**Comparative Study of Databases (supported on cloud )**

It is also important to differentiate between cloud databases which are relational as opposed to non-relational or NoSQL:

* **SQL database**, such as [NuoDB](http://en.wikipedia.org/wiki/NuoDB), [Oracle Database](http://en.wikipedia.org/wiki/Oracle_Database), [Microsoft SQL Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server), and [MySQL](http://en.wikipedia.org/wiki/MySQL" \o "MySQL), are one type of database which can be run on the cloud (either as a Virtual Machine Image or as a service, depending on the vendor). SQL databases are difficult to scale, although cloud database services based on SQL are attempting to address this challenge.
* **NoSQL databases**, such as [Apache Cassandra](http://en.wikipedia.org/wiki/Apache_Cassandra), [CouchDB](http://en.wikipedia.org/wiki/CouchDB" \o "CouchDB) and [MongoDB](http://en.wikipedia.org/wiki/MongoDB" \o "MongoDB), are another type of database which can run on the cloud. NoSQL databases are built to service heavy read/write loads and are able scale up and down easily, and therefore they are more natively suited to running on the cloud. However, most contemporary applications are built around an SQL data model, so working with NoSQL databases often requires a complete rewrite of application code.

**MySQL:**

MySQL is offered under two different editions: the open source MySQL Community Server and the commercial [Enterprise Server](http://en.wikipedia.org/wiki/MySQL_Enterprise). MySQL Enterprise Server is differentiated by a series of commercial extensions which install as server plug-in, but otherwise shares the version numbering system and is built from the same code base.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, [Joomla](http://en.wikipedia.org/wiki/Joomla" \o "Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress" \o "WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB" \o "PhpBB), [MyBB](http://en.wikipedia.org/wiki/MyBB), [Drupal](http://en.wikipedia.org/wiki/Drupal) and other software. MySQL is also used in many high-profile,large-scale websites including Wikipedia, Google (though not for searches), Facebook,Twitter, Flickr,[[21]](http://en.wikipedia.org/wiki/MySQL#cite_note-22) and YouTube.[[22]](http://en.wikipedia.org/wiki/MySQL#cite_note-23)

**PostgreSQL:**

PostgreSQL works on any of the following instruction set architectures: x86 and x86-64 on Windows and other operatings systems; other than Windows: IA-64 PowerPC 64, S/390, S/390x, SPARC, SPARC 64, Alpha, ARMv8-A (64-bit)and older ARM (32bit), MIPS, MIPSel, M68k, and PA-RISC. It is also known to work on M32R, NS32k, and VAX. In addition to these, it is possible to build PostgreSQL for an unsupported CPU by disabling spinlocks.

**SQLite :**

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example it has partial support for triggers, and it can't write to views (however it supports INSTEAD OF triggers that provide this functionality). While it supports complex queries, it still has limited ALTER TABLE support, as it can't modify or delete columns. SQLite uses an unusual type system for an SQL-compatible DBMS; instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is *dynamically typed*. Moreover, it is *weakly typed* in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the column's preferred type is integer). A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products.

**Microsoft SQL Server:**

 It also has the ability to link to data in its existing location and use it for viewing, querying, editing, and reporting. This allows the existing data to change while ensuring that Access uses the latest data. It can perform heterogeneous joins between data sets stored across different platforms. Access is often used by people downloading data from enterprise level databases for manipulation, analysis, and reporting locally.

**Microsoft Access:**

Users can create tables, queries, forms and reports, and connect them together with [macros](http://en.wikipedia.org/wiki/Macro_(computer_science)). Advanced users can use VBA to write rich solutions with advanced [data manipulation](http://en.wikipedia.org/wiki/Data_manipulation) and user control. Access also has report creation features that can work with any data source that Access can "access".

The original concept of Access was for end users to be able to "access" data from any source. Other features include: the import and export of data to many formats including Excel, Outlook, ASCII, dBase, Paradox, FoxPro, SQL Server, Oracle, ODBC, etc. It also has the ability to link to data in its existing location and use it for viewing, querying, editing, and reporting.

**IBM DB2:**

IBM DB2 is a family of database server products developed by IBM. These products all support the relational model, but in recent years some products have been extended to support object-relational features and non-relational structures, in particular XML.

Other things are there but we can always use MySQL as its free and trusted by many like Twitter,Facebook YouTube.

The following links will guide you through:

<https://docs.djangoproject.com/en/1.6/topics/db/multi-db/>

<https://docs.djangoproject.com/en/dev/topics/db/multi-db/>