

**Systems and Database Administration – Journal Entry 5**

**DT211C**

**BSc in Computer Science**

**Tim Browning**

**C14519547**

School of Computing

Dublin Institute of Technology



# MongoDB

**How does MongoDB conduct transactions?**

MongoDB does support transaction-*like* semantics using two-phase commits. It doesn’t support transactions out of the box but you can implement optimistic transactions on your own. MongoDB is restricted to single document writes.

The main idea behind MongoDB transactions:

* Stores the information from the steps in a transaction and stores the transaction in memory.
* The information is retrieved from memory and executed. After each process the state is updated.

In MongoDB, you can emulate a two-phase commit to achieve a comparable result.

[2]*“Consider a scenario where you want to transfer funds from account A to account B. In a relational database system, you can subtract the funds from A and add the funds to B in a single multi-statement transaction. In MongoDB, you can emulate a two-phase commit to achieve a comparable result.”*

1. BEGIN TRANSACTION
2. Debit checking account $20(A).
3. Credit savings account $20(B).
4. COMMIT TRANSACTION

If the database crashes between step 2 and 3 without transaction control, you could end up with one account debited but the other one not being credited.

MongoDB 4.0 will add support for multi-document transactions.

**What features does MongoDB have that would relate to USERS, ROLES and PROFILEs in Oracle?**

1. MongoDB uses BSON format

BSON is a JSON-like storage format. BSON stands for Binary. It adds support for data types like Date and binary that are not supported in JSON.

* Create User with Roles
* Create Administrative User with Roles
* Create User with Authentication Restrictions

The createUser() function uses BSON format.

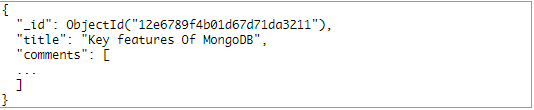


Figure 1 \_BSON format

BSON allows MongoDB to index and map document properties.

1. MongoDB Ad hoc queries

MongoDB uses Ad hoc queries to search for USERS, ROLES and PROFILES. MongoDB is able to support ad hoc queries by indexing BSON.

db.example.find() – Ad hoc function

1. MongoDB uses Indexing

Creating indexes for USERS, ROLES AND PROFILES can improve performance of searches

db.collection.createIndex() – index function

**What auditing is possible in MongoDB?**

The auditing system in MongoDB can record the following operations:

* Schema (DDL),
* Replica set and sharded cluster,
* Authentication and authorization, and
* CRUD operations

Schema DDL defines the different structures in a database.

**How does MongoDB ensure eventual consistency?**

Quorum Reads and Writes: Consistently can be provided by configuring both read and writes to require access the correct number of replicas in order to succeed.

Number of Replicas to Reads and writes:[1] Specifying the number of replicas to read from in a query allows an application to trade-off read query performance and consistency for each use case.

Writes with Unavailable Replicas: [1]When a write cannot succeed at the number of replicas specified for a write request, the database may roll back the writes or employ a mechanism to ensure the write succeeds on the failed replica at a later time.

Resolving Write Conflicts: If a write issue occurs that, an object has different values in both the original and the replica. Timestamps can be used to resolve this error.

MongoDB is consistent by default. Applications can read from replicas where data is eventually consistent by default. Applications can also read from the closest copy of the data when latency is more important than consistency.

**What is replication in MongoDB?**

Replication is the process of keeping identical copies of data on different servers. It is the idea of synchronising data across multiple servers.

MongoDB achieves replication by the use of a replica set. A replica set is a group instances that host the same data set. In a replica, a primary node that receives operations can be replicated onto a secondary node. Each node will have the same operations at the end of a replication

Replication is used to:

* Keep data safe
* Recover corrupted data
* Read scaling
* Ensure a high availability of data

[2]

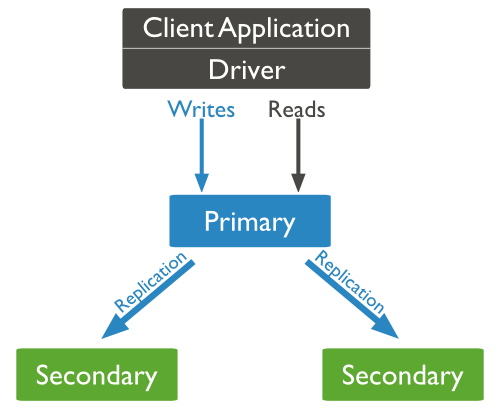


Figure 2 - Primary and secondary nodes (replication)

**What is 'write concern' and how does it work?**

Write concerns report the success of a write operation. There are multiple levels, which are available for reporting. They are implemented to better address the specific needs of applications.

Write Concern features:

* Ensure faster performance
* Make sure operations persist successfully

Unacknowledged: MongoDB does not acknowledge the receipt of write operations. It is similar to ignoring errors but drivers will attempt to handle errors

Acknowledged: The mongod confirms the receipt of the write operation. Acknowledged write concern allows clients to catch network and duplicate keys. This is default write concern.

Journaled: The MongoDB acknowledges the write operation only after committing the data to the journal. This write concern ensures that MongoDB can recover the data following a shutdown

Replica Acknowledged: [3] With replica acknowledged write concern, you can guarantee that the write operation propagates to additional members of the replica set

[2] “Perform Two Phase Commits — MongoDB Manual 3.6,” https://github.com/mongodb/docs/blob/master/source/tutorial/perform-two-phase-commits.txt[1] “FAQ,” *MongoDB*. [Online]. Available: https://www.mongodb.com/faq. [Accessed: 16-Mar-2018].

[2] tutorialspoint.com, “MongoDB Replication,” *www.tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/mongodb/mongodb\_replication.htm. [Accessed: 16-Mar-2018].

[3] “What is Write Concern in MongoDB? - DZone Java,” *dzone.com*. [Online]. Available: https://dzone.com/articles/what-write-concern-mongodb. [Accessed: 16-Mar-2018].

. [Online]. Available: https://docs.mongodb.com/manual/tutorial/perform-two-phase-commits. [Accessed: 12-Mar-2018].