FAITHINFOTECH
Discover Potential. Deliver Performance

Introduction to Database

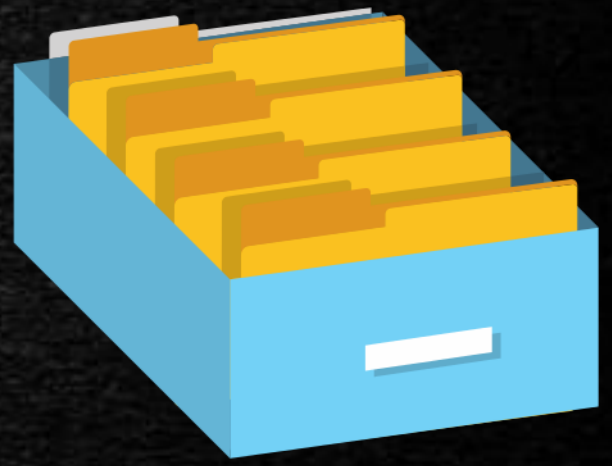
Introduction to DB, Why it's needed

Here is A Problem…

- A store deals with the products of many companies.
- Whenever there is a shortage of material the store owner places the order for the material to the concerned vendor.
- How would he manage the data for his store?

Solution 1 : Using Paper Files

- Paper files
  - A register/ file for one main item
  - Product register, Vendor List, Billing, Accounts



- Drawback:
  - Difficult to maintain and search
  - Repetition of data becomes necessary to make work easy

Solution 2 : Using Custom Software

- Data Management Software

  - Advantage
    - Easy to search
    - Easy to manipulate data
    - Easy to filter data according to a criteria
  - Drawback:
    - Application bound to the data storage
    - Modification is difficult

The Ultimate Solution : Use a Database

•Why Database

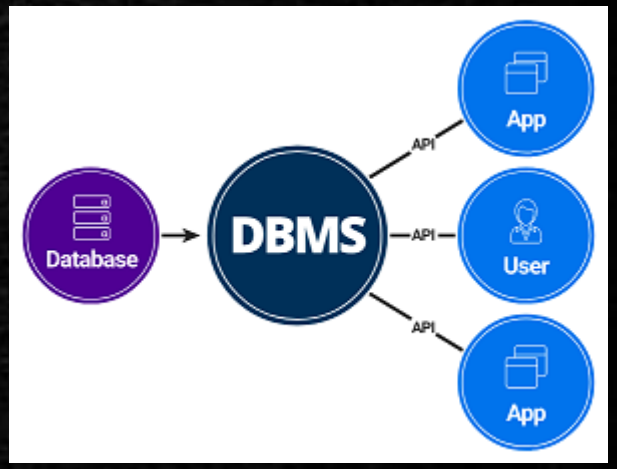
- Data represented in the form of logical tables.
- Modification of the structure of data is easy
- The programmer does not have to worry about the ‘How’ part
- Simple language to communicate with the database (SQL)
- Searching for data and sorting are easy

What is database and DBMS?

- Database is an organized collection of interrelated data

- The data is stored together without harmful or unnecessary redundancy

- A Database Management System (DBMS) is software designed to store, retrieve, define, and manage data in a database.



Characteristics of a good database:

- **Performance**:
  - Facility for the retrieval and manipulation of data irrespective of the number of tables with minimum time

- **Minimal redundancy**:
  - The database system should support minimal redundancy of data.

- **MultiUser**:
  - The db should provide multiuser support.

Characteristics of a good database:

- **Integrity**:
  - When multiple users uses the db the data items and the associations b/w the data should not be destroyed.

- **Privacy and security**:
  - The data should be protected against accidental or intentional access by unauthorized persons.

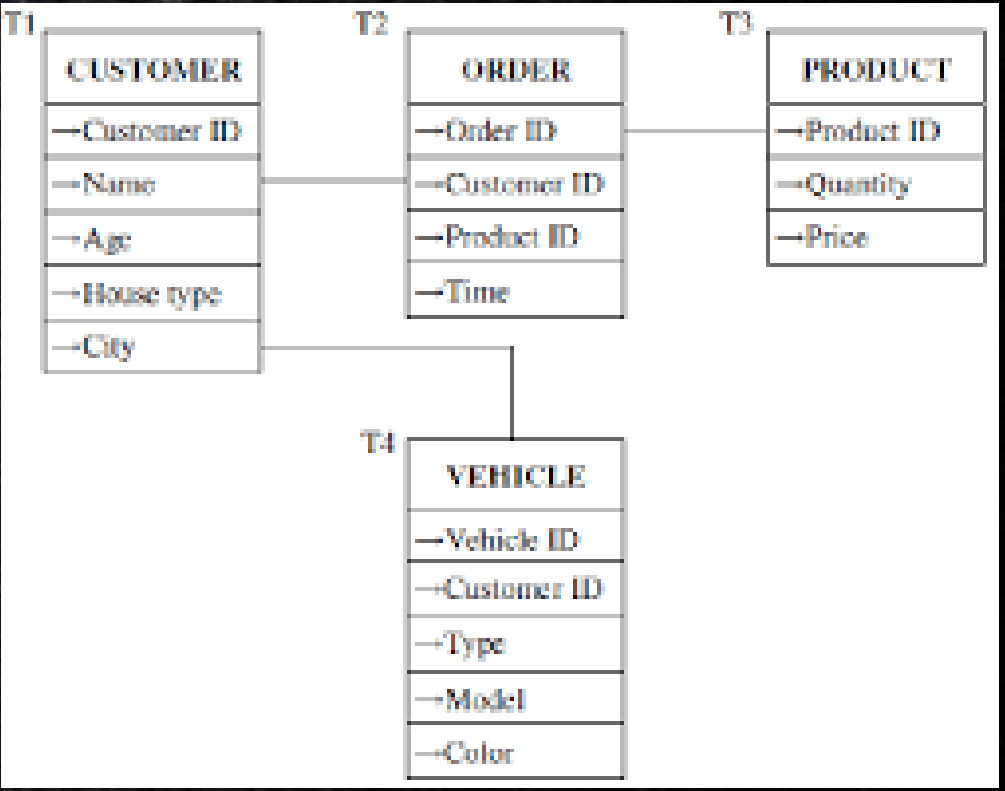
- **DB Language**:
  - The Db language used should be easy and powerful .
  - The most popular Db language is SQL

SQL

Introduction to RDBMS Concepts

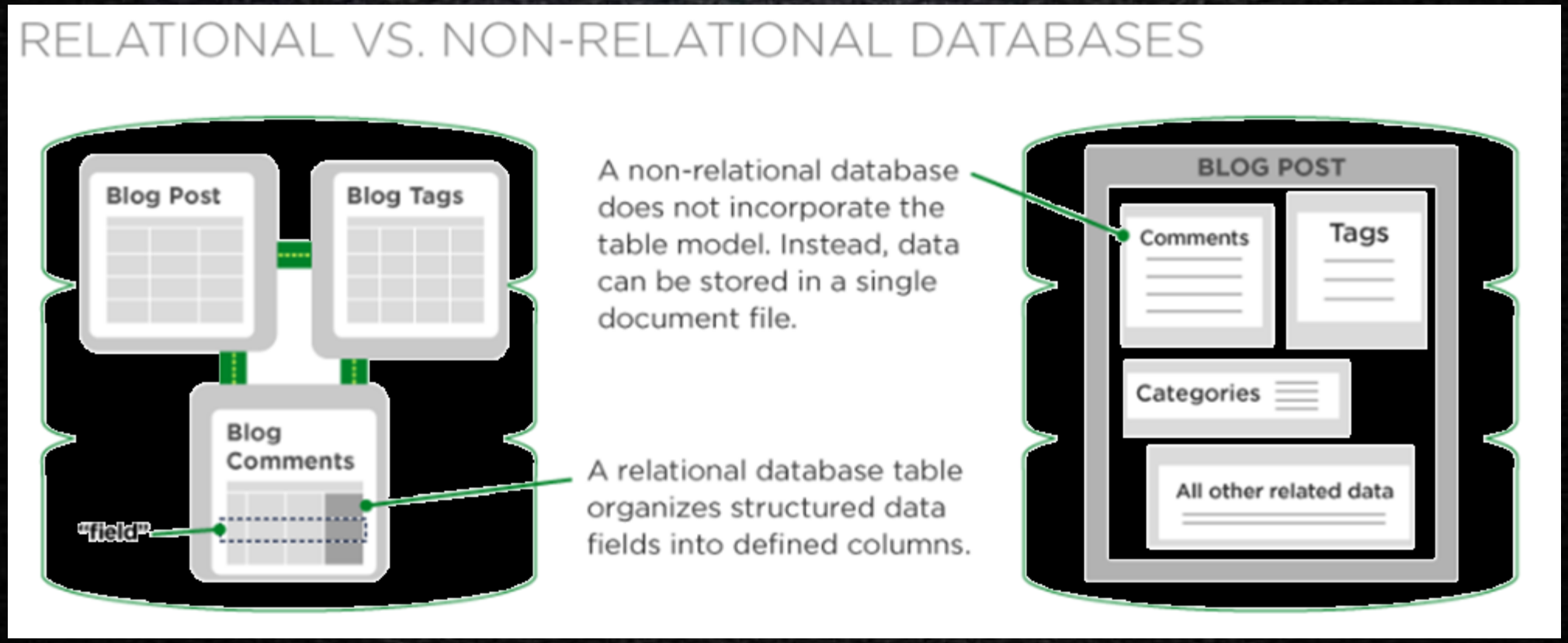
What is a Relational Database

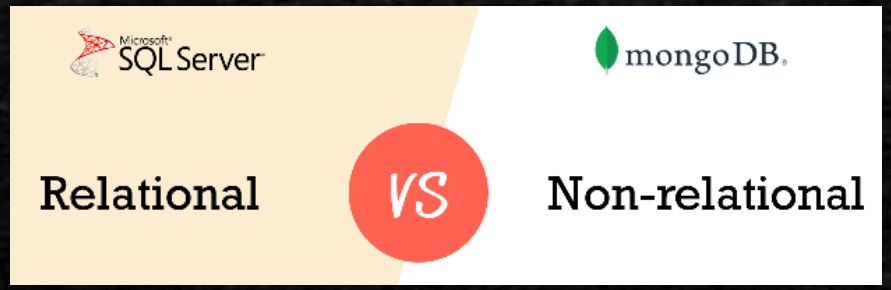
- A relational database is a collection of data items with pre-defined relationships between them.
- These items are organized as a set of tables with columns and rows. ...
- Each row in a table could be marked with a unique identifier called a primary key, and rows among multiple tables can be made related using foreign keys.



Eg: All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL and Microsoft Access are based on RDBMS.

Relational Database (SQL) vs Non-Relational (NoSQL)





SQL Model to represent Employee Data in Table

COLUMN

ROW

EmpID	EmpName	DepartmentID	Salary
ED101	Kannan	101	8000
ED102	Daasan	100	9000
ED103	Sathru	102	10000

FK

PK

Relational Database vs Non-Relational

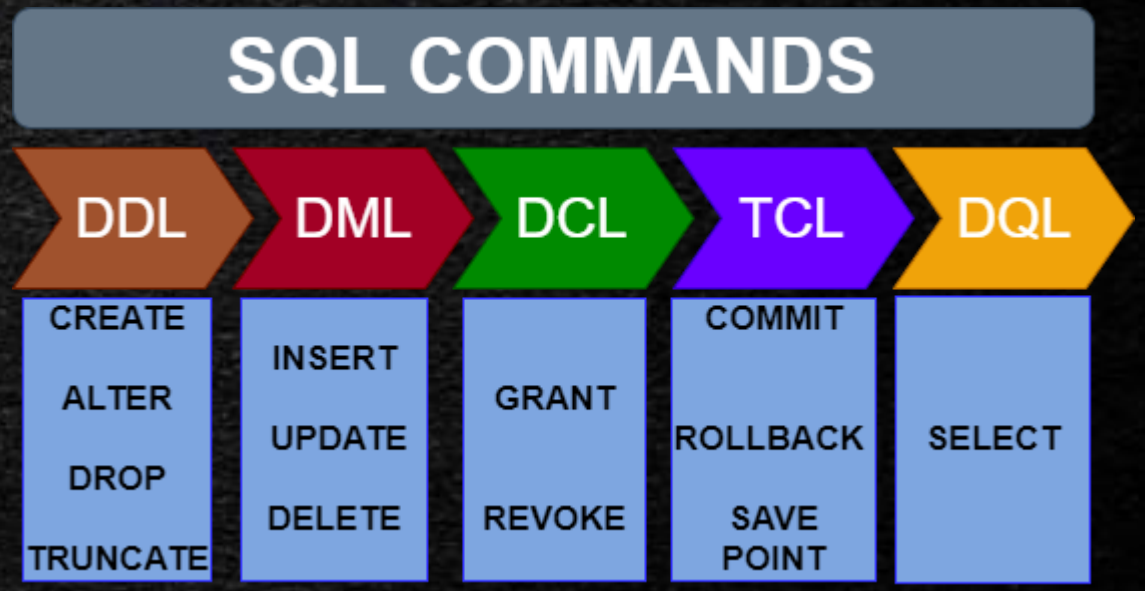
- This is how data is represented in a non-relational DB

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

SQL - Structured Query Language

- **SQL** is the common language to communicate with Database

- Parts of SQL
  - DDL-Data Definition Language
  - DML-Data Manipulation Language
  - TCL-Transaction Control Language
  - DCL-Data Control Language
  - DQL-Data Query Language

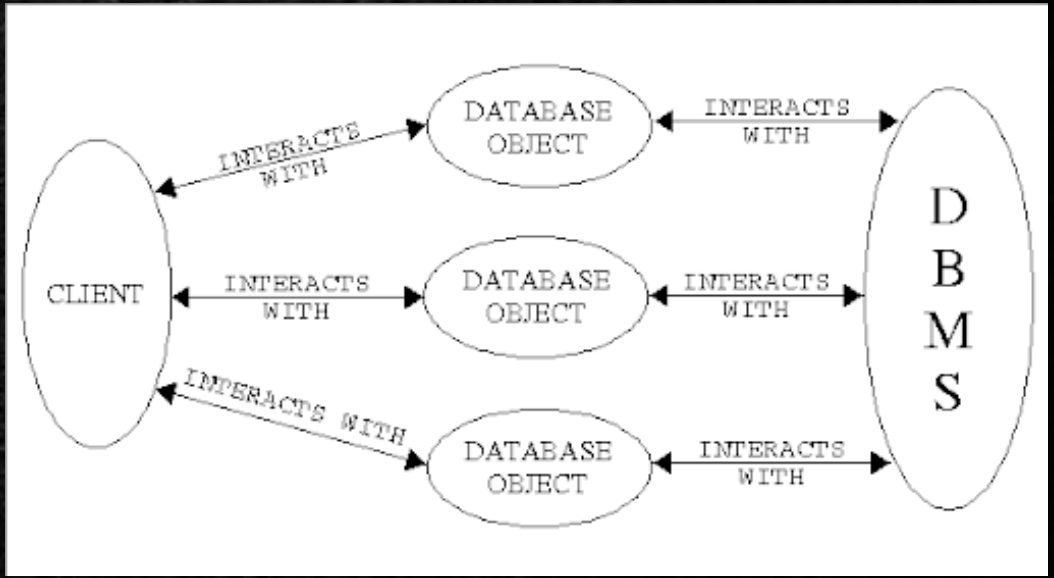


SQL Database Objects

- Table – Basic Unit of Storage
- View – Logical Subset of data from one or more tables
- Index – Improves the performance of some queries
- Synonym – Gives alternative names to objects

Object Naming Rules

- Must begin with a letter
- Must be 1 –30 chars long
- Must contain only A-Z, a-z, 0-9, $ and #
- Must not duplicate the name of another object owned by the same user
- Must not be database reserved word
  - A table can have upto 1000 columns



SQL Server

Basics of MS SQL Server

Microsoft SQL Server

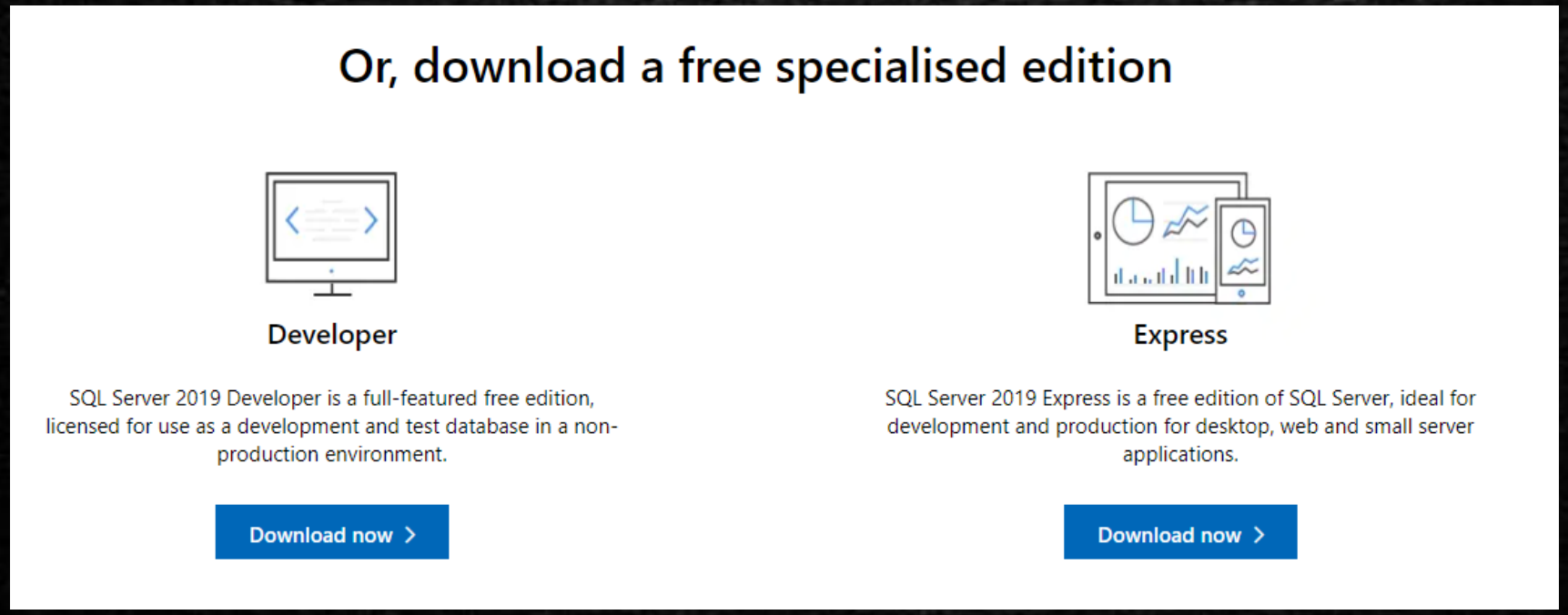
- SQL Server is software (A Relational Database Management System) developed by Microsoft.
- It is also called MS SQL Server. It is implemented from the specification of RDBMS
- The interface tool for SQL Server is SQL Server Management Studio (SSMS)

Usage of Microsoft SQL Server

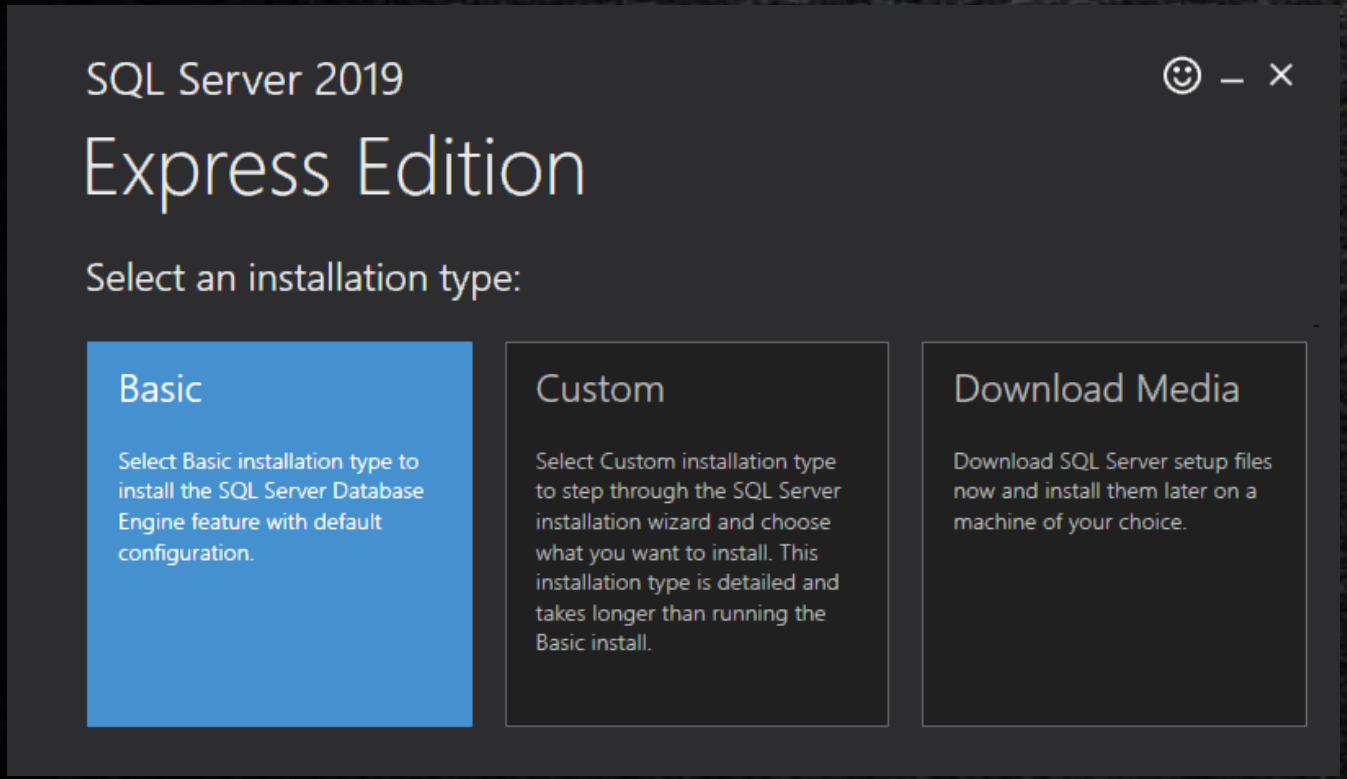
- To build and maintain databases.
- To analyze the data using SQL Server Analysis Services (SSAS).
- To generate reports using SQL Server Reporting Services (SSRS).
- To perform Extract Transform Load operations using SQL Server Integration Services (SSIS).

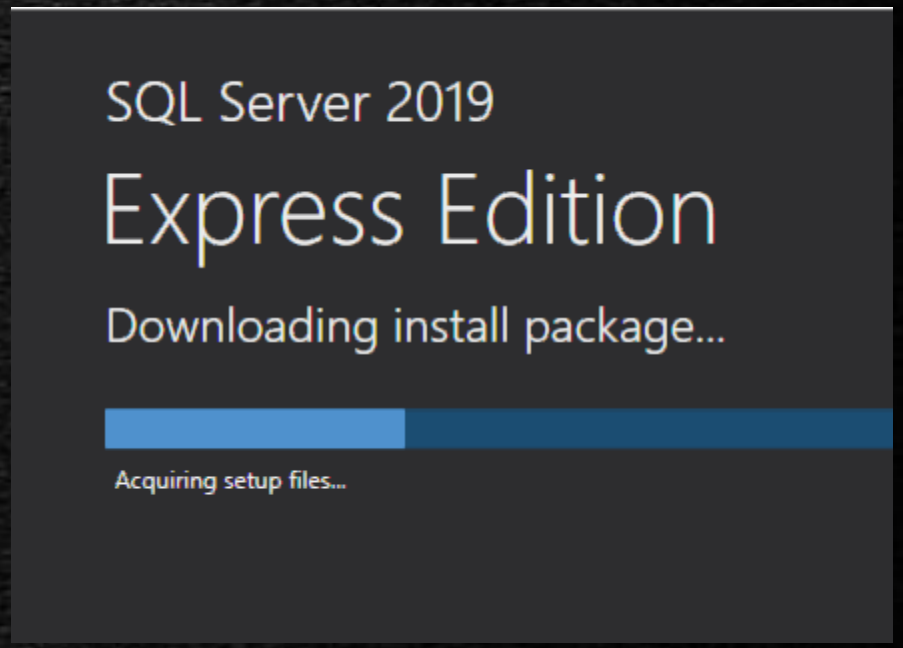
Installing Microsoft SQL Server

- Go to https://www.microsoft.com/en-in/sql-server/sql-server-downloads
- Download the Express Edition

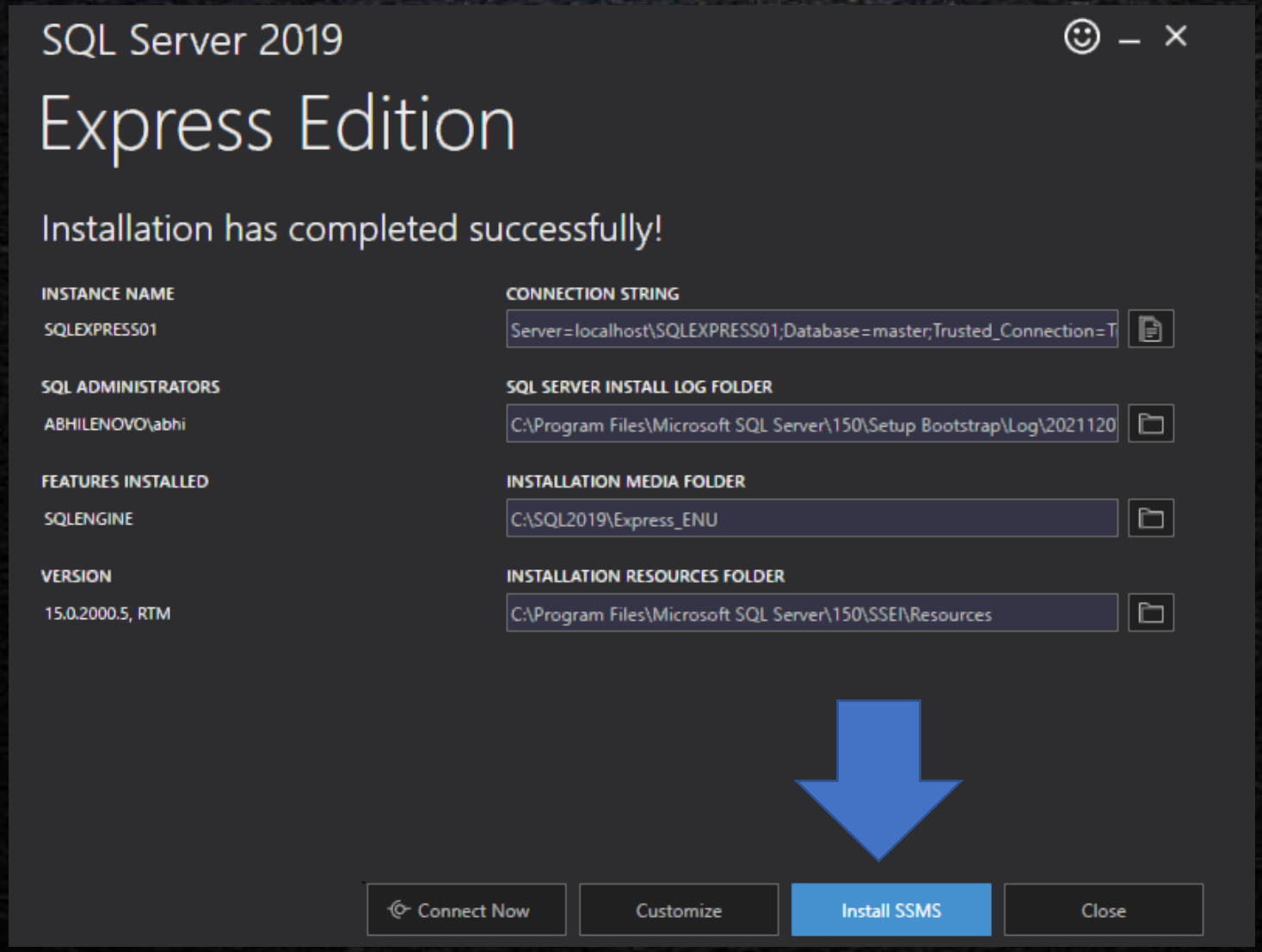


Installing Microsoft SQL Server





Installing Microsoft SQL Server



Installing Microsoft SQL Server Management Studio

Download SQL Server Management Studio (SSMS)

Article • 12/04/2021 • 7 minutes to read • +34 Is this page helpful?

Applies to: SQL Server (all supported versions) Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

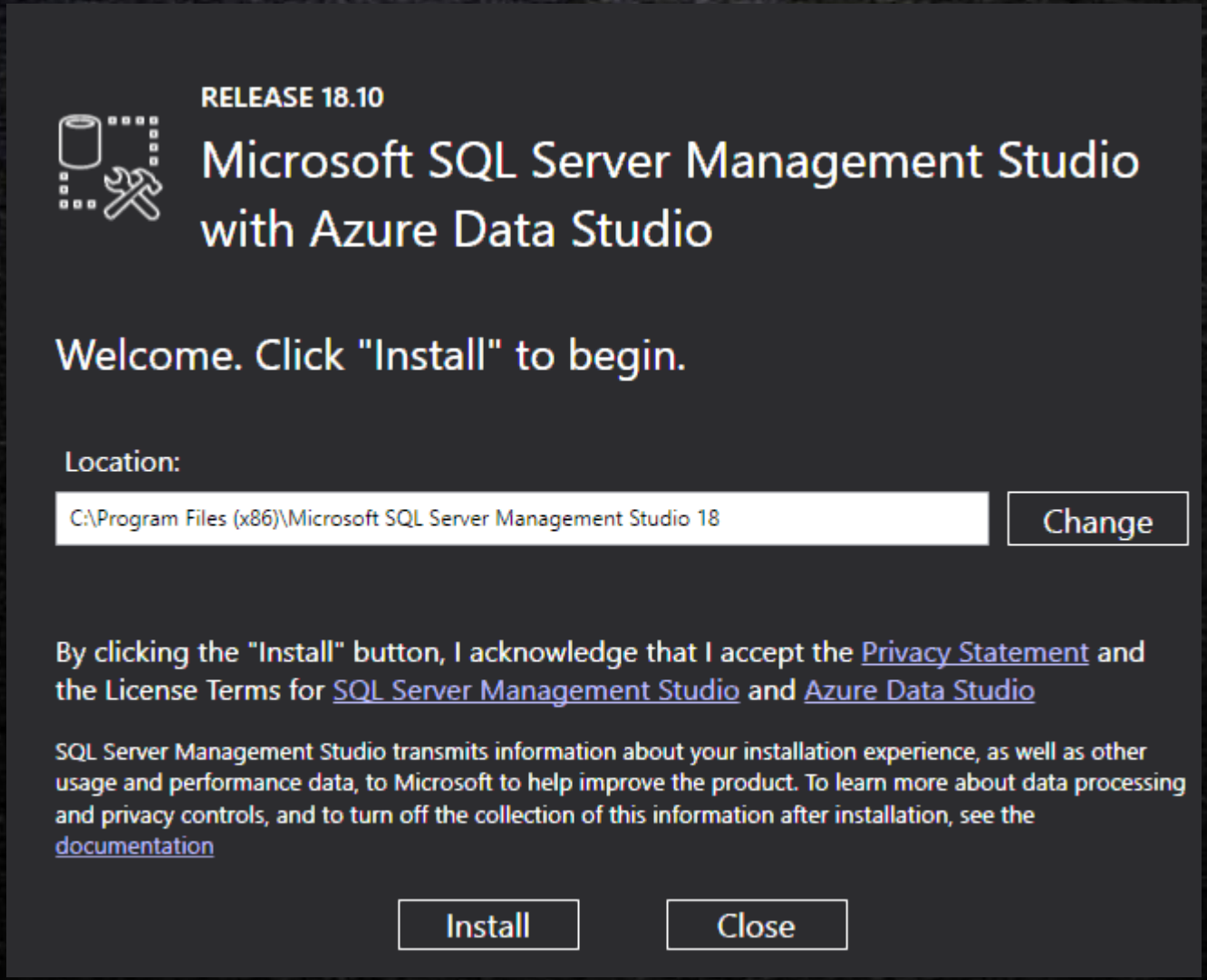
SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications, and build queries and scripts.

Use SSMS to query, design, and manage your databases and data warehouses, wherever they are - on your local computer, or in the cloud.

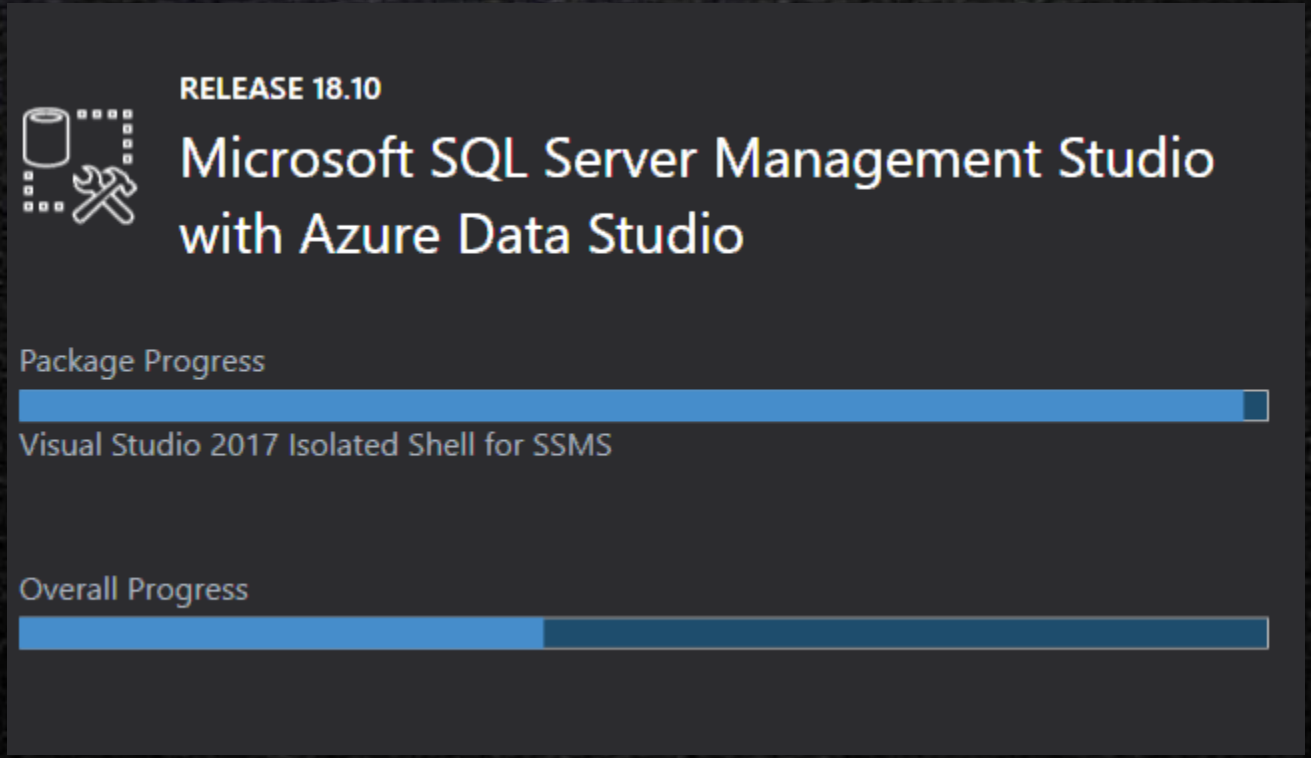
Download SSMS

Free Download for SQL Server Management Studio (SSMS) 18.10

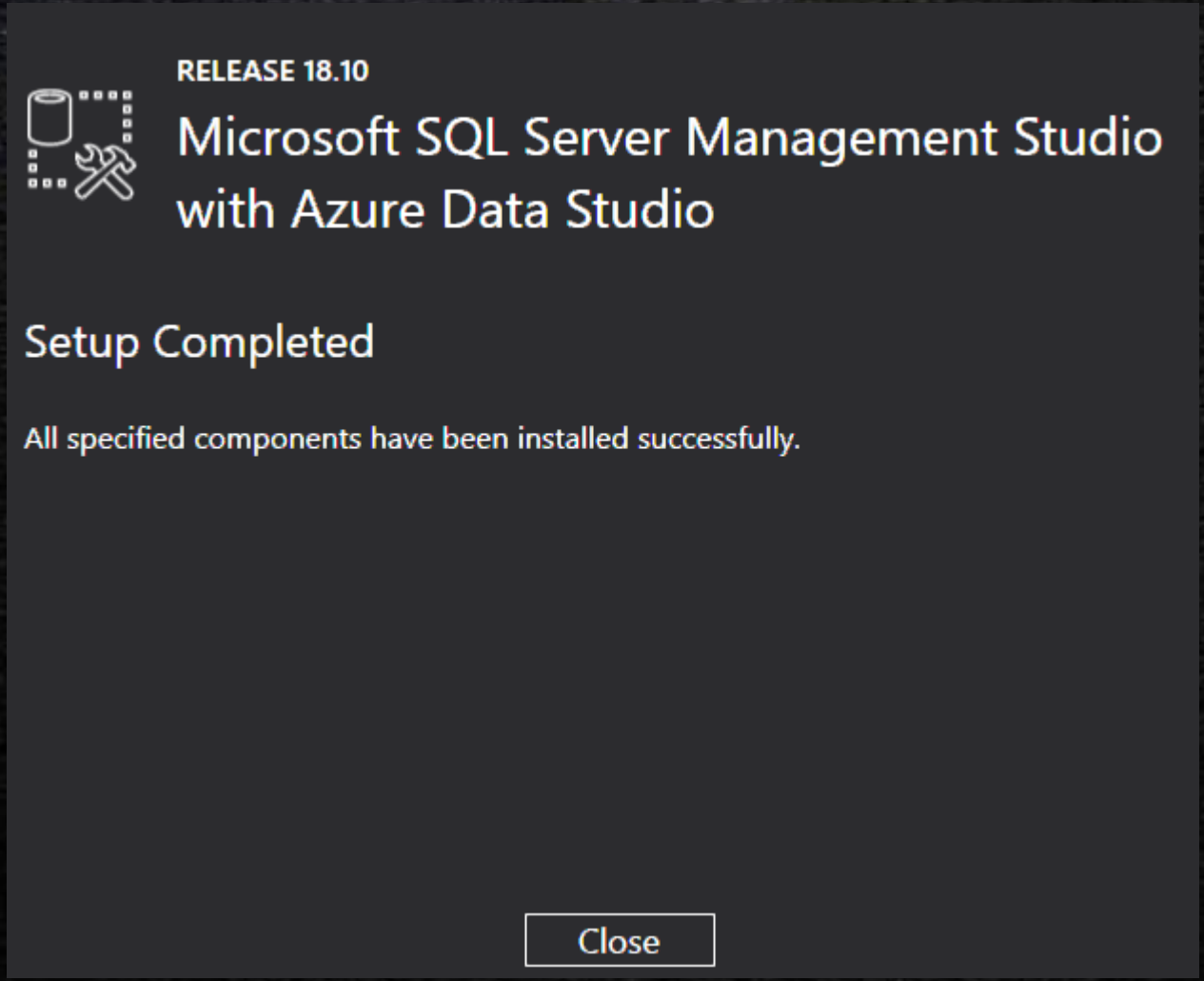
Installing Microsoft SQL Server Management Studio



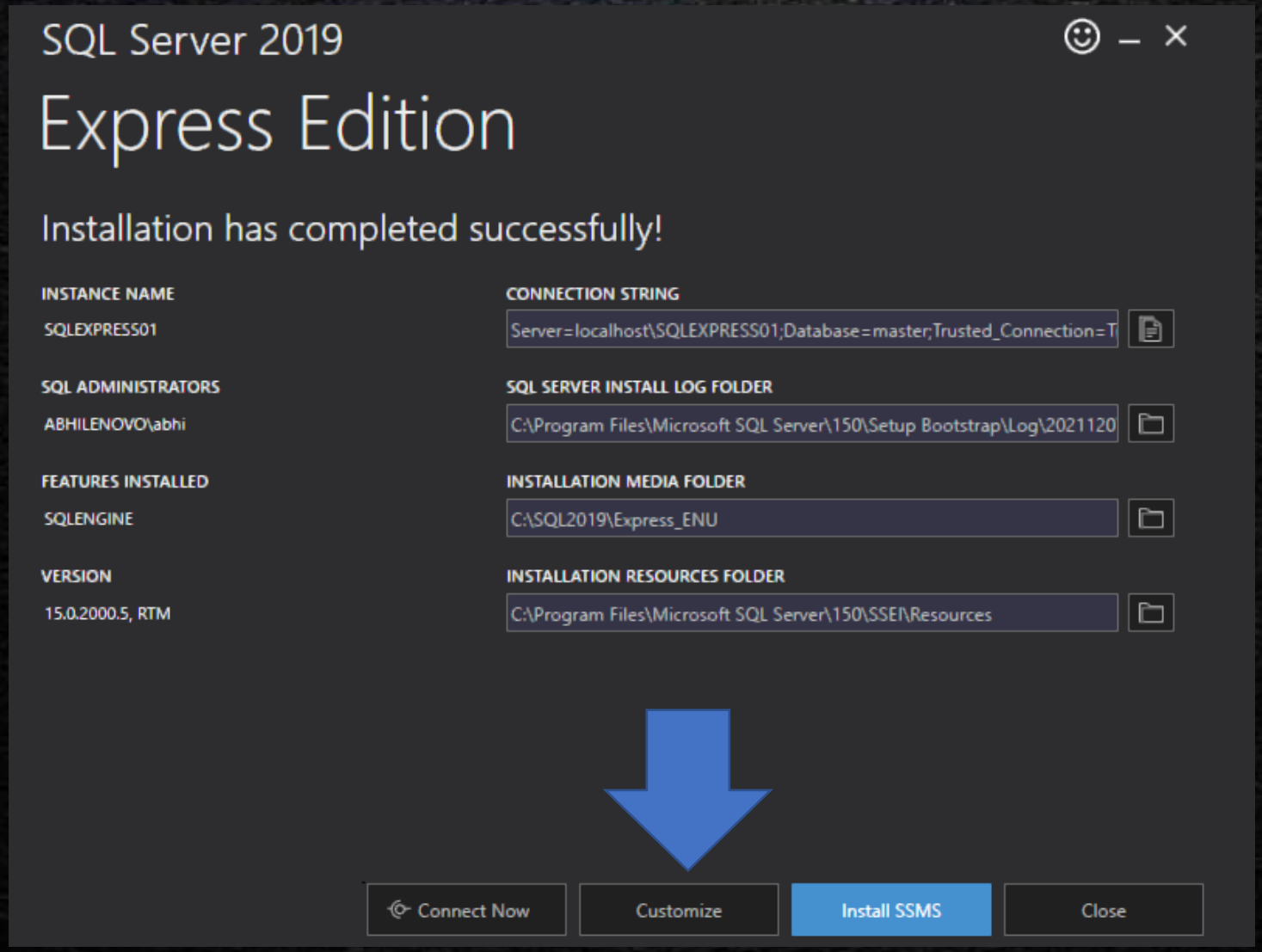
Installing Microsoft SQL Server Management Studio



Installing Microsoft SQL Server Management Studio



Installing Microsoft SQL Server



Installing Microsoft SQL Server

Global Rules

Setup Global Rules identify problems that might occur when you install SQL Server Setup support files. Failures must be corrected before Setup can continue.

- **Global Rules**
- Microsoft Update
- Product Updates
- Install Setup Files
- Install Rules
- Installation Type
- License Terms
- Feature Selection
- Feature Rules
- Feature Configuration Rules
- Installation Progress
- Complete

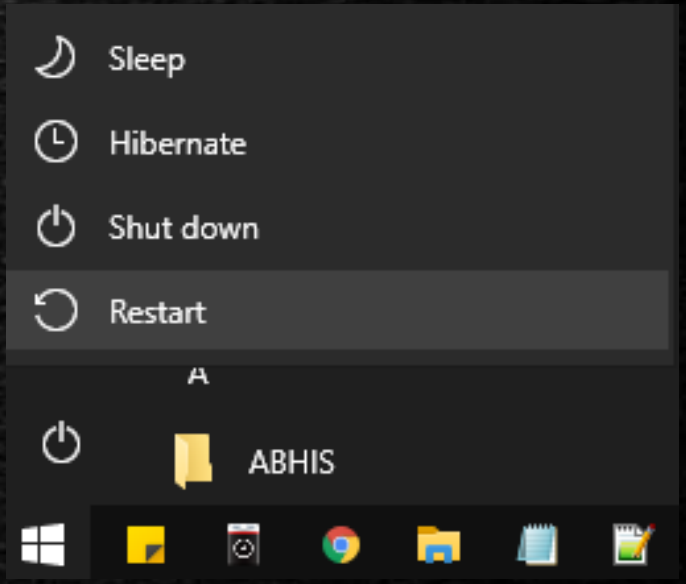
Operation completed. Passed: 7. Failed 1. Warning 0. Skipped 0.

Hide details <<

Re-run

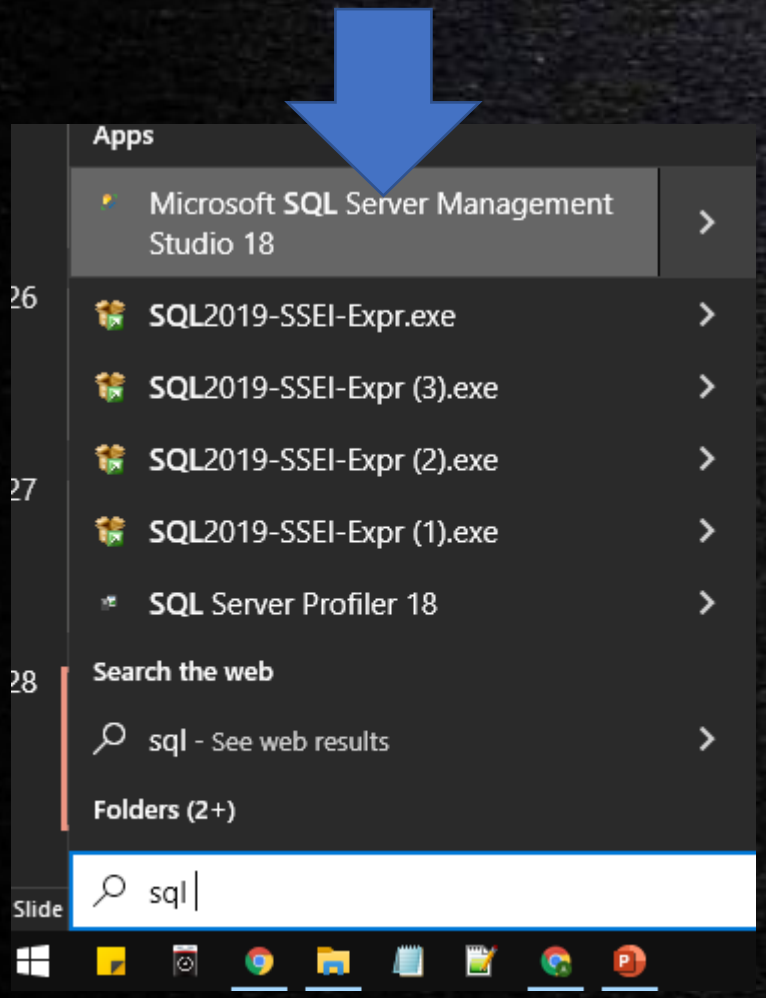
View detailed report

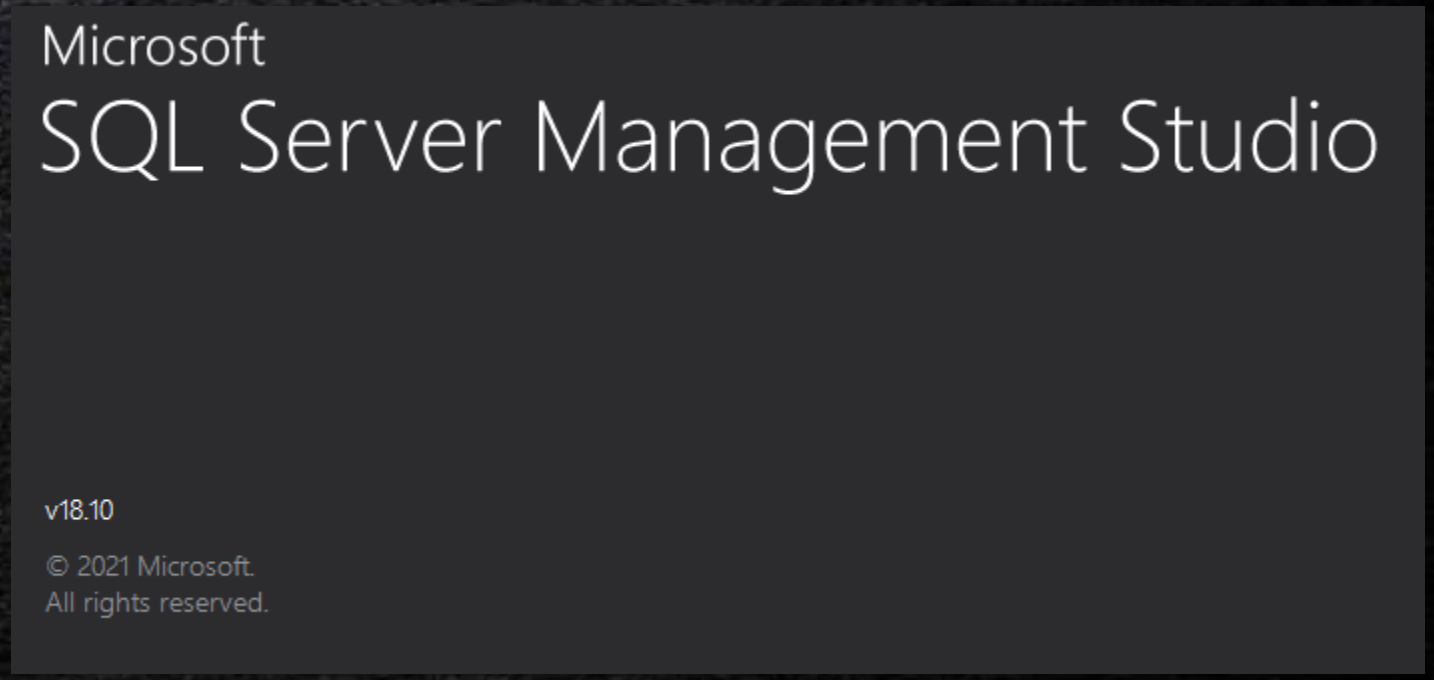
Result	Rule	Status
✓	Setup administrator	Passed
✓	Setup account privileges	Passed
✗	Restart computer	Failed
✓	Windows Management Instrumentation (WMI) service	Passed
✓	Consistency validation for SQL Server registry keys	Passed
✓	Long path names to files on SQL Server installation media	Passed
✓	SQL Server Setup Product Incompatibility	Passed
✓	Edition WOW64 platform	Passed



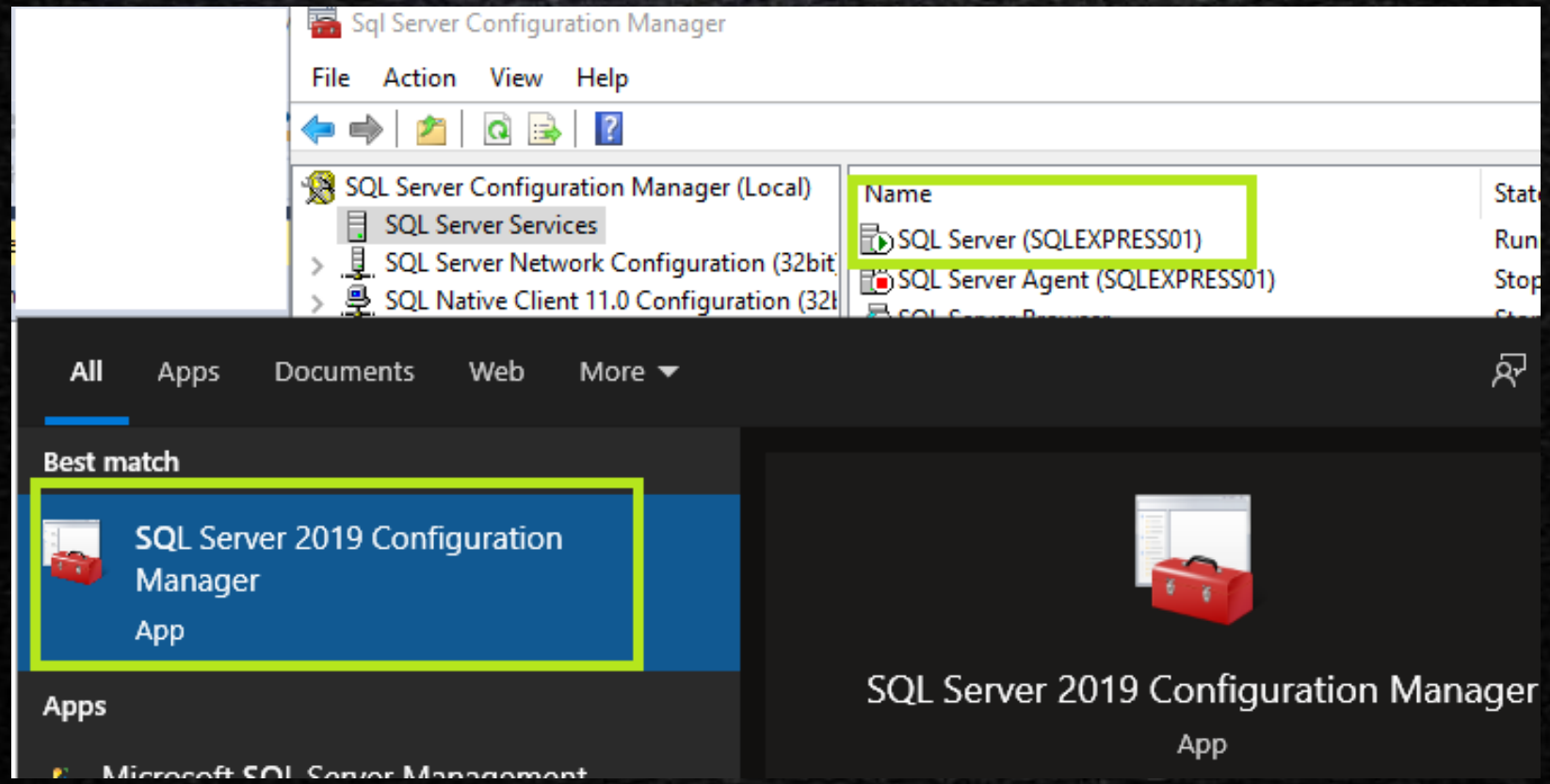
Close and Quit setup, then Restart your computer

Open Microsoft SQL Server Management Studio

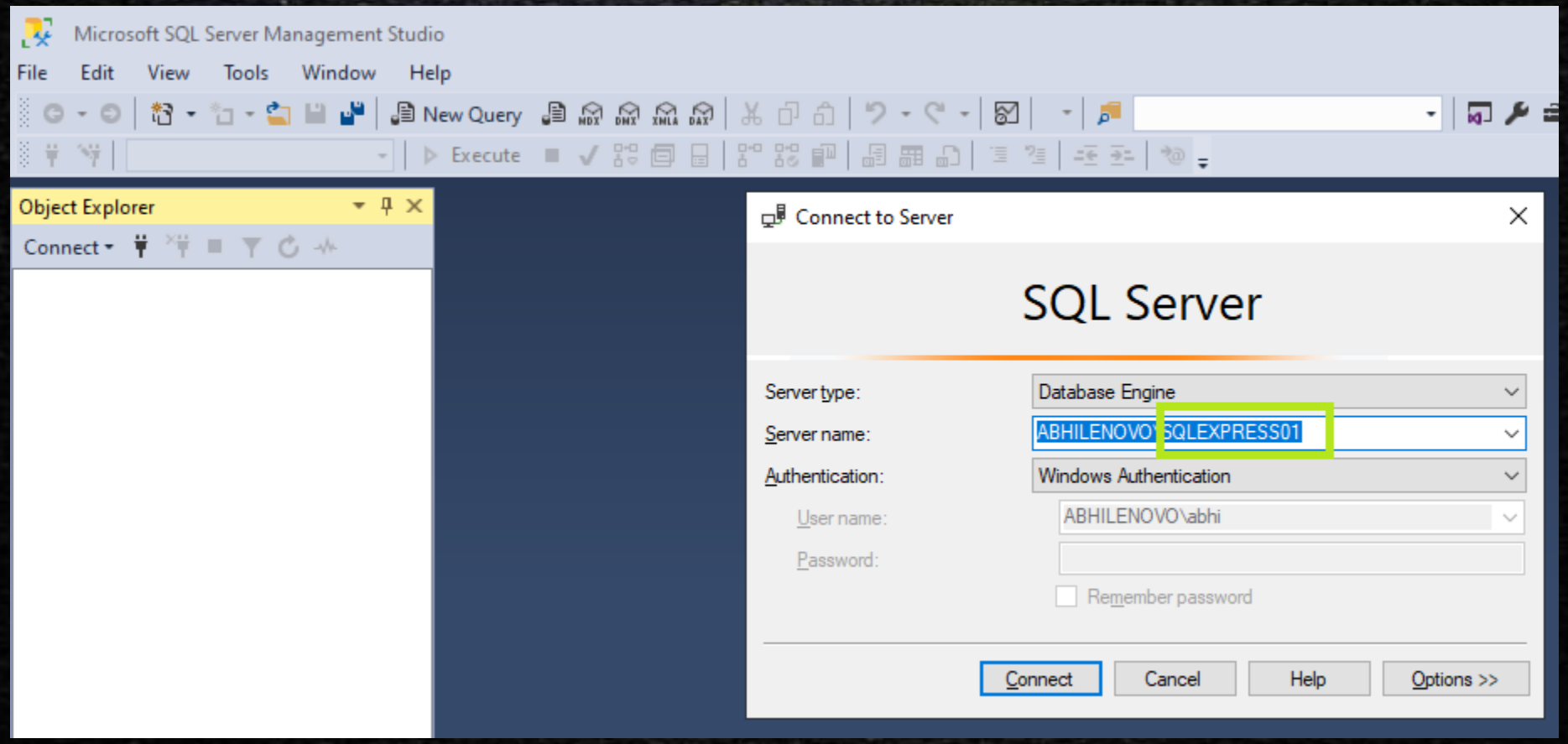




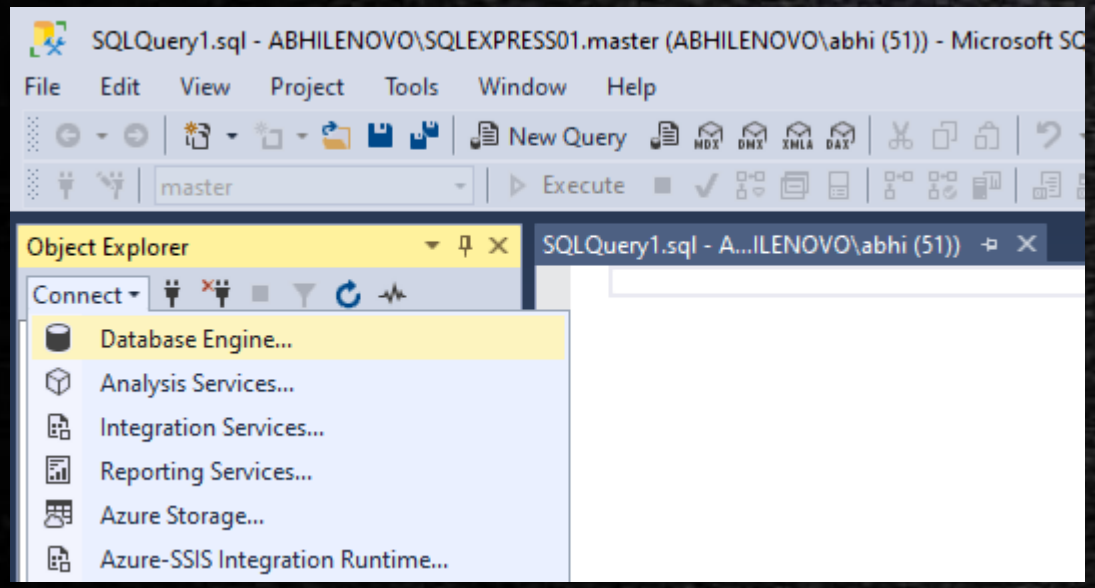
Open Microsoft SQL Server Config and find server name

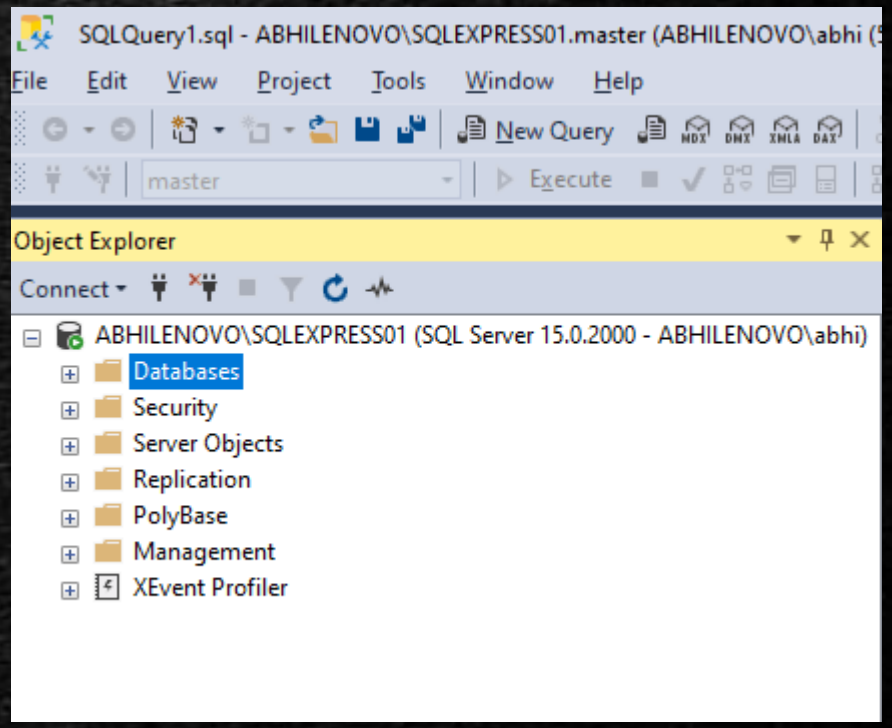


In SQL Server Management Studio connect to that server

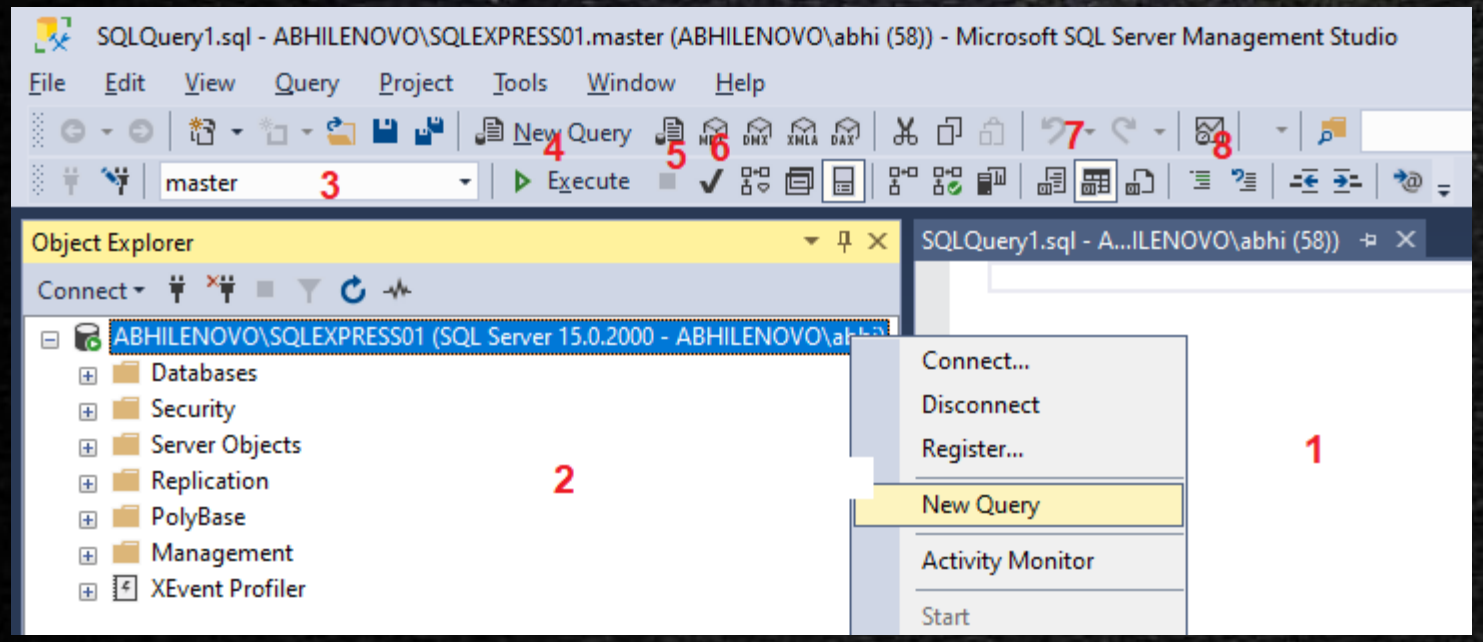


In SQL Server Management Studio connect to that server





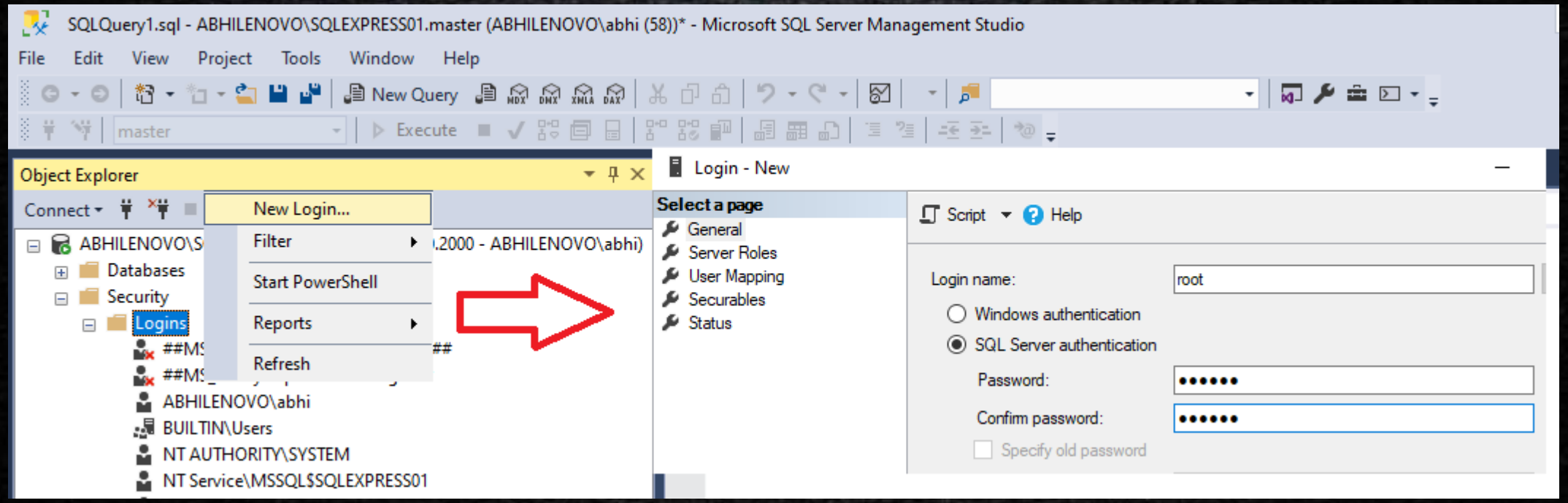
In SQL Server Management Studio Interface



7. Change Query Result Destination
8. Comment or uncomment (comment using - - before line)

1. Query Editor: This section is used to write the queries.
2. Object Explorer: Shows the database objects contained on the server in a tree format.
3. Databases Selection Dropdown : Select database to run the query
4. Execution button: Execute the query and get results
5. Cancel Query : Stop currently running query
6. Parse : Validate query syntax without checking the db objects

Option to Set a custom Login for SQL Server



- Just in case If we want to set a custom login for SQL Server.
- For our exercises we will be using the default 'Windows Authentication'

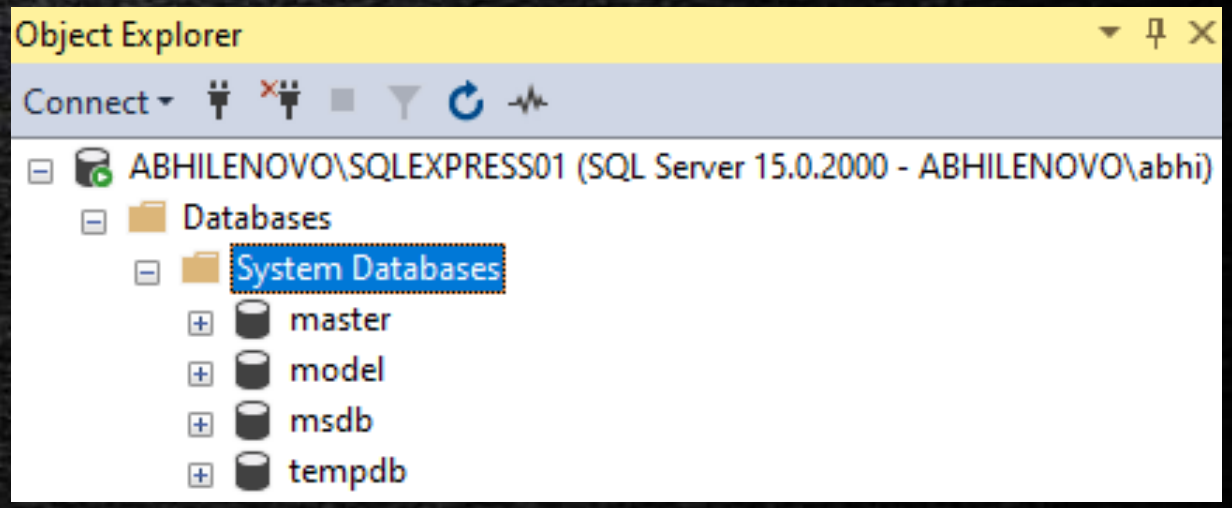
SQL Basic DB Operations

CREATE, USE, DROP and BACKUP DB

Types of DB in SQL Server

SQL Server has two types of database:

- System databases
- User Databases



- System databases are created automatically while installing the MS SQL Server.

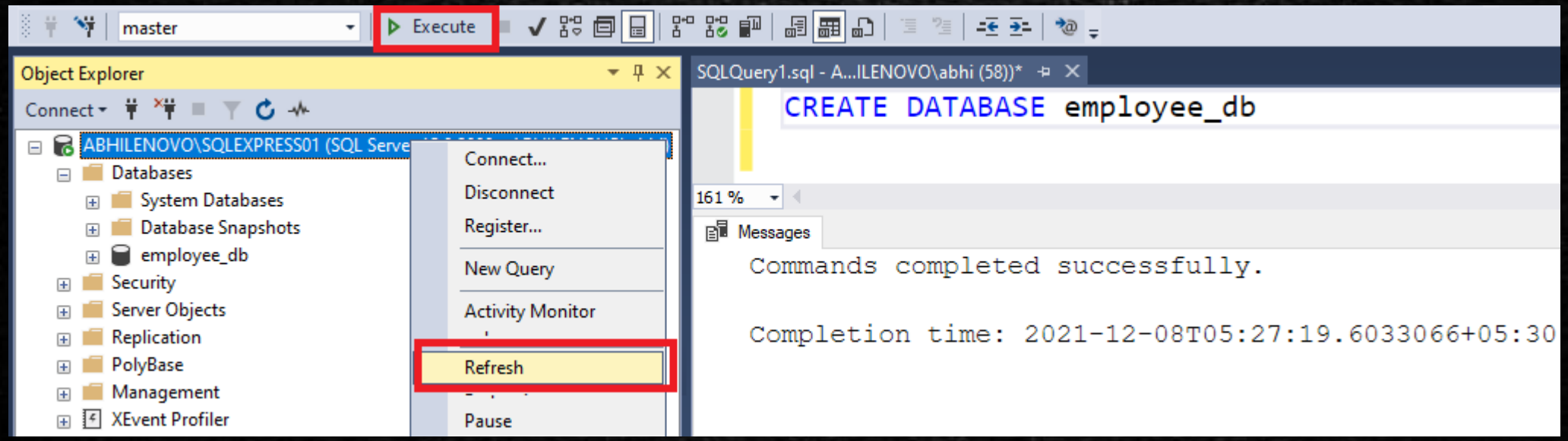
- It is essential to run the server efficiently.

Create a new user database

A new database in SQL Server can be created in two ways:

- Transact-SQL Command
- SQL Server Management Studio

- Using the TSQL Command: ‘CREATE DATABASE db_name ‘

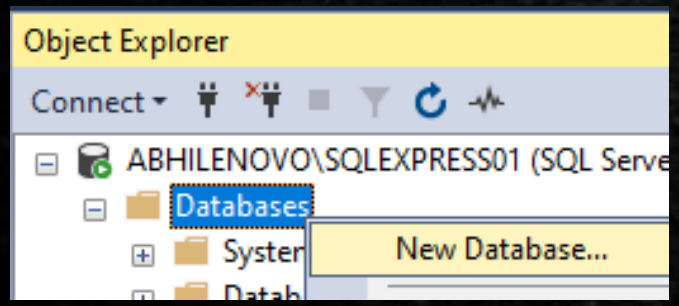


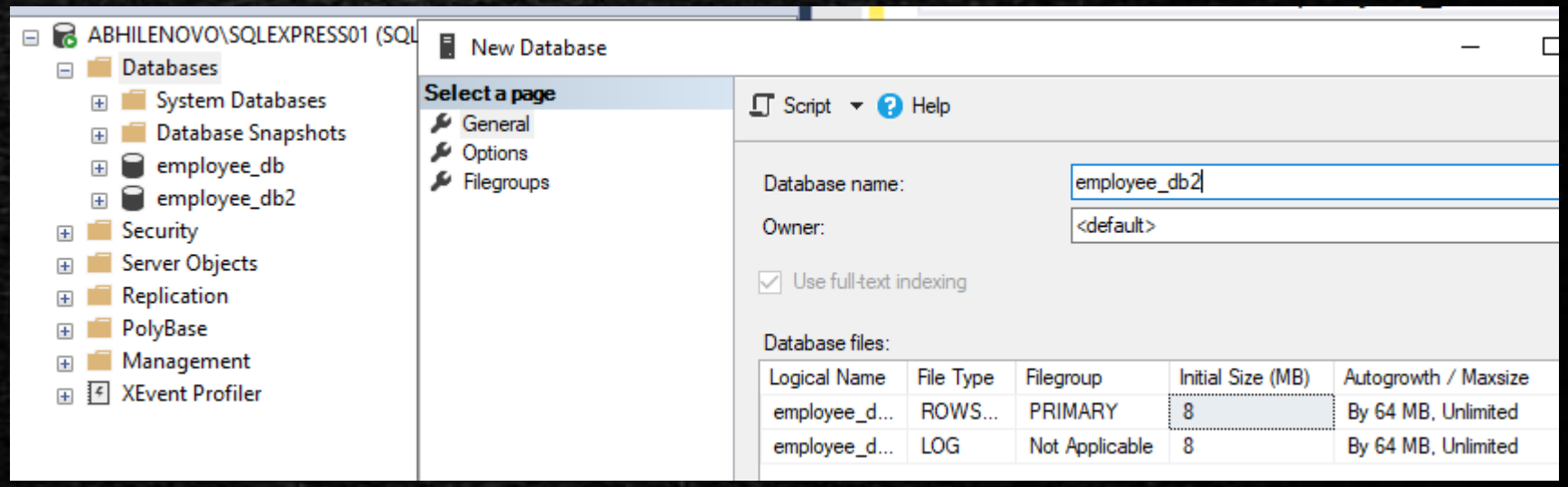
Create a new user database

A new database in SQL Server can be created in two ways:

- Transact-SQL Command
- SQL Server Management Studio

- Using SQL Server Management Studio





List All Databases (MSSQL)

Executing this stored procedure will display the ‘view’ of all databases

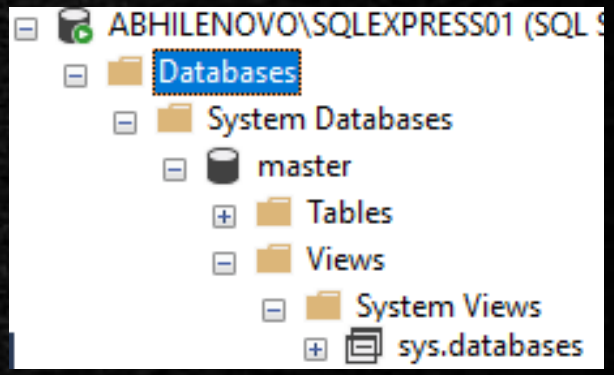
SQLQuery1.sql - A...ILENOVO\abhi (58))*

```
--CREATE DATABASE employee_db
SELECT name FROM master.sys.databases ORDER BY name;
```

161 %

Results | Messages

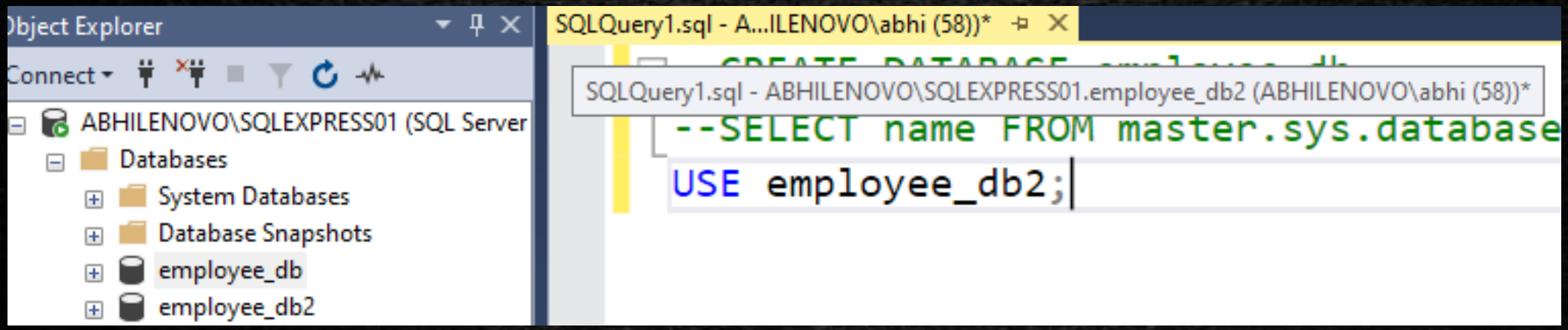
	name
1	employee_db
2	employee_db2
3	master
4	model
5	msdb
6	tempdb



Select a database (SQL)

You may either click on the db on the left side and execute query OR

Use the ‘USE db_name;’ SQL query

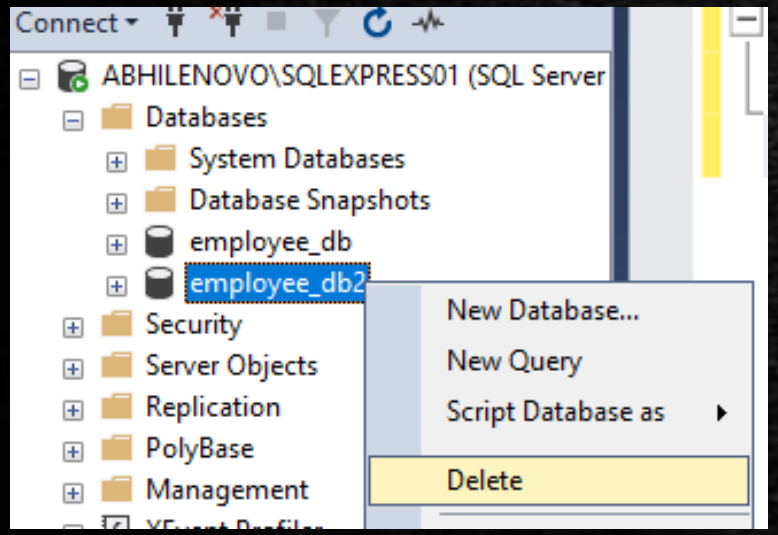


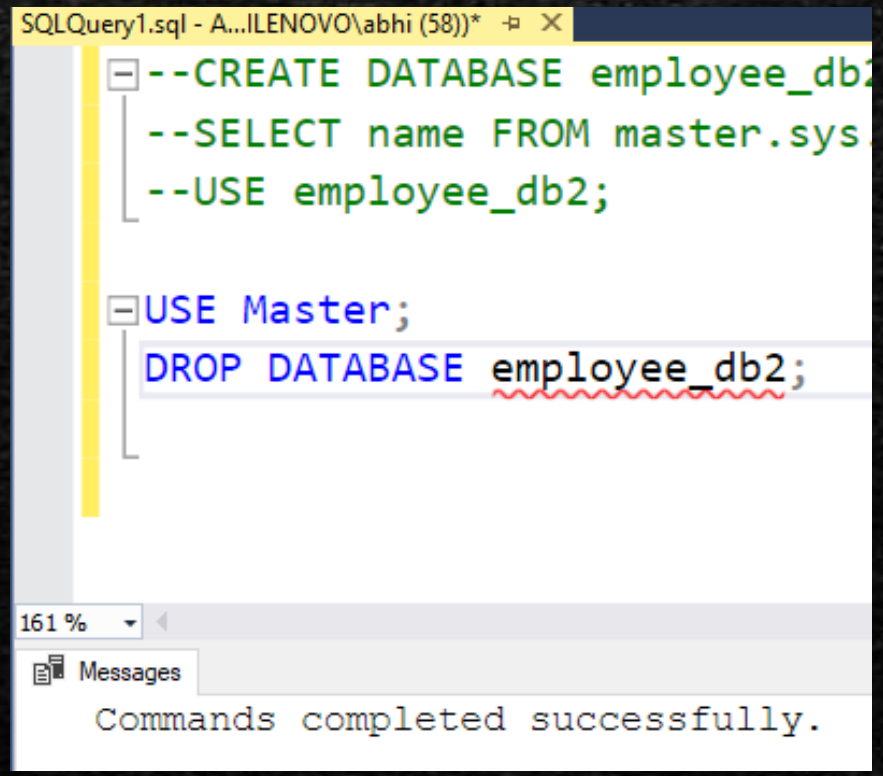
Delete a database (SQL)

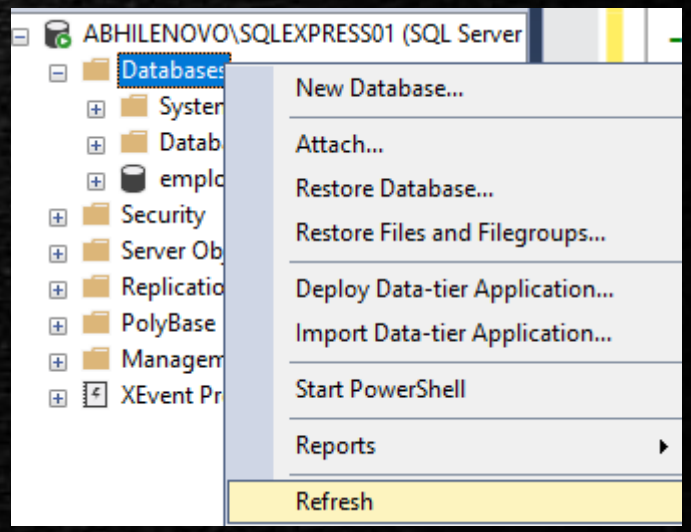
You may either right click on the db and select ‘Delete’ OR

Use the ‘DROP DATABASE db_name;’ SQL query

(Make sure to ‘unuse’ the db before drop)







Backup a database (SQL)

You may either right click on the db and select ‘Tasks >> Backup’ OR

Use the ‘BACKUP DATABASE db_name TO DISK = ‘path’;’ SQL query

```
BACKUP DATABASE employee_db TO DISK = 'D:\db_backup\employee_db.bak'
```

61 %

Messages

```
Processed 376 pages for database 'employee_db', file 'employee_db' on file 1.
Processed 2 pages for database 'employee_db', file 'employee_db_log' on file 1.
BACKUP DATABASE successfully processed 378 pages in 0.289 seconds (10.204 MB/sec).
```

This PC › LENOVO (D:) › db_backup

Name

employee_db.bak

Backup a database (SQL)

You may either right click on the db and select ‘Tasks >> Backup’ OR

Use the ‘BACKUP DATABASE db_name TO DISK = ‘path’ WITH DIFFERENTIAL;’ SQL query to backup only the changes

```
BACKUP DATABASE employee_db TO DISK = 'D:\db_backup\employee_db.bak' WITH DIFFERENTIAL
```

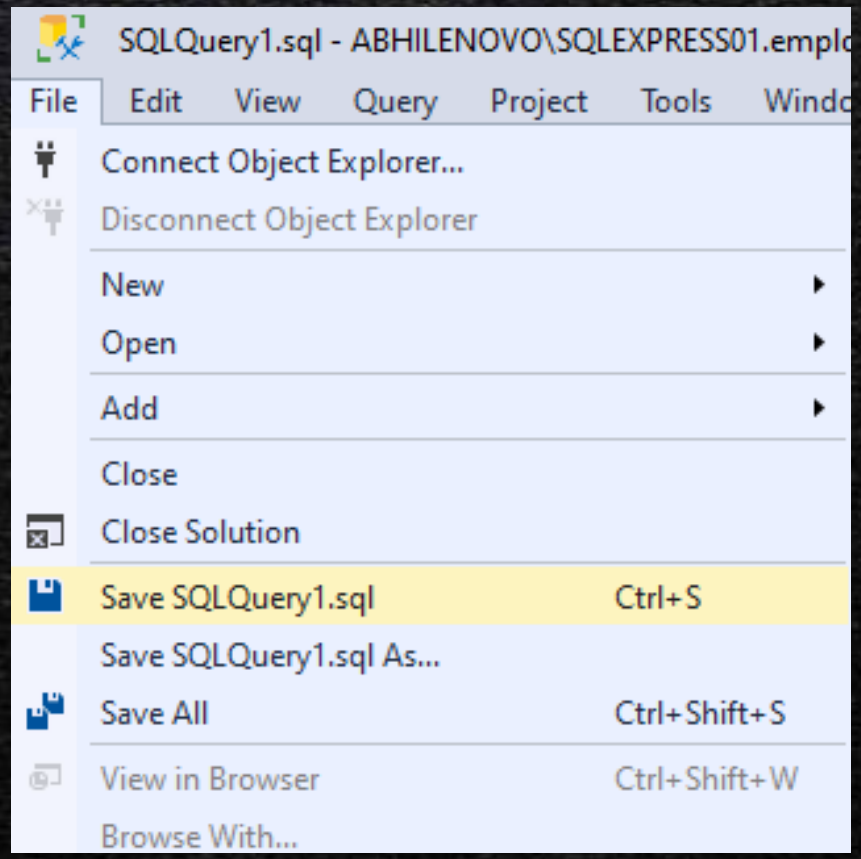
Messages

```
Processed 56 pages for database 'employee_db', file 'employee_db' on file 2.
Processed 2 pages for database 'employee_db', file 'employee_db_log' on file 2.
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 58 pages in 0.062 seconds (7.245 MB/sec).
```

Backup the SQL file

We can re-use the queries by saving them as a .sql file

File >> Save

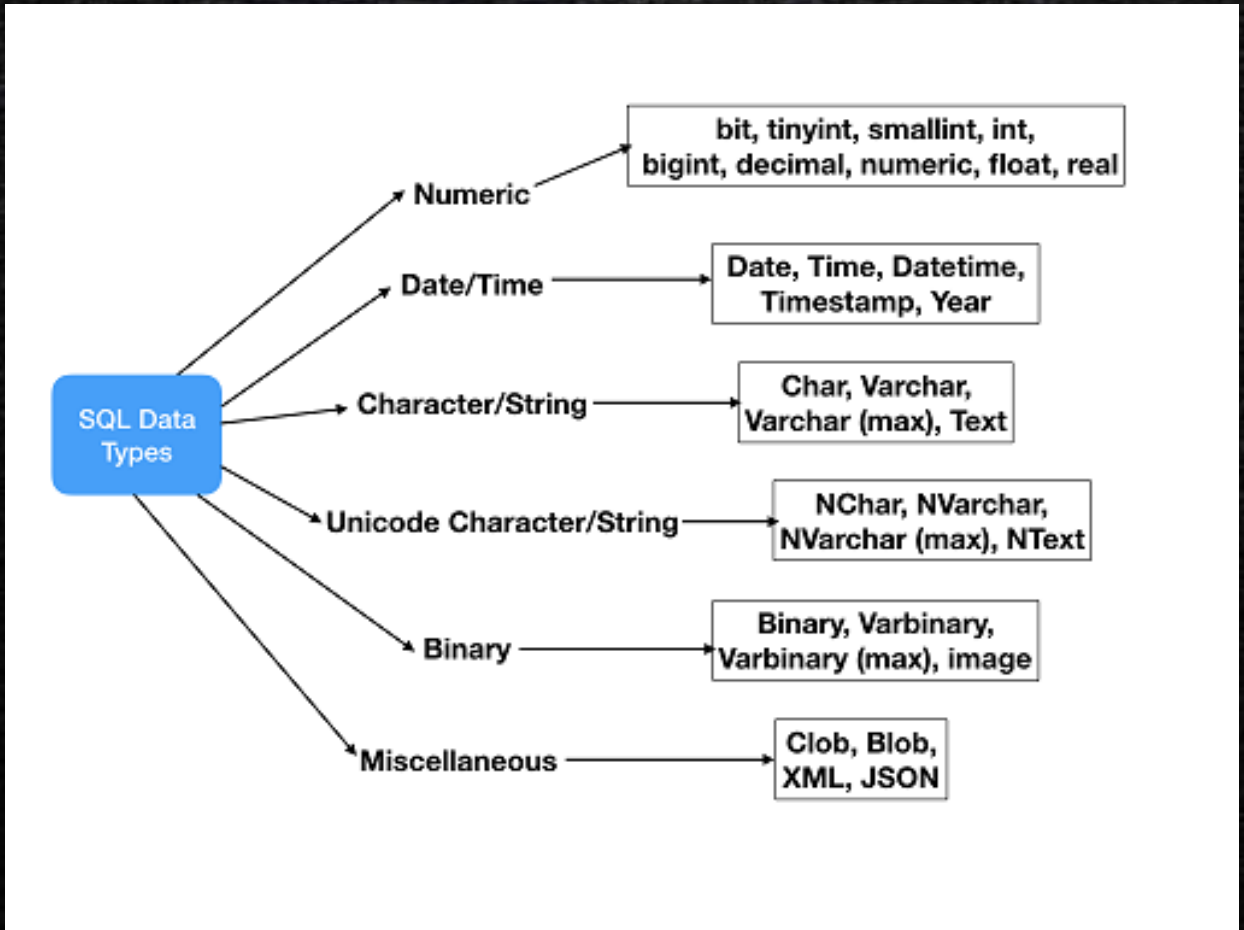


SQL Basic Table Operations

Basic DDL Operations: CREATE, DROP, ALTER

SQL Data Types

Using Data Type, we specify what type of data is expected inside of each column. The common data types are :



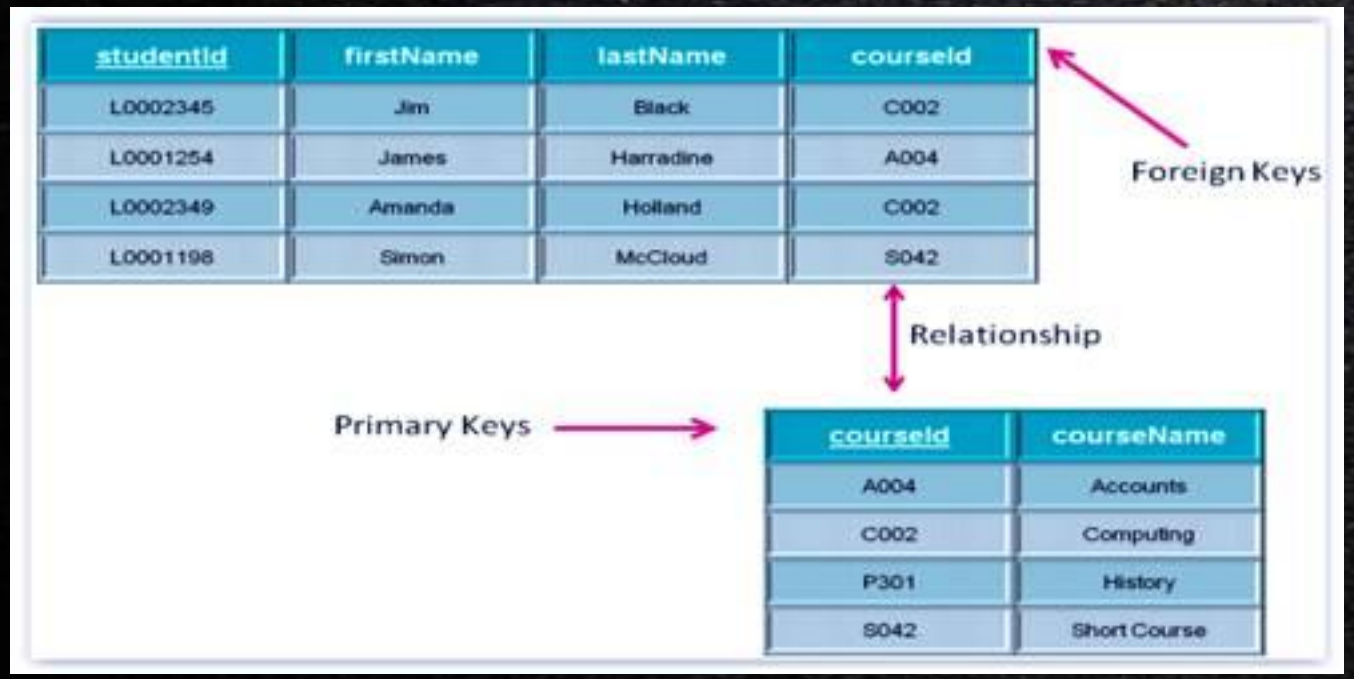
SQL Constraints

SQL constraints are used to specify rules for the data in a table.

Column level constraints apply to a column, and table level constraints apply to the whole table.

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE.
- IDENTITY data type generates autoincrementing integer
- FOREIGN KEY - links between tables
- CHECK - Ensures values satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - To create and get data from database very quickly

Primary Key vs Foreign Key

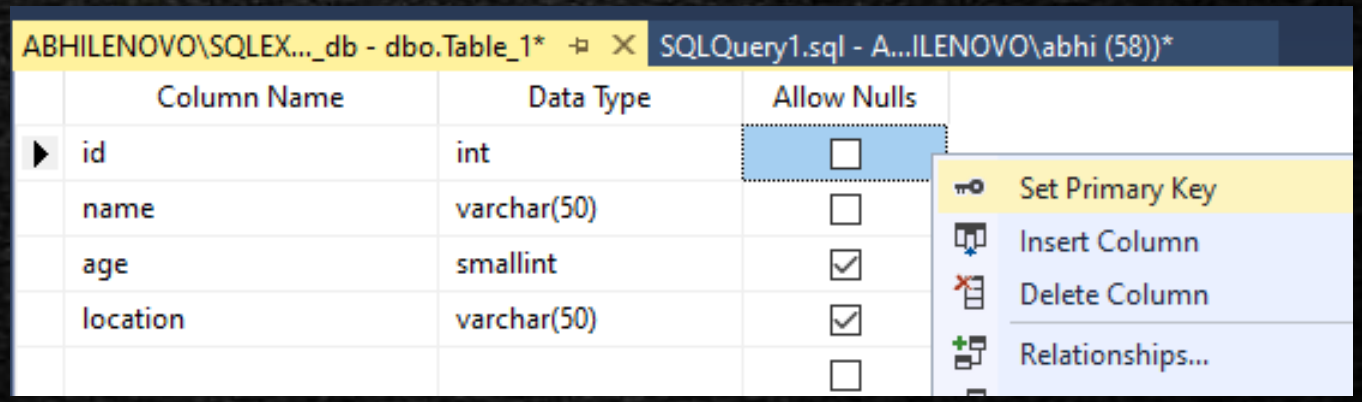


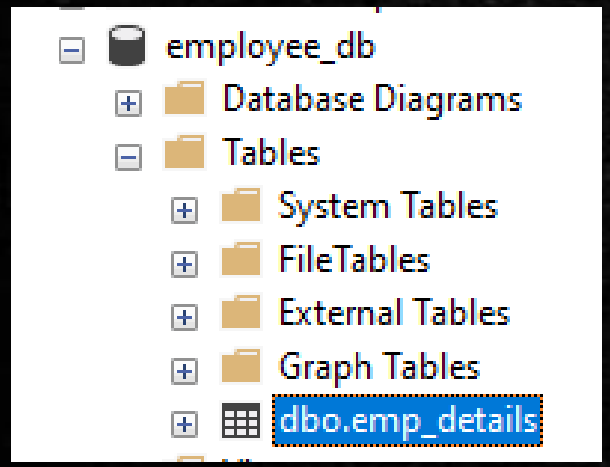
Item	Primary Key	Foreign Key
Consists of One or More Columns	Yes	Yes
Duplicate Values Allowed	No	Yes
NULLs Allowed	No	Yes
Uniquely Identify Rows In a Table	Yes	Maybe
Number allowed per table	One	Zero or More
Indexed	Automatically Indexed	No Index Automatically created

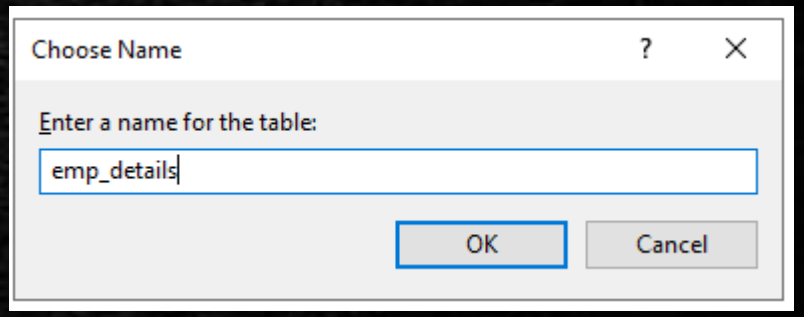
Create Table in the database

Using Management Studio:

After adding columns press ctrl+S to save the table

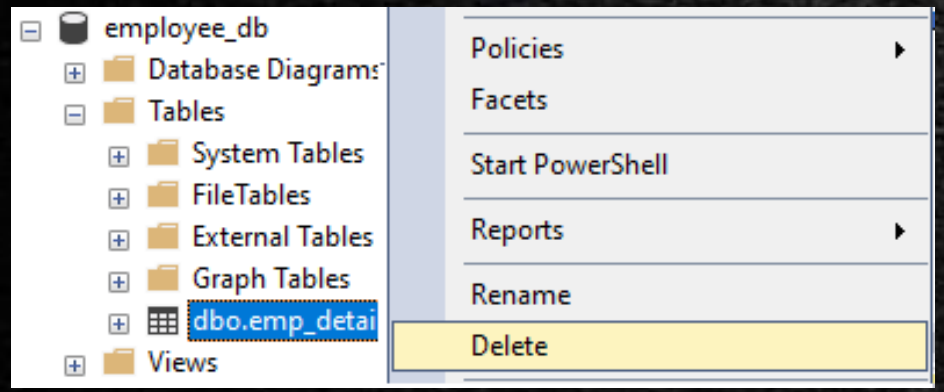


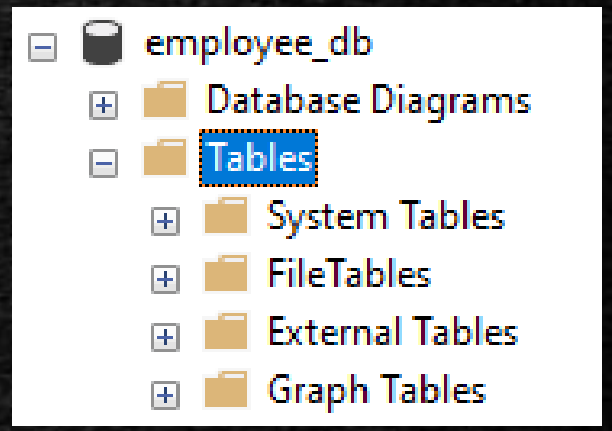




Delete Table from the database

Using Management Studio:





CREATE a table in database

Use the CREATE TABLE [database_name.] table_name (

```
    column_definition1,
    column_definition2,
    ........,
    table_constraints
);
```

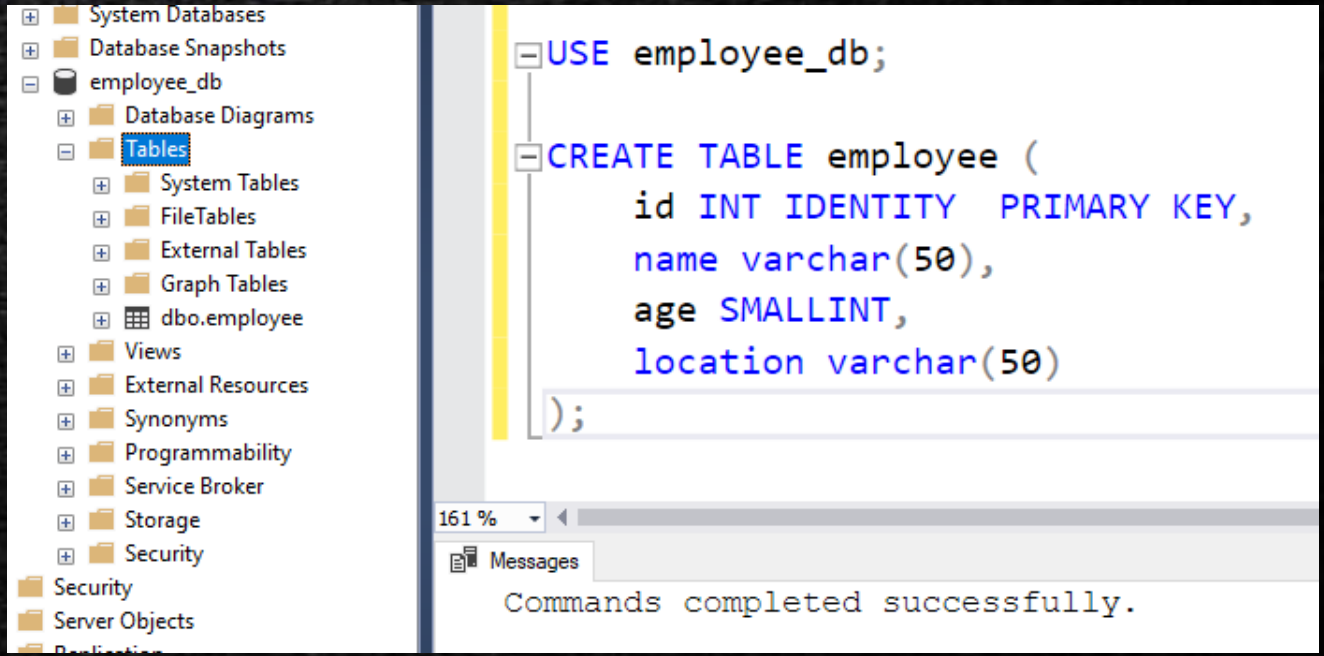
SQL query to create new table

CREATE an employee table in database

Use the CREATE TABLE table_name (

```
    column_definition1,
    column_definition2,
    ........,
    table_constraints
);
```

SQL query to create new table



ALTER a table in database

ALTER TABLE is used to add, delete, or edit columns in an existing table.

```
ALTER TABLE table_name ADD column_name datatype;

ALTER TABLE table_name DROP COLUMN column_name;

ALTER TABLE table_name ALTER COLUMN column_name datatype;
```

ALTER employee table example

```sql
--CREATE TABLE employee (
--     id INT IDENTITY  PRIMARY KEY,
--     name varchar(50),
--     age SMALLINT,
--     location varchar(50)
--);

ALTER TABLE employee
ADD DOB date;
```

%

Messages

```
Commands completed successfully.
```

DESCRIBE (View) Table Schema

SQL Server has built-in system stored procedure sp_help which we can Execute to view table schema

```
--ALTER TABLE employee
--ADD DOB date;

EXEC sp_help employee;
```

161 %

Results | Messages

	Name	Owner	Type	Created_datetime
1	employee	dbo	user table	2021-12-08 07:09:16.723

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLe
1	id	int	no	4	10	0	no	(n/a)	(n/a)
2	name	varchar	no	50			yes	no	yes
3	age	smallint	no	2	5	0	yes	(n/a)	(n/a)
4	location	varchar	no	50			yes	no	yes
5	DOB	date	no	3	10	0	yes	(n/a)	(n/a)

	Identity	Seed	Increment	Not For Replication
1	id	1	1	0

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

	index_name	index_description	index_keys
1	PK__employee__3213E83F9F9C90AC	clustered, unique, primary key located on PRIMARY	id

INSERT Data into table

We can

- Add data in a single row
- Add data in multiple rows

```
INSERT INTO [table_name]
(col_name1, col_name2, ... )
VALUES
(value1, value2, ... );
```

```
INSERT INTO [table_name]
(col_name1, col_name2, ... )
VALUES
(value1, value2, ... )
(value1, value2, ... )
(value1, value2, ... )
(value1, value2, ... );
```

INSERT Data into table

```sql
INSERT INTO employee (name, age, location,dob)
VALUES ('Tom', 2, 'USA','2018-10-20'),
('Jerry', 1, 'USA','2018-10-20'),
('Mickey', 3, 'USA','2018-10-20');
```

essages

```
(3 rows affected)

Completion time: 2021-12-08T08:07:32.7467824+05:30
```

Fetch data in the table using SELECT statement

To fetch all the columns

SELECT * FROM table_name;

```sql
INSERT INTO employee (name, age, location,dob)
VALUES ('Tom', 2, 'USA','2018-10-20'),
('Jerry', 1, 'USA','2018-10-20'),
('Mickey', 3, 'USA','2018-10-20');

GO

select * from employee;
```

161 %

Results | Messages

	id	name	age	location	DOB
1	1	donald	2	USA	NULL
2	2	Jerry	1	USA	NULL
3	3	Mickey	3	USA	NULL
4	4	Tom	2	USA	2018-10-20
5	5	Jerry	1	USA	2018-10-20
6	6	Mickey	3	USA	2018-10-20
7	7	Tom	2	USA	2018-10-20
8	8	Jerry	1	USA	2018-10-20
9	9	Mickey	3	USA	2018-10-20

Query executed successfully. ABHILEN

Update rows of data using UPDATE statement

```
UPDATE table_name
SET column1 = new_value1,
      column2 = new_value2, ...
[WHERE Clause] ;
```

Update a row of data using UPDATE statement

```sql
UPDATE employee
SET name = 'donald'
WHERE name='Tom';
```

61 %

Results | Messages

```
(1 row affected)
```

```sql
select * from employee;
```

161 %

Results | Messages

	id	name	age	location	DOB
1	1	donald	2	USA	NULL
2	2	Jerry	1	USA	NULL
3	3	Mickey	3	USA	NULL

DELETE a row of data using DELETE statement

DELETE FROM table_name
[WHERE Clause] ;

```
DELETE FROM employee WHERE name='Tom'
GO
select * from employee;
```

161 %

Results | Messages

	id	name	age	location	DOB
1	1	donald	2	USA	NULL
2	2	Jerry	1	USA	NULL
3	3	Mickey	3	USA	NULL

DROP a table

USE db_name;

DROP TABLE table_name ;

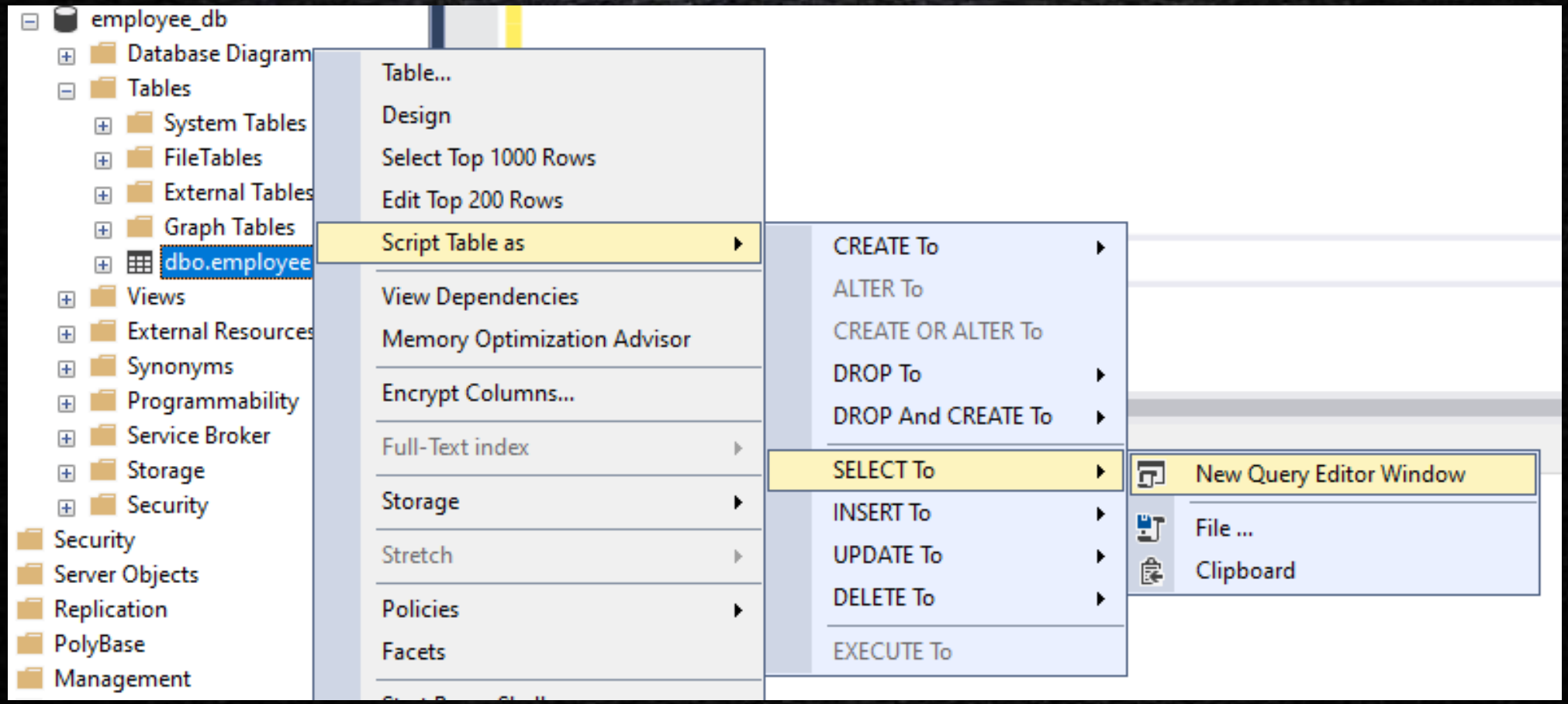
```
DROP TABLE employee;
```

1 %

Messages

```
Commands completed successfully.
```

CREATE, DROP , SELECT, UPDATE, DELTE using SQL Server



DAY 1 Assignment

Create a database called ‘hospital_db’

Create a table called ‘patient’ and add the following fields

- Patient Record no (primary key)
- Patient name
- Phone number
- Gender
- Age
- Location

Do insert alter and delete operations