**DATA TYPES : MongoDB Server stores data using the** [**BSON**](https://www.mongodb.com/docs/manual/reference/bson-types/) **format which supports some additional data types that are not available using the** [**JSON**](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-JSON) **format.**

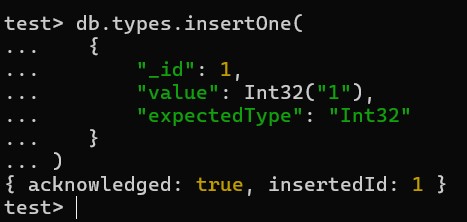
* **Date**
* **Int32**
* **Decimal**
* **Timestamp**

**DATE :**

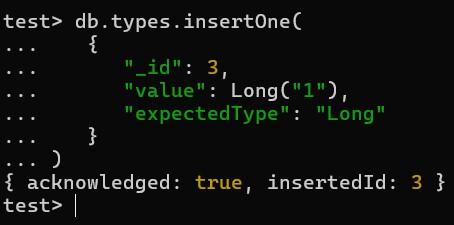
* **Date() method which returns the current date as a string.**
* **new Date() constructor which returns a Date object using the ISODate()wrapper.**
* **ISODate() constructor which returns a Date object using the ISODate()wrapper.**

**Int32 :**

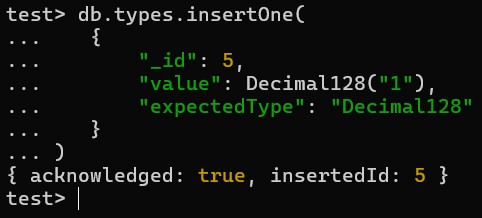
* **If a number can be converted to a 32-bit integer, mongosh will store it as Int32. If not, mongosh defaults to storing the number as a Double. Numerical values that are stored as Int32 in mongosh would have been stored by default as Double in the mongo shell.The** [**Int32()**](http://mongodb.github.io/node-mongodb-native/3.6/api/Int32.html) **constructor can be used to explicitly specify 32-bit integers.**



**Long :[Long()](http://mongodb.github.io/node-mongodb-native/3.6/api/Long.html) constructor can be used to explicitly specify a 64-binteger.**



**Decimal128 :** [**Decimal128()**](http://mongodb.github.io/node-mongodb-native/3.6/api/Decimal128.html) **values are 128-bit decimal-based floating-point numbers that emulate decimal rounding with exact precision.**



**Timestamp : MongoDB uses a** [**BSON**](https://www.mongodb.com/docs/manual/reference/bson-types/#timestamps)[**Timestamp**](https://www.mongodb.com/docs/manual/reference/bson-types/#timestamps) **internally in the** [**oplog**](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-oplog)**. The**

**Timestamp type works similarly to the Java Timestamp type. Use the** [**Date**](https://www.mongodb.com/docs/mongodb-shell/reference/data-types/#std-label-mongo-shell-date-type) **type for operations involving dates. A Timestamp signature has two optional parameters.**

# WHERE CLAUSE,AND,OR&CRUD

**LOAD THE DOCUMENT**

* **Download the student csv from this** [**link**](https://drive.google.com/file/d/1L3FhU59Tw-a-o-8esmKrhb4GhXGW_soe/view?usp=sharing)
* **Import the data to the collection created** [**link**](https://www.mongodb.com/docs/compass/current/import-export/)

**WHERE : Given a Collection you want to FILTER a subset based on a condition. That is the place WHERE is used.For queries that cannot be done any other way, there are "$where" clauses, which allow you to execute arbitrary JavaScript as part of your query. This allows you to do (almost) anything within a query. For security, use of "$where" clauses should be highly restricted or eliminated. End users should never be allowed to execute arbitrary "$where" clauses.**





**AND : Given a Collection you want to FILTER a subset based on multiple conditions.This operator is used to perform logical AND operation on the array of one or more expressions and select or retrieve only those documents that match all the given expression in the array.**

* **This operator performs short-circuit evaluation.**
* **If the first expression of $and operator evaluates to false, then MongoDB will not evaluate the remaining expressions in the array.**
* **You can also use AND operation implicitly with the help of comma(, ).**



**OR : Given a Collection you want to FILTER a subset based on multiple conditions but Any One is Sufficient.**

* **You can use this operator in methods like find(), update(), etc. according to your requirements.**
* **You can also use this operator with text queries, GeoSpatial queries, and sort operations.**
* **When MongoDB evaluating the clauses in the $or expression, it performs a collection scan.**

