# Database Management System

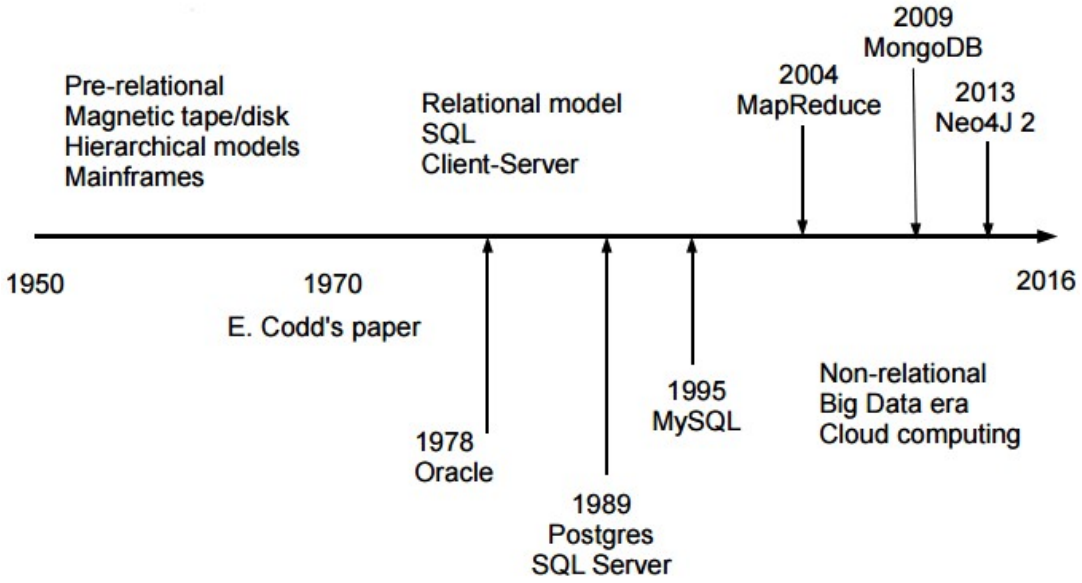
Dr. Tran Anh Tuan,

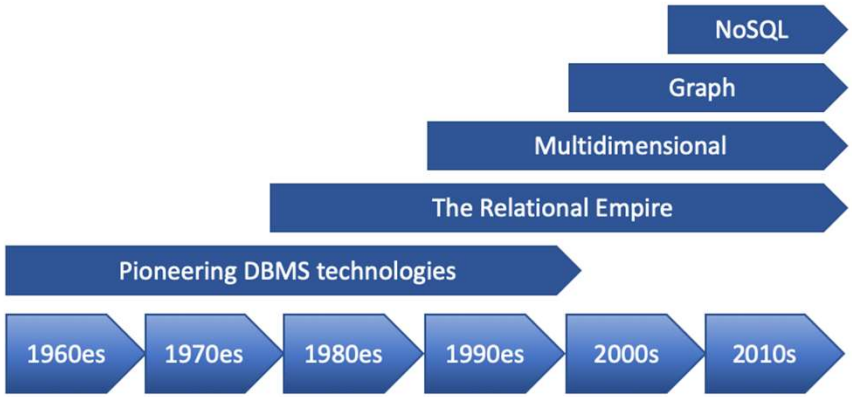
Faculty of Mathematics and Computer Science,

University of Science, HCMC

The database evolution happened in five “waves”:

- The first wave consisted of network, hierarchical, inverted list, and (in the 1990’s) object-oriented DBMSs; it took place from roughly 1960 to 1999.
- The relational wave introduced all of the SQL products (and a few non-SQL) around 1990 and began to lose users around 2008.
- The decision support wave introduced Online Analytical Processing (OLAP) and specialized DBMSs around 1990, and is still in full force today.
- The graph wave began with The Semantic Web stack from the Worldwide Web Consortium in 1999, with property graphs appearing around 2008
- The NoSQL wave includes big data and much more; it began in 2008.



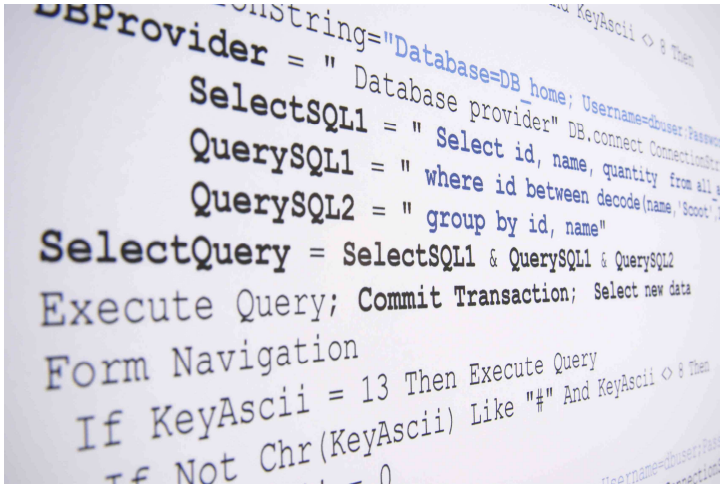



# Syllabus

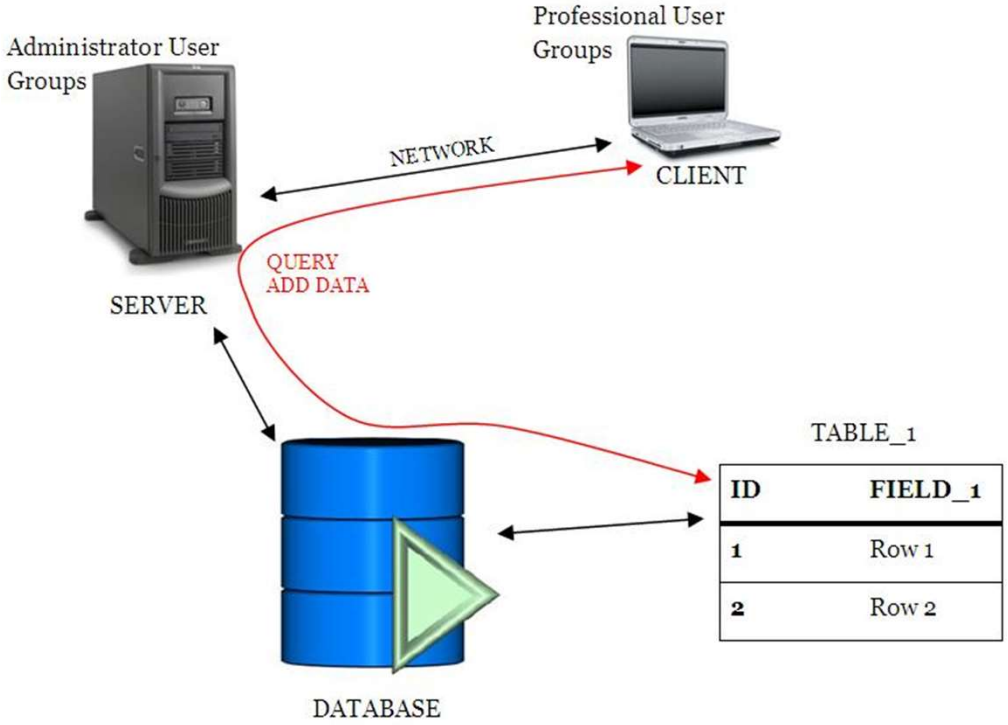
- Lecture 1 : Basic SQL Commands
- Lecture 2 : Advanced SQL Commands
- Lecture 3 : Stored Procedures and Functions
- Lecture 4 : Trigger, Transaction, Cursor and Temporary Table
- Lecture 5 : Security, Role, Server Backup, and Server Recovery
- Lecture 6 : Data Synchronization, SQL Profiler, Linked Server, Job Schedule
- Lecture 7 : Basic MongoDB Commands
- Lecture 8 : Advanced MongoDB Commands
- Lecture 9 : SQL Server and MongoDB connection from Application
- Lecture 10 : Final Project

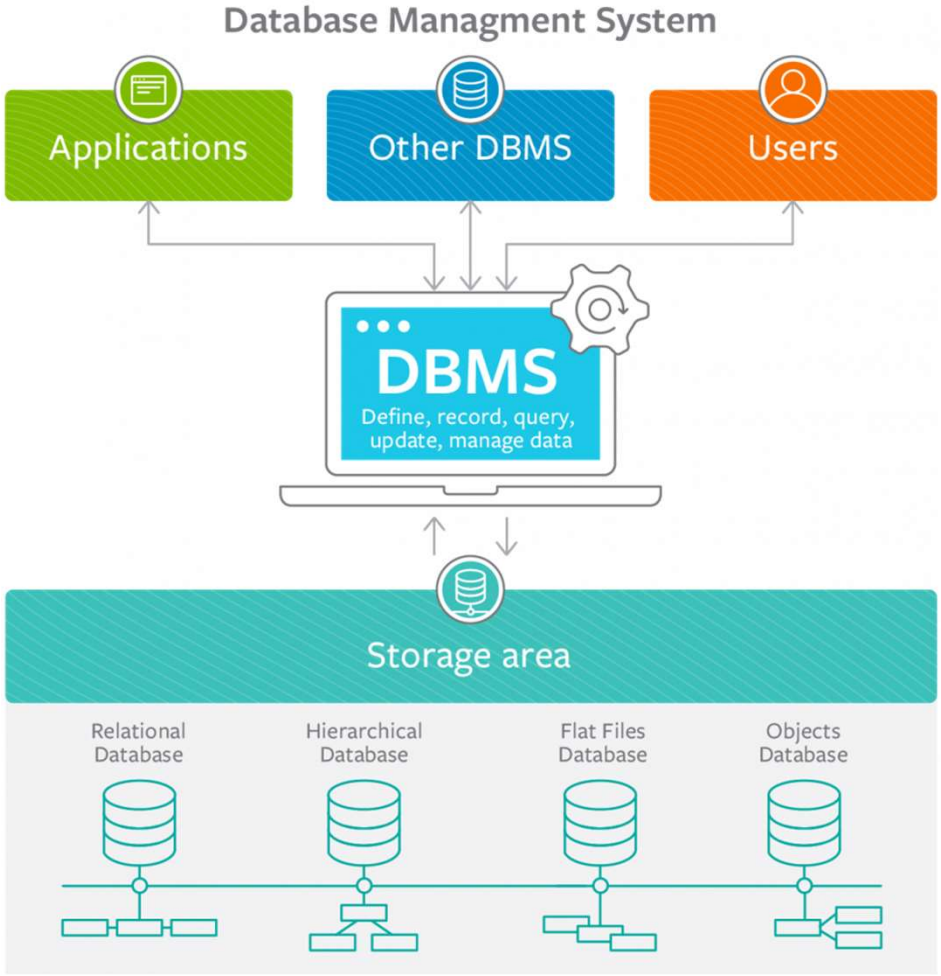
# Introduction

- The term **'database'** is defined as any collection of electronic records that can be processed to produce useful information. The data can be accessed, modified, managed, controlled and organized to perform various data-processing operations.
- The data is typically indexed across rows, columns and tables that make workload processing and data querying efficient. Different types of databases include: object-oriented, relational, distributed, hierarchical, network, and others.



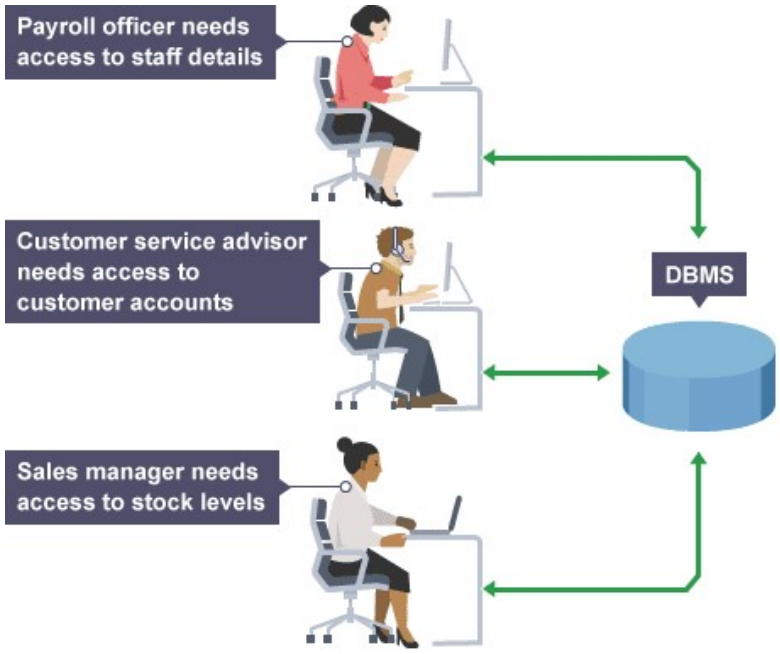

# Introduction

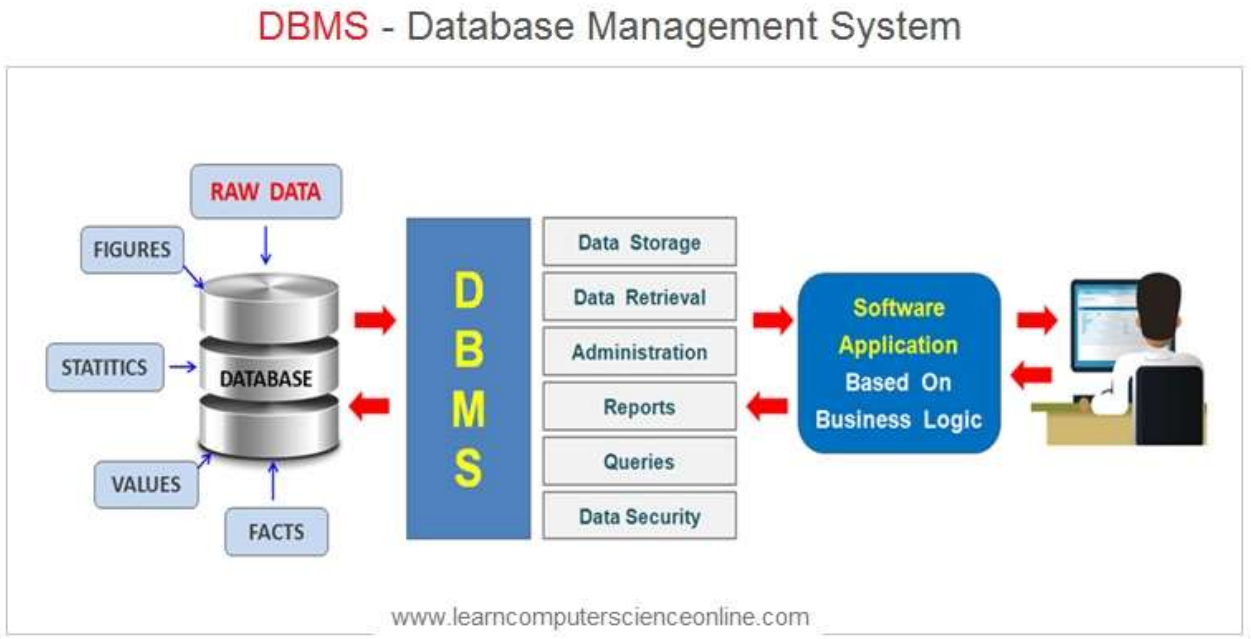



In enterprise applications, databases involve mission-critical, security-sensitive, and compliance-focused record items that have complicated logical relationships with other datasets and grow exponentially over time as the userbase increases.

# Introduction



# Introduction

- In order to facilitate these functions, DBMS has the following key components:
  - **Software.** DBMS is primarily a software system that can be considered as a management console or an interface to interact with and manage databases. The interfacing also spreads across real-world physical systems that contribute data to the backend databases. The OS, networking software, and the hardware infrastructure is involved in creating, accessing, managing, and processing the databases.



1. SolarWinds Database Performance Analyzer
2. Oracle RDBMS
3. IBM DB2
4. Altibase
5. Microsoft SQL Server
6. SAP Sybase ASE
7. Teradata
8. ADABAS
9. MySQL
10. FileMaker
11. Microsoft Access
12. Informix
13. SQLite
14. PostgresSQL
15. AmazonRDS
16. MongoDB
17. Redis
18. CouchDB
19. Neo4j
20. OrientDB
21. Couchbase
22. Toad
23. phpMyAdmin
24. SQL Developer
25. Seqel PRO
26. Robomongo
27. DbVisualizer
28. Hadoop HDFS
29. Cloudera
30. MariaDB
31. Informix Dynamic Server
32. 4D (4th Dimension)

# Introduction

**Software**

## #5) Microsoft SQL Server

Microsoft

Developed in the year 1989. The latest updated version came in 2016. The language used is Assembly C, Linux, C++ for writing it.

Works on Linux and Windows operating systems.

**Few features of MS SQL server include:**

Compatible with Oracle provides efficient management of workload and allows multiple users to use the same database.

**Costs:** It's a commercial tool.

**Website:** Microsoft SQL server

# Introduction

**Software**

#9) MySQL

MySQL

Latest version 8. The language used is C and C++.

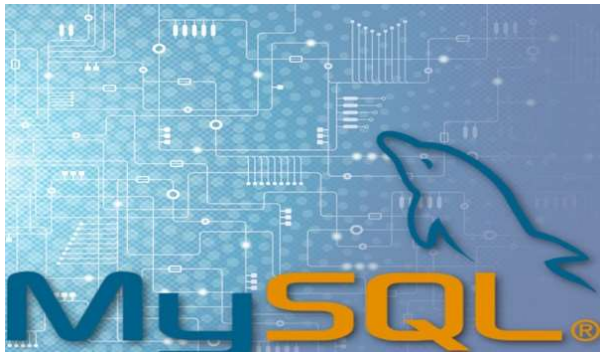
Works on Linux and Windows.

**Few features of this tool are:**

High-speed data processing, use of triggers increases productivity, with rollback and commit helps in data recovery if required.

**Costs:** It's a commercial tool.

**Website:** MySQL



# Introduction

**Software**

## #13) SQLite


It is used as a database system for mobiles. It is coded in C language.

It can work on Linux, Windows, and Mac operating systems.

**Few features of this tool are:**

It does not need much space hence, it can be used for storing small to medium size websites. It is fast and does not need to set up.

**Costs:** It's an open-source tool.

**Website:** SQLite


# Introduction

**Software**

## #16) MongoDB

mongoDB

**Few features of MongoDB are:**

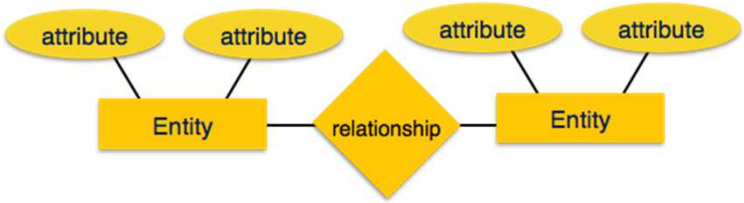
It can process a large amount of data simultaneously and uses internal memory so the data is easily accessible, the use of very complex joins is not supported, scaling is easily possible. Queries can be easily optimized for output.

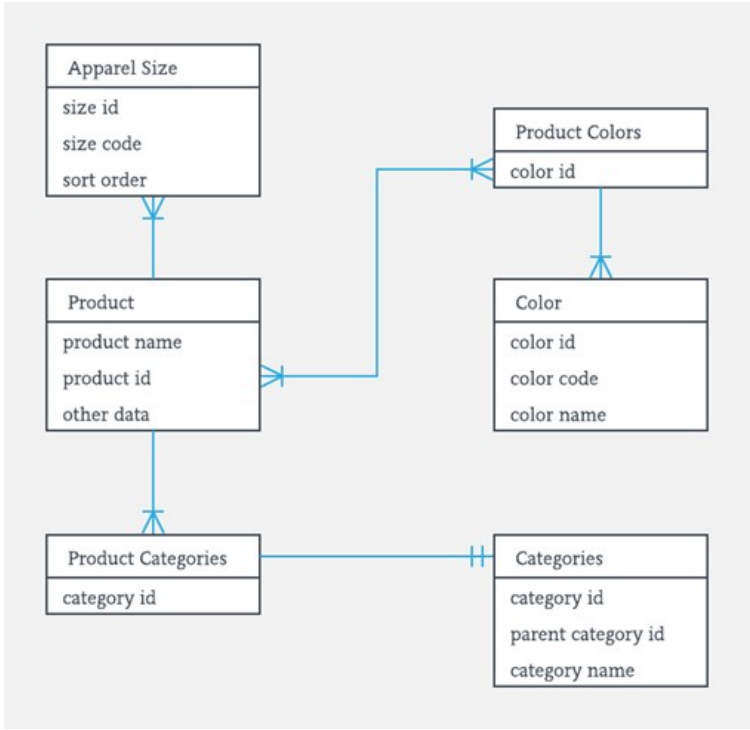
**Costs:** It's an open-source tool

**Website:** Mongo DB

# Introduction

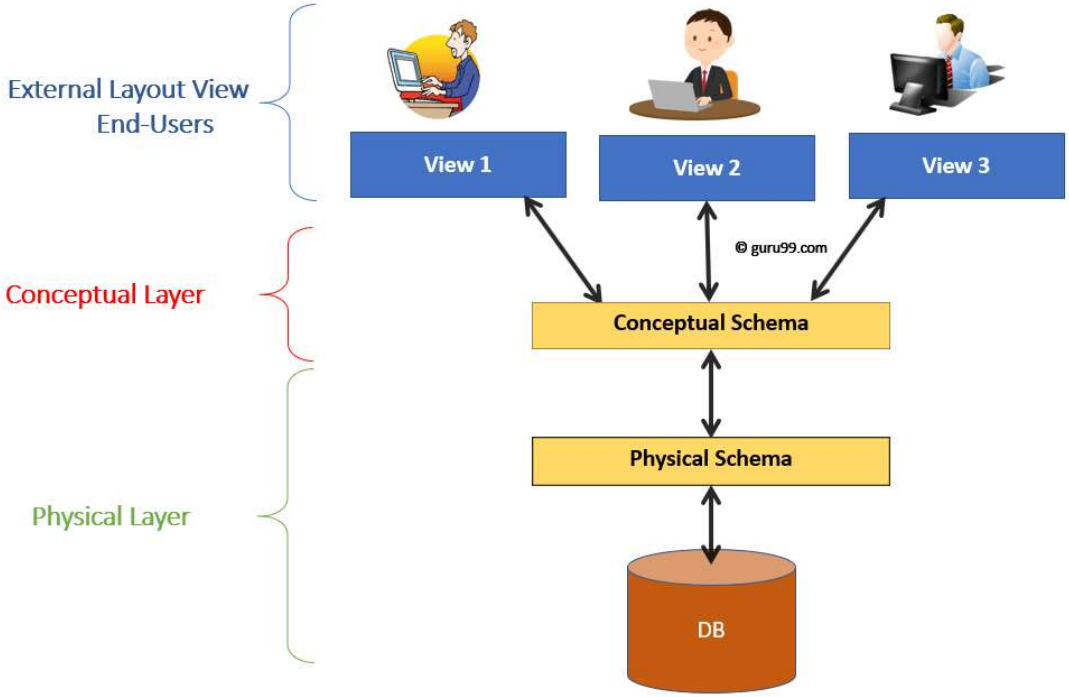
- In order to facilitate these functions, DBMS has the following key components:
  - **Data.** DBMS contains operational data, access to database records and metadata as a resource to perform the necessary functionality. The data may include files with such as index files, administrative information, and data dictionaries used to represent data flows, ownership, structure, and relationships to other records or objects.



# Introduction

**Data**

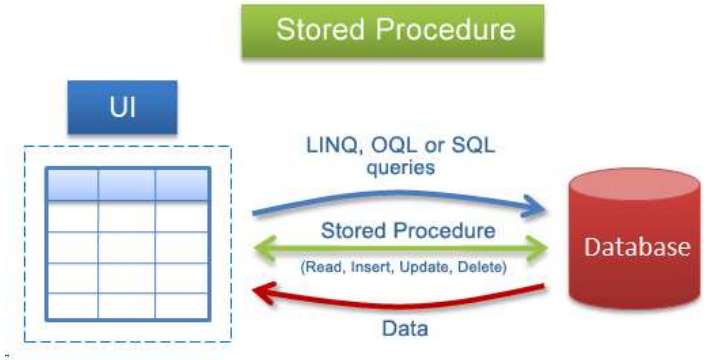


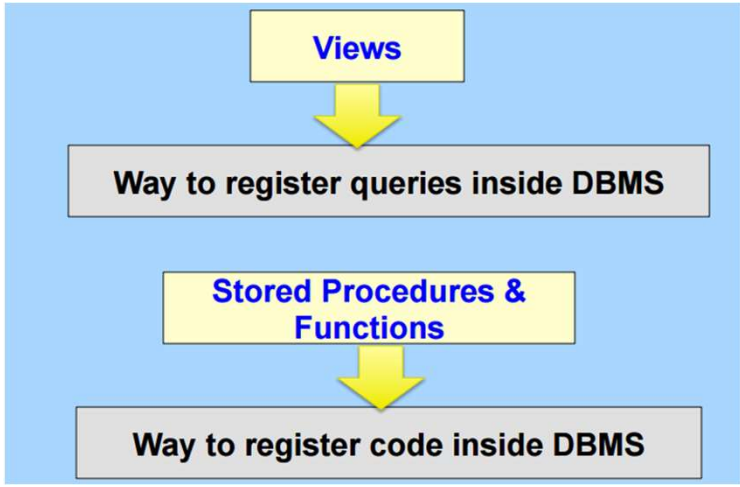

Consider an Example of a University Database. At the different levels this is how the implementation will look like:

| Type of Schema | Implementation |
|---|---|
| External Schema | **View 1**: Course info(cid:int,cname:string)<br>**View 2**: studeninfo(id:int. name:string) |
| Conceptual Shema | `Students(id: int, name: string, login: string, age: integer)`<br>`Courses(id: int, cname.string, credits:integer)`<br>`Enrolled(id: int, grade:string)` |
| Physical Schema | • Relations stored as unordered files.<br>• Index on the first column of Students. |

# Introduction

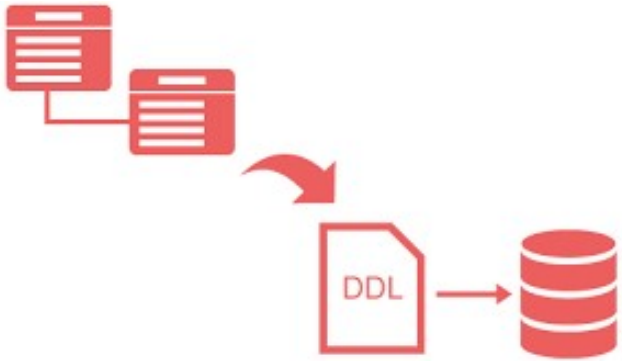
- In order to facilitate these functions, DBMS has the following key components:
  - **Procedures.** While not a part of the DBMS software, procedures can be considered as instructions on using DBMS. The documented guidelines assist users in designing, modifying, managing, and processing databases.



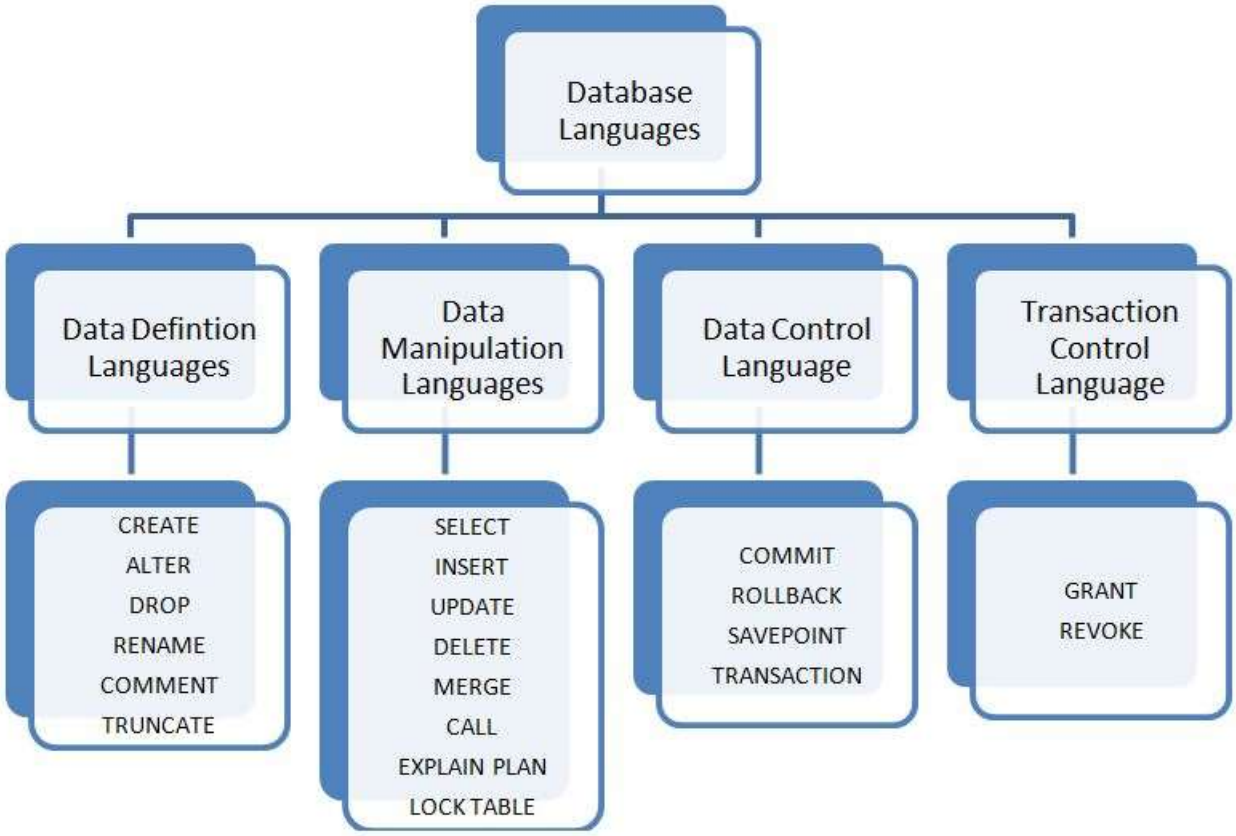
# Introduction

- In order to facilitate these functions, DBMS has the following key components:
  - **Database languages.** These are components of the DBMS used to access, modify, store, and retrieve data items from databases; specify database schema; control user access; and perform other associated database management operations. Types of DBMS languages include Data Definition Language (DDL), Data Manipulation Language (DML), Database Access Language (DAL) and Data Control Language (DCL).



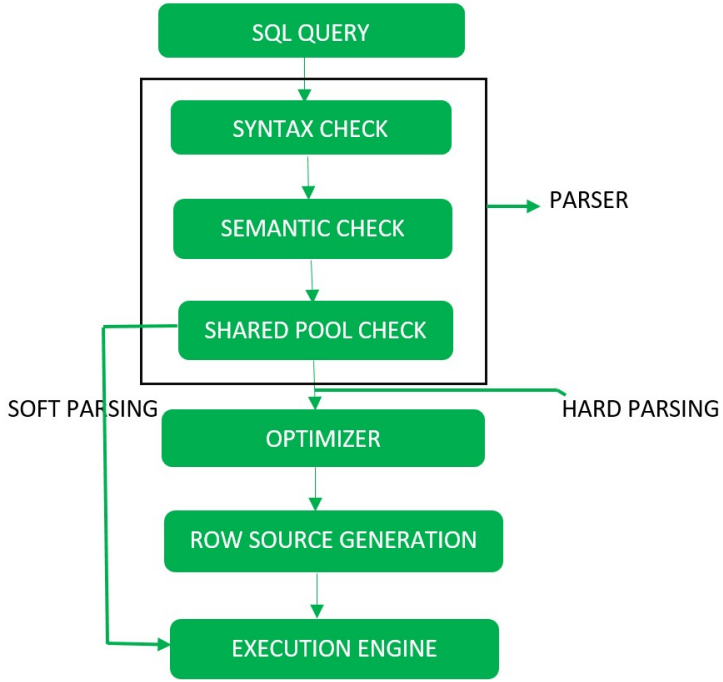

# Introduction

**Database languages**



# Introduction

- In order to facilitate these functions, DBMS has the following key components:
  - **Query processor.** As a fundamental component of the DBMS, the query processor acts as an intermediary between users and the DBMS data engine in order to communicate query requests. When users enter an instruction in SQL language, the command is executed from the high-level language instruction to a low-level language that the underlying machine can understand and process to perform the appropriate DBMS functionality. In addition to instruction parsing and translation, the query processor also optimizes queries to ensure fast processing and accurate results.




# Introduction

**Query processor**

- **Step-1:**

  **Parser:** During parse call, the database performs the following checks- Syntax check, Semantic check and Shared pool check, after converting the query into relational algebra.

  Parser performs the following checks as (refer detailed diagram):

  1. **Syntax check –** concludes SQL syntactic validity. Example:

     ```
     SELECT * FORM employee
     ```

     Here error of wrong spelling of FROM is given by this check.

  2. **Semantic check –** determines whether the statement is meaningful or not. Example: query contains a tablename which does not exist is checked by this check.
  3. **Shared Pool check –** Every query possess a hash code during its execution. So, this check determines existence of written hash code in shared pool if code exists in shared pool then database will not take additional steps for optimization and execution.

  **Hard Parse and Soft Parse –**

  If there is a fresh query and its hash code does not exist in shared pool then that query has to pass through from the additional steps known as hard parsing otherwise if hash code exists then query does not passes through additional steps. It just passes directly to execution engine (refer detailed diagram). This is known as soft parsing.

  Hard Parse includes following steps – Optimizer and Row source generation.

# Introduction

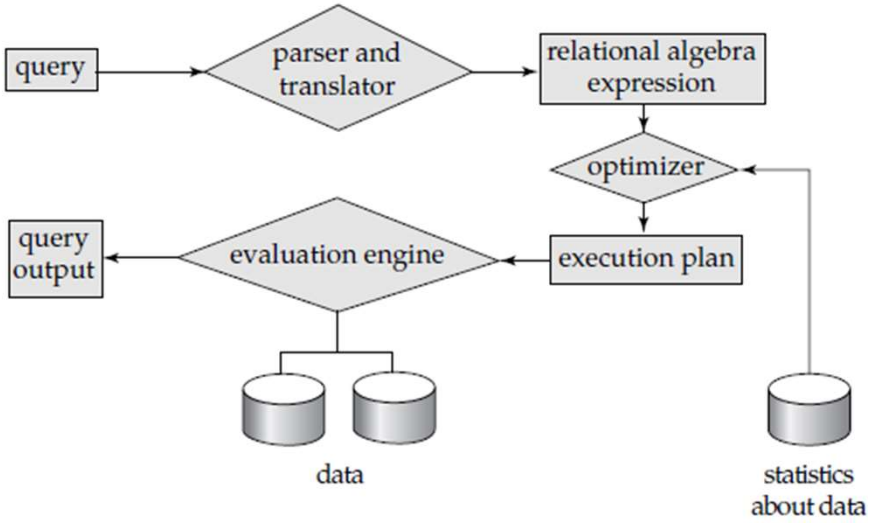
**Query processor**

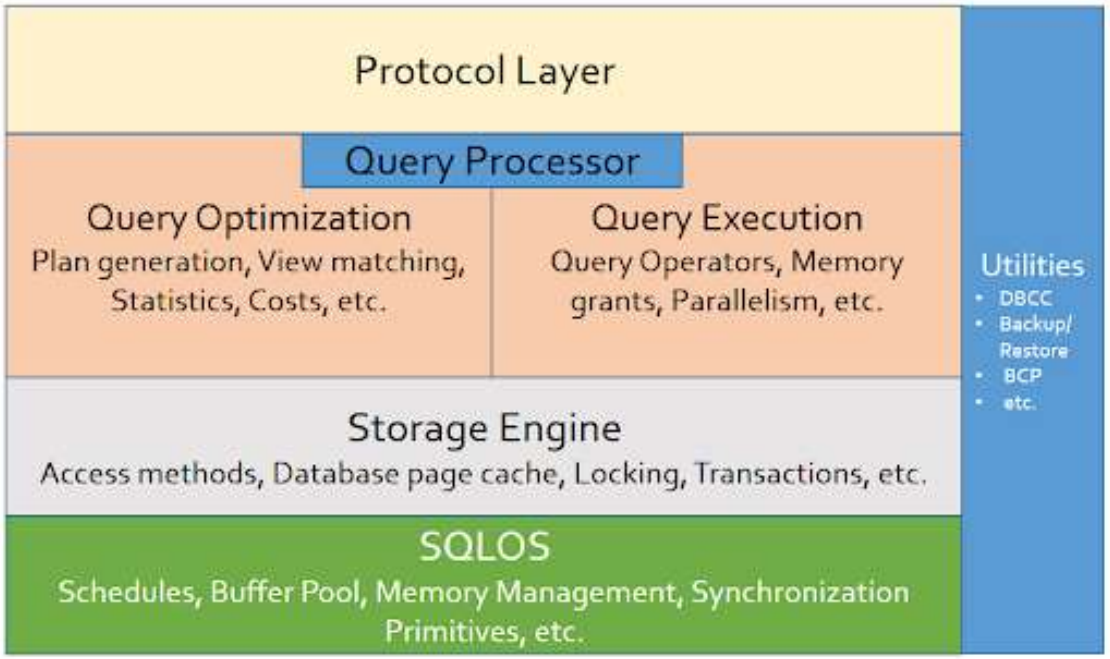
- **Step-2:**
  **Optimizer:** During optimization stage, database must perform a hard parse atleast for one unique DML statement and perform optimization during this parse. This database never optimizes DDL unless it includes a DML component such as subquery that require optimization.
  It is a process in which multiple query execution plan for satisfying a query are examined and most efficient query plan is satisfied for execution.
  Database catalog stores the execution plans and then optimizer passes the lowest cost plan for execution.

  **Row Source Generation –**
  The Row Source Generation is a software that receives a optimal execution plan from the optimizer and produces an iterative execution plan that is usable by the rest of the database. the iterative plan is the binary program that when executes by the sql engine produces the result set.
- **Step-3:**
  **Execution Engine:** Finally runs the query and display the required result.

# Introduction

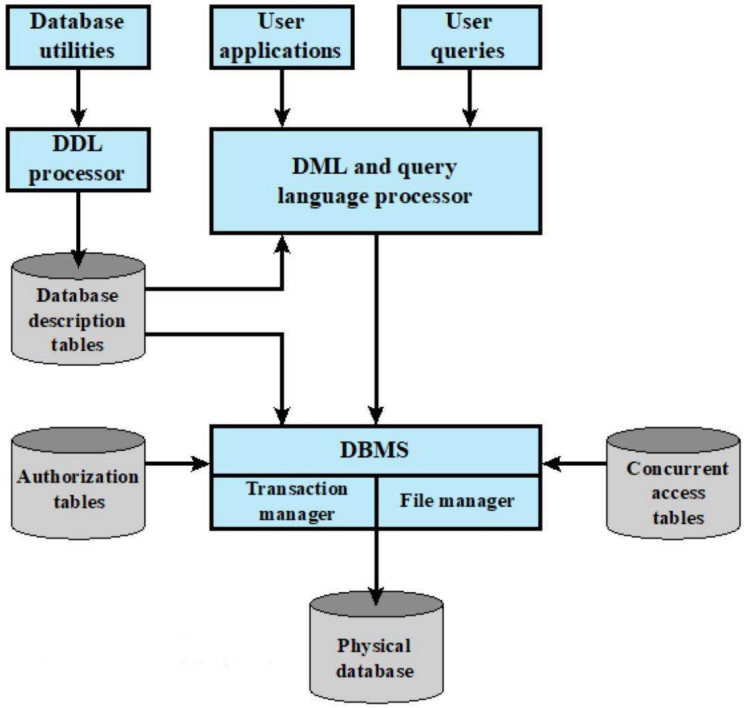
**Query processor**



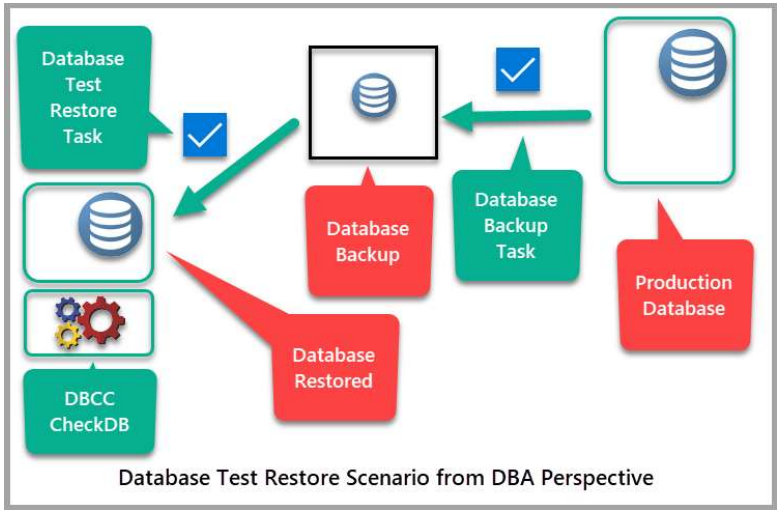
# Introduction

- In order to facilitate these functions, DBMS has the following key components:
  - **Runtime database manager.** A centralized management component of DBMS that handles functionality associated with runtime data, which is commonly used for context-based database access. This component checks for user authorization to request the query; processes the approved queries; devises an optimal strategy for query execution; supports concurrency so that multiple users can simultaneously work on same databases; and ensures integrity of data recorded into the databases.



# Introduction

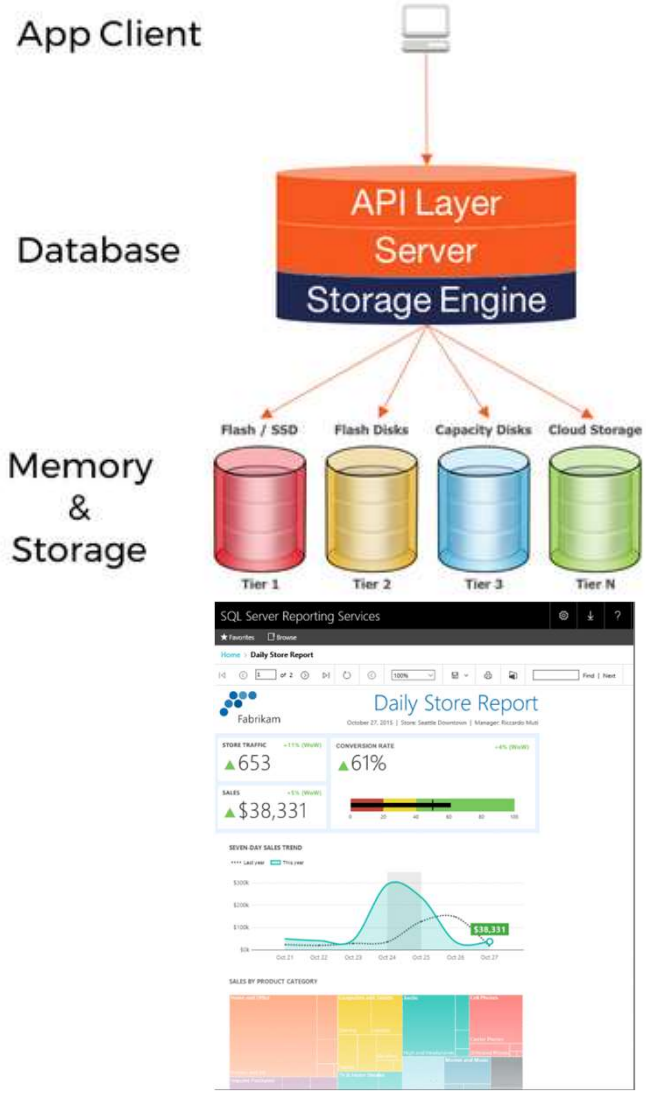
- In order to facilitate these functions, DBMS has the following key components:
  - **Database manager.** Unlike the runtime database manager that handles queries and data at runtime, the database manager performs DBMS functionality associated with the data within databases. Database manager allows a set of commands to perform different DBMS operations that include creating, deleting, backup, restoring, cloning, and other database maintenance tasks. The database manager may also be used to update the database with patches from vendors.



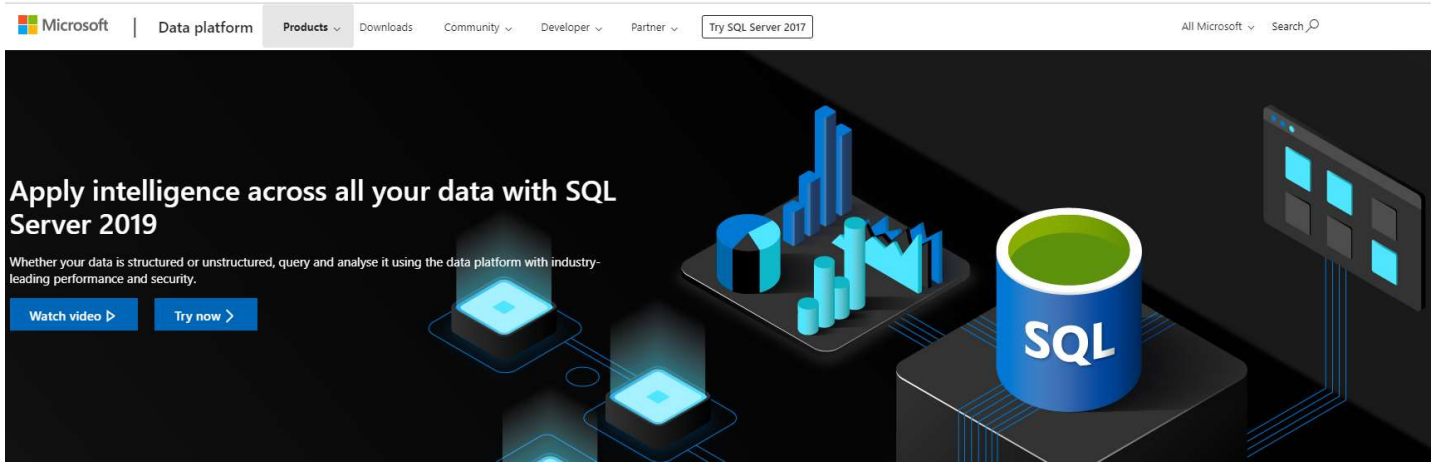
Database Test Restore Scenario from DBA Perspective

# Introduction

- In order to facilitate these functions, DBMS has the following key components:
  - **Database engine.** This is the core software component within the DBMS solution that performs the core functions associated with data storage and retrieval. A database engine is also accessible via APIs that allow users or apps to create, read, write, and delete records in databases.
  - **Reporting.** The report generator extracts useful information from DBMS files and displays it in structured format based on defined specifications. This information may be used for further analysis, decision making, or business intelligence.



# SQL Server




## What you'll love about SQL Server 2019

### Analyse every type of data

Gain insights from all your data by querying across relational, non-relational, structured and unstructured data, for a complete picture of your business using SQL Server 2019 with Apache Spark built in.

### Choose your language and platform

Get the flexibility to use the language and platform of your choice with open source support. Run SQL Server on Linux containers with Kubernetes support or on Windows.

### Rely on industry-leading performance

Take advantage of breakthrough scalability and performance to improve the stability and response time of your database – without making app changes. Get high availability for mission-critical applications, data warehouses and data lakes.

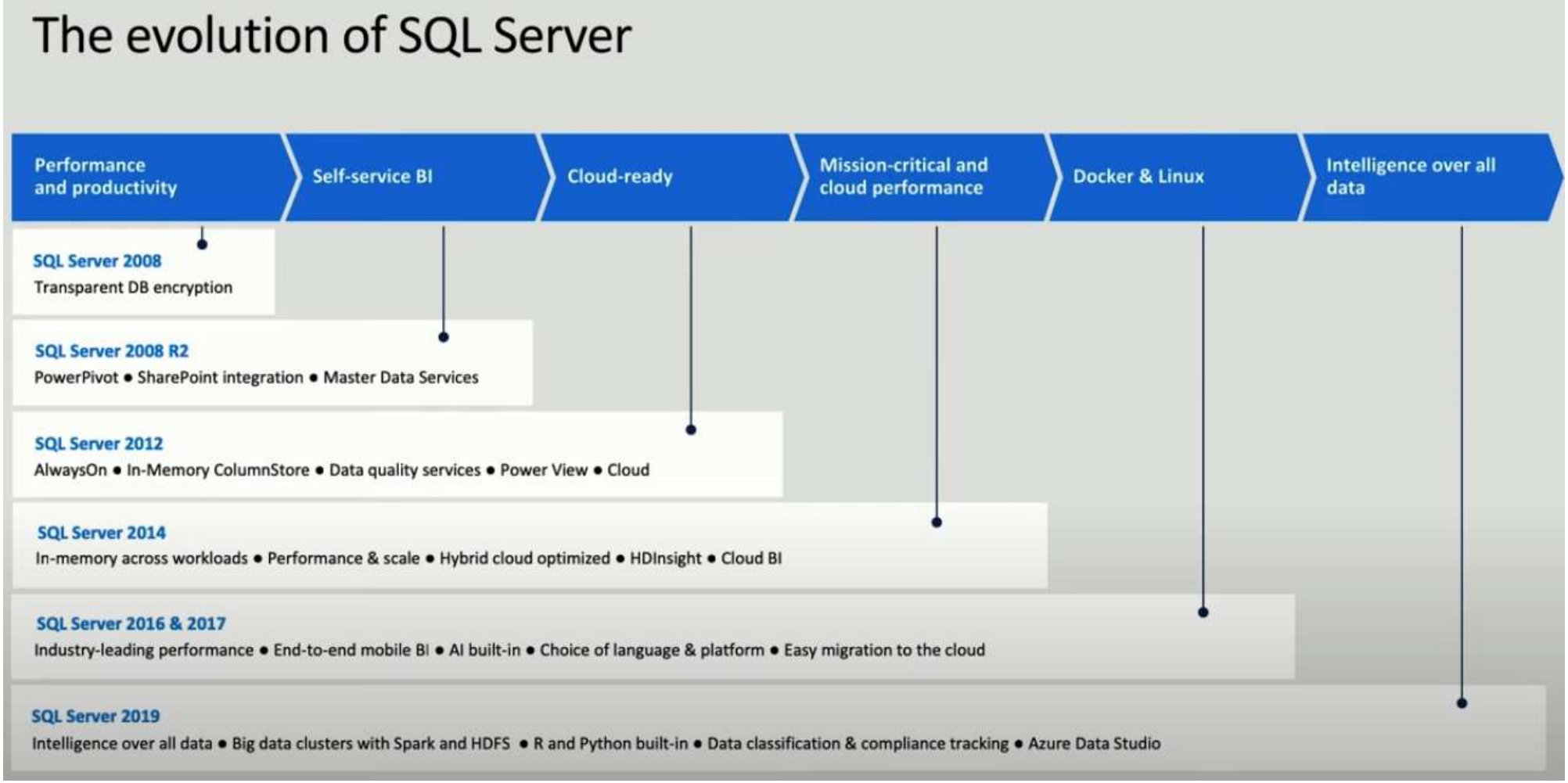
### Trust nine years of proven security leadership

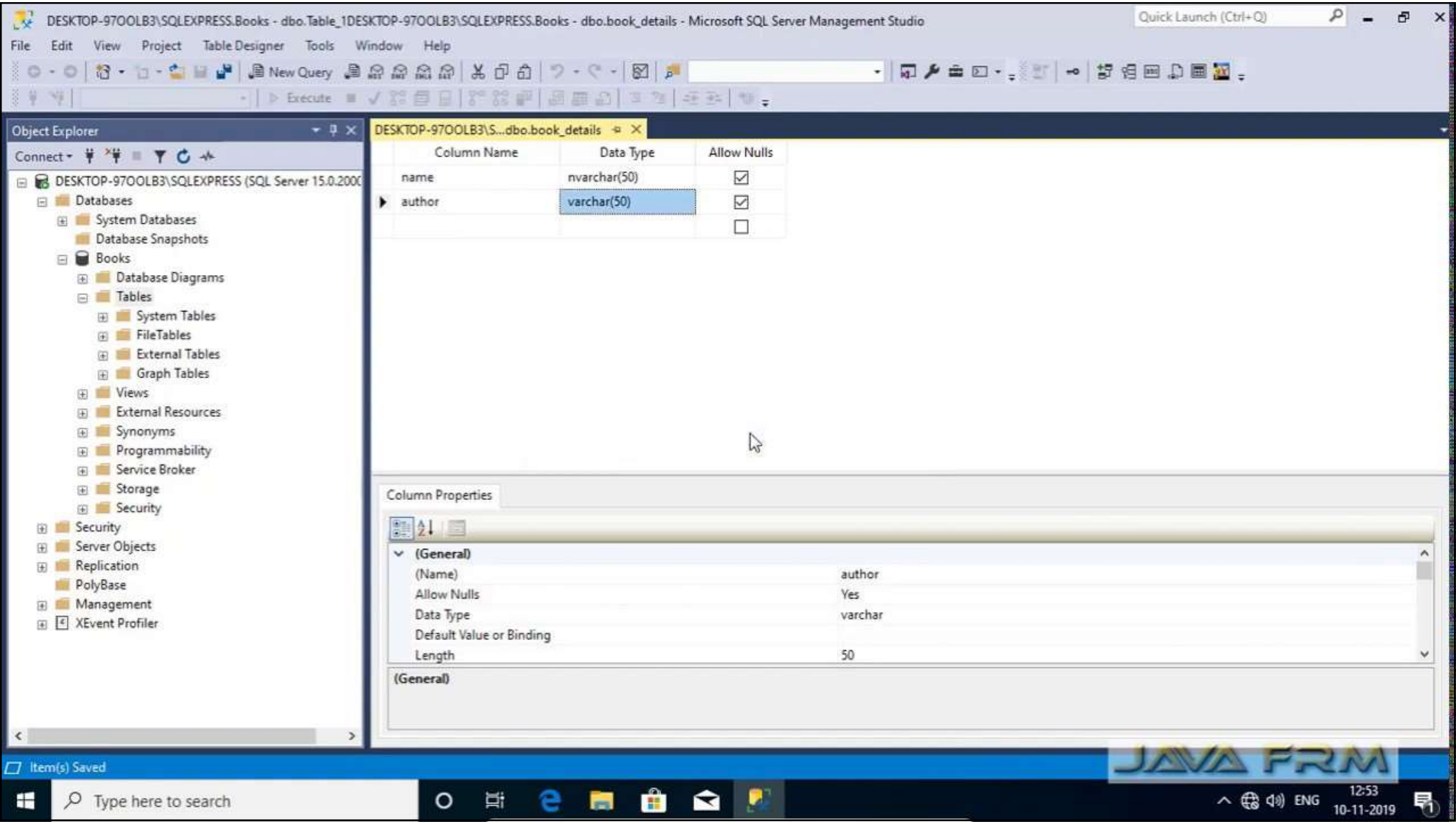
Achieve your security and compliance goals using the database rated as least vulnerable over the last nine years. Stay a step ahead using built-in features for data classification, data protection and monitoring and alerts [1].

### Make faster, better decisions

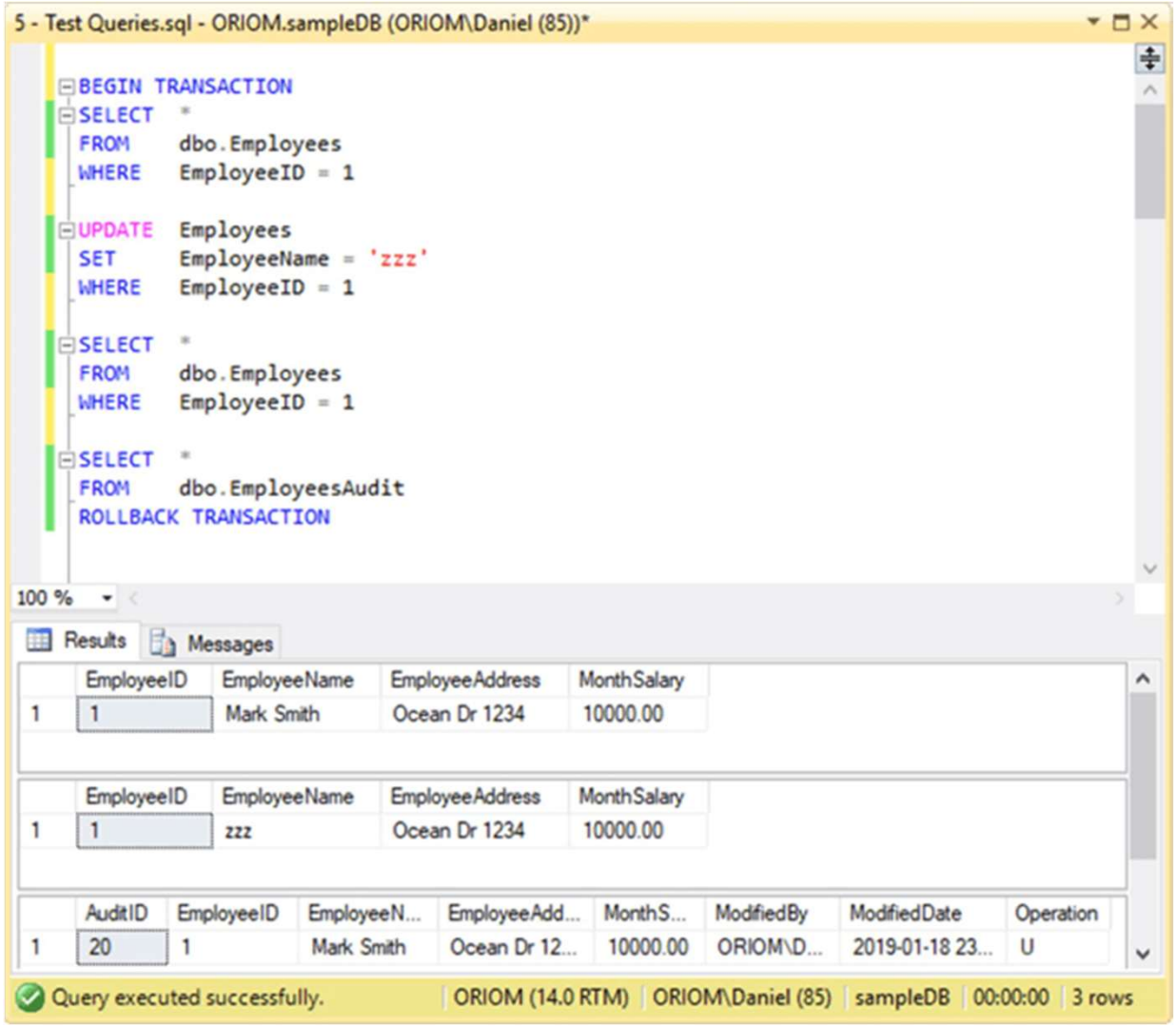
Turn data into answers using the enterprise reporting capabilities of SQL Server Reporting Services along with the included Power BI Report Server, which gives your users access to rich, interactive Power BI reports on any device.

# The evolution of SQL Server




DESKTOP-97OOLB3\SQLEXPRESS.Books - dbo.Table_1DESKTOP-97OOLB3\SQLEXPRESS.Books - dbo.book_details - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Project Table Designer Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-97OOLB3\SQLEXPRESS (SQL Server 15.0.200
Databases
System Databases
Database Snapshots
Books
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
DESKTOP-97OOLB3\S...dbo.book_details
Column Name
Data Type
Allow Nulls
name
nvarchar(50)
author
varchar(50)
Column Properties
(General)
(Name)
author
Allow Nulls
Yes
Data Type
varchar
Default Value or Binding
Length
50
(General)
Item(s) Saved
Type here to search
ENG
12:53
10-11-2019

# SQL Server Example



# SQL Server Example

```
1  BEGIN TRY
2      SELECT CAST('SQL' AS INT) [INT]
3  END TRY
4  BEGIN CATCH
5      SELECT ERROR_NUMBER() AS ErrNum
6           , ERROR_SEVERITY() AS ErrSeverity
7           , ERROR_STATE() AS ErrState
8           , ERROR_PROCEDURE() AS ErrProcedure
9           , ERROR_LINE() AS ErrLine
10          , ERROR_MESSAGE() AS ErrMsg;
11 END CATCH;
12 GO
```

82 %

Results | Messages

| INT |
|---|

| | ErrNum | ErrSeverity | ErrState | ErrProcedure | ErrLine | ErrMsg |
|---|---|---|---|---|---|---|
| 1 | 245 | 16 | 1 | NULL | 2 | Conversion failed when converting the varchar value 'SQL' to data type int. |

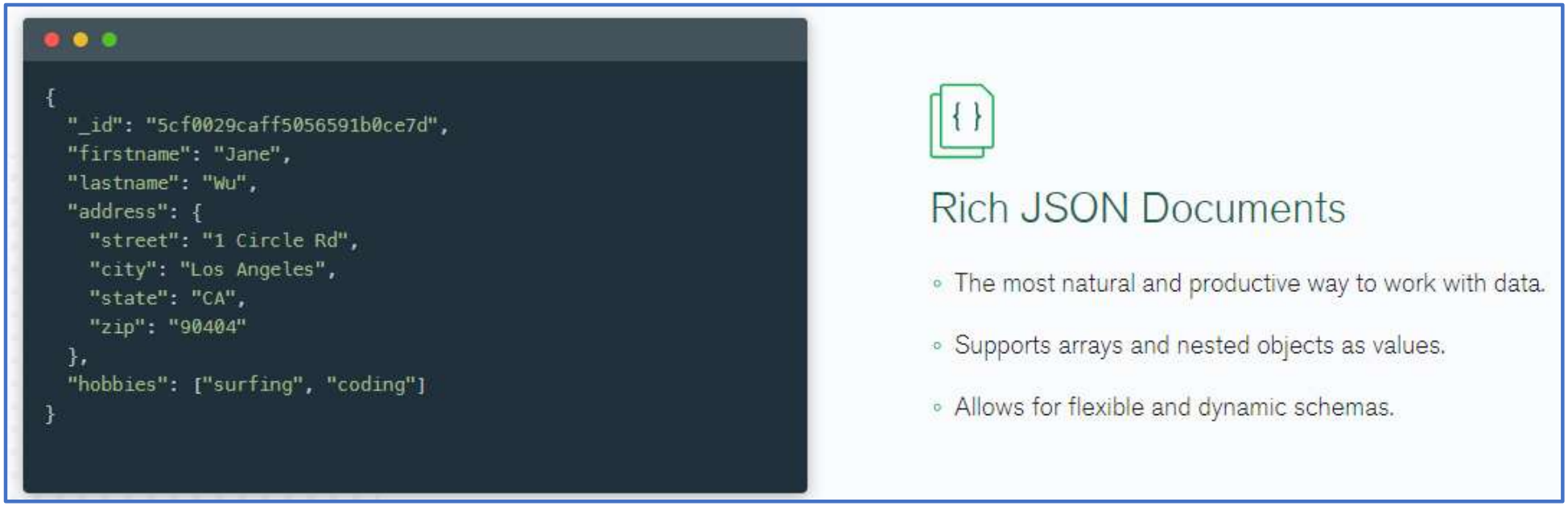
# MongoDB

Used by millions of developers to power the world's most innovative products and services


## As a programmer, you think in objects. Now your database does too.

MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model.

# MongoDB


{
"_id": "5cf0029caff5056591b0ce7d",
"firstname": "Jane",
"lastname": "Wu",
"address": {
"street": "1 Circle Rd",
"city": "Los Angeles",
"state": "CA",
"zip": "90404"
},
"hobbies": ["surfing", "coding"]
}
Rich JSON Documents
The most natural and productive way to work with data.
Supports arrays and nested objects as values.
Allows for flexible and dynamic schemas.

# MongoDB

## Powerful query language

- Rich and expressive query language that allows you to filter and sort by any field, no matter how nested it may be within a document.
- Support for aggregations and other modern use-cases such as geo-based search, graph search, and text search.
- Queries are themselves JSON, and thus easily composable. No more concatenating strings to dynamically generate SQL queries.

```
> db.users.find({ "address.zip" : "90404" })
{ "_id": "5cf0029caff5056591b0ce7d", "firstname": "Jane", "lastname"
{ "_id": "507f1f77bcf86cd799439011", "firstname": "Jon", "lastname":
{ "_id": "5349b4ddd2781d08c09890f3", "firstname": "Jim", "lastname":
{ "_id": "5bf142459b72e12b2b1b2cd", "firstname": "Jeff", "lastname":
{ "_id": "5cf003283b23d04a40d5f88a", "firstname": "Jerry", "lastname
{ "_id": "5bf142459b72e12b2b1b2cd", "firstname": "Jai", "lastname":
{ "_id": "5cf0036deaa1742dd225ea35", "firstname": "Jess", "lastname"
{ "_id": "54495ad94c934721ede76d90", "firstname": "Jill", "lastname"
{ "_id": "566eb3c704c7b31facbb0007", "firstname": "Janet", "lastname
{ "_id": "5a999cc461d36489a27f2563", "firstname": "Jan", "lastname":
```

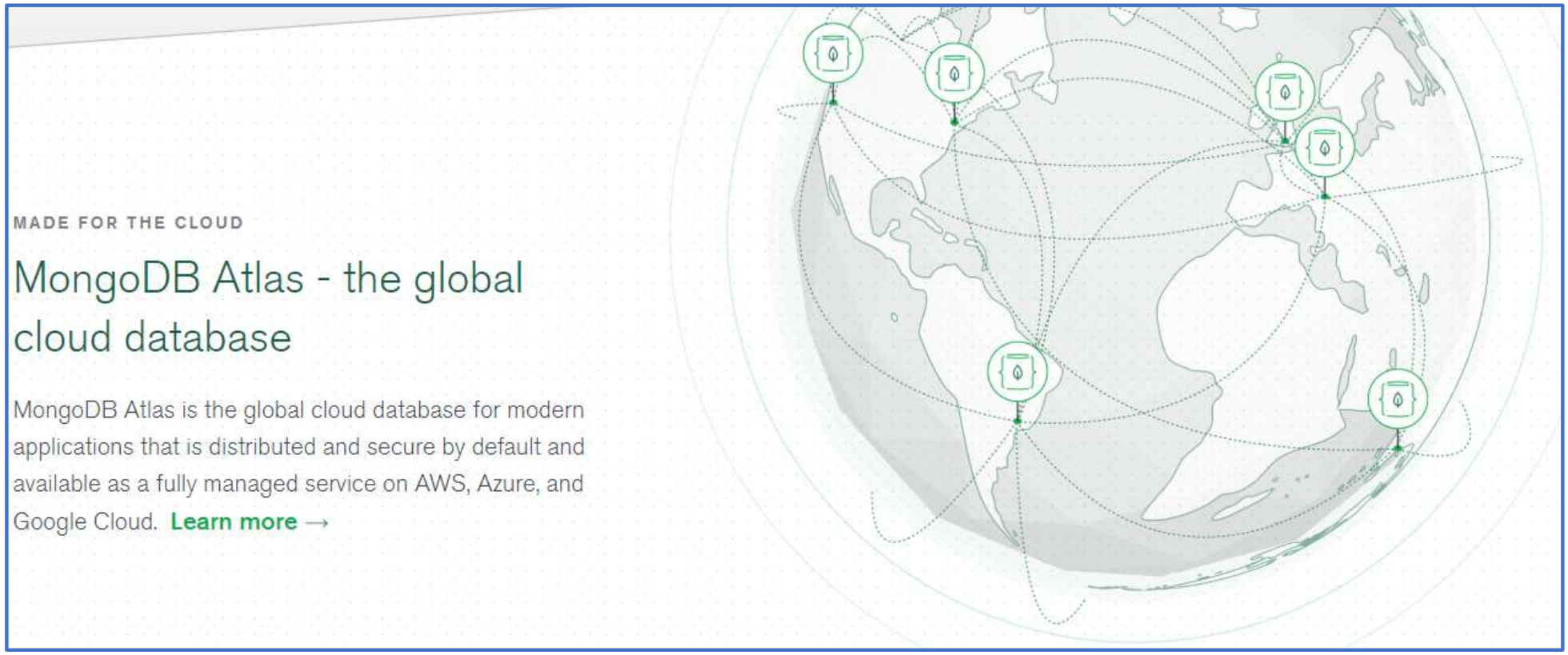
# MongoDB

```
session.start_transaction()
order = { line_items : [ { item : 5, quantity: 6 } ] }
db.orders.insertOne( order, session=session );
for x in order.line_items:
  db.inventory.update(
    { _id  : x.item } ,
    { $inc : { number : -1 * x.quantity } },
    session=session
  )
session.commit_transaction()
```

## All the power of a relational database, and more...

- Full ACID transactions.
- Support for joins in queries.
- Two types of relationships instead of one: reference and embedded.

# MongoDB


MADE FOR THE CLOUD
MongoDB Atlas - the global cloud database
MongoDB Atlas is the global cloud database for modern applications that is distributed and secure by default and available as a fully managed service on AWS, Azure, and Google Cloud. Learn more →

# MongoDB Example


```
{
  "address": {
     "building": "1007",
     "coord": [ -73.856077, 40.848447 ],
     "street": "Morris Park Ave",
     "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
     { "date": { "$date": 1393804800000 }, "grade": "A",
     { "date": { "$date": 1378857600000 }, "grade": "A",
     { "date": { "$date": 1358985600000 }, "grade": "A",
     { "date": { "$date": 1322006400000 }, "grade": "A",
     { "date": { "$date": 1299715200000 }, "grade": "B",
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```


Find | Aggregate

Query Reference

Query: {borough: {$in: ["Brooklyn", "Queens"]}}

Fields: {_id:0, grades:0}

Sort: {}

Limit: 100

Skip: 0

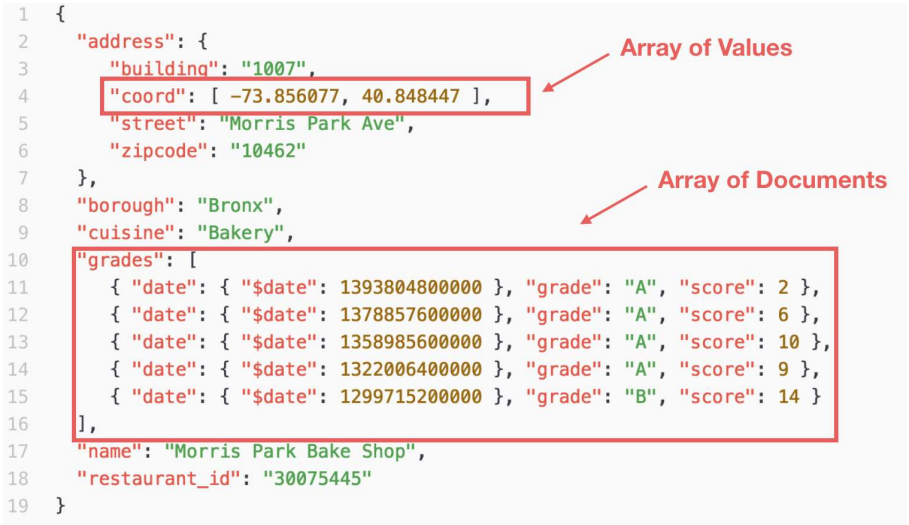
Flatten: true

▶ Get Data

**100** of **25,359** rows returned

| | borough character | cuisine character | name character | restaurant_id integer | ad cha |
|---|---|---|---|---|---|
| 1 | Queens | Jewish/Kosher | Tov Kosher Kitchen | 40356068 | 97- |
| 2 | Queens | American | Brunos On The Boulevard | 40356151 | 882 |
| 3 | Brooklyn | Hamburgers | Wendy'S | 30112340 | 469 |
| 4 | Brooklyn | American | Riviera Caterer | 40356018 | 278 |
| 5 | Brooklyn | Ice Cream, Gelato, Yogurt, Ices | Taste The Tropics Ice Cream | 40356731 | 183 |
| 6 | Brooklyn | American | Regina Caterers | 40356649 | 640 |
| 7 | Brooklyn | American | C & C Catering Service | 40357437 | 771 |
| 8 | Brooklyn | Delicatessen | Wilken'S Fine Food | 40356483 | 711 |

# MongoDB Example




```
{
  "address": {
     "building": "1007",
     "coord": [ -73.856077, 40.848447 ],
     "street": "Morris Park Ave",
     "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
     { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
     { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
     { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
     { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
     { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

Find | Aggregate

Query Reference

Query: {"grades.0.grade": "A"}

Fields: {_id:0, borough:1, name:1, grades:1}  Sort: {}

Limit: 100  Skip: 0  Flatten: true

▶ Get Data

**100** of **25,359** rows returned

| | borough<br>character | grades<br>list |
|---|---|---|
| 1 | Brooklyn | list(date = c(1419897600, 1404172800, 1367280000, 1336435200), grade = c("A", "B", "A", "A"), score = c(8, 23, 12, 12)) |
| 2 | Bronx | list(date = c(1393804800, 1378857600, 1358985600, 1322006400, 1299715200), grade = c("A", "A", "A", "A", "B"), score = |
| 3 | Staten Island | list(date = c(1412553600, 1400544000, 1365033600, 1327363200), grade = c("A", "A", "A", "A"), score = c(9, 12, 12, 9)) |
| 4 | Brooklyn | list(date = c(1402358400, 1370390400, 1334275200, 1318377600), grade = c("A", "A", "A", "A"), score = c(5, 7, 12, 12)) |
| 5 | Brooklyn | list(date = c(1405296000, 1373414400, 1341964800, 1329955200), grade = c("A", "A", "A", "A"), score = c(12, 8, 5, 8)) |
| 6 | Manhattan | list(date = c(1409961600, 1374451200, 1343692800, 1325116800), grade = c("A", "A", "A", "A"), score = c(2, 11, 12, 12)) |
| 7 | Brooklyn | list(date = c(1405641600, 1375142400, 1360713600, 1345075200, 1313539200), grade = c("A", "A", "A", "A", "A"), score = |
| 8 | Bronx | list(date = c(1401235200, 1371600000, 1339718400), grade = c("A", "A", "A"), score = c(11, 4, 3)) |

# Summary

## What is SQL Server?

SQL Server is a Microsoft relational database management system(RDBMS). The competitors are Oracle DB and MySQL. It supports a 32-bit and 64-bit environment. It also is known as MSSQL and Microsoft SQL Server. Some more details regarding SQL Server are given below:

- Its first version was released in 1989 by Microsoft.
- It supports XML data type support, dynamic management views and database mirroring.
- It supports e-commerce and data warehousing.
- It has several editions: Enterprise, Standard, Web, Business Intelligence, Express.

# Summary

## MongoDB

"MongoDB is Open-Source, cross-platform, NoSQL document database written in C++ that provides high performance, high availability and high scalability."

Let us discuss what does each term signifies in this definition

- **High Performance-**It means it provides faster read and writes scan.
- **High Availability**-Many replicated servers are used to provide high availability of data without delay
- **Scalability-**Automatic SHARDING distributes collection data across machines and eventually, consistent read scan can be distributed over replicated servers.

# JOB Description


**Database Administrator Duties & Responsibilities**

Database administrators (DBAs) have a variety of duties and tasks, such as the following:

- Responsibility for the evaluation of database software purchases
- Supervision of modifications to any existing database software to meet the needs of their employer
- Responsibility for maintaining the integrity and performance of company databases
- Must guarantee that data is stored securely and optimally
- Informing end users of changes in databases and train them to utilize systems
- Ensuring the security of company data

# THANK YOU