Databasteknik

MongoDB

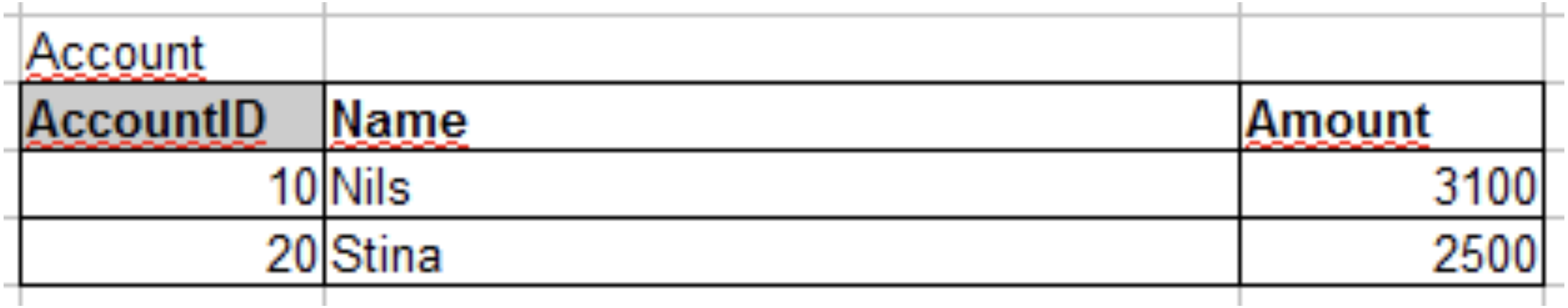
Utbildare: Mikael Olsson

mikael.olsson@emmio.se
076-174 90 43

NACKADEMIN

Transaktioner

Relationsdatabaser



Account

AccountID	Name	Amount
10	Nils	3100
20	Stina	2500

Transaktioner

Relationsdatabaser

- Stina vill överföra 200:- till Nils.

  -
    ```
    UPDATE Account SET Amount = Amount + 200 WHERE AccountID = 10;
    ```

  -
    ```
    UPDATE Account SET Amount = Amount - 200 WHERE AccountID = 20;
    ```
- Vad händer om systemet kraschar mellan dessa uppdateringar?
- Hur kan vi bibehålla dataintegiteten?

Transaktioner

Relationsdatabaser

```
START TRANSACTION;

UPDATE Account SET Amount = Amount + 200 WHERE AccountID =
10;

UPDATE Account SET Amount = Amount - 200 WHERE AccountID =
20;

COMMIT;
```

MongoDB

Transaktioner

- Transaktioner i MongoDB är delvis mer komplicerade eftersom data kan vara uppdelade på flera noder.
    - https://developer.mongodb.com/quickstart/node-transactions/ - exempel med Node.js
- An operation on a single document is atomic - för operationer i ett dokument kommer allt att funka utan inblandning av oss
- Because you can use embedded documents and arrays to capture relationships between data in a single document structure instead of normalizing across multiple documents and collections, this single-document atomicity obviates the need for multi-document transactions for many practical use cases.

MongoDB

Transaktioner

- For situations that require atomicity of reads and writes to multiple documents (in a single or multiple collections), MongoDB supports multi-document transactions.

- With distributed transactions, transactions can be used across multiple operations, collections, databases and documents.

Transaktioner

- The following mongo shell methods are available for transactions:
    - `Session.startTransaction()`
    - `Session.commitTransaction()`
    - `Session.abortTransaction()`

Transaktioner

Air bnb

- En början på ett exempel
  - Lösningen är för komplicerad för att gå igenom i sin helhet men ett exempel på när man kan behöva använda transaktioner
- Collection users
  - Easily view reservations
  - Reservations embedded in users collection

```
{
  "_id": {"$oid":"5dd589544f549efc1b0320a5"},
  "email": "leslie@example.com",
  "name": "Leslie Yepp",
  "reservations":  [
    {
      "name":"Infinite Views",
      "dates": [
         {"$date": {"$numberLong":"1577750400000"}},
         {"$date": {"$numberLong":"1577836800000"}}
      ],
      "pricePerNight": {"$numberInt":"180"},
      "specialRequests": "Late checkout",
      "breakfastIncluded":true
    },
    {
      "name": "Lovely Loft",
       "dates": [
         {"$date": {"$numberLong": "1585958400000"}}
      ],
      "pricePerNight": {"$numberInt":"210"}
      "breakfastIncluded":false
    }
  ]
}
```

Transaktioner

Air bnb

- Users need to know if the listing is already booked for their travel dates

- Store the dates the listing is reserved in the listingsAndReviews collection

```
{
  "_id": {"$oid":"5dbc20f942073d6d4dabd730"},
  "name":"Infinite Views",
  "summary":"Modern home with infinite views from the infinity pool",
  "property_type": "House",
  "bedrooms": {"$numberInt": "6"},
  "bathrooms":{"$numberDouble":"4.5"},
  "beds":{"$numberInt":"8"},
  "datesReserved": [
    {"$date": {"$numberLong": "1577750400000"}},
    {"$date": {"$numberLong": "1577836800000"}}
  ]
}
```

Transaktioner

Air bnb

- Vi behöver hålla användarens bokningsdatum och rummets bokningsdatum i synk.
- Det är dock komplicerat i MongoDB och kräver involvering av backend
- Kolla gärna lite på egen hand och om inte annat:
    - Minns att MongoDB har stöd för transaktioner!
    - https://developer.mongodb.com/quickstart/node-transactions/

Säkerhet

Authentication

- Authentication is the process of verifying the identity of a client that is trying to connect with a database.

Säkerhet

Authentication

- Authentication is the process of verifying the identity of a client that is trying to connect with a database.
    - Enable access control.
    - Always start by creating an administrator user. Then add additional users as needed.
    - Encrypt all communications between mongod and mongos instances as well as internal and external communications using TLS/SSL.
    - Encrypt data on each MongoDB host using filesystem, device, or physical encryption.
    - Run MongoDB on a trusted network only. Do not allow your database to be routable outbound to the public internet, even when inside a trusted network, and don't let it run on any more interfaces than it has to. This prevents a bad actor from having a means of moving your data from the server to another offsite location (for hardware, that is—software-based routers and static routing tables can still be modified by hackers).
    - Make a habit of tracking changes in both the database and data.

Säkerhet

Authorization/role-based security

- Role-based access control (RBAC) is one of MongoDB's best features. While you can find well-defined roles within MongoDB that can cover most users, custom roles can be created as well.

- A role essentially determines what permissions a user has and what he/she can access. Once a user has been defined by a role, the user cannot access the system beyond it.

Säkerhet

Access-control best practices

- Giving users too much access gives way to potential misuse of privileges, which is why exercising due diligence is important when assigning roles.
- A report by Gartner reveals that 62% of company insiders indulged in actions that gave them a second income. This is usually done by misusing company data.
- Too much access is an issue that constantly figures in the top-10 lists of IT issues companies wrestle with.

Säkerhet

Guidelines

- Understand each role right down to its most minute detail. The better roles are understood, the more accurately privileges can be assigned to them.

- It's best to follow a principle of least privilege. Assign to users only those roles they need to get the job done. More privileges can be assigned if needed.

- Create a new MongoDB user for each application/use case of the database. For example, create a user named "webapp" (with least privileges) to run your web application, while creating another user, "analytics," (perhaps read-only) for a business analytics system. This creates isolated privileges and allows granular control of application usage of the database.

Säkerhet

Guidelines

- Create a resource to help users understand basic information security. Run drills to ensure employees understand the security requirements and are clear on what consequences they can face should the requirements not be met.

- Revoke access of users no longer with your organization as soon as they leave.

- Implement user provisioning software to manage multiple users more efficiently.

- MongoDB 3.5 onwards comes with client source filtering that allows you to filter connections based on IP addresses or IP ranges. Use it to have better control over who can access an environment.

- Implement user-level access control lists to grant permissions to individual users.

Säkerhet

Consider the following steps

- Always set your database management system (DBMS) to require a strong password.
- Stay away from default users and demo databases. Since this information is public, it can be used against you.
- Never use standard usernames like root, user, or app, as they are the easiest to guess.
- Restrict public network access to the greatest extent possible. Only IP addresses that need to talk to your database server should be given access to it. Using a good quality VPN is highly recommended.
- Set up monitoring systems to look out for high CPU usage and I/O activity. Doing so will alert you to unusual patterns that are typical with cyberattacks.

Säkerhet

Consider the following steps

- Again, follow the principle of least privilege when assigning user roles. This cannot be emphasized enough.
- Conduct database audits at regular intervals. The longer your audit trail, the more secure you are.
- Encrypt your backup data. Ransomware attacks have begun to infect backups as well.
- Consider hiring ethical hackers to gain an outsider perspective and probe your database for weaknesses.

Säkerhet

Consider the following steps

- Consider hiring ethical hackers to gain an outsider perspective and probe your database for weaknesses.
  - Always ascertain the identity of a person before accepting any communications from them. Ask yourself:
  - Do I know this person?
  - Do I absolutely have to click on the link or open that attachment?
  - Is the name and email of the person the same as what's in my contact list?
  - Was I expecting an email from them?
  - Keep yourself up to date on the latest goings-on in the security world.

Säkerhet

Ett sista tips

- Digital security is a shifting target that's never the same from one moment to the next. As every business has its own set of strengths and weaknesses, it is best that every organization's systems and policies be based on them.