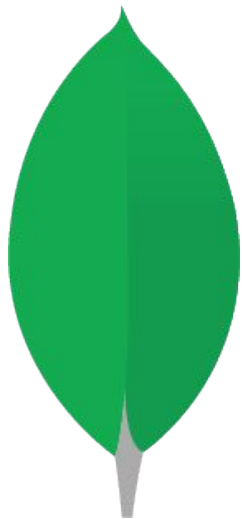

Node Driver

Adventures in Document Mapping and Transactions





mongoose

- type casting
- validation
- query building
- hooks

`mongodb` driver

- no logo
- no cool name
- ?

Example: User

```
class User {  
  constructor(doc = {}) {  
    this._id = doc._id || new ObjectId();  
    this.firstName = doc.firstName;  
    this.lastName = doc.lastName;  
  }  
  getFullName() { return `${this.firstName} ${this.lastName}`; }  
  static map(doc) { return new User(doc); }  
  static unmap(user) {  
    return { _id: user._id, firstName: user.firstName, lastName: user.lastName };  
  }  
}
```

Example: User (continued)

```
const coll = db.collection('foo', {
  map: User.map,
  unmap: User.unmap
});

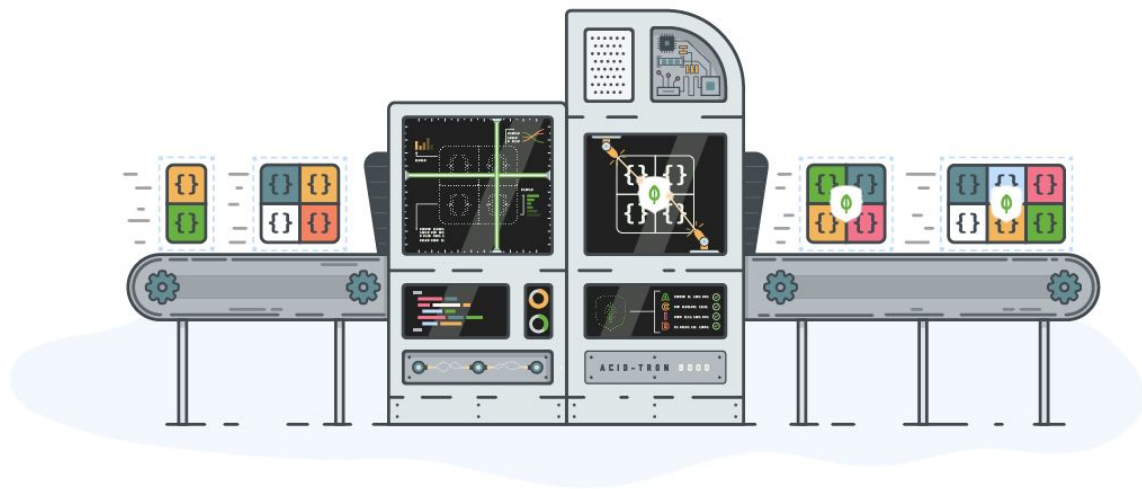
const user = new User({ firstName: 'john', lastName: 'doe' });
const insertResult = await coll.insertOne(user);
const lookedUpUser =
  await coll.find({ _id: result.insertedId });
assert.deepEqual(user, lookedUpUser);
```



Plugins?

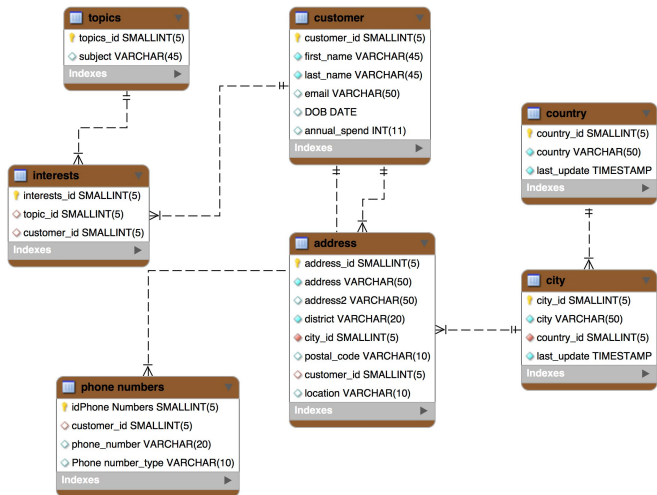
MongoDB 4.0:

Multi-Document ACID Transactions



Data Models and Transactions

Different databases take **different approaches**



Tabular (Relational) Database

Related data split across multiple records and tables.
Multi-record transactions essential

```
_id: 12345678
> name: Object
> address: Array
> phone: Array
email: "john.doe@mongodb.com"
dob: 1966-07-30 01:00:00.000
✓ interests: Array
  0: "Cycling"
  1: "IoT"
```

Document Database

Related data contained in a single, rich document.
Transaction scoped to the document

Data Models and Transactions

For many apps, single document transactions meet data integrity needs

```
_id: 12345678
> name: Object
> address: Array
> phone: Array
  email: "john.doe@mongodb.com"
  dob: 1966-07-30 01:00:00.000
  interests: Array
    0: "Cycling"
    1: "IoT"
```

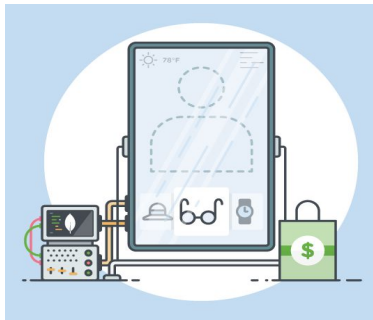
Document Database

Multi-Document Transaction Examples



Finance

Account debit/credit



Retail

Insert orders, decrement inventory



Telco

Insert call detail, update subscriber's plan

MongoDB Multi-Document Transactions



Just like relational transactions

- Multi-statement, familiar relational syntax
- Easy to add to any application
- Multiple documents in 1 or many collections

ACID guarantees

- Snapshot isolation, all or nothing execution
- No performance impact for non-transactional operations

Schedule

- MongoDB 4.0, Summer '18: replica set
- MongoDB 4.2: extended to sharded clusters

Syntax*

```
await client.withSession(session =>
  async function () {
    session.startTransaction();
    try {
      await collection.insertOne(doc1, { session });
      await collection.insertOne(doc2, { session });
      session.commitTransaction();
    } catch (e) {
      session.abortTransaction();
    }
  }
);
```

Natural for developers

- Idiomatic to the programming language
- Familiar to relational developers
- Simple

Comparing Syntax with Relational

```
db.start_transaction()  
cursor.execute(orderInsert, orderData)  
cursor.execute(stockUpdate, stockData)  
db.commit()
```



```
s.start_transaction()  
orders.insert_one(order, session=s)  
stock.update_one(item, stockUpdate, session=s)  
s.commit_transaction()
```



*Python. Syntax is subject to change



Thanks!

@mbroadst