Madan Bhandari Memorial College

Lab 1: Introduction and installation of SQL Server for Database Management

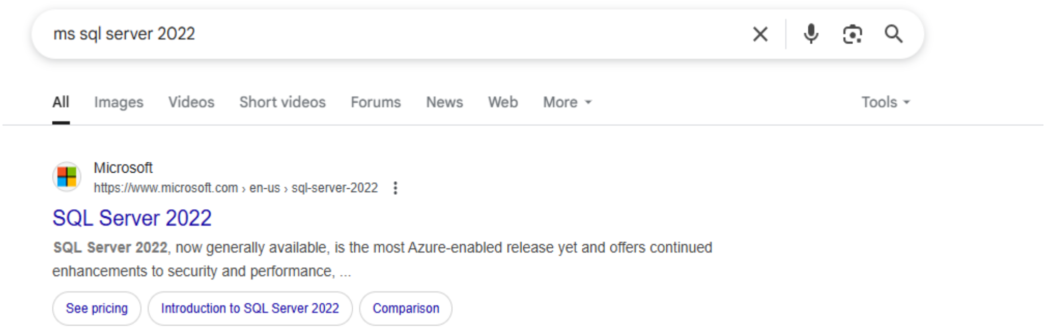
Introduction:

Microsoft SQL Server is a relational database management system (RDBMS) that supports a wide variety of transaction processing, business intelligence (BI) and data analytics applications in corporate IT environments. Like other RDBMS software, Microsoft SQL Server is built on top of Structured Query Language (SQL), a standardized programming language that database administrators (DBAs) and other IT professionals use to manage databases and query the data they contain. SQL Server is tied to Transact-SQL (T-SQL), Microsoft's proprietary query language that enables applications and tools to communicate and also connect to a SQL Server instance or database.

Installation:

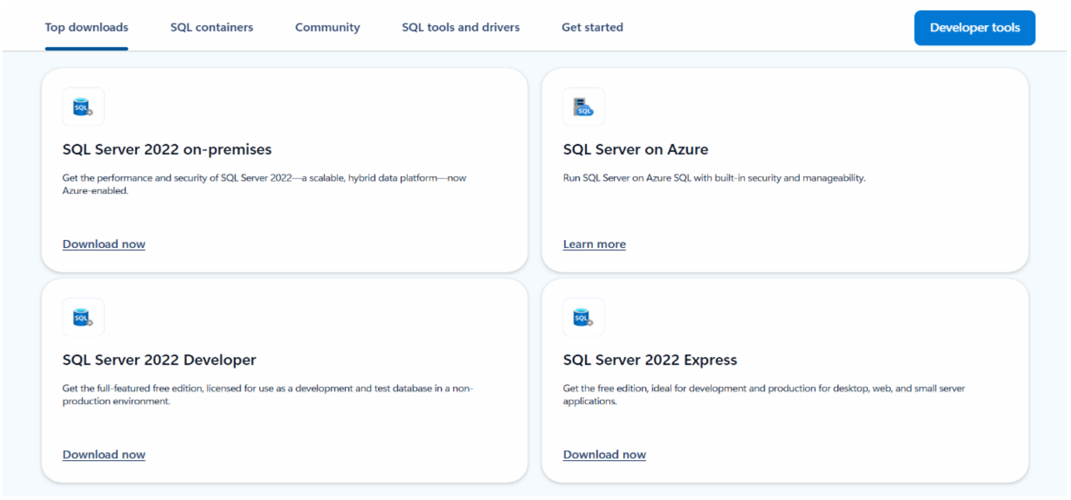
In this Installation we are going to download MS SQL Server 2022 Developer edition.

Step1: Firstly, let's go to the link microsoft.com/en-us/sql-server/sql-server-downloads or simply search ms sql server 2022 in google and download the developer edition from the options provided.



SQL Server 2022, now generally available, is the most Azure-enabled release yet and offers continued enhancements to security and performance, ...

Figure 1: Search result for "MS SQL Server 2022" showing the official Microsoft download link.



Get the performance and security of SQL Server 2022—a scalable, hybrid data platform—now Azure-enabled.

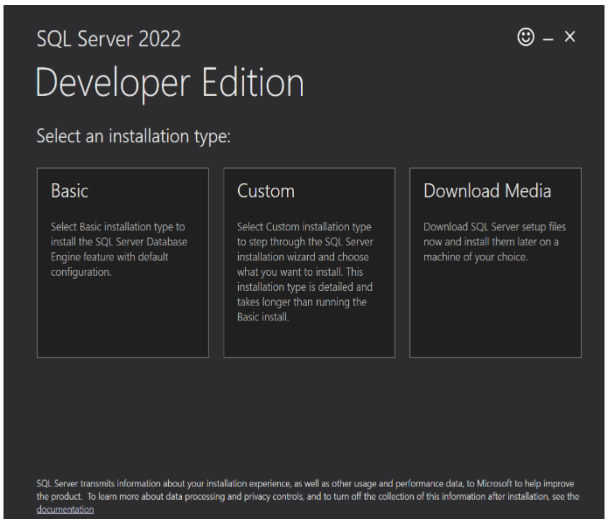
Run SQL Server on Azure SQL with built-in security and manageability.

Get the full-featured free edition, licensed for use as a development and test database in a non-production environment.

Get the free edition, ideal for development and production for desktop, web, and small server applications.

Figure 2:The various SQL Server 2022 editions available for download

Step 2: A SQL2022-SSEI-Dev.exe file is now downloaded. After this run the file and installation process will start. For this installation we are going we with basic option from the options provided to us.


SQL Server 2022

Developer Edition

Select an installation type:

Basic

Select Basic installation type to install the SQL Server Database Engine feature with default configuration.

Custom

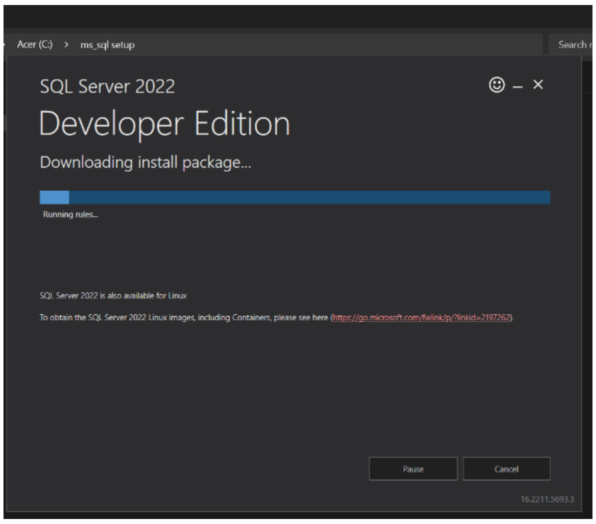
Select Custom installation type to step through the SQL Server installation wizard and choose what you want to install. This installation type is detailed and takes longer than running the Basic install.

Download Media

Download SQL Server setup files now and install them later on a machine of your choice.

SQL Server transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the documentation

Figure 3: The figure displays the installation screen for SQL Server 2022 Developer Edition, with three options: Basic, Custom, and Download Media.


Acer (C:) > ms_sql setup

SQL Server 2022

Developer Edition

Downloading install package...

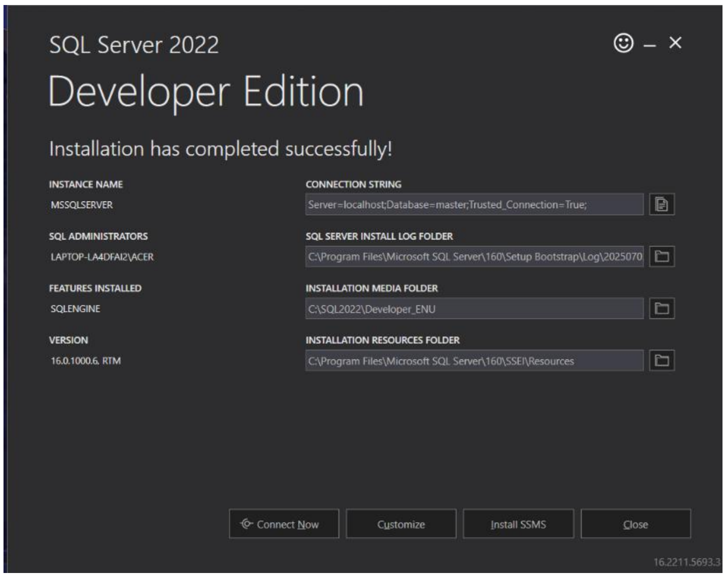
Running rules...

SQL Server 2022 is also available for Linux

To obtain the SQL Server 2022 Linux images, including Containers, please see here (https://go.microsoft.com/fwlink/p/?linkid=2197262).

Figure 4: Installing the basic option

Step 3: After completion of this process, a download and install summary page is shown which also has a important link for us so, do not cross out this window. On this page you have click Install SSMS which is an important file required for running SQL Server. If you have already downloaded SSMS than you can skip this step and continue later.


SQL Server 2022

Developer Edition

Installation has completed successfully!

INSTANCE NAME
MSSQLSERVER

SQL ADMINISTRATORS
LAPTOP-LA4DFAI2\ACER

FEATURES INSTALLED
SQLENGINE

VERSION
16.0.1000.6, RTM

CONNECTION STRING
Server=localhost;Database=master;Trusted_Connection=True;

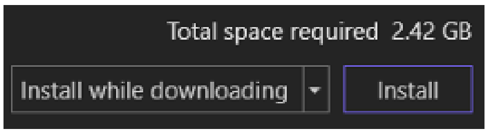
SQL SERVER INSTALL LOG FOLDER
C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap\Log\2025070

INSTALLATION MEDIA FOLDER
C:\SQL2022\Developer_ENU

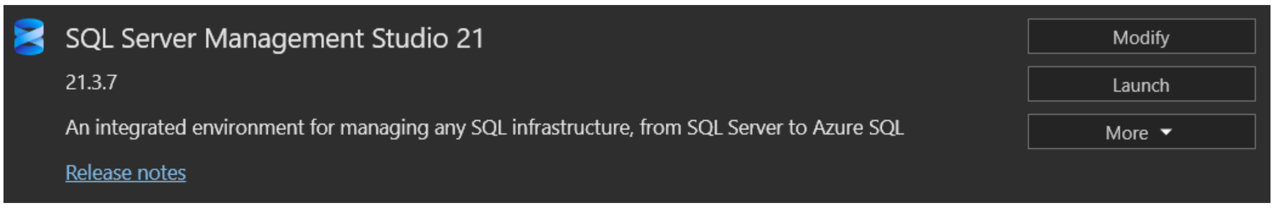
INSTALLATION RESOURCES FOLDER
C:\Program Files\Microsoft SQL Server\160\SSEI\Resources

Figure 5: Installation successfully completed

Step 4: Click the Install SSMS button and follow the steps there to completely install SSMS for your device. After that a bootstrapper has been downloaded click it and run the processes as the instructions instructs and installer should start installing after execution of the file click the install button on the bottom left of your screen.


Total space required 2.42 GB

Step 5: The installation process should start. After completion, now click launch to start the program.


SQL Server Management Studio 21

21.3.7

An integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL

Release notes

Figure 6: Launch the SSMS

Step 6: After clicking on launch a sign in window should open. Create a account or sign in to a existing account or you can skip for now to sign in later.

Conclusion:

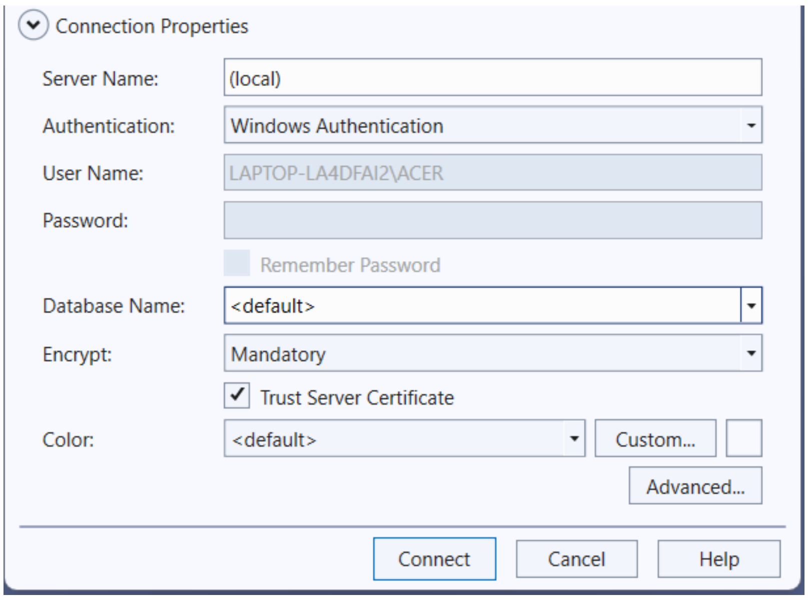
In this lab, we successfully installed MySQL and ensured that it runs properly on our system. This setup provides the foundation for practicing SQL commands and developing database-driven applications. By completing this lab, we are now ready to explore the core functionalities of MySQL and perform operations such as creating tables, inserting data, and running queries.

LAB 2: Creating DataBase and tables using query

In this lab, we learn how to create a database and define tables using SQL queries. Understanding the concepts about how to structure a database is essential for organizing data efficiently. With CREATE DATABASE and CREATE TABLE commands, we can design a relational database that suits the needs of any application. This lab focuses on writing basic SQL commands to set up a database and its tables from scratch.

Creating a database and connecting to local server

Step 1: After sign in our installation process is completed and now a connect window will pop up in this window. Here write the server name (local) and then click Trust Server Certificate and hit connect.


Connection Properties

Server Name: (local)

Authentication: Windows Authentication

User Name: LAPTOP-LA4DFAI2\ACER

Password:

Remember Password

Database Name: <default>

Encrypt: Mandatory

✓ Trust Server Certificate

Color: <default>

Figure 7: Database connect

Step 2: Create a new database in local computer. Right click on new database and create a new database in your local compuer.

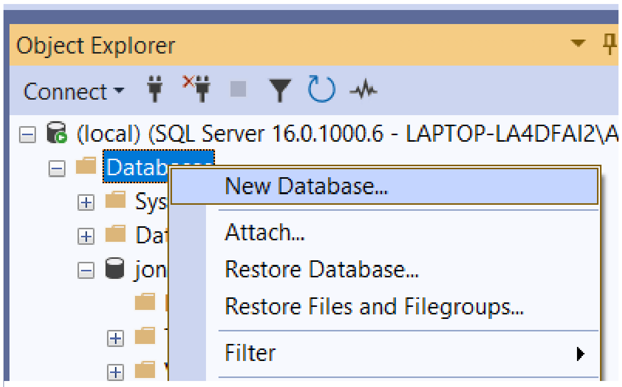
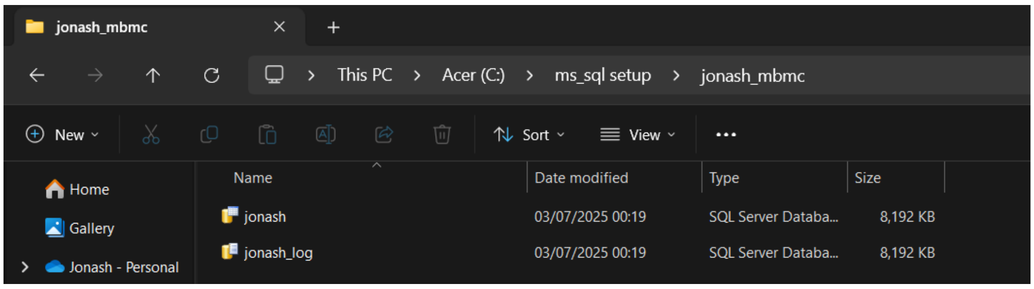


Figure 7: 1 Create a new database



Name	Date modified	Type	Size
jonash	03/07/2025 00:19	SQL Server Databa...	8,192 KB
jonash_log	03/07/2025 00:19	SQL Server Databa...	8,192 KB

Figure 7: 2 Database created in computer

Creating a table using query

Step 1: Go to your database and right click and select new query.

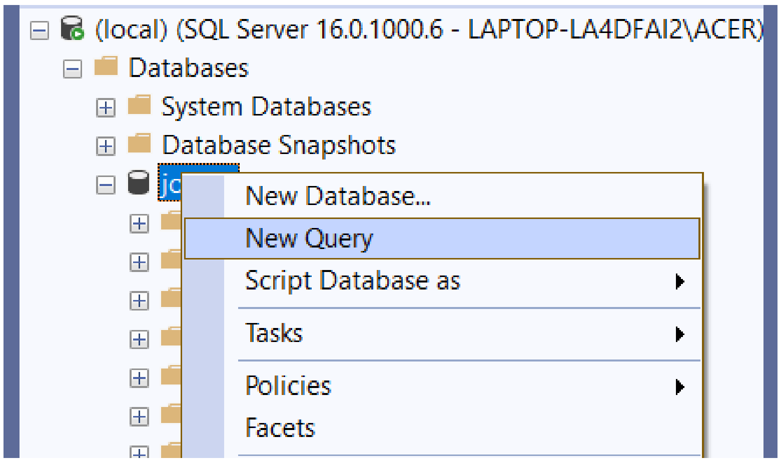


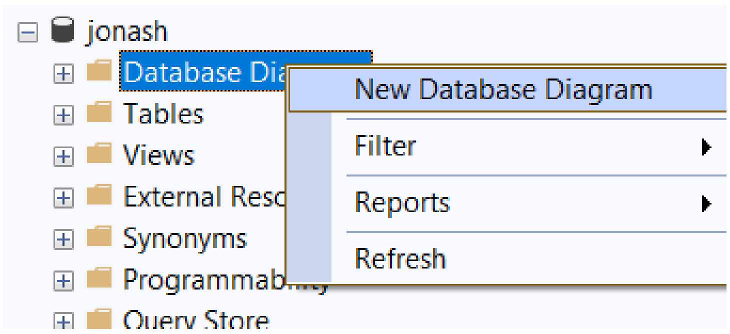
Figure 7: 3 Creating a new query

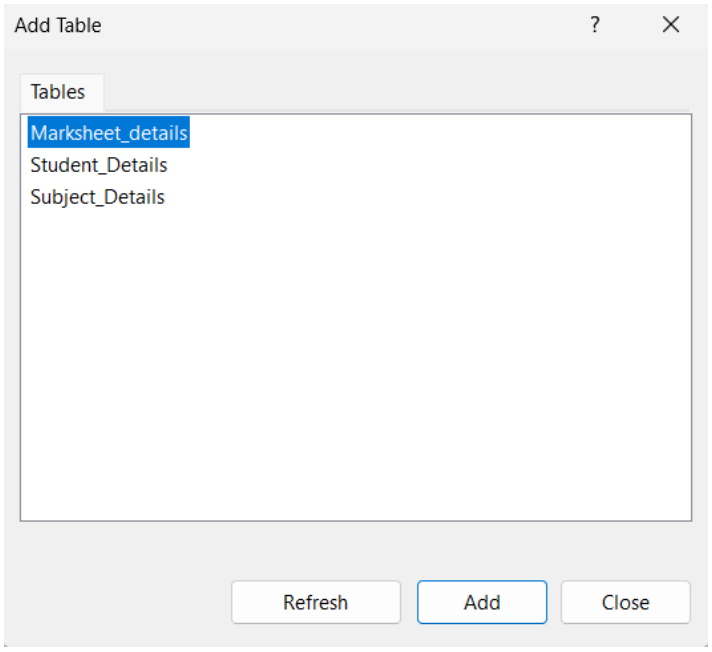
Step 2: Write the following query and execute the code.

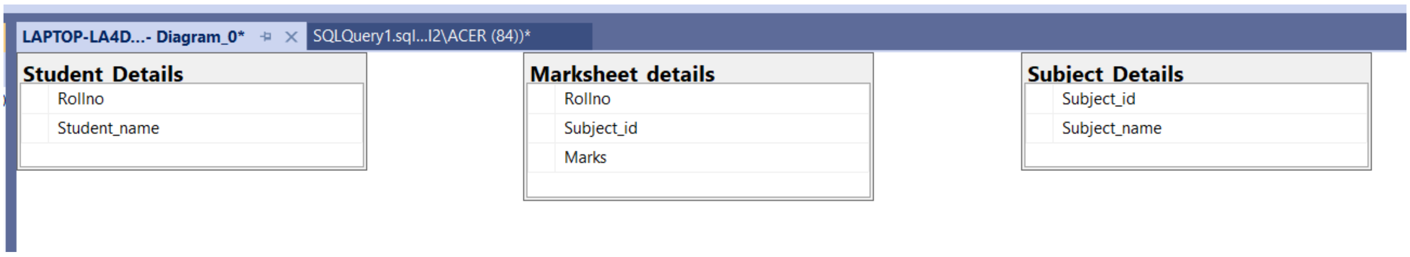
```
create table student_details
(
	roll_no int,
	student_name varchar(100)
)

create table subject_details
(
	student_id int,
	subject_name varchar(100)
)
```

```
create table marksheet_details
(
	roll_no int,
	subject_id int,
	marks float
)
```

Step 3: Right click and click new database diagram and add the necessary tables.







Student Details
- Rollno
- Student_name

Marksheet details
- Rollno
- Subject_id
- Marks

Subject Details
- Subject_id
- Subject_name

Conclusion:

In this lab, we successfully created a database and multiple tables using SQL queries. This practical experience helped us learning to create databases and tables using query.

Lab 3: Concept of keys

In SQL, the different types of keys are the set of attributes used to identify a specific row from a table and to find or create the relation between two or more tables.

Types of keys:

Key type	Purpose
Primary Key:	Used to uniquely identify a row in a table
Foreign Key:	Used to maintain referential integrity between tables
Composite Key:	Used to uniquely identify a row when a single column is not sufficient
Unique Key:	Used to prevent duplicate values in a column

Database Language in SQL

SQL (Structured Query Language) includes different types of commands grouped as database languages. These are used to create, manage, and manipulate data in a database.

Types of Database Languages:

1. DDL (Data Definition Language) – Defines the structure of the database.
   Commands: CREATE, ALTER, DROP, TRUNCATE
2. DML (Data Manipulation Language) – Deals with data manipulation.
   Commands: SELECT, INSERT, UPDATE, DELETE
3. DCL (Data Control Language) – Controls access to data.
   Commands: GRANT, REVOKE
4. TCL (Transaction Control Language) – Manages transactions in the database.
   Commands: COMMIT, ROLLBACK, SAVEPOINT

Some Functions in DDL

CREATE(): Used to create a new table or database.

ALTER(): Used to change the structure of an existing table (e.g., add or remove columns).

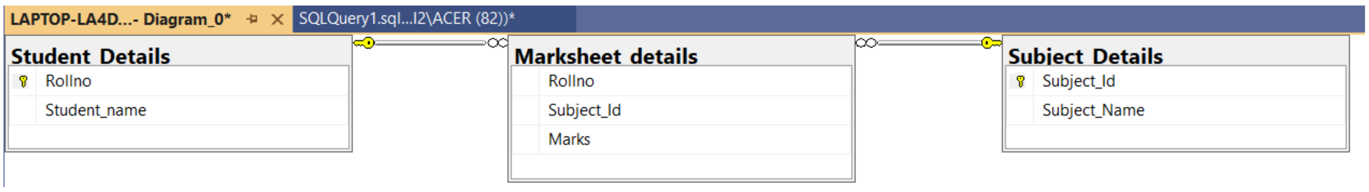
DROP(): Used to delete a table or database completely.

Steps for showing keys and DDL language in a relational database

Step1: Create a new query and write the following code which includes primary key and composite primary key and DDL functions then executing it.

```sql
-- Dropping table if it exists

DROP TABLE IF EXISTS Student_Details;
DROP TABLE IF EXISTS Subject_Details;
DROP TABLE IF EXISTS Marksheet_details;
GO

--CreatingStudentDetailstable

CREATE TABLE Student_Details

(

    Rollno INT NOT NULL,
    Student_name VARCHAR(100),

    PRIMARY KEY (RollNo)

);

-- Creating SubjectDetails table

CREATE TABLE Subject_Details

(

    Subject_Id INT NOT NULL,

    Subject_Name VARCHAR(100),

    PRIMARY KEY (SubjectId)

);

-- Creating MarksheetDetails table

CREATE TABLE Marksheet_details

(

    Rollno INT NOT NULL,

    Subject_Id INT NOT NULL,

    Marks INT,

FOREIGN KEY (Rollno) REFERENCES Student_Details(Rollno),
FOREIGN KEY (Subject_Id) REFERENCES Subject_Details(Subject_Id)

);
```

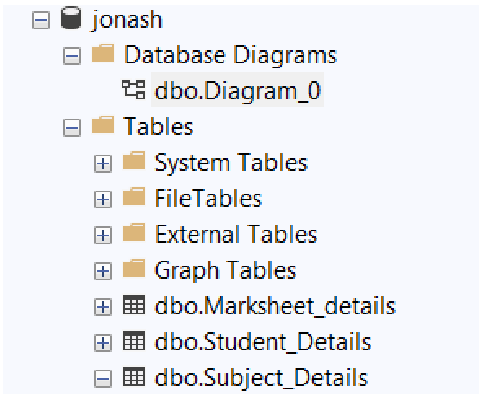
Step2: The diagrammatic view of relational tables including the database tables is given below:



Student_Details
Rollno
Student_name

Marksheet_details
Rollno
Subject_Id
Marks

Subject_Details
Subject_Id
Subject_Name



```
jonash
  Database Diagrams
    dbo.Diagram_0
  Tables
    System Tables
    FileTables
    External Tables
    Graph Tables
    dbo.Marksheet_details
    dbo.Student_Details
    dbo.Subject_Details
```

Conclusion

In this lab, we learned about various DDL commands to create, alter, and drop database objects, and we learned how keys are used to uniquely identify and relate records in tables.

Lab 4: Insertion of Data into Relational Tables Using Queries with Foreign Keys

Intoduction:

In this lab, we focus on DML (Data Manipulation Language) commands, which are used to manage the data within relational database tables. These commands do not modify the table structure but allow users to:

- SELECT – Retrieve data from tables.
- INSERT – Add new records into a table.
- UPDATE – Modify existing records.
- DELETE – Remove records from a table.

The lab demonstrates the use of the INSERT and SELECT commands to insert and retrieve student-related data from multiple interrelated tables, while ensuring referential integrity using foreign keys.

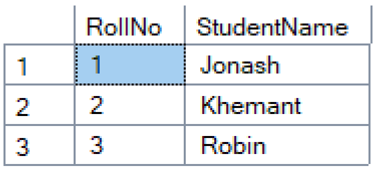
Creating Tables with Foreign Keys

Three relational tables are created using SQL Server Management Studio (SSMS):

1. Studen_tDetails – Stores student information.
2. Subject_Details – Stores subject information.
3. Marksheet_details – Stores marks obtained by students, linked to the other two tables using foreign keys.

This design enforces relational integrity and ensures that entries in Marksheet_details cannot exist unless corresponding records exist in Student_Details and Subject_Details.

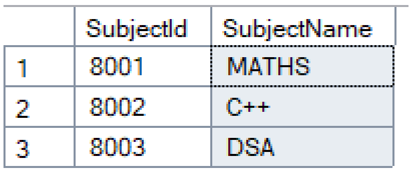
```sql
--SQL Source Code:
-- Dropping tables if they exist
DROP TABLE IF EXISTS Student_Details;
DROP TABLE IF EXISTS Subject_Details;
DROP TABLE IF EXISTS Marksheet_details;
GO

-- Creating StudentDetails table
CREATE TABLE StudentDetails (
    RollNo INT NOT NULL,
    StudentName VARCHAR(100),
    PRIMARY KEY (RollNo)
);

-- Creating SubjectDetails table
CREATE TABLE SubjectDetails (
    SubjectId INT NOT NULL,
    SubjectName VARCHAR(100),
    PRIMARY KEY (SubjectId)
);

-- Creating MarksheetDetails table
CREATE TABLE MarksheetDetails (
    RollNo INT NOT NULL,
    SubjectId INT NOT NULL,
    Marks FLOAT,
    FOREIGN KEY (RollNo) REFERENCES StudentDetails(RollNo),
    FOREIGN KEY (SubjectId) REFERENCES SubjectDetails(SubjectId)
);

-- Query to Insert Data into Tables:
-- Inserting students
INSERT INTO StudentDetails (RollNo, StudentName) VALUES
(1, 'Jonash'),
(2, 'Khemant'),
(3, 'Robin');
```



	RollNo	StudentName
1	1	Jonash
2	2	Khemant
3	3	Robin

Figure 7: 4 Displaying student details

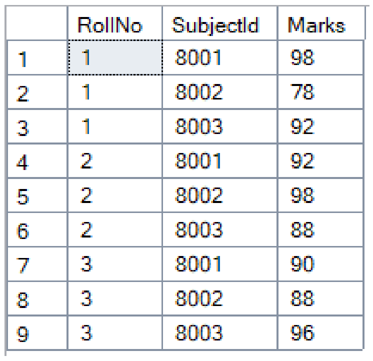
```sql
-- Inserting subjects
INSERT INTO SubjectDetails (SubjectId, SubjectName) VALUES
(8001, 'MATHS'),
(8002, 'C++'),
(8003, 'DSA');
```



	SubjectId	SubjectName
1	8001	MATHS
2	8002	C++
3	8003	DSA

Figure 7: 5 Displaying output of subject details

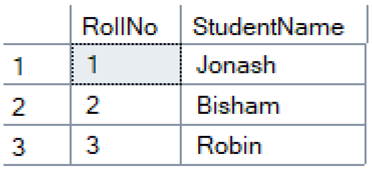
```sql
-- Inserting marks
INSERT INTO MarksheetDetails (RollNo, SubjectId, Marks) VALUES
(1, 8001, 98), (1, 8002, 78), (1, 8003, 92), (2, 8001, 92), (2, 8002, 98), (2, 8003, 88),
(3, 8001, 90), (3, 8002, 88), (3, 8003, 96);
```



	RollNo	SubjectId	Marks
1	1	8001	98
2	1	8002	78
3	1	8003	92
4	2	8001	92
5	2	8002	98
6	2	8003	88
7	3	8001	90
8	3	8002	88
9	3	8003	96

Figure 7: 6 Displaying the marksheet details

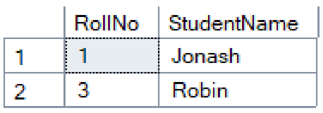
```sql
--Query to update data in Table
Update StudentDetails
Set StudentName = 'Bisham'
where RollNo = 2
```



	RollNo	StudentName
1	1	Jonash
2	2	Bisham
3	3	Robin

Figure 7: 7 Update student details

```sql
--Query to delete data from Table
Delete from MarksheetDetails where RollNo = 2
Delete From StudentDetails
where RollNo = 2
```



	RollNo	StudentName
1	1	Jonash
2	3	Robin

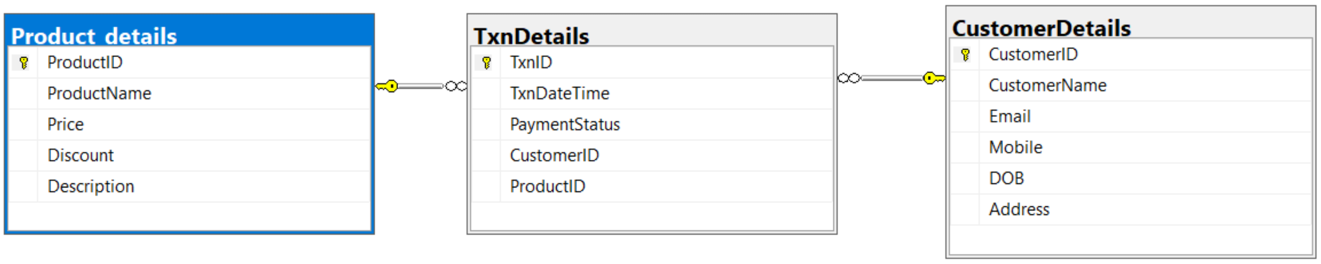
Figure 7: 8 Delete a row

Lab 5:

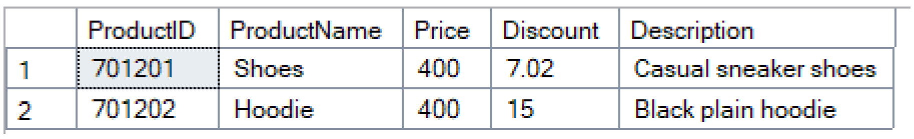
Title: Create a Table for Customer Details, Product Details and Transaction Details each having a Primary key and also create foreign keys Referencing Customer details and Product Details in Transaction Details. Also insert at least 2 data sets in each table.

--SQL Source Code:

```
--Creating tables
Create table Product_details
(
ProductID int NOT NULL,
ProductName varchar(255),
Price float,
Discount float,
Description varchar(500),
PRIMARY KEY(ProductID)
);

Create table CustomerDetails
(
CustomerID int NOT NULL,
CustomerName varchar(100),
Email varchar(255),
Mobile Bigint,
DOB date,
Address Varchar(255)
PRIMARY KEY(CustomerID)
);

Create table TxnDetails
(
TxnID int NOT NULL,
TxnDateTime datetime,
PaymentStatus char,
PRIMARY KEY(TxnID),
CustomerID int,
Foreign Key (CustomerID) References CustomerDetails(CustomerID),
ProductID int,
Foreign Key (ProductID) References Product_Details(ProductID),
);
```

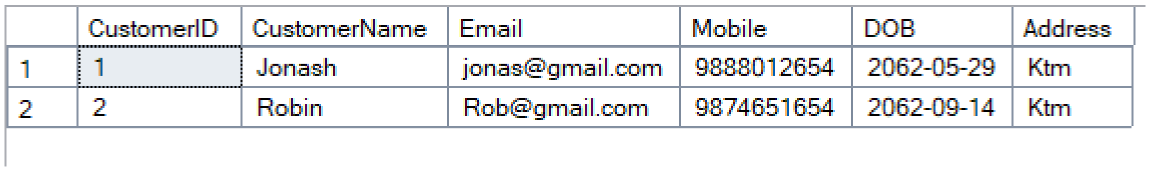


```
--Query to Inset data into the tables
Insert into Product_details
(ProductID,ProductName,Price,Discount,Description)
values
(701201,'Shoes',400,7.02,'Casual sneaker shoes')
Insert into Product_details
(ProductID,ProductName,Price,Discount,Description)
values
(701202,'Hoodie',400,15,'Black plain hoodie')
```



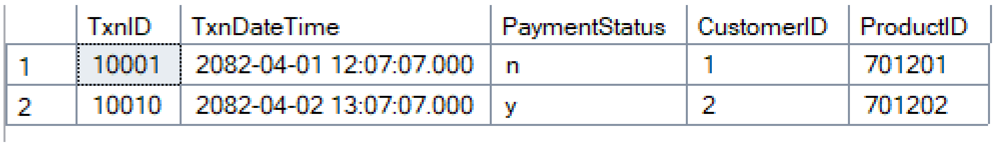
	ProductID	ProductName	Price	Discount	Description
1	701201	Shoes	400	7.02	Casual sneaker shoes
2	701202	Hoodie	400	15	Black plain hoodie

```sql
Insert into CustomerDetails
(CustomerID,CustomerName,Email,Mobile,DOB,Address)
values
(1,'Jonash','jonas@gmail.com',9888012654,'2062/05/29','Ktm')
Insert into CustomerDetails
(CustomerID,CustomerName,Email,Mobile,DOB,Address)
values
(2,'Robin','Rob@gmail.com',9874651654,'2062/09/14','Ktm')
```



	CustomerID	CustomerName	Email	Mobile	DOB	Address
1	1	Jonash	jonas@gmail.com	9888012654	2062-05-29	Ktm
2	2	Robin	Rob@gmail.com	9874651654	2062-09-14	Ktm

```sql
Insert into TxnDetails
(TxnID,TxnDateTime,CustomerID,ProductID,PaymentStatus)
values
(10001,'2082/04/01 12:07:07',1,701201,'n')
Insert into TxnDetails
(TxnID,TxnDateTime,CustomerID,ProductID,PaymentStatus)
values
(10010,'2082/04/02 13:07:07',2,701202,'y')
```



	TxnID	TxnDateTime	PaymentStatus	CustomerID	ProductID
1	10001	2082-04-01 12:07:07.000	n	1	701201
2	10010	2082-04-02 13:07:07.000	y	2	701202

Conclusion:

In this lab we create table for Customer Details, Product Details and Transaction Details each having a Primary key and also create foreign keys Referencing Customer details and Product Details in Transaction Details. Also inserted 2 data sets in each table.