_status)

values

(1, 'Jianghui','Fiction', 'Available'), --1

(2, 'Crocodile','Non-Fiction', 'Not Available'), --2

(3, 'Snakes','Comic', 'Available'), --3

(4, 'Kangaroo','Fiction', 'Not Available'), --4

(5, 'Oh Life ','Non-Fiction', 'Available'), --5

(6, 'Giorgio Amarni','Fiction', 'Not Available'), --6

(7, 'Oh Jojo ','Non-Fiction', 'Available'), --7

(8, 'Jujutsu Kaisen', 'Fiction', 'Available'),--8

(9, 'Haikyuu', 'Fiction', 'Available'), --9

(10, 'Death Note' , 'Fiction', 'Available'), -- 10

(11, 'How to start your own business', 'Textbook', 'Available'), --11

(12, 'SQL for dummies', 'Textbook', 'Not Available'), --12

(13, 'Course of Theoretical Physics', 'Textbook', 'Available'), --13

(14, 'An introduction to statistical infernce', 'Textbook', 'Available'), --14

(15, 'Python for everybody','Textbook', 'Available'), --15

(16, 'Life of Pi', 'Action and Adventure', 'Not Available'), --16

(17, 'The Three Musketeers', 'Action and Adventure', 'Not Available'), --17

(18, 'The Call of the Wild', 'Action and Adventure', 'Available'), --18

(19, 'Watchmen', 'Comic', 'Not Available') --19

insert into writers (writer\_first\_name, writer\_last\_name)

values

('Maxwell','Guy'),--1

('Stephen','Rieks'),--2

('Mike','White'),--3

('Emily','Burrows'),--4

('Angela','Samson'),--5

('Rosie','loski'),--6

('Sean','Michaels'),--7

('Olivia','Jameson'),--8

('Samuel', 'Jackson'),--9

('Robert', 'Stew'),--10

('Bob', 'Builder'),--11

('Mackenzie', 'Rowe'),--12

('Abhijith', 'Vamadev'),--13

('Trina', 'Trisha'),--14

('Balthazar', 'Necromancer'),--15

('Shniua', 'Legal'),--16

('Lee', 'Sichuan'),--17

('Michael', 'Fudge'),--18

('Jayce', 'Talis'), --19

('Shauna', 'Vayne')--20

insert into book\_authors(book\_author\_book\_id, book\_author\_author\_id)

values

(1, 1),

(2, 1),

(3, 1),

(4, 2),

(5, 2),

(6, 3),

(7, 4),

(7, 5),

(7, 6),

(7, 7),

(7, 8),

(8, 2),

(9, 3),

(10, 4),

(10, 5),

(10, 1),

(11, 4),

(12, 5),

(12, 6),

(12, 7),

(12, 9),

(13, 14),

(13, 13),

(13, 12),

(14, 20),

(14, 19),

(15, 15),

(16, 16),

(17, 1),

(17, 17),

(18, 10),

(18, 18),

(19, 11),

(19, 9),

(19, 10),

(19, 8)

go

insert into members (member\_firstname, member\_lastname, member\_joined\_date, member\_active\_status, member\_email\_address, member\_address)

values

('Abhijith', 'Vamadev', '2020-05-01', 'Active', 'aavamadev@syr.edu', 'Syracuse University'),

('Arsh', 'Singh', '2020-06-02', 'Active', 'arsh@syr.edu', 'Purdue University'),

('Jeffrey', 'Wong', '2020-01-04', 'Active', 'wong@purdue.edu', 'NorthWestern University'),

('Violet', 'Caitlin', '2020-09-15', 'Active', 'violetcaitlin@gmail.com', 'Zaun 201 Street'),

('Jayce', 'Tallis', '2020-10-13', 'Not-Active', 'jaycetallis@gmail.com', 'Magicland 202 Street'),

('Jinx','', '2020-10-15', 'Not-Active', 'jinxcrazy@gmail.com', 'Crazyland 2020 Street' ),

('Himerdinger', '', '2020-10-10', 'Active', 'councilor@gmail.com', 'Upperandlower 201 Street'),

('Caitlin', 'Kirreman', '2020-05-10', 'Active', 'cupcake@gmail.com', 'Piltover 2021 Street'),

('Viktor', 'Evolution', '2020-02-20', 'Active', 'thegreatestevolution@gmail.com', 'LowerCity 202 Street'),

('Akali', 'Tethi', '2019-05-19', 'Not-Active', 'fistofshadow@gmail.com', 'Kinko Order Street'),

('Irelia', 'Firremen', '2019-06-01', 'Active', 'bladedancer@hotmail.com', 'Navori Ionia 505 Street'),

('Draven', 'Darius', '2020-05-05', 'Not-Active', 'itsdraventime@hotmail.com', 'Noxus 200 Street'),

('Shauna', 'Vayne', '2020-02-20', 'Active', 'tumblecity@gmail.com', 'Tumble place 202 street'),

('Master', 'Yi', '2020-03-19', 'Not-Active', 'runfast@yahoo.com', '300 University Avenue'),

('Mel','', '2019-05-20', 'Active', 'influencer@gmail.com', 'Garudian 2020 Navel Place')

go

insert into reservations(reservation\_member\_id, reservation\_book\_id, reservation\_date, reservation\_return\_date)

values

(1, 1, '2020-03-03', '2020-05-03'),

(1, 3, '2020-03-03', '2020-05-03'),

(2, 3, '2020-06-09', '2020-09-09'),

(3, 18, '2020-05-03', '2020-08-03'),

(3, 8, '2020-05-03', '2020-08-03'),

(4, 13, '2020-01-01', '2020-04-01'),

(5,11, '2020-05-03', '2020-08-03'),

(6, 14, '2020-02-03', '2020-05-03'),

(7, 15, '2020-08-03', '2020-11-03'),

(8,10, '2020-04-19', '2020-07-19')

insert into employees(employee\_ssn, employee\_firstname, employee\_lastname, employee\_email, employee\_hiredate, employee\_termdate, employee\_jobtitle, employee\_department, employee\_payrate, employee\_department\_head, employee\_hours\_worked, employee\_pay\_date)

values

(123345556, 'Abhijith', 'Vamadev', 'aavamadev@gmail.com', '2019-01-04', '', 'Manager', 'Administration', 20, 'Abhijith', 105,'2020-01-18'),

(233445567, 'Nyugen', 'Benezal', 'nygugen@gmail.com','2019-02-04', '', 'Manager', 'Catalouging', 20, 'Nyugen', 115, '2020-01-18'),

(234232131, 'Nickson', 'Davidson', 'nickson@gmail.com', '2020-04-05','', 'Student Employee', 'Administration', 12, 'Abhijith', 35, '2020-01-18'),

(459334213, 'Joe', 'Goldburg', 'joe@gmail.com', '2020-04-03', '', 'Student Employee', 'Administration', 12, 'Abhijith', 30, '2020-01-18'),

(293219321, 'Harley', 'Quinn', 'harley@gmail.com', '2020-03-04', '', 'Employee', 'Adminisration', 15, 'Abhijith', 55, '2020-01-18'),

(488233948, 'Melanie', 'Mellow','melanie@gmail.com', '2019-09-04', '2020-02-05', 'Employee', 'Catalouging', 15, 'Nyugen', 66, '2020-02-05'),

(872732739, 'Bob', 'Builder', 'bob@gmail.com', '2019-10-10', '', 'Employee', 'Catalouging', 15, 'Nyugen', 88, '2020-01-18'),

(090832983, 'Dim', 'Dwight', 'dim@gmail.com','2020-03-01', '', 'Student Employee', 'Catalouging', 12, 'Nyugen', 53, '2020-01-18'),

(928398123, 'Jackway', 'Mclovin', 'hackway@gmail.com','2020-01-03', '', 'Employee', 'Catalouging', 15, 'Nyugen', 44, '2020-01-18'),

(928392183, 'Tim', 'Jonathan', 'tim@gmail.com','2019-05-15', '', 'Employee', 'Catalouging', 15, 'Nyugen', 100, '2020-01-18'),

(623727382, 'Jacksay', 'Witman','jackshawy@gmail.com', '2019-05-15', '', 'Employee', 'Administration', 15, 'Abhijith', 99, '2020-01-18')

go

-- Verify

select \* from books

select \* from writers

select \* from book\_authors

select \* from members

select \* from reservations

select \* from employees

* SQL Up/Down Script to load / migrate in existing data.

--View Code

drop view if exists view\_books\_and\_authors

drop view if exists students\_cost

drop view if exists view\_library\_cost\_students

drop view if exists view\_reserved\_books

go

create view view\_reserved\_books as

   select m.member\_firstname + ' ' + m.member\_lastname as 'Member Name',

       b.book\_title as 'Book Title', b.book\_category\_name as 'Book Genre',

       r.reservation\_date as 'Reservation Date',

       r.reservation\_return\_date as 'Return Date for the book: '

       from reservations r

           join members m  on m.member\_id = r.reservation\_member\_id

           join books b on b.book\_id = r.reservation\_book\_id

       group by m.member\_firstname, m.member\_lastname, b.book\_title, b.book\_category\_name, r.reservation\_date, r.reservation\_return\_date

go

create view view\_library\_cost\_students as

   select e.employee\_firstname + ' ' + e.employee\_lastname as 'Employee Name',

        e.employee\_payrate as 'Employee Pay rate',

        e.employee\_hours\_worked as 'Total hours worked',

        e.employee\_payrate \* e.employee\_hours\_worked as 'Total Gross Pay',

        e.employee\_department\_head as 'Reports to: '

        from employees e

        where employee\_jobtitle = 'Student Employee'

go

create view students\_cost as

   select SUM(e.employee\_hours\_worked \* e.employee\_payrate) as 'Total Cost of Hiring Students'

        from employees e

        where employee\_jobtitle = 'Student Employee'

go

go

create view view\_books\_and\_authors as

   select b.book\_title as 'Book Title',

   b.book\_category\_name as 'Category',

   w.writer\_first\_name + ' ' + w.writer\_last\_name as 'Writers Name'

       from books b

           join book\_authors a  on a.book\_author\_book\_id = b.book\_id

           join writers w on w.writer\_id = a.book\_author\_author\_id

       group by b.book\_title, b.book\_category\_name, w.writer\_first\_name, w.writer\_last\_name

go

Team Log recording:

1. Abhijith Anil Vamadev:
2. Data Entities, Attributes, PK/FK relationships – October 31st
3. Conceptual Model Diagram – November 7th
4. Logical Model Diagram – November 7th
5. Identification of your external data model and data logic. – November 21st
6. Basic layout of all application screens. – November 28th
7. Diagram of each screen used in the application. – November 28th
8. SQL Up/Down script to implement the internal model with initial data. – November 21st
9. SQL Up/Down Script to load / migrate in existing data. – November 21st
10. SQL Up/Down script of data logic for the external data model – November 21st
11. A working implementation of the application – November 29th
12. Presentation – 4th December
13. Report – 4th December
14. Kelvin:
15. Data Entities, Attributes, PK/FK relationships – October 31st
16. Conceptual Model Diagram – November 7th