Data Types in SQL

October 31, 2012

SQL Data Types

SQL data can be put into fields that are defined as strings, numeric and date and time. When we create a table, we name the column and define the data type for each. The following are SQL data types.

Strings	Numeric	Date and Time
Fixed Length String	Number(p,s)	Date
Variable Length String	Number(p)	
Large Object Types	Integer	
	Small Integer	

Fixed Length String

Fixed Length String is represented by character(n). The number of characters will always be the same length such as abbreviated names of US states. These could be OH, NY and KY. We can type numbers, letters and special characters as long as it is fixed length.

Example:

```
Create table order_guide
(
State char(2)
)
;
```

Will create a table called order_guide and it has a column called State that allows for a two character fixed length string.

N Fixed Length Character

N Character Fixed Length String is represented by nchar(n). The number of characters will always be the same length such as A is 65 and B is 66 when handling ASCII numbers for keyboard characters. These could be OH, NY and KY. We can type numbers, letters and special characters as long as it is fixed length.

Strings
Fixed Length String
Variable Length String
Large Object Types

Example:

Alter table order_guide

Add CharacterNumber nchar(3);

Will add a column called CharacterNumber that allows for a 3 place number.

Variable Length String

A Variable Length String is represented by varchar2(n). We can type numbers, letters and special characters as long as it fits into the maximum number of characters defined by the number in (n).

Strings
Fixed Length String
Variable Length String
Large Object Types

Example:

Alter table order_guide

Add city varchar2(25);

Will add a column called city that allows for to a 25 characters string to a table called order_guide.

Large Object Types

Large Date Types are entered using clob. Clob is used for large data strings that can go up into terabytes of characters.

Strings
Fixed Length String
Variable Length String
Large Object Types

Example:

Alter table order_guide

Add comments clob;

Will add a column called comments that allows Clob is used for large data strings that can go up into terabytes of characters.

Number(p,s)

Number data types contain number that have a maximum of p places and a maximum of s decimal places.

Numeric
Number(p,s)
Number(p)
Integer
Small Integer

Example:

Alter table order_guide

Add weight number(4,2);

Will add a column called weight that allows for a 4 place number that has two decimal places. Customers enter 12.55, 11.10 and 99.51.

Number(p)

Number data types contain number that have a maximum of p places and no decimal places.

Numeric
Number (p,s)
Number (p)
Integer
Small Integer

Example:

Alter table order_guide

Add height_cm number(4);

Will add a column called height_cm that allows for a 4 place number that has no decimal places. Customers enter 1255, 1110 and 9951.

Integer

Integer data types contain whole numbers that can be negative, zero or positive. Their numeric values can range from -2,147,483,648 to 2,147,483,647.

Numeric
Bit(n)
Decimal (p,s)
Integer
Smallint
Bigint
Float (p,s)
Double Precision(p,s)
Real(s)

Example:

Alter table order_guide

Add quantity integer;

Will add a column called quantity that allows for a whole number. Customers enter 1, 2, or any other whole number. We can deny negative integers in our code.

Small Integer

Small Integer data types contain whole numbers that can be negative, zero or positive. Their numeric values can range from -32,768 to 32,767.

Numeric
Number(p,s)
Number(p)
Integer
Small Integer

Example:

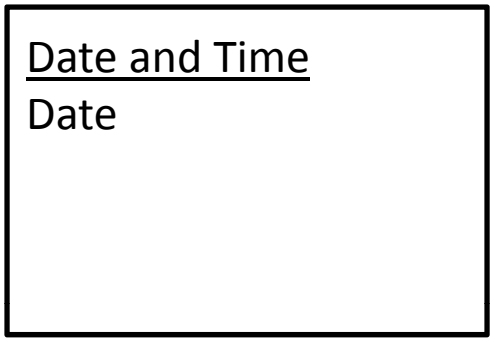
Alter table order_guide

Add days smallint;

Will add a column called days that allows for a whole number. Customers enter 1, 2, or any other whole number. We can deny negative integers in our code.

Date

To input the date into a field, we will use the date function.



Example:

Alter table order_guide

Add date_ordered date;

Will add a column called date_ordered that allows for the date from the user.

Experiment with this Table

```
SQL>
SQL> -- create demo table
SQL> create table Employee(
2 ID VARCHAR2(4 BYTE) NOT NULL,
3 First_Name VARCHAR2(20 BYTE),
4 Last_Name VARCHAR2(20 BYTE),
5 Start_Date DATE,
6 End_Date DATE,
7 Salary Number(8,2),
8 City VARCHAR2(20 BYTE),
9 Description VARCHAR2(80 BYTE)
10 )
11 /

Table created.

SQL>
```

Insert this Data

Insert this data:

01 Jason Martin 25-JUL-96 25-JUL-06 1234.56 Toronto Programmer
02 Alison Mathews 21-MAR-76 21-FEB-86 2234.78 Vancouver Tester
03 James Smith 12-DEC-78 15-MAR-90 2324.78 Vancouver Tester
04 Celia Rice 24-OCT-82 21-APR-99 3334.78 Vancouver Manager
05 Robert Black 15-JAN-84 08-AUG-98 4334.78 Vancouver Tester
06 Linda Green 30-JUL-87 04-JAN-96 5334.78 New York Tester
07 David Larry 31-DEC-90 12-FEB-98 6334.78 New York Manager
08 James Cat 17-SEP-96 15-APR-02 7334.78 Vancouver Tester

Query the Date

Experiment with this SQL function:

```
SQL>
SQL>
SQL> SELECT id, first_name, salary
2 FROM employee
3 WHERE salary > 3000
4 ORDER BY salary;

ID FIRST_NAME SALARY
---- -------------------- ----------
04 Celia 3334.78
05 Robert 4334.78
06 Linda 5334.78
07 David 6334.78
08 James 7334.78
```