

**w17-p1: create user\_xx in postgresQL server, enter two data, the first one is about your**

Materialized Views

Operators

Procedures

1.3 Sequences

Tables (7)

> book\_58

> cart\_58

> category\_58

> category\_xx

> shop\_58

> shop\_xx

> user\_58

> Columns (3)

user\_id

name

email

> Constraints

> shop\_58

> shop\_xx

> user\_58

> Columns (3)

user\_id

name

email

> Constraints

> Indexes

> RLS Policies

> Rules

Query Editor

Query History

```
1 INSERT INTO user_58(user_id,name,email)
2 VALUES
3 (1,'TingAN','409411658@gms.tku.edu.tw'),
4 (2,'john','john@gmail.com')
5 /*
6 CREATE TABLE user_58 (
7     user_id int,
8     name varchar(255),
9     email varchar(255),
10     PRIMARY KEY (user_id)
11 )
```

Data Output

Explain

Messages

Notifications

INSERT 0 2

Query returned successfully in 54 msec.

Data Output

Explain

Messages

Notifications

	user_id [PK] integer	name character varying (255)	email character varying (255)
1	1	TingAN	409411658@gms.tku.edu.tw
2	2	john	john@gmail.com

**w17-p2: create cart\_xx in postgresQL server, enter five data, one for each category**

Procedures

1.3 Sequences

Tables (8)

Cart\_58

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

cart\_58\_product\_id\_fkey

cart\_58\_user\_id\_fkey

Indexes

RLS Policies

Rules

Triggers

category\_58

category\_xx

shop\_58

Query Editor

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

INSERT INTO

VALUES

(1,1,6,1),

(2,1,15,2),

(3,1,20,1),

(4,1,28,3),

(5,1,35,2)

/\*

CREATE TABLE cart\_58(

cart\_id int,

user\_id int REFERENCES user\_58 ON DELETE SET NULL ON UPDATE CASCADE,

product\_id int REFERENCES shop\_58 ON DELETE SET NULL ON UPDATE CASCADE,

amount int,

PRIMARY KEY(cart\_id)

)

\*/

Data Output

Explain

Messages

Notifications

INSERT 0 5

Query returned successfully in 44 msec.

1.3 Sequences

Tables (8)

Cart\_58

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

cart\_58\_product\_id\_fkey

cart\_58\_user\_id\_fkey

Indexes

RLS Policies

Data Output

Explain

Messages

Notifications

	cart_id [PK] integer	user_id integer	product_id integer	amount integer
1	1	1	6	1
2	2	1	15	2
3	3	1	20	1
4	4	1	28	3
5	5	1	35	2

w17-p3: give query to get related data for cart info shown in ejs

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

cart\_58\_product\_id\_fkey

cart\_58\_user\_id\_fkey

Indexes

RLS Policies

Rules

Triggers

category\_58

category\_xx

shop\_58

Columns (6)

id

name

cat\_id

price

remote\_url

local\_url

Constraints

Indexes

Query Editor

```

1 SELECT cart_id,U.name as user_name,S.name as product_name,price,amount
2 FROM cart_58 as C,user_58 as U,shop_58 as S
3 WHERE C.user_id=U.user_id AND C.product_id=S.id

```

Data Output

	cart_id	user_name	product_name	price	amount
	integer	character varying (255)	character varying (45)	real	integer
1	1	TingAN	Brown Brim	25	1
2	2	TingAN	Black Jean Shearling	125	2
3	3	TingAN	Adidas NMD	220	1
4	4	TingAN	Blue Tanktop	25	3
5	5	TingAN	Camo Down Vest	325	2

## w17-p4: insert data in order to violate the key constraints, foreign key constraints

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

cart\_58\_product\_id\_fkey

cart\_58\_user\_id\_fkey

Indexes

RLS Policies

Rules

Triggers

category\_58

category\_xx

shop\_58

Columns (6)

id

name

Query Editor

```

1 INSERT INTO cart_58(cart_id,user_id,product_id,amount)
2 VALUES
3 (1,3,6,1)
4
5 /*
6 CREATE TABLE cart_58(
7     cart_id int,
8     user_id int REFERENCES user_58 ON DELETE SET NULL ON UPDATE CASCADE,
9     product_id int REFERENCES shop_58 ON DELETE SET NULL ON UPDATE CASCADE,
10    amount int,
11    PRIMARY KEY(cart_id)
12 )
13 */

```

Data Output

ERROR: duplicate key value violates unique constraint "cart\_58\_pkey1"

DETAIL: Key (cart\_id)=(1) already exists.

SQL state: 23505

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

cart\_58\_product\_id\_fkey

cart\_58\_user\_id\_fkey

Indexes

RLS Policies

Rules

Triggers

category\_58

category\_xx

shop\_58

Columns (6)

id

Query Editor

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

INSERT INTO

VALUES

(6,3,6,1)

/\*

CREATE TABLE cart\_58(

cart\_id int,

user\_id int REFERENCES user\_58 ON DELETE SET NULL ON UPDATE CASCADE,

product\_id int REFERENCES shop\_58 ON DELETE SET NULL ON UPDATE CASCADE,

amount int,

PRIMARY KEY(cart\_id)

)

/\*

Data Output

Explain

Messages

Notifications

ERROR: insert or update on table "cart\_58" violates foreign key constraint "cart\_58\_user\_id\_fkey"

DETAIL: Key (user\_id)=(3) is not present in table "user\_58".

SQL state: 23503

Query Editor

Query History

1

2

3

4

5

6

INSERT INTO

VALUES

(6,5,5,1)

/\*

CREATE TABLE cart\_58(

Data Output

Explain

Messages

Notifications

ERROR: insert or update on table "cart\_58" violates foreign key constraint "cart\_58\_product\_id\_fkey"

DETAIL: Key (product\_id)=(5) is not present in table "shop\_58".

SQL state: 23503

## w17-p5.1: test ON UPDATE CASCADE

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (8)

Cart\_58

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Query Editor

Query History

1

2

SELECT \*

FROM public.user\_58

ORDER BY user\_id ASC

Data Output

Explain

Messages

Notifications

	user_id [PK] integer	name character varying (255)	email character varying (255)
1	1	TingAN	409411658@gms.tku.edu.tw
2	5	john	john@gmail.com

Functions

Materialized Views

Operators

Procedures

1.3 Sequences

Tables (8)

Cart\_58

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

2 ORDER BY cart\_id ASC

Data Output

Explain

Messages

Notifications

	cart_id [PK] integer	user_id integer	product_id integer	amount integer
1	1	1	6	1
2	2	1	15	2
3	3	1	20	1
4	4	1	28	3
5	5	1	35	2
6	7	5	10	1
7	8	5	20	2

## w17-p5.2: test ON DELETE SET NULL

Foreign Tables

Functions

Materialized Views

Operators

Procedures

1.3 Sequences

Tables (8)

Cart\_58

book\_58

cart\_58

Columns (4)

cart\_id

user\_id

product\_id

amount

Constraints (3)

cart\_58\_pkey1

cart\_58\_product\_

1 SELECT \* FROM public.cart\_58

2 ORDER BY cart\_id ASC

Data Output

Explain

Messages

Notifications

	cart_id [PK] integer	user_id integer	product_id integer	amount integer
1	1	1	6	1
2	2	1	15	2
3	3	1	20	1
4	4	1	28	3
5	5	1	35	2
6	7	[null]	10	1
7	8	[null]	20	2

- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- 1.3 Sequences
- Tables (8)
  - Cart\_58
  - book\_58
  - cart\_58
    - Columns (4)
      - cart\_id

```
2 ORDER BY user_id ASC
```

	user_id [PK] integer	name character varying (255)	email character varying (255)
1	1	TingAn	409411658@gms.tku.edu.tw
2	5	john	john@gmail.com

Data Output
Explain
Messages
Notifications

w17-p6: implement /book\_xx/delete/:id

book\_58.js M

index.ejs M

w17\_58.md

views > book\_58 > index.ejs > html > body > div.container > div.card > div.card-body > table.table > tbody > tr > td > a.btn.btn-danger.delete

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

```

<td>
  <a
    class="btn btn-success edit"
    href="/book_58/edit/<%= data[i].id %>"
    >Edit</a>
  <a
    class="btn btn-danger delete"
    onclick="return alert('Are You sure?')"
    href="/book_58/delete/<%= data[i].id %>"
    >Delete</a>
  </td>
</tr>
</table>

```

book\_58.js M

routes > book\_58.js > router.get('/delete/:id') callback > query

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

```

//delete
router.get('/delete/:id', async(req, res)=>{
  try{
    console.log('delete id', req.params.id);
    const query = {
      text: 'DELETE FROM book_58 WHERE id = $1',
      values: [req.params.id],
    };
    const results= await db.query(query);
    res.redirect('/book_58');
  }catch(err){
    console.log(err);
  }
});

```

Commits · 40941

https://tku365.sh

Books

localhost:3000/book\_58

409411658 TingAn

Books

Add Book

Id	Name	Author	Price	Action
1	JavaScript Primers	Brad Traversy	100	<div>Edit</div> <div>Delete</div>

w17-p7: implement /book\_xx/update

The screenshot shows the VS Code editor with the `index.ejs` file open. The HTML code for the edit button is highlighted in red:

```
class="btn btn-success edit"
href="/book_58/edit/<%= data[i].id %>"
>Edit</a>
```

The browser window shows the page `localhost:3000/book_58` with a table of books. The 'Edit' button for the first book is highlighted in red:

Id	Name	Author	Price	Action
1	JavaScript Primers	Brad Traversy	100	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

The VS Code editor also shows the `book_58.js` file with the `router.get('/edit/:id')` callback function highlighted in red:

```
//edit
router.get('/edit/:id', async (req, res) => {
  const id = req.params.id;
  console.log('id', id);
  try {
    const query = {
      text: `SELECT * FROM book_58 WHERE id = ${id}`,
      values: [id],
    };
    const results = await db.query(query);
    data = results.rows;
    res.render('/book_58/edit_58', {
      id: data[0].id,
      name: data[0].name,
      author: data[0].author,
      price: data[0].price,
    });
  } catch (err) {
    console.log(err);
  }
});
```

The screenshot shows the VS Code editor with the `book_58.js` file open. The `router.get('/update')` callback function is highlighted in red:

```
//update
router.get('/update', async (req, res) => {
  try {
    const query = {
      text: `UPDATE book_58 SET name = ${req.body.name}, author = ${req.body.author}, price = ${req.body.price} WHERE id = ${req.params.id}`,
      values: [req.body.name, req.body.author, req.body.price, req.params.id],
    };
    await db.query(query);
    res.redirect('/book_58');
  } catch (err) {
    console.log(err);
  }
});
```

The browser window shows the page `localhost:3000/book_58` with a form to update a book. The form fields are highlighted in red:

```
<form action="/book_58/update" method="post">
  <div class="form-group">
    <input type="hidden" name="id" value="<%= id %>" autocomplete="off" />
  </div>
  <div class="form-group">
    <label>Name:</label>
    <input type="text" class="form-control" name="name" value="<%= name %>" autocomplete="off" />
  </div>
  <div class="form-group">
    <label>Author:</label>
    <input type="text" class="form-control" name="author" value="<%= author %>" autocomplete="off" />
  </div>
  <div class="form-group">
    <label>Price:</label>
    <input type="text" class="form-control" name="price" value="<%= price %>" autocomplete="off" />
  </div>
  <input type="submit" value="Update" />
</form>
```

## w17-LAST-log

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%" --after="2022-06-13"
ccf0fe1 409411658 Tue Jun 14 19:14:56 2022 +0800 w17-p7: implement /book_xx/update
ff087a5 409411658 Tue Jun 14 18:47:30 2022 +0800 w17-p6: implement /book_xx/delete/:id
c0f2940 409411658 Tue Jun 14 17:25:55 2022 +0800 w17-p5.2: test ON DELETE SET NULL
289af1d 409411658 Tue Jun 14 17:20:35 2022 +0800 w17-p5.1: test ON UPDATE CASCADE
23bdd0d 409411658 Tue Jun 14 17:05:07 2022 +0800 w17-p4: insert data in order to violate the key constraints, foreign key constraints
cfd62cd 409411658 Tue Jun 14 16:40:20 2022 +0800 w17-p3: give query to get related data for cart info shown
8b9ac45 409411658 Tue Jun 14 16:11:17 2022 +0800 w17-p2: create cart_xx in postgresql server, enter five data, one for each category
ac1715a 409411658 Tue Jun 14 15:53:59 2022 +0800 w17-p1: create user_xx in postgresql server, enter two data, the first one is about your
```

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-06-13"
ccf0fe1 409411658      Tue Jun 14 19:14:56 2022 +0800  w17-p7: implement /book_xx/update
ff087a5 409411658      Tue Jun 14 18:47:30 2022 +0800  w17-p6: implement /book_xx/delete/:id
c0f2940 409411658      Tue Jun 14 17:25:55 2022 +0800  w17-p5.2: test ON DELETE SET NULL
289af1d 409411658      Tue Jun 14 17:20:35 2022 +0800  w17-p5.1: test ON UPDATE CASCADE
23bddd0 409411658      Tue Jun 14 17:05:07 2022 +0800  w17-p4: insert data in order to violate the key constrain
cfd62cd 409411658      Tue Jun 14 16:40:20 2022 +0800  w17-p3: give query to get related data for cart info show
8b9ac45 409411658      Tue Jun 14 16:11:17 2022 +0800  w17-p2: create cart_xx in postgresQL server, enter five c
ac1715a 409411658      Tue Jun 14 15:53:59 2022 +0800  w17-p1: create user_xx in postgresQL server, enter two da
```