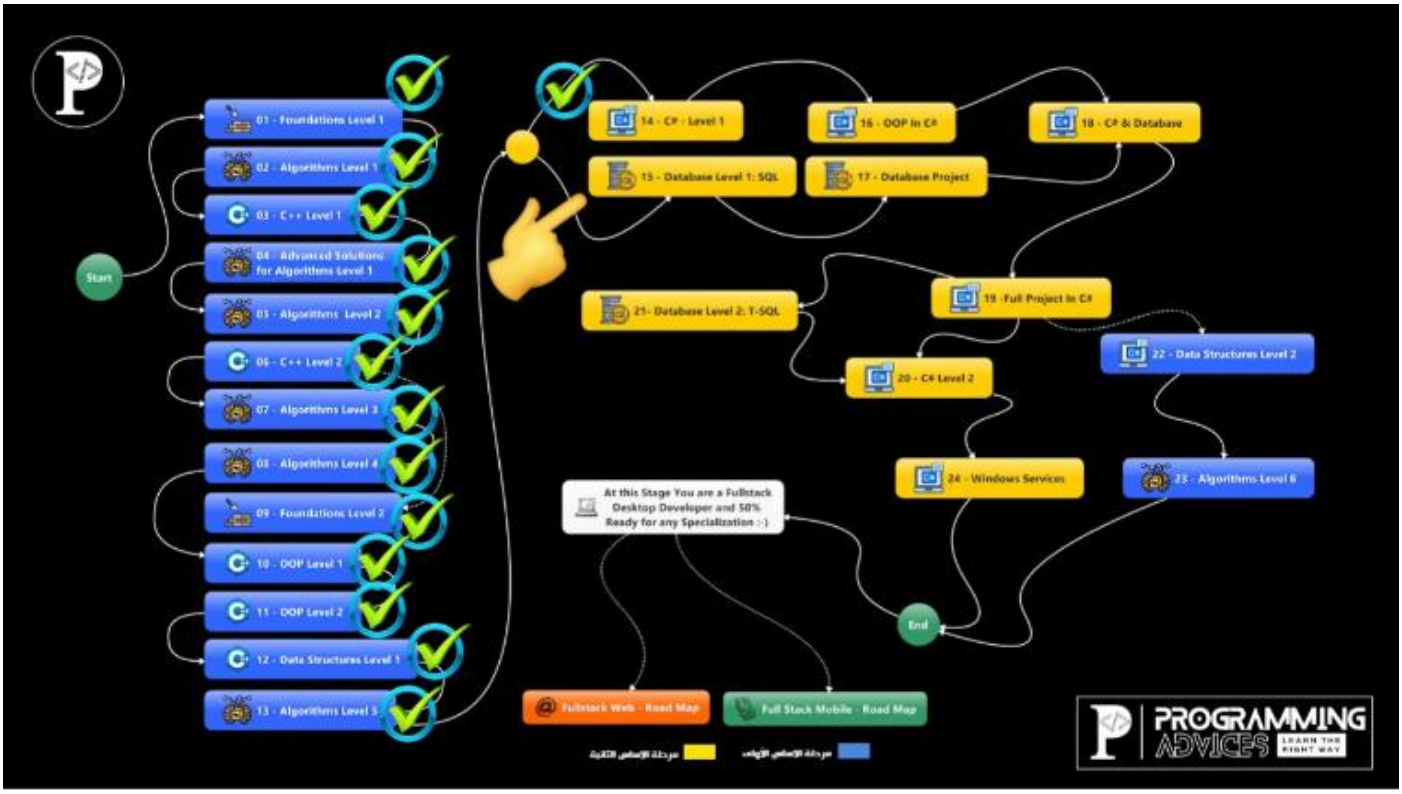
Intoduction

About This Course

من الحاجات اللي بتؤثر في سرعة البرنامج هوا الكود اللي بتعمله في الداتا بيز



Telegram Group

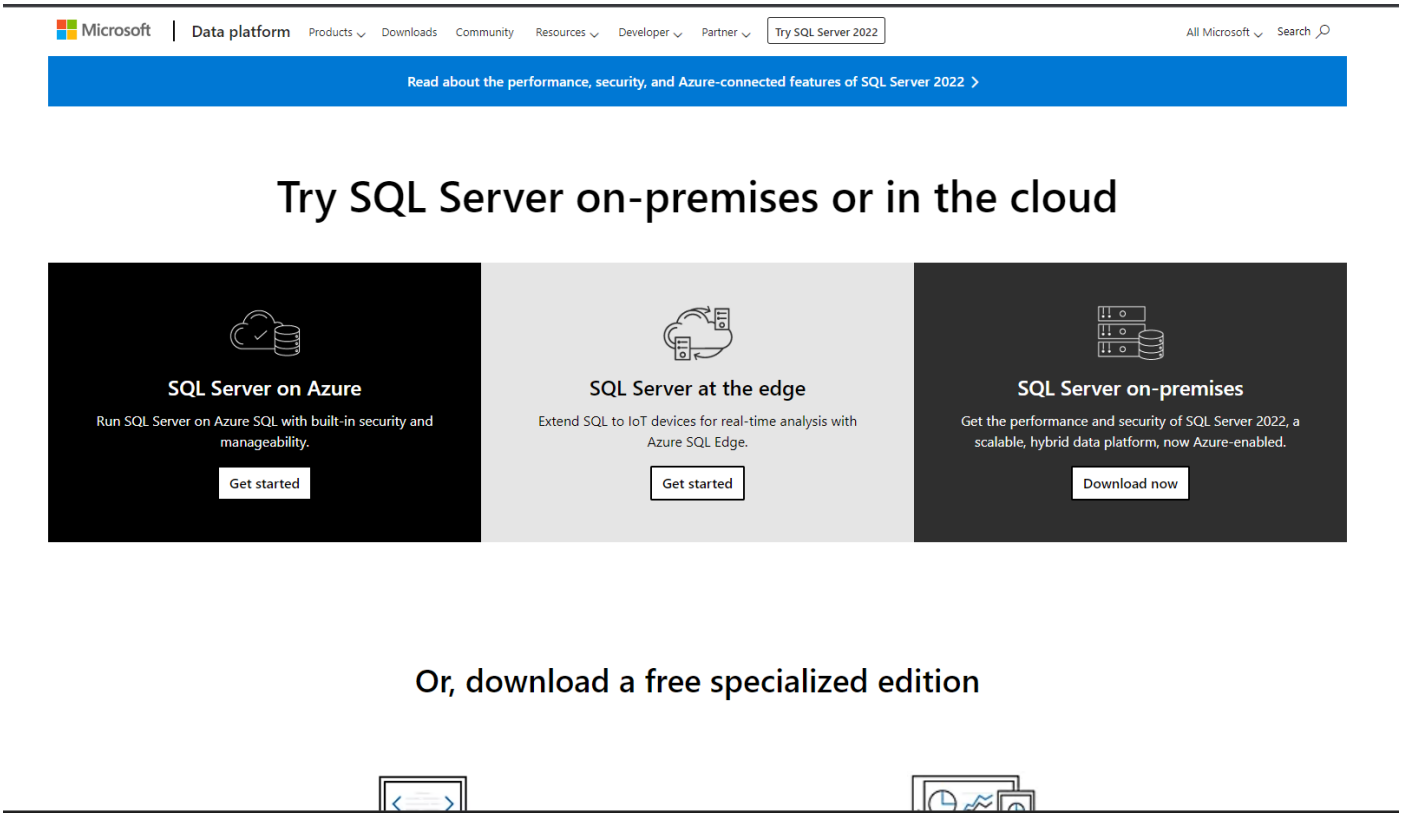
https://t.me/+qXIggnw944VkNThk

How to install SQL Server 2022

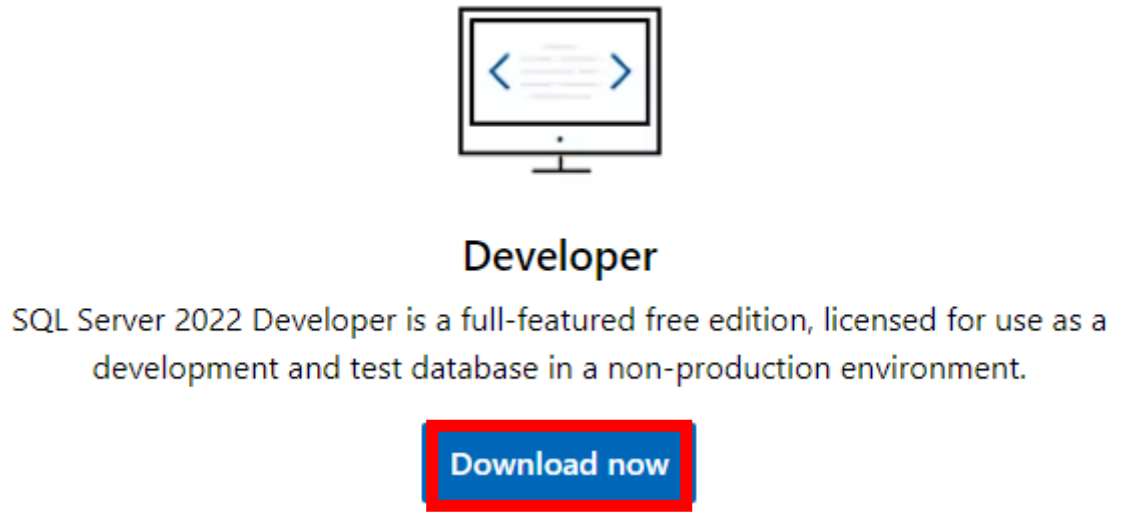
هنزل SQL server developer edition وهيا full featured من ال SQL server وبتمكنك انك تعمل اللي انت عايزه علي الداتا بيز

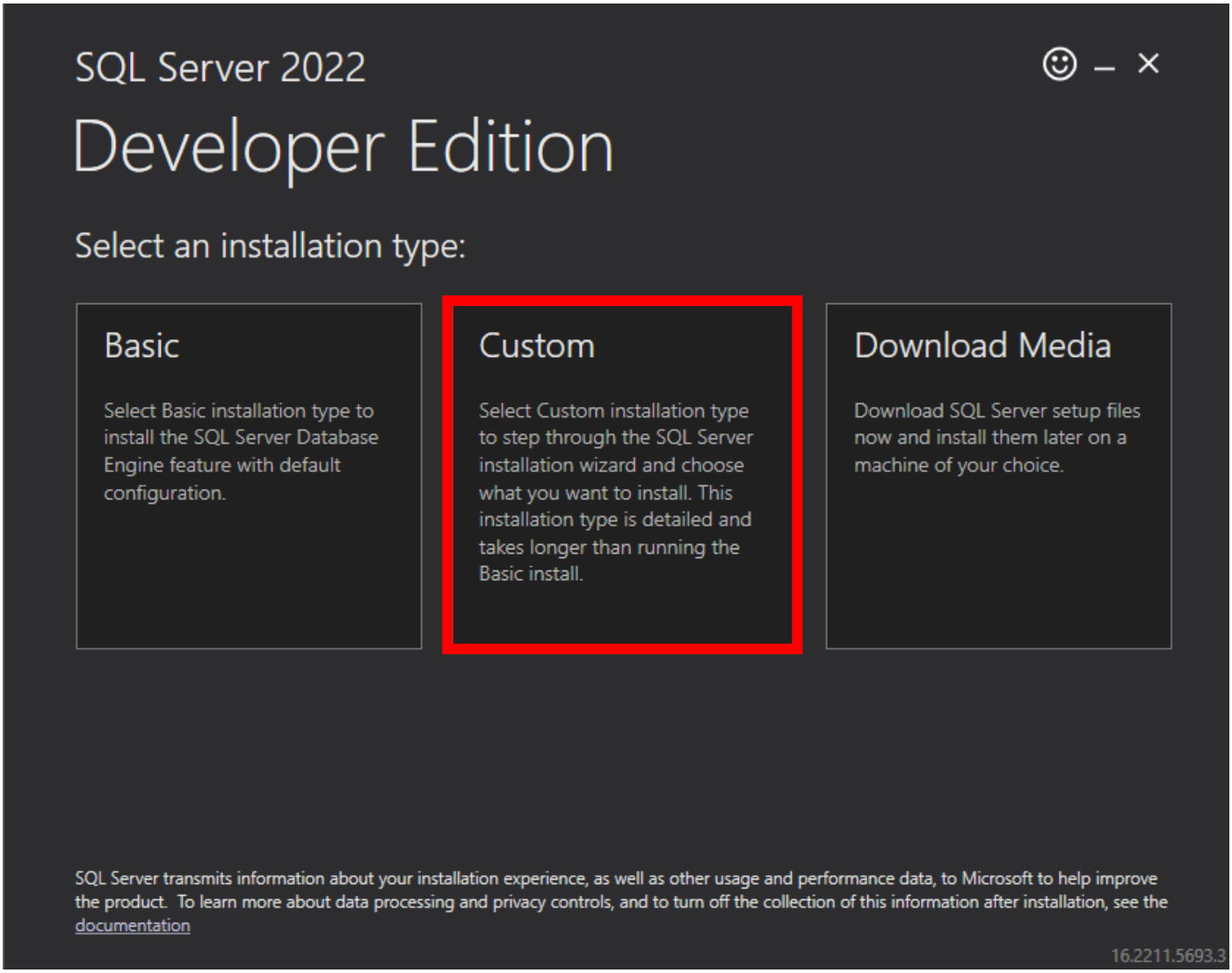
هندخل عالرابط ده

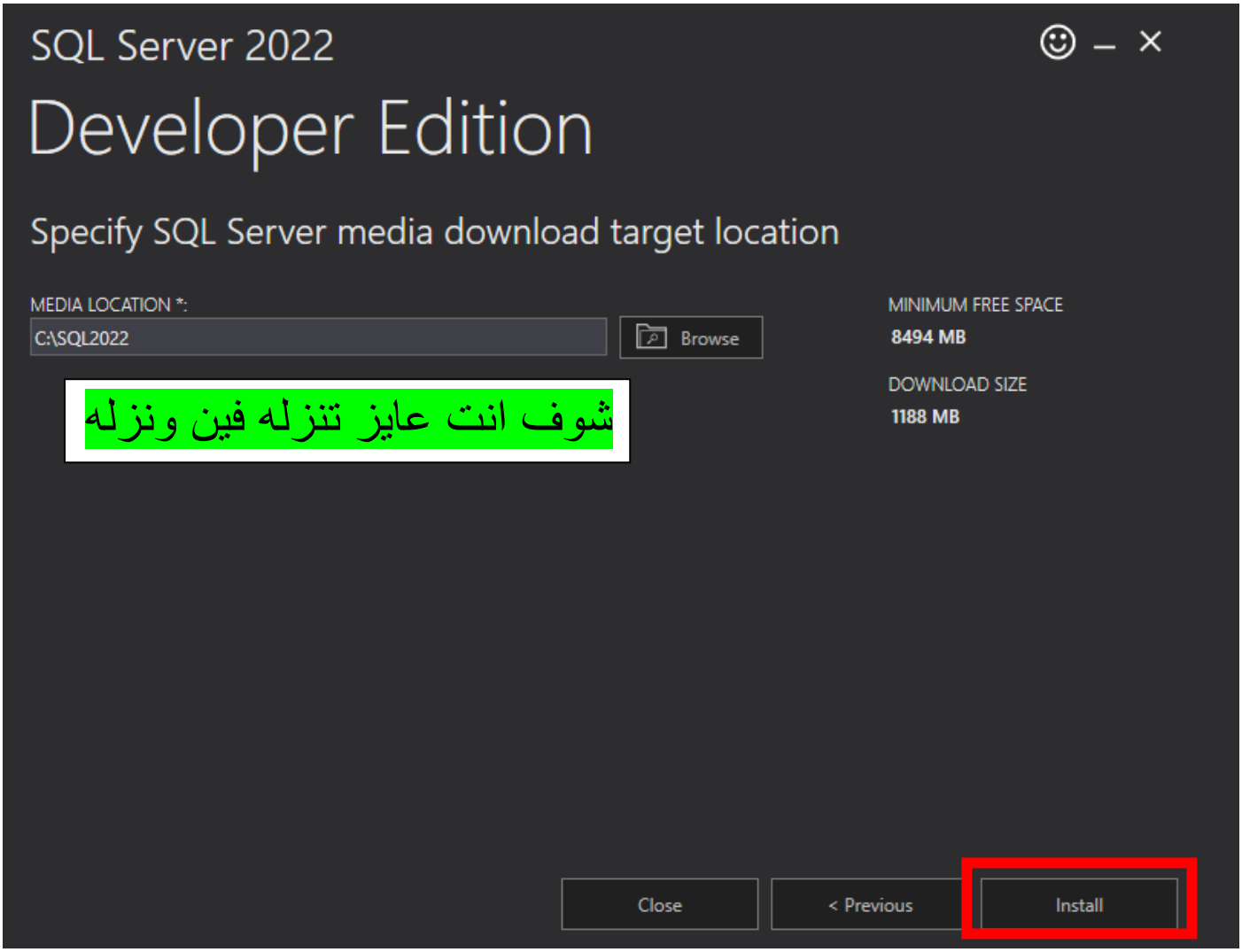
https://www.microsoft.com/en-us/sql-server/sql-server-downloads

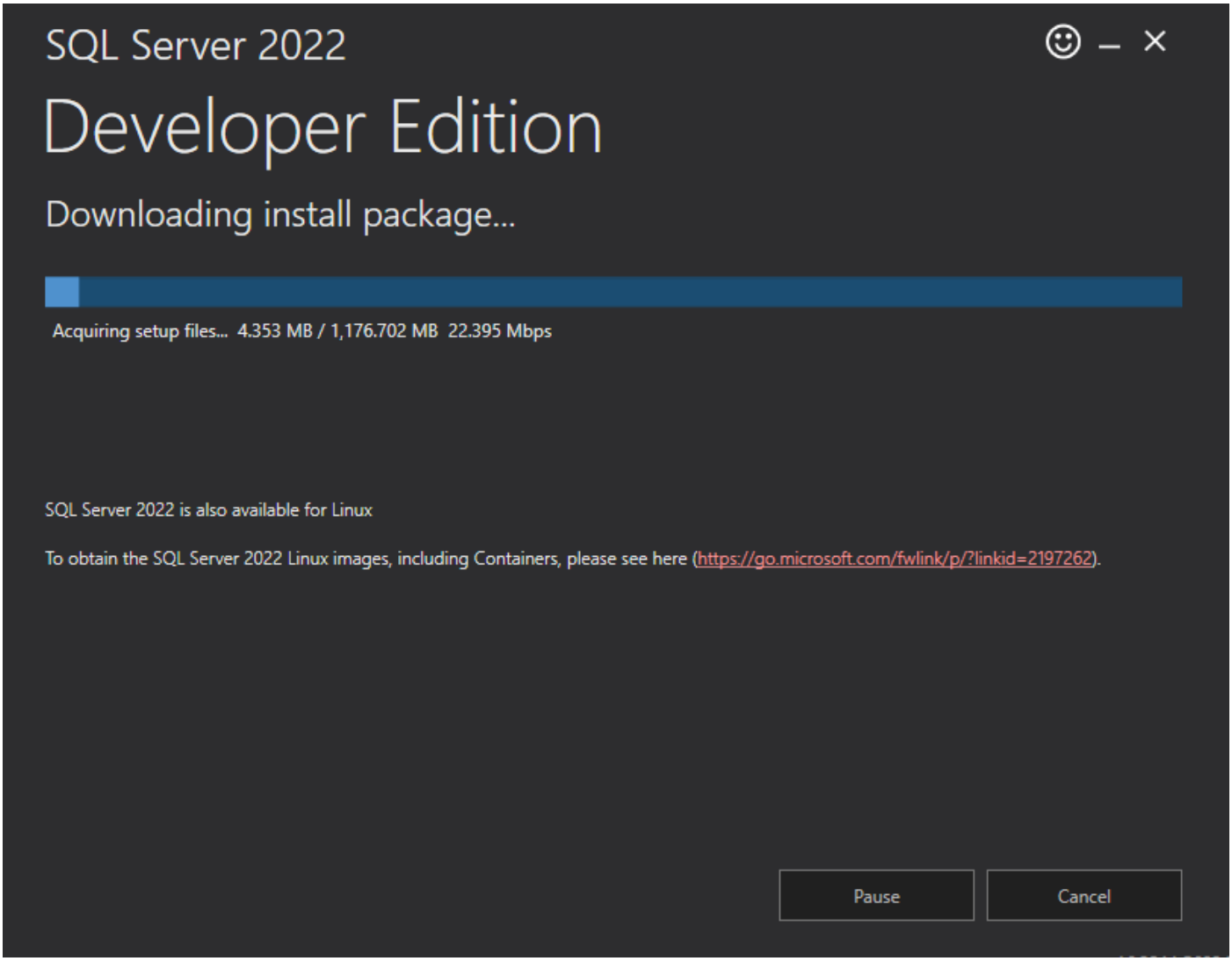


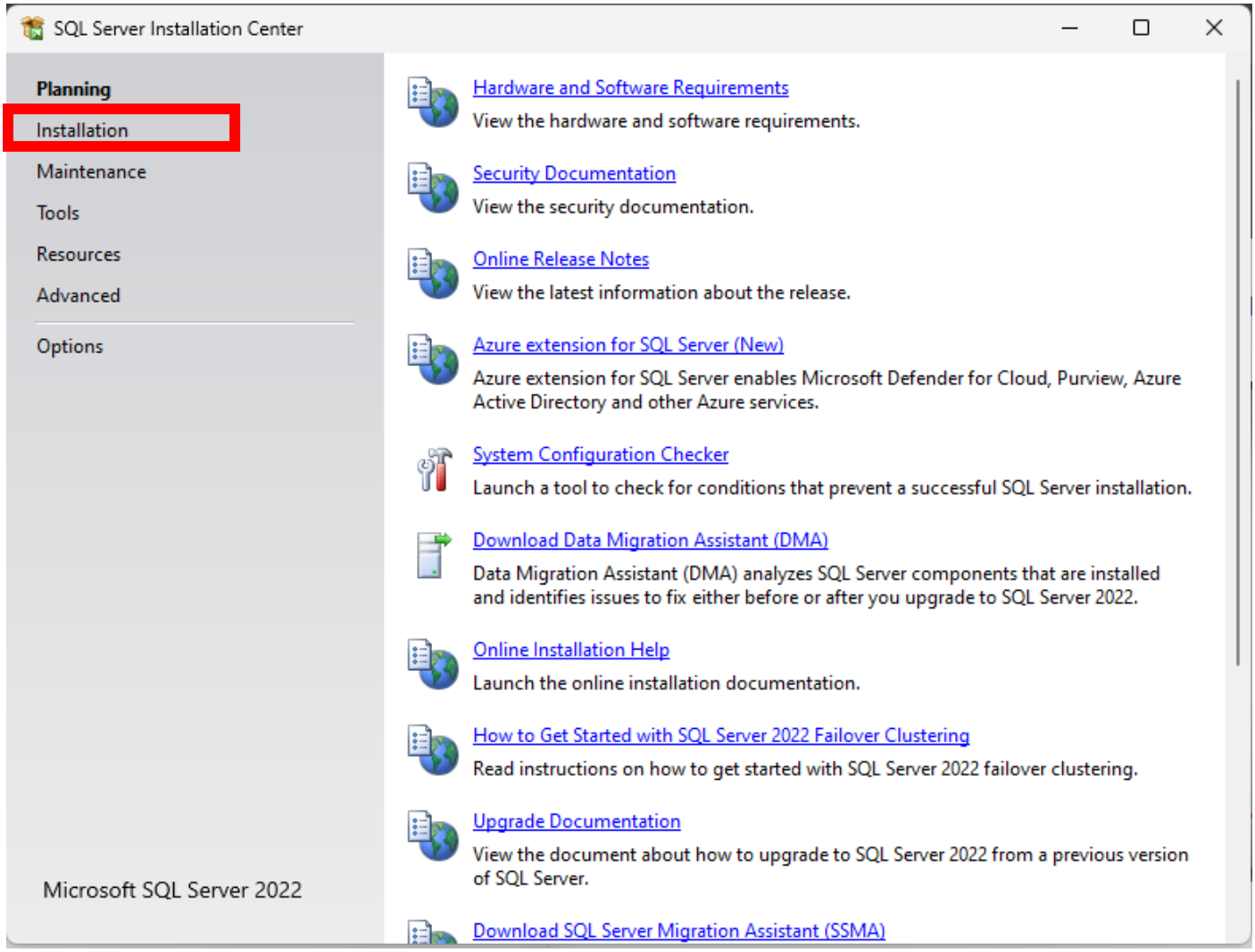
لازم تمشي معاه خطوة خطوه وماتعملش حاجه من نفسك عشان لو عملت حاجه غلط هتحصل مشاكل وهتضطر تعيده من الأول

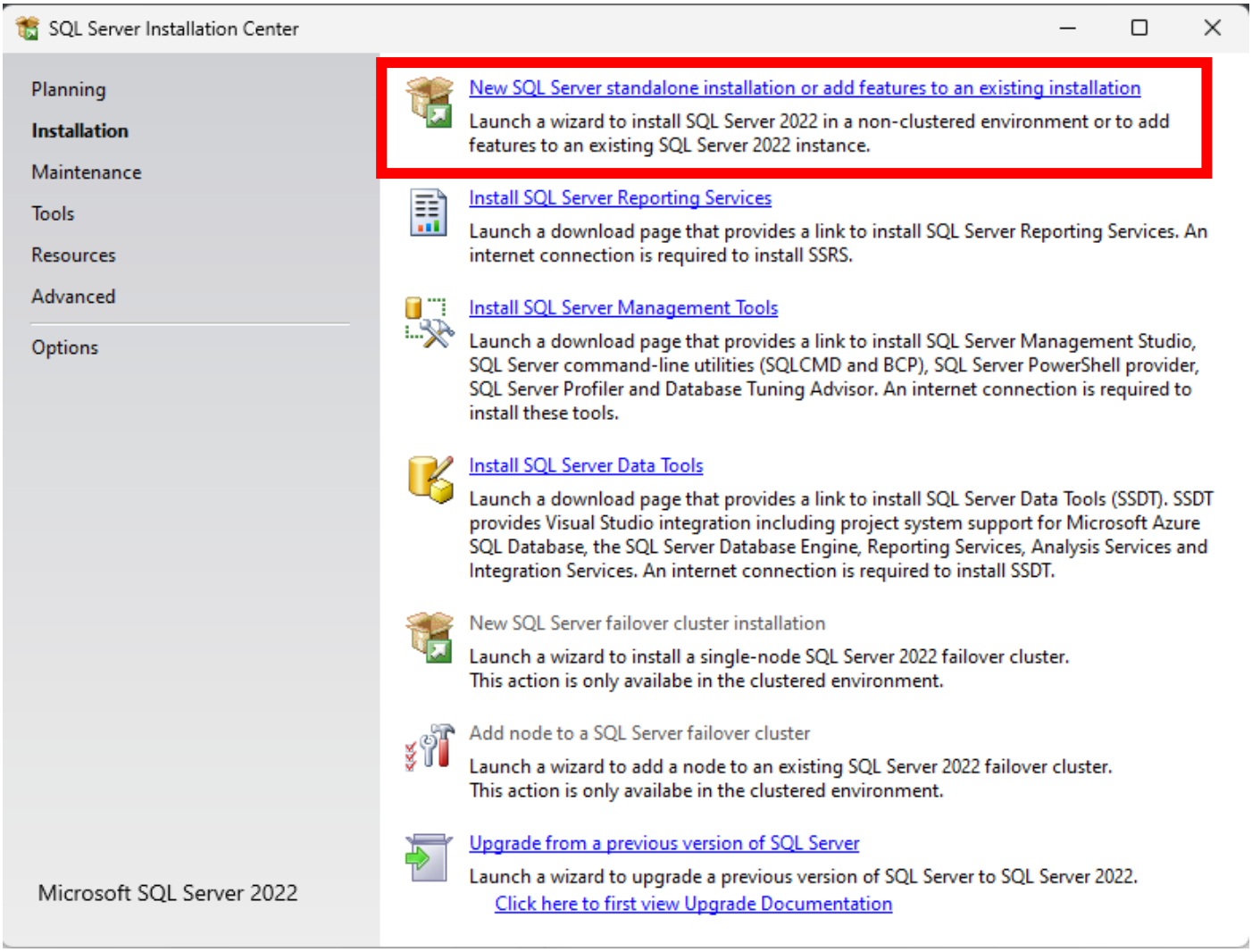


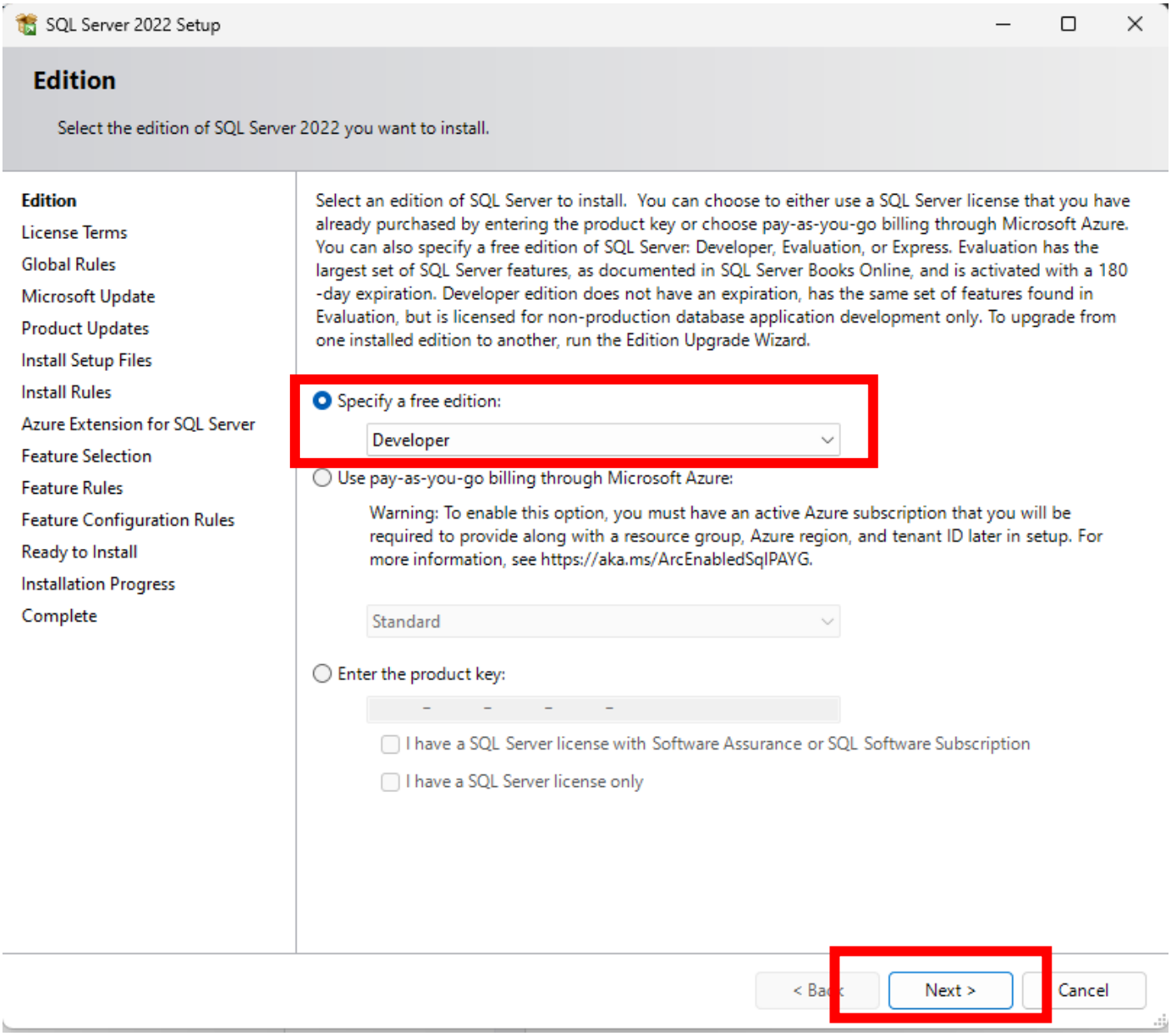

SQL Server 2022
Developer Edition
Select an installation type:
Basic
Select Basic installation type to install the SQL Server Database Engine feature with default configuration.
Custom
Select Custom installation type to step through the SQL Server installation wizard and choose what you want to install. 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
16.2211.5693.3

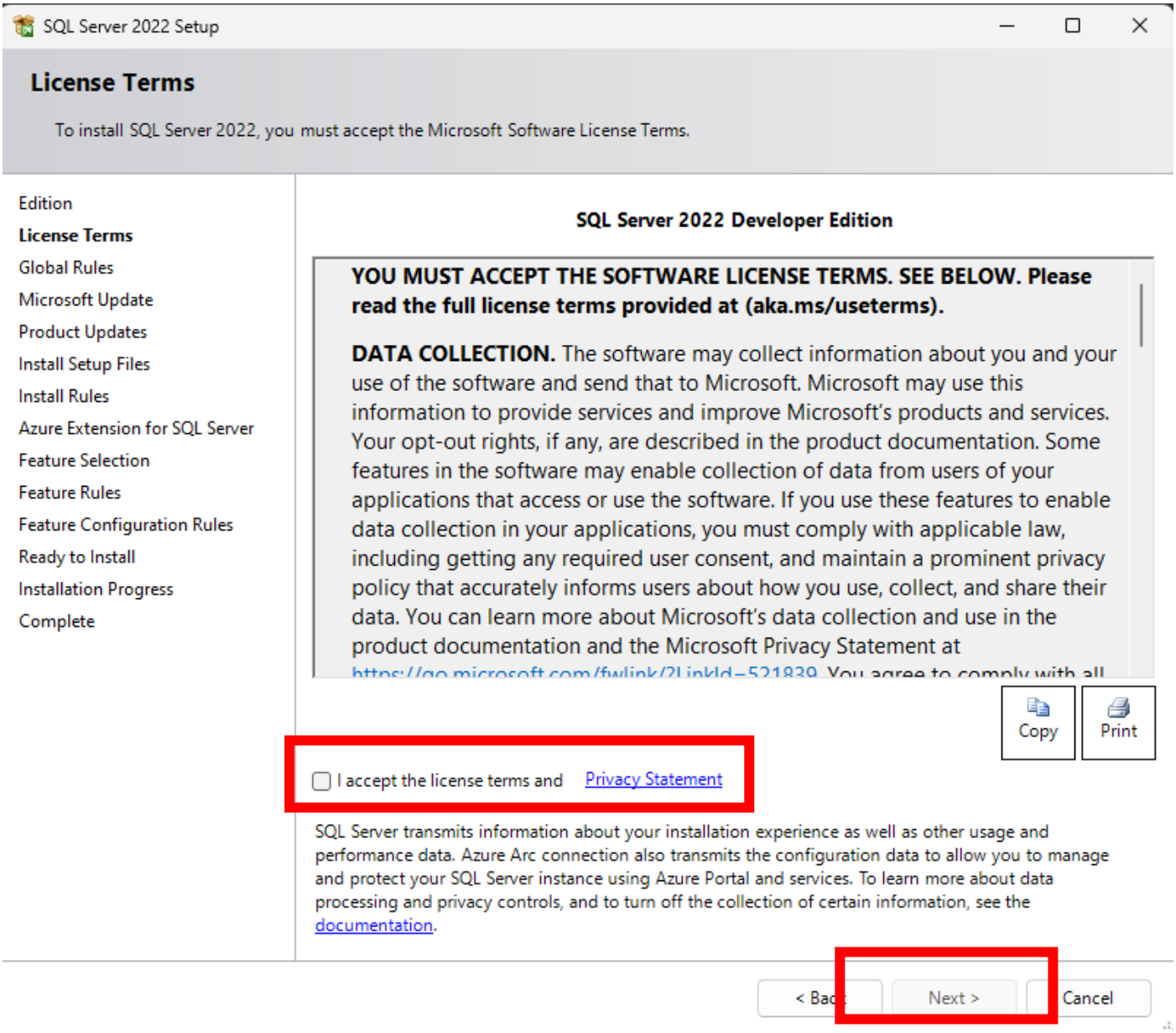

SQL Server 2022
Developer Edition
Specify SQL Server media download target location
MEDIA LOCATION *:
C:\SQL2022
Browse
MINIMUM FREE SPACE
8494 MB
DOWNLOAD SIZE
1188 MB
شوف انت عايز تنزله فين ونزله
Close
< Previous
Install

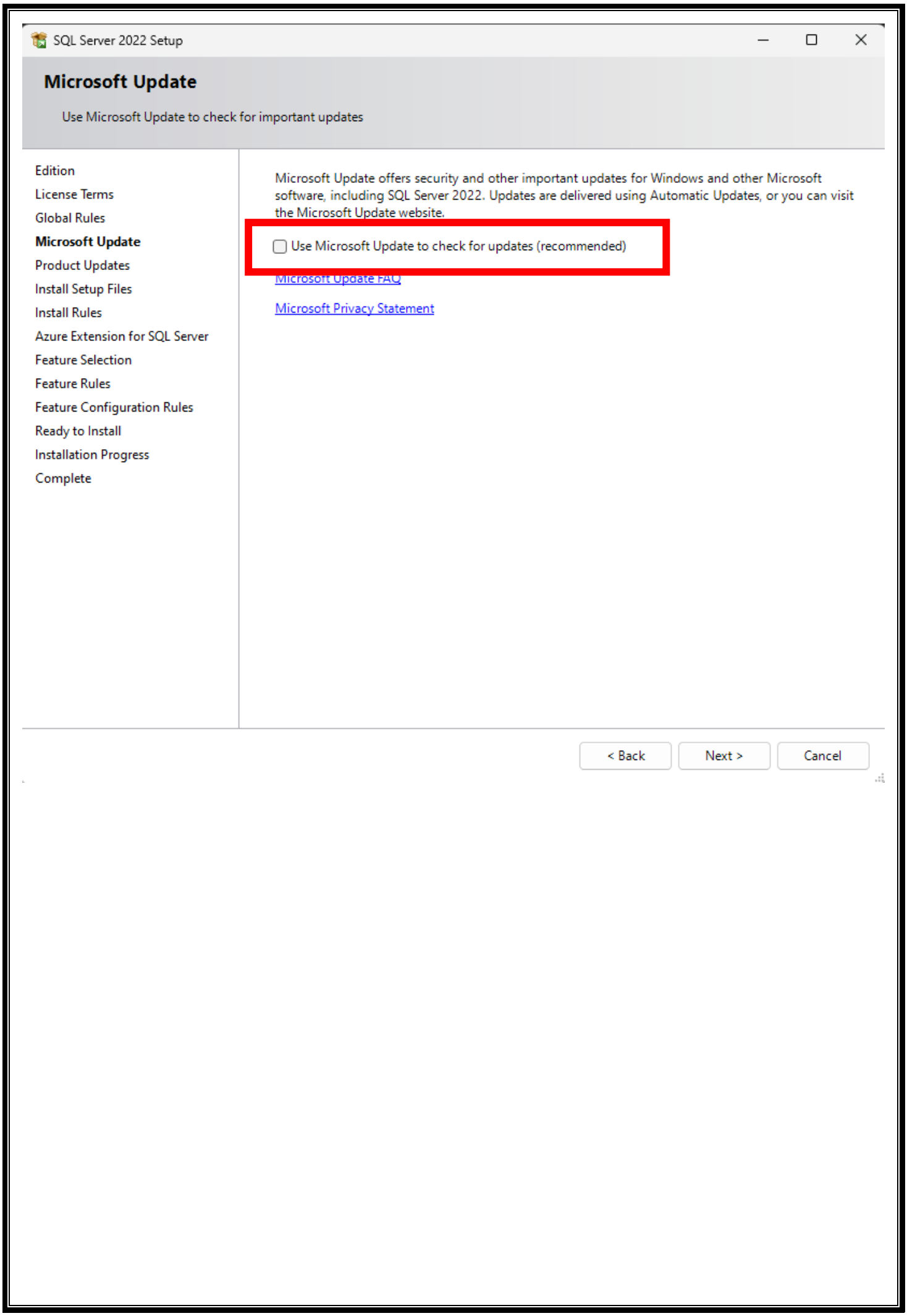

SQL Server 2022
Developer Edition
Downloading install package...
Acquiring setup files... 4.353 MB / 1,176.702 MB 22.395 Mbps
SQL Server 2022 is also available for Linux
To obtain the SQL Server 2022 Linux images, including Containers, please see here (https://go.microsoft.com/fwlink/p/?linkid=2197262).
Pause
Cancel

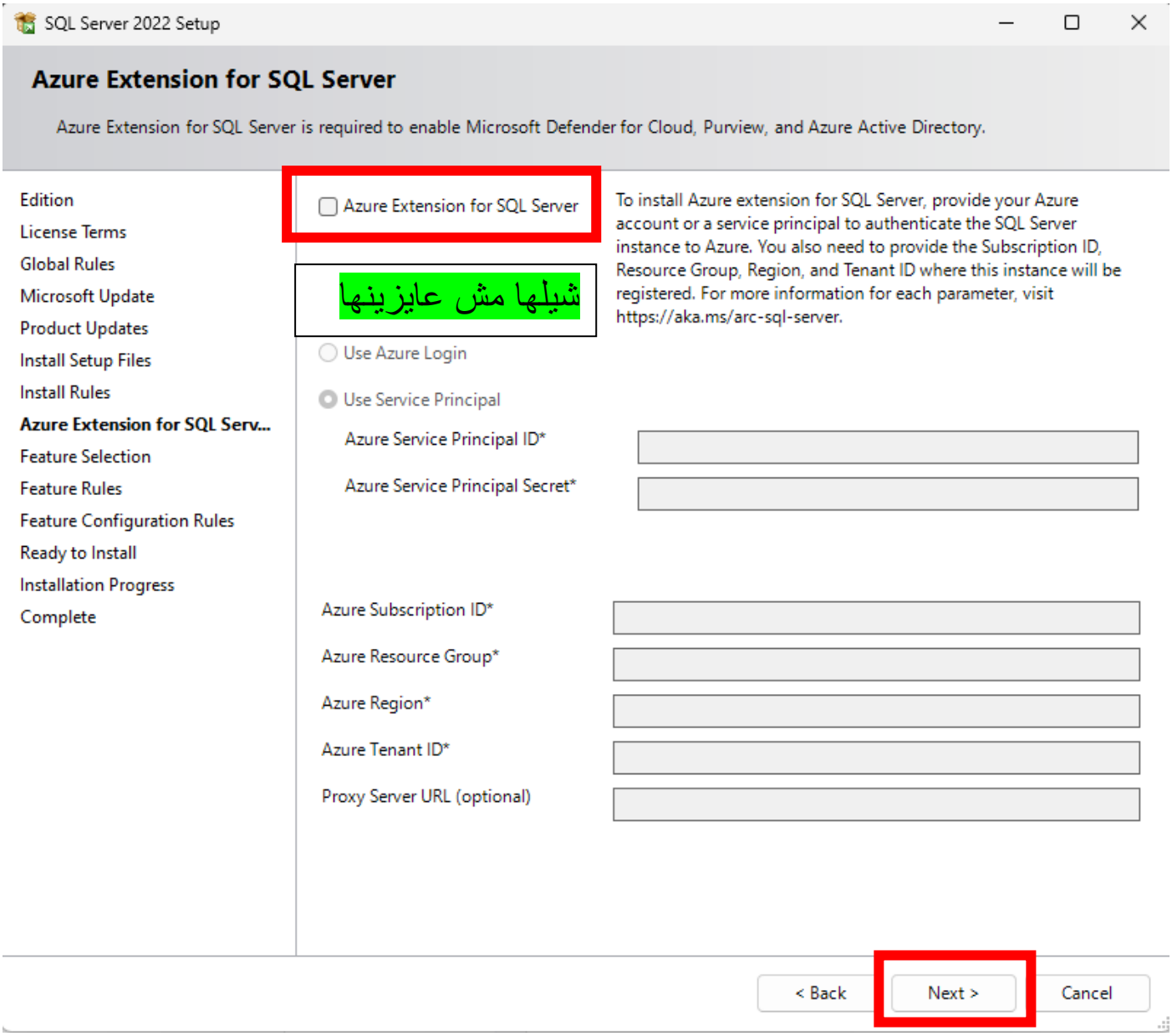

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
Microsoft SQL Server 2022
Hardware and Software Requirements
View the hardware and software requirements.
Security Documentation
View the security documentation.
Online Release Notes
View the latest information about the release.
Azure extension for SQL Server (New)
Azure extension for SQL Server enables Microsoft Defender for Cloud, Purview, Azure Active Directory and other Azure services.
System Configuration Checker
Launch a tool to check for conditions that prevent a successful SQL Server installation.
Download Data Migration Assistant (DMA)
Data Migration Assistant (DMA) analyzes SQL Server components that are installed and identifies issues to fix either before or after you upgrade to SQL Server 2022.
Online Installation Help
Launch the online installation documentation.
How to Get Started with SQL Server 2022 Failover Clustering
Read instructions on how to get started with SQL Server 2022 failover clustering.
Upgrade Documentation
View the document about how to upgrade to SQL Server 2022 from a previous version of SQL Server.
Download SQL Server Migration Assistant (SSMA)

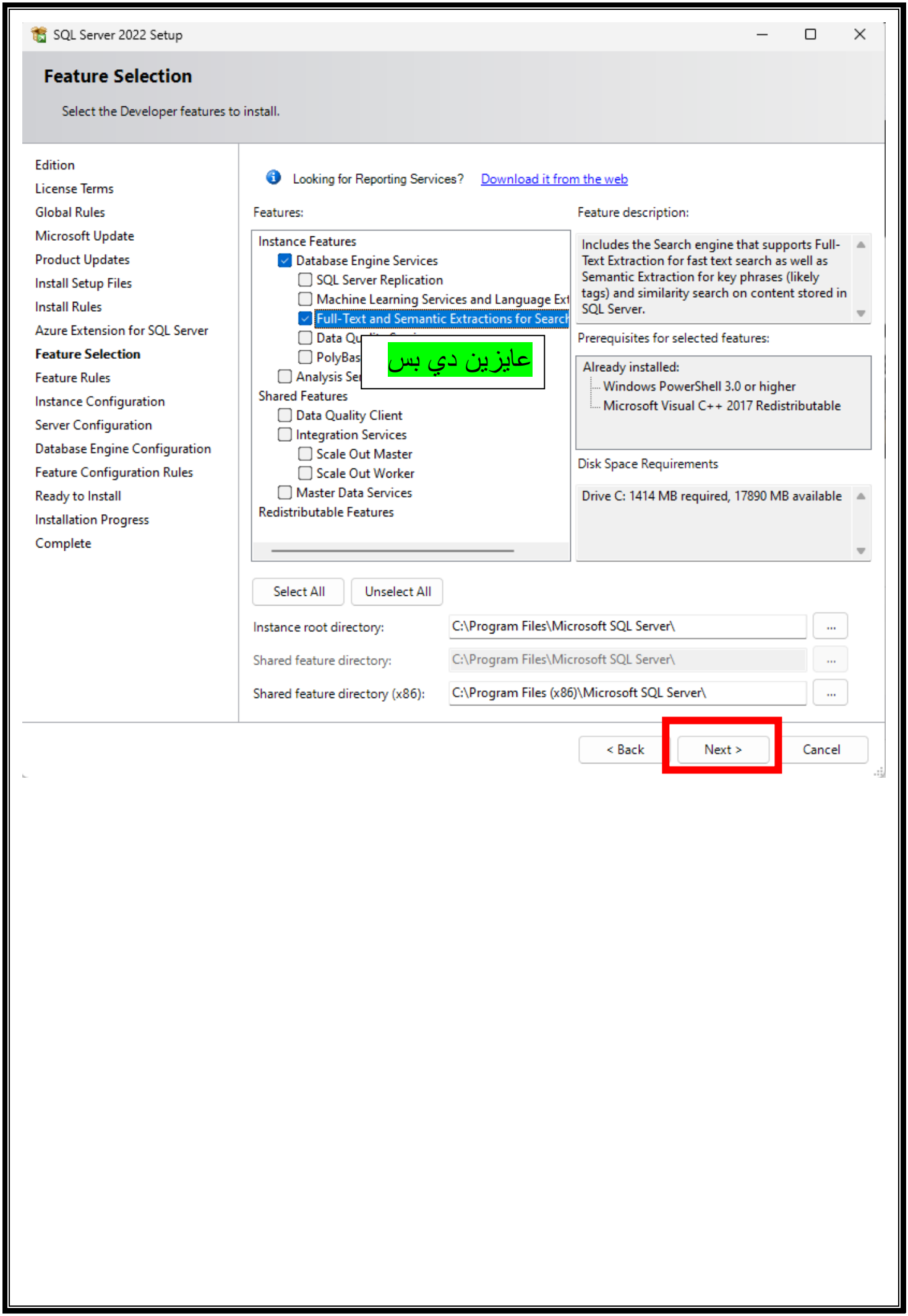

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
Microsoft SQL Server 2022
New SQL Server standalone installation or add features to an existing installation
Launch a wizard to install SQL Server 2022 in a non-clustered environment or to add features to an existing SQL Server 2022 instance.
Install SQL Server Reporting Services
Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.
Install SQL Server Management Tools
Launch a download page that provides a link to install SQL Server Management Studio, SQL Server command-line utilities (SQLCMD and BCP), SQL Server PowerShell provider, SQL Server Profiler and Database Tuning Advisor. An internet connection is required to install these tools.
Install SQL Server Data Tools
Launch a download page that provides a link to install SQL Server Data Tools (SSDT). SSDT provides Visual Studio integration including project system support for Microsoft Azure SQL Database, the SQL Server Database Engine, Reporting Services, Analysis Services and Integration Services. An internet connection is required to install SSDT.
New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2022 failover cluster.
This action is only availabe in the clustered environment.
Add node to a SQL Server failover cluster
Launch a wizard to add a node to an existing SQL Server 2022 failover cluster.
This action is only availabe in the clustered environment.
Upgrade from a previous version of SQL Server
Launch a wizard to upgrade a previous version of SQL Server to SQL Server 2022.
Click here to first view Upgrade Documentation

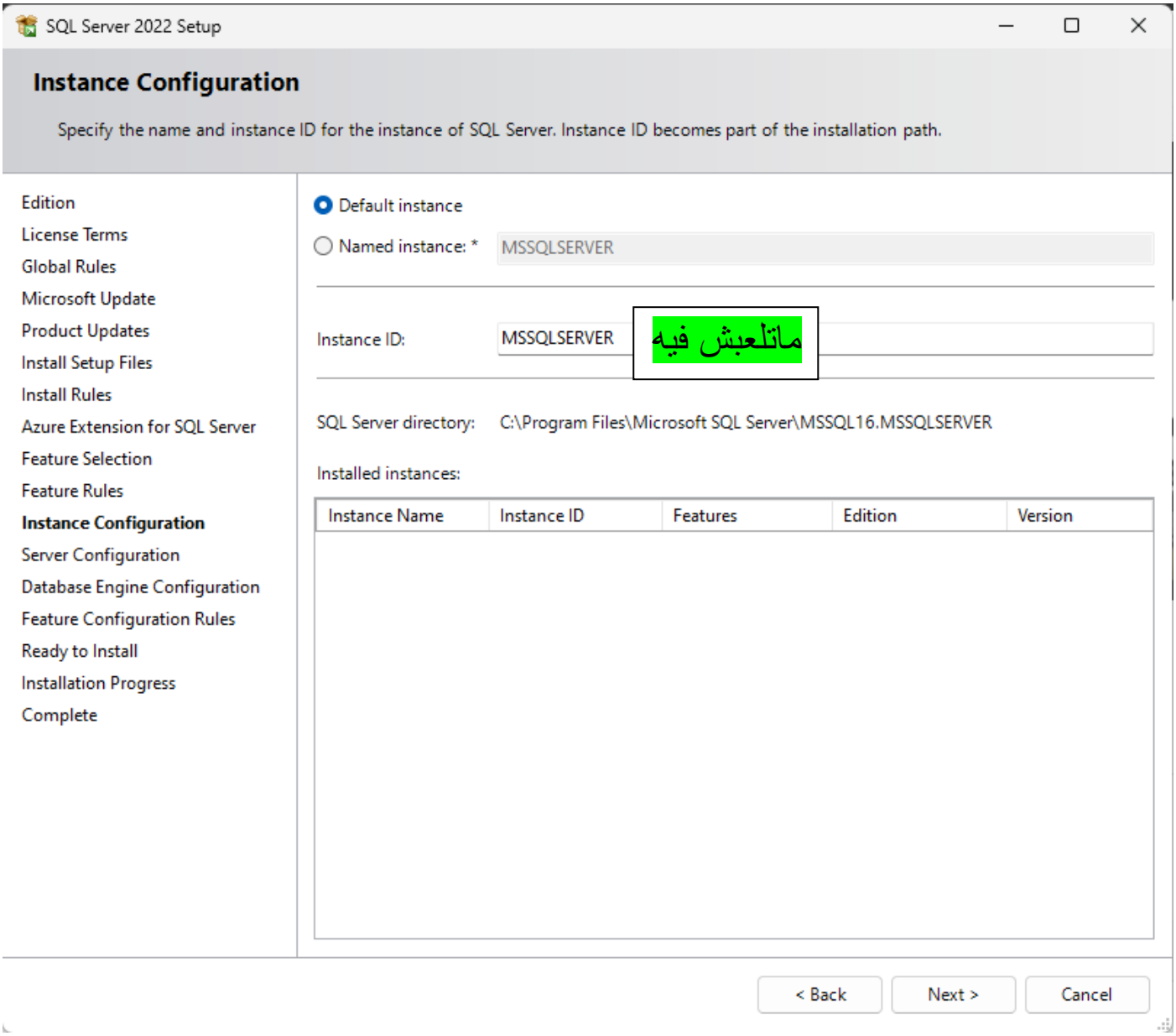

SQL Server 2022 Setup
Edition
Select the edition of SQL Server 2022 you want to install.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Select an edition of SQL Server to install. You can choose to either use a SQL Server license that you have already purchased by entering the product key or choose pay-as-you-go billing through Microsoft Azure. You can also specify a free edition of SQL Server: Developer, Evaluation, or Express. Evaluation has the largest set of SQL Server features, as documented in SQL Server Books Online, and is activated with a 180 -day expiration. Developer edition does not have an expiration, has the same set of features found in Evaluation, but is licensed for non-production database application development only. To upgrade from one installed edition to another, run the Edition Upgrade Wizard.
Specify a free edition:
Developer
Use pay-as-you-go billing through Microsoft Azure:
Warning: To enable this option, you must have an active Azure subscription that you will be required to provide along with a resource group, Azure region, and tenant ID later in setup. For more information, see https://aka.ms/ArcEnabledSqlPAYG.
Standard
Enter the product key:
I have a SQL Server license with Software Assurance or SQL Software Subscription
I have a SQL Server license only
< Back
Next >
Cancel

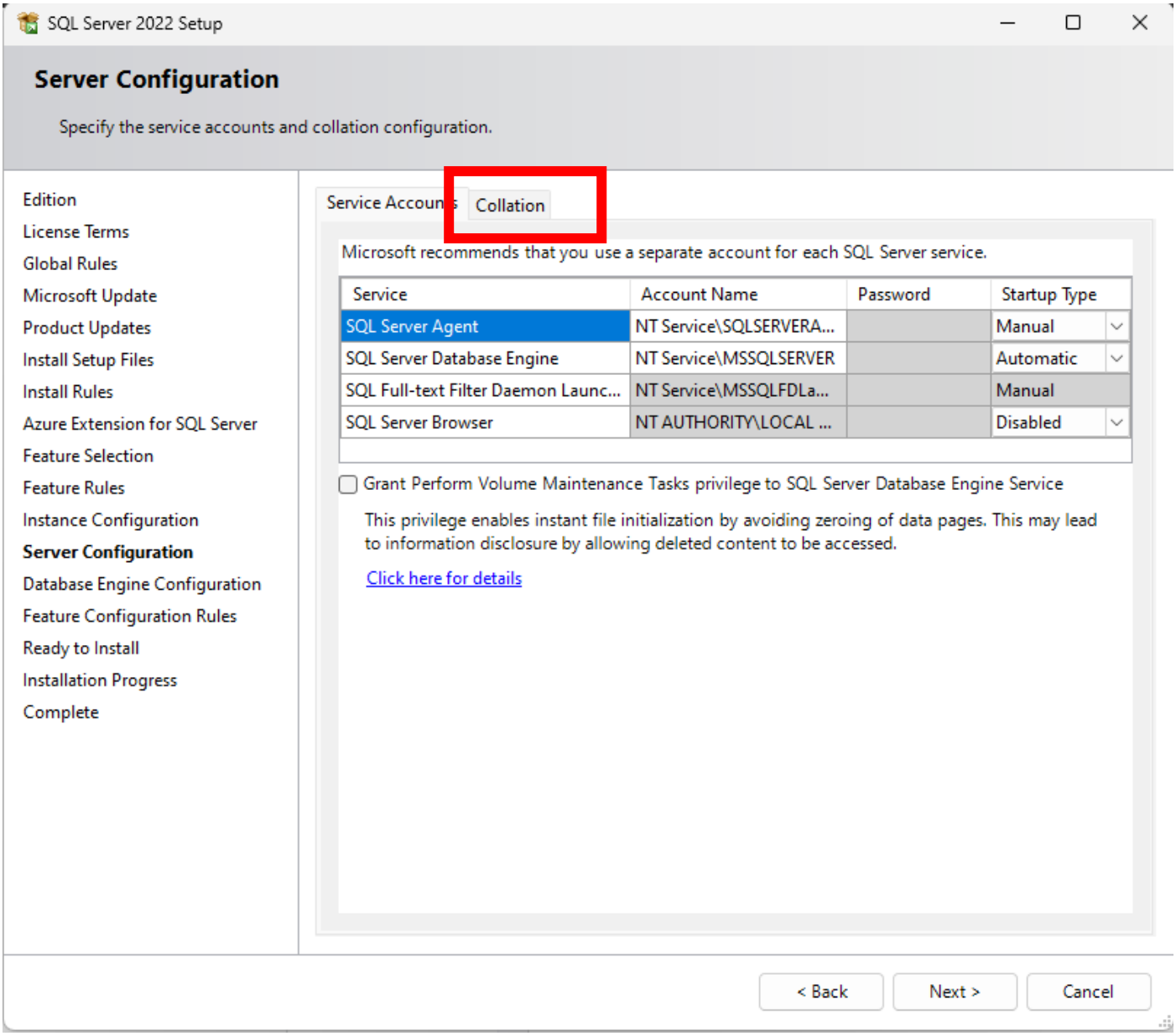

SQL Server 2022 Setup
License Terms
To install SQL Server 2022, you must accept the Microsoft Software License Terms.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
SQL Server 2022 Developer Edition
YOU MUST ACCEPT THE SOFTWARE LICENSE TERMS. SEE BELOW. Please read the full license terms provided at (aka.ms/useterms).
DATA COLLECTION. The software may collect information about you and your use of the software and send that to Microsoft. Microsoft may use this information to provide services and improve Microsoft's products and services. Your opt-out rights, if any, are described in the product documentation. Some features in the software may enable collection of data from users of your applications that access or use the software. If you use these features to enable data collection in your applications, you must comply with applicable law, including getting any required user consent, and maintain a prominent privacy policy that accurately informs users about how you use, collect, and share their data. You can learn more about Microsoft's data collection and use in the product documentation and the Microsoft Privacy Statement at
Copy
Print
I accept the license terms and
Privacy Statement
SQL Server transmits information about your installation experience as well as other usage and performance data. Azure Arc connection also transmits the configuration data to allow you to manage and protect your SQL Server instance using Azure Portal and services. To learn more about data processing and privacy controls, and to turn off the collection of certain information, see the
documentation.
< Back
Next >
Cancel

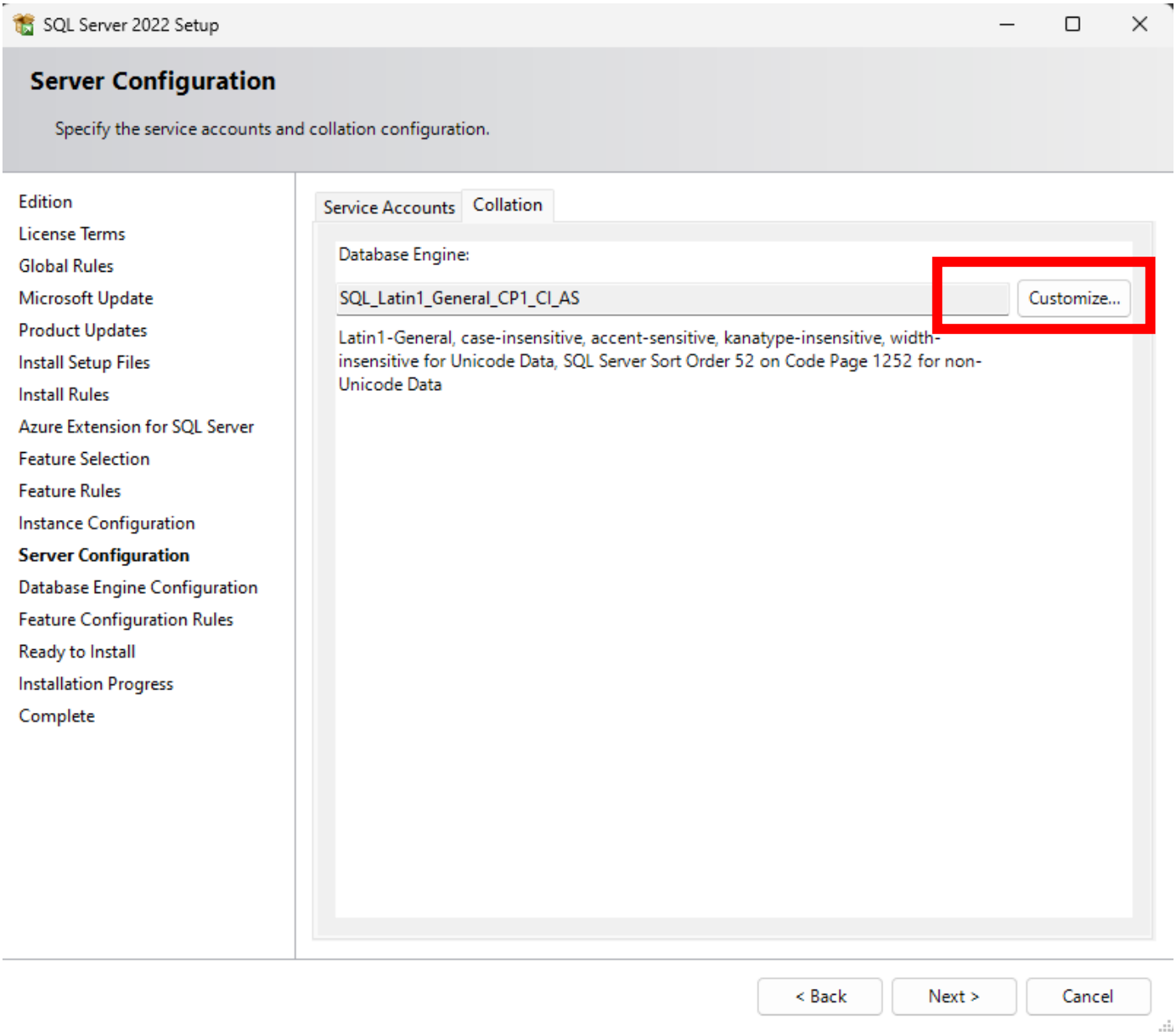

SQL Server 2022 Setup
Microsoft Update
Use Microsoft Update to check for important updates
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Microsoft Update offers security and other important updates for Windows and other Microsoft software, including SQL Server 2022. Updates are delivered using Automatic Updates, or you can visit the Microsoft Update website.
Use Microsoft Update to check for updates (recommended)
Microsoft Update FAQ
Microsoft Privacy Statement
< Back
Next >
Cancel

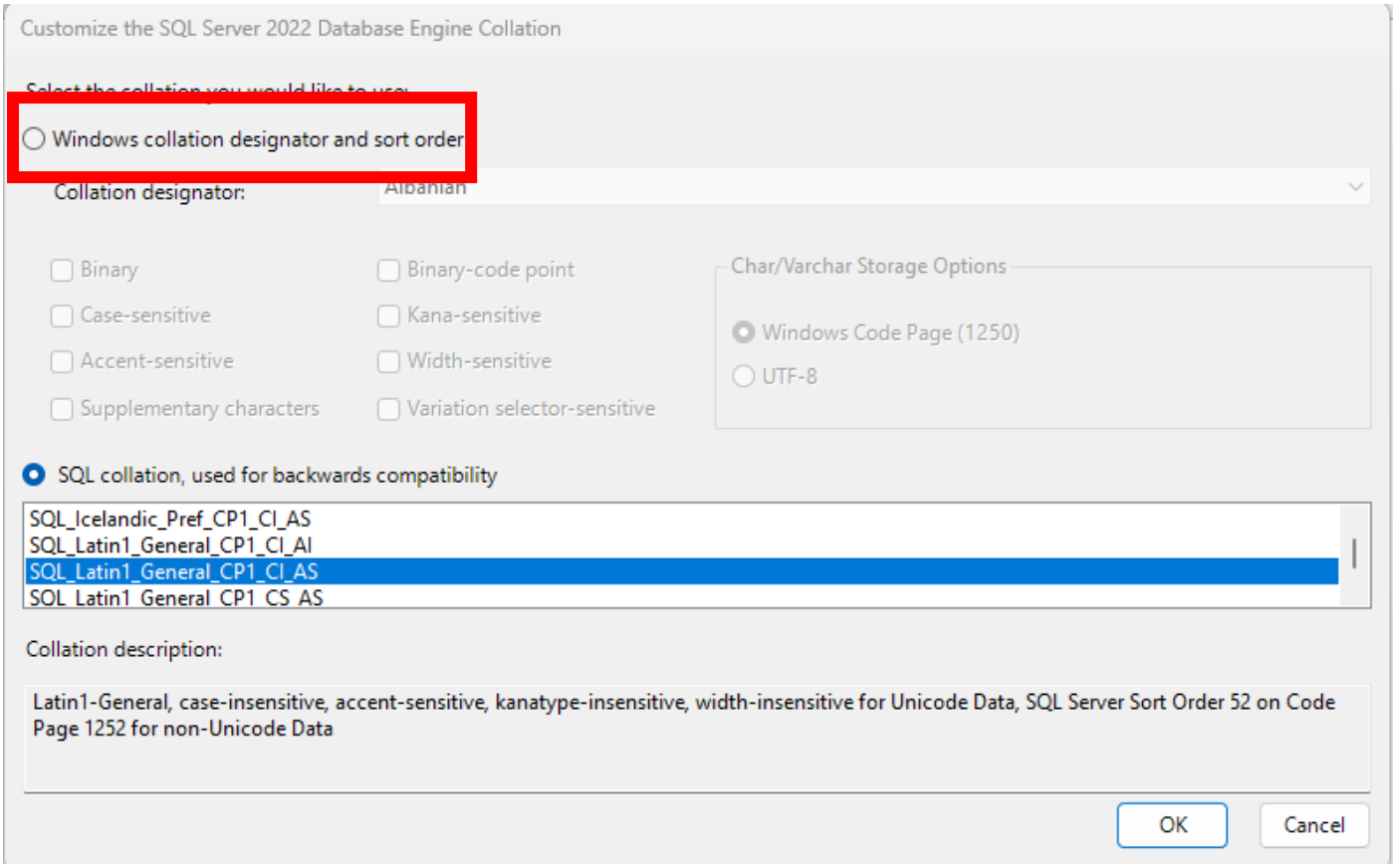

SQL Server 2022 Setup
Azure Extension for SQL Server
Azure Extension for SQL Server is required to enable Microsoft Defender for Cloud, Purview, and Azure Active Directory.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Serv...
Feature Selection
Feature Rules
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Azure Extension for SQL Server
شيلها مش عايزينها
To install Azure extension for SQL Server, provide your Azure account or a service principal to authenticate the SQL Server instance to Azure. You also need to provide the Subscription ID, Resource Group, Region, and Tenant ID where this instance will be registered. For more information for each parameter, visit https://aka.ms/arc-sql-server.
Use Azure Login
Use Service Principal
Azure Service Principal ID*
Azure Service Principal Secret*
Azure Subscription ID*
Azure Resource Group*
Azure Region*
Azure Tenant ID*
Proxy Server URL (optional)
< Back
Next >
Cancel

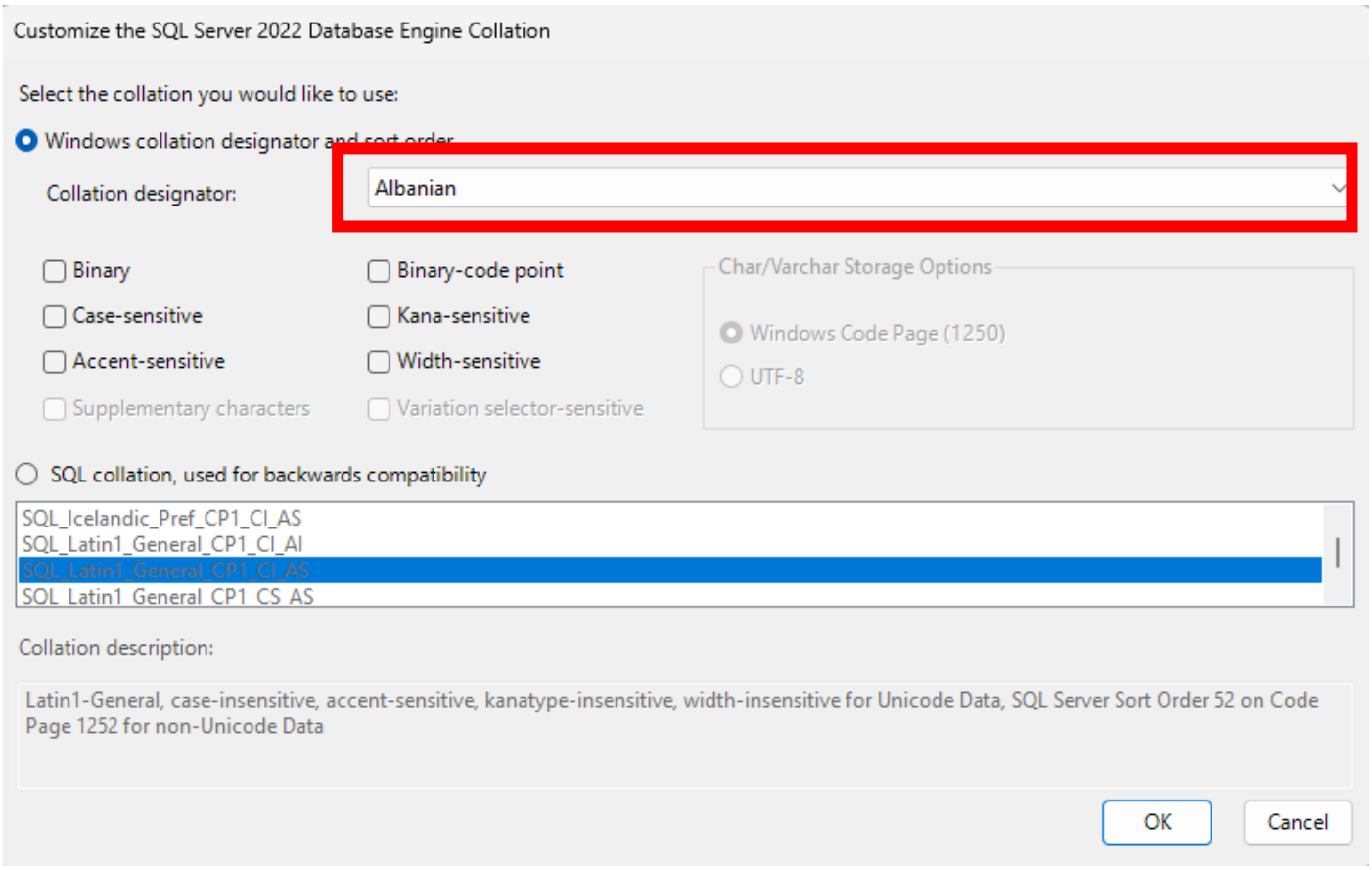

SQL Server 2022 Setup
Feature Selection
Select the Developer features to install.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Looking for Reporting Services?
Download it from the web
Features:
Instance Features
Database Engine Services
SQL Server Replication
Machine Learning Services and Language Ext
Full-Text and Semantic Extractions for Search
Data Qu
PolyBas
Analysis Se
عايزين دي بس
Shared Features
Data Quality Client
Integration Services
Scale Out Master
Scale Out Worker
Master Data Services
Redistributable Features
Feature description:
Includes the Search engine that supports Full-Text Extraction for fast text search as well as Semantic Extraction for key phrases (likely tags) and similarity search on content stored in SQL Server.
Prerequisites for selected features:
Already installed:
Windows PowerShell 3.0 or higher
Microsoft Visual C++ 2017 Redistributable
Disk Space Requirements
Drive C: 1414 MB required, 17890 MB available
Select All
Unselect All
Instance root directory:
C:\Program Files\Microsoft SQL Server\
Shared feature directory:
C:\Program Files\Microsoft SQL Server\
Shared feature directory (x86):
C:\Program Files (x86)\Microsoft SQL Server\
< Back
Next >
Cancel

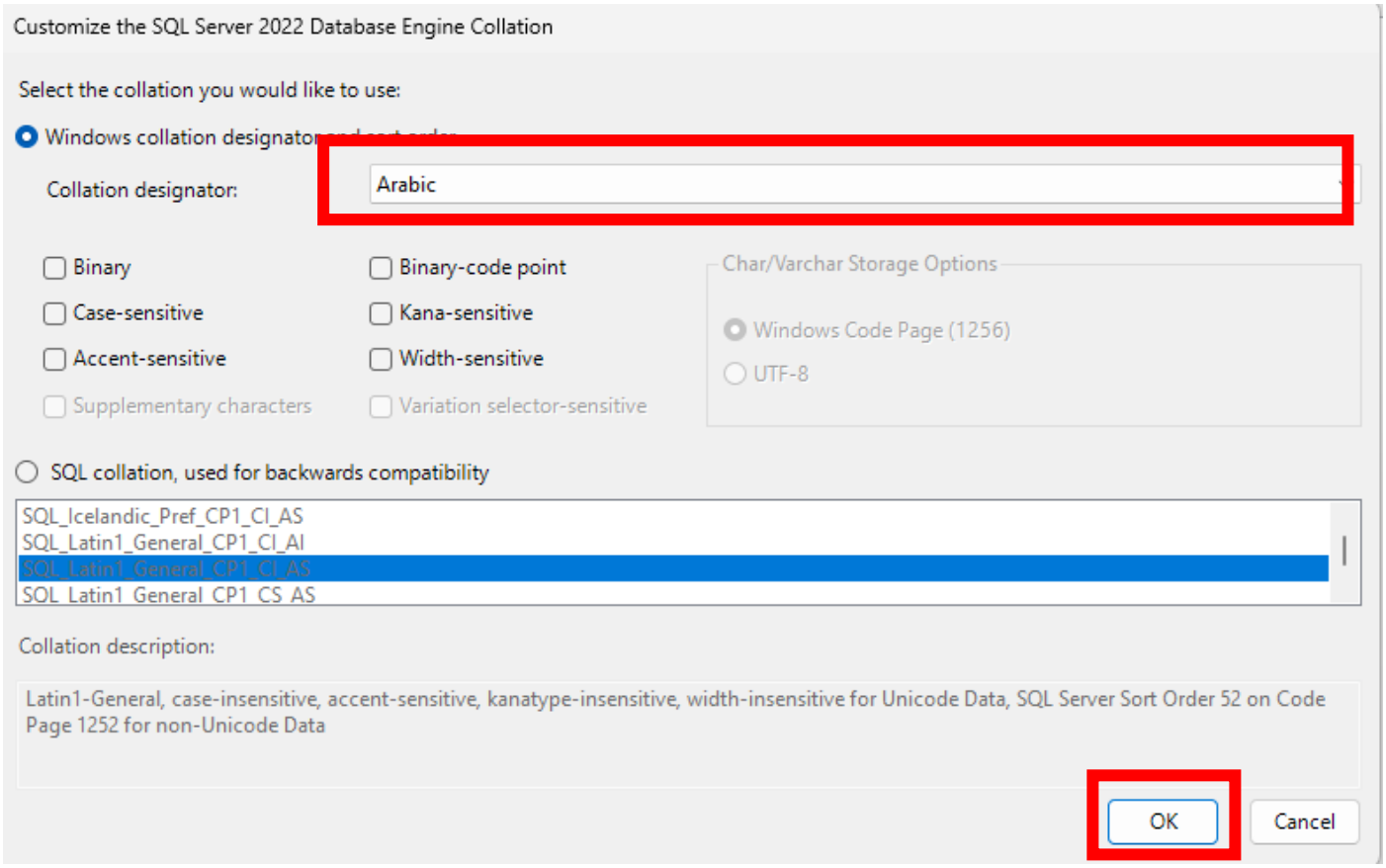

SQL Server 2022 Setup
Instance Configuration
Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Default instance
Named instance: *
MSSQLSERVER
Instance ID:
MSSQLSERVER
ماتلعبش فيه
SQL Server directory:
C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER
Installed instances:
Instance Name
Instance ID
Features
Edition
Version
< Back
Next >
Cancel

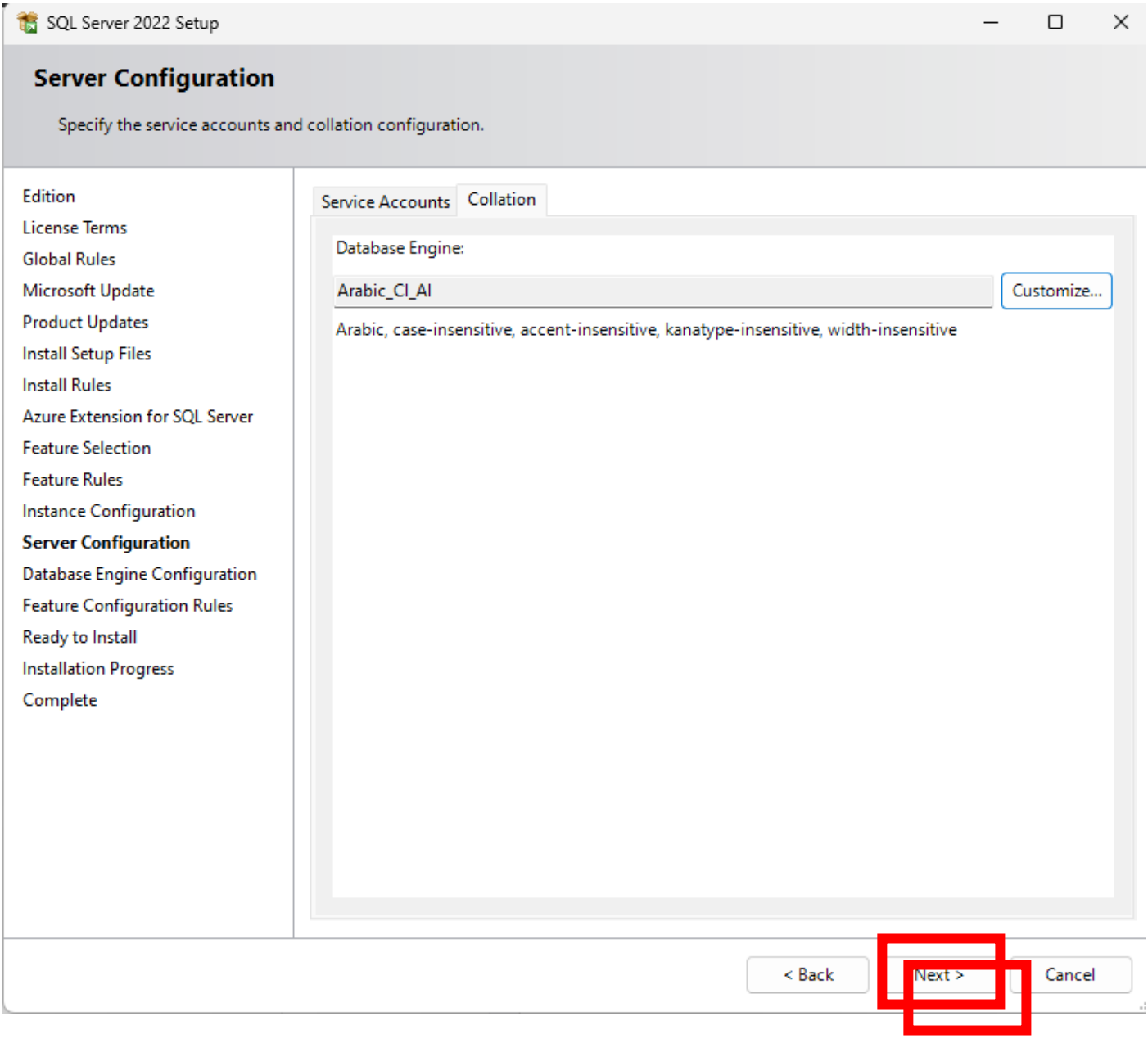

SQL Server 2022 Setup
Server Configuration
Specify the service accounts and collation configuration.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Service Accounts
Collation
Microsoft recommends that you use a separate account for each SQL Server service.
Service
Account Name
Password
Startup Type
SQL Server Agent
NT Service\SQLSERVERA...
Manual
SQL Server Database Engine
NT Service\MSSQLSERVER
Automatic
SQL Full-text Filter Daemon Launc...
NT Service\MSSQLFDLa...
Manual
SQL Server Browser
NT AUTHORITY\LOCAL ...
Disabled
Grant Perform Volume Maintenance Tasks privilege to SQL Server Database Engine Service
This privilege enables instant file initialization by avoiding zeroing of data pages. This may lead to information disclosure by allowing deleted content to be accessed.
Click here for details
< Back
Next >
Cancel

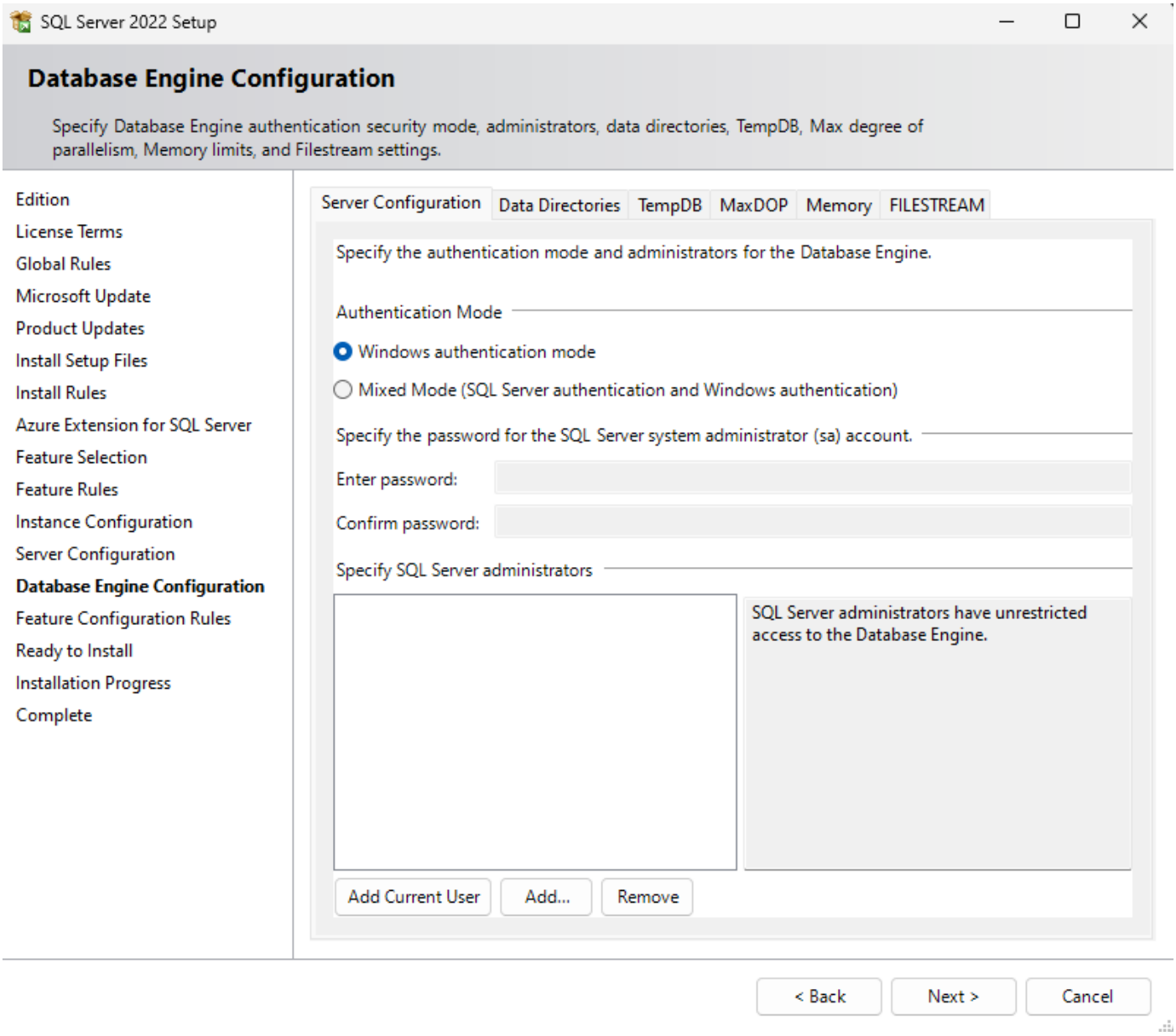

SQL Server 2022 Setup
Server Configuration
Specify the service accounts and collation configuration.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Service Accounts
Collation
Database Engine:
SQL_Latin1_General_CP1_CI_AS
Customize...
Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data
< Back
Next >
Cancel


Customize the SQL Server 2022 Database Engine Collation
Windows collation designator and sort order
Collation designator:
Albanian
Binary
Binary-code point
Case-sensitive
Kana-sensitive
Accent-sensitive
Width-sensitive
Supplementary characters
Variation selector-sensitive
Char/Varchar Storage Options
Windows Code Page (1250)
UTF-8
SQL collation, used for backwards compatibility
SQL_Icelandic_Pref_CP1_CI_AS
SQL_Latin1_General_CP1_CI_AI
SQL_Latin1_General_CP1_CI_AS
SQL_Latin1_General_CP1_CS_AS
Collation description:
Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data
OK
Cancel


Customize the SQL Server 2022 Database Engine Collation
Select the collation you would like to use:
Windows collation designator and sort order
Collation designator:
Albanian
Binary
Binary-code point
Case-sensitive
Kana-sensitive
Accent-sensitive
Width-sensitive
Supplementary characters
Variation selector-sensitive
Char/Varchar Storage Options
Windows Code Page (1250)
UTF-8
SQL collation, used for backwards compatibility
SQL_Icelandic_Pref_CP1_CI_AS
SQL_Latin1_General_CP1_CI_AI
SQL_Latin1_General_CP1_CI_AS
SQL_Latin1_General_CP1_CS_AS
Collation description:
Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data
OK
Cancel


Customize the SQL Server 2022 Database Engine Collation
Select the collation you would like to use:
Windows collation designator and sort order
Collation designator:
Arabic
Binary
Binary-code point
Case-sensitive
Kana-sensitive
Accent-sensitive
Width-sensitive
Supplementary characters
Variation selector-sensitive
Char/Varchar Storage Options
Windows Code Page (1256)
UTF-8
SQL collation, used for backwards compatibility
SQL_Icelandic_Pref_CP1_CI_AS
SQL_Latin1_General_CP1_CI_AI
SQL_Latin1_General_CP1_CI_AS
SQL_Latin1_General_CP1_CS_AS
Collation description:
Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data
OK
Cancel


SQL Server 2022 Setup
Server Configuration
Specify the service accounts and collation configuration.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Service Accounts
Collation
Database Engine:
Arabic_CI_AI
Customize...
Arabic, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
< Back
Next >
Cancel



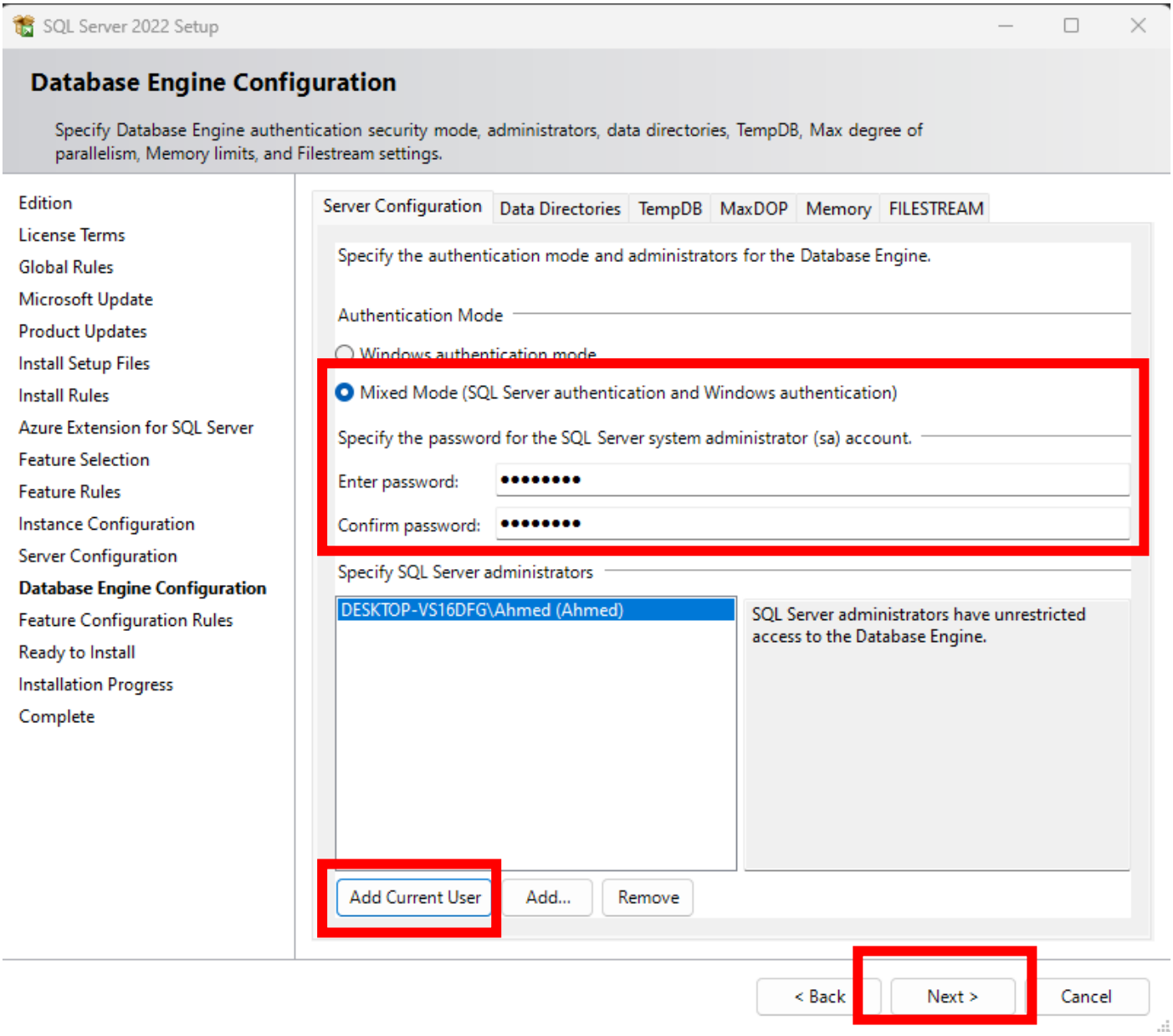
هوا دلوقتي هينزلك ال sql server من البرنامج بتاعك ففيه طريقتين عشان يشتغل اول واحده عن طريق اليوزر والباسورد بتوع الويندوز ودي اسمها windows authentication

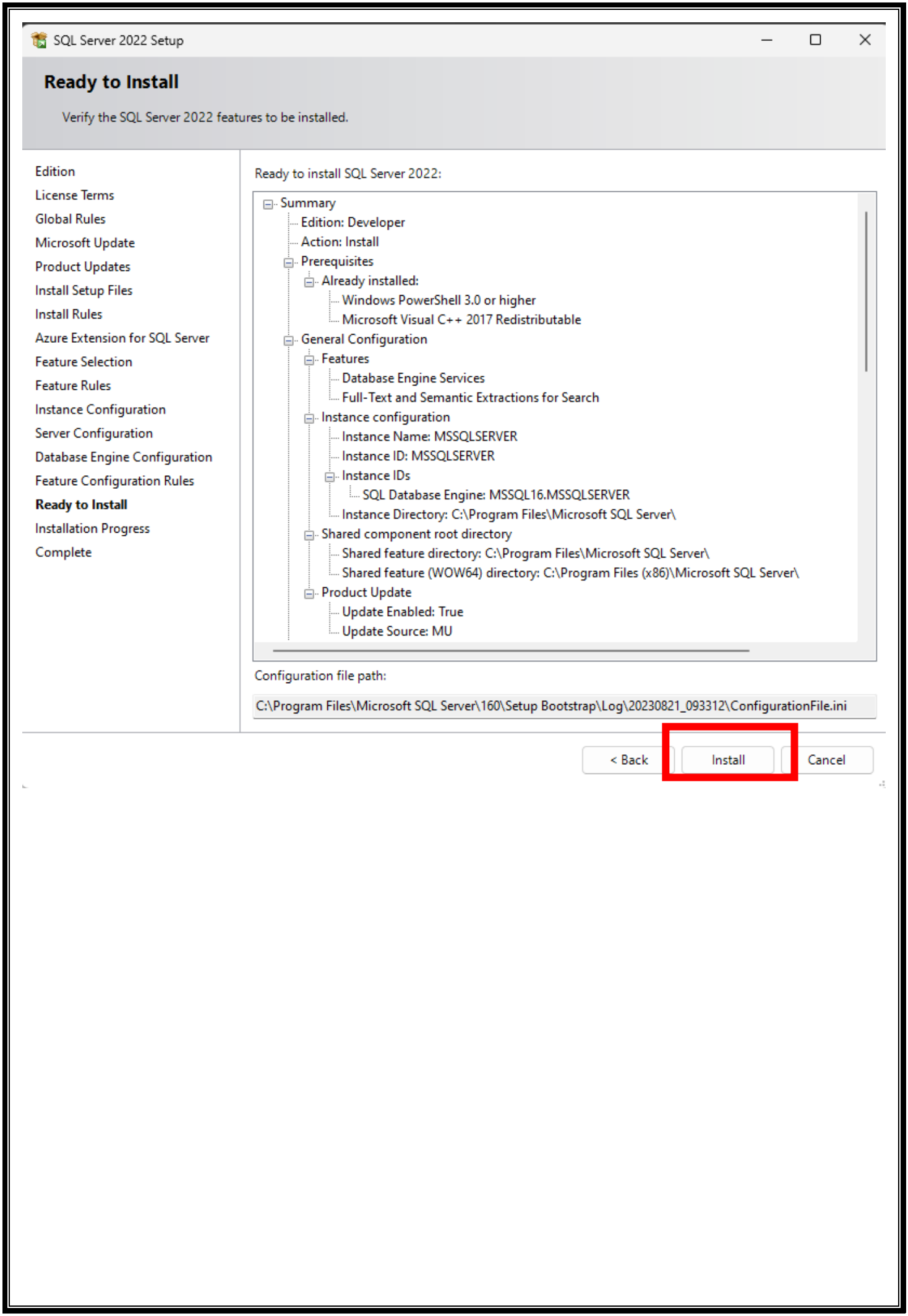
والطريقه التانيه عن طريق يوزر وباسوورد خاصين بيك

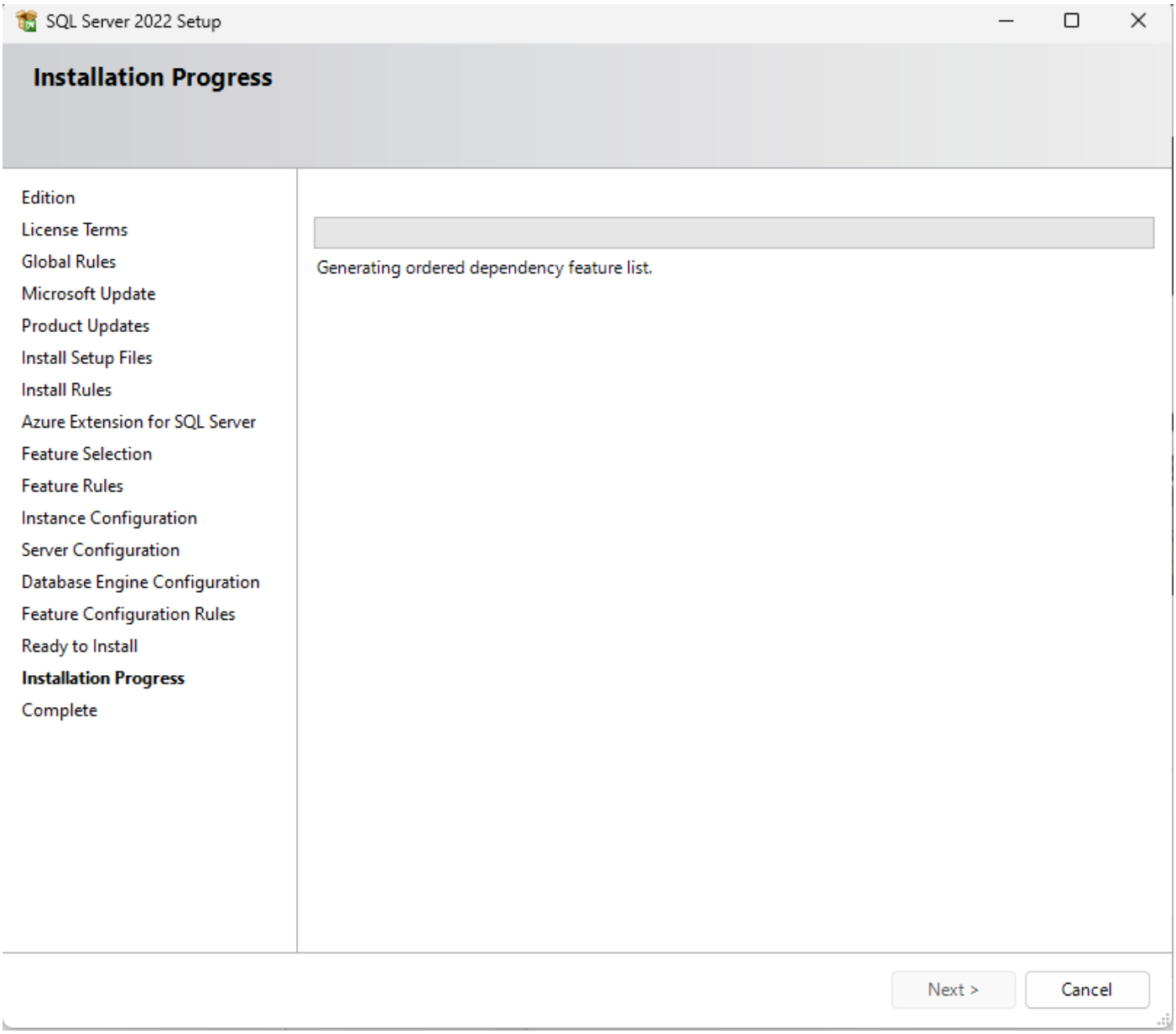
فااحنا هنختار mixed mode عشان يدعم الاتنين

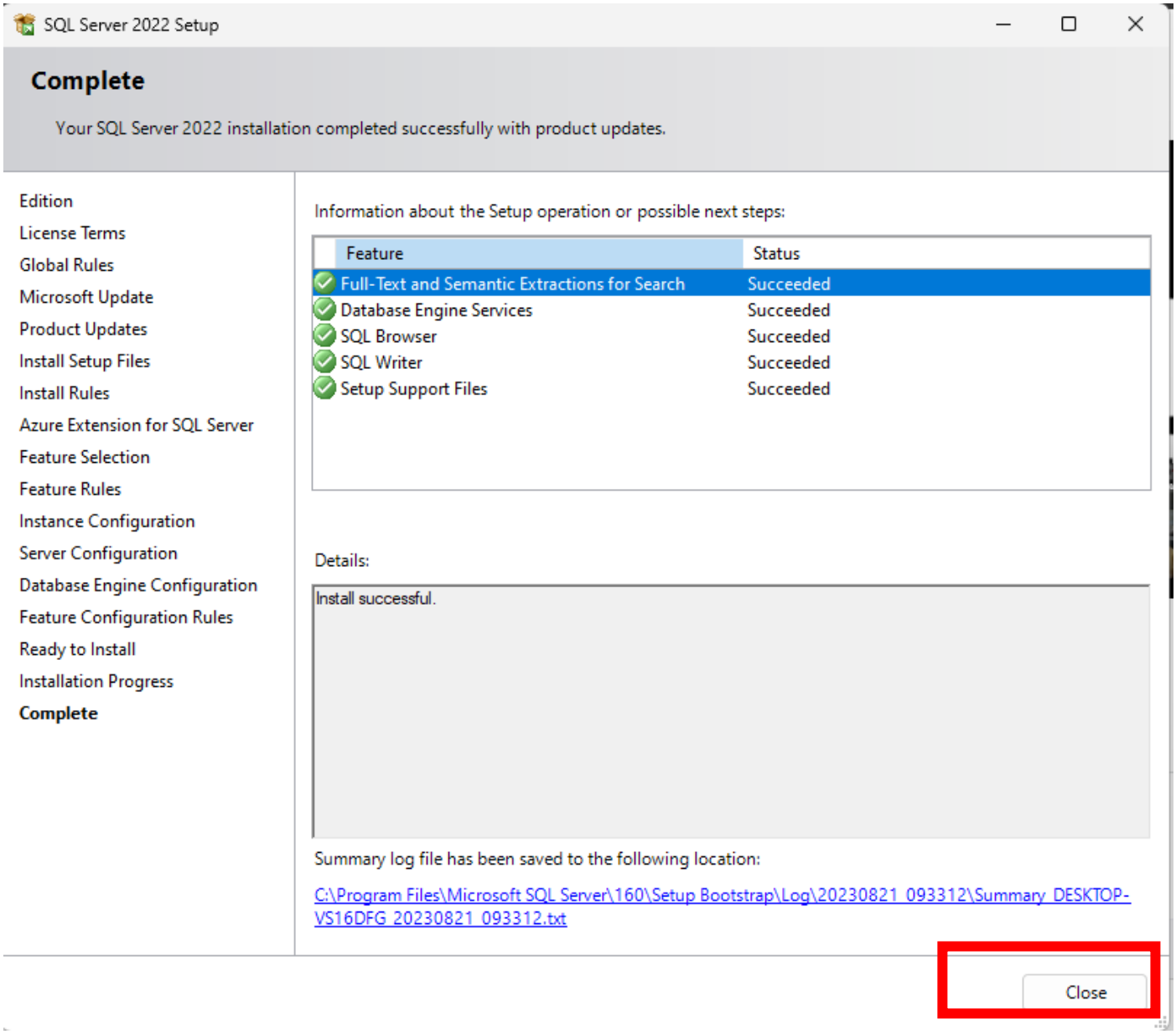
بعدين في الباسورد خليه sa123456

وبعدين بتدوس علي add current user واصبر عليه هياخد وقت

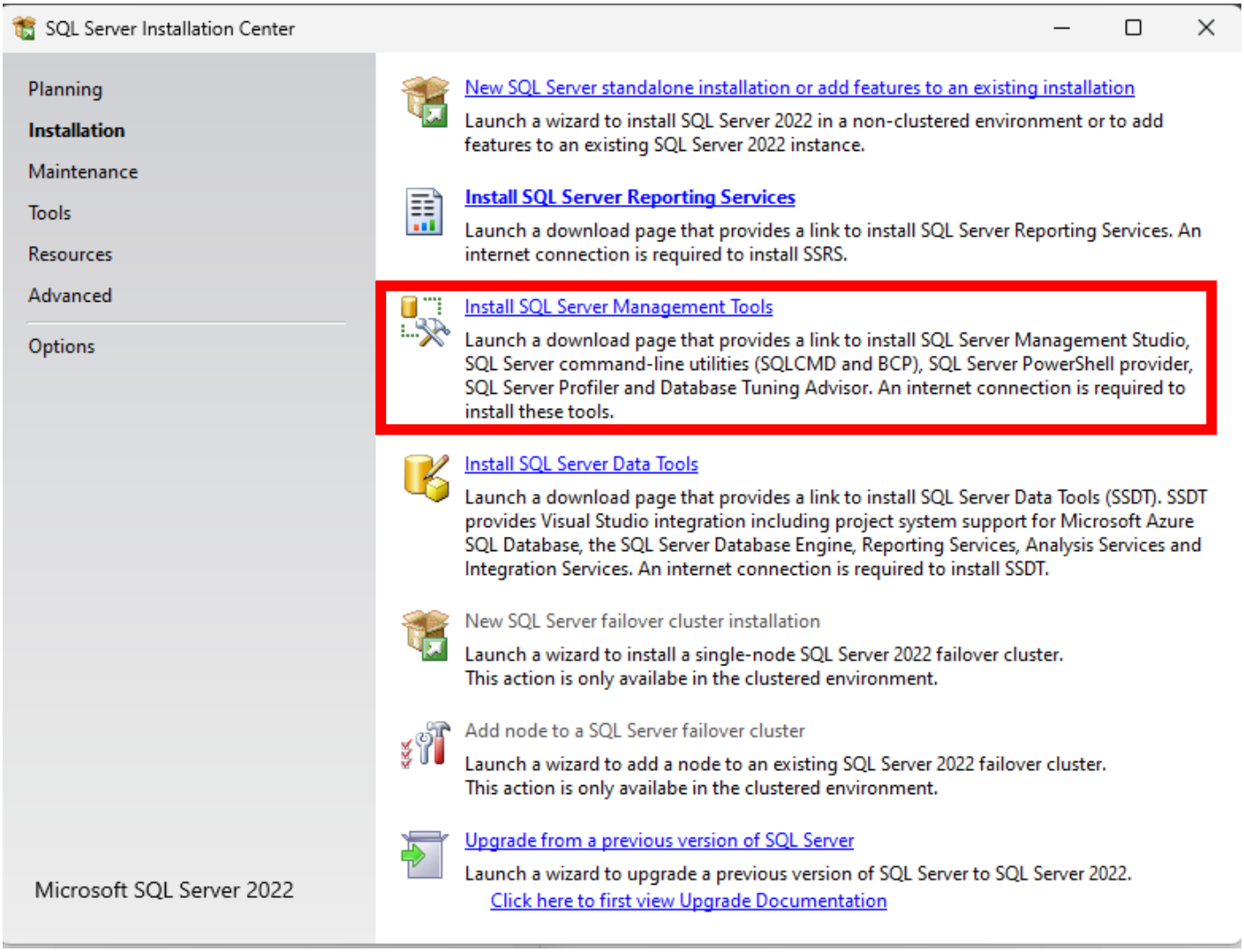

SQL Server 2022 Setup
Database Engine Configuration
Specify Database Engine authentication security mode, administrators, data directories, TempDB, Max degree of parallelism, Memory limits, and Filestream settings.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Server Configuration
Data Directories
TempDB
MaxDOP
Memory
FILESTREAM
Specify the authentication mode and administrators for the Database Engine.
Authentication Mode
Windows authentication mode
Mixed Mode (SQL Server authentication and Windows authentication)
Specify the password for the SQL Server system administrator (sa) account.
Enter password:
Confirm password:
Specify SQL Server administrators
DESKTOP-VS16DFG\Ahmed (Ahmed)
SQL Server administrators have unrestricted access to the Database Engine.
Add Current User
Add...
Remove
< Back
Next >
Cancel

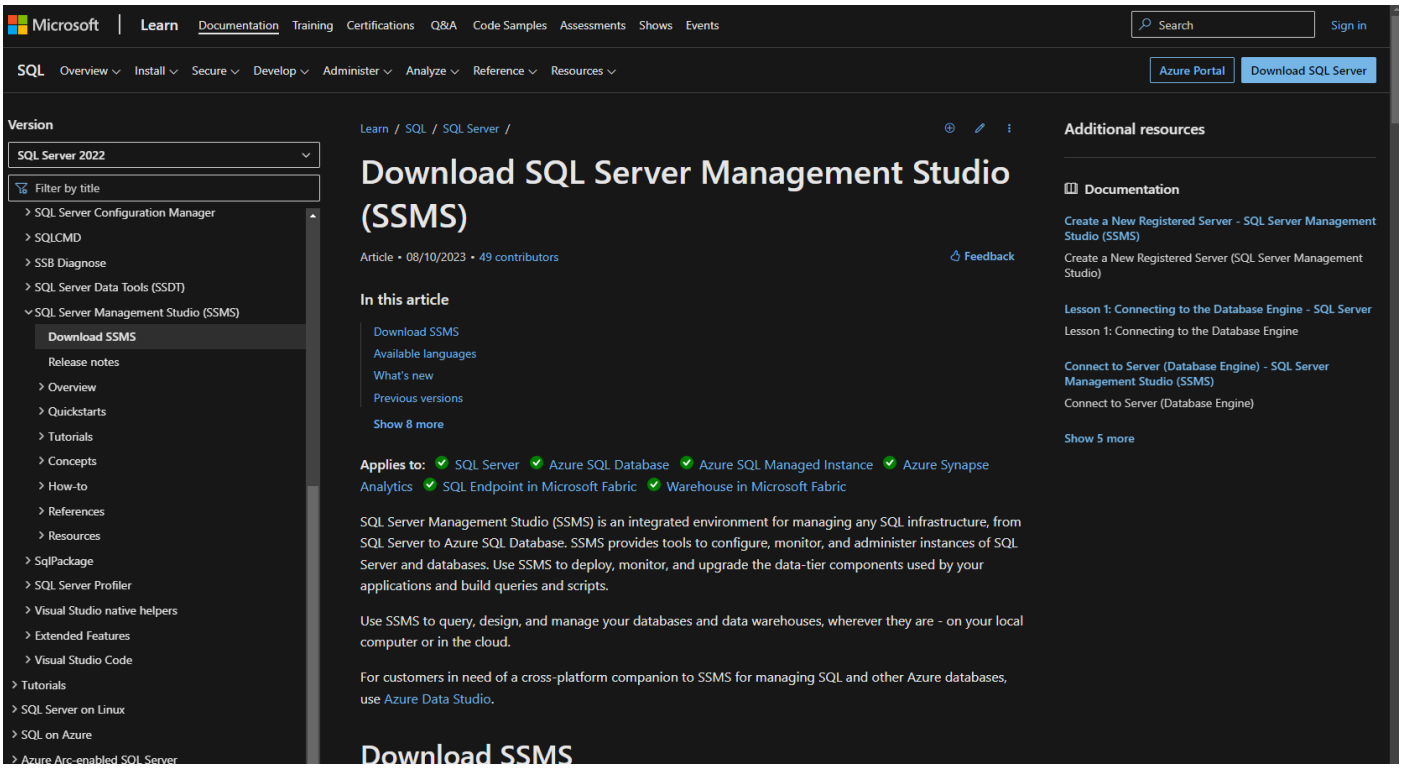

SQL Server 2022 Setup
Ready to Install
Verify the SQL Server 2022 features to be installed.
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Ready to install SQL Server 2022:
Summary
Edition: Developer
Action: Install
Prerequisites
Already installed:
Windows PowerShell 3.0 or higher
Microsoft Visual C++ 2017 Redistributable
General Configuration
Features
Database Engine Services
Full-Text and Semantic Extractions for Search
Instance configuration
Instance Name: MSSQLSERVER
Instance ID: MSSQLSERVER
Instance IDs
SQL Database Engine: MSSQL16.MSSQLSERVER
Instance Directory: C:\Program Files\Microsoft SQL Server\
Shared component root directory
Shared feature directory: C:\Program Files\Microsoft SQL Server\
Shared feature (WOW64) directory: C:\Program Files (x86)\Microsoft SQL Server\
Product Update
Update Enabled: True
Update Source: MU
Configuration file path:
C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap\Log\20230821_093312\ConfigurationFile.ini
< Back
Install
Cancel


SQL Server 2022 Setup
Installation Progress
Edition
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Azure Extension for SQL Server
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Generating ordered dependency feature list.
Next >
Cancel



كده نزلنا ال engine دلوقتي هننزل ال sql server management studio واللي من خلاله هتقدر تعمل الداتا بيز بتاعتك


SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New SQL Server standalone installation or add features to an existing installation
Launch a wizard to install SQL Server 2022 in a non-clustered environment or to add features to an existing SQL Server 2022 instance.
Install SQL Server Reporting Services
Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.
Install SQL Server Management Tools
Launch a download page that provides a link to install SQL Server Management Studio, SQL Server command-line utilities (SQLCMD and BCP), SQL Server PowerShell provider, SQL Server Profiler and Database Tuning Advisor. An internet connection is required to install these tools.
Install SQL Server Data Tools
Launch a download page that provides a link to install SQL Server Data Tools (SSDT). SSDT provides Visual Studio integration including project system support for Microsoft Azure SQL Database, the SQL Server Database Engine, Reporting Services, Analysis Services and Integration Services. An internet connection is required to install SSDT.
New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2022 failover cluster.
This action is only availabe in the clustered environment.
Add node to a SQL Server failover cluster
Launch a wizard to add a node to an existing SQL Server 2022 failover cluster.
This action is only availabe in the clustered environment.
Upgrade from a previous version of SQL Server
Launch a wizard to upgrade a previous version of SQL Server to SQL Server 2022.
Click here to first view Upgrade Documentation
Microsoft SQL Server 2022


Microsoft | Learn
Documentation Training Certifications Q&A Code Samples Assessments Shows Events
Search
Sign in
SQL
Overview Install Secure Develop Administer Analyze Reference Resources
Azure Portal
Download SQL Server
Version
SQL Server 2022
Filter by title
SQL Server Configuration Manager
SQLCMD
SSB Diagnose
SQL Server Data Tools (SSDT)
SQL Server Management Studio (SSMS)
Download SSMS
Release notes
Overview
Quickstarts
Tutorials
Concepts
How-to
References
Resources
SqlPackage
SQL Server Profiler
Visual Studio native helpers
Extended Features
Visual Studio Code
Tutorials
SQL Server on Linux
SQL on Azure
Azure Arc-enabled SQL Server
Learn / SQL / SQL Server /
Download SQL Server Management Studio (SSMS)
Article • 08/10/2023 • 49 contributors
Feedback
In this article
Download SSMS
Available languages
What's new
Previous versions
Show 8 more
Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics SQL Endpoint in Microsoft Fabric Warehouse in Microsoft Fabric
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.
For customers in need of a cross-platform companion to SSMS for managing SQL and other Azure databases, use Azure Data Studio.
Download SSMS
Additional resources
Documentation
Create a New Registered Server - SQL Server Management Studio (SSMS)
Create a New Registered Server (SQL Server Management Studio)
Lesson 1: Connecting to the Database Engine - SQL Server
Lesson 1: Connecting to the Database Engine
Connect to Server (Database Engine) - SQL Server Management Studio (SSMS)
Connect to Server (Database Engine)
Show 5 more

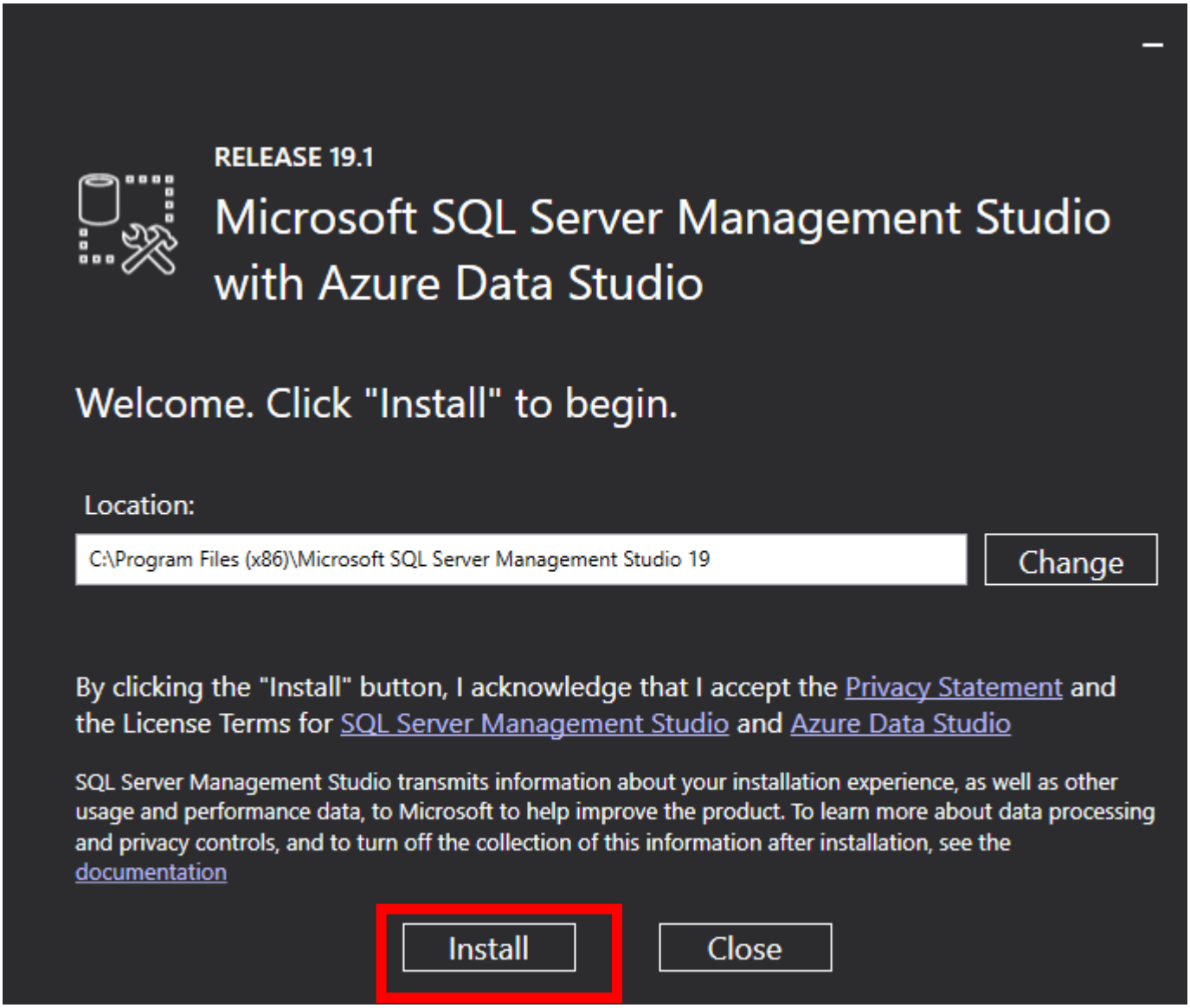
Download SSMS

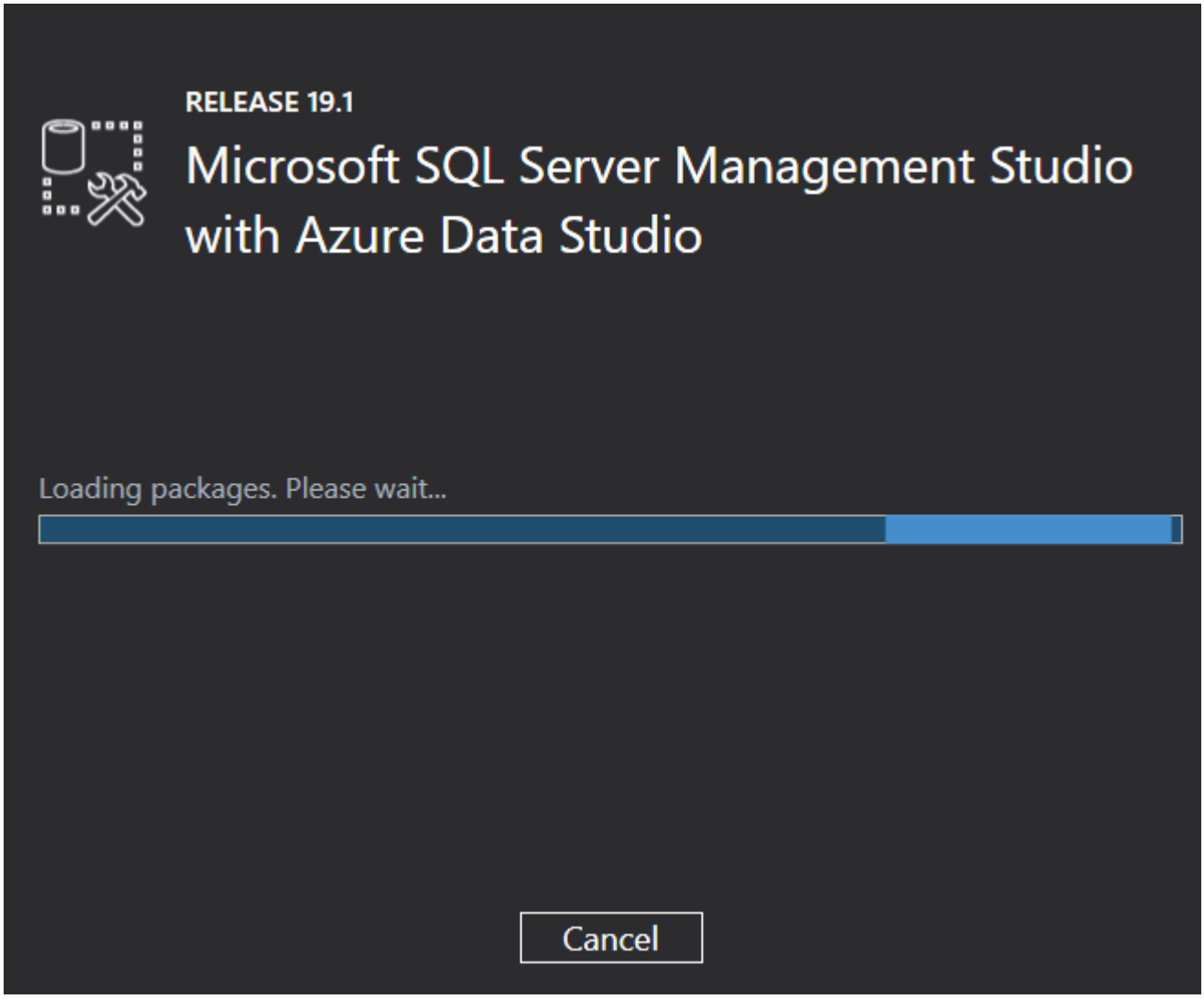
Free Download for SQL Server Management Studio (SSMS) 19.1

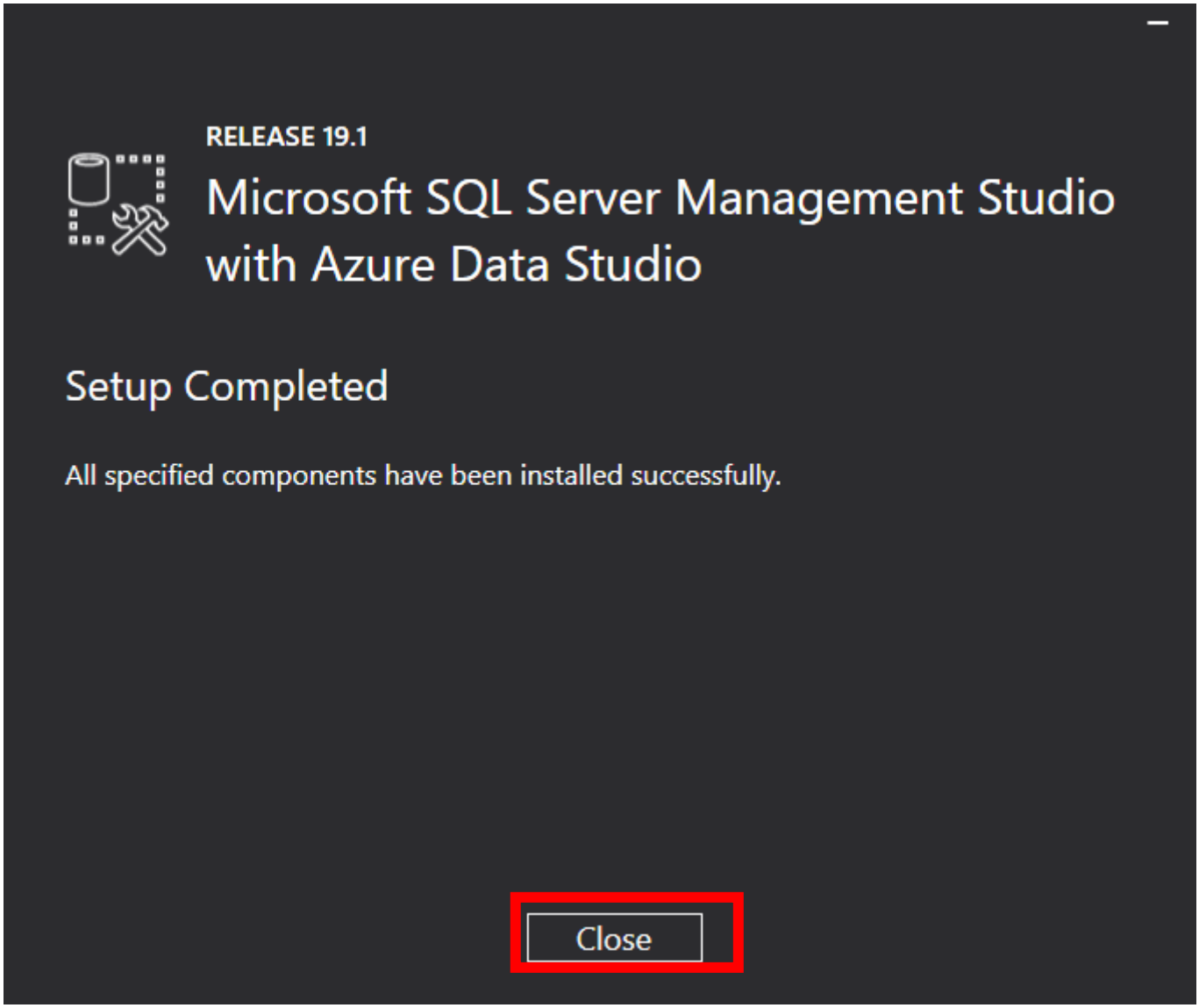
SSMS 19.1 is the latest general availability (GA) version. If you have a *preview* version of SSMS 19 installed, you should uninstall it before installing SSMS 19.1. If you have SSMS 19.x installed, installing SSMS 19.1 upgrades it to 19.1.

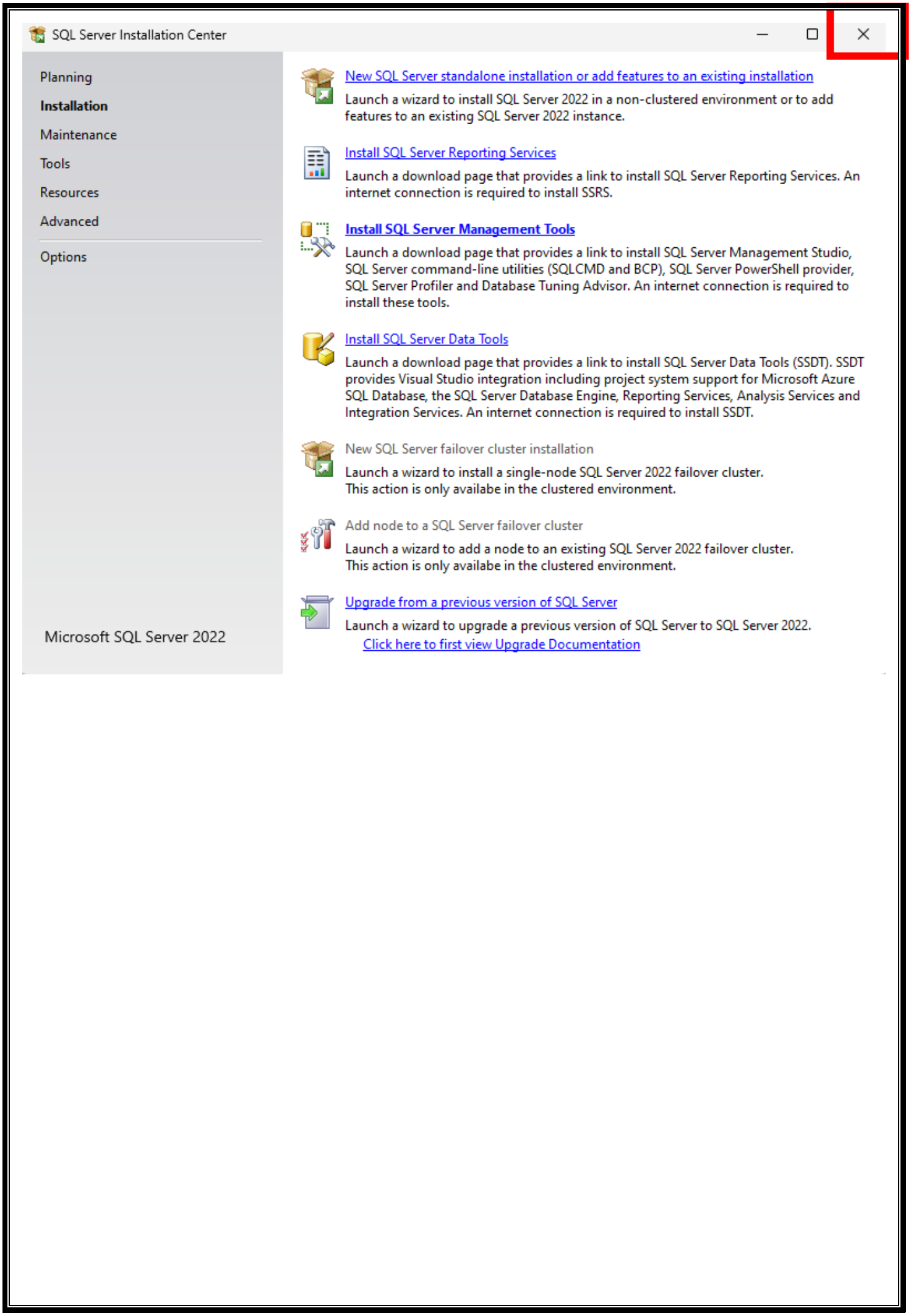
- Release number: 19.1
- Build number: 19.1.56.0
- Release date: May 24, 2023

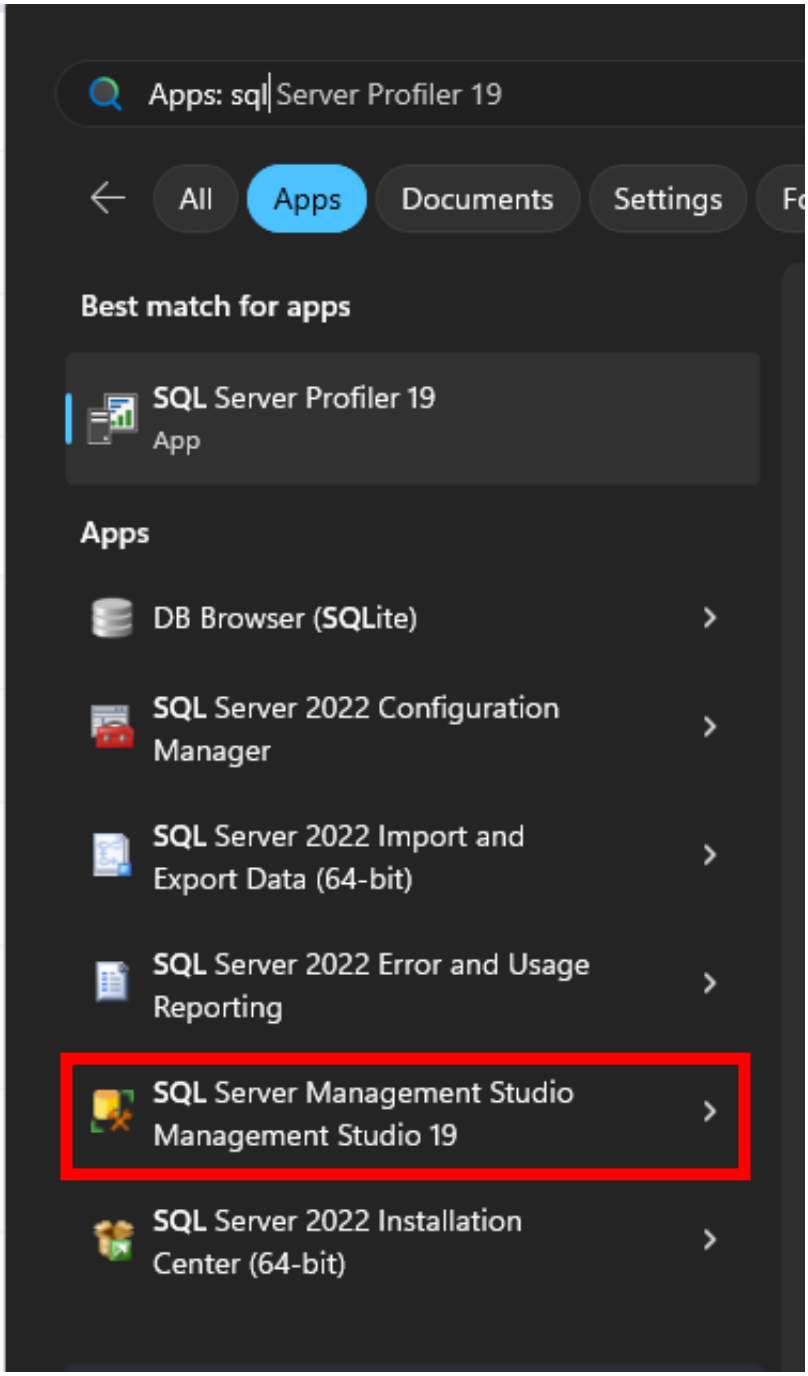
By using SQL Server Management Studio, you agree to its license terms and privacy statement. If you have comments or suggestions or want to report issues, the best way to contact the SSMS team is at SQL Server user

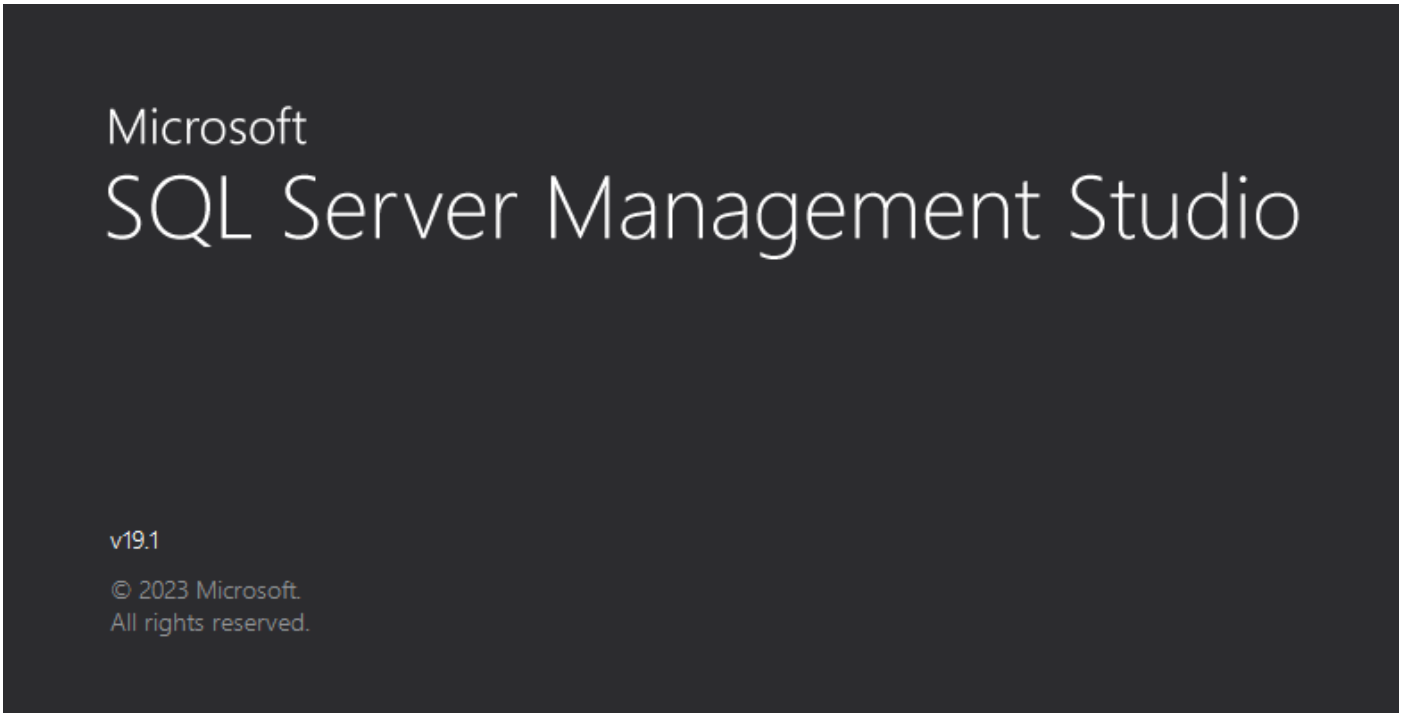


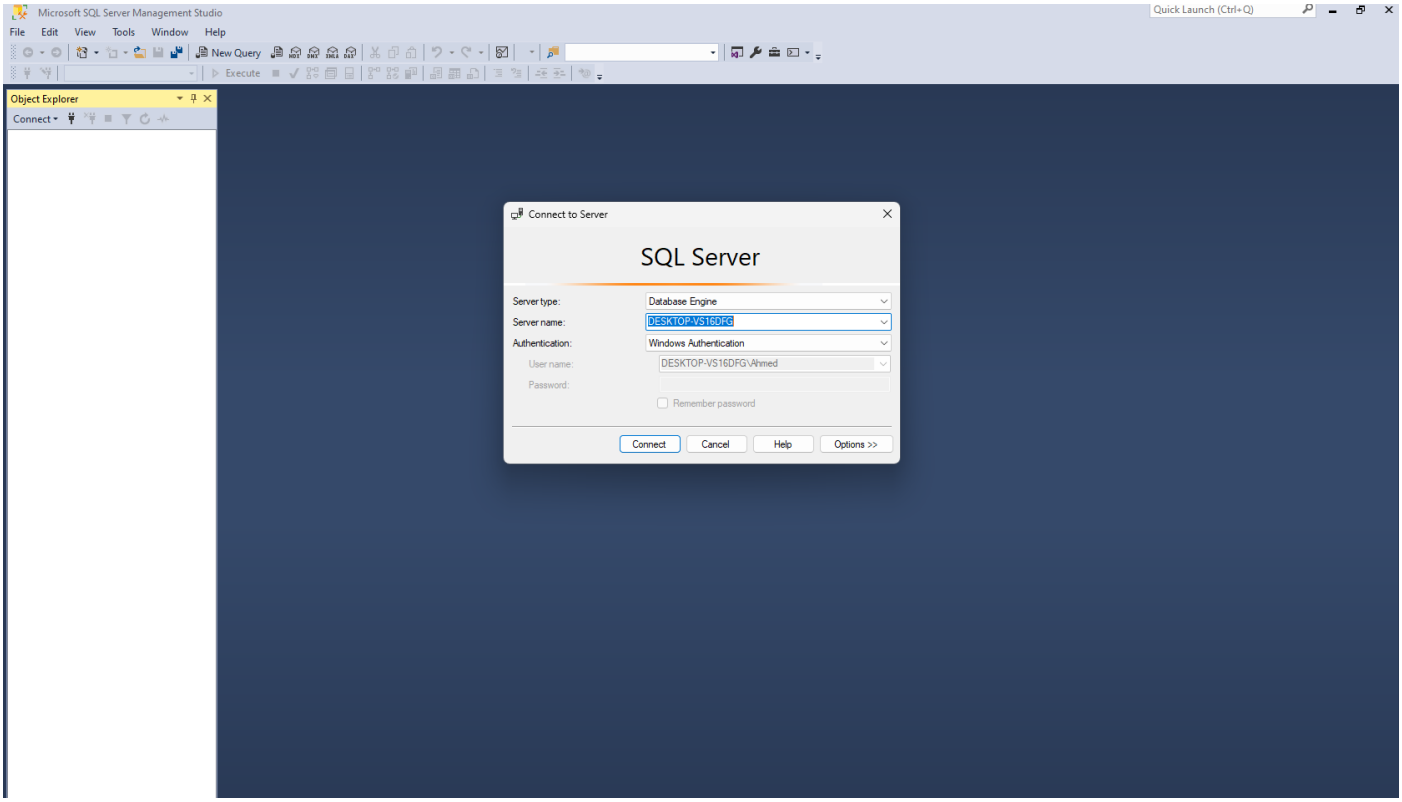

RELEASE 19.1
Microsoft SQL Server Management Studio with Azure Data Studio
Loading packages. Please wait...
Cancel


RELEASE 19.1
Microsoft SQL Server Management Studio with Azure Data Studio
Setup Completed
All specified components have been installed successfully.
Close


SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
Microsoft SQL Server 2022
New SQL Server standalone installation or add features to an existing installation
Launch a wizard to install SQL Server 2022 in a non-clustered environment or to add features to an existing SQL Server 2022 instance.
Install SQL Server Reporting Services
Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.
Install SQL Server Management Tools
Launch a download page that provides a link to install SQL Server Management Studio, SQL Server command-line utilities (SQLCMD and BCP), SQL Server PowerShell provider, SQL Server Profiler and Database Tuning Advisor. An internet connection is required to install these tools.
Install SQL Server Data Tools
Launch a download page that provides a link to install SQL Server Data Tools (SSDT). SSDT provides Visual Studio integration including project system support for Microsoft Azure SQL Database, the SQL Server Database Engine, Reporting Services, Analysis Services and Integration Services. An internet connection is required to install SSDT.
New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2022 failover cluster.
This action is only availabe in the clustered environment.
Add node to a SQL Server failover cluster
Launch a wizard to add a node to an existing SQL Server 2022 failover cluster.
This action is only availabe in the clustered environment.
Upgrade from a previous version of SQL Server
Launch a wizard to upgrade a previous version of SQL Server to SQL Server 2022.
Click here to first view Upgrade Documentation

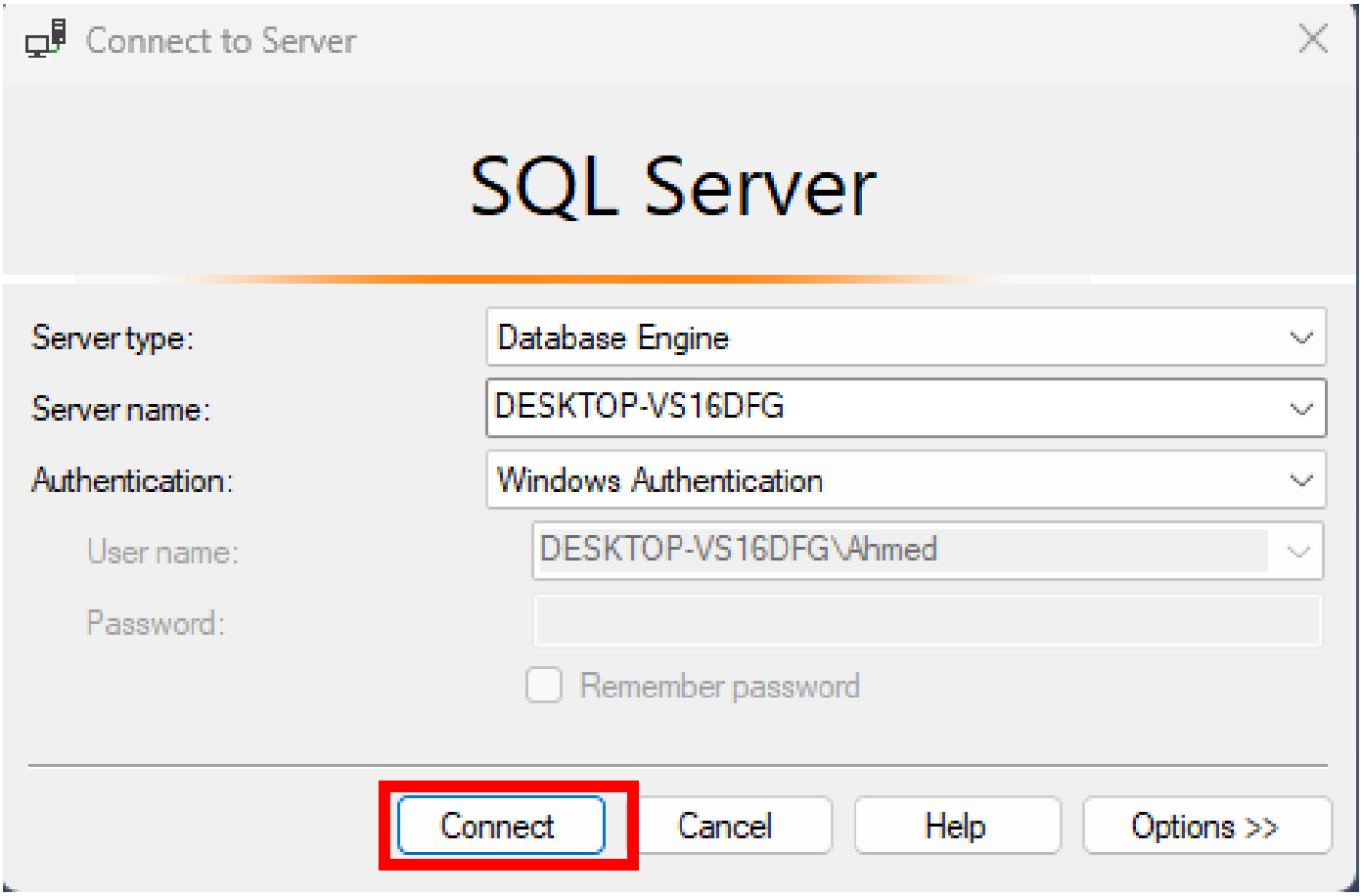

Apps: sql Server Profiler 19
All
Apps
Documents
Settings
Best match for apps
SQL Server Profiler 19
App
Apps
DB Browser (SQLite)
SQL Server 2022 Configuration Manager
SQL Server 2022 Import and Export Data (64-bit)
SQL Server 2022 Error and Usage Reporting
SQL Server Management Studio Management Studio 19
SQL Server 2022 Installation Center (64-bit)

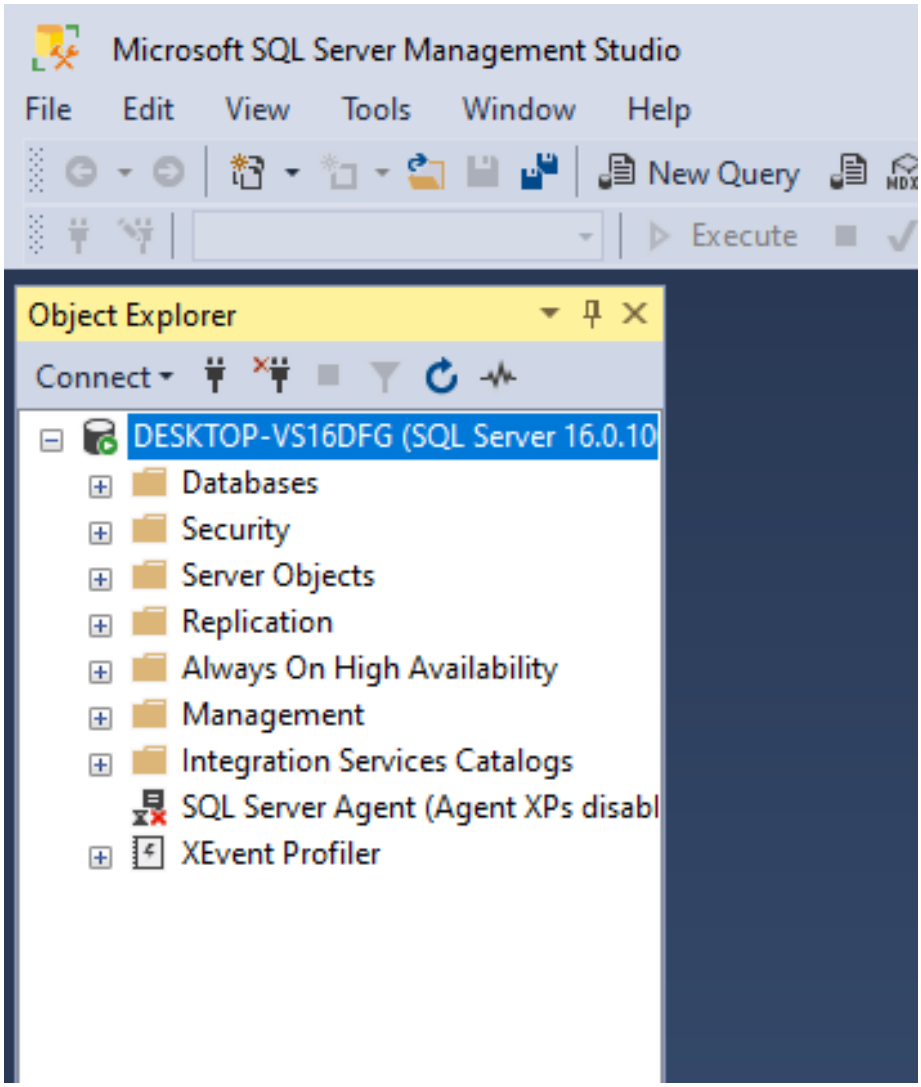




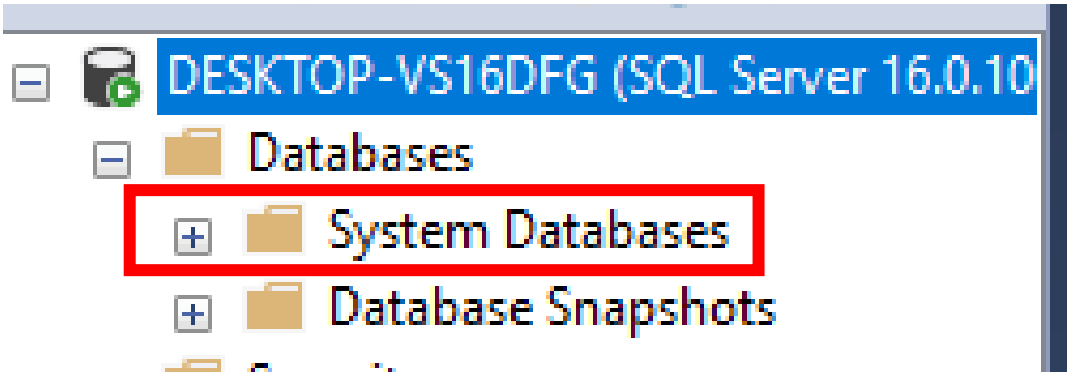
دلوقتي احنا عاوزين نعمل coneection عال sql server اللي موجود عندنا

وتقدر تدخل باليوزر بتاعك بتاع الويندرز او باليوزر والباسورد اللي حطيتهم


Connect to Server
SQL Server
Server type:
Database Engine
Server name:
DESKTOP-VS16DFG
Authentication:
Windows Authentication
User name:
DESKTOP-VS16DFG\Ahmed
Password:
Remember password
Connect
Cancel
Help
Options >>


Microsoft SQL Server Management Studio
File
Edit
View
Tools
Window
Help
New Query
Execute
Object Explorer
Connect
DESKTOP-VS16DFG (SQL Server 16.0.10
Databases
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler

اوعي تلعب في دي



Differences between Data, Information , Knowledge and Wisdom

يجب مراجعة هذا الدرس الذي اخذناه في سلسلة اساسيات مهمة لكل مبرمج

https://youtu.be/qTVRs1wTwho

Differences between Data Structure and Database

يجب مراجعة هذا الدرس الذي اخذناه في كورس

Data Structure

https://programmingadvices.com/courses/database-level-1-sql-concepts-and-practice/lectures/46066083

What is Database?

نتعرف علي كل الأساسيات المشتركه بين كل ال databases والفروق بين كل داتا بيز والتانيه بتكون بسيطه

ماتفوتش ولا درس وافهم كل حاجه كويس

الداتا بيز هيا مجموعه من البيانات او الحقائق المنظمه عشان تقدر توصلها بسهوله وعشان تدير الداتا بيز لازم تستخدم DBMS وهيا اختصار ل data base management system ودي بتنقسم لنوعين :-

1- Non relational DBMS :- البيانات موجوده وبتتخزن بس مفيش علاقه بينها زي ماكنا بنعمل ملف TEXT وبرضه زي ال XML والملفات

2- Relational RDBMS :- زي ال SQL SERVER وال ORACLE وهيا عباره عن داتا فيه علاقات بينها وبين بعضها

What is Database?

A database is an organized collection of data so that it can be easily accessed. To manage these databases, Database Management Systems (DBMS) are used.

Types of DBMS:

In general, there are two common types of databases:

- Non-Relational (DBMS): File System, XML..etc.
- Relational (RDBMS): enhanced version of DBMS but with relations, examples SQLServer, Oracle, MySQL ..etc.

وده مثال عال FILE SYSTEM زي مشروع البنك كده

ID	FirstName	LastName	Gender	Birthdate	Salary	DepID	DeptName	DeptLocation
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1	IT	Amman
2	Ali	Zaher	M	5/7/1990	3000	2	Finance	Amman
3	Lubna	Aqel	F	5/5/2000	500	1	IT	Amman
5	Fadi	Khalil	M	6/6/1980	1400	4	Marketing	Qatar
6	Maha	Majed	M	7/7/2001	300	3	HR	UAE
7	Omar	Ali	M	6/6/1977	2000	1	IT	Amman
8	Huda	Omar	F	4/2/1990	1000	1	IT	Amman

التعامل مع ال files صعب وفيه مشاكل زي ماشوفنا قبل كده ولما تيجي تدخل في الكورس هتعرف قيمة الداتا بيز

والداتا بيز بيتحط الداتا في جداول ويتعمل علاقات بين الجداول دي زي مثلا عندك جدول للموظفين ومن ضمن البيانات بتاعت الموظفين هيا البيانت بتاعت الأقسام اللي هما فيها ف ليه تحط بيانات الأقسام معاهم وانت ممكن تحطهم في جدول منفصل وتربط بين الجدولين برقم بيمثل القسم

In RDBMS.

Data that is stored in an organized fashion in tables containing rows and columns along with relations between these tables.

Employees Table

ID	FirstName	LastName	Gender	Birthdate	Salary	DepartmentID
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1
2	Ali	Zaher	M	5/7/1990	3000	2
3	Lubna	Aqel	F	5/5/2000	500	1
5	Fadi	Khalil	M	6/6/1980	1400	4
6	Maha	Majed	Omar	7/7/2001	300	3
7	Omar	Ali	Salim	6/6/1977	2000	1
8	Hadi	Omar	Abed	4/2/1990	1000	1

Departments Table

ID	Name	Location
1	IT	Amman
2	Finance	Amman
3	HR	UAE
4	Marketing	Qatar

الصف في الجدول بيسموه row او record او entity

العمود في الجدول اسمه column او field او attribute

الجدول كله علي بعضه اسمه entity set او table

In RDBMS.

Data that is stored in an organized fashion in tables containing rows and columns along with relations between these tables.

Table/Entity Set

Column/Field/Attribute

Row/Entity/Record

Employees Table

ID	FirstName	LastName	Gender	Birthdate	Salary	DepartmentID
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1
2	Ali	Zaher	M	5/7/1990	3000	2
3	Lubna	Aqel	F	5/5/2000	500	1
5	Fadi	Khalil	M	6/6/1980	1400	4
6	Maha	Majed	M	7/7/2001	300	3
7	Omar	Ali	M	6/6/1977	2000	1
8	Huda	Omar	F	4/2/1990	1000	1

Departments Table

ID	Name	Location
1	IT	Amman
2	Finance	Amman
3	HR	UAE
4	Marketing	Qatar

الواجب

DBMS and RDBMS are the same.

True

False

DBMS stands for Database Management Systems

True

False

DBMS has relations between Data

True

False

RDBMS has relations between data.

True

False

RDBMS stands for Relational Database Management Systems .

True

False

DBMS has two types: None Relational Database and Relational Database.

True

False

File System and XML are samples of RDBMS.

True

False

File System, XML are samples of DBMS.

True

False

SQL Server , Oracle , MySQL are examples of RDBMS.

True

False

Dealing with RDBMS is much easier than dealing with DBMS.

True

False

In RDBMS: Data that is stored in an organized fashion in tables containing rows and columns along with relations between these tables.

True

False

Column/Field/Attribute are the same.

True

False

Row/Entity/Record are the same

True

False

Table/Entity Set are the same.

True

False

Relationships are represented using references to data from other tables.

True

False

What is NULL?

ساعات بيكون فيه معلومات مش متوفره ومش عايز اعملها initial value عشان امشي الدنيا فبروح للعمود او ال field وبقوله allow null يعني اسمح بانه يكون في العمود ده قيم null وهيا معناها nothing يعني مفيش داتا يعني ماتجيش انه null معناها صفر او مسافه

وكمان بيعرفوا انها distinct value يعني قيمه مميزه ومهم استخدامها بحذرلانه بتأثر علي ال queries والعمليات الحسابيه في الداتا بيز

What is Null?

Employees Table

ID	FirstName	LastName	Gender	Birthdate	Salary	DepartmentID
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1
2	Ali	Zaher	M	Null	Null	2
3	Lubna	Aqel	F	5/5/2000	500	1
5	Fadi	Khalil	M	6/6/1980	1400	4
6	Maha	Majed	M	7/7/2001	300	3
7	Omar	Ali	M	6/6/1977	2000	1
8	Huda	Omar	F	4/2/1990	1000	1

Departments Table

ID	Name	Location
1	IT	Amman
2	Finance	Amman
3	HR	UAE
4	Marketing	Qatar

What is Null

- In a database, "NULL" is a special marker used to indicate that a data value does not exist in the database. It represents a missing or unknown value in a table column.
- When a field or column in a table has a NULL value, it means that the field has no value at all, and it's different from having an empty or zero value. NULL is not the same as a blank space or a zero, and it's a distinct value in the database.
- NULL can be used in several ways, such as indicating missing data, representing optional fields, or as a placeholder for values that are not yet known. However, it's essential to use NULL values carefully because they can affect the results of queries and calculations in unexpected ways.

Summary

- In summary, NULL is a special value in a database that represents the absence of a value in a table column.
- It's used to indicate missing or unknown data and should be used carefully to avoid unexpected results in queries and calculations.

الواجب

NULL is a special value in a database that represents the absence of a value in a table column.

True

False

NULL is used to indicate missing or unknown data and should be used carefully to avoid unexpected results in queries and calculations.

True

False

When a salary field in a table has a NULL value it means its value is zero

True

False

When a salary field in a table has a NULL value it means its value is missing or unknown.

True

False

When a field in a table has a NULL value it means its optional.

True

False

NULL is not the same as a blank space or a zero.

True

False

When a NumberOfChildren field in Employees table has a NULL value this means that employee has zero children.

True

False

When a NumberOfChildren field in Employees table has a NULL value this means that we don't know yet that value, it's missing or unknown to us.

True

False

Primary Key vs Foreign Key / Referential Integrity

ال primary key وال foreign key بيتم استخدامهم في الربط بين الجداول في الداتا بيز

عشان يكون عندي relations بين الجداول لازم اقدر اربطهم مع بعض وعشان اقدر اربطهم مع بعض لازم يكون عندي field من ضمن ال fields بتاعت الجدول تكون القيم اللي فيه مش متكرره

فلو بصيت لجدول الموظفين تحت هتلاقي انه ال id عباره عن primary key وزي ماانت شايف هوا عباره عن ارقام والأرقام دي بتمثل مرجع لل record اللي هيا فيه فماينفعش ابدا ان الرقم ده يتكرر باي شكل زي رقم البطاقه او رقم موبايلك كده ماينفعش يتكرر

لو بصيت في جدول ال department برضه هتلاقي عمود ال id بيمثل primary key للجدول بحيث انك لما تيجي لجدول الموظفين وتقول انه محمد أبو هدهود بيشتغل في ال department رقم واحد لما تروح لجدول ال department وتدور عالرقم ده تبع اني قسم مايجيبلكش قسمين لا هوا قسم واحد

يعني من شروط ال primary key انه مايكونش متكرر ومايكونش ب null ومايتغيرش عشان هيكون مربوط علي اكتر من حاجه فلو غيرته يبقي انت كده بوظت الداتا بيز

بيقول انه ال primary key ممكن يكون اتنين مش واحد بس لا يفضل استخدامه

فيه ناس بتيجي تصمم الداتا بيز وتعمل 4 اعمده مثلا ويحطوهم كلهم primary key ممكن تعمله بس مش بيكون عملي وهيكون ابطأ

طيب بص في جدول ال department هل ينفع اخلي عمود ال name يشتغل primary key ؟

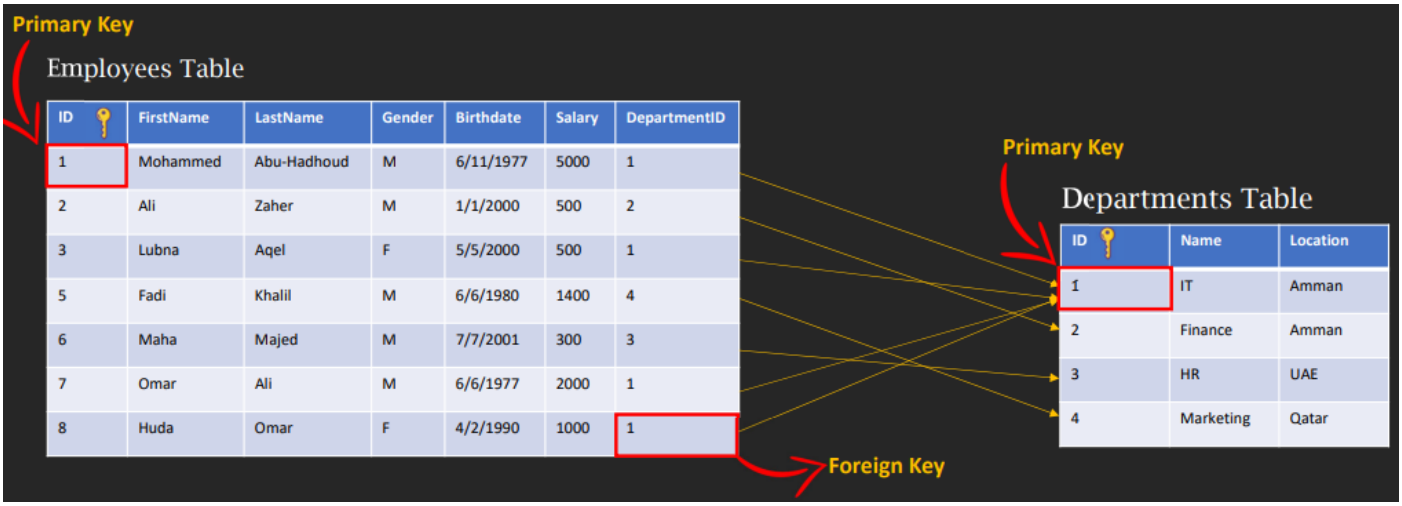
قالك اه ينفع لانه مش بيتكرر ومش بيساوي null لكن لايفضل لانه من النوع string لانه هيكون ابطأ في البحث وهياخد مساحات اكبر

ال primary key لما بيتنقل لجدول تاني عشان يدل عليه بيكون اسمه foreign key زي عمود ال department id في جدول ال employees كده

ال foreign key هوا مؤشر علي primary key في جدول تاني وماينفعش تحط فيه رقم من دماغك عشان مايزعلكش لانه من نفسه بيروح يشوف الرقم ده موجود في الجدول التاني ولا لا ودي اسمها referential integrity

ولو جيت تحذف record مربوط بجداول تانيه مش هيقبل منك ده

وفيه اوبشن بيمثل مصيبه واسمه cascade delete وده لو جيت تحذف record من جدول معين بيحذفه وبيحذف أي record تاني في أي جدول تاني مرتبط بال record اللي انت عاوز تحذفه



Primary Key

- A primary key is a column or set of columns in a relational database table that uniquely identifies each row or record in the table.
- It is a unique identifier for each record and serves as a reference point for other tables that have a relationship with the table in question.
- Each table in a relational database must have a primary key, and it should be a non-null value that is unique and stable over time.
- Primary Key should not be changed.

Foreign Key

- A foreign key, on the other hand, is a column or set of columns in a table that refers to the primary key of another table.
- It establishes a relationship between two tables, allowing data to be shared and linked between them.
- The foreign key ensures that referential integrity is maintained by ensuring that any value in the foreign key column must exist in the primary key column of the related table.

Summary

- In summary, a primary key uniquely identifies a record in a table, while a foreign key establishes a relationship between two tables by referencing the primary key of another table.

الواجب

A primary key is a column or set of columns in a relational database table that uniquely identifies each row or record in the table.

True

False

Primary Key is a unique identifier for each record and serves as a reference point for other tables that have a relationship with the table in question.

True

False

Each table in a relational database must have a primary key, and it should be a non-null value that is unique and stable over time.

True

False

Primary Key should not be changed.

True

False

A foreign key, on the other hand, is a column or set of columns in a table that refers to the primary key of another table.

True

False

Foreign Key establishes a relationship between two tables, allowing data to be shared and linked between them.

True

False

The foreign key ensures that referential integrity is maintained by ensuring that any value in the foreign key column must exist in the primary key column of the related table.

True

False

In summary, a primary key uniquely identifies a record in a table, while a foreign key establishes a relationship between two tables by referencing the primary key of another table.

True

False

Primary Key can be more than one filed.

True

False

It's better to use numbers as primary keys because they are fast in search.

True

False

Primary Key can be NULL.

True

False

Primary Key can be repeated.

True

False

Primary Key should be Unique, Not NULL , and should not be changed.

True

False

Value in the foreign key column must exist in the primary key column of the related table.

True

False

What is Redundancy? and why it's a problem?

ال data redundancy معناها انه عندك تكرار في الداتا الديزاين السيئ للداتا بيز بيؤدي لحدوث تكرارات كتير في الداتا بيز وده ليه مميزاته وليه عيوبه

بص للجدول ده اللي معمول في text file فيه تكرارات

Employees File: Redundancy Problem

ID	FirstName	LastName	Gender	Birthdate	Salary	DepID	DeptName	DeptLocation
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1	IT	Amman
2	Ali	Zaher	M	5/7/1990	3000	2	Finance	Amman
3	Lubna	Aqel	F	5/5/2000	500	1	IT	Amman
5	Fadi	Khalil	M	6/6/1980	1400	4	Marketing	Qatar
6	Maha	Majed	M	7/7/2001	300	3	HR	UAE
7	Omar	Ali	M	6/6/1977	2000	1	IT	Amman
8	Huda	Omar	F	4/2/1990	1000	1	IT	Amman

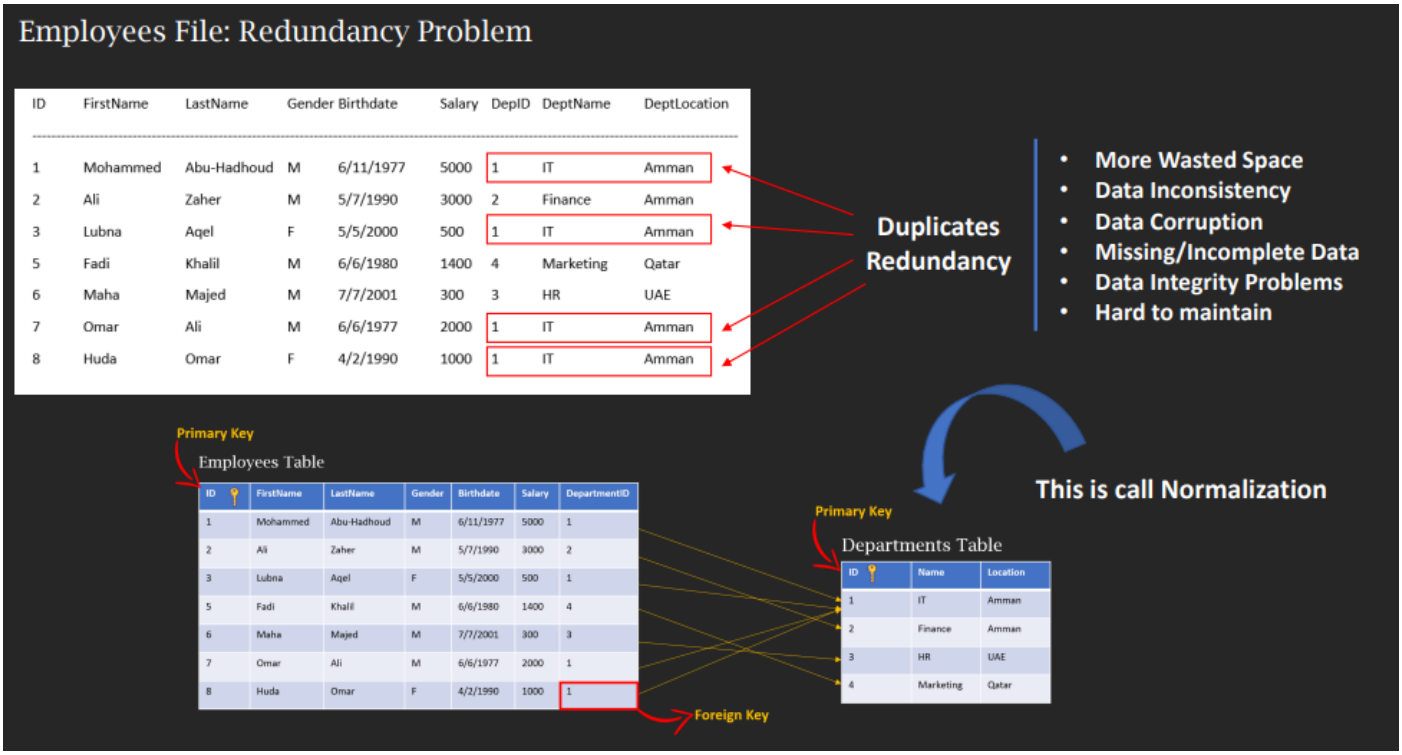
الكلام ده بياخد مساحه في الجهاز عالفاضي وده بيكون بدايه لمشاكل تانيه زي :-

1- المساحه المهدره
2- Data inconsistency :-
3- Data Corruption :-
4- Missing /incomplete data
5- Data integrity
6- Hard to maintain

مثلا في الجدول اللي فوق انت حبيت تغير اسم ال it ل information technology انت كده هتضطر انك تغيره في كل ال records ويه واحد ممكن تنسي تعدل فيه

فلما جت الداتا بيز فصلت بين الداتا فقالك خلي بيانات الموظفين لوحده وبيانات الأقسام لوحدها وبالتالي لو عايز تعدل علي بيانات قسم معين هتعدله مره واحده وخلصنا وده شكل من اشكال ال normalization وهوا حل لمشكلة ال redundancy

ال redundancy مش دايما بتكون ضاره لانها ساعات بتكون مفيده واسرع في اني مثلا عاوز ازود عمود في جدول الأقسام يقولي عدد الموظفين في القسم ده دي هتكون معلومه مشتقه من جدول الموظفين فالافضل هنا انك تستخدم ال redundancy



Redundancy

- Redundancy refers to the presence of duplicated data within the database. Redundancy can occur in different ways, such as storing the same information multiple times, or storing information that can be derived from other data in the database.

- While redundancy can sometimes be useful, it can also cause problems. For example, redundant data takes up additional storage space and can make it more difficult to maintain consistency within the database. If one copy of the data is updated, the other copies may become outdated, leading to inconsistencies and errors.

- To avoid redundancy in databases, normalization techniques can be used to organize the data in a way that minimizes duplication.

Normalization

- Normalization is the process of organizing data in a database in a way that reduces redundancy and improves data integrity. There are several levels of normalization, also known as normal forms, each with its own set of rules.
- To avoid redundancy in databases, normalization techniques can be used to organize the data in a way that minimizes duplication.
- This involves breaking down the data into smaller, more atomic pieces and linking them together through relationships, which can reduce the amount of redundant data and make it easier to manage and update the database.
- Additionally, enforcing data constraints, such as unique keys and foreign key relationships, can help prevent redundant data from being inserted into the database.
- We will talk more about normalization in the future not now.

الواجب

Redundancy refers to the presence of duplicated data within the database.

True

False

Redundancy can occur in different ways, such as storing the same information multiple times, or storing information that can be derived from other data in the database.

True

False

Redundancy is always bad.

True

False

Redundancy can sometimes be useful.

True

False

Redundant data takes up additional storage space.

True

False

Redundancy can make it more difficult to maintain consistency within the database. If one copy of the data is updated, the other copies may become outdated, leading to inconsistencies and errors.

True

False

To avoid redundancy in databases, normalization techniques can be used to organize the data in a way that minimizes duplication.

True

False

Normalization is the process of organizing data in a database in a way that reduces redundancy.

- True
- False

Normalization improves data integrity.

- True
- False

Normalization involves breaking down the data into smaller, more atomic pieces and linking them together through relationships, which can reduce the amount of redundant data and make it easier to manage and update the database.

- True
- False

Enforcing data constraints, such as unique keys and foreign key relationships, can help prevent redundant data from being inserted into the database.

- True
- False

Constraints are Restrictions/Rules on Data.

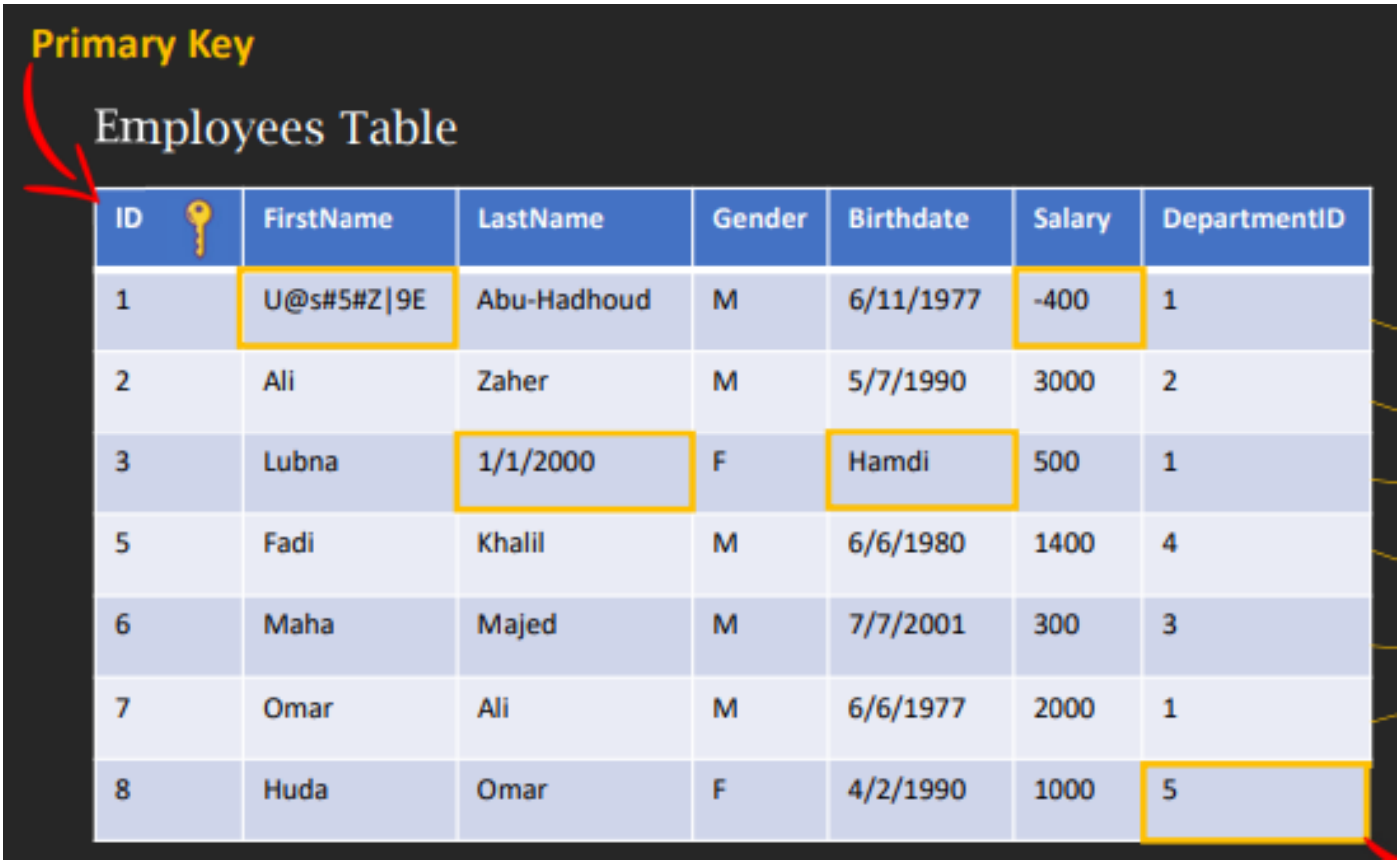
True

False

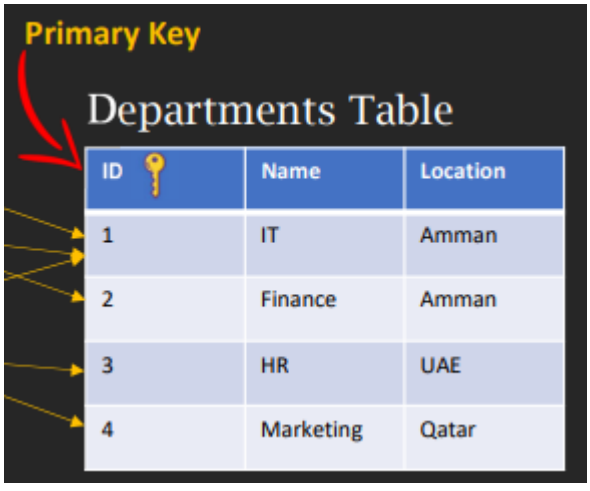
What is Data Integrity? and Why it's Important and Critical?

ال data integrity معناها تكامل البيانات وهنا بنقصد بيها انك لازم تتأكد ان كل البيانات اللي بتدخل الجدول عندك تكون مظبوطه وصحيحه لانه ساعات بيحصل أخطاء او مشاكل في الداتا بيز

بص كده الجدول ده



ID	FirstName	LastName	Gender	Birthdate	Salary	DepartmentID
1	U@s#5#Z|9E	Abu-Hadhoud	M	6/11/1977	-400	1
2	Ali	Zaher	M	5/7/1990	3000	2
3	Lubna	1/1/2000	F	Hamdi	500	1
5	Fadi	Khalil	M	6/6/1980	1400	4
6	Maha	Majed	M	7/7/2001	300	3
7	Omar	Ali	M	6/6/1977	2000	1
8	Huda	Omar	F	4/2/1990	1000	5



ID	Name	Location
1	IT	Amman
2	Finance	Amman
3	HR	UAE
4	Marketing	Qatar

اول مشكلة عندك انه ال first name في ال record الأول corrupted ودي احد مشاكل ال data integrity وهيا ال data corruption وده بيحصل لما بتعمل migration للداتا من سيستم قديم لسيتم جديد يعني اثناء نقل الداتا من سيستم للتاني بيحصل خطأ معين هوا اللي بيبوظ الداتا دي

تاني مشكله هيا ال salary في اول record جايبلك قيمته بالسالب ممكن تكون عامل data validation في الكود بتاعك بس بيقولك ده مش كفايه وانك لازم تعمله تاني علي مستوي الداتا بيز بانك تعمل constraint يعني قيد عال field ده انه مايقبلش غير ارقام موجبه فقط

بص في ال last name في ال record رقم 3 هتلاقيه كاتب تاريخ بدل مايكتب اسم الشخص دي برضه ممكن تكون ناتجه عن ال data migration او يكون حد مش واخد باله وكتب التاريخ بدل الاسم

بص برضه عال field بتاع التاريخ في نفس ال record حاطط الاسم مكان التاريخ ده برضه من مشاكل ال data integrity ان انا مش عامل ال data type صح عامله string بدل مايكون date

بص عال record الأخير هتلاقيه كاتب في ال department id الرقم 5 وده مش موجود في الجدول بتاع الأقسام وده بيكون بسبب ان انا مش عامل referential integrity بين الجدولين يعني مش معرف العلاقه بين الجدولين

عشان كده بيقولك انك تعمل داتا بيز صح يعتمد عليها ده اهم بكتير من انك تكتب query وده مش حته ادخال البيانات بس لا ده كمان بيشمل الحذف والتعديل والنقل واي حد بيتعامل مع الداتا لازم يحط الأنواع دي من ال data integrity في اعتباره :-

1- Entity integrity :- لازم تتاكد انه كل record ليه id من خلاله تقدر تستدعي أي شيء في ال record ده
2- Referential integrity :- يعني الجداول مش مرتبطه ببعضها زي ماحصل وضفت الرقم 5 في عمود ال department id ومفيش في جدول ال department الرقم ده
3- Domain integrity :- يعني ماتجيش تخلي ال salary بالسالب او التاريخ مثلا مايكونش من ضمن فتره معينه
4- Business inregrty :- انه الداتا تكون متوافقه مع شروط وقواعد الشركه زي مثلا انه الحسابات البنكيه ماينفعش تقل او تزيد عن رقم معين او انك في مستشفي والمستشفي مانعه انه بيانات المرضي فيه يتم مشاركتها مع حد

Data Integrity

- Data integrity refers to the accuracy, consistency, and reliability of data over its entire life cycle, from creation to deletion. In other words, it refers to the assurance that data is complete, accurate, and trustworthy.

- There are several factors that can impact data integrity, including human error, hardware or software failure, security breaches, and data transfer errors.

- To maintain data integrity, it is important to establish appropriate policies and procedures, and to implement appropriate technologies, such as encryption, backups, and access controls.

Data Integrity

There are different types of data integrity that organizations need to consider:

1. Entity integrity: This ensures that each row or record in a table is unique and can be uniquely identified. This is typically achieved through the use of primary keys.

2. Referential integrity: This ensures that relationships between tables are maintained and that there are no orphaned records. This is typically achieved through the use of foreign keys.

3. Domain integrity: This ensures that data is within acceptable ranges or values. For example, a date field should only contain valid dates, and a numeric field should only contain valid numbers.

Data Integrity

4. Business integrity: This ensures that data meets business rules and requirements. For example, a bank might have rules around minimum and maximum account balances, or a hospital might have rules around patient data confidentiality.

Data Integrity

- Maintaining data integrity is critical for organizations that rely on accurate and trustworthy data to make informed decisions. Without data integrity, organizations risk making decisions based on incomplete, inaccurate, or unreliable data, which can lead to poor outcomes, financial losses, and damage to reputation.

- To maintain data integrity we use **Constraints**, we will explain them in the next lesson ☺.

الواجب

Data integrity refers to the accuracy, consistency, and reliability of data over its entire life cycle, from creation to deletion.

True

False

Data Integrity refers to the assurance that data is complete, accurate, and trustworthy.

True

False

There are several factors that can impact data integrity, including human error, hardware or software failure, security breaches, and data transfer errors.

True

False

To maintain data integrity, it is important to establish appropriate policies and procedures, and to implement appropriate technologies, such as encryption, backups, and access controls.

True

False

There are different types of data integrity that organizations need to consider: Entity integrity, Referential integrity, Domain integrity, and Business integrity.

True

False

Referential integrity: This ensures that each row or record in a table is unique and can be uniquely identified. This is typically achieved through the use of primary keys.

True

False

Entity integrity: This ensures that each row or record in a table is unique and can be uniquely identified. This is typically achieved through the use of primary keys.

True

False

Referential integrity: This ensures that relationships between tables are maintained and that there are no orphaned records. This is typically achieved through the use of foreign keys.

True

False

Domain integrity is the same as Business Integrity.

True

False

Domain integrity: This ensures that data is within acceptable ranges or values. For example, a date field should only contain valid dates, and a numeric field should only contain valid numbers.

True

False

Business integrity: This ensures that data meets business rules and requirements. For example, a bank might have rules around minimum and maximum account balances, or a hospital might have rules around patient data confidentiality.

True

False

What is Constraint? and Why it's Important?

ال constraints هيا شروط وقواعد بتطبقها علي الداتا عشان تضمن او تحقق ال data integrity ودي تقدر تعمل علي مستوي الاعمده او الجدول كله ودي بعض أنواع ال constrains :-

1- Primary key :- وده بيضمنلك انه كله صف في الجدول ليه رقم خاص بيه وماينفعش يتغير زي مااتكلمنا قبل كده
2- Foreign key :- وده بيضمنلك انه كل صف في الجدول بتاعك ليه علاقه بصف تاني في جدول تاني وان مافيش حد بيحط رقم مش موجود في الجدول التاني
3- Unique constraints :- وده انك بتعرف عمود معين انه الداتا اللي فيه لا تقبل التكرار زي انك مثلا بتدخل بيانات الشيكات رقم الشيك ماينفعش يتغير او بيانات عملاء ليهم رقم قومي ماينفعش يتكرر العمود ده لا هوا primary key ولا حتي foreign key بس هوا بنفسه الداتا اللي فيه ماينفعش تتكرر زي ال serial number بتاع الموبايل

4- Not null :- انك بتيجي تعرف عمود معين انه لازم يكون فيه داتا ماينفعش يكون ب null
5- Check constraint :- انك تحط مثلا انه ماينفعش ال age يكون اقل من 18 سنه

وفيه معلومه هنا بيقولهالك انه الفرق بين ال primary key وا unique constraint انه ال unique constraint بتقبل ال null

Constraints

- In the context of databases, constraints are rules or conditions that are applied to the data to ensure its integrity and consistency. Constraints can be applied to individual columns or to entire tables, and they are used to enforce various rules and restrictions on the data.

- By using constraints, you can help ensure that your data is accurate, consistent, and easy to manage.

Constraints

Here are some common types of constraints used in databases:

1. Primary Key Constraint: This constraint ensures that a column or a set of columns uniquely identifies each row in a table. This constraint helps to enforce data integrity and ensure that there are no duplicate rows in the table.

2. Foreign Key Constraint: This constraint establishes a relationship between two tables based on a key field. The foreign key constraint ensures that data in one table matches data in another table, and it helps to maintain referential integrity in the database.

3. Unique Constraint: This constraint ensures that the data in a column or set of columns is unique across all rows in the table. This constraint helps to enforce data integrity and prevent duplicate values from being inserted into the table.

Constraints

Here are some common types of constraints used in databases:

4. Not Null Constraint: This constraint ensures that a column or set of columns cannot contain null (empty) values. This constraint helps to ensure that the data is complete and accurate, and it can help prevent errors in queries and calculations.

5. Check Constraint: This constraint ensures that the data in a column or set of columns meets a specified condition. This constraint helps to enforce data integrity and prevent invalid data from being inserted into the table.

Interview Question?

What is the difference between Primary Key Constraint and Unique Constraint?

Primary Key is Unique but it does not allow NULL ☺ while Unique allows NULL.

الواجب

In the context of databases, constraints are rules or conditions that are applied to the data to ensure its integrity and consistency. Constraints can be applied to individual columns or to entire tables, and they are used to enforce various rules and restrictions on the data.

True

False

By using constraints, you can help ensure that your data is accurate, consistent, and easy to manage.

True

False

The constraint that ensures that a column or a set of columns uniquely identifies each row in a table. This constraint helps to enforce data integrity and ensure that there are no duplicate rows in the table is?

- Foreign Key Constraint
- Unique Constraint
- Primary Key Constraint
- Check Constraint
- Not Null Constraint

Which constraint ensures that the data in a column or set of columns meets a specified condition. This constraint helps to enforce data integrity and prevent invalid data from being inserted into the table?

- Not Null Constraint
- Primary Key Constraint
- Check Constraint
- Foreign Key Constraint
- Unique Constraint

Which constraint establishes a relationship between two tables based on a key field. The foreign key constraint ensures that data in one table matches data in another table, and it helps to maintain referential integrity in the database?

- Unique Constraint
- Foreign Key Constraint
- Primary Key Constraint
- Not Null Constraint
- Check Constraint

Which constraint ensures that a column or set of columns cannot contain null (empty) values. This constraint helps to ensure that the data is complete and accurate, and it can help prevent errors in queries and calculations?

- Foreign Key Constraint
- Primary Key Constraint
- Unique Constraint
- Not Null Constraint
- Check Constraint

Which constraint ensures that the data in a column or set of columns is unique across all rows in the table. and allows Null as well, This constraint helps to enforce data integrity and prevent duplicate values from being inserted into the table?

- Check Constraint
- Unique Constraint
- Primary Key Constraint
- Foreign Key Constraint
- Not Null Constraint

Which constraint(s) ensures that the data in a column or set of columns is unique across all rows in the table. This constraint helps to enforce data integrity and prevent duplicate values from being inserted into the table.

- Primary Key Constraint
- Foreign Key Constraint
- Not Null Constraint
- Unique Constraint
- Check Constraint

Which of the following is TRUE?

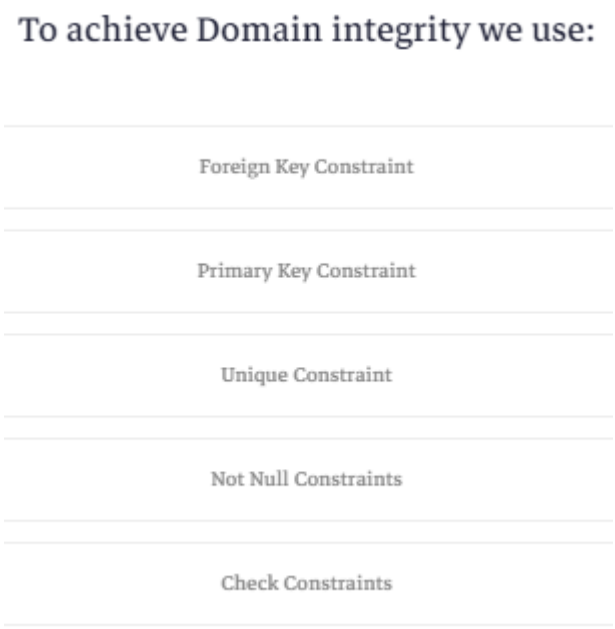
- Primary Key is Unique and allows NULL while Unique Does not allow NULL.
- Primary Key is Unique Constraint but does not allow NULL while Unique Constraint allows NULL.
- Both Primary Key Constraint and Unique Constraint prevent duplicates.

To achieve Entity integrity we use:

- Foreign Key Constraints
- Unique Constraints
- Primary Key Constraints
- Check Constraints
- Not Null Constraints

To achieve Referential integrity we use:

- Foreign Key Constraints
- Unique Constraints
- Primary Key Constraints
- Check Constraints
- Not Null Constraints



What is SQL?

بص كده عالجدول ده وتخيل اننا كنا بنحفظ الداتا دي في ملف text

ID	FirstName	LastName	Gender	Birthdate	Salary	DepID	DeptName	DeptLocation
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1	IT	Amman
2	Ali	Zaher	M	5/7/1990	3000	2	Finance	Amman
3	Lubna	Aqel	F	5/5/2000	500	1	IT	Amman
5	Fadi	Khalil	M	6/6/1980	1400	4	Marketing	Qatar
6	Maha	Majed	M	7/7/2001	300	3	HR	UAE
7	Omar	Ali	M	6/6/1977	2000	1	IT	Amman
8	Huda	Omar	F	4/2/1990	1000	1	IT	Amman

لو جيت قولتلك عايزين بيانات الموظفين الاناث هتعمل ايه ؟

كنا هنحمل الداتا اللي في الملف ده كلها وبعدين نلف عليهم كهم عشان نعملهم separation لانه كل record هوا عباره عن string وبعدين نخزنهم في شكل معين structure او كلاس وبعدين نلف عليهم كلهم ونشوف لو ال gender كان female بناخدها نعرضها

فعشان بس تعمل فلتر للداتا اخدت وقت وقت كبير واستهلكت موارد كتير

طيب لو عايز اعمل نفس الحوار ده في ال sql كل اللي هعمله اني استخدم حاجه اسمها sql query ودي بتكون بسيطه وسريعه

فكل اللي هنعمله اننا نكتب select statement زي كده

```
Select * from Employees
Where Gender = 'F'
```

اللغه اللي كتبنا بيها هيا عباره عن query language وهيا لغه قريبه من لغة الانسان واللغه دي هيا جزء من حاجه اسمها sql وده اختصار ل structured query language وياما بننطقها s q l او بننطقها سيكوال see qual

ال sql بنستخدمها عشان نتواصل مع الداتا بيز عشان نوصل للداتا او نعمل عليها عمليات معينه

وهنا بيقولك انه ال sql هيا standard يعني اللي هتتعلمه هنا هينفع تطبقه علي كل أنواع الداتا بيز اللي موجوده سواء sql server او oracle او حتي Microsoft access

هيا لغه مبنيه علي أوامر وتقدر بيها تعمل ايه حاجه عالداتا بيز من اضافه وحذف وتعديل

وبعد كده بيديك امثله للاوامر دي

وبعدين بيقولك ان ال sql جواها 5 لغات :-

1- Data definition language(DDL) :- ودي من خلالها بتتم عمليات خاصه بالداتا بيز او الجدوال نفسها نفسها التعريف بالجداول الجديده والحذف والاستبدال والاوامر اللي فيها هيا (create-Drop – Alter - truncate)

2- Data manipulation language(DML) :-ودي من خلالها بتتم عمليات التعامل مع الداتا او البيانات نفسها او ال زي (insert – update – delete)

3- DATA Control Language(DCL) :-زي حوار الصلاحيات بتاعت المستخدمين والاوامر اللي فيها (grant – revoke)

4- Transaction Control language (TCL) :- ودي هنجيلها بعدين والاوامر اللي فيها (commit – rollback – save point)

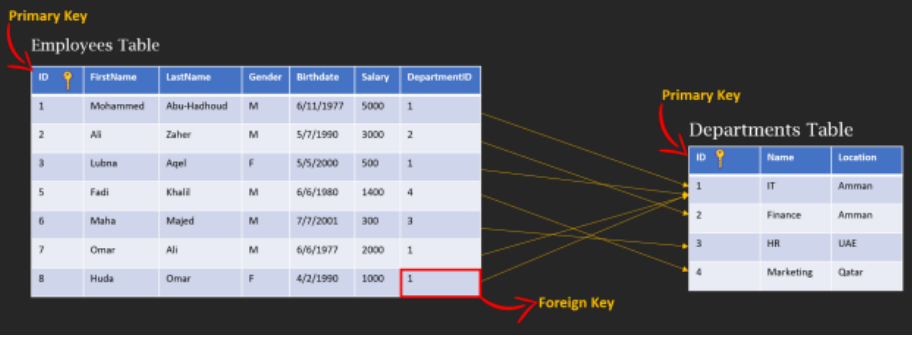
5- Data Query language (DQL) :- ودي اللي هيا الامر select ومن خلاله بتقدر تستخرج تقارير

Employees File:

ID	FirstName	LastName	Gender	Birthdate	Salary	DepID	DeptName	DeptLocation
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1	IT	Amman
2	Ali	Zaher	M	5/7/1990	3000	2	Finance	Amman
3	Lubna	Aqel	F	5/5/2000	500	1	IT	Amman
5	Fadi	Khalil	M	6/6/1980	1400	4	Marketing	Qatar
6	Maha	Majed	M	7/7/2001	300	3	HR	UAE
7	Omar	Ali	M	6/6/1977	2000	1	IT	Amman
8	Huda	Omar	F	4/2/1990	1000	1	IT	Amman

To Get All Female Employees

- You have to write code and loops.
- Slow process
- Slow performance



- You Use SQL Query ☺
- Simple and fast.

```
Select * from Employees
Where Gender = 'F'
```

What is SQL

- SQL stands for Structured Query Language
- Pronounced as "S-Q-L" or sometimes as "See-Quel".
- SQL is used to communicate with a database.
- SQL lets you access and manipulate databases
- Database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.

What Can You Do With SQL?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Examples Of SQL Statements:

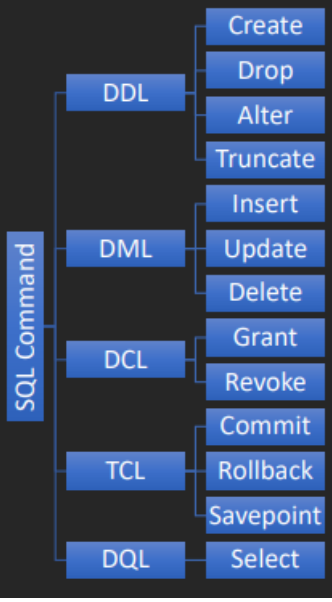
- SELECT * FROM Employees WHERE Salary < 1000;
- SELECT FirstName , LastName FROM Employees WHERE Salary < 1000 and Gender='M';
- SELECT * FROM Employees WHERE Salary between 500 and 1000;
- Select Count(*) from Employees;
- Select Sum(Salary) from Employees;
- Select Avg(Salary) from Employees;
- Delete from Employees where ID=10;
- Update Employees set FirstName = 'Amjad' where ID = 10;

Examples:

- CREATE DATABASE MyDatabase;
- DROP DATABASE MyDatabase;
- CREATE TABLE Employees (
  ID int,
  FirstName varchar(255),
  LastName varchar(255),
  Address varchar(255),
  City varchar(255)
  );

Types of SQL Statements:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- Data Query Language (DQL)



الواجب

SQL stands for Structured Query Language

True

False

SQL is Pronounced as “S-Q-L” or sometimes as “See-Quel”.

True

False

SQL is used to communicate with a database.

True

False

SQL lets you access and manipulate databases.

True

False

Database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.

True

False

SQL can execute queries against a database.

True

False

SQL can retrieve data from a database.

True

False

SQL can insert and update records in a database.

True

False

SQL can delete records from a database.

- True
- False

SQL can create new databases, Tables, Views ..etc in the database

- True
- False

SQL can set permissions on tables, procedures, and views.

- True
- False

What are 5 Types of SQL Statements?

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- Data Query Language (DQL)
- Data Handling Language (DHL)

DBMs vs RDBMS Summary

What is Database?

A database is an organized collection of data so that it can be easily accessed. To manage these databases, Database Management Systems (DBMS) are used.

Types of DBMS:

In general, there are two common types of databases:

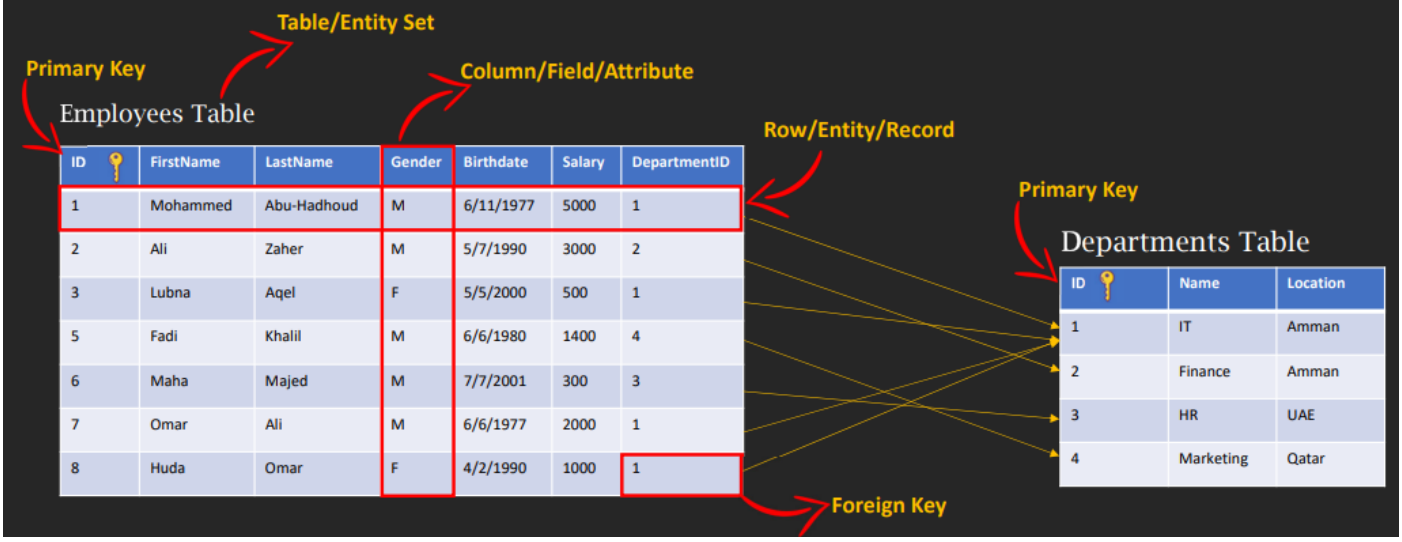
- Non-Relational (DBMS): File System, XML..etc.
- Relational (RDBMS): enhanced version of DBMS but with relations, examples SQLServer, Oracle, MySQL ..etc.

Employees File

ID	FirstName	LastName	Gender	Birthdate	Salary	DepID	DeptName	DeptLocation
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1	IT	Amman
2	Ali	Zaher	M	5/7/1990	3000	2	Finance	Amman
3	Lubna	Aqel	F	5/5/2000	500	1	IT	Amman
5	Fadi	Khalil	M	6/6/1980	1400	4	Marketing	Qatar
6	Maha	Majed	M	7/7/2001	300	3	HR	UAE
7	Omar	Ali	M	6/6/1977	2000	1	IT	Amman
8	Huda	Omar	F	4/2/1990	1000	1	IT	Amman

In RDBMS.

Data that is stored in an organized fashion in tables containing rows and columns along with relations between these tables.

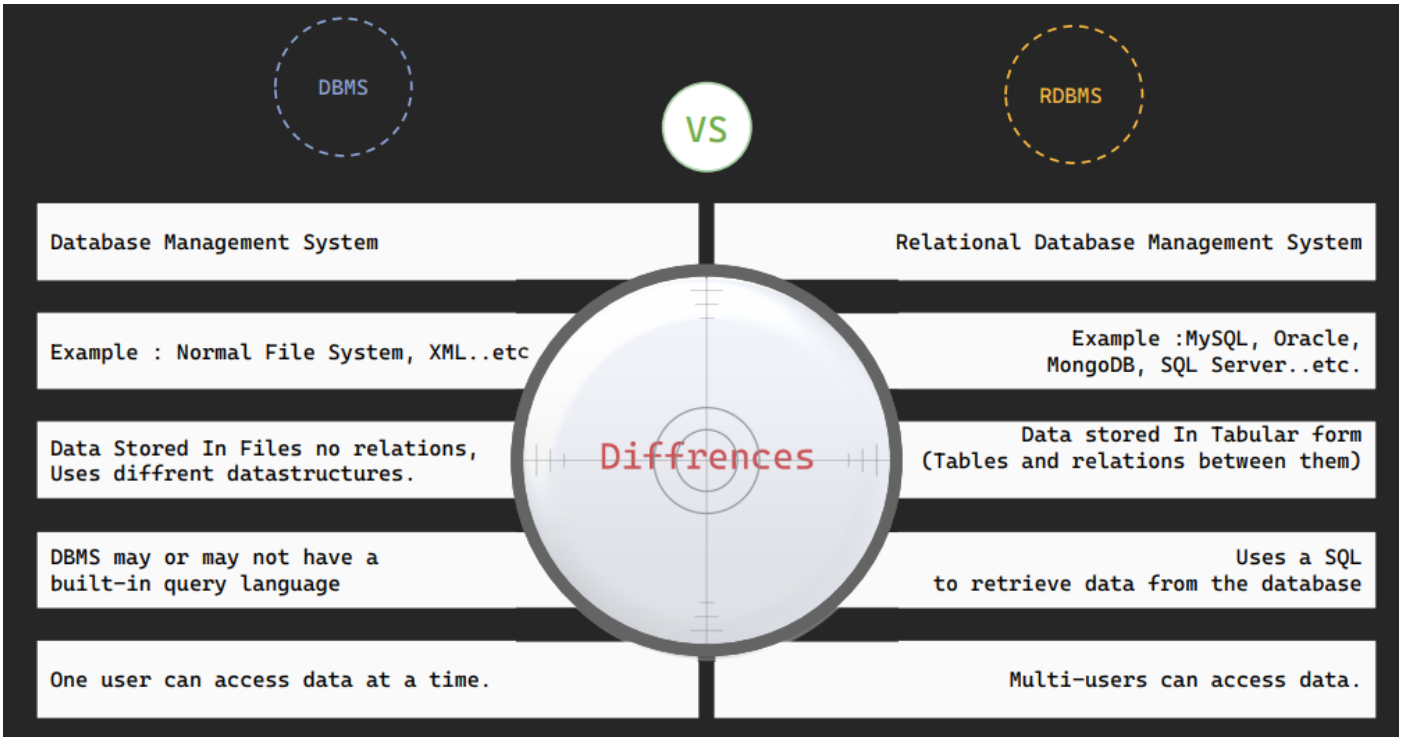


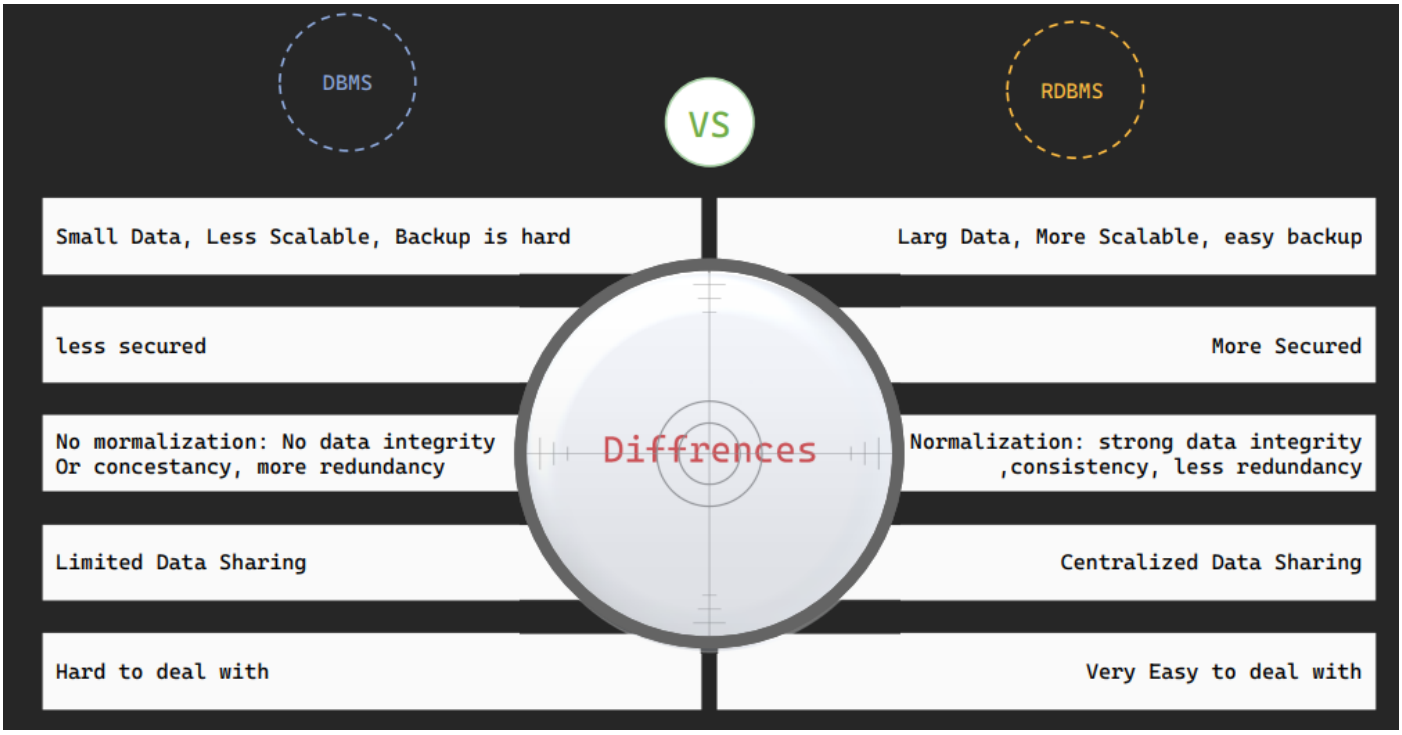
Employees Table

ID	FirstName	LastName	Gender	Birthdate	Salary	DepartmentID
1	Mohammed	Abu-Hadhoud	M	6/11/1977	5000	1
2	Ali	Zaher	M	5/7/1990	3000	2
3	Lubna	Aqel	F	5/5/2000	500	1
5	Fadi	Khalil	M	6/6/1980	1400	4
6	Maha	Majed	M	7/7/2001	300	3
7	Omar	Ali	M	6/6/1977	2000	1
8	Huda	Omar	F	4/2/1990	1000	1

Departments Table

ID	Name	Location
1	IT	Amman
2	Finance	Amman
3	HR	UAE
4	Marketing	Qatar





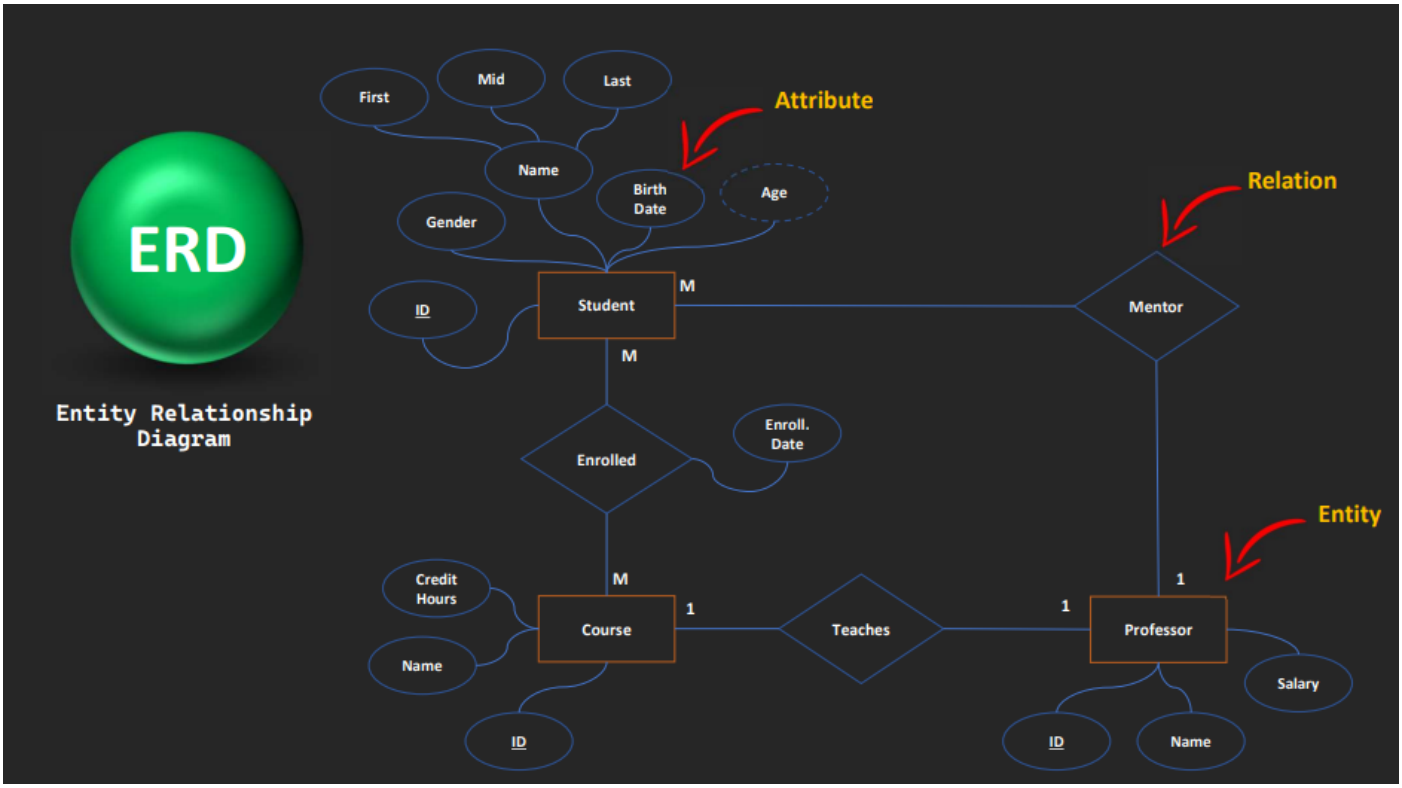
Database Design: Conceptual Design

What is ERD? and Why?

هنا بيقولك انه الديزاين بتاع الداتا بيز اهم حاجه لانك لو عامله غلط او مش زي مالعميل عايزه وبدأت تبني عليه البرنامج بتاعك هتكون اهدرت وقت وتكاليف وجهد عالفاضي

فعشان تعمل ديزاين مبدأي او conceptual design بتستخدم حاجه اسمها ERD diagram وهوا اختصار ل entity relationship diagram او ER Model وبيكون زي مخططات المباني

وده شكله



من خلال النظر لل diagram ده بتعرف انه هيكون عندك 3 جداول في الداتا بيز هما student و course و professor والعلاقات بين الجداول وكل جدول فيه انهي داتا

فانت لازم تحلل متطلبات النظام كلها قبل ماتبدأ في البرنامج بتاعك وبتقعد تعدل عليه لحد ماتوصل للديزاين الصحيح

What is ERD?

- An Entity Relationship Diagram (ER Diagram) pictorially explains the relationship between entities to be stored in a database.
- Fundamentally, the ER Diagram is a structural design of the database.
- It acts as a framework created with specialized symbols for the purpose of defining the relationship between the database entities.
- ER diagram is created based on three principal components: entities, attributes, and relationships.

What is an ER Model?

- An Entity-Relationship Model represents the structure of the database with the help of a diagram.
- ER Modelling is a systematic process to design a database as it would require you to analyze all data requirements before implementing your database.

Why Use ER Diagrams in DBMS?

- ER Diagram helps you conceptualize the database and lets you know which fields need to be embedded for a particular entity.
- ER Diagram gives a better understanding of the information to be stored in a database.
- It reduces complexity and allows database designers to build databases quickly.
- It helps to describe elements using Entity-Relationship models.
- It allows users to get a preview of the logical structure of the database.

Conclusion

- ER Diagram in RDBMS is widely used to describe the conceptual design of databases.
- It helps both users and database developers to preview the structure of the database before implementing the database.

الواجب

An Entity Relationship Diagram (ER Diagram) pictorially explains the relationship between entities to be stored in a database.

True

False

ER Diagram is a physical design of the database.

True

False

ER Diagram is a structural design of the database.

True

False

ER Diagram is a conceptual design of the database.

True

False

ER diagram is created based on three principal components: entities, attributes, and relationships.

True

False

An Entity-Relationship Model represents the structure of the database with the help of a diagram.

True

False

ER Modelling is a systematic process to design a database as it would require you to analyze all data requirements before implementing your database.

True

False

The cost of updating ERD is much cheaper than the cost of updating Database after you build it.

True

False

ER Diagram helps you conceptualize the database and lets you know which fields need to be embedded for a particular entity.

True

False

ER Diagram gives a better understanding of the information to be stored in a database.

True

False

ERD reduces complexity and allows database designers to build databases quickly.

True

False

ER Diagram in RDBMS is widely used to describe the conceptual design of databases.

True

False

ERD helps both users and database developers to preview the structure of the database before implementing the database.

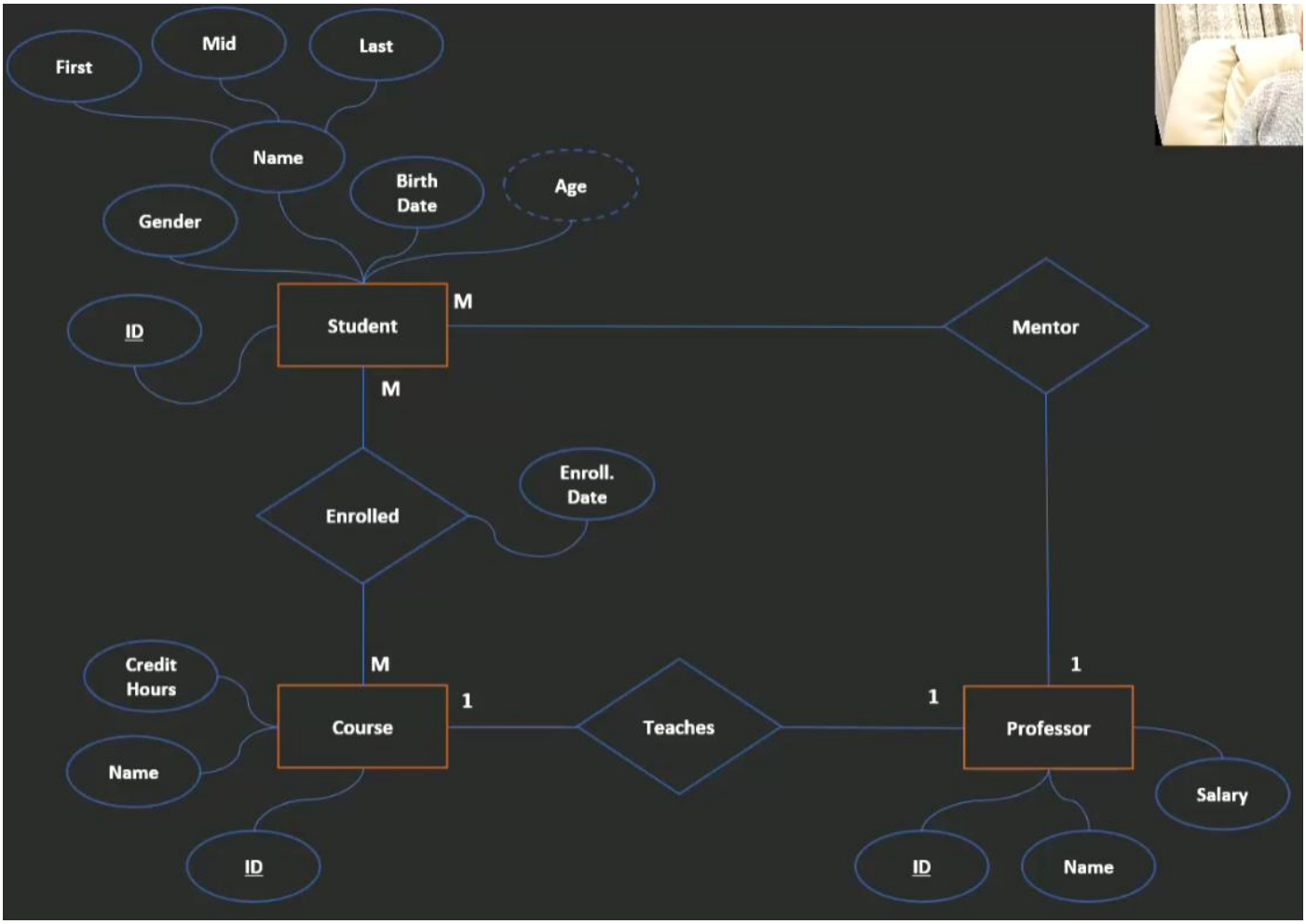
True

False

ERD Symbols

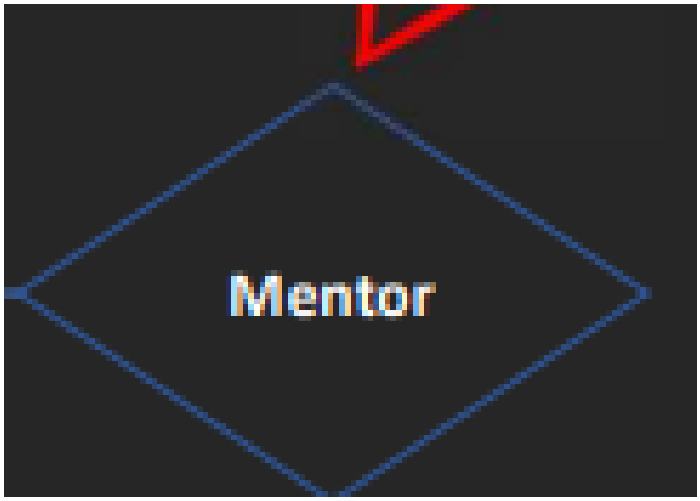
زي ال flow chart كان بيتكون من رموز كانت ليها معاني بتساعدك في حل ال algorithms هنا ال erd ليها برضه رموز او symbols عشان تعرف تبنيها

بص كده عال diagram ده

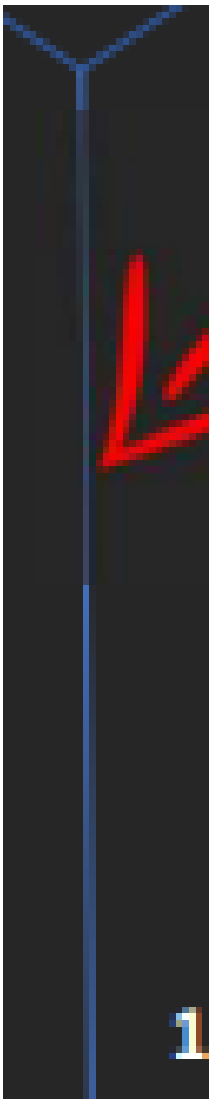


هتلاقي أي entity بتمثلها في مستطيل وبتعرف ان كنت محتاج تخزن الداتا دي علي الجهاز لو محتاج يبقي الداتا entity وبيكون اسمها strong entity لانه ليه primary key ولو ملوش primary key بيكون اسمه weak entity وبيكون مستطيلين جوا بعض وده لاينصح باستخدامه

وبعدين بتسال هل فيه علاقه بين entity و entity تانيه لو كانت الاجابه اه يبقي بتعمل relationship بينهم ودي بتديها الشكل ده اللي بيمثل العلاقه نفسها



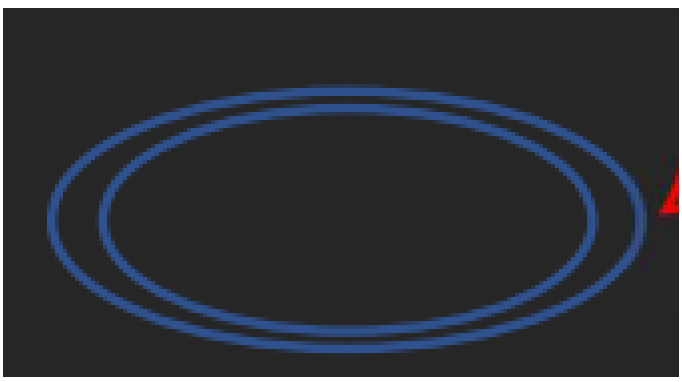
وعشن تربط entity بالتانيه بتوصلهم بخط



والعلاقات دي فيه منها أنواع هنشرحها بعدين

قالك بعد كده انه كل entity او صف بيكون فيه عدة اعمده او attributes ودي بتمثلها في شكل بيضاوي

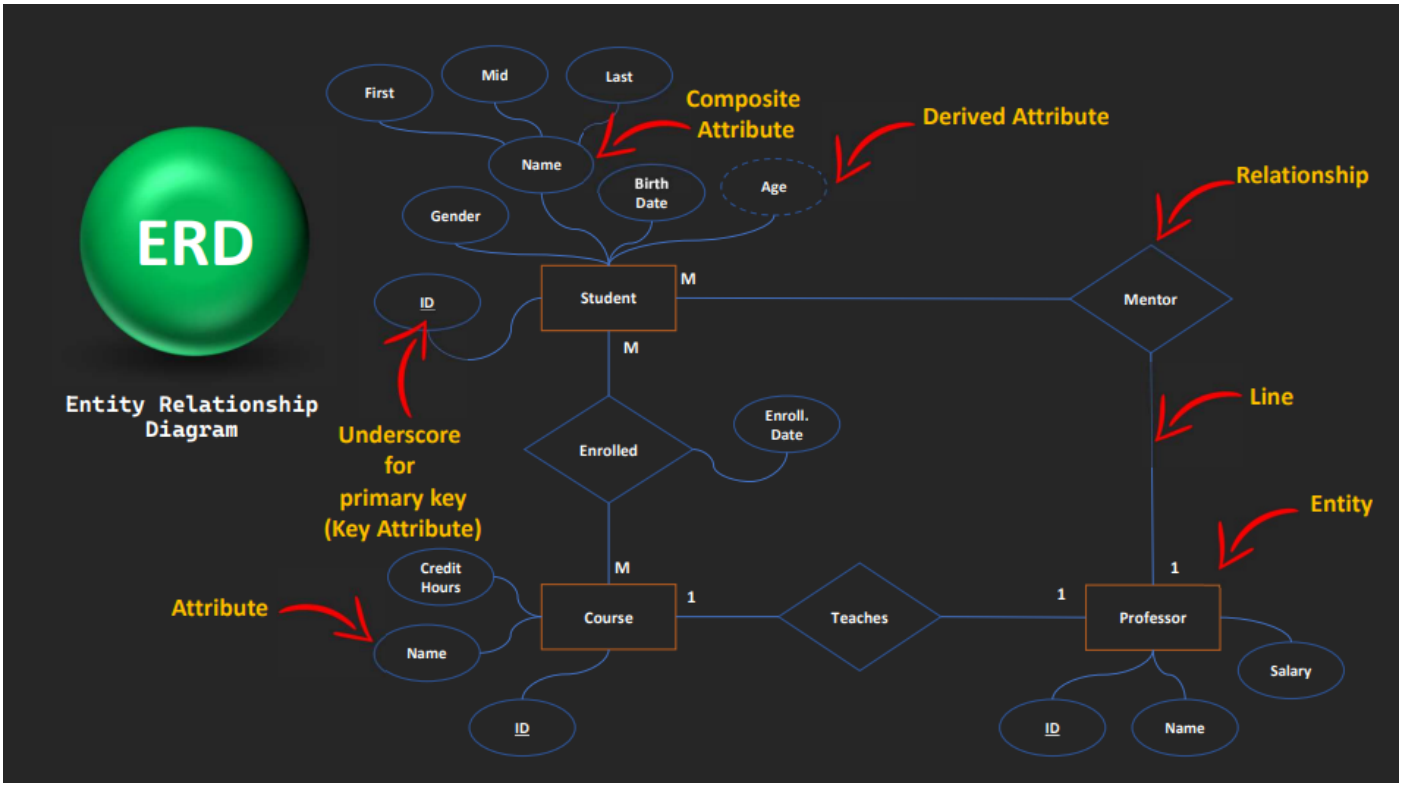
فيه حاجه اسمه multi valued attribute وده bad practice وهوا عباره عن انك تستخدم field عشان يمثل اكتر من قيمه وبيكون شكلين بيضاويين جوه بعض

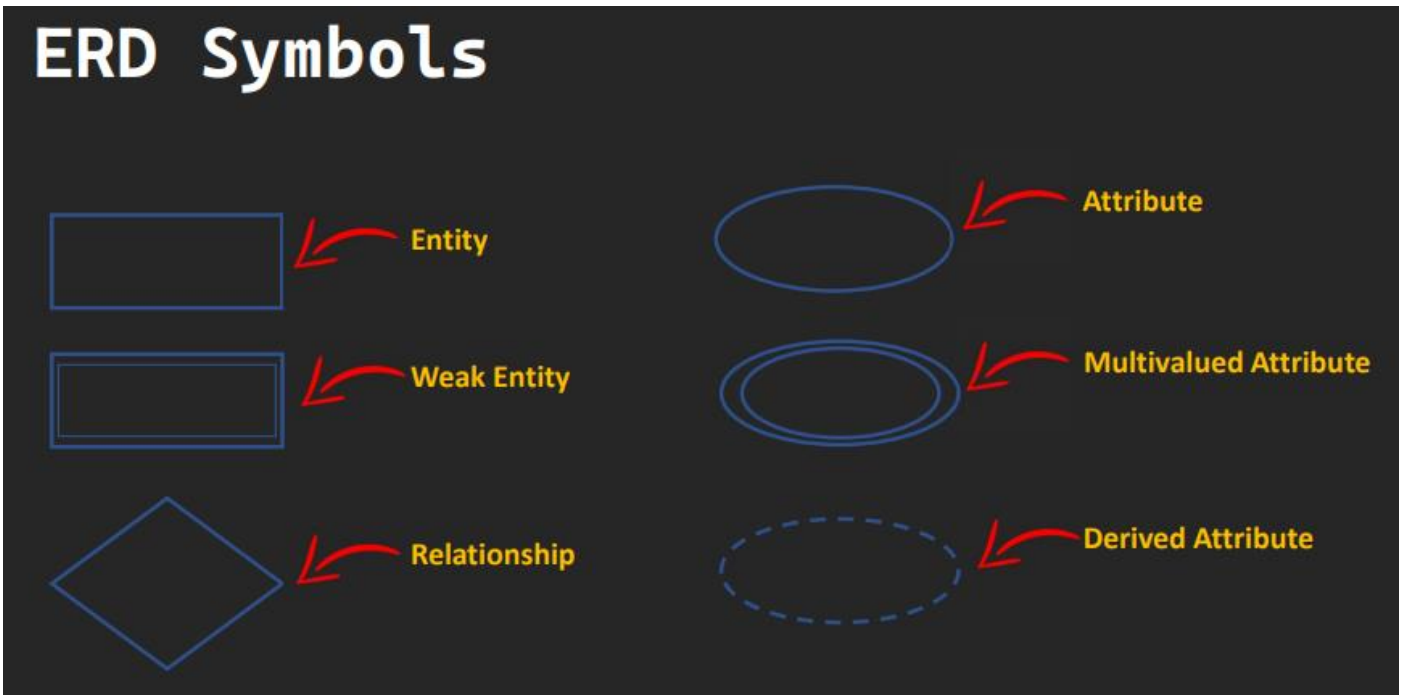


ولو فيه attribute بينقسم لعدة انوع بيتسمي composite attribute زي كده

وفيه حاجه اسمها composite attribute او derived attribute وده عباره عن attribute عادي بس مش بتخزن فيه داتا بس بيتم اشتقاقه من اعمده تاني زي مايكون عندك تاريخ الميلاد وعايز تعمل عمود للعمر انت هنا مش محتاج تخزن العمر لانك مجرد ماتطرح تاريخ اليوم من تاريخ الميلاد هيكون عندك العمر وده بيتمثل في شكل بيضاوي بس بخط متقطع

أي attribute موجود تحتيه خط ده بيكون primary key


ERD
Entity Relationship Diagram
First
Mid
Last
Composite Attribute
Name
Birth Date
Age
Derived Attribute
Gender
Relationship
ID
Student
M
Mentor
M
Underscore for primary key (Key Attribute)
Enrolled
Enroll. Date
Line
Entity
Credit Hours
M
1
1
Attribute
Course
1
Teaches
Professor
Name
Salary
ID
ID
Name


ERD Symbols
Entity
Attribute
Weak Entity
Multivalued Attribute
Relationship
Derived Attribute

Symbols Used in ER Diagrams:

- Rectangles: This Entity Relationship Diagram symbol represents entity types
- Ellipses: This symbol represents attributes
- Diamonds: This symbol represents relationship types
- Lines: It links attributes to entity types and entity types with other relationship types
- Primary key: Here, it underlines the attributes
- Double Ellipses: Represents multi-valued attributes

الواجب

Which ERD symbol represents entity types?

- Ellipses
- Rectangles
- Diamonds
- Double Ellipses

Which ERD symbol represents attributes?

- Rectangles
- Diamonds
- Ellipses
- Lines

Which ERD symbol represents relationship types?

- Diamonds
- Ellipses
- Rectangles
- Double Ellipses

Which ERD symbol links attributes to entity types and entity types with other relationship types?

- Rectangles
- Ellipses
- Diamonds
- Lines

How to represent primary key in ERD?

- Underline the attributes name
- Double Ellipses

Strong Entity is the entity that has primary key(s).

- True
- False

Weak Entity is represented by double rectangles because it has no primary key for it and it depends on other entities to be identified.

True

False

Components of ERD

هنا بيلملك الموضوع عشان يكون منظم اكتر وبيقولك انه بينقسم ل 3 أجزاء رئيسيه ودي التقسيمه بتاعتهم

Components of ER Diagram:

We base an ER Diagram on three basic concepts:

- Entities
  - Entity (Strong Entity)
  - Weak Entity
- Attributes
  - Attribute
  - Key Attribute
  - Composite Attribute
  - Multivalued Attribute
  - Derived Attribute
- Relationships
  - One-to-One Relationships
  - One-to-Many Relationships
  - Many-to-One Relationships
  - Many-to-Many Relationships

Entity (Strong) and Weak Entity

قولنا قبل كده انه ال entity نوعين ياما strong ياما weak والفرق بينهم بيكون انه ال strong entity فيها primary key انم ال weak entity مافيها primary key وقولنا انه ال entity بيمثل صف في الجدول بيعبر عن بيانات عن أي حاجه وبيتم تمثيله في شكل مستطيل

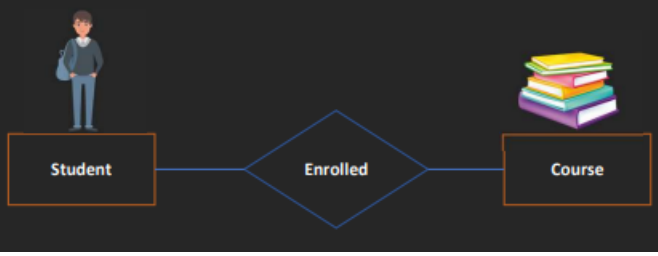
ال weak entity بيتم تعريفها او استدعاءها عن طريق ال foreign key وانه يكون مربوط ب entity تانيه ليها primary key

Entities

- Entity (Strong Entity).
- Weak Entity.

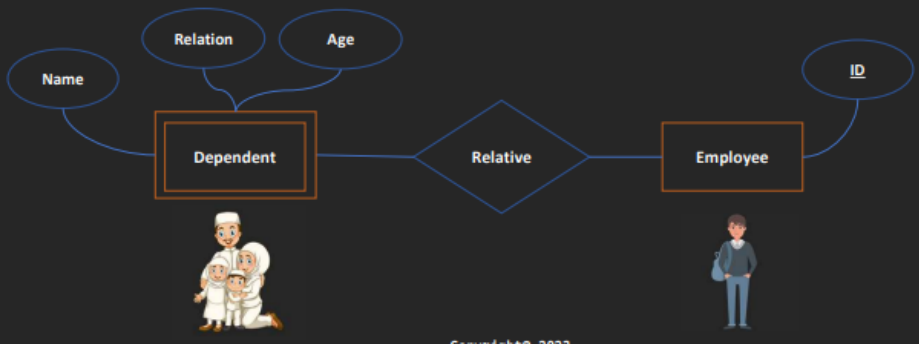
Entities(Strong Entities)

- An entity can be either a living or non-living component.
- It showcases an entity as a rectangle in an ER diagram.
- Each Entity has a primary key
- For example, in a student study course, both the student and the course are entities.



Weak Entities

- An entity that makes reliance over another entity is called a weak entity
- You showcase the weak entity as a double rectangle in ER Diagram.
- Weak Entity has no primary key.
- In the example below, Employee is a strong entity because it has a primary key attribute – ID. Unlike Dependent, the dependent is a weak entity because it does not have any primary key and the name here acts only as a discriminator.



الواجب

There are two types of entities, strong entities and weak entities

True

False

Strong Entity does not have primary key.

True

False

Strong Entity has to have primary key.

True

False

Weak Entity has no primary key(s)

True

False

An entity can be either a living or non-living component.

True

False

Which of the following assures the Entity Integrity.

Strong Entities

Weak Entities

Strong Entity is represented by double rectangles in ERD.

True

False

Strong Entity is represented by single rectangle in ERD.

True

False

Weak Entity is represented by double rectangles in ERD.

True

False

Attributes

زي ماالحنا عارفين انه كل جدول مكون من صفوف واعمده والصفوف بنسميها entities وبتعبر عن object او شيء معين

الاعمده بقي بيكون اسمها attributes او columns او fields وكل عمود بيمثل خاصيه او صفه للشئ الممثل في شكل صف او entity

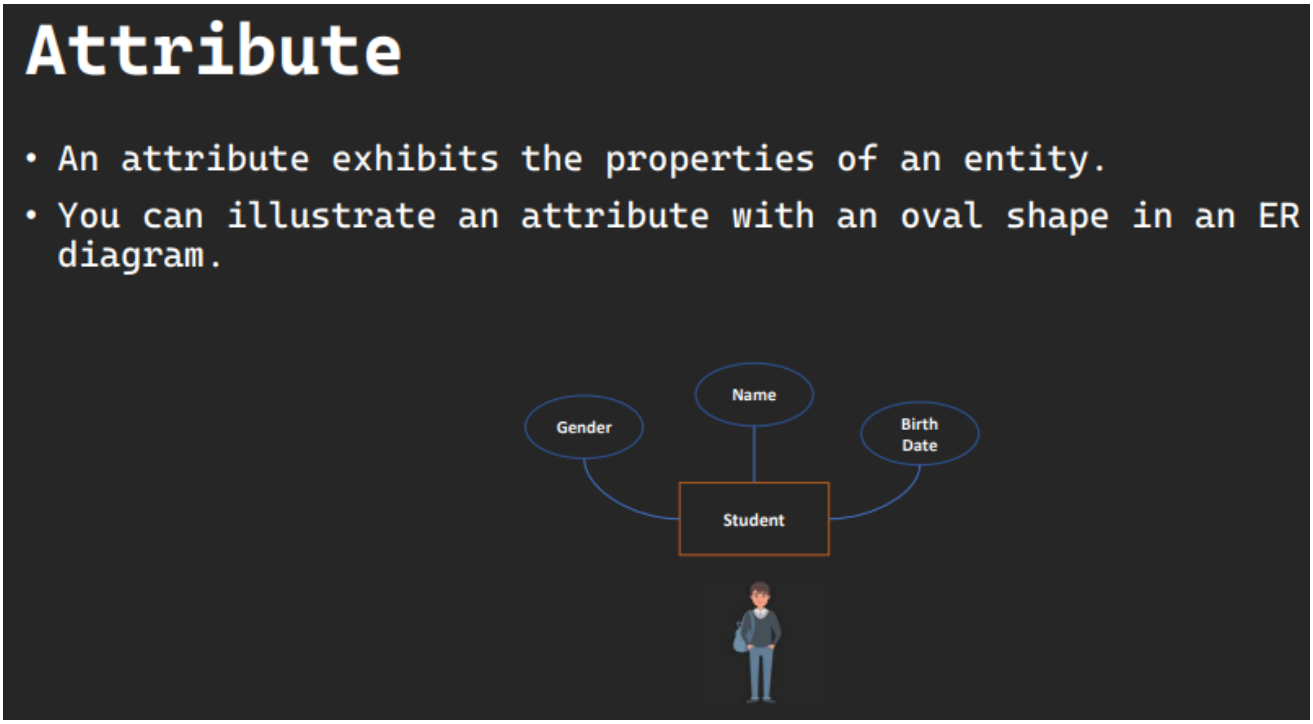
وال attributes بتمثل في شكل بيضاوي في ال erd

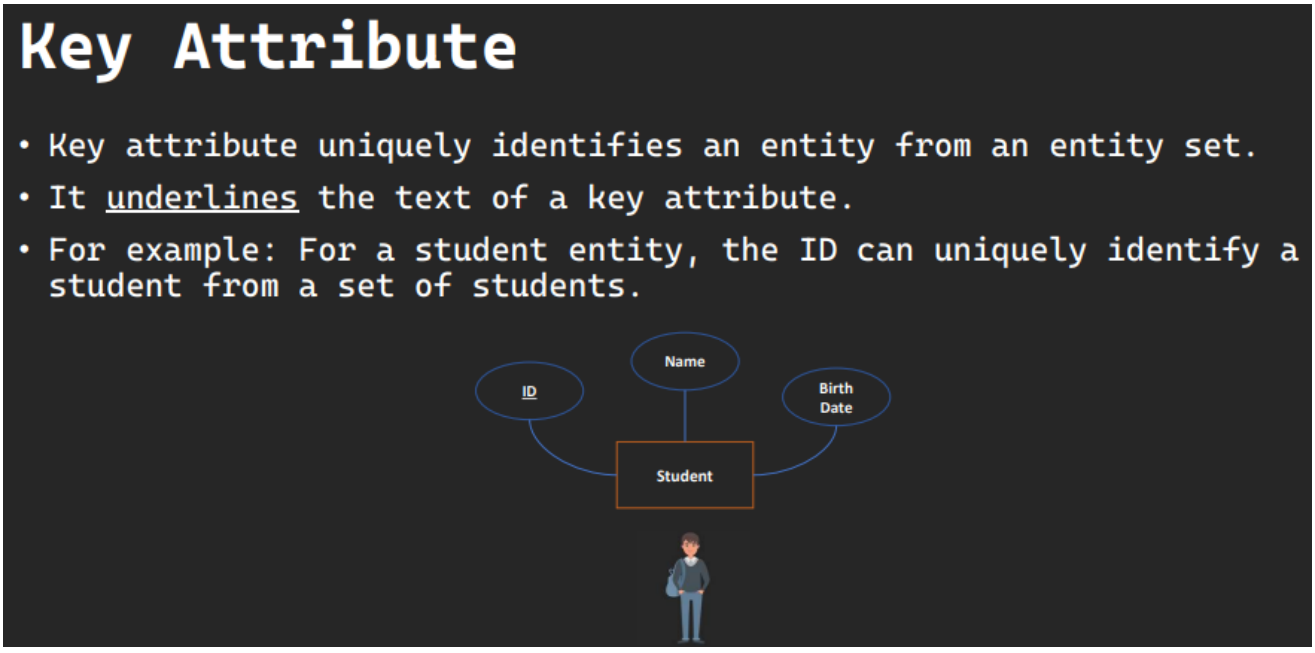
ال key attribute هوا عمود بيكون primary key وبيتم تمثيله في ال erd في شكل بيضاوي وبتحط خط تحت اسمه

ال composite attribute هوا عباره عن attribute عادي بس مركب بيتكون من عدة attributes تانيه وبيتم تمثيله بشكل بيضاوي عادي وطالع منه attributes تانيه

ال multivalued attribute بيمثل بشكلين بيضاويين جوا بعض وده بيعبر عن عدة قيم في وقت واحد زي مثلا انه يتم تخزين عدة ارقام تيليفونات لنفس الشخص في عمود واحد وده شيء مش كويس

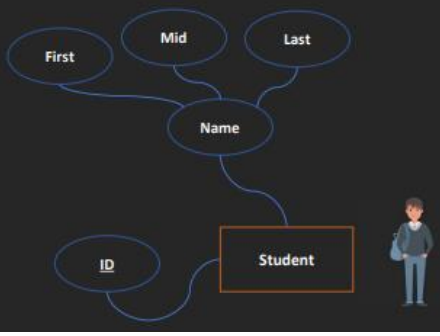
ال derived attribute وده بيمثل بشكل بيضاوي متقطع وده مش داتا بتتخزن لا ده بيعتمد علي عمود تاني وبيجيبلك منه معلومات تانيه





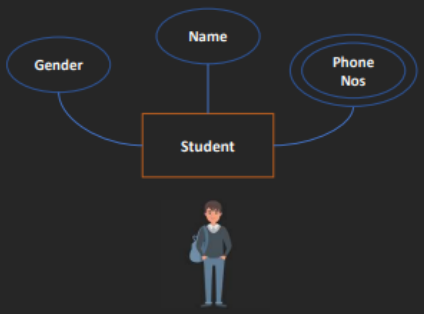
Composite Attribute

- An attribute that is composed of several other attributes is known as a composite attribute.
- An oval showcases the composite attribute, and the composite attribute oval is further connected with other ovals.



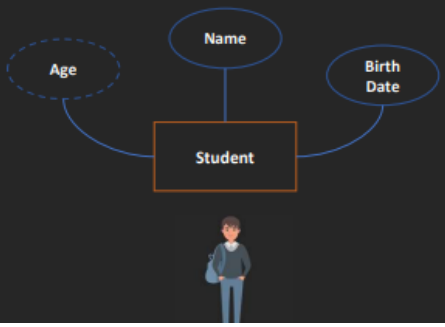
Multivalued Attribute

- Some attributes can possess over one value, those attributes are called multivalued attributes.
- The double oval shape is used to represent a multivalued attribute.



Derived Attribute

- An attribute that can be derived from other attributes of the entity is known as a derived attribute.
- In the ER diagram, the dashed oval represents the derived attribute.



الواجب

An attribute exhibits the properties of an entity.

True

False

You can illustrate an attribute with an oval shape in an ER diagram.

True

False

Key attribute uniquely identifies an entity from an entity set.

True

False

Key Attribute is represented by underlining the text of a key attribute.

True

False

An attribute that is composed of several other attributes is known as a composite attribute.

True

False

An oval showcases the composite attribute, and the composite attribute oval is further connected with other ovals.

True

False

Some attributes can possess over one value, those attributes are called multivalued attributes.

True

False

The double oval shape is used to represent a multivalued attribute.

True

False

It's not recommended to have multivalued attributes.

True

False

An attribute that can be derived from other attributes of the entity is known as a derived attribute.

True

False

In the ER diagram, the dashed oval represents the derived attribute.

True

False

At the end the attribute is a field or column in the database.

True

False

Relationships

ال relationships او العلاقات من اهم الحاجات اللي لازم تحددها صح في الداتا بيز بعد ما بتحدد ال entities بتيجي خطوة تحديد ال relationships

باختصار انت بتسال هل فيه علاقه بين الجدولين ولا لا لو فيه بترسم معين وبتحط فيه ايه هيا العلاقه وبعد كده بترسم خط بيمشي من اول جدول بيروح للعلاقه وبينتهي عند الجدول التاني

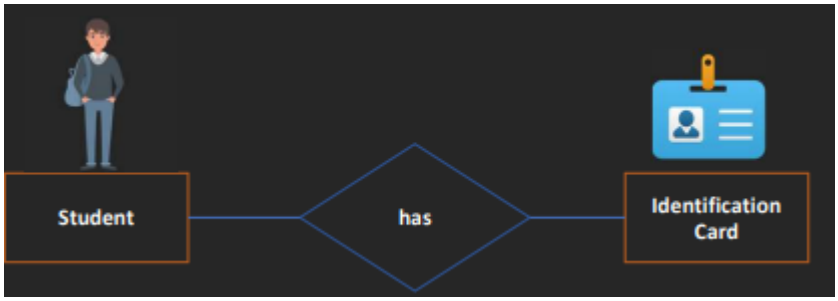
العلاقات فيه منها انوع ومنها العلاقه بين الشئ ونفسه او علاقتين بين شيء و شئ اخر

Relationship

- The diamond shape showcases a relationship in the ER diagram.
- It depicts the relationship between two entities.
- In the example below, both the student and the course are entities, and Enrolled is the relationship between them.

Relationship

- In the example below, both the student and the Identification-Card are entities, and “has” is the relationship between them.



Relationship

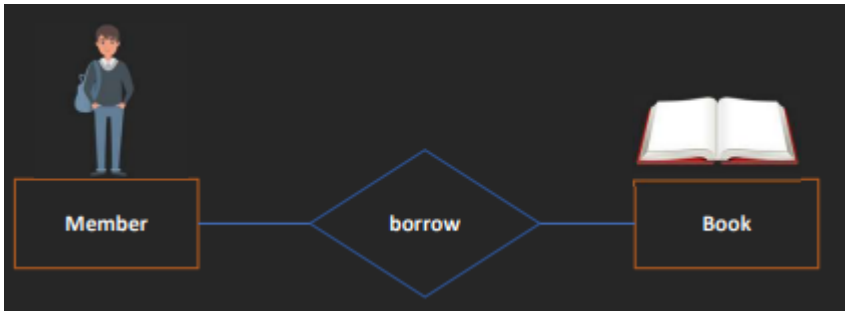
- In the example below, both the customer and the order are entities, and “Place” is the relationship between them.

Relationship

- In the example below, both the customer and the order are entities, and “Place” is the relationship between them.
- Order can have more than one product.

Relationship

- In the example below, both the Member and the Book are entities, and “borrow” is the relationship between them.

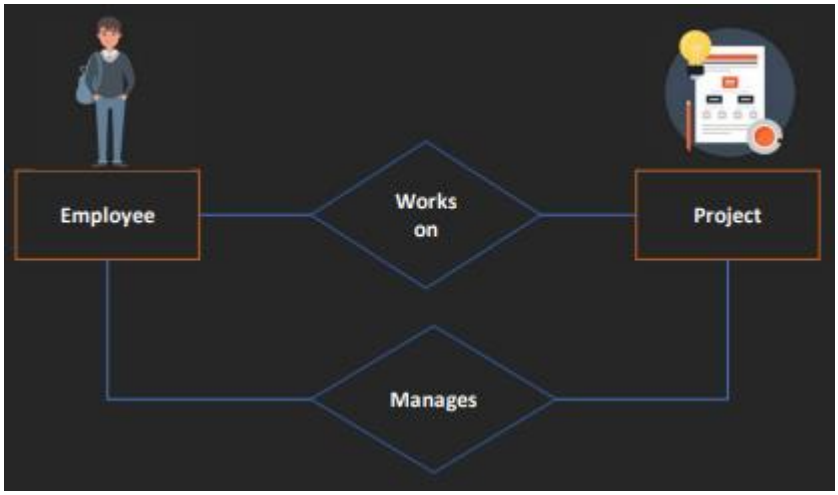


Relationship

- In the example below, both the Customer and the Car are entities, and “Rent” is the relationship between them.

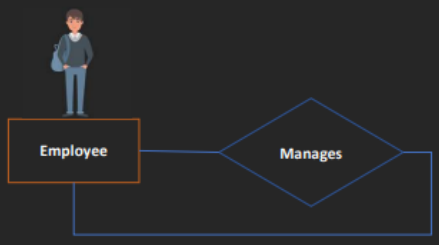
Relationship

- In the example below, both the Employee and the Project are entities, and “Works on” is the relationship between them.
- Also Employee manages projects.



Self Referencing Relationship

- In the example below, an employee has only one manager and manager is employee.
- It depicts the relationship between one entity.
- When an element of an entity is associated with a an element of same entity, it is called self relationship.



Employees

ID (PK)	Name	Salary	ManagerID (FK)
1	Ahmed	5000	Null
2	Amjad	1500	1
3	Maher	1300	2
4	Alia	500	2

وهنا بيقولك انه العلاقات أنواع ودي انواعها

Relationship Types

- One-to-One Relationship.
- One-to-Many Relationship.
- Many-to-One Relationship.
- Many-to-Many Relationship.

الواجب

Relationship is always between two entities.

True

False

Relationship can be on one entity and we call it Self Reference Relationship.

True

False

We can have only one relationship between two entities.

True

False

We can have more than one relationship between two entities.

True

False

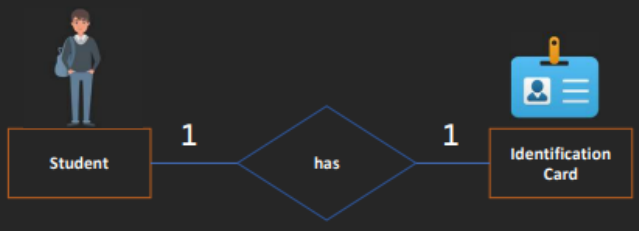
One-to-One Relationship

هنتكلم عن علاقة ال one to one وهيا انه كل entity بيمتلك او بيكون ليه علاقه مع record واحد بس من ال entity التانيه ماينفعش يكون ليه علاقه مع اكتر من record زي الانسان ماينفعش يكون ليه اكتر من رقم قومي

والعلاقه دي بتمثل بانك تكتب رقم واحد جنب طرفي العلاقه بص في الأمثلة الجايه

One-to-One Relationship

- In the example below, both the Student and the Identification-Card are entities, and “has” is the relationship between them.
- For example, a student has only one identification card and an identification card is given to one person.



One-to-One Relationship

- In the example below, both the Student and the Person are entities, and “is a” is the relationship between them.
- In the example below, both the Employee and the Person are entities, and “is a” is the relationship between them.

Copyright© 2023

One-to-One Relationship

- In the example below, both the Member and the Book are entities, and “borrow” is the relationship between them.
- Member can borrow only one book, and book can be borrowed by one member.

One-to-One Relationship

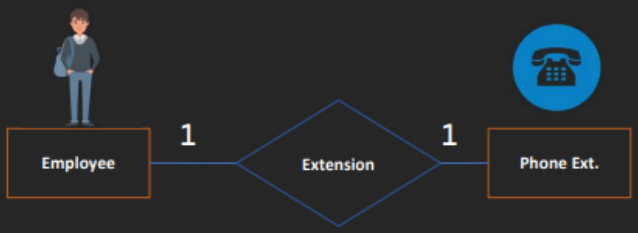
- In the example below, both the Traveler and the Seat are entities, and “Set” is the relationship between them.
- Traveler can set on only one seat, and seat can be assigned to one traveler.

One-to-One Relationship

- In the example below, both the Employee and the Desk are entities, and “Set” is the relationship between them.
- Employee can set on only one Desk, and Desk can be assigned to one Employee.

One-to-One Relationship

- In the example below, both the Employee and the phone extension are entities, and “Extension” is the relationship between them.
- Each employee has a specific phone extension, which can only reach one employee.



One-to-One Relationship

- In the example below, both the Employee and the Task are entities, and “Assigned” is the relationship between them.
- For example, employee can work on a single Task, and a Task can be assigned to one employee.

One-to-One Relationship

- In the example below, both the citizen and the Car are entities, and “Own” is the relationship between them.
- Citizen can own only one car, a car can be owned by one citizen.

الواجب

When a single element of an entity is associated with a single element of another entity, it is called a one-to-one relationship

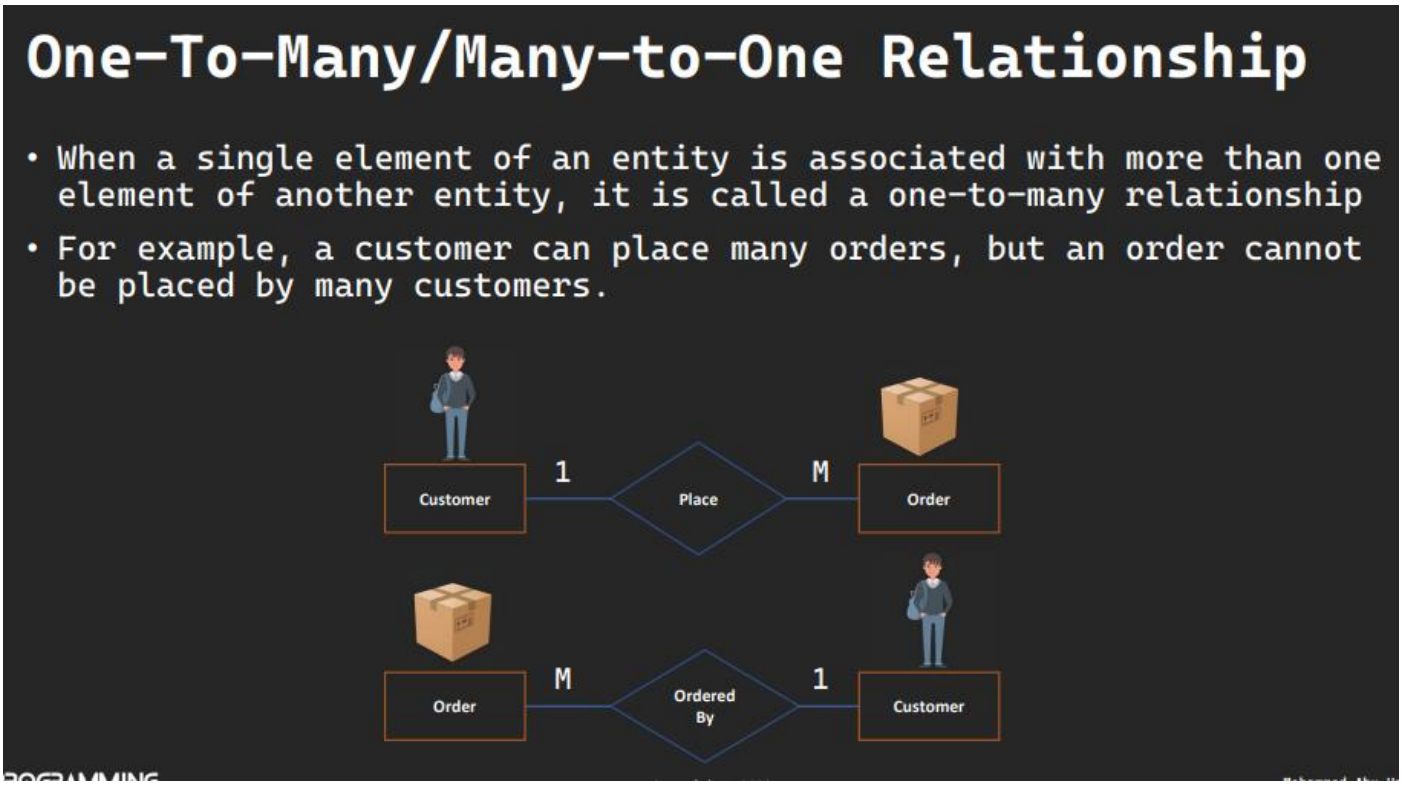
True

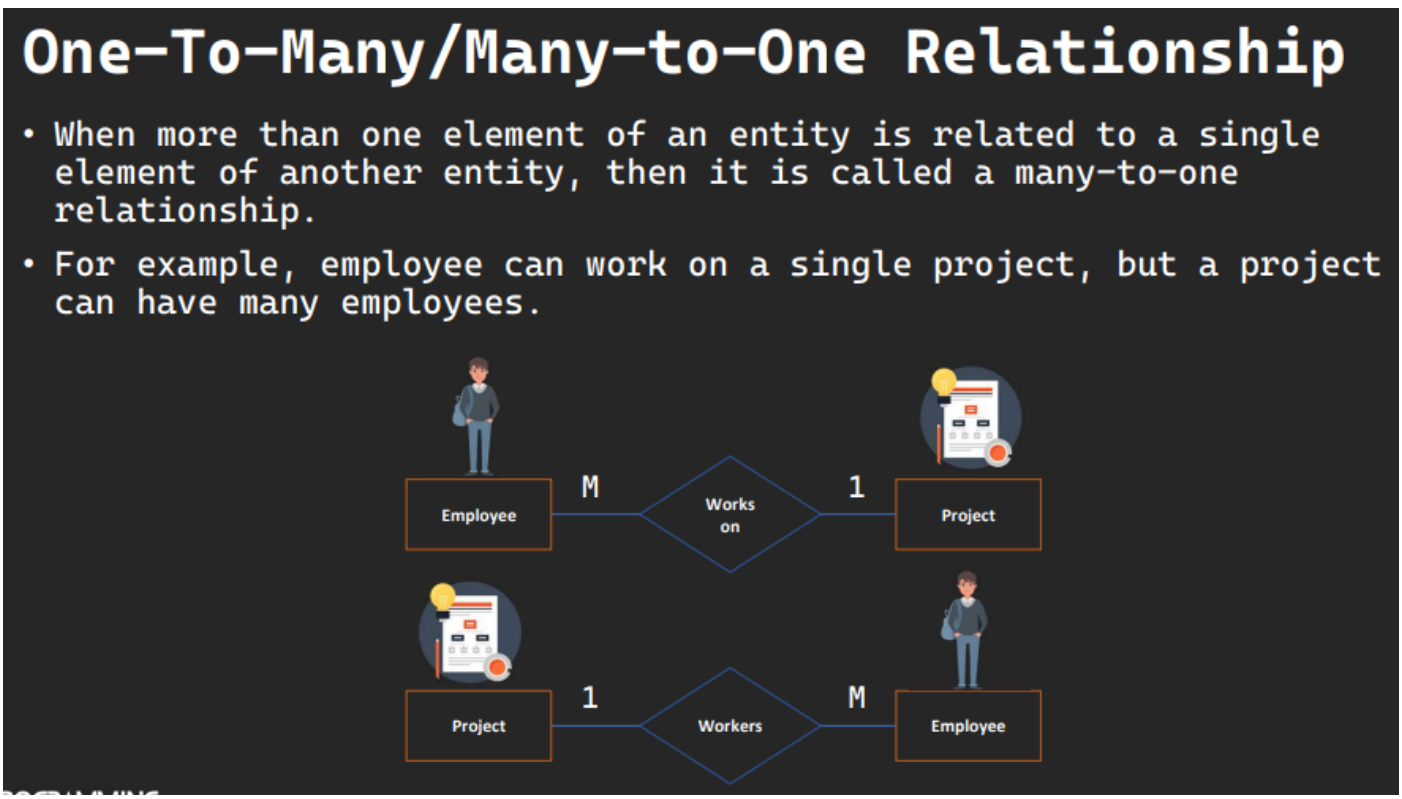
False

One-to-Many/Many-to-One Relationship

علاقة ال one to many وال many to one وهيا انه لما تكون entity معينه تقدر ترتبط باكتر من entity تانيه زي مدرس واحد بيدرس لعدة طلاب ودي بتمثل بان ال entity الأول اللي هوا المدرس بيتحط جنبها الرقم 1 وال entity التانيه اللي هيا الطلاب بيتحط جنبها حرف ال m وبتتقرا من الشمال لليمين

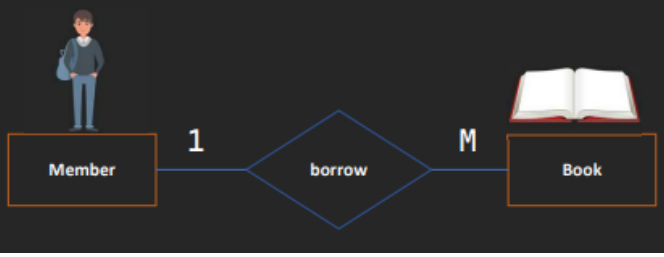
يعني لو كان المدرس عالشمال فبتقول one to many ولو كان المدرس عاليمين بتقول many to one او علي حسب انت بدأت منين





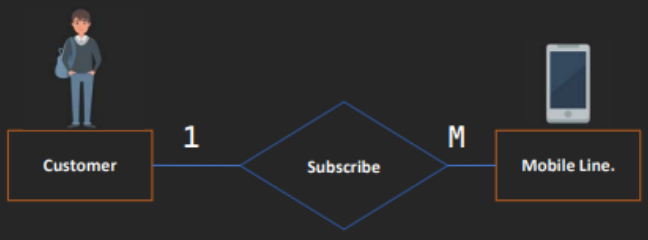
One-To-Many/Many-to-One Relationship

- In the example below, both the Member and the Book are entities, and “borrow” is the relationship between them.
- Member can borrow Many books, and book can be borrowed by one member.



One-To-Many/Many-to-One Relationship

- In the example below, both the Customer and the Mobile-Line are entities, and “Subscribe” is the relationship between them.
- Each Customer can subscribe to many lines, but each line can only be owned by one.

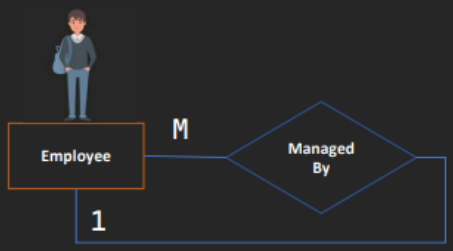


One-To-Many/Many-to-One Relationship

- In the example below, both the citizen and the Car are entities, and “Own” is the relationship between them.
- Citizen can own many cars, a car can be owned by one citizen.

One-To-Many/Many-to-One Relationship

- When an element of an entity is associated with a an element of same entity, it is called self relationship.
- For example, an employee has only one manager and manager can manage many employees.



Employees

ID (PK)	Name	Salary	ManagerID (FK)
1	Ahmed	5000	Null
2	Amjad	1500	1
3	Maher	1300	2
4	Alia	500	2

الواجب

When a single element of an entity is associated with more than one element of another entity, it is called a one-to-many relationship.

True

False

When more than one element of an entity is associated with a single element of another entity, it is called a many-to-one relationship.

True

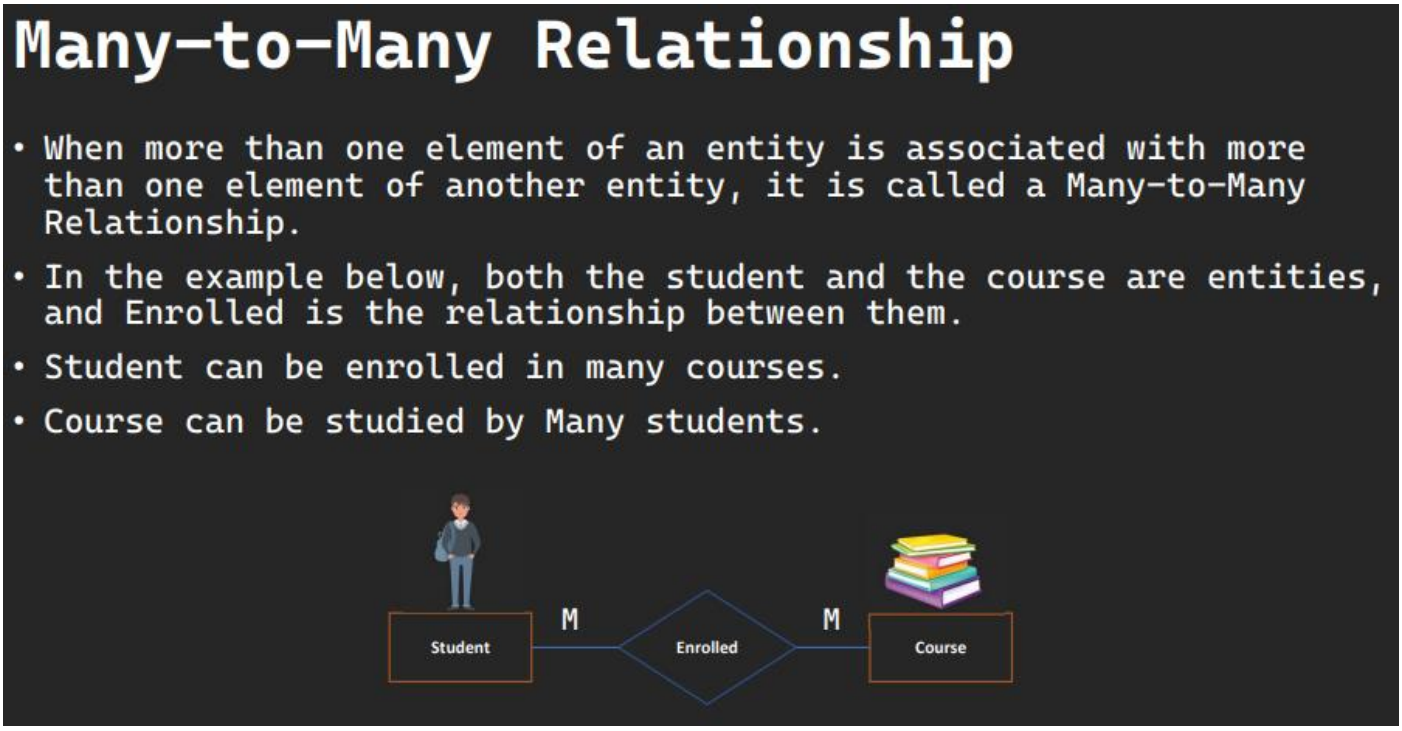
False

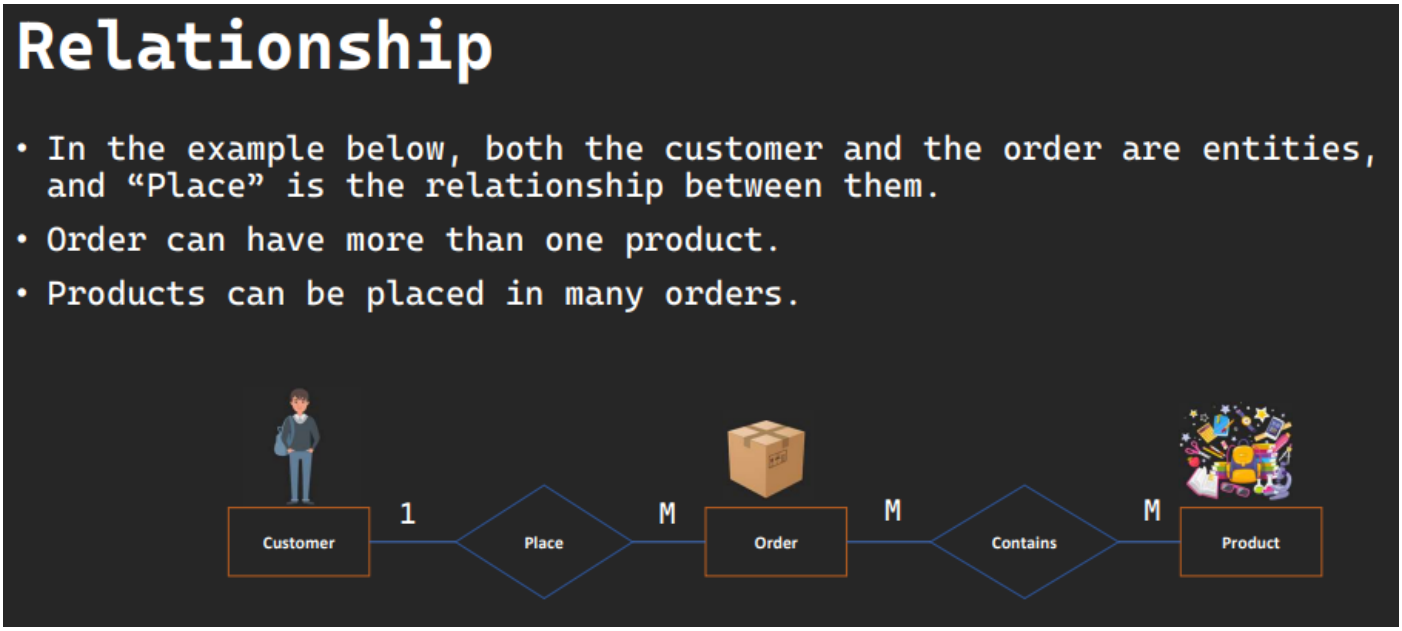
Many-to-Many Relationship

علاقة ال many to many وهيا بتكون لما تقدر تربط اكتر من entity باكتر من entity في جدول تاني مثلا اكتر من طالب يقدروا يشتركوا باكتر من كورس والكورس الواحد يقدر يشترك فيه اكتر من طالب وبتمثل العلاقه بكتابة ال m بجانب طرفي العلاقه

حتة انك تحدد ايه اكتر عدد ممكن يدخل في العلاقه ده اسمه cardinality

شوف الامثله





الواجب

When more than one element of an entity is associated with more than one element of another entity, it is called a Many-to-Many Relationship.

True

False

Cardinality vs Ordinality

لما كنا بنحدد نوع العلاقه قبل كده كنا بنسأل نفسنا ايه اكبر عدد من ال entities في الجدول التاني يقدروا يرتبطوا ب entity واحد في الجدول الأول والعكس

زي ايه عدد الكورسات اللي الطالب يقدر يشترك فيها وايه اكبر عدد من الطلاب يقدرو يشتركوا في الكورس الواحد

ده كده اسمه cardinality انك تسال ايه اكبر عدد مطلوب لتحقيق العلاقه بين جدولين

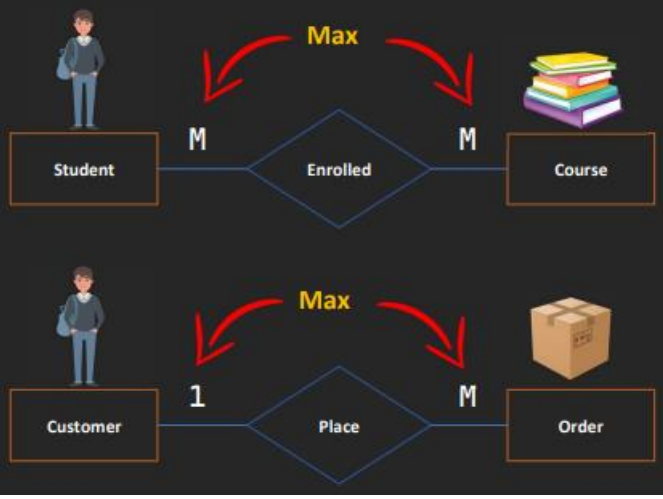
ال ordinality هوا العكس انك تسال عن اقل عدد من ال entities مطلوب لتحديد العلاقه

زي مثلا انه العميل اقل عدد من ال orders اللي المفروض يعملها هوا صفر لكن اقل عدد من العملاء اللي لازم يطلبوا ال order ده هوا واحد لانه الاوردر هوا record وطالما فيه اوردر اتعمل يبقي لازم يكون فيه عميل هوا اللي طلبه

ال ordinality وال cardinality ممكن يتكتبوا مع بعض زي كده (min,max) علي جانب طرفي العلاقه

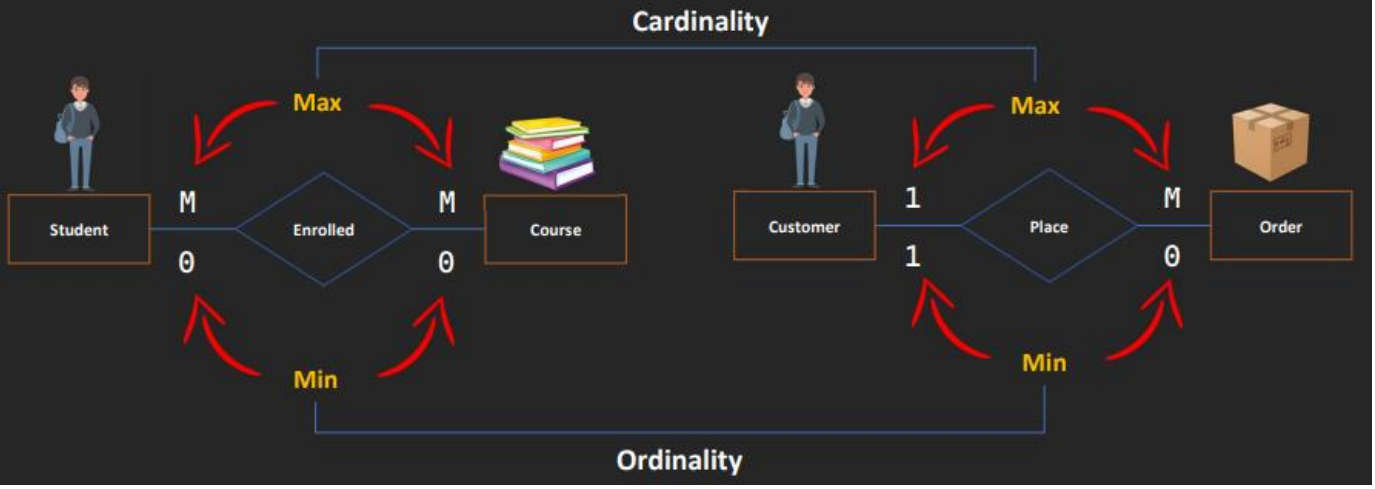
Cardinality

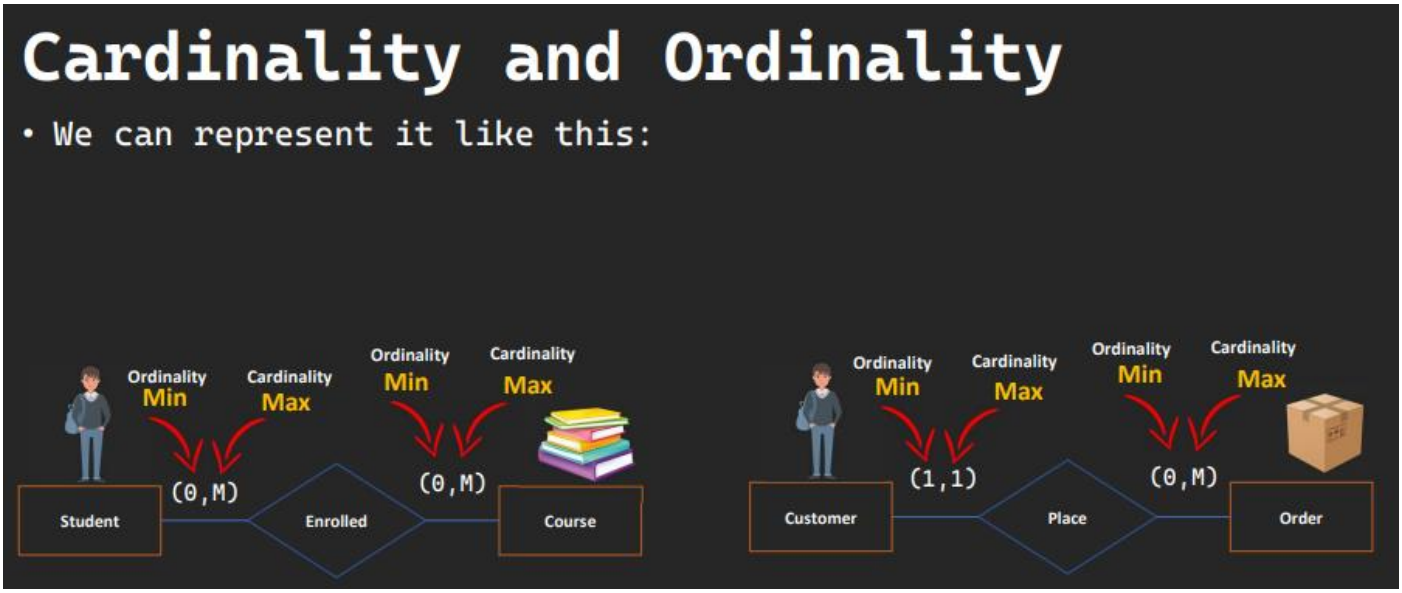
- Cardinality refers to the maximum number of times an instance in one entity can relate to instances of another entity



Ordinality

- Ordinality, on the other hand, is the minimum number of times an instance in one entity can be associated with an instance in the related entity. (in other words It specify if it's optional/mandatory/required or not).





الواجب

Cardinality refers to the maximum number of times an instance in one entity can relate to instances of another entity.

True

False

Ordinality, on the other hand, is the minimum number of times an instance in one entity can be associated with an instance in the related entity. (in other words It specifies if it's optional/mandatory/required or not).

True

False

(0,M) : the first number represents the cardinality and the second one represents the ordinality.

True

False

(0,M) : the first number represents the ordinality and the second one represents the cardinality.

True

False

When you ask "What is the Max?" you are identifying the cardinality.

True

False

When you ask "What is the Min?" you are identifying the ordinality.

True

False

Cardinality Symbols and Practices

هوا بيقولك انه لما حد يقولك cardinality هوا بيقصد الاتنين ال cardinality وال ordinality

وان ال ordinality موجوده عشان تحدد انه ال لازم يكون الطالب مشترك في كورس ولا لا يعني لازم تحدد entity من الجدول التاني ولو لاز كم عدد ال entities دي

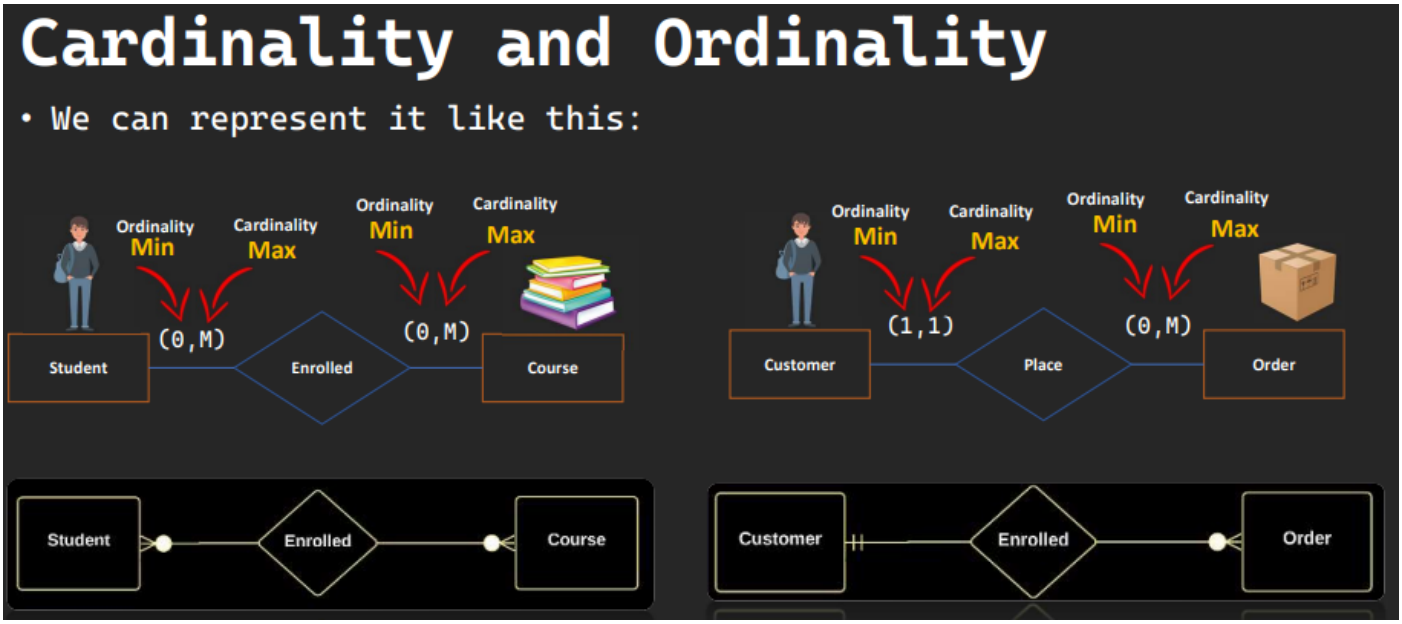
وكنا بنرمز لل cardinality بالطريقه دي (0,M) لكن بيقولك انه فيه sympols او رموز تانيه لتوضيح العلاقه زي دي مثلا

ال sympol دي اسمها crow-s-foot-notation يعني رجل الغراب احنا بنسميها رجل الفرخه

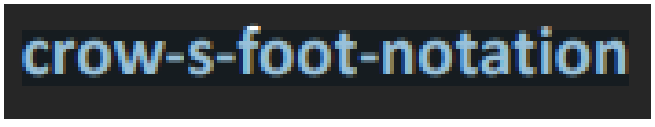
والدايرة بترمز للصفر  لانما الجزء ده  بيرمز لل m وده بيرمز للرقم واحد 

الجزء القريب من المستطيل او ال entity بيرمز لل cardinality او ال max انما الجزء البعيد بيرمز لل ordinality او ال min

بص كده للمثال ده

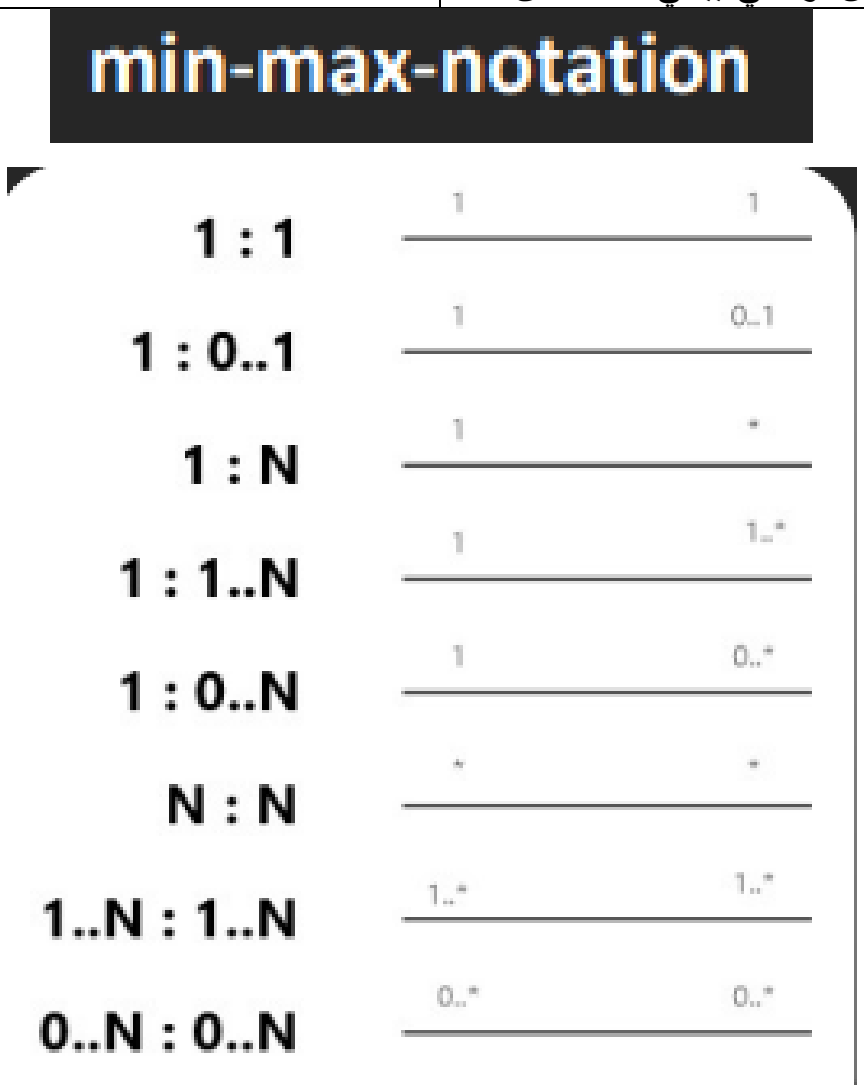


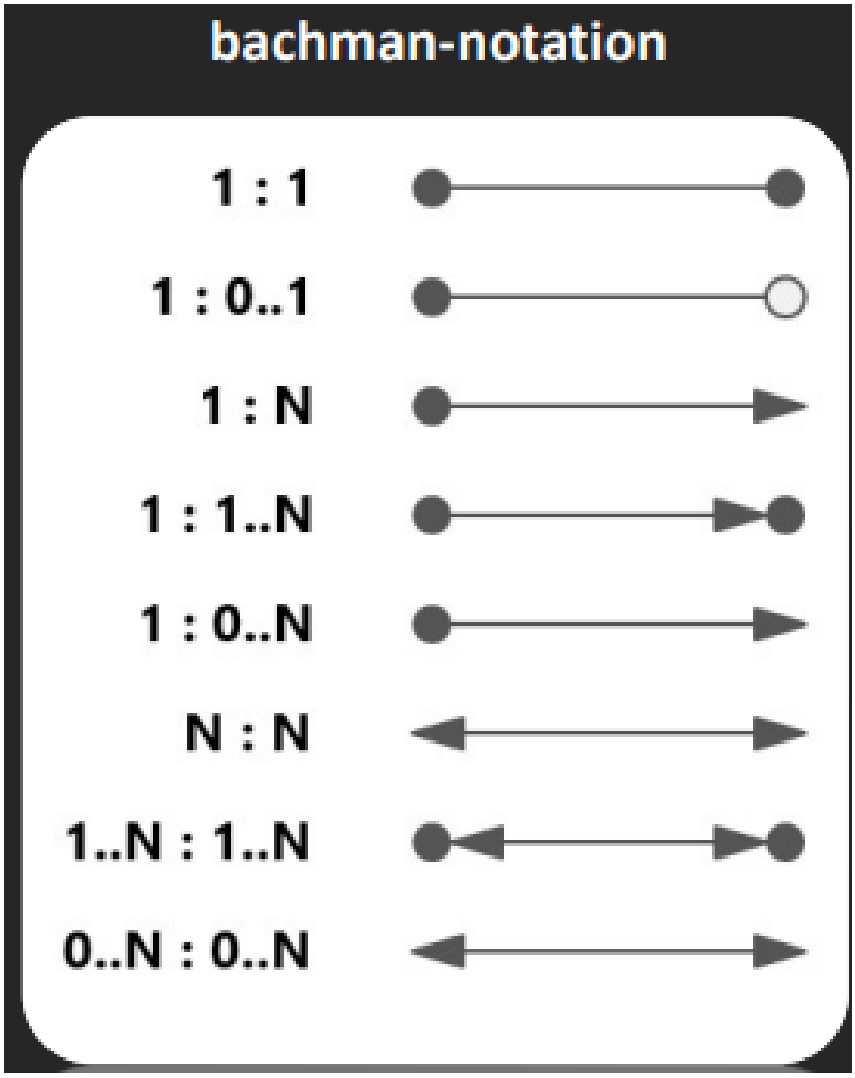
ودول 3 انوع من الرموز



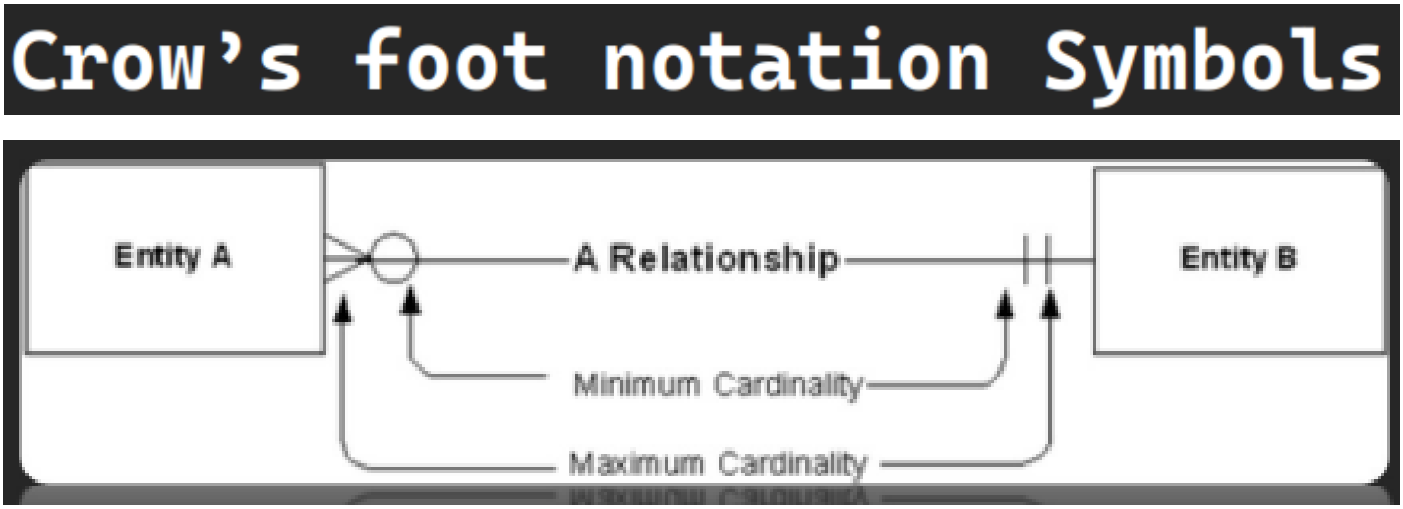
1 : 1 	One to One واحد بس من الجدول الأول بيتربط بواحد بس من الجدول التاني
1 : 0..1 	One to zero or One واحد بس من الجدول الأول بيرتبط مع واحد بس او صفر من الجدول التاني ال ordinality بصفر وال cardinality بواحد

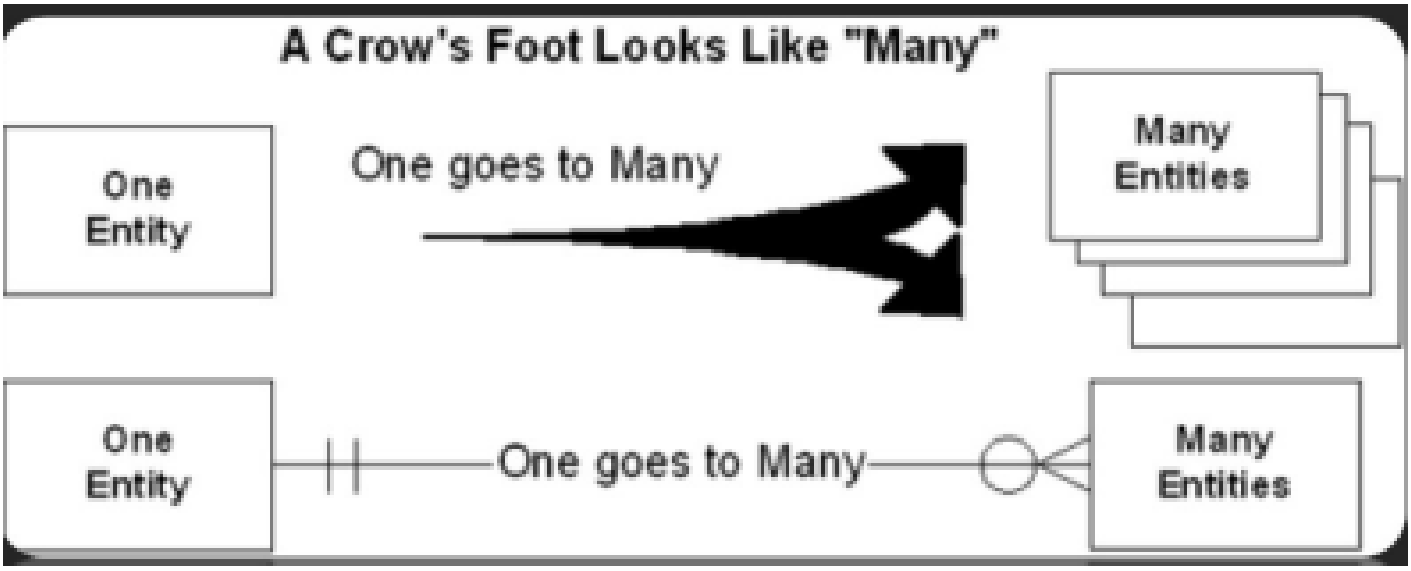
1 : N 	One to Many كل عنصر من الجدول الأول يمكن ربطه مع اكتر من عنصر بالجدول التاني
1 : 1..N 	One to one or many عنصر واحد من الجدول الأول يرتبط مع عنصر او اكتر من الجدول التاني
1 : 0..N 	One to zero or many عنصر واحد من الجدول الأول مايرتبطش مع حد خالص او يرتبط مع أي عدد من العناصر من الجدول التاني
N : N 	Many to many أي عدد من الجدول الأول يرتبط مع أي عدد من الجدول التاني
1..N : 1..N 	One or many to one or many عالاقل واحد من الجدول الأول يرتبط مع واحد او اكتر من الجدول التاني
0..N : 0..N 	Zero or many to zero or many كل اللي يجيبه ربنا كويس او اللي ييجي منه احسن منه

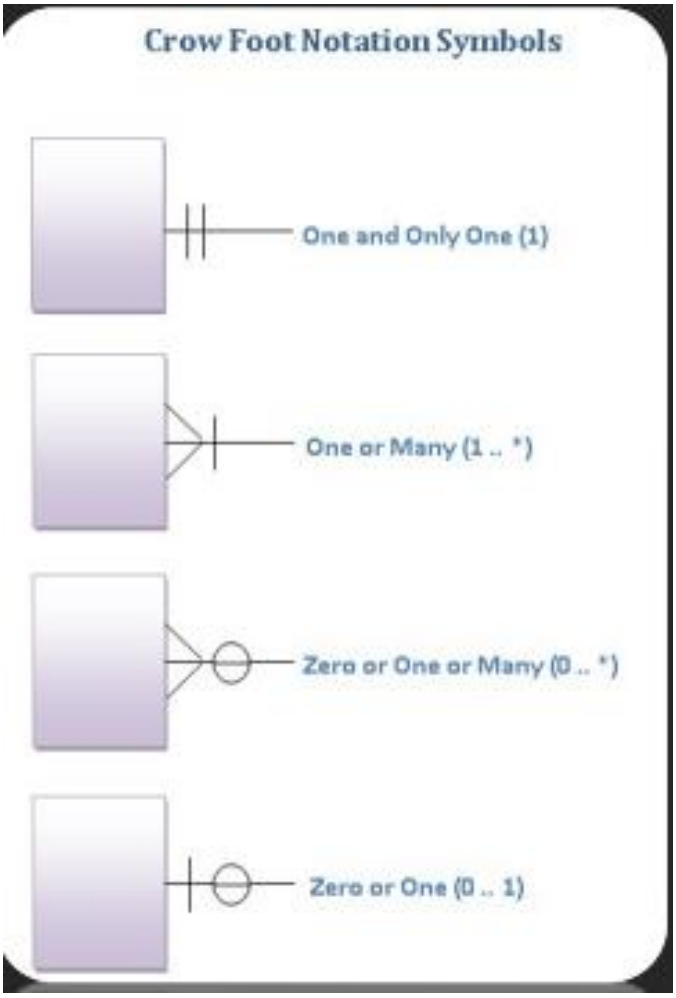



bachman-notation
1 : 1
1 : 0..1
1 : N
1 : 1..N
1 : 0..N
N : N
1..N : 1..N
0..N : 0..N

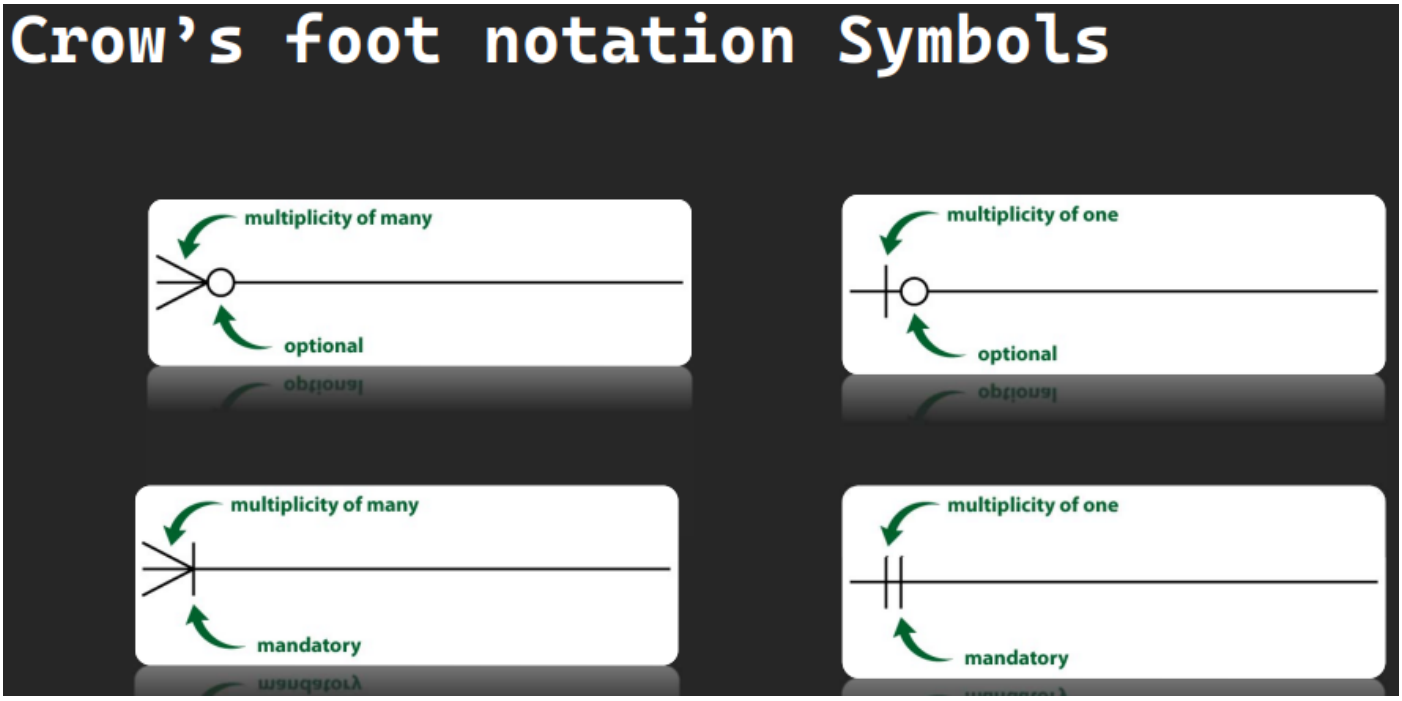
ودي امثله


Crow’s foot notation Symbols
Entity A
A Relationship
Entity B
Minimum Cardinality
Maximum Cardinality

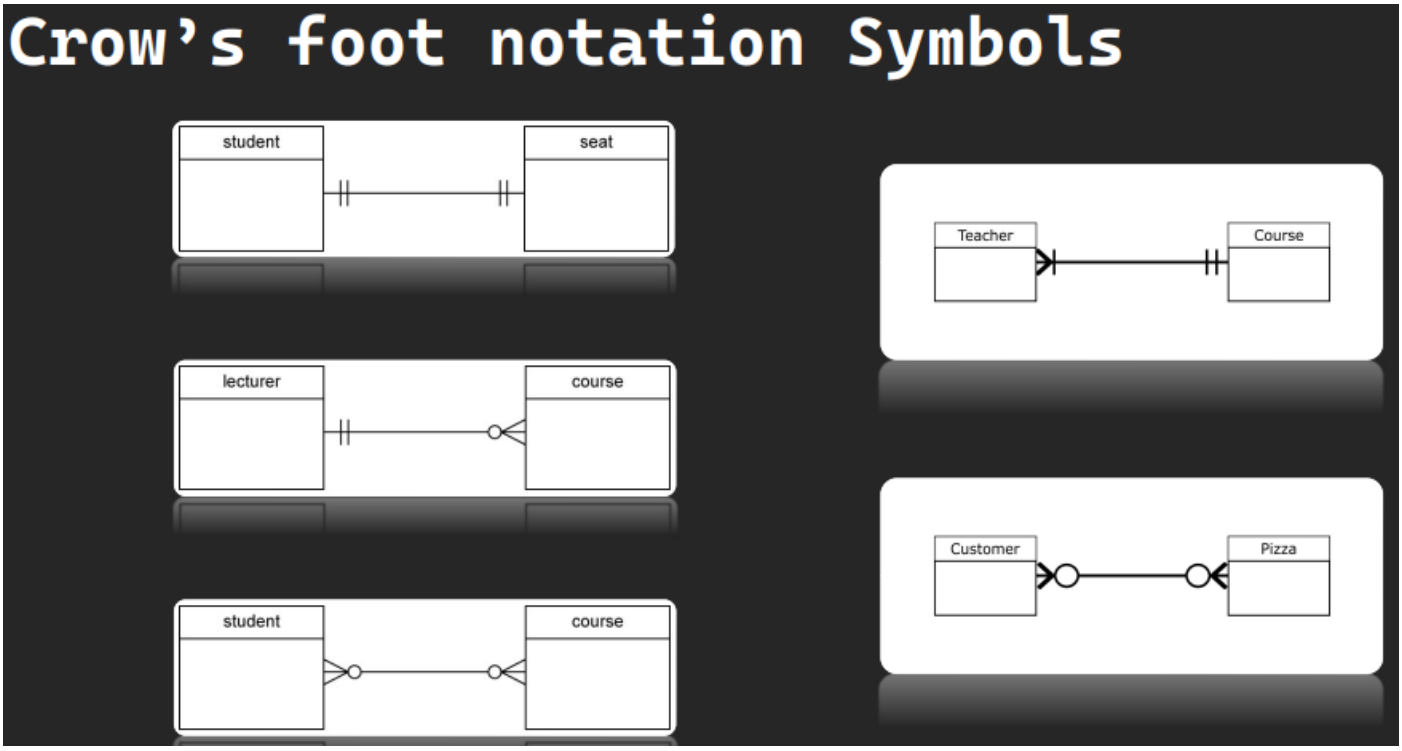

A Crow's Foot Looks Like "Many"
One Entity
One goes to Many
Many Entities
One Entity
One goes to Many
Many Entities

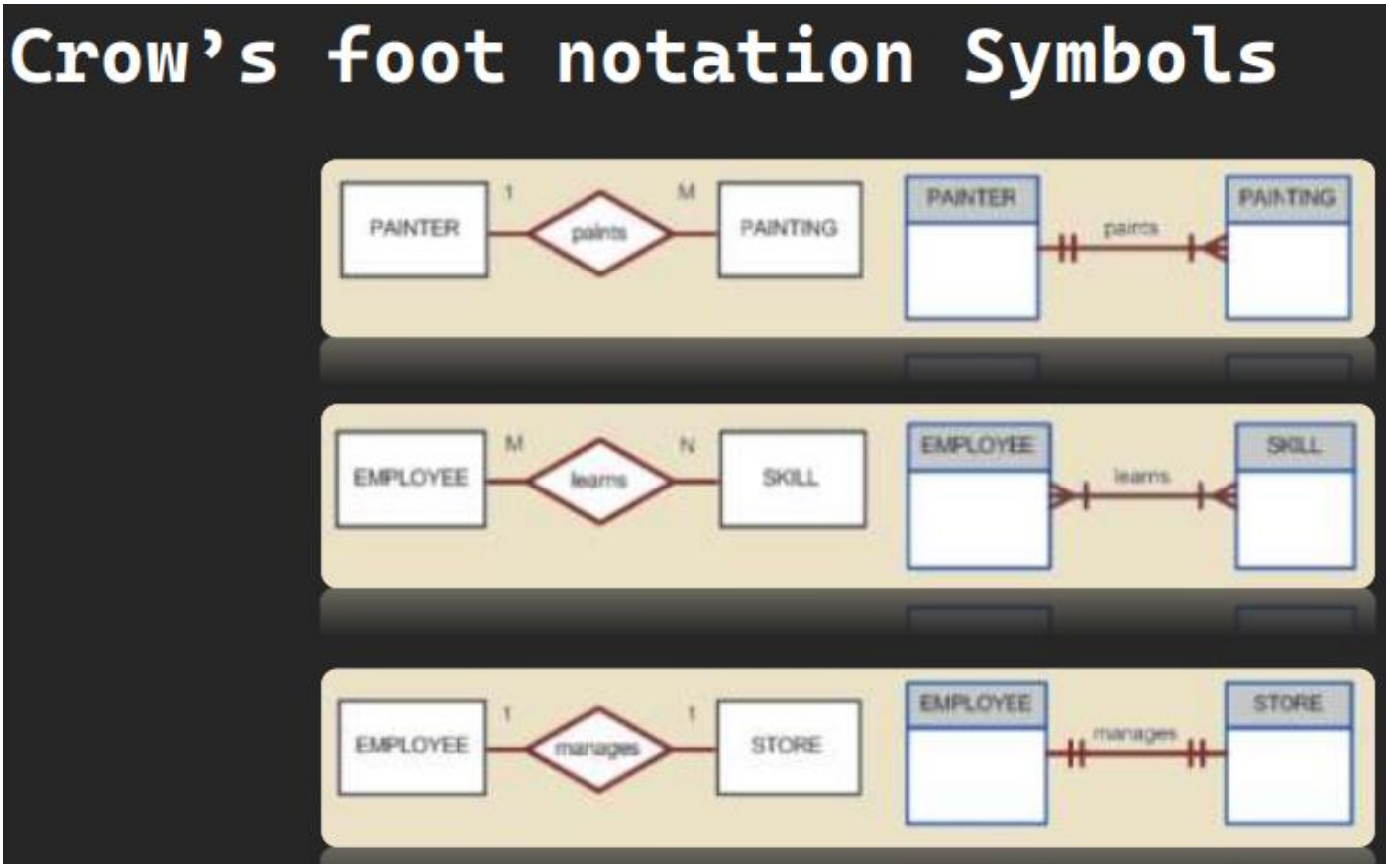

Crow Foot Notation Symbols
One and Only One (1)
One or Many (1 .. *)
Zero or One or Many (0 .. *)
Zero or One (0 .. 1)

Crow's foot notation Symbols



Crow's foot notation Symbols





في اخر مثال هوا جايبلك مثال من النت وبيوريلك انه مجرد مثال اكاديمي لكن في ارض الواقع الكلام ده مش صح لانه لو كل موظف بيدير فرع طيب والعمال والسكرتاريه والمحاسبين وباقي الموظفين وديتهم فين ياما انت بقي شركتك عباره عن محلات وجايب في كل محل مرمطون بيعملك كل حاجه وده شيء خيالي او انك فاتح ورشه وشغال فيها لوحد وجايبلي سيستم وبرضه هتكون العلاقه غلط تتحط كده وانه الصح انه تكون العلاقه one or one to zero or one

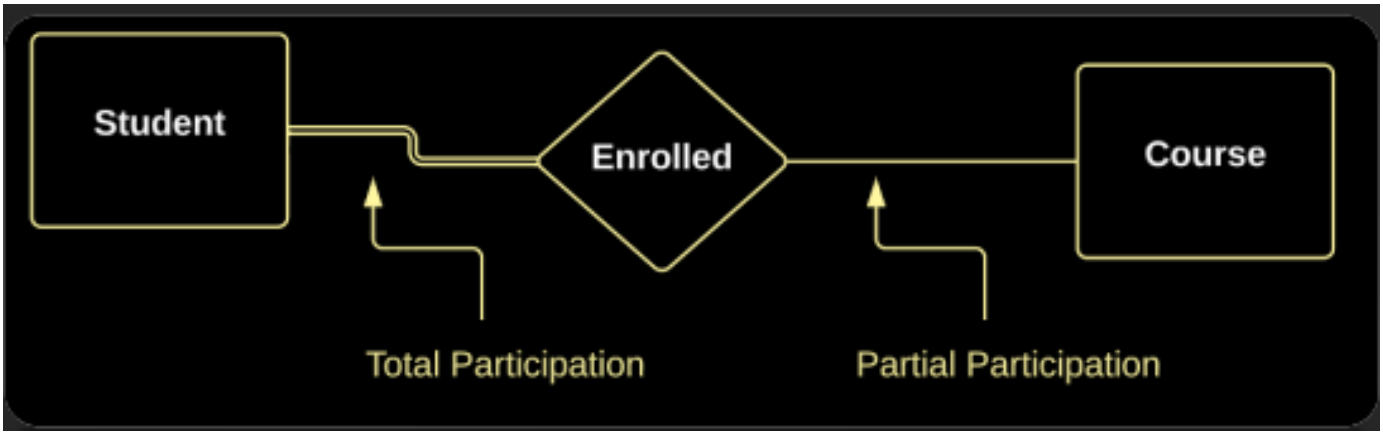
Total Vs Partial Participation

بيقولك انه زمان كانوا بيستخدموا ال partial participation وال total participation

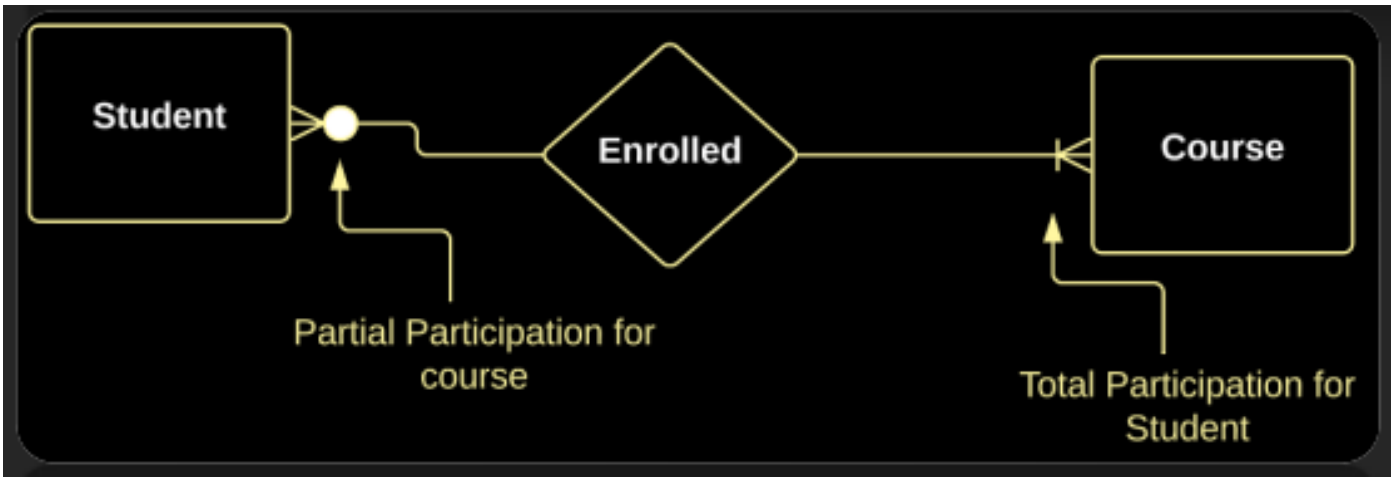
ال total participation معناها انه كل عنصر من الجدول لازم يكون مرتبط بعنصر او اكتر من الجدول التاني بغض النظر عن نوع العلاقه نفسها يعني كل طالب لازم غصب عنه ياخد كورس وماعندناش طلاب في الشارع

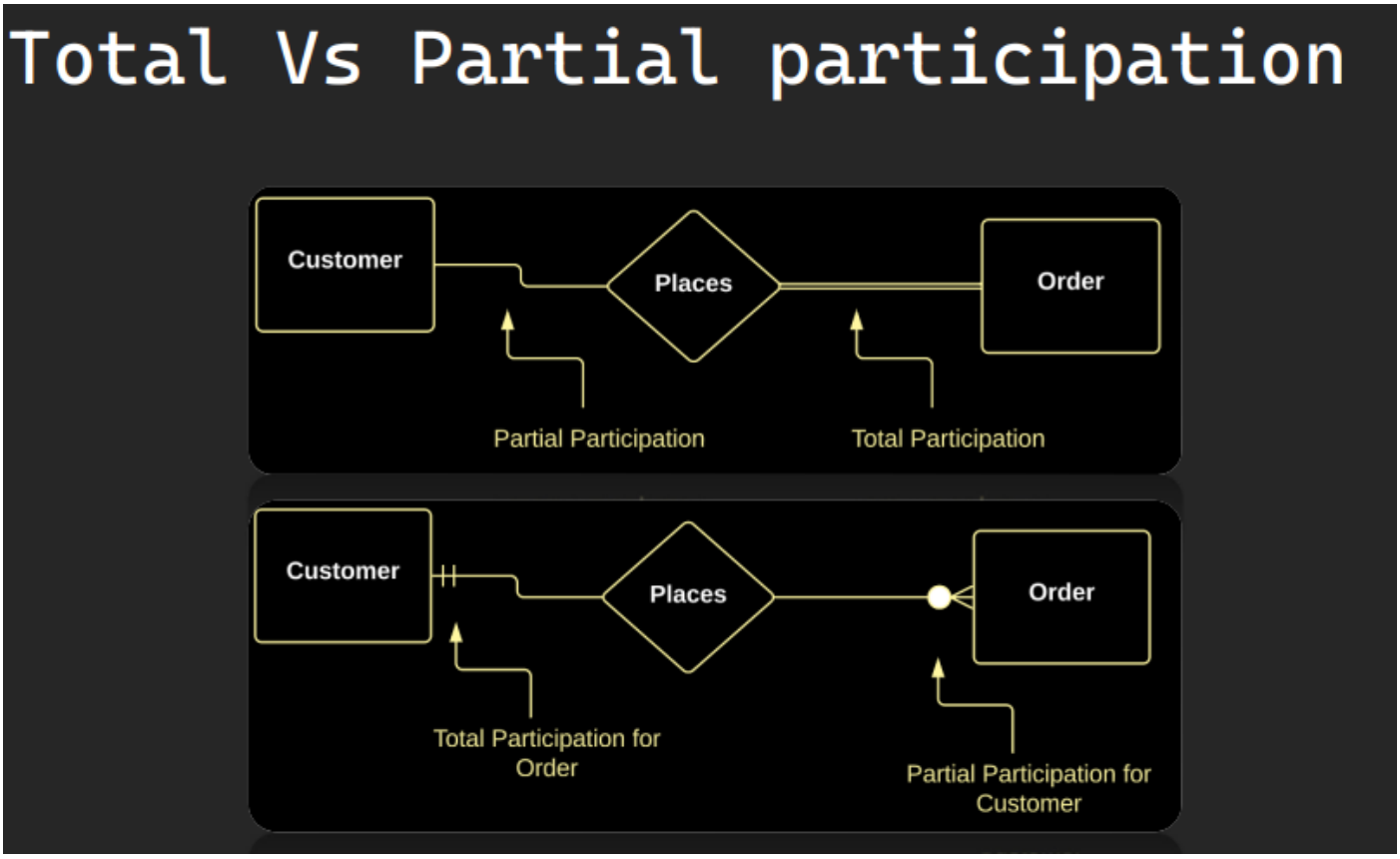
ال partial participation انه مش لازم كل العناصر من الجدول الأول تكون مرتبطه بعناصر من الجدول التاني زي انه الكورسات مش لازم يكون فيه طلبه مسجلين فيها

ودي بتبقي عباره عن انه يجيلك في ناحيه من العاقه ويرسم خطين بدل خط واحد زي كده



ال erd اللي فوق ده احنا بنمثه كده

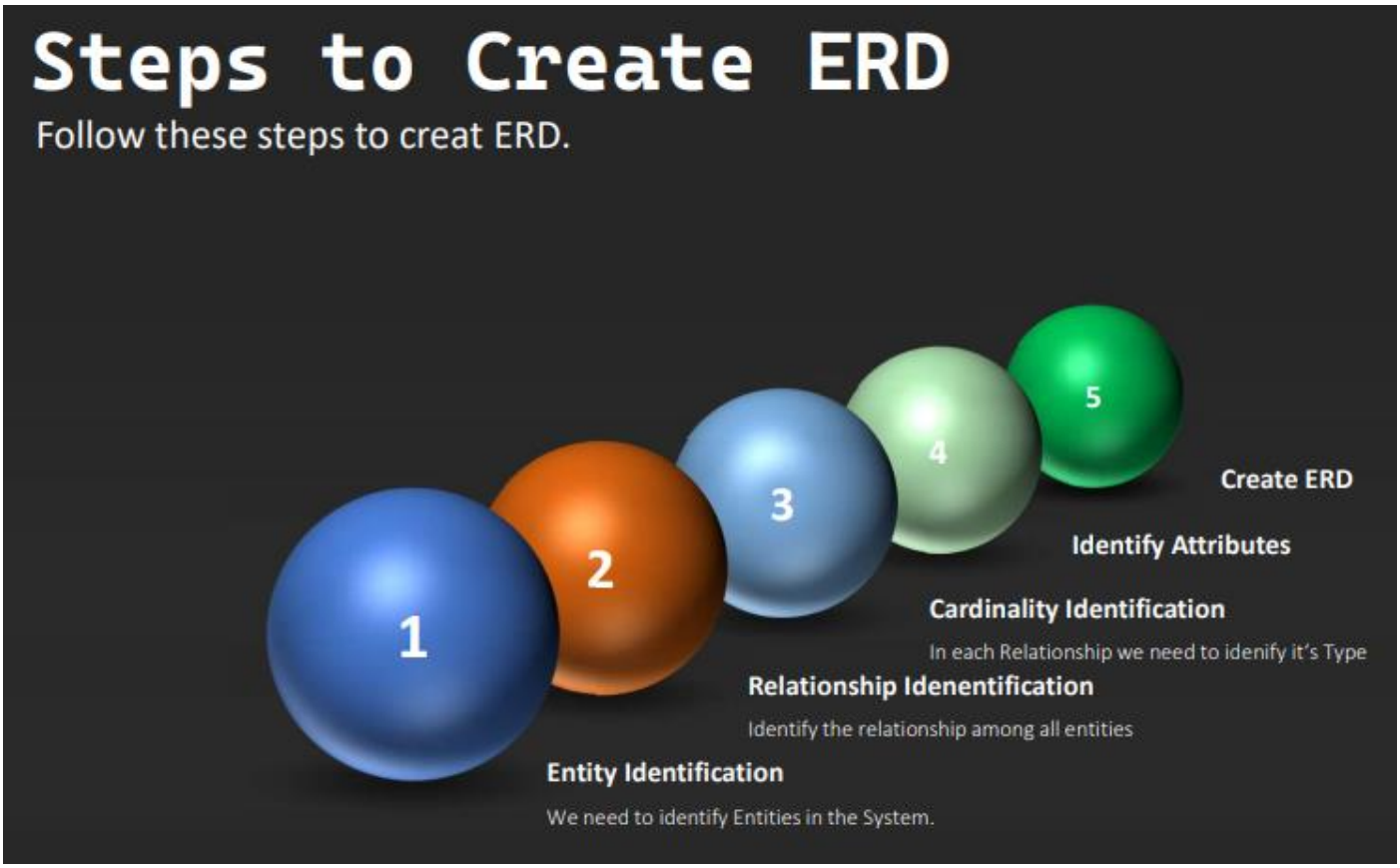




Process of Creating ERD Step by Step - Small Project

هنا بيقولك انك عشان تطلع erd عندك خطوات بتمشي عليها واحده واحده هوب تلاقي نفسك عامل ال diagram :-

1- Entity identification :- وهيا انك تحدد ايه الجدوال اللي محتاج تعمليها بناءا علي المتطلبات بتاعت المشروع
2- Relationship identification :- هل يوجد علاقه بين الجداول ببعضها ولا لا ؟
3- Cardinality identification :- لو وجدت علاقات ايه هوا نوع العلاقات دي
4- Identify attributes :- انك تحدد الاعمده بتاعت كل جدول
5- مبروك عليك ال diagram



تعالي نعمل ال diagram للمشروع ده :-

Create ERD for University

Requirements:

- In a university, a Student enrolls in Courses. A student can enroll in more than one course, Each course can have more than one student.
- Each student has only one mentor(prof), Professor can mentor more than one student.
- Each Professor can teach only one course.
- Each Course can have only one professor.
- Course can have un-assigned professor.
- Professor can teach no courses.

اول خطوه انك تحدد ال entities وهيا أي حاجه محتاج تخزن عنها معلومات

فاحنا دوقتي محتاجين enttites لل student وال courses وال professor

Step 1:

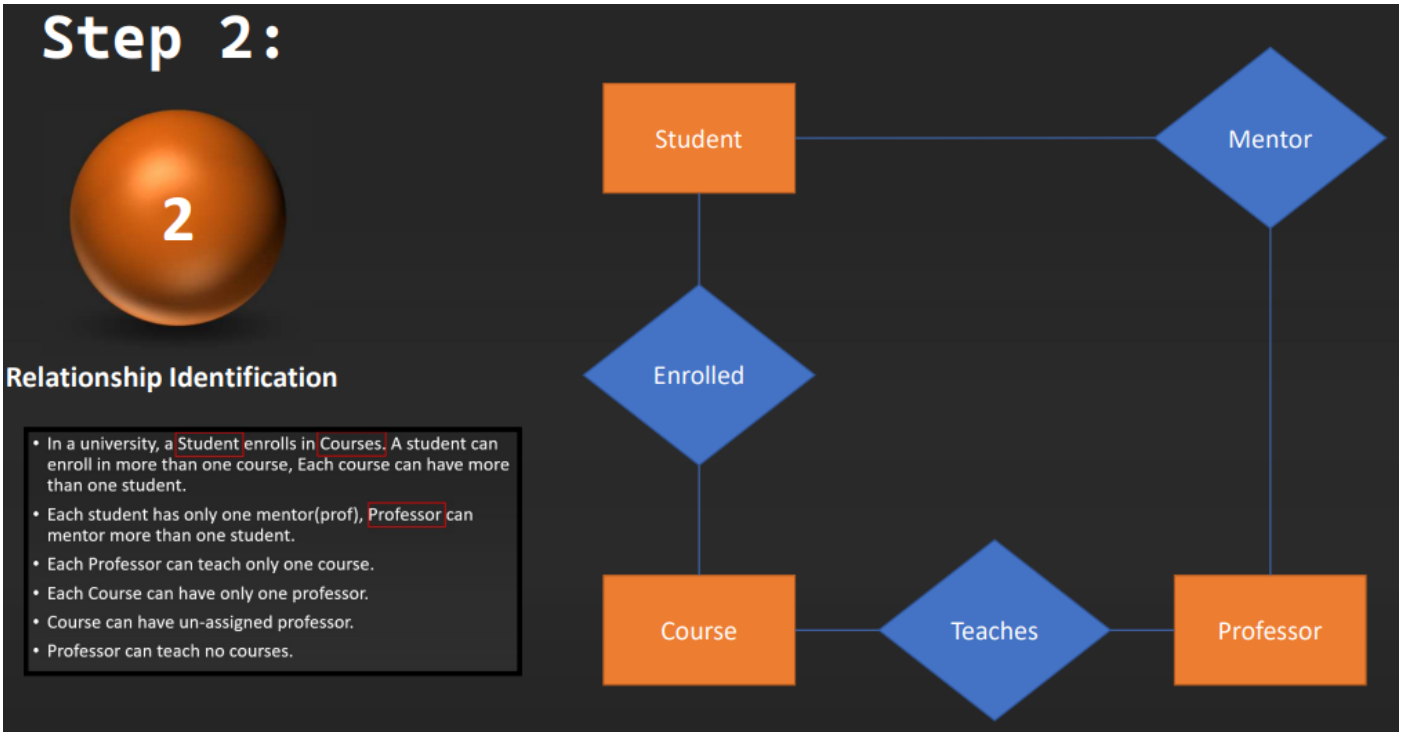
Entity Identification

Requirements:

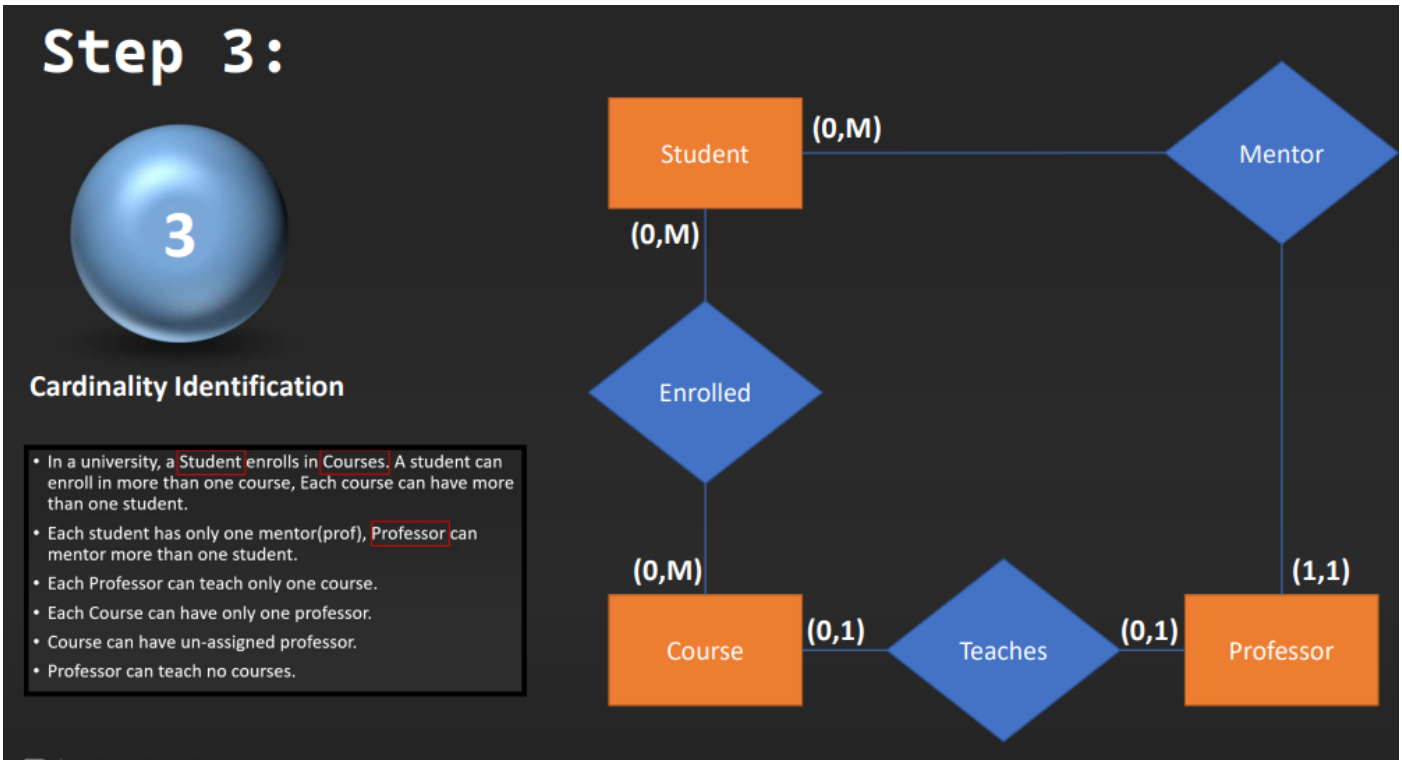
- In a university, a Student enrolls in Courses. A student can enroll in more than one course, Each course can have more than one student.
- Each student has only one mentor(prof), Professor can mentor more than one student.
- Each Professor can teach only one course.
- Each Course can have only one professor.
- Course can have un-assigned professor.
- Professor can teach no courses.

Student | Course | Professor

بعدين تعالي نشوف هل فيه علاقات بين ال entities وبعضها ولا لا



بعد كده بنحدد نوع كل علاقه واي شيء لم يذكر فهوا optional



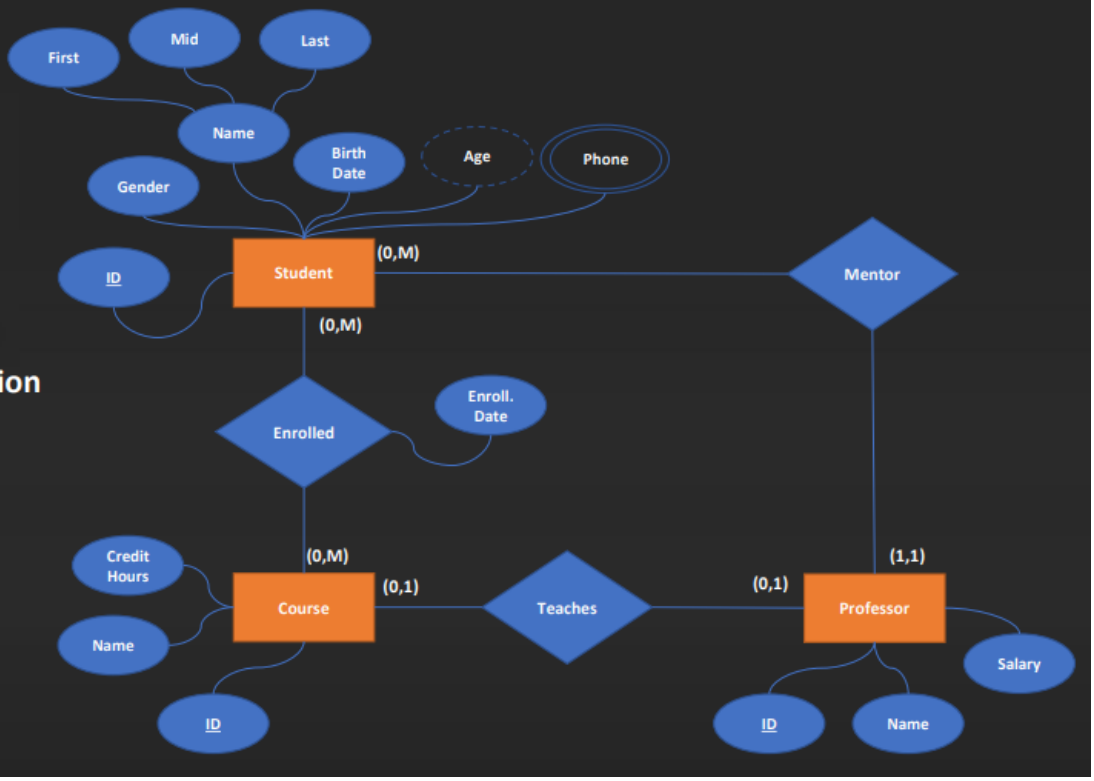
بعد كده هنحدد الاعمده بتاعت كل جدول

وبتسال بعدين هل لما بيحصل حدث معين زي انه الطالب يسجل في كورس هل فيه معلومات زياده محتاج تخزنها ؟

هقولك اه التاريخ بتاع الاشتراك ده مهم بعمله attribute بس بحطه عال relation نفسها

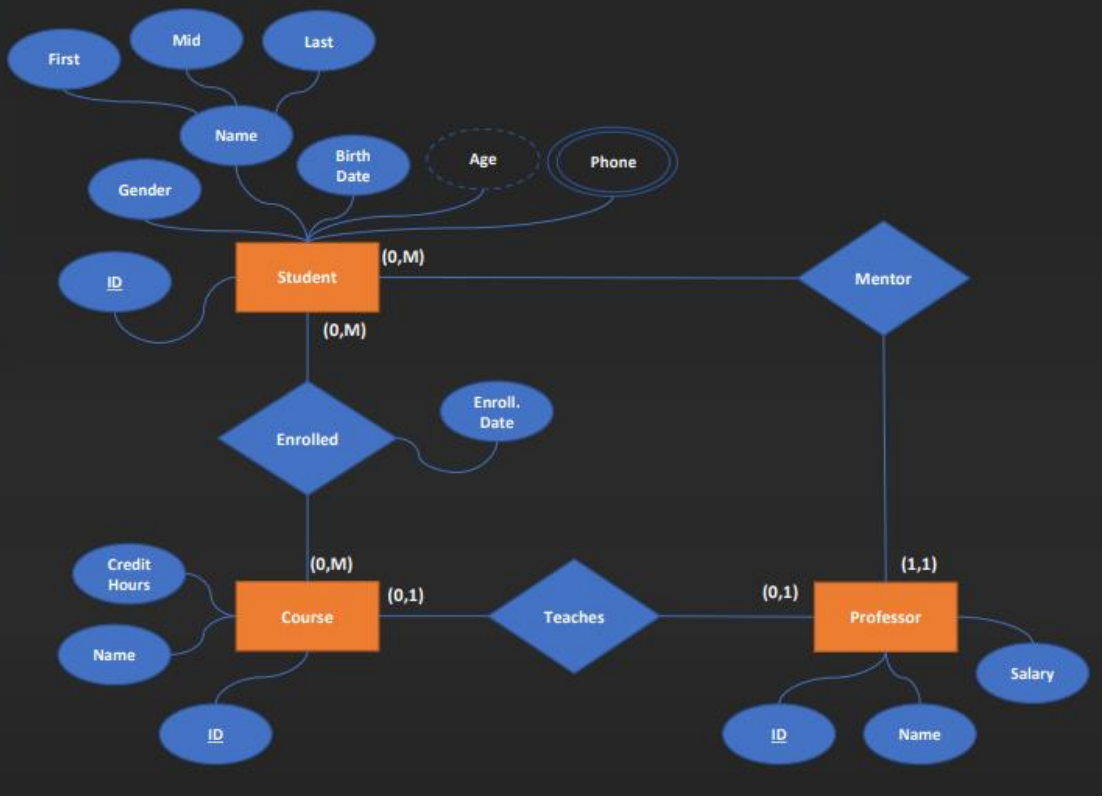
Step 4:

Attributes Identification

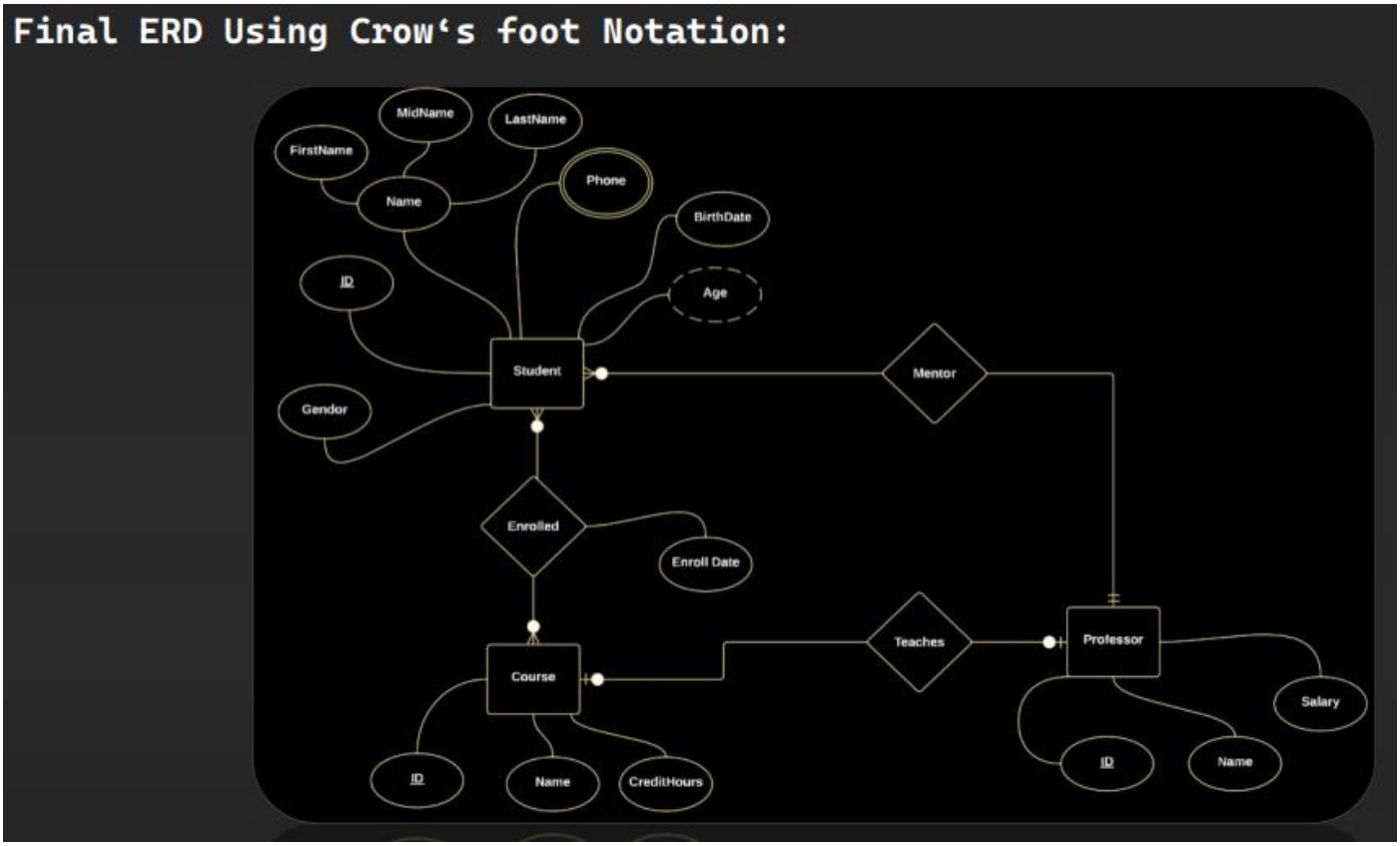


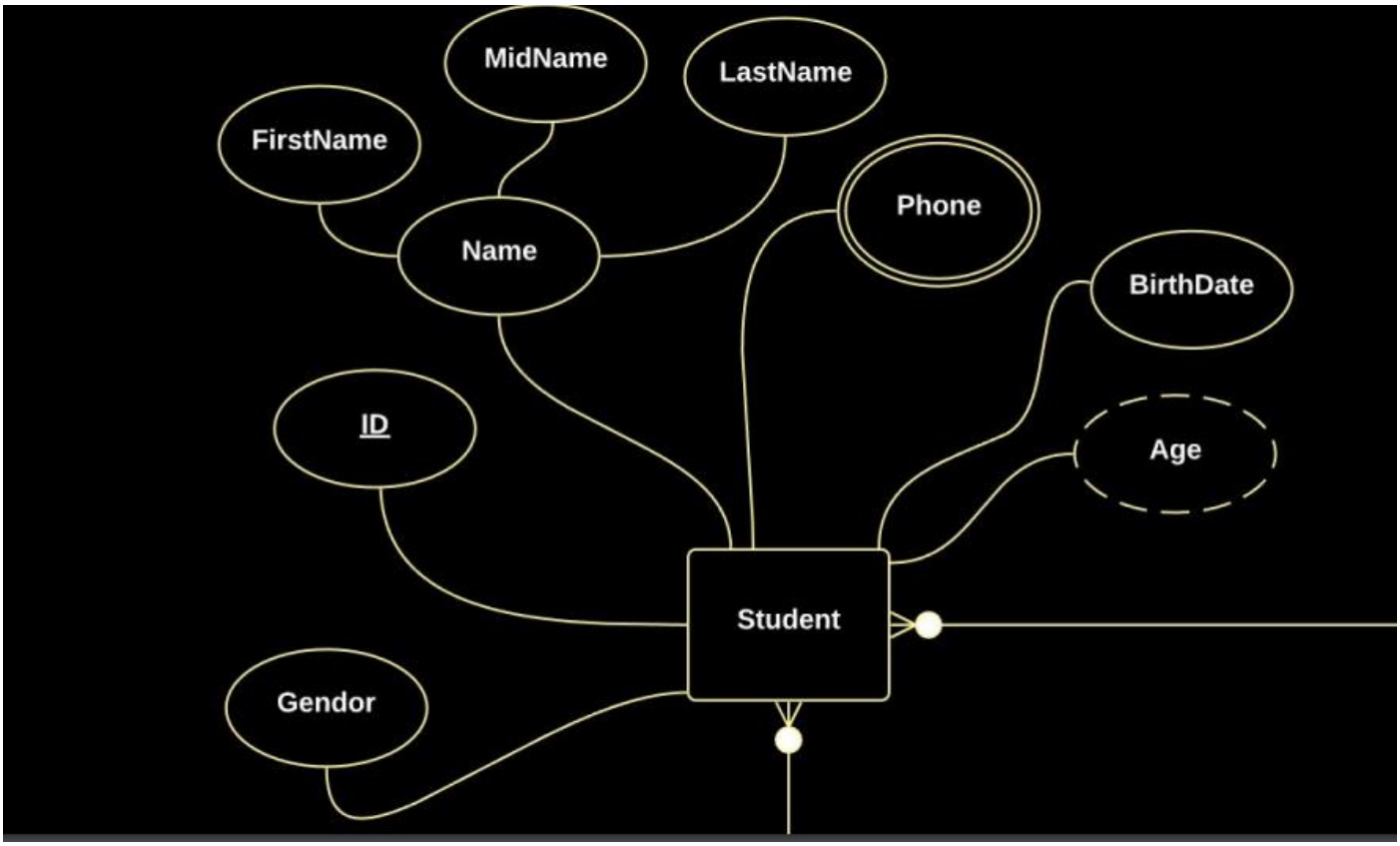
Step 5:

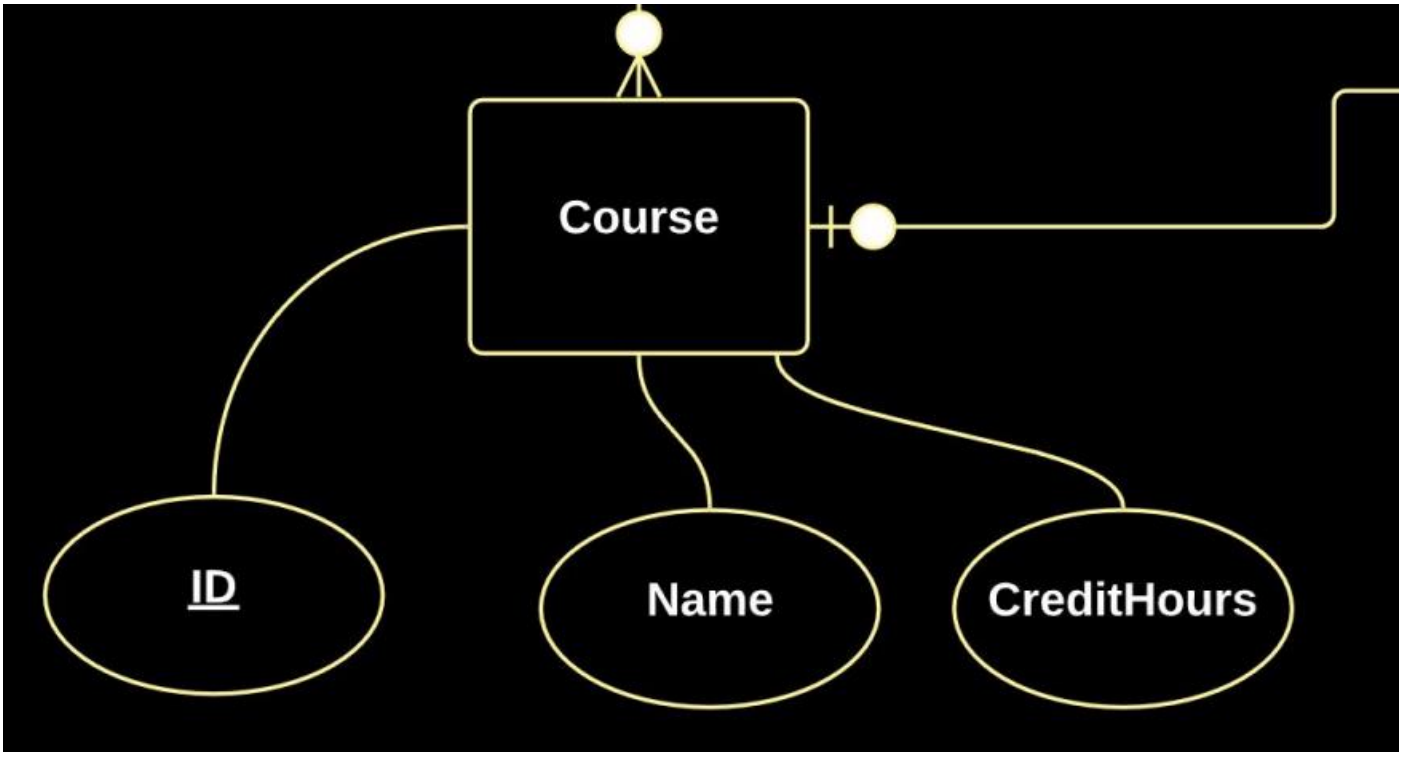
Creat ERD

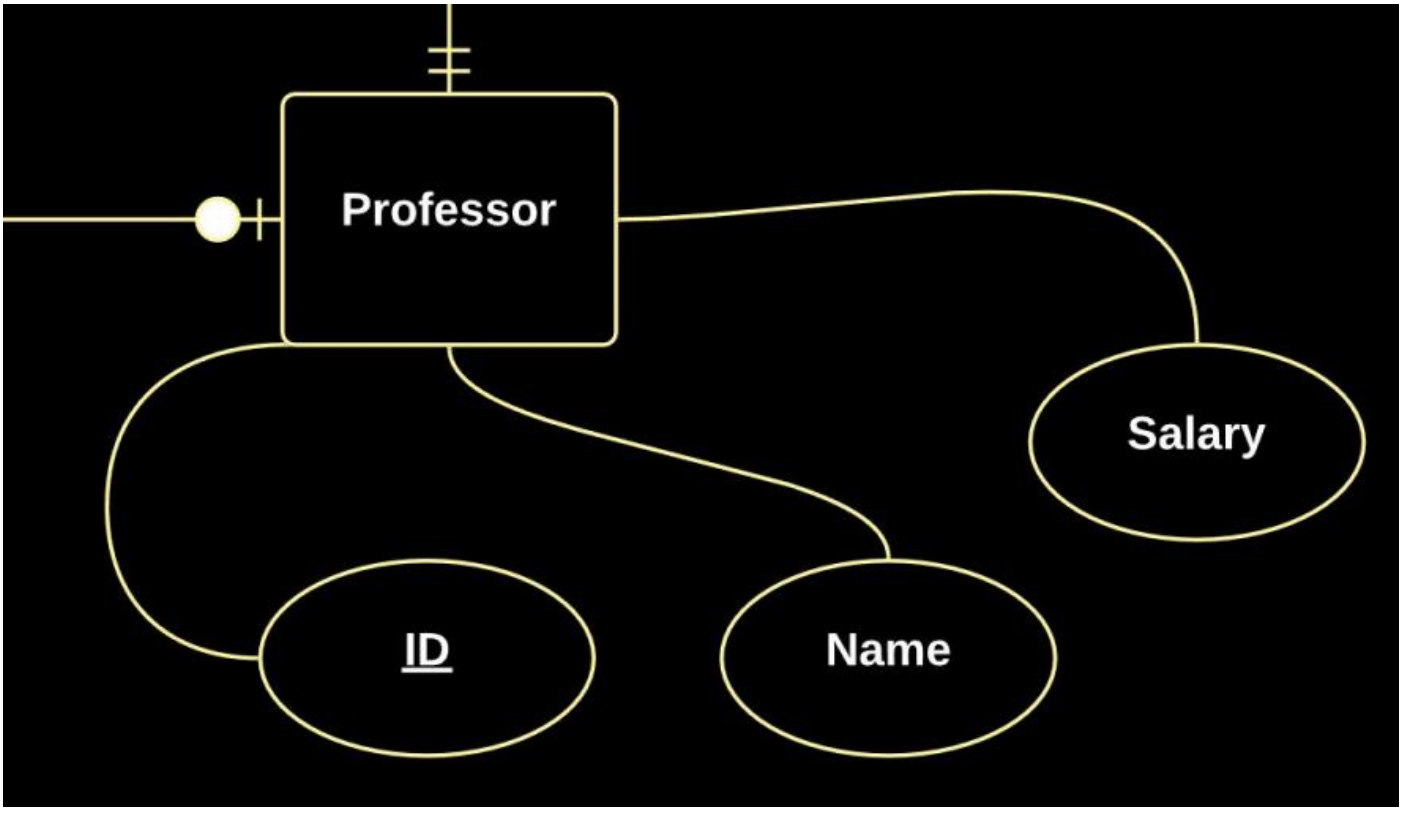


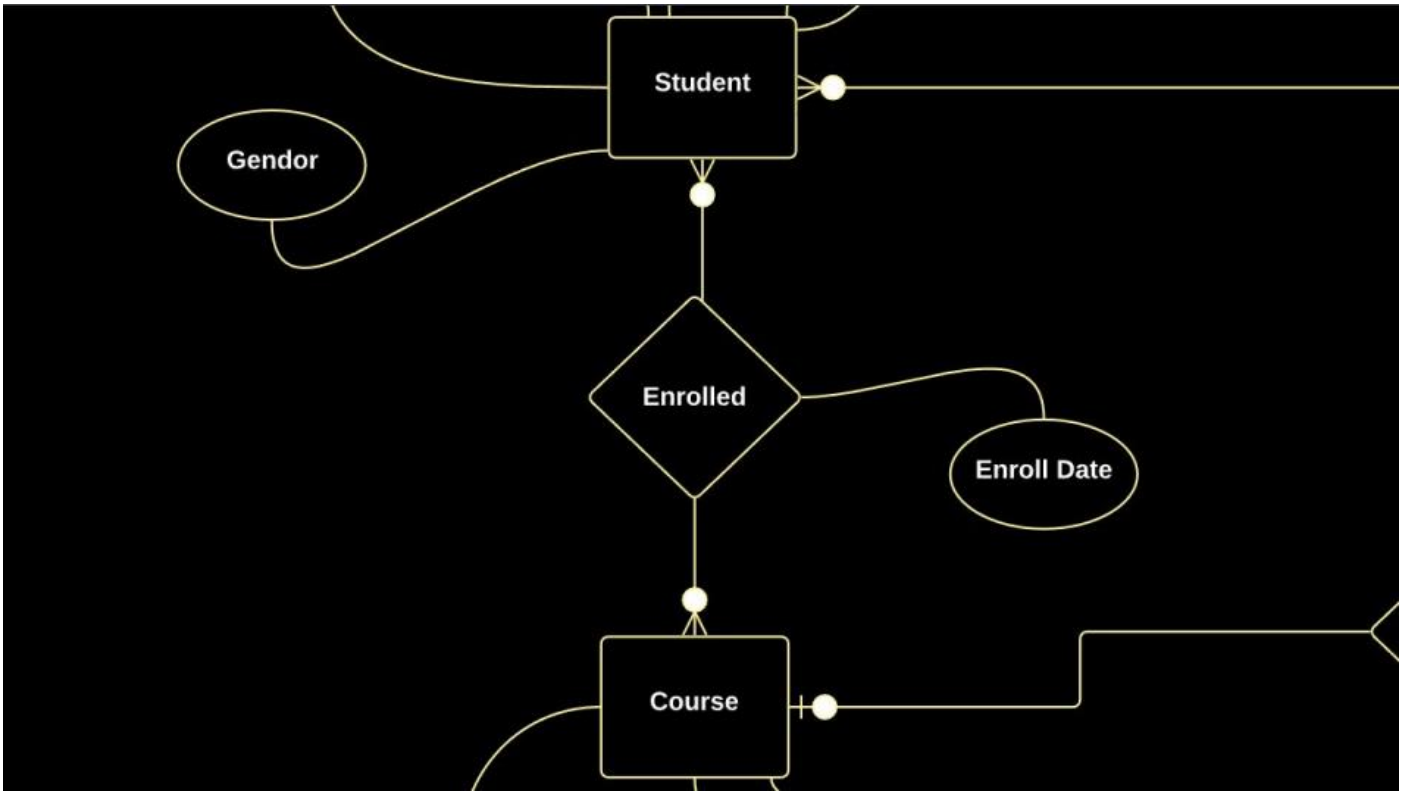
Final ERD Using Crow‘s foot Notation:

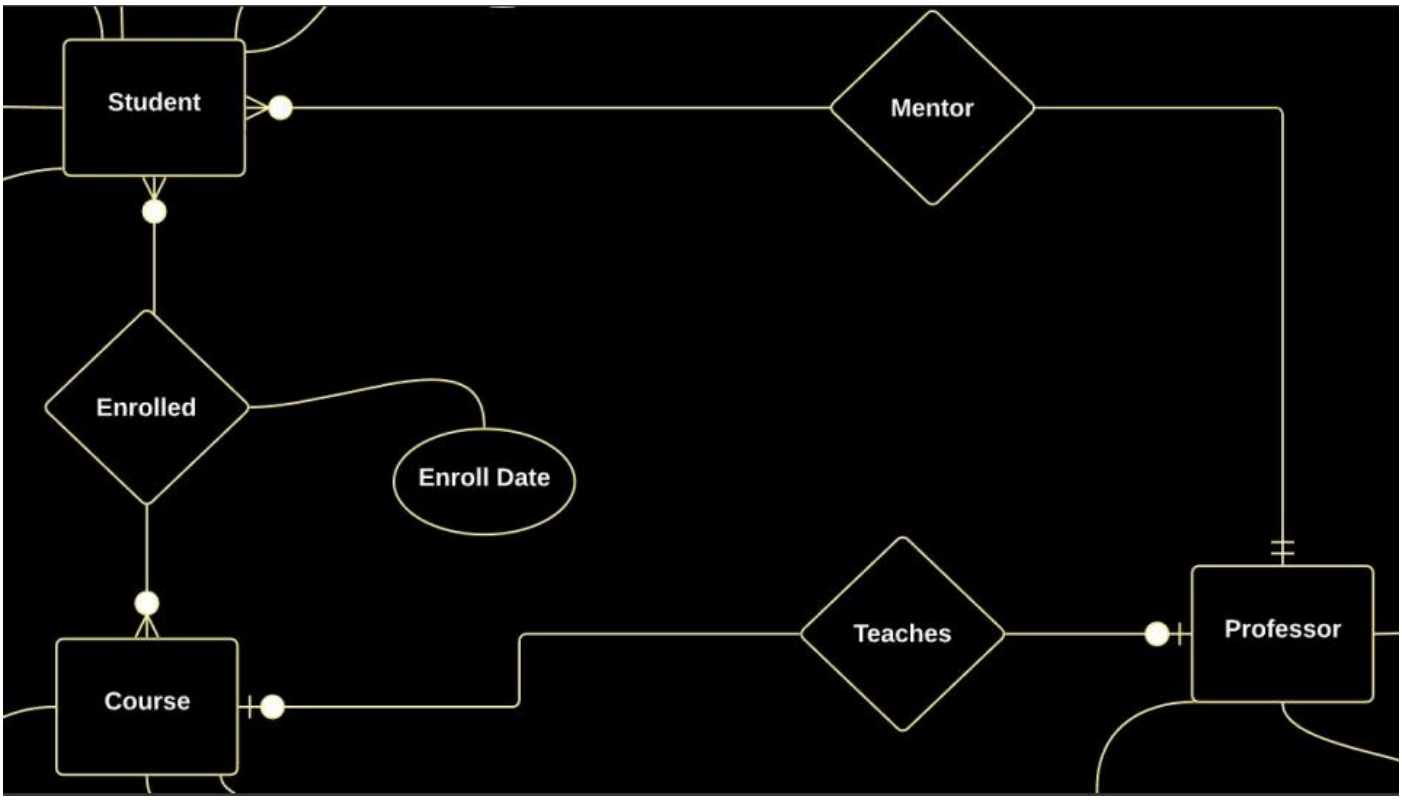





Course
ID
Name
CreditHours


Professor
Salary
ID
Name





Recommended ERD Software to Use

يمكنكم استخدام هذا البرنامج سيسهل عليكم رسم ال

ERD

البرنامج موجود اونلاين على الرابط التالي

http://erdplus.com

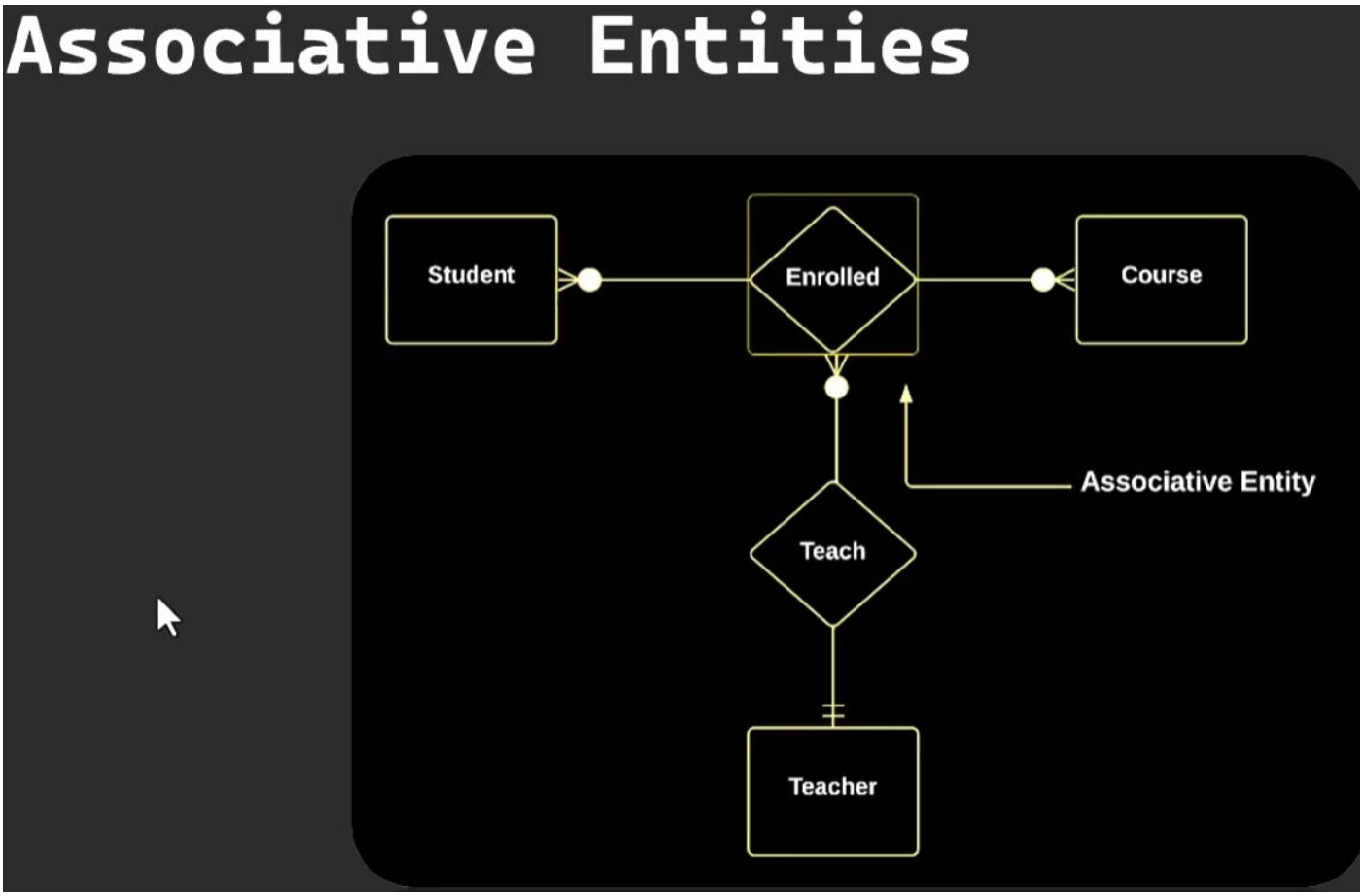
/http://erdplus.com

Aggregation / Associative Entities

ال associative entities هيا علاقه بتتبني علي علاقه زي ان المدرس بيدرس الماده للطالب هنا فيه علاقه غير مباشره بين المدرس والطالب والكورس لانه التدريس مش هيحصل الا لما يتواجد علاقه بين الطالب والكورس وهيا انه الطالب يشترك في الكورس

في الحاله دي انت بتحول العلاقه ل entity وبترمز ليها بشكل معين داخل مستطيل

ملحوظه :- الassociative entity دايما بتكون ناتجه عن علاقه many to many



الواجب

An associative entity is a type of entity in a database that is used to model a many-to-many relationship between two other entities. It is also sometimes called a junction table, a linking table, or a cross-reference table.

True

False

By using an associative entity, we can represent complex relationships between entities in a structured and efficient way, without having to duplicate data or create confusing relationships between tables. It allows us to model many-to-many relationships and avoid data redundancy, making it a useful concept in database design.

True

False

Representing relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher level entity. Aggregation is an abstraction through which we can represent relationships as higher level entity sets.

True

False

Aggregation is a specialized form of association between two or more Entities in which each Entity has its own Existence.

True

False

In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.

True

False

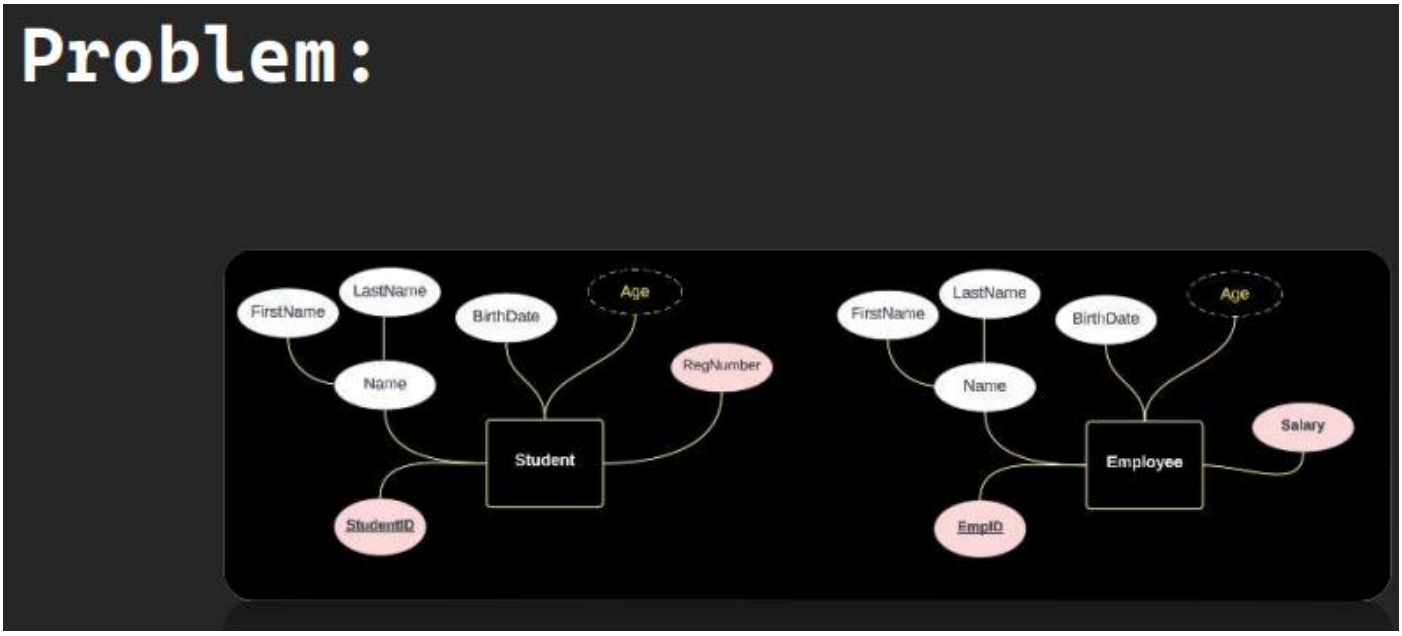
Generalization

الفكره فيه زي ماكنت بتعمل كلاس ال person وتعمل كلاس بال client واليورز بيورثوا منه

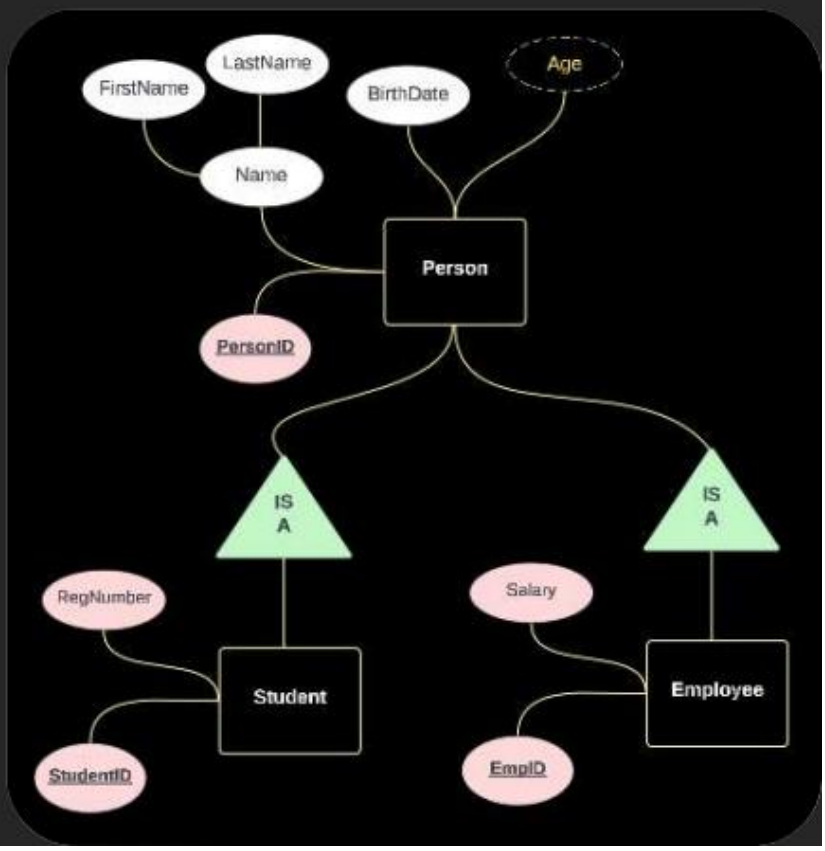
انك لما كون عندك اكتر من جدول بيتقاسموا attributes واحده فبدل ما تكتب نفس الاعمده في الجداول كلها لا انت لم الحاجات المشتركه دي وحطها في جدول واحد عام وخلي باقي الجداول تاخد منه

ماتفكرش ازاي هنعمل ده في الداتا بيز فكر دلوقتي في ال erd وبس ومالكش دعوه بالداتا بيز

ال generalization بيمثل بمثلث



Generalization:



Generalization

- Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.

- It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common.

الواجب

Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.

True

False

Generalization is Top-Down approach.

True

False

Generalization is Bottom-Up approach.

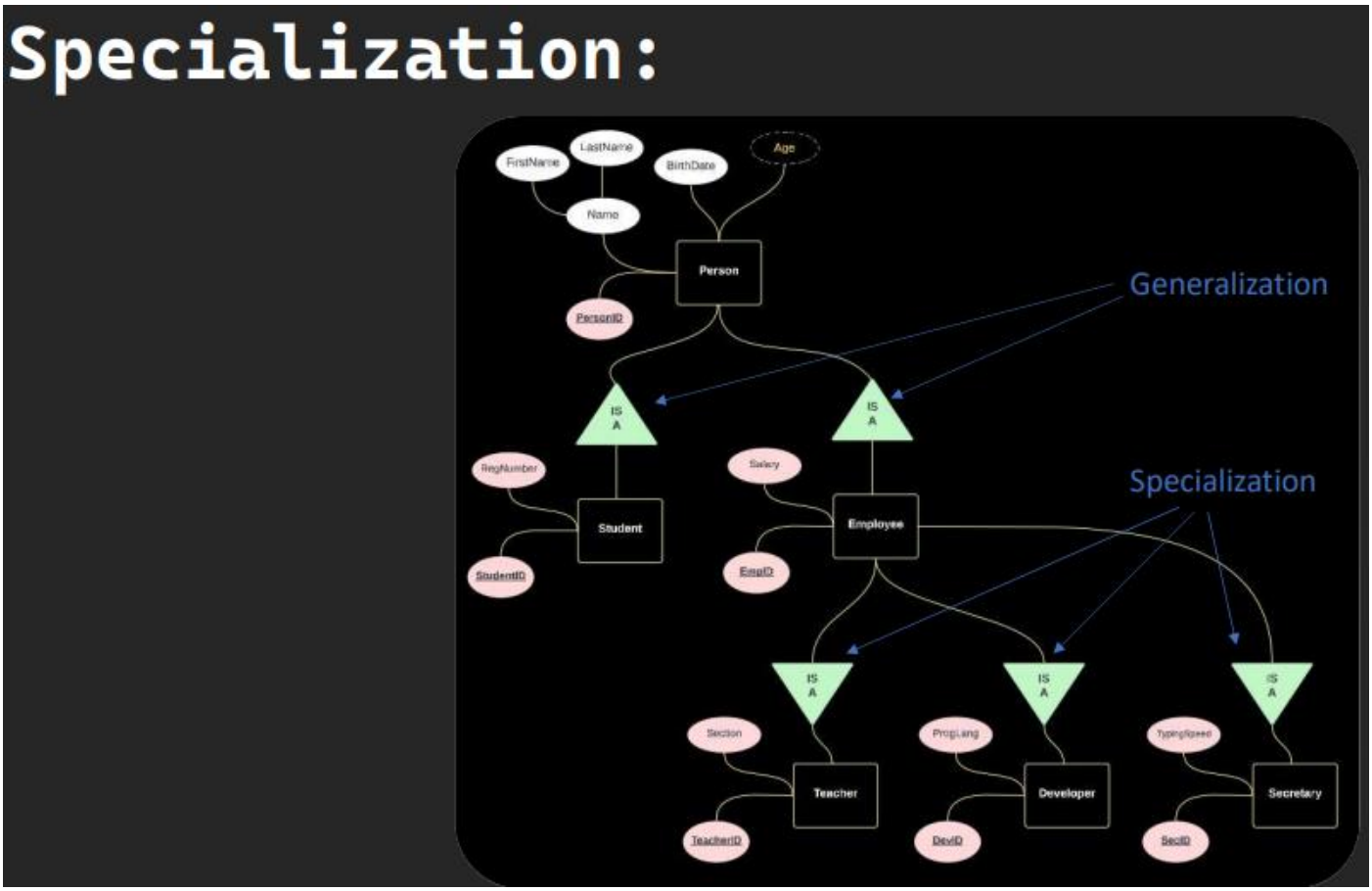
True

False

Specialization

ال specialization هوا عكس ال generalization وهوا انك بتحدد او بتعرف اكتر او بتقسم ال entity ل entities تانيه

الفرق بينهم انه ال generalization انت ممكن تحط فيه بيانات حد من الشارع انما ال specialization لا انت الناس موجوده عندك وبتقسمهم



Specialization

- In specialization, an entity is divided into sub-entities based on their characteristics.
- Specialization is a top-down approach in which a higher-level entity is divided into multiple specialized lower-level entities.

الواجب

In specialization, an entity is divided into sub-entities based on their characteristics.

True

False

Specialization is a bottom-up approach

True

False

Specialization is a top-down approach.

True

False

Specialization is a top-down approach in which a higher-level entity is divided into multiple specialized lower-level entities.

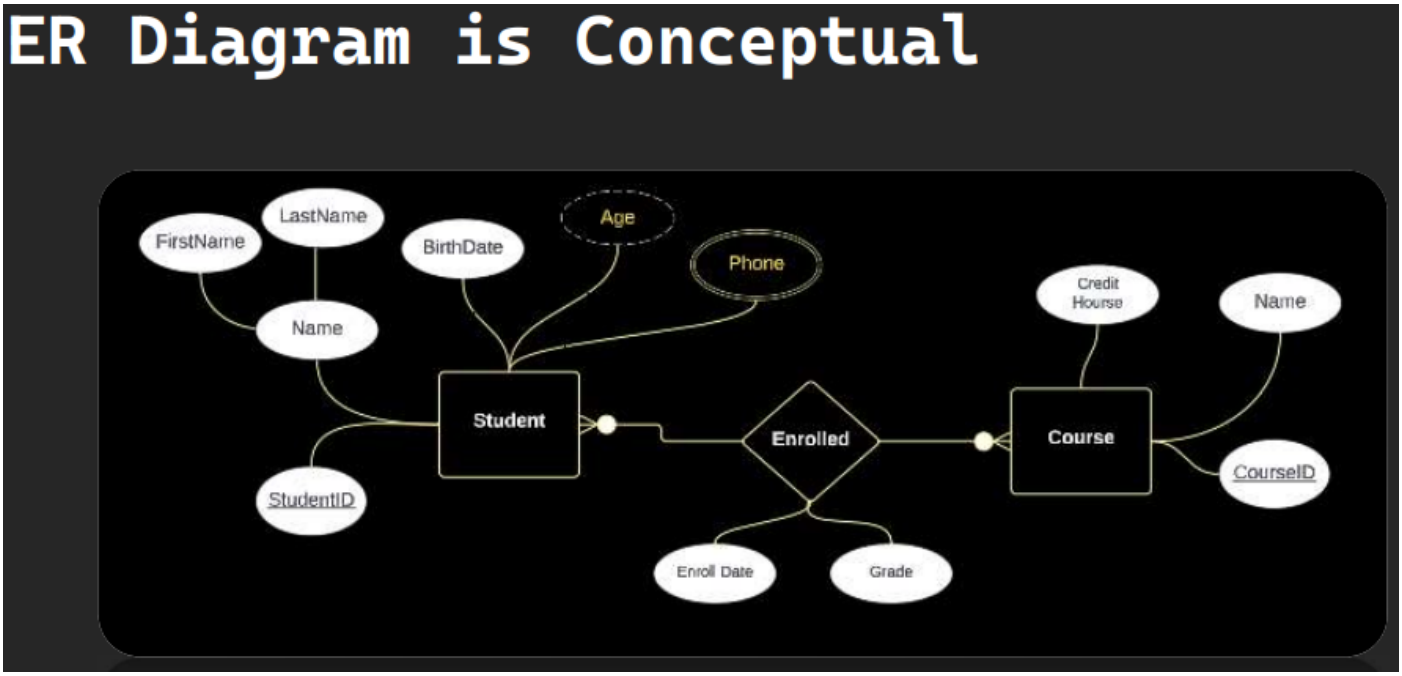
True

False

What is Relational Schema?

هنا بيقولك انه ال erd مهم في انه يديك تصور عام لشكل الداتا بيز وبعد مانرسمه بنحوله لحاجه اسمها relational schema وهيا جزء من ال logical diagram وبيخلي ال erd اقرب للواقعيه وسهل في تحويله لداتا بيز

ده erd

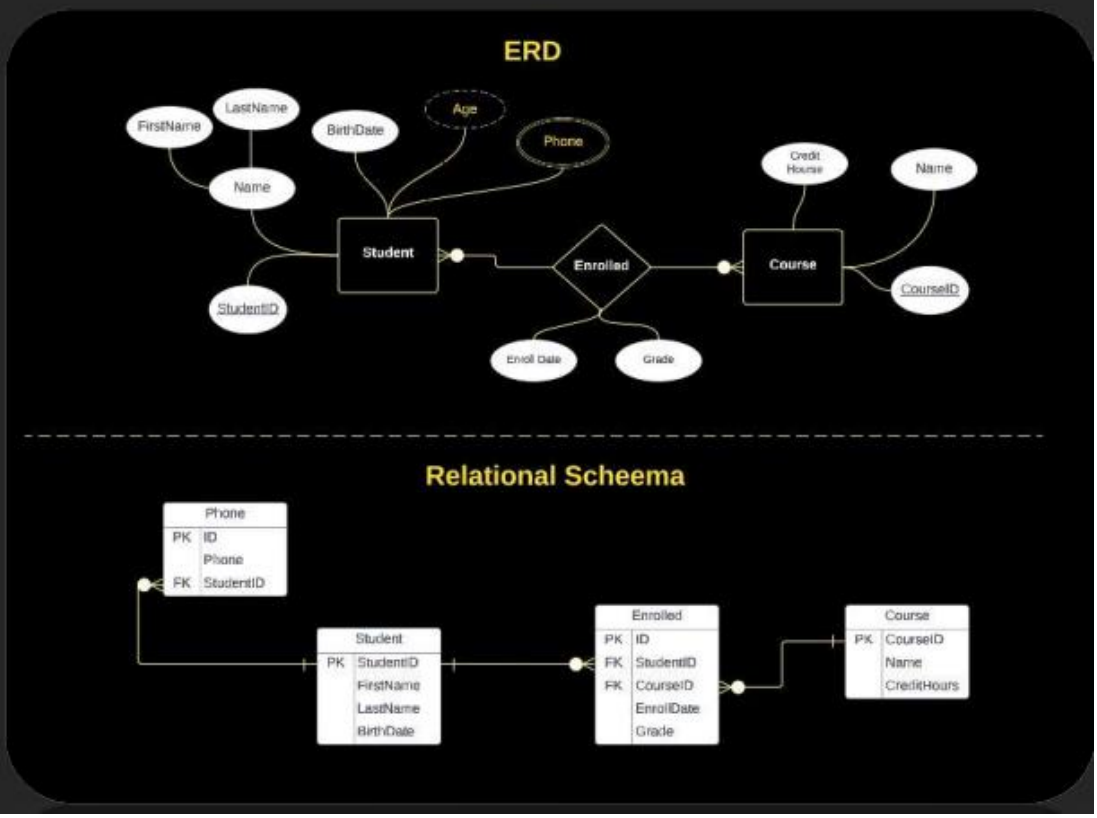


هل بالضرورة اني راسم جدولين يبقي انا هعمل داتا بيز فيها جدولين ؟

قالك لا لانه فيه حاجات تانيه هتحتاج اني اعملها جداول ازاي وامتي وفين ده هييجي بعدين المطلوب منك انك تعرف انك بعد ماتعمل ال erd بتعمل relational schema

فلما احول ال erd اللي فوق ل relational schema هيطلع كده

ER Diagram to Relational Schema



Relational Schema

- A relational schema is a set of relational tables and associated items that are related to one another.
- Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. every attribute would have an associated domain.

الواجب

A relational schema is a set of relational tables and associated items that are related to one another.

True

False

Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. every attribute would have an associated domain.

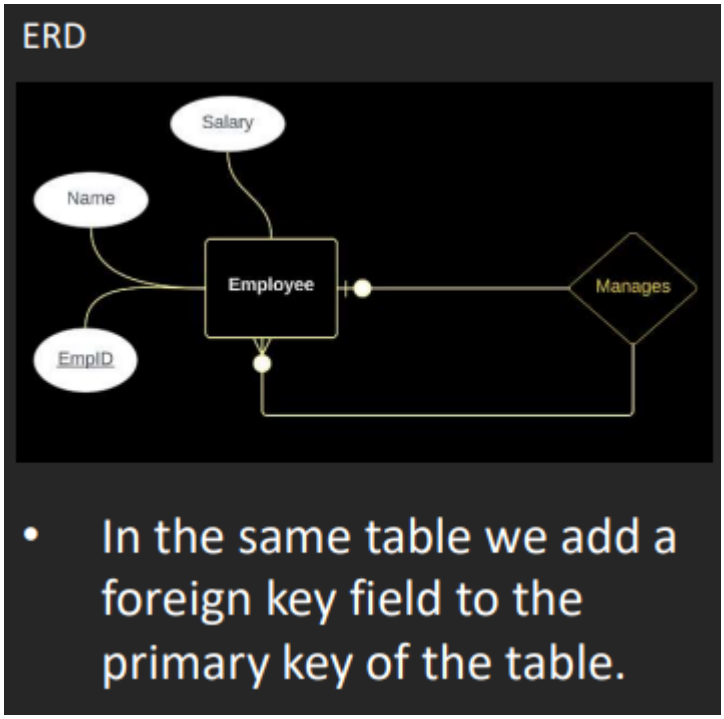
True

False

Convert Self Referential ==> Relational Schema

هنبدأ في تحويل ال erd ل relational schema

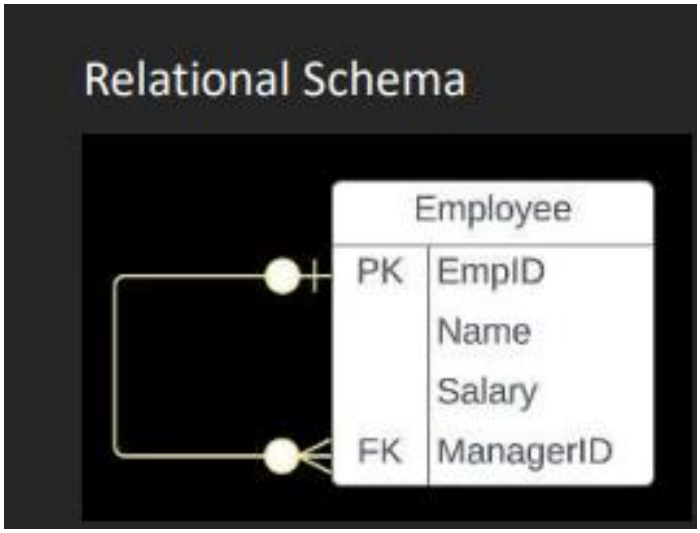
وأول حاجه هنبدأ بيها هيا ال self refrential entity وهيا ال entity اللي ليها علاقه مع نفسها

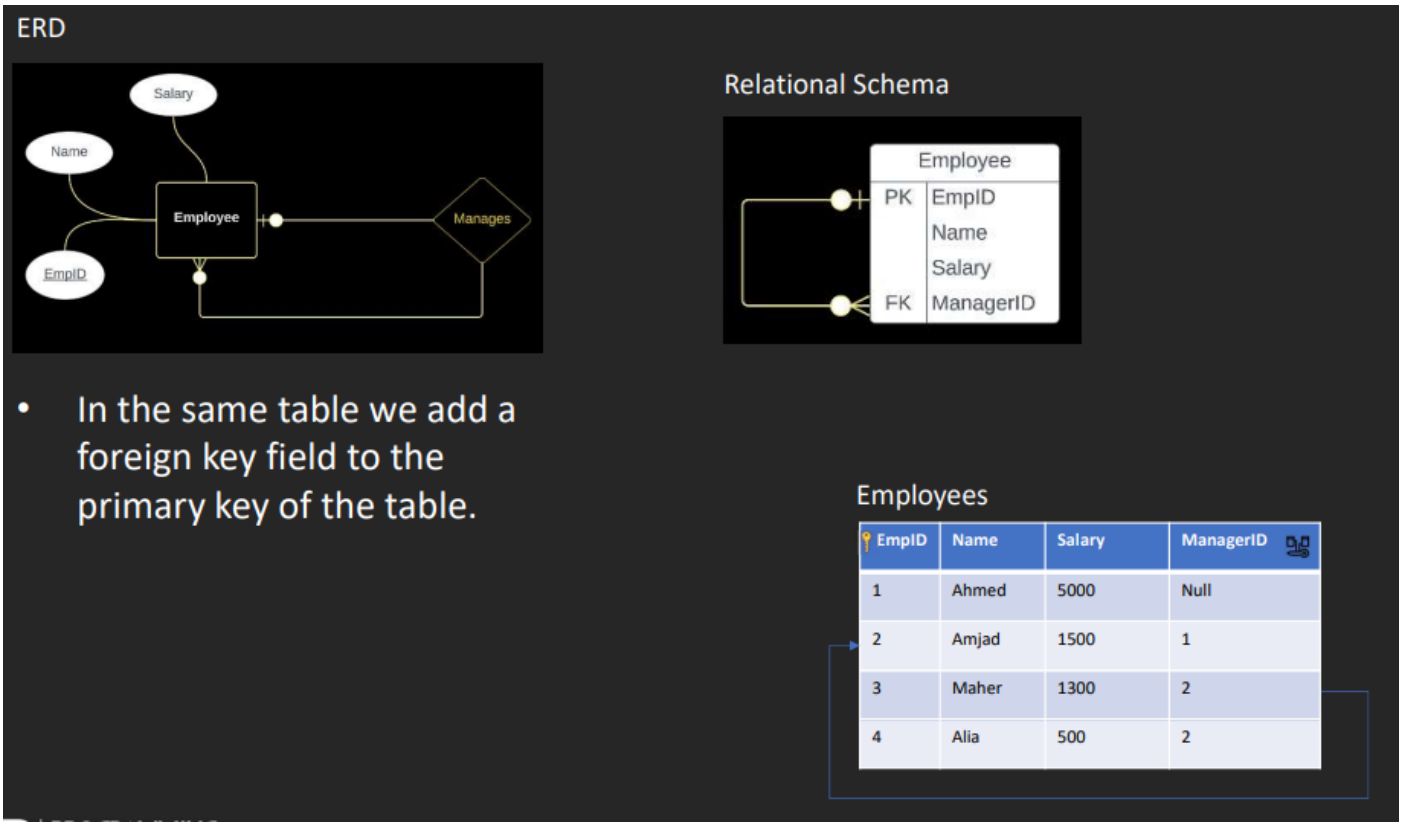


كل اللي بتعمله انك بتعمل مربع بكتب فيه اسم الجدول او ال entity وتشد سطر وبعدين تكتب ال attributes وجنب العمود اللي عايز تعمله primary key بتكتب pk

وبعدين بتزود عمود بيمثل ال foreign key وتكتب جنبه fk والعلاقه نفسها بتمثلها بانك بتوصل ال primary key بال foreign key وتحط نوع العلاقه

زي كده





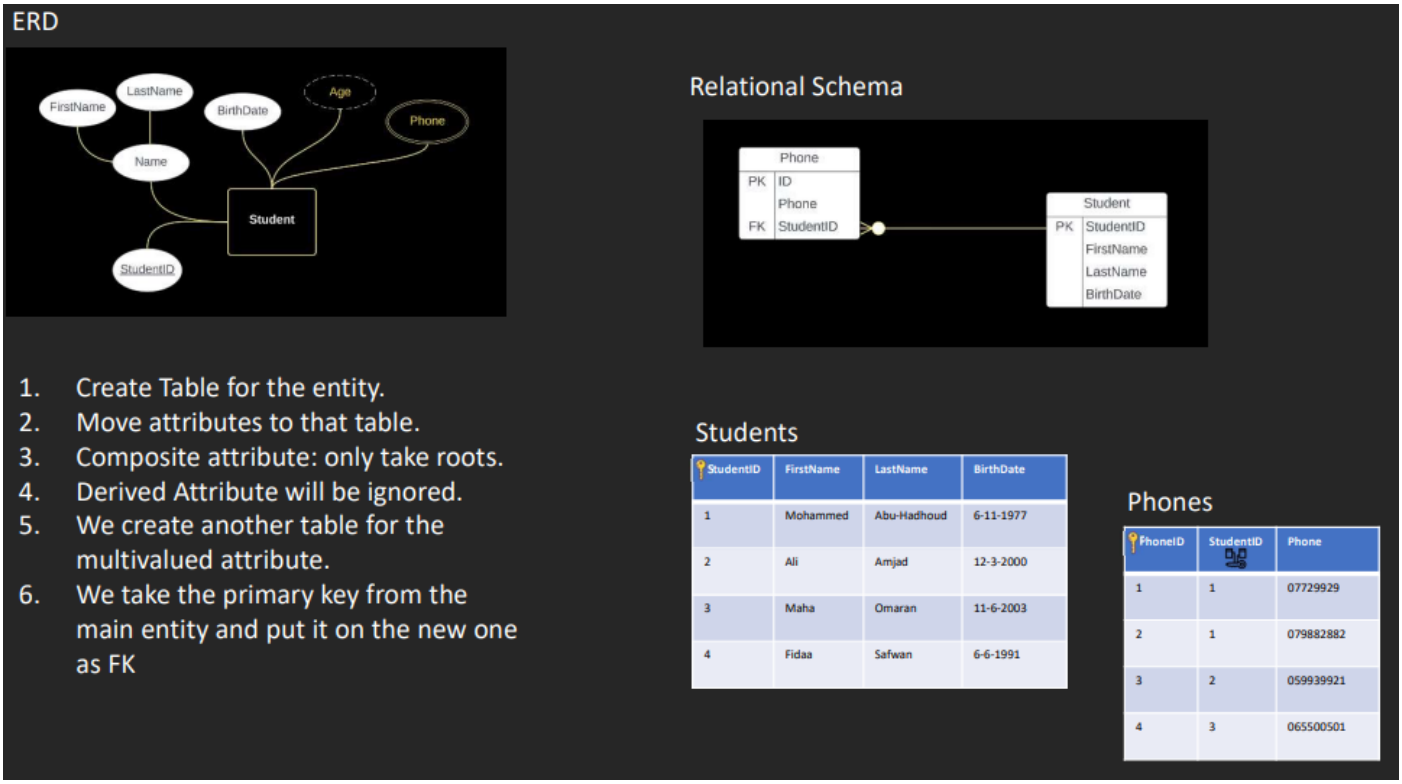
Convert Composite-Multivalued-Derived Attributes ==> Relational Schema

هنا بيقولك وانت بتعمل ال schema ولقيت في وشك composite attribute زي ال name ماتتغضش لان كل اللي هتعمله هتاخد ال attributes اللي هوا واخد منها قيمته زي ال first name وال last name وتحطهم في نفس الجدول ولا كانك شايف ال name ده ولا يشغلك في حاجه

لو لقيت في وشك derived attribute ولا كأنك شايفه

لو لقيت multi valued attribute هتاخد تديله كف تعمل منه entity صغيره يعني هتاخده تعمله جدول لوحده زي ال phone مثلا لو الشخص ليه اكتر من رقم هتعمل جدول فيه id والتيليفون وال student id اللي هيكون foreign key يشاور علي الطالب صاحب التيليفون ده

زي كده

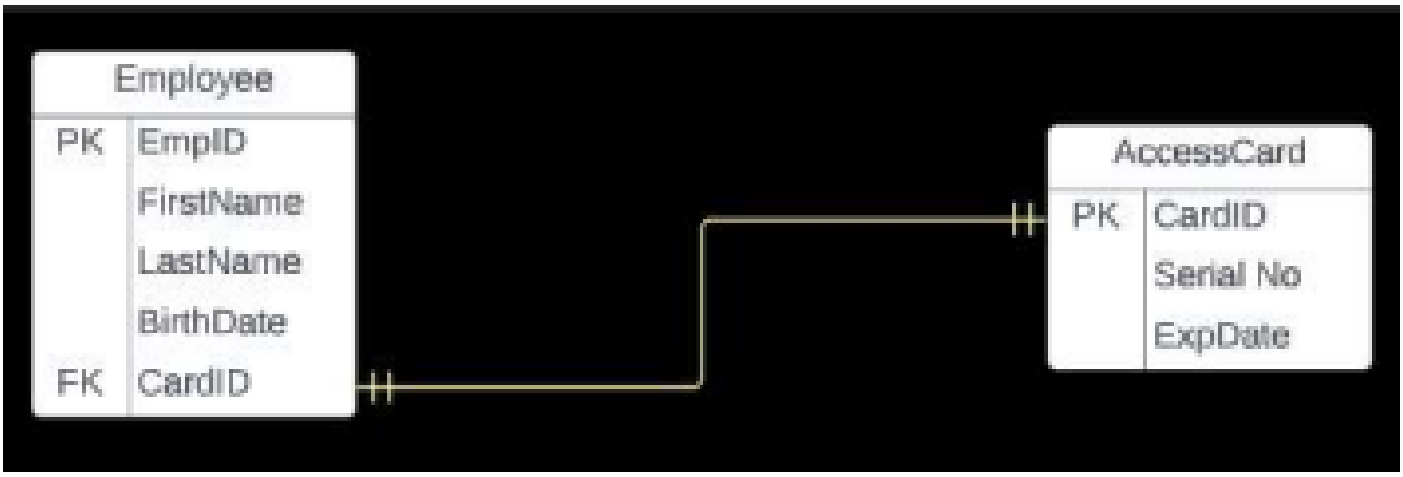


فيه ملحوظه هنا انك لما تتعامل مع ال erd او ال schema انت بتتعامل مع entity واحده لكن لما تتعامل مع جدول اصبح set of entities يعني ماتجيش تقولي ده جدول الطالب هقولك انت عامل جدول بحاله عشان طالب واحد؟

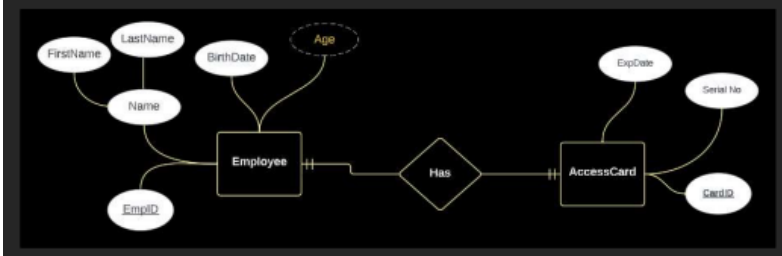
Convert One-to-One ==> Relational Schema

عشان تعبر عن ال one to one بتعمل الجدولين بال schema عادي بس بتيجي تزود عمود foreign key ومش هتفرق معاك تعمله في انهي جدول من الاتنين بس يفضل انك تعمل ال foreign key في الجدول اللي استخدامه اكتر في الداتا بيز عشان يكون فيه المعلومات كامله

زي كده



ERD

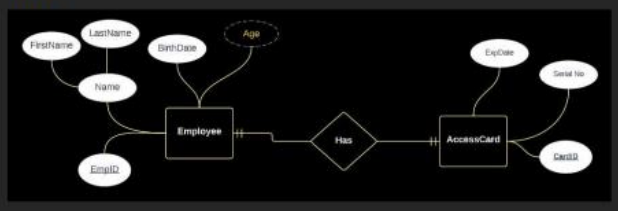


1. We create a table for each entity with it's attributes.
2. We take the primary key from any side and put it in the other side ☺

Relational Schema

OR

ERD



Relational Schema

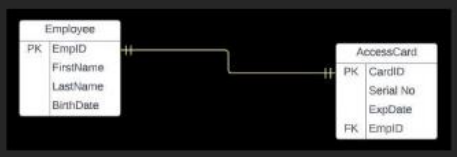
Employees

EmpID	FirstName	LastName	BirthDate	CardID
1	Mohammed	Abu-Hadhoud	6-11-1977	2
2	Ali	Amjad	12-3-2000	1
3	Maha	Omaran	11-6-2003	4
4	Fidaa	Safwan	6-6-1991	3

AccessCards

CardID	SerialNo	ExpDate
1	AH22002	1/1/2030
2	MA9938	1/1/2025
3	QA88381	1/1/2022
4	KW9K222	3/3/2026

Relational Schema



Employees

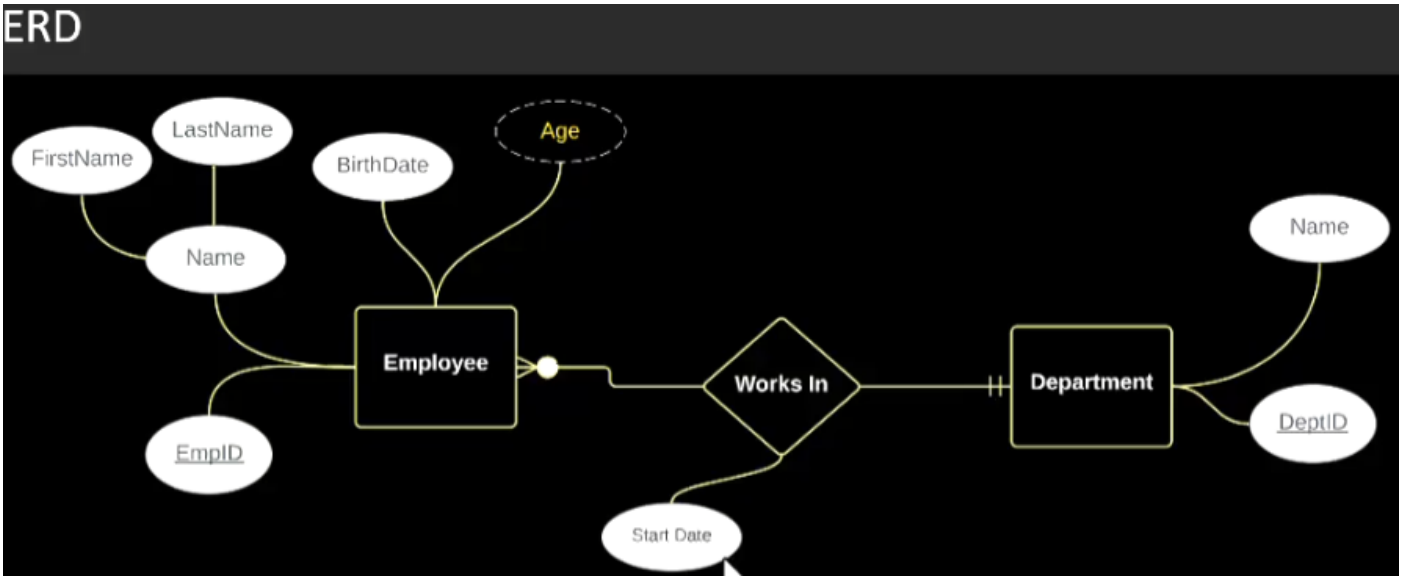
EmpID	FirstName	LastName	BirthDate
1	Mohammed	Abu-Hadhoud	6-11-1977
2	Ali	Amjad	12-3-2000
3	Maha	Omaran	11-6-2003
4	Fidaa	Safwan	6-6-1991

AccessCards

CardID	SerialNo	ExpDate	EmpID
1	AH22002	1/1/2030	2
2	MA9938	1/1/2025	1
3	QA88381	1/1/2022	4
4	KW9K222	3/3/2026	3

Convert One-to-Many/Many-to-One ==> Relational Schema

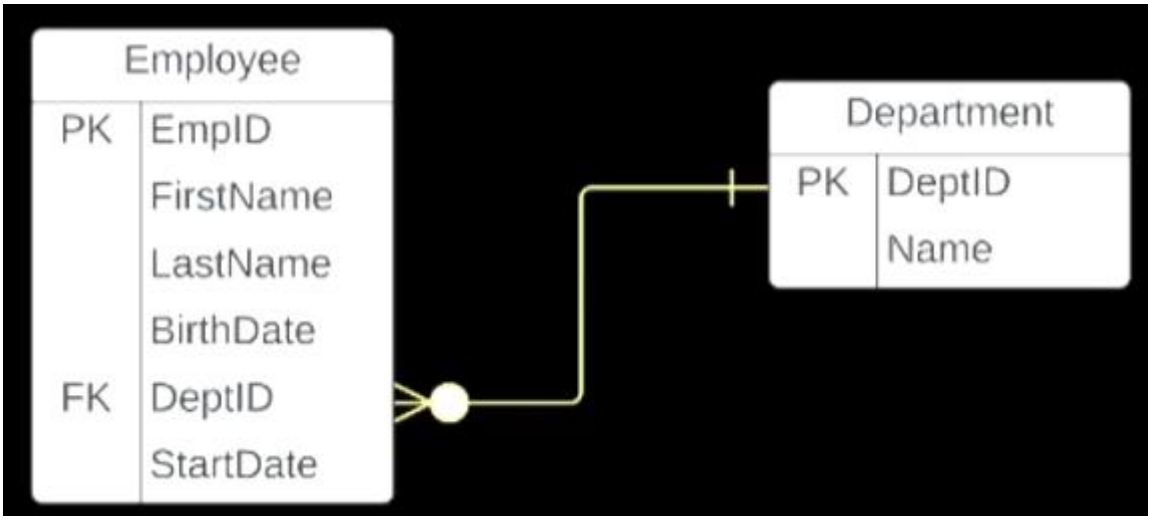
عاوزين نحول دي

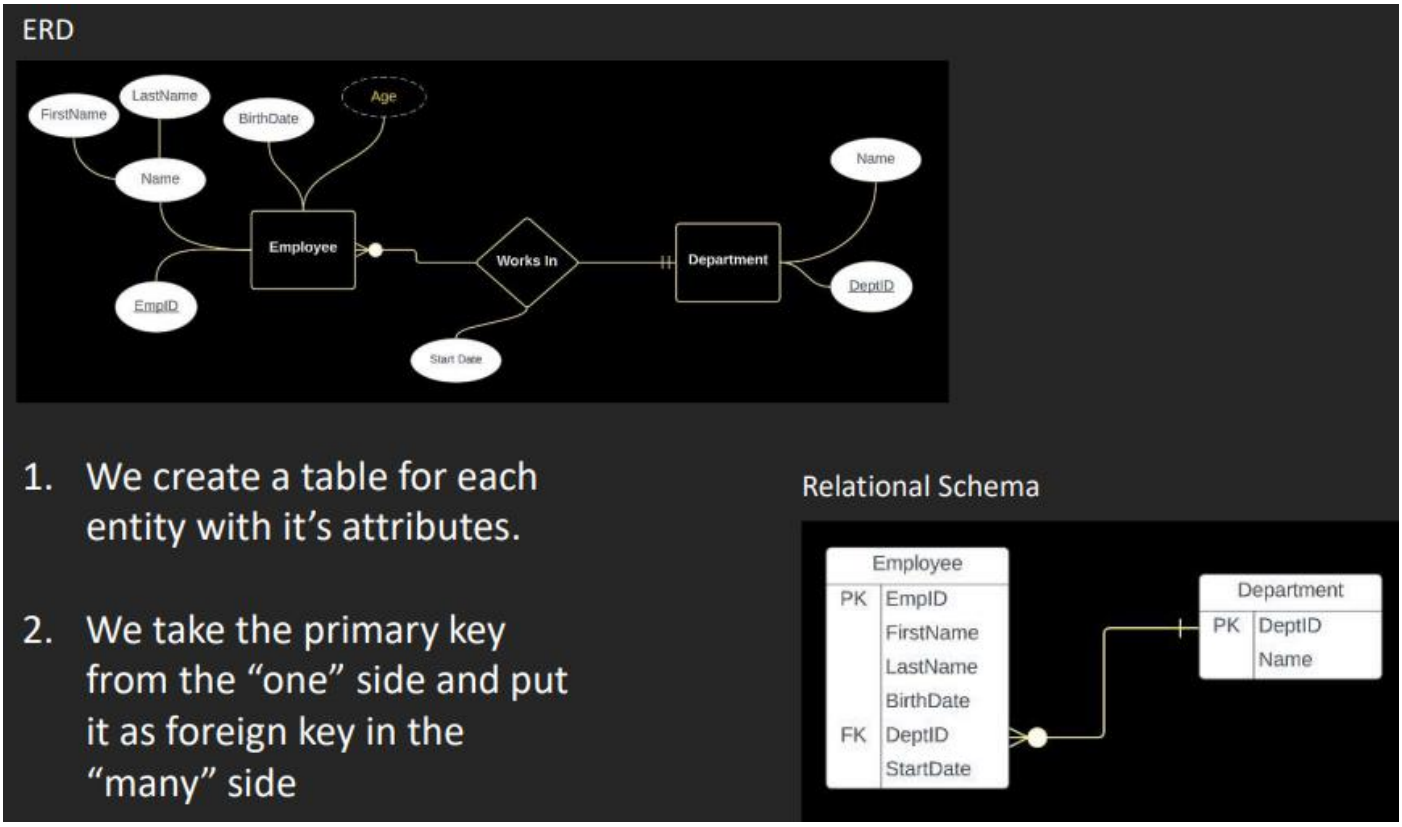


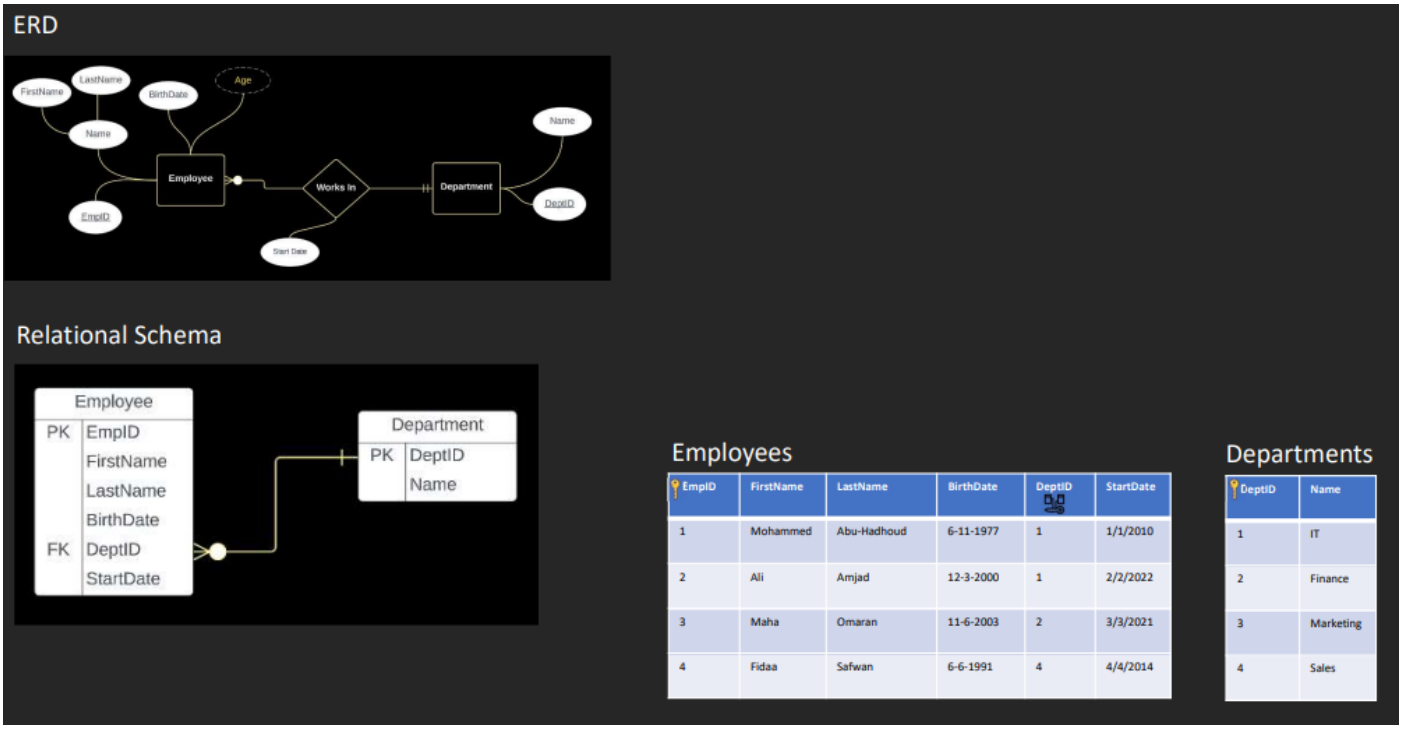
لاحظ ان العلاقه هيا many to one وانه لما تحصل العلاقه بين الجدولين عاوز اخزن ال

start date

قالك عشان تعمل ال schema بتروح للجهه اللي ال one ناحيتها بتاخد ال primary key بتاعها وترجع الناحيه التانيه بتعمله foreign key وبتحط معاه ال attribute المرتبط بالعلاقه اللي هوا في حالتنا ال start date زي كده







Convert Many-to-Many ==> Relational Schema

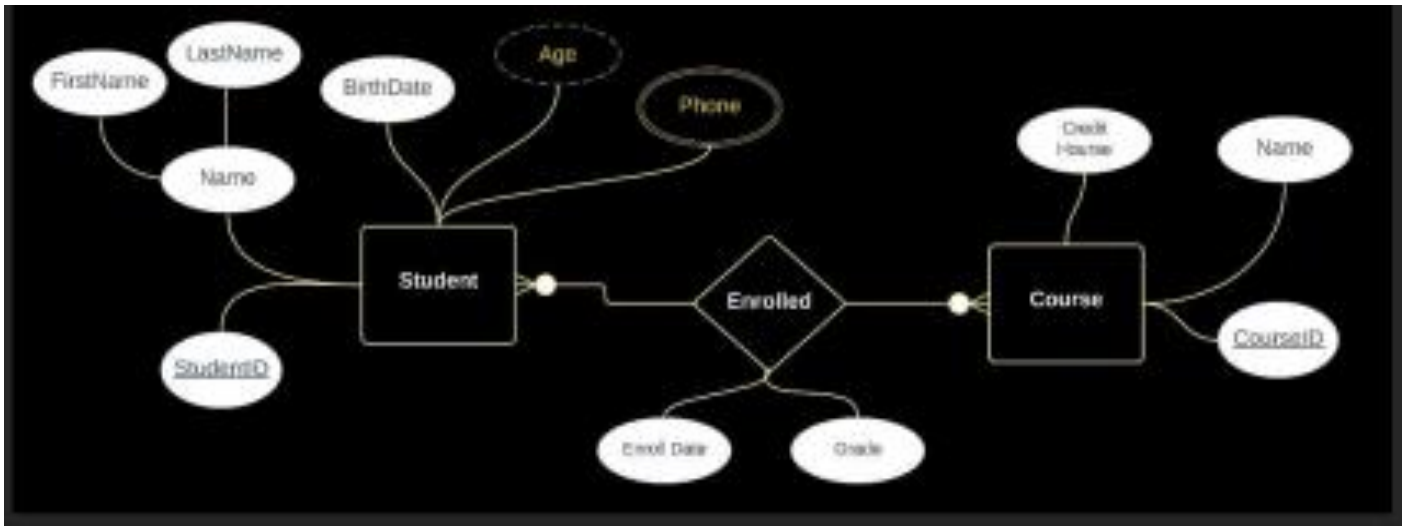
هنا بيقولك انك اول ماتلاقي علاقه many to many لازم تعمل جدول وسيط بينهم

الجدول ده بتعمله id خاص بيه وبتعمل اتنين foreign keys كل واحد فيهم بيشاور علي ال primary key بتاع كل جدول من الجدولين التانيين وفي الجدول الوسيط ده بتحط فيه أي معلومات ناتجه عن العلاقه دي

العلاقه بين الجدولين والجدول الوسيط هيا علاقه one two many او many to one حسب الاتجاه بس ال many بنكون ناحية الجدول الوسيط

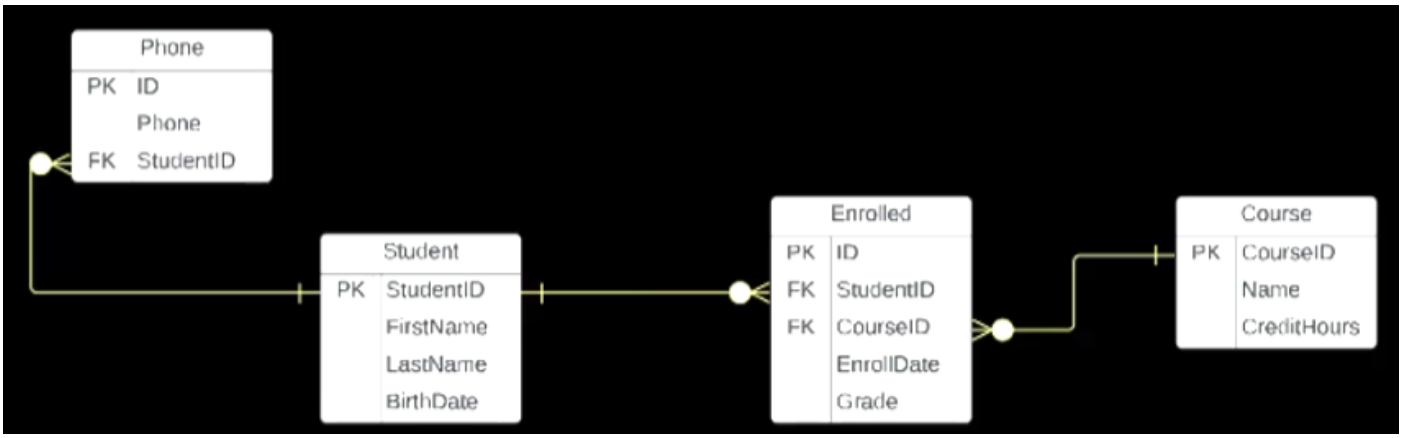
فيه ناس بتشيل ال primary key وبيستعملوا الاتنين foreign keys مع بعض علي انهم primary key هنا بيقولك ده هيخلي السيستم ابطأ وهيقيده بعدين لانه مثلا لو فيه طالب عايزيعيد نفس الكورس مش هيقدر او العميل مش هيقدر يطلب نفس الاوردر تاني لانك في الحاله دي انت هتكرر عمود ال primary key اللي في الجدول الوسيط فعشان تقوله اعملي record تاني الprimary key بتاعه مكن من نفس الكود ونفس الطالب هيرفض

عاوزين نحول ال erd ده



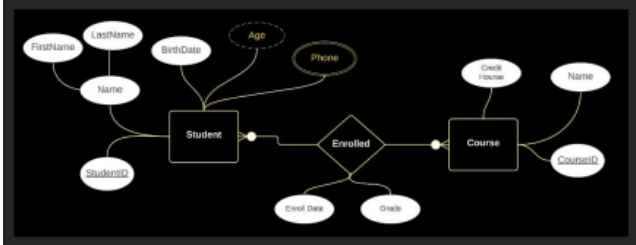
العلاقه فيه many to many وفيه عندي وعايز اخزن enroll date و grade لوحدهم

هنا بيقولك خزن ال enroll date وال grade مع الجدول الوسيط



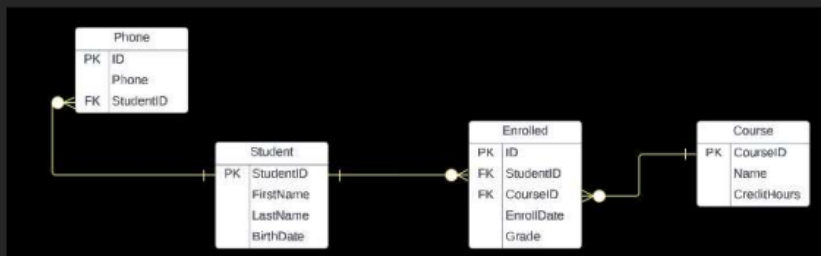
جدول ال phone ناتج عن ال multi valued attribute مالهوش علاقه بموضوعنا

ERD

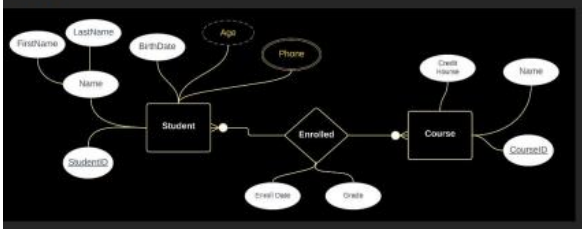


1. We create a table for each entity with it's attributes.
2. Create a bridge table for the many to many relationship.
3. Take the primary key from each tables in the relation and put them in the bridge table.
4. Add a primary key to the bridge table.

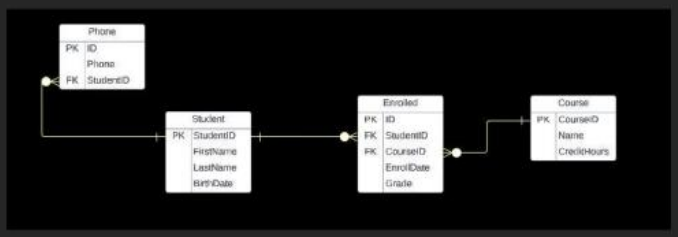
Relational Schema



ERD



Relational Schema



Phones

PhoneID	StudentID	Phone
1	1	07729929
1	1	079882882
3	2	059939921
4	3	065500501

Students

StudentID	FirstName	LastName	BirthDate
1	Mohammed	Abu-Hadhoud	6-11-1977
2	Ali	Amjad	12-3-2000
3	Maha	Omaran	11-6-2003
4	Fidaa	Safwan	6-6-1991

Enrollments

EnrollmentID	StudentID	CourseID	EnrollDate	Grade
1	1	2	1/1/2000	95
2	1	3	20/1/2005	80
3	2	2	5/5/2022	50
4	3	4	15/1/2021	45
5	3	4	25/6/2022	Null

Courses

CourseID	Name	CreditHours
1	OOP	3
2	Database	3
3	C#	3
4	Marketing	1

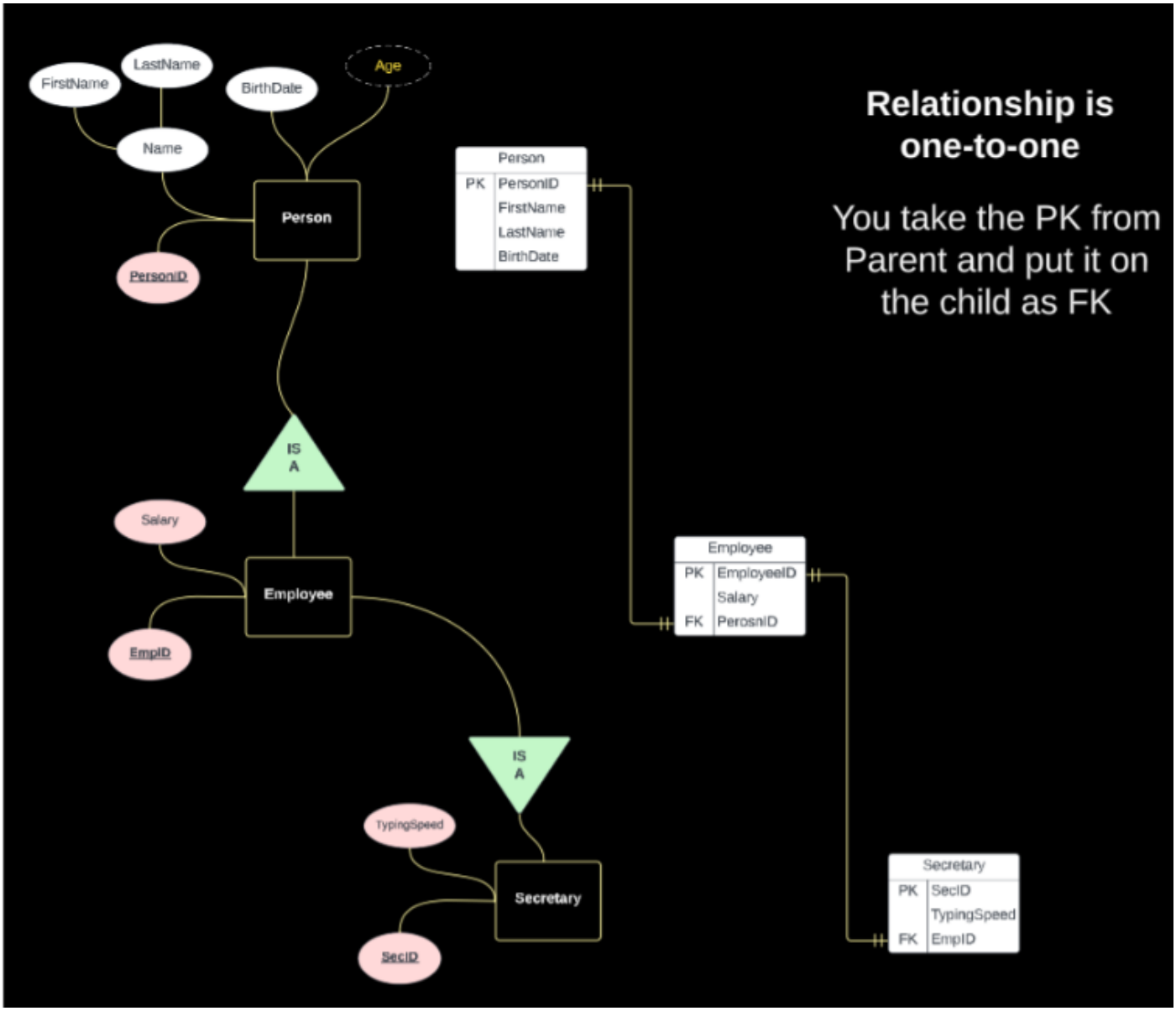
Generalization and Specialization to Relational Schema

هنا بيقولك عشان تستخدم ال generalization انت بتربط بين الجدول الاب والجدول الابن بعلاقه واحد لواحد وال foreign key بتحطه في الجدول الابن لانك لو حطيته في جدول الاب هتعمل بيه ايه؟

Generalization and Specialization to Relational Schema

To convert Generalization of Specialization into Relational Schema:

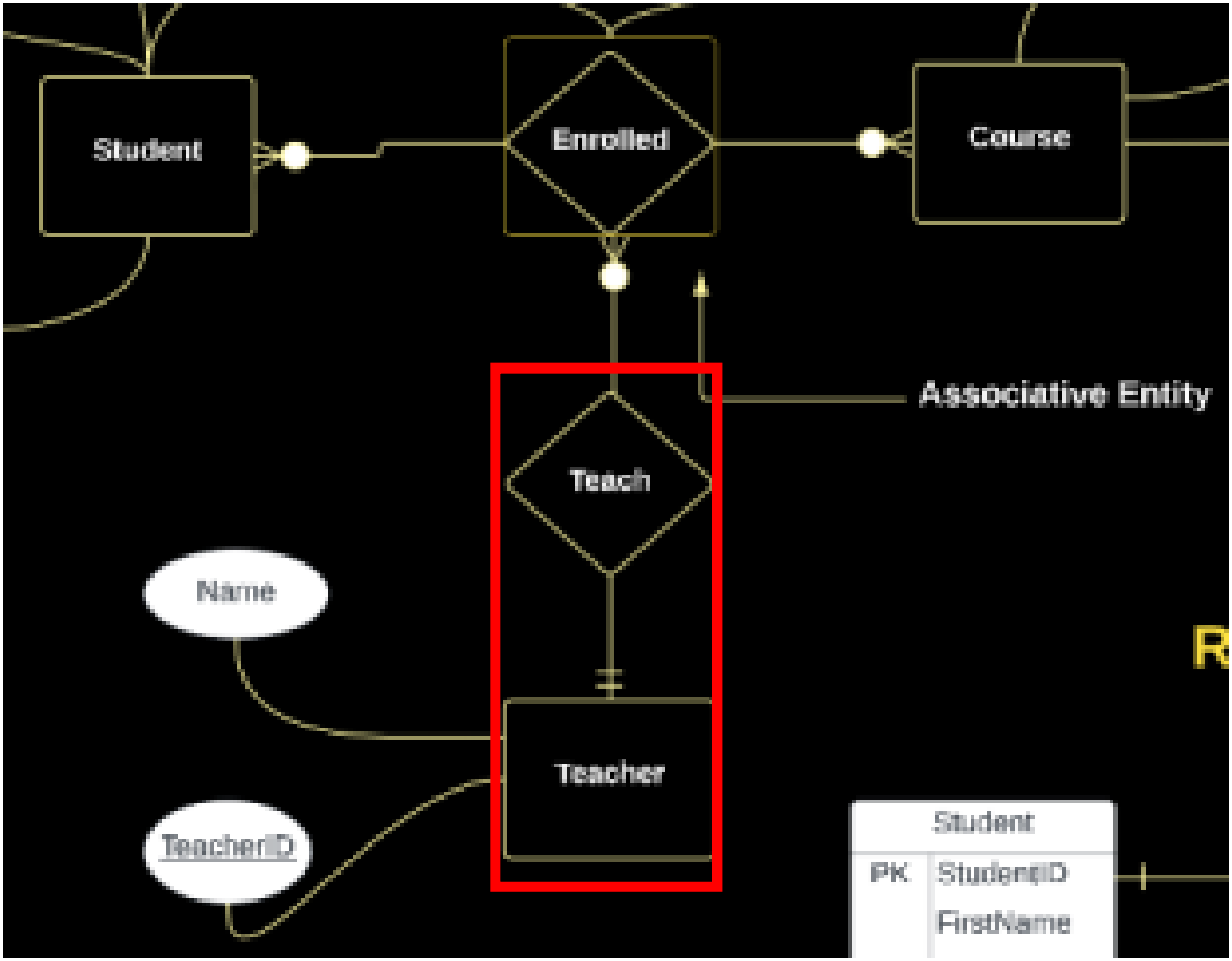
- Remember that the relationship is always one-to-One.
- You take the primary key from the parent entity and put it as foreign key in the child entity.

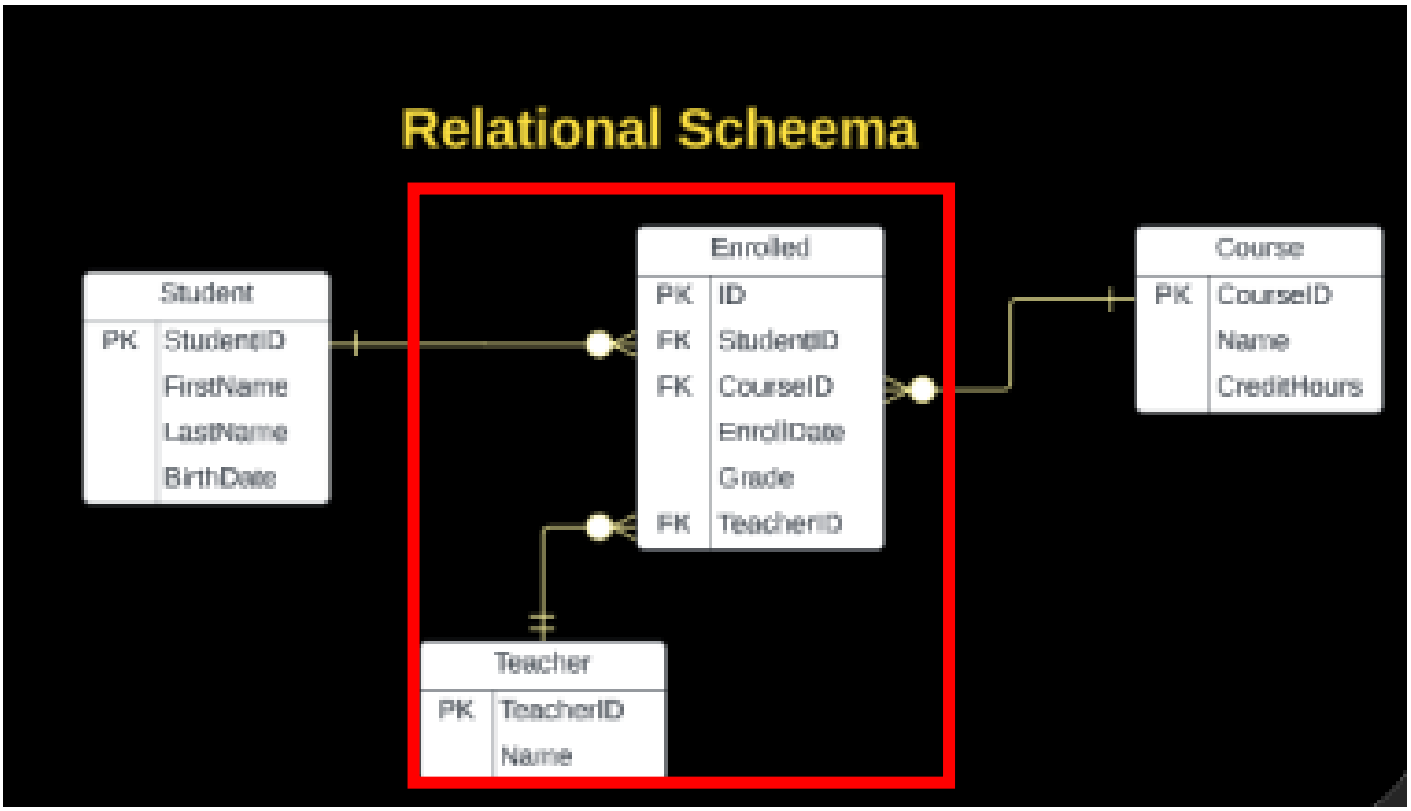


Convert Associative Entity to Relational Schema

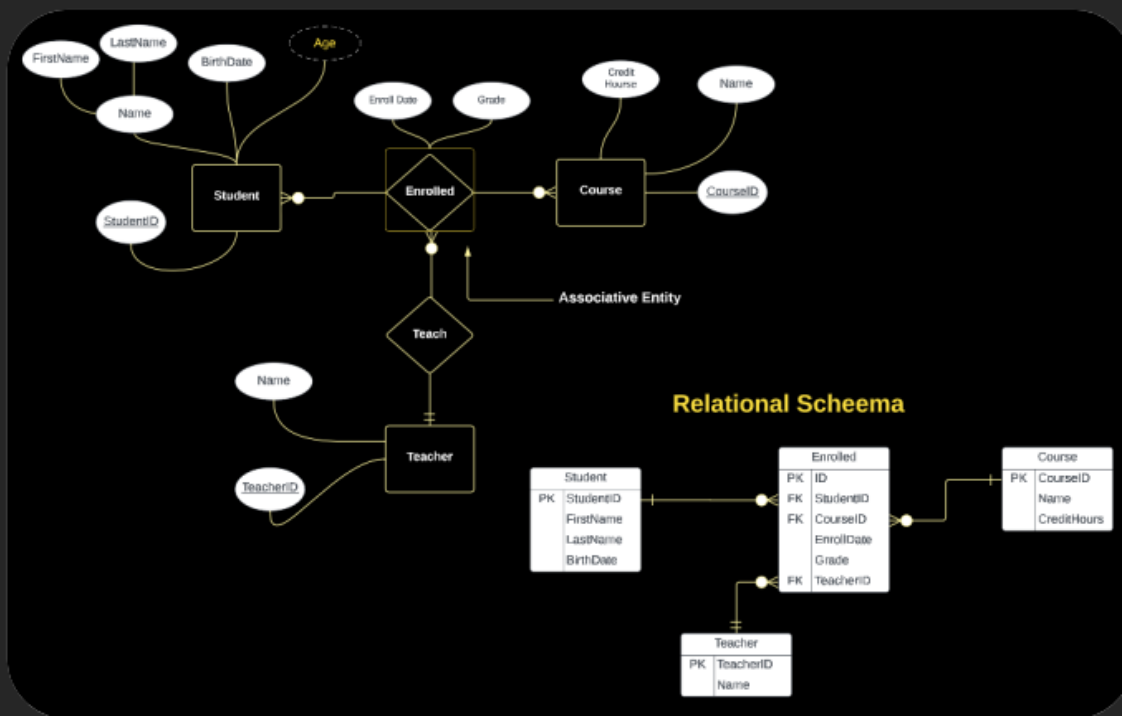
ال associative entity هيا ال entity اللي مبنيه علي علاقه تانيه والعلاقه التانيه دي بتكون many to many

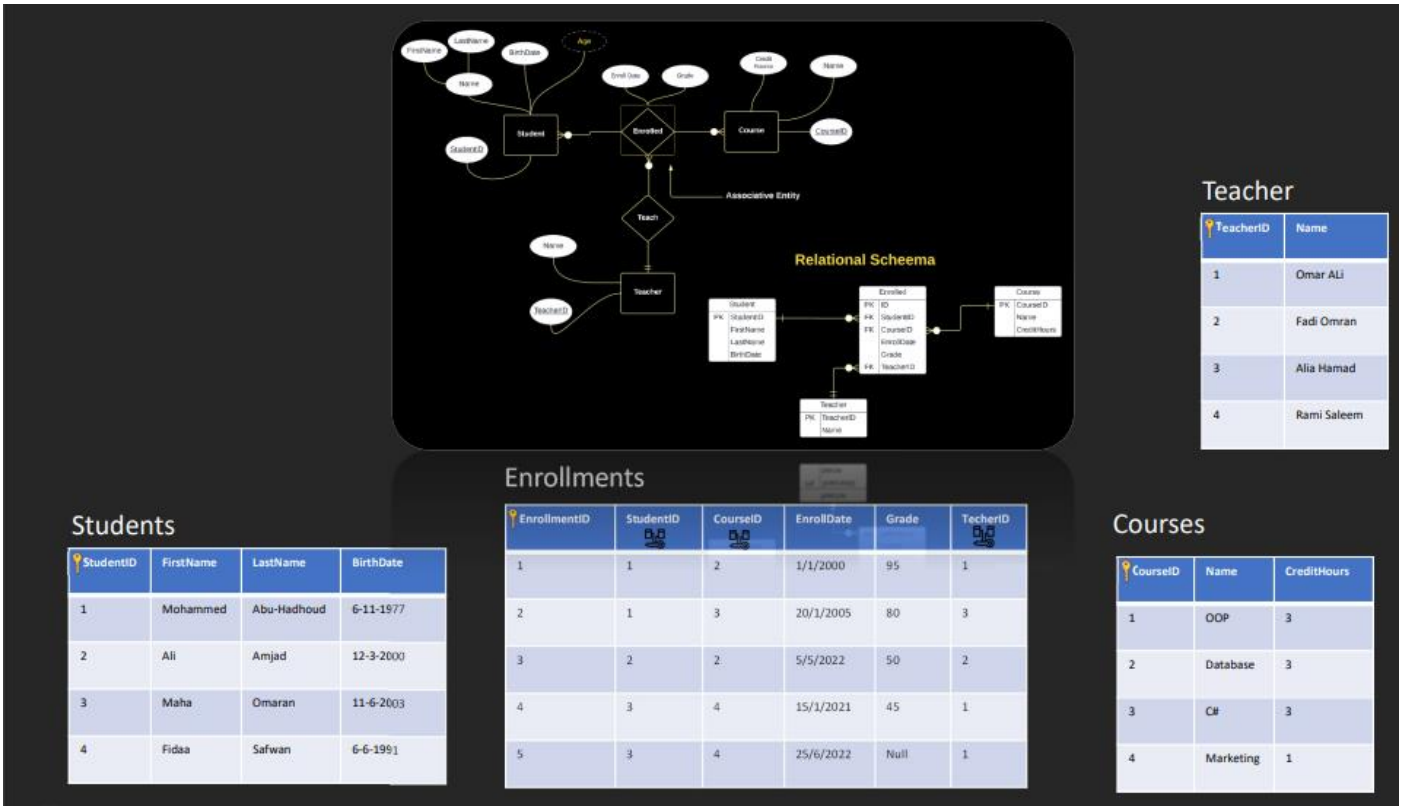
فعشان امثلها كل اللي بعمله اني بعمل جدول وبربطه بعلاقه one to many بالجدول الوسيط


Student
Enrolled
Course
Associative Entity
Teach
Name
Teacher
TeacherID
Student
PK
StudentID
FirstName


Relational Scheema
Student
PK
StudentID
FirstName
LastName
BirthDate
Enrolled
PK
ID
FK
StudentID
FK
CourseID
EnrollDate
Grade
FK
TeacherID
Course
PK
CourseID
Name
CreditHours
Teacher
PK
TeacherID
Name

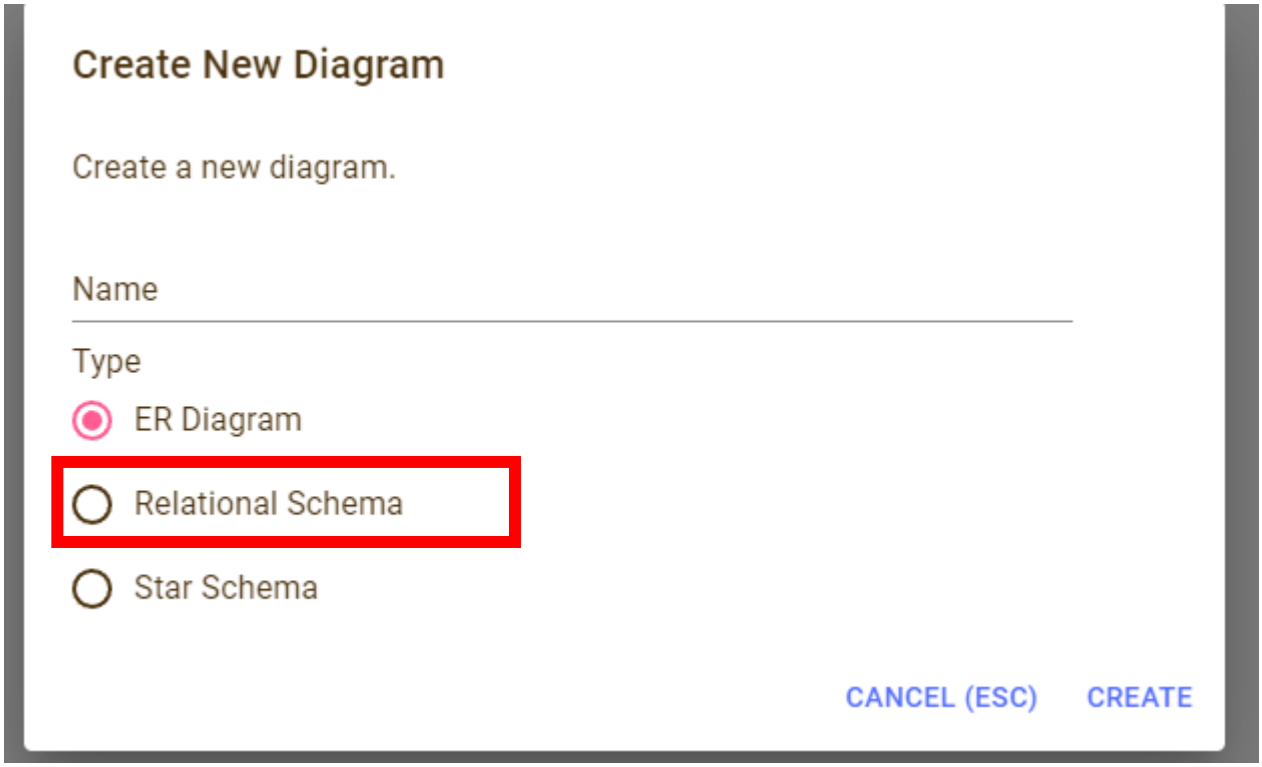
1. We create a table for each entity with it's attributes.
2. Create a bridge table for the many to many relationship.
3. Take the primary key from each tables in the relation and put them in the bridge table. Also Take the primary key from the associated relation and also put it in the bridge table .





How to create Relational Schema on ERDPlus.com?

هنا عشان تعمل ال relational shema في الموقع بتختارها من هنا



والباقي سهل

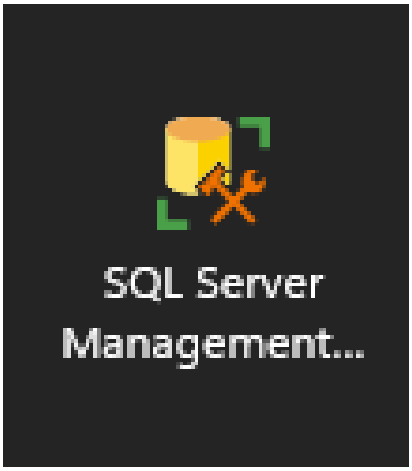
(revision) What is SQL?

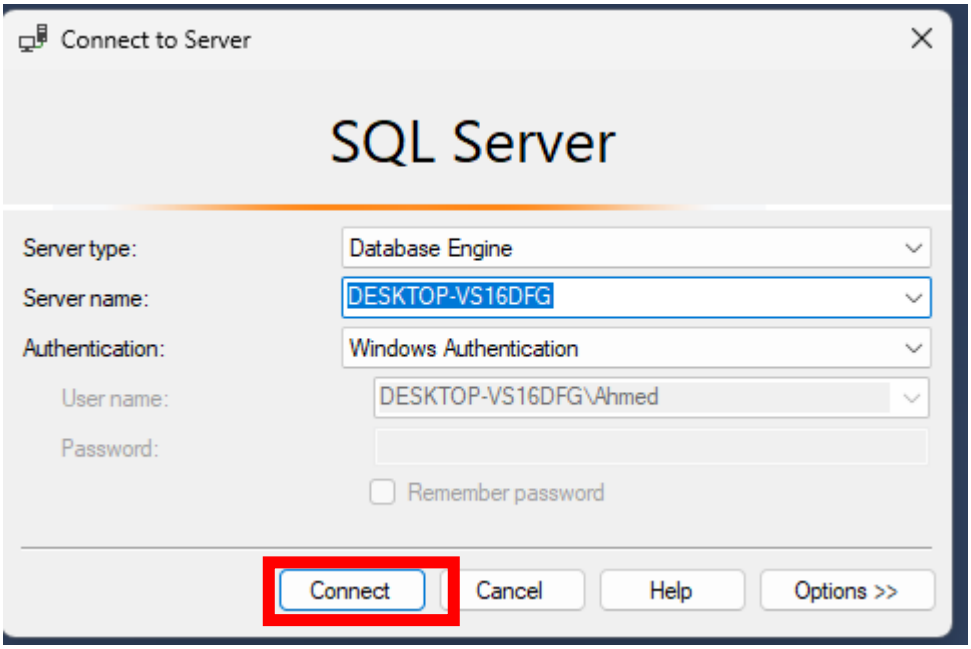
https://programmingadvices.com/courses/database-level-1-sql-concepts-and-practice/lectures/46511156

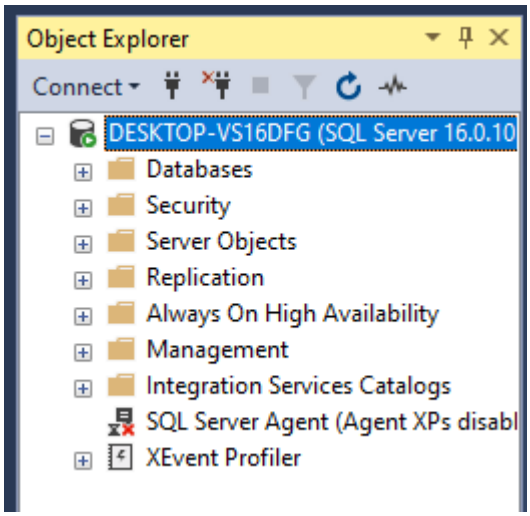
SQL - Data Definition Language - DDL

Create Database

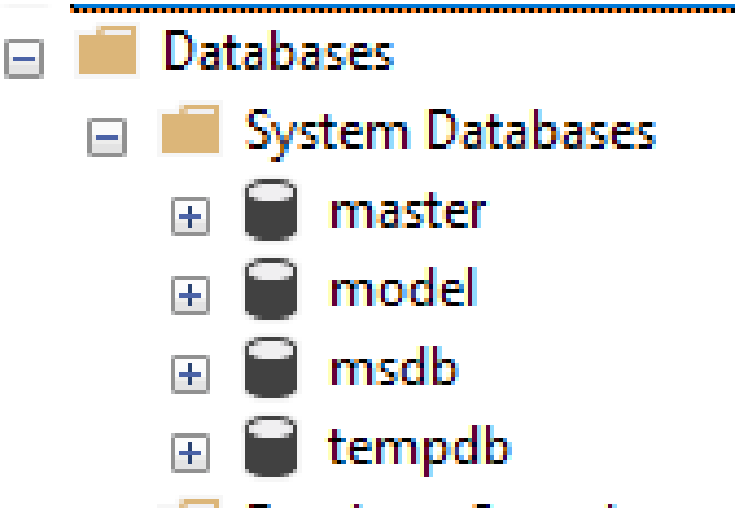
عشان نتواصل مع الداتا بيز لازم الاتصال يتم عن طريق sql management studio



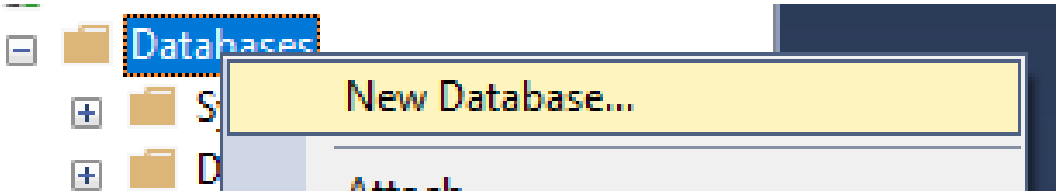


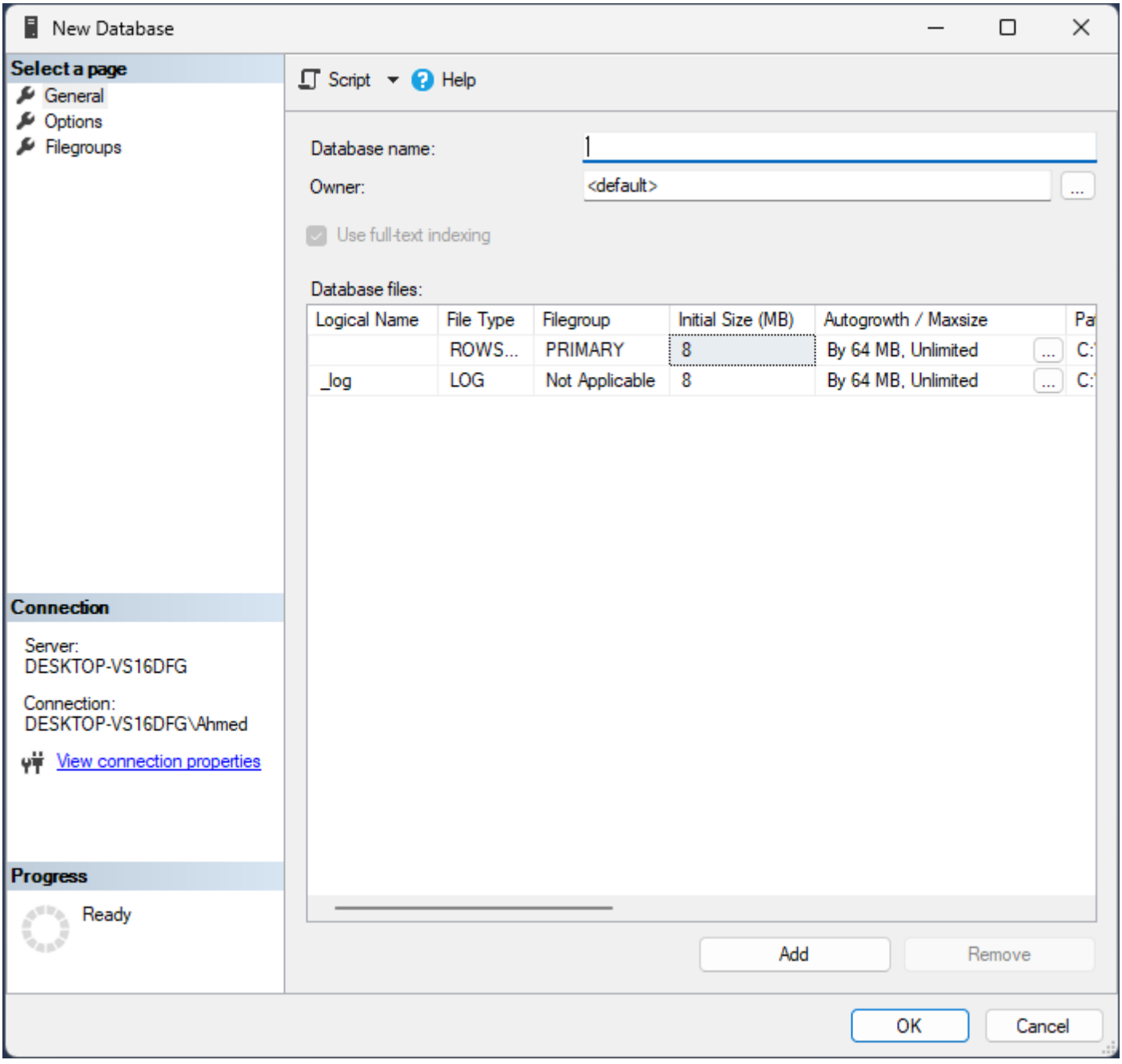


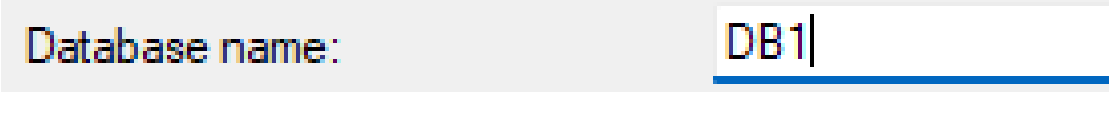
وزي ماقولنا اوعي تلعب في دول



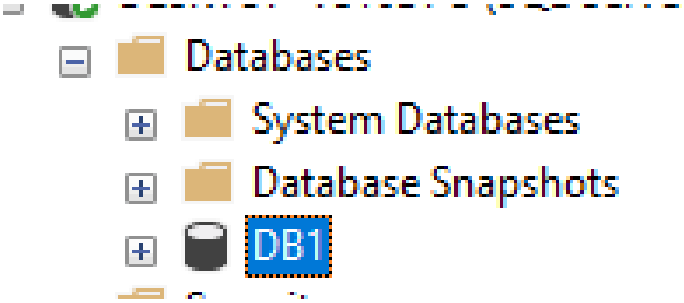
وهنا بيقولك انه فيه طريقتين عشان تعمل داتابيز طريقه بالماوس وطريقة بالكود







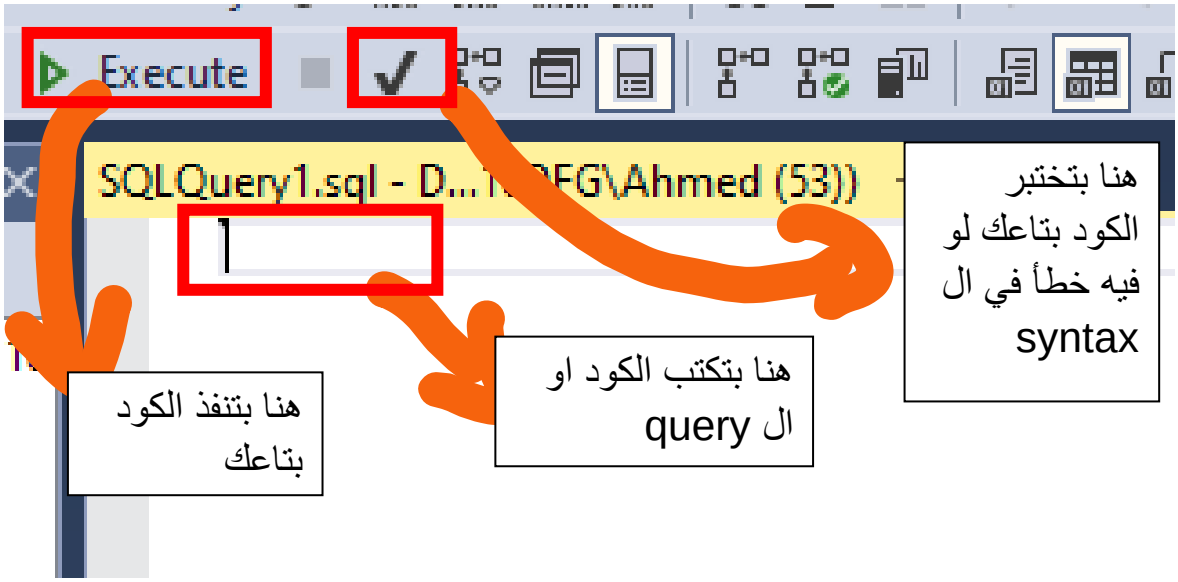
وبعدين دوس OK هتظهرلك في الجنب



طيب دي الطريقه الاولي

الطريقه التانيه من خلال الكود وهنا بيقولك انه أي حاجه في ال SQL SERVER بتقدر تعملها بالماوس وبالكود عادي

عشان تكتب أي كود بتدوس علي NEW query من القايمه اللي فوق

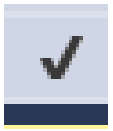


عاوزين بقي نعمل داتا بيز جديده من خلال الكود

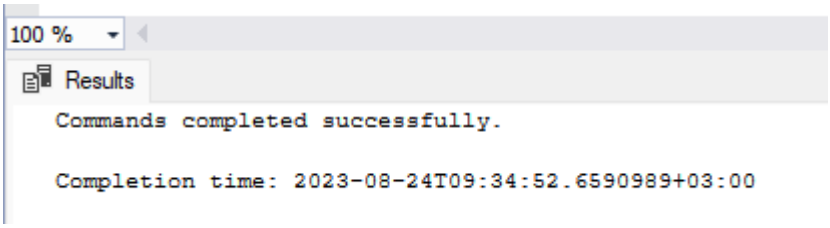
قالك كل اللي بتعمله انك تكتب create وبعدين الحاجه اللي عايز تعملها في حالتنا هنا هنكتب database وبعدين بتعين اسم للداتا بيز

```
create database DB2
```

وبعدين لو عايز تعمل check عالكود بتدوس هنا

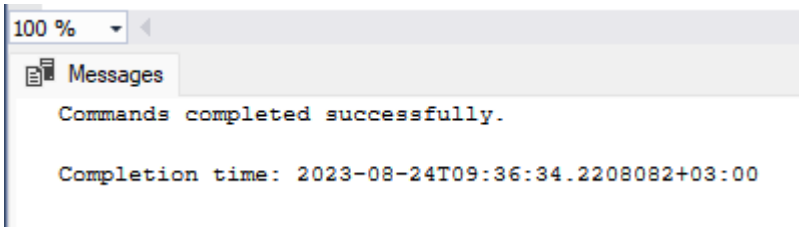


والنتيجه بتظهر تحت

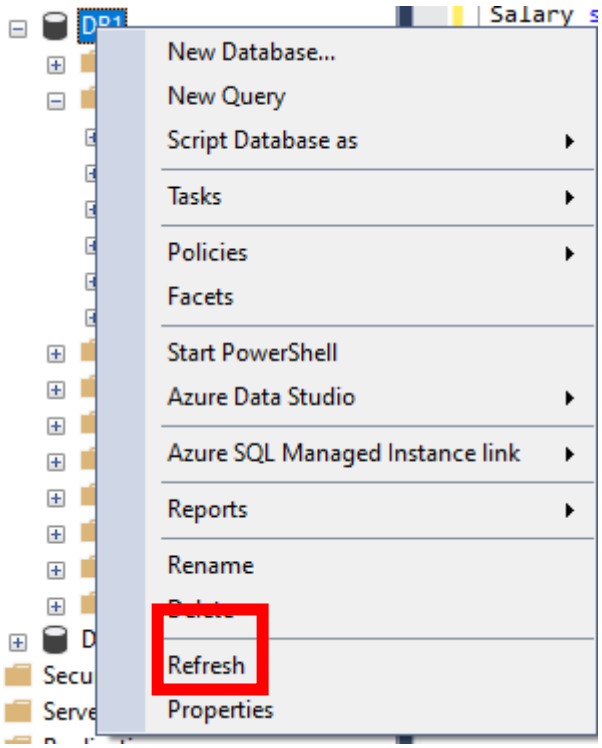


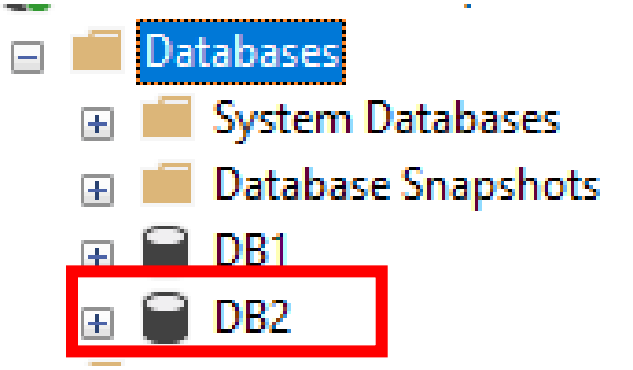
لو عايز تنفذ الكود من هنا

والنتيجه برضه بتظهر تحت



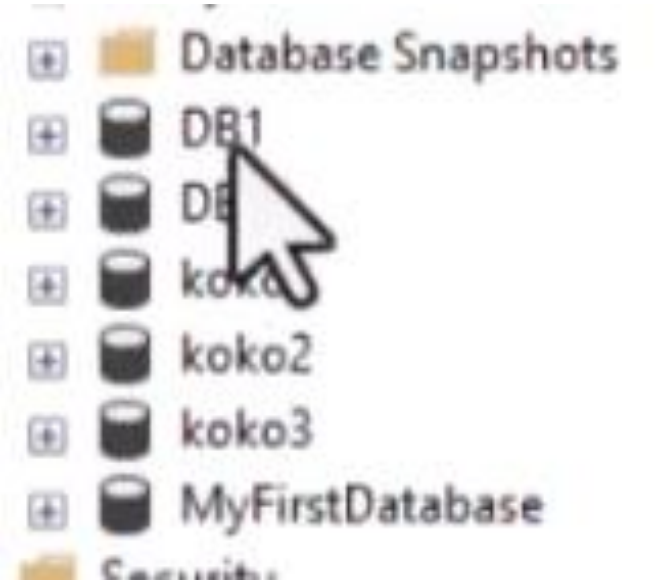
وبعدين بتيجي تعمل refresh للقايمه عشان تظهرلك الداتا بيز





وهنا هوا بيعمل اكتر من واحده في نفس الوقت

```
create database koko;

create database koko2;

create database koko3;
```



SQL CREATE DATABASE Statement

Before we can work with database tables, we must create a database first.

The `CREATE DATABASE` statement is used to create database tables. For example,

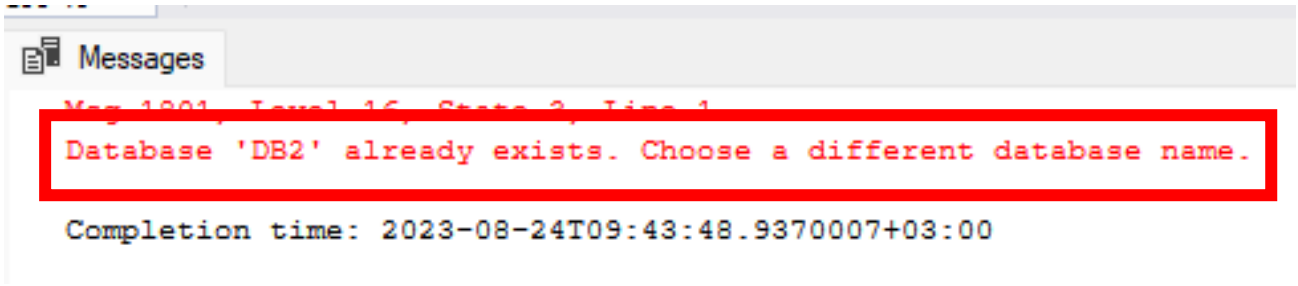
```
CREATE DATABASE Koko;
```

Here, the SQL command creates a database named Koko.

Create Database IF NOT EXISTS

هنا بيقولك لو حاولت تعمل داتا بيز جديده بنفس الاسم بتاع داتا بيز موجوده عندك قبل كده هيضرب منك

```
create database DB2
```



عشان تعمل comment هنا بتكتب – علامتين طرح

```
--create database DB2
```

طيب احنا دلوقتي عاوزين نتجنب الخطأ ده ونقوله لو مالقيتش داتا بيز بالاسم ده اعمل واحده جديده

طيب جملة ال if statement في ال sql server بتتكون من ايه؟

قالك بتكتب If والشرط وبعدها بتكتب begin وبعدين الامر اللي عاوز تنفذه لو الشرط اتحقق ولما تخلص تكتب end

دي كده اول حاجه

تاني حاجه وهيا NOT ودي زيها زي علامه ال ! في ال C++ او ال C# اللي بتغير ال true بتقلبه false والعكس

تالت حاجه وهيا ال function اللي اسمها exists ودي بتاخد جمله او sub query وترجع Boolean بتقولك ال query دي موجوده ولا لا او تم تنفيذ الكود ده ولا لا

تعالي بقي نركب البازل

اول حاجه هحط ال if statement

```
IF
BEGIN

END;
```

بعدين انا عايز أقوله لو مش موجوده

```
IF NOT EXISTS()
BEGIN

END;
```

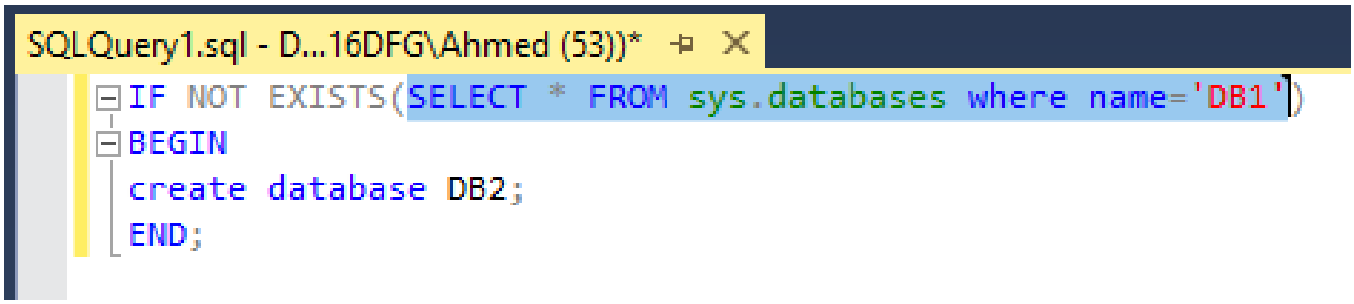
عشان أقوله يختار الداتا بيز اللي اسمها DB1 مثلا بكتب الجمله دي واللي معناها اختار كل حاجه من ال DATABASES اللي موجوده في السيستم بشرط انه الداتا بيز يكون اسمها كذا

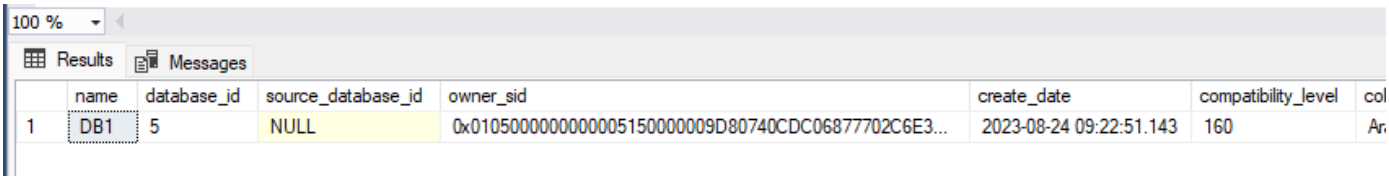
```
IF NOT EXISTS(SELECT * FROM sys.databases where name='DB1')
BEGIN

END;
```

كده ناقص اكتب الكود اللي عايز انفذه

```
IF NOT EXISTS(SELECT * FROM sys.databases where name='DB1')
BEGIN
create database DB2;
END;
```

فيه حاجه هنا عاوز يقولهالك وهيا انك لو حددت كود معين وجيت تشغل الأوامر اللي هيتنفذ هوا الكود المظلل بس





CREATE DATABASE IF NOT EXISTS

If there is already a database with the same name, SQL will throw an error while creating a database.

In such situations, we can use the the following code to create a database only if there is no existing database with the same name. For example,

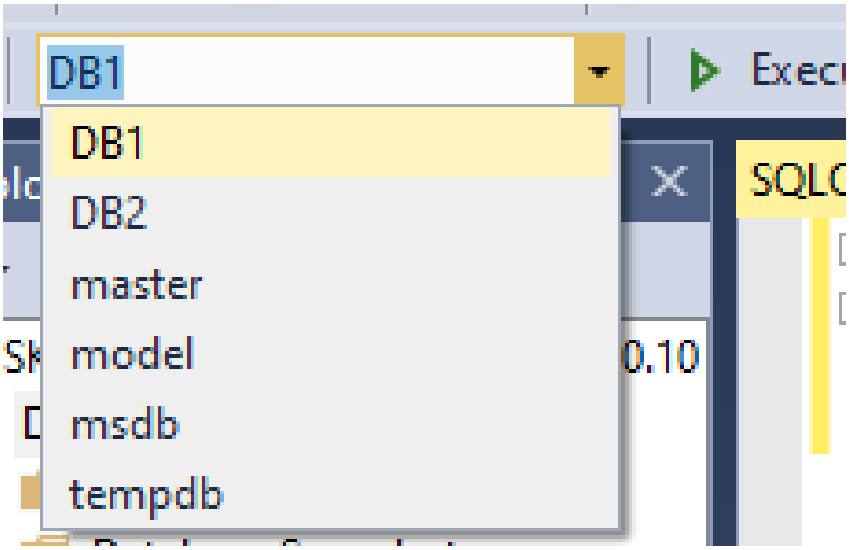
```
IF NOT EXISTS(SELECT * FROM sys.databases WHERE name = 'koko')
 BEGIN
   CREATE DATABASE koko;
 END
```

Here, the code creates a database named koko only if there is no existing database with the same name.

Switch Database

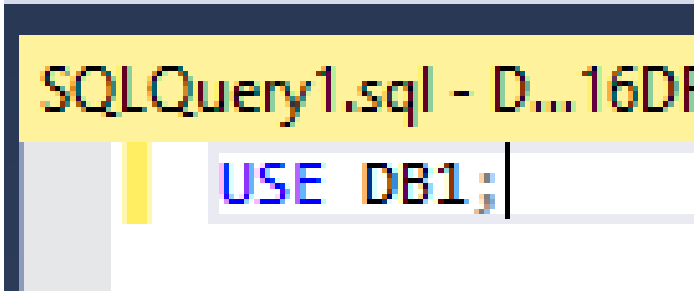
هنا بيقولك انك ممكن تكون بتكتب query علي داتا بيز غير اللي انت عايزها ودي مصيبه فلازم تتاكد من الداتا بيز اللي انت عايز تنفذ عليها الأوامر بتاعتك

عشان نعرف انهي داتا بيز اللي بيتنفذ عليها الأوامر بنلاقي اسمها مكتوب هنا وبنقدر نغيره لو عايزين



طيب ده ساعات وبيحصل اننا ساعات بيكون علي الله حكايتنا وبننسي نبص عليها او نغيرها

فممكن نحدد الداتا بيز عن طريق الامر use وبعده نكتب اسم الداتا بيز



Switch Databases

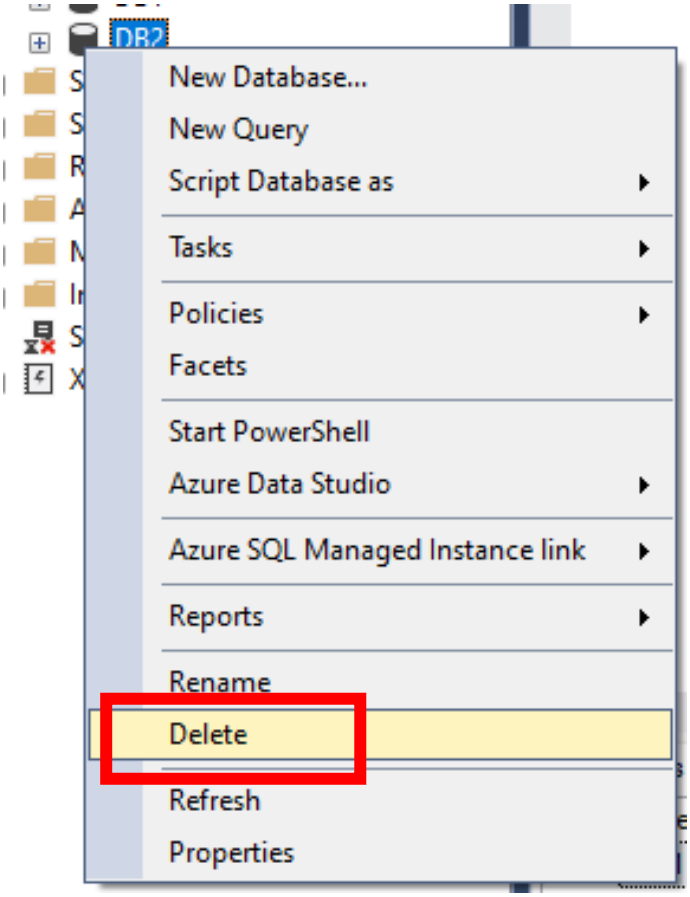
We have to work across multiple databases from time to time. To switch between available databases, we can run the following statement.

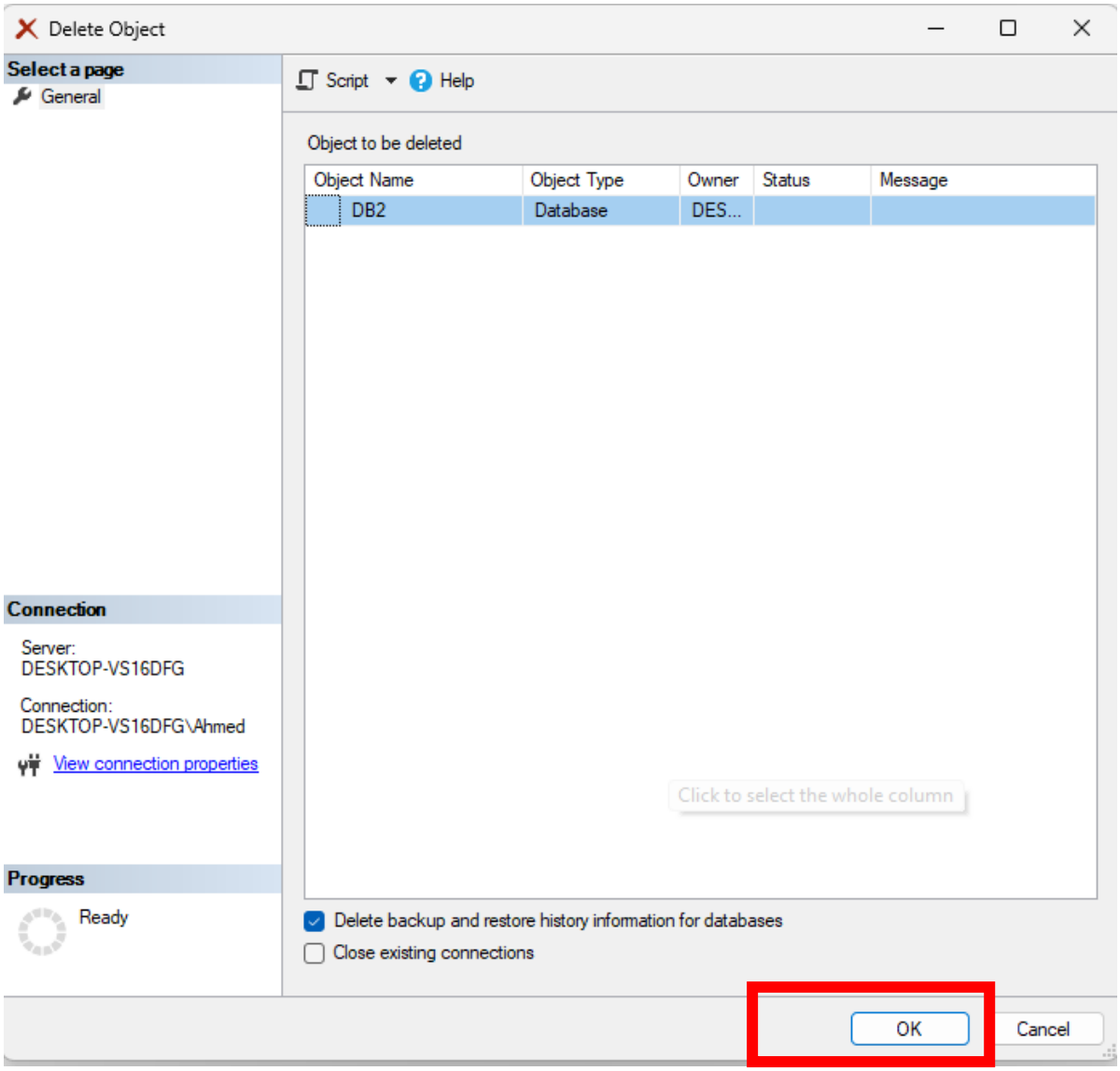
```
USE koko;
```

This code selects the koko database and all SQL operations will be performed inside this database.

Drop Database

عاوزين نلغي او نحذف داتابيز معينه لو بالماوس فالطريقه اهي





لو عايزين نعملها بالكود بنكتب drop وبعدها ايه نوع الحاجه اللي عاوزين نحذفها وفي حالتنا هتكون database وبعدين اسم الداتا بيز

```
drop database DB1;
```

SQL DROP DATABASE Statement

In SQL, `DROP DATABASE` is used to delete the database in our Database Management System. For example,

```
DROP DATABASE koko;
```

Here, the SQL command will delete a database named koko.

Also make sure you have **admin** or **DROP** permission to run this command.

Note: When we delete a database, all tables and records within a database are also deleted.

Drop Database IF EXISTS

مش محتاجه شرح لانه نفس اللي عملناه قبل كده عاوزين نقوله لو موجوده احذفها نفس اللي عملناه بس بدل CREATE هنكتب DROP وهنشيل كلمه NOT

```
IF EXISTS(SELECT * FROM sys.databases WHERE
NAME='DB1')
BEGIN
DROP database DB1;
END ;
```

Drop DATABASE IF EXISTS

If there is no database with the same name, SQL will throw an error while dropping a database.

In such situations, we can use the the following code to drop a database only if there is existing database with the same name. For example,

```
IF EXISTS(SELECT * FROM sys.databases WHERE name = 'koko')
 BEGIN
   Drop DATABASE koko;
 END
```

Here, the code drops a database named koko only if there is an existing database with the same name.

Create Table

قبل ماندخل نعمل جدول جديد فيه حاجه لازم نعرفها وهناخد نبذه عنها دلوقتي

ال int وده عادي مش هنختلف عليه وعارفينه

ال char() وده بيعبر عن string وبياخد رقم الرقم ده هوا عباره عن عدد الاحرف اللي عاوز اليوزر ميزودش عنها لو قولتله 3 احرف هيدخل لحد 3 بس ولو عملتها 50 حرف واليوزر دخل حرف واحد بس هيبدل ال 49 حرف بمسافات فهيكون في مساحه مهدره

المشكله التانيه اللي فيه انه مش بيحفظ غير حروف انجليزيه بس ماتقدرش تكتب جواه باي لغه تانيه

ال nchar() وده زي اللي فات بس بيقبل أي لغه تانيه عادي

ال varchar() وده زي ال char() بس لو جيت قولتله انه عدد الحروف المسموح بيها 50 واليوزر دخل حرف واحد بس بيخزن حرف واحد بس من غير مسافات زياده وده احسن

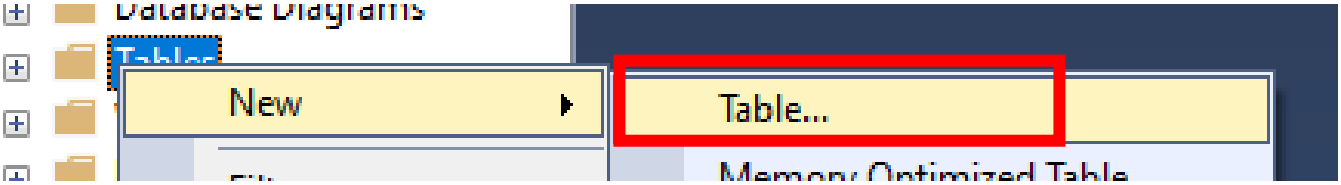
ال nvarchar() وده زيه زي ال varchar() بس بيقبل باي لغه غير الإنجليزي وهوا المفضل في الاستخدام

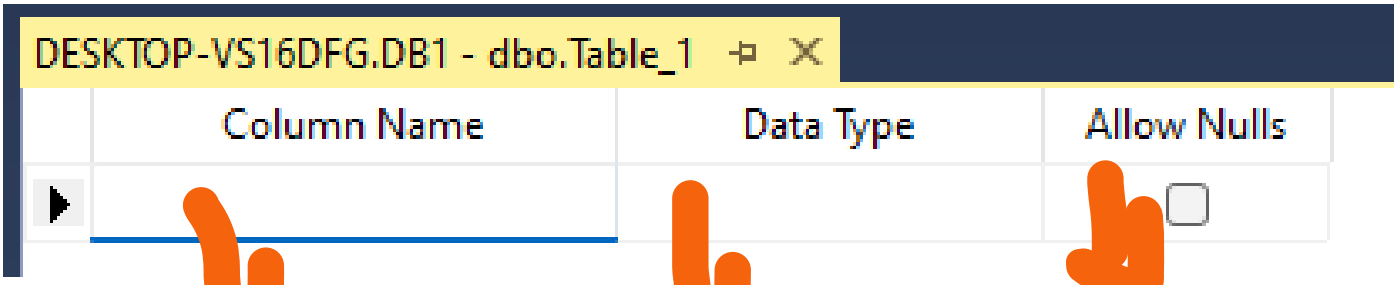
ال smallmoney هنتكلم عليها بعدين

طيب تعالي بقي نعمل جدول جديد بالماوس

هنفتح الداتا بيز اللي عايزين نحط فيها الجدول

وبعدين كليك يمين علي الملف اللي اسمه tables



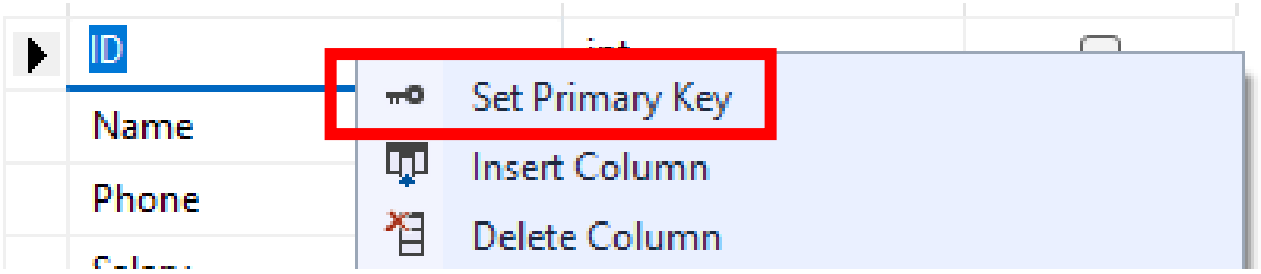


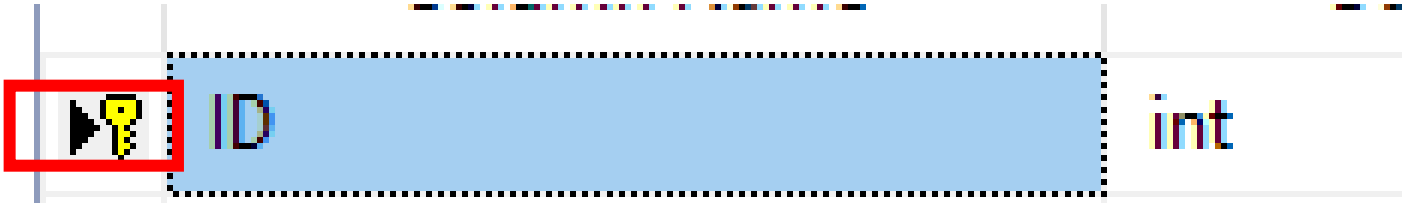
عايز العمود يقبل ال null ولا لا يعني عايز اليوزر يدخل في الداتا اجباري ولا لا	هنا ال data type	هنا بتكتب اسم العمود

DESKTOP-VS16DFG.DB1 - dbo.Table_1*

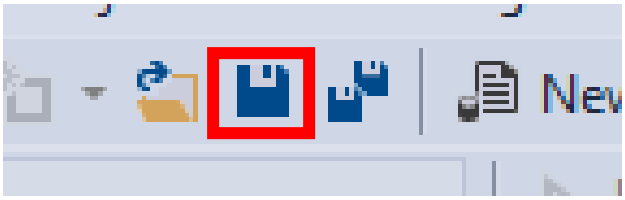
Column Name	Data Type	Allow Nulls
ID	int	☐
Name	nvarchar(50)	☐
Phone	nvarchar(10)	☑
Salary	smallmoney	☑
		☐

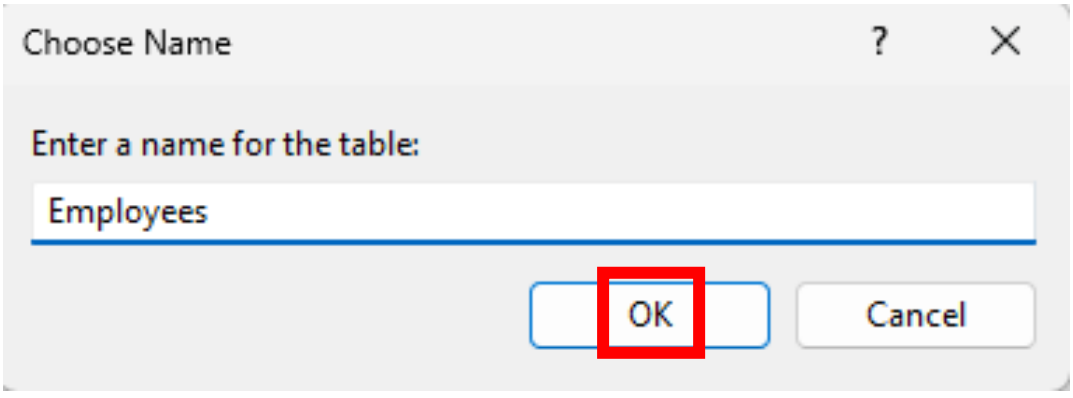
طيب دلوقتي عايزين نخلي ال id يكون primary key

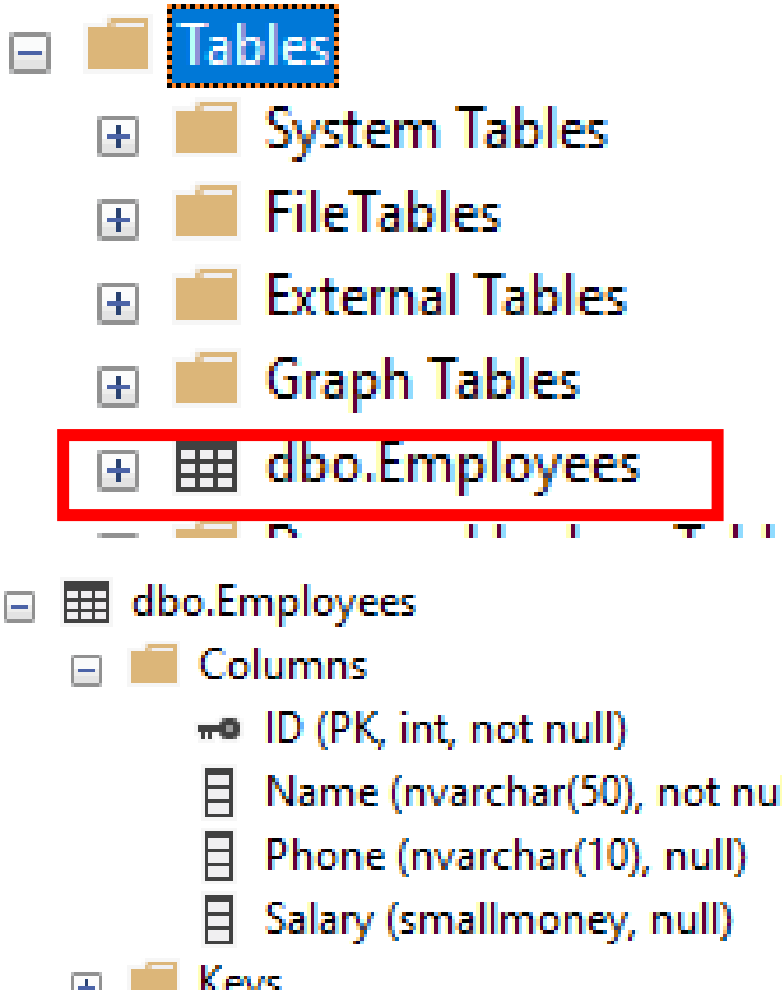




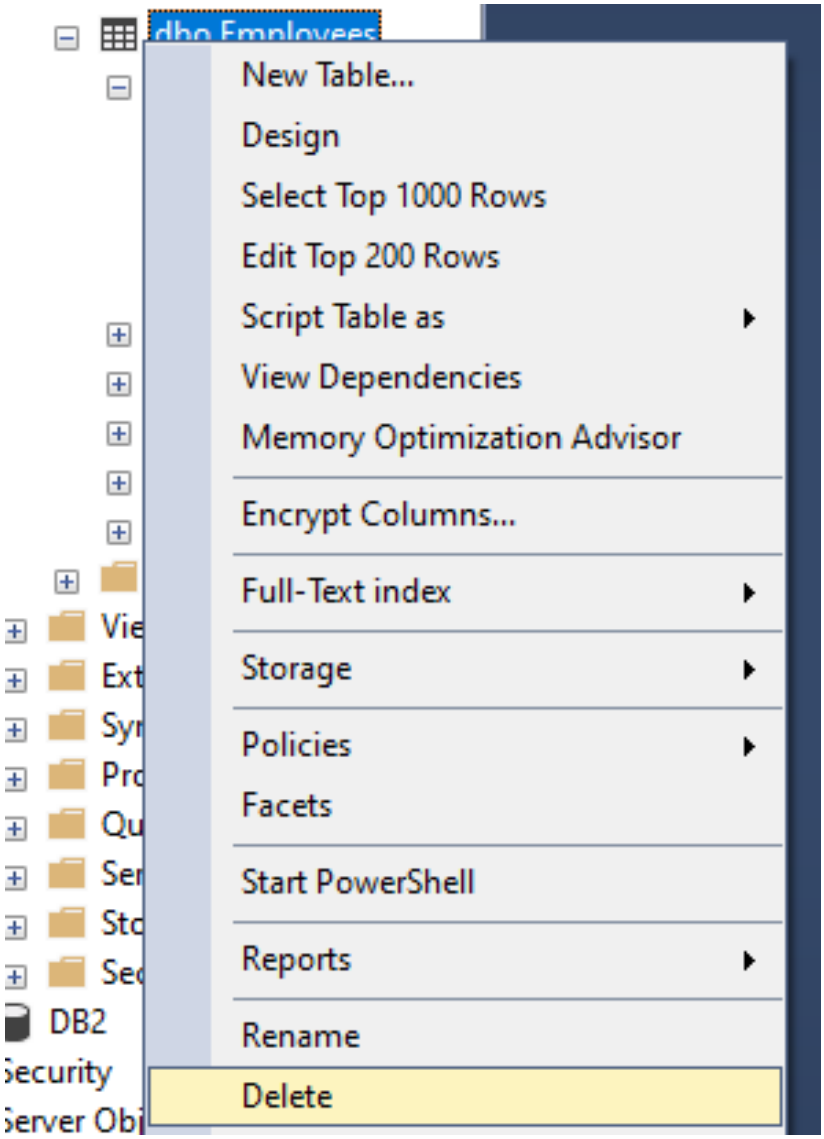
عاوزين نحفظ الجدول







دلوقتي هنحذف الجدول ده عشان نروح نعمله بالكود



تعالي بقي نعمله بالكود

خلينا فاكرين انه اولمر ال DDL خاصه بالعمليات علي الداتابيز نفسها

واحنا لحد دلوقتي عرفنا امرين واحد اسمه create وواحد اسمه delete

طيب عشان نعمل جدول جديد هنستخدم الامر create وبعدين نسأل احنا عاوزين نعمل ايه ؟

عاوزين نعمل جدول نقوم نكتب table وبعديه نحط اسم للجدول ونفتح قوسين

```
use DB1;

Create Table Employees();
```

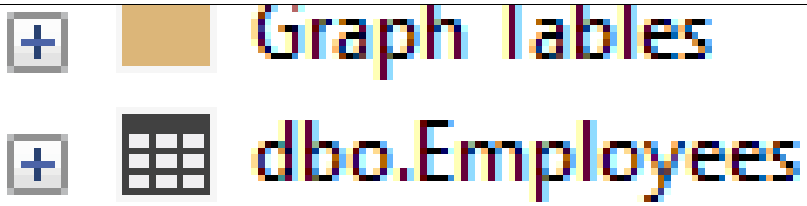
طيب عاوزين نعمل الاعمده :-

الاعمده هتكون جوه القوسين وبين كل عمود والتاني فاصله

هنعرف العمود ازاي قالك هتكتب اسم العمود وبعديه ال data type وبعدين null او not null

ولو فيه عمود عاوز تعرفه علي انه primary key هتكتب في الاخر كلمه primary key وبين قوسين اسم العمود

```
use DB1;

Create Table Employees(
ID int Not Null,
Name nvarchar(50)Not Null,
Phone nvarchar(10)NULL,
Salary smallmoney Null,
PRIMARY KEY(ID)
);
```



SQL CREATE TABLE Statement

A database table is used to store records (data). To create a database table, we use the SQL `CREATE TABLE` statement. Syntax:

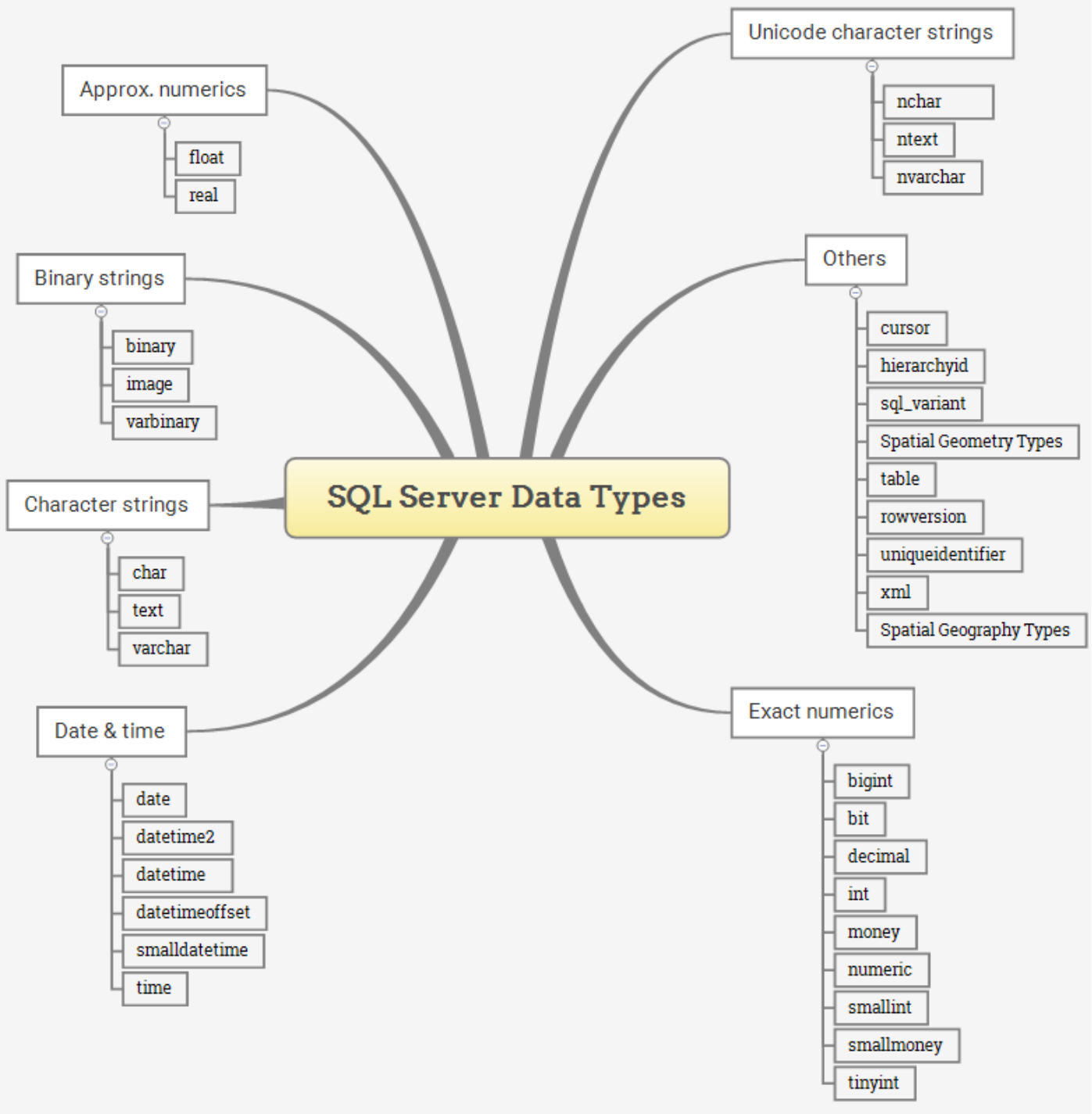
```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

Note: We must provide data types for each column while creating a table. Learn more about SQL Data Types in the next lesson.

SQL DataTypes



بيقولك ماتستخدمش الانوع دي عشان هتتلغي قريب ntext وال text وال image واستخدم بدالهم ال nvarchar وال varchar وال varbinary

ودي الأنواع وبتشيل لحد قد ايه

Exact numeric data types

Exact numeric data types store exact numbers such as integer, decimal, or monetary amount.

- The bit store one of three values 0, 1, and NULL
- The int, bigint, smallint, and tinyint data types store integer data.
- The decimal and numeric data types store numbers that have fixed precision and scale. Note that decimal and numeric are synonyms.
- The money and smallmoney data type store currency values.

The following table illustrates the characteristics of the exact numeric data types:

Data Type	Lower limit	Upper limit	Memory
bigint	−2^63 (−9,223,372, 036,854,775,808)	2^63−1 (−9,223,372, 036,854,775,807)	8 bytes
int	−2^31 (−2,147, 483,648)	2^31−1 (−2,147, 483,647)	4 bytes
smallint	−2^15 (−32,767)	2^15 (−32,768)	2 bytes
tinyint	0	255	1 byte
bit	0	1	1 byte/8bit column
decimal	−10^38+1	10^381−1	5 to 17 bytes
numeric	−10^38+1	10^381−1	5 to 17 bytes
money	−922,337, 203, 685,477.5808	+922,337, 203, 685,477.5807	8 bytes
smallmoney	−214,478.3648	+214,478.3647	4 bytes

Approximate numeric data types

The approximate numeric data type stores floating point numeric data. They are often used in scientific calculations.

Data Type	Lower limit	Upper limit	Memory	Precision
float(n)	−1.79E+308	1.79E+308	Depends on the value of n	7 Digit
real	−3.40E+38	3.40E+38	4 bytes	15 Digit

Character strings data types

Character strings data types allow you to store either fixed-length (char) or variable-length data (varchar). The text data type can store non-Unicode data in the code page of the server.

Data Type	Lower limit	Upper limit	Memory
char	0 chars	8000 chars	n bytes
varchar	0 chars	8000 chars	n bytes + 2 bytes
varchar (max)	0 chars	2^31 chars	n bytes + 2 bytes
text	0 chars	2,147,483,647 chars	n bytes + 4 bytes

Unicode character string data types

Unicode character string data types store either fixed-length (nchar) or variable-length (nvarchar) Unicode character data.

Data Type	Lower limit	Upper limit	Memory
nchar	0 chars	4000 chars	2 times n bytes
nvarchar	0 chars	4000 chars	2 times n bytes + 2 bytes
ntext	0 chars	1,073,741,823 char	2 times the string length

Date & Time data types

The date and time data types store data and time data, and the date time offset.

Data Type	Storage size	Accuracy	Lower Range	Upper Range
datetime	8 bytes	Rounded to increments of .000, .003, .007	1753-01-01	9999-12-31
smalldatetime	4 bytes, fixed	1 minute	1900-01-01	2079-06-06
date	3 bytes, fixed	1 day	0001-01-01	9999-12-31
time	5 bytes	100 nanoseconds	00:00:00.0000000	23:59:59.9999999
datetimeoffset	10 bytes	100 nanoseconds	0001-01-01	9999-12-31
datetime2	6 bytes	100 nanoseconds	0001-01-01	9999-12-31

If you develop a new application, you should use the **time**, **date**, **datetime2** and **datetimeoffset** data types. Because these types align with the SQL Standard and more portable. In addition, the **time**, **datetime2** and **datetimeoffset** have more seconds precision and **datetimeoffset** supports time zone.

Binary string data types

The binary data types stores fixed and variable length binary data.

Data Type	Lower limit	Upper limit	Memory
binary	0 bytes	8000 bytes	n bytes
varbinary	0 bytes	8000 bytes	The actual length of data entered + 2 bytes
image	0 bytes	2,147,483,647 bytes	

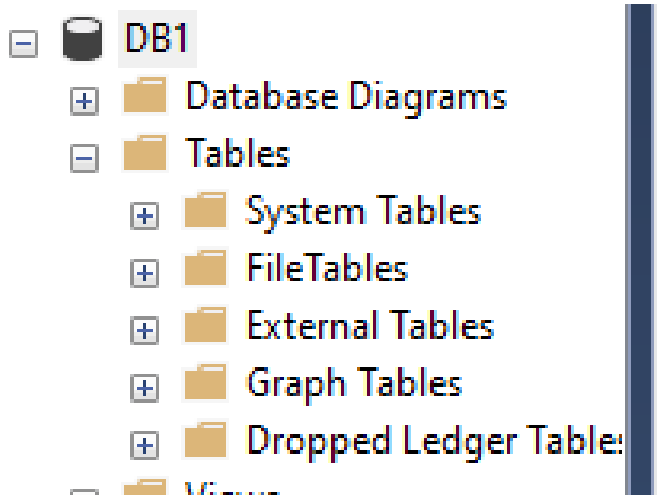
Other data types

Data Type	Description
cursor	for variables or stored procedure OUTPUT parameter that contains a reference to a cursor
rowversion	expose automatically generated, unique binary numbers within a database.
hierarchyid	represent a tree position in a tree hierarchy
uniqueidentifier	16-byte GUID
sql_variant	store values of other data types
XML	store XML data in a column, or a variable of XML type
Spatial Geometry type	represent data in a flat coordinate system.
Spatial Geography type	store ellipsoidal (round-earth) data, such as GPS latitude and longitude coordinates.
table	store a result set temporarily for processing at a later time

Drop Table

بنحذف الجدول بالماوس او بالامر drop table وبعدين اسمه الجدول

```
use DB1;

drop table Employees;
```



SQL DROP TABLE Statement

In SQL, `DROP TABLE` is used to delete the tables in our database. For example,

```
DROP TABLE Employees;
```

Here, the SQL command will delete a table named Employees.

Also make sure you have **admin** or **DROP** permission to run this command.

Note: When we delete a database table, all records within a table are also deleted.

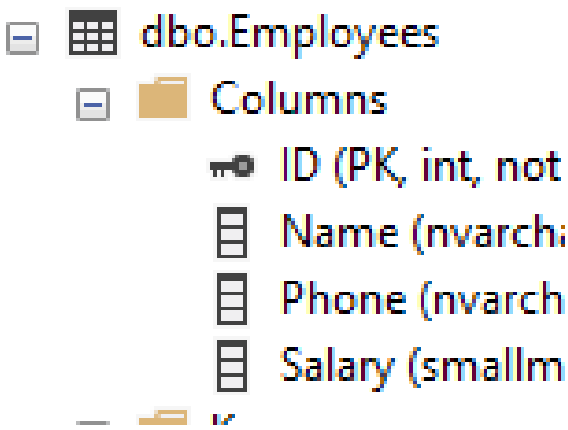
DDL - Alter Table Statement

Add Column

لحد دلوقتي احنا عملنا جدول وداتا بيز وحذفناهم عاوزين بقي نعدل عالجدول

وأول تعديل هوا اننا نضيف عمود عالجدول

دلوقتي احنا عندنا الجدول ده



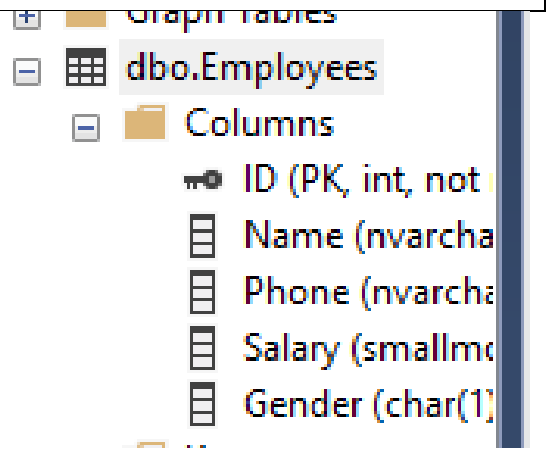
عاوزين نضيف عمود جديد اسمه gender ونوعه char يا male يا female

عشان اعملها بالكود بقوله alter table يعني عد عالجدول وبتكتب بعدها اسم الجدول

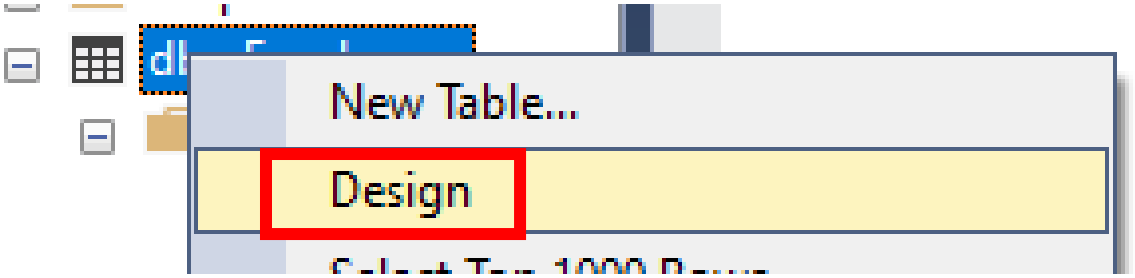
وبعدين بتكتب add وبعده اسم العمود وال data type

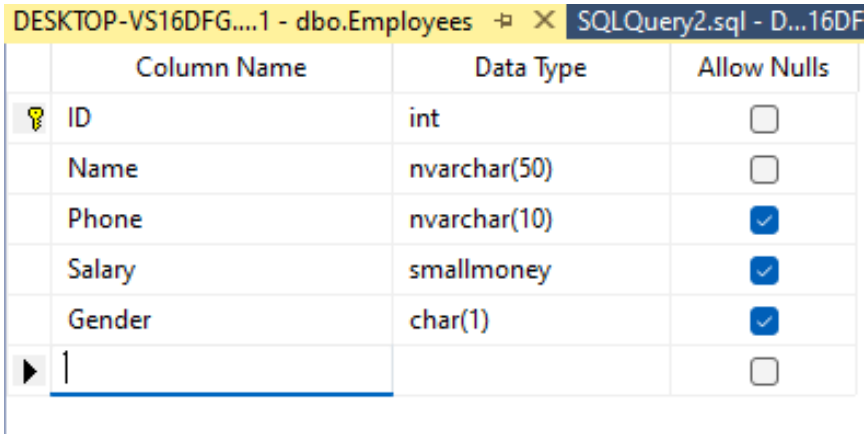
لو ماحددتش null ولا not null هوا من نفسه بيعمله null يعني بيقبل ال null

```
use DB1;

alter table Employess
add Gender char(1);
```



لو عايز تضيف عمود بالماوس





Add Column in a Table

We can add columns in a table using the `ALTER TABLE` command with the `ADD` clause. For example,

```
ALTER TABLE Employees
ADD Gendor char(1);
```

Here, the SQL command adds a column named Gendor in the Employees table.

Rename Column

لو عايز تعمل rename لعمود معين بالماوس بتروح كليك يمين علي الجدول وتختار design او كليك يمين عالعمود وتختار RENAME وتعدله

انما بالكود هتكتب نفس السطر الأول اللي هوا

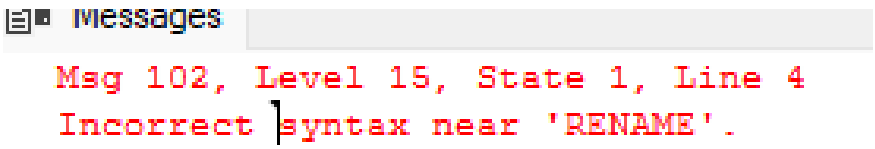
ALTER TABLE Employees

وتحتيه بتكتب rename column وبعدها الاسم القديم وبعدين to وبعدين الاسم الجديد

زي كده

```
ALTER TABLE Employees
RENAME COLUMN Gendor TO Gender;
```

بس هنا هيجيبلك ال ERROR ده



هنا بيقولك انه بيبقي فيه فروقات بسيطه بين كل داتابيز والتانيه وهنا مش هيقبل ال SYNTAX اللي كتبناه

طب ايه البديل؟

قالك انه فيه stored procedures علي مستوي الداتا بيز ودي بتكون function عاديه

ال procedure دي اسمها sp_rename بياخد اسم العمود والاسم الجديد وكلمة column لو في حالتنا هنغير اسم عمود

عشان نستدعي ال procedure ونشغلها بنكتب قبلها كلمه exec

زي كده

```
use DB1;
```

```sql
--ALTER TABLE Employees
--RENAME COLUMN Gendor TO Gender;

exec sp_rename 'Employees.Gendor','Gender','COLUMN';
```

Rename Column in a Table (Most Databases)

We can rename columns in a table using the `ALTER TABLE` command with the `RENAME COLUMN` clause. For example,

```
ALTER TABLE Employees
RENAME COLUMN Gendor TO Gender;
```

Here, the SQL command changes the column name of Gendor to Gender in the Employees table.

Rename column in table (in SQL Server)

You can not use the ALTER TABLE statement in SQL Server to rename a column in a table. However, you can use sp_rename, though Microsoft recommends that you drop and recreate the table so that scripts and stored procedures are not broken.

Syntax

The syntax to rename a column in an existing table in SQL Server (Transact-SQL) is:

```
exec sp_rename 'table_name.old_column_name', 'new_column_name', 'COLUMN';
```

Rename a table

عاوزين نغير اسم الجدول مش العمود

في كل لغات ال sql بتستخدم الكود ده :-

بتكتب alter table وبعدها اسم الجدول وبعدين to وبعدين الاسم الجديد

زي كده

```sql
ALTER TABLE Emp
 RENAME TO Employees;
```

هنا لا بتستخدم الprocedure اللي اسمها sp_rename

بس المرادي بتديها اسم الجدول القيم وبعدين الاسم الجديد يااما غيره بالماوس واخلص

زي كده

```
use DB1;

--ALTER TABLE Emp
-- RENAME TO Employees;

exec sp_rename 'Emp','Employees';
```

Rename a Table (Most Databases)

We can change the name of a table using the `ALTER TABLE` command with the `RENAME` clause. For example,

```
ALTER TABLE OldTableName
RENAME TO NewTableName;
```

Rename table (In SQL Server)

You can not use the ALTER TABLE statement in SQL Server to rename a table. However, you can use sp_rename, though Microsoft recommends that you drop and recreate the table so that scripts and stored procedures are not broken.

Syntax

The syntax to rename a table in SQL Server (Transact-SQL) is:

```
exec sp_rename 'old_table_name', 'new_table_name';
```

Modify a column

عاوزين نعدل علي ال data type بتاعت العمود

هنستخدم برضه ال alter table

وبعدها هنقوله alter column وبعدها اسم العمود وبعدها بتكتب التعديلات اللي عاوز تعملها

زي كده

```
use DB1;

ALTER TABLE Employees
ALTER COLUMN Name nvarchar(100) NOT NULL;
```

لو ماكتبتش NOT NULL هيعملها NULL تلقائي

في بعض أنواع الداتا بيز التانيه بدل كلمه ALTER COLUMN بيكتب MODIFY COLUMN

Modify Column in a Table

We can also change the column's data type using the `ALTER TABLE` command with `MODIFY` or `ALTER COLUMN` clause. For example,

SQL Server

```
ALTER TABLE Employees
ALTER COLUMN Name VARCHAR(100);
```

MySQL

```
ALTER TABLE Employees
MODIFY COLUMN Name VARCHAR(100);
```

Oracle

```
ALTER TABLE Employees
MODIFY Name VARCHAR(100);
```

PostgreSQL

```
ALTER TABLE Employees
ALTER COLUMN Name VARCHAR(100);
```

Here, the SQL command changes the data type of the Name column to VARCHAR in the Employees table.

Delete a column

عشان تحذف عمود بتكتب DROP وبعدها اسم العمود

وطبعا ماتنساش السطر بتاع ALTER TABLE

```
ALTER TABLE Employees
DROP COLUMN Gender;
```

Delete Column in a Table

We can also drop (remove) columns in a table using the `ALTER TABLE` command with the `DROP` clause. For example,

```
ALTER TABLE Employees
DROP COLUMN Gender;
```

Here, the SQL command removes the Gender column from the Employees table.

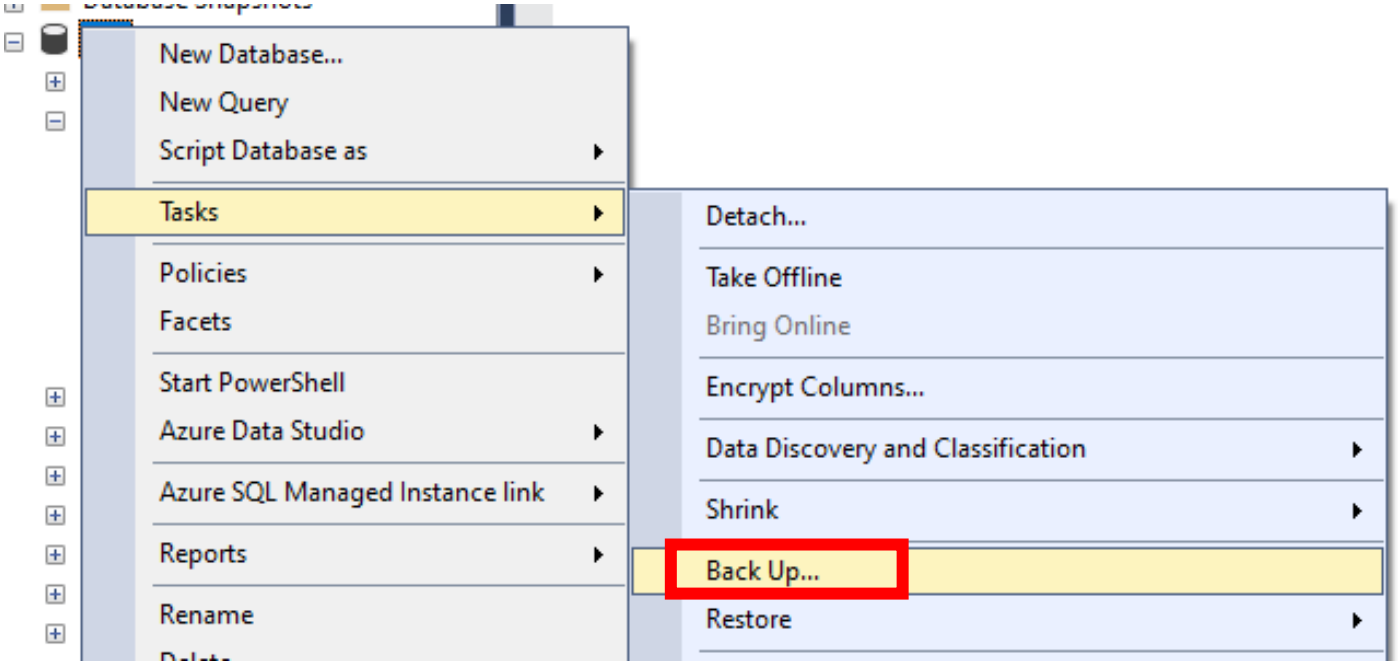
Backup & Restore Database

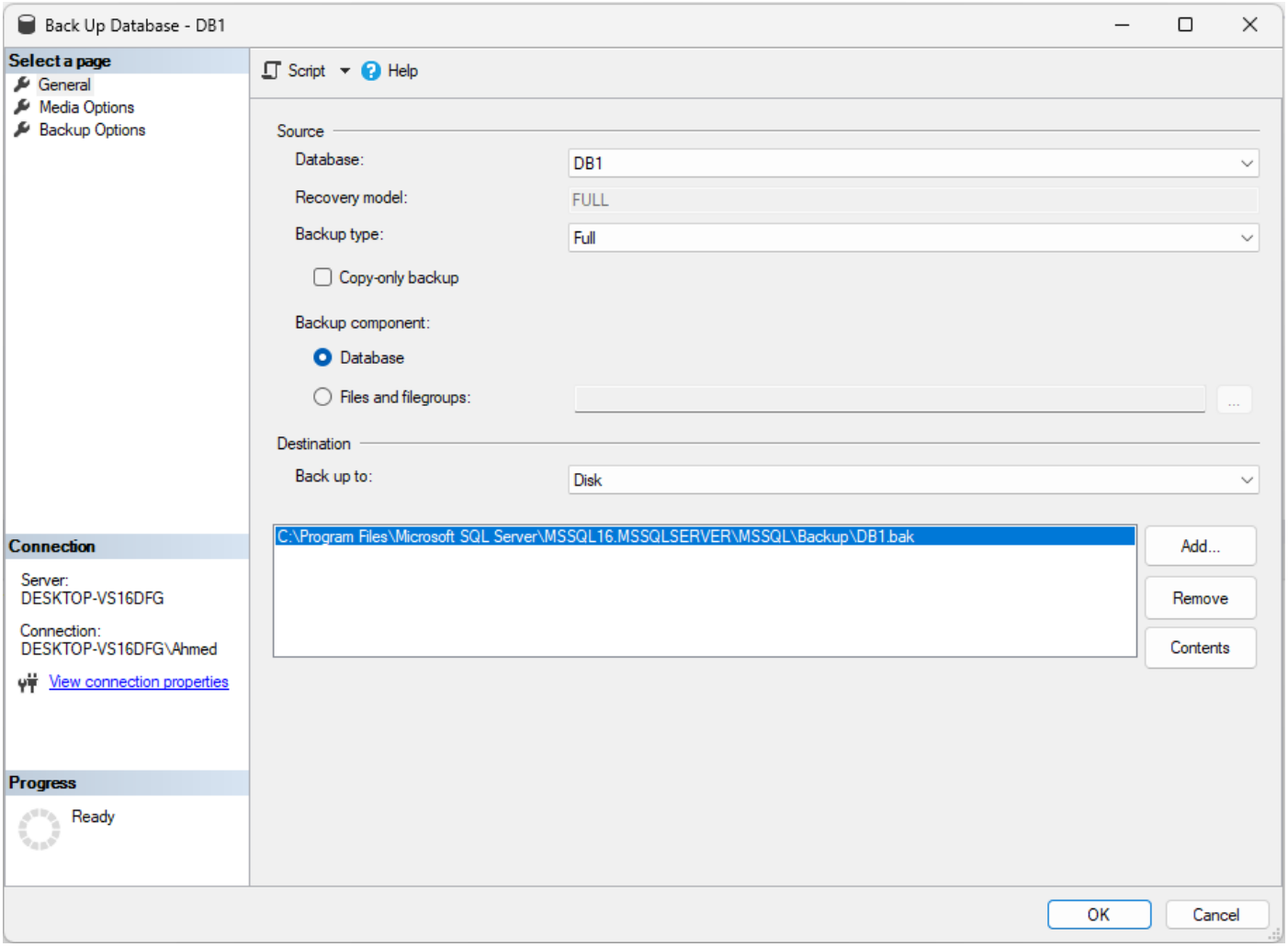
Backup Database

عاوزين ناخد نسخه احتياطيه من الداتابيز

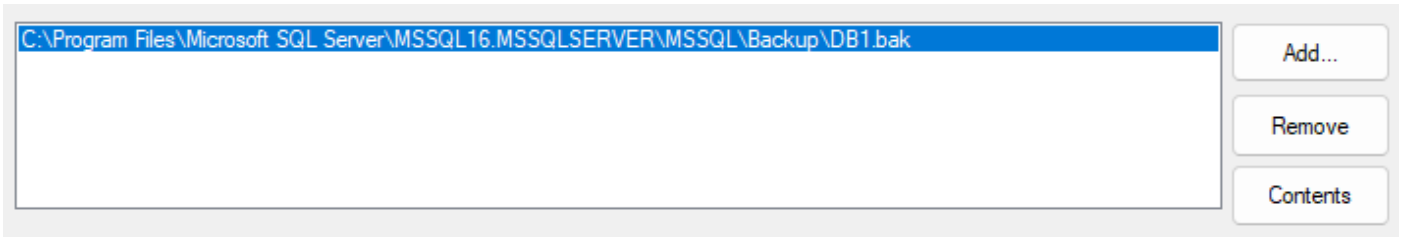
لو عاوزين نعملها بالماوس

كليك يمين عالداتابيز اللي عاوزين نعملها BACKUP



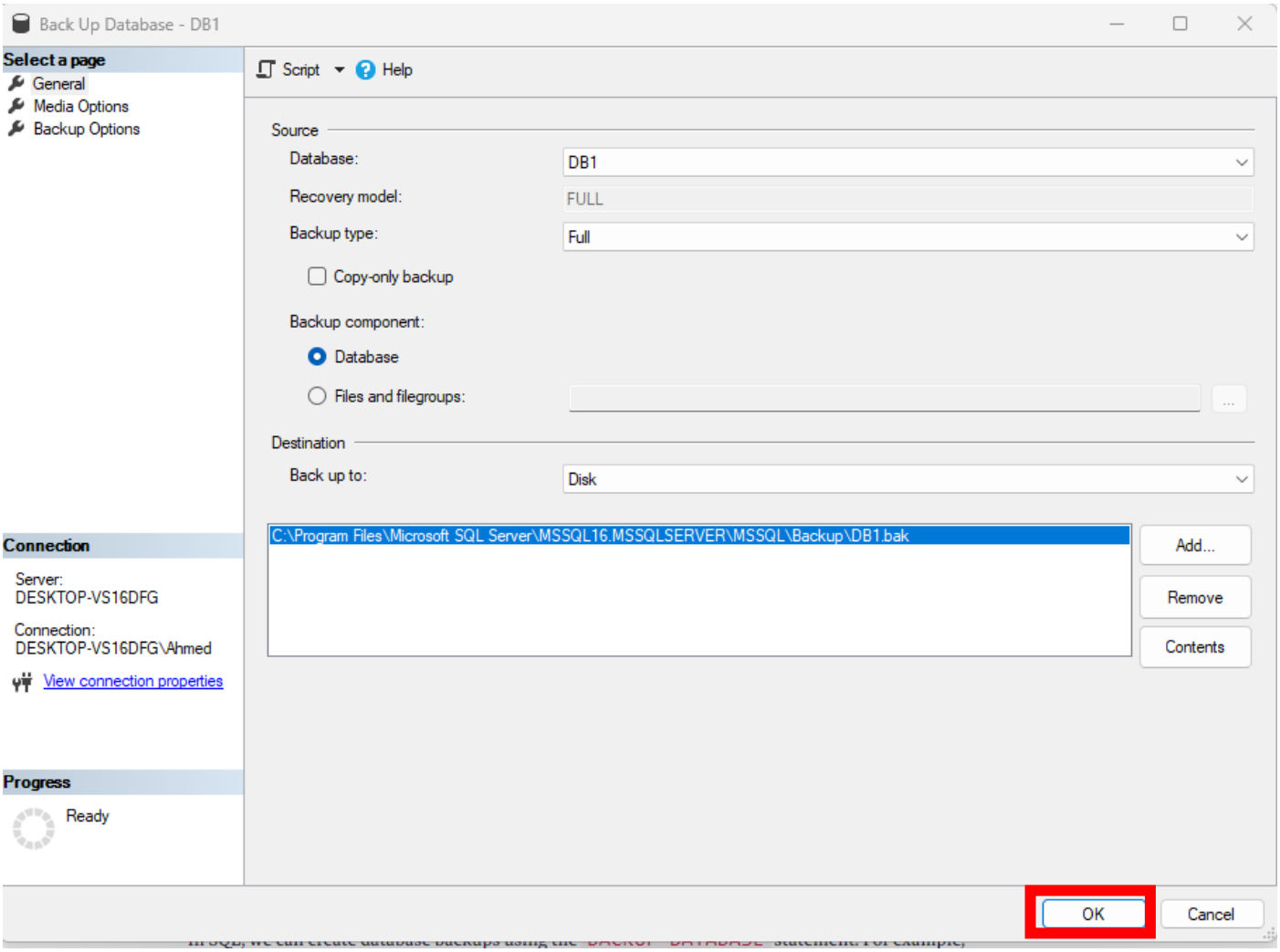


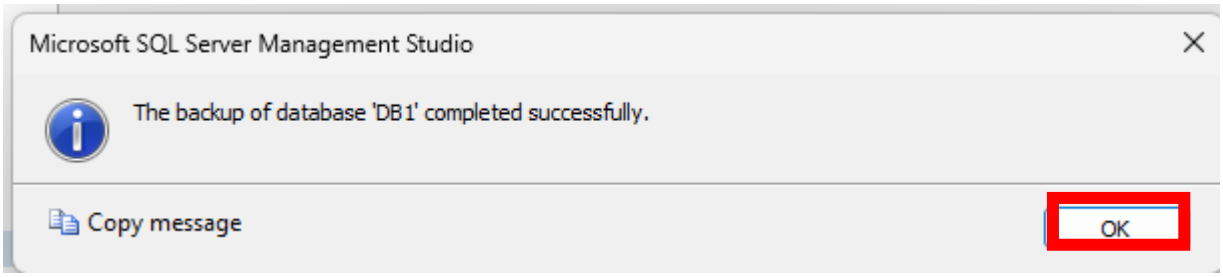
وبعدين من هنا بتشوف عاوز تحط ال BACKUP فين تقدر تشيل ال PATH ده وتعمل ADD وتحدد المكان اللي انت عاوزه



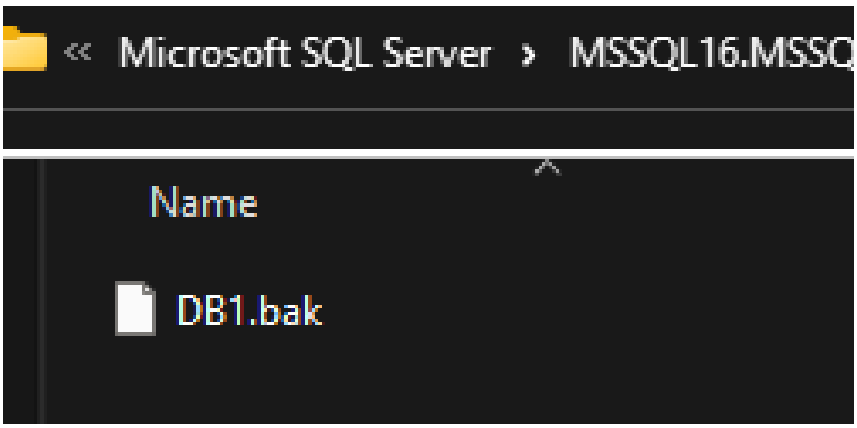
عادة بيحطوا bak. بعد اسم الملف بس لو ماجطيتهوش مش هيأثر

وبعدين دوس ok





ده الملف اهو



وهنا بيقولك لازم ال backup لازم يتعمل في حته غير اللي محطوط فيها الويندوز عشان لو الويندوز او السيستم وقع انت هتقع معاه ختي ماتحطهاش علي نفس الجهاز

طيب عشان نعملها بالكود بنكتب backup database وبعدها اسم الداتا بيز وبعدين نكتب to disk ونكتب المسار

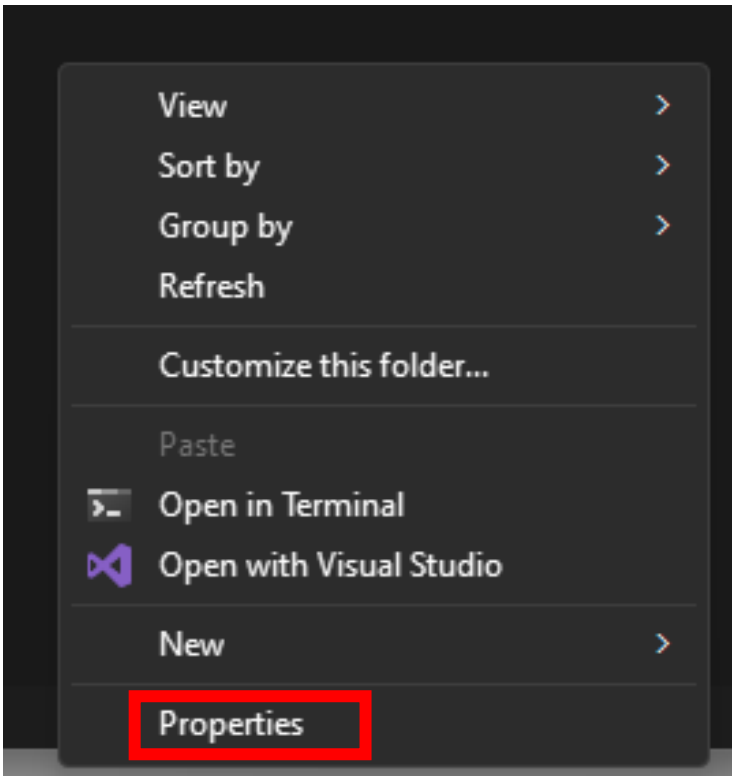
```
use DB1;

BACKUP database DB1
TO DISK ='C:\DB1.bak';
```

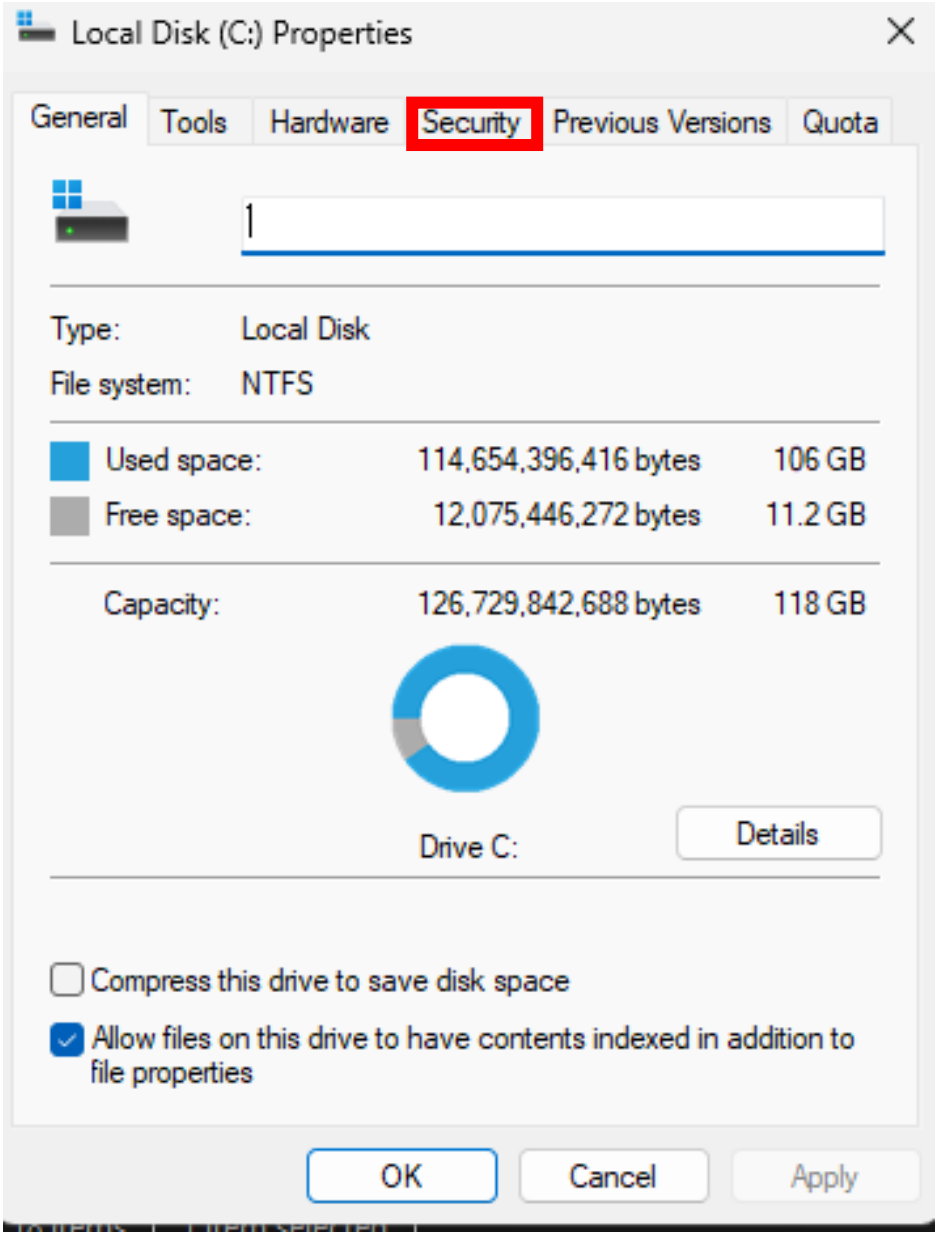
لو اداك ERROR من النوع ACCESS DENIED زي كده

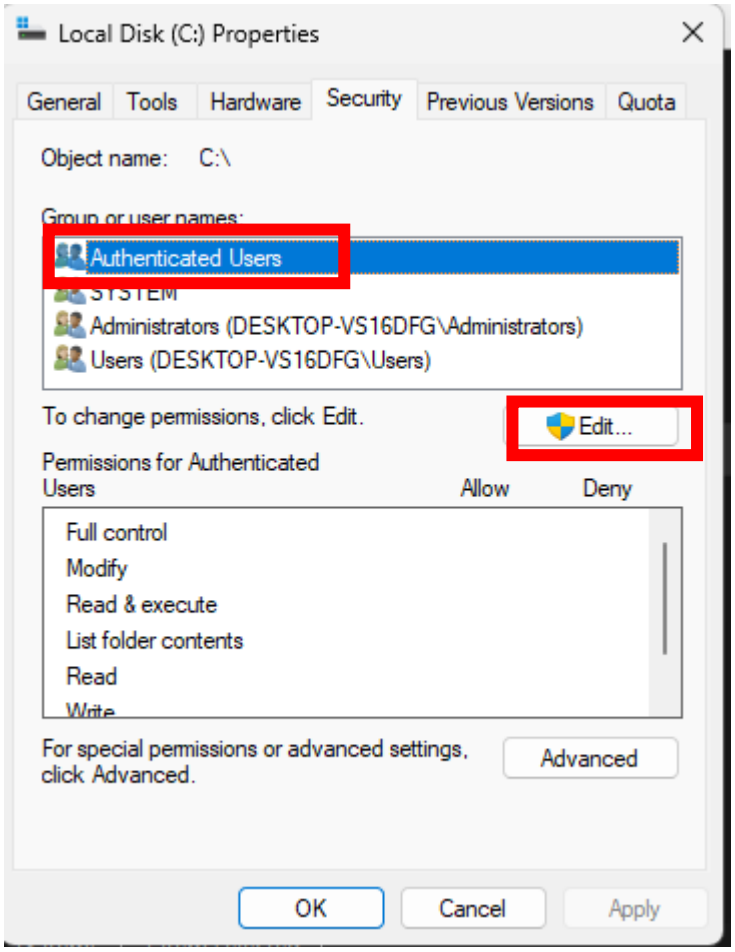
```
Messages
Msg 3201, Level 16, State 1, Line 3
Cannot open backup device 'C:\DB1.bak'. Operating system error 5(Access is denied.).
Msg 3013, Level 16, State 1, Line 3
BACKUP DATABASE is terminating abnormally.
```

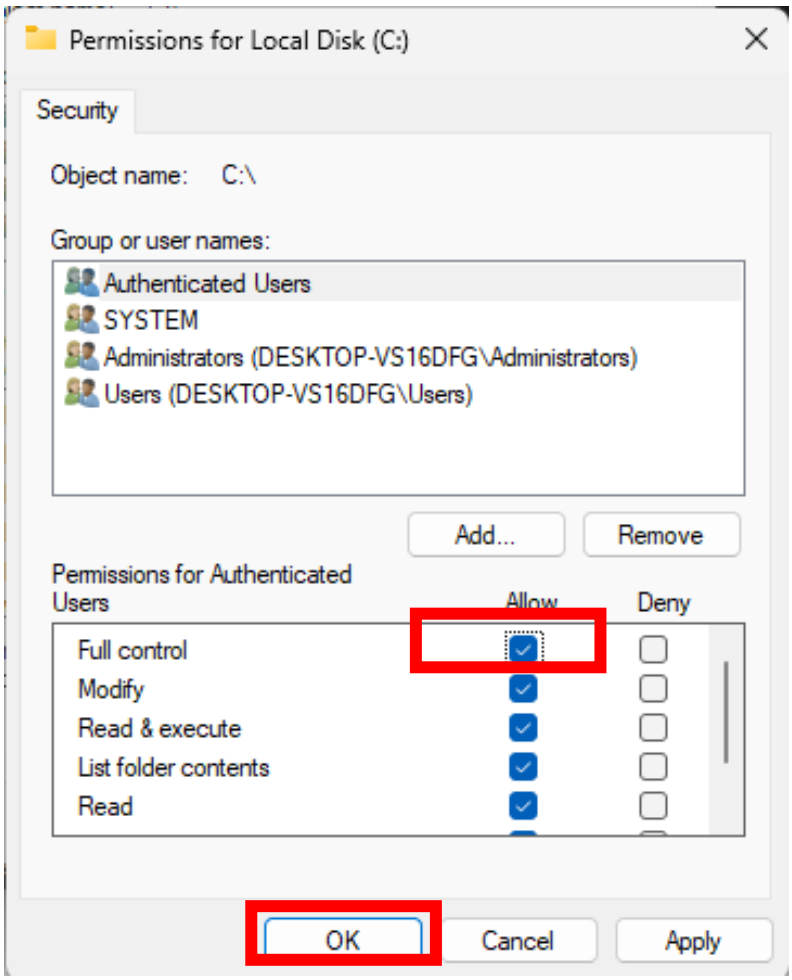
ده معناه انه اليوزر بتاعك بتاع الويندوز ماعندوش صلاحيات يدخل عال C او المكان اللي انت عاوز تحط فيه ال BACKUP

فبتروح لل C DRIVE وكليك يمين




Local Disk (C:) Properties
General
Tools
Hardware
Security
Previous Versions
Quota
1
Type: Local Disk
File system: NTFS
Used space: 114,654,396,416 bytes 106 GB
Free space: 12,075,446,272 bytes 11.2 GB
Capacity: 126,729,842,688 bytes 118 GB
Drive C:
Details
Compress this drive to save disk space
Allow files on this drive to have contents indexed in addition to file properties
OK
Cancel
Apply





واقعد وافق علي كل حاجه

احنا كده بناخد FULL BACKUP لأننا بناخد الداتا بيز كلها هيله بيله وننسخها

SQL BACKUP DATABASE Statement

It is important to create database backups regularly so that our data won't get lost if the database gets corrupted.

In SQL, we can create database backups using the `BACKUP DATABASE` statement. For example,

```
BACKUP DATABASE MyDatabase1
TO DISK = 'C:\MyDatabase1_backup.bak';
```

Here, the SQL command creates a backup file of the MyDatabase1 database inside **C** drive, named MyDatabase1_backup.bak.

Note: It's a common convention to use the **.bak** file extension for database backup files, however, it's not mandatory.

Tip: Always back up the database to a different drive than the actual database. Then, if you get a disk crash, you will not lose your backup file along with the database.

Differential Backup

ال full backup مكلف من ناحية المساحه ومن ناحية الوقت

عشان كده فيه حاجه تانيه اسمها deferential backup ودي فكرتها انك اول مره بتاخد full backup عادي لكن بعد كده بتشوف ايه اللي اتحط زيادهاو اتعدل في البيانات وبيعدلها طريقتها هيا نفس الطريقه اللي فاتت بس بزود كلمتين WITH DIFFERENTIAL

```
use DB1;

BACKUP database DB1
TO DISK ='C:\DB1.bak'
WITH DIFFERENTIAL;
```

SQL DIFFERENTIAL BACKUP DATABASE Statement

A differential back up only backs up the parts of the database that have changed since the last full database backup.

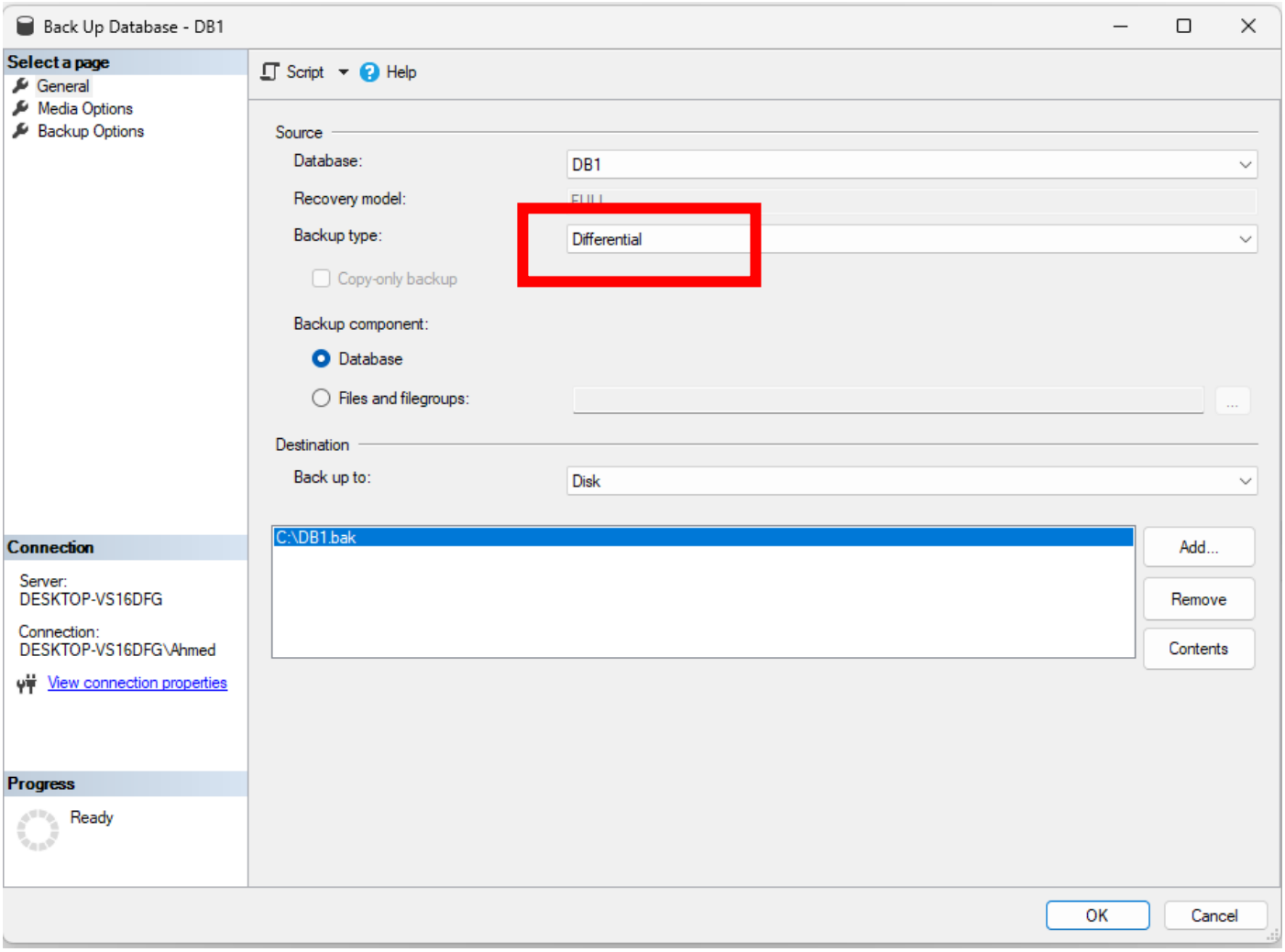
In SQL, we can create differential database backups using the `BACKUP DATABASE With Differential` statement. For example,

```
BACKUP DATABASE MyDatabase1
TO DISK = 'C:\MyDatabase1_backup.bak'
WITH DIFFERENTIAL;
```

Here, the SQL command appends only new changes to the previous backup file. Hence, this command may work faster.

Tip: A differential back up reduces the back up time (since only the changes are backed up).

وبالماوس نفس الطريقه بس بغير دي

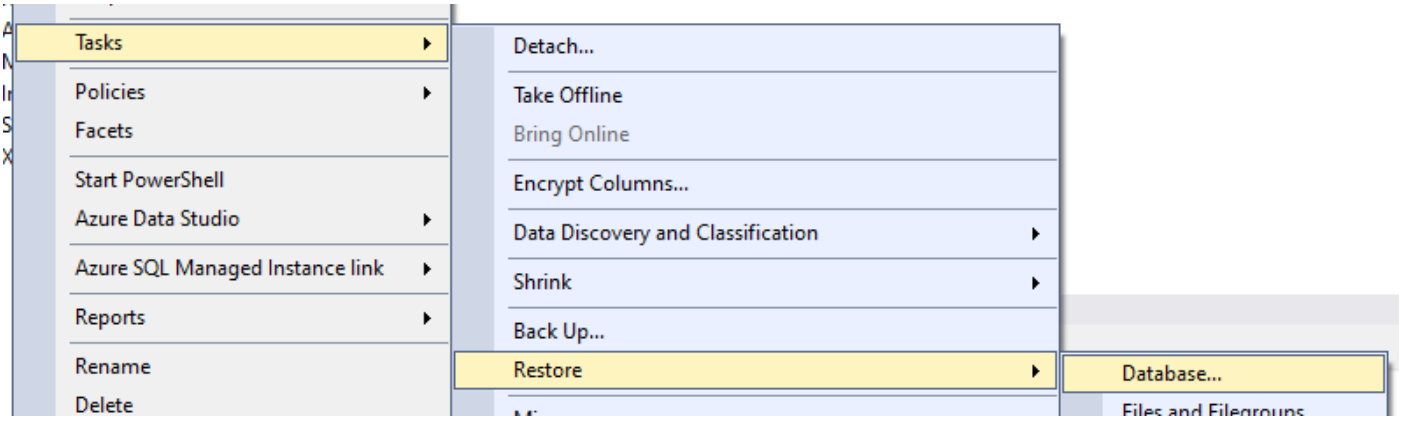


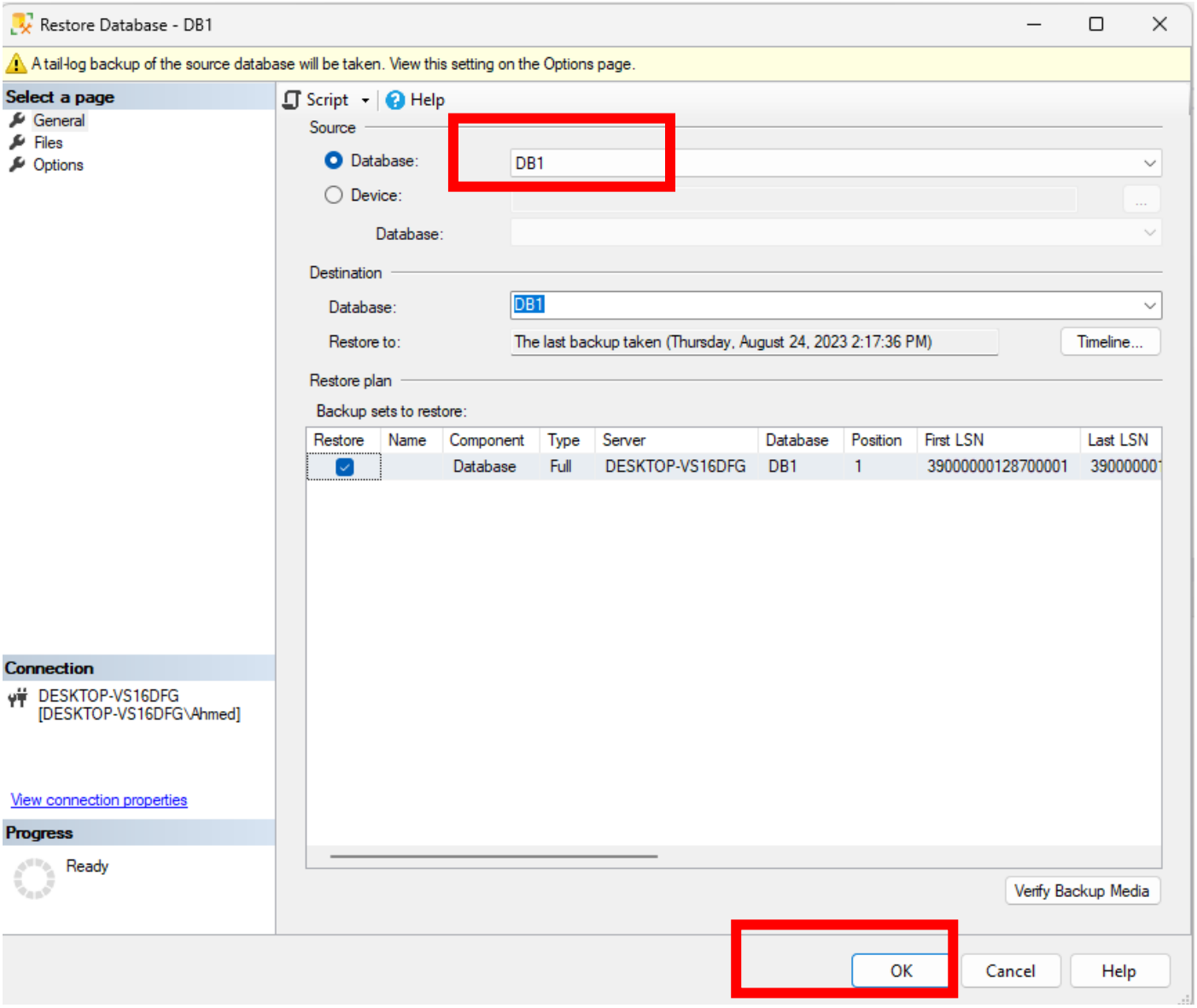
Restore Database

عشان اعمل RESTORE لل backup بكتب نفس الكود بتاع ال full backup بس بغير كلمه backup بكلمة restore وبدل to بكتب from

```
use master;
RESTORE database DB1
FROM DISK ='C:\DB1.bak';
```

وبالماوس من هنا





Restore Database From Backup

To restore a backup file to the database management system, we can use the `RESTORE DATABASE` statement. For example,

```
RESTORE DATABASE MyDatabase1
FROM DISK = 'C:\MyDatabase1.bak';
```

Here, the SQL command restores the MyDatabase1.bak file in the database named MyDatabase1.

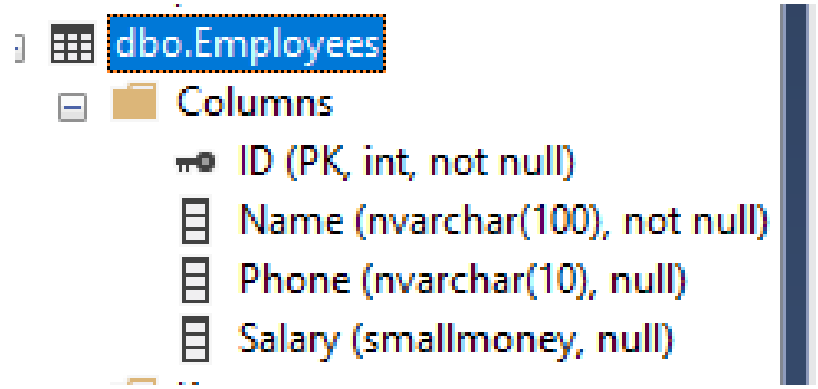
Data Manipulation Language - DML

Insert Into Statement

ال DML ودي اوامرها خاصه بالداتا نفسها او ال records

ال insert statement هيا بتخليك تدخل داتا جديده او records عالجدول

ده الجدول بتاعنا

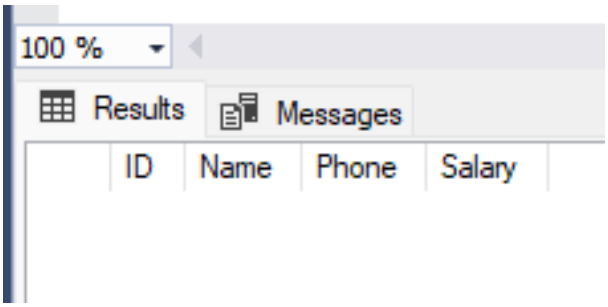


اول حاجه عايز اعملها اني ابص عالجدول اشوف موجود فيه ايه وده مالوش علاقه بالكود اللي عايز اكتبه

عشان اعمل ده هقوله السطر ده واللي معناه اعرضلي كل حاجه من الجدول اللي اسمه employees

```
USE DB1;
SELECT * FROM Employees;
```

وده الجدول اهو فاضي



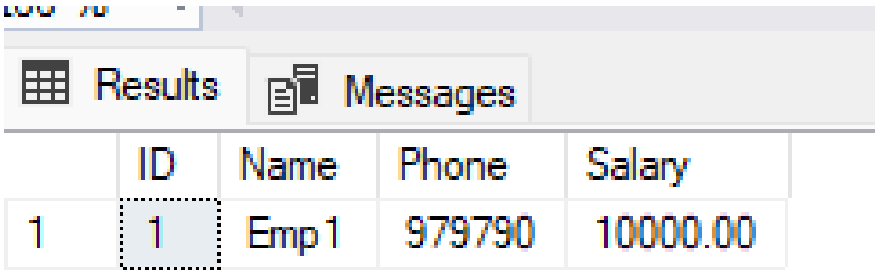
طيب عاوزين بقي ندخل داتا عالجدول

قالك هتكتب insert into وبعدها اسم الجدول اللي هتدخل فيه البيانات وبعدين تكتب values وتفتح قوسين تحط جواهم القيم اللي عاوز تحطها بنفس ترتيب الاعمده اللي في الجدول

زي كده

```
INSERT INTO Employees
VALUES(1,'Emp1','979790',10000);
```

وخليك فاكر انك ممكن تحدد جزئية معينه بالماوس وينفذهالك هيا بس



لو جيت تدخل بيانات جديده وجيت تحط نفس ال id هيجيبلك خطأ انه ال id ده موجود قبل كده

تقدر تدخل اكتر من record بجمله insert واحده عن طريق تكرار الاقواس زي كده

```
INSERT INTO Employees
values
(2,'Emp2','1234',700),
(3,'Emp3','5464',400),
(4,'Emp4','7897',900);
```

هنا لو عايز ادخل عمودين بس مش عايز ادخل الباقي

اول حاجه لازم الاعمدة اللي مش عايز تدخلها دي تكون بتقبل ال null

طيب عشان ادخل العمودين هعمل ايه ؟

قالك هتيجي جنب ال insert into وتكتب اسم الجدول وتفتح قوسين تكتب فيهم اسامي الاعمده اللي عاوز تحط فيهم الداتا وبعدين تشتغل عادي بس بالترتيب اللي انت حاطه يعني لو كاتب انك عايز تدخل في ال name وبعدين ال id يبقي تكتب الداتا بتاعت ال name الأول

زي كده

```
INSERT INTO Employees(ID,Name)
VALUES(5,'Emp5');
```

100 %

Results | Messages

	ID	Name	Phone	Salary
1	1	Emp1	979790	10000.00
2	2	Emp2	1234	700.00
3	3	Emp3	5464	400.00
4	4	Emp4	7897	900.00
5	5	Emp5	NULL	NULL

اقدر اعمل كده كمان

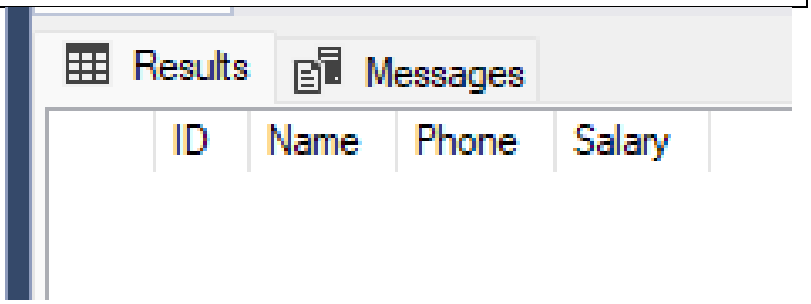
```
INSERT INTO Employees
VALUES(6,'Emp6',null,null);
```

Results | Messages

	ID	Name	Phone	Salary
1	1	Emp1	979790	10000.00
2	2	Emp2	1234	700.00
3	3	Emp3	5464	400.00
4	4	Emp4	7897	900.00
5	5	Emp5	NULL	NULL
6	6	Emp6	NULL	NULL

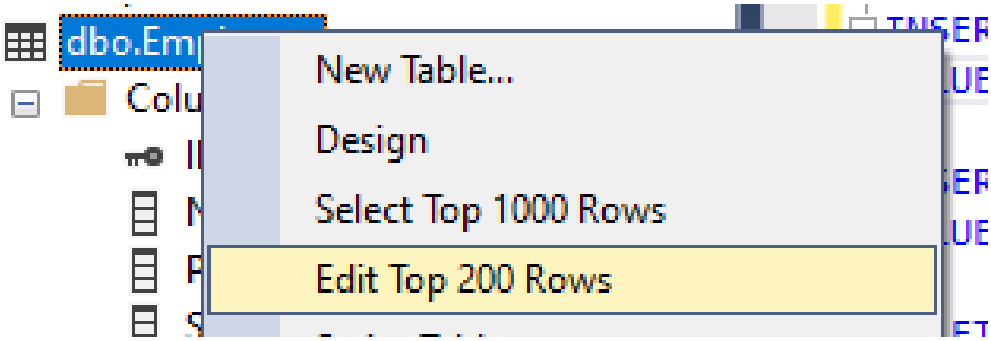
لو عايز احذف الداتا كلها بكتب delete from وبعدها اسم الجدول

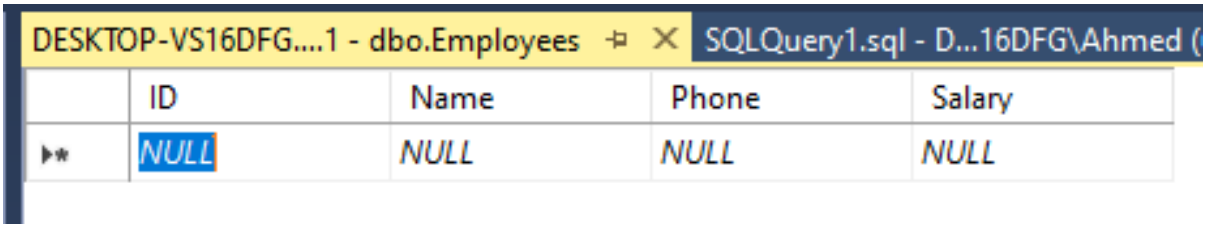
```
DELETE FROM Employees;
```



لو حددت امر معين حتي لو كان معموله COMMENT تقدر تعمله excute عادي

لو عايز تدخل بيانات بالماوس كليك يمين عالجدول وبتختار edit





وبعدين اكتب اللي انت عاوزه

ممكن تعمل ده لو عاوز تجرب انما اعقتد في العادي انت هتحط جمله ال insert نفسها في البرنامج بتاعك

INSERT INTO Statement

The `INSERT INTO` statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the `INSERT INTO` statement in two ways:

1. Specify both the column names and the values to be inserted:

INSERT INTO *table_name* (*column1, column2, column3, ...*)

VALUES (*value1, value2, value3, ...*);

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the `INSERT INTO` syntax would be as follows:

INSERT INTO *table_name*

VALUES (*value1, value2, value3, ...*);

وده الكود كله

```
--this will show all data in the table
select * from Employees;


--Insert one record at a time
Insert Into Employees
values
(10,'Emp10','079939',1000);

--Insert one record at a time with some null values
Insert Into Employees
values
(11,'Emp11',null,null);


--insert multiple records at a time.
Insert Into Employees
values
(2,'Emp2','552221',700),
(3,'Emp3','55554',300),
(4,'Emp4','322344',400);


--insert only selected fields
Insert Into Employees (ID, Name)
values
(5,'Emp5');


--if you forget to insert not null filed an error will occure.
Insert Into Employees (ID)
values (5);
```

```
select * from Employees;

--this will delete all records in table.
--delete from Employees;
```

Update Statement

هنا عاوز اقولك انه التعديل عالبيانات اللي في الداتا بيز مافيهاش undo يعني الداتا اللي هتعدلها مش هتعرف ترجعها تاني الا لو كنت عامل backup

ليه بقولك كده ؟

بقولك كده عشان أوامر الحذف والتعديل لو ماحطيتش فيها جملة شرطيه بتحذف كل الداتا اللي في الجدول

او بتعدل علي كل الداتا في العمود اللي في الجدول

وقبل ماتعمل جمله حذف او تعديل لازم تتاكد من الكود بتاعك قبل ماتشغله

طيب عشان نغير قيمة خليه او قيمه معينه في record معين بنكتب كلمة update وبعدها اسم الجدول اللي عايزين نعدل عليه وبعدين بنكتب كلمة set واسم العمود = القيمه الجديده ولو فيه حاجه تانيه في نفس ال record عايزين نعدلها بنعمل فاصله ونكتب اسم العمود = القيمه الجديده برضه

لحد هنا لو جينا شغلنا الامر ده هيغير كل القيم اللي في العمود بنفس القيمه اللي كتبناها

وعشان ده مايحصلش بنحط شرط وده بيتم بكلمة where

بص المثال ده

طيب دلوقتي احنا عندنا الجدول ده

Results Messages

	ID	Name	Phone	Salary
1	2	Emp2	552221	700.00
2	3	Emp3	55554	300.00
3	4	Emp4	322344	400.00
4	5	Emp5	NULL	NULL
5	10	Emp10	079939	1000.00
6	11	Emp11	NULL	NULL

عاوزين نغير اسم الموظف اللي ال id بتاعه = 2 نخليه Mohamed abu hadhoud

```
UPDATE Employees
SET Name='Mohamed Abu-Hadhoud'
where ID=2;
```

وده لو عاوزين نغير الاسم والمرتب

```
UPDATE Employees
SET Name='Mohamed Abu-Hadhoud',Salary=5000
where ID=2;
```

Results Messages

	ID	Name	Phone	Salary
1	2	Mohamed Abu-Hadhoud	552221	5000.00
2	3	Emp3	55554	300.00
3	4	Emp4	322344	400.00
4	5	Emp5	NULL	NULL
5	10	Emp10	079939	1000.00
6	11	Emp11	NULL	NULL

هنا ممكن نستخدم الشرط في صالحنا زي اننا مثلا نعدل علي كل الرواتب اللي اقل من 500 نزودها ب 200

```
UPDATE Employees
SET Salary=Salary+200
WHERE Salary<500;
```

	ID	Name	Phone	Salary
1	2	Mohamed Abu-Hadhoud	552221	5000.00
2	3	Emp3	55554	500.00
3	4	Emp4	322344	600.00
4	5	Emp5	NULL	NULL
5	10	Emp10	079939	1000.00
6	11	Emp11	NULL	NULL

طيب عاوزين المرتبات اللي اقل من 1000 نزودها ب 10%

```
UPDATE Employees
SET Salary=Salary*1.1
WHERE Salary<=1000;
```

Results Messages

	ID	Name	Phone	Salary
1	2	Mohamed Abu-Hadhoud	552221	5000.00
2	3	Emp3	55554	550.00
3	4	Emp4	322344	660.00
4	5	Emp5	NULL	NULL
5	10	Emp10	079939	1100.00
6	11	Emp11	NULL	NULL

UPDATE Statement

The `UPDATE` statement is used to modify the existing records in a table.

UPDATE Syntax

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Note: Be careful when updating records in a table! Notice the `WHERE` clause in the `UPDATE` statement. The `WHERE` clause specifies which record(s) that should be updated. If you omit the `WHERE` clause, all records in the table will be updated!

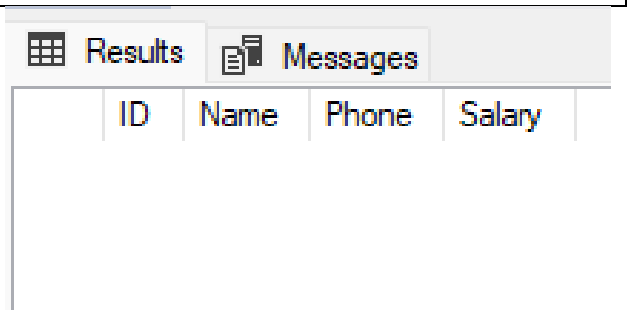
```
--this will show all data in the table
select * from Employees;

-- this will update one filed at a time
Update Employees
set Name ='Mohammed Abu-Hadhoud'
where ID=2;

-- this will update multiple fields at a time.
Update Employees
set Name ='Mohammed Abu-Hadhoud' ,  Salary=5000
where ID=2;

-- this will increase the salary by 200 for all employees that their salaries are less than 500
update Employees
set Salary = Salary+ 200
where  Salary < 500 ;

-- this will increase the salary by 10% for all employees that their salaries are less than or equal
1000
update Employees
set Salary = Salary *1.1
where  Salary <= 1000;
```

Delete Statement

لو عايز احذف الداتا كلها في الجدول كتب delete from وبعدها اسم الجدول

ولو عايز احذف record معين بحط شرط باستخدام ال where

```
DELETE FROM Employees;
```



عبينا الجدول تاني

```
INSERT INTO Employees
VALUES
(1,'Emp1','079939',1000),
(2,'Emp2','552221',700),
(3,'Emp3','55554',300),
(4,'Emp4','322344',400),
(5,'Emp5',null,null),
(11,'Emp11',null,null);
```

Results Messages

	ID	Name	Phone	Salary
1	1	Emp1	079939	1000.00
2	2	Emp2	552221	700.00
3	3	Emp3	55554	300.00
4	4	Emp4	322344	400.00
5	5	Emp5	NULL	NULL
6	11	Emp11	NULL	NULL

دلوقتي عاوزين نحذف أي record ال salary بتاعه ب null

كل اللي هنعمله هوا اننا هنزود شرط علي امر الحذف

هنا مفيش حاجه اسمه =null بتقول بدالها is null

```
DELETE FROM Employees
WHERE Salary is null;
```

Results Messages

	ID	Name	Phone	Salary
1	1	Emp1	079939	1000.00
2	2	Emp2	552221	700.00
3	3	Emp3	55554	300.00
4	4	Emp4	322344	400.00

عاوزين نحذف اخر واحد اللي ال id بتاعه ب4

```
DELETE FROM Employees
WHERE ID=4;
```

Results Messages

	ID	Name	Phone	Salary
1	1	Emp1	079939	1000.00
2	2	Emp2	552221	700.00
3	3	Emp3	55554	300.00

لو كتبت شرط مش متحقق مش هيعمل حاجه

DELETE Statement

The `DELETE` statement is used to delete existing records in a table.

DELETE Syntax

```
DELETE FROM table_name WHERE condition;
```

Note: Be careful when deleting records in a table! Notice the `WHERE` clause in the `DELETE` statement. The `WHERE` clause specifies which record(s) should be deleted. If you omit the `WHERE` clause, all records in the table will be deleted!

```
--this will show all data in the table
select * from Employees;

-- this will delete all employees which their salary is null
delete from Employees
where salary is null;

-- this will delete all employees that have their id=4 , which is one record in our case
delete from Employees
where ID=4;
```

Select Into Statement

رجعنا الداتا اللي حذفناها

```
INSERT INTO Employees
VALUES
(1,'Emp1','079939',1000),
(2,'Emp2','552221',700),
(3,'Emp3','55554',300),
(4,'Emp4','322344',400),
(5,'Emp5',null,null),
(11,'Emp11',null,null);
```

Results Messages

	ID	Name	Phone	Salary
1	1	Emp1	079939	1000.00
2	2	Emp2	552221	700.00
3	3	Emp3	55554	300.00
4	4	Emp4	322344	400.00
5	5	Emp5	NULL	NULL
6	11	Emp11	NULL	NULL

دلوقتي انا عايز انسخ الداتا اللي في الجدول ده لجدول تاني

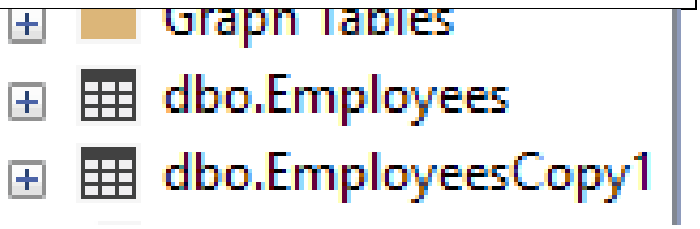
فيه اكتر من طريقه عشان تعمل ده

بس فيه طريقه سهله عشان تعمل كده

بنكتب SELECT * INTO وكده هينسخ كل حاجه وبعدين بنعين اسم للجدول الجديد اللي هننسخ فيه الداتا والامر ده هوا اللي هينشئ الجدول وبعدين تكتب FROM وبعدها اسم الجدول اللي هتنسخ منه الداتا

لو عايز تنسخ داتا معينه بتزود الشرط ولو عايز تنسخ اعمده محدده بتكتب اسم الاعمده بدل ال *

```
SELECT * INTO EmployeesCopy1
FROM Employees;
```



Results Messages

	ID	Name	Phone	Salary
1	1	Emp1	079939	1000.00
2	2	Emp2	552221	700.00
3	3	Emp3	55554	300.00
4	4	Emp4	322344	400.00
5	5	Emp5	NULL	NULL
6	11	Emp11	NULL	NULL

لو جيت انت عملت الجدول وكتبت نفس الامر هيطلعلك ERROR

وهنا هننسخ عمودين بس

```
SELECT ID,Name INTO EmployeesCopy2
From Employees;

SELECT * FROM EmployeesCopy2;
```

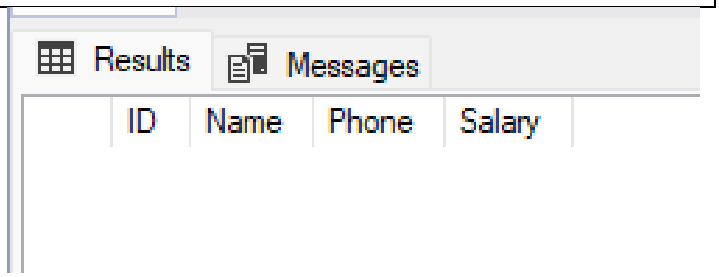
Results Messages

	ID	Name
1	1	Emp1
2	2	Emp2
3	3	Emp3
4	4	Emp4
5	5	Emp5
6	11	Emp11

طيب لو عايز انسخ اسامي الاعمده بس

هحط شرط النتيجه بتاعته ب false

```sql
SELECT * INTO EmployeesCopy3
FROM Employees
WHERE 5=6;

SELECT * FROM EmployeesCopy3;
```



SELECT INTO Statement

In SQL, we can copy data from one database table to a new table using the `SELECT INTO` command. For example,

```sql
SELECT *
INTO EmpoyeesCopy
FROM Employees;
```

Here, the SQL command copies all data from the Employees table to the new EmployeesCopy table.

Note: The `SELECT INTO` statement creates a new table. If the database already has a table with the same name, `SELECT INTO` gives an error.

If we want to copy data to an existing table (rather than creating a new table), we should use the INSERT INTO SELECT statement.

```sql
--this will show all data in the table
select * from Employees;

-- this will create a new table named EmployeesCopy1 based on the selected coumns then it will copy
the data from Employees table based on the condition provided
SELECT *
INTO EmpoyeesCopy1
FROM Employees;

select * from EmpoyeesCopy1;



-- this will create a new table named EmployeesCopy1 based on the selected coumns then it will copy
the data from Employees table based on the condition provided
SELECT ID, Name
INTO EmpoyeesCopy2
FROM Employees;

select * from EmpoyeesCopy2;
```

```
-- this will create a new table named EmployeesCopy1 based on the selected coumns then it will
--  copy the data from Employees table based on the condition provided which is false means no data
will be copied
SELECT *
INTO EmpoyeesCopy3
FROM Employees
where 5=6;

select * from EmpoyeesCopy3;
```

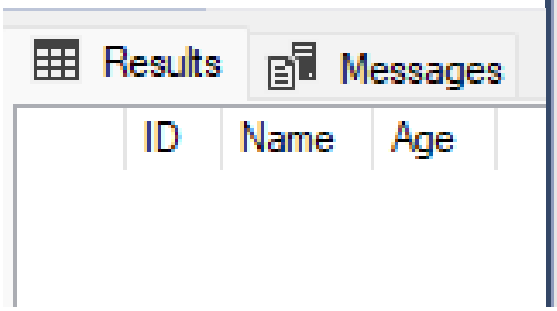
Insert Into ..Select From Statement

حذفنا الجداول القديمه

```
use DB1;
DROP TABLE Employees;
DROP TABLE EmployeesCopy1;
DROP TABLE EmployeesCopy2;
DROP TABLE EmployeesCopy3;
```

هنعمل جدولين تانيين

```
use DB1;

CREATE TABLE Persons(
ID int not null,
Name nvarchar(50)not null,
Age tinyint not null,
PRIMARY KEY(id)
);

SELECT * INTO OldPersons
FROM Persons;

ALTER TABLE OldPersons
ADD PRIMARY KEY(ID);

Insert INTO Persons
VALUES
(1,'Mohamed',45),
(2,'Ali',30),
(3,'Amjad',25),
(4,'Maha',20),
(5,'Shadi',22);

SELECT*FROM Persons;
SELECT*FROM OldPersons;
```



	ID	Name	Age
1	1	Mohamed	45
2	2	Ali	30
3	3	Amjad	25
4	4	Maha	20
5	5	Shadi	22

دلوقتي انت لما تيجي تدخل داتا جديده عالجدول بتعمل ايه ؟

بكتب امر insert into

طيب للو عايز تختار كل الداتا من الجدول بتعمل ايه ؟

بكتب امر select from

طيب احنا دلوقتي عاوزين ننسخ الداتا اللي موجوده في ال persons ونحطها في ال old persons

لو ماكانش جدول ال old persons موجود كنا استخدمنا ال select into عادي

لكن الجدول موجود عندنا هنعمل ايه ؟

هستخدم جمله ال insert مع جملة ال select يعني هعمل select للداتا الموجوده في الجدول بتاع ال persons وهعمل بيها insert في الجدول بتاع ال old persons طيب ازاي ؟

بكل بساطه تكتبهم تحت بعض

```
INSERT INTO OldPersons
SELECT * FROM Persons;
```

Results Messages

	ID	Name	Age
1	1	Mohamed	45
2	2	Ali	30
3	3	Amjad	25
4	4	Maha	20
5	5	Shadi	22

هوا كده هياخد الداتا اللي ناتجه عن ال SELECT STATEMENT وينسخها في الجدول

وبرضه تقدر تستخدم الشروط

طيب هنحذف الداتا اللي موجوده في ال old persons

ونرجع ننسخ الأشخاص اللي سنهم اكبر من 30

```
DELETE FROM OldPersons;
INSERT INTO OldPersons
SELECT * FROM Persons
WHERE AGE>=30;
```

Results | Messages

	ID	Name	Age
1	1	Mohamed	45
2	2	Ali	30

INSERT INTO SELECT Statement

We'll learn to copy records from one table to another with the help of examples.

The `INSERT INTO SELECT` statement is used to copy records from one table to another existing table. For example,

```
INSERT INTO OldPersons
SELECT *
FROM Persons;
```

Here, the SQL command copies all records from the Persons table to the OldPersons table.

Note: To run this command,

- the database must already have a table named OldPersons
- the column names of the OldPersons table and the Persons table must match

If we want to copy data to a new table (rather than copying in an existing table), we should use the SELECT INTO statement.

Copy all columns from one table to another table:

INSERT INTO *table2*

SELECT * FROM *table1*

WHERE *condition*;

Copy only some columns from one table into another table:

INSERT INTO *table2* (*column1, column2, column3, ...*)

SELECT *column1, column2, column3, ...*

FROM *table1*

WHERE *condition*;

MYSQL

```
insert into OldPersons
select  * from Persons
where age >=30;
```

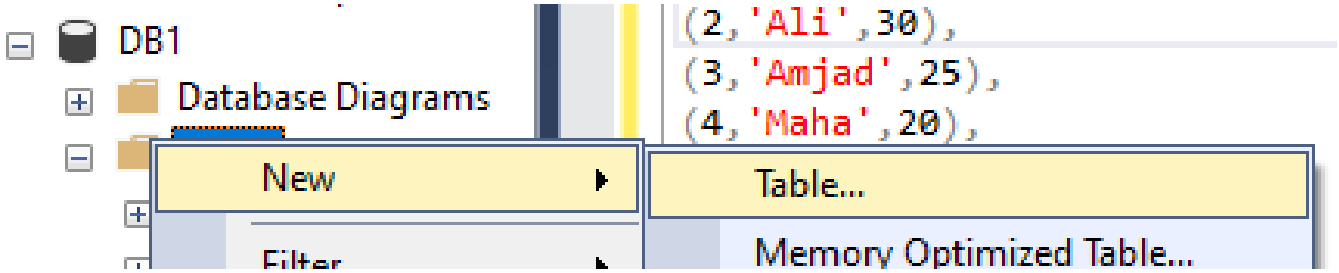
Misc 1

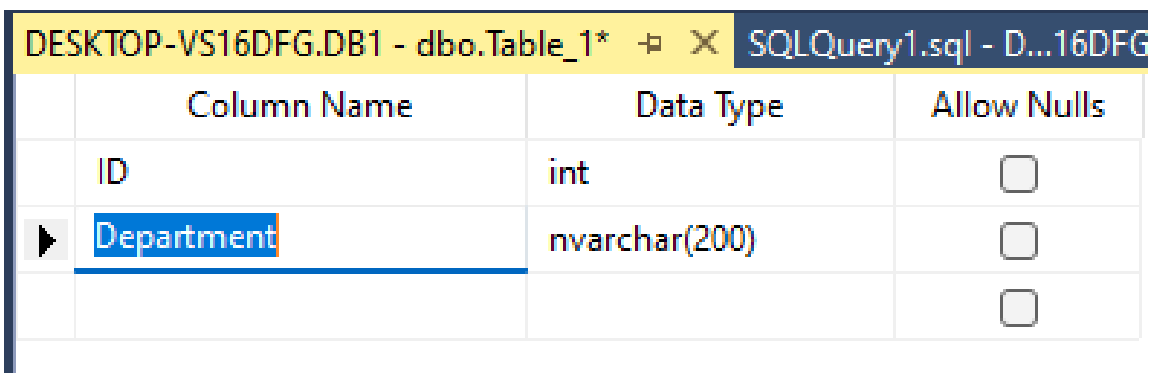
Identity Field (Auto Increment)

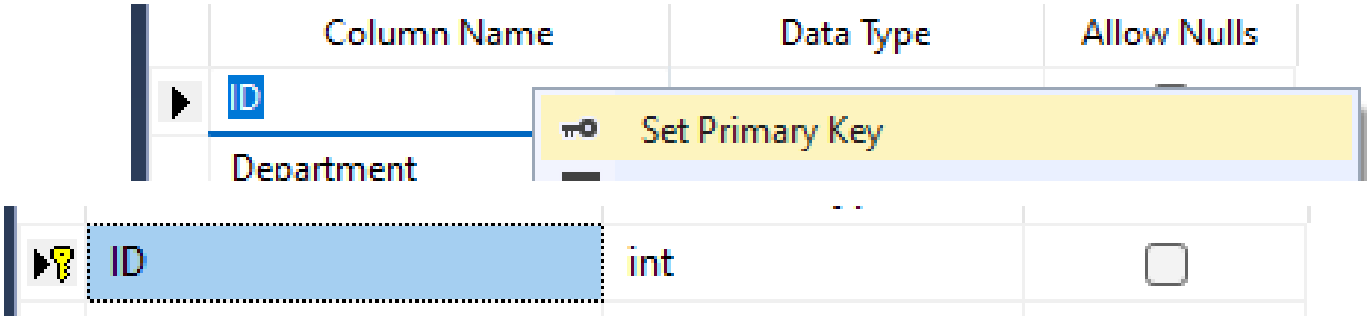
دلوقتي احنا عاوزين عمود ال ID يدخل الرقم لوحده من غير ماكتبه بايدي الحركه دي اسمها auto numbering او auto increment

الفكره منها انه هيرقم ال id بدالك احسن ماتتلخبط وانت بترقم

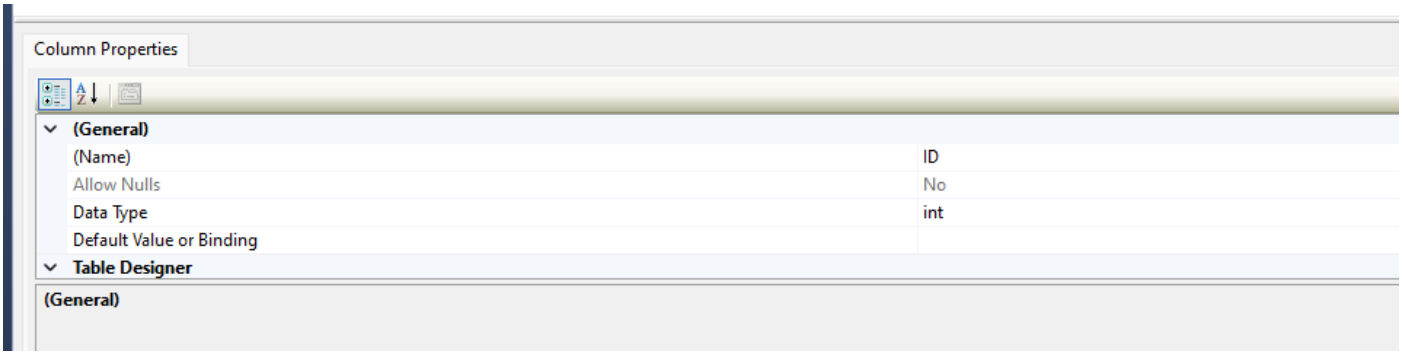
تعالي نعمل جدول جديد بالماوس ونعمل ال auto increment فيه

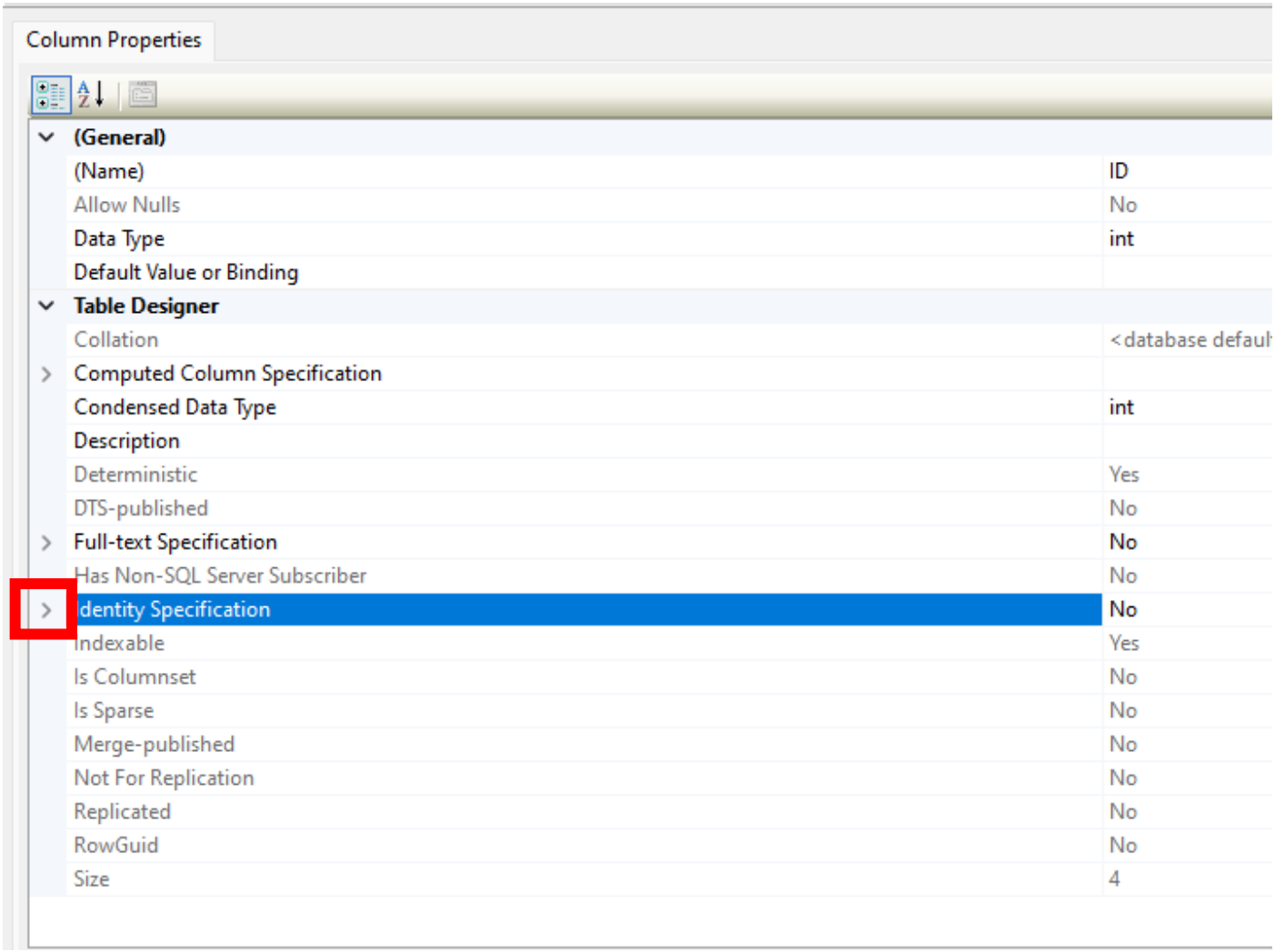






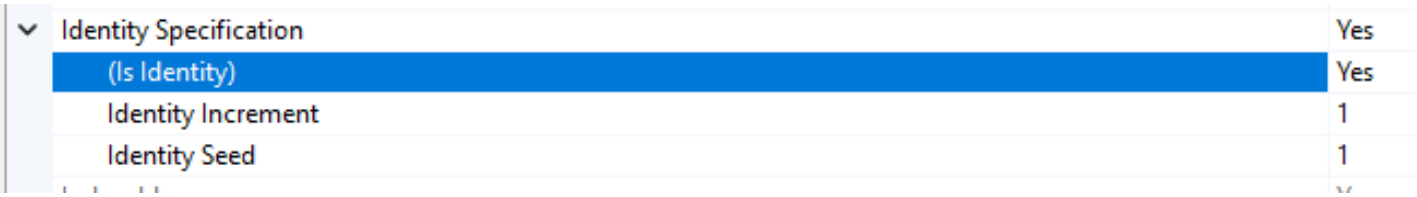
ال auto increment بنلاقيها في الجزء اللي تحت اللي اسمه column properties

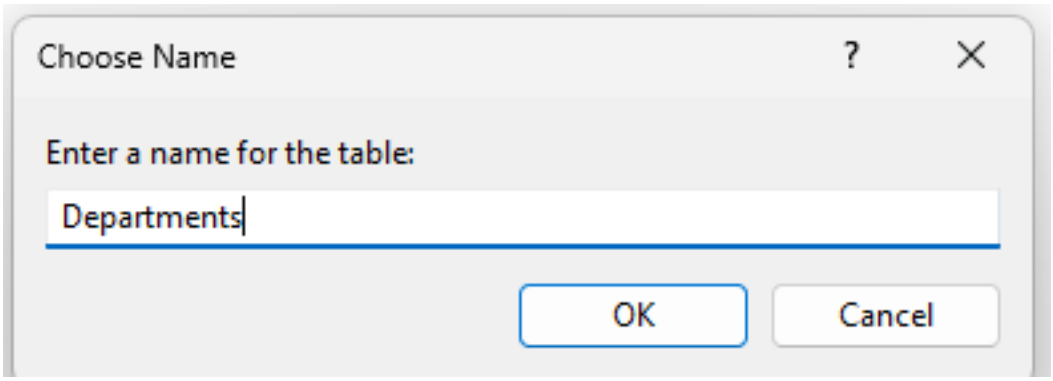




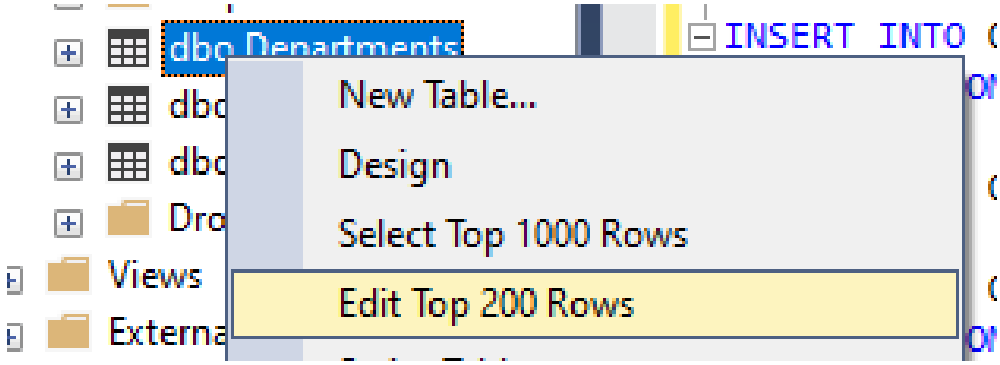
بتعدين بتدوس 2 كليك بالماوس علي is identity هتتقلب ب yes وبعدين هيظهر تحتها identity increment ودي بتقولك عاوز لما تحط record جديد ال id يزيد بكام

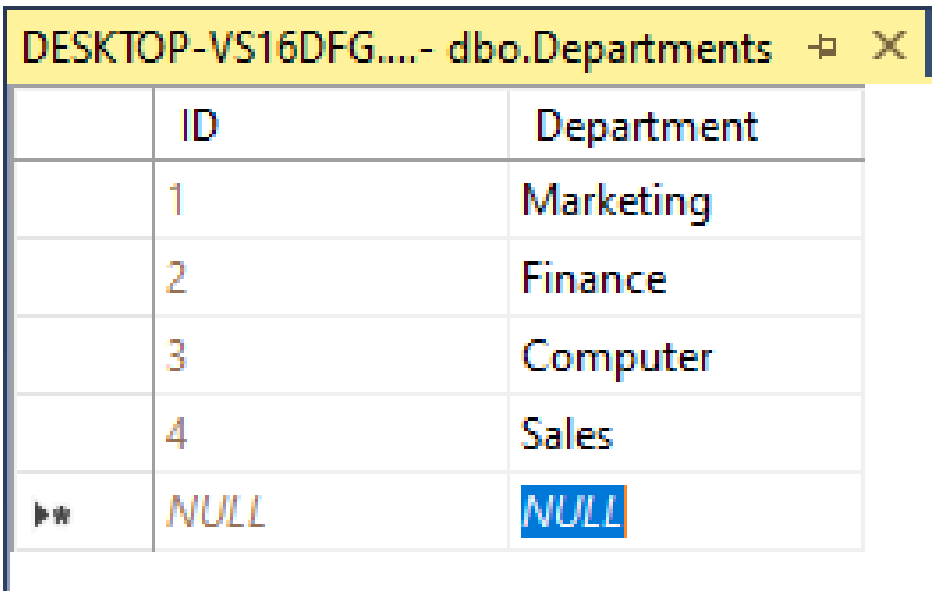
ال identity seed يتقولك عاوز العد يبدأ من كام





تعالي بقي نجرب ندخل أي حاجه بالماوس في الجدول لو جيت تكتب ال id بايديك مش هيقبل لانه اصبح بيتعدل من نفسه





حتي لما تيجي تعمل أوامر ال insert وجيت تحط قيمه لل id هيطلعلك error عشان كده طالما عملت عمود من الاعمده auto increment ماتحاولش تدخل قيم ليه

```
use DB1;

SELECT * FROM Departments;

INSERT INTO Departments
values('HR');
```

Results Messages

	ID	Department
1	1	Marketing
2	2	Finance
3	3	Computer
4	4	Sales
5	5	HR
6	6	HR
7	7	HR
8	8	HR
9	9	HR
10	10	HR
11	11	HR

طيب لو عايز اعرف اخر id موجود في الجدول بكتب السطر ده هنيجي لشرحه بعدين

```
print @@identity
```

Messages

```
11

Completion time: 2023-08-26T12:52:25.6505757+03:00
```

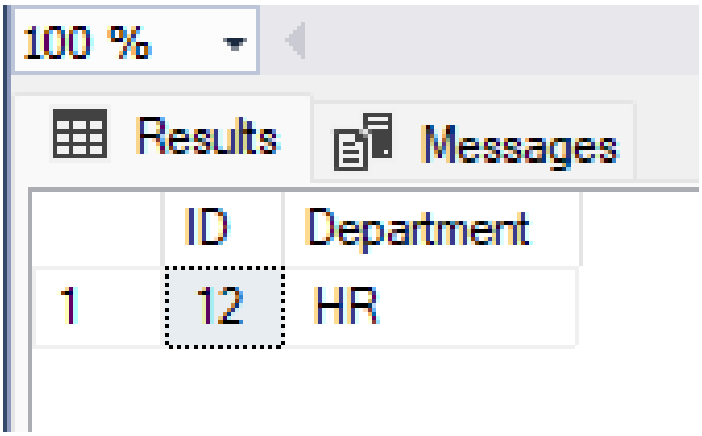
هوا هنا بيخزن ال id في متغير وبيبدأ يزود عليه

طيب لو انا جيت وحذفت الداتا اللي في الجدول كلها وبعدها ضيفت records جديده هل هيبدأ يعد من 1 تاني ؟

قالك لا بيكمل من العدد من حيث انتهي

```
use DB1;

SELECT * FROM Departments;

INSERT INTO Departments
```

```
values('HR');

print @@identity

delete from Departments;
```



عشان اعمل نفس الجدول بالكود بعمله عادي جدا بس باجي بعد ال DATA TYPE وبكتب identity وبحط الرقم اللي هيزيد بيه والرقم اللي هيبدأ بيه العدد

```
create table Departments(
ID int identity(1,1)not null,
Department nvarchar(200) not null,
Primary key(ID)
);
```

AUTO INCREMENT Field

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

Syntax for SQL Server

The following SQL statement defines the "Personid" column to be an auto-increment primary key field in the "Persons" table:

```
CREATE TABLE Persons (
  Personid int IDENTITY(1,1) PRIMARY KEY,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int
);
```

The MS SQL Server uses the `IDENTITY` keyword to perform an auto-increment feature.

In the example above, the starting value for `IDENTITY` is 1, and it will increment by 1 for each new record.

Tip: To specify that the "Personid" column should start at value 10 and increment by 5, change it to `IDENTITY(10,5)`.

To insert a new record into the "Persons" table, we will NOT have to specify a value for the "Personid" column (a unique value will be added automatically):

```
INSERT INTO Persons (FirstName,LastName)
VALUES ('Mohammed','Abu-Hadhoud');
```

The SQL statement above would insert a new record into the "Persons" table. The "Personid" column would be assigned a unique value. The "FirstName" column would be set to "Mohammed" and the "LastName" column would be set to "Abu-Hadhoud".

```
CREATE TABLE Departments (
    ID int identity(1,1)  NOT NULL,
    Name nvarchar(50) NOT NULL,
    PRIMARY KEY (ID)
);



-----------------------

insert into Departments
values ('HR');

print @@identity;
```

Delete vs Truncate statement.

ال truncate هيا هيا ال delete بس بتفرق عنها في حاجتين اول حاجه انك ماينفعش تستخدم فيها ال where يعني ماينفعش تضيف شروط لانها بتحذف كل ال records وتاني حاجه انها بتصفر ال identity يعني بعد ماتحذف وتضيف record جديد هيبدا يعد معاك من البدايه

ال delete بيمسحهم واحد واحد انما ال truncate بيعمل format للجدول

```
SELECT * FROM Departments;

insert into Departments
values ('HR');
```

Results Messages

	ID	Department
1	1	HR
2	2	HR
3	3	HR

```
truncate table Departments;
```

100 %

Results Messages

	ID	Department
1	1	HR

Delete Vs Truncate

The main difference between both statements is that `DELETE FROM` statement supports `WHERE` clause whereas `TRUNCATE` does not.

Also the `DELETE FROM` statement does not reset the auto number (identity field) while the `TRUNCATE` does reset the identity fields.

That means, we can delete single or multiple rows using the `DELETE FROM` statement while the `TRUNCATE` statement deletes all records from the table at once.

We can mimic the `TRUNCATE` statement with `DELETE FROM` statement by omitting the `WHERE` clause. For example,

```
DELETE FROM Customers;
```

is similar to,

```
TRUNCATE TABLE Customers;
```

```
select * from Departments;

--this will delete all rows but will not reset the identity counter.
delete from Departments;


--this will delete all rows and reset the identity counter.
truncate table departments;

insert into Departments
values ('HR');

print @@identity;
```

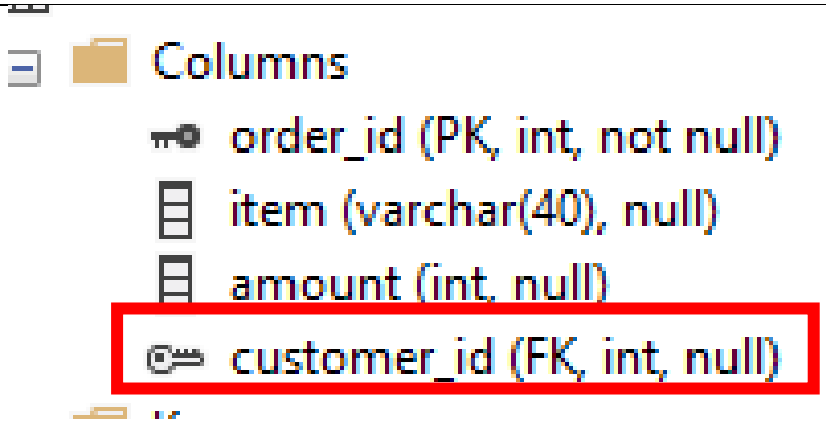
Foreign Key Constraint

ال primary key هوا constraint وال not null برضه constraint وكمان ال foreign key هوا عباره عن constraint واتكلمنا عن ال foreign key قبل كده

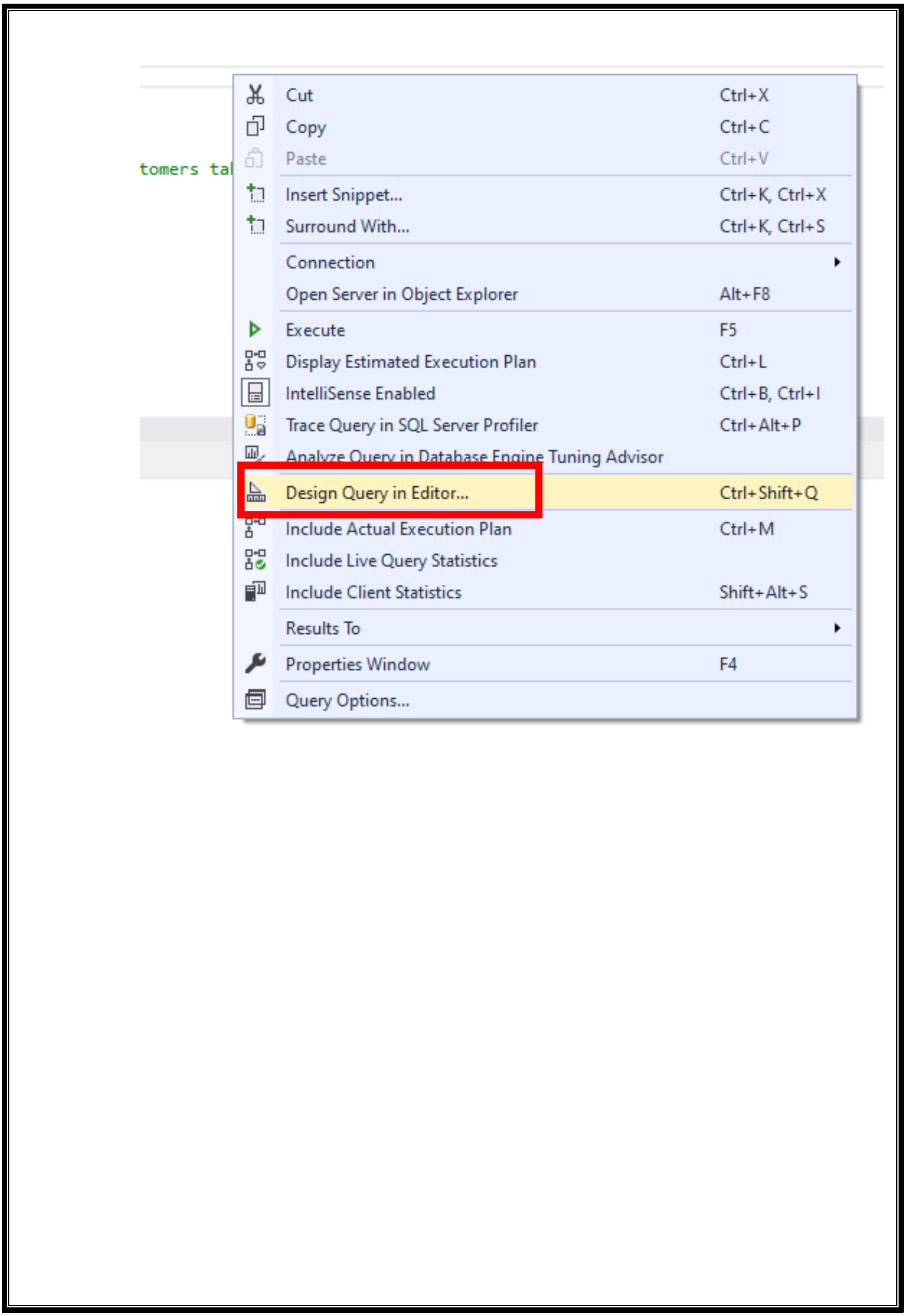
تعالي نعمل الجدولين دول جدول customers وجدول orders هنحط foreign key لجدول ال orders عشان يقولنا مين اللي اكل الجبنه (مين اللي طلب الاوردر) فبيقولك الجدول اللي من غير foreign key خليه هوا الأول في الكود

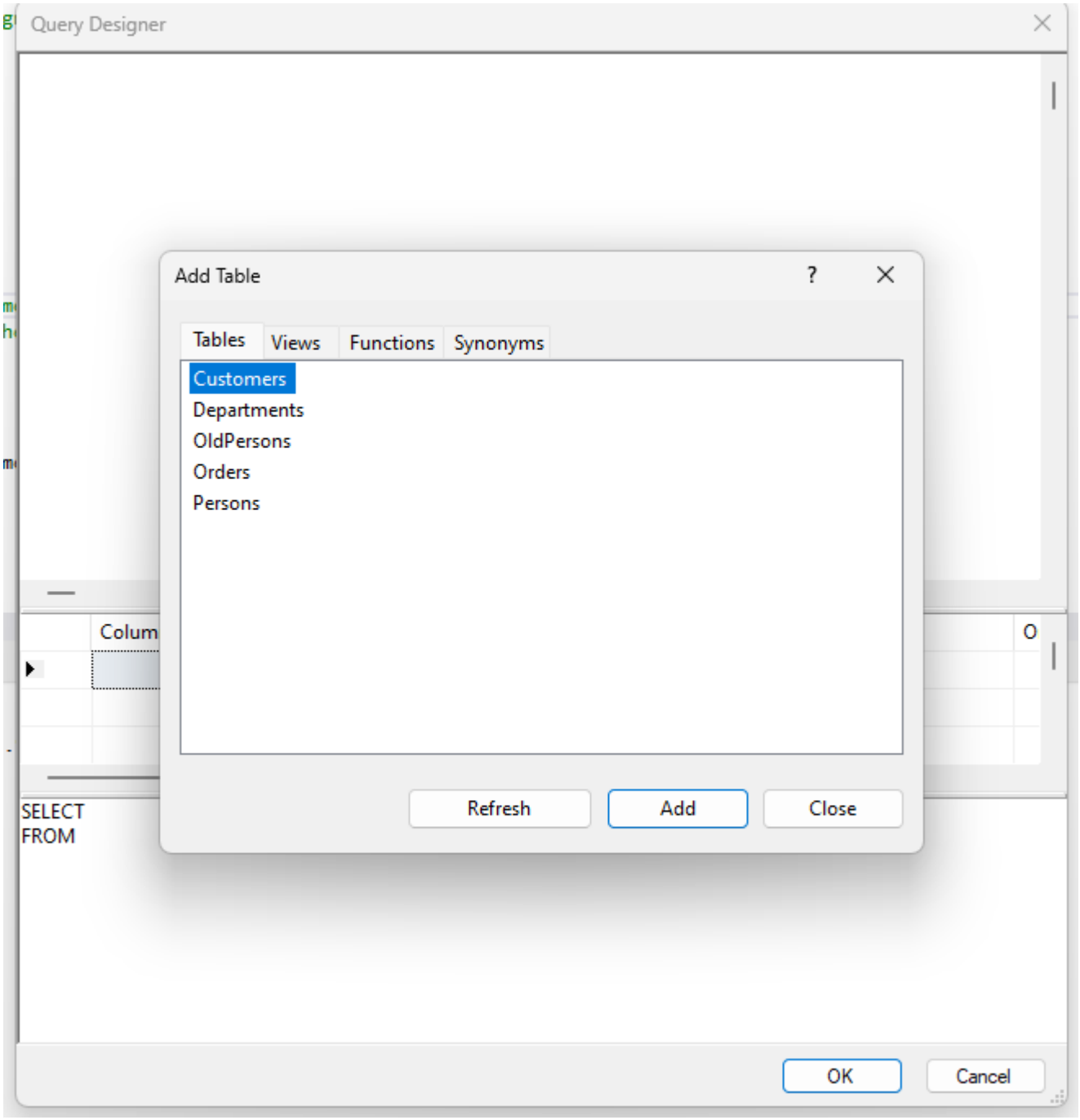
عشان اعمل foreign key بعمل عمود عادي وبعد ال data type بكتب references وبعدين اسم الجدول اللي عايز اربط بيه وافتح قوسين اكتب فيهم اسم العمود اللي بيمثل ال primary key في الجدول الاولاني

```sql
use DB1;

-- This table doesn't have a foreign key
CREATE TABLE Customers (
  id INT ,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  PRIMARY KEY (id)
);


-- Adding foreign key to the customer_id field
-- The foreign key references to the id field of the Customers
table
CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT REFERENCES Customers(id),
  PRIMARY KEY (order_id)
);
```

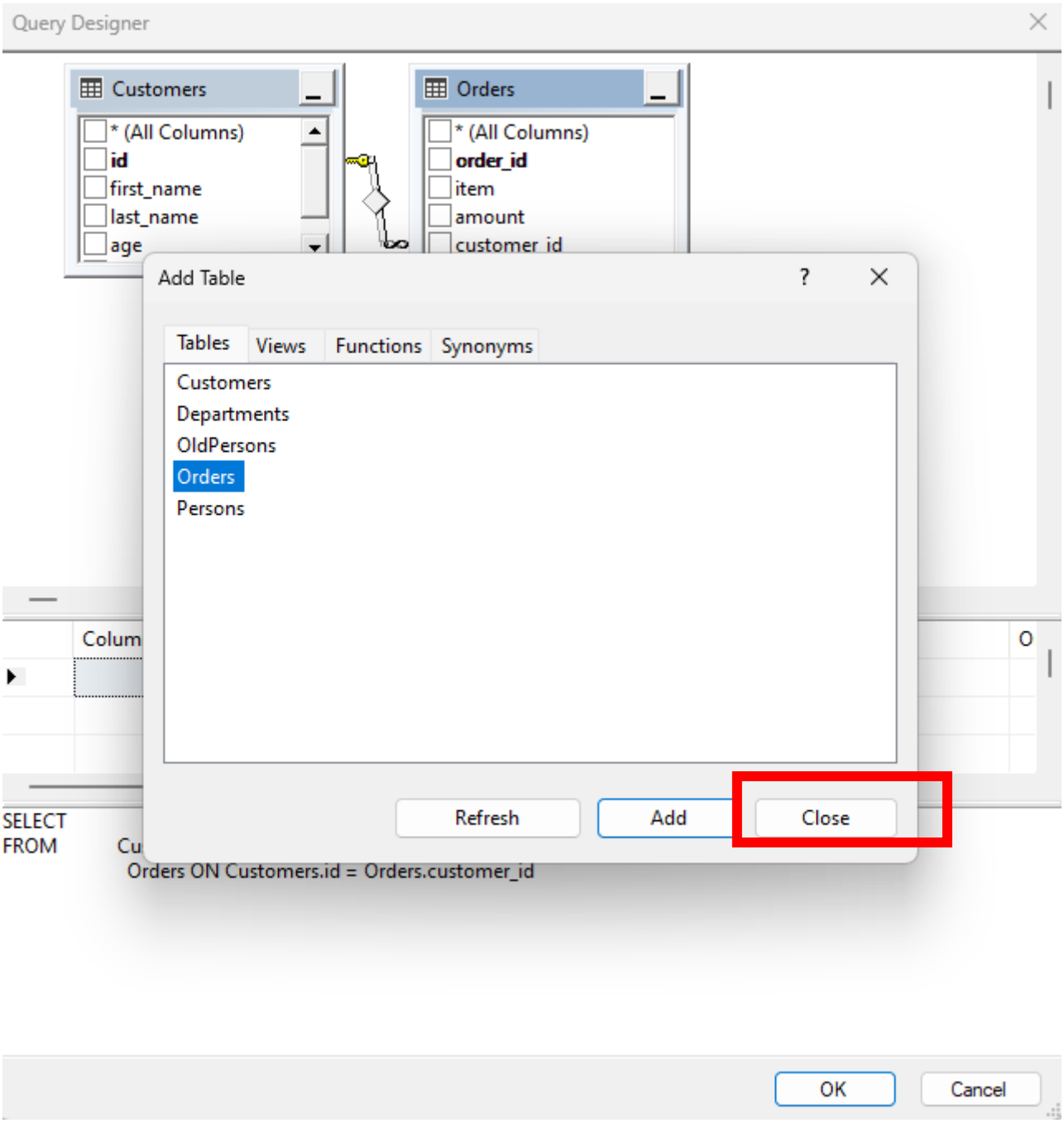


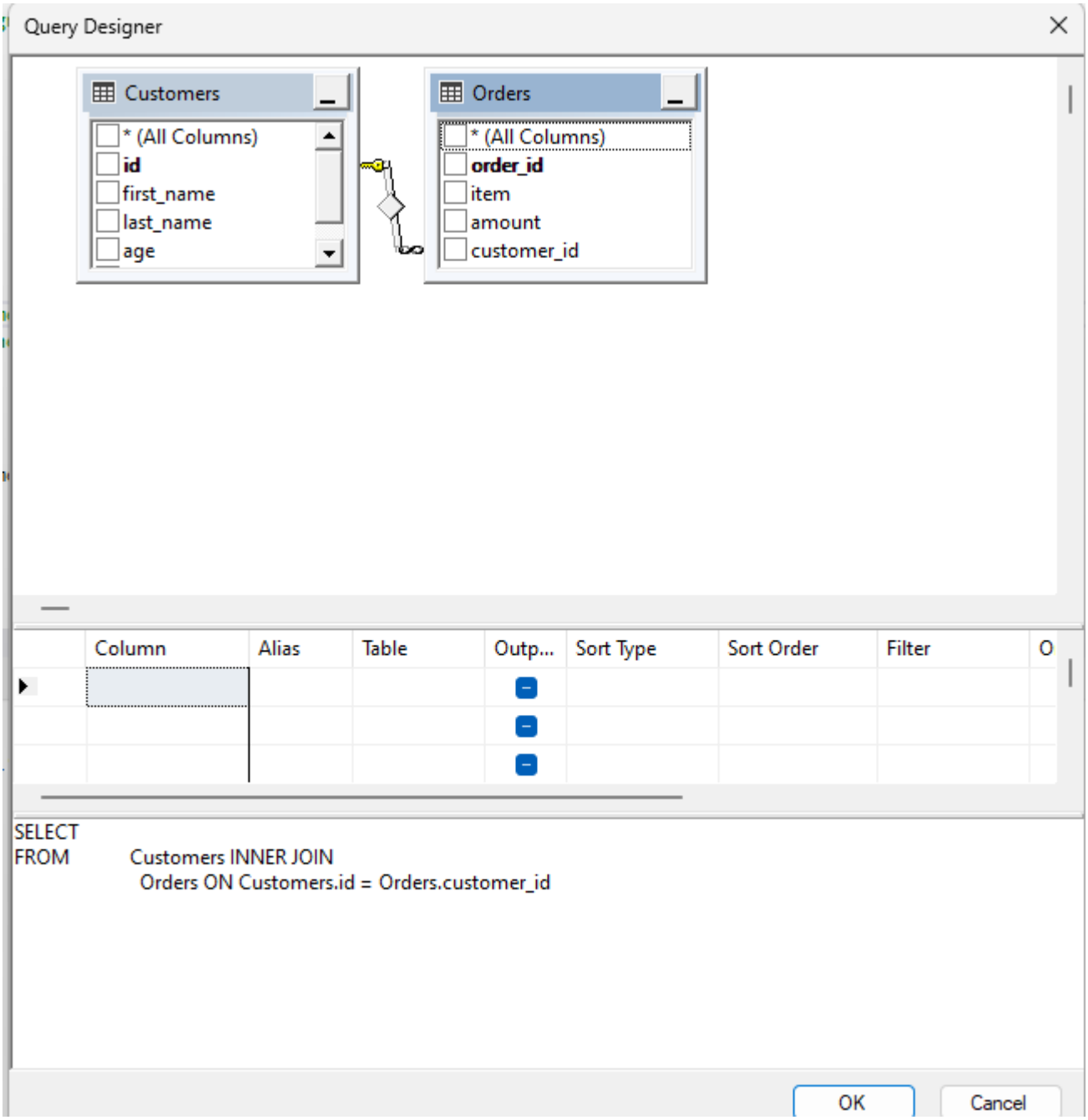
عشان تشوف العلاقه كليك يمين علي شاشة ال query


tomers tal
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Insert Snippet... Ctrl+K, Ctrl+X
Surround With... Ctrl+K, Ctrl+S
Connection
Open Server in Object Explorer Alt+F8
Execute F5
Display Estimated Execution Plan Ctrl+L
IntelliSense Enabled Ctrl+B, Ctrl+I
Trace Query in SQL Server Profiler Ctrl+Alt+P
Analyze Query in Database Engine Tuning Advisor
Design Query in Editor... Ctrl+Shift+Q
Include Actual Execution Plan Ctrl+M
Include Live Query Statistics
Include Client Statistics Shift+Alt+S
Results To
Properties Window F4
Query Options...

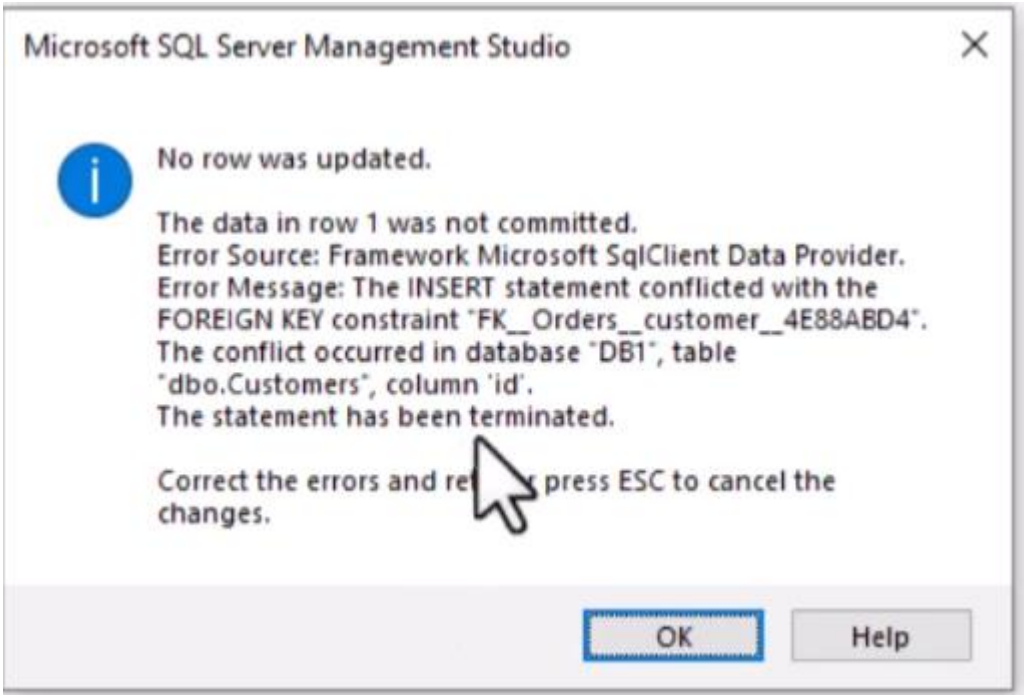


نقرتين بالماوس علي اسم الجدول هيظهرلك ورا

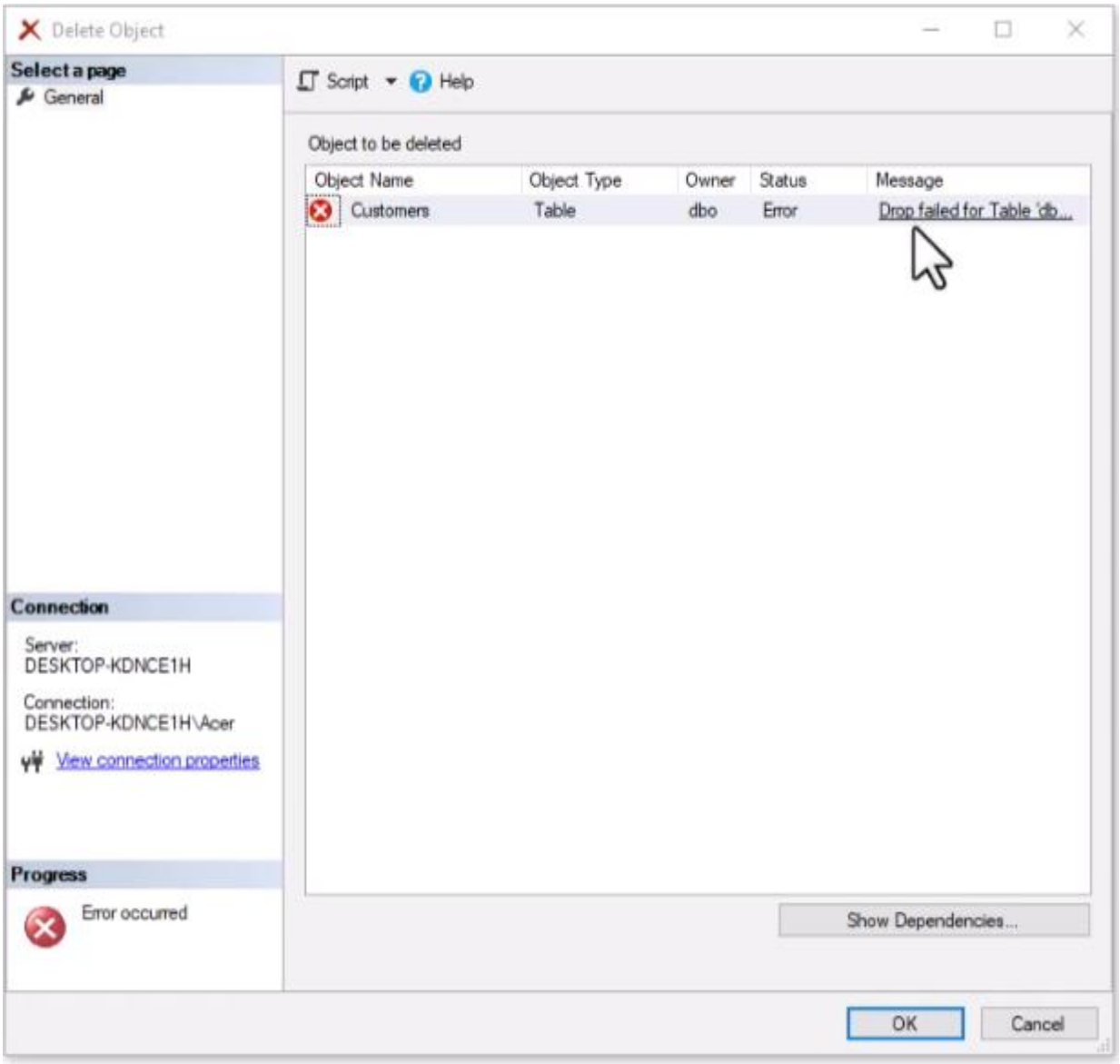

Query Designer
Customers
* (All Columns)
id
first_name
last_name
age
Orders
* (All Columns)
order_id
item
amount
customer_id
Add Table
Tables
Views
Functions
Synonyms
Customers
Departments
OldPersons
Orders
Persons
Refresh
Add
Close
SELECT
FROM
Orders ON Customers.id = Orders.customer_id
OK
Cancel



لوجينا في جدول ال orders ونزود اوردر وجينا نحط في ال customer id رقم مش موجود في جدول ال customers هيطلعلي خطأ



لو جيت تحذف جدول ال customers هيطلعلك error لانه مرتبط بجدول تاني لازم احذف جدول ال orders الأول عشان اقدر احذف جدول ال customers



لو جيت تعمل الجدولين من غير ماتضيف السطر ده هيتعملوا مش هيقولك لا بس كل جدول مش هيكون ليه علاقه بالجدول التاني هيكون كل واحد لوحده

```
INT REFERENCES Customers(id),
```

```
use DB1;

-- This table doesn't have a foreign key
CREATE TABLE Customers (
  id INT ,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  PRIMARY KEY (id)
);


-- Adding foreign key to the customer_id field
-- The foreign key references to the id field of the Customers
table
CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT,
  PRIMARY KEY (order_id)
);
```

طيب عشان اضيف ال reference هعمل alter table واقوله يضيف foreign key واكتب اسم العمود اللي عايز اعمله ال fk وبعدها بكتب السطر بتاع ال references عادي

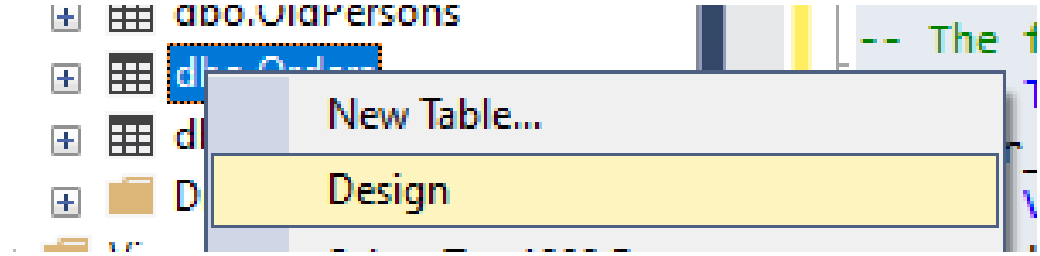
```
alter table Orders
add FOREIGN KEY(order_id) references Customers(ID)
```

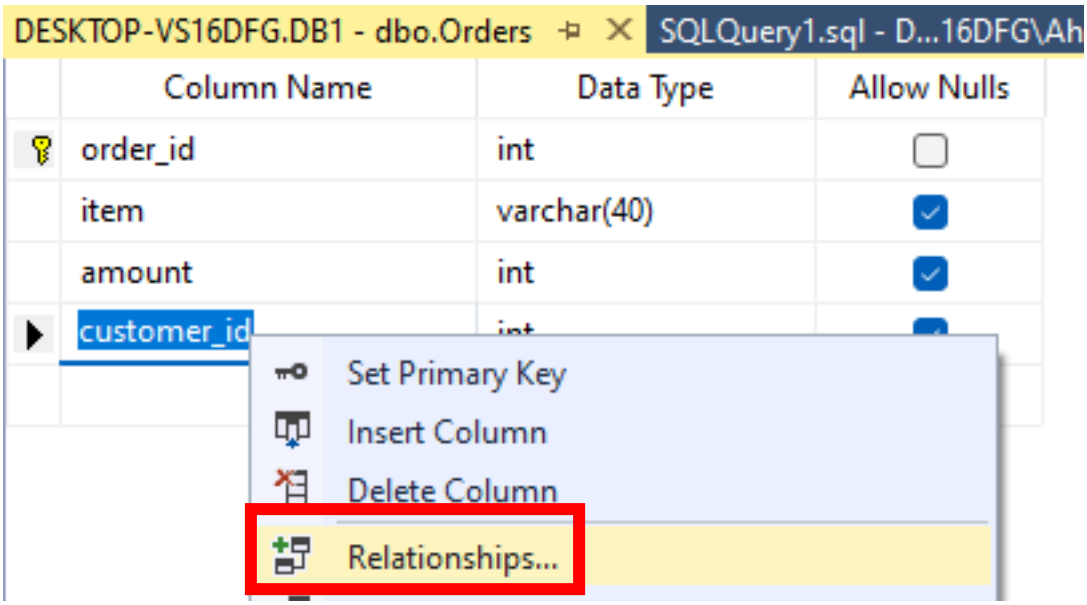
طيب هنحذف الجدولين ونعملهم تاني من غير مانربطهم

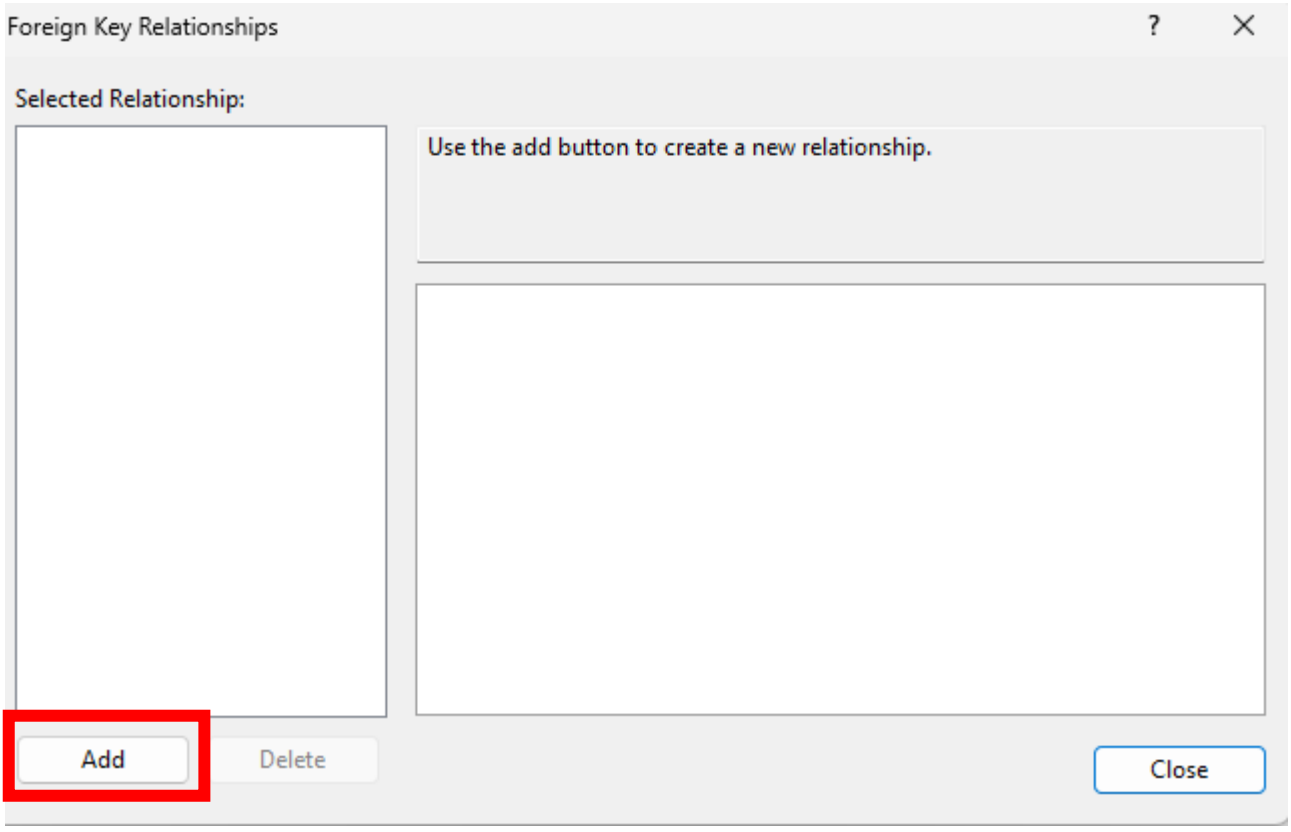
```
DROP TABLE Orders;
DROP TABLE Customers;

-- This table doesn't have a foreign key
CREATE TABLE Customers (
  id INT ,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  PRIMARY KEY (id)
);


-- Adding foreign key to the customer_id field
-- The foreign key references to the id field of the Customers table
```

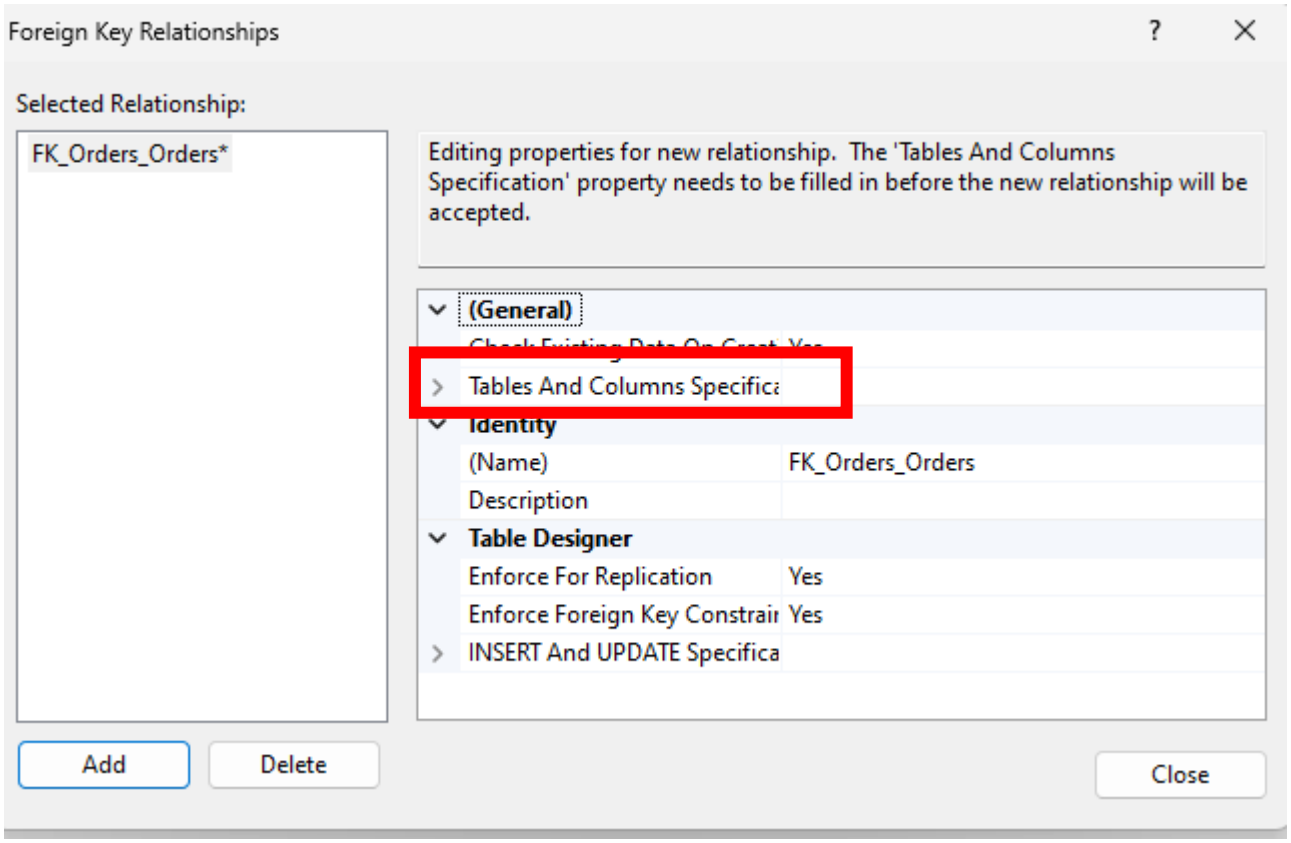
```sql
CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT ,
  PRIMARY KEY (order_id)
);
```

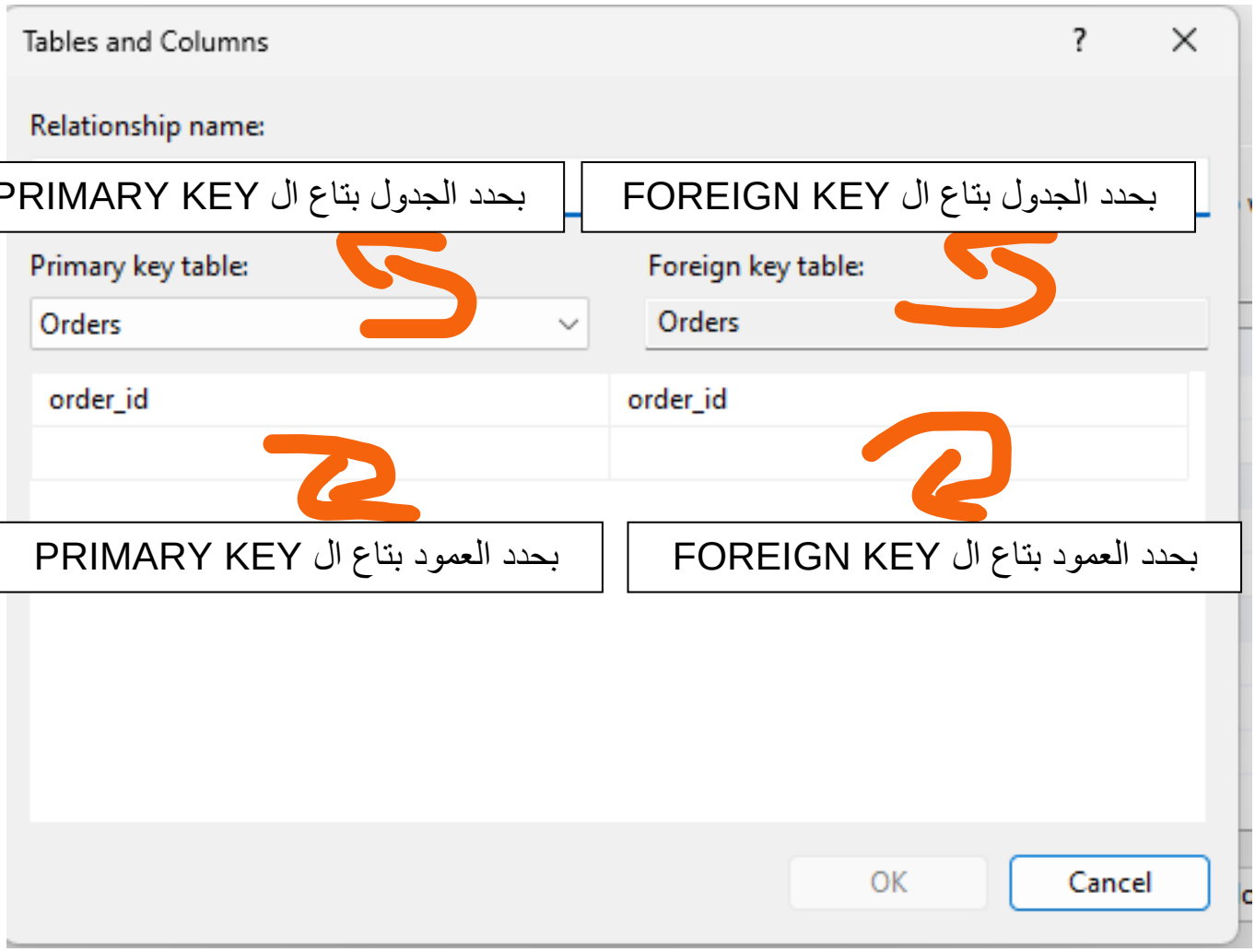
عاوزين نضيف ال FOREIGN KEY عن طريق ال DESIGN

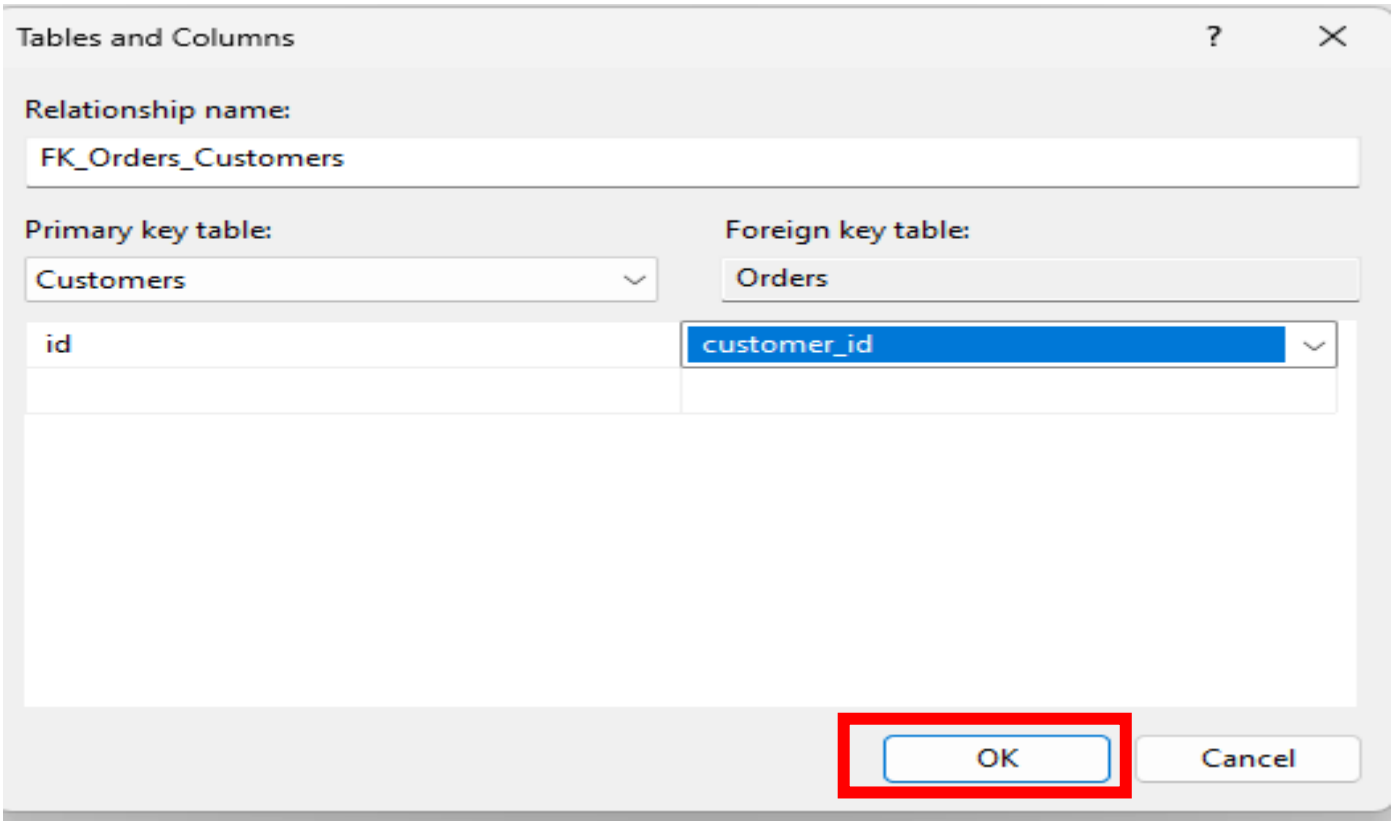


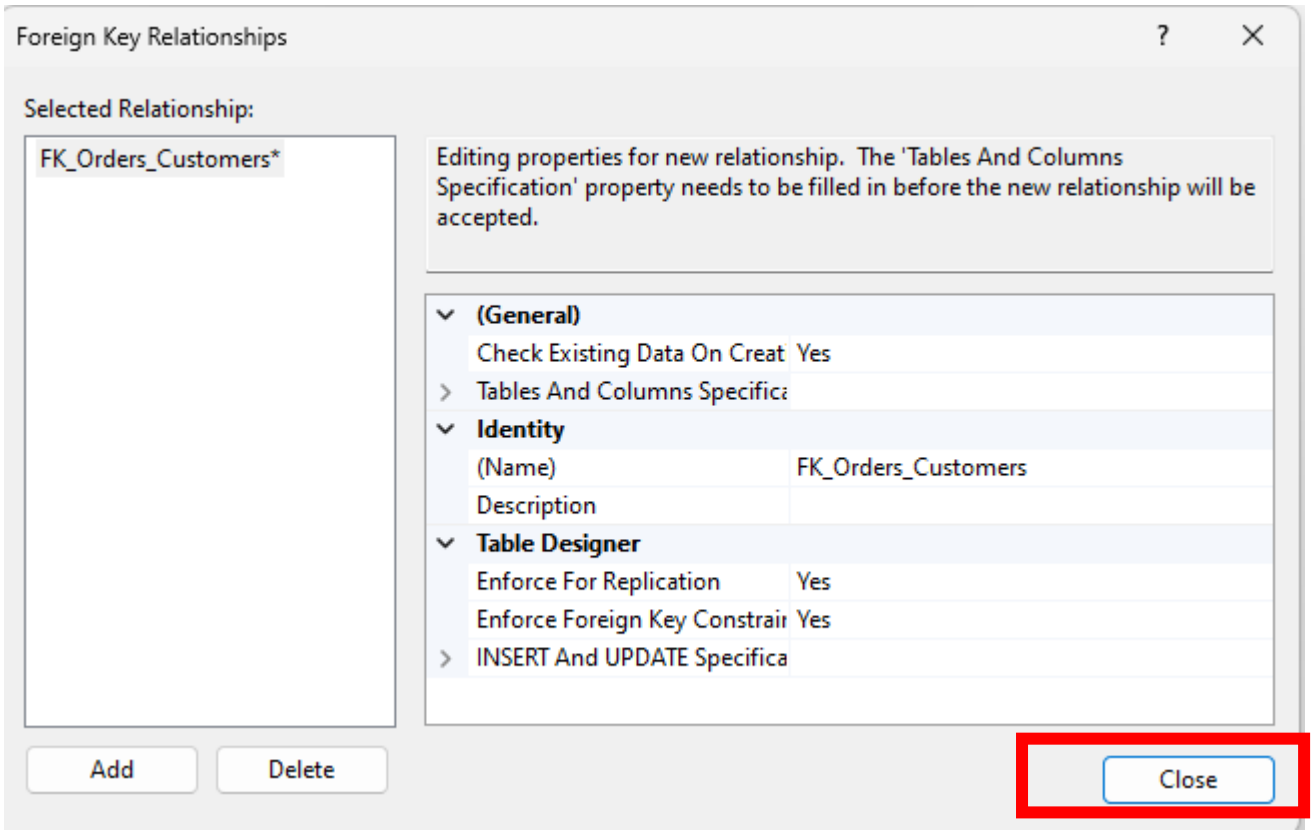


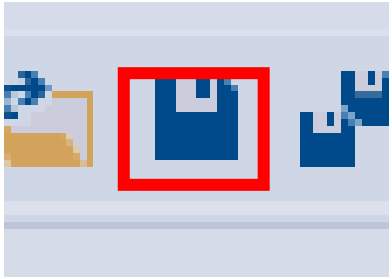


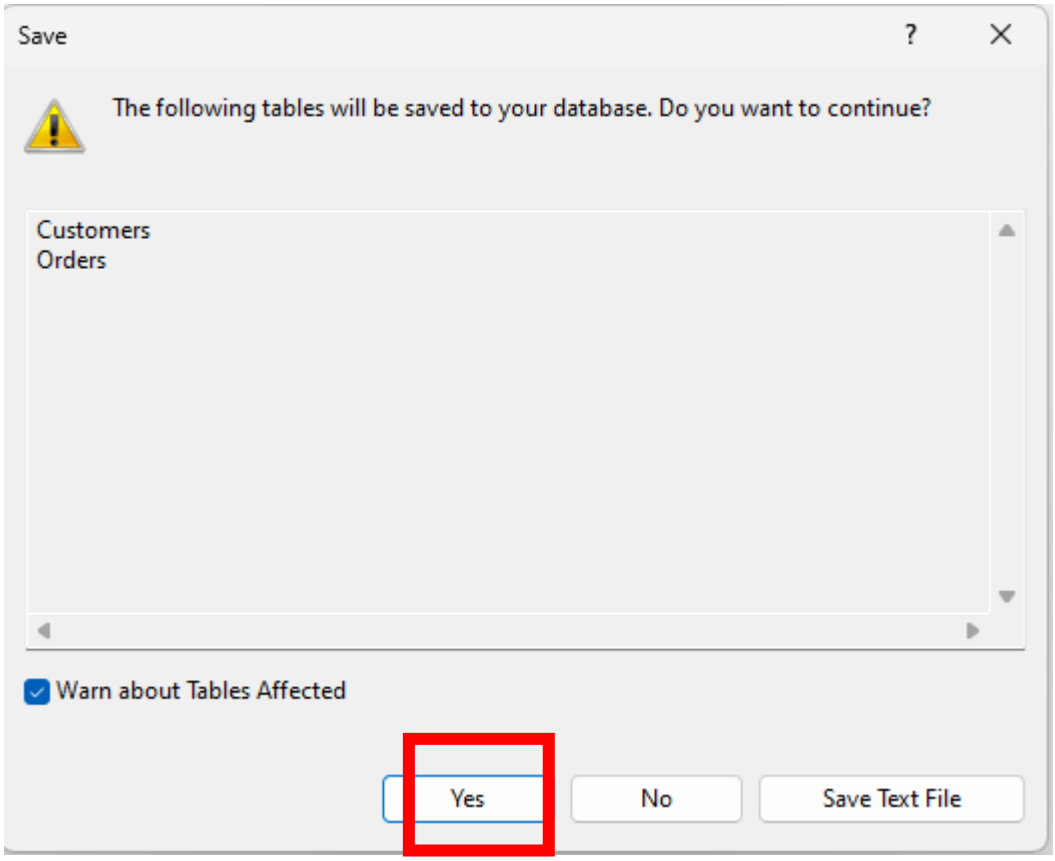

Foreign Key Relationships
Selected Relationship:
FK_Orders_Orders*
Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.
(General)
Tables And Columns Specifica
Identity
(Name)
FK_Orders_Orders
Description
Table Designer
Enforce For Replication
Yes
Enforce Foreign Key Constrai
Yes
INSERT And UPDATE Specifica
Add
Delete
Close


Tables and Columns
Relationship name:
بحدد الجدول بتاع ال PRIMARY KEY
بحدد الجدول بتاع ال FOREIGN KEY
Primary key table:
Orders
Foreign key table:
Orders
order_id
order_id
بحدد العمود بتاع ال PRIMARY KEY
بحدد العمود بتاع ال FOREIGN KEY
OK
Cancel


Tables and Columns
Relationship name:
FK_Orders_Customers
Primary key table:
Customers
Foreign key table:
Orders
id
customer_id
OK
Cancel


Foreign Key Relationships
Selected Relationship:
FK_Orders_Customers*
Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.
(General)
Check Existing Data On Creat Yes
Tables And Columns Specifica
Identity
(Name) FK_Orders_Customers
Description
Table Designer
Enforce For Replication Yes
Enforce Foreign Key Constrair Yes
INSERT And UPDATE Specifica
Add
Delete
Close





SQL FOREIGN KEY

In this tutorial, we'll learn about the FOREIGN KEY in SQL and how to use them with the help of examples.

In SQL, we can create a relationship between two tables using the `FOREIGN KEY` constraint.

Here, the customer_id field in the Orders table is `FOREIGN KEY` which references the id field in the Customers table.

This means that the value of the customer_id (of the Orders table) must be a value from the id column (of the Customers table).

Note: The Foreign key can be referenced to any column in the parent table. However, it is general practice to reference the foreign key to the primary key of the parent table.

Creating FOREIGN Key

Now, let's see how we can create foreign key constraints in a database.

```
-- This table doesn't have a foreign key
CREATE TABLE Customers (
  id INT ,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  PRIMARY KEY (id)
);


-- Adding foreign key to the customer_id field
-- The foreign key references to the id field of the Customers table
CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT REFERENCES Customers(id),
  PRIMARY KEY (order_id)
);
```

Here, the value of the customer_id column in the Orders table references the row in another table named Customers with its id column.

Note: The above code works in all major database systems. However, there may be the alternate syntax to create foreign keys depending on the database. Refer to their respective database documentation for more information.

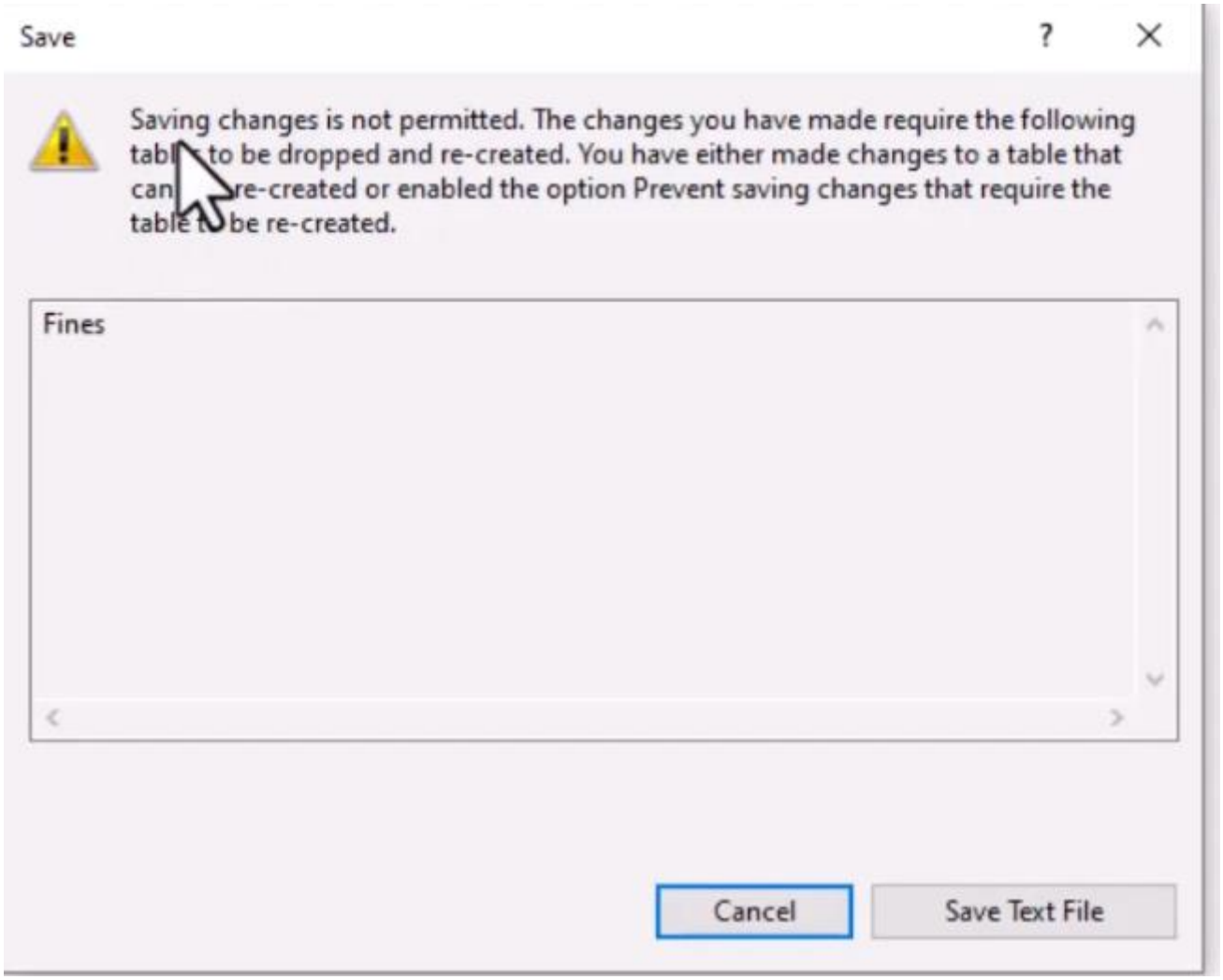
Foreign Key with Alter Table

It is possible to add the `FOREIGN KEY` constraint to an existing table using the `ALTER TABLE` command. For example,

```
-- This table doesn't have a foreign key
CREATE TABLE Customers (
  id INT ,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  PRIMARY KEY (id)
);

CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT ,
  PRIMARY KEY (order_id)
);

-- Adding foreign key to the customer_id field using alter

ALTER TABLE Orders
ADD FOREIGN KEY (customer_id) REFERENCES Customers(id);
```

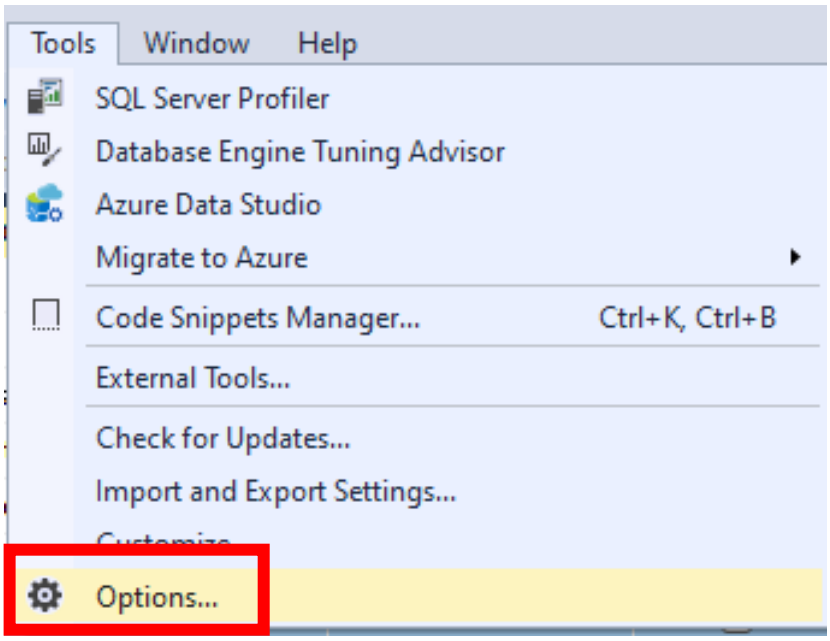
Solution To: "Saving changes is not permitted" error

هنا بيقولك انه فيه ناس لما تيجي تعدل علي جدول وتحفظه تطلعلك الرسااله دي

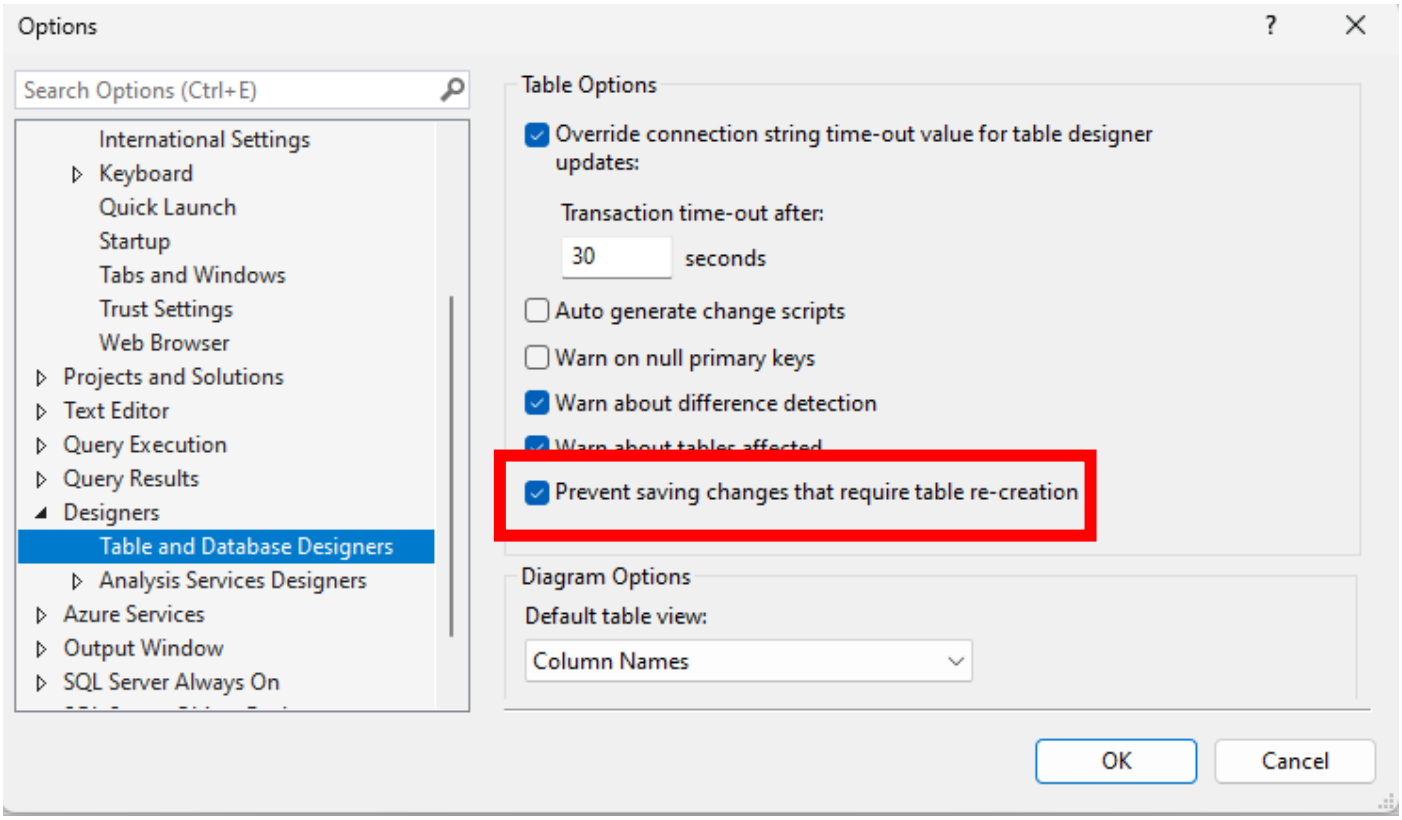


بيقولك انه ده بيحصل لانه فيه اوبشن بيخليك لما تيجي تعدل علي عمود او تضيف عمود جديد بيحذف الجدول ويعمل جدول جديد مكانه بنفس المواصفات

عشان تصلحها بتعمل كده



وبعدين بتشيل الcheck من هنا



Solution To: "Saving changes is not permitted" error:

If you encounter the "Saving changes is not permitted" error in SQL Server when attempting to modify a table that requires re-creation, you can change a setting in SQL Server Management Studio (SSMS) to allow saving changes that require table re-creation. Here's how you can do it:

1. Open SQL Server Management Studio.
2. Go to the "Tools" menu and select "Options."
3. In the Options window, navigate to "Designers" > "Table and Database Designers."
4. Uncheck the option "Prevent saving changes that require table re-creation."
5. Click "OK" to save the changes.

After making this configuration change, you should be able to modify and save changes to your tables without encountering the "Saving changes is not permitted" error. However, keep in mind that making significant changes to a table's structure can have implications on existing data and may require careful consideration and backup procedures to avoid data loss or inconsistencies. Exercise caution when making structural changes to production databases.

SQL - Queries

Restore Sample HR Database and Get Ready .

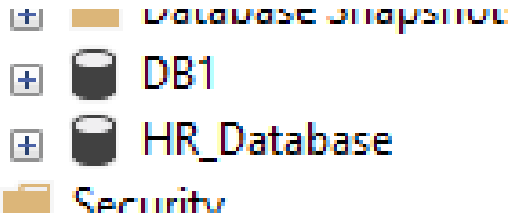
هتنزل الداتا بيز دي

https://cdn.fs.teachablecdn.com/vB7M2lSPQ7WwpJrk8Ggi

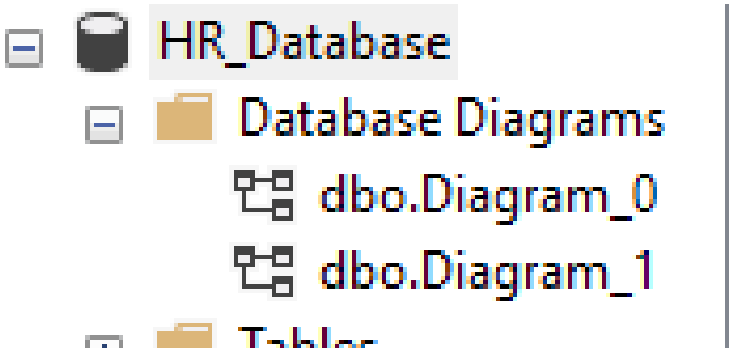
وتحطها عندك عال sql server

```
RESTORE DATABASE HR_Database
FROM DISK='D:\HR_Database.bak';
```

وبيقولك لو حصل خطأ حط السطر ده



وبعدين هنروح نبص عال DIAGRAM



لو مالقيتش الملفين دول وظهرلك رسالة خطأ

هتكتب السطرين دول في ال QUERY وتشغلهم

```
use HR_Database;

EXEC sp_changedbowner 'sa';
```

Select Statement

هنبدأ في ال DQL اللي هيا عباره عن ال select statement

Query معناها استعلام

زي ماعرفنا قبل كده انه الجمله دي بترجعلي كل حاجه من الجدول

```
USE HR_Database;

SELECT * FROM Countries;

SELECT * FROM Departments;

SELECT * FROM Employees;
```

فيه طريقه تانيه اكتب بيها ال QUERY دي وهيا اني اذكر اسم الجدول قبل علامة النجمه

```
SELECT Countries.* FROM Countries;

SELECT Departments.* FROM Departments;
```

```
SELECT Employees.* FROM Employees;
```

	ID	Name
1	1	Engineering
2	2	Accounting
3	3	Marketing
4	4	IT
5	5	HR
6	6	Finance
7	7	Sales

Results Mes

	ID	Name
1	1	USA
2	2	UK
3	3	China

Results Messages

	ID	FirstName	LastName	Gendor	DateOfBirth	CountryID	DepartmentID	HireDate	ExitDate	MonthlySalary	BonusPerc
1	285	Kai	Le	M	1972-09-20 00:00:00	1	1	2022-02-05 00:00:00	NULL	709.00	0
2	286	Robert	Patel	M	1981-06-10 00:00:00	1	7	2013-10-23 00:00:00	NULL	2708.00	0
3	287	Cameron	Lo	M	1994-05-26 00:00:00	2	4	2019-03-24 00:00:00	NULL	372.00	0
4	288	Harper	Castillo	F	1996-01-21 00:00:00	1	4	2018-04-07 00:00:00	NULL	922.00	0
5	289	Harper	Dominguez	F	1964-12-19 00:00:00	1	1	2005-06-18 00:00:00	NULL	535.00	0.24
6	290	Ezra	Vu	M	1994-02-12 00:00:00	1	4	2004-04-22 00:00:00	2014-02-14 00:00:00	2701.00	0
7	291	Jade	Hu	F	1997-04-17 00:00:00	2	2	2009-06-27 00:00:00	NULL	395.00	0
8	292	Miles	Chang	M	1987-06-14 00:00:00	2	6	1999-02-19 00:00:00	NULL	1415.00	0
9	293	Gianna	Holmes	F	1994-09-10 00:00:00	1	4	2011-09-09 00:00:00	NULL	694.00	0
10	294	Jameson	Thomas	M	1962-12-23 00:00:00	1	6	2015-02-05 00:00:00	NULL	2377.00	0.1
11	295	Jameson	Pena	M	1979-01-07 00:00:00	1	4	2003-10-12 00:00:00	NULL	1258.00	0
12	296	Bella	Wu	F	1995-02-09 00:00:00	1	6	2014-08-03 00:00:00	NULL	794.00	0
13	297	Jose	Wong	M	1990-07-03 00:00:00	2	4	2017-11-15 00:00:00	NULL	2374.00	0.23
14	298	Lucas	Richardson	M	1983-10-04 00:00:00	1	3	2018-07-22 00:00:00	NULL	1751.00	0.08
15	299	Jacob	Moore	M	1977-12-29 00:00:00	1	3	2021-03-24 00:00:00	NULL	659.00	0.15
16	300	Luna	Lu	F	1998-01-15 00:00:00	1	4	1997-07-26 00:00:00	NULL	1929.00	0
17	301	Bella	Tran	F	1967-06-01 00:00:00	2	1	2010-08-05 00:00:00	NULL	2555.00	0.33
18	302	Ivy	Chau	F	1988-01-04 00:00:00	2	7	2019-03-03 00:00:00	NULL	465.00	0

طيب انا جدول الموظفين مشش عايز منه كل الداتا دي عايز اعمده معينه زي ال ID والاسم والمرتب

قالك كل اللي هتعمله انك هتكتب اسم العمود او الاعمدة اللي عايزها بدل علامة النجمه

يعني علامة النجمه وظيفتها انها تختار كل الاعمده والشروط اللي كنا بنعملها بال WHERE دي بتتحكم في عدد ال RECORDS

```
SELECT ID,FirstName,LastName,MonthlySalary FROM Employees;
```

طيب لو عايزين نرجع عمود تاريخ الميلاد والاسم الأول

```
SELECT ID,FirstName,DateOfBirth FROM Employees;
```

The SQL SELECT Statement

The `SELECT` statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

SELECT Syntax

SELECT *column1, column2, ...*

FROM *table_name*;

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

SELECT * FROM *table_name*;

MYSQL

```
Select * from Employees;

Select Employees.* from Employees;

Select ID, FirstName, LastName,MonthlySalary From Employees;

Select ID, FirstName, DateOfBirth From Employees;

Select * from Departments;

Select * from Countries;
```

Select Distinct Statement

كلمة DISTINCT بترجع الداتا بدون تكرار

لو عايز مثلا اني اعرف ايه الأقسام اللي موجود فيها موظفين عندي وجيت اعملها بامر

SELECT departmentid هترجع مكرره انما لو زودت كلمة distinct هترجع من غير تكرار

```
SELECT DepartmentID FROM Employees;

SELECT DISTINCT DepartmentID FROM Employees;
```

	DepartmentID
1	3
2	6
3	7
4	1
5	4
6	5
7	2

Results | Messa

	DepartmentID
1	1
2	7
3	4
4	4
5	1
6	4
7	2
8	6
9	4
10	6
11	4
12	6
13	4
14	3
15	3
16	4
17	1
18	7

اقدر اعمل ده علي أي عمود تاني زي ال FIRST NAME مثلا

```
SELECT FirstName FROM Employees;

SELECT DISTINCT FirstName FROM Employees;
```

	FirstName
195	Skylar
196	Sofia
197	Sophia
198	Sophie
199	Stella
200	Theodore
201	Thomas
202	Valentina
203	Victoria
204	Violet
205	Vivian
206	Wesley
207	William
208	Willow
209	Wyatt
210	Xavier
211	Z e
212	Z ey

	FirstName
983	Alexander
984	Logan
985	Henry
986	Delilah
987	Caroline
988	Jack
989	Luna
990	John
991	Charlotte
992	Miles
993	Violet
994	Isaac
995	Ian
996	Melody
997	Eliza
998	Layla
999	Thomas
1000	Willow

لو جيت اعملها علي عمودين هيعمل ايه ؟

اعتبر ان النتيجه اللي طالعه من العمودين هما record واحد علي بعضه عادي بغض النظر عن كام عمود في الجدول اللي انا جايب منه الداتا وهيشوف هل ال record ده كله علي بعضه متكرر في الجدول اللي طلع ولا لا لو متكرر هيشيل التكرار ولو مش متكرر هيسيبه

طيب لو فيه قيم في العمود الأول او العمود التاني متكرره هيشيلهم ؟

ماخلاص بقي احنا قولنا انه هيتعامل مع القيم اللي في الجدولين كانهم record واحد علي بعضه وبيشوف هل ال record ده نفسه متكرر ولا لا

عاوز اشوف الأسماء اللي في كل قسم مثلا مر بسمح بالتكرار ومره لا

هلاقي فيه قسم فيه واحد اسمه احمد مثلا هل فيه حد تاني في نفس القسم اسمه احمد؟ لو فيه هيشيله

```sql
SELECT FirstName,DepartmentID FROM Employees;

SELECT DISTINCT FirstName,DepartmentID FROM Employees;
```

	FirstName	DepartmentID
696	Wesley	7
697	William	4
698	William	5
699	Willow	1
700	Willow	2
701	Willow	3
702	Willow	4
703	Willow	5
704	Willow	7
705	Wyatt	1
706	Wyatt	4
707	Wyatt	6
708	Wyatt	7
709	Xavier	3
710	Zoe	2
711	Zoey	1
712	Zoey	3
713	Zoey	4

Messages

	FirstName	DepartmentID
983	Alexander	7
984	Logan	5
985	Henry	7
986	Delilah	4
987	Caroline	6
988	Jack	2
989	Luna	4
990	John	1
991	Charlotte	7
992	Miles	7
993	Violet	3
994	Isaac	6
995	Ian	1
996	Melody	4
997	Eliza	6
998	Layla	7
999	Thomas	1
1000	Willow	7

```sql
USE HR_Database;

SELECT DepartmentID FROM Employees;

SELECT DISTINCT DepartmentID FROM Employees;

SELECT FirstName FROM Employees;

SELECT DISTINCT FirstName FROM Employees;

SELECT FirstName,DepartmentID FROM Employees;

SELECT DISTINCT FirstName,DepartmentID FROM Employees;
```

The SQL SELECT DISTINCT Statement

The `SELECT DISTINCT` statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT Syntax

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

MYSQL

```
Select DepartmentID from Employees;

Select Distinct  DepartmentID from Employees;



Select FirstName from Employees;

Select Distinct  FirstName from Employees;


Select FirstName, DepartmentID from Employees;

Select distinct FirstName, DepartmentID from Employees;
```

Where Statement + AND , OR, NOT

هنا بيتكلم عن اضافه شروط لأمر ال select

مثلا عايز اجيب كل الموظفين الاناث

```
USE HR_Database;

SELECT * FROM Employees
WHERE Gendor='F';
```

	ID	FirstName	LastNam	Gendor	ateOfBirth	CountryID	DepartmentID	HireDate	ExitDate	MonthlySalary	BonusPerc
1	288	Harper	Castillo	F	996-01-21 00:00:00	1	4	2018-04-07 00:00:00	NULL	922.00	0
2	289	Harper	Dominguez	F	964-12-19 00:00:00	1	1	2005-06-18 00:00:00	NULL	535.00	0.24
3	291	Jade	Hu	F	997-04-17 00:00:00	2	2	2009-06-27 00:00:00	NULL	395.00	0
4	293	Gianna	Holmes	F	994-09-10 00:00:00	1	4	2011-09-09 00:00:00	NULL	694.00	0
5	296	Bella	Wu	F	995-02-09 00:00:00	1	6	2014-08-03 00:00:00	NULL	794.00	0
6	300	Luna	Lu	F	998-01-15 00:00:00	1	4	1997-07-26 00:00:00	NULL	1929.00	0
7	301	Bella	Tran	F	967-06-01 00:00:00	2	1	2010-08-05 00:00:00	NULL	2555.00	0.33
8	302	Ivy	Chau	F	988-01-04 00:00:00	2	7	2019-03-03 00:00:00	NULL	465.00	0
9	304	Sophia	Gutierrez	F	971-09-28 00:00:00	1	2	2009-02-08 00:00:00	NULL	2081.00	0.06
10	306	Lillian	Lewis	F	989-01-05 00:00:00	1	4	2013-08-14 00:00:00	2019-03-31 00:00:00	2790.00	0
11	307	Serenity	Cao	F	992-07-08 00:00:00	2	7	2018-10-21 00:00:00	NULL	2684.00	0
12	312	Ivy	Thompson	F	964-04-15 00:00:00	1	3	2004-08-11 00:00:00	NULL	2811.00	0.05
13	313	Peyton	Wright	F	960-02-09 00:00:00	1	3	2017-05-13 00:00:00	NULL	2066.00	0.1
14	315	Ruby	Alexander	F	994-03-07 00:00:00	1	6	2001-08-13 00:00:00	NULL	2982.00	0.36
15	318	Liliana	Chang	F	978-08-19 00:00:00	2	4	2018-01-11 00:00:00	NULL	2043.00	0
16	321	Amelia	Dominguez	F	973-08-17 00:00:00	3	2	2015-09-23 00:00:00	2018-07-27 00:00:00	1229.00	0.15
17	324	Chloe	Chin	F	983-01-07 00:00:00	1	6	2021-11-13 00:00:00	NULL	2872.00	0.1
18	325	Ella	Martinez	F	988-06-19 00:00:00	1	6	2012-04-06 00:00:00	NULL	1465.00	0

عايزين الموظفين اللي مرتباتهم اقل من 500

```
SELECT * FROM Employees
WHERE MonthlySalary<500;
```

	ID	FirstName	LastName	Gendor	DateOfBirth	CountryID	DepartmentID	HireDate	ExitDate	MonthlySalary	BonusPerc
1	287	Cameron	Lo	M	1994-05-26 00:00:00	2	4	2019-03-24 00:00:00	NULL	372.00	0
2	291	Jade	Hu	F	1997-04-17 00:00:00	2	2	2009-06-27 00:00:00	NULL	395.00	0
3	302	Ivy	Chau	F	1988-01-04 00:00:00	2	7	2019-03-03 00:00:00	NULL	465.00	0
4	326	Gianna	Jones	F	1965-12-26 00:00:00	1	2	1998-09-26 00:00:00	2016-11-02 00:00:00	362.00	0.09
5	343	Jacob	Cheng	M	1994-07-09 00:00:00	1	7	1999-11-26 00:00:00	2003-11-27 00:00:00	330.00	0.17
6	359	Axel	Patel	M	1966-07-20 00:00:00	1	7	2022-09-13 00:00:00	NULL	213.00	0
7	362	Mia	Grant	F	1982-02-10 00:00:00	1	4	2007-02-16 00:00:00	NULL	453.00	0
8	381	Connor	Lai	M	1986-03-13 00:00:00	1	1	2019-09-22 00:00:00	NULL	336.00	0.12
9	404	Sadie	Singh	F	1960-08-22 00:00:00	1	1	2022-03-16 00:00:00	NULL	481.00	0.26
10	407	Logan	Young	M	1961-07-27 00:00:00	1	4	2010-08-16 00:00:00	NULL	459.00	0.33
11	410	Emily	Hong	F	1981-09-19 00:00:00	2	4	2021-01-09 00:00:00	NULL	383.00	0
12	411	Luke	Ramos	M	1997-01-22 00:00:00	3	7	2018-07-28 00:00:00	NULL	235.00	0
13	414	Lillian	Cheng	F	1971-01-27 00:00:00	2	6	1996-08-20 00:00:00	NULL	395.00	0
14	427	Raelynn	Espinoza	F	1980-06-22 00:00:00	3	2	2002-09-12 00:00:00	NULL	396.00	0
15	430	Riley	King	F	1962-12-31 00:00:00	1	3	2013-08-10 00:00:00	NULL	404.00	0.15
16	432	Caleb	Jones	M	1980-10-14 00:00:00	1	3	1996-07-10 00:00:00	NULL	264.00	0.08
17	449	Addison	Lo	F	1973-06-18 00:00:00	1	4	2020-08-28 00:00:00	NULL	367.00	0
18	454	Mia	Jiang	F	1972-10-26 00:00:00	1	2	2015-06-19 00:00:00	NULL	389.00	0.38

لو عايز اجيب الموظفين اللي مرتباتهم اكبر من 500

هنا بيقولك ممكن تستخدم NOT وهيا بتنفي الشرط

```
SELECT * FROM Employees
WHERE MonthlySalary>500;
```

```
SELECT * FROM Employees
WHERE NOT MonthlySalary<=500;
```

Results Messages

	ID	FirstName	LastName	Gendor	DateOfBirth	CountryID	DepartmentID	HireDate	ExitDate	MonthlySalary	BonusPerc
1	285	Kai	Le	M	1972-09-20 00:00:00	1	1	2022-02-05 00:00:00	NULL	709.00	0
2	286	Robert	Patel	M	1981-06-10 00:00:00	1	7	2013-10-23 00:00:00	NULL	2708.00	0
3	288	Harper	Castillo	F	1996-01-21 00:00:00	1	4	2018-04-07 00:00:00	NULL	922.00	0
4	289	Harper	Dominguez	F	1964-12-19 00:00:00	1	1	2005-06-18 00:00:00	NULL	535.00	0.24
5	290	Ezra	Vu	M	1994-02-12 00:00:00	1	4	2004-04-22 00:00:00	2014-02-14 00:00:00	2701.00	0
6	292	Miles	Chang	M	1987-06-14 00:00:00	2	6	1999-02-19 00:00:00	NULL	1415.00	0
7	293	Gianna	Holmes	F	1994-09-10 00:00:00	1	4	2011-09-09 00:00:00	NULL	694.00	0
8	294	Jameson	Thomas	M	1962-12-23 00:00:00	1	6	2015-02-05 00:00:00	NULL	2377.00	0.1
9	295	Jameson	Pena	M	1979-01-07 00:00:00	1	4	2003-10-12 00:00:00	NULL	1258.00	0
10	296	Bella	Wu	F	1995-02-09 00:00:00	1	6	2014-08-03 00:00:00	NULL	794.00	0
11	297	Jose	Wong	M	1990-07-03 00:00:00	2	4	2017-11-15 00:00:00	NULL	2374.00	0.23
12	298	Lucas	Richardson	M	1983-10-04 00:00:00	1	3	2018-07-22 00:00:00	NULL	1751.00	0.08
13	299	Jacob	Moore	M	1977-12-29 00:00:00	1	3	2021-03-24 00:00:00	NULL	659.00	0.15
14	300	Luna	Lu	F	1998-01-15 00:00:00	1	4	1997-07-26 00:00:00	NULL	1929.00	0
15	301	Bella	Tran	F	1967-06-01 00:00:00	2	1	2010-08-05 00:00:00	NULL	2555.00	0.33
16	303	Jordan	Kumar	M	1992-10-31 00:00:00	1	4	2017-11-11 00:00:00	NULL	1085.00	0
17	304	Sophia	Gutierrez	F	1971-09-28 00:00:00	1	2	2009-02-08 00:00:00	NULL	2081.00	0.06
18	305	Eli	Dang	M	1992-11-08 00:00:00	1	2	2015-11-16 00:00:00	NULL	654.00	0.12

طيب عاوزين كل الموظفين الانااث اللي مرتباتهم اقل من 500 هنا ممكن استخدم AND

```
SELECT * FROM Employees
WHERE MonthlySalary<500 AND Gendor='F';
```

	ID	FirstName	LastName	Gendor	DateOfBirth	CountryID	DepartmentID	HireDate	ExitDate	MonthlySalary	BonusPerc
1	291	Jade	Hu	F	1997-04-17 00:00:00	2	2	2009-06-27 00:00:00	NULL	395.00	0
2	302	Ivy	Chau	F	1988-01-04 00:00:00	2	7	2019-03-03 00:00:00	NULL	465.00	0
3	326	Gianna	Jones	F	1965-12-26 00:00:00	1	2	1998-09-26 00:00:00	2016-11-02 00:00:00	362.00	0.09
4	362	Mia	Grant	F	1982-02-10 00:00:00	1	4	2007-02-16 00:00:00	NULL	453.00	0
5	404	Sadie	Singh	F	1960-08-22 00:00:00	1	1	2022-03-16 00:00:00	NULL	481.00	0.26
6	410	Emily	Hong	F	1981-09-19 00:00:00	2	4	2021-01-09 00:00:00	NULL	383.00	0
7	414	Lillian	Cheng	F	1971-01-27 00:00:00	2	6	1996-08-20 00:00:00	NULL	395.00	0
8	427	Raelynn	Espinoza	F	1980-06-22 00:00:00	3	2	2002-09-12 00:00:00	NULL	396.00	0
9	430	Riley	King	F	1962-12-31 00:00:00	1	3	2013-08-10 00:00:00	NULL	404.00	0.15
10	449	Addison	Lo	F	1973-06-18 00:00:00	1	4	2020-08-28 00:00:00	NULL	367.00	0
11	454	Mia	Jiang	F	1972-10-26 00:00:00	1	2	2015-06-19 00:00:00	NULL	389.00	0.38
12	455	Skylar	Parker	F	1987-01-21 00:00:00	1	4	2018-08-05 00:00:00	NULL	330.00	0
13	492	Layla	Nunez	F	1993-09-07 00:00:00	1	6	2017-09-10 00:00:00	NULL	220.00	0.05
14	534	Genesis	Herrera	F	1987-05-30 00:00:00	3	4	2015-10-03 00:00:00	NULL	331.00	0.1
15	537	Olivia	Mendoza	F	1974-02-05 00:00:00	1	7	2017-05-07 00:00:00	NULL	473.00	0
16	538	Skylar	Xu	F	1986-05-13 00:00:00	2	1	1997-10-16 00:00:00	NULL	263.00	0
17	539	Eloise	Williams	F	1971-09-21 00:00:00	1	7	2019-02-12 00:00:00	NULL	316.00	0.36
18	546	Luna	Lu	F	1965-02-02 00:00:00	2	3	2007-03-06 00:00:00	NULL	201.00	0

عاوز كل الناس اللي من البلد اللي رقمها 1

```
SELECT * FROM Employees
WHERE CountryID =1;
```

عاوز كل الناس اللي مش من البلد اللي رقمها 1

```
SELECT * FROM Employees
WHERE NOT CountryID =1;
```

```
SELECT * FROM Employees
WHERE CountryID <>1;
```

عاوز كل الناس اللي من البلد اللي رقمها ب 1 والنوع ذكر

```
SELECT * FROM Employees
WHERE CountryID =1 AND Gendor='M';
```

عاوز كل الموظفين اللي في القسم رقم 1 ورقم 2 هنا نستخدم OR

```
SELECT * FROM Employees
WHERE CountryID =1 OR CountryID=2;
```

لو استخدمت AND بدل OR مش هيرجعلك حاجه لانه علاقة الموظفين بالقسم هيا علاقه

MANY TO ONE يعني عمرك ماهتلاقي موظف شغال في قسمين مختلفين لكن العكس ممكن انه القسم فيه اكتر من موظف

عاوز كل الناس اللي لسه شغالين معانا هنستخدم IS NULL

```
SELECT * FROM Employees
WHERE ExitDate IS NULL;
```

عاوز كل الناس اللي مشيت من الشركه

```sql
SELECT * FROM Employees
WHERE NOT ExitDate IS NULL;
```

```sql
SELECT * FROM Employees
WHERE ExitDate IS NOT NULL;
```

The SQL WHERE Clause

The `WHERE` clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

WHERE Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Note: The `WHERE` clause is not only used in `SELECT` statements, it is also used in `UPDATE`, `DELETE`, etc.!

The SQL AND, OR and NOT Operators

The `WHERE` clause can be combined with `AND`, `OR`, and `NOT` operators.

The `AND` and `OR` operators are used to filter records based on more than one condition:

- The `AND` operator displays a record if all the conditions separated by `AND` are TRUE.
- The `OR` operator displays a record if any of the conditions separated by `OR` is TRUE.

The `NOT` operator displays a record if the condition(s) is NOT TRUE.

AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

```
Select * from Employees
where Gendor='F';

Select * from Employees
where  MonthlySalary<=500;


Select * from Employees
where  MonthlySalary>500;

Select * from Employees
where Not MonthlySalary<=500;


Select * from Employees
where MonthlySalary<500 and Gendor='F';

select * from Employees
where CountryID=1;


select * from Employees
where Not CountryID=1;


select * from Employees
where  CountryID <> 1;


select * from Employees
where DepartmentID=1;

select * from Employees
where DepartmentID=1 and Gendor='M';


select * from Employees
where DepartmentID=1 Or DepartmentID=2;

select * from Employees
where DepartmentID=1 AND DepartmentID=2;


Select * from Employees
where ExitDate is Null;


Select * from Employees
```

```
where ExitDate  is Not Null;
```

"In" Operator

هنا بيقولك بل ماتفضل كل شوية تكتب OR لما تكون عاوز مثلا تجيب الموظفين اللي في اكتر من قسم بتكتب IN وتفتح قوسين وتكتب كل القيم اللي انت عاوزها

```
select * from Employees
where DepartmentID=1 Or DepartmentID=2 or DepartmentID=5 or
DepartmentID=7;
```

```
select * from Employees
where DepartmentID IN(1,2,5,7);
```

لو عايز اجيب كل الموظفين اللي اساميهم يعقوب وبروك وهاربر

```
SELECT * FROM Employees
WHERE  FirstName IN('Jacob','Brooks','Harper');
```

عايز اجيب ارقام الأقسام اللي فيها موظفين بيقبضوا اقل من او يساوي 210

```
SELECT DepartmentID FROM Employees
WHERE MonthlySalary<=210;
```

عايز اجيب اسامي الأقسام اللي فيها موظفين بيقبضوا اقل من او يساوي 210

هنا بيقولك انك ممكن تحط جملة SQL في ال IN

يعني هجيب كل اسامي الأقسام واحط شرط الشرط ده هيكون IN وهيكون جوا ال IN دي QUERY تانيه تجيبلي ال ID بتاع الأقسام اللي فيها موظفين بيقبضوا اقل من 210

```
SELECT Name FROM Departments
WHERE ID IN(SELECT DepartmentID FROM Employees WHERE
MonthlySalary<=210);
```

عايز اسامي الأقسام اللي فيها موظفين بيقبضوا اعلي من210

```
SELECT Name FROM Departments
WHERE ID NOT IN(SELECT DepartmentID FROM Employees WHERE
MonthlySalary<=210);
```

The SQL IN Operator

The `IN` operator allows you to specify multiple values in a `WHERE` clause.

The `IN` operator is a shorthand for multiple `OR` conditions.

IN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

or:

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

```sql
select * from Employees
where DepartmentID=1 Or DepartmentID=2;


select * from Employees
where DepartmentID=1 Or DepartmentID=2 or DepartmentID=7;


select * from Employees
where DepartmentID=1 Or DepartmentID=2 or DepartmentID=5 or DepartmentID=7;


select * from Employees
where DepartmentID in (1,2,5,7);


select * from Employees
where FirstName in ('Jacob','Brooks','Harper');


select Departments.Name from Departments
where
ID in ( select DepartmentID from Employees where MonthlySalary <=210 );


select Departments.Name from Departments
where
ID not in ( select DepartmentID from Employees where MonthlySalary <=210 );
```

Sorting : Order By

لما بترجع داتا من جدول بترجعلك حسب ماهيا متخزنه ولو عايز ترتبها بتيجي في اخر جملة ال SQL وبتكتب ORDER BY وبعدها اسم العمود وبعدها ASC لو تصاعدي او DESC لو تنازلي

ال DEFAULT بتاعه انه يكون تصاعدي يعني مش لازم تكتب ASC

عاوز ارجع ال ID والاسم الأول والمرتب من القسم رقم 1 وارتبهم من حيث المرتب الأقل للاعلي

```
SELECT ID,FirstName,MonthlySalary FROM Employees
Where DepartmentID=1
ORDER BY MonthlySalary;
```

```
SELECT ID,FirstName,MonthlySalary FROM Employees
Where DepartmentID=1
ORDER BY MonthlySalary ASC;
```

عاوزه تنازليا

```
SELECT ID,FirstName,MonthlySalary FROM Employees
Where DepartmentID=1
ORDER BY MonthlySalary DESC;
```

عاوز ارتب تصاعديا حسب الاسم الأول والمرتب هنا هوا بيرتب الاسامي حسب الحروف الأول وبيشوف لو فيه اسامي متكرره بيحط الاسم اللي مرتبه اقل الاول

```
SELECT ID,FirstName,MonthlySalary FROM Employees
Where DepartmentID=1
ORDER BY FirstName, MonthlySalary ;
```

طيب لو عايز ارتبهم حسب الاسم تصاعديا وحسب المرتب تنازليا هنا هيرتب الاسامي تصاعديا عادي بس لو لقي اسم مكرر هيحط اللي مرتبه اعلي الأول

```
SELECT ID,FirstName,MonthlySalary FROM Employees
Where DepartmentID=1
ORDER BY FirstName ASC, MonthlySalary DESC;
```

The SQL ORDER BY Keyword

The `ORDER BY` keyword is used to sort the result-set in ascending or descending order.

The `ORDER BY` keyword sorts the records in ascending order by default. To sort the records in descending order, use the `DESC` keyword.

ORDER BY Syntax

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

```
select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
```

```sql
select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By FirstName ;

select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By FirstName ASC;

select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By FirstName desc;

select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By MonthlySalary ;


select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By MonthlySalary Asc;


select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By MonthlySalary Desc;



select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By FirstName , MonthlySalary ;


select ID, FirstName,MonthlySalary from Employees
where DepartmentID=1
Order By FirstName ASC, MonthlySalary Desc;
```

Select Top Statement

كلمة TOP لو كتبتها بعد كلمة SELECT وكتبت بعدها رقم وليكن 10 مثلا هيجيبلك اول 10 RECORDS من الجدول

لو كتبت بعدها كلمة PERCENT هيجيبلي اول 10% من ال RECORDS يعني لو عندي 100 سجل هيجيبلي اول 10 منهم

عاوز اول خمس موظفين

```sql
SELECT TOP 5 * FROM Employees;
```

عاوز اول 5% من الموظفين

```sql
SELECT TOP 5 PERCENT *  FROM Employees;
```

عاوز أسماء الموظفين اللي واخدين اعلي 3 مرتبات

امشي من الاخر للأول عشان توصل

عاوزين الأول نجيب المرتبات غير مكرره ومرتبه تنازليا

```sql
SELECT DISTINCT MonthlySalary FROM Employees
ORDER BY MonthlySalary DESC
```

هاتلي اول 3 منهم

```sql
SELECT DISTINCT TOP 3 MonthlySalary FROM Employees
ORDER BY MonthlySalary DESC
```

هاتلي اسامي الموظفين بقي

```sql
SELECT ID,FirstName  FROM Employees
WHERE MonthlySalary IN(
SELECT DISTINCT TOP 3 MonthlySalary FROM Employees
ORDER BY MonthlySalary DESC);
```

عاوز اللي واخدين اقل 3 مرتبات

```sql
SELECT ID,FirstName  FROM Employees
WHERE MonthlySalary IN(
SELECT DISTINCT TOP 3 MonthlySalary FROM Employees
ORDER BY MonthlySalary ASC);
```

The SQL SELECT TOP Clause

The `SELECT TOP` clause is used to specify the number of records to return.

The `SELECT TOP` clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

Note: Not all database systems support the `SELECT TOP` clause. MySQL supports the `LIMIT` clause to select a limited number of records, while Oracle uses `FETCH FIRST n ROWS ONLY` and `ROWNUM`.

SQL Server / MS Access Syntax:

```
SELECT TOP number|percent column_name(s)
FROM table_name
WHERE condition;
```

```sql
Select * from Employees;


-- This will show top 5 employees.
Select top 5 * from Employees;


-- This will show top 10% of the data.
```

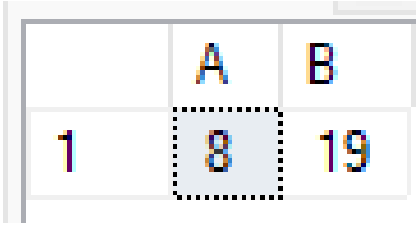
```sql
select top 10 percent * from Employees;

-- this will show the all salaries ordered from the heighest to lowest.
select  MonthlySalary from employees
order by MonthlySalary Desc;



-- this will show the all salaries ordered from the heighest to lowest without redundancy.
select distinct MonthlySalary from employees
order by MonthlySalary Desc;

-- this will show the heighest 3 salaries.
select  distinct  top 3 MonthlySalary from employees
order by MonthlySalary Desc;


--This will show all employees who takes one of the heighest 3 salaries.

select ID , FirstName, MonthlySalary from Employees where MonthlySalary In
(
	select  distinct  top 3 MonthlySalary from employees
	order by MonthlySalary Desc
)

Order By MonthlySalary Desc



--This will show all employees who takes one of the Lowest 3 salaries.
select ID , FirstName, MonthlySalary from Employees where MonthlySalary In
(
	select  distinct  top 3 MonthlySalary from employees
	order by MonthlySalary ASC
)

Order By MonthlySalary ASC
```

Select As

هنا بيقولك انك ممكن بد ماتكتب SELECT تعمل عمليات رياضيه

```sql
SELECT A=5+3,B=98/5;
```

طلعلك عمودين بالاسامي اللي انا حطيتها وحط تحت كل واحد الناتج بتاعه



	A	B
1	8	19

ولو كتبت FROM Employees هيككر النتيجه بعدد ال records اللي في جدول الموظفين بدون سبب

```sql
SELECT A=5+3,B=98/5 FROM Employees;
```

عاوز جدول باسامي الموظفين ومرتباتهم مقسومه علي 2

```sql
SELECT ID,FirstName,A=MonthlySalary/2 FROM Employees;
```

عاوز اجيب ال id وال full name بتوع كل موظف

```
SELECT ID,FullName=FirstName+' '+LastName FROM Employees;
```

بيقولك انه ممكن تحط ال EXPRESSION او المعادله الأول وبعدين تكتب AS وبعدها تسمي العمود

```
SELECT ID,FirstName+' '+LastName AS FullName FROM Employees;
```

عاوز ال id والاسم الأول والمرتب الشهري والسنوي

```
SELECT ID,FirstName,MonthlySalary,YearlySalary=MonthlySalary*12
FROM Employees;
```

عاوز ازود البونص كقيمه

```
SELECT ID,FirstName,MonthlySalary,
YearlySalary=MonthlySalary*12,
BonusAmount=MonthlySalary*BonusPerc
 FROM Employees;
```

فيه function في الsql server اسمها DATEDIFF بتاخد 3 متغيرات عاوز النتيجه تكون ازاي والتاريخ الأول والتاريخ التاني وبتطرح التاريخين وبتطلعهولك بال FORMAT اللي انت حددته

لو كتبت ال FORMAT خليتها YEAR هتطلعهولك بالسنين

عاوزين نجيب ال ID وال FULL NAME وال AGE

```
SELECT ID,
FullName=FirstName+' '+LastName,
Age=DATEDIFF(YEAR,DateOfBirth,GETDATE())
FROM Employees;
```

عاوزين تاريخ النهارده

```
SELECT NOW=GETDATE();
```

SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of that query.

An alias is created with the `AS` keyword.

Alias Column Syntax

```
SELECT column_name AS alias_name
FROM table_name;
```

```
Select A= 5 * 4 , B= 6/2


Select A= 5 * 4 , B= 6/2
from employees


Select ID, FirstName,  A = MonthlySalary/2
from employees


Select ID, FirstName + ' ' + LastName as FullName From Employees;


Select ID, FullName = FirstName + ' ' + LastName  From Employees;


select ID, FirstName , MonthlySalary , YealySalary = MonthlySalary * 12  from employees;


select ID, FirstName , MonthlySalary , YealySalary =MonthlySalary* 12 , BonusAmount= MonthlySalary *
BonusPerc  from employees;


select Today = getDate()


select ID, FullName= FirstName + ' ' + LastName, Age = DATEDIFF(Year , DateOfBirth ,getDate()) from
Employees;
```

Between Operator

كلمة BETWEEN بتحط بعدها قيمه صغري وبعدين AND وبعدين قيمه كبري وبيرجعلك كل ال RECORDS اللي بين القيمتين دول تقدر تستخدمها علي الأرقام والتواريخ وال STRING

عاوز اجيب الموظفين اللي مرتباتهم بين ال 500 وال 1000

```
SELECT * FROM Employees
WHERE MonthlySalary >=500 AND MonthlySalary<=1000;
```

```sql
SELECT * FROM Employees
WHERE MonthlySalary BETWEEN 500 AND 1000;
```

The SQL BETWEEN Operator

The `BETWEEN` operator selects values within a given range. The values can be numbers, text, or dates.

The `BETWEEN` operator is inclusive: begin and end values are included.

BETWEEN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

MYSQL

```sql
Select * from Employees where
(MonthlySalary >=500 and MonthlySalary <=1000)

Select * from Employees where
MonthlySalary Between 500 and 1000;
```

Count, Sum, Avg, Min, Max Functions

هما FUNCTIONS بتديهم عمود معين وبيرجعولك القيمه

تعالي نجربهم علي عمود المرتبات

```sql
SELECT
TotalCount=Count(MonthlySalary),
TotalSum=Sum(MonthlySalary),
Average=AVG(MonthlySalary),
MinimumSalary=MIN(MonthlySalary),
MaximumSalary=MAX(MonthlySalary)
FROM Employees
```

احسبهم علي القسم رقم 1

```sql
SELECT
TotalCount=Count(MonthlySalary),
TotalSum=Sum(MonthlySalary),
```

```
Average=AVG(MonthlySalary),
MinimumSalary=MIN(MonthlySalary),
MaximumSalary=MAX(MonthlySalary)
FROM Employees

WHERE DepartmentID=1;
```

ال COUNT بتعد الRECORDS اللي العمود اللي انت مختاره فيه قيم يعني مش بتعد ال NULL

هاتلي اجمالي عدد الموظفين.

```
SELECT TotalEmployees=Count(ID)FROM Employees;
```

هاتلي عدد الموظفين اللي مشيوا

```
SELECT ResignedEmployees=Count(ExitDate)FROM Employees;
```

ال average برضه مابتاخدش القيم اللي ب null

The SQL COUNT(), AVG() and SUM() Functions

The `COUNT()` function returns the number of rows that matches a specified criterion.

COUNT() Syntax

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

The `AVG()` function returns the average value of a numeric column.

AVG() Syntax

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

The `SUM()` function returns the total sum of a numeric column.

SUM() Syntax

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

The SQL MIN() and MAX() Functions

The `MIN()` function returns the smallest value of the selected column.

The `MAX()` function returns the largest value of the selected column.

MIN() Syntax

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

MAX() Syntax

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

```
select TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees;



select  TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees where DepartmentID=1


select * from employees;

select TotalEmployees = count (ID) from Employees;

--count function only counts the not null values.
select ResignedEmployees= count(ExitDate)  from employees;
```

Group By

Group by بنستخدمها في اخراج تقرير بالاجماليات والعمود اللي بتاخده بيتم التجميع علي أساسه

عاوزين الدوال بتاعت الدرس اللي فات نعملها علي مستوي الأقسام

```
SELECT DepartmentID,
TotalCount=Count(MonthlySalary),
TotalSum=Sum(MonthlySalary),
Average=AVG(MonthlySalary),
```

```
MinimumSalary=MIN(MonthlySalary),
MaximumSalary=MAX(MonthlySalary)
FROM Employees

GROUP BY DepartmentID
ORDER BY DepartmentID;
```

The SQL GROUP BY Statement

The `GROUP BY` statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The `GROUP BY` statement is often used with aggregate functions (`COUNT()`, `MAX()`, `MIN()`, `SUM()`, `AVG()`) to group the result-set by one or more columns.

GROUP BY Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

```
select TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees;


select  TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees where DepartmentID=3


select DepartmentID, TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees
       Group By DepartmentID
          order by DepartmentID
```

Having

ال HAVING بتعمل فلتر للنتايج للي طالعه من ال GROUP BY وبتتكتب بعد ال GROUP BY

عندنا ال QUERY دي

```
SELECT DepartmentID,
TotalCount=Count(MonthlySalary),
TotalSum=Sum(MonthlySalary),
Average=AVG(MonthlySalary),
MinimumSalary=MIN(MonthlySalary),
MaximumSalary=MAX(MonthlySalary)
FROM Employees

GROUP BY DepartmentID
ORDER BY DepartmentID;
```

عاوزين نفلتر عالاقسام اللي فيها اكبر من 100 موظف

```
SELECT DepartmentID,
TotalCount=Count(MonthlySalary),
TotalSum=Sum(MonthlySalary),
Average=AVG(MonthlySalary),
MinimumSalary=MIN(MonthlySalary),
MaximumSalary=MAX(MonthlySalary)
FROM Employees

GROUP BY DepartmentID
HAVING Count(MonthlySalary)>100
ORDER BY DepartmentID;
```

لو عايز اعمل نفس الشغل من غير having بحط الكود في sub query وبديها اسم الاسم ده هيكون اسم الجدول اللي هيطلع وبعدين استخدم where عادي

```
select * from(
SELECT DepartmentID,
TotalCount=Count(MonthlySalary),
TotalSum=Sum(MonthlySalary),
Average=AVG(MonthlySalary),
MinimumSalary=MIN(MonthlySalary),
MaximumSalary=MAX(MonthlySalary)
FROM Employees

GROUP BY DepartmentID)R2

where R2.TotalCount>100;
```

The SQL HAVING Clause

The `HAVING` clause was added to SQL because the `WHERE` keyword cannot be used with aggregate functions in a direct way.

HAVING Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

```
select DepartmentID, TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees

       Group By DepartmentID

          order by DepartmentID


--Having is the where satement for group by
select DepartmentID, TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees
       Group By DepartmentID
         having Count(MonthlySalary) > 100



-- Same solution without having :-)
select * from
(

   select DepartmentID, TotalCount=Count(MonthlySalary),
          TotalSum=Sum(MonthlySalary),
          Average=Avg(MonthlySalary),
          MinSalary=Min(MonthlySalary),
          MaxSalary=Max(MonthlySalary)

          from Employees

       Group By DepartmentID

) R1

where R1.TotalCount> 100;
```

Like

ال like بتخليك تعمل بحث علي ال records اللي فيها قيمه شبه القيمة اللي انت دخلتها كانك تبحث عن اسم شخص ومش عارف مكتوب ازاي مثلا فبتكتب حرف او جزء من الكلمة وهوا بيجيلك كل الاسامي اللي بتحتوي علي المقطع ده

طريقتها انك مكان الجزء اللي مش عارفه بتكتب %

عاوز كل الموظفين اللي بيبدأ اساميهم بحرف a

```
SELECT *FROM Employees
where FirstName like 'a%';
```

عاوز اللي بينتهي بحرف ال a

```
SELECT *FROM Employees
where FirstName like '%a';
```

عاوز كل الموظفين اللي اساميهم فيه المقطع tell

```
SELECT *FROM Employees
where FirstName like '%tell%';
```

عاوز اللي بيبدا وبينتهي بحرف a

```
SELECT *FROM Employees
where FirstName like 'a%a';
```

عاوز اللي تاني حرف في اسمه بيكون a

```
SELECT *FROM Employees
where FirstName like '_a%';
```

عاوز الحرف الثالث يكون a

```
SELECT *FROM Employees
where FirstName like '___a%';
```

عاوز اول حرف يكون a وعدد الحروف عالاقل يكونوا 3 حروف (ال a من ضمنهم)

```
SELECT *FROM Employees
where FirstName like 'a__%';
```

عالاقل يكونوا 4 حروف

```
SELECT *FROM Employees
where FirstName like 'a___%';
```

عاوزين اللي اساميهم بتبدأ بحرف ال a او ال b

```
SELECT *FROM Employees
where FirstName like 'a%' or FirstName like 'b%';
```

The SQL LIKE Operator

The `LIKE` operator is used in a `WHERE` clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the `LIKE` operator:

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (_) represents one, single character

Note: MS Access uses an asterisk (*) instead of the percent sign (%), and a question mark (?) instead of the underscore (_).

The percent sign and the underscore can also be used in combinations!

LIKE Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

Tip: You can also combine any number of conditions using `AND` or `OR` operators.

Here are some examples showing different `LIKE` operators with '%' and '_' wildcards:

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

```sql
USE HR_Database;


select * from Employees;

--Finds any values that start with "a"
select ID, FirstName from Employees
where FirstName like 'a%';


--Finds any values that end with "a"
select ID, FirstName from Employees
where FirstName like '%a';


--Finds any values that have "tell" in any position
select ID, FirstName from Employees
where FirstName like '%tell%';

--     Finds any values that start with "a" and ends with "a"
select ID, FirstName from Employees
where FirstName like 'a%a';

--Finds any values that have "a" in the second position
select ID, FirstName from Employees
where FirstName like '_a%';

--Finds any values that have "a" in the third position
select ID, FirstName from Employees
where FirstName like '__a%';


--Finds any values that start with "a" and are at least 3 characters in length
select ID, FirstName from Employees
where FirstName like 'a__%';

--Finds any values that start with "a" and are at least 4 characters in length
select ID, FirstName from Employees
where FirstName like 'a___%';



--Finds any values that start with "a"
select ID, FirstName from Employees
where FirstName like 'a%' or FirstName like 'b%' ;
```

WildCards

هنعمل الأول update للجدول (لاحظ انه كاتب محمد مره بال a ومره بال e)

```sql
Update Employees
set FirstName ='Mohammed' , LastName='Abu-Hadhoud'
where ID= 285;


Update Employees
set FirstName ='Mohammad' , LastName='Maher'
where ID= 286;
```

ال wild cards بيتم استخدامهم مع ال like

لو انا عايز اجيب اللي اساميهم محمد وانا عارف انه فيه ناس بتكتبه ad وفيه ناس بتكتبه ed

```sql
select ID, FirstName, LastName from Employees
Where firstName = 'Mohammed' or FirstName ='Mohammad';
```

طالما انا عارف انه الاختلاف ممكن يكون في مكان واحد في الكلمه ممكن اكتب [] واحط بينهم الحرفين اللي شاكك فيهم

```
select ID, FirstName, LastName from Employees
Where firstName like 'Mohamm[ae]d';
```

عاوز كل الناس اللي اساميهم مش محمد

```
select ID, FirstName, LastName from Employees
Where firstName not like 'Mohamm[ae]d';
```

عاوز كل الناس اللي اساميهم بتبدأ بال a او b او c

```
select ID, FirstName, LastName from Employees
Where firstName like 'a%' or firstName like 'b%' or firstName like
'c%';
```

```
select ID, FirstName, LastName from Employees
Where firstName  like '[abc]%';
```

علامة – الطرح بتعبر عن المدي او ال range

عاوز كل اللي بيبدأ من حرف a او b او c لحد حرف ال l

```
select ID, FirstName, LastName from Employees
Where firstName  like '[a-l]%';
```

SQL Wildcard Characters

A wildcard character is used to substitute one or more characters in a string.

Wildcard characters are used with the `LIKE` operator. The `LIKE` operator is used in a `WHERE` clause to search for a specified pattern in a column.

Wildcard Characters in SQL Server

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents any single character within the specified range	c[a-b]t finds cat and cbt

```
--Execute these satatements to update data
Update Employees
set FirstName ='Mohammed' , LastName='Abu-Hadhoud'
where ID= 285;
```

```sql
Update Employees
set FirstName ='Mohammad' , LastName='Maher'
where ID= 286;

--------------------------------

select ID, FirstName, LastName from Employees
Where firstName = 'Mohammed' or FirstName ='Mohammad';


-- will search form Mohammed or Mohammad
select ID, FirstName, LastName from Employees
Where firstName like 'Mohamm[ae]d';

-----------------------------------

--You can use Not
select ID, FirstName, LastName from Employees
Where firstName Not like 'Mohamm[ae]d';

--------------------

select ID, FirstName, LastName from Employees
Where firstName like 'a%' or firstName like 'b%' or firstName like 'c%';


-- search for all employees that their first name start with a or b or c
select ID, FirstName, LastName from Employees
Where firstName like '[abc]%';


--------------------------------
-- search for all employees that their first name start with any letter from a to l
select ID, FirstName, LastName from Employees
Where firstName like '[a-l]%';
--------------------------------
```

Restore Shop Database

https://cdn.fs.teachablecdn.com/7SRuonRdq1Cu2nea0SAS

```sql
RESTORE DATABASE Shop_Database FROM DISK= 'D:\Shop_Database.bak';
USE Shop_Database;
EXEC sp_changedbowner 'sa';
```

(Inner) Join

قبل كده كنا بنجيب الداتا من جدول واحد فلو عايزين نجيب داتا من اكتر من جدول بنستخدم ال JOIN STATEMENT ودي عباره عن جملة بتربط اكتر من جدول ببعضهم

طيب هنا بيقولك انه فيه اربع انوع من ال JOINS :-

1- INNER JOIN :- وده معظم شغلك هيكون عليه وده بيطلع الداتا المشتركه بين الجدولين او ال RECORDS اللي حصل بينهم ربط زي انه يرجعلك العملاء اللي عملوا اوردرات بس

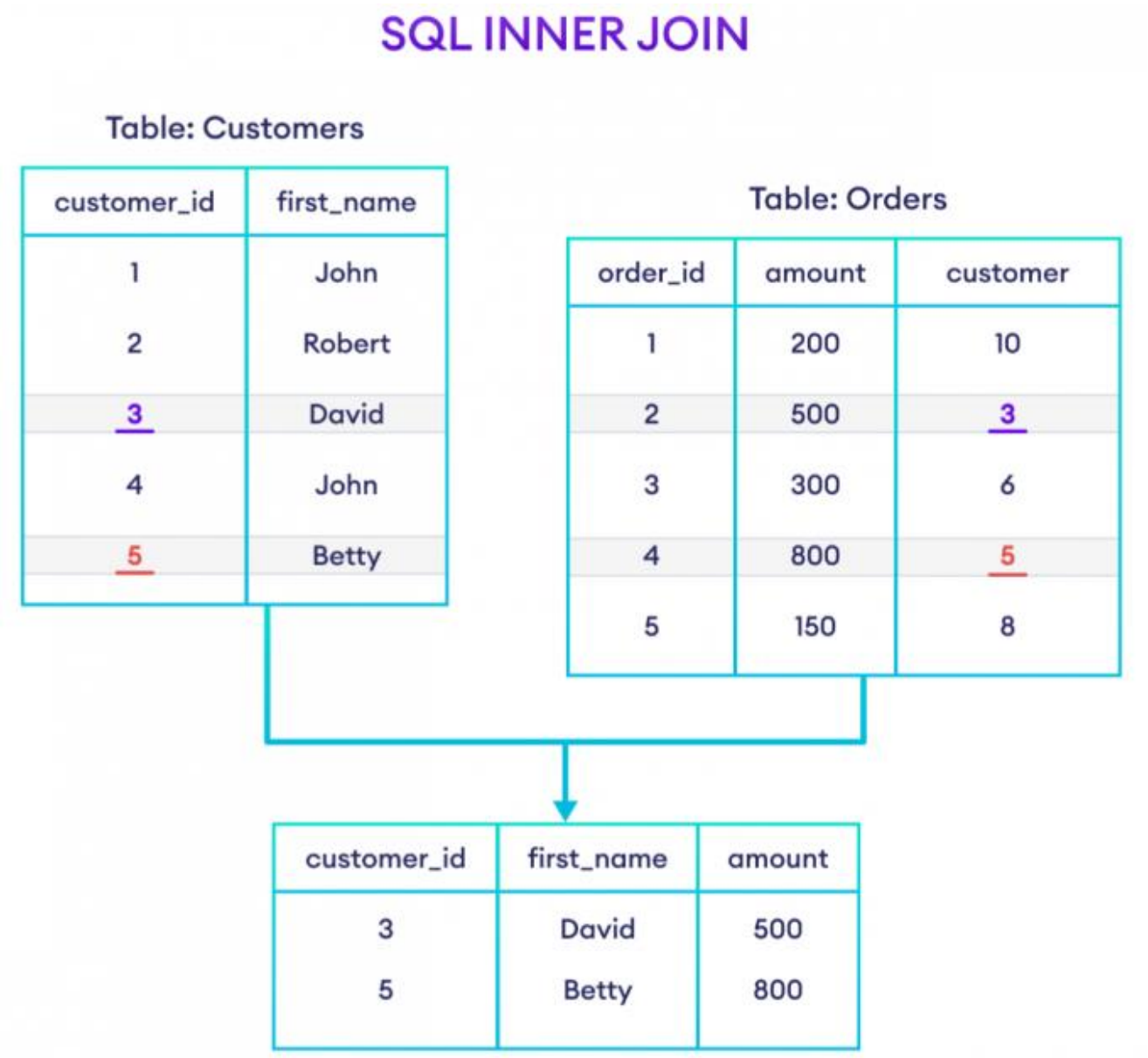
2- LEFT JOIN :- وده بيرجعلك كل الداتا بتاعت الجدول اللي عالشمال بما فيهم الداتا المشتركه من الجدول اللي عاليمين

3- RIGHT JOIN :- بيرجعلك كل الداتا بتاعت الجدول اللي عاليمين ومعاه الداتا المرتبطه بيه من الجدول اللي عالشمال

4- FULL OUTER JOIN :- وده بيجيبلك الداتا بتاعت الجدولين كلهم

هنبدأ في ال inner join وساعات بيقولوا عليها join

بص كده عالجداول دي



الجدول بتاعت ال orders بتلاقي فيه عملاء مش موجودين في جدول ال customers وده مش موجود في الحياه العمليه لانه عادة بتلاقي الجدولين مربوطين ببعض انما هنا الجدولين مالهومش علاقه ببعض

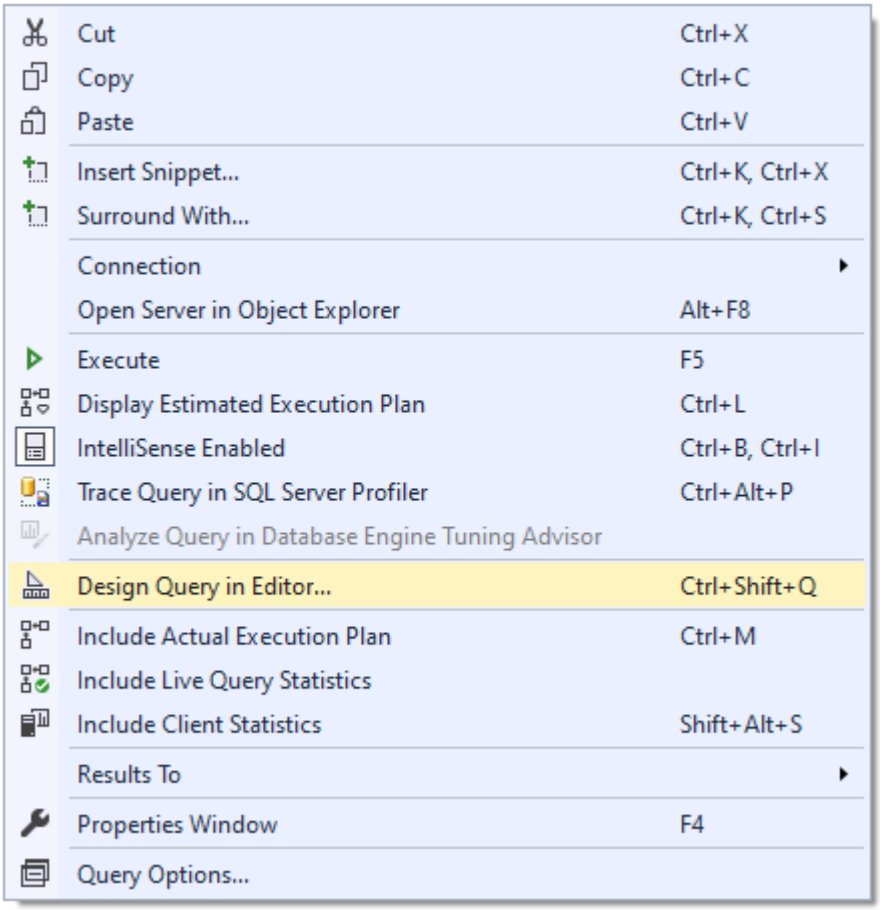
هنا لما جينا نعمل inner join جابلي الداتا المشتركه بس بين الجدولين او الاوردرات اللي ليها عملاء

طريقة كتابة ال select statement هيا انك تكتب اسامي الاعمده اللي انت عاوزها بس تكتب اسم الجدول قبليه عشان تعرف انت هتجيب العمود منين وبعدين from وبعدها اسم الجدول الأول وبعدين inner join واسم الجدول التاني وبعدين on وبعدين الشرط او ال keys اللي بتربط الجدولين ببعض

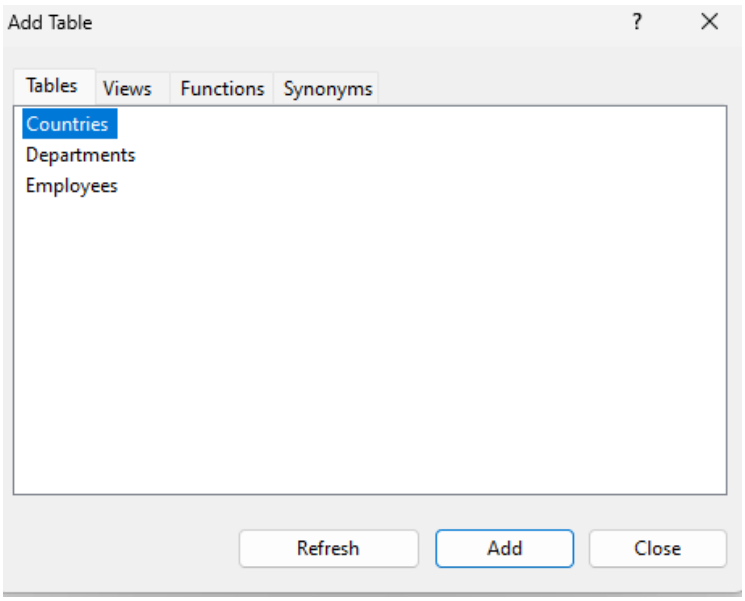
ودي ال select statement بتاعتها

```
SELECT Customers.CustomerID,Customers.Name,Orders.Amount
FROM Customers INNER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID;
```

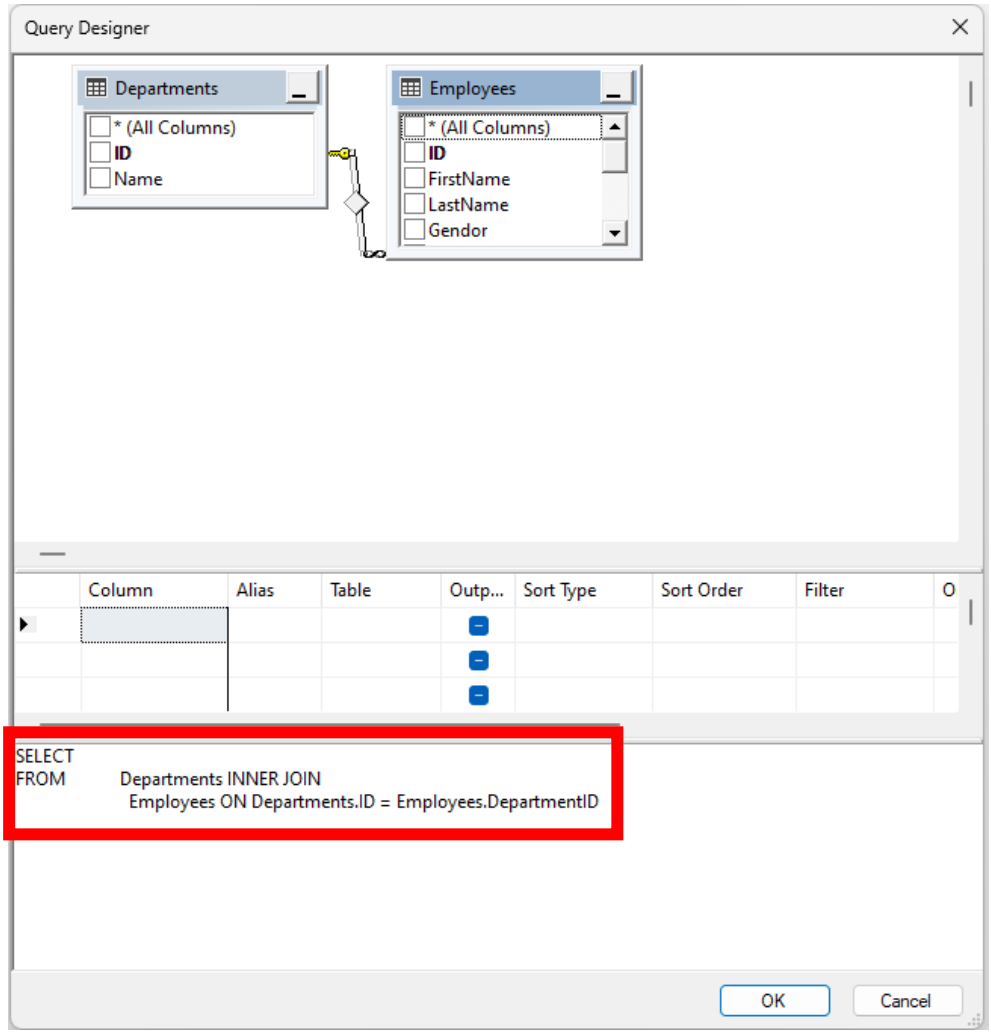
نقدر نعمل ده بالديزاين زي كده هنعملها عالداتا بيز بتاعت ال HR



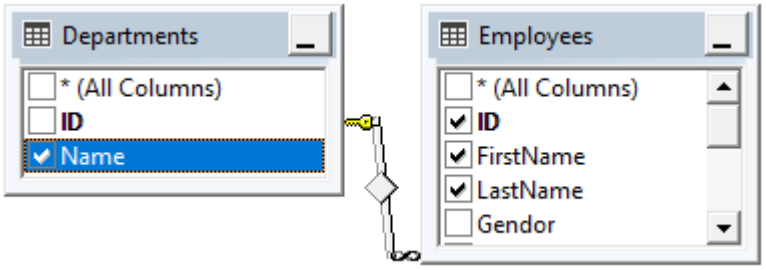
وبعدين هنختار جدول ال EMPLOYEES وجدول ال DEPARTMENT



هلاقيه كتبلي ال SELECT STATEMENT لوحده في الشاشه تحت



هنختار الاعمده اللي عاوزين نعرضها




```
SELECT    Employees.ID, Employees.FirstName, Employees.LastName, Departments.Name
FROM      Departments INNER JOIN
          Employees ON Departments.ID = Employees.DepartmentID
```

```
SELECT          Employees.ID, Employees.FirstName, Employees.LastName, Departments.Name
FROM              Departments INNER JOIN
                  Employees ON Departments.ID = Employees.DepartmentID
```

Results | Messages

	ID	FirstName	LastName	Name
1	285	Mohammed	Abu-Hadhoud	Engineering
2	286	Mohammad	Maher	Sales
3	287	Cameron	Lo	IT
4	288	Harper	Castillo	IT
5	289	Harper	Dominguez	Engineering
6	290	Ezra	Vu	IT
7	291	Jade	Hu	Accounting

عاوز الموظفين اللي شغالين في ال it بس

```
SELECT Employees.ID, Employees.FirstName, Employees.LastName,
Departments.Name AS DeptName
FROM Departments INNER JOIN
 Employees ON Departments.ID = Employees.DepartmentID
WHERE Departments.Name='IT';
```

لو ماشتغلتش روح اكتب أسماء الأقسام تاني في جدول ال DEPARTMENT

عاوزين نجيب الاسم والقسم والبلد

عشان تربط جدولين مع جدول بتكتب اول علاقه وبعدها تاني علاقه من غير ماتكرر اسم الجدول اللي مرتبط بيهم

```
SELECT
Employees.ID,Employees.FirstName,Employees.LastName,Departments.Nam
e AS Department,Countries.Name AS Country

FROM Employees INNER JOIN Departments ON
Employees.DepartmentID=Departments.ID
 INNER JOIN Countries ON Employees.CountryID=Countries.ID;
```

فلتر الداتا علي اللي بلدهم أمريكا

```
SELECT
Employees.ID,Employees.FirstName,Employees.LastName,Departments.Nam
e AS Department,Countries.Name AS Country

FROM Employees INNER JOIN Departments ON
Employees.DepartmentID=Departments.ID
 INNER JOIN Countries ON Employees.CountryID=Countries.ID

 WHERE Countries.Name='USA';
```

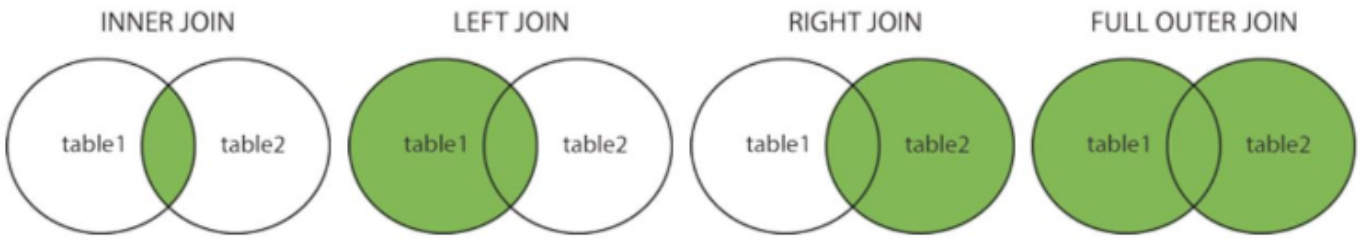
SQL JOIN

A `JOIN` clause is used to combine rows from two or more tables, based on a related column between them.

Different Types of SQL JOINs

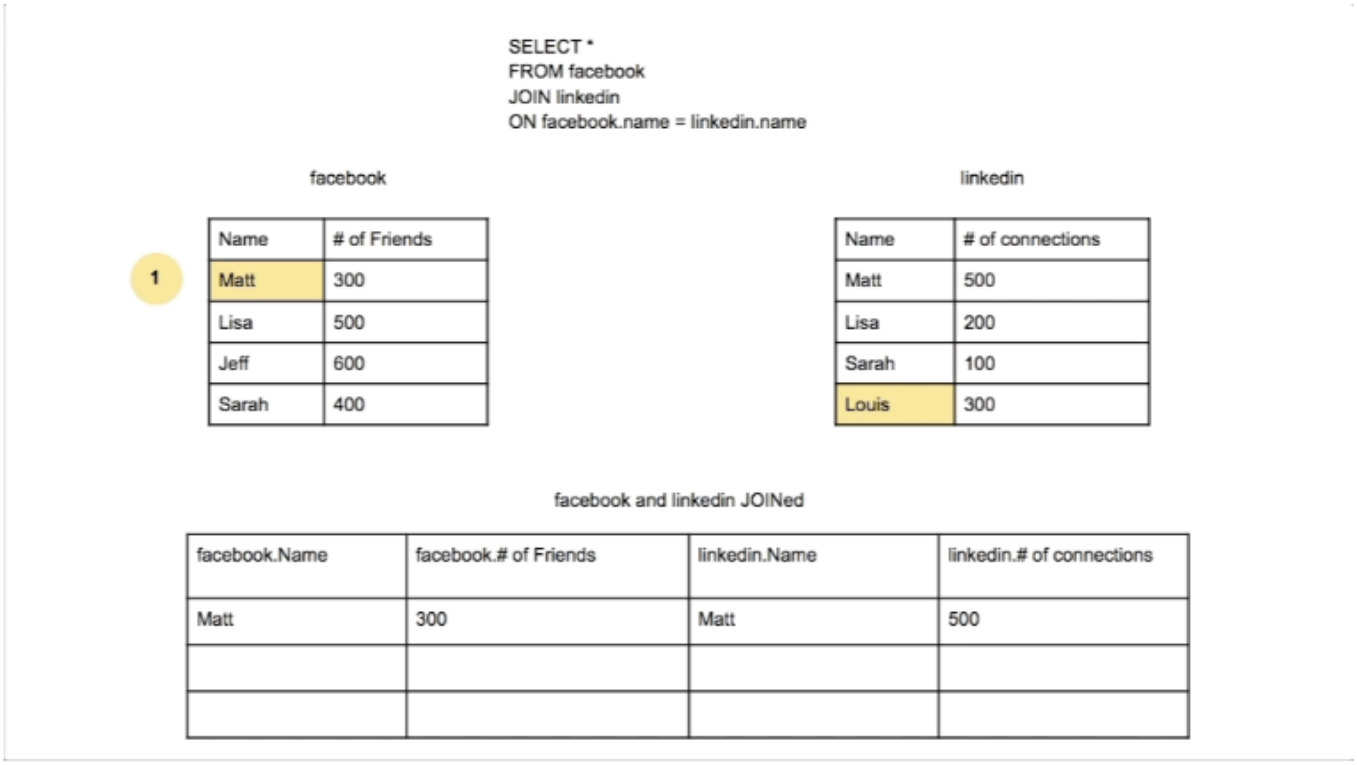
Here are the different types of the JOINs in SQL:

- `(INNER) JOIN` : Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN` : Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN` : Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN` : Returns all records when there is a match in either left or right table



SQL INNER JOIN

The SQL `INNER JOIN` joins two tables based on a common column, and selects records that have matching values in these columns.



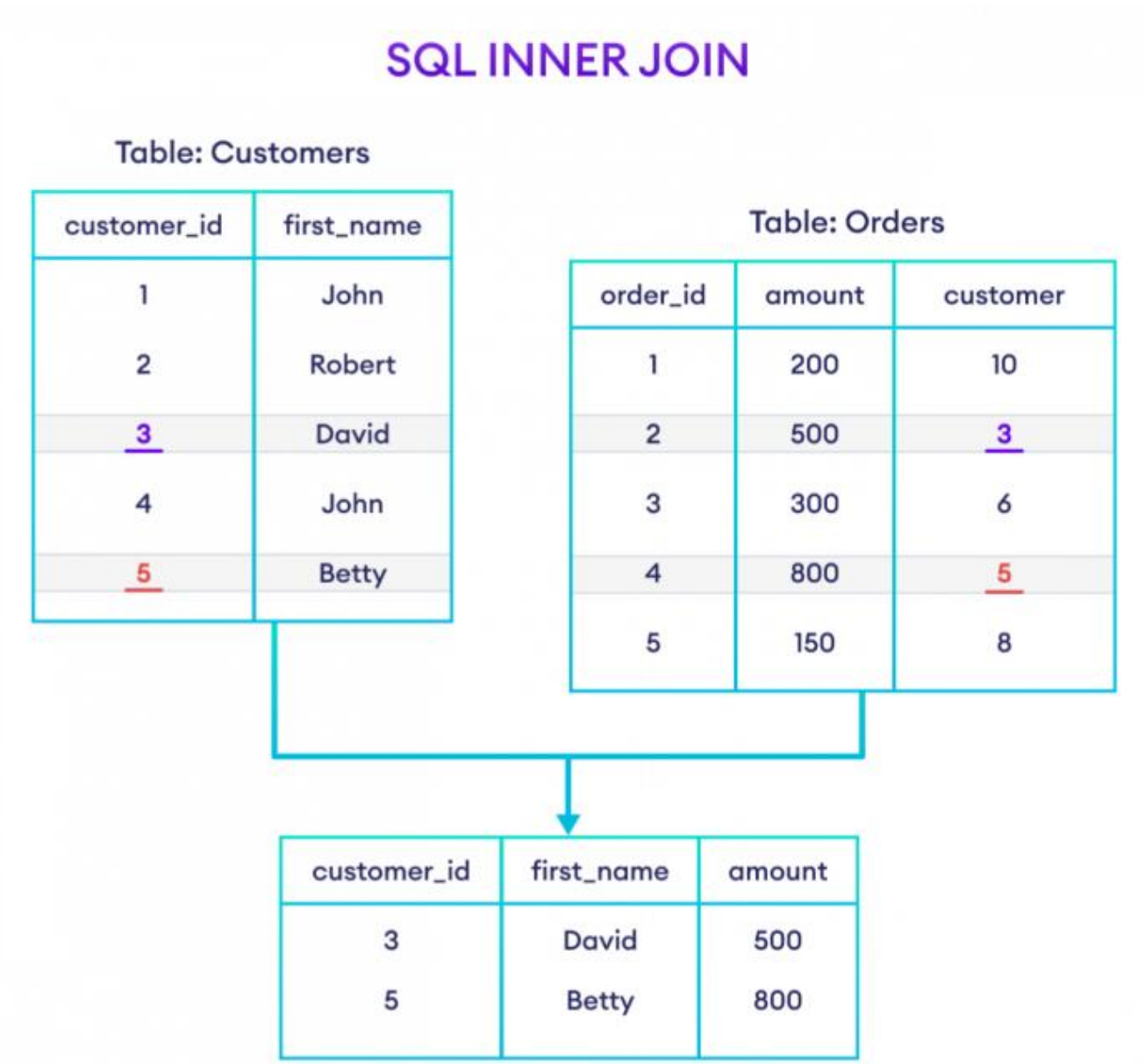
Name	# of Friends
Matt	300
Lisa	500
Jeff	600
Sarah	400

Name	# of connections
Matt	500
Lisa	200
Sarah	100
Louis	300

facebook.Name	facebook.# of Friends	linkedin.Name	linkedin.# of connections
Matt	300	Matt	500

Example

```
SELECT Customers.customer_id, Customers.first_name, Orders.amount
FROM Customers
INNER JOIN Orders
ON Customers.customer_id = Orders.customer;
```



Here, the SQL command selects **customer_id** and **first_name** columns (from the **Customers** table) and the **amount** column (from the **Orders** table).

And, the result set will contain those rows where there is a match between **customer_id** (of the **Customers** table) and **customer** (of the **Orders** table).

Syntax of INNER JOIN

The syntax of `INNER JOIN` is:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

INNER JOIN With WHERE Clause

Here's an example of the `INNER JOIN` with the WHERE clause:

```
SELECT Customers.customer_id, Customers.first_name, Orders.amount
FROM Customers
INNER JOIN Orders
ON Customers.customer_id = Orders.customer
WHERE Orders.amount >= 500;
```

Here, the SQL command joins two tables and selects rows where the amount is **greater than or equal to 500**.

```
-- Join and Inner Join are the same

select * from Customers;

select * from Orders;


SELECT Customers.CustomerID, Customers.Name, Orders.Amount
FROM Customers
JOIN Orders
ON Customers.CustomerID = Orders.CustomerID;



SELECT Customers.CustomerID, Customers.Name, Orders.Amount
FROM Customers
Inner JOIN Orders
ON Customers.CustomerID = Orders.CustomerID;
```

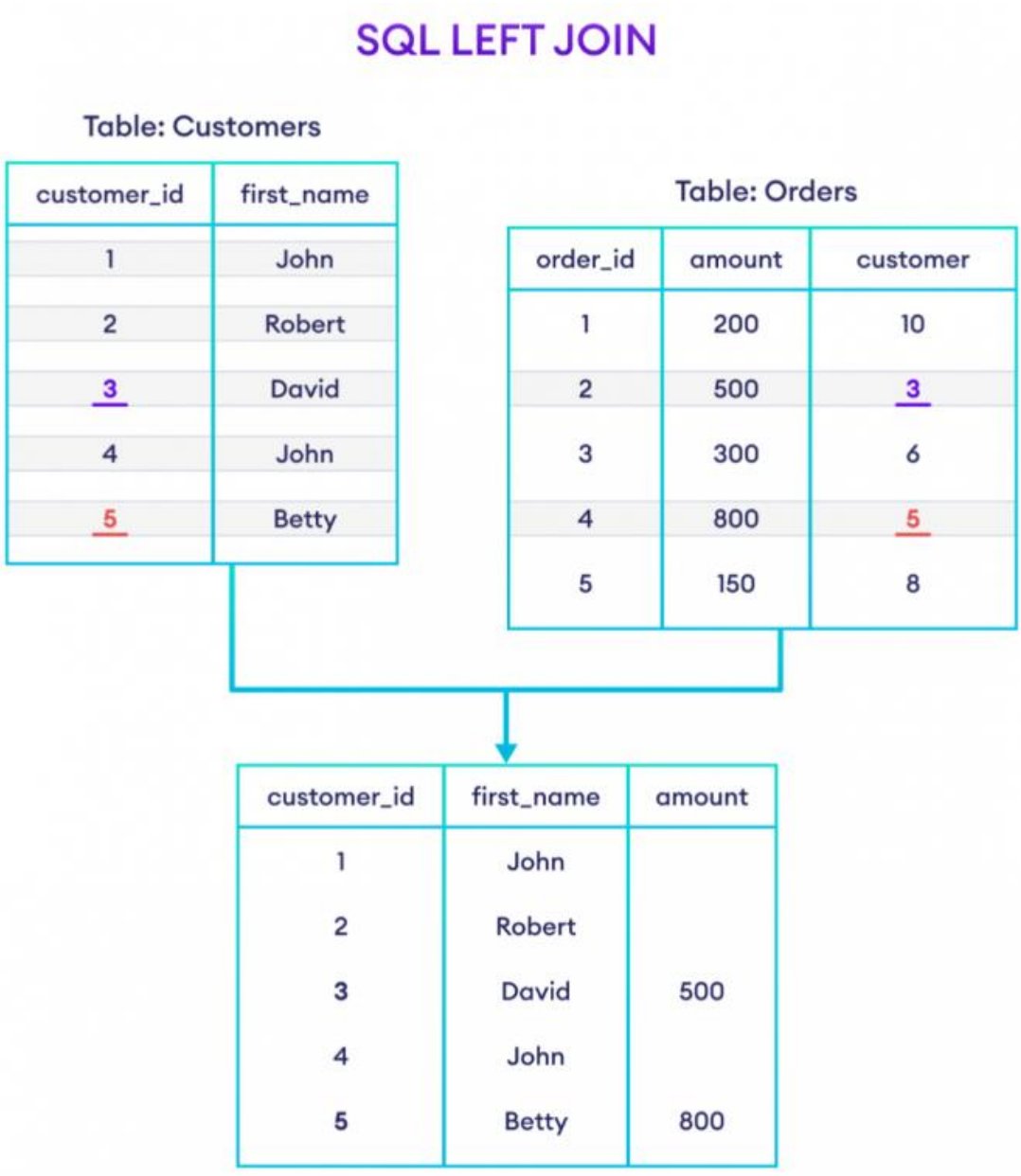
```
--This code for HR_Database

--Inner Join two Tables
SELECT        Employees.ID, Employees.FirstName, Employees.LastName, Departments.Name as DeptName
FROM            Employees INNER JOIN
                         Departments ON Employees.DepartmentID = Departments.ID


--Inner joind with where
SELECT        Employees.ID, Employees.FirstName, Employees.LastName, Departments.Name as DeptName
FROM            Employees INNER JOIN
                         Departments ON Employees.DepartmentID = Departments.ID
where Departments.Name = 'IT';
```

```sql
--Inner Join Three Tables
SELECT        Employees.ID, Employees.FirstName, Employees.LastName, Departments.Name as DeptName,
Countries.Name AS CountryName
FROM            Employees INNER JOIN
                         Departments ON Employees.DepartmentID = Departments.ID INNER JOIN
                         Countries ON Employees.CountryID = Countries.ID


--Inner Join Three Tables with where
SELECT        Employees.ID, Employees.FirstName, Employees.LastName, Departments.Name as DeptName,
Countries.Name AS CountryName
FROM            Employees INNER JOIN
                         Departments ON Employees.DepartmentID = Departments.ID INNER JOIN
                         Countries ON Employees.CountryID = Countries.ID
       where Countries.Name = 'USA';
```

Left (Outer) Join

ال LEFT JOIN بياخد كل الداتا من الجدول اللي عالشمال والداتا المشتركه بس من اللي عاليمين وساعات بيقولوا عليه LEFT OUTER JOIN

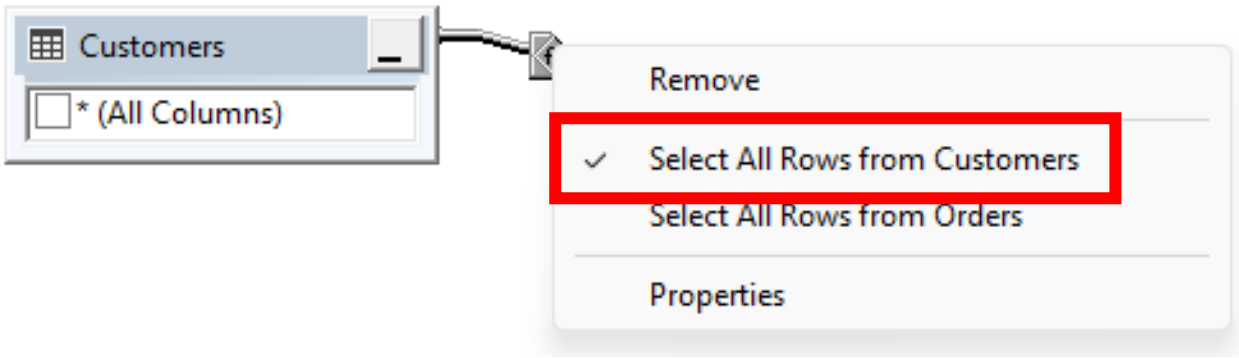
اني اجيب كل العملاء والاوردرات اللي عملوها في الجدول ده



طريقتها اني بدل ماكنت بكتب INNER لا بكتب LEFT

```
USE Shop_Database;
SELECT Customers.CustomerID,Customers.Name,Orders.Amount
FROM Customers LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID;
```

عشان اعملها في ال DESIGNER بعمل كليك يمين عالسهم بين الجدولين



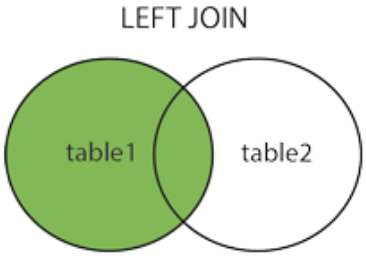
SQL LEFT JOIN Keyword

The `LEFT JOIN` keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

LEFT JOIN Syntax

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases LEFT JOIN is called LEFT OUTER JOIN.



The SQL `LEFT JOIN` joins two tables based on a common column, and selects records that have matching values in these columns and remaining rows from the left table.

3

```
SELECT *
FROM facebook
LEFT JOIN linkedin
ON facebook.name = linkedin.name
```

facebook

Name	# of Friends
Matt	300
Lisa	500
Jeff	600
Sarah	400

linkedin

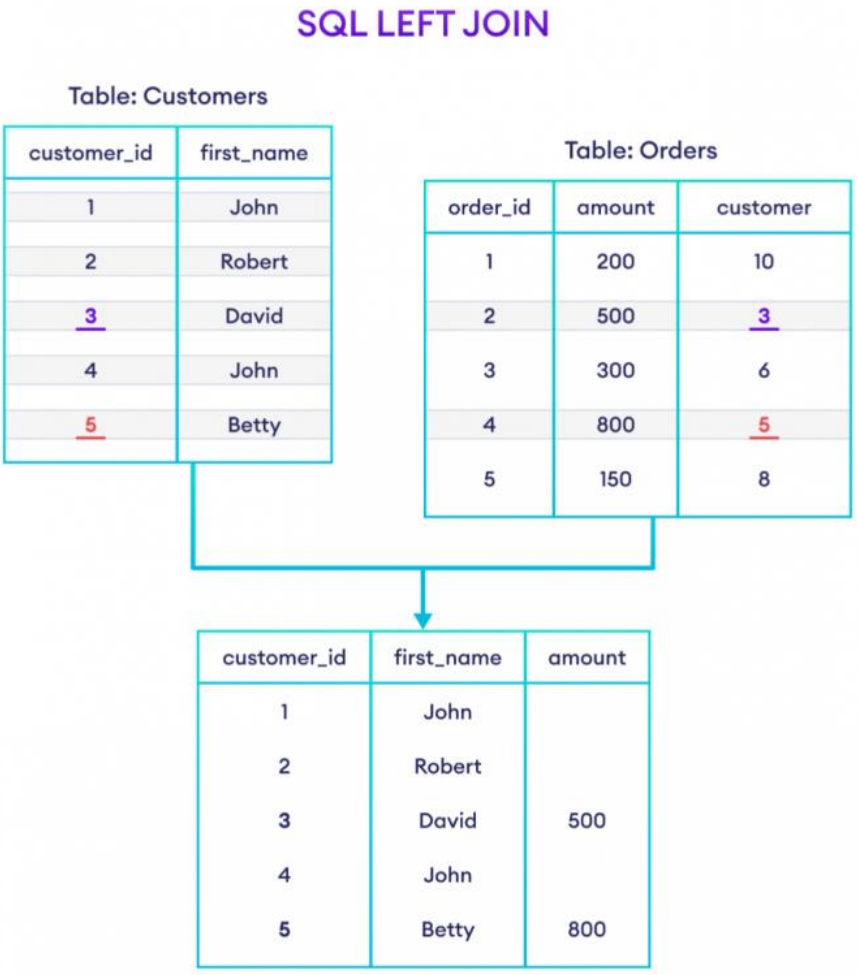
Name	# of connections
Matt	500
Lisa	200
Sarah	100
Louis	300

facebook and linkedin JOINed

facebook.Name	facebook.# of Friends	linkedin.Name	linkedin.# of connections
Matt	300	Matt	500
Lisa	500	Lisa	200

Example

```
SELECT Customers.customer_id, Customers.first_name, Orders.amount
FROM Customers
LEFT JOIN Orders
ON Customers.customer_id = Orders.customer;
```



Here, the SQL command selects **customer_id** and **first_name** columns (from the **Customers** table) and the **amount** column (from the **Orders** table).

And, the result set will contain those rows where there is a match between **customer_id** (of the **Customers** table) and **customer** (of the **Orders** table) along with all the remaining rows from the **Customers** table.

```
--Left Join and Left Outer Join are the same.

--Left Join: gets all data from table customers and only matched data from table orders
SELECT Customers.CustomerID, Customers.Name, Orders.Amount
FROM Customers
Left JOIN Orders
ON Customers.CustomerID = Orders.CustomerID;


SELECT Customers.CustomerID, Customers.Name, Orders.Amount
FROM Customers
Left Outer JOIN Orders
ON Customers.CustomerID = Orders.CustomerID;
```

Right (Outer) Join + Full (Outer) Join

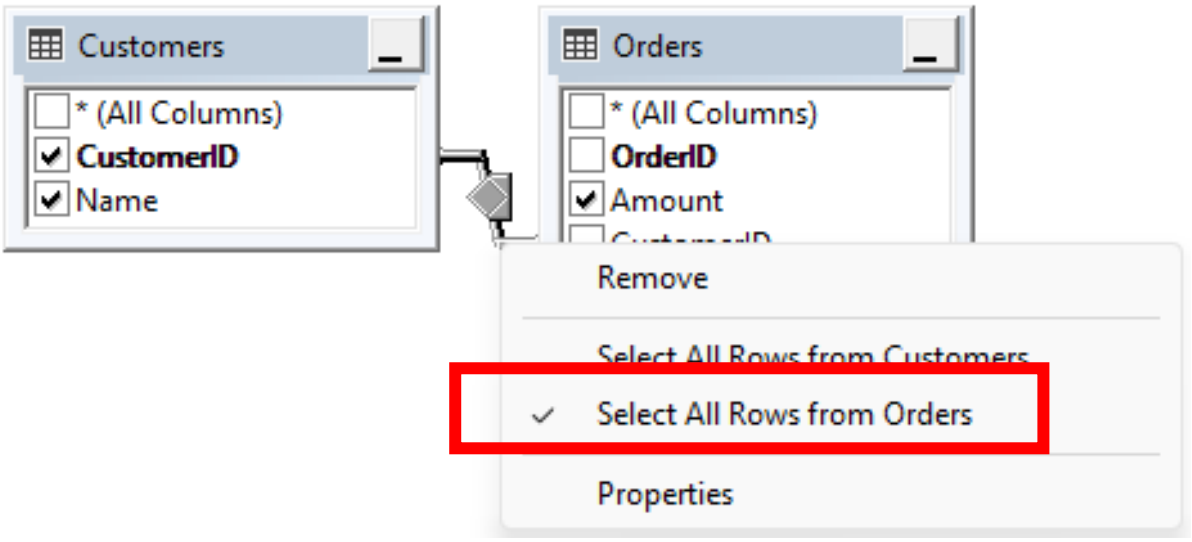
ال RIGHT JOIN او ال RIGHT OUTER JOIN بيجيب كل الداتا من اللي عاليمين والمشترك اللي في الشمال

وال FULL JOIN او ال FULL OUTER JOIN بيجيب كل الداتا من الجدولين

طريقتهم انك بس بتكتب اساميم

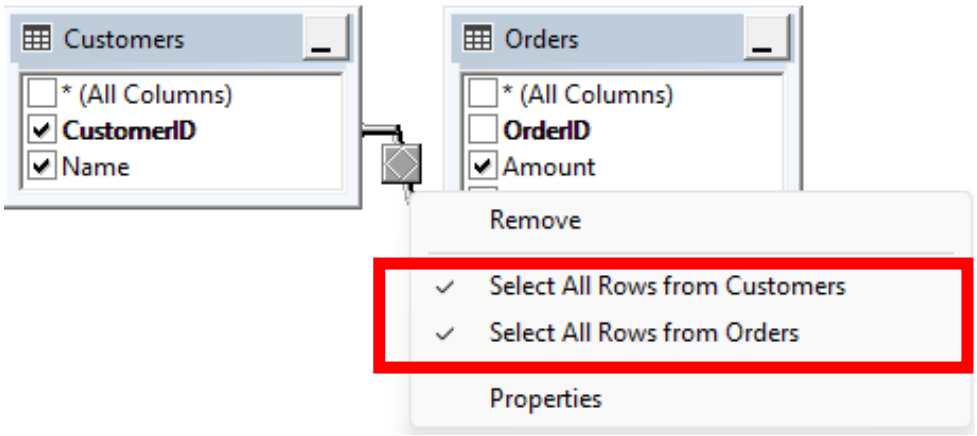
دي نفس ال QUERY بس بال RIGHT JOIN

```
SELECT Customers.CustomerID,Customers.Name,Orders.Amount
FROM Customers RIGHT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID;
```



دي نفس ال QUERY بس بال FULL JOIN

```
SELECT Customers.CustomerID,Customers.Name,Orders.Amount
FROM Customers FULL JOIN Orders
ON Customers.CustomerID=Orders.CustomerID;
```



SQL RIGHT JOIN Keyword

The `RIGHT JOIN` keyword returns all records from the right table (table2), and the matching records from the left table (table1).

The result is 0 records from the left side, if there is no match.

```
SELECT *
FROM facebook
JOIN linkedin
ON facebook.name = linkedin.name
```

facebook

Name	# of Friends
Matt	300
Lisa	500
Jeff	600
Sarah	400

linkedin

Name	# of connections
Matt	500
Lisa	200
Sarah	100
Louis	300

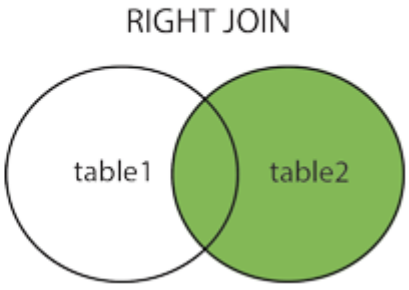
facebook and linkedin JOINed

facebook.Name	facebook.# of Friends	linkedin.Name	linkedin.# of connections
Matt	300	Matt	500
Lisa	500	Lisa	200

RIGHT JOIN Syntax

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases `RIGHT JOIN` is called `RIGHT OUTER JOIN`.



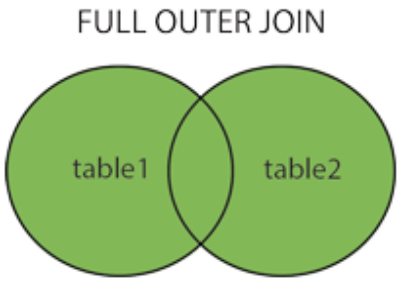
SQL FULL OUTER JOIN Keyword

The `FULL OUTER JOIN` keyword returns all records when there is a match in left (table1) or right (table2) table records.

Tip: `FULL OUTER JOIN` and `FULL JOIN` are the same.

FULL OUTER JOIN Syntax

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```



Note: `FULL OUTER JOIN` can potentially return very large result-sets!

```
SELECT *
FROM facebook
FULL OUTER JOIN linkedin
ON facebook.name = linkedin.name
```

facebook

Name	# of Friends
Matt	300
Lisa	500
Jeff	600
Sarah	400

linkedin

Name	# of connections
Matt	500
Lisa	200
Sarah	100
Louis	300

6

facebook and linkedin JOINed

facebook.Name	facebook.# of Friends	linkedin.Name	linkedin.# of connections
Matt	300	Matt	500
Lisa	500	Lisa	200
Jeff	600	*Null*	*Null*
Sarah	400	Sarah	100

```sql
--Inner Join
SELECT        Customers.CustomerID, Customers.Name, Orders.Amount
FROM            Customers INNER JOIN
                         Orders ON Customers.CustomerID = Orders.CustomerID
--Left Join
SELECT        Customers.CustomerID, Customers.Name, Orders.Amount
FROM            Customers LEFT OUTER JOIN
                         Orders ON Customers.CustomerID = Orders.CustomerID
--Right Join
SELECT        Customers.CustomerID, Customers.Name, Orders.Amount
FROM            Customers RIGHT OUTER JOIN
                         Orders ON Customers.CustomerID = Orders.CustomerID
--Full Join
SELECT        Customers.CustomerID, Customers.Name, Orders.Amount
FROM            Customers FULL OUTER JOIN
                         Orders ON Customers.CustomerID = Orders.CustomerID
```

Views

لو فيه عندي QUERY بستخدمها كتير اقدر احطها جوه VIEW وهوا زيه زي ال FUNCTION كده

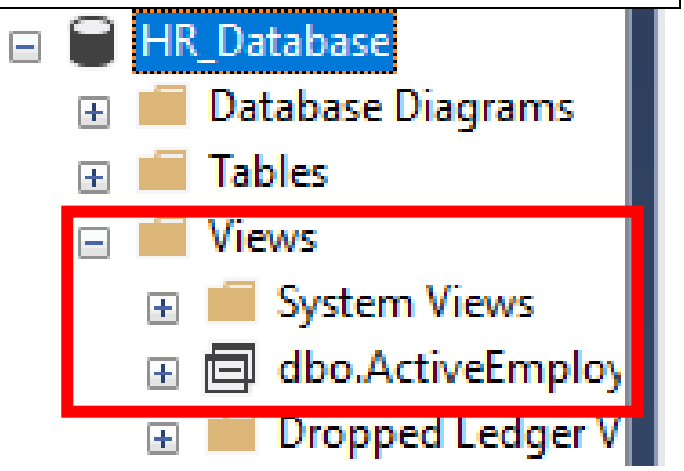
عشان اعمل VIEW بكتب CREATE VIEW وبعدين بكتب اسم ليه وبعدين AS وبعدين بحط ال QUERY جواه

عاوز الموظفين اللي لسه موجودين في الشركه

```sql
SELECT * FROM Employees
WHERE ExitDate IS NULL;
```

عاوز احط ال QUERY اللي عملتها في VIEW

```sql
CREATE VIEW ActiveEmployees AS
SELECT * FROM Employees
WHERE ExitDate IS NULL;
```



كده اقدر اتعامل مع ال VIEW دي كانها جدول بس هيا مش جدول هيا اختصار للكود

```sql
SELECT* FROM ActiveEmployees;
```

عاوز اعمل view للناس اللي مشيت

```sql
CREATE VIEW ResignedEmployees AS
SELECT * FROM Employees
WHERE ExitDate IS NOT NULL
```

```sql
SELECT * FROM ResignedEmployees
```

عاوز اعمل VIEW للاسم والجنس بس

```sql
CREATE VIEW ShortDetailedEmployees AS
SELECT ID,FirstName,LastName,Gendor FROM Employees
```

```sql
SELECT* FROM ShortDetailedEmployees
```

اقدر اعمل صلاحيات علي اليوزر انه مايشوفش غير ال VIEW ده

SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the `CREATE VIEW` statement.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Note: A view always shows up-to-date data! The database engine recreates the view, every time a user queries it.

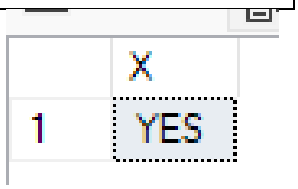
SQL - More Queries

Exists

ال EXIST زي ماقولنا قبل كده بتكتب جواها QUERY ولو في RECORD واحد او اكتر طلع بترجعلك TRUE

اعملي عمود اسمه X وحط فيه كلمةYES لو عندك عملاء ال ID بتاعهم ب 3 والطلبات بتاعتهم عامله اقل من 600

```
SELECT X='YES'
WHERE EXISTS(
SELECT * FROM Orders
WHERE Orders.CustomerID=3 AND Orders.Amount<600)
```



	X
1	YES

رجعلي جدول ال customers كله وسميه t1 لو فيه عميل من جدول ال order ال id بتاعه موجود في ال t1 وال فلوس اقل من 600

```
SELECT *from Customers T1
WHERE EXISTS(
SELECT * FROM Orders
WHERE Orders.CustomerID=T1.CustomerID AND Orders.Amount<600)
```

ال queryاللي عملناها دي بطيئه لان ال exists مهما كانت الداتا اللي هترجع هترجع true او false

عشان كده بيقولك بدل مايرجع الداتا كلها لا خليه يرجع اول record بس لانه ده اقل شيء مطلوب عشان يرجع true

```
SELECT *from Customers T1
WHERE EXISTS(
SELECT top 1 * FROM Orders
WHERE Orders.CustomerID=T1.CustomerID AND Orders.Amount<600)
```

ممكن اخليها اسرع واسرع وهيا بدل مايجيبلي كل الاعمده لا هخليه يرجع أي حاجه من عندي

```
SELECT *from Customers T1
WHERE EXISTS(
SELECT top 1 x='y' FROM Orders
WHERE Orders.CustomerID=T1.CustomerID AND Orders.Amount<600)
```

The SQL EXISTS Operator

The `EXISTS` operator is used to test for the existence of any record in a subquery.

The `EXISTS` operator returns TRUE if the subquery returns one or more records.

EXISTS Syntax

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);
```

The SQL `EXISTS` operator executes the outer SQL query if the subquery is not `NULL` (empty result-set).

```
select X='yes'
where  exists
   (
       select * from Orders
       where customerID= 3 and Amount < 600
       )


select * from Customers T1
where
  exists
   (
       select * from Orders
       where customerID= T1.CustomerID and Amount < 600
       )

--More optimized and faster
select * from Customers T1
where
  exists
   (
       select top 1 * from Orders
       where customerID= T1.CustomerID and Amount < 600
       )


--More optimized and faster
select * from Customers T1
where
  exists
   (
       select top 1 R='Y'  from Orders
       where customerID= T1.CustomerID and Amount < 600
       )
```

Union

لو عندي جدولين او اتنين result sets ليهم نفس الاعمده بنفس المسميات اقدر احطهم كلهم في query واحده عن طريق انك تكتب union بينهم

```sql
select * from ActiveEmployees
union
select * from ResignedEmployees
```

لو فيه records متكرره بتشيلهم

```sql
select * from Departments
union
select * from Departments
```

لو كتبت كلمة ALL مش هيشيل المتكرر

```sql
select * from Departments
union ALL
select * from Departments
```

The SQL UNION Operator

The `UNION` operator is used to combine the result-set of two or more `SELECT` statements.

- Every `SELECT` statement within `UNION` must have the same number of columns
- The columns must also have similar data types
- The columns in every `SELECT` statement must also be in the same order

UNION Syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

UNION ALL Syntax

The `UNION` operator selects only distinct values by default. To allow duplicate values, use `UNION ALL`:

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```

Note: The column names in the result-set are usually equal to the column names in the first `SELECT` statement.

وتقدر تضيف عمود يميز الداتا اللي راجعه من كل جدول عن طريق اضافه عمود جديد زي كده

```
select *,X=1 from Departments
union ALL
select *,X=2  from Departments
```

لو عندك جدول فيه فواتير مشتريات وجدول تاني فيه فواتير مبيعات وعايز تجيبهم كلهم في جدول واحد مثلا وتضيف عمود عمود يبينلك نوع الفاتوره ومن خلاله تقدر تطلع تقرير بارصدة الأصناف

Case

ال CASE هنا شبه مفهوم ال SWICH في البرمجه

طريقتها اني بكتب case وتحتها كلمة end وفي النص هحط الشروط كانها if else statement

بكتب when وبعدها الشرط وبعدين then وبعدين اللي عايز اعمله

ولو فيه حالات تانيه بكتبها بنفس الطريقه

وفي الاخر بكتب else وبحط الحاجه ال default

لو عندي ال query دي

```
SELECT ID,FirstName,LastName from Employees;
```

وعايز استبدل الحرف m بكلمة male والحرف f بكلمة female وحطهم في عمود جديد

```
SELECT ID,FirstName,LastName ,GenderTitle=
CASE
WHEN Gendor='M' THEN 'Male'
WHEN Gendor='F' THEN'Female'
ELSE 'Unknown'
END
from Employees;
```

عاوز ازود عمود يقولي لو الموظف شغال لا مشي

```
SELECT ID,FirstName,LastName ,GenderTitle=
CASE
WHEN Gendor='M' THEN 'Male'
WHEN Gendor='F' THEN'Female'
ELSE 'Unknown'
END
,
Status=
CASE
WHEN ExitDate IS NULL THEN 'NOT ACTIVE'
WHEN ExitDate IS NOT NULL THEN 'ACTIVE'
ELSE 'Unknown'
```

```
END
from Employees;
```

هنا لو النوع ذكر بيزوده 10% من المرتب ولو انثي بتزيد 15%

```
select ID, FirstName, LastName,MonthlySalary,

NewSalaryToBe =
CASE
    WHEN Gendor='M' THEN MonthlySalary * 1.1
    WHEN Gendor='F' THEN MonthlySalary * 1.15

END
from Employees
```

The SQL CASE Expression

The `CASE` expression goes through conditions and returns a value when the first condition is met (like an if-then-else statement).

So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in

the `ELSE` clause.

If there is no `ELSE` part and no conditions are true, it returns NULL.

CASE Syntax

```
CASE
  WHEN condition1 THEN result1
  WHEN condition2 THEN result2
  WHEN conditionN THEN resultN
  ELSE result
END;
```

```
select ID, FirstName, LastName, GendorTitle =
CASE
    WHEN Gendor='M' THEN 'Male'
    WHEN Gendor='F' THEN 'Female'
    ELSE 'Unknown'
END

from Employees

----------------------------------------

select ID, FirstName, LastName,  GendorTitle =
CASE
    WHEN Gendor='M' THEN 'Male'
    WHEN Gendor='F' THEN 'Female'
    ELSE 'Unknown'
END,

Status =
CASE
    WHEN ExitDate is null THEN 'Active'
    WHEN Gendor is Not null THEN 'Resigned'
END
from Employees
```

```
-----------------------------------------------------

select ID, FirstName, LastName,MonthlySalary,

NewSalaryToBe =
CASE
    WHEN Gendor='M' THEN MonthlySalary * 1.1
    WHEN Gendor='F' THEN MonthlySalary * 1.15

END
from Employees
```

More about Constraints

Constraints

SQL Create Constraints

Constraints can be specified when the table is created with the `CREATE TABLE` statement, or after the table is created with the `ALTER TABLE` statement.

Syntax

```
CREATE TABLE table_name (
  column1 datatype constraint,
  column2 datatype constraint,
  column3 datatype constraint,
  ....
);
```

SQL Constraints

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

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- `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`. Uniquely identifies each row in a table
- `FOREIGN KEY` - Prevents actions that would destroy links between tables
- `CHECK` - Ensures that the values in a column satisfies a specific condition
- `DEFAULT` - Sets a default value for a column if no value is specified
- `CREATE INDEX` - Used to create and retrieve data from the database very quickly

More about Constraints

What is Constraint? and Why it's Important?

https://programmingadvices.com/courses/database-level-1-sql-concepts-and-practice/lectures/46898430

Primary Key Constraint

هنا فيه معلومه لو عايز تستخدم اكتر من عمود ك PRIMARY KEY

بتكتب CONSTRAINT وبعدها بتكتب اسم عمود جديد وبعدين PRIMARY KEY وتفتح قوسين وتحط فيهم اسامي الاعمده اللي عاوزها تبقي PRIMARY KEYمع بعض

```sql
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
     CONSTRAINT PK_Person PRIMARY KEY (ID,LastName)
);
```

	Column Name	Data Type	Allow Nulls
🔑	ID	int	☐
	LastName	varchar(255)	☐
	FirstName	varchar(255)	☑
	Age	int	☑
			☐

عشان تحذف عمود ال primary key اللي عملته من عمودين بتكتب قبل اسمه constraint

```
ALTER TABLE Persons
DROP CONSTRAINT PK_Person;
```

SQL PRIMARY KEY Constraint

The `PRIMARY KEY` constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

SQL PRIMARY KEY on CREATE TABLE

The following SQL creates a `PRIMARY KEY` on the "ID" column when the "Persons" table is created:

```
CREATE TABLE Persons (
  ID int NOT NULL PRIMARY KEY,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int
);
```

To allow naming of a `PRIMARY KEY` constraint, and for defining a `PRIMARY KEY` constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  CONSTRAINT PK_Person PRIMARY KEY (ID,LastName)
);
```

Note: In the example above there is only ONE `PRIMARY KEY` (PK_Person). However, the VALUE of the primary key is made up of TWO COLUMNS (ID + LastName).

SQL PRIMARY KEY on ALTER TABLE

To create a `PRIMARY KEY` constraint on the "ID" column when the table is already created, use the following SQL:

```
ALTER TABLE Persons
ADD PRIMARY KEY (ID);
```

To allow naming of a `PRIMARY KEY` constraint, and for defining a `PRIMARY KEY` constraint on multiple columns, use the following SQL syntax:

```
ALTER TABLE Persons
ADD CONSTRAINT PK_Person PRIMARY KEY (ID,LastName);
```

Note: If you use `ALTER TABLE` to add a primary key, the primary key column(s) must have been declared to not contain NULL values (when the table was first created).

DROP a PRIMARY KEY Constraint

To drop a `PRIMARY KEY` constraint, use the following SQL:

```
ALTER TABLE Persons
DROP CONSTRAINT PK_Person;
```

Foreign Key Constraint

SQL FOREIGN KEY Constraint

The `FOREIGN KEY` constraint is used to prevent actions that would destroy links between tables.

A `FOREIGN KEY` is a field (or collection of fields) in one table, that refers to the `PRIMARY KEY` in another table.

SQL FOREIGN KEY on CREATE TABLE

The following SQL creates a `FOREIGN KEY` on the "PersonID" column when the "Orders" table is created:

```
CREATE TABLE Orders (
  OrderID int NOT NULL PRIMARY KEY,
  OrderNumber int NOT NULL,
  PersonID int FOREIGN KEY REFERENCES Persons(PersonID)
);
```

To allow naming of a `FOREIGN KEY` constraint, and for defining a `FOREIGN KEY` constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE Orders (
  OrderID int NOT NULL,
  OrderNumber int NOT NULL,
  PersonID int,
   PRIMARY KEY (OrderID),
   CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)
   REFERENCES Persons(PersonID)
);
```

SQL FOREIGN KEY on ALTER TABLE

To create a `FOREIGN KEY` constraint on the "PersonID" column when the "Orders" table is already created, use the following SQL:

```
ALTER TABLE Orders
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

To allow naming of a `FOREIGN KEY` constraint, and for defining a `FOREIGN KEY` constraint on multiple columns, use the following SQL syntax:

```
ALTER TABLE Orders
ADD CONSTRAINT FK_PersonOrder
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

DROP a FOREIGN KEY Constraint

To drop a `FOREIGN KEY` constraint, use the following SQL:

```
ALTER TABLE Orders
DROP CONSTRAINT FK_PersonOrder;
```

Not Null Constraint

SQL NOT NULL on CREATE TABLE

The following SQL ensures that the "ID", "LastName", and "FirstName" columns will NOT accept NULL values when the "Persons" table is created:

Example

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255) NOT NULL,
  Age int
);
```

SQL NOT NULL on ALTER TABLE

To create a `NOT NULL` constraint on the "Age" column when the "Persons" table is already created, use the following SQL:

```
ALTER TABLE Persons
ALTER COLUMN Age int NOT NULL;
```

DEFAULT Constraint

ده بيبقي default value لو اليوزر مادخلش داتا بحط قيمه افتراضيه ليها وطريقتها اني بكتب كلمة default بعد تعريف العمود في الجدول القيمه الافتراضيه تقدر تحطها بايدك او تستخدم function يجيبهالك

هنعمل داتا بيز جديده

```
Create database DB3;
```

هنعمل جدول اسمه persons وبنخلي ال city ال default بتاعها amman

```
use DB3;

CREATE TABLE Persons(
ID INT NOT NULL ,
LastName varchar(255) NOT NULL,
FirstName varchar(255) NOT NULL,
```

```
Age int,
City varchar(255) DEFAULT 'Amman'
);
```

لو مادخلتش قيمه للمدينه وجيت قفلت الجدول وفتحته تاني هتلاقيها اتحولت بقت عمان

DESKTOP-VS16DFG.DB3 - dbo.Persons | SQLQuery1.sql - D...16DFG\Ahmed (74))*

	ID	LastName	FirstName	Age	City
!	1	Abu-Hadhud	Mohamed	45	NULL
▸*	NULL	NULL	NULL	NULL	NULL

DESKTOP-VS16DFG.DB3 - dbo.Persons | SQLQuery1.sql - D...16DFG\Ahmed (74))*

	ID	LastName	FirstName	Age	City
▸	1	Abu-Hadhud	Mohamed	45	Amman
*	NULL	NULL	NULL	NULL	NULL

وطبعا تقدر تعدلها

عاوزين نعمل جدول لل orders ونخلي التاريخ بتاعه هوا تاريخ السيستم

```
CREATE TABLE Orders(
ID INT NOT NULL ,
OrderNumber int not null,
OrderDate date default GETDATE());
```

DESKTOP-VS16DFG.DB3 - dbo.Orders | SQLQuery1.sql - D...

	ID	OrderNumber	OrderDate
!	1	1223123	NULL

ID	OrderNumber	OrderDate
1	1223123	2023-08-29

طيب هنحذف الجدولين ونعمل جدول ال PERSONS من غير default

```
drop table Persons;
drop table Orders;

CREATE TABLE Persons(
ID INT NOT NULL ,
LastName varchar(255) NOT NULL,
FirstName varchar(255) NOT NULL,
```

```
Age int,
City varchar(255)
);
```

عاوزين نرجع ال DEFAULT تاني

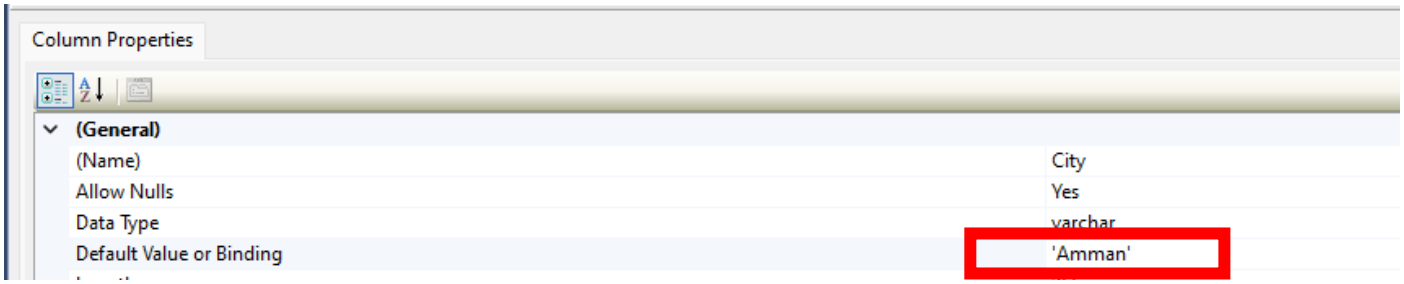
بنكتب ADD CONSTRAINT ونحط اسم لل constraint عشان تقدر تحذفها بعدين لو حبيت دي وبعدين نكتب default ونحط القيمه وبعدين for ونكتب اسم العمود

```
ALTER TABLE Persons
ADD CONSTRAINT df_City
Default 'Amman' FOR City;
```

احذف ال constraint اللي عملتها

```
ALTER TABLE Persons
DROP CONSTRAINT df_City;
```

تقدر تعدلها من هنا



SQL DEFAULT Constraint

The `DEFAULT` constraint is used to set a default value for a column.

The default value will be added to all new records, if no other value is specified.

SQL DEFAULT on CREATE TABLE

The following SQL sets a `DEFAULT` value for the "City" column when the "Persons" table is created:

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  City varchar(255) DEFAULT 'Amman'
);
```

The `DEFAULT` constraint can also be used to insert system values, by using functions like `GETDATE()`:

```
CREATE TABLE Orders (
  ID int NOT NULL,
  OrderNumber int NOT NULL,
  OrderDate date DEFAULT GETDATE()
);
```

SQL DEFAULT on ALTER TABLE

To create a `DEFAULT` constraint on the "City" column when the table is already created, use the following SQL:

```
ALTER TABLE Persons
ADD CONSTRAINT df_City
DEFAULT 'Amman' FOR City;
```

DROP a DEFAULT Constraint

To drop a `DEFAULT` constraint, use the following SQL:

```
ALTER TABLE Persons
DROP Constraint  df_City;
```

Check Constraint

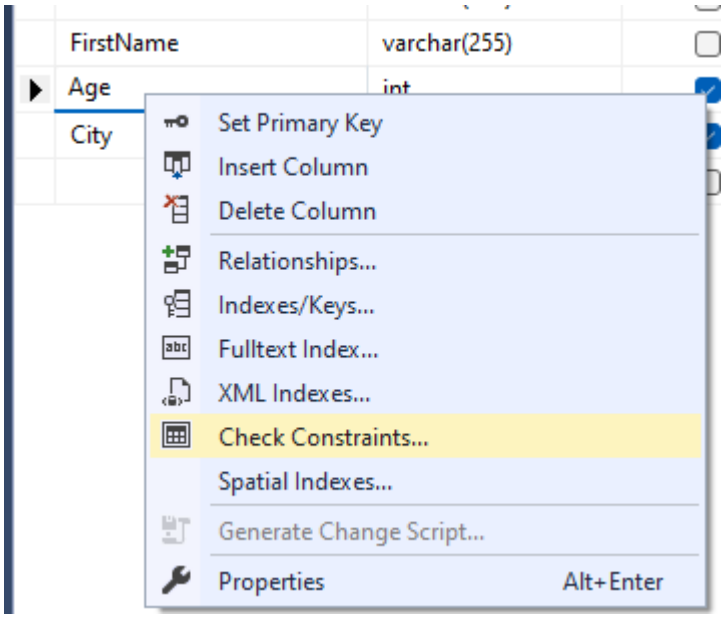
Check constraint لو عايز احط قيود عالداتا اللي هتدخل انه تكون مثلا اكبر او اصغر من قيمة معينه

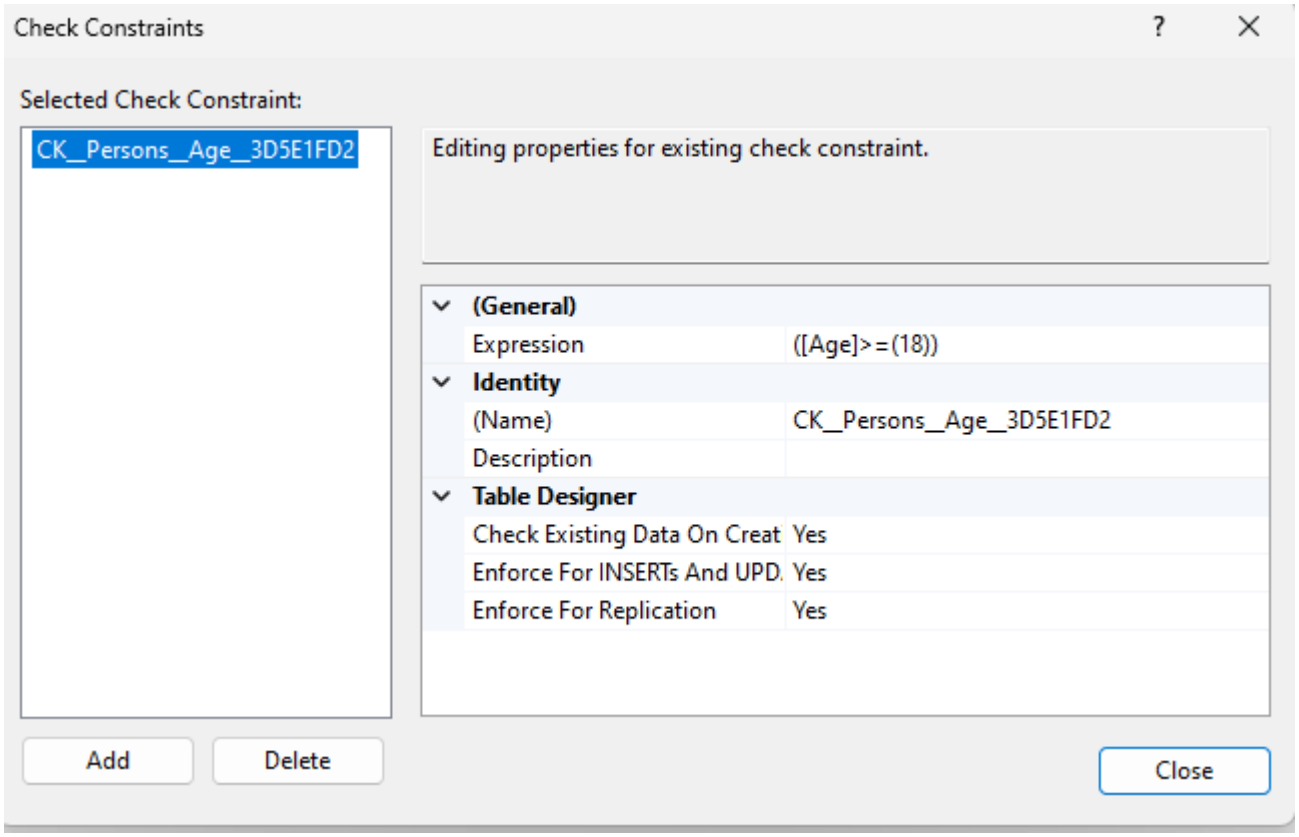
طريقتها بكتب check وبعدين الشرط

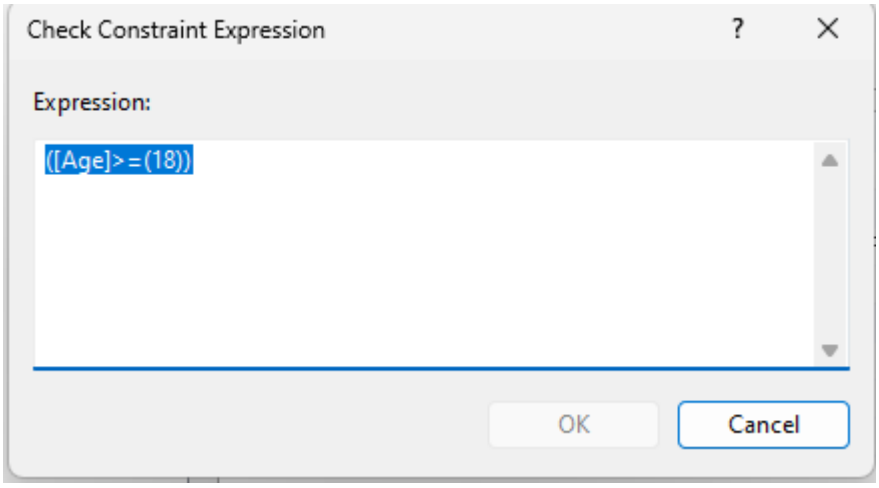
عاوز احط شرط للسن يكون 18 او اكبر

```
CREATE TABLE Persons(
ID INT NOT NULL ,
LastName varchar(255) NOT NULL,
FirstName varchar(255) NOT NULL,
Age int  check(Age>=18),
City varchar(255)
);
```

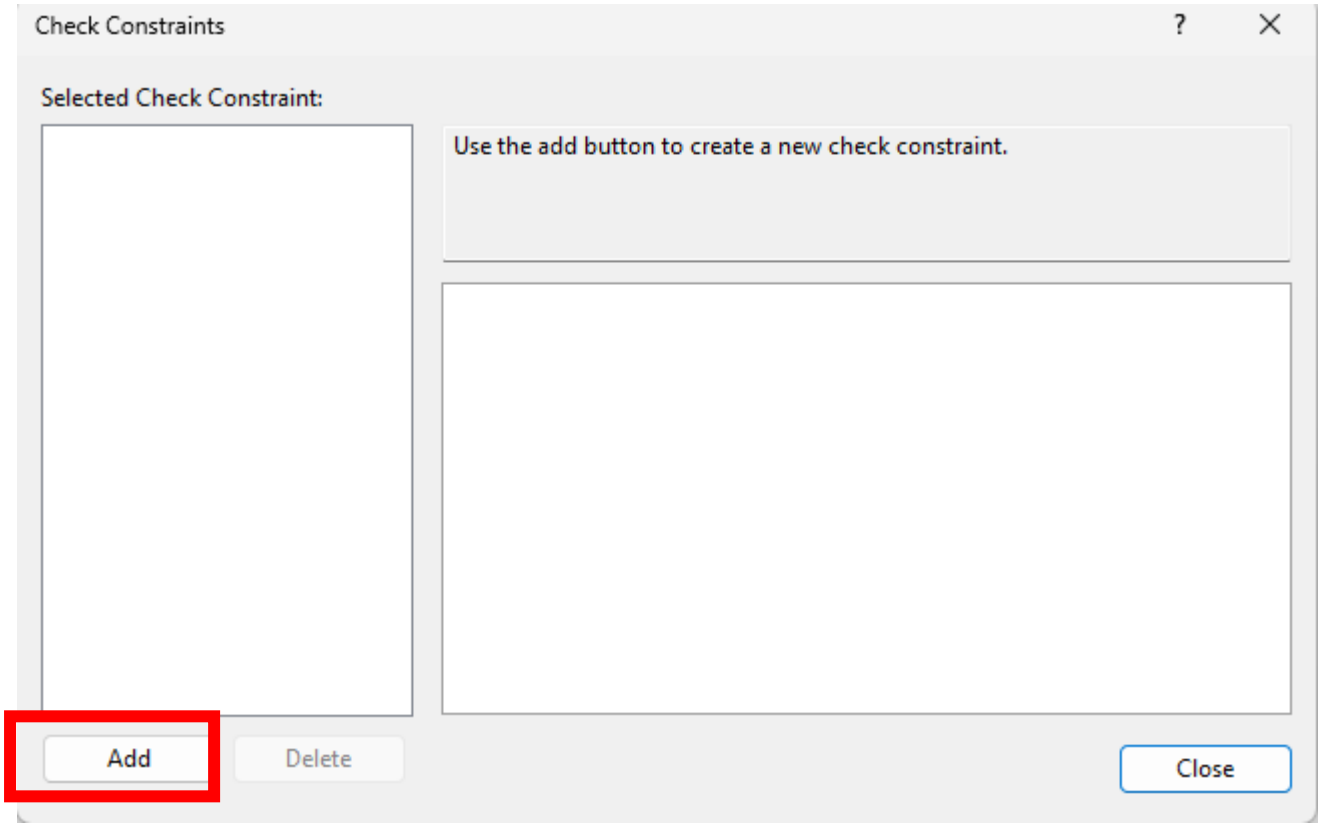
لو عايز اعملها في الديزاين

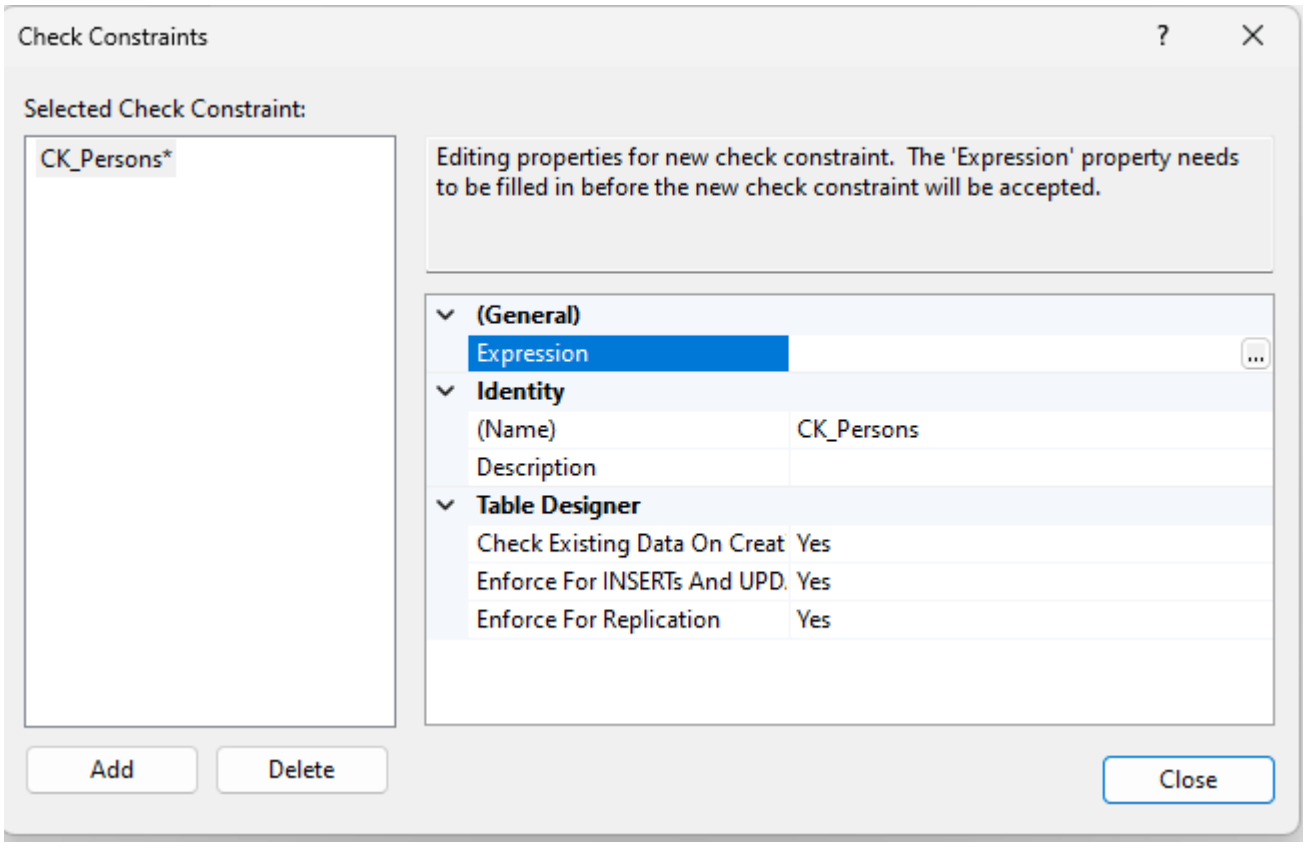


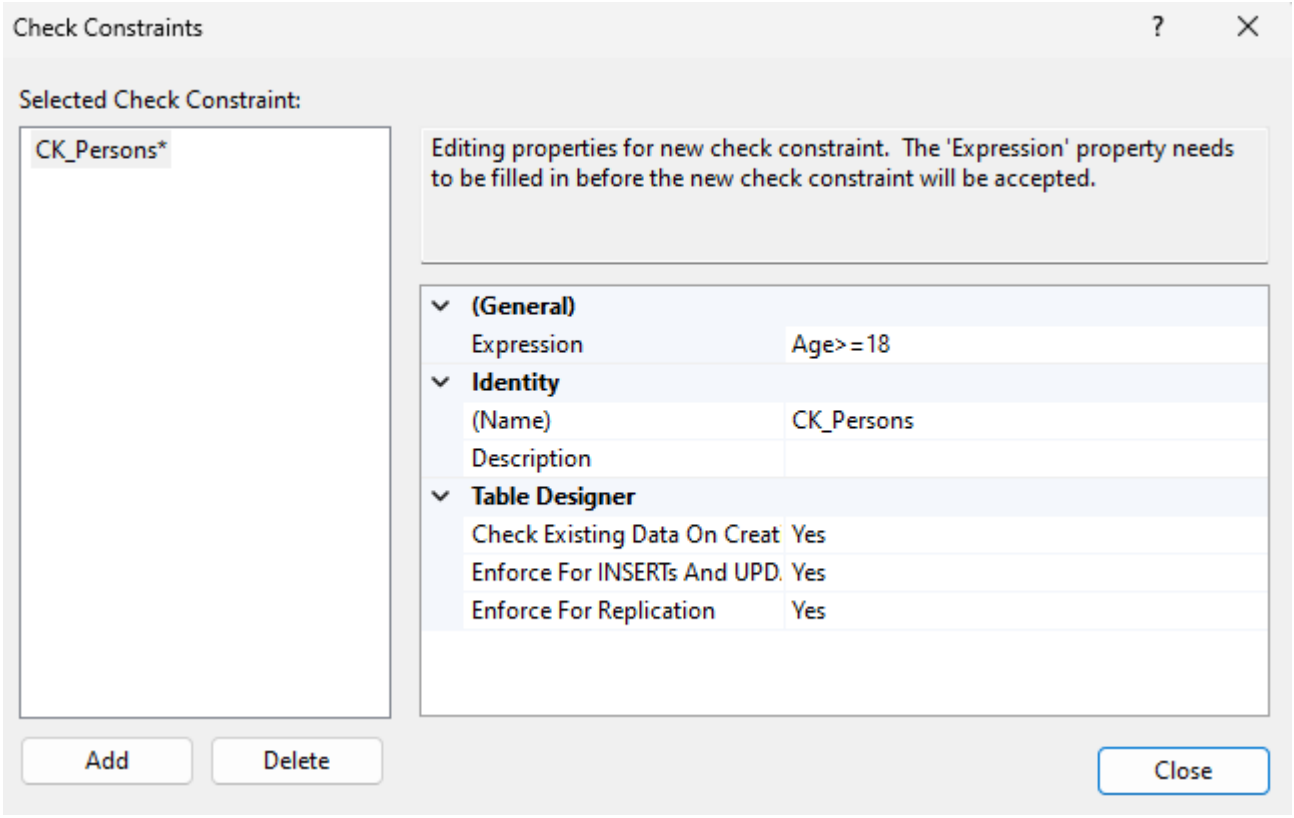




حذفتها وعملتها تاني


Check Constraints
Selected Check Constraint:
Use the add button to create a new check constraint.
Add
Delete
Close


Check Constraints
Selected Check Constraint:
CK_Persons*
Editing properties for new check constraint. The 'Expression' property needs to be filled in before the new check constraint will be accepted.
(General)
Expression
Identity
(Name)
CK_Persons
Description
Table Designer
Check Existing Data On Creat
Yes
Enforce For INSERTs And UPD.
Yes
Enforce For Replication
Yes
Add
Delete
Close



احذف الجدول تاني

```
drop table Persons
```

عاوزين نعمل الجدول تاني بس هنضيف constraint نقول فيها ان ال city بتاعتها تكون عمان اجباري والعمر اكبر من او يساوي 18

في نفس ال constraint

```
CREATE TABLE Persons(
ID INT NOT NULL ,
LastName varchar(255) NOT NULL,
FirstName varchar(255) NOT NULL,
Age int,
City varchar(255),
constraint chk_Persons  check(Age>=18 AND City='Amman')

);
```

عايز احذف ال constraint

```
ALTER TABLE Persons
drop constraint chk_Persons
```

SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

SQL CHECK on CREATE TABLE

The following SQL creates a CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that the age of a person must be 18, or older:

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int CHECK (Age>=18)
);
```

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  City varchar(255),
   CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Amman')
);
```

DROP a CHECK Constraint

To drop a CHECK constraint, use the following SQL:

```
ALTER TABLE Persons
DROP CONSTRAINT CHK_Person;
```

Unique Constraint

ال unique constraint مش بتقبل تكرار بس بيسمحلك تدخل null مره واحده بس في العمود وطريقتها انك تكتب كلمة unique جنب العمود

```
CREATE TABLE Persons(
ID INT unique,
LastName varchar(255) NOT NULL,
FirstName varchar(255) ,
Age int,
City varchar(255),
);
```

عايز احط ال unique علي عمودين

```
CREATE TABLE Persons(
ID INT not null,
LastName varchar(255) NOT NULL,
FirstName varchar(255) ,
Age int,
City varchar(255),

constraint UC_UNIQUE UNIQUE(ID,LastName)
);
```

لو عايز اعمله في الalter table

```
alter table Persons
add unique(ID);
```

SQL UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

SQL UNIQUE Constraint on CREATE TABLE

The following SQL creates a `UNIQUE` constraint on the "ID" column when the "Persons" table is created:

SQL Server:

```
CREATE TABLE Persons (
  ID int NOT NULL UNIQUE,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int
);
```

To name a `UNIQUE` constraint, and to define a `UNIQUE` constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
   CONSTRAINT UC_Person UNIQUE (ID,LastName)
);
```

SQL UNIQUE Constraint on ALTER TABLE

To create a `UNIQUE` constraint on the "ID" column when the table is already created, use the following SQL:

```
ALTER TABLE Persons
ADD UNIQUE (ID);
```

To name a `UNIQUE` constraint, and to define a `UNIQUE` constraint on multiple columns, use the following SQL syntax:

```
ALTER TABLE Persons
ADD CONSTRAINT UC_Person UNIQUE (ID,LastName);
```

DROP a UNIQUE Constraint

To drop a `UNIQUE` constraint, use the following SQL:

SQL Server :

```
ALTER TABLE Persons
DROP CONSTRAINT UC_Person;
```

SQL Index

ال indexes هيا طريقه لترتيب الداتا في الجدول بطريقه معينه

لما اليوزر بيدخل داتا في الجدول وانا مثلا عامل ال id ك auto increment هلاقيه بيرقم الrecords بالترتيب اللي اليوزر دخلها بيه وال index هنا اسمه cluster index يعني مجمد وبيكون اولويته لل primary key وبيكون اسرع في الاسترجاع لان الداتا مرتبه بحسب ماليوزر دخلها

طيب لو انا عايز ارتبها حسب ال last name مثلا

هنا بتكتب جملة sql لما بتنفذها بيروح يعملك جدول ماتقدر تشوفه ولا توصله وبيحط الداتا فيه مترتبه حسب ال last name

فلما تروح تعمل query تور علي اسم محمد مثلا بيروح عالجدول الجديد اللي هوا عمله اللي الداتا في مرتبه ابجديا حسب ال last name وبيدور علي محمد ويرجعهولك وهنا السرعه بتكون اضعاف السرعه لو كنت استعملت الجدول الأصلي

هنا انا بشوف اكتر اعمده بعمل عليها بحث وبروح احط عليها index

ماينفعش تستحلي الموضوع وتعملها علي كل الاعمده لانه ال index بيبطأ عملية ال update وال insert لانه بيروح يعملها مرتين مره في الجدول الأصلي ومره في الجدول المرتب حسب ال index

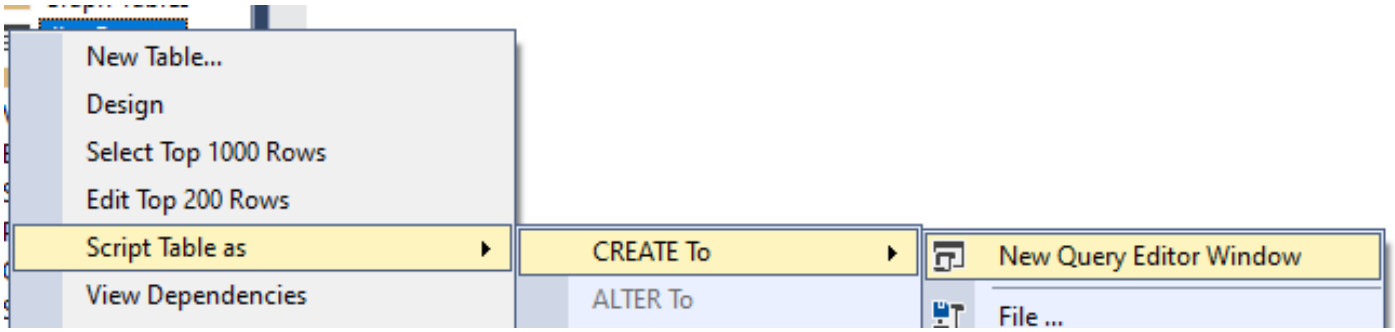
طريقته انك تكتب create index وتكتب اسم ليه براحتك وبعدين تكتب on وبعدها اسم الجدول وتفتح قوسين تحط فيهم اسم العمود او الاعمده

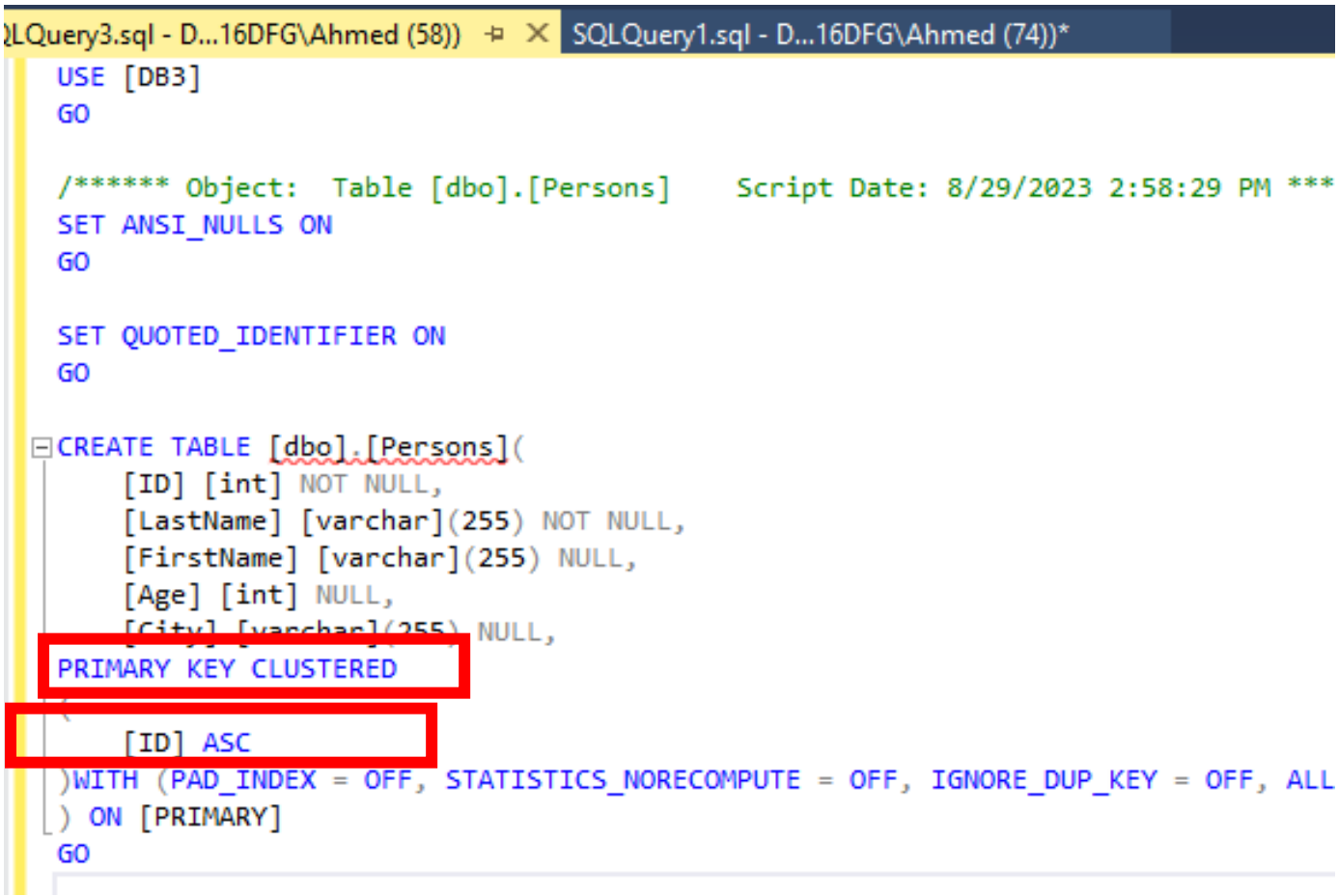
تعالي نعمل جدول

```sql
drop table Persons

CREATE TABLE Persons(
ID INT not null primary key,
LastName varchar(255) NOT NULL,
FirstName varchar(255) ,
Age int,
City varchar(255),
);
```

بمجرد اني حددت ال primary key وشغلت الكود هوا راح عمل ال clustered index

ولو عايز تعملها كليك يمين عالجدول





هنا هوا رتب ال id تصاعديا تقدر تخليه تنازليا وعمله cluster index

تعالي نعمل index لل last name

```sql
create index idx_lastname
on Persons(LastName);
```

عاوزين نعمل واحد تاني للfisrt name والlast name

```sql
CREATE INDEX idx_pname
ON Persons (LastName, FirstName);
```

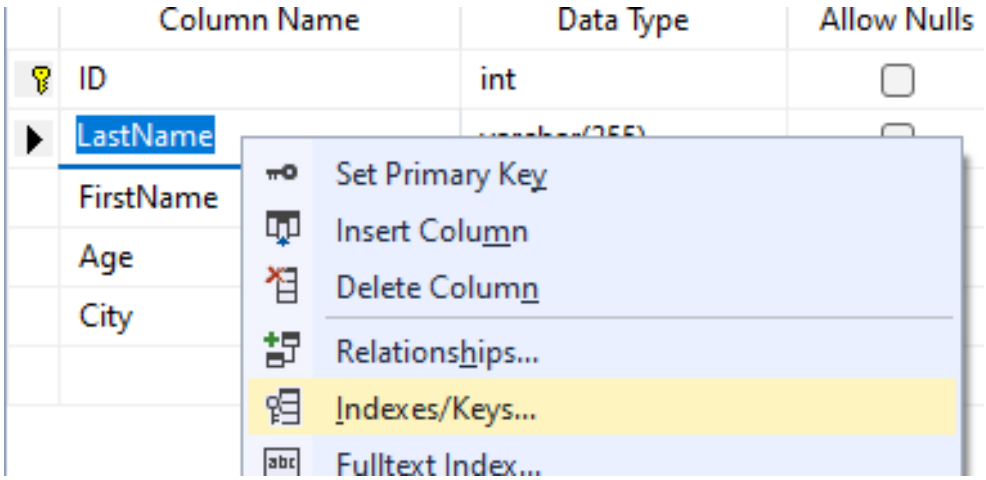
عايز احذف ال index الاولاني

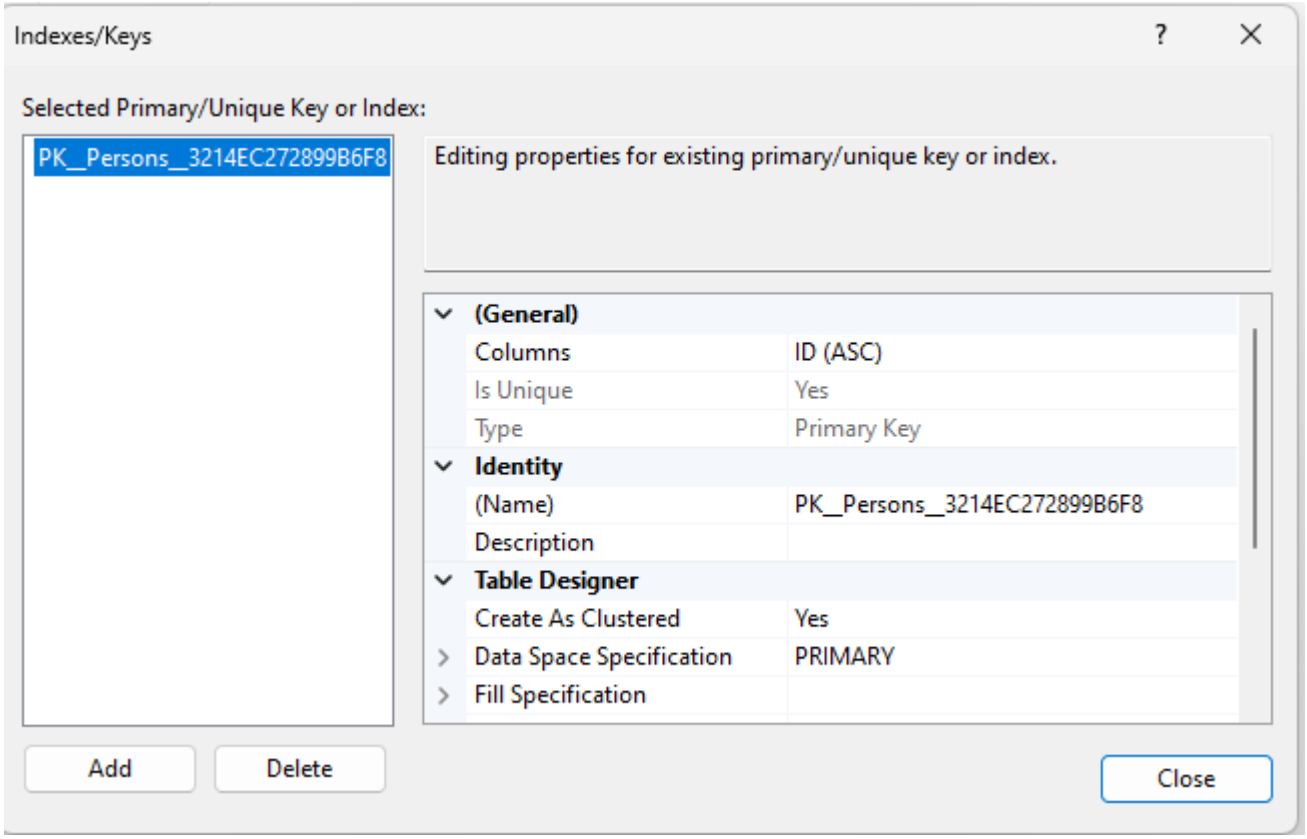
```sql
drop index Persons.idx_lastname
```

عايز احذف التاني

```sql
drop index Persons.idx_pname
```

لو عايز اعمل INDEX عن طريق الماوس





تقدر تعدل وتضيف وتحذف

SQL CREATE INDEX Statement

The `CREATE INDEX` statement is used to create indexes in tables.

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

Note: Updating a table with indexes takes more time than updating a table without (because the indexes also need an update). So, only create indexes on columns that will be frequently searched against.

CREATE INDEX Syntax

Creates an index on a table. Duplicate values are allowed:

CREATE INDEX *index_name*
ON *table_name* (*column1*, *column2*, ...);

CREATE UNIQUE INDEX Syntax

Creates a unique index on a table. Duplicate values are not allowed:

CREATE UNIQUE INDEX *index_name*
ON *table_name* (*column1*, *column2*, ...);

CREATE INDEX Example

The SQL statement below creates an index named "idx_lastname" on the "LastName" column in the "Persons" table:

CREATE INDEX idx_lastname
ON Persons (LastName);

If you want to create an index on a combination of columns, you can list the column names within the parentheses, separated by commas:

CREATE INDEX idx_pname
ON Persons (LastName, FirstName);

DROP INDEX Statement

The `DROP INDEX` statement is used to delete an index in a table.

DROP INDEX *table_name.index_name*;

SQL Server Clustered Index and Primary key constraint

When you create a table with a primary key, SQL Server automatically creates a corresponding clustered index that includes primary key columns.

Clustered Index is much faster than normal Index.

Normalization

What is Normalization?

ال normalization هيا عباره عن تقسيم الداتا لاجزاء اصغر عشان تقدر تتحكم فيها (فرق تسد) عشان مايكونش فيه تكرار او تعقيد في الداتا

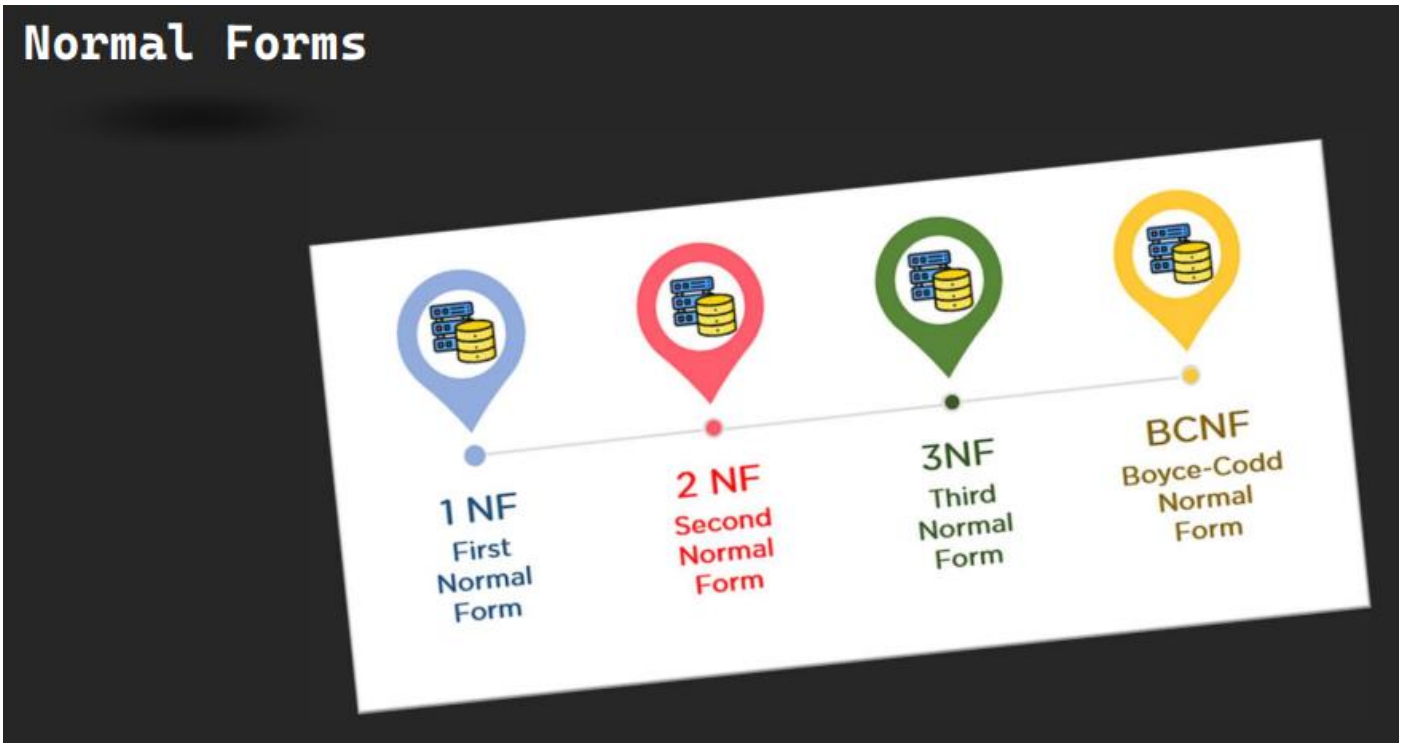
ال normalization بتتحقق عن طريق ال normal forms وده عباره عن مجموعه شروط لو حققتها بتكون حققت ال normalization وده بيتكون من :-

1- First normal form :-
2- Second normal form :-
3- Third normal form :-
4- Boyce codd normal form :- وده يعتبر تطوير لل third normal form

What is Normalization?

- Normalization is the process of organizing data in a database to reduce redundancy and improve data consistency.
- In other words, it is the process of breaking down a larger database into smaller, more manageable pieces, while ensuring that the data is logically organized and free from redundant information.
- Normalization is achieved by applying a set of rules, called Normal Forms, to the database tables. The higher the normal form, the more strictly the rules are applied, and the more normalized the database is.
- The goal of normalization is to eliminate data anomalies, such as duplicate or inconsistent data, which can lead to errors and inconsistencies in the database.
- A normalized database is easier to maintain, update and modify, and can be queried more efficiently.

Normal Forms

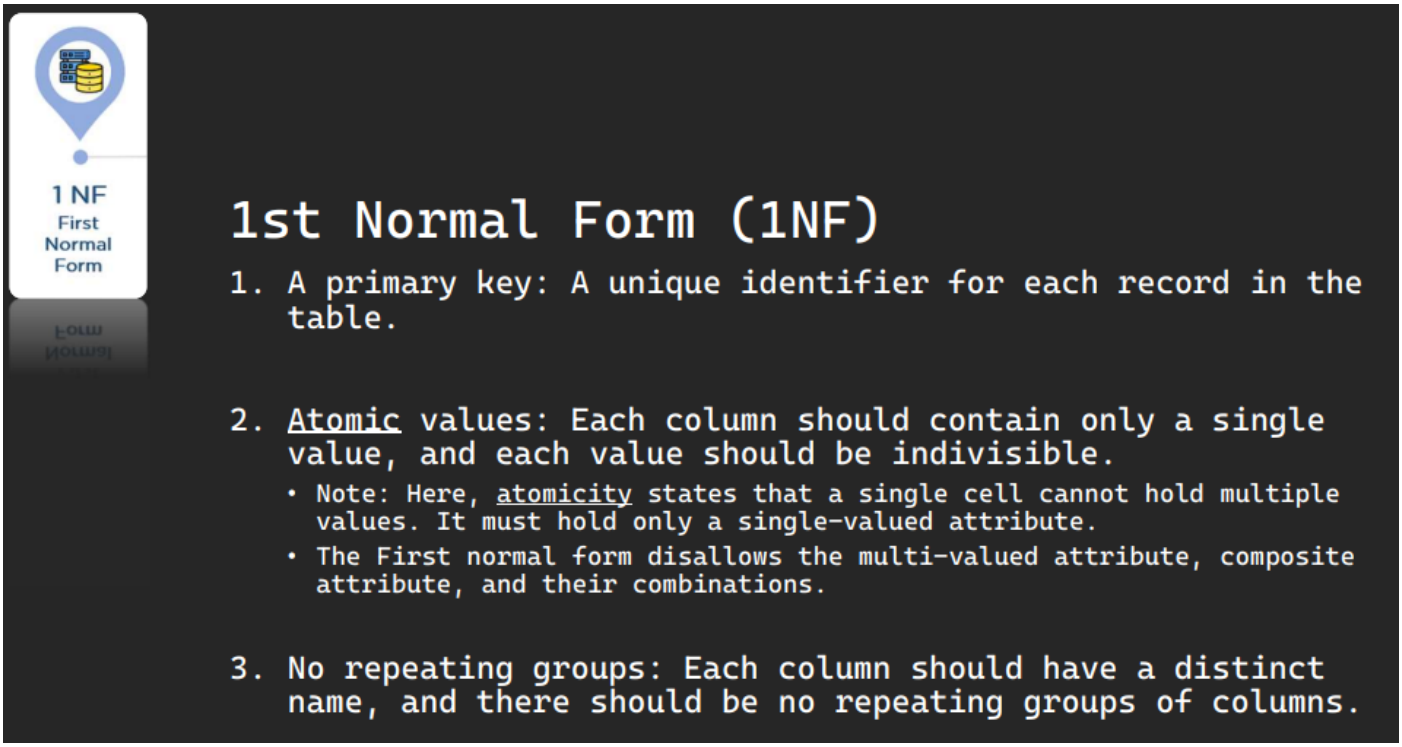


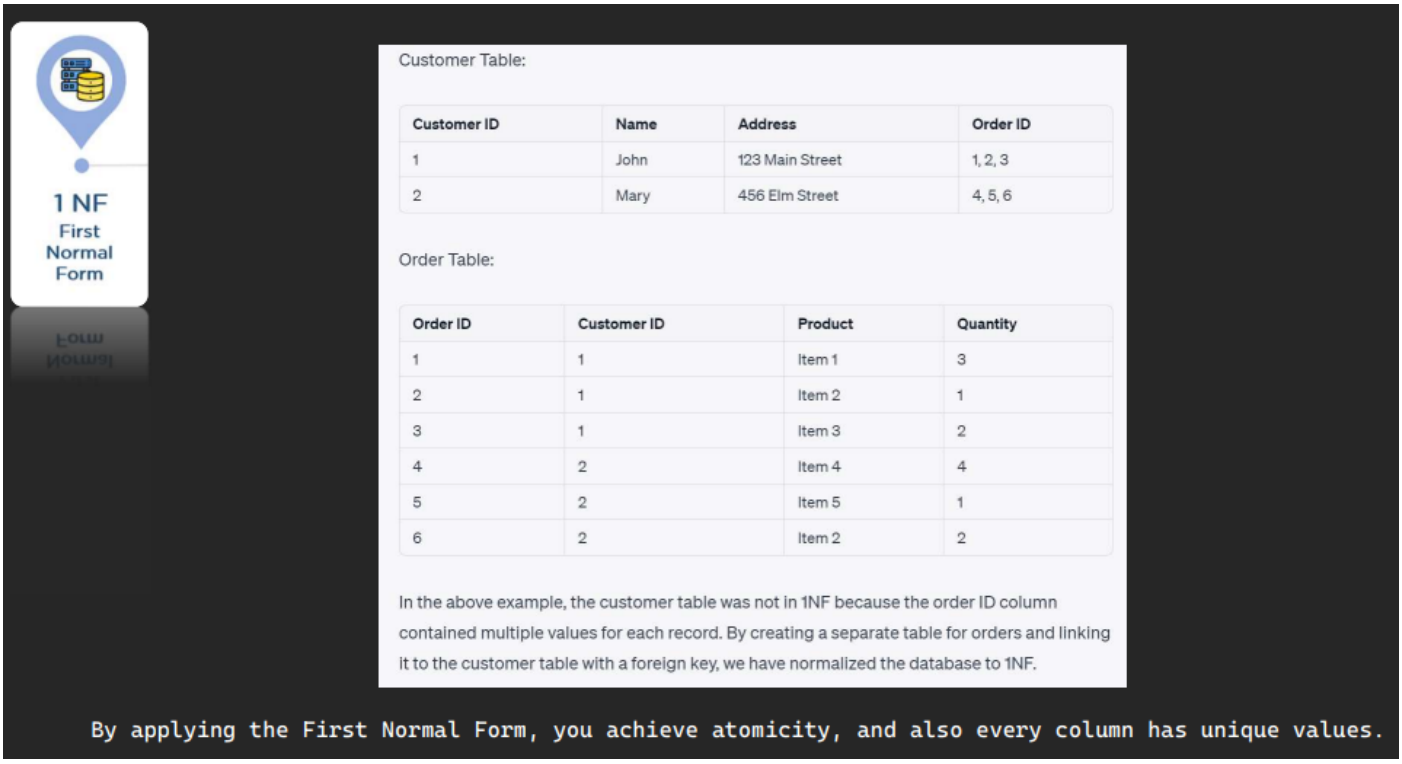
First Normal Form 1NF

هنا بيقولك مافيش أي normal form يقدر يتحقق من غير ماتحقق الشروط بتاعت ال first normal form

الشروط هيا :-

1- لازم يكون عندك primary key لكل record
2- Atomic values :- انه العمود الواحد بيمثل قيمه واحده ماينفعش مثلا تحط رقمين هاتف في نفس المكان
3- كل عمود يكون ليه اسم مايتكررش



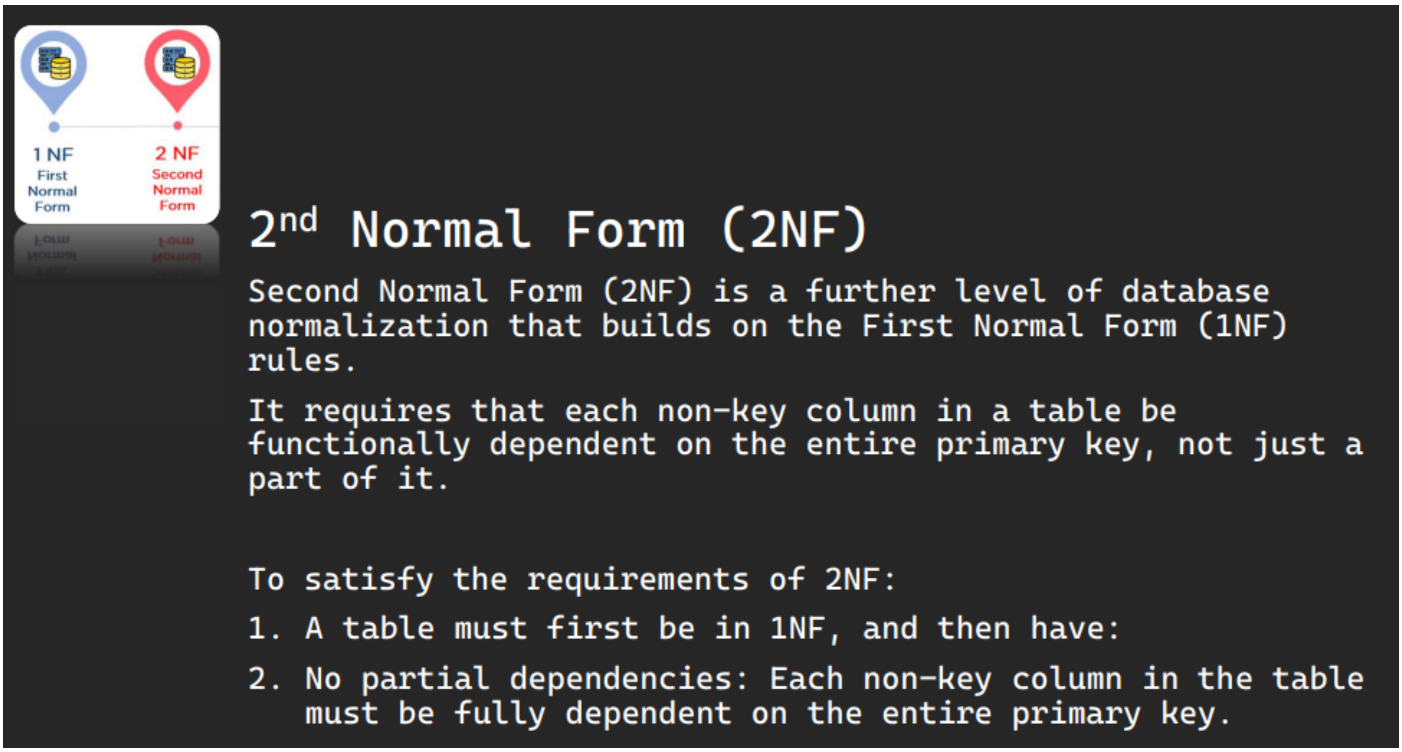


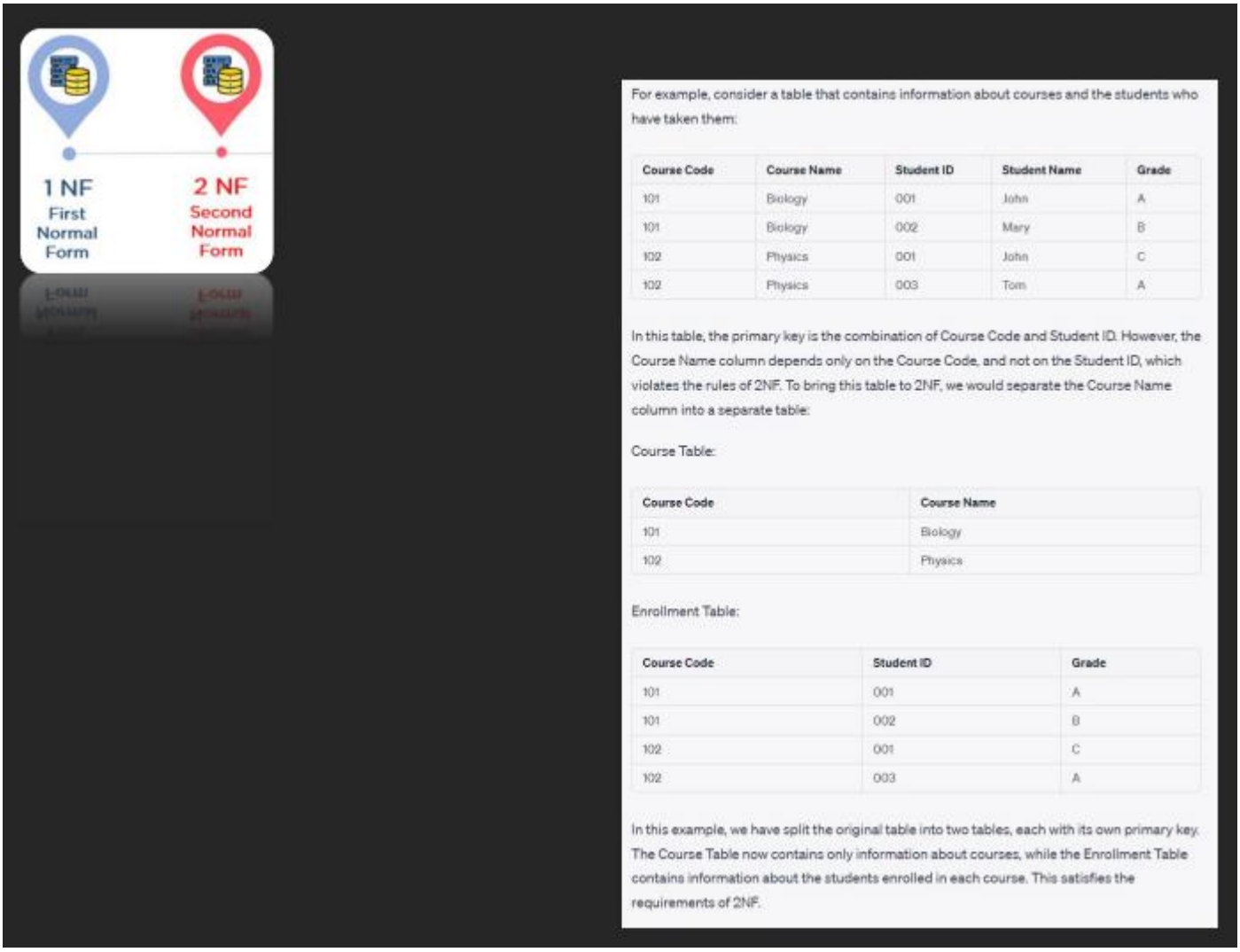
Second Normal Form 2NF

ال second normal form بيعتمد اعتماد كلي علي الfirst normal form

الشروط :-

1- تكون محقق ال first normal form
2- No partial dependencies :- انه كل عمود في الجدول تقدر ترجعه عن طريق primary key خاص بيه وكامل يعني لو فيه primary key معمود من عمودين او عمود واحد او اكتر لازم باقي الاعمده في الجدول تكون معتمده عالمفتاح كله يعني ماينفعش تيجي تقول انه العمود الفلاني اقدر اجيبه من العمود رقم 1 بينما عندك ال primary key معمول من العمودين 1 و 2 اللي هوا شيء غير منطقي





Third Normal Form 3NF

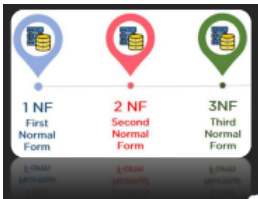
ال third normal form بيعتمد اعتماد كلي علي ال second normal form وبالتالي علي ال first normal form

وهوا level اعلي من الشروط

الشروط :-

1- تكون محقق ال first normal form وال second normal form
2- No transitive dependency :- انه كل عمود يكون معتمد فقط علي ال primary key ولا يعتمد علي عمود اخر

يعني باختصار لو لقيت انه فيه داتا عندك هتتكرر حطها في جدول تاني وكل عمود بيمثل قيمه واحده بس وده اللي كلنا بنعمله



3rd Normal Form (3NF)

Third Normal Form (3NF) is a higher level of database normalization that builds on the rules of First Normal Form (1NF) and Second Normal Form (2NF). It requires that each non-key column in a table be dependent only on the primary key, and not on any other non-key columns.

To satisfy the requirements of 3NF:

1. Table must first be in 1NF and 2NF, and then have:
2. No transitive dependencies: Each non-key column in the table must be dependent only on the primary key, and not on any other non-key columns.

Books

Book ID	Book Title	Author Name	Author Email
1	Pride and Prejudice	Jane Austen	jausten@example.com
2	1948	George Orwell	gorwell@example.com
3	Emma	Jane Austen	jausten@example.com
4	Animal Farm	George Orwell	gorwell@example.com
5	Sense and Sensibility	Jane Austen	jausten@example.com

Authors Table:

Author ID	Author Name	Author Email
1	Jane Austen	jausten@example.com
2	George Orwell	gorwell@example.com

Books Table:

Book ID	Book Title	Author ID
1	Pride and Prejudice	1
2	1984	2
3	Emma	1
4	Animal Farm	2
5	Sense and Sensibility	1

In this example, we have split the original table into two tables, each with its own primary key. The Authors Table now contains only information about authors, while the Books Table contains information about the books and the authors who wrote them. This satisfies the requirements of 3NF.

Boyce Codd normal form (BCNF) , 4NF and 5NF

BCNF , 4NF, and 5 NF will be discussed in Database Level 2.

بالنسبة للNormalization انتم لن تحتاجو اكثر من 3NF باقي الانواع هي وضعت لمعالجة الاخطاء في الديزاين 😊 وهذا الاخطاء انتم لن ترتكبوها لطالما استخدمتم سياسة سياسة فرق تسد

Course Completed

Message

تم بحمد الله الانتهاء من هذا الكورس

لا تنسى مواصلة التدريب بشكل دائم

سيكون هنالك كورس تطبيقي كامل على جميع ما تم اخذه في هذا الكورس

كل التوفيق للجميع