>> Relational Databases >>

Pros:

- Simplicity
- Data Accuracy
- Easy Access
- Data Integrity
- Flexibility
- Security

Cons:

- Performance Issues
- Long Time to Set-up
- No Support of complex data type

MySQL >>

WordPress + Drupal + joomla

Also used by >> YouTube + Flicker + twitter

Functionality (pros):

- Performance Monitoring
- Support of >> Linux, windows, Mac OS & others
- Password Encryption
- Great performance of up to 50M data row

Cons:

- Slow data transaction
- Weak debugging algorithms

SQL Server >>

Functionality (pros):

- In-Memory Analytics
- Bi Semantic model
- Customization Capabilities
- Integration with oracle DB
- Error Management

Cons:

- Expensive Licensing Plans
- Windows-based servers only

Oracle >> On the Cloud

Functionality (pros):

- 4 levels of data transaction protection
- Grouped transactions
- Real Application cluster
- Multi-OS Support

Cons:

- High Price
- Difficult to find developers
- Not easy to learn & use

IBM DB2

Functionality (pros):

- Powerful SQL Modification
- Efficient memory handling

- Support of IBM infrastructure

Cons:

- Requires Add-ons to unlock full functionality
- Free IBM Support for the first 3 years only

>> Non-Relational Databases >>

Pros:

- Handling Unstructured Data
- Agility
- Readability
- Open-Source

Cons:

- Dependency on a specific database management system
- Limited Functionality
- Hiring Difficulties

Mango DB >>

Functionality (pros):

- Support of various data types
- Ability to distribute data automatically between different servers
- Fast performance

Cons:

- Requires more memory increasingly
- No control of duplication

- Lack of Documentation

Document DB >>

Example > Amazon

Functionality (pros):

- Integration with mongo DB
- Database migration

Cons:

- No control of duplication
- Scarce educational resources
- The database isn't organized well

Cassandra >>

Created by Facebook team

Functionality (pros):

- Scalability
- Support of unstructured, structured & semi-structured data
- Support for multiple data centers

Cons:

- No Acid Support & Aggregates support
- Latency problems (it's not good at reading high volumes of data simultaneously)

How to choose a database??

1- What type of data will you analyzing?

- A lot of factual & Numeric data > Relational database
- Large Amount of messy data > Non-relational database

2- How much data are you dealing with?

- Huge Amount of data > Non-relational database

3- Are you ready to invest time & budget in the setup of your database?

- Relational database > harder to setup / easier to support
- Non-relational database > easier to setup / harder to support

4- Do you need real-time data? Yes

- Non-relational database