Database fundamentals: Understanding RDBMS

The purpose of RDBMS

Collect & store related data

Organize data into logical structures

Data can be stored as tables & relations

Fields can be used to store values

Easily query the stored data

Easily analyze stored data

Allows real-time online data processing (OLTP)

Data is consistently updated

Short & fast inserts & updates

Speed is fast

Data is highly normalized

Easily define schemas & sub-schemas

Easily establish relationships between DB elements

Provides the ability of independence of data from programs

Provides backup & recovery features

Popular RDBMS

Oracle

High performance

Spawn multiple database instances – instance caging

100% uptime – high availability

Excellent backup & recovery features with low recovery time

Highly secure

SQL Server

Insights from a data estate

Multiplatform support

High performance

Allows easy data classification, protection & monitoring

High availability

Cloud-based

Highly scalable

Hybridization – part of database could be in cloud

The software even tries to fix malicious activity if detected

MySQL

Open source although owned by Oracle

2 licensing models – community, enterprise version

ACID – Atomicity, Consistency, Isolation, Durability

Multimodal – can support structured (based on SQL) & semi-structured data (based on JSON)

Scalable & fast

memory based

built on client-server architecture

transactional

supports large databases

MySQL HeatWave – In-memory query accelerator – designed for fast execution of queries, analytical & transactional queries

PostgreSQL

Completely open-source

Very good reliability & active community

ACID

Triggers can be programmed

Stored procedures can be programmed

We can modify source code

Provides immediate consistency on a single server

Feature history of Oracle DBMS

Created by Larry Ellison

Application clusters, advanced queuing, data-mining, streams

Automated DB management, grid infrastructure & online indexing

Transparent data encryption

Active data guard, secure files, exit data, data reduction

Multi-tenant architecture, in-memory column store, support for JSON

Active directory integration

Online merging of partitions & sub-partitions

Automatic index creation, real-time statistics maintenance, SQL queries on object stores

In-memory for IoT data stream

Support for blockchain tables, self-managing in-memory features

Sharding advisor tools, property graph visualization, automatic zone maps

Oracle’s future

Customers are slowly shifting to cloud databases but Oracle made the jump a bit too late. It’s hard for companies on Oracle to jump. They are working on their 2nd generation of cloud services to catch up with the competition.

Features of SQL Server

Networking features

Replication

Query analyzer, OLAP – using which SQL could be extended

Stored procedures

OLAP replaced by Analysis services

XML support, user-defined fns, indexed views, log shipping, replication enhancements

SQL Server Management Studio

Business Intelligence Development Studio

Create customized audits & monitor data warehouse

Replaced BI tools with SQL Server Data tools

Enhanced clustering features

Ease of integration into Azure cloud

Memory optimized tables

PolyBase – query noSQL data i.e, JSON, CSV files etc. – made archiving a lot easier

Complete support for Linux, Python

MySQL Features

Fast & stable

Good optimization

Transactional capabilities

Supported prepared statements

Stored procedures, server-side cursors, triggers, views, XA transaction

Data partitioning, row-based replication, event scheduler

Database terminology

Relational database – database objects are related

Database instances – same database, several manifestations so several people can work on it at the same time

In-memory – database which is in working memory

Table – collection of related columns & rows

Record – a row in a table

Primary key – way of identifying each record using one or a combination of columns

Different relationship types

One-to-many – records in 1 table match one or more in another table

One-to-one – records in 1 table match those in 1 other table

SQL – standardized language used to query data in tables

Query – obtain values stored in fields of a table

Recordset – result of a specific query

Join – special query to merge data from several different tables

Referential integrity – foreign key in one table matches rows in the main table

DDL – Data Definition Language, refers to database schema & descriptions

DML – Data Manipulation Language, includes statements like UPDATE, INSERT, DELETE etc.

DCL – Data Control Language – commands to control database access

Locking – restrict multiple users manipulating the same record at the same time

Rollback – reverse all changes made by SQL statements

Atomicity – a single cell can’t contain multiple values, or, 1 transaction must be done completely, parts of a transaction can’t be successful & the rest can’t fail the whole thing has to run or nothing runs

Normalization of data

It’s a Data organization process. We breakdown the data into related group & each group will have it’s separate table & then we define the rules & relationships between tables.

This improves database flexibility & organization.

Normal forms – measure depth of normalization

1 NF –

eliminate repletion of data

separate tables for sets of related data

use primary keys

2NF

Separate tables for values that apply to multiple records

Use of foreign/compound keys

3NF

Elimination of fields that are not dependent on the key

Eg: A customers can have multiple orders. Instead of having a single table that stores customer details along with order details, store customer info in 1 table & order info. This is 1NF. In another & have a foreign key that points to customers. This is 2NF. We see that we are storing Zip code, City & street name of customer but realize we can get all these things from Zip code so we remove City & other fields that can be derived from Zip code.

Structuring data with Normalization

EF Codd – developed the relational model of data representation

Before Codd, DBs were laregely unorganized & didn’t have a theoretical foundation. He used mathematical logic to come up with relational data models & also came up with the first abstract data model.

Boyce-Codd Normal Form (BCNF) a.k.a 3.5 NF

The DB must already be in 3NF

The form refers to dependency of objects

If 1 column implies another, then one of these columns must be super key of the other

In this relationship A->B, A implies B so B is dependent on A, therefore A has to be a super key.