

Designing and Implementing Tables



Gerald Britton

IT SOLUTIONS DESIGNER, SQL SERVER SPECIALIST

@GeraldBritton www.linkedin.com/in/geraldbritton



Creating Tables

Database name

Schema name

- Default is dbo

Table name

Column names

Data types

Constraints

- Primary key
- Nullability



Regular Identifiers

Letter
Underscore (_)
At sign (@)
Number sign (#)

First character

Letters
Decimal numbers
@, \$, # or _

**Subsequent
characters**

Database
Table
Schema
Column

Not a reserved word

No embedded
spaces or special
characters

Not these!

Rule breakers
enclosed in brackets
[My Table]

Delimited identifiers

128 characters

Maximum length



Naming Conventions For Database Objects

Do not use @ as the first character

- Used for variable names in T-SQL

Do not use # as the first character

- Used for objects in tempdb

Use a consistent style for naming objects

- CamelCase: MyTable
- Underscore separated: my_table
- Hybrid: UN_Agency

Avoid delimited identifiers: [My Table]



```
USE Rodents;
```

```
GO
```

```
CREATE TABLE Mice.TheQuestion
```

```
(TheAnswer int NOT NULL);
```

```
INSERT INTO Mice.TheQuestion
```

```
(TheAnswer) VALUES (42);
```

◀ Enter the database context

◀ Create a new table in the Mice schema

◀ The table has one column, TheAnswer

◀ Populate the table

Adams, Douglas (1979) *The Hitchhiker's Guide to the Galaxy* London, UK: Pan Macmillan



Bob's Order Data

Text	Customer name, Address, Shoe Style, SKU, Salesman name
Integer	Quantity, Order ID
Decimal	Size, Unit price, Total price, Discount percentage
Date	Order date, Requested delivery date, Actual delivery date



Data Types for Textual Data

`char(n)`

`n = 1...8000`

**Fixed length
Non-Unicode**

`varchar(n)`

`n = 1...8000`

`varchar(max)`

**Variable length
Non-Unicode**

`nchar(n)`

`n = 1...4000`

**Fixed length
Unicode**

`nvarchar(n)`

`n = 1...4000`

`nvarchar(max)`

**Variable length
Unicode**



Data Types for Integer Data

tinyint

0 to 255

1 byte

smallint

-2^{15} to $2^{15}-1$

2 bytes

int

-2^{31} to $2^{31}-1$

4 bytes

bigint

-2^{63} to $2^{63}-1$

8 bytes



Data Types for Decimal Data

**decimal [(p, s)]
and
numeric [(p, s)]**

$-10^{38} + 1$ to $10^{38} - 1$

5 to 17 bytes

money

4 decimal places

-

922,337,203,685,477.5

808 to

922,337,203,685,477.5

807

8 bytes

smallmoney

4 decimal places

-214,748.3648

to

214,748.3647

4 bytes



Data Types for Dates and Times

date

0001-01-01 to 9999-12-31

3 bytes

time[(n)]

n = 0 to 7

5 bytes

datetime

Jan 1, 1753 to Dec 31, 9999

8 bytes

smalldatetime

Jan 1, 1900 to Jun 6, 2079

4 bytes

datetime2(n)

0001-01-01 to 9999-12-31

6 to 8 bytes

datetimeoffset(n)

0001-01-01 to 9999-12-31

10 bytes

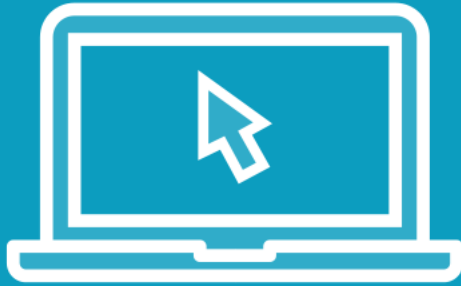


Hold on! What about
expedited orders?

bit type!



Demo



Create database

Create order tracking table

Key constraint

- Matching (backing) index

Index \neq Constraint



“A collation specifies the bit patterns that represent each character in a data set. Collations also determine the rules that sort and compare data.”

docs.microsoft.com/en-us/sql/relational-databases/collations/collation-and-unicode-support



Collations

Unicode vs. non-Unicode

nchar/nvarchar vs. char/nvarchar

All character data has some collation

Column level, Database level, Instance level

Also on expressions

Used for sorting rules, case, and accent sensitivity

Code page for non-Unicode data



Demo



Collations



Summary



Creating tables

Naming rules and conventions

Data types

CREATE TABLE

PRIMARY KEY constraint

Collations

Other data types: <https://bit.ly/2PXeEGv>

