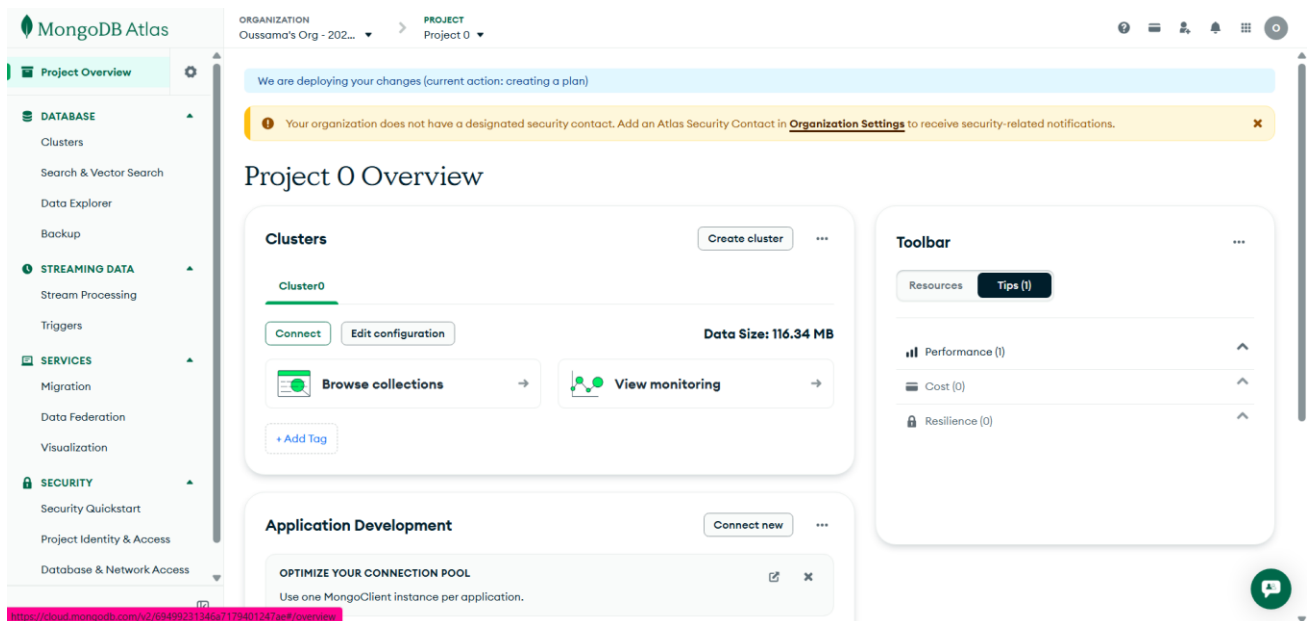


# Lab 4: Initiation à MongoDB

## Oussama Oukhmir 2ème Année Cycle d'Ingénieur GI 2025-2026 BIG DATA

### 1. Création d'un compte et déploiement d'un cluster



### 2. Connexion à MongoDB Atlas

```
PS C:\Users\oussa> mongosh --version or mongo --version
2.5.10
PS C:\Users\oussa> mongosh "mongodb+srv://cluster0.rwjga68.mongodb.net/" --apiVersion 1 --username oukhmir_oussama
Enter password: *****
Current Mongosh Log ID: 6949a57c16df987b671e2620
Connecting to: mongodb+srv://<credentials>@cluster0.rwjga68.mongodb.net/?appName=mongosh+2.5.10
Using MongoDB: 8.0.17 (API Version 1)
Using Mongosh: 2.5.10

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

Atlas atlas-jh69i3-shard-0 [primary] test> |
```

### 3. Création d'une base de données et d'une collection

- ✓ Pour se placer dans une base
- ✓ Créer une collection nommée users
- ✓ Lister les collections de la base
- ✓ Insérer un document JSON dans la collection users
- ✓ Insérer maintenant plusieurs tuples dans la collection users

```
mongosh mongodb+srv://<a> x + v
Enter password: *****
Current Mongosh Log ID: 6949a57c16df987b671e2620
Connecting to:  mongodb+srv://<credentials>@cluster0.rwjga68.mongodb.net/?appName=mongosh+2.5.10
Using MongoDB:  8.0.17 (API Version 1)
Using Mongosh:  2.5.10

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

Atlas atlas-jh69i3-shard-0 [primary] test> show db
MongoshInvalidInputError: [COMMON-10001] 'db' is not a valid argument for "show".
Atlas atlas-jh69i3-shard-0 [primary] test> show dbs
sample_mflix  114.65 MiB
admin          360.00 KiB
local         3.02 GiB
Atlas atlas-jh69i3-shard-0 [primary] test> use BankDB
switched to db BankDB
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.createCollection("users")
{ ok: 1 }
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.insertOne({ name: "Mohamed", age: 28, city: "Casablanca" })
{
  acknowledged: true,
  insertedId: ObjectId('6949a94516df987b671e2621')
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.insertMany([
...   { name: "Alice", age: 25, city: "Paris" },
...   { name: "Bob", age: 30, city: "Lyon" },
...   { name: "Charlie", age: 28, city: "Marseille" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6949a94c16df987b671e2622'),
    '1': ObjectId('6949a94c16df987b671e2623'),
    '2': ObjectId('6949a94c16df987b671e2624')
  }
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

## 4.Requêtes pour récupérer des données

- ✓ L'objet (javascript) implicite, db, permet de soumettre des demandes d'exécution de certaines méthodes. Afficher tous les utilisateurs de la collection:

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.insertMany([
...   { name: "Alice", age: 25, city: "Paris" },
...   { name: "Bob", age: 30, city: "Lyon" },
...   { name: "Charlie", age: 28, city: "Marseille" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6949a94c16df987b671e2622'),
    '1': ObjectId('6949a94c16df987b671e2623'),
    '2': ObjectId('6949a94c16df987b671e2624')
  }
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.find()
[
  {
    _id: ObjectId('6949a94516df987b671e2621'),
    name: 'Mohamed',
    age: 20,
    city: 'Casablanca'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2622'),
    name: 'Alice',
    age: 25,
    city: 'Paris'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2623'),
    name: 'Bob',
    age: 30,
    city: 'Lyon'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2624'),
    name: 'Charlie',
    age: 28,
    city: 'Marseille'
  }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

- ✓ Compter le nombre de documents dans la collection et Récupérer un utilisateur spécifique :

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
4
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.findOne({ name: "Alice" })
{
  _id: ObjectId('6949a94c16df987b671e2622'),
  name: 'Alice',
  age: 25,
  city: 'Paris'
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.find({ age: { $gt: 20 } })
[
  {
    _id: ObjectId('6949a94c16df987b671e2622'),
    name: 'Alice',
    age: 25,
    city: 'Paris'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2623'),
    name: 'Bob',
    age: 30,
    city: 'Lyon'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2624'),
    name: 'Charlie',
    age: 28,
    city: 'Marseille'
  }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.updateOne({ name: "Alice" }, { $set: { age: 26 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

- ✓ Filtrer les utilisateurs par âge (age>20) et Modifier l'âge d'un utilisateur et Supprimer un utilisateur et supprimer une collection:

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.findOne({ name: "Alice" })
{
  _id: ObjectId('6949a94c16df987b671e2622'),
  name: 'Alice',
  age: 25,
  city: 'Paris'
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.find({ age: { $gt: 20 } })
[
  {
    _id: ObjectId('6949a94c16df987b671e2622'),
    name: 'Alice',
    age: 25,
    city: 'Paris'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2623'),
    name: 'Bob',
    age: 30,
    city: 'Lyon'
  },
  {
    _id: ObjectId('6949a94c16df987b671e2624'),
    name: 'Charlie',
    age: 28,
    city: 'Marseille'
  }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.updateOne({ name: "Alice" }, { $set: { age: 26 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.deleteOne({ name: "Charlie" })
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.users.drop()
true
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

## 5.Importation de fichier dans une collection

- ✓ Télécharger le fichier transactions.json. et Après avoir créé la base de données appelée « BankDB » et la collection « transactions »,importer le fichier dans la collection

```
Administrateur: Windows Po...
PS C:\Users\oussa\Documents> mongoimport --uri "mongodb+srv://oukhmir_oussama@cluster0.rwjga68.mongodb.net/BankDB" --collection transactions --file transact
ions.json --jsonArray
Enter password for mongo user:
2025-12-22T21:43:17.214+0100    connected to: mongodb+srv://[**REDACTED**]@cluster0.rwjga68.mongodb.net/BankDB
2025-12-22T21:43:19.959+0100    1000 document(s) imported successfully. 0 document(s) failed to import.
```

- ✓ Dans MongoDB Shell ,connectez-vous à la base de données BankDB et afficher son contenu

```
PS C:\Users\oussa\Documents> mongosh "mongodb+srv://cluster0.rwjga68.mongodb.net/BankDB" --username oukhmir_oussama
Enter password: *****
Current Mongosh Log ID: 6949ae3557aed587f01e2620
Connecting to:      mongodb+srv://<credentials>@cluster0.rwjga68.mongodb.net/BankDB?appName=mongosh+2.5.10
Using MongoDB:      8.0.17
Using Mongosh:      2.5.10

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

Atlas atlas-jh69i3-shard-0 [primary] BankDB> use BankDB
already on db BankDB
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
1060
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.find().pretty()
[
  {
    _id: ObjectId('6949ad657d6782d09c255372'),
    'Transaction ID': 'TXN9412011085',
    'Sender Account ID': 'ACC58958',
    'Receiver Account ID': 'ACC32826',
    'Transaction Amount': 529.62,
    'Transaction Type': 'Withdrawal',
    'Timestamp': '2025-01-17 10:51:00',
    'Transaction Status': 'Success',
    'Fraud Flag': 'False',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice2',
    'Latency (ms)': 11,
    'Slice Bandwidth (Mbps)': 89,
    'PIN Code': 2369,
    'Geolocation': { type: 'Point', coordinates: [ 118.2437, 35.6895 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c255373'),
    'Transaction ID': 'TXN6682052855',
    'Sender Account ID': 'ACC68764',
    'Receiver Account ID': 'ACC53807',
    'Transaction Amount': 1171.23,
    'Transaction Type': 'Transfer',
    'Timestamp': '2025-01-17 10:29:00',
    'Geolocation': { type: 'Point', coordinates: [ 118.2437, 35.6895 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c255374'),
    'Transaction ID': 'TXN3914566506',
    'Sender Account ID': 'ACC45926',
    'Receiver Account ID': 'ACC76590',
    'Transaction Amount': 979.89,
    'Transaction Type': 'Withdrawal',
    'Timestamp': '2025-01-17 10:25:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'False',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice2',
    'Latency (ms)': 15,
    'Slice Bandwidth (Mbps)': 225,
    'PIN Code': 7119,
    'Geolocation': { type: 'Point', coordinates: [ -2.3522, 40.7128 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c255385'),
    'Transaction ID': 'TXN6302601632',
    'Sender Account ID': 'ACC91042',
    'Receiver Account ID': 'ACC98945',
    'Transaction Amount': 129.78,
    'Transaction Type': 'Transfer',
    'Timestamp': '2025-01-17 10:11:00',
    'Transaction Status': 'Success',
    'Fraud Flag': 'True',
    'Device Used': 'Desktop',
    'Network Slice ID': 'Slice1',
    'Latency (ms)': 19,
    'Slice Bandwidth (Mbps)': 106,
    'PIN Code': 5199,
    'Geolocation': { type: 'Point', coordinates: [ 118.2437, 51.5074 ] }
  }
]
Type "it" for more
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

```
'Slice Bandwidth (Mbps)': 61,
'PIN Code': 3326,
'Geolocation': { type: 'Point', coordinates: [ -37.6173, 35.6895 ] }
},
{
  _id: ObjectId('6949ad657d6782d09c255384'),
  'Transaction ID': 'TXN3914566506',
  'Sender Account ID': 'ACC45926',
  'Receiver Account ID': 'ACC76590',
  'Transaction Amount': 979.89,
  'Transaction Type': 'Withdrawal',
  'Timestamp': '2025-01-17 10:25:00',
  'Transaction Status': 'Failed',
  'Fraud Flag': 'False',
  'Device Used': 'Mobile',
  'Network Slice ID': 'Slice2',
  'Latency (ms)': 15,
  'Slice Bandwidth (Mbps)': 225,
  'PIN Code': 7119,
  'Geolocation': { type: 'Point', coordinates: [ -2.3522, 40.7128 ] }
},
{
  _id: ObjectId('6949ad657d6782d09c255385'),
  'Transaction ID': 'TXN6302601632',
  'Sender Account ID': 'ACC91042',
  'Receiver Account ID': 'ACC98945',
  'Transaction Amount': 129.78,
  'Transaction Type': 'Transfer',
  'Timestamp': '2025-01-17 10:11:00',
  'Transaction Status': 'Success',
  'Fraud Flag': 'True',
  'Device Used': 'Desktop',
  'Network Slice ID': 'Slice1',
  'Latency (ms)': 19,
  'Slice Bandwidth (Mbps)': 106,
  'PIN Code': 5199,
  'Geolocation': { type: 'Point', coordinates: [ 118.2437, 51.5074 ] }
}
]
Type "it" for more
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

- ✓ Affiche les collections de la base de données courante.
- ✓ Afficher les informations sur une transaction pour comprendre la structure
- ✓ Combien y a-t-il de transactions dans la base de données ?
- ✓ Trouver toutes les transactions échouées ("transaction status": "failed")
- ✓ Vérifier les transactions suspectes ("fraud flag" : "true")

```

mongosh mongodb+srv://< > + v
Atlas atlas-jh69i3-shard-0 [primary] BankDB> show collections
transactions
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.findOne()
{
  _id: ObjectId('6949ad657d6782d09c255372'),
  'Transaction ID': 'TXN9412011085',
  'Sender Account ID': 'ACC58958',
  'Receiver Account ID': 'ACC32826',
  'Transaction Amount': 529.62,
  'Transaction Type': 'Withdrawal',
  'Timestamp': '2025-01-17 10:51:00',
  'Transaction Status': 'Success',
  'Fraud Flag': 'False',
  'Device Used': 'Mobile',
  'Network Slice ID': 'Slice2',
  'Latency (ms)': 11,
  'Slice Bandwidth (Mbps)': 89,
  'PIN Code': 2369,
  'Geolocation': { type: 'Point', coordinates: [ 118.2437, 35.6895 ] }
}
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.countDocuments()
1000
Atlas atlas-jh69i3-shard-0 [primary] BankDB>

```

```

mongosh mongodb+srv://< > + v
1000
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.find({ "Transaction Status": "Failed" }).pretty()
[
  {
    _id: ObjectId('6949ad657d6782d09c255374'),
    'Transaction ID': 'TXN4361744832',
    'Sender Account ID': 'ACC55704',
    'Receiver Account ID': 'ACC24911',
    'Transaction Amount': 817.55,
    'Transaction Type': 'Deposit',
    'Timestamp': '2025-01-17 10:13:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'False',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice2',
    'Latency (ms)': 8,
    'Slice Bandwidth (Mbps)': 82,
    'PIN Code': 2892,
    'Geolocation': { type: 'Point', coordinates: [ -2.3522, 40.7128 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c255375'),
    'Transaction ID': 'TXN9743166792',
    'Sender Account ID': 'ACC82066',
    'Receiver Account ID': 'ACC33985',
    'Transaction Amount': 570.65,
    'Transaction Type': 'Transfer',
    'Timestamp': '2025-01-17 10:17:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'False',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice2',
    'Latency (ms)': 14,
    'Slice Bandwidth (Mbps)': 246,
    'PIN Code': 9680,
    'Geolocation': { type: 'Point', coordinates: [ -0.1278, 51.5074 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c255377'),
    'Transaction ID': 'TXN3008400720',
    'Sender Account ID': 'ACC96801',
    'Receiver Account ID': 'ACC33985',
    'Transaction Amount': 570.65,
    'Transaction Type': 'Transfer',
    'Timestamp': '2025-01-17 10:17:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'False',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice2',
    'Latency (ms)': 14,
    'Slice Bandwidth (Mbps)': 246,
    'PIN Code': 9680,
    'Geolocation': { type: 'Point', coordinates: [ -0.1278, 51.5074 ] }
  }
]

```

```
mongosh mongodb+srv://<ci> x + v
'Receiver Account ID': 'ACC27800',
'Transaction Amount': 485.37,
'Transaction Type': 'Transfer',
'Timestamp': '2025-01-17 10:50:00',
'Transaction Status': 'Failed',
'Fraud Flag': 'True',
'Device Used': 'Mobile',
'Network Slice ID': 'Slice3',
'Latency (ms)': 20,
'Slice Bandwidth (Mbps)': 173,
'PIN Code': 9631,
'Geolocation': { type: 'Point', coordinates: [ -37.6173, 34.0522 ] }
},
{
  _id: ObjectId('6949ad657d6782d09c255391'),
  'Transaction ID': 'TXN9520068950',
  'Sender Account ID': 'ACC14994',
  'Receiver Account ID': 'ACC16656',
  'Transaction Amount': 495.0,
  'Transaction Type': 'Deposit',
  'Timestamp': '2025-01-17 10:14:00',
  'Transaction Status': 'Failed',
  'Fraud Flag': 'True',
  'Device Used': 'Desktop',
  'Network Slice ID': 'Slice3',
  'Latency (ms)': 10,
  'Slice Bandwidth (Mbps)': 179,
  'PIN Code': 3075,
  'Geolocation': { type: 'Point', coordinates: [ 74.006, 34.0522 ] }
}
]
Type "it" for more
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.find({ "Fraud Flag": true }).pretty()
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.find({ "Fraud Flag": true }).pretty()
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.countDocuments({ "Fraud Flag": true })
0
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.countDocuments({ "Transaction Status": "Failed" })
513
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Nombre total de transactions par statut

```
mongosh mongodb+srv://<ci> x + v
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... { $group: { _id: "$Transaction Status", total: { $count: {} } } }
... ])
[ { _id: 'Failed', total: 513 }, { _id: 'Success', total: 487 } ]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Montant moyen des transactions par type

```
mongosh mongodb+srv://<ci> x + v
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... { $group: { _id: "$Transaction Type", avgAmount: { $avg: "$Transaction Amount" } } }
... ])
[ { _id: 'Transfer', avgAmount: 780.1512032085561 },
  { _id: 'Deposit', avgAmount: 797.6032278481013 },
  { _id: 'Withdrawal', avgAmount: 733.3745806451612 }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Top 5 des comptes qui envoient le plus d'argent

```
mongosh mongodb+srv://<ci> x + v
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... { $group: { _id: "$Sender Account ID", totalSent: { $sum: "$Transaction Amount" } } }, { $sort: { totalSent: -1 } },
... { $limit: 5 }
... ])
[ { _id: 'ACC37810', totalSent: 2757.77 },
  { _id: 'ACC75741', totalSent: 1918.82 },
  { _id: 'ACC89865', totalSent: 1692.55 },
  { _id: 'ACC44804', totalSent: 1497.76 },
  { _id: 'ACC67128', totalSent: 1495.01 }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Calculer le montant total des transactions par type

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... { $group: { _id: "$Transaction Type", total: { $sum: "$Transaction Amount" } } }
... ])
[
  { _id: 'Withdrawal', total: 227346.12 },
  { _id: 'Transfer', total: 291776.55 },
  { _id: 'Deposit', total: 252042.62 }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Vérifier la latence moyenne par "Network Slice ID"

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... { $group: { _id: "$Network Slice ID", avgLatency: { $avg: "$Latency (ms)" } } }
... ])
[
  { _id: 'Slice2', avgLatency: 11.3 },
  { _id: 'Slice1', avgLatency: 11.91640866873065 },
  { _id: 'Slice3', avgLatency: 11.86053412462908 }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Transactions échouées par appareil utilisé

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... { $match: { "Transaction Status": "Failed" } },
... { $group: { _id: "$Device Used", totalFailed: { $count: {} } } }
... ])
[
  { _id: 'Desktop', totalFailed: 243 },
  { _id: 'Mobile', totalFailed: 270 }
]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

✓ Trouver les transactions suspectes (fraude et montant élevé > 1000)

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.find({ "Fraud Flag": "True", "Transaction Amount": { $gt: 1000 } })
[
  {
    _id: ObjectId('6949ad657d6782d09c25537e'),
    'Transaction ID': 'TXN2109277527',
    'Sender Account ID': 'ACC48227',
    'Receiver Account ID': 'ACC93536',
    'Transaction Amount': 1135.8,
    'Transaction Type': 'Deposit',
    'Timestamp': '2025-01-17 10:56:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'True',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice2',
    'Latency (ms)': 11,
    'Slice Bandwidth (Mbps)': 69,
    'PIN Code': 8607,
    'Geolocation': { type: 'Point', coordinates: [ 118.2437, 48.8566 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c255383'),
    'Transaction ID': 'TXN7417651729',
    'Sender Account ID': 'ACC26106',
    'Receiver Account ID': 'ACC21043',
    'Transaction Amount': 1371.8,
    'Transaction Type': 'Transfer',
    'Timestamp': '2025-01-17 10:09:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'True',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice1',
    'Latency (ms)': 8,
    'Slice Bandwidth (Mbps)': 61,
    'PIN Code': 3326,
    'Geolocation': { type: 'Point', coordinates: [ -37.6173, 35.6895 ] }
  },
  {
    _id: ObjectId('6949ad657d6782d09c25538c'),
    'Transaction ID': 'TXN2214150284',
    'Sender Account ID': 'ACC48650',
    'Receiver Account ID': 'ACC48650',
    'Transaction Amount': 1000,
    'Transaction Type': 'Transfer',
    'Timestamp': '2025-01-17 10:09:00',
    'Transaction Status': 'Failed',
    'Fraud Flag': 'True',
    'Device Used': 'Mobile',
    'Network Slice ID': 'Slice1',
    'Latency (ms)': 8,
    'Slice Bandwidth (Mbps)': 61,
    'PIN Code': 3326,
    'Geolocation': { type: 'Point', coordinates: [ -37.6173, 35.6895 ] }
  }
]
```



- ✓ Combien transactions ont un montant entre 100 et 200 ans ?

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.countDocuments({ amount: { $gte: 100, $lte: 200 } })
0
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

- ✓ Afficher Montant total des transaction

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([{$group: { _id: null, totalAmount: { $sum: "$Transaction Amount" } } }])
[ { _id: null, totalAmount: 771165.29 } ]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

- ✓ afficher **Taux d'échec des transactions (%)** et Afficher le Nombre de transactions frauduleuses

```
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.aggregate([
... {$group: { _id: "$Transaction Status", count: { $count: {} } } }
... ])
[ { _id: 'Success', count: 487 }, { _id: 'Failed', count: 513 } ]
Atlas atlas-jh69i3-shard-0 [primary] BankDB> db.transactions.countDocuments({ "Fraud Flag": "True" })
481
Atlas atlas-jh69i3-shard-0 [primary] BankDB> |
```

## 6.Manipuler MapReduce sur la collection transactions

- ✓ Exécutez la cellule suivante dans Google Colab pour installer PyMongo :

!pip install pymongo

The screenshot shows a Google Colab notebook with the following content:

```
[1]
✓ 10 s

!pip install pymongo

Collecting pymongo
  Downloading pymongo-4.15.5-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (22 kB)
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo)
  Downloading dnspython-2.8.0-py3-none-any.whl.metadata (5.7 kB)
  Downloading pymongo-4.15.5-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (1.7 MB)
    1.7/1.7 MB 65.5 MB/s eta 0:00:00
  Downloading dnspython-2.8.0-py3-none-any.whl (331 kB)
    331.1/331.1 kB 24.6 MB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.8.0 pymongo-4.15.5

[82]
✓ 0 s

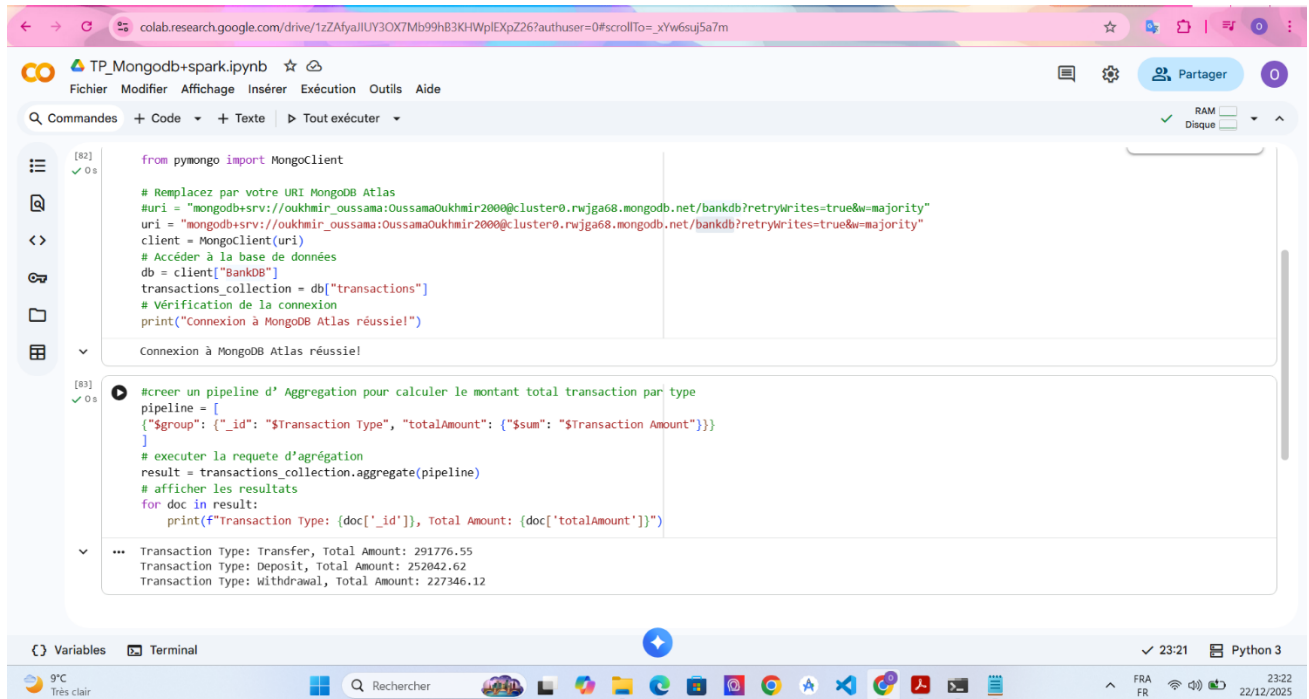
from pymongo import MongoClient

# Remplacez par votre URI MongoDB Atlas
uri = "mongodb+srv://oukhmir_oussama:OussamaOukhmir2000@cluster0.rwjga68.mongodb.net/bankdb?retrywrites=true&w=majority"
uri = "mongodb+srv://oukhmir_oussama:OussamaOukhmir2000@cluster0.rwjga68.mongodb.net/bankdb?retrywrites=true&w=majority"
client = MongoClient(uri)
# Accéder à la base de données
db = client["BankDB"]
transactions_collection = db["transactions"]
# Vérification de la connexion
print("connexion à MongoDB Atlas réussie!")

Connexion à MongoDB Atlas réussie!

[83]
#créer un pipeline d' Aggregation pour calculer le montant total transaction par type
```

- ✓ Ensuite, Se connecter à MongoDB Atlas et Etant donné que map\_reduce() est obsolète, utilisez le framework d'agrégation de MongoDB pour additionner les montants des transactions par type de transaction



The screenshot shows a Google Colab notebook titled "TP\_Mongodb+spark.ipynb". The notebook contains two code cells. The first cell connects to MongoDB Atlas using the pymongo library. The second cell creates an aggregation pipeline to calculate the total amount of transactions by type. The output of the second cell shows the results of the aggregation.

```
[82] ✓ 0 s
from pymongo import MongoClient

# Remplacez par votre URI MongoDB Atlas
#uri = "mongodb+srv://oukhir_oussama:OussamaOukmir2000@cluster0.rvjga68.mongodb.net/bankdb?retryWrites=true&w=majority"
uri = "mongodb+srv://oukhir_oussama:OussamaOukmir2000@cluster0.rvjga68.mongodb.net/bankdb?retryWrites=true&w=majority"
client = MongoClient(uri)
# Accéder à la base de données
db = client["BankDB"]
transactions_collection = db["transactions"]
# Vérification de la connexion
print("Connexion à MongoDB Atlas réussie!")

Connexion à MongoDB Atlas réussie!

[83] ✓ 0 s
#créer un pipeline d' Aggregation pour calculer le montant total transaction par type
pipeline = [
    {"$group": {"_id": "$Transaction Type", "totalAmount": {"$sum": "$Transaction Amount"}}}
]
# executer la requete d'agrégation
result = transactions_collection.aggregate(pipeline)
# afficher les résultats
for doc in result:
    print(f"Transaction Type: {doc['_id']}, Total Amount: {doc['totalAmount']}")

...
Transaction Type: Transfer, Total Amount: 291776.55
Transaction Type: Deposit, Total Amount: 252042.62
Transaction Type: Withdrawal, Total Amount: 227346.12
```

Variables Terminal

9°C Très clair

Rechercher

FRA FR 23:21 Python 3 22/12/2025

## 7. Dashboard avec MongoDB Atlas

<https://charts.mongodb.com/charts-project-0-ixdjewa/public/dashboards/6949c5a0-2d7b-48bf-8bc7-43dee00a4c00>

