**Boston public library:**

The library offering a diverse range of services that cater to the needs of its vibrant community. It provides extensive physical collections that free for all the citizens such as e-books, audiobooks, magazines, fiction. Educational programs, workshops, and events are also included. The dedicated librarians and staff stand ready to assist visitors in finding specific information, conducting research.

**Mission statement:**

Maintain the data generated by Boston public library and ensure an efficient resources and services for both customers and employees.

**Mission Objectives:**

To maintain data for entities below:

Library Info

Store basic information of Boston Public Libraries, such as address, name

Employee

Store the information for employee who work for each library.

Customer Info

Store information for customers who registered in one of Boston Public Libraries.

Type

The type of one a collection in the library. Such as fiction, nonfiction, music, CDs…

Collections (Book)

The collection such as books stored in a specific library.

Items

A physical copy of a collections that people can borrow.

Publisher

The publisher of a collection.

Contributor

The author or contributor of a collection

Special Events

Events held by a library for people to attend.

Appointments

Customers can make an appointment with employees for some services. Such as help on research and request for special resources.

Class

Boston public library held some regular class, speech, workshop to publics.

Reservation

Record the reservation when a customer reserve or borrow a book in the library.

Department

A library usually consists of multiple department such as Children’s Library, Community Learning Center, Teen Central.

Schedule

Store open time on each week of day for events, library’s open time, class.

**Requirement and relationships:**

**For Customer:**

The database must store customer information, including their name, Social Security Number (SSN), email, and phone number. Each customer should have a unique customer ID (CustomerID) as the primary key.

The system should support customer registration and manage the issuance of unique library cards. Each library card should have a unique card ID (CardID) associated with a specific customer (CustomerID). One customer can only have one card at a time.

Customers should be able to make reservations for library items like books, and CDs, and the database should store reservation details, including the reservation date, status, associated library card (CardID), and reserved item (ItemID). The system should allow customers to access their reservation history and view details of reserved items and their status. When they reserve an item, the database update the item status and record the reservation. One customer can hold 0-5 item at maximum at then same time.

Customer should also be able to make appointments with library staff if they need and help. The time schedule, appointment type (AppointmentType), ID (AppointmentID), and the status would be recorded. One customer can have many appointments and each appointment is unique and just for one customer.

One customer can register at only one library, but they can use the service from all Boston Public Library. A library can have one to many customers. After they registered at a library, they will receive their unique library card. Each library card can hold zero to five reservations at the same time.

**The employees:**

The database must store employee information, including their name, Social Security Number (SSN), gender, email, phone number, and employee status. Each employee should have a unique employee ID (EmployeeID) as the primary key.

Employees should be associated with specific departments within the library. An employees can only work in one department at the same time. The database should store information about the appointments they served. One appointment can be served by multiple employees depends on the appointment details.

The employee can work for one department at the same time. One department can have many employees.

**For library:**

The database will store the basic information for all related Boston Public Library. Information includes name (LibraryName), contact, email, address, and the specific time schedule to show when the library open on each day.

Libraries have one to many different departments; each department have a unique open time or schedule to public. Each department have a unique id can belong to only one library.

Each library can hold some special events that free for publics to join. One library can hold multiple events at the same time, but every event at different library should be unique. The database will store the ID (EventID), name, description, and status of every events. Each course has one to many unique schedule.

Library also provides many general courses to the publics. Similarly, the course name, schedule, status would be recorded. Each course have one to many unique schedule.

The database should also be able to manage its collections. The collection is virtual item in the library. Each collection has a unique ID, name, the data added to the library (adding date), description and the ISBN. The collection would be classified to different types like music, fictions, with a TypeID. The public information will include the publisherID, name, country, and the city. One collection can be published by many publishers which means different version of an item. Each publisher can publish many collections. Each collection can have many copies in different library but every copy is unique and we need to track the status. For every collection, the database will also record the contributors. One collection can have one to many contributors. Each contributors’ name, gender, age, education, major, job would also be recorded.

**Summary of relationship:**

**Library - Collection:**

The "Library" table has a one-to-many relationship with the "Collection" table. Each collection belongs to one library, and each library can have multiple collections.

**Collection - CollectionType:**

The "Collection" table has a many-to-one relationship with the "CollectionType" table. Each collection has a specific collection type, and multiple collections can belong to the same collection type.

**Collection - Contributor** (Many-to-Many Relationship):

The "Collection" table has a many-to-many relationship with the "Contributor" table through the "collection\_contributors" table. Each collection can have multiple contributors, and each contributor can contribute to multiple collections.

**Collection - Publisher** (Many-to-Many Relationship):

The "Collection" table has a many-to-many relationship with the "Publisher" table through the "publisher\_\_publish\_collection" table. Each collection can have multiple publishers, and each publisher can publish multiple collections.

**Item - Collection:**

The "Item" table has a many-to-one relationship with the "Collection" table. Each item belongs to a specific collection, and each collection can have multiple items.

**Customer - Library:**

The "Customer" table has a many-to-one relationship with the "Library" table. Each customer is associated with one library, and each library can have multiple customers.

**LibraryCard - Customer:**

The "LibraryCard" table has a many-to-one relationship with the "Customer" table. Each library card is associated with one customer, and each customer can have one library card.

**Reservation - LibraryCard:**

The "Reservation" table has a many-to-one relationship with the "LibraryCard" table. Each reservation is linked to one library card, and each library card can have multiple reservations.

**Reservation - Item:**

The "Reservation" table has a many-to-one relationship with the "Item" table. Each reservation is linked to one item, and each item can have multiple reservations.

**Appointment - LibraryCard:**

The "Appointment" table has a many-to-one relationship with the "LibraryCard" table. Each appointment is associated with one library card, and each library card can have multiple appointments.

**Appointment - Library:**

The "Appointment" table has a many-to-one relationship with the "Library" table. Each appointment is associated with one library, and each library can have multiple appointments.

**SpecialEvent - Library:**

The "SpecialEvent" table has a many-to-one relationship with the "Library" table. Each special event is associated with one library, and each library can have multiple special events.

**Course - Library:**

The "Course" table has a many-to-one relationship with the "Library" table. Each course is associated with one library, and each library can have multiple courses.

**Department - Library:**

The "Department" table has a many-to-one relationship with the "Library" table. Each department is associated with one library, and each library can have multiple departments.

**Employee - Department:**

The "Employee" table has a many-to-one relationship with the "Department" table. Each employee is associated with one department, and each department can have multiple employees.

**Employee\_serve\_Appointment (Many-to-Many Relationship):**

The "employee\_serve\_appointment" table establishes a many-to-many relationship between the "Employee" and "Appointment" tables. Each employee can serve multiple appointments, and each appointment can be served by multiple employees.

**TimeSchedule - Library:**

The "TimeSchedule" table has a many-to-one relationship with the "Library" table. Each time schedule entry is associated with one library, and each library can have multiple time schedule entries.

**TimeSchedule - Appointment:**

The "TimeSchedule" table has a many-to-one relationship with the "Appointment" table. Each time schedule entry is associated with one appointment, and each appointment can have multiple time schedule entries.

**TimeSchedule - Course:**

The "TimeSchedule" table has a many-to-one relationship with the "Course" table. Each time schedule entry is associated with one course, and each course can have multiple time schedule entries.

**TimeSchedule - SpecialEvent:**

The "TimeSchedule" table has a many-to-one relationship with the "SpecialEvent" table. Each time schedule entry is associated with one special event, and each special event can have multiple time schedule entries.

**TimeSchedule - Department:**

The "TimeSchedule" table has a many-to-one relationship with the "Department" table. Each time schedule entry is associated with one department, and each department can have multiple time schedule entries.

**Primary key:**

For efficiency considerations under Mysql database, all primary key is generated using auto increment id. (Data index in Mysql stored as black + tree. Using discontinuous value like SSN will influence the insertion efficiency while keep the query efficiency the same.)

**Foreign key:**

All foreign key is assigned properly in Relational model below.

**ER diagram.**

图示, 示意图

描述已自动生成

**Relational model**

图示

描述已自动生成

**Normalization:**

All functional dependencies:

Library table:

LibraryID -> LibraryName, phone, email, address

CollectionType table:

CollectionTypeID -> CollectionTypeName

Contributor table:

ContributorID -> ContributorName, Gender, Country, Age, Education, Major, Job

Publisher table:

PublisherID -> PublisherName, Country, City

Collection table:

CollectionID -> CollectionName, AddingDate, ColleDescript, ISBN, CollectionTypeID

collection\_contributors table:

(ContributorID, CollectionID) -> {no other attributes are present}

publisher\_\_publish\_collection table:

(PublisherID, CollectionID) -> {no other attributes are present}

Item table:

ItemID -> ItemStatus, CollectionID, libraryID

Customer table:

CustomerID -> CustomerName, SSN, email, phone, libraryID

LibraryCard table:

CardID -> CardStatus, CustomerID

Reservation table:

ReservationID -> ReservationDate, ReturnedDate, CardID, ItemID

Appointment table:

AppointmentID -> AppointmentType, AppointmentStatus, CardID, libraryID

SpecialEvent table:

EventID -> EventName, EventDescription, EventStatus, libraryID

Course table:

CourseID -> CourseName, CourseDescription, CourseStatus, libraryID

Department table:

DepartmentId -> DepartmentName, email, phone, libraryID

Employee table:

EmployeeID -> EmployeeName, ssn, gender, email, phone, EmployeeStatus, DepartmentId

employee\_serve\_appointment table:

(AppointmentID, EmployeeID) -> {no other attributes are present}

TimeSchedule table:

ScheduleId -> DayOfWeek, StartTime, EndTime, libraryID, AppointmentID, CourseID, EventID, DepartmentID

2NF:

Database is already in third normal form since there is no partial dependencies and every non-prime attribute is fully functionally dependent on the entire primary key. Except publisher\_\_publish\_collection table and employee\_serve\_appointment table which are used to store information for M:N relationship, all other table only have one primary key to avoid partial dependencies.

3NF:

According to the dependencies above, we can see there is no transitive dependency and normalization is completed during design the database. For all the functional dependencies in the tables, all non-prime attributes are fully dependent on the primary key. The database satisfies the conditions for 3rd Normal Form (3NF) since there are no partial or transitive dependencies.

**SQL commands:**

The project is completed using Mysql.

The SQL code is in external file: boston public library.sql.

Testing data insertion is in external file: boston public library testing data.sql.

Testing query is in file: boston public library query test.sql

**Date queries:**

**Able to:**

1. Customers find information for all related Boston public libraries.
2. Find information for a book: Given a collection name, return the status of all such item under the collection and also return the collection name, publisher name, isbn, contributors name and collection which library hold this item.
3. Customer search for the open schedule for one library.
4. Customer doing research based on list all collection type in a library.
5. Customers want to join an event, list the recent event and time schedule for a library.
6. Customer wants to borrow one item: add the reservation automatically change the status of an item
7. Customer return one item: update the reservation, change of item status back to available.
8. Employees check the appointment they assigned by the manager or customers.
9. Check for all active courses in Boston public library and the schedule for each course.
10. Find the most popular collection in all libraries (highest reservation number).
11. Find all collection and collection name from a given contributor.
12. Find all collection and collection name from a given publisher.
13. Employee check all unreturned item reserved by a customer.
14. Employee count all unreturned item reserved by a customer.
15. Automatically avoid someone borrow more than 5 book, there will be a warning!
16. Automatically avoid someone borrow an unavailable item, there will be a warning!