

Context

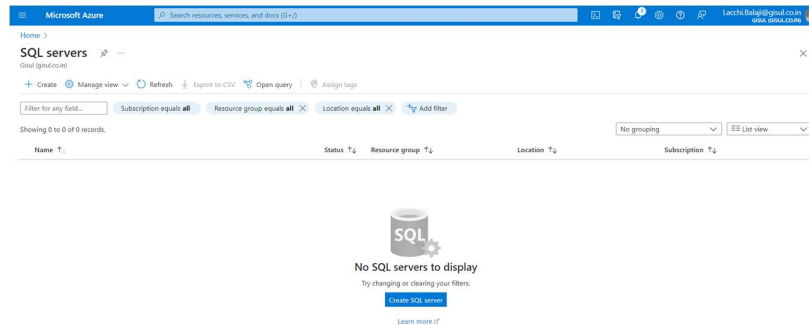
| | |
|---------------------|---------|
| Create SQL Server | 2 – 4 |
| Create SQL database | 4 – 5 |
| Table Creations | 6 – 9 |
| Insert Data | 9 – 12 |
| Grant | 13 |
| Revoke | 13 |
| Update | 14 |
| Truncate | 14 |
| Filtering Data | 15 |
| Group by | 15 |
| Having | 16 |
| Sorting Data | 16 – 17 |

Create SQL Server

1. Login to Microsoft Azure Portal Account
2. On the Home page click on SQL servers. If you don't see the SQL Server click on Create a resource and search for SQL Server.



3. Click on Create SQL Server as shown below.



4. Under basic give the below properties as shown below.

Basics Networking Additional settings Tags Review + create

SQL database server is a logical container for managing databases and elastic pools. Complete the Basic tab, then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Server details

Enter required settings for this server, including providing a name and location.

Server name * ☒ .database.windows.net

Location *

5. Scroll down and give the login name and password then click on the Next option.

Authentication

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Azure AD authentication [Learn more](#) or using an existing Azure AD user, group, or application as Azure AD admin [Learn more](#), or select both SQL and Azure AD authentication.

Authentication method

☐ Use only Azure Active Directory (Azure AD) authentication

☐ Use both SQL and Azure AD authentication

☒ Use SQL authentication

Server admin login * ☒

Password * ☒

Confirm password * ☒

[Review + create](#) [Next: Networking >](#)

6. Set the below properties and click on Review+Create

Create SQL Database Server

Microsoft

[Basics](#) [Networking](#) [Additional settings](#) [Tags](#) [Review + create](#)

Configure networking access for your server.

Firewall rules

Allow Azure services and resources to access this server

☒ Yes ☐ No

Review + create

< Previous

Next : Additional settings >

7. Next click on Create.

[Basics](#) [Networking](#) [Additional settings](#) [Tags](#) [Review + create](#)

Product details

SQL Database Server
by Microsoft
[Terms of use](#) | [Privacy policy](#)

Estimated cost per month
No additional charges

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and third-party offerings. For additional details see [Azure Marketplace Terms](#).

Basics

| | |
|-----------------------|---------------------------|
| Subscription | Microsoft Partner Network |
| Resource group | ADFResource |
| Server name | adfsqservertrail |
| Authentication method | SQL authentication |
| Server admin login | adminsql |
| Location | East US |

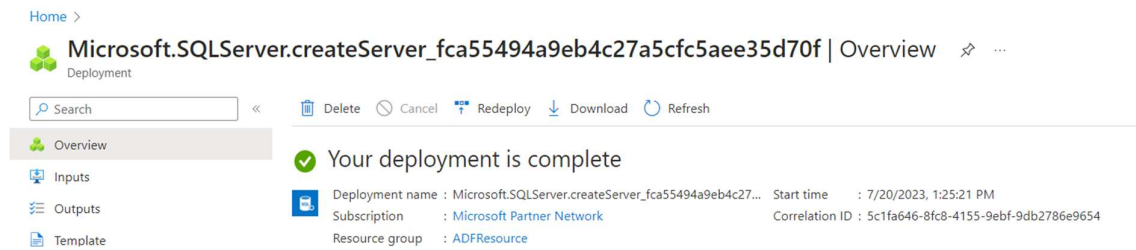
Networking

Create

< Previous

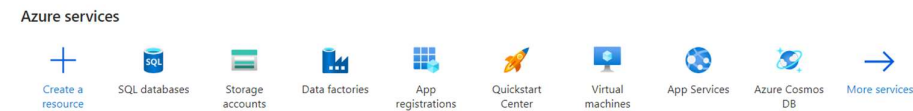
[Download a template for automation](#)

8. After some time you will see a screen like below.

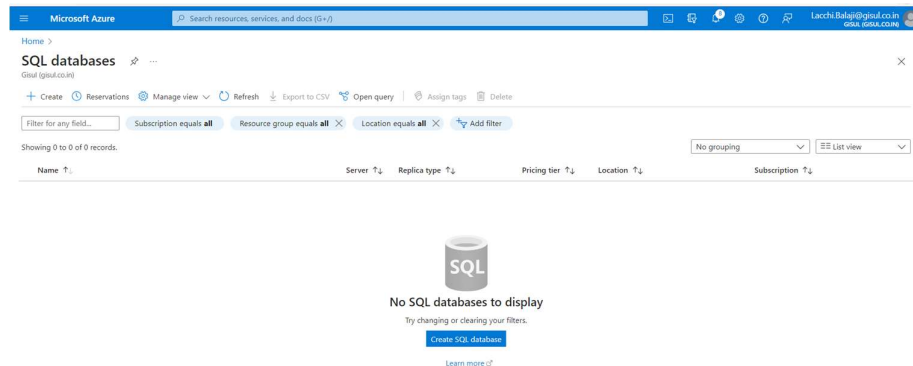


Create SQL database

1. On the Home page click on SQL database.



2. Next click on Create SQL database.



3. Give the Database name as sqldb and select the server that we created before then set the properties as shown below. And click on Review+Create.

Create SQL Database

Microsoft

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name *

Server * [Create new](#)

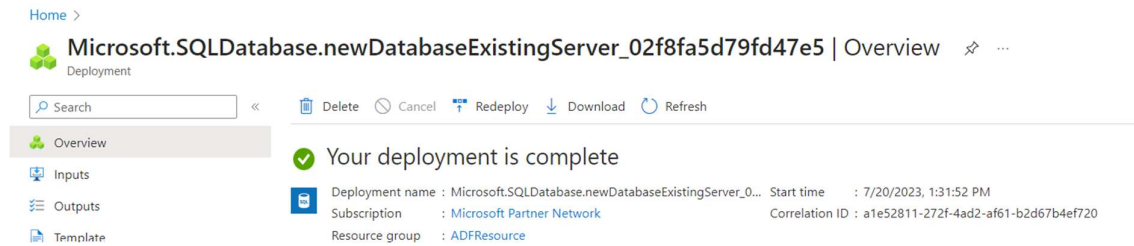
Want to use SQL elastic pool? ☐ Yes ☒ No

Workload environment ☒ Development ☐ Production

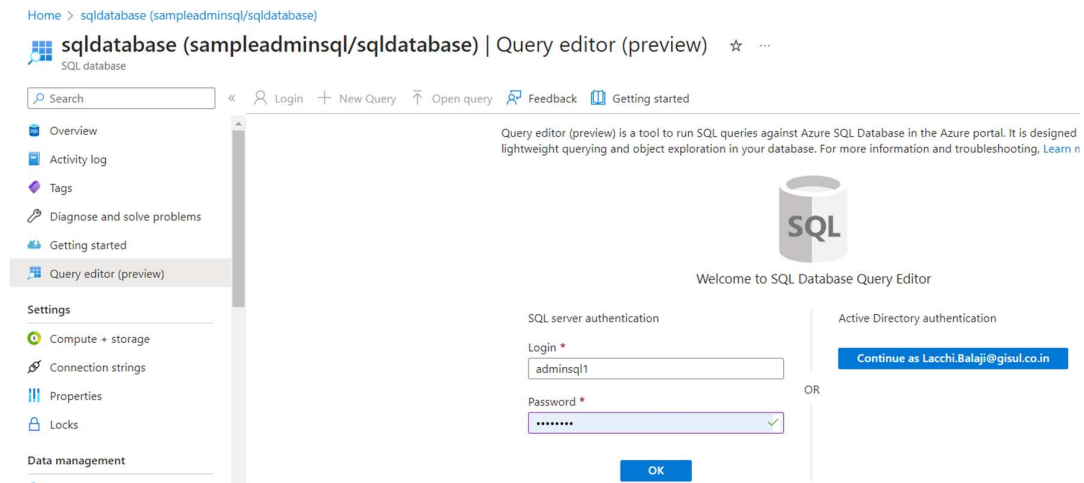
Default settings provided for Development workloads. Configurations can be modified as needed.

[Review + create](#) [Next: Networking >](#)

- Next click on Create.
- After some time you will see the below screen as shown.



- Now click on the SQL Database that we created just now.
- Go to Query editor, give login credentials that we created in the sql server click on Ok.
- If it asks to allow access to an IP address allow it.



- Next you will see a Query window like below where we create our tables, views stored procedure etc.

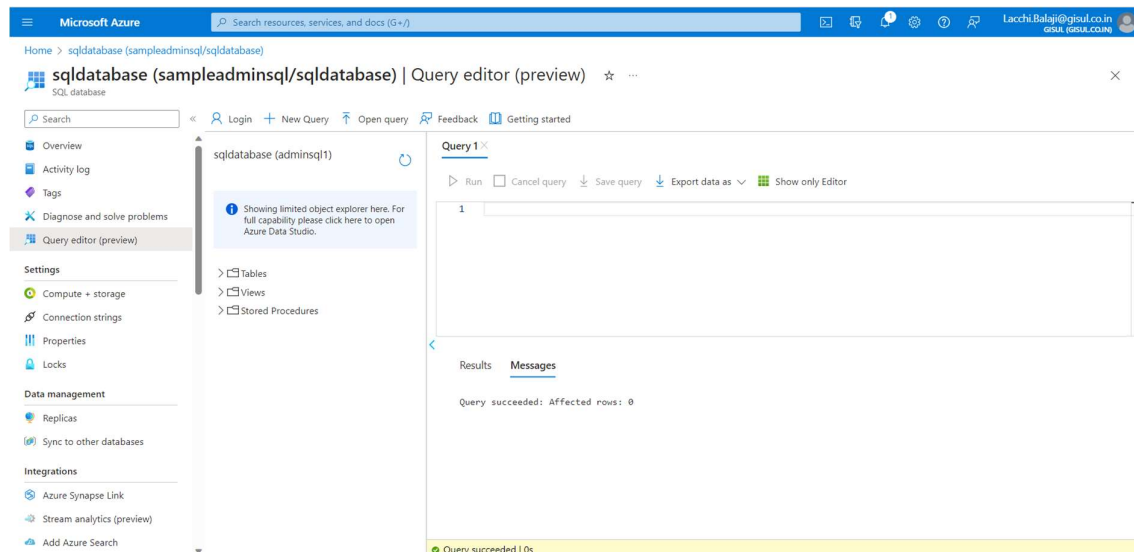




Table Creations

Exercise: Create and implement schema as per the ER Model design in the previous module and apply Constraints.

1. Now here we are creating an Author Table. Use the below query to create.

Query: CREATE TABLE Author (
 AuthorID INT PRIMARY KEY,
 FirstName VARCHAR(50),
 LastName VARCHAR(50),
 DateOfBirth DATE,
 Nationality VARCHAR(50)
);

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as

```
1 CREATE TABLE Author (  
2     AuthorID INT PRIMARY KEY,  
3     FirstName VARCHAR(50),  
4     LastName VARCHAR(50),  
5     DateOfBirth DATE,  
6     Nationality VARCHAR(50)  
7 );
```



Results Messages

Query succeeded: Affected rows: 0

2. Next we will create a Publisher table by using the below Query.

Query: CREATE TABLE Publisher (
 PublisherID INT PRIMARY KEY,
 Name VARCHAR(100),
 Location VARCHAR(100)
);

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as

```
1 CREATE TABLE Publisher (  
2     PublisherID INT PRIMARY KEY,  
3     Name VARCHAR(100),  
4     Location VARCHAR(100)  
5 );
```



Results Messages

Query succeeded: Affected rows: 0

3. Next create a customer table by using the below query.

Query: CREATE TABLE Customer (
CustomerID INT PRIMARY KEY,
FirstName VARCHAR(50),
LastName VARCHAR(50),
Email VARCHAR(100),
PhoneNumber VARCHAR(20),
Address VARCHAR(200)
);

Query 1 ✕

Run ☐ Cancel query  Save query  Ex

```
1 CREATE TABLE Customer (  
2     CustomerID INT PRIMARY KEY,  
3     FirstName VARCHAR(50),  
4     LastName VARCHAR(50),  
5     Email VARCHAR(100),  
6     PhoneNumber VARCHAR(20),  
7     Address VARCHAR(200)  
8 );
```

Results Messages

Query succeeded: Affected rows: 0

4. Create Book table and use Foreign key to make connection with Publisher table by using below Query.

Query: CREATE TABLE Book (
ISBN VARCHAR(20) PRIMARY KEY,
Title VARCHAR(200),
Genre VARCHAR(50),
PublicationYear INT,
Price DECIMAL(10, 2),
PublisherID INT,
FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID)
);

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 CREATE TABLE Book (  
2     ISBN VARCHAR(20) PRIMARY KEY,  
3     Title VARCHAR(200),  
4     Genre VARCHAR(50),  
5     PublicationYear INT,  
6     Price DECIMAL(10, 2),  
7     PublisherID INT,  
8     FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID)  
9 );
```




Results Messages

Query succeeded: Affected rows: 0

5. Create an Order table by using the below Query.

```
Query: CREATE TABLE [Order] (  
    OrderID INT PRIMARY KEY,  
    OrderDate DATE,  
    TotalAmount DECIMAL(10, 2),  
    CustomerID INT,  
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
);
```

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as  Show only Ed

```
1 CREATE TABLE [Order](  
2     OrderID INT PRIMARY KEY,  
3     OrderDate DATE,  
4     TotalAmount DECIMAL(10, 2),  
5     CustomerID INT,  
6     FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
7 );
```





Results Messages

Query succeeded: Affected rows: 0

6. Next create BookAuthor table by using below query.

```
Query: CREATE TABLE BookAuthor (  
    ISBN VARCHAR(20),  
    AuthorID INT,  
    PRIMARY KEY (ISBN, AuthorID),  
    FOREIGN KEY (ISBN) REFERENCES Book(ISBN),  
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID)  
);
```

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 CREATE TABLE BookAuthor (  
2     ISBN VARCHAR(20),  
3     AuthorID INT,  
4     PRIMARY KEY (ISBN, AuthorID),  
5     FOREIGN KEY (ISBN) REFERENCES Book(ISBN),  
6     FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID)  
7 );
```




Results Messages

Query succeeded: Affected rows: 0

7. Create our last table OrderBook by using the below query.

Query: CREATE TABLE OrderBook (
 OrderID INT,
 ISBN VARCHAR(20),
 PRIMARY KEY (OrderID, ISBN),
 FOREIGN KEY (OrderID) REFERENCES Order(OrderID),
 FOREIGN KEY (ISBN) REFERENCES Book(ISBN)
);

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show

```
1 CREATE TABLE OrderBook (  
2     OrderID INT,  
3     ISBN VARCHAR(20),  
4     PRIMARY KEY (OrderID, ISBN),  
5     FOREIGN KEY (OrderID) REFERENCES [Order](OrderID),  
6     FOREIGN KEY (ISBN) REFERENCES Book(ISBN)  
7 );
```

Results Messages

Query succeeded: Affected rows: 0

Insert Data

1. Insert data into the Author table by using the below query.

Query: INSERT INTO Author (AuthorID, FirstName, LastName, DateOfBirth, Nationality)
VALUES
 (1, 'J.K.', 'Rowling', '1965-07-31', 'British'),
 (2, 'George', 'Orwell', '1903-06-25', 'British')

2. To check the Data, use the below query.

Query: Select * from Author;

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO Author (AuthorID, FirstName, LastName, DateOfBirth, Nationality)  
2 VALUES  
3     (1, 'J.K.', 'Rowling', '1965-07-31', 'British'),  
4     (2, 'George', 'Orwell', '1903-06-25', 'British')  
5  
6 select * from Author;
```

Results Messages

 Search to filter items...

| AuthorID | FirstName | LastName | DateOfBirth | Nationality |
|----------|-----------|----------|-------------------------------|-------------|
| 1 | J.K. | Rowling | 1965-07-31T00:00:00.000000... | British |
| 2 | George | Orwell | 1903-06-25T00:00:00.000000... | British |

3. Insert data into the Publisher table by using the below Query.




Query: INSERT INTO Publisher (PublisherID, Name, Location)
VALUES

(1, 'Penguin Books', 'London'),
(2, 'HarperCollins', 'New York')

4. To check the Data, use the below query.

Query: Select * from Publisher;

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO Publisher (PublisherID, Name, Location)
2 VALUES
3     (1, 'Penguin Books', 'London'),
4     (2, 'HarperCollins', 'New York')
5
6 Select * from Publisher
```

Results Messages

Search to filter items...

| PublisherID | Name | Location |
|-------------|---------------|----------|
| 1 | Penguin Books | London |
| 2 | HarperCollins | New York |

5. Insert data into the Customer table by using the below Query.

Query: INSERT INTO Customer (CustomerID, FirstName, LastName, Email, PhoneNumber, Address)




VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),
(2, 'Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '456 Elm St')

6. To check the Data, use the below query.

Query: Select * from Customer;

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO Customer (CustomerID, FirstName, LastName, Email, PhoneNumber, Address)
2 VALUES
3     (1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),
4     (2, 'Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '456 Elm St')
5
6 Select * from Customer;
```

Results Messages

Search to filter items...

| CustomerID | FirstName | LastName | Email | PhoneNumber | Address |
|------------|-----------|----------|-------------------------|--------------|-------------|
| 1 | John | Doe | john.doe@example.com | 123-456-7890 | 123 Main St |
| 2 | Jane | Smith | jane.smith@example.c... | 987-654-3210 | 456 Elm St |

7. Insert data into the Book table by using the below query.


Query: INSERT INTO Book (ISBN, Title, Genre, PublicationYear, Price, PublisherID)
VALUES

('978-0-553-21311-0', 'Harry Potter and the Sorcerer's Stone', 'Fantasy', 1997, 19.99, 1),
('978-0-452-28423-4', '1984', 'Dystopian', 1949, 12.99, 2)

8. To check the Data, use the below query.


Query: Select * from Book;

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO Book (ISBN, Title, Genre, PublicationYear, Price, PublisherID)
2 VALUES
3 ('978-0-553-21311-0', 'Harry Potter and the Sorcerer's Stone', 'Fantasy', 1997, 19.99, 1),
4 ('978-0-452-28423-4', '1984', 'Dystopian', 1949, 12.99, 2)
5
6 Select * from Book;
```

Results Messages

 Search to filter items...

| ISBN | Title | Genre | PublicationYear | Price | PublisherID |
|-------------------|---------------------------|-----------|-----------------|-------|-------------|
| 978-0-452-28423-4 | 1984 | Dystopian | 1949 | 12.99 | 2 |
| 978-0-553-21311-0 | Harry Potter and the S... | Fantasy | 1997 | 19.99 | 1 |

9. Insert data into the Order table by using the below query.




Query: INSERT INTO [Order] (OrderID, OrderDate, TotalAmount, CustomerID)
VALUES

(1, '2023-08-01', 33.98, 1),
(2, '2023-08-15', 12.99, 2)

10. To check the Data, use the below query.


Query: Select * from [Order];

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO [Order] (OrderID, OrderDate, TotalAmount, CustomerID)
2 VALUES
3 (1, '2023-08-01', 33.98, 1),
4 (2, '2023-08-15', 12.99, 2)
5
6 Select * from [Order];
```

Results Messages

 Search to filter items...

| OrderID | OrderDate | TotalAmount | CustomerID |
|---------|-----------------------------|-------------|------------|
| 1 | 2023-08-01T00:00:00.0000000 | 33.98 | 1 |
| 2 | 2023-08-15T00:00:00.0000000 | 12.99 | 2 |

11. Insert data into the BookAuthor table by using the below query.




Query: INSERT INTO BookAuthor (ISBN, AuthorID)
VALUES

('978-0-553-21311-0', 1),
('978-0-452-28423-4', 2)

12. To check the Data, use the below query.


Query: Select * from BookAuthor;

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO BookAuthor (ISBN, AuthorID)
2 VALUES
3 ('978-0-553-21311-0', 1),
4 ('978-0-452-28423-4', 2)
5
6 Select * from BookAuthor;
```

Results Messages

 Search to filter items...

| ISBN | AuthorID |
|-------------------|----------|
| 978-0-452-28423-4 | 2 |
| 978-0-553-21311-0 | 1 |

13. Insert data into the OrderBook table by using the below query.




Query: INSERT INTO OrderBook (OrderID, ISBN)
VALUES

(1, '978-0-553-21311-0'),
(2, '978-0-452-28423-4')

14. To check the Data, use the below query.


Query: Select * from OrderBook;

Query 1 ✕

Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 INSERT INTO OrderBook (OrderID, ISBN)
2 VALUES
3 (1, '978-0-553-21311-0'),
4 (2, '978-0-452-28423-4')
5
6 Select * from OrderBook;
```

Results Messages

 Search to filter items...




| OrderID | ISBN |
|---------|-------------------|
| 1 | 978-0-553-21311-0 |
| 2 | 978-0-452-28423-4 |

Grant

1. Grant SELECT permission on the Author table to a user.

Query: GRANT SELECT ON Author TO dbo;

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as ▾

1 GRANT SELECT ON Author TO dbo;

Results Messages




Query succeeded: Affected rows: 0

Revoke

1. Revoke SELECT permission on the Author table from a user.

Query: REVOKE SELECT ON Author FROM dbo;

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data

1 REVOKE SELECT ON Author FROM dbo;

Results Messages

Query succeeded: Affected rows: 0

Update





1. The UPDATE operation is used to modify existing records in a table.
2. Update the price of a book with ISBN '978-0-553-21311-0'

Query: UPDATE Book SET Price = 24.99 WHERE ISBN = '978-0-553-21311-0';

3. To check the data use the query below.


Query: Select * from Book;

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 UPDATE Book SET Price = 24.99 WHERE ISBN = '978-0-553-21311-0';
2
3 Select * from Book;
```

Results Messages

 Search to filter items...




| ISBN | Title | Genre | PublicationYear | Price | PublisherID |
|-------------------|---------------------------|-----------|-----------------|-------|-------------|
| 978-0-452-28423-4 | 1984 | Dystopian | 1949 | 12.99 | 2 |
| 978-0-553-21311-0 | Harry Potter and the S... | Fantasy | 1997 | 24.99 | 1 |

Truncate

1. The TRUNCATE operation is used to remove all records from a table while keeping the table structure intact.

Query: TRUNCATE TABLE OrderBook;

Query 1 ✕

 Run ☐ Cancel query  Save query  Export

```
1 TRUNCATE TABLE OrderBook;
```

Results Messages

Query succeeded: Affected rows: 0

Filtering Data

1. Filtering data involves selecting specific rows based on certain conditions.
2. Retrieve books with a price greater than \$15

Query: `SELECT * FROM Book WHERE Price > 15.00;`

Query 1 ✕

[Run](#) ☐ Cancel query [Save query](#) [Export data as](#) ▼ [Show only Editor](#)

```
1 SELECT * FROM Book WHERE Price > 15.00;
```

Results Messages

| ISBN | Title | Genre | PublicationYear | Price | PublisherID |
|-------------------|---------------------------|---------|-----------------|-------|-------------|
| 978-0-553-21311-0 | Harry Potter and the S... | Fantasy | 1997 | 24.99 | 1 |

3. Retrieve customers from a specific city.

Query: `SELECT * FROM Customer WHERE Address LIKE '%Main St%';`

Query 1 ✕

[Run](#) ☐ Cancel query [Save query](#) [Export data as](#) ▼ [Show only Editor](#)

```
1 SELECT * FROM Customer WHERE Address LIKE '%Main St%';
```

Results Messages

| CustomerID | FirstName | LastName | Email | PhoneNumber | Address |
|------------|-----------|----------|----------------------|--------------|-------------|
| 1 | John | Doe | john.doe@example.com | 123-456-7890 | 123 Main St |

Grouping Data using GROUP BY

1. Grouping data allows you to aggregate data based on specific columns.
2. Get the total sales amount for each customer.

Query: `SELECT CustomerID, SUM(TotalAmount) AS TotalSales
FROM [Order]
GROUP BY CustomerID;`

Query 1 ✕

[Run](#) ☐ Cancel query [Save query](#) [Export data as](#) ▼ [Show only Editor](#)

```
1 SELECT CustomerID, SUM(TotalAmount) AS TotalSales  
2 FROM [Order]  
3 GROUP BY CustomerID;
```

Results Messages





| CustomerID | TotalSales |
|------------|------------|
| 1 | 33.98 |
| 2 | 12.99 |

Filtering Aggregated Data using HAVING

1. The HAVING clause is used with GROUP BY to filter aggregated data.
2. Get customers who have made total purchases greater than \$30.

Query: SELECT CustomerID, SUM(TotalAmount) AS TotalPurchases
FROM [Order]
GROUP BY CustomerID
HAVING SUM(TotalAmount) > 30.00;

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 SELECT CustomerID, SUM(TotalAmount) AS TotalPurchases
2 FROM [Order]
3 GROUP BY CustomerID
4 HAVING SUM(TotalAmount) > 30.00;
```

Results Messages

 Search to filter items...




| CustomerID | TotalPurchases |
|------------|----------------|
| 1 | 33.98 |

Sorting Data

1. Sorting data allows you to order the result set based on specific columns.
2. Retrieve books sorted by title in ascending order

Query: SELECT * FROM Book ORDER BY Title;

Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as  Show only Editor

```
1 SELECT * FROM Book ORDER BY Title;
```

Results Messages

 Search to filter items...

| ISBN | Title | Genre | PublicationYear | Price | PublisherID |
|-------------------|---------------------------|-----------|-----------------|-------|-------------|
| 978-0-452-28423-4 | 1984 | Dystopian | 1949 | 12.99 | 2 |
| 978-0-553-21311-0 | Harry Potter and the S... | Fantasy | 1997 | 24.99 | 1 |

3. Retrieve customers sorted by last name in descending order.

Query: `SELECT * FROM Customer ORDER BY LastName DESC;`

Query 1

Run

Cancel query

Save query

Export data as

Show only Editor

1

`SELECT * FROM Customer ORDER BY LastName DESC;`

Results

Messages

Search to filter items...

| CustomerID | FirstName | LastName | Email | PhoneNumber | Address |
|------------|-----------|----------|-------------------------|--------------|-------------|
| 2 | Jane | Smith | jane.smith@example.c... | 987-654-3210 | 456 Elm St |
| 1 | John | Doe | john.doe@example.com | 123-456-7890 | 123 Main St |