



SQL bulletin

Group8

Anna Thunberg || Anders Pettersson
Emanuele De Negri || Marko Päätaalo





Frågor?





Projektbeskrivning

Backend-delen av en bulletin-board med hjälp av
node.js & sql



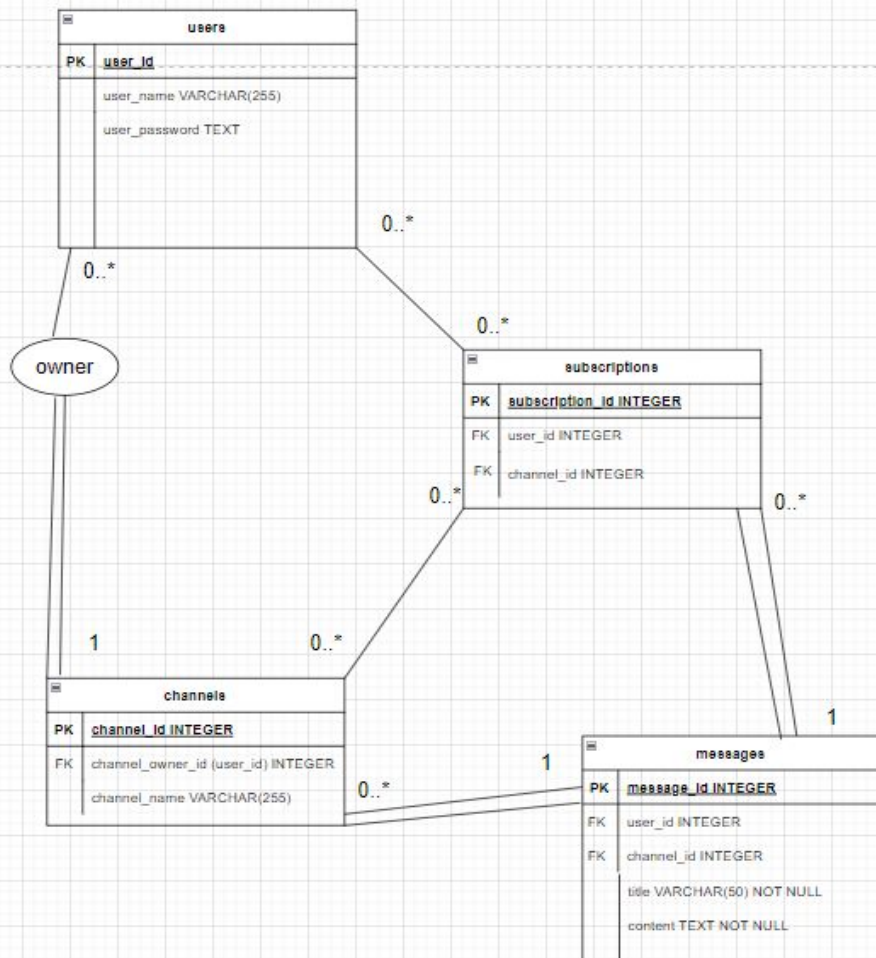


Arbetsmetod

- Github => Branches
- MVC
- Trello – tickets



ER-diagram



Databas-struktur

```
let sql_messages = `CREATE TABLE IF NOT EXISTS messages (  
  message_id INTEGER PRIMARY KEY,  
  user_id INTEGER,  
  channel_id INTEGER,  
  title VARCHAR(50) NOT NULL,  
  content TEXT NOT NULL,  
  FOREIGN KEY (user_id) REFERENCES users(user_id),  
  FOREIGN KEY (channel_id) REFERENCES channels(channel_id)  
);`;
```

```
let sql_subscription = `CREATE TABLE IF NOT EXISTS subscriptions (  
  subscription_id INTEGER PRIMARY KEY,  
  user_id INTEGER,  
  channel_id INTEGER,  
  FOREIGN KEY (user_id) REFERENCES users(user_id),  
  FOREIGN KEY (channel_id) REFERENCES channels(channel_id)  
);`;
```

```
function initDatabase() {  
  let sql_user = `CREATE TABLE IF NOT EXISTS "users" (  
    user_id INTEGER PRIMARY KEY,  
    user_name VARCHAR(255),  
    user_password TEXT  
  )`;  
};
```

```
let sql_channels = `CREATE TABLE IF NOT EXISTS "channels" (  
  channel_id INTEGER PRIMARY KEY,  
  channel_name VARCHAR(255),  
  channel_owner_id INTEGER,  
  FOREIGN KEY(channel_owner_id) REFERENCES users(user_id)  
)`;
```

```
let sql_channel = `CREATE TABLE IF NOT EXISTS channels (  
  channel_id INTEGER PRIMARY KEY,  
  channel_name VARCHAR(255),  
  owner_id INTEGER,  
  FOREIGN KEY (owner_id) REFERENCES users(user_id)  
)`;
```



Endpoint: /channels/post/:id (PUT)

MODEL

```
async function isMessageAuthor(message_id, user_id) {
  return new Promise((resolve, reject) => {
    db.get("SELECT user_id FROM messages WHERE message_id = ?",
      [message_id], (err, row) => {
        if (err) {
          reject(err);
        } else {
          const isAuthor = row && row.user_id === user_id;
          resolve(isAuthor);
        }
      });
  });
}
```

```
async function editMessageById(message_id, title, content) {
  return new Promise((resolve, reject) => {
    let sql = `UPDATE messages SET `;
    const params = [];

    if (title && content) {
      sql += `title = ?, content = ?`;
      params.push(title, content);
    } else if (title) {
      sql += `title = ?`;
      params.push(title);
    } else if (content) {
      sql += `content = ?`;
      params.push(content);
    } else {
      resolve({ changes: false });
      return;
    }

    sql += ` WHERE message_id = ?`;
    params.push(message_id);

    db.run(sql, params, function (err) {
      if (err) {
        reject(err);
      } else {
        if (this.changes > 0) {
          resolve({ changes: true });
        } else {
          resolve({ changes: false });
        }
      }
    });
  });
}
```



Endpoint: /channels/post/:id (PUT)

ROUTES

```
router.put("/post/:id", auth, updateMessage);
```

CONTROLLER

```
async function updateMessage(req, res) {
  const message_id = req.params.id;
  const { title, content } = req.body;
  const user_id = req.user.id;

  if (!title && !content) {
    return res.status(400).json({ error: "Title or content are required" });
  }


  if (title.length > 50) {
    return res.status(400).json({ error: "Title must be less than 50 characters" });
  }

  try {
    const isAuthor = await isMessageAuthor(message_id, user_id);
    if (!isAuthor) {
      return res.status(403).json({ error: "You are not authorized to update this message" });
    }

    const result = await editMessageById(message_id, title, content);

    if (!result.changes) {
      return res.status(404).json({ error: "Message not found or no changes made" });
    }

    res.status(200).json({
      message: "Message updated successfully",
      updatedMessage: { message_id, title, content },
    });
  } catch (error) {
    console.error("Error updating message", error);
    return res.status(500).json({ error: "Internal server error" });
  }
}
```





Grupparbetet

- Utmaningar
- Problem som uppstått
- Delar som fungerat bra
- Delar som har utvecklingspotential att fungera bättre





Frågor?

