Microsoft SQL Server Design & Develop

Masoud Mirzakhani

Masoud Mirzakhani

Senior DW/ ETL/ BI Architect

Master of Science in Information Technology
QAZVIN AZAD UNIVERSITY

Bachelor of Science in Information Technology
PARAND AZAD UNIVERSITY

md.mirzakhani@gmail.com
(+98) 919 475 25 75
linkedin.com/in/masoudmirzakhani

Syllabus

Design and implement database objects

Design and Implement a Relational Database Schema	• Design tables and schemas based on business requirements.
	• Improve the design of tables by using normalization.
	• Write table create statements.
	• Determine the most efficient data types to use.
Ensure Data Integrity with Constraints	• Define table and foreign key constraints to enforce business rules.
	• Write **Transact-SQL** statements to add constraints to tables.
	• Identify results of Data Manipulation Language (**DML**) statements g iven existing tables and constraints.
	• Identify proper usage of **PRIMARY KEY** constraint.

Syllabus

Manage data with Transact-SQL

Create Transact-SQL SELECT queries	• Identify proper **SELECT** query structure.
	• Write specific queries to satisfy business requirements.
	• Construct results from multiple queries using set operators.
	• Distinguish between **UNION** and **UNION ALL** behavior.
	• Identify the query that would return expected results based on provided table structure and/or data.
Query multiple tables by using joins	• Write queries with join statements based on provided tables, data, and requirements.
	• Determine proper usage of **INNER JOIN**, **LEFT/RIGHT/FULL OUTER JOIN**, and **CROSS JOIN**.
	• Construct multiple **JOIN** operators using **AND** and **OR**.
	• Determine the correct results when presented with multi-table **SELECT** statements and source data;
	• Write queries with **NULLs** on joins.
Implement functions and aggregate data	• Construct queries using **scalar-valued** and **table-valued functions.**
	• Use built-in **aggregate functions.**

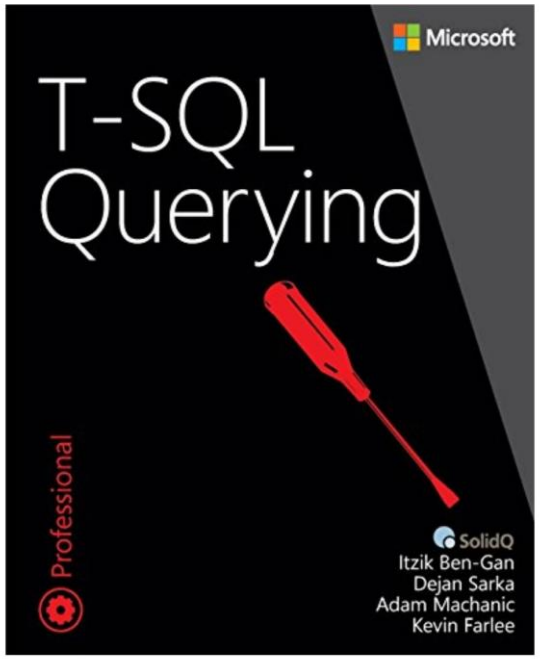
Syllabus

Query data with advanced Transact-SQL components

Query data by using Subqueries and APPLY	• Determine the results of queries using subqueries and table joins.
	• Evaluate performance differences between table joins and correlated subqueries based on provided data and query plans.
	• Distinguish between the use of **CROSS APPLY** and **OUTER APPLY.**
	• Write **APPLY** statements that return a given data set based on supplied data.
Group and pivot data by using queries	• Use **windowing functions** to group and rank the results of a query.
	• Distinguish between using **Windowing Functions** and **GROUP BY.**
	• Construct complex **GROUP BY** clauses using **GROUPING SETS**, and **CUBE.**
	• Construct **PIVOT** and **UNPIVOT** statements to return desired results based on supplied data.
	• Determine the impact of **NULL** values in **PIVOT** and **UNPIVOT** queries.

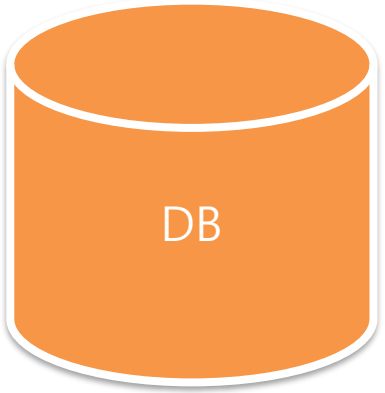
Reference

Itzik Ben-Gan



Question?

What is Database?



Spread Sheet VS Database

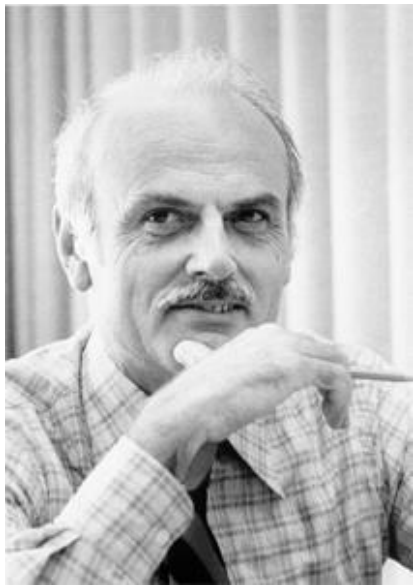
Spread Sheet	Database
Optimized for simple data analysis	Powerful relational analysis
Rich formatting features	Tabular report format
Limited ability to compare data from differ ent sources	Manage data from different sources
Self-contained documents	Dedicated database servers
Limited security options.	Permissions increase security
One user at a time	Multiple simultaneous users
Data Volume Restricted	Able to access and manage large amounts of data

History

- The **Navigational** Database Model (1960)
- The **Relational Database Model** (1970)
- **NoSQL** and **NewSQL** (2000)

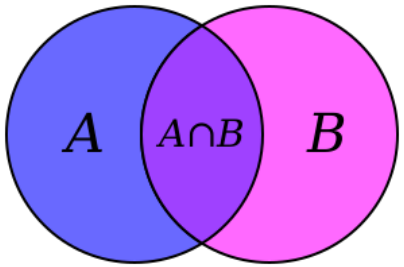
Relational Database Model

Edgar Frank "Ted" Codd (1923 – 2003)



Relational Database Model

- (1969)
- Set Theory
- Relational Algebra



Operation	My HTML	Symbol
Projection	PROJECT	π
Selection	SELECT	σ
Renaming	RENAME	ρ
Union	UNION	$\cup$
Intersection	INTERSECTION	$\cap$
Assignment	<-	$\leftarrow$

Operation	My HTML	Symbol
Cartesian product	X	$\times$
Join	JOIN	$\bowtie$
Left outer join	LEFT OUTER JOIN	$⟕$
Right outer join	RIGHT OUTER JOIN	$⟖$
Full outer join	FULL OUTER JOIN	$⟗$
Semijoin	SEMIJOIN	$\ltimes$

Relational Database Model

Enterprise	Open Source
IBM DB2	MySQL
Microsoft Access	PostgreSQL
Microsoft SQL Server	SQLite
Oracle	
SAP HANA	
Teradata	

SEQUEL

- ***SEQUEL*** (***S****tructured* ***E****nglish* ***Q****uery* ***L****anguage*)
- (1970)
- IBM
- Donald D. Chamberlin and Raymond F. Boyce

- The acronym **SEQUEL** changed to **SQL**
- ***S****tructured* ***Q****uery* ***L****anguage*
- **ANSI** and **ISO** adopted the standard "Database Language SQL"
- (1986)

SQL Languages

RDBMS	Language	Full Name
IBM DB2	SQL PL	SQL Procedural Language
Microsoft SQL Server	T-SQL	Transact-SQL
MySQL	SQL/PSM	SQL/Persistent Stored Module
Oracle	PL/SQL	Procedural Language/SQL
PostgreSQL	PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language
SAP HANA	SQLScript	SQL Script
Teradata	SPL	Stored Procedural Language

حساب و کتاب یه بقالی

Factor						
Total Price	Quantity	Unit Price	Product	Date	Customer Name	Factor Number
2000	2	1000	Snack	1398-01-01	Ali	1
15000	3	5000	Chips	1398-01-01	Ali	1
8000	2	4000	Coca	1398-01-01	Ali	1
20000	4	5000	7UP	1398-01-02	Ahmad	2
20000	4	5000	Pepsi	1398-01-02	Ahmad	2
20000	4	5000	Fanta	1398-01-02	Ahmad	2

حساب و کتاب یه بقالی

Factor Header		
Date	Customer Name	Factor Number
1398-01-01	Ali	1
1398-01-02	Ahmad	2

Factor Detail				
Total Price	Quantity	Unit Price	Product	Factor Number
2000	2	1000	Snack	1
15000	3	5000	Chips	1
8000	2	4000	Coca	1
20000	4	5000	7UP	2
20000	4	5000	Pepsi	2
20000	4	5000	Fanta	2

حساب و کتاب یه بقالی

Factor Header		
Date	Customer Code	Factor Number
1398-01-01	11	1
1398-01-02	2	2

Customer			
Customer Code	Customer Name	Birth Date	Sex
1	Ali		
2	Ahmad		

Factor Detail					
Total Price	Quantity	Unit Price	Product Code	Row Number	Factor Number
2000	2	1000	1	1	1
15000	3	5000	2	2	1
8000	2	4000	3	3	1
20000	4	5000	1	1	2
20000	4	5000	5	2	2
20000	4	5000	6	3	2

Product	
Product Name	Product Code
Snack	1
Chips	2
Coca	3
7UP	4
Pepsi	5
Fanta	6

حساب و کتاب یه بقالی

Factor Header		
Date *	Customer Code *	Factor Number *U
تاریخ	عدد	عدد
1398-01-01	11	1
1398-01-02	2	2

Customer			
Customer Code *U	Customer Name *	Birth Date	Sex
عدد	متن	تاریخ	؟
1	Ali		
2	Ahmad		

Factor Detail					
Total Price *	Quantity *	Unit Price *	Product Code *	Row Number *U1	Factor Number * U1
2000	2	1000	1	1	1
15000	3	5000	2	2	1
8000	2	4000	30	3	1
20000	4	5000	1	1	2
20000	4	5000	5	2	20
20000	4	5000	6	3	2

Product	
Product Name *	Product Code *U
Snack	1
Chips	2
Coca	3
7UP	4
Pepsi	5
Fanta	6

Naming Conventions

Rule	Example
Plural ‘S’	Customers
Finglish	Kala
English Alphabetic Characters	Factor-Header
Abbreviation	TelNo
KeyWord	Date

Multiple Word Identifiers

Title	Example
Snake Case	snake_case
Snake Case (All Caps)	SNAKE_CASE
Camel Case	camelCase
Kebab-case	kebab-case
Pascal Case	PascalCase