Project Step 4 DRAFT:

Website URL: https://web.engr.oregonstate.edu/~davisg4/CS340/index.html

---

Team Members:

- Graham Davis
- Josh Gage

Project Title:

- Local Library Information Management System

Feedback from Step 1 Draft:

Feedback by the TAs and peer reviewer:

Peer Feedback #1:

Josh Pulattie

5 days ago

First off, thanks for the 162 nightmares...

- **Does the overview describe what problem is to be solved by a website with DB back end?**
- The overview outlines the problem to be solved adequately. It is built to track the books and people within a small library.
- **Does the overview list specific facts?**
- The overview lists specific facts that the library will handle approximately 6,000 transactions per year. It also lists what other information will be tracked in the database. Depending on the rubrics definition of specific facts, this may suffice. However, it may be beneficial to add in more numerical data points.
- **Are at least four entities described, and does each one represent a single idea to be stored as a list?**
- At least four entities are described in depth throughout the overview. The descriptions are great and help the understanding of the project.

- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?**
- The entity details appear to list all attribute data types, constraints and correction relationships. The description appears to not include Genres which is on the ER.
- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**
- The ER has the relationships but does not include PKs and FKs, which may not be required. It also does not include Order_books_Linked. If I am understanding the diagram correctly, Authors and Books relationship may be backward. It looks to me like one book can have many authors. This may be intentional, but it could also be the case that you've intended to flip this and have a single author having many books.
- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**
- All naming conventions appear to be consistent throughout and look like they match the requirements.

Peer Feedback #2:

Auberon Orbock

5 days ago

Hello,

Nice topic, libraries are very important community resources! Great job on describing the problem and coming up with the required entities. My suggestions would be that the Order_Books_Linked entity, I believe, should be in PascalCase, and be included on the ERD graph, and that the ERD graph could also include more than just the entity names, and have the attributes listed, including showing exactly which PKs link to which FKs. The Authors and Genres entities should probably have a M:M relationships with the Books entity too, as there is not one book per author, or genre, and books can have multiple authors and genres as you indicated in the ERD. Overall looking good!

Peer Feedback #3:

Annabel Wang

4 days ago

Hi! I thought you guys had a solid idea, and the draft was very detailed and descriptive. The overview accurately describes the problem and provides statistics and relationships. There are at least

4 entities, and each of them have a good list of attributes and clear relationships. The 1:1 and M:M relationships also make sense, and I don't have any recommendations for those. The naming conventions also look good! Overall great work, and very straight forward.

Peer Feedback #4:

Josue Bustamante

4 days ago

Hi Group 35! Here is my review!

· **Does the overview describe what problem is to be solved by a website with DB back end?**

Yes, it presents a scenario that is befitting the use of a database with multiple transactions and entities with their respective attributes: A library is an appropriate real-world application considering the multiple objects to be recorded and the transactions it implies.

·**Does the overview list specific facts?**

Yes, it lists the amount of transactions per year and the information the database will be recording. It also includes descriptions of what will be done with the information as implemented in the library's processes such as calculating fines and tracking book availability.

·**Are at least four entities described, and does each one represent a single idea to be stored as a list?**

Yes, the library members, books, authors, orders, and employees are appropriate entities for a library database and each are separate objects or a collection of objects.

· **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?**

Yes, it includes descriptions of each entity as well as the significance of the information being stored. The attributes have the appropriate datatypes although the capitalization is not all uniform. The order books linked description is a bit difficult to understand at first, and it would be helpful to include that it is a linking table in the first description to clarify.

· **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

Yes, the 1:M relationships from Authors to Books and Members to Books are implemented well. The M:M relationship between Book and Orders is intuitive but its implementation through a linking table is a bit unclear, especially since the linking table is not present in the ERD. Additionally, the ERD has a genres table which is not present in the list of entities.

· **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

Yes, all the names have the correct plurality. The names and their capitalizations are also uniform.

TA Feedback:

Hi team, good work on your draft! Some points: 1. How do you think you could add some more numeric data? For example, how many members are you expecting to manage per day? How many reservations? How many books are in your database? (All of these are imaginary, but just help us understand the potential scale and make better design choices). 2. Table names should be written in PascalCase, without underscores. So Order_Books_Linked could be written as OrderBooksLinked or, simply, BookOrders. 3. good work on not setting all values to be required; this allows us to fill in optional fields or update them later after creating our instance of the entity. 4. authorID and genreID should be corrected in the Books table. 5. The ERD table is empty?

- Zoha Zainab Akbarzadeh

Actions based on the feedback:

Based on the peer and TA feedback, we updated the naming convention to PascalCase, added in more numeric data in the project outline / overview, updated the ERD table to include attributes, and updated the Database Outline to include the entities Genres, BooksAuthorsLinked, and a BooksGenresLinked along with the associated attributes for each. We also removed the foreign keys from the Books entity attributes since these will actually need to go through the linking table due to the M:M relationships.

The only specific note we did not address was adding in the linking tables to the ERD. Based on our understanding of the ERD, the linked tables are implied by the M:M relationships and do not need to be added in. However, when the database schematic is created, it will include these linking entities.

Upgrades to the Draft version:

All of the updates and upgrades we have made have been referenced in the above section. Either directly or indirectly, all changes were made based on our review of the feedback provided.

Feedback from Step 2 Draft:

Feedback by the TAs and peer reviewer:

TA Feedback

- Add Commands: Include SET FOREIGN_KEY_CHECKS and AUTO_COMMIT in your DDL.
- FK Display: Ensure foreign keys are visible in the schema.

Peer Feedback

Christopher Blum

- CASCADE Operations: Consider adding these for better data management.

Katie Kimura

- Attribute Naming: Use camelCase for attributes (e.g., bookID).
- Data Type Suggestions: Consider YEAR for YearPublished.
- CASCADE Operations: Implement for related table deletions.

Russell Myers

- CASCADE Operations: Add for Member or Employee in Orders table to handle deletions

---

Feedback from Step 3 Draft:

Feedback by the TAs and peer reviewer:

TA Feedback:

- Hi team, good work on keeping your PDF updated! Some pointers: 1. I am unsure if this is what you intended, but there is no database information on the currently provided link as I'm looking at it! I will assume it worked previously per comments, but please fix this as soon as possible.
  - Zoha Zainab Akbarzadeh

Peer Feedback

Steven Cleasby-Mayeda3d

Hey Graham and Joshua,

Does the UI utilize a SELECT for every table in the schema?

Yes, there are queries to select each table in the DML.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

It seems all the selects simply select all the rows in each table.

Does the UI implement an INSERT for every table in the schema?

Not in the UI on the website, there are insert queries in the DML doc.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

Yes, the insert queries are correct in the DML sql.

Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship?

There are delete queries in the doc but, there is no delete for the intersection table.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

There are update queries for updating tables in the database.

Is at least one relationship NULLable?

It seems so, there are foreign keys that can be NULL defined in the DDL doc. The UI simply needs to allow NULL as an option in entry.

Do you have any other suggestions for the team to help with their HTML UI?

Currently, they still need to implement the tables and interaction with the backend.

Reply

R

Russell Myers3d

- *Does the UI utilize a SELECT for every table in the schema?* In other

The UI utilizes a SELECT for every table in the schema from Members, Books, Authors, Employees, orders, to Genres. Each table has a link to select from on the UI that will connect to the coresponding database table in future implementations.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

No, there is no search or filter functionality. The UI should support such a feature in further implementation.

- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.

The UI input fields are not implemented for each INSERT in the table schema. This should be implemented in future releases.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

The m:m relatioships of BooksAuthorsLinked, BooksGenresLinked and OrderBooksLinked are in the schema. Each Insert query include the corresponsing FK attributes.

- *Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Theres DELETE queries for each table and they do have removal without deleting from the entities. Although the UI doesnt show any delete functionalities. Ex, Authors: 'DELETE FROM Authors WHERE AuthorID = :authorID;'

- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

There are UPDATE queries, for each table. Ex, UPDATE Authors
SET Name = ':name', BirthDate = ':birthDate', Nationality = ':nationality'
WHERE AuthorID = :authorID;

- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus, it should be feasible to edit an Order and change the value of Employee to be empty.

Yes the nullable functionality is implemented in the table schema and should reflect as in the given "e.g." example. If you wanted to implement another NULLable feature, 'OrderBooksLinked' table could have 'ReturnedDate' NULL which would allow an empty value to be blank in the case the book wasnt returned.

- *Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.*

When implementing the UI/html, it would be could User functionality to give confirmation dialog when the user attempts to use the DELETE operation. This would limit mistakes and insure no accidental data loss. Renaiming table aliases for better recognition is good practice as well, for example, Members.Name could be 'Member Name'.

Reply

M

Monica Zhang5d

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

- The UI has 7 pages with a one sentence description on each page but no tables are displayed. The pages include the entities on the ERD: Home, Books, Authors, Members, Employees, Orders, and Genre. The schema shows 9 tables and should have 9 tables displayed on the UI.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

- There is no search/filter implemented in the UI.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

- There is currently no table displayed on the UI and no INSERT feature implemented. The data manipulation file has an INSERT query for 6 tables: Member, Books, Authors, Orders, Employees, and Genre.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

- The HTML UI does not have an INSERT feature that adds the FK attributes. In the Order table, there is a query that adds the corresponding foreign keys memberID and employeeID.

Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

- There is no DELETE implementation in the HTML UI. There are 6 delete queries for the data manipulation file.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

- No UPDATE feature in the UI but there are 6 UPDATE queries in the data manipulation file.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus, it should be feasible to edit an Order and change the value of Employee to be empty.

- All the foreign keys are defined as NOT NULL. In the ERD Orders entity, MemberID and Employee ID are defined as NOT NULL. In the Books entity, AuthorID and GenreID are defined as NOT NULL.

Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.

- There are currently 6 pages set up to display the tables for Books, Authors, Members, Employees, Orders, and Genres. The provided schema shows that there are a total of 9 tables, therefore data from all 9 tables, including the intersection tables should be displayed on the HTML UI. Also, make sure to add a drop-down filter/search to select foreign key IDs.

Reply

C

Chris Allen5d

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Currently the UI doesn't have table views implemented for any table at the moment. Group35_DML.sql does have individual SELECT queries for each table.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

None of the SELECT queries in Group35_DML.sql use a search filter.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

The UI currently does not have an implementation to add items to a table. However the Group35_DML.sql file has example INSERT commands to eventually be used by the UI.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products,

INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

INSERT commands in Group35_DML.sql for tables that require a FK currently only have placeholders for those FKs (i.e. :memberID and :orderID) and don't specify how those IDs will be obtained, either by subquery or their specific ID. There is no reference query in the DML for the M:M Intersectiont table, but there is an INSERT example for this table in the Group_35_DDL.sql file.

Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

DELETE examples exist for each table in the DML file, save for the M:M relation table, BookGenresLinked. DELETE examples for Books and Genres make no mention of the intersection table.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Each table referenced in the Group35_DML.sql file has a corresponding UPDATE command.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus, it should be feasible to edit an Order and change the value of Employee to be empty.

All foreign keys in the Group35_DDL.sql file are specified as NOT NULL, so currently all relationships are not optional.

Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.

I see that you still have a lot of work to do to finish implementing the UI. I'm in the same boat as well. Just keep chugging away at it. While that is in process, you should also keep working out how the backend queries will work. As I've been working on the implementation, I keep discovering things I didn't consider and realized it's going to be a bunch of more work to finish, but don't let that discourage you!

Actions based on the feedback:

Based on the TA feedback we fixed the UI to include the example sample data. Based on the peer feedback from peers, we added in a DML query to allow a delete on a M:N table (BooksAuthorsLinked

Table) and allowed the foreign key in Books of Genre to be Null) in the DDL file.

Project Outline:

The Local Library Information Management System aims to streamline the operations of a small local library that handles approximately 6,000 transactions per year. The system will manage the daily interactions of around 500 members and handle roughly 50 transactions per day. The database will store information on roughly 2,000 books across 5 or more genres and will also maintain records of current library members and employees, as well as the authors of the available titles. We are expecting to keep data on 1,000 or more authors, and 5 or more employees. This comprehensive data collection will assist in making informed decisions regarding the library's operations and services.

Database Outline:

Entities and Attributes:

1. **Members: keeps track of the library members. Members can have multiple orders of books so long as they are in good standing. Fines calculated on the orders are rolled up into the current fines and are tracked by member. If the current fines become too large, the member will be marked as not in good standing, and will not be able to check out any additional books.**
    - MemberID: int, auto_increment, unique, not NULL, PK
    - Email: varchar, not NULL
    - Name: varchar, not NULL
    - Phone: varchar, NULL
    - Address: varchar, NULL
    - Standing: varchar, NULL
    - CurrentFines: decimal(10,2), NULL

Relationship: One-to-many with Orders (customerID as FK in Orders)

2. **Books: keeps track of the library books and their availability. Books can be in multiple orders. Books are tracked for availability with the boolean markers in onHold and checkedOut attributes.**
    - BookID: int, auto_increment, unique, not NULL, PK
    - Title: varchar, not NULL
    - ISBN: varchar, not NULL
    - YearPublished: int, not NULL
    - Publisher: varchar, not NULL
    - PageCount: int, not NULL

- Language: varchar, not NULL
- OnHold: boolean, not NULL
- CheckedOut: boolean, Not NULL

Relationship: Many-to-many with Orders through OrderBooksLinked

3. **Authors**: **Keeps track of the authors of the available books. An author can write many books, and a book can have multiple authors. A query can be run by an AuthorID to see their list of books and the books' availability.**
    - AuthorID: int, auto_increment, unique, not NULL, PK
    - Name: varchar, not NULL
    - BirthDate: date, NULL
    - Nationality: varchar, NULL

Relationships: Many-to-many with Books through the BooksAuthorsLinked

4. **BooksAuthorsLinked: Links the Authors to Books since there is a many to many relationship between these two tables.**
    - BooksAuthorID: int, auto_increment, unique, not NULL, PK
    - BookID: int, not NULL, FK referencing Books
    - AuthorID: int, not NULL, FK referencing Authors

Relationship: Acts as a linking table between Authors and Books

5. **Orders: Keeps track of an order. An order can have multiple books (through the OrderBooksLinked table) and must have a memberID and employeeID associated.**
    - OrderID: int, auto_increment, unique, not NULL, PK
    - MemberID: int, not NULL, FK referencing Members
    - EmployeeID: int, not NULL, FK referencing Employee
    - OrderDate: date, not NULL

Relationship: Many-to-many with Books through OrderBooksLinked

6. **OrderBooksLinked: Links the Orders to Books since there is a many to many relationship between these two tables. Keeps track of all the line item level data on a given order for a given book.**
    - OrderBooksID: int, auto_increment, unique, not NULL, PK
    - OrderID: int, not NULL, FK referencing Orders
    - BookID: int, not NULL, FK referencing Books
    - CheckoutDate: date, NULL
    - DueDate: date, not NULL
    - ReturnedDate: date, NULL
    - FineIncurred: decimal(10,2), NULL

Relationship: Acts as a linking table between Orders and Books

7. **Employees: Keeps track of the current employees. Each order must have an associated employee.**
   - EmployeeID: int, auto_increment, unique, not NULL, PK
   - Name: varchar, not NULL
   - Position: varchar, not NULL
   - Email: varchar, not NULL
   - Phone: varchar, NULL

Relationship: One-to-many with Orders

8. **Genres: Keeps a listing of book genres. A book can have multiple genres, and a genre can have multiple books. They are connected with the BooksGenresLinked entity.**
   - GenreID: varchar, unique, not NULL, PK
   - GenreDescription: varchar, NULL

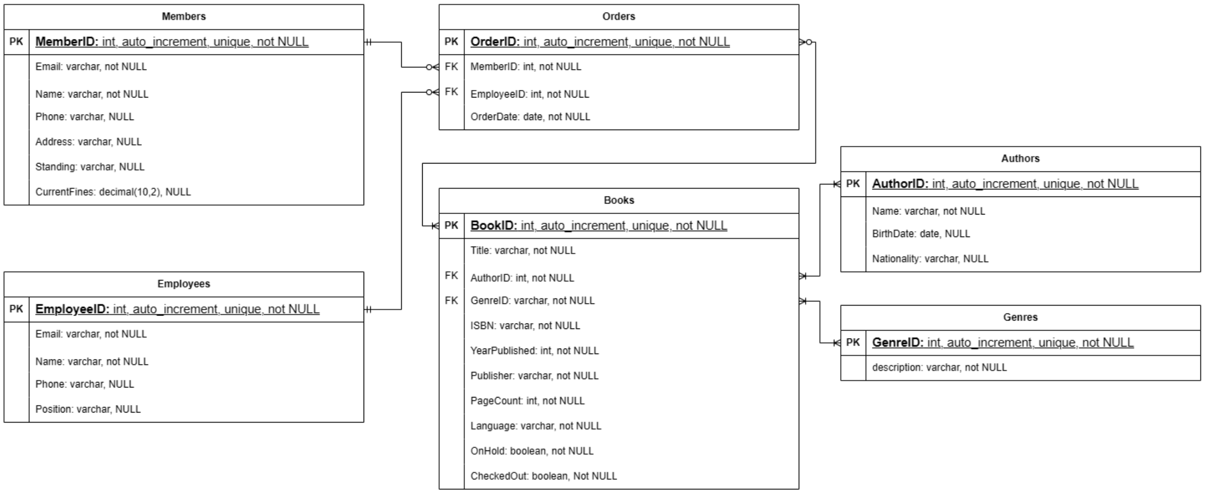
Relationship: Many-to-many with Books through BooksGenresLinked

9. **BooksGenresLinked: Links the Genres to Books since there is a many to many relationship between these two tables.**
   - BooksGenreID: int, auto_increment, unique, not NULL, PK
   - BookID: int, not NULL, FK referencing Books
   - GenreID: int, not NULL, FK referencing Genres

Relationship: Acts as a linking table between Genres and Books

---

Entity-Relationship Diagram (ERD)



Members

PK **MemberID:** int, auto_increment, unique, not NULL

Email: varchar, not NULL

Name: varchar, not NULL

Phone: varchar, NULL

Address: varchar, NULL

Standing: varchar, NULL

CurrentFines: decimal(10,2), NULL

Orders

PK **OrderID:** int, auto_increment, unique, not NULL

FK MemberID: int, not NULL

FK EmployeeID: int, not NULL

OrderDate: date, not NULL

Authors

PK **AuthorID:** int, auto_increment, unique, not NULL

Name: varchar, not NULL

BirthDate: date, NULL

Nationality: varchar, NULL

Books

PK **BookID:** int, auto_increment, unique, not NULL

Title: varchar, not NULL

FK AuthorID: int, not NULL

FK GenreID: varchar, not NULL

ISBN: varchar, not NULL

YearPublished: int, not NULL

Publisher: varchar, not NULL

PageCount: int, not NULL

Language: varchar, not NULL

OnHold: boolean, not NULL

CheckedOut: boolean, Not NULL

Genres

PK **GenreID:** int, auto_increment, unique, not NULL

description: varchar, not NULL

Employees

PK **EmployeeID:** int, auto_increment, unique, not NULL

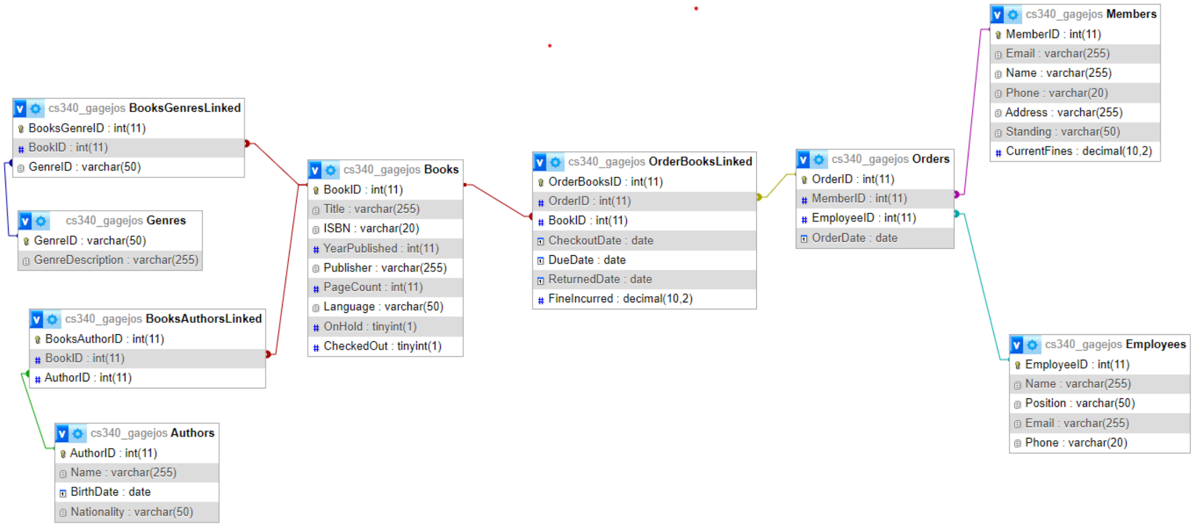
Email: varchar, not NULL

Name: varchar, not NULL

Phone: varchar, NULL

Position: varchar, NULL

Schema:



cs340_gagejos **Members**
MemberID : int(11)
Email : varchar(255)
Name : varchar(255)
Phone : varchar(20)
Address : varchar(255)
Standing : varchar(50)
CurrentFines : decimal(10,2)

cs340_gagejos **BooksGenresLinked**
BooksGenreID : int(11)
BookID : int(11)
GenreID : varchar(50)

cs340_gagejos **Books**
BookID : int(11)
Title : varchar(255)
ISBN : varchar(20)
YearPublished : int(11)
Publisher : varchar(255)
PageCount : int(11)
Language : varchar(50)
OnHold : tinyint(1)
CheckedOut : tinyint(1)

cs340_gagejos **OrderBooksLinked**
OrderBooksID : int(11)
OrderID : int(11)
BookID : int(11)
CheckoutDate : date
DueDate : date
ReturnedDate : date
FineIncurred : decimal(10,2)

cs340_gagejos **Orders**
OrderID : int(11)
MemberID : int(11)
EmployeeID : int(11)
OrderDate : date

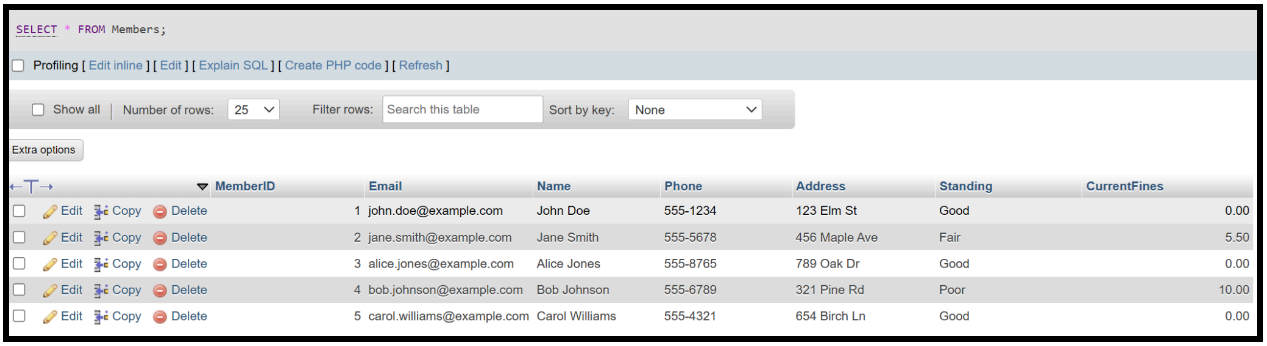
cs340_gagejos **Genres**
GenreID : varchar(50)
GenreDescription : varchar(255)

cs340_gagejos **BooksAuthorsLinked**
BooksAuthorID : int(11)
BookID : int(11)
AuthorID : int(11)

cs340_gagejos **Employees**
EmployeeID : int(11)
Name : varchar(255)
Position : varchar(50)
Email : varchar(255)
Phone : varchar(20)

cs340_gagejos **Authors**
AuthorID : int(11)
Name : varchar(255)
BirthDate : date
Nationality : varchar(50)

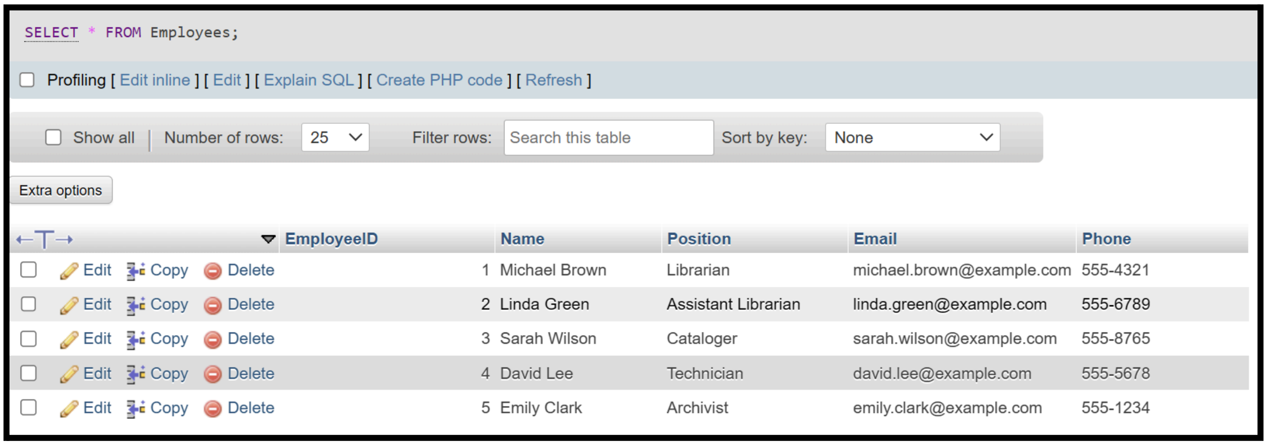
Example Data:



```
SELECT * FROM Members;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

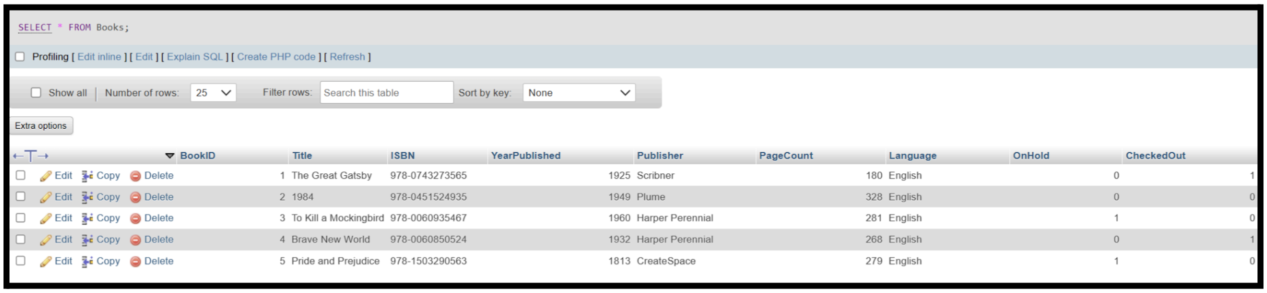
←T→	MemberID	Email	Name	Phone	Address	Standing	CurrentFines
□ Edit Copy Delete	1	john.doe@example.com	John Doe	555-1234	123 Elm St	Good	0.00
□ Edit Copy Delete	2	jane.smith@example.com	Jane Smith	555-5678	456 Maple Ave	Fair	5.50
□ Edit Copy Delete	3	alice.jones@example.com	Alice Jones	555-8765	789 Oak Dr	Good	0.00
□ Edit Copy Delete	4	bob.johnson@example.com	Bob Johnson	555-6789	321 Pine Rd	Poor	10.00
□ Edit Copy Delete	5	carol.williams@example.com	Carol Williams	555-4321	654 Birch Ln	Good	0.00



```
SELECT * FROM Employees;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

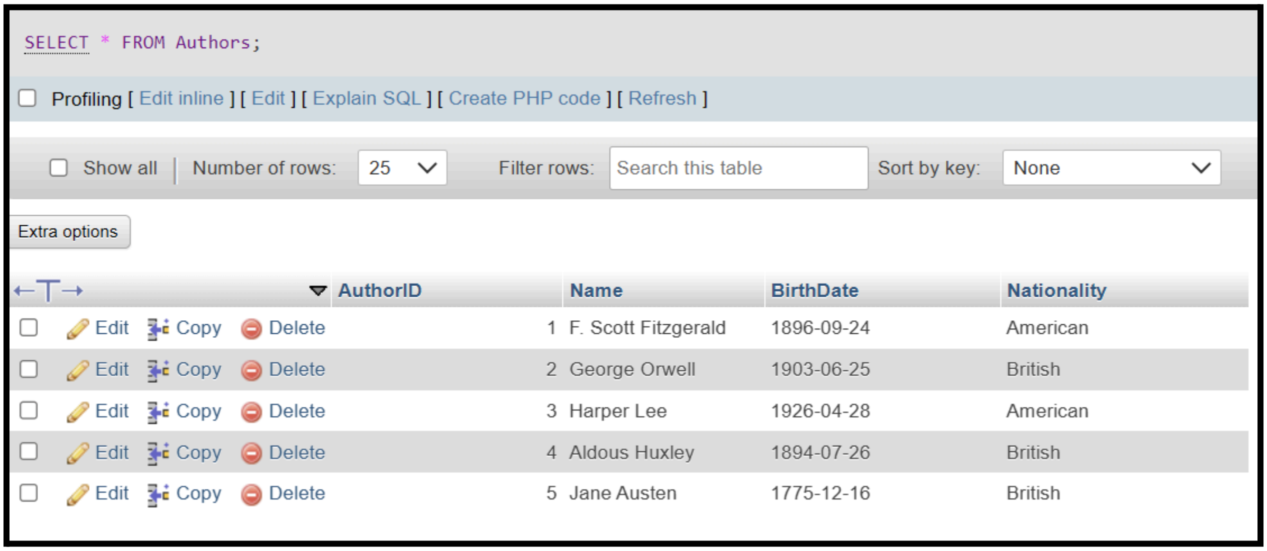
←T→	EmployeeID	Name	Position	Email	Phone
□ Edit Copy Delete	1	Michael Brown	Librarian	michael.brown@example.com	555-4321
□ Edit Copy Delete	2	Linda Green	Assistant Librarian	linda.green@example.com	555-6789
□ Edit Copy Delete	3	Sarah Wilson	Cataloger	sarah.wilson@example.com	555-8765
□ Edit Copy Delete	4	David Lee	Technician	david.lee@example.com	555-5678
□ Edit Copy Delete	5	Emily Clark	Archivist	emily.clark@example.com	555-1234



```
SELECT * FROM Books;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

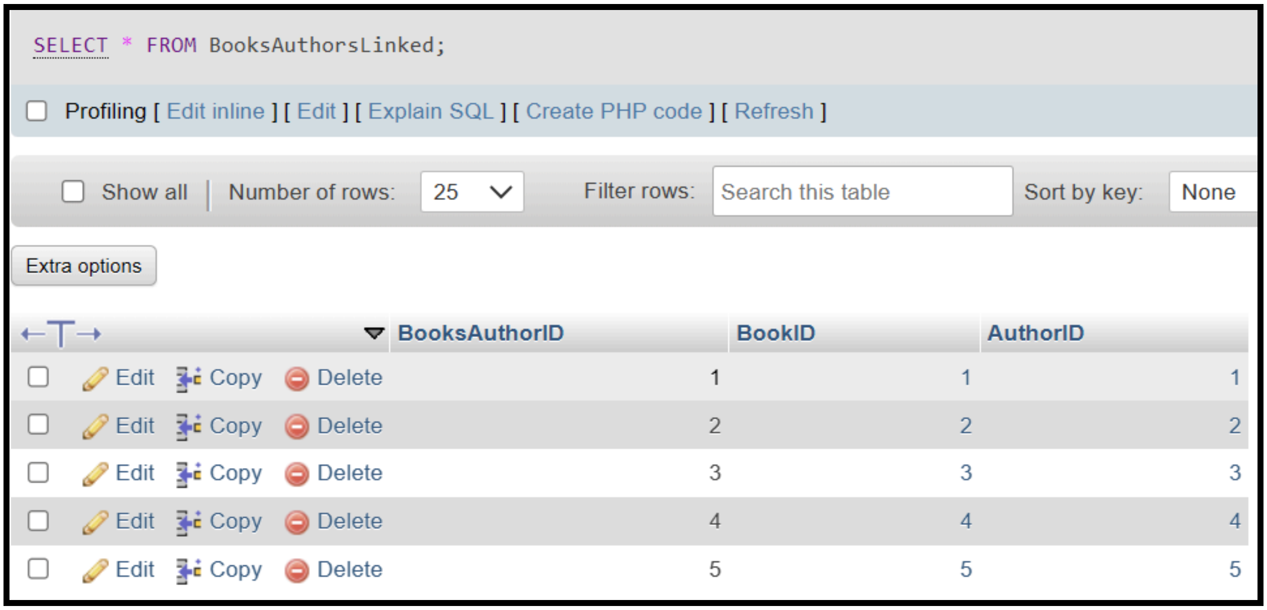
←T→	BookID	Title	ISBN	YearPublished	Publisher	PageCount	Language	OnHold	CheckedOut
□ Edit Copy Delete	1	The Great Gatsby	978-0743273565	1925	Scribner	180	English	0	1
□ Edit Copy Delete	2	1984	978-0451524935	1949	Plume	328	English	0	0
□ Edit Copy Delete	3	To Kill a Mockingbird	978-0060935467	1960	Harper Perennial	281	English	1	0
□ Edit Copy Delete	4	Brave New World	978-0060850524	1932	Harper Perennial	268	English	0	1
□ Edit Copy Delete	5	Pride and Prejudice	978-1503290563	1813	CreateSpace	279	English	1	0



```
SELECT * FROM Authors;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

←T→	AuthorID	Name	BirthDate	Nationality
□ Edit Copy Delete	1	F. Scott Fitzgerald	1896-09-24	American
□ Edit Copy Delete	2	George Orwell	1903-06-25	British
□ Edit Copy Delete	3	Harper Lee	1926-04-28	American
□ Edit Copy Delete	4	Aldous Huxley	1894-07-26	British
□ Edit Copy Delete	5	Jane Austen	1775-12-16	British

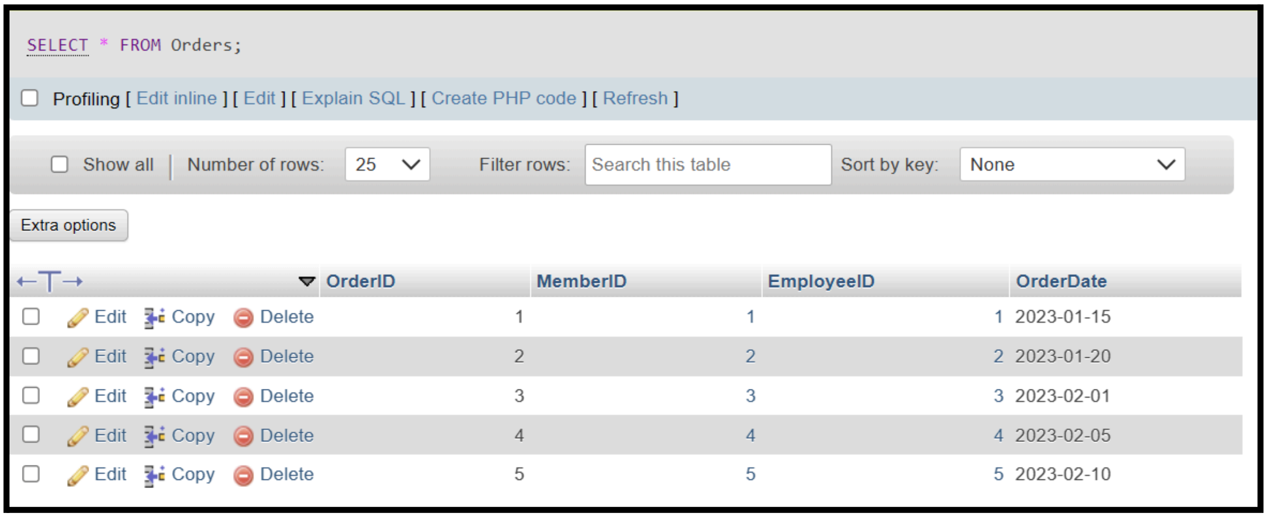


```
SELECT * FROM BooksAuthorsLinked;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options
←T→

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	BooksAuthorID	BookID	AuthorID
□ Edit Copy Delete	1	1	1
□ Edit Copy Delete	2	2	2
□ Edit Copy Delete	3	3	3
□ Edit Copy Delete	4	4	4
□ Edit Copy Delete	5	5	5



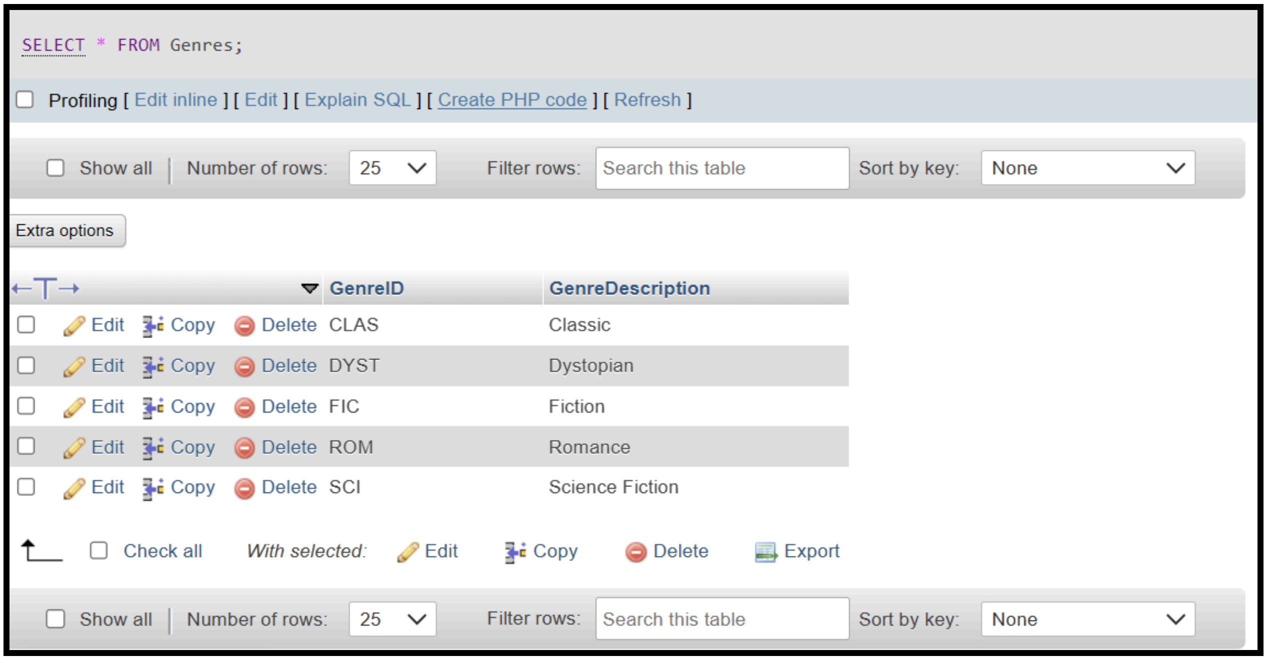
```
SELECT * FROM Orders;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options
←T→

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	OrderID	MemberID	EmployeeID	OrderDate
□ Edit Copy Delete	1	1	1	2023-01-15
□ Edit Copy Delete	2	2	2	2023-01-20
□ Edit Copy Delete	3	3	3	2023-02-01
□ Edit Copy Delete	4	4	4	2023-02-05
□ Edit Copy Delete	5	5	5	2023-02-10

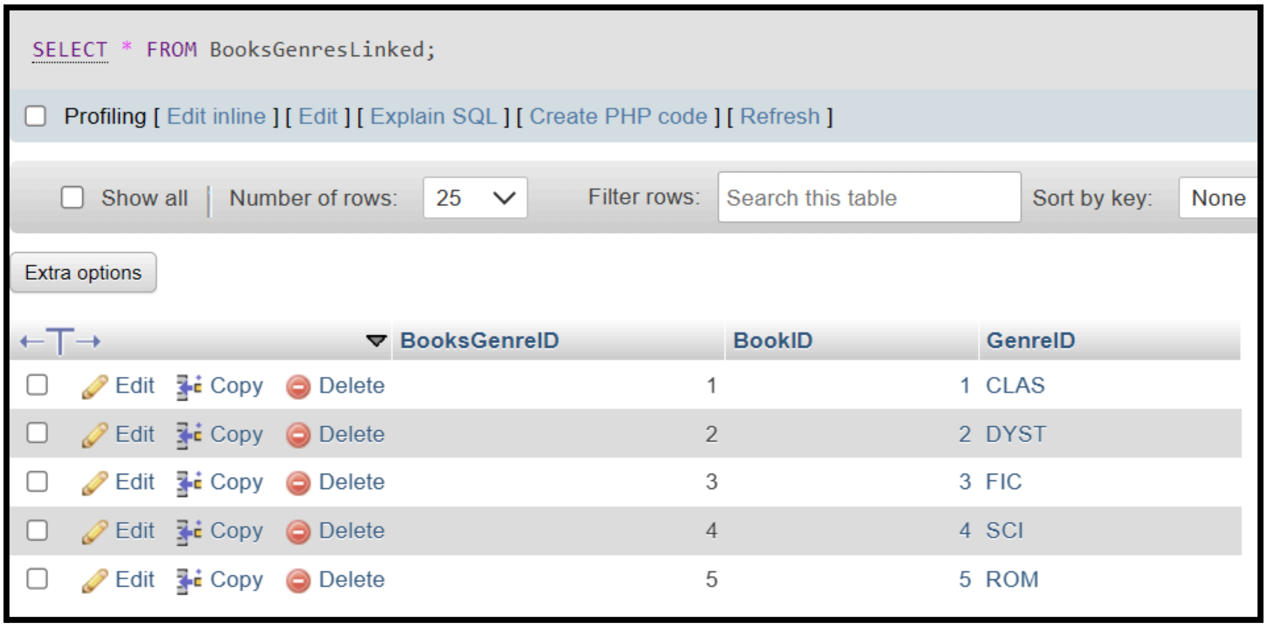
Ord



```
SELECT * FROM Genres;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

GenreID	GenreDescription
CLAS	Classic
DYST	Dystopian
FIC	Fiction
ROM	Romance
SCI	Science Fiction



```
SELECT * FROM BooksGenresLinked;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

BooksGenreID	BookID	GenreID
1	1	CLAS
2	2	DYST
3	3	FIC
4	4	SCI
5	5	ROM