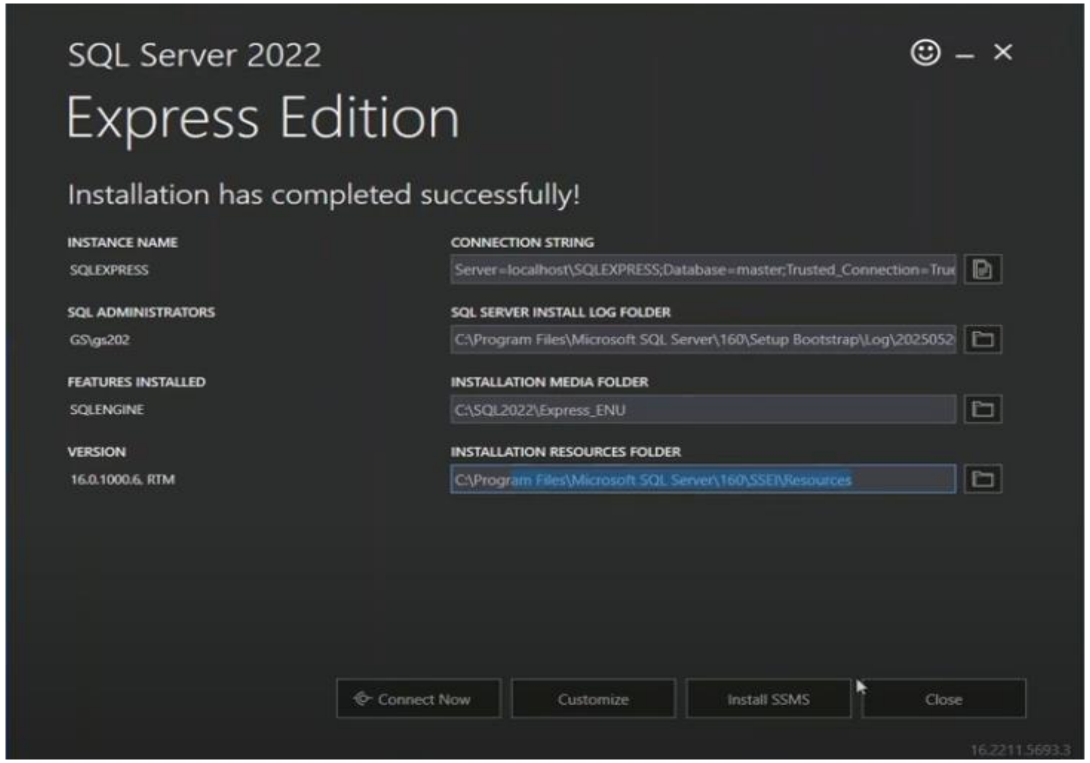
ALL YOU NEED TO KNOW ABOUT SQL

- 12 August 2025

- By Deboleena Thakur, B. Pharm, MS-HCDA

- Intermediate Data Analyst- IDHI, College of Medicine,
- University of Arkansas for Medical Sciences

Install Microsoft SSMS and Database Engine-


SQL Server 2022

Express Edition

Installation has completed successfully!

INSTANCE NAME
SQLEXPRESS

SQL ADMINISTRATORS
GS\gs202

FEATURES INSTALLED
SQLENGINE

VERSION
16.0.1000.6, RTM

CONNECTION STRING
Server=localhost\SQLEXPRESS;Database=master;Trusted_Connection=Tru

SQL SERVER INSTALL LOG FOLDER
C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap\Log\2025052

INSTALLATION MEDIA FOLDER
C:\SQL2022\Express_ENU

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

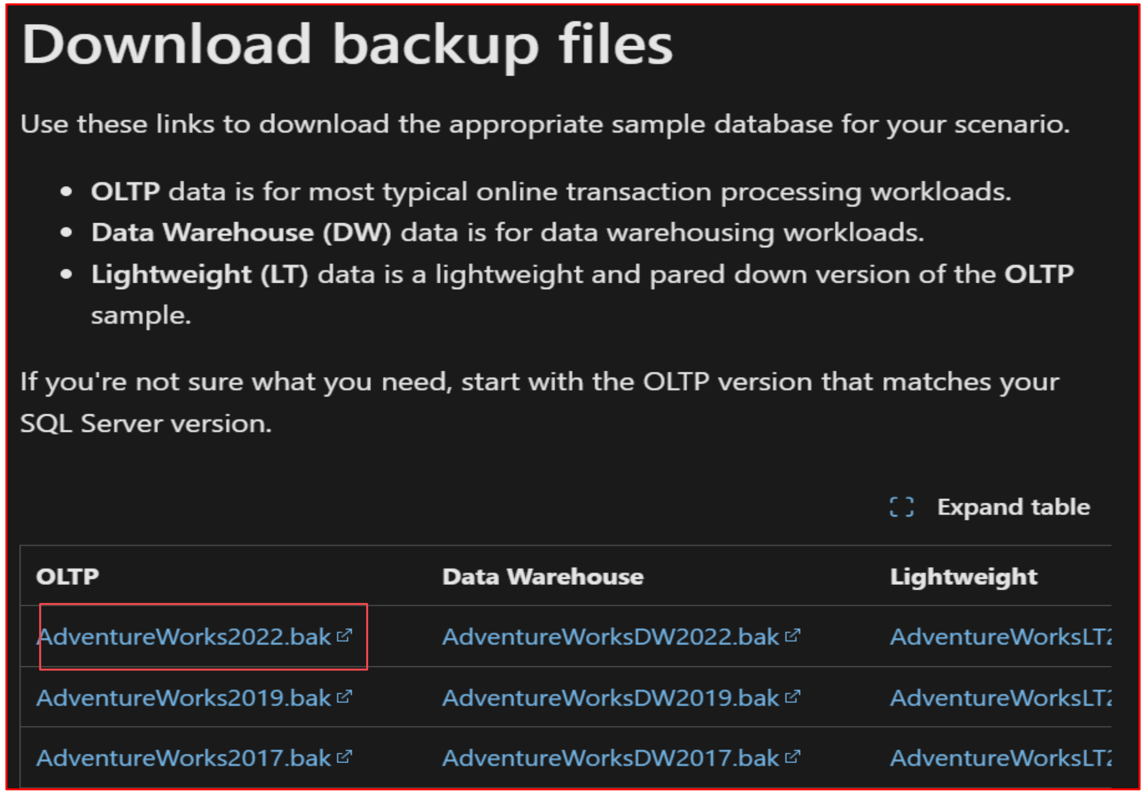
Step 1- For personal use, install a SQL Server database engine like SQL Express 2022- https://www.microsoft.com/en-us/download/details.aspx?id=104781

- SQL Server Express allows you to run a Microsoft SQL Server database locally on your device without needing external servers or special permissions. It works well with SQL Server Management Studio (SSMS), including version 21, for database management.
- Click on 'install SSMS' button at the end of installation OR click on the link below.

Step 2- Download the latest version of Microsoft SQL Server Management Studio (SSMS)- https://learn.microsoft.com/en-us/ssms/install/install

Download the free sample database-

- **Step 3-** Download AdventureWorks sample databases- https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver17&tabs=ssms
- - This is a free sample database by Microsoft compatible with its SQL Server
- You can select either the OLTP or the Data Warehouse version
- **OLTP- Online Transaction Processing, data refers to the transactional data managed by systems designed to handle a high volume of short, frequently updated transactions.



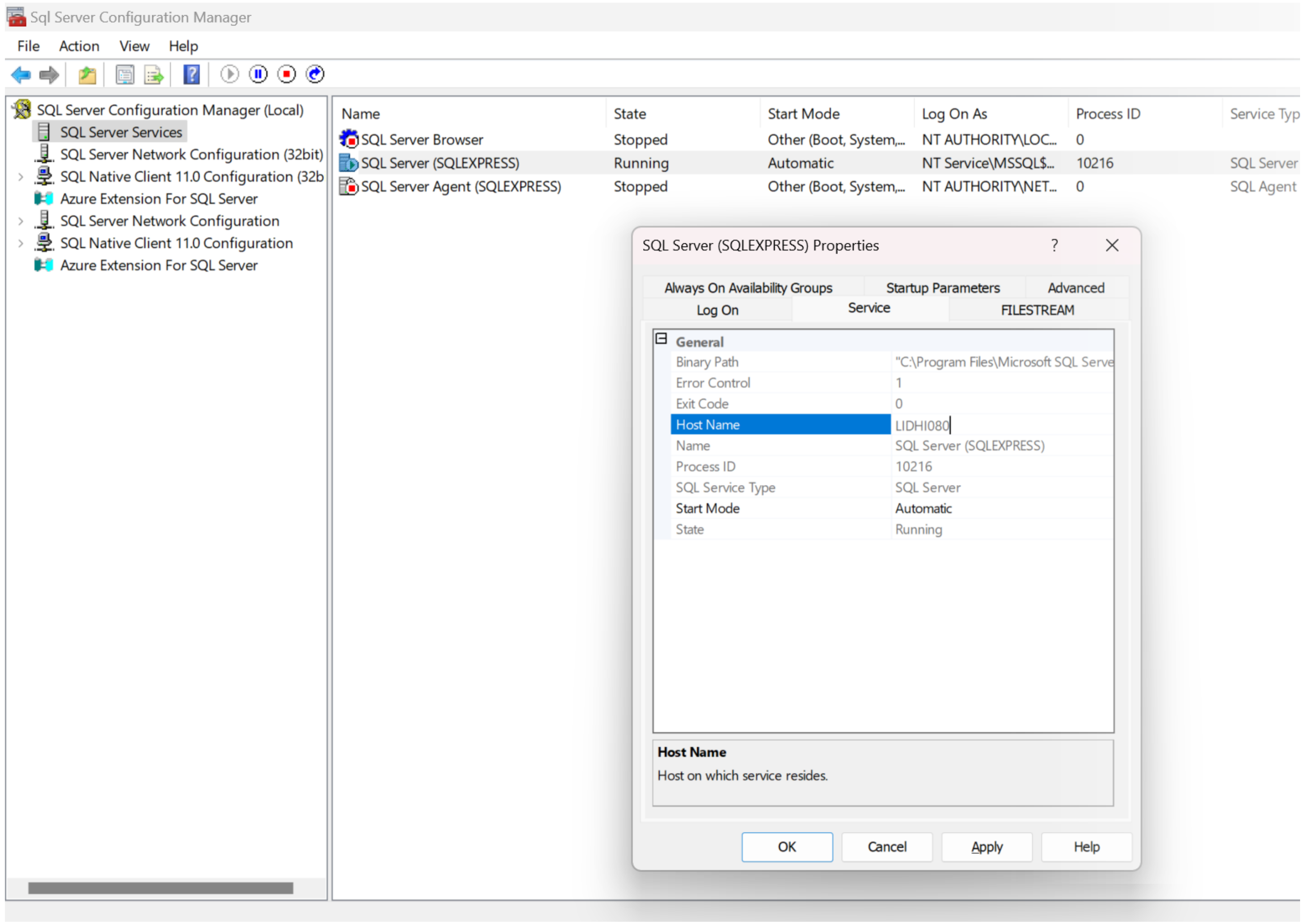
Download backup files

Use these links to download the appropriate sample database for your scenario.

- **OLTP** data is for most typical online transaction processing workloads.
- **Data Warehouse (DW)** data is for data warehousing workloads.
- **Lightweight (LT)** data is a lightweight and pared down version of the **OLTP** sample.

If you're not sure what you need, start with the OLTP version that matches your SQL Server version.

OLTP	Data Warehouse	Lightweight
AdventureWorks2022.bak	AdventureWorksDW2022.bak	AdventureWorksLT2
AdventureWorks2019.bak	AdventureWorksDW2019.bak	AdventureWorksLT2
AdventureWorks2017.bak	AdventureWorksDW2017.bak	AdventureWorksLT2


Sql Server Configuration Manager

SQL Server Configuration Manager (Local)
- SQL Server Services
- SQL Server Network Configuration (32bit)
- SQL Native Client 11.0 Configuration (32b
- Azure Extension For SQL Server
- SQL Server Network Configuration
- SQL Native Client 11.0 Configuration
- Azure Extension For SQL Server

Name	State	Start Mode	Log On As	Process ID	Service Typ
SQL Server Browser	Stopped	Other (Boot, System,...	NT AUTHORITY\LOC...	0	
SQL Server (SQLEXPRESS)	Running	Automatic	NT Service\MSSQL$...	10216	SQL Server
SQL Server Agent (SQLEXPRESS)	Stopped	Other (Boot, System,...	NT AUTHORITY\NET...	0	SQL Agent

SQL Server (SQLEXPRESS) Properties

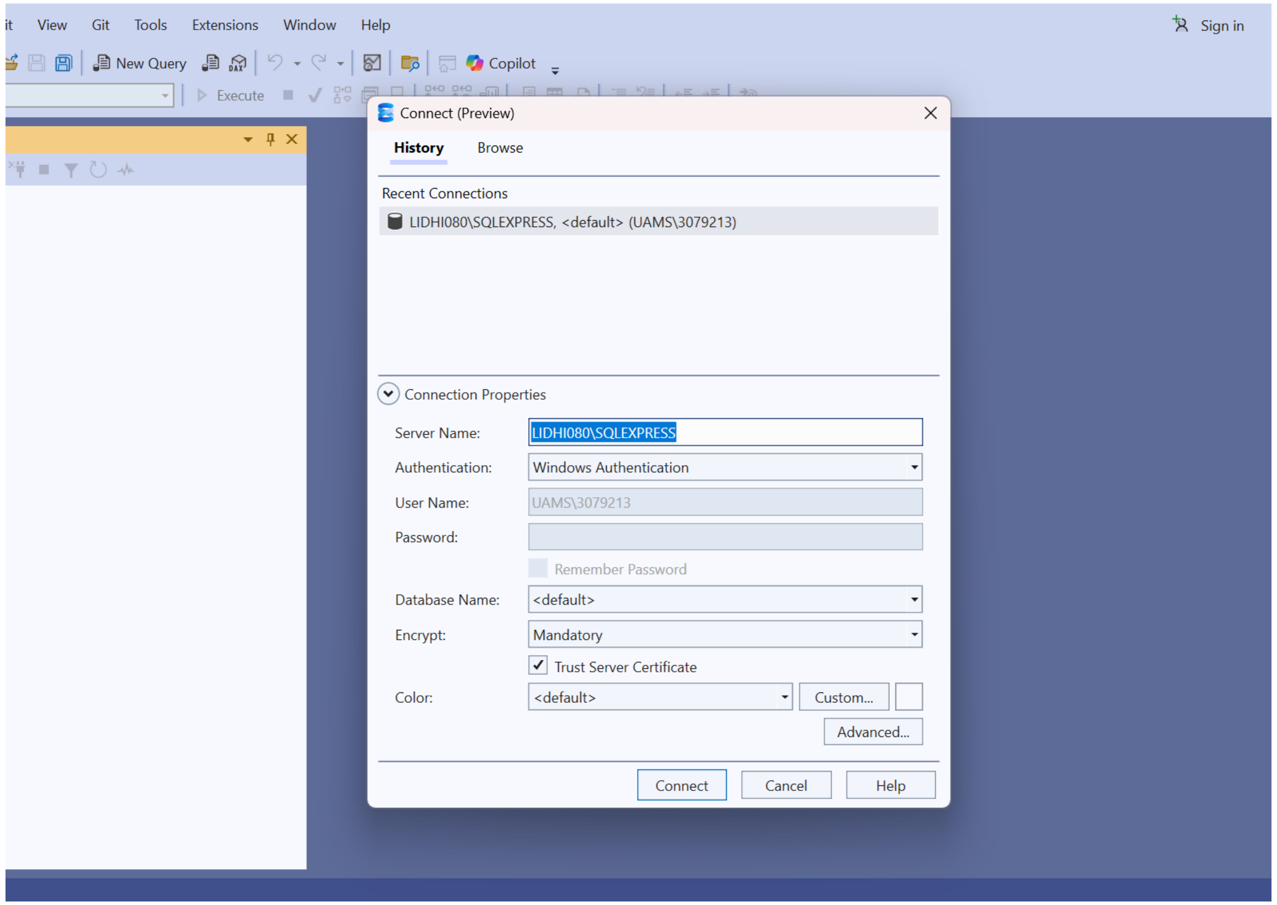
General

Binary Path	"C:\Program Files\Microsoft SQL Serve
Error Control	1
Exit Code	0
Host Name	LIDHI080
Name	SQL Server (SQLEXPRESS)
Process ID	10216
SQL Service Type	SQL Server
Start Mode	Automatic
State	Running

Host Name
Host on which service resides.

Step 4- Open the SSMS tool and connect to personal device's server

- Go to Windows
- Type Sql Server Configuration Manager
- Click on SQL Server Services on the left side panel
- Right click on SQL Server (SQL EXPRESS) and go to Properties
- Copy the Host Name and enter **<host name>\SQLEXPRESS** on to the Server Name on SSMS21 tool.



Recent Connections

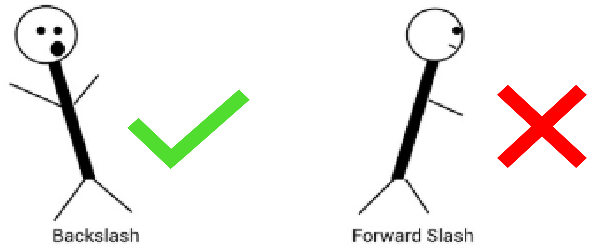
LIDHI080\SQLEXPRESS, <default> (UAMS\3079213)

Connection Properties

Server Name:	LIDHI080\SQLEXPRESS
Authentication:	Windows Authentication
User Name:	UAMS\3079213
Password:	
	Remember Password
Database Name:	<default>
Encrypt:	Mandatory
	✓ Trust Server Certificate
Color:	<default>

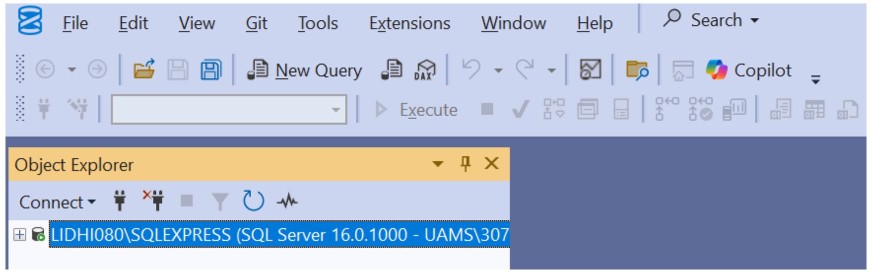
OR

- Simply go to Windows search for System Information
- Copy the System Name and enter <*System Name*>\SQLEXPRESS
- Click Connect.



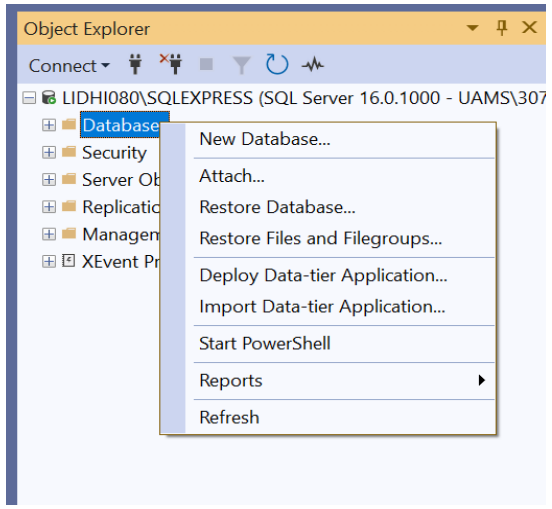
Step 5- Load the **AdventureWorks** database into the RDBMS

-Click on the + next to the device server name



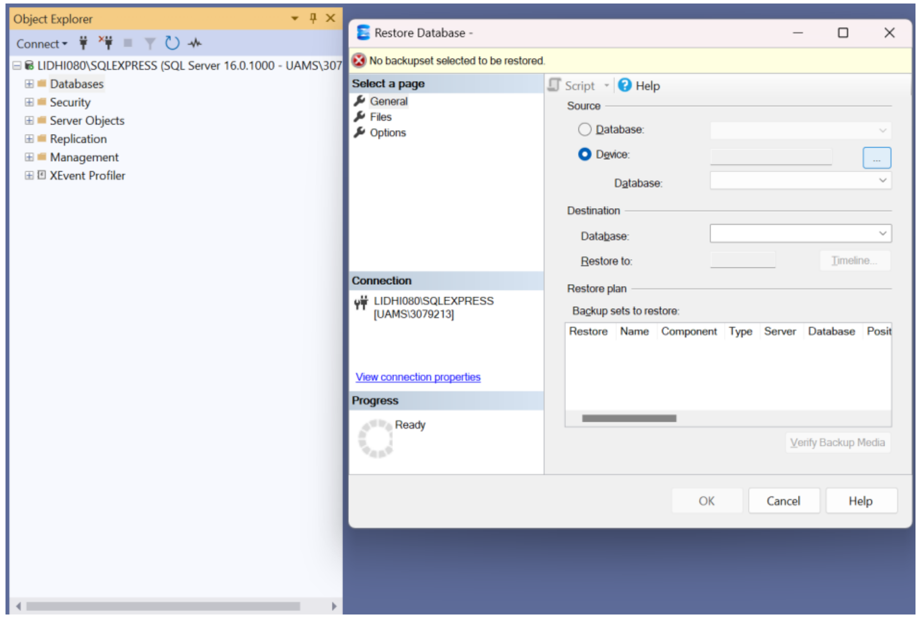
LIDHI080\SQLEXPRESS (SQL Server 16.0.1000 - UAMS\307

-Right click next to +Databases and the Restore Database…



LIDHI080\SQLEXPRESS (SQL Server 16.0.1000 - UAMS\307

-Select the **Device** option and click on the **three dots** and **Add** the file from the path location.
Click **OK**.



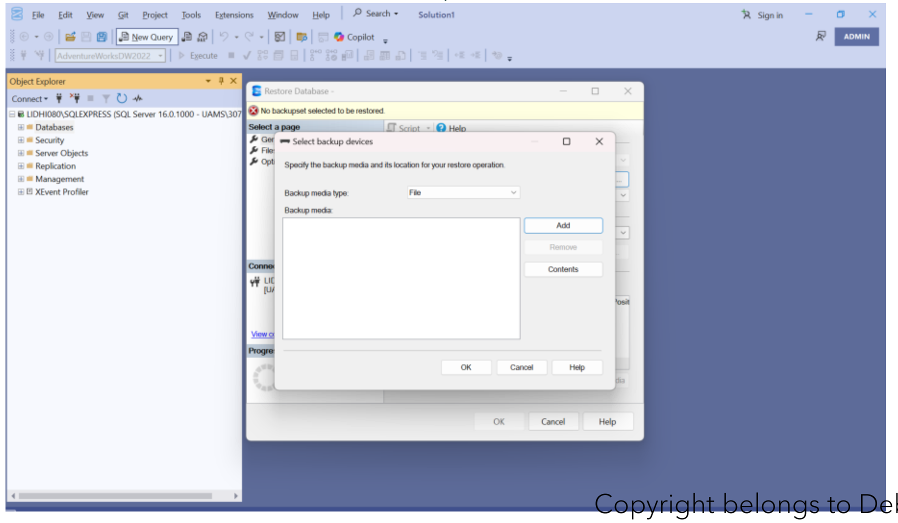
No backupset selected to be restored.

Source: [✓ Device] Database
Destination: Database, Restore to, Timeline...
Restore plan — Backup sets to restore:

Restore	Name	Component	Type	Server	Database	Posit

Connection: LIDHI080\SQLEXPRESS [UAMS\3079213]

Progress: Ready

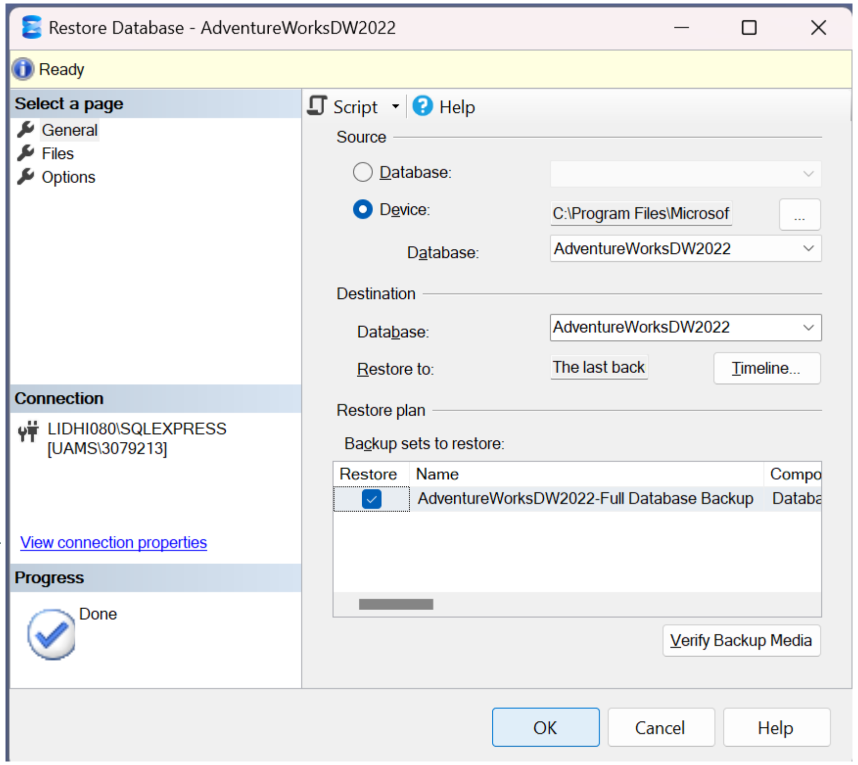


No backupset selected to be restored.

Specify the backup media and its location for your restore operation.

Backup media type: File

Backup media:



Ready

Source
[✓ Device]: C:\Program Files\Microsof
Database: AdventureWorksDW2022

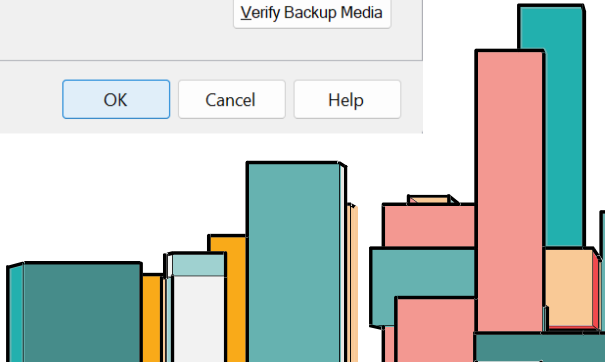
Destination
Database: AdventureWorksDW2022
Restore to: The last back

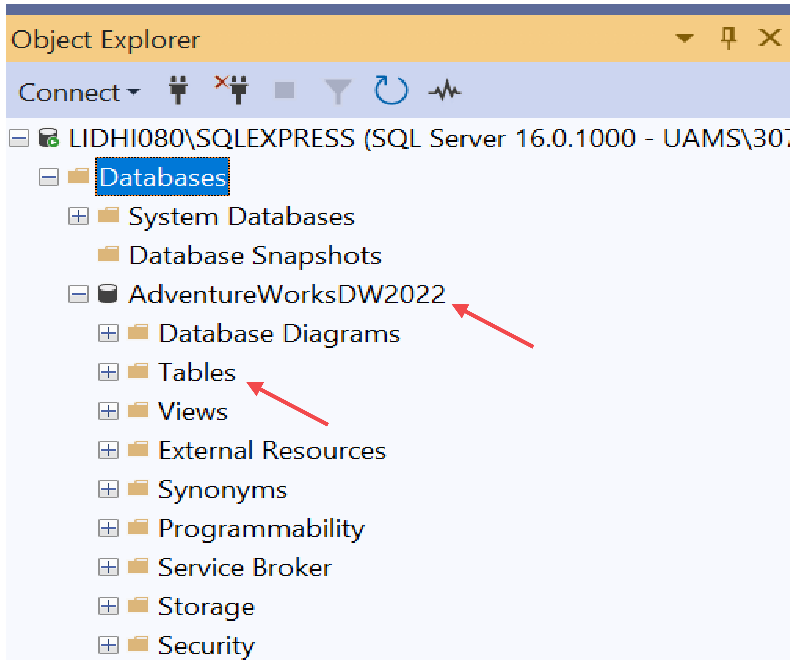
Restore plan
Backup sets to restore:

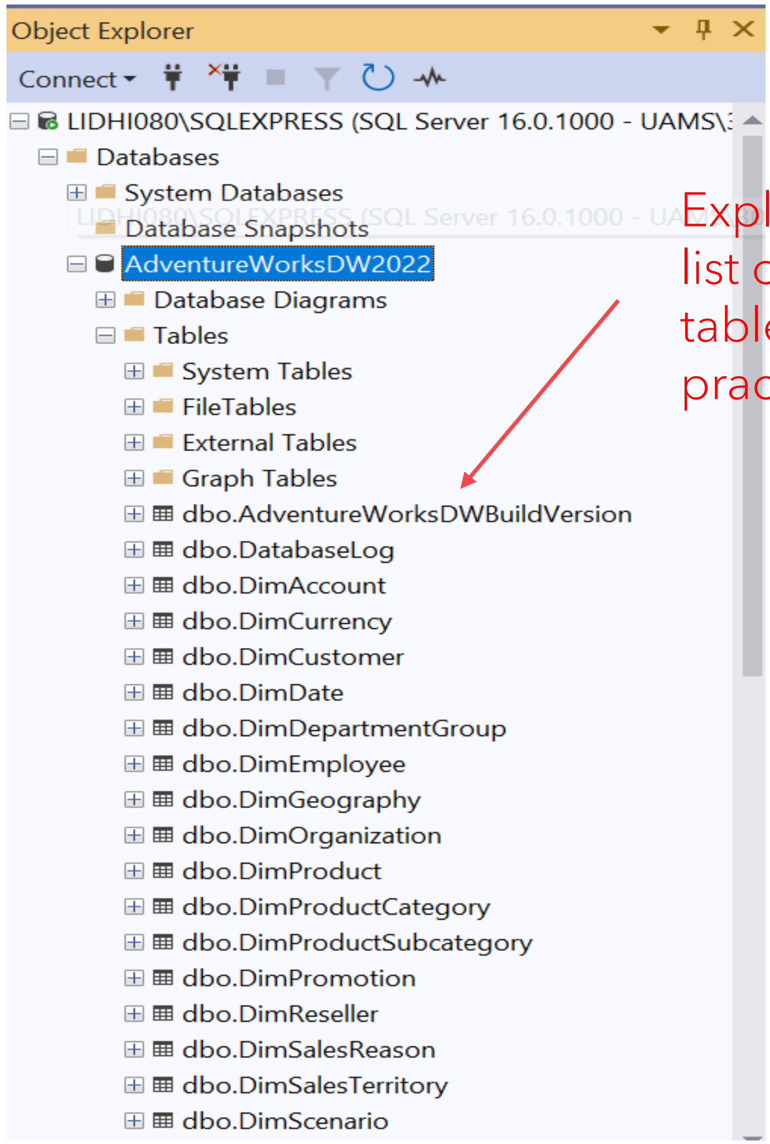
Restore	Name	Compo
✓	AdventureWorksDW2022-Full Database Backup	Databa

Connection: LIDHI080\SQLEXPRESS [UAMS\3079213]

Progress: Done







Explore the list of example tables for practice

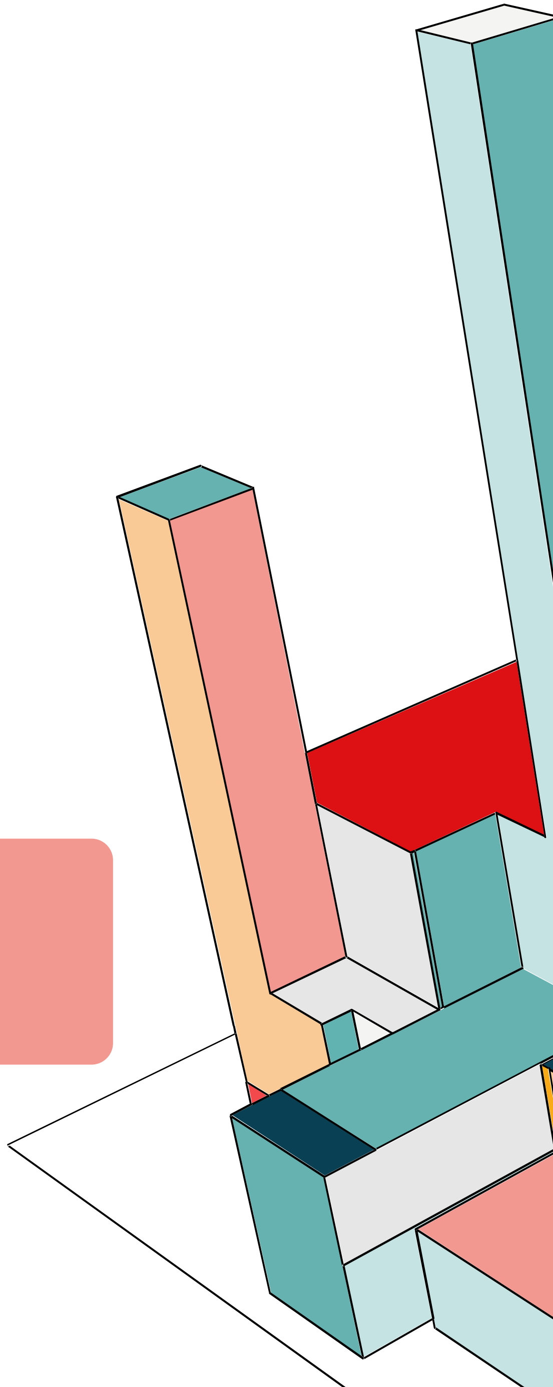
Click on the +AdventureWorksDW2022 and then go to +Tables to check out all the tables under this database.

Course Breakdown-

Basics of SQL

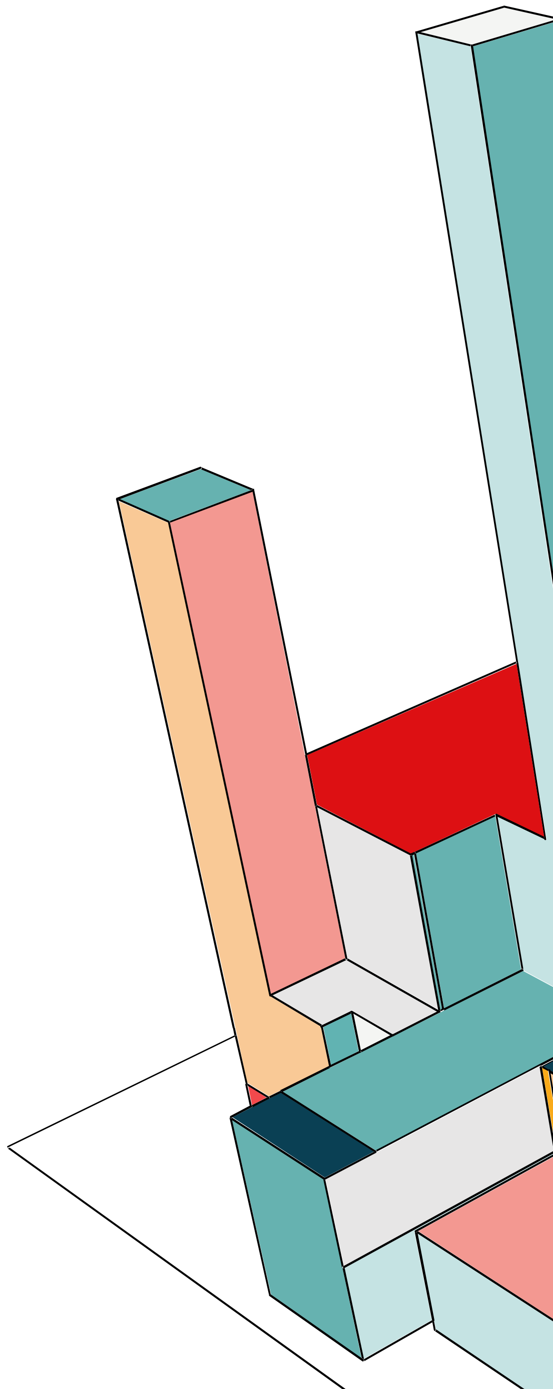
Demonstration of the SQL commands in the RDBMS

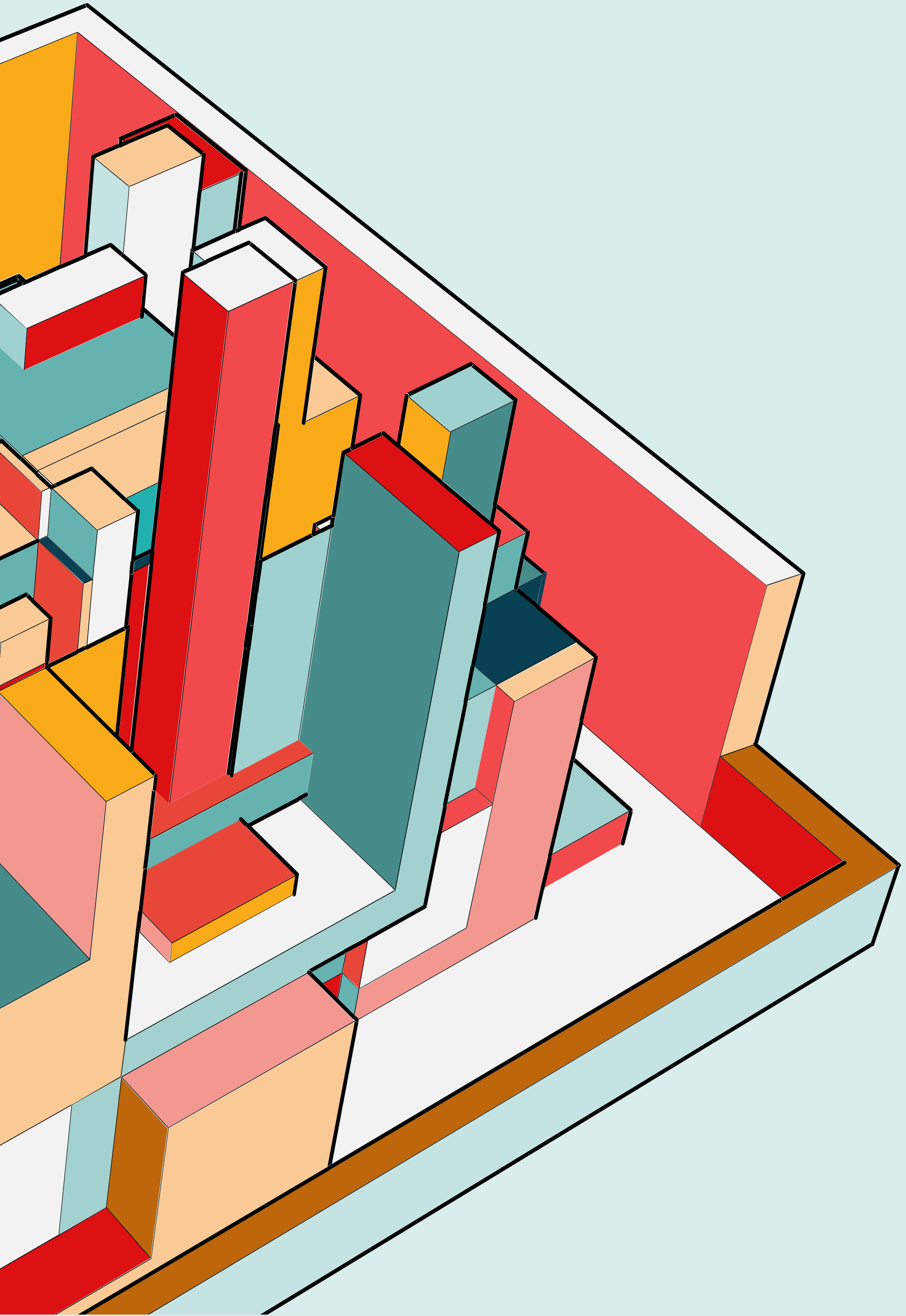
How to use SQL in Research (Project)



Agenda

1. Introduction to SQL
2. Join types
3. Practice SQL commands using Adventure Works database
4. Entity Relationship Diagrams
5. Tiny Exercise
6. Create a database using the Mimic-III Clinical data
7. Epic Cosmos for Data Architecture
8. Q&A





INTRODUCTION TO SQL

What is SQL?

SQL (Structured Query Language) is a standardized language used to manage and interact with data in **relational databases** (e.g., SQL Server, MySQL, PostgreSQL, Oracle).

- **Brief History-**

-SQL was created in the 1970s by IBM
-adopted commercially by Oracle in 1979
-standardized by American National Standards Institute (ANSI) in 1986.

- **Key SQL Functions:**

-DDL (Data Definition Language): Create, alter, or drop tables and structures.
-DML (Data Manipulation Language): Insert, update, delete, and query data.
-DCL (Data Control Language): Grant or revoke user access and permissions.

•SQL is **declarative** language- you describe *what* you want, not *how* to do it.

•While syntax may vary slightly across systems, core SQL commands are consistent.

SQL Order of Writing vs. Order of Execution

1. Order of Writing (How You Write the Query)

SELECT column
FROM table
WHERE condition
GROUP BY column
HAVING condition
ORDER BY column
LIMIT number;

2. Order of Execution (How SQL Actually Processes It)

1. FROM → Identify the tables and join them if needed.
2. WHERE → Filter rows that meet the condition.
3. GROUP BY → Group the remaining rows.
4. HAVING → Filter groups based on conditions.
5. SELECT → Choose the columns or calculations to output.
6. ORDER BY → Sort the final results.
7. LIMIT / OFFSET (if used) → Return only the required number of rows.

"Silly Frog Wear Green Hat On Lake"

Some common terminologies used–

•**Syntax** → The *rules and structure* for writing SQL statements (like grammar in a language).
Example: SELECT column_name FROM table_name WHERE condition; is the syntax pattern.

•**Command** → A specific SQL *instruction* that tells the database what to do.
Example: SELECT * FROM Customers; is a command to retrieve all customer data.

•**Clause** → A *component* or *section* of an SQL command that performs a specific task.
Example: In SELECT Name FROM Customers WHERE Country = 'USA';, the WHERE Country = 'USA' part is a clause that filters results.

Data types: SQL Server (SSMS) vs. SQLite

Usage	SQL Server (SSMS) Data Type	SQLite Data Type	Example Usage
Whole numbers	INT, BIGINT, SMALLINT, TINYINT	INTEGER	Storing counts, IDs (CustomerID = 101)
Decimal numbers	DECIMAL(p,s), NUMERIC(p,s)	REAL (for floating point), NUMERIC (for fixed precision)	Storing prices (Price = 19.99), weights
Floating point	FLOAT, REAL	REAL	Scientific values, measurements
Fixed-length text	CHAR(n)	TEXT	Country codes ('US')
Variable-length text	VARCHAR(n), NVARCHAR(n), TEXT	TEXT	Names, descriptions
Large text	TEXT (deprecated: NTEXT)	TEXT	Articles, comments
Date and time	DATE, DATETIME, DATETIME2, TIME	TEXT (ISO format), REAL (Julian), or INTEGER (Unix timestamp)	Birthdates, timestamps
Boolean values	BIT	No dedicated type – use INTEGER (0/1)	Yes/No flags (IsActive = 1)
Binary data	VARBINARY, BINARY, IMAGE (deprecated)	BLOB	Storing images, files
Unique identifier	UNIQUEIDENTIFIER (GUID)	TEXT	UUIDs for unique records
Money values	MONEY, SMALLMONEY	No dedicated type – use NUMERIC or REAL	Product prices, salaries
Arrays / JSON	Not native – use NVARCHAR(MAX) or JSON (in Azure SQL)	TEXT (store JSON)	Storing structured JSON data

Notes:

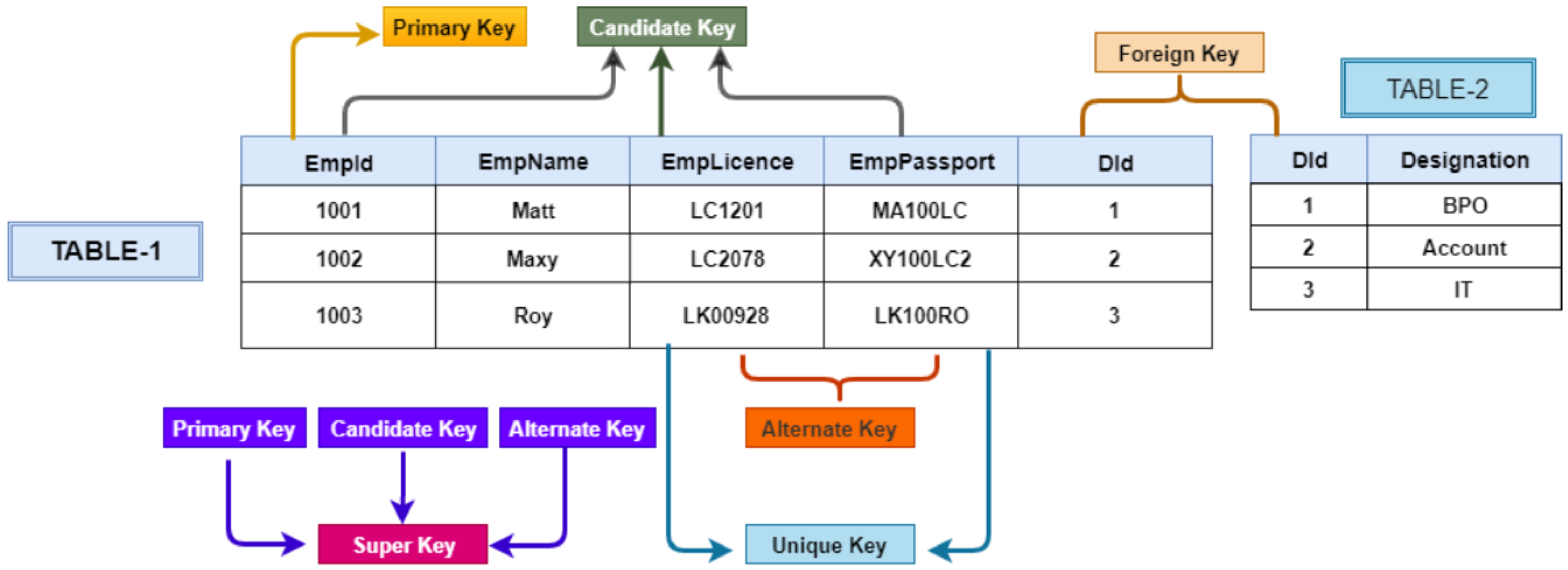
•SQLite doesn't have strict types like SQL Server, and it uses something called *type affinity*, meaning the type you declare is more flexible.

•SQL Server strictly enforces data types, while SQLite is more relaxed, which means you could accidentally store different kinds of data in the same column if you're not careful.

•In SQLite, the main data type categories are TEXT, REAL, INTEGER, BLOB, and NUMERIC.

Important terms

1. Database Structure
•**Database** – A structured collection of data.
•**Table** – A collection of rows and columns storing related data.
•**Record/Tuple** – A single data entry in a table or the row.
•**Field/Attribute** – A specific category of data within a table or the column.
•**Schema** – Blueprint or structure of the database.
•**Index** – Improves search speed in a table.
•**View** – A saved query that displays data from one or more tables.



EmpId	EmpName	EmpLicence	EmpPassport	DId
1001	Matt	LC1201	MA100LC	1
1002	Maxy	LC2078	XY100LC2	2
1003	Roy	LK00928	LK100RO	3

DId	Designation
1	BPO
2	Account
3	IT

2. Keys & Relationships

•Primary Key – Unique identifier for a row in a table.
•Foreign Key – A field linking to the primary key of another table.
•Composite Key – Combination of two or more columns used as a unique identifier.
•Candidate Key – Possible columns that could serve as a primary key.
•Unique Key – Ensures all values in a column are unique.

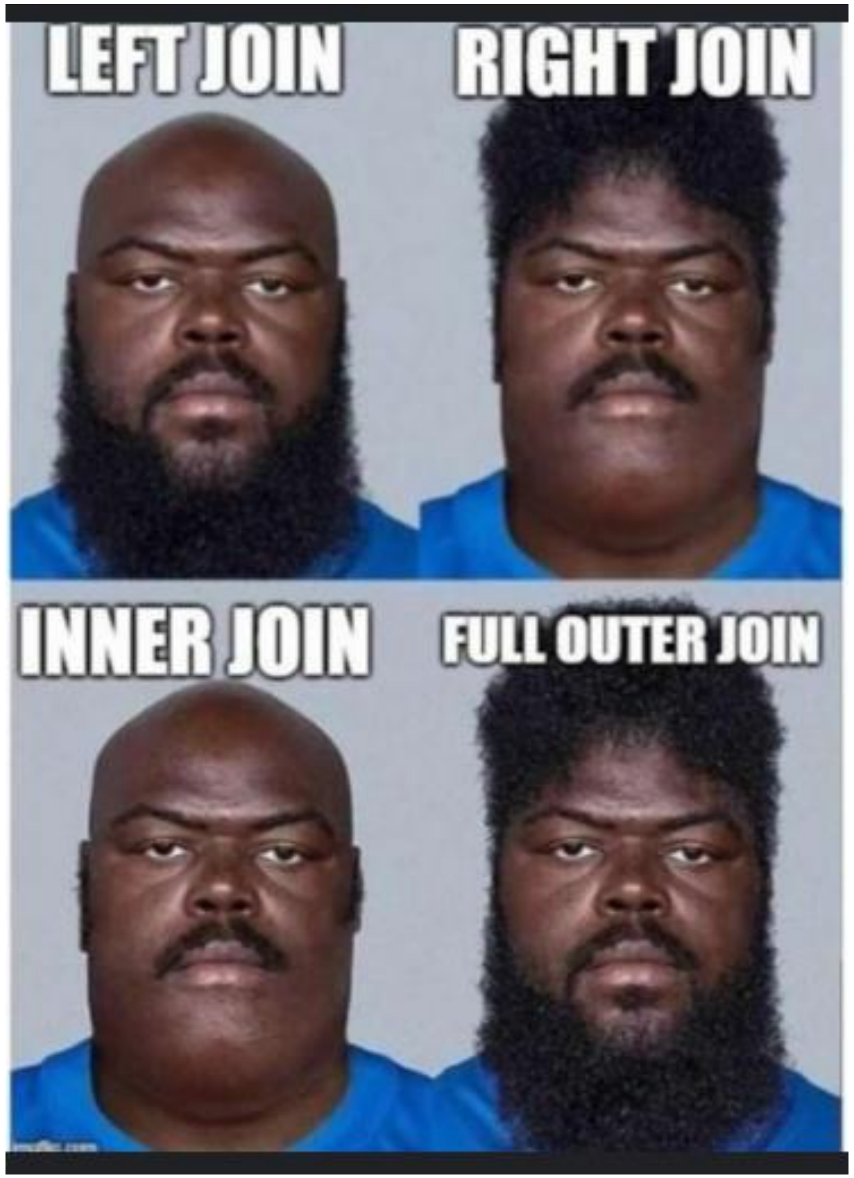
3. SQL Commands

- **DDL (Data Definition Language)** – CREATE, ALTER, DROP, TRUNCATE (structure-related commands).
- **DML (Data Manipulation Language)** – SELECT, INSERT, UPDATE, DELETE (data-related commands).
- **DCL (Data Control Language)** – GRANT, REVOKE (permissions).
- **TCL (Transaction Control Language)** – COMMIT, ROLLBACK, SAVEPOINT (transactions).

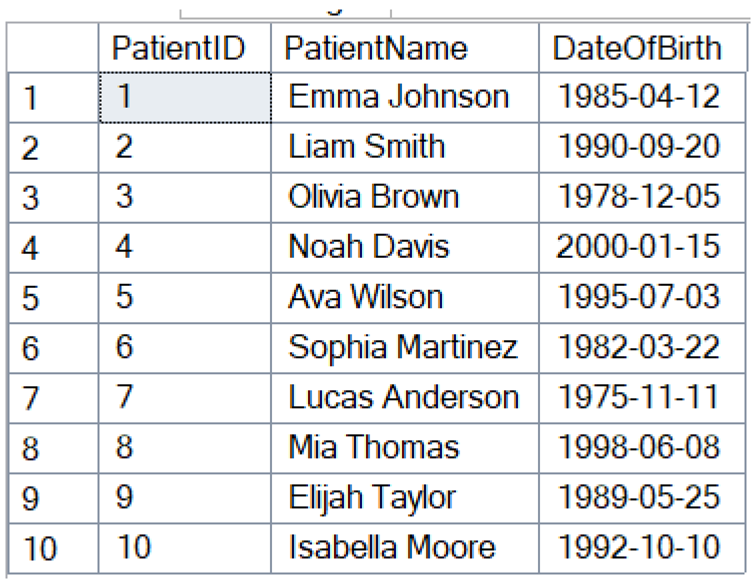
4. Data & Query Concepts

- **Data Type** – Defines the type of data (INT, VARCHAR, DATE, BOOLEAN).
- **NULL** – Represents missing or unknown data.
- **Constraint** – Rules applied to data (NOT NULL, CHECK, DEFAULT).
- **Join** – Combines rows from two or more tables (INNER, LEFT, RIGHT, FULL).
- **Subquery** – A query inside another query.
- **Alias** – Temporary name for a table or column.

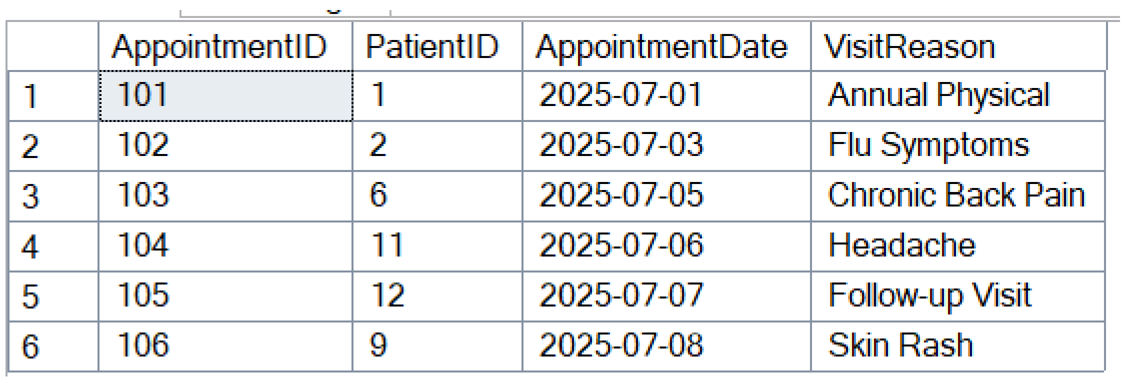
JOIN TYPES



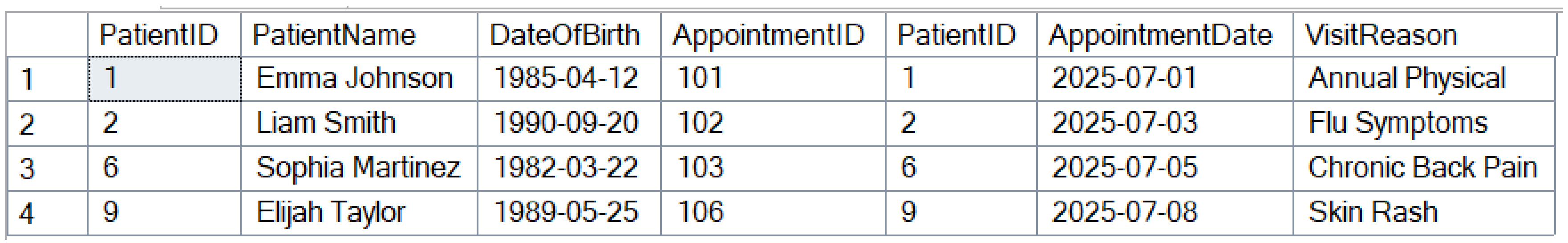
INNER JOIN



	PatientID	PatientName	DateOfBirth
1	1	Emma Johnson	1985-04-12
2	2	Liam Smith	1990-09-20
3	3	Olivia Brown	1978-12-05
4	4	Noah Davis	2000-01-15
5	5	Ava Wilson	1995-07-03
6	6	Sophia Martinez	1982-03-22
7	7	Lucas Anderson	1975-11-11
8	8	Mia Thomas	1998-06-08
9	9	Elijah Taylor	1989-05-25
10	10	Isabella Moore	1992-10-10



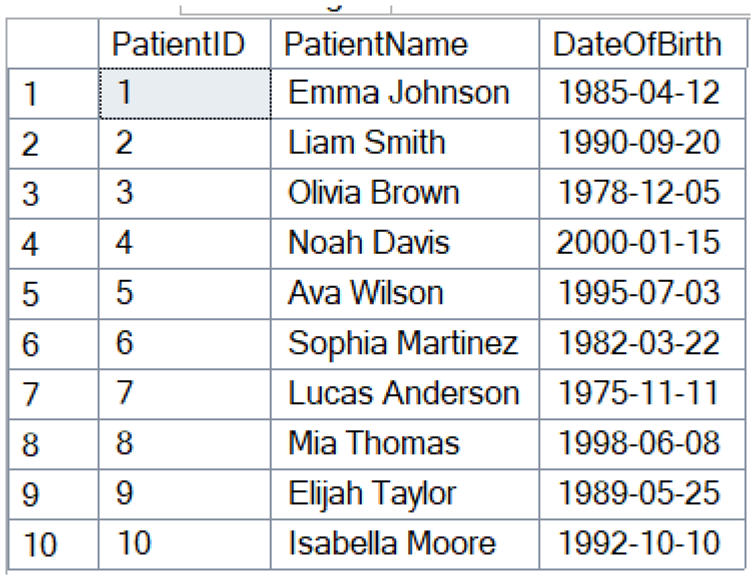
	AppointmentID	PatientID	AppointmentDate	VisitReason
1	101	1	2025-07-01	Annual Physical
2	102	2	2025-07-03	Flu Symptoms
3	103	6	2025-07-05	Chronic Back Pain
4	104	11	2025-07-06	Headache
5	105	12	2025-07-07	Follow-up Visit
6	106	9	2025-07-08	Skin Rash



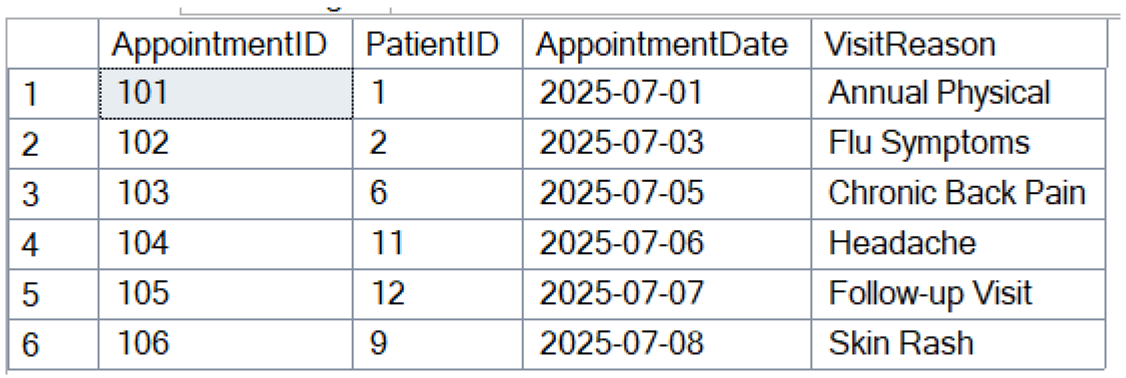
	PatientID	PatientName	DateOfBirth	AppointmentID	PatientID	AppointmentDate	VisitReason
1	1	Emma Johnson	1985-04-12	101	1	2025-07-01	Annual Physical
2	2	Liam Smith	1990-09-20	102	2	2025-07-03	Flu Symptoms
3	6	Sophia Martinez	1982-03-22	103	6	2025-07-05	Chronic Back Pain
4	9	Elijah Taylor	1989-05-25	106	9	2025-07-08	Skin Rash

Şḥộẋș ộŋľỳ ţḩê ņăţçḩêđ șộẋș xḥêsê ţḩê ŘăţîêŋţÍĐ êỳîșţș îŋ čộţḩ ţḩê Řăţîêŋţș ţăčľê ăŋđ ţḩê Ăřřộîŋţņêŋţș ţăčľê

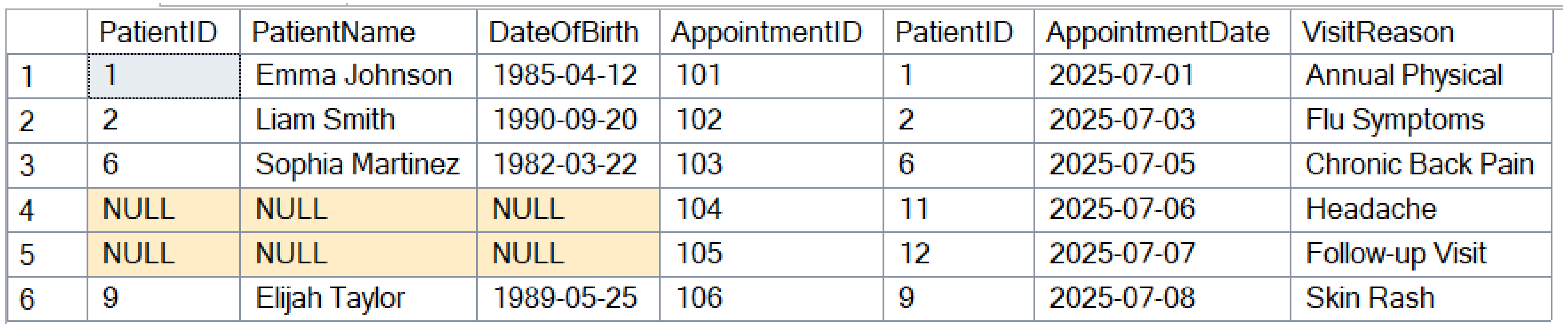
RIGHT JOIN



	PatientID	PatientName	DateOfBirth
1	1	Emma Johnson	1985-04-12
2	2	Liam Smith	1990-09-20
3	3	Olivia Brown	1978-12-05
4	4	Noah Davis	2000-01-15
5	5	Ava Wilson	1995-07-03
6	6	Sophia Martinez	1982-03-22
7	7	Lucas Anderson	1975-11-11
8	8	Mia Thomas	1998-06-08
9	9	Elijah Taylor	1989-05-25
10	10	Isabella Moore	1992-10-10

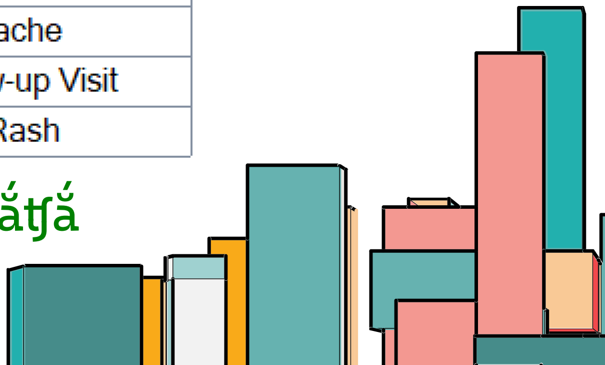


	AppointmentID	PatientID	AppointmentDate	VisitReason
1	101	1	2025-07-01	Annual Physical
2	102	2	2025-07-03	Flu Symptoms
3	103	6	2025-07-05	Chronic Back Pain
4	104	11	2025-07-06	Headache
5	105	12	2025-07-07	Follow-up Visit
6	106	9	2025-07-08	Skin Rash

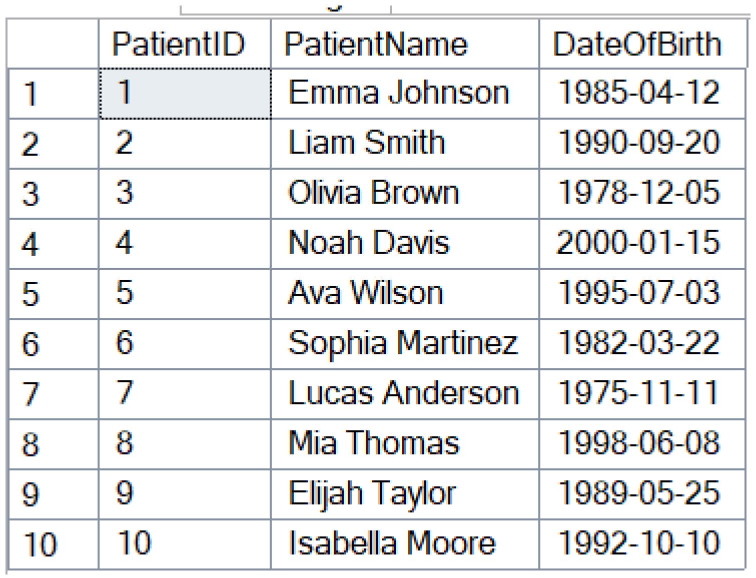


	PatientID	PatientName	DateOfBirth	AppointmentID	PatientID	AppointmentDate	VisitReason
1	1	Emma Johnson	1985-04-12	101	1	2025-07-01	Annual Physical
2	2	Liam Smith	1990-09-20	102	2	2025-07-03	Flu Symptoms
3	6	Sophia Martinez	1982-03-22	103	6	2025-07-05	Chronic Back Pain
4	NULL	NULL	NULL	104	11	2025-07-06	Headache
5	NULL	NULL	NULL	105	12	2025-07-07	Follow-up Visit
6	9	Elijah Taylor	1989-05-25	106	9	2025-07-08	Skin Rash

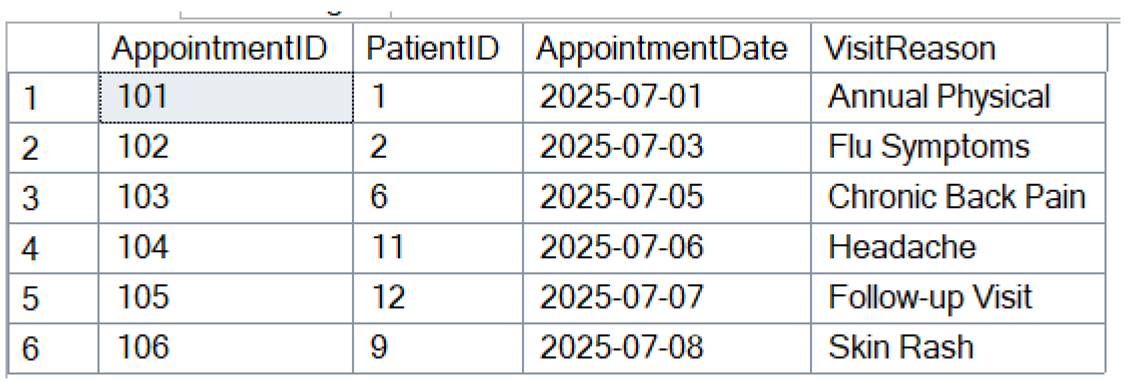
Ķêêřș ắľľ șộxș ğsộṇ ţḥê Ařřộîŋţṇêŋţș ţắčľê ắŋđ ắđđș đắţắ ğsộṇ ţḥê Rắţîêŋţș ţắčľê ộŋľỳ xḥêŋ ţḥêsê ș ắ ṇắţçḥ



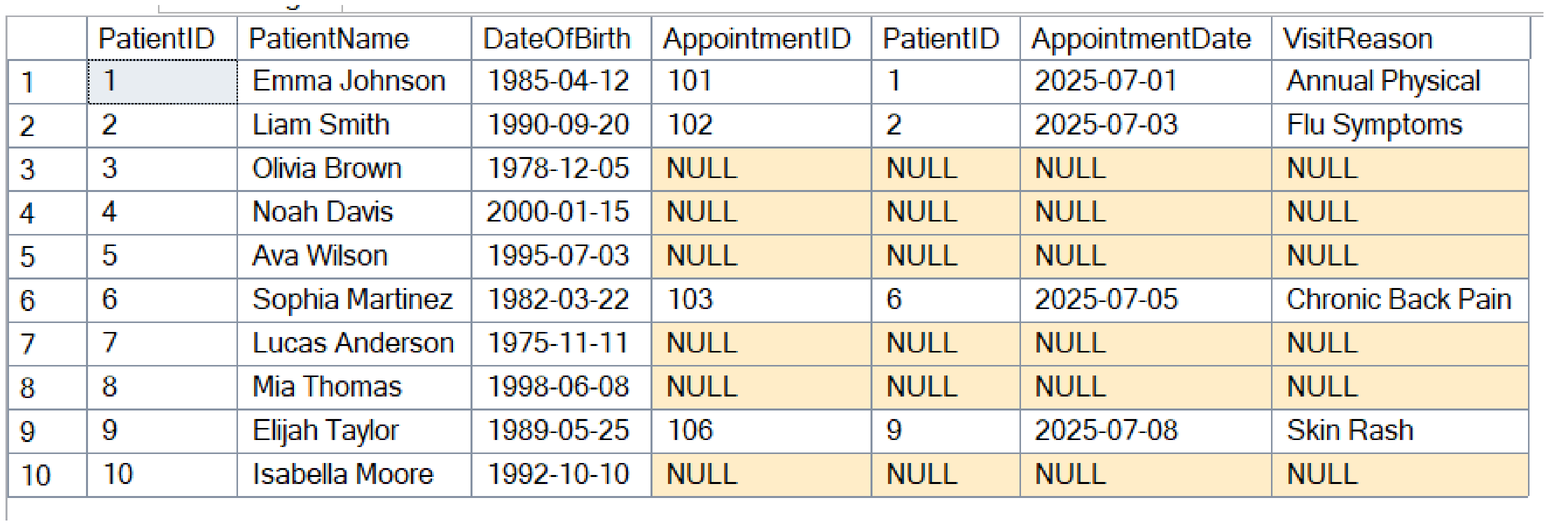
LEFT JOIN



	PatientID	PatientName	DateOfBirth
1	1	Emma Johnson	1985-04-12
2	2	Liam Smith	1990-09-20
3	3	Olivia Brown	1978-12-05
4	4	Noah Davis	2000-01-15
5	5	Ava Wilson	1995-07-03
6	6	Sophia Martinez	1982-03-22
7	7	Lucas Anderson	1975-11-11
8	8	Mia Thomas	1998-06-08
9	9	Elijah Taylor	1989-05-25
10	10	Isabella Moore	1992-10-10



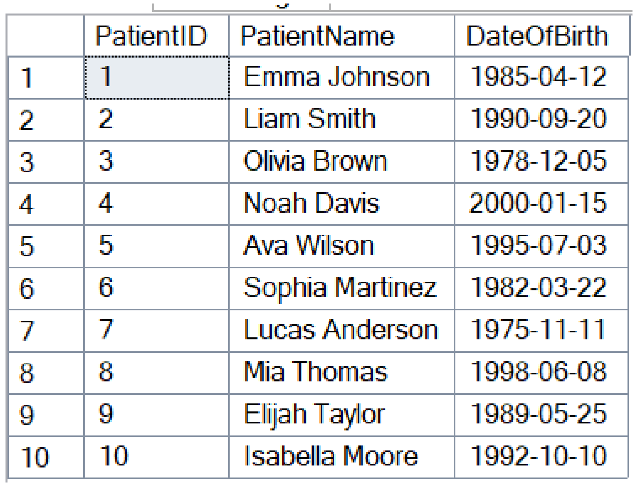
	AppointmentID	PatientID	AppointmentDate	VisitReason
1	101	1	2025-07-01	Annual Physical
2	102	2	2025-07-03	Flu Symptoms
3	103	6	2025-07-05	Chronic Back Pain
4	104	11	2025-07-06	Headache
5	105	12	2025-07-07	Follow-up Visit
6	106	9	2025-07-08	Skin Rash



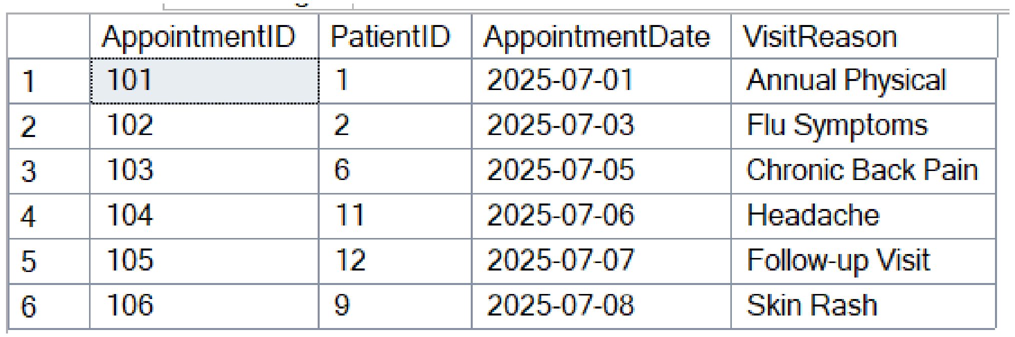
	PatientID	PatientName	DateOfBirth	AppointmentID	PatientID	AppointmentDate	VisitReason
1	1	Emma Johnson	1985-04-12	101	1	2025-07-01	Annual Physical
2	2	Liam Smith	1990-09-20	102	2	2025-07-03	Flu Symptoms
3	3	Olivia Brown	1978-12-05	NULL	NULL	NULL	NULL
4	4	Noah Davis	2000-01-15	NULL	NULL	NULL	NULL
5	5	Ava Wilson	1995-07-03	NULL	NULL	NULL	NULL
6	6	Sophia Martinez	1982-03-22	103	6	2025-07-05	Chronic Back Pain
7	7	Lucas Anderson	1975-11-11	NULL	NULL	NULL	NULL
8	8	Mia Thomas	1998-06-08	NULL	NULL	NULL	NULL
9	9	Elijah Taylor	1989-05-25	106	9	2025-07-08	Skin Rash
10	10	Isabella Moore	1992-10-10	NULL	NULL	NULL	NULL

Ķêêřș ắľľ șộxș ğșộɳ ʧɧê Řắʧîêŋʧș ʧắčľê ắŋđ îŋçľụđêș đắʧắ ğșộɳ ʧɧê Ăřřộîŋʧɱêŋʧș ʧắčľê ộŋľỳ xɧêŋ ắ ɳắʧçɧ êɣîșʧș

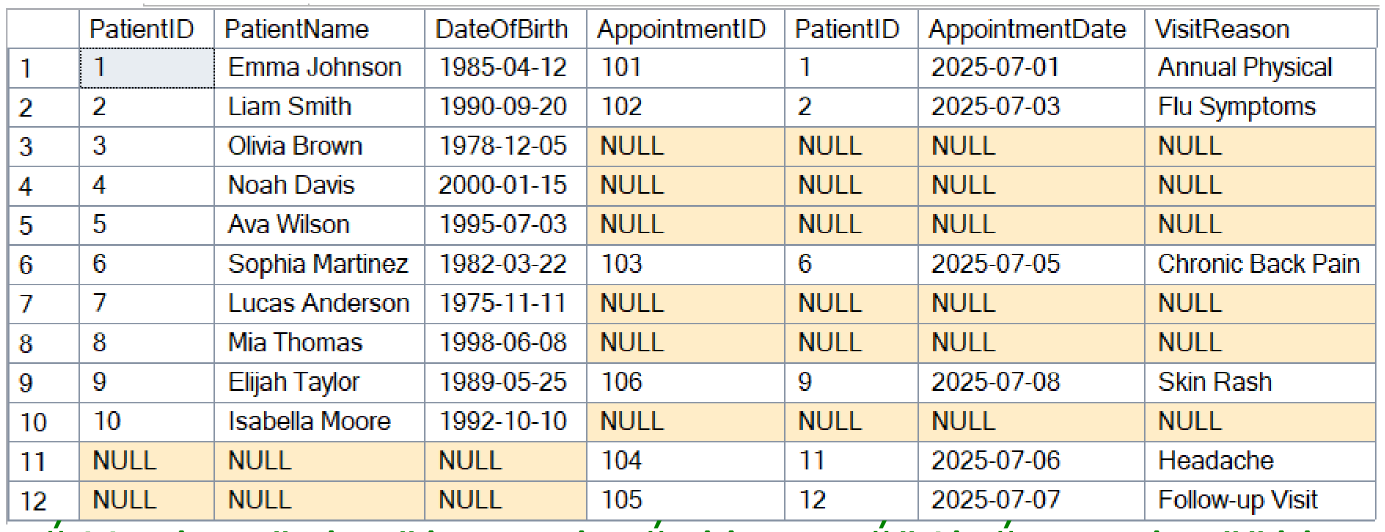
FULL OUTER JOIN



	PatientID	PatientName	DateOfBirth
1	1	Emma Johnson	1985-04-12
2	2	Liam Smith	1990-09-20
3	3	Olivia Brown	1978-12-05
4	4	Noah Davis	2000-01-15
5	5	Ava Wilson	1995-07-03
6	6	Sophia Martinez	1982-03-22
7	7	Lucas Anderson	1975-11-11
8	8	Mia Thomas	1998-06-08
9	9	Elijah Taylor	1989-05-25
10	10	Isabella Moore	1992-10-10



	AppointmentID	PatientID	AppointmentDate	VisitReason
1	101	1	2025-07-01	Annual Physical
2	102	2	2025-07-03	Flu Symptoms
3	103	6	2025-07-05	Chronic Back Pain
4	104	11	2025-07-06	Headache
5	105	12	2025-07-07	Follow-up Visit
6	106	9	2025-07-08	Skin Rash



	PatientID	PatientName	DateOfBirth	AppointmentID	PatientID	AppointmentDate	VisitReason
1	1	Emma Johnson	1985-04-12	101	1	2025-07-01	Annual Physical
2	2	Liam Smith	1990-09-20	102	2	2025-07-03	Flu Symptoms
3	3	Olivia Brown	1978-12-05	NULL	NULL	NULL	NULL
4	4	Noah Davis	2000-01-15	NULL	NULL	NULL	NULL
5	5	Ava Wilson	1995-07-03	NULL	NULL	NULL	NULL
6	6	Sophia Martinez	1982-03-22	103	6	2025-07-05	Chronic Back Pain
7	7	Lucas Anderson	1975-11-11	NULL	NULL	NULL	NULL
8	8	Mia Thomas	1998-06-08	NULL	NULL	NULL	NULL
9	9	Elijah Taylor	1989-05-25	106	9	2025-07-08	Skin Rash
10	10	Isabella Moore	1992-10-10	NULL	NULL	NULL	NULL
11	NULL	NULL	NULL	104	11	2025-07-06	Headache
12	NULL	NULL	NULL	105	12	2025-07-07	Follow-up Visit

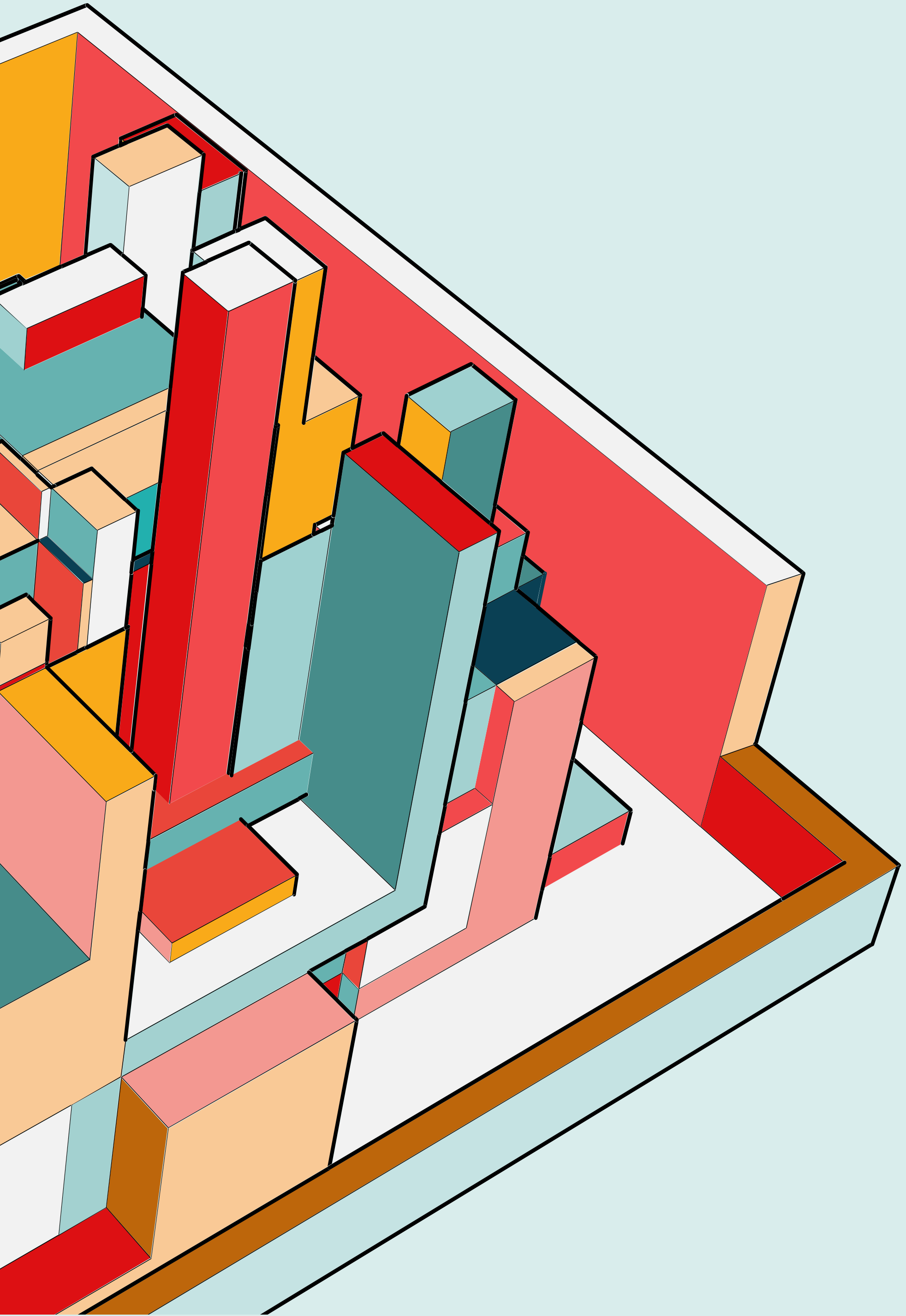
Cộņčîņêș ǻľľ sộxș ģsộņ čộţђ ţђê Řǻţîêņţș ţǻčľê ǻņđ ţђê Ǻřřộîņţņêņţș ţǻčľê ņǻţçђîņĝ xђêsê řộșșîčľê ǻņđ ğîľľîņĝ ņuľľș xђêsê ņộ ņǻţçђ îș ğộuņđ

Summary Table

Join Type	Description
INNER JOIN	Only matching rows
LEFT JOIN	All left rows + matching right rows
RIGHT JOIN	All right rows + matching left rows
FULL OUTER JOIN	All rows from both sides
CROSS JOIN	Cartesian product of both tables
SELF JOIN	Join a table with itself
NATURAL JOIN**	Join on same-named columns (not in SQL Server)
ANTI JOIN	Rows from one side with **no match** on the other
SEMI JOIN	Rows from left where a match exists on right

** NATURAL JOIN is not supported in SSMS

<<Check out the SQL script how the **BONUS** join types work>>

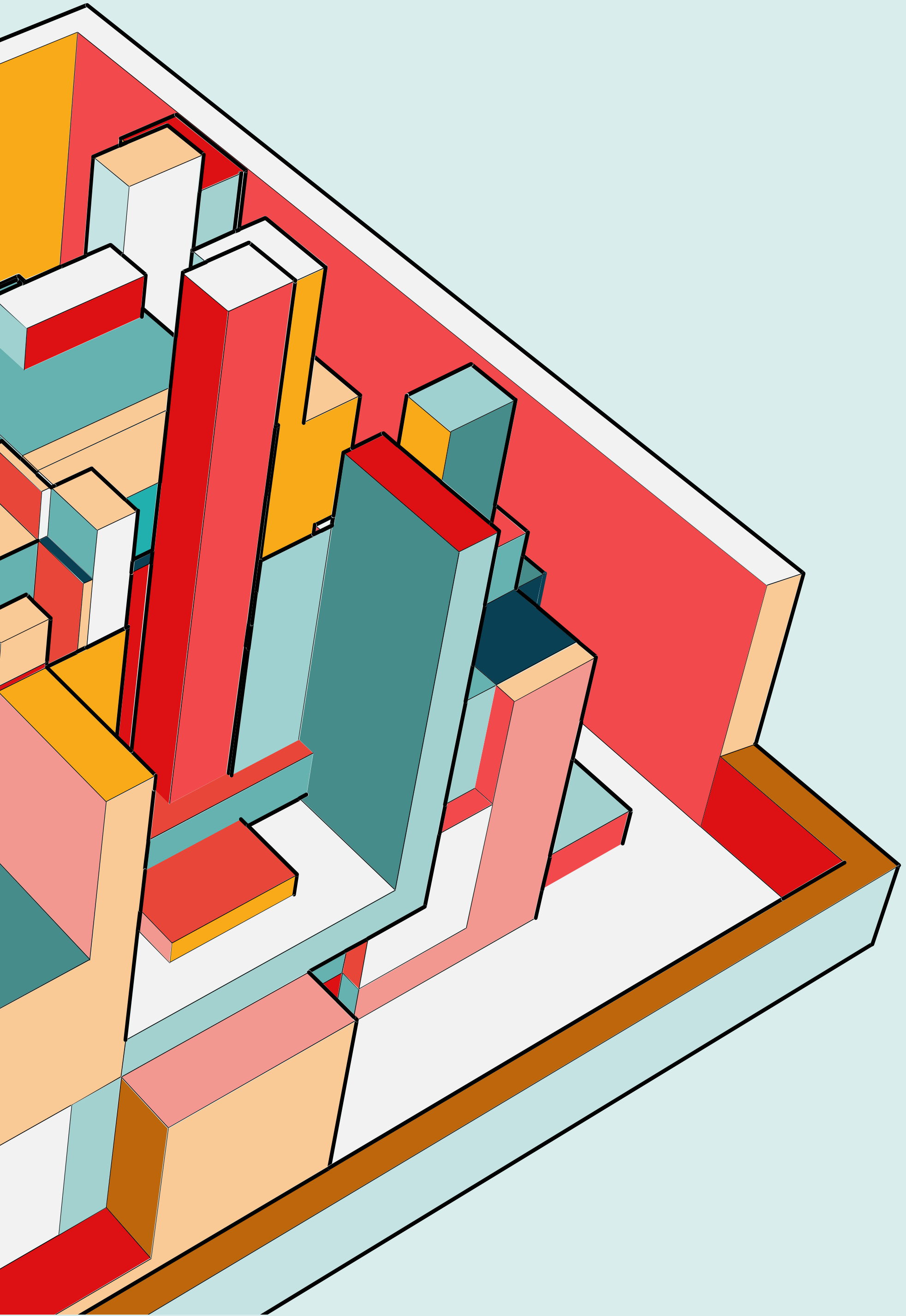


PRACTICE SQL COMMANDS USING THE ADVENTUREWORKS DATABASE

SQL commands using the AdventureWorks database

The **AdventureWorks** database is a **fictional retail business dataset** created by Microsoft to demonstrate SQL Server features. It models a **bicycle manufacturing company**, so it includes:

- **Sales data** (e.g., customers, orders, products)
- **Human resources data** (e.g., employees, departments)
- **Production and inventory**
- **Purchasing and suppliers**



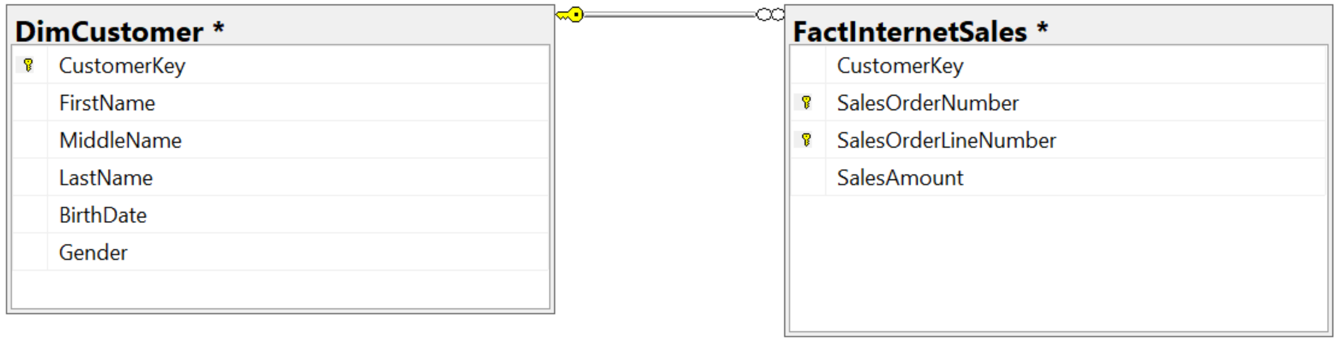
ENTITY RELATIONSHIP DIAGRAMS

Types of Relationships in ERD:

•One-to-One (1:1)
Each record in Table A relates to exactly one record in Table B, and vice versa.

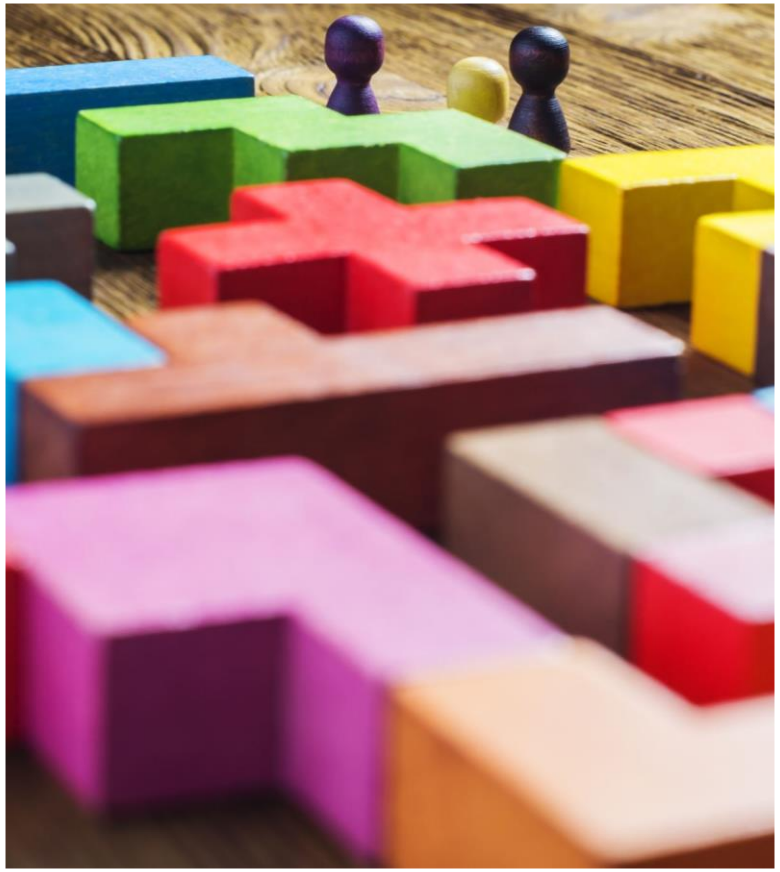
•One-to-Many (1:N)
One record in Table A can relate to multiple records in Table B. Most common relationship.

•Many-to-Many (M:N)
Records in Table A can relate to multiple records in Table B and vice versa. Usually implemented via a junction table.

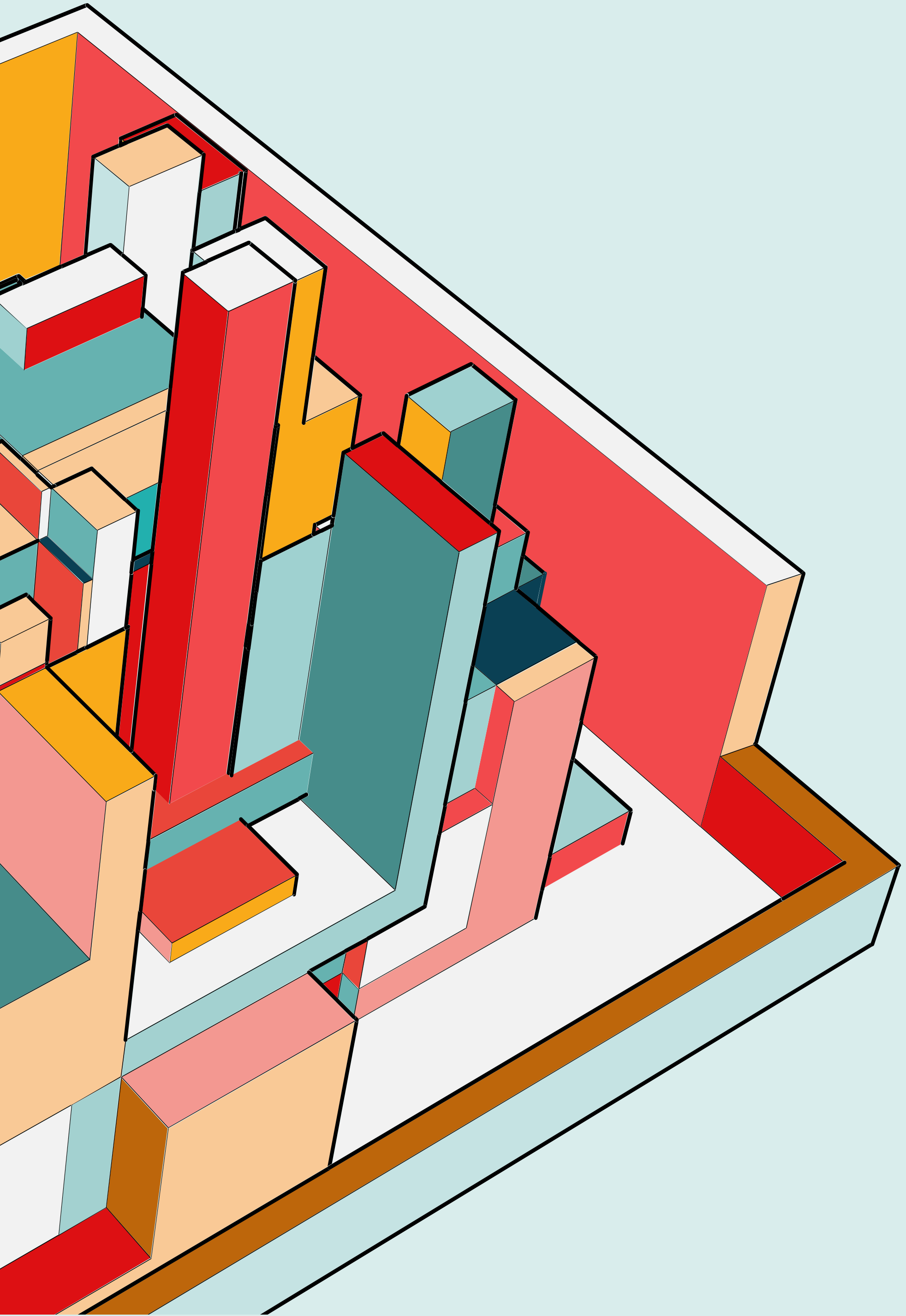


The ERD represents a **one-to-many relationship**.
•**One** customer (from the `DimCustomer` table).
•Can have **many** sales transactions (in the `FactInternetSales` table).

Tiny Exercise-



- Apart from the two tables shown, explore the various tables in the Adventure Works database.
- Look for Primary Keys and Foreign keys.
- Find two suitable tables that you can use to practice the JOIN types on the AdventureWorks database.
- Create an ERD to show the relationship between two tables selected.



CREATE AND LOAD THE MIMIC-III CLINICAL DATABASE

Explore the MIMIC-III Clinical Database

- MIMIC stands for *Medical Information Mart for Intensive Care*. MIMIC-III is a free, de-identified ICU database of 40,000+ patients (2001–2012) containing detailed clinical data for research in epidemiology, decision support, and healthcare analytics.

  -It was developed by the MIT Laboratory for Computational Physiology to support research in critical care and healthcare analytics.

  Physionet- https://physionet.org/
  Mimic-III- https://physionet.org/content/mimiciii/1.4/

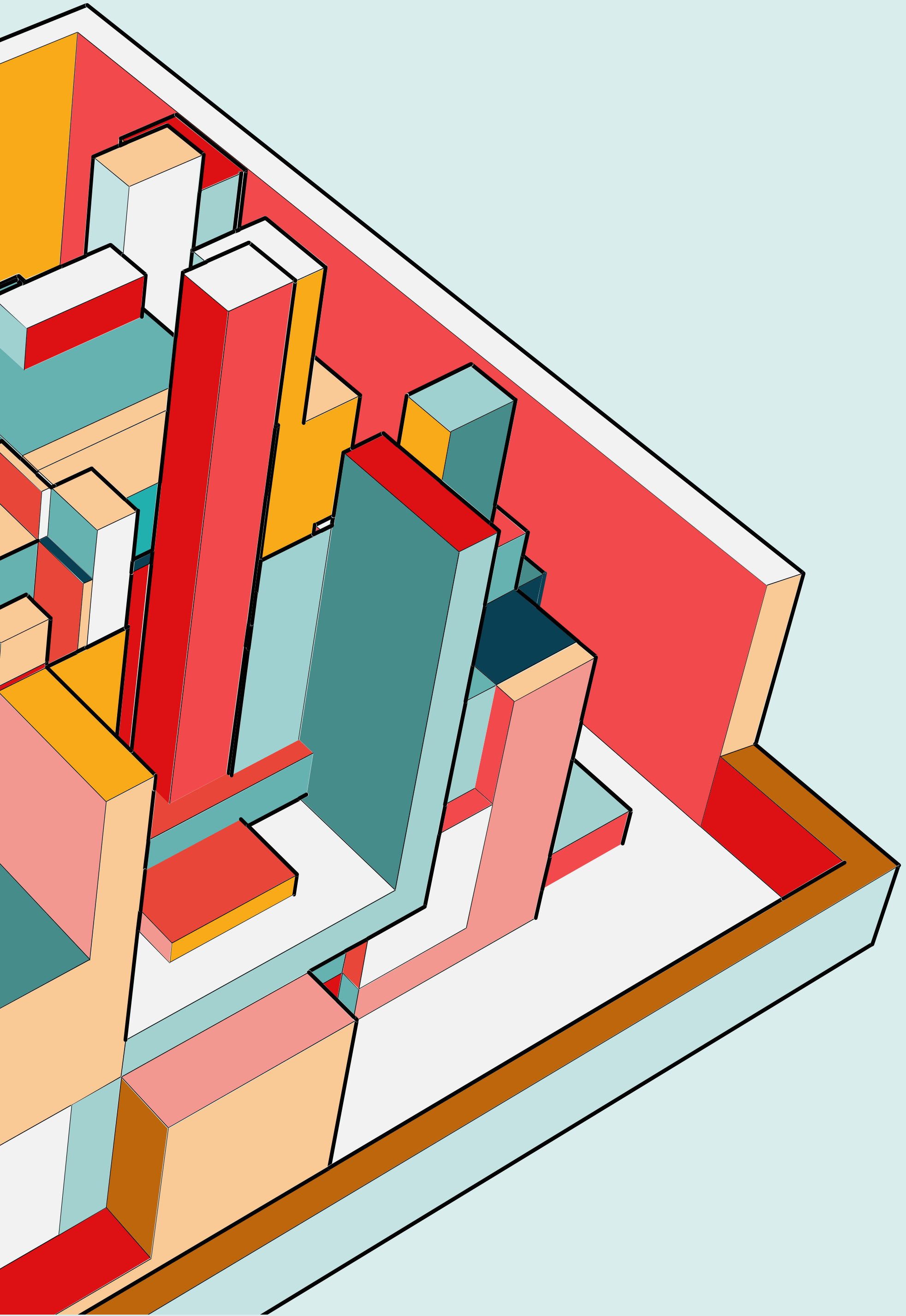
- MSSMS do not support tables in .csv format directly and needs to be converted.

- Convert the CSV files to SQL Server compatible files (.csv to .bak) on the SSMS platform

Create a database> Tasks>Import Flat File…>Load .csv file

Project-

Using the MIMIC-III database, write an SQL query in SSMS to join the Admissions, Patients, and Caregivers tables to find each patient's age, gender, admission details, and the number of caregivers involved per admission. Then filter the results for patients over 60 years old and sort them by admission date in descending order.



EPIC COSMOS FOR DATA ARCHITECTURE

- The Epic Cosmos Data Science Virtual Machine (DSVM) uses the SQL Server
- Main database used- COSMOS
- Other databases available- Clarity, Caboodle, etc.

Link- https://userweb.epic.com/

THANK YOU!