Database Management Systems

Module 60: Widely Used DBMSs and Course Summarization

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Module Recap

- Understood the issues in Big Data
- Understood the approach of NOSQL and CAP theorem viz-a-viz ACID
- Took a tour of common types of NOSQL database
- Compared Relational with Non-relational

Module Objectives

- The space of RDBMSs is crowded. We take a look into widely used RDBMS systems
- We recap the weeks of the course

Module Outline

- Widely Used RDBMSs
- Course Recap

Widely used RDBMS

Ref: https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems(Accessed:26-08-2021)
Ref: http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38151.1540/doc/html/san1278453579697.html(Accessed:26-08-2021)
Ref: https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.welcome.doc/doc/welcome.html(Accessed:26-08-2021)
Ref: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-develop-cplusplus-simple(Accessed:26-08-2021)

Relational Databases

- The relational model of data organizes data into one or more tables (or *relations*) of *rows* and *columns*, with a *unique key* for each row
- Since each row in a table has its own unique key, rows in a table can be linked to rows in other tables by storing the unique key of the row to which it should be linked (where such unique key is known as a *foreign key*)
- Mostly, the relational databases use *SQL* as the language for *querying and maintaining the database*
- The reasons for the dominance of relational databases are:
  - simplicity,
  - robustness,
  - flexibility,
  - performance,
  - scalability, and
  - compatibility in managing generic data
- The RDBMSs are mostly used in large enterprise scenarios

Widely used RDBMS

- **Commercial / Proprietary** with *Market Share*[1]
  - **Oracle** (Oracle): *Market Share of Oracle: 48.8%*
  - **Db2** (IBM): *Market Share of IBM: 20.2%*
  - **SQL Server** (Microsoft): *Market Share of Microsoft: 17.0%*
  - **Sybase** (Sybase Corporation / SAP AG): *Market Share of SAP: 4.7%*
  - **Teradata** (Caltech and Citibank): *Market Share of Teradata: 3.7%*
  - Others: Microsoft Access, Microsoft Azure SQL Database
- **Free / GPL**[2] **/ Open Source**
  - **PostgreSQL** (PostgreSQL Global Development Group)
  - **MySQL** (MySQL AB / Oracle Corporation)
  - **SQLite** (SQLite Developers)
  - Others: MariaDB, Hive
- **Object–Relational Database (ORD) or Object–RDBMS (ORDBMS)**
  - **Illustra** (Informix / IBM)
  - **Objectivity/DB** (Objectivity, Inc.)

[1] Gartner, in 2011, listed the five leading proprietary software relational database vendors by revenue

[2] GNU General Public License (*GPLv3*)

Global DBMS Software Market Share (%): 2021

Company Name	DBMS Market Share
Oracle	45.60 %
Microsoft	19.10 %
IBM	15.70 %
SAP	9.60 %
Teradata	3.20 %
Others	6.80 %

Source: *DBMS Customers List* (Accessed 28-Aug-21)

DB-Engines Ranking (August 2021): Relational DBMS

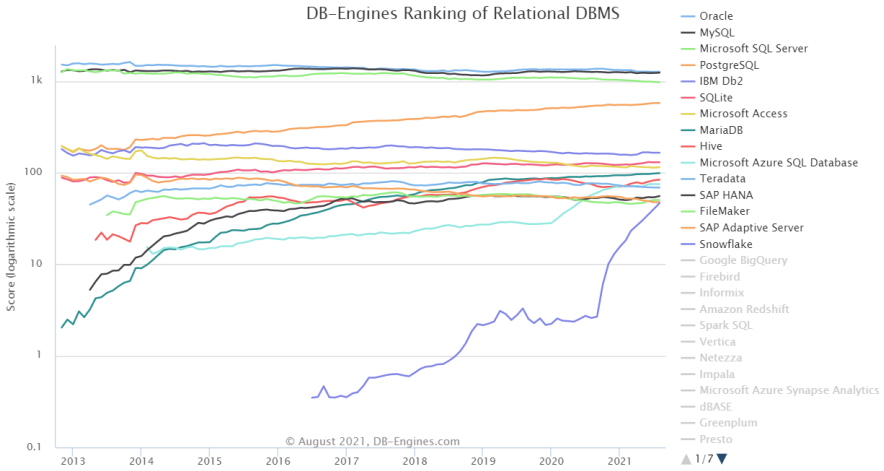
☐ include secondary database models

147 systems in ranking, August 2021

Rank			DBMS	Database Model	Score		
Aug 2021	Jul 2021	Aug 2020			Aug 2021	Jul 2021	Aug 2020
1.	1.	1.	Oracle	Relational, Multi-model	1269.26	+6.59	-85.90
2.	2.	2.	MySQL	Relational, Multi-model	1238.22	+9.84	-23.36
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	973.35	-8.61	-102.53
4.	4.	4.	PostgreSQL	Relational, Multi-model	577.05	-0.10	+40.28
5.	5.	5.	IBM Db2	Relational, Multi-model	165.46	+0.31	+3.01
6.	6.	6.	SQLite	Relational	129.81	-0.39	+3.00
7.	7.	7.	Microsoft Access	Relational	114.84	+1.39	-5.02
8.	8.	8.	MariaDB	Relational, Multi-model	98.98	+0.99	+8.06
9.	9.	↑ 10.	Hive	Relational	83.93	+1.26	+8.64
10.	10.	↑ 11.	Microsoft Azure SQL Database	Relational, Multi-model	75.15	-0.06	+18.31
11.	11.	↓ 9.	Teradata	Relational, Multi-model	68.82	-0.13	-7.96
12.	12.	↑ 13.	SAP HANA	Relational, Multi-model	55.57	+1.76	+2.46
13.	13.	↑ 14.	FileMaker	Relational	50.28	-0.01	+2.24
14.	14.	↓ 12.	SAP Adaptive Server	Relational, Multi-model	47.61	-0.04	-6.35
15.	↑ 16.	↑ 54.	Snowflake	Relational	46.54	+6.50	+43.95
16.	↓ 15.	↓ 15.	Google BigQuery	Relational	42.03	+0.77	+9.43
17.	17.	↑ 19.	Firebird	Relational	25.35	+0.26	+4.42
18.	↑ 19.	↓ 16.	Informix	Relational, Multi-model	23.02	-0.21	-1.35
19.	↓ 18.	↓ 17.	Amazon Redshift	Relational	22.86	-0.67	+0.49
20.	20.	↑ 21.	Spark SQL	Relational	20.56	-0.83	+1.94

Source: *DB-Engines Ranking of Relational DBMS* (Accessed 28-Aug-21)

DB-Engines Ranking (August 2021): Trend of Relational DBMS Popularity



Source: *DB-Engines Ranking - Trend of Relational DBMS Popularity* (Accessed 28-Aug-21)

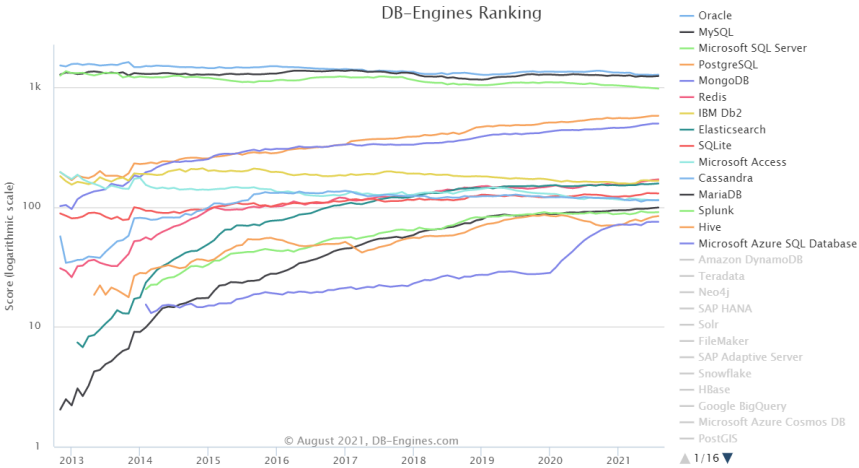
DB-Engines Ranking (August 2021): Complete

373 systems in ranking, August 2021

Rank			DBMS	Database Model	Score		
Aug 2021	Jul 2021	Aug 2020			Aug 2021	Jul 2021	Aug 2020
1.	1.	1.	Oracle	Relational, Multi-model	1269.26	+6.59	-85.90
2.	2.	2.	MySQL	Relational, Multi-model	1238.22	+9.84	-23.36
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	973.35	-8.61	-102.53
4.	4.	4.	PostgreSQL	Relational, Multi-model	577.05	-0.10	+40.28
5.	5.	5.	MongoDB	Document, Multi-model	496.54	+0.38	+52.98
6.	6.	↑ 7.	Redis	Key-value, Multi-model	169.88	+1.58	+17.01
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	165.46	+0.31	+3.01
8.	8.	8.	Elasticsearch	Search engine, Multi-model	157.08	+1.32	+4.76
9.	9.	9.	SQLite	Relational	129.81	-0.39	+3.00
10.	↑ 11.	10.	Microsoft Access	Relational	114.84	+1.39	-5.02
11.	↓ 10.	11.	Cassandra	Wide column	113.66	-0.35	-6.18
12.	12.	12.	MariaDB	Relational, Multi-model	98.98	+0.99	+8.06
13.	13.	13.	Splunk	Search engine	90.60	+0.55	+0.69
14.	14.	↑ 15.	Hive	Relational	83.93	+1.26	+8.64
15.	15.	↑ 17.	Microsoft Azure SQL Database	Relational, Multi-model	75.15	-0.06	+18.31
16.	16.	16.	Amazon DynamoDB	Multi-model	74.90	-0.30	+10.15
17.	17.	↓ 14.	Teradata	Relational, Multi-model	68.82	-0.13	-7.96
18.	18.	↑ 21.	Neo4j	Graph	56.95	-0.21	+6.77
19.	19.	19.	SAP HANA	Relational, Multi-model	55.57	+1.76	+2.46
20.	20.	20.	Solr	Search engine, Multi-model	51.06	-0.73	-0.63

Source: *DB-Engines Ranking* (Accessed 28-Aug-21)

DB-Engines Ranking (August 2021): Trend Popularity



Source: *DB-Engines Ranking - Trend Popularity* (Accessed 28-Aug-21)

Oracle

- Multi-model commercial DBMS produced and marketed by **Oracle Corporation**.
- Larry Ellison, Bob Miner and Ed Oates started a consultancy called Software Development Laboratories (SDL) in 1977, and developed the original version of Oracle.
- Latest Version: **Oracle Database 19c** is the current long term release. Oracle Database 21c is available for production use as an innovation release (August 2021)
- Application Domains: Online Transaction Processing (OLTP), Data Warehousing (DW) and Mixed (OLTP & DW) database workloads including Oracle Human Capital Management (HCM), Oracle Enterprise Resource Planning (ERP), Oracle Customer Experience (CX), Oracle Supply Chain Management (SCM), Oracle Enterprise Performance Management (EPM), Oracle Construction and Engineering
- Languages: Structured Query language (SQL), Procedural SQL (PL-SQL)
- Tools / Editions: Oracle SQL Developer, Oracle Forms, Oracle Jdeveloper, Oracle Reports for development of applications, Oracle Live SQL for test environment
- Connectivity: Java (JDBC), Microsoft.NET (ODP.NET), C/C++ (OCI, ODBC, ODPI-C), Python (cx_Oracle)

Db2

- Db2 contains database-server products developed by **IBM**. Mostly relational models, but now includes object relational models
- In 1970, Edgar F.Codd, researcher in IBM published the model for data manipulation.
- Latest Version: **Db2 11.5** (June 2019)
- Application Domains: Online Transaction Processing (OLTP), Data Warehousing (DW) and Mixed (OLTP & DW) database workloads
- Languages: Structured Query language (SQL), XML Query
- Tools / Editions: Advanced Enterprise Server Edition, Enterprise Server Edition, Advanced Workgroup Server Edition, Workgroup Server Edition, Direct and Developer Editions and Express-C.
- Connectivity: C/C++, Java, Ruby, Perl through a package of DB2 API's

SQL Server

- Relational database management system developed by **Microsoft**.
- SQL Server 1.0, a 16-bit server for the OS/2 operating system in 1989
- Latest Version: **Microsoft SQL Server 2019** (November 2019)
- Application Domains: Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP)
- Languages: Transact SQL
- Tools / Editions: Enterprise, Standard, Web, Business Intelligence, WorkGroup, Express
- Connectivity: Java (JDBC), C/C++ (ODBC)

Sybase

- Relational model database server product for businesses developed by **Sybase Corporation** which became part of **SAP AG**.
- Originally meant for Unix platforms in 1987, Sybase Corporation's primary DBMS product was initially marketed under the name Sybase SQL Server.
- Latest Version: **SAP ASE 16** (April 2014)
- Languages: Sybase IQ, Transact-SQL
- Tools / Editions: Sybase SQL server for development of applications. Has a developer and express edition.
- Connectivity: C/C++ (SQLAPI++), Java (JDBC)

Teradata

- Relational database management system developed by **Caltech and Citibank's advanced technology group**
- In 1984, the first version of Teradata was released
- Latest Version: **Teradata 17.10.08.00** (August 2021)
- Application Domains: Online Transaction Processing (OLTP), Data Warehousing (DW) and Mixed (OLTP & DW) database workloads
- Languages: BTEQ (Basic Teradata Query)
- Tools / Editions: Developer Edition, Express Edition
- Connectivity: Java (JDBC), C/C++ (ODBC)

PostgreSQL

- Open source relational database management system produced by **PostgreSQL Global Development Group**, a diverse group of many companies and individual contributors.
- First version in 1988 by researchers of POSTGRES project
- Latest Version: **PostgreSQL 14.0** (June, 2021)
  - For this course, we using **PostgreSQL 10.18** (*Download Link*)
- Application Domains: Online Transaction Processing (OLTP), Data Warehousing (DW) and Mixed (OLTP & DW) database workloads, Supports Big Data Analytics
- Languages: Structured Query language (SQL), Procedural SQL (PL- SQL)
- Connectivity: Java (JDBC), Microsoft.NET (npgsql), C/C++ (libpq), Python (psycopg2 and several others)

MySQL

- Open source relational database management system produced by **Swedish company MySQL AB, owned by Oracle Corporation**
- First internal release on 23 May 1995
- Latest Version: **MySQL 8.0.26** (July 2021)
- Application Domains: Online Transaction Processing (OLTP), Data Warehousing (DW) and Mixed (OLTP & DW) database workloads
- Languages: Structured Query language (SQL), Procedural SQL (PL- SQL)
- Connectivity: Java (JDBC), Microsoft.NET (ADO.NET), C/C++ (ODBC)

SQLite

- SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine
- It is in the public domain and is thus free for use for any purpose, commercial or private
- It is an RDBMS contained in a C library and is not a client–server database engine. Rather, it is embedded into the end program
- It is supported by an international team of developers who work on SQLite full-time
- First release on 29 May 2000
- Latest Version: **SQLite 3.36.0** (June 2021)
  - For Application Development course, we are going to use **SQLite**. Check version from Instructor
- Application Domains:
  - Photoshop Lightroom (Adobe), A350 XWB family of aircraft (Airbus), GM, Nissan, and Suzuki automobiles (Bosch), Dropbox, osquery (Facebook), Android cell-phone OS and Chrome Web Browser (Google), Library of Congress, McAfee, Firefox, etc.
- Languages: Structured Query language (SQL)
- Connectivity: Java (JDBC), Microsoft.NET (Microsoft.Data.Sqlite), C/C++ (SQLite C/C++ Interface), Python (sqlite3)

Object–Relational Database (ORD) or Object–RDBMS (ORDBMS)

- Combines database capabilities with object oriented programming language capabilities
- *Objects have a many to many relationship and are accessed by the use of pointers*
- *Access to data can be faster because an object can be retrieved directly without a search, by following pointers*
- Most object databases also offer some kind of query language, allowing objects to be found using a declarative programming approach
- Examples:
  - **Illustra**: A commercialized version of the Postgres ORD. It was sold to Informix Corp. in 1997, folded into the Informix 7 Product Line, eventually sold to IBM
  - **Objectivity/DB**: It is a commercial ORD by Objectivity, Inc. It allows applications to make standard C++, C#, Java, or Python objects persistent without having to convert the data objects into the rows and columns used by a RDBMS. It supports OO languages, SQL/ODBC and XML
  - **SQL:1999**: Many of the ideas of early ORD efforts have largely become incorporated into SQL:1999 via structured types. Any product compliant to OO features of SQL:1999 could be described as an ORD product. For example, Db2, Oracle, and SQL Server, make claims to support this technology and do so with varying degrees of success

Source: *Object–relational database*, *Object database*

Parameters

- We compare the RDMBSs based on the following parameters:
  - OS support
  - Fundamental features
  - Limits
  - Tables and views
  - Indexes
  - Database capabilities
  - Data types
  - Other objects
  - Partitioning
  - Access control
  - Programming Language Support

Source: *Comparison of relational database management systems*

OS support

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Linux, Window, Mac, Unix, Haiku, z/OS, OpenVMS	Linux, Window, Mac, Unix, BareMetal, Android, Solaris	Linux, Window	Linux, Window, Mac (Express-C image), OS/2, Unix, z/OS, iOS	Linux, Window, Mac, Unix,	Linux, Window, Mac, Unix, BSD, Solaris AmigaOS, z/OS, Android	Linux, Window, Mac, Unix, BSD, AmigaOS, z/OS, Android

Basic Features

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports ACID properties for transactions, implicit commit for DDL, referential integrity, row level locking for fine grained locking, Concurrency control	Supports ACID properties for transactions, referential integrity, row level locking for fine grained locking Concurrency control	Supports ACID properties for transactions, referential integrity, row level locking for fine grained locking, Concurrency control	Supports ACID properties for transactions, referential integrity, row level locking for fine grained locking, Concurrency control	Supports ACID properties for transactions, referential integrity, hash and partition for fine grained locking, Concurrency control	Supports ACID properties for transactions, referential integrity, row level locking for fine grained locking, Concurrency control	Supports ACID properties for transactions, referential integrity, row level locking for fine grained locking, Concurrency control

Limits

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Max DB Size: 8589 PB	Max DB Size: 104TB	Max DB Size: 524,272 TB	Max DB Size: Unlimited	Max DB Size: Unlimited	Max DB Size: Unlimited	Max DB Size: Unlimited
Max Table Size: 4GB * block size	Max Table Size: File size	Max Table Size: 524,272 TB	Max Table Size: 2 ZB	Max Table Size: Unlimited	Max Table Size: 32 TB	Max Table Size: 256 TB
Max Row Size: 8KB	Max Row Size: File size	Max Row Size: 2TB	Max Row Size: 32,677 B	Max Row Size: 64 GB	Max Row Size: 1.6TB	Max Row Size: 64KB
Max Column per Row: 1,000	Max Column per Row: 45,000	Max Column per Row: 1,024	Max Column per Row: 1,012	Max Column per Row: 2048	Max Column per Row: 1600	Max Column per Row: 4096
Max CHAR size: 32,767 B	Max CHAR size: 2GB	Max CHAR size: 2GB	Max CHAR size: 32 KB	Max CHAR size: 64,000 bits	Max CHAR size: 1GB	Max CHAR size: 64 KB
Max Number size: 126 bits	Max Number size: 64 bits	Max Number size: 126 bits	Max Number size: 64 bits	Max Number size: 38 bits	Max Number size: Unlimited	Max Number size: 64 bits
Max Column Name size: 128		Max Column Name size: 128	Max Column Name size: 128	Max Column Name size: 128	Max Column Name size: 63	Max Column Name size: 64

Tables and Views

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports Temporary tables and Materialised views (apart from basic)	Supports Temporary tables and Materialised views (apart from basic)	Supports Temporary tables and Materialised views (apart from basic)	Supports Temporary tables and Materialised views (apart from basic)	Supports Temporary tables and Materialised views (apart from basic)	Supports Temporary tables and Materialised views (apart from basic)	Supports Temporary tables (apart from basic)

Type System

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Static+Dynamic	Static	Static	Static+Dynamic	Static	Static	Static

Comparative Study

Data Types

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; And other miscellaneous types like Spacial, Image, Audio, Dicom, Video	Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; Bit And other miscellaneous types like Money	Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; Bit And other miscellaneous types like Timestamp, Rowversion, UniqueIdentifier identity	Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; And other miscellaneous types like Graphic, Vargraphic, xml, DbClob	Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; And other miscellaneous types like Period, Interval, Geometry, xml, json	Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; Boolean And other miscellaneous types like Enum, xml Circle, Path, UUID	Supports various variants of Integer; Floating Point; Decimal; String; Binary; Date/Time; Bit And other miscellaneous types like Enum, Set, MultiCurve, Geometry, LineString, Surface, Polygon

Indexes

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports R/R++, Hash, Partial, Bitmap, Reverse Apart from Basic B/B++ indexes	Supports only Basic B/B++ indexes	Supports R/R++, Hash, Partial, Bitmap, Reverse Apart from Basic B/B++ indexes	Supports R/R++, Hash, Partial, Bitmap, Reverse Apart from Basic B/B++ indexes	Supports Hash, Partial, Bitmap, Apart from Basic B/B++ indexes	Supports R/R++, Hash, Partial, Bitmap, Reverse Apart from Basic B/B++ indexes	Supports R/R++, Hash, Apart from Basic B/B++ indexes

Comparative Study

Database Capabilities

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports Union, Intersect, Inner Joins, Outer Joins, Except, Inner Selects, Merger Joins, Blobs and Clobs, Common Table Expressions, Windowing Functions, Parallel Query	Supports Union, Intersect, Inner Joins, Outer Joins, Except, Inner Selects, Merger Joins, Blobs and Clobs, Common Table Expressions, Windowing Functions, Parallel Query	Supports Union, Intersect, Inner Joins, Outer Joins, Except, Inner Selects, Merger Joins, Blobs and Clobs, Common Table Expressions, Windowing Functions, Parallel Query	Supports Union, Intersect, Inner Joins, Outer Joins, Except, Inner Selects, Merger Joins, Blobs and Clobs, Common Table Expressions, Windowing Functions, Parallel Query	Supports Union, Intersect, Inner Joins, Outer Joins, Except, Inner Selects, Merger Joins, Blobs and Clobs, Common Table Expressions, Windowing Functions, Parallel Query	Supports Union, Intersect, Inner Joins, Outer Joins, Except, Blobs and Clobs	Supports Union, Outer Joins, Except, Inner Selects, Blobs and Clobs, Common Table Expressions

Other Objects

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports Data Domain, Cursor, Trigger, Function, Procedure, External Routine	Supports Data Domain, Cursor, Trigger, Function, Procedure, External Routine	Supports Data Domain, Cursor, Trigger, Function, Procedure, External Routine	Supports Data Domain, Cursor, Trigger, Function, Procedure, External Routine	Supports Cursor, Trigger, Function, Procedure, External Routine	Supports Data Domain, Cursor, Trigger, Function, Procedure, External Routine	Supports Cursor, Trigger, Function, Procedure, External Routine

Partitioning

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports Range, Hash, Composite, List	Supports none	Supports Range, Hash, Composite, List	Supports Range, Hash, Composite, List	Supports Range, Hash, Composite, List	Supports Range, Hash, Composite, List	Supports Range, Hash, Composite, List

Access Control

Oracle	Sybase	SQL Server	DB2	Teradata	PostgreSQL	MySQL
Supports Native network encryption, Separation of Duties, Password Complexity Rules, Enterprise Directory compatibility, Audit, Resource Limit,	Supports Native network encryption, Separation of Duties, Password Complexity Rules, Enterprise Directory compatibility, Audit, Resource Limit,	Supports Native network encryption, Separation of Duties, Password Complexity Rules, Enterprise Directory compatibility, Audit, Resource Limit, Patch Access	Supports Native network encryption, Separation of Duties, Password Complexity Rules, Enterprise Directory compatibility, Audit, Resource Limit,	Supports Native network encryption, Separation of Duties, Password Complexity Rules, Enterprise Directory compatibility, Audit, Resource Limit, Patch Access	Supports Native network encryption, Separation of Duties, Password Complexity Rules, Enterprise Directory compatibility, Audit, Resource Limit, Patch Access	Supports Native network encryption, Enterprise Directory compatibility, Patch Access

Course Recap

- **Module 01: Course Overview**
  - Why Databases?
  - KYC: Know Your Course
- **Module 02: Why DBMS?/1**
  - Evolution of Data Management
  - History of DBMS
- **Module 03: Why DBMS?/2**
  - File Systems vs Databases
- **Module 04: Introduction to DBMS/1**
  - Levels of Abstraction
  - Schema and Instance
  - Data Models
  - DDL and DML
  - SQL
  - Database Design
- **Module 05: Introduction to DBMS/2**
  - Database Design
  - Database Engine
  - Database Users and Administrators

- **Module 06: Introduction to Relational Model/1**
  - Example of a Relation
  - Attributes
  - Schema and Instance
  - Keys
  - Relational Query Languages
- **Module 07: Introduction to Relational Model/2**
  - Relational Operators
  - Aggregation Operators
- **Module 08: Introduction to SQL/1**
  - History of SQL
  - Data Definition Language
  - Data Manipulation Language
- **Module 09: Introduction to SQL/2**
  - Additional Basic Operations
    - Cartesian Product
    - Rename AS
    - String Values
    - Order By Clause
    - Select Top/Fetch Clause
    - Where Clause Predicates
    - Duplicates
- **Module 10: Introduction to SQL/3**
  - Set Operations
  - Null Values
  - Aggregate Functions
    - Group By
    - Having
    - Null Values

- **Module 11: SQL Examples**
  - Cartesian Product
  - Rename AS
  - Where AND/OR
  - String Values
  - Order By Clause
  - in
  - Set Operations
  - Aggregation Operations
- **Module 12: Intermediate SQL/1**
  - Nested Subqueries
  - Modification of the Database
- **Module 13: Intermediate SQL/2**
  - Join Expressions
  - Views
- **Module 14: Intermediate SQL/3**
  - Transactions
  - Integrity Constraints
  - SQL Data Types and Schemas
  - Authorization
- **Module 15: Advanced SQL**
  - Functions and Procedural Constructs
  - Triggers

- **Module 16: Formal Relational Query Languages/1**
  - Relational Algebra
- **Module 17: Formal Relational Query Languages/2**
  - Predicate Logic
  - Tuple Relational Calculus
  - Domain Relational Calculus
  - Equivalence of Algebra and Calculus
- **Module 18: Entity-Relationship Model/1**
  - Design Process
  - ER Model
- **Module 19: Entity-Relationship Model/2**
  - ER Diagram
  - ER Model to Relational Schema
- **Module 20: Entity-Relationship Model/3**
  - ER Features

- **Module 26: Relational Database Design/6: Normal Forms**
  - Normal Forms
- **Module 27: Relational Database Design/7: Normal Forms**
  - Decomposition to 3NF
  - Decomposition to BCNF
- **Module 28: Relational Database Design/8: Case Study**
  - Library Information System (LIS) (*Specification of LIS shared separately*)
- **Module 29: Relational Database Design/9: MVD and 4NF**
  - Multivalued Dependencies
  - Decomposition to 4NF
- **Module 30: Relational Database Design/10: Design Summary and Temporal Data**
  - Database-Design Process
  - Temporal Databases

- **Module 31: Application Design and Development/1: Architecture**
  - Application Programs and Architecture
- **Module 32: Application Design and Development/2: Web Applications**
  - WWW
  - Scripting
- **Module 33: Application Design and Development/3: SQL and Native Language**
  - SQL and Native Language
  - ODBC
  - JDBC
  - Bridge
  - Embedded SQL
- **Module 34: Application Design and Development/4: Python and PostgreSQL**
  - PostgreSQL and Python
  - Python Frameworks for PostgresSQL
  - Flask
- **Module 35: Application Design and Development/5: Application Development and Mobile**
  - Rapid Application Development
  - Application Performance and Security
  - Challenges in Web Application Development
  - Mobile Apps

Week 08: Storage Management

- **Module 36: Algorithms and Data Structures/1: Algorithms and Complexity Analysis**
  - Algorithms
  - Analysis of Algorithms
  - Complexity Chart
- **Module 37: Algorithms and Data Structures/2: Data Structures/1**
  - Data Structures
  - Linear Data Structures
  - Linear and Binary Search
- **Module 38: Algorithms and Data Structures/3: Data Structures/2**
  - Data Structures
  - Non-linear Data Structures
  - Binary Search Tree
  - Comparison
- **Module 39: Storage and File Structure/1: Physical Storage**
  - Overview of Physical Storage Media
  - Magnetic Disk
  - Magnetic Tapes
  - Cloud Storage
  - Other Storage
  - Future of Storage
- **Module 40: Storage and File Structure/2: File Structure**
  - File Organization
  - Organization of Records in Files
  - Data Dictionary Storage
  - Storage Access

- **Module 41: Indexing and Hashing/1: Indexing/1**
  - Concepts of Indexing
  - Ordered Indices
- **Module 42: Indexing and Hashing/1: Indexing/2**
  - Balanced Binary Search Trees
  - 2-3-4 Tree
- **Module 43: Indexing and Hashing/1: Indexing/3**
  - B^+-Tree Index Files
  - B-Tree Index Files
- **Module 44: Indexing and Hashing/4: Hashing**
  - Static Hashing
  - Dynamic Hashing
  - Comparison Schemes
  - Bitmap Indices
- **Module 45: Indexing and Hashing/5: Index Design**
  - Index Definition in SQL
  - Guidelines for Indexing

- **Module 46: Transactions/1**
  - Transaction Concept
  - Transaction States
  - Concurrent Executions
- **Module 47: Transactions/2: Serializability**
  - Serializability
  - Conflict Serializability
- **Module 48: Transactions/3: Recoverability**
  - Recovery
  - Transaction Definition in SQL
  - View Serializability
  - Complex Notions of Serializability
- **Module 49: Concurrency Control/1**
  - Concurrency Control
  - Lock-Based Protocols
  - Implementation of Locking
- **Module 50: Concurrency Control/2**
  - Deadlock Handling
  - Timestamp-Based Protocols

- **Module 51: Backup and Recovery/1: Backup/1**
  - What is Backup and Recovery?
  - Why Backup?
  - Backup Data: Types
  - Backup Strategies
  - Case: Monthly Schedule
  - Hot Backup
- **Module 52: Backup and Recovery/2: Recovery/1**
  - Failure Classification
  - Storage Structure
  - Log-Based Recovery
- **Module 53: Backup and Recovery/3: Recovery/2**
  - Transactional Logging
  - Recovery Algorithm
- **Module 54: Backup and Recovery/4: Recovery/3**
  - Recovery with Early Lock Release
  - Plan for Backup and Recovery
- **Module 55: Backup and Recovery/5: Backup/2: RAID**
  - RAID: Redundant Array of Independent Disks

Week 12: Query Optimization, Performance and Architecture, NOSQL, Widely used RDBMSs, and Course Summarization

- **Module 56: Query Processing and Optimization/1: Processing**
  - Query Processing
  - Query Cost
  - Selection Operation
  - Sorting
  - Join Operation
  - Other Operations
- **Module 57: Query Processing and Optimization/2: Optimization**
  - Introduction to Query Optimization
  - Transformation of Relational Expressions
- **Module 58: RDBMS Performance and Architecture**
  - RDBMS Performance and Scalability
  - RDBMS Architecture
  - Scaling Databases
- **Module 59: Non-Relational DBMS: NOSQL**
  - What is Big Data?
  - What is NOSQL?
  - CAP Theorem
  - Types of NOSQL Databases
  - Relational vs. Non-Relational
- **Module 60: Widely used DBMSs and Summarization**
  - Widely used RDBMSs
  - Course Recap

- Read the DBMS Text book thoroughly and solve exercises
- Practice query coding
- Practice database design from specs
- Besides DBMS, develop good knowledge in programming, data structure, algorithms and discrete structures
- Seek help, if you need to – mail us
- To learn more online you may refer to the resources mentioned in: *What is the best possible way to learn DBMS online ?*

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.
Edited and new slides are marked with "PPD".