

Library Database

Group 1

Javid Ditty, Abdu Hijazi, Andre Floyd, Samuel Stenerson,
Hashem Boussi, Caleb Nickens



Team Responsibilities

| Team Member | Responsibilities |
|------------------|---|
| Javid Ditty | Database Requirements, ER Diagram, Relational Database Schema |
| Abdu Hijazi | Database Requirements, Sample Database Instance, SQL Statements and Results |
| Andre Floyd | Application Background, Sample Database Instance |
| Samuel Stenerson | Application Background, ER Diagram, Relational Database Schema |
| Hashem Boussi | Application Background, Sample Database Instance |
| Caleb Nickens | Sample Database Instance, SQL Statements and Results |



Application Background

Resource Management

- Manage books, journals, newspapers, magazines, etc.
- Facilitate and streamline library operations through proper and efficient resource tracking and management.

Transaction Handling

- Libraries are fundamentally about the circulation of material.
- Efficient and accurate processing of lending, rentals, renewals, and reservations is important for an effective management system.
- Designed to keep track of these activities, ensuring timely return of resources and availability for other members.

Member Services

- Involves registration processes, tracking borrowing history and overdue books, and managing member dues and fines.
- Maintaining detailed member records is vital to keep an accurate account of borrowing activities.

Staff Administration

- Involves managing staff records, defining roles and responsibilities, and handling payroll.
- Efficient management of staff records and roles ensures smooth operation and effective allocation of human resources.



Database Requirements

Each person has a unique id, a first name, a middle initial, and a last name and can be staff members, library members, both, or neither. Each staff member has a unique ssn, a role, a salary, a birthday, an age, and a sex and can have multiple addresses and multiple phone numbers. Some staff members are supervisors and can have multiple other staff members as supervisees. Each library member has a unique membership id, a membership status, a membership expiration date, and a join date.

Each staff member can be assigned multiple service tickets, and each service ticket has a unique id, an issue description, a start date, a due date, a completion date, and a status.

Exactly one staff member presides over all transactions with library members, especially those related to fines. Each transaction has a unique id, the date something was borrowed, the date the borrowed thing is due, the date the borrowed thing is returned, and a status. Transactions can have fines, and each fine has a motive, a date it was issued, and a cost. Multiple transactions can have the same fines, but the same transaction cannot have multiple fines with the same motive. Transactions and their fines are concerned with at least one stock item.

Each library member can reserve multiple locations in the library, and each reservation has a start date, an end date, a title, and a purpose. Each location has a unique room number, a name, and a type and can be a space, with a capacity, a shelf, with a category, or neither. Each shelf can contain multiple stock items.

Each supplier of created stock has a unique id and a type and can be either a person or an institution. Each institution has a unique id, a name, and a type and can have multiple addresses and phone numbers. Each supplier can supply multiple stock items.

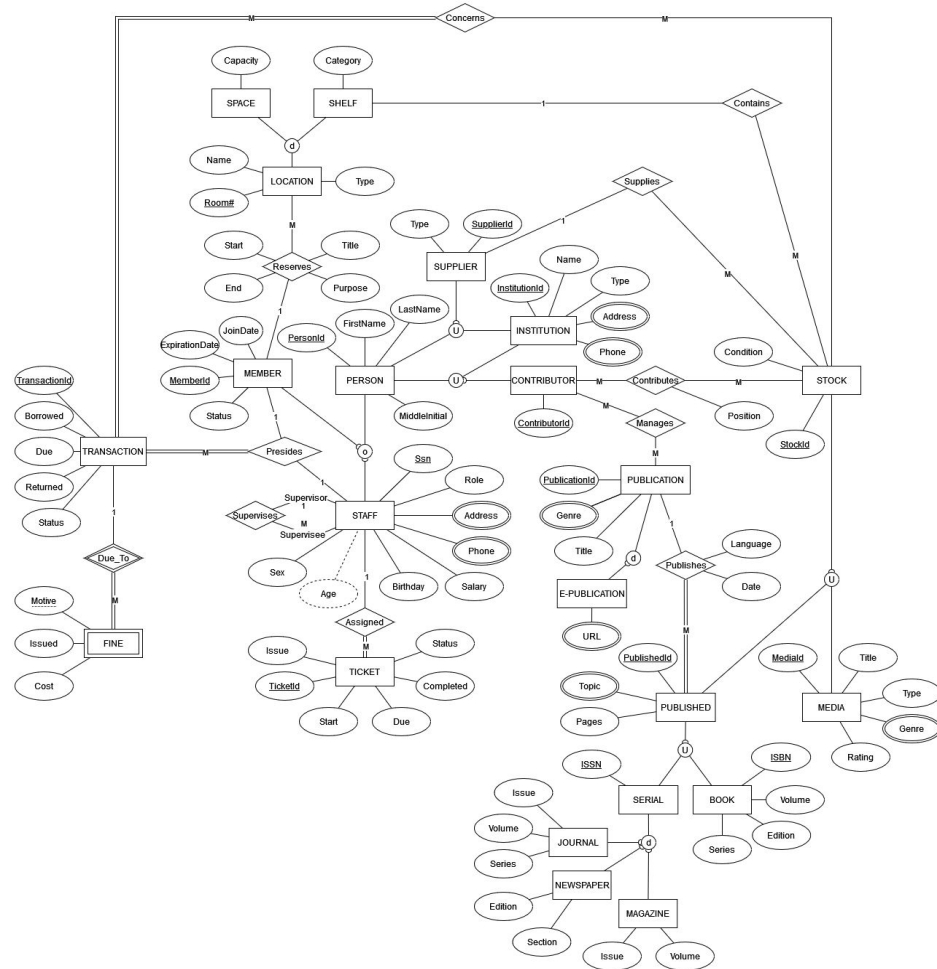
Each contributor to the creation of stock has a unique id and can be a person or an institution. Multiple contributors can contribute to many stock items, and each contribution has its own type. Multiple contributors can manage multiple publications, and each publication has a unique id and a title and can have multiple genres. Publications can be e-publications, and each e-publication can have multiple urls.

Each stock item has a unique id and a condition. Each stock item is media or published. Each media item has a unique id, a title, a type, and a rating and can have multiple genres. Each published stock item has a unique id and a number of pages and can have multiple topics. Multiple published stock items must be published by a publication in a language and with a publication date.

Each published stock item is either a serial item or a book. Each serial item has a unique ISSN, and each book has a unique ISBN, volume, edition, and series. Each serial item can be either a journal, newspaper, or magazine. Each journal has an item, a volume, and a series. Each newspaper has an edition and a section. Each magazine has an issue and a volume.



ER Diagram





Relational Database Schema

PHONE (PhoneId, Phone, Ssn, InstitutionId)

- Foreign Key: Ssn (STAFF.Ssn), InstitutionId (INSTITUTION.InstitutionId)

| | | | |
|----------------|-------|-----|---------------|
| <u>PhoneId</u> | Phone | Ssn | InstitutionId |
|----------------|-------|-----|---------------|

STAFF (Ssn, Role, Salary, Birthday, Age, Sex, Supervisor)

- Foreign Key: Supervisor (STAFF.Ssn)

- Functional Dependency: Birthday → Age

| | | | | | | |
|------------|------|--------|----------|-----|-----|------------|
| <u>Ssn</u> | Role | Salary | Birthday | Age | Sex | Supervisor |
|------------|------|--------|----------|-----|-----|------------|

SHELF (Room#, Category)

- Foreign Key: Room# (LOCATION.Room#)

| | |
|--------------|----------|
| <u>Room#</u> | Category |
|--------------|----------|

RESERVES (Reservee, Room#, Title, Purpose, Start, End)

- Foreign Key: Reservee (MEMBER.MemberId), Room# (LOCATION.Room#)

| | | | | | |
|-----------------|--------------|-------|---------|-------|-----|
| <u>Reservee</u> | <u>Room#</u> | Title | Purpose | Start | End |
|-----------------|--------------|-------|---------|-------|-----|

ADDRESS (AddressId, Address, Ssn, InstitutionId)

- Foreign Key: Ssn (STAFF.Ssn), InstitutionId (INSTITUTION.InstitutionId)

| | | | |
|------------------|---------|-----|---------------|
| <u>AddressId</u> | Address | Ssn | InstitutionId |
|------------------|---------|-----|---------------|

NEWSPAPER (ISSN, Edition, Section)

| | | |
|-------------|---------|---------|
| <u>ISSN</u> | Edition | Section |
|-------------|---------|---------|

JOURNAL (ISSN, Issue, Volume, Series)

| | | | |
|-------------|-------|--------|--------|
| <u>ISSN</u> | Issue | Volume | Series |
|-------------|-------|--------|--------|

BOOK (ISBN, Volume, Edition, Series)

| | | | |
|-------------|--------|---------|--------|
| <u>ISBN</u> | Volume | Edition | Series |
|-------------|--------|---------|--------|

TICKET (TicketId, Issue, Start, Due, Completed, Status, Assignee)

- Foreign Key: Assignee (STAFF.Ssn)

| | | | | | | |
|-----------------|-------|-------|-----|-----------|--------|----------|
| <u>TicketId</u> | Issue | Start | Due | Completed | Status | Assignee |
|-----------------|-------|-------|-----|-----------|--------|----------|

PERSON (PersonId, FirstName, MiddleInitial, LastName)

| | | | |
|-----------------|-----------|---------------|----------|
| <u>PersonId</u> | FirstName | MiddleInitial | LastName |
|-----------------|-----------|---------------|----------|

SUPPLIER (SupplierId, Type, InstitutionId, PersonId)

- Foreign Key: INSTITUTION (INSTITUTION.InstitutionId), PERSON (PERSON.PersonId)

| | | | |
|-------------------|------|---------------|----------|
| <u>SupplierId</u> | Type | InstitutionId | PersonId |
|-------------------|------|---------------|----------|

INSTITUTION (InstitutionId, Name, Type)

| | | |
|----------------------|------|------|
| <u>InstitutionId</u> | Name | Type |
|----------------------|------|------|

CONTRIBUTOR (ContributorId, PersonId, InstitutionId)

- Foreign Key: PERSON (PERSON.PersonId), INSTITUTION (INSTITUTION.InstitutionId)

| | | |
|----------------------|----------|---------------|
| <u>ContributorId</u> | PersonId | InstitutionId |
|----------------------|----------|---------------|

MANAGES (ContributorId, PublicationId)

- Foreign Key: ContributorId (CONTRIBUTOR.ContributorId), PublicationId (PUBLICATION.PublicationId)

| | |
|----------------------|----------------------|
| <u>ContributorId</u> | <u>PublicationId</u> |
|----------------------|----------------------|

CONTRIBUTES (ContributorId, StockId, Position)

- Foreign Key: ContributorId (CONTRIBUTOR.ContributorId), StockId (STOCK.StockId)

| | | |
|----------------------|----------------|----------|
| <u>ContributorId</u> | <u>StockId</u> | Position |
|----------------------|----------------|----------|

GENRE (GenreId, Genre, PublicationId, MediaId, PublishedId)

- Foreign Key: PublicationId (PUBLICATION.PublicationId), MediaId (MEDIA.MediaId), PublishedId (PUBLISHED.PublishedId)

| | | | | |
|----------------|-------|---------------|---------|-------------|
| <u>GenreId</u> | Genre | PublicationId | MediaId | PublishedId |
|----------------|-------|---------------|---------|-------------|

PUBLICATION (PublicationId, Title)

| | |
|----------------------|-------|
| <u>PublicationId</u> | Title |
|----------------------|-------|

E-PUBLICATION (PublicationId)

- Foreign Key: PublicationId (PUBLICATION.PublicationId)

| |
|----------------------|
| <u>PublicationId</u> |
|----------------------|

URL (Url, PublicationId)

- Foreign Key: PublicationId (E-PUBLICATION.PublicationId)

| | |
|------------|---------------|
| <u>Url</u> | PublicationId |
|------------|---------------|

STOCK (StockId, Condition, Room#, PublishedId, MediaId, SupplierId)

- Foreign Key: PublishedId (PUBLISHED.PublishedId), MediaId (MEDIA.MediaId), Room# (SHELF.Room#), SupplierId (SUPPLIER.SupplierId)

| | | | | | |
|----------------|-----------|-------|-------------|---------|------------|
| <u>StockId</u> | Condition | Room# | PublishedId | MediaId | SupplierId |
|----------------|-----------|-------|-------------|---------|------------|

MEDIA (MediaId, Title, Type, Rating)

| | | | |
|----------------|-------|------|--------|
| <u>MediaId</u> | Title | Type | Rating |
|----------------|-------|------|--------|

PUBLISHED (PublishedId, Pages, Language, Date, PublicationId, ISSN, ISBN)

- Foreign Key: PublicationId (PUBLICATION.PublicationId), ISSN (SERIAL.ISSN), ISBN (BOOK.ISBN)

| | | | | | | |
|--------------------|-------|----------|------|---------------|------|------|
| <u>PublishedId</u> | Pages | Language | Date | PublicationId | ISSN | ISBN |
|--------------------|-------|----------|------|---------------|------|------|

SERIAL (ISSN)

| |
|-------------|
| <u>ISSN</u> |
|-------------|

MAGAZINE (ISSN, Issue, Volume)

| | | |
|-------------|-------|--------|
| <u>ISSN</u> | Issue | Volume |
|-------------|-------|--------|

CONCERNS (TransactionId, StockId)

- Foreign Key: TransactionId (TRANSACTION.TransactionId), StockId (STOCK.StockId)

| | |
|----------------------|----------------|
| <u>TransactionId</u> | <u>StockId</u> |
|----------------------|----------------|

PRESIDES (TransactionId, Ssn, MemberId)

- Foreign Key: TransactionId (TRANSACTION.TransactionId), Ssn (STAFF.Ssn), MemberId (MEMBER.MemberId)

| | | |
|----------------------|------------|-----------------|
| <u>TransactionId</u> | <u>Ssn</u> | <u>MemberId</u> |
|----------------------|------------|-----------------|

TRANSACTION (TransactionId, Borrowed, Due, Returned, Status)

| | | | | |
|----------------------|----------|-----|----------|--------|
| <u>TransactionId</u> | Borrowed | Due | Returned | Status |
|----------------------|----------|-----|----------|--------|

FINE (TransactionId, Motive, Issued, Cost)

- Foreign Key: TransactionId (TRANSACTION.TransactionId)

| | | | |
|----------------------|---------------|--------|------|
| <u>TransactionId</u> | <u>Motive</u> | Issued | Cost |
|----------------------|---------------|--------|------|

MEMBER (MemberId, Status, JoinDate, ExpirationDate)

| | | | |
|-----------------|--------|----------|----------------|
| <u>MemberId</u> | Status | JoinDate | ExpirationDate |
|-----------------|--------|----------|----------------|

LOCATION (Room#, Name, Type)

| | | |
|--------------|------|------|
| <u>Room#</u> | Name | Type |
|--------------|------|------|

SPACE (Room#, Capacity)

- Foreign Key: Room# (LOCATION.Room#)

| | |
|--------------|----------|
| <u>Room#</u> | Capacity |
|--------------|----------|

SHELF (Room#, Category)


- Foreign Key: Room# (LOCATION.Room#)

| | |
|--------------|----------|
| <u>Room#</u> | Category |
|--------------|----------|



Sample Database Instance

* Only some tuples in each table are included.



Book

| ISBN | Volume | Edition | Series |
|-------------------|--------|---------|--------|
| 978-3-16-148410-0 | 2 | 3 | 1 |
| 978-1-4028-9462-6 | 1 | 2 | 1 |
| 978-0-596-52068-7 | 11 | 2 | 2 |
| 978-0-306-40615-7 | 4 | 3 | 2 |
| 978-3-540-00238-3 | 1 | 1 | 1 |
| 978-0-12-374856-0 | 2 | 21 | 1 |
| 978-0-471-48648-0 | 5 | 1 | 1 |
| 978-1-86197-876-9 | 3 | 2 | 1 |
| 978-1-59853-436-6 | 4 | 3 | 1 |
| 978-0-14-044913-6 | 5 | 1 | 2 |

Journal

| ISSN | Issue | Volume | Series |
|-----------|-------|--------|--------|
| 2928-8383 | 4 | 9 | 3 |
| 7236-8373 | 2 | 3 | 1 |
| 6638-3733 | 2 | 7 | 2 |
| 1217-8637 | 8 | 12 | 4 |
| 0282-8282 | 1 | 3 | 2 |

Serial

| ISSN |
|-----------|
| 0200-2822 |
| 7373-8239 |
| 8923-8282 |
| 1070-8089 |
| 8128-9292 |

Magazine

| ISSN | Issue | Volume |
|-----------|-------|--------|
| 1234-5678 | 4 | 10 |
| 1212-0101 | 8 | 8 |
| 9060-5656 | 3 | 7 |
| 3487-3448 | 9 | 4 |
| 2040-6080 | 2 | 12 |

Media

| MediaID | Title | Type | Rating |
|---------|-----------|-------|--------|
| 123456 | Shrek | Movie | PG |
| 234561 | Plane | Movie | R |
| 848484 | Missing | Movie | PG-13 |
| 100200 | Kill Shot | Movie | R |

LibraryMember

| MemberID | PersonID | Status | JoinDate | ExpirationDate |
|----------------|----------|----------|------------|----------------|
| 72316513783412 | 12345 | ACTIVE | 01/12/2021 | 01/12/2024 |
| 82372617123571 | 81282 | ACTIVE | 05/02/2023 | 05/02/2026 |
| 72521781251235 | 47472 | INACTIVE | 11/10/2020 | 11/10/2023 |
| 76123512351235 | 30643 | ACTIVE | 08/03/2022 | 08/03/2025 |

Person

| PersonID | FirstName | MiddleInitial | LastName |
|----------|-----------|---------------|----------|
| 12345 | John | A | Hunter |
| 81282 | Alex | II | Denson |
| 47472 | Ally | N | Garrett |
| 30643 | Tonya | G | Duncan |
| 11223 | Ali | J | Bazzi |

Staff

| SSN | Role | Salary | Birthday | Age | Sex | Supervisor |
|-----------|----------------------|----------|------------|-----|-----|------------|
| 123456789 | Librarian | 52000.00 | 09/01/1967 | 56 | F | 555555555 |
| 111223333 | Library Assistant | 25000.00 | 03/18/2000 | 23 | F | 555555555 |
| 444556666 | Teen Librarian | 48000.00 | 07/06/1995 | 28 | F | 555555555 |
| 777889999 | Children's Librarian | 45554.67 | 04/05/1982 | 41 | F | 555555555 |

URL

| URL | PublicationID |
|---|---------------|
| http://www.example.com/publication/121283 | 121283 |
| http://www.example.com/publication/723822 | 723822 |
| http://www.example.com/publication/905143 | 905143 |
| http://www.example.com/publication/534021 | 534021 |

Publication

| PublicationID | Title |
|---------------|-------------------------|
| 121283 | Gone with the Wind |
| 723822 | Lord of the Rings |
| 905143 | Cooking for Dummies |
| 534021 | Fundamentals of Physics |

E-Publication

| PublicationID |
|---------------|
| 121283 |
| 723822 |
| 905143 |
| 534021 |
| 637683 |

Published

| PublishedID | Pages | Language | Date | PublicationID | ISSN | ISBN |
|-------------|-------|----------|------------|---------------|-----------|-------------------|
| 2345678 | 300 | English | 2023-01-15 | 121283 | 1234-5678 | 978-3-16-148410-0 |
| 3456789 | 220 | Spanish | 2023-02-10 | 723822 | 2345-6789 | 978-1-4028-9462-6 |
| 4567890 | 150 | French | 2023-03-05 | 905143 | 3456-7890 | 978-0-596-52068-7 |
| 5678901 | 500 | German | 2023-04-20 | 534021 | 4567-8901 | 978-0-306-40615-7 |
| 6789012 | 350 | Italian | 2023-05-15 | 637683 | 5678-9012 | 978-3-540-00238-3 |

Fine

| TransactionID | Motive | Issued | Cost |
|---------------|--------------|------------|---------|
| 838328 | Overdue Item | 05/09/2023 | \$0.00 |
| 832893 | Damaged Item | 12/11/2021 | \$10.00 |
| 812962 | Lost Item | 12/15/2021 | \$27.45 |
| 289161 | Overdue Item | 03/07/2020 | \$0.00 |

Transaction

| TransactionID | Borrowed | Due | Returned | Status |
|---------------|------------|------------|------------|----------|
| 838328 | 10/02/2023 | 10/30/2023 | 10/30/2023 | Complete |
| 832893 | 10/05/2023 | 10/19/2023 | 10/18/2023 | Complete |
| 812962 | 08/19/2023 | 09/17/2023 | 00/00/0000 | Pending |

Location

| RoomNumber | Name | Type |
|------------|--------------|----------|
| 200 | Main | General |
| 101 | Reading Room | Children |
| 201 | Reading Room | Teens |
| 301 | Reading Room | Adults |
| 108 | Computer Lab | Children |

Space

| RoomNumber | Capacity |
|------------|----------|
| 200 | 500 |
| 101 | 256 |
| 201 | 235 |
| 301 | 231 |
| 108 | 120 |

Phone

| PhoneID | Phone | SSN | InstitutionID |
|---------|------------|-----------|---------------|
| 1234 | 3135555555 | 123456789 | 10012 |
| 5678 | 1234567890 | 111223333 | 10023 |
| 9012 | 3130000000 | 444556666 | 10034 |

Ticket

| TicketID | Issue | Start Date | Due Date | Completed Date | Status | Assignee |
|----------|------------------------------|------------|------------|----------------|--------|-----------|
| 1001 | Book Cataloging Error | 2023-11-01 | 2023-11-10 | 2023-11-09 | Closed | 777665555 |
| 1002 | Library Network Outage | 2023-11-02 | 2023-11-12 | 2023-11-11 | Closed | 100000001 |
| 1003 | RFID System Malfunction | 2023-11-03 | 2023-11-13 | Not Completed | Open | 100000001 |
| 1004 | Staff Training on New System | 2023-11-04 | 2023-11-14 | Not Completed | Open | 100000001 |

Supplier

| SupplierID | Type | InstitutionID | PersonID |
|------------|----------------------|---------------|----------|
| 10101 | Book Supplier | 10012 | 5001 |
| 10102 | Journal Subscription | 10023 | 5002 |
| 10103 | Digital Resources | 10034 | 5003 |

Contributor

| ContributorID | PersonID | InstitutionID |
|---------------|----------|---------------|
| 154279 | 80101 | 10012 |
| 738249 | 80102 | 10023 |
| 904612 | 80103 | 10034 |
| 531420 | 80104 | 10041 |

Edits

| ContributorID | PublicationID |
|---------------|---------------|
| 154279 | 121283 |
| 738249 | 723822 |
| 904612 | 905143 |
| 531420 | 534021 |
| 639874 | 637683 |

Address

| AddressID | Address | SSN | InstitutionID |
|-----------|-----------------------|-----------|---------------|
| 1 | 123 Maple St, Newview | 123456789 | 10012 |
| 2 | 456 Oak St, Newview | 111223333 | 10023 |
| 3 | 789 Pine St, Newview | 444556666 | 10034 |
| 4 | 321 Birch St, Newview | 777889999 | 10041 |
| 5 | 654 Elm St, Parkway | 101010101 | 10050 |

Concerns

| TransactionID | StockID |
|---------------|---------|
| 838328 | 1234567 |
| 832893 | 2345678 |
| 812962 | 3456789 |

Presides

| TransactionID | SSN | MemberID |
|---------------|-----------|----------------|
| 838328 | 123456789 | 72316513783412 |
| 832893 | 111223333 | 82372617123571 |
| 812962 | 444556666 | 72521781251235 |

Contributes

| ContributorID | StockID | Position |
|---------------|---------|-----------------------|
| 154279 | 500123 | Librarian |
| 738249 | 500124 | Cataloging Specialist |
| 904612 | 500125 | Reference Librarian |
| 531420 | 500126 | Collection Manager |
| 639874 | 500127 | Archivist |

Genre

| GenreID | Genre | PublicationID | PublishedID | MediaID |
|---------|-----------------|---------------|-------------|---------|
| 101100 | Science | 301010 | NULL | NULL |
| 102200 | Mystery | NULL | 404030 | NULL |
| 119900 | Poetry | 311110 | NULL | NULL |
| 120000 | Science Fiction | NULL | NULL | 516140 |

Stock

| StockID | ItemCondition | RoomNumber | PublishedID | MediaID |
|---------|---------------|------------|-------------|---------|
| 1234567 | New | 101 | 2345678 | 3456789 |
| 2345678 | Good | 102 | 3456789 | 4567890 |
| 3456789 | Fair | 103 | 4567890 | 5678901 |
| 4567890 | Excellent | 104 | 5678901 | 6789012 |

Shelf

| RoomNumber | Category |
|------------|----------|
| 200 | Books |
| 101 | Books |
| 201 | Books |
| 301 | Books |

Reserves

| Reservee | RoomNumber | Title | Purpose | Start | End |
|----------------|------------|-----------------------|---|------------|------------|
| 72316513783412 | 220 | Anti-Bullying | Stop people from bullying other people | 11/12/23 | 11/14/23 |
| 82372617123571 | 322 | Business Meeting | Coworkers getting together in a meeting | 10/13/2023 | 10/13/2023 |
| 23785782152363 | 326 | Group Project Meeting | Classmates working on a project | 11/05/2023 | 11/05/2023 |



SQL Statements and Results



SQL DDL

CREATE SCHEMA LIBRARY**CREATE TABLE CONCERNS (**

| | | |
|---------------|-----|-----------|
| TransactionId | INT | NOT NULL, |
| StockId | INT | NOT NULL, |

PRIMARY KEY (TransactionId, StockId),**FOREIGN KEY (TransactionId) REFERENCES TRANSACTION (TransactionId) ON DELETE CASCADE ON UPDATE CASCADE,****FOREIGN KEY (StockId) REFERENCES STOCK (StockId) ON DELETE CASCADE ON UPDATE CASCADE**
));**CREATE TABLE PRESIDES (**

| | | |
|---------------|-----|-----------|
| TransactionId | INT | NOT NULL, |
| Ssn | INT | NOT NULL, |
| MemberId | INT | NOT NULL, |

PRIMARY KEY (TransactionId, Ssn, MemberId),**FOREIGN KEY (TransactionId) REFERENCES TRANSACTION (TransactionId) ON DELETE CASCADE ON UPDATE CASCADE,****FOREIGN KEY (Ssn) REFERENCES STAFF (Ssn) ON DELETE CASCADE ON UPDATE CASCADE,****FOREIGN KEY (MemberId) REFERENCES MEMBER (MemberId) ON DELETE CASCADE ON UPDATE CASCADE**
));**CREATE TABLE TRANSACTION (**

| | | |
|---------------|-------------|-----------|
| TransactionId | INT | NOT NULL, |
| Borrowed | DATE | NOT NULL, |
| Due | DATE | NOT NULL, |
| Returned | DATE | NOT NULL, |
| Status | VARCHAR (5) | NOT NULL, |

PRIMARY KEY (TransactionId)
));**CREATE TABLE FINE (**

| | | |
|---------------|--------------|-----------------------------------|
| TransactionId | INT | NOT NULL, |
| Motive | VARCHAR (30) | NOT NULL, DEFAULT OVERDUE, |
| Issued | DATE | NOT NULL, |
| Cost | FLOAT | NOT NULL, |

PRIMARY KEY (TransactionId, Motive),**FOREIGN KEY (TransactionId) REFERENCES TRANSACTION (TransactionId) ON DELETE CASCADE ON UPDATE CASCADE**
));**CREATE TABLE LIBRARYMEMBER (**

| | | |
|----------------|-------------|------------------------------------|
| MemberId | INT | NOT NULL, |
| PersonId | INT | NOT NULL, |
| Status | VARCHAR (5) | NOT NULL, DEFAULT INACTIVE, |
| JoinDate | DATE | NOT NULL, |
| ExpirationDate | DATE | NOT NULL, |

PRIMARY KEY (MemberId),**FOREIGN KEY (PersonId) REFERENCES PERSON (PersonId) ON DELETE CASCADE ON UPDATE CASCADE**
));**CREATE TABLE LOCATION (**

| | | |
|-------|--------------|-----------------------------------|
| Room# | INT | NOT NULL, |
| Name | VARCHAR (15) | NOT NULL, |
| Type | VARCHAR (15) | NOT NULL, DEFAULT GENERAL, |

PRIMARY KEY (Room#)
));**CREATE TABLE SPACE (**

| | | |
|----------|-----|-----------------------------|
| Room# | INT | NOT NULL, |
| Capacity | INT | NOT NULL, DEFAULT 5, |

PRIMARY KEY (Room#),**FOREIGN KEY (Room#) REFERENCES LOCATION (Room#) ON DELETE CASCADE ON UPDATE CASCADE**
));**CREATE TABLE SHELF (**

| | | |
|----------|--------------|--------------------------------|
| Room# | INT | NOT NULL, |
| Category | VARCHAR (15) | NOT NULL, DEFAULT MISC, |

PRIMARY KEY (Room#),**FOREIGN KEY (Room#) REFERENCES LOCATION (Room#) ON DELETE CASCADE ON UPDATE CASCADE**
));**CREATE TABLE RESERVES (**

| | | |
|----------|--------------|---------------------------------------|
| Reservee | INT | NOT NULL, |
| Room# | INT | NOT NULL, |
| Title | VARCHAR (50) | NOT NULL, DEFAULT RESERVATION, |
| Purpose | VARCHAR (50) | NOT NULL, DEFAULT MISC, |
| Start | DATE | NOT NULL, |
| End | DATE | NOT NULL, |

PRIMARY KEY (Reservee, Room#),**FOREIGN KEY (Reservee) REFERENCES MEMBER (MemberId) ON DELETE CASCADE ON UPDATE CASCADE,****FOREIGN KEY (Room#) REFERENCES LOCATION (Room#) ON DELETE CASCADE ON UPDATE CASCADE**
));

```
CREATE TABLE ADDRESS (
  AddressId          INT          NOT NULL,
  Address            VARCHAR (50) NOT NULL,
  Ssn                VARCHAR (9),
  InstitutionId      INT,
  PRIMARY KEY (AddressId),
  FOREIGN KEY (Ssn) REFERENCES STAFF (Ssn) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (InstitutionId) REFERENCES INSTITUTION (InstitutionId) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE PHONE (
  PhoneId           INT          NOT NULL,
  Phone             VARCHAR (20) NOT NULL,
  Ssn               VARCHAR (9),
  InstitutionId     INT,
  PRIMARY KEY (PhoneId),
  FOREIGN KEY (Ssn) REFERENCES STAFF (Ssn) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (InstitutionId) REFERENCES INSTITUTION (InstitutionId) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE STAFF (
  Ssn                VARCHAR (9)          NOT NULL,
  Role               VARCHAR (15)         NOT NULL, DEFAULT None,
  Salary             FLOAT                NOT NULL,
  Birthday           DATE                 NOT NULL,
  Age                INT                  NOT NULL,
  Sex                VARCHAR(1)           NOT NULL,
  Supervisor         VARCHAR (9),
  PRIMARY KEY (Ssn),
  FOREIGN KEY (Ssn) REFERENCES STAFF (Ssn) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE TICKET (
  TicketId          INT          NOT NULL,
  Issue             VARCHAR (100) NOT NULL,
  Start             DATE          NOT NULL,
  Due               DATE          NOT NULL,
  Completed         DATE          NOT NULL,
  Status            VARCHAR (15)  NOT NULL, DEFAULT INCOMPLETE,
  Assignee          INT          NOT NULL,
  PRIMARY KEY (TicketId),
  FOREIGN KEY (Assignee) REFERENCES STAFF (Ssn) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE PERSON (
  PersonId          INT          NOT NULL,
  FirstName          VARCHAR (20) NOT NULL,
  MiddleInitial      VARCHAR (1),
  LastName           VARCHAR (20) NOT NULL,
  PRIMARY KEY (PersonId)
);
```

```
CREATE TABLE SUPPLIER (
  SupplierId        INT          NOT NULL,
  Type              VARCHAR (15) NOT NULL,
  InstitutionId     INT,
  PersonId          INT,
  PRIMARY KEY (SupplierId),
  FOREIGN KEY (InstitutionId) REFERENCES INSTITUTION (InstitutionId) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (PersonId) REFERENCES PERSON (PersonId) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE INSTITUTION (
  InstitutionId      INT          NOT NULL,
  Name              VARCHAR (30) NOT NULL,
  Type              VARCHAR (15) NOT NULL,
  PRIMARY KEY (InstitutionId)
);
```

```
CREATE TABLE CONTRIBUTOR (
  ContributorId     INT          NOT NULL,
  PersonId          INT,
  InstitutionId     INT,
  PRIMARY KEY (ContributorId),
  FOREIGN KEY (PersonId) REFERENCES PERSON (PersonId) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (InstitutionId) REFERENCES INSTITUTION (InstitutionId) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE MANAGES(
  ContributorId     INT          NOT NULL,
  PublicationId     INT          NOT NULL,
  PRIMARY KEY (ContributorId, PublicationId),
  FOREIGN KEY (ContributorId) REFERENCES CONTRIBUTOR (ContributorId) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (PublicationId) REFERENCES PUBLICATION (PublicationId) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE CONTRIBUTES (
  ContributorId     INT          NOT NULL,
  StockId           INT          NOT NULL,
  Position          VARCHAR (15) NOT NULL, DEFAULT MISC,
  PRIMARY KEY (ContributorId, StockId),
  FOREIGN KEY (ContributorId) REFERENCES CONTRIBUTOR (ContributorId) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (StockId) REFERENCES STOCK (StockId) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```

CREATE TABLE GENRE (
  GenreId          INT          NOT NULL,
  Genre            VARCHAR (15) NOT NULL,
  PublicationId     INT,
  MediaId          INT,
  PublishedId      INT,
  PRIMARY KEY (GenreId),
  FOREIGN KEY (PublicationId) REFERENCES PUBLICATION (PublicationId) ON DELETE CASCADE ON
  UPDATE CASCADE,
  FOREIGN KEY (MediaId) REFERENCES MEDIA (MediaId) ON DELETE CASCADE ON UPDATE
  CASCADE,
  FOREIGN KEY (PublishedId) REFERENCES PUBLISHED (PublishedId) ON DELETE CASCADE ON
  UPDATE CASCADE
);

```

```

CREATE TABLE PUBLICATION (
  PublicationId     INT          NOT NULL,
  Title            VARCHAR (50) NOT NULL,
  PRIMARY KEY (PublicationId)
);

```

```

CREATE TABLE E-PUBLICATION (
  PublicationId     INT          NOT NULL,
  PRIMARY KEY (PublicationId),
  FOREIGN KEY (PublicationId) REFERENCES PUBLICATION (PublicationId) ON DELETE CASCADE ON
  UPDATE CASCADE
);

```

```

CREATE TABLE URL (
  Url              VARCHAR (255) NOT NULL,
  PublicationId     INT          NOT NULL,
  PRIMARY KEY (URL, PublicationId),
  FOREIGN KEY (PublicationId) REFERENCES E-PUBLICATION (PublicationId) ON DELETE CASCADE
  ON UPDATE CASCADE
);

```

```

CREATE TABLE STOCK (
  StockId          INT          NOT NULL,
  Condition        VARCHAR (15) NOT NULL,
  Room#           INT,
  PublishedId      INT,
  MediaId         INT,
  SupplierId       INT,
  PRIMARY KEY (StockId),
  FOREIGN KEY (PublishedId) REFERENCES PUBLISHED (PublicationId) ON DELETE CASCADE ON
  UPDATE CASCADE,
  FOREIGN KEY (MediaId) REFERENCES MEDIA (MediaId) ON DELETE CASCADE ON UPDATE
  CASCADE,
  FOREIGN KEY (Room#) REFERENCES SHELF (Room#) ON DELETE CASCADE ON UPDATE
  CASCADE,
  FOREIGN KEY (SupplierId) REFERENCES SUPPLIER (SupplierId) ON DELETE CASCADE ON UPDATE
  CASCADE
);

```

```

CREATE TABLE MEDIA (
  MediaId          INT          NOT NULL,
  Title            VARCHAR (30) NOT NULL,
  Type             VARCHAR (15) NOT NULL,
  Rating           VARCHAR (1),
  PRIMARY KEY (MediaId)
);

```

```

CREATE TABLE PUBLISHED (
  PublishedId      INT          NOT NULL,
  Pages           INT,
  Language        VARCHAR (15),
  Date            DATE          NOT NULL,
  PublicationId     INT          NOT NULL,
  ISSN            INT,
  ISBN            INT,
  PRIMARY KEY (PublishedId),
  FOREIGN KEY (PublicationId) REFERENCES PUBLICATION (PublicationId) ON DELETE CASCADE ON UPDATE
  CASCADE,
  FOREIGN KEY (ISSN) REFERENCES SERIAL (ISSN) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (ISBN) REFERENCES BOOK (ISBN) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE SERIAL (
  ISSN            INT          NOT NULL,
  PRIMARY KEY (ISSN)
);

```

```

CREATE TABLE MAGAZINE (
  ISSN            INT          NOT NULL,
  Issue           INT          NOT NULL,
  Volume          INT          NOT NULL,
  PRIMARY KEY (ISSN)
);

```

```

CREATE TABLE NEWSPAPER (
  ISSN            INT          NOT NULL,
  Edition         INT          NOT NULL,
  Section         INT          NOT NULL,
  PRIMARY KEY (ISSN)
);

```

```

CREATE TABLE JOURNAL (
  ISSN            INT          NOT NULL,
  Issue           INT          NOT NULL,
  Volume          INT          NOT NULL,
  Series          INT          NOT NULL,
  PRIMARY KEY (ISSN)
);

```

```

CREATE TABLE BOOK (
  ISBN            INT          NOT NULL,
  Volume          INT          NOT NULL,
  Edition         INT          NOT NULL,
  Series          INT          NOT NULL,
  PRIMARY KEY (ISBN)
);

```



SQL DML

1. Retrieve the full details of all books published in a certain language after a specific date.

```
SELECT Title, book.ISBN, ISSN, Date, Pages, Language, volume, edition, series
FROM Supplier
JOIN contributor on contributor.InstitutionId = supplier.InstitutionId
JOIN edits on contributor.ContributorId = edits.ContributorId
JOIN published on published.PublicationId = edits.PublicationId
JOIN publication on published.PublicationId = publication.PublicationId
JOIN book on published.ISBN = book.ISBN
WHERE Language = 'English' AND
Date > '2023-4-20'
```

Result Grid

Filter Rows:




Export:

Wrap Cell Content:

| | Title | ISBN | ISSN | Date | Pages | Language | volume | edition | series |
|--|-------------|---------------|----------|------------|-------|----------|--------|---------|--------|
| | Animal Farm | 9780123748560 | 67890123 | 2023-06-30 | 280 | English | 2 | 21 | 1 |

2. All members who have overdue fines before a certain date.

```
SELECT presides.MemberId, MemberStatus, Issued, Motive, Cost, FirstName, LastName
FROM librarymember
JOIN presides ON librarymember.MemberId = presides.MemberId
JOIN transaction ON presides.TransactionId = transaction.TransactionId
JOIN fine ON transaction.TransactionId = fine.TransactionId
JOIN person on librarymember.PersonId = person.PersonId
WHERE
Issued < '2019-11-27' AND
Cost > 0
```

| | | | | | | | |
|-------------|----------------|---|-----------------------------------|---|--|-----------|----------|
| Result Grid | |  | Filter Rows: <input type="text"/> | Export:  | Wrap Cell Content:  | | |
| | MemberId | MemberStatus | Issued | Motive | Cost | FirstName | LastName |
| ▶ | 72316513783412 | ACTIVE | 2015-04-04 | Lost Item | 48.98 | John | Hunter |
| | 23785782152363 | ACTIVE | 2019-10-02 | Damaged Item | 28.56 | Ali | Bazzi |

3. List of All Books Borrowed by a Specific Member.

```
SELECT published.ISBN, Title, Borrowed, Due, ItemCondition, Pages, volume, edition, series,
FirstName, LastName
FROM transaction
JOIN presides ON transaction.TransactionId = presides.TransactionId
JOIN concerns ON transaction.TransactionId = concerns.TransactionId
JOIN stock ON concerns.StockId = stock.StockId
JOIN published ON stock.PublishedId = published.PublishedId
JOIN publication ON published.PublicationId = publication.PublicationId
JOIN book on published.ISBN = book.ISBN
JOIN librarymember on librarymember.MemberId = presides.MemberId
JOIN person on librarymember.PersonId = person.PersonId
WHERE presides.MemberId = 72316513783412
```

| | | | | | | | | | | | | | | | | | |
|-------------|---------------|----------------------------------|------------|------------|---------------|-------|--------|---------|--------|-----------|----------|--------------|--|---------|--|--------------------|--|
| Result Grid | | | | | | | | | | | | Filter Rows: | | Export: | | Wrap Cell Content: | |
| | ISBN | Title | Borrowed | Due | ItemCondition | Pages | volume | edition | series | FirstName | LastName | | | | | | |
| ▶ | 9781598534366 | Fundamentals of Database Systems | 2023-08-12 | 2023-09-12 | Fair | 410 | 4 | 3 | 1 | John | Hunter | | | | | | |
| | 9783161484100 | Gone with the Wind | 2023-10-02 | 2023-10-30 | New | 300 | 2 | 3 | 1 | John | Hunter | | | | | | |

4. Total fines collected each month.

```
SELECT SUM(Cost) AS TotalFines
FROM Fine
WHERE EXTRACT(MONTH FROM Issued) = 12 AND EXTRACT(YEAR FROM Issued) = 2021;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

| | |
|---|-------------------|
| | TotalFines |
| ▶ | 60.70000076293945 |

5. List of suppliers providing books in "Language"

```
SELECT SupplierID, TYPE, Language, supplier.InstitutionId
FROM Supplier
JOIN contributor on contributor.InstitutionId = supplier.InstitutionId
JOIN edits on contributor.ContributorId = edits.ContributorId
JOIN published on published.PublicationId = edits.PublicationId
WHERE Language = 'Spanish'
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|----------------------|----------|--------------------|
| SupplierID | TYPE | Language | InstitutionId |
| 10102 | Journal Subscription | Spanish | 10023 |
| 10107 | Archival Supplies | Spanish | 10074 |
| 10108 | Office Supplies | Spanish | 10085 |

6. Top 3 most popular book genres.

```
SELECT Genre, COUNT(*) AS Count
FROM Genre
GROUP BY Genre
ORDER BY Count DESC
LIMIT 3;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
|-----------------|--------------|---------|--------------------|
| Genre | Count | | |
| Science | 4 | | |
| Science Fiction | 4 | | |
| Mystery | 2 | | |

7. Top 3 most active members based on transactions

```
SELECT librarymember.MemberID, COUNT(TransactionId) AS TotalTransactions, FirstName,
LastName
FROM Presides
JOIN librarymember on librarymember.MemberId = presides.MemberId
JOIN person on librarymember.PersonId = person.PersonId
GROUP BY MemberID
ORDER BY TotalTransactions DESC
LIMIT 3;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
|----------------|-------------------|-----------|--------------------|
| MemberID | TotalTransactions | FirstName | LastName |
| 72316513783412 | 2 | John | Hunter |
| 9126821684801 | 1 | Patrick | Robbins |
| 9127238648923 | 1 | Jackson | Waters |

8. Upcoming Library Events ordered by start date

```
SELECT Title, Start, End, RoomNumber
FROM Reserves
WHERE Start > '2023-11-15'
ORDER BY Start;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
|--------------------------------|--------------|------------|--------------------|
| Title | Start | End | RoomNumber |
| What Thanksgiving is All About | 2023-11-21 | 2023-11-21 | 224 |
| Business Meeting | 2023-11-27 | 2023-11-27 | 322 |
| Construction Club | 2023-11-28 | 2023-11-28 | 226 |
| Story Time: The Cat in the Hat | 2023-11-28 | 2023-11-28 | 220 |
| Toys for Tots | 2023-12-01 | 2023-12-23 | 228 |
| Book Club | 2023-12-05 | 2023-12-05 | 320 |

9. Open tickets older than two weeks

```
SELECT TicketId, Issue, Start, memberStatus
FROM ticket
WHERE memberStatus = 'Open' AND
Start < DATE_SUB(CURDATE(), INTERVAL 14 DAY);
```

| Result Grid Filter Rows: Export: Wrap Cell Content: | | | | |
|---|----------|------------------------------|------------|--------------|
| | TicketId | Issue | Start | memberStatus |
| ▶ | 1003 | RFID System Malfunction | 2023-11-03 | Open |
| | 1004 | Staff Training on New System | 2023-11-04 | Open |
| | 1009 | Wi-Fi Access Interruption | 2023-11-09 | Open |
| | 1011 | Printer Network Connection | 2023-11-11 | Open |
| | 1014 | Self-Checkout Machine Error | 2023-11-14 | Open |

10. Find available rooms on a specific date within a specific capacity

```
SELECT space.RoomNumber, space.capacity
FROM space
LEFT JOIN reserves ON space.RoomNumber = reserves.RoomNumber
AND '2023-11-15' BETWEEN reserves.start AND reserves.end
WHERE reserves.RoomNumber IS NULL
AND space.capacity >= 97
ORDER by space.RoomNumber ASC
```

| Result Grid Filter Rows: Export: Wrap Cell Content: | | |
|---|------------|----------|
| | RoomNumber | capacity |
| ▶ | 101 | 256 |
| | 108 | 120 |
| | 113 | 145 |
| | 200 | 500 |
| | 201 | 235 |
| | 207 | 150 |
| | 230 | 372 |
| | 301 | 231 |
| | 305 | 250 |
| | 315 | 153 |
| | 328 | 107 |

Step 1: Set an INACTIVE member to ACTIVE

UPDATE librarymember

SET MemberStatus = "ACTIVE"

WHERE MemberID = 89126376376365;

SELECT *

FROM librarymember

ORDER by MemberStatus;

Step 2: Update any member who should be INACTIVE due to expiration to ACTIVE

UPDATE librarymember

SET MemberStatus = 'INACTIVE'

WHERE ExpirationDate < CURRENT_DATE

AND MemberStatus = 'ACTIVE';

SELECT*

FROM librarymember

ORDER by MemberStatus;

| Result Grid Filter Rows: Exports: Wrap Cell Contents: | | | | | |
|---|----------------|----------|--------------|------------|----------------|
| | MemberId | PersonId | MemberStatus | JoinDate | ExpirationDate |
| ▶ | 23785782152363 | 11223 | ACTIVE | 2021-07-24 | 2024-07-24 |
| | 52762523652371 | 67584 | ACTIVE | 2022-05-09 | 2025-05-09 |
| | 66448373731238 | 73789 | ACTIVE | 2022-05-01 | 2025-05-01 |
| | 71281256129356 | 20402 | ACTIVE | 2023-06-17 | 2026-06-17 |
| | 72316513783412 | 12345 | ACTIVE | 2021-01-12 | 2024-01-12 |
| | 76123512351235 | 30643 | ACTIVE | 2022-08-03 | 2025-08-03 |
| | 82191008383976 | 23782 | ACTIVE | 2023-08-21 | 2026-08-21 |
| | 82372617123571 | 81282 | ACTIVE | 2023-05-02 | 2026-05-02 |
| | 82638926286383 | 41451 | ACTIVE | 2022-07-17 | 2025-07-17 |
| | 89126376376365 | 83332 | ACTIVE | 2020-10-24 | 2023-10-24 |
| | 89126891278534 | 77773 | ACTIVE | 2021-03-17 | 2024-03-17 |
| | 89216812821358 | 47556 | ACTIVE | 2023-06-12 | 2026-06-12 |
| | 89268286268361 | 55553 | ACTIVE | 2021-04-01 | 2024-04-01 |
| | 90272862682682 | 23451 | ACTIVE | 2023-05-14 | 2026-05-14 |
| | 92872626376223 | 34726 | ACTIVE | 2022-01-23 | 2025-01-23 |
| | 9126821684801 | 98712 | INACTIVE | 2018-03-13 | 2021-03-13 |
| | 9127238648923 | 30913 | INACTIVE | 2020-09-08 | 2023-09-08 |
| | 72521781251235 | 47472 | INACTIVE | 2020-11-10 | 2023-11-10 |
| | 91268127812363 | 18283 | INACTIVE | 2010-01-07 | 2013-01-07 |
| | 96382621861268 | 32423 | INACTIVE | 2015-02-10 | 2018-02-10 |

| MemberId | PersonId | MemberStatus | JoinDate | ExpirationDate |
|------------------|----------|--------------|------------|----------------|
| 23785782152363 | 11223 | ACTIVE | 2021-07-24 | 2024-07-24 |
| 52762523652371 | 67584 | ACTIVE | 2022-05-09 | 2025-05-09 |
| 66448373731238 | 73789 | ACTIVE | 2022-05-01 | 2025-05-01 |
| 71281256129356 | 20402 | ACTIVE | 2023-06-17 | 2026-06-17 |
| 72316513783412 | 12345 | ACTIVE | 2021-01-12 | 2024-01-12 |
| 76123512351235 | 30643 | ACTIVE | 2022-08-03 | 2025-08-03 |
| 82191008383976 | 23782 | ACTIVE | 2023-08-21 | 2026-08-21 |
| 82372617123571 | 81282 | ACTIVE | 2023-05-02 | 2026-05-02 |
| 82638926286383 | 41451 | ACTIVE | 2022-07-17 | 2025-07-17 |
| 89126891278534 | 77773 | ACTIVE | 2021-03-17 | 2024-03-17 |
| 89216812821358 | 47556 | ACTIVE | 2023-06-12 | 2026-06-12 |
| 89268286268361 | 55553 | ACTIVE | 2021-04-01 | 2024-04-01 |
| 90272862682682 | 23451 | ACTIVE | 2023-05-14 | 2026-05-14 |
| 92872626376223 | 34726 | ACTIVE | 2022-01-23 | 2025-01-23 |
| 9126821684801 | 98712 | INACTIVE | 2018-03-13 | 2021-03-13 |
| 9127238648923 | 30913 | INACTIVE | 2020-09-08 | 2023-09-08 |
| 72521781251235 | 47472 | INACTIVE | 2020-11-10 | 2023-11-10 |
| ▶ 89126376376365 | 83332 | INACTIVE | 2020-10-24 | 2023-10-24 |
| 91268127812363 | 18283 | INACTIVE | 2010-01-07 | 2013-01-07 |
| 96382621861268 | 32423 | INACTIVE | 2015-02-10 | 2018-02-10 |