SQL SQL Database 95 Node SQL with Node 121 SQL+ Node (CRUD op.) Node 131 (+ Express)

SQL Database

· Database

It is a collection of data in a format that can be easily accessed. Mary and Control of the State of the

The bearing the same and the same and the

Astronomical Company of the second

3 Benefits:

- can store large data

- features like security, scalability, etc.

-easier to insert, update on delete data

+ SQL VIS NOSQL

SQL NOSQL

Reserve March States

1934M - Nottolladent &

Relational Database Non Relational Database (data stored in) (data stored in document/) Tables key-value/graphs, etc.

eg mysal, Oracle, eg Mongodb, Cassandra, Postgresal, etc. Neoaj, etc.

SQL - Structured Query Language

interact with relational databases.

* Installation - Mysel

- visit www.mysql.com/downloads/

- scool down to " MysQL Community (GPL) Downloads. and click on this link

SOL Databace

agnidation +

- click on "MysQL Installer for Windows"

- click on "download" button (2.2 mb msi file)
- click on "No thanks, just start my download" link

- "install" downloaded file

- select "full" & click on "Next"

- click on "Execute", "Execute", "Next", "Next", ...

- enter root password for when we have to access MysQL server

(Here, I am using = mys91@12345)

- click on "Next", "Next", ..., "Execute", "Finish", "Next". - enter spot password we had just entered and

- click on "Next"

- click on "Execute", "Anish", "Execute"

- Uncheck "Start MySQL shell after setup" - click on "Finish"

> anced remainde Stanchused

> > Tanattalia Allerial

MySQL Workbench software

- on left side panel, first icon is for "Mysal connections", click on @ icon
- "setup New Connection" window will open connection Name = New Connection password = click on "Store in vault" button & enter root password.
- -click on "Test Connection" button. It has to be dialog box for "successfully made the Mysel. Connection" & click on "ok" button.

- DROP DATABASE college;

- dick on connection, we have just made.
- -NOW, you can see some windows such as Navigatos, Query 1 (tile) with editor in which we will write commands (code), outputs, etc.
 - Note: To execute selected code in editor,
 press ctrl + enter

CREATE TABLE table name (

Compart Consults

Mysselv Washbench * Create a database database name - CREATE DATABASE college; → In Marigator Panel = Schemas tab, click on 69 (refresh) icon at the top-right corner, you can see our "college" named database. Tolk Congression Fully and The Harmes diet * Delete a database - DROP DATABASE college; * Use a database To create anything in the database, we have to select that database. - USE college; * Create a Table - CREATE TABLE table-name (datatype column_names constraint. datatype column_name2 constraint,

datatype constraint

column-names

0.9.	* Patabase Quest
CREATE TABLE student	CDARAGATE WARRY &
THICKTHIC (30)	
1.	- DROP PRINBASE ALD
Di Linner de 201x3	AP PAUMING DAG
	: ≥ secon _ paragases :
Insert data into the	table 31 AAD WANDES
INSERT INTO student	
YALUES	
(101, "adam", 25), (102, "eve", 24);	* Table Guesies
	adages =
	A TANKS TANKS &
show Table	Service of the service of the
	at thunder
SELECT * FROM stude	ent;
	ar description for 7
"Result Grid" window	t named table in
rollno name	age
101 adam	25
102 eve	24

7	Database Queries
-)	CREATE DATABASE OF NOT EXTSTS db_name;
->	DROP DATABASE db-name; DROP DATABASE OF EXTSTS db-name;
->	SHOW DATABASES;
->	SHOW TABLES ; LAST & STATE ALOR TROOPS :
	TAISERT TAITO STUDENT
*	Table Queries
-	Create
->	Insert
->	Update
	Alter Short
→ →	Pruncate Delete : Inchie
	Delete ; trobide, MOSE + TOSISE-
101	ships became the last one may be
	Court to the second of the sec

3.75

MAIN

prelies

* Datatypes

- THAR = String (0-255), can store characters of fixed length

 CHAR (50)
- > VARCHAR = string (0-255), can stone characters of upto given length

 VARCHAR (50)
- -> BLOB = string (0-65535), can store binary large object
 BLOB (1000)
- → INT = integer (-2,147,483,648 to 2,147,483,647)

INT

→ TINYINT = înteger (-128 to 127). If we use "TINYINT UNSIGNED" then range will be (0 - 255).

2 TINYINT

→ BIGINT = integer (-9, 223, 372, 036, 854, 775, 808 to 9, 223, 372, 036, 854, 775, 807)

BIGINT

- -> BIT: can store x-bit values. x can range from 1 to 64.

 BIT (2)
- → FLOAT = decimal number with precision to 23 digits

- DOUBLE : decimal number with 24, to 53 xight
 - BOOLEAN : boolean values o on 1

 BOOLEAN

Fed.

- DATE = date in format of YYYY-MM-DD ranging from 1000-01-01 to 9999-12-31

 DATE
 - → YEAR = year in 4 digits format ranging from
 1901 to 2155

 YEAR

2 147 485 849 2

AND SPECIAL PROPERTY OF THE COMMENTAL PROPERTY.

* No Constaints

(di spino)

CREATE TABLE temps (

* Constraints

-> Rules for data in the table

NOT NULL column can not have a null value

UNIQUE all values in column are different

DEFAULT sets the default value of a column

the day tolderak hylotopyth tell offet

CHECK it can limit the values allowed in a

ess.

- name VARCHAR (30) NOT NULL
- email VARCHAR (OD) UNIQUE
- following INT DEFAULT O
- Salary INT DEFAULT 25000
- age INT CONSTRAINT CHECK (age >= 13)
- CONSTRAINT age-check CHECK (age = 18 AND city = "Delhi")
- age INT CHECK (age >= 18)
- age INT, CHECK (age >= 18)

Constaniates * Key Constraints -> Keys are special columns in the tables > Primary Key -> Primary key constraint makes a column unique & not null but used only for one column. the later the party to the time to make all the said - It is a column (or set of columns) in a table that uniquely identifies each now (a unique id). - There is only 1 PK & it should be NOT NULL. TOWN COST PRINCE SON MOT MILE 0.9 BURTELL (OD) SHERISON MONEY - CREATE TABLE temp (id INT NOT NULL, PRIMARY KEY (id)); Salasy dat necess 2 wages CALICATAT CHECK (APR 25 (S) e.9. - CREATE TABLE temp (INT PRIMARY KEY (8) 54 080) KD3HD TOT 590

elessente.

- > Foreign Key it and I solvery &
- Foreign Key prevents actions that would destroy links between tables
- A Foreign Key is a column (or set of columns) in a table that refers to the primary key in another table.
- -> FKs can have duplicate & null values.
- -> There can be multiple Fks.

e.g.

- CREATE TABLE temp (temp-id INT PRIMARY KEY, cust-id INT,

FOREIGN KEY (cust-id) REFERENCES temp1 (Id)

The Martin May - deligation

);

id is primary key in temps table

```
* Visualise Tables (EER Diagram)
→ Database menu > Reverse Engineer = Next, Next, L., Execute
         sould especial wall social
  29
  I A FOLGEST May be a solution of the last
- CREATE TABLE teacher (
     teach-id INT PRIMARY KEY,
     name VARCHAR (30)
on Di lun a stadiote a aul il an
  CREATE TABLE student (
     id INT PRIMARY KEY,
     sollno INT,
     name VARCHAR (30),
    teacher-id INT,
    FORETON KEY (teacher-id) REFERENCES teacher (teach-id)
  );
               THE REST NO. 1711
EER Diagram:
        teacher V
         teach-id
         name VARCHARGOD
          student 🔻
          id INT
          foll no INT
          name VAR (HAR (30)
         teacher id INT
```

- * Insert data into Table
- INSERT INTO USER

 (id. age, name, email, followers, followings)

 VALUES

 (1, 14, "abc", "abc@email.com", 105, 10),

 (5, 12, "xyz", "xyz@email.com", 120, 50);

WHERE SMALLSHES

ELECT + FROM DELL

3 Openators in Wene Clause

- Companicon Operators

- + Select Command & Manual & Sastal
- selects & shows data from the DB.

 syntax: SELECT col1, col2 FROM table-name;
 - syntax (to show all): SELECT * FROM table-name;
- ⇒ DISTINCT keyword used to show distinct values from duplicate values from a table column
- SELECT DISTINCT age FROM user;

output: If age column has 14, 15, 15, 14, 16 values than output will be:

14

15

16

* Where clause star will stat to the s to define some conditions Cleatenth prototol liones, swam, ou his syntax: Car Car Con Constant Sale , Sale 142 (1) - SELECT col1, col2 FROM table-name WHERE conditions; eg SELECT * FROM USER WHERE FOllowers >= 2000; 1000 1000 eg SELECT name followers FROM uses WHERE followers >= 200; symple about Mary & TOTIES & Cup was at a volage * Operators in Where Clause S DICTING POPULATION INCOME -> Arithmetic Operators + addition - subtraction DATE TO TO TO THE * multiplication 1 division % modulus → Companison Operators =, != , >, >= , < , <=

-> Logical Operators

AND, OR, NOT, IN, BETWEEN, ALL, LIKE, ANY

Many and Amer 137, 32 Dec

Coloniania Tablas

FROM WEEK

; S TIMIN

- Bitwise Operators

& Bitwise AND

1 Bitwise or

* Logical Operators

AND to check for both conditions to be true of the conditions to be true BETWEEN selects for a given range

IN matches any value in the list

NOT to negate the given condition
e.g. SELECT name, age

FROM user

WHERE age > 18 AND followers > 200;

eg select name, age FROM usen

WHERE age BETWEEN 15 AND 17;

15,16,17

eg SELECT name, age, email

FROM user:
WHERE email IN ("abc@email.com", "del@email.com");

CNOT to negate This condition

* Limit clause Logical operators -> sets an upper limit on number of (tuples) nows to be returned - SELECT col1, col2 a Situation Operators FROM table-name LIMIT number; Bifusice AND eg select name, age, email FROM USER LIMIT 2; & Legical Operators eg SELECT * FROM user WHERE age > 14 LIMIT 3; Call Best Charas Charles Clark Steppen as The * Order by clause -> To sort in ascending (ASC) or descending (DESC) order - SELECT col1, col2 FROM MISS IN FROM table-name ORDER BY col-name(s) ASC; eg select * From user ORDER BY followess DESC; These, if do not with anything then by default It is asc.

FREM LLES

TO HOUSE TO 3132 -

FROM table name

HAVING CONJUGERS

- Angulassa II account des des

- * Aggregate Functions
- on a set of values, and returns a single value.
 - COUNT ()
 - MAX ()
 - MEN()
 - SUM ()
 - AVGO

eg SELECT max (followers) From user;

eg SELECT count (age) FROM USER
WHERE age > 18;

- * Group By Clause Comments va 911540
- into summary rows.
- The collects data from multiple records and groups the result by one or more column.
- SELECT col1, col2 From table-name GROUP BY col-name (s);
- > Generally, we use group by clause with some aggregation function.

eg select age, count (id) FROM USER Agantary Salagaraph = GROUP BY age; Sylen Shrin output: 14 2 15 3 (A major ~ 16 2 17 1 Charles -() m(/-a -* Having Clause SELLECT THAN (Allowers) FROM USON: -> Similar to where; i.e. applies some conditions on sows, But it is used when we want to apply any conditions after grouping. - SELECT col1, col2 FROM table-name GROUP BY col-name(s) suil (8 quoin) HAVING condition; -> WHERE is for the table, HAVING is for a group -> Grouping is necessary for HAVING. e.g. SELECT age, max (followers) FROM WER GROUP BY age HAVING max (followers) > 200;

MATERIAL VALUE OF THE PARTY OF

- * General Order
- SELECT column (s)

 FROM table-name

 WHERE condition

 GROUP BY column (s)

 HAVING condition

 ORDER BY column (s) DESC;
- * Update Table Pows
- > 70 update existing ROWS
- UPDATE table-name

 SET coll = vall, col2 = val2

 WHERE condition;
 - e.g. UPDATE USER SET followers = 600 WHERE age > 18;
- The sal gives an error when above code executes, then run following line of code:

SET SQL-SAFE- UPDATES = 0;

- * Delete Table Rows
- -> To delete existing nows
- DELETE FROM table-name
 WHERE condition;
 - WHERE age = 14;
- * Alter Table Queries
- → To change the schema
- => ADD Column
- ALTER TABLE table-name
 ADD COLUMN column-name datatype constraint;
- > DROP Column
- ALTER TABLE table_name;

 DROP COLUMN column-name;
- =) RENAME Table
- ALTER TABLE table-name;

- => CHANGE COLUMN (RENAME)
 - ALTER TABLE table-name CHANGE COLUMN old-name new-name new-datatype newconstraint,

OF DETECT FROM MAKE

& Alter Table Quegles

MANUAL COLUMN

Troubles gasar a

- > MODIFY column (modify datatype / constraint)
- ALTER TABLE table-name MODIFY col-name new-databype new-constraint;

- * Truncate Table Query
- > To delete table data

the first the second property of the second

- TRUNCATE TABLE table-name;

* SQL in Terminal MATAN MOIR & Jerry

of SQL Workberch software.

In Terminal

- HPQDP MINGW64 ~ Desktopl demol SQL-CLASS
\$ /USR/local/mysql/bin/mysql - U ROOT - P

**TIT This command gives an excost them

\$ /c/"Program Files"/MysQL/"MysQLsower 8.0'/bin/mysql -4 800t -p
Enter password: mysql@12345
Welcome to the MysQL monitor. Commands end with
jor \q.
Your sQL Connection id is 42.

Server version: 8.0.34 MYSQL Community Server + GPL

Copyright (c) 2000, 2023, Oracle and/ox its affiliates.

Oracle is a

Type "help;" or "h" for help. Type "\c" to clear the current input statement.

- mysgl>

- mysal > SHOW DATABASES; Database 11 1/1 1/1 1/1 information_schema insta mysgl pestosmance_schema sakila lains d sys - would see house had not a some state of 7 sows in set (0.00 sec) hand walled trade some and some some state of - mysgl > CREATE DATABASE delta-app; - mysgl > USE delta-app; Database changed - mysgl > CREATE TABLE temp (id INT PRIMARY KEY); - mysql > SHOW TABLES; Tables_in_delta_app_ 1 80W in set (0.00 sec) (Josym

* Source .sql file

In demolsal-class/schema.sql

- CREATE TABLE USER (

id varihar (50) PROMARY KEY,
USERNAME VARIHAR (50) UNIQUE,
email varihar (50) UNIQUE NOT NULL,
password varihar (50) NOT NULL

In Terminal

- mysql > source schema.sql; Query OK, o rows affected (0.01 sec)

- mysql > SHOW TABLES;

);

Tables_in_delta_app__ temp

2 rows in set (0.00 sec)

2) xabor about I -

SQL with Node

Lineary Alland

- * Faker Package (@faker-js1faker)
- > 76 generate fake data

Note: From this page forward in examples, we are assuming that we have two data tables (temp k. user) in database (delta_app) as per page no. 112 k 119.

CO AT A STATE OF THE STATE OF T

O her in this selection of the special series

() dubility in step, though to be 2 to 3

In Terminal

- HP@DP MINGWEG ~ | Desktop | demo | SQL_CLASS | & npm init -y
- \$ npm 1 @faker-js/faker

```
SQL with Node
In SQL-CLASS/index.js

- const & faker 3 = require ("@faker.js/faker");
  let getRandomUser = () => {
     return {
  userId: faker string unid (),
  username: faker. internet. username ().
       email: faker internet email(),
       avatar: faker image avator (),
       password: faker internet password (),
       bisthdate: faker . date . bisthdate (),
     registeredAt: faker date past ()
  3; 22/12 182 Januar Lacytest In ABLUSTIAN 900 AN -
                             1 April Wit -1
  console. log (getRandomUser ());
           In Terminal
- $ node index.js
   usered: 3a3b047 c-97 c1-abfo-8046-50347 ac 6815c',
   username: 'RORY, Johns',
   email: Jason - Braku 17 @ hotmail. com!,
   avatar: "https://avators.githubusercontent.com/u/18632780,
   password: 'dvHQVbdolaws3Va',
   bisthdate: 1947-09-30 T21: 21: 27. 3 182",
   RegisteredAt: 2023 - 02-23T 12: 17: 23.713Z
```

```
* My SQL Package (mysql2)
3 To connect Node with Mysal
             In Terminal
- HP@DP MINGW64 N/ Desktopp/demolsQL-CLASS
  $ nom i mysgl2
                           ( catch ( catch)
             In SQL-CLASS/index.js
- const mysql = sequise ("mysql2");
  const connection = mysql. create Connection ( {
      host: "localhost",
      user: "Root",
      password: " mysql@ 12345",
     database: "delta_app"
  3); 1 co. 15 L. d. sain seen with all shall
```

let 9 = "SHOW TABLES";

dus sand ways France To an and

Constant they borrow tuning with

Mysal Rockage (meglz) 3 yest connection. query (q, (ess, result) =) { if (ear) throw ear; console.log (sesult); 4 11411 1 mysel2 3 catch (esh) { console. log (ess); connection. end (); cond connection swysel. the warm In Terminal: - \$ node index.js [{ Tables-in-delta-app: 'temp' }, in { Tables - in - delta - app: 'uses' }] It we lad only one table then \$ 1 output will be [1 3] "If we didn't write connection.end() then cussos won't appear here but after output (cursor will be there). =) In REST (Neb app), we need phave connection with database, so need to end when we working with express.

* INSERT into TABLE (single ROW) In index.js table in database - let 9 = "INSERT INTO user (id, username, email, password) YALUES (?, ?, ?, ?)"; let user = ["123", "123-rewuser", "abc@gmail.com", "abc"]; try 2 connection. quest (9, user, (ess, result) =) { if (en) throw err; console. (og (sesuit); 3); & catch (est) { console. log (ess); In Terminal - \$ node index.js Result Set Header { field Count: 0, affected ROWS: 1, insested: 0, info: " " serverstatus: 2, warning Status: 0, changedlows: 0

```
* INSERT into Table (multiple rows)
                                                                                  In index.js
- let wers = Com 11) roza man manint
                      ["1236", "123_newwesb", "abc@gmail.comb", "abcb"],
                     ["123c", "123-newwesc", "abc@gmil.comc", "abcc"]
           I to the terminal of the second of the secon
                                                                                                                                    Etable of database
           let 9=" INSERT INTO USER (id, username, enail, password)
          VALUES ?";
           connection. query (9; [users], (ers, result) =) &
            3);
                                                                                                                                          ( Dryson ) red . Starred
```

(Class) Joseph L

(CARE) sal . Il man

LOWER TO NOT

127

```
* INSERT into Table (Bulk data using faker)
               In index is
- const { faker } = sequise ( " @ faker-js / faker");
 const mysql = require ("mysql2");
 const connection = mysql. create Connection ( }
      host: "localhost",
      uses: " root".
      password: "mysgl@12345",
      database: " delta-app"
 4);
 let getRandom User = () => 2
   setuan [
      takes. datatype. unid (),
      faker datat internet wername (),
      faker internet email (),
      faker internet password ()
   1;
3;
let data = [];
for (let i=1, i <= 100, i++) {
     data. push (get Random User ());
```

rake in database let 9 = "INSERT DINTO user (id, username; email, password)

VALUES?"; tay {
connection.query(q,[data],(err, result)=){ if (ess.) throw ess.; and connection = m. console. log(sesult);

3);

3 catel (esg) { console. log (ess); connection. end (); () In Terminal - \$ node index.js Result SetHeader { field Count: 0, affected Rows: 100, insertId:0, into: Records: 100, Duplicates: 0, Warnings: 0', serverstatus: 2, warningstatus: 0, changed Rows: 0

show bla -

CRUD Operations (SQL + Ex. + Node)

> sal database = we have database = "data-app" table = "user" Total Records = 103

-> packages = express,

uuid,

takes (optional if we want to add bulk take data),

method-overside,

mysql2

- Home Poute

GET / show no. of users in DB home. egs

-) Show Route

GET /user show all users data users.ejs

-) Edit Route

GET /user/:id/edit edit form edit ejs

PATCH / User/: id update in database

- Delete Poute

GET /user/:id/delete delete form delete.ejs

DELETE /user/: id delete record in DB

CRUD OPERALIONS (SOLL + Cx. + Hode) - Add Route as which " a sindefine avel all a sindefine 1820 GET /user/new new user addition new ejs form POST / user new user data store in DB when I have so the see you have the add a like feel shipped-ballem URL => Home - Home Poute Show all Users Edit RECORD Useas DET Delete Record Edit edited RECORD submit DELETE Delete Tin submit stor the New REORD New Mew Water Trans PAICH THESPIT HORETE IN Astabonie Dilete Poute GET / LESS/11/ Lelston 10/06 (One BE OF LOWER PARTY OF BUILDING STEELS

All ".ejs" files of "views" folder < ' poetres = Northead > > home ejs DOCTYPE LIMI <title> Home Page </title> Whead? 2 body> (5/1/2) < h27 Total Numbers of Users are: <1.= count 1.7 </h2> (form action = "/user" method = "get"> (button) click me to see All ugess </button> 4/form> </ html > CU12 61 (343) CHAS GROBEL CARS CH > LICERCHANDOLIKADS CLERLY Capo feet Coonet used of users) to obs Starry was was sitely show a war and to did

CLAND THE WARRENCE WARRENCE AND THE CALLS

```
=) users. ejs who "envir" to estil ejo. " Il ;
    <! DOCTYPE html>
        < title> All Usess </ title>
        < style>
           table, th, td &
              border: 1px solid black;
       4/style>
(head)
    < body >
       < form action="/user/new" method="get">
 < button > Add New User </button>
       </508m>
       <h2> List of all users </h2>
       < tables
        (th)
          4th) Id 4/th>
          4th> Email 4/th>
          > Username 
        (/ta)
        (%) for (const user of users) { ·1.>
              (42)
                 <!- user.id 1.7 </td>
                 >
```

```
< form action= "user/e1= werid x. >/edit"
                  method = "get" >
                 < button> Edit username </button>
    4 form>
                            ( D35 N )
          1/td>
                            X 10-42
          >
  < form action="/user/e/= user.ld 1.7/delete"
                method = "get">
                < button > Delete Record </button >
    </form>
         1/td>
      </ta>
 <1. 3 1.>
  Calleto Richies Tallanda Sala
</ body >
           K nother School of the Killian S
</ html >
                    - CAMB 1 12
                         se chief he
                           K hated 32
```

```
=) new es
                     (11)
       < form a line handers with a selection
      <! DOCTYPE html)
Chattans all continues
         <title> Add New User </title>
      1/head>
                             CLFLS
      < body>
 <form action = "/user" method = "prost">
          <input name = "email"
 type = "email"
                placeholder="enter email">
          < input name = "wername"
                   type = "text"
                   placeholder = " enter username">
          <input name = "password"</pre>
                   type = " password"
                   placeholder="enter password")
          < button > submit </button>
       </ form >
      </ body >
```

</html>

```
a edit. ejs
                           fortal allegant li
 LIDOCTYPE html>
    (title) Edit Page (/title)
                                CAMPIA DE
 1 head >
                                    KARDI >
 2 body >
     < h2> You are about to edit this user: <1= user. eneil 1.>
     </k2>
     < form action="/usex/</= usex.id +>? = method = PATCH"
          method = "post">
        ctextarea name "username"
            (.1. = user username 1.)
        (/textasea)
        < input name="password"
    type = "password"
           placeholder="enter password">
         <button> submit </button>
                                   ( boltz
    4/form>
                                   < ladd's
 4/body>
 Vhtml>
```

```
=> delete. ejs
                   STOOTS HANDS
  ! DOCTYPE html
   <title> Delete Page </title>
 4head?
                           Khonits
 < body>
                            s pend s
    < h2> You are about to Delete this uses:
      < != user . username 1.> with email:
      <1.= user. email 1.>
    </h2>
    < form action="/usex/</= usex.id 1.>?-method=DELETE"
         method = "post" >
     <input name="password"
           type = "password"
            placeholder="enter password")
   </toban>
 1/body>
                          Z/10000 20
 </html>
```

```
"index.js" is in "SQL-CLASS" folder
                                       and her Cold to the second of the contract of the second o
of index is
      const mysql = require (" mysql2");
      const express = sequire ( express");
       const app = service express();
       const path = require ("path");
       const methodo verride = require ("method-override");
       const { v4: unidv4} = sequire("unid");
      app. set ("view engine", "ejs");
      app. set ("views", path. join ( -- dismame, "/views"));
      app. use (express. unlencoded (textended: true }));
      app. use ( method overside ( "_method") );
     app. listen ("8020", () => {
console. log ("Seaver is listening on post 8080");
      Di
      const connection = mysql. create Connection ( }
                            host: localhost; Tolle = 1 331
                            usea: "root".
                         password: "mysql@12345",
                            database: " delta-app"
       3);
                                                                                                               P.TO --
```

// Home Route & SALD 193" M at 19- value app. get ("/", (reg, res) =) } · table mana let q = ' SELECT count (+) FROM user ; New of Destinoyant Sandows, 320'es tay & Description of the state connection. quesy (q, (ess, result) =) { if (ess) throw ess; let count = result [0] ["count (*)"]; nes. render ("home.ejs", 1 count 3); 3 catch (exa) { console. (og (ess); ses. send (" something wrong"); The state of the s 11 Show Route app. get ("/uses", (seg, ses) =) } let ? = 'SELECT * FROM user'; tay 1 connection. query (2, (ess, users) =) { if (est) throw est; res. render ("users. ejs", ¿users }); } catch(err) { console. log (ess); ses . send i "something went wrong"); 3);

```
// Edit Poute suite state state
app. get ("/user/:id/edit", (reg, res) = {
  let { id } = 8 eq. params;
                  Vet & our round & designation ( 13)
  let 9 = SELECT * FROM USER WHERE id="${id}"
 The fat the sound was been been a followed by the
  try ?
     connection. query (q, (ess, result) =) {
          if (ess) throw ers;
          let user = result [0];
          res. Render ("edit.ejs", { user });
      3);
  } catch (ess.) {
  res. send ("Something went wrong");
Et PI UPPATE WICE SET ; [
som en till man i 3 " = tilbra bek si R
 17 TEALS TELL BRIEFIN . P.T.O. -
```

Dinne and the large mais some

(fice thought (mas)

```
// Edit Poute - update database
app. patch ("/uses1:id", (seq, ses) =) {
                  let { id } = req. params;
                  let & password: torm Pass, username: new Username } = 809. body
                 let 9 = SELECT * FROM USER WHERE id="$ { id 3" },
                tay ?
    Commeching association ( as a security)
                          connection. query (2, (ess, result) => {
                                         if (est) throw ess;
                                    il de la company de la company
let user = result [0];
                                     It (form Pass != user. password) {
                                                ses send ("WRONG Password");
                        3 else {
                                                     let 92= LIPDATE USER SET
                                                                                       username = "$ { new Username }"
                                                                                      WHERE Id = "$ { id }" );
                                                   connection. query ( 92, (err, result) => {
                                                                     if (ess.) throw ess;
                                                                     res. redirect("/user");
                                                  3);
            } catch(err) {
```

```
console log (est.);
          Res. send ("something went wrong");
3);
// Delete Route
      let the Serect is FROM upen Where
app. get ("/user/: id/delete", (Req, res) > {
    let { id } = Req. params;
    let 9 = "SELECT * FROM uses WHERE id="$ 2id3";
    try {
       connection. query ( &, (ess, result) =) }
           if (ess) throw ess;
   let user = result[0];
          ses, render ("delete.ejs", {user });
    } catch (est) {
             o word (not)
        console. log (ess);
        res. send ("Something went wrong");
    4
3);
```

```
// Delete Route - update database
    app. delete ("/user/:id", (Reg, Res) => {
       let { id } = req. params;
let { password: formPass } = req. body;
                                     stuck stilled in
       let q = 'SELECT * FROM USER WHERE id="stid;";
          connection. query (2, (ess, sesult) =) {
if (ess) throw ess)
           let user = result[0];
      if (formpass != wer. password) {
               res. send ("WRONG Password");
            } else {
     let 92 = DELETE FROM USER WHERE

id = "$ {id}" ;
              connection. query ( 92, (ess, result) = {
                   if (ess) throw ess;
       3);

3 catch (ess.) & console. (og (ess.);

ses. send("something went wkong");

3
```

```
11 Add new Record Poute
app. get ("/uses / new", ( neg, nes) => }
res. sender ("new.ejs");
11 add new record Route - add in database
app. post("/user", (seq, ses) => }
   let 2 username, email, password 3 = req. body;
let id = uuidv4();
   let 9= INSERT INTO user (id, username, email, password) VALUES ("$ Lid3", "$ { username }",
            "$ { email }", "$ { password}");
   try {
      connection. query (9, (ess, sesult) => 1
         if (ess) throw ess;
      nes. redirect ( */user");
   } catch (ess) {
        console.log (est);
res.send("something went wrong");
3);
```

* start the server - HP@DP MINGW64 ~/ Desktop/demo/SQL_CLASS & nodemon index js App is listening on post 2020 I add new second route - add in solatore