

Introduction

NGUYEN Hong Phuong

Email: phuongnh@soict.hust.edu.vn

Site: <http://users.soict.hust.edu.vn/phuongnh>

Contents

- ☐ What is SQL?
- ☐ When do you need a Database?
- ☐ Uses of Databases
- ☐ Ways to Use SQL
- ☐ Some Relational Database Concepts
- ☐ Finding Your Way Around the Server
- ☐ What is SQL Server?
- ☐ Usage of SQL Server
- ☐ Versions of SQL Server
- ☐ SQL Server Components
- ☐ Instance of SQL Server
- ☐ Advantages of Instances
- ☐ SQL Server – Editions
- ☐ Create databases
- ☐ Create table

What is SQL?

- ❑ Structured Query Language
- ❑ Usually "talk" to a database server
- ❑ Used as front end to many databases (mysql, postgresql, oracle, Sybase,...)
- ❑ Three Subsystems: data description, data access and privileges
- ❑ Optimized for certain data arrangements
- ❑ The language is case-sensitive/insensitive, but you can use upper case for keywords.

When do you need a Database?

- ❑ Multiple simultaneous changes to data (concurrency)
- ❑ Data changes on a regular basis
- ❑ Large data sets where you only need some observations/variables
- ❑ Share huge data set among many people
- ❑ Rapid queries with no analysis
- ❑ Web interfaces to data, especially dynamic data

Uses of Databases

□ Traditional Uses:

- Live Queries
- Report Generation
- Normalization, foreign keys, joins, etc.

□ Newer uses:

- Storage - data is extracted and analyzed in another application
- Backends to web sites
- Traditional rules may not be as important

Ways to Use SQL

- ❑ console command
- ❑ GUI interfaces are often available
- ❑ Interfaces to many programming languages: R, python, perl, PHP, etc.
- ❑ SQLite - use SQL without a database server
- ❑ PROC SQL in SAS

Some Relational Database Concepts

- ❑ A database server can contain many databases
- ❑ Databases are collections of tables
- ❑ Tables are two-dimensional with rows (observations) and columns (variables)
- ❑ Limited mathematical and summary operations available
- ❑ Very good at combining information from several tables

Finding Your Way Around the Server

- Since a single server can support many databases, each containing many tables, with each table having a variety of columns, it's easy to get lost when you're working with databases. These commands will help figure out what's available:
 - `SHOW DATABASES;`
 - `SHOW TABLES IN database;`
 - `SHOW COLUMNS IN table;`
 - `DESCRIBE table;` - shows the columns and their types

□ Show some information with SQL Server

```
SELECT name, database_id, create_date  
FROM sys.databases;
```

```
SELECT name FROM master.dbo.sysdatabases
```

```
SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE  
TABLE_TYPE='BASE TABLE'
```

```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME =  
'GiangVien'
```

```
EXEC sp_help 'Sach'
```

```
select * from sys.all_columns where object_id =  
OBJECT_ID('TacGia')
```

```
SELECT s.name as schema_name, t.name as table_name,  
c.* FROM sys.columns AS c  
INNER JOIN sys.tables AS t ON t.object_id = c.object_id  
INNER JOIN sys.schemas AS s ON s.schema_id = t.schema_id  
WHERE t.name = 'Lecturers' AND s.name = 'dbo';
```

What is SQL Server?

- ❑ It is a software, developed by Microsoft, which is implemented from the specification of RDBMS.
- ❑ It is also an ORDBMS.
- ❑ It is platform dependent.
- ❑ It is both GUI and command based software.
- ❑ It supports SQL (SEQUEL) language which is an IBM product, non-procedural, common database and case insensitive language.

Usage of SQL Server

- ❑ To create databases.
- ❑ To maintain databases.
- ❑ To analyze the data through SQL Server Analysis Services (SSAS).
- ❑ To generate reports through SQL Server Reporting Services (SSRS).
- ❑ To carry out ETL operations through SQL Server Integration Services (SSIS)

Versions of SQL Server

Version	Year	Code Name
6.0	1995	SQL95
6.5	1996	Hydra
7.0	1998	Sphinx
8.0 (2000)	2000	Shiloh
9.0 (2005)	2005	Yukon
10.0 (2008)	2008	Katmai
0.5 (2008 R2)	2010	Kilimanjaro
11.0 (2012)	2012	Denali
12 (2014)	2014	Hekaton (initially), SQL 14 (current)

SQL Server Components

- ❑ SQL Server works in client-server architecture, hence it supports two types of components: (a) Workstation and (b) Server.
- ❑ Workstation components are installed in every device/SQL Server operator's machine. These are just interfaces to interact with Server components. Example: SSMS, SSCM, Profiler, BIDS, SQLEM etc.
- ❑ Server components are installed in centralized server. These are services. Example: SQL Server, SQL Server Agent, SSIS, SSAS, SSRS, SQL browser, SQL Server full text search etc.

Instance of SQL Server

- ❑ An instance is an installation of SQL Server.
- ❑ An instance is an exact copy of the same software.
- ❑ If we install 'n' times, then 'n' instances will be created.
- ❑ There are two types of instances in SQL Server a) Default b) Named.
- ❑ Only one default instance will be supported in one Server.
- ❑ Multiple named instances will be supported in one Server.
- ❑ Default instance will take the server name as Instance name.
- ❑ Default instance service name is MSSQLSERVER.
- ❑ 16 instances will be supported in 2000 version.
- ❑ 50 instances will supported in 2005 and later versions.

Advantages of Instances

- ❑ To install different versions in one machine.
- ❑ To reduce cost.
- ❑ To maintain production, development, and test environments separately.
- ❑ To reduce temporary database problems.
- ❑ To separate security privileges.
- ❑ To maintain standby server.

SQL Server – Editions

- ❑ SQL Server is available in various editions. This chapter lists the multiple editions with its features.
- ❑ Enterprise: This is the top-end edition with a full feature set.
- ❑ Standard: This has less features than Enterprise, when there is no requirement of advanced features.
- ❑ Workgroup: This is suitable for remote offices of a larger company.
- ❑ Web: This is designed for web applications.
- ❑ Developer: This is similar to Enterprise, but licensed to only one user for development, testing and demo. It can be easily upgraded to Enterprise without reinstallation.

SQL Server – Editions (cont'd)

- ❑ Express: This is free entry level database. It can utilize only 1 CPU and 1 GB memory, the maximum size of the database is 10 GB.
- ❑ Compact: This is free embedded database for mobile application development. The maximum size of the database is 4 GB.
- ❑ Datacenter: The major change in new SQL Server 2008 R2 is Datacenter Edition. The Datacenter edition has no memory limitation and offers support for more than 25 instances.

SQL Server – Editions (cont'd)

- ❑ Business Intelligence: Business Intelligence Edition is a new introduction in SQL Server 2012. This edition includes all the features in the Standard edition and support for advanced BI features such as Power View and PowerPivot, but it lacks support for advanced availability features like AlwaysOn Availability Groups and other online operations.
- ❑ Enterprise Evaluation: The SQL Server Evaluation Edition is a great way to get a fully functional and free instance of SQL Server for learning and developing solutions. This edition has a built-in expiry of 6 months from the time that you install it.

SQL Server – Editions (cont'd)

2005	2008	2008 R2	2012	2014
Enterprise	✓	✓	✓	✓
Standard	✓	✓	✓	✓
Developer	✓	✓	✓	✓
Workgroup	✓	✓	✗	✗
Win Compact Edition - Mobile	✓	✓	✓	✓
Enterprise Evaluation	✓	✓	✓	✓
Express	✓	✓	✓	✓
Web	✓	✓	✓	
Datacenter	✗	✗		
Business Intelligence	✓			

Data Types

- ❑ SQL supports a very large number of different formats for internal storage of information.
- ❑ Numeric
 - INTEGER, SMALLINT, BIGINT
 - NUMERIC(w,d), DECIMAL(w,d) - numbers with width w and d decimal places
 - REAL, DOUBLE PRECISION - machine and database dependent
 - FLOAT(p) - floating point number with p binary digits of precision

Data Types (cont'd)

☐ Character

- CHARACTER(L) - a fixed-length character of length L
- CHARACTER VARYING(L) or VARCHAR(L) – supports maximum length of L

☐ Binary

- BIT(L), BIT VARYING(L) - like corresponding characters
- BINARY LARGE OBJECT(L) or BLOB(L)

☐ Temporal

- DATE
- TIME
- TIMESTAMP

CREATE DATABASE

- Two ways:
 - Use SQL Server Management Studio
 - Use Transact-SQL

Use SQL Server Management Studio

□ To create a database

- In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
- Right-click Databases, and then select New Database.
- In New Database, enter a database name.
- To create the database by accepting all default values, select OK;
- See for more information:
<https://docs.microsoft.com/en-us/sql/relational-databases/databases/create-a-database?view=sql-server-ver15>

Use Transact-SQL

□ To create a database

- Connect to the Database Engine.
- From the Standard bar, select New Query.
- Copy and paste the following example into the query window and select Execute.

This example creates the database QLKH. Because the keyword PRIMARY is not used, the first file (QLKH_dat) becomes the primary file. Because neither MB nor KB is specified in the SIZE parameter for the QLKH _dat file, it uses MB and is allocated in megabytes. The QLKH_log file is allocated in megabytes because the MB suffix is explicitly stated in the SIZE parameter.

Use Transact-SQL (cont'd)

```
USE master;
GO
CREATE DATABASE QLKH
ON
( NAME = QLKH_dat,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\
  MSSQL13.MSSQLSERVER\MSSQL\DATA\QLKHdat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = QLKH_log,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\
  MSSQL13.MSSQLSERVER\MSSQL\DATA\QLKHlog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB );
GO
```

CREATE TABLE statement

- Suppose we have data measured on the height and weight of children over a range of ages. The first step is deciding on the appropriate variable types, and creating the table with the CREATE TABLE command.

```
CREATE TABLE kids(  
    id CHAR(6),  
    race SMALLINT,  
    age DECIMAL(6,3),  
    height DECIMAL(7,3),  
    weight DECIMAL(7,3),  
    sex SMALLINT);
```



Now, create three tables

```
CREATE TABLE Lecturers(  
  LID char(4) NOT NULL,  
  FullName nchar(30) NOT NULL,  
  Address nvarchar(50) NOT NULL,  
  DOB date NOT NULL,  
  CONSTRAINT pkLecturers PRIMARY KEY (LID)  
)
```

```
CREATE TABLE Projects(  
  PID char(4) NOT NULL,  
  Title nvarchar(50) NOT NULL,  
  Level nchar(12) NOT NULL,  
  Cost integer,  
  CONSTRAINT pkProjects PRIMARY KEY (PID)  
)
```

```
CREATE TABLE Participation(  
  LID char(4) NOT NULL,  
  PID char(4) NOT NULL,  
  Duration smallint,  
  CONSTRAINT pkParticipation PRIMARY KEY (LID, PID),  
  CONSTRAINT fk1 FOREIGN KEY (LID) REFERENCES Lecturers (LID),  
  CONSTRAINT fk2 FOREIGN KEY (PID) REFERENCES Projects (PID)  
)
```

```
INSERT INTO Lecturers VALUES('GV01',N'Vũ Tuyết Trinh',N'Hoàng Mai, Hà Nội',
'1975/10/10'),
('GV02',N'Nguyễn Nhật Quang',N'Hai Bà Trưng, Hà Nội','1976/11/03'),
('GV03',N'Trần Đức Khánh',N'Đồng Đa, Hà Nội','1977/06/04'),
('GV04',N'Nguyễn Hồng Phương',N'Tây Hồ, Hà Nội','1983/12/10'),
('GV05',N'Lê Thanh Hương',N'Hai Bà Trưng, Hà Nội','1976/10/10')
```

```
INSERT INTO Projects VALUES ('DT01',N'Tính toán lưới',N'Nhà nước','700'),
('DT02',N'Phát hiện tri thức',N'Bộ','300'),
('DT03',N'Phân loại văn bản',N'Bộ','270'),
('DT04',N'Dịch tự động Anh Việt',N'Trường','30')
```

```
INSERT INTO Participation VALUES ('GV01','DT01','100'),
('GV01','DT02','80'),
('GV01','DT03','80'),
('GV02','DT01','120'),
('GV02','DT03','140'),
('GV03','DT03','150'),
('GV04','DT04','180')
```