University of Bisha
Faculty Of Computing and Information Technology

Part (8)

Structure Query Language (SQL)

Outlines of Part 8 → use Chapter 6

- Introduction and SQL Data type and Literals
- Type of SQL Commands
- Operators
- Creating Queries
- Aggregate Functions

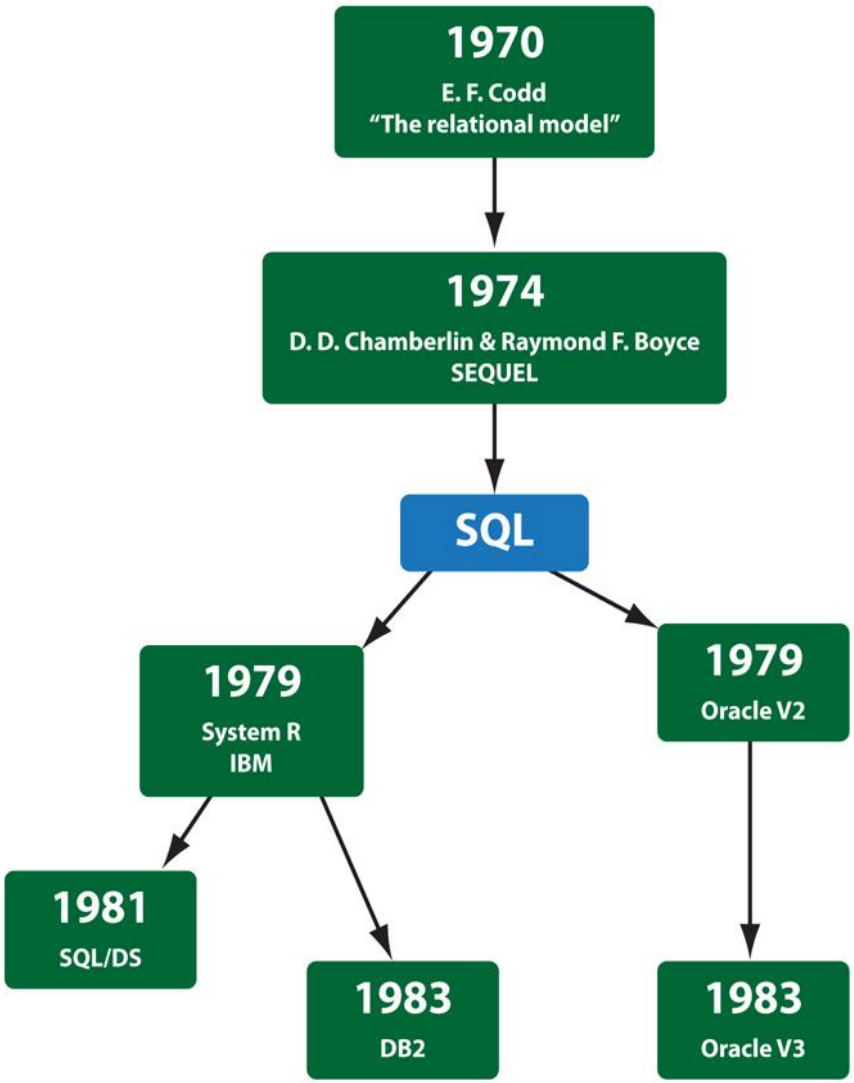
Introduction and SQL Data type and Literals

- **SQL** (***Structured Query Language***)

is a standardized programming language used for managing relational databases and performing various operations on the data in them.

- The standard for relational database management systems (RDBMS)

History of SQL



Introduction and SQL Data type and Literals

- SQL ***Data Types*** define the type of value that can be stored in a table column.

- SQL Server offers six categories of data types for your use which are listed below –

  1. **Numeric** data types such as int, tinyint, bigint, float, real etc.
  2. **Date and Time** data types such as Date, Time, Datetime etc.
  3. **Character and String** data types such as char, varchar, text etc.
  4. **Unicode character string** data types, for example nchar, nvarchar, ntext etc. (Is the universal character-encoding standard **used** for representation of text).
  5. **Binary** data types such as binary, varbinary etc.
  6. **Miscellaneous** data types – clob, blob, xml, cursor, table etc.

SQL Numeric Data Types

Datatype	From	To
bit	0	1
tinyint	0	255
smallint	-32,768	32,767
int	-2,147,483,648	2,147,483,647
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
decimal	-10^38 +1	10^38 -1
numeric	-10^38 +1	10^38 -1
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

SQL Date and Time Data Types

Datatype	Description
DATE	Stores date in the format YYYY-MM-DD
TIME	Stores time in the format HH:MI:SS
DATETIME	Stores date and time information in the format YYYY-MM-DD HH:MI:SS
TIMESTAMP	Stores number of seconds passed since the Unix epoch ('1970-01-01 00:00:00' UTC)
YEAR	Stores year in 2 digit or 4 digit format. Range 1901 to 2155 in 4-digit format. Range 70 to 69, representing 1970 to 2069.

SQL Character and String Data Types

Datatype	Description
CHAR	Fixed length with maximum length of 8,000 characters
VARCHAR	Variable length storage with maximum length of 8,000 characters
VARCHAR(max)	Variable length storage with provided max characters, not supported in MySQL
TEXT	Variable length storage with maximum size of 2GB data

SQL Unicode Character and String Data Types

Datatype	Description
NCHAR	Fixed length with maximum length of 4,000 characters
NVARCHAR	Variable length storage with maximum length of 4,000 characters
NVARCHAR(max)	Variable length storage with provided max characters
NTEXT	Variable length storage with maximum size of 1GB data

Note that above data types are not supported in MySQL database.

Introduction and SQL Data type and Literals

SQL Binary Data Types

Datatype	Description
BINARY	Fixed length with maximum length of 8,000 bytes
VARBINARY	Variable length storage with maximum length of 8,000 bytes
VARBINARY(max)	Variable length storage with provided max bytes
IMAGE	Variable length storage with maximum size of 2GB binary data

SQL Miscellaneous Data Types

Datatype	Description
CLOB	Character large objets that can hold up to 2GB
BLOB	For binary large objects
XML	for storing xml data
JSON	for storing JSON data

Introduction and SQL Data type and Literals

In SQL, a ***literal*** is the same as a constant. We'll cover several types of literals - string, integer, decimal, and datetime literals.

❖String Literals

String literals are always surrounded by single quotes (').

Examples: :

- 'TechOnTheNet.com' 'This is a literal
- ' 'XYZ'
- '123'

These string literal examples contain of strings enclosed in single quotes.

❖Integer Literals

Integer literals can be either positive numbers or negative numbers, but do not contain decimals. If you do not specify a sign, then a positive number is assumed. ***Examples:***

536 +536 -536

Introduction and SQL Data type and Literals

❖Decimal Literals

Decimal literals can be either positive numbers or negative numbers and contain decimals. If you do not specify a sign, then a positive number is assumed.

Examples:

- 24.7
- +24.7
- -24.7

❖Datetime Literals

Datetime literals are character representations of datetime values that are enclosed in single quotes.

Examples:

- 'April 30, 2015'
- '2015/04/30'
- '2015/04/30 08:34:25‘

Type of SQL Commands

The four main ***categories*** of SQL statements are as follows:

1. DML (Data Manipulation Language)
2. DDL (Data Definition Language)
3. DCL (Data Control Language)
4. TCL(TransactionControl Language)

Type of SQL Commands…..Cont.
Data Definition Language (DDL)

Data Definition Language or DDL commands are used for changing the structure of a table. In other words, DDL commands are capable of creating, deleting, and modifying data.

Main SQL DDL statements are:

- **CREATE:** create database and its objects like (table, index, views, store procedure, function and triggers).

- **ALTER:** alters the structure of the existing database.Typically, the ALTER command is used either to add a new attribute or modify the characteristics of some existing attribute.

- **DROP:** delete objects from the database

Type of SQL Commands…..Cont.
Data Manipulation Language (DML)

Data Manipulation Language (DM) commands help in modifying a relational database.

Main SQL DML statements are:

- **DELETE:** Used for removing one or more rows from a table.
- **INSERT:**Used for inserting data into the row of a table.
- **UPDATE:**Used to modify or update the value of a column in a table. It can update all rows or some selective rows in the table.

Type of SQL Commands…..Cont.
Data Control Language (DCL)

Data Control Language (DCL).In order to protect the information in a table from unauthorized access, DCL commands are used. A DCL command can either enable or disable a user from accessing information from a database. List of user access privileges:

- ALTER
- DELETE
- INDEX
- INSERT
- SELECT
- UPDATE

Main SQL DDL statements are:

- **GRANT :**Used for granting user access privileges to a database.
- **REVOKE:** Used for taking back permission given to a user.

Operators

SQL operators are reserved keywords used in the WHERE clause of a SQL statement to perform arithmetic, logical and comparison operations.

Types of SQL Operators

- **Arithmetic Operators**
- **Comparison Operators**
- **Logical Operators**

Operators…..Cont.

Arithmetic Operators

These operators are used to perform operations such as addition, multiplication, subtraction etc.

Operator	Operation	Description
+	Addition	Add values on either side of the operator
–	Subtraction	Used to subtract the right hand side value from the left hand side value
*	Multiplication	Multiples the values present on each side of the operator
/	Division	Divides the left hand side value by the right hand side value
%	Modulus	Divides the left hand side value by the right hand side value; and returns the remainder

Example:

SELECT ename, sal, sal+500 FROM ***emp***;

Operators…..Cont.

Comparison Operators

These operators are used to perform operations such as equal to, greater than, less than etc.

Operator	Operation	Description
=	Equal to	Used to check if the values of both operands are equal or not. If they are equal, then it returns TRUE.
>	Greater than	Returns TRUE if the value of left operand is greater than the right operand.
<	Less than	Checks whether the value of left operand is less than the right operand, if yes returns TRUE.
>=	Greater than or equal to	Used to check if the left operand is greater than or equal to the right operand, and returns TRUE, if the condition is true.
<=	Less than or equal to	Returns TRUE if the left operand is less than or equal to the right operand.
<> or !=	Not equal to	Used to check if values of operands are equal or not. If they are not equal then, it returns TRUE.
!>	Not greater than	Checks whether the left operand is not greater than the right operand, if yes then returns TRUE.
!<	Not less than	Returns TRUE, if the left operand is not less than the right operand.

Example:

SELECT * FROM ***students*** WHERE Age > 25;

Operators…..Cont.

Logical Operators

The logical operators are used to perform operations such as ALL, ANY, NOT, BETWEEN etc.

Operator	Description
ALL	Used to compare a specific value to all other values in a set
ANY	Compares a specific value to any of the values present in a set.
IN	Used to compare a specific value to the literal values mentioned.
BETWEEN	Searches for values within the range mentioned.
AND	Allows the user to mention multiple conditions in a WHERE clause.
OR	Combines multiple conditions in a WHERE clause.
NOT	A negate operators, used to reverse the output of the logical operator.
EXISTS	Used to search for the row's presence in the table.
LIKE	Compares a pattern using wildcard operators.
SOME	Similar to the ANY operator, and is used compares a specific value to some of the values present in a set.

Example

```
SELECT * FROM Students
WHERE Age > ANY (SELECT Age FROM Students WHERE Age > 21);
```

Creating Queries

CREATE Statement

The **CREATE DATABASE** statement is used to create a new database in SQL.

CREATE	Use	Syntax	Example
CREATE Database	Is used to create a new SQL database.	CREATE DATABASE DatabaseName;	SQL> CREATE DATABASE testDB;
CREATE Table	Is used to create a new table.	CREATE TABLE table_name(column1 datatype, column2 datatype, column3 datatype, ... columnN datatype, PRIMARY KEY(one or more columns));	SQL> CREATE TABLE CUSTOMERS(ID INT NOT NULL, NAME VARCHAR (20) NOT NULL, AGE INT NOT NULL, ADDRESS CHAR (25) , SALARY DECIMAL (18, 2), PRIMARY KEY (ID));

CREATE Statement Example:

- Create the table "student" with the following properties

Field name	Data type	length	Constrains
st_no	number	8	Primary key
name	varchar2	30	Not null
address	varchar2	40	
age	number	2	

CREATE TABLE student(st_no number(8), name varchar2(30) Not null, address varchar2(40), age number(2), constraints student_pk primary key(st_no));

ALTER TABLE Statement:

- Add a new column to a table.
- Drop a column from a table.
- Add a new table constraint.
- Drop a table constraint.
- Set a default for a column.
- Drop a default for a column.

ALTER TABLE Examples:

- Change "**student**" table by adding new field named "**mobile**".

```
ALTER TABLE student
        ADD mobile number;
```

- Change "**student**" table by adding primary key to the field named "**StID**".

```
ALTER TABLE student ADD CONSTRAINT st_pk
PRIMARY KEY(StID);
```

Creating Queries……Cont.

SELECT Statement:

- The most commonly used SQL command is SELECT statement.
- SQL SELECT statement is used to query or retrieve data from a table in the database.
- A query may retrieve information from specified columns or from all of the columns in the table.
- To create a simple SQL SELECT Statement, you must specify the column(s) name and the table name.
- The whole query is called SQL SELECT Statement.

Basic form:

```
SELECT  <attributes>
FROM    <one or more relations>
WHERE   <conditions>
```

Creating Queries…..Cont.

SELECT Statement Example:

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

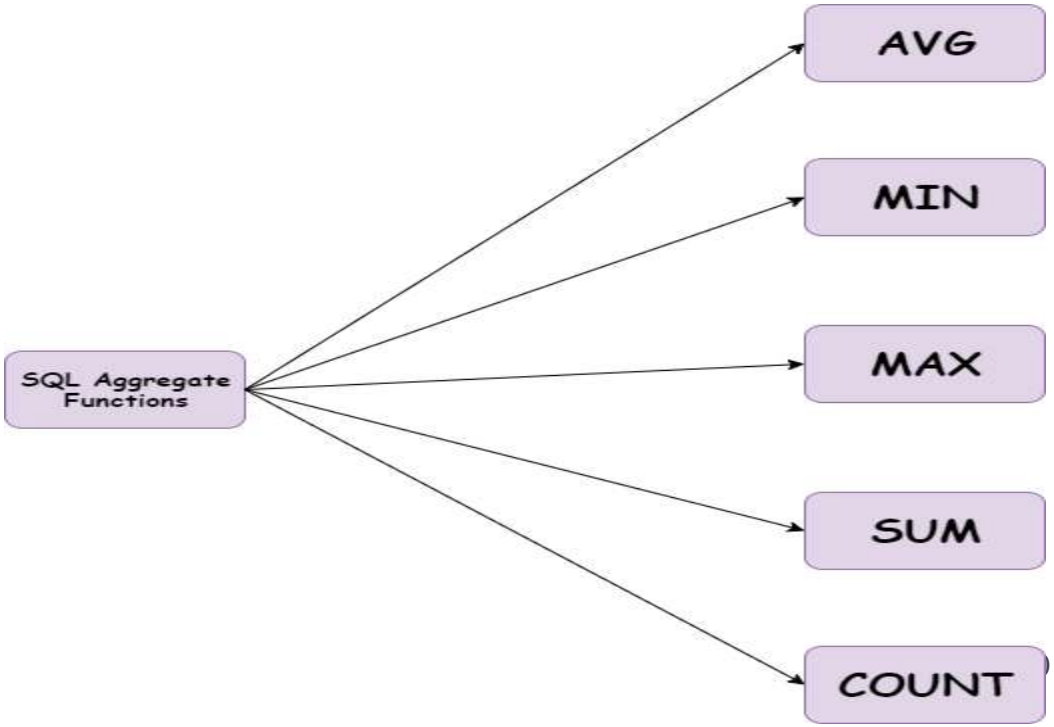
```
SELECT  *
FROM    Product
WHERE   category='Gadgets'
```

"selection"

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks

Aggregate Functions

- In database management an aggregate function is a function where the values of multiple rows are grouped together as input on certain criteria to form a single value of more significant meaning.
- SQL supports several aggregation operations:



Aggregate Functions….Cont.

List of SQL Aggregate functions are :

Functions	Description
SQL Count function	The SQL COUNT function returns the number of rows in a table satisfying the criteria specified in the WHERE clause. It sets the number of rows or non NULL column values.
SQL Sum function	The SQL AGGREGATE SUM() function returns the sum of all selected column.
SQL Avg function	The SQL AVG function calculates the average value of a column of numeric type. It returns the average of all non NULL values
SQL Max function	The aggregate function SQL MAX() is used to find the maximum value or highest value of a certain column or expression. This function is useful to determine the largest of all selected values of a column.
SQL Min function	The aggregate function SQL MIN() is used to find the minimum value or lowest value of a column or expression. This function is useful to determine the smallest of all selected values of a column.

Aggregate Functions….Cont.

- *Examples*

```
SELECT  avg(price)
FROM    Product
WHERE   maker="Toyota"
```

```
SELECT  count(*)
FROM    Product
WHERE   year > 1995
```


Thank you