Database Management
Course Introduction

By: Meyer Tanuan (2022), Glenn Paulley (2015), John Mckay (2011)

Course Outline

- Course Description
- Resources
- Course Outcomes
- Unit Outcomes
- Evaluation – Exercises, Assignments, Exams
- Schedule (see Instructional Plan)

Optional Resources

- Microsoft SQL Server technical documentation
  - https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15
- “SQL: A Visual Quickstart Guide, 3rd Edition (2008)” by Chris Fehily, published by Peachpit Press. ISBN 978-0-321-55357-7.
- “Microsoft SQL Server 2012 – Step by Step (2014)” by Mike Hotek, published by Microsoft Press.
- Essential SQL: http://www.essentialsql.com/, authored by Kris Wenzel
- Stanford University online database and SQL course (curated by Jennifer Widom): https://class.stanford.edu/courses/Home/Databases/Engineering/about

Optional Videos: Database Management

- SS2019 Essentials: https://www.linkedin.com/learning/microsoft-sql-server-2019-essential-training [DB: LandonHotel]
- Querying SS2019: https://www.linkedin.com/learning/querying-microsoft-sql-server-2019 [DB: AdventureWorks2017]
- SS2016 Query Data (Subquery and CTE): https://www.linkedin.com/learning/microsoft-sql-server-2016-query-data/writing-subqueries [DB: AdventureWorks2014]
- DB Transactions: https://www.linkedin.com/learning/program-databases-with-transact-sql/ensure-data-consistency-with-transactions [DB: WideWorldImporters]
- What is NoSQL? https://www.linkedin.com/learning/nosql-essential-training [DB: AirBnB]

What is a relational database?

- A collection of shared, related objects that can be simultaneously queried and updated by multiple users at the same time
- Users are authenticated using either Windows authentication or a userid/password combination
- Object access requires access privileges that are controlled by either the owner of the object, or by a Database Administrator (a “super user”)
- A relational database guarantees the durability of updates even in the case of catastrophic failures, such as a loss of power to the machine

Fundamentals of relational databases

- The fundamental data object in a relational database is a **table**
- Tables consist of **columns** and **rows** (More formally: relations, attributes, tuples)
- Tables have primary keys and foreign keys which relate them to each other
- For the most part, the values stored in each column of each row is a single piece of data, such as the string “Canadian” or the number 42.
- Tables are defined and queried using a query language
- Microsoft SQL Server supports **SQL** (Structured Query Language); its dialect is called **Transact-SQL**
- Originally, Microsoft SQL Server came from Sybase in 1993

History (1)

- History of Relational Database Management Systems
- 1960’s – file systems (often tape)
- 1970’s – CODASYL and hierarchical (SYSTEM 2000, IMS) databases
  - Multi-user access, notion of transactions and serializability
  - Navigation interface API – write a program to retrieve data
  - IMS is still extremely popular; efficient and well-engineered
  - 1980’s – introduction of relational database systems
  - Oracle, Sybase SQL Server, INGRES are three pioneering products
  - Sybase introduced stored procedures in 1983
  - TERADATA introduced a hardware-based engine in 1984
  - Ad-hoc query languages (QUEL, SEQUEL, SQL)

History (2)

- The last two decades:
- Object-oriented database systems were introduced but have become niche products for specific application domains
- Relational systems have improved tremendously in Sophistication – Supports a wide variety of data types, including spatial data, structured types, and arrays – Much more powerful, feature-rich SQL language
- The last 10 years has seen a wave of research into cloud computing and a resurgence of simple data models and navigational interfaces to provide predictable performance

SQL Beginnings

- Structured Query Language (**SQL**) is a relational tuple-calculus language
  - Specifies what to retrieve, not how
- Requires a query optimizer to translate the request into an execution plan
  - Execution plans are typically based on relational algebra
- Won out over domain calculus languages such as QUEL (Ingres) and RDO from DEC, for Digital's Rdb product

SQL Standards

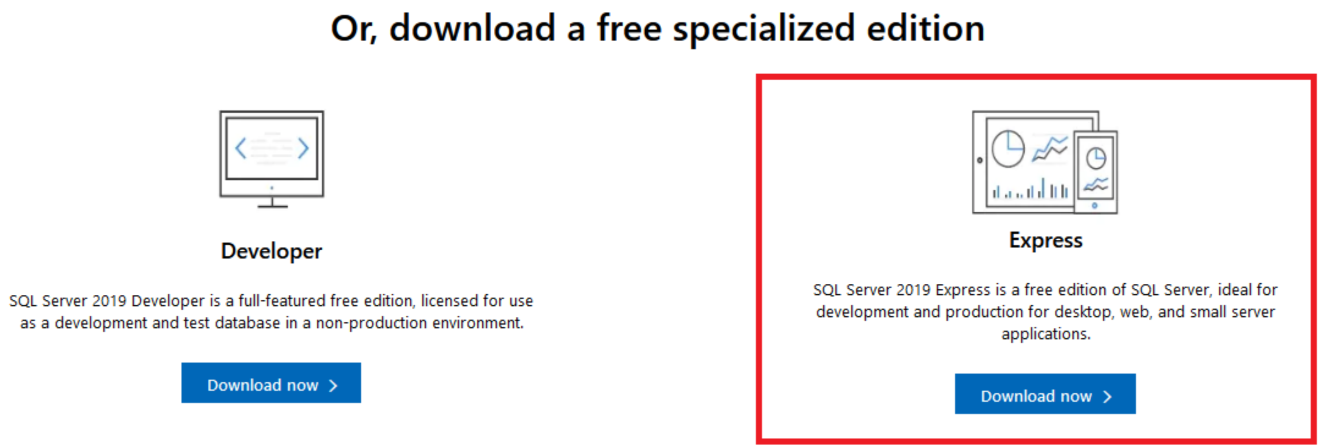
- In the early 1980’s, standardization efforts for a retrieval language for CODASYL (network) database was underway
- Language was called NDS
- IBM’s addition of SQL/DS to the marketplace in 1983, and DB2 in 1985, forced a change to standardize a new query language to support the relational model
- First attempt was to extend SEQUEL, an academic prototype
- The X3H2 committee decided to adopt SQL as the adhoc query language of choice, based on the availability of commercial implementations
- Other contenders were QUEL (Ingres) and RDO (Rdb from DEC)

Client Server vs. Personal Databases

- Microsoft SQL server is a “server-based” or “networked” database that permits client-server deployments
- Comparable products include:
  - Oracle
  - DB2 UDB, Informix (both IBM products)
  - SAP Adaptive Server Enterprise, SAP SQL Anywhere
  - Ingres (then Actian, now Ingres once again)
  - MySQL, PostgreSQL (both open source)
- Personal databases – designed to run in-process (linked with application code) or standalone, deployed with the application
  - SQL Anywhere, SQL Server for CE, Sybase UltraLite, SQL Server

Install SQL Server Express Edition

- Download and install **SQL Server Express 2019**
- Follow the instructions from:
  - **InstallingSQLServerExpress2019.pdf**



Or, download a free specialized edition

Developer

SQL Server 2019 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.

Download now >

Express

SQL Server 2019 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.

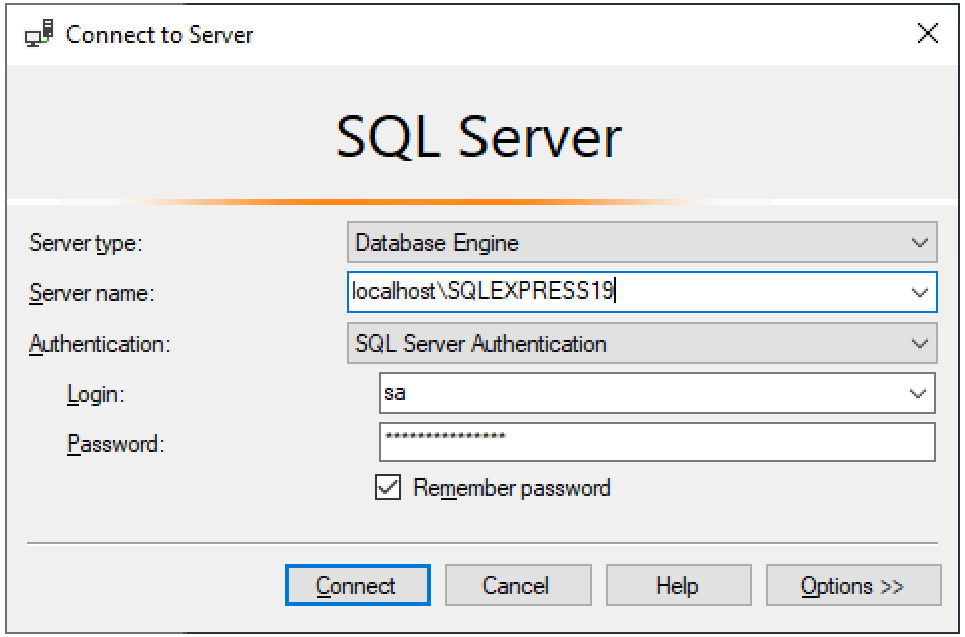
Download now >

MS SQL Docs: https://docs.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server?view=sql-server-ver15

CONESTOGA
Connect Life and Learning

Connect to SQL Server Express 2019

- At the end of the SSE install process, install SQL Server Management Studio (**SSMS**)
- Once installed, type **ssms** into the Windows search text to start the Management Studio application
- Once running, it will show a connection dialog.
- Server type: **Database Engine**
- Server name: **localhost\SQLEXPRESS19**
- Authentication: **SQL Server Authentication**
  - Login/Password: **sa/Conestoga1**
  - *Option 2: Use Windows Authentication*
- Press **Connect**
  - SQL Server Management Studio window appears



Connect to Server

SQL Server

Server type: Database Engine

Server name: localhost\SQLEXPRESS19

Authentication: SQL Server Authentication

Login: sa

Password: ****************

☑ Remember password

Create SIS Database

- File -> Open -> File… and load the file **CreateSISDatabase.sql** from eConestoga (download from **Databases** folder to your **C: drive** first)
- Press !Execute to create the **SIS** database
- Refresh the list of databases in the left-hand side bar by right-clicking on “**Databases**”, then “**Refresh**”
- From the left side bar, you can expand the **SIS** database to see the list of tables
- Note: The tables are “owned” by the user “**dbo**” (“database owner”)
- See “**SIS Database Diagrams.pdf**” on **eConestoga** for a sample database design

Playing Around (1)

- Right-click on a table *(e.g., dbo.Employee)* in the left-hand side bar and click on “SELECT TOP 1000 rows”
- You’ll see the first 1000 rows of that table, along with the SQL statement that was generated to produce that result
- To see all of the data in a table:

```
SELECT *
FROM table_name
```

- Example:

```
SELECT *
FROM dbo.Employee
```

Playing Around (2)

- To see all of the “User” tables in your database – execute the following script in a “New Query” window:

```
Use SIS; /* set the naming scope to objects within the SIS database */
SELECT *
FROM sysobjects
WHERE xtype = 'U'
```

- The **catalog** of a database is also a set of tables that can be queried just like any other table in the database