**How to create a Database?**

**A) Using SQL Server Management Studio:**

1. Connect to an SQL instance of the SQL Server Database Engine then expand that instance.
2. Right-click Databases, and then click New Database.
3. Enter a database name.
4. To create the database by with default values, click OK.

**B) Using a Query:**

**Open Query Window, and Type the below command and execute it.**

CREATE DATABASE SAMPLEDB1

**How to delete a Database?**

**A) Using SQL Server Management Studio**

1. In Object Explorer, connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand Databases, right-click the database to delete, and then click Delete.
3. Confirm the correct database is selected, and then click OK.

**B) Using a Query**

drop database sample

**DATA MODEL**

The graphical representation of business is known as a data model. Data model represents the way of implementing the data base for the business.

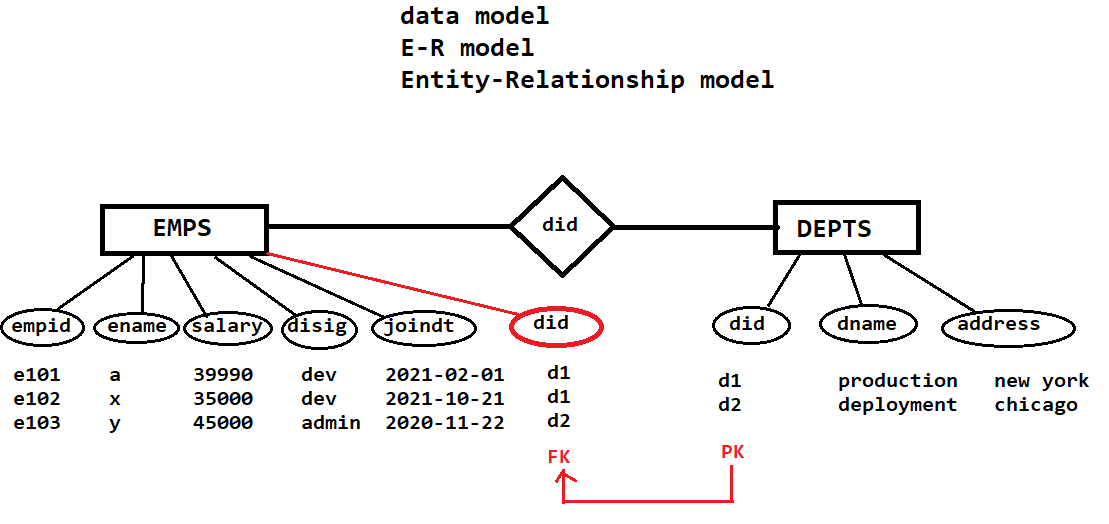
For any ***OLTP RDBMS*** the data model name is ***E-R (Entity-Relationship) Model*.**

In E-R model,

* Any object is represented with **RECTANGLE**.
* Object has Properties. Each property is represented with **Ellipse.**
* The relation between objects is represented with **Rhombus**.

Entity : Any real time object is known as entity

Ex : Emps, Depts, Products, Customers, . . .



From the above diagram,

🡪consider Entity names as Tables names

Ex: EMPS, DEPTS

🡪Consider property names as Column names.

Ex: In emps table , columns are empid, ename, salary, disig, joindt and did

🡪Relation is defined with Referential Integrity Constraint.

**DDL command:**

**CREATE**

Used to create any data base object like tables, views, indexes and so on ..

***HOW TO CREATE A TABLE?***

Syntax:

CREATE TABLE <table\_name>

(

<colname-1> DATATYPE (size),

<colname-2> DATATYPE (size),

: : : :,

: : : :

);

Ex:

write a query to create a table **emps** with *eno, ename, sal, desg, jdate, gender, mobile, mailid?*

create table emps

(

eno numeric(3),

ename varchar(20),

sal numeric(5),

desg varchar(20),

jdate date,

gender varchar(7),

mobile numeric(10),

mailid varchar(30),

);

**DATA TYPES**

SQL SERVER Bulit-In DATATYPES

Microsoft is providing, a set of predefined datatype names.

Each data type name represents one type of data, that is allowed into specific column.

Based on data type physical memory is allocated for each value in the table.

**String Data Types**

|  |  |  |  |
| --- | --- | --- | --- |
| char(n) | Fixed width character string | 8,000 characters |  |

|  |  |  |
| --- | --- | --- |
| varchar(n) | Variable width character string | 8,000 characters |

|  |  |  |
| --- | --- | --- |
| varchar(max) | Variable width character string | 1,073,741,824 characters |

|  |  |  |
| --- | --- | --- |
| Text | Variable width character string | 2GB of text data |

Unicode string data types:

Nchar(), nvarchar(), nvarchar(max), ntext

|  |  |  |  |
| --- | --- | --- | --- |
| binary(n) | Fixed width binary string | 8,000 bytes |  |
| Varbinary | Variable width binary string | 8,000 bytes |  |
| varbinary(max) | Variable width binary string | 2GB |  |
| Image | Variable width binary string | 2GB |  |

**Numeric Data Types**

|  |  |  |
| --- | --- | --- |
| Bit | Integer that can be 0, 1, or NULL |  |
| Tinyint | Allows whole numbers from 0 to 255 | 1 byte |
| Smallint | Allows whole numbers between -32,768 and 32,767 | 2 bytes |
| Int | Allows whole numbers between -2,147,483,648 and 2,147,483,647 | 4 bytes |
| Bigint | Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807 | 8 bytes |
| decimal(p,s) | Fixed precision and scale numbers.  Allows numbers from -10^38 +1 to 10^38 –1.  The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.  The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0 | 5-17 bytes |
| numeric(p,s) | Fixed precision and scale numbers.  Allows numbers from -10^38 +1 to 10^38 –1.  The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.  The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0 | 5-17 bytes |
| Smallmoney | Monetary data from -214,748.3648 to 214,748.3647 | 4 bytes |
| Money | Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807 | 8 bytes |
| float(n) | Floating precision number data from -1.79E + 308 to 1.79E + 308.  The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53. | 4 or 8 bytes |
| Real | Floating precision number data from -3.40E + 38 to 3.40E + 38 | 4 bytes |

**Date and Time Data types**

|  |  |  |
| --- | --- | --- |
| Datetime | From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds | 8 bytes |
| datetime2 | From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds | 6-8 bytes |
| Smalldatetime | From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute | 4 bytes |
| Date | Store a date only. From January 1, 0001 to December 31, 9999 | 3 bytes |
| Time | Store a time only to an accuracy of 100 nanoseconds | 3-5 bytes |
| Datetimeoffset | The same as datetime2 with the addition of a time zone offset | 8-10 bytes |
| Timestamp | Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable |  |