# Static DB

Sourav Das | 8100448204 | souravbumbadas25@gmail.com

# Description

Static DB is a NoSQL or SL-SQL (Schema Less – SQL) Database Written entirely in C++.

This database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in RDBMS.

### Features**: -**

1. **Variable Caching System**:

* Cache Size can be Change In-between statements.

1. **Convert JSON to BSON**:

* JSON object are converted to Faster Binary Formats- BSON.

1. **Very Light Weight**:

* Execution of Statement takes minimal time due to raw-data formatting.

1. **Embedded Database**:

* Static DB is not a client–server database engine. It is embedded into the end program. Making it enormously fast in running query as no inter – process communication takes place.

## Advantages are: -

1. Simplicity of Design.
2. Simpler "horizontal" scaling to clusters of machines (which is a problem for relational databases)
3. Operations are faster in NoSQL.
4. Finer Control Over Datasets.

# Overview: -

## SL-SQL statements: -

1. **Create**

Creating a document or a key value (K-V pair) in the database.



A document can be any JSON document.



All Document are Schema Less. Any key can represent anything.





1. **Get**

Retrieving Documents from the database.



Get command is not for querying data it is only for data retrival.





****

1. **Insert**

Inserting Values into the existing Documents.









1. **Delete**

Deleting keys or values from documents.





**>>>** **DELETE** kilo**;**

1. **Update**

Update the values present in the database.







1. **Find**

Query a document for values.

**SQL:** SELECT **\*** FROM test**;**



**SQL:** SELECT item**,** qty FROM test**;**



**SQL:** SELECT item**,** qty FROM test WHERE status **=** 'A'**;**



**SQL:** SELECT item**,** qty FROM test WHERE qty **<** 50**;**



**SQL**: SELECT item**,** qty FROM test WHERE qty **>** 50 AND qty **<** 100**;**



**$lt = < (less than) and &gt = > (greater than).**

**SQL:** SELECT item**,** qty FROM test WHERE qty **>=** 50 AND qty **<=** 100**;**

****

**$eq is =(equal to)**

**SQL:** SELECT item**,** qty FROM test WHERE qty **<** 50 OR status **=** 'D'**;**



1. **Commit**

Write all the Document to Permanent Disk.

**>>>** **COMMIT**;

RETURN MSG**:** Database Committed Successfully.

1. **Help**

Gives Details Of all the other command.

**>>>** **HELP**;

RETURN MSG**:**

1. **Import**

Execute Database Statement which has been written on a file.

**>>>** **IMPORT** "imports.sdb"

RETURN MSG**:** All statement Imported Successfully.

All statement imported successfully.

1. **Export**

Export all the document present in the database to a file in the specified format.

* JSON – export in JSON format
* SDB – export as Static DB statements

**>>>** **EXPORT** "export.sdb" **SDB**;

RETURN MSG**:** All collection Exported Successfully.

**>>>** **EXPORT** "export.json" **JSON**;

RETURN MSG**:** All collection Exported Successfully.

# Using Static Db for Social Networking

Let’s take an Example

The picture below is a snapshot from a comment section in Facebook.

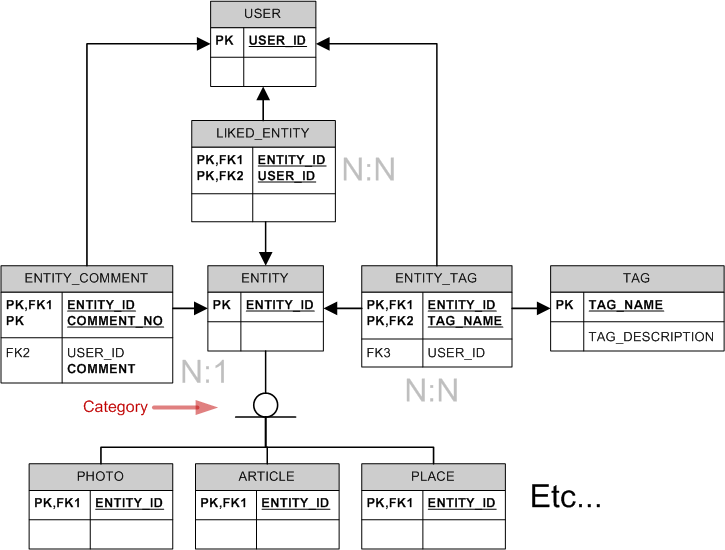


It is observed in the above picture that a hierarchy of comments and replies are accumulated on all posts.

Now if we look into the Facebook site we are left to wondered that how these comments are stored inside a database so that efficient retrieval can be performed.

At first, we may think about a RDBMS i.e. is mapping all these comments and replies to a table structed datastore where all comments and replies are stored with proper user id and no of likes.

* **Let’s see how Using RDBMS we can Store Facebook Comments.**



Now that’s a lot of tables thus this very inefficient as most of the time will be spend of joining the tables.

* Let’s see how Static Db does it



The entire post is stored Way more easily in Static Db than the RDBMS implementation.

The advantages are: -

1. The entire data is stored in one collection (RDBMS uses many table and JOINs)
2. No overhead of table joining.
3. Efficiently queries data that is only needed.
4. Reduced redundancy as foreign key and primary key are not used.

Even static Db support querying making it easily retrievable.

