RDBMS (SQL Database)	**MongoDB (NoSQL Database)**
Relational database	Non-relational and document-oriented database
Need to design your tables, data structure, relations first, then only you can start coding.	You can start coding without worrying about tables. You can modify your objects at a lesser cost of development.
Supports SQL query language	Supports JSON query language
Does not provide JavaScript client for querying	Provides JavaScript client for querying
Table based	Collection based and key-value pair
Row based	Document based
Column based	Field based
Indexing	Indexing
Primary Key	Primary Key (Default key **_id** provided by MongoDB itself)
Group By	Aggregation
Not that easy to set up.	Comparatively easy to set up and get it running. It’s Java client is also very simple.
Support foreign key	No support for the foreign key. But if you need these type of constraint, you have to handle it in the code itself which is a bit complex.

Support for Joins	No support for Joins. But you can change your document structure and embed the other document inside the first document. But keep in mind that MongoDB has maximum document size of 16MB.
Support for triggers	No Support for triggers
Provides very fine granularity of locking	Provides only one level of locking (i.e. document (row) level). In the previous version of MongoDB(2), it supported collection (table) level locking.
Contains schema which is predefined	Contains dynamic schema
Not suitable for hierarchical data storage	Best suitable for hierarchical data storage
Vertically scalable – increasing RAM	Horizontally scalable – add more servers (i.e Sharding)
SQL injection vulnerability	Unaffected by SQL injection
Emphasizes on ACID properties (Atomicity, Consistency, Isolation and Durability)	Emphasizes on CAP theorem (Consistency, Availability, and Partition tolerance)
Slower as compared to NoSQL databases	MongoDB is almost 100 times faster than traditional database systems.