



DATABASE

Gehad ElSharkawy

Teaching assistance at Helwan
University, BIS Department



Agenda

Introduction

Data Types

Constraints

Case Study

Demo



Program

SQL SERVER 2014 EXPRESS

Link to download:

<https://www.microsoft.com/en-gb/download/details.aspx?id=42299>

How to setup:

<https://www.youtube.com/watch?v=EzFM7mzFUg&t=10s>

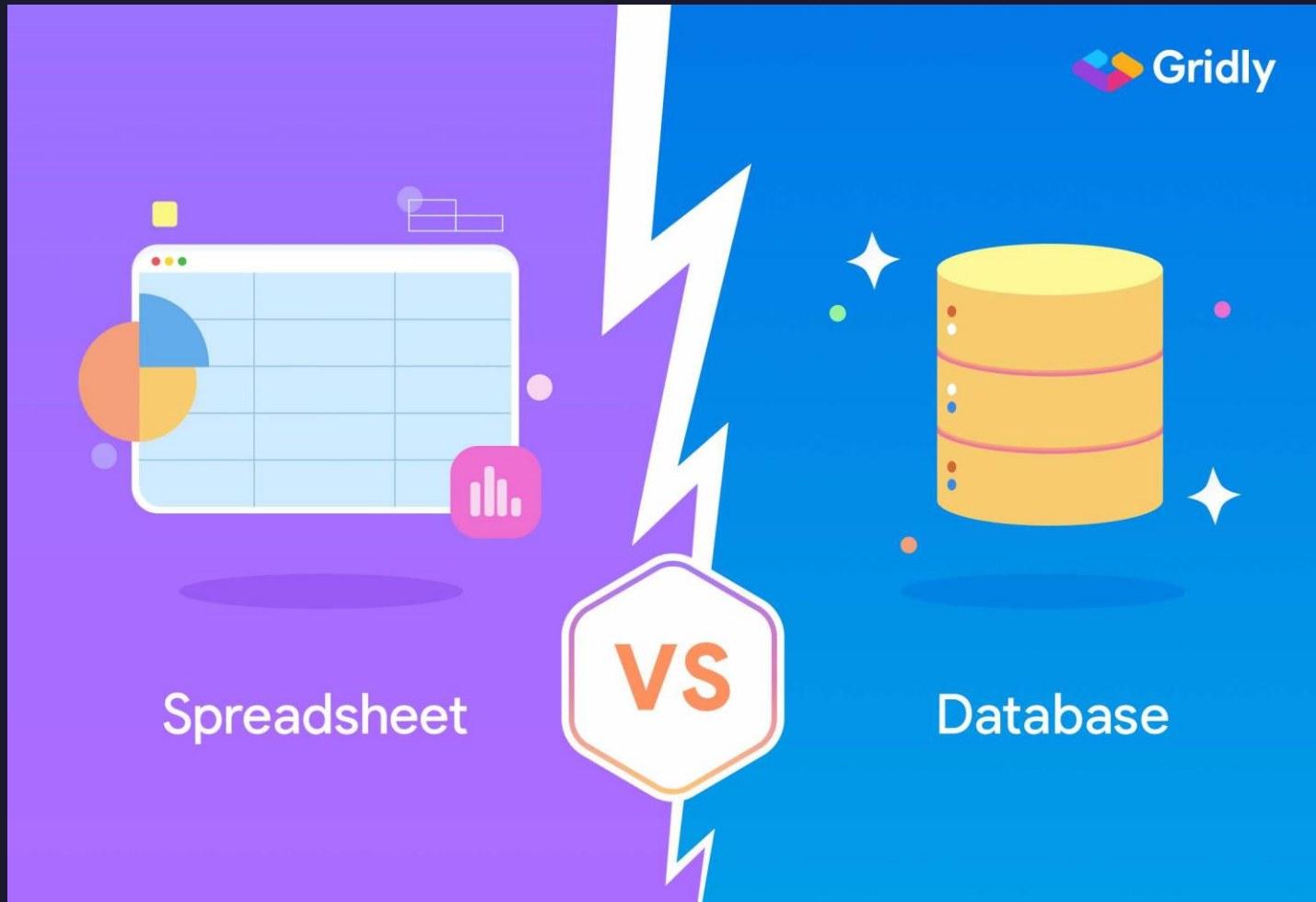




Introduction

- A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a [database management system \(DBMS\)](#). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.
- Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized.

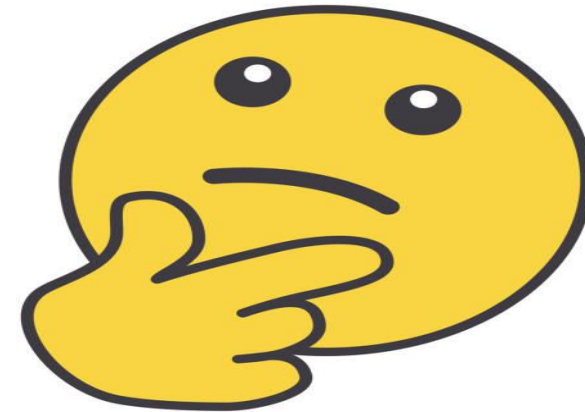
THINK



Tables are related
The amount of data
Who can access data
Data redundancy
Data consistency
Data integration
Data quality

THINK

- Database
- Data mining
- Data warehouse



What is SQL?

- SQL stands for Structured Query Language
- SQL is a programming language used by nearly all [relational databases](#) to query, manipulate, and define data, and to provide access control



What Can SQL do?

- 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

Table (Relation)

COLUMN

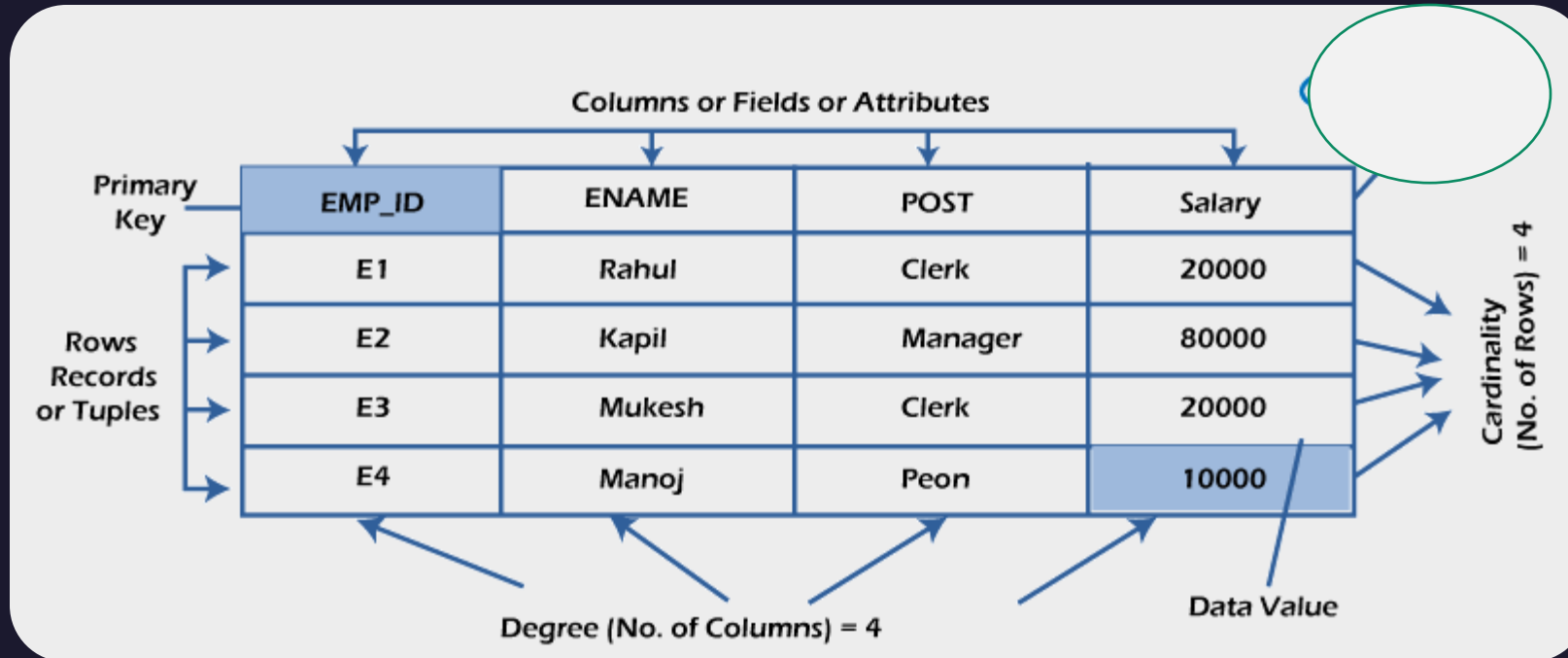
- A field is a column in a table that is designed to maintain specific information about every record in the table.
- A column is a vertical entity in a table that contains all information associated with a specific field in a table.

ROW

- A record, also called a row, is each individual entry that exists in a table.
- A record is a horizontal entity in a table.

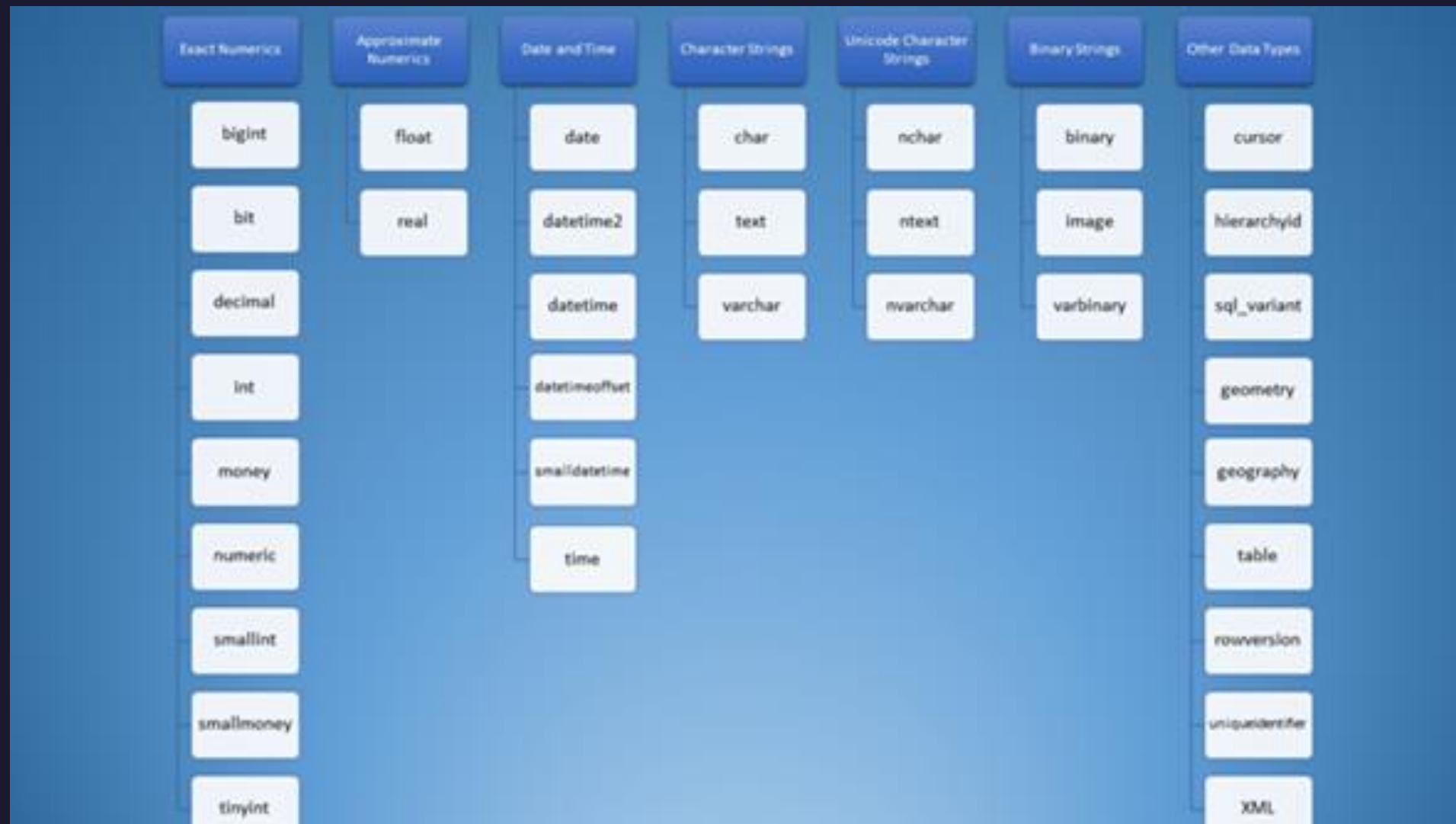


Table (Relation)



Data Types

The data type of a column defines what value the column can hold (integer, character, money, date and time, binary, and so on).



Data Types

- Each column in a database table is required to have a name and a data type.

Int	Varbinary(max)
Float 3.8887 >= 3.8887	Date dd/mm/yy
Money 3.8887 => 4	Time hh:mm:ss
Bit	Datetime dd/mm/yyyy hh:mm:ss
Image	



String Data Types

- Char (15)
- Nchar(250)
- Varchar(40)
- Nvarchar(250)



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.

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



DEMO

SQL SERVER

Create database and tables

- `create database BIS_db`
- `create table level3 (`
- `column1 datatype constraint1,`
- `column2 datatype constraint2,`
- `column3 datatype constraint3,`
- `column4 datatype);`



Case study

customer

CustomerID	CustomerN	gender	email	phone	CAddress
int	nvarchar(50)	bit	varchar(40)	char(11)	Varchar(35)
pk	not null	not null		Not null	

product

ProductID	Pname	Price	Pdescription	CustomerID
char(14)	nvarchar(30)	money	nvarchar(150)	int
pk	not null unique	not null check (price>0)	not null	fk

Case study

Category

cID	cName
int	nvarchar(40)
pk	not null

order

orderID	orderDate	CustomerID
int	datetime	int
pk	not null default (getdate())	fk

order_details

ProductID	orderID	Order_qty
char(14)	int	int
Fk pk	fk Pk	not null default(1)



Thank You

Gehad ElSharkawy

Gehad.magdi.mohammed@commerce.Hel
wan.edu.eg

