[Beshoy Ayad AbdelMalak]

How to not Show Duplicates in SQL

Reference : https://learnsql.com/

Problem:

You'd like to display non-duplicate records in SQL.

Example:

Our database has a table named `City` with data in the columns `id`, `name`, and `country`.

id	name	country
1	Madrid	Spain
2	Barcelona	Spain
3	Warsaw	Poland
4	Cracow	Poland

Let's get the names of the countries without duplicates.

Solution:

We'll use the keyword `DISTINCT`. Here's the query:

```sql
SELECT DISTINCT country FROM City;
```

Here's the result of the query:

country
Spain
Poland

Database Scheme

```sql
CREATE TABLE users(
    id SERIAL PRIMARY KEY,
    firstName VARCHAR(100),
    lastName VARCHAR(100),
    password VARCHAR
);
```

Model

```
export type User = {
 id?: number;
 firstname: string;
 lastname: string;
 password: string;
};

export class UserStore {
sync create(Usr: User): Promise<User> {
   try {
     const conn = await client.connect();

     const sql =
       'INSERT INTO  users (firstname, lastname, password) VALUES($1, $2, $3) RETURNING *
';

     const result = await conn.query(sql, [Usr.firstname, Usr.lastname, password]);

     const user = result.rows[0];

     conn.release();
     return user;
   } catch (err) {
     throw new Error(
       `Unable to create user (${Usr.firstname} ${Usr.lastname}): Error : ${err}`
     );
   }
 }

}
```

Handler

```
import { User, UserStore } from 'Model'
const store = new UserStore();

const create = async (req: Request, res: Response) => {
  const { firstname, lastname, password } = req.body;
  if (
    firstname == undefined ||
    lastname == undefined ||
    password == undefined
  ) {
    res.status(406);
    return res.send(
      'one or more parameters is invaled or typed incorrectly check your data and try again'
    );
  }
  const Usr: User = { firstname, lastname, password };
  try {
    const newUser = await store.create(Usr);
     res.json(token);
    } catch (err) {
    const e = err as Error;
    if (e.message.includes('Unable to create user')) {
      res.status(500).json(e.message);
    } else {
      res.status(401).json(e.message);
    }
  }
};
```