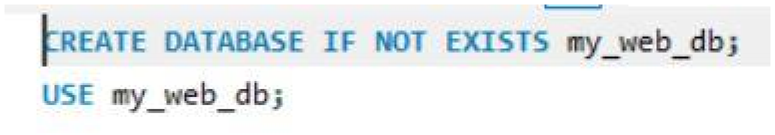
Kristin Pflug

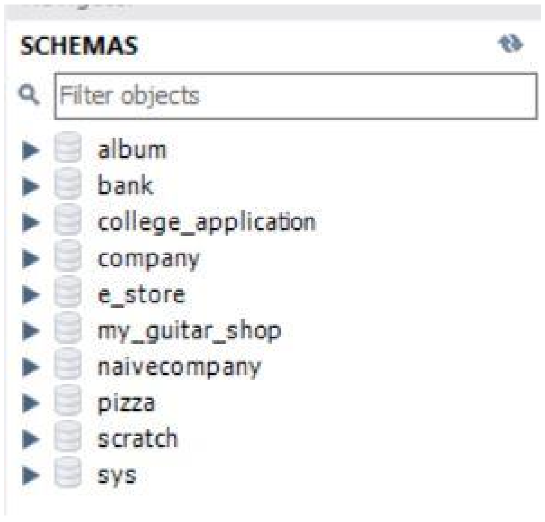
Lab 8: Creating a Database in MySQL

1.) The IF NOT EXISTS keyword verifies if there is a table with the specified name already in the database. If an identical table exists, then the query will not execute.



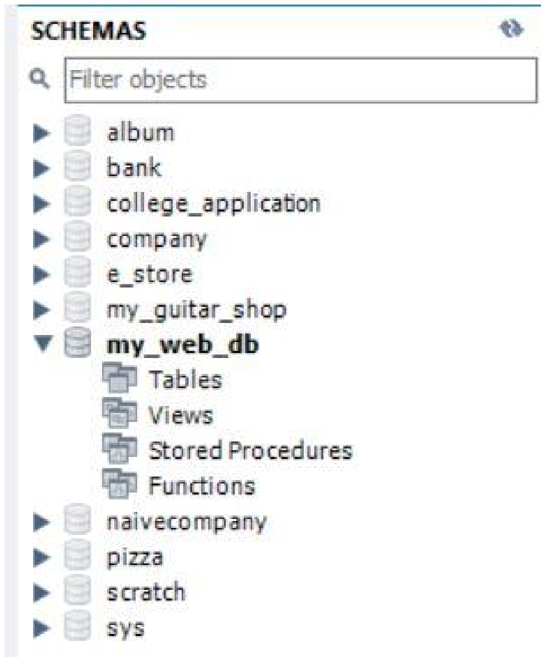
```
CREATE DATABASE IF NOT EXISTS my_web_db;
USE my_web_db;
```

Before running the queries:



```
SCHEMAS
Filter objects
▶ album
▶ bank
▶ college_application
▶ company
▶ e_store
▶ my_guitar_shop
▶ naivecompany
▶ pizza
▶ scratch
▶ sys
```

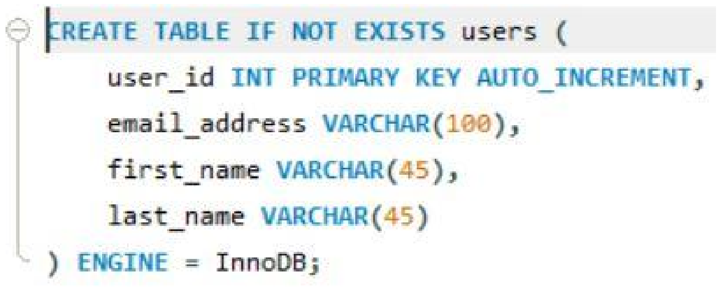
After running the queries:



```
SCHEMAS
Filter objects
▶ album
▶ bank
▶ college_application
▶ company
▶ e_store
▶ my_guitar_shop
▼ my_web_db
    Tables
    Views
    Stored Procedures
    Functions
▶ naivecompany
▶ pizza
▶ scratch
▶ sys
```

2.) CREATE TABLE IF NOT EXISTS users (

```
        user_id       INT        PRIMARY KEY AUTO_INCREMENT,
        email_address  _________________ ,
        first_name    ____________________ ,
        last_name     ____________________
    ) ENGINE = InnoDB;
```



```
CREATE TABLE IF NOT EXISTS users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    email_address VARCHAR(100),
    first_name VARCHAR(45),
    last_name VARCHAR(45)
) ENGINE = InnoDB;
```

Query Results:

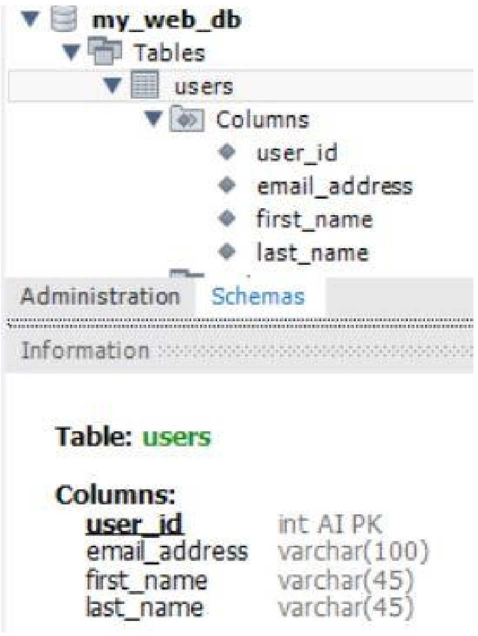
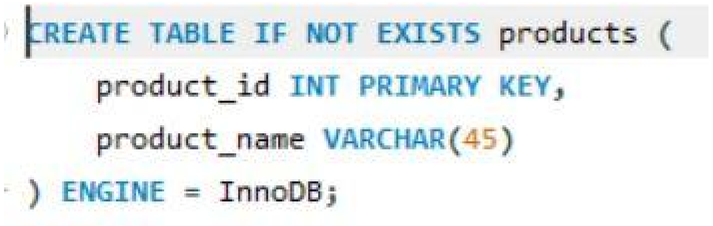


Table: users

Columns:

```
user_id         int AI PK
email_address   varchar(100)
first_name      varchar(45)
last_name       varchar(45)
```

3.)



```
CREATE TABLE IF NOT EXISTS products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(45)
) ENGINE = InnoDB;
```

Query Results:

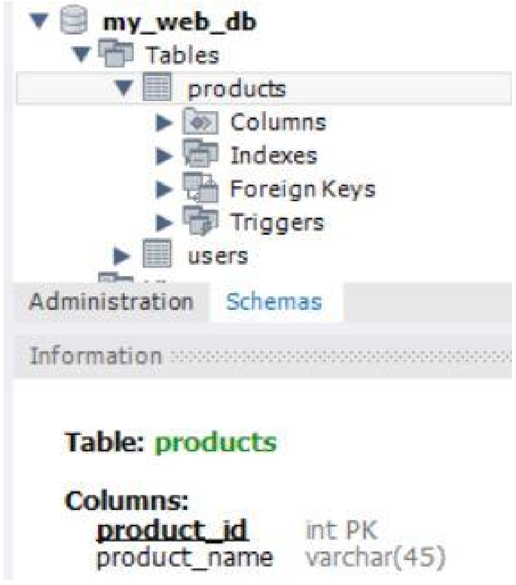
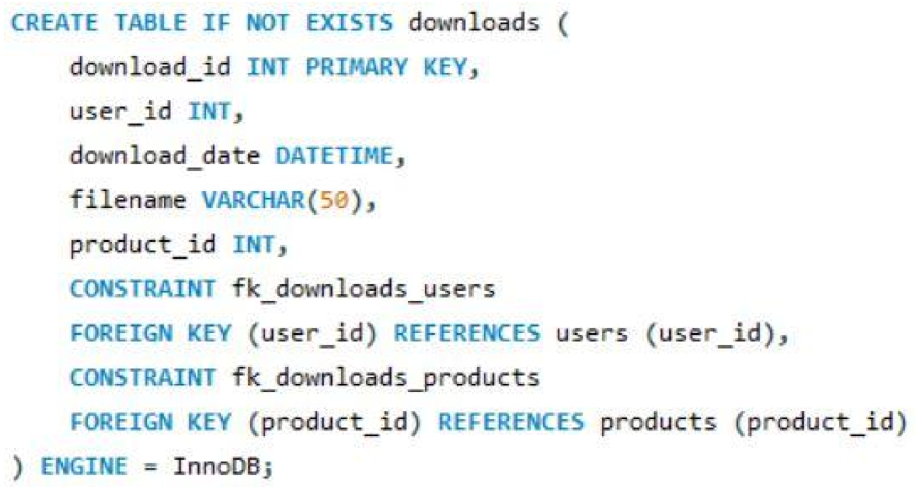


Table: products

Columns:

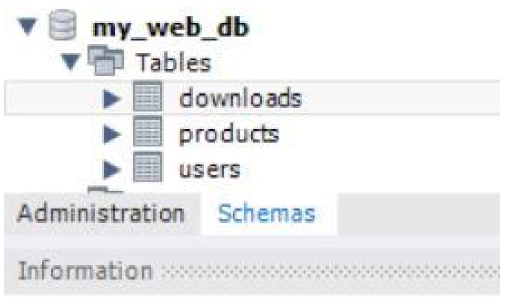
```
product_id     int PK
product_name   varchar(45)
```

```
4.) CREATE TABLE IF NOT EXISTS downloads (
        download_id   INT _____________ ,
        user_id        _________________ ,
        download_date _________________ ,
        filename       _________________ ,
        product_id     _________________ ,

        CONSTRAINT fk_downloads_users
        FOREIGN KEY (_____) REFERENCES _____ (_____),

        CONSTRAINT fk_downloads_products
        FOREIGN KEY (_____) REFERENCES _____ (_____)
  ) ENGINE = InnoDB;
```



```
CREATE TABLE IF NOT EXISTS downloads (
    download_id INT PRIMARY KEY,
    user_id INT,
    download_date DATETIME,
    filename VARCHAR(50),
    product_id INT,
    CONSTRAINT fk_downloads_users
    FOREIGN KEY (user_id) REFERENCES users (user_id),
    CONSTRAINT fk_downloads_products
    FOREIGN KEY (product_id) REFERENCES products (product_id)
) ENGINE = InnoDB;
```

Query Results:



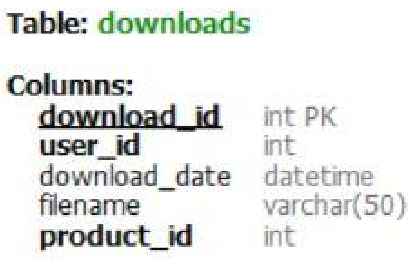
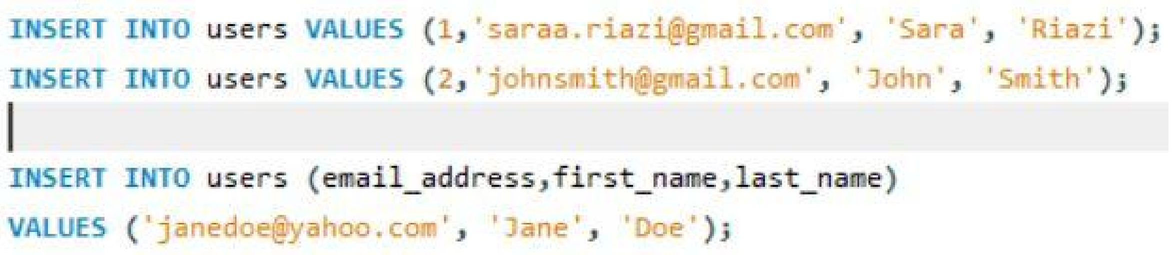


Table: downloads

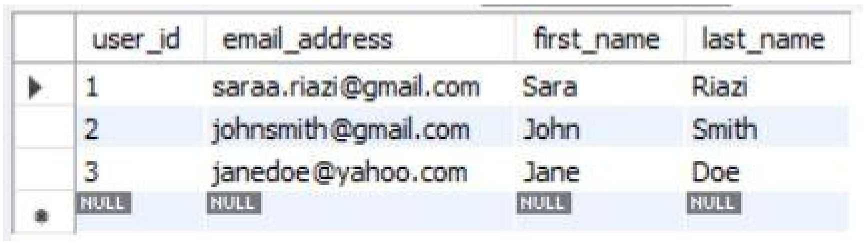
Columns:

download_id	int PK
user_id	int
download_date	datetime
filename	varchar(50)
product_id	int

5.) Jane’s user_id is 3 because the user_id attribute is set to AUTO_INCREMENT, which means that the user_id value will increment by one each time a new row is added to the database. Since the last row in the database before Jane Doe has a user_id of 2, Jane Doe’s user_id is set to 3.



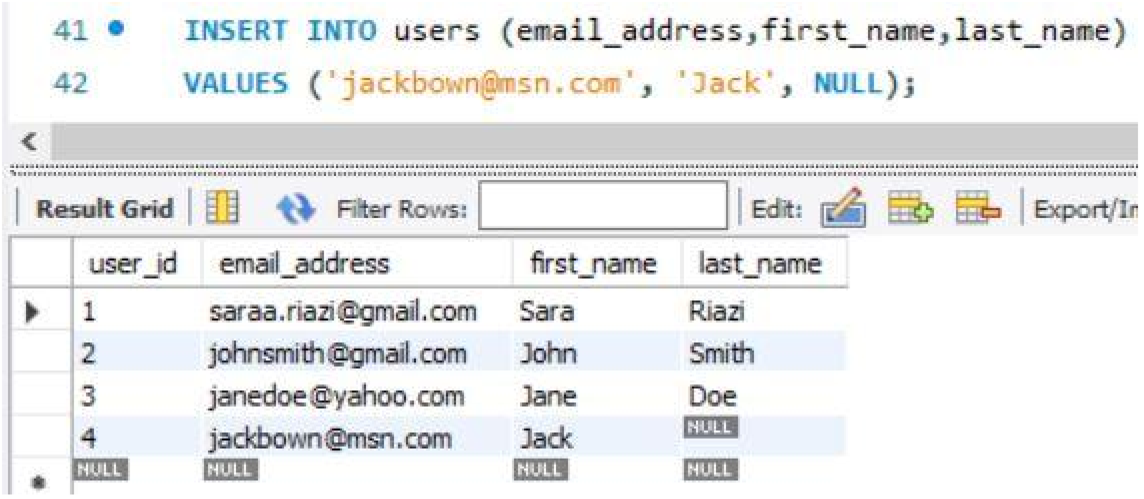
```
INSERT INTO users VALUES (1,'saraa.riazi@gmail.com', 'Sara', 'Riazi');
INSERT INTO users VALUES (2,'johnsmith@gmail.com', 'John', 'Smith');

INSERT INTO users (email_address,first_name,last_name)
VALUES ('janedoe@yahoo.com', 'Jane', 'Doe');
```



	user_id	email_address	first_name	last_name
▶	1	saraa.riazi@gmail.com	Sara	Riazi
	2	johnsmith@gmail.com	John	Smith
	3	janedoe@yahoo.com	Jane	Doe
*	NULL	NULL	NULL	NULL

6.) INSERT INTO users (email_address,first_name,last_name)
VALUES ('jackbown@msn.com', 'Jack', NULL);

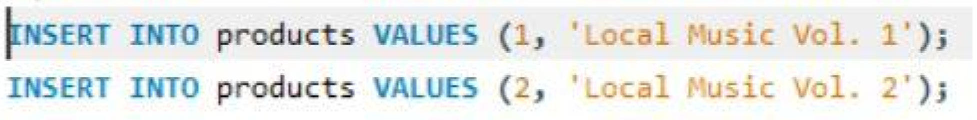
Query Results:



```
INSERT INTO users (email_address,first_name,last_name)
VALUES ('jackbown@msn.com', 'Jack', NULL);
```

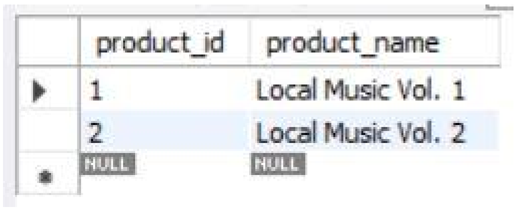
	user_id	email_address	first_name	last_name
▶	1	saraa.riazi@gmail.com	Sara	Riazi
	2	johnsmith@gmail.com	John	Smith
	3	janedoe@yahoo.com	Jane	Doe
	4	jackbown@msn.com	Jack	NULL
*	NULL	NULL	NULL	NULL

7.) INSERT INTO products VALUES (1, 'Local Music Vol. 1');
INSERT INTO products VALUES (2, 'Local Music Vol. 2');



```
INSERT INTO products VALUES (1, 'Local Music Vol. 1');
INSERT INTO products VALUES (2, 'Local Music Vol. 2');
```

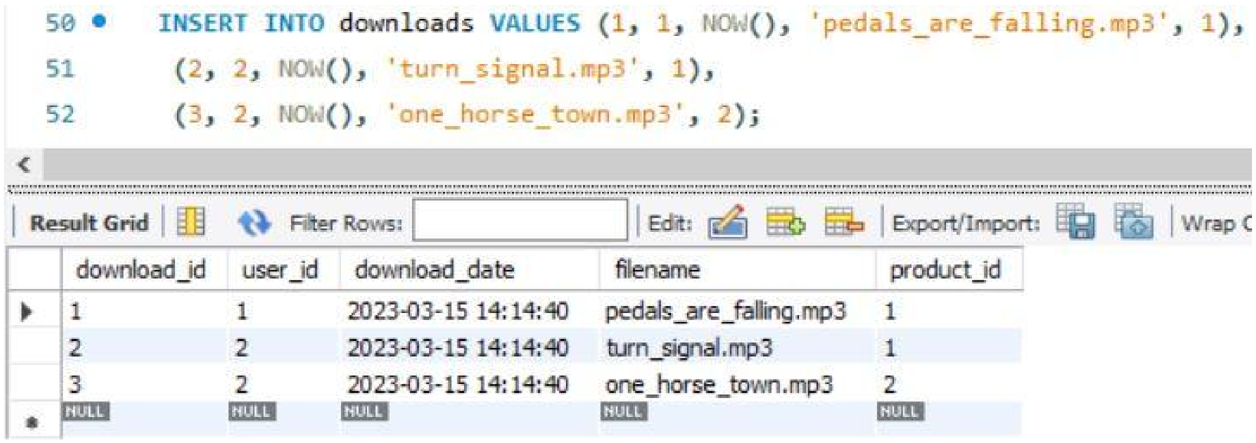
Query Results:



	product_id	product_name
▶	1	Local Music Vol. 1
	2	Local Music Vol. 2
*	NULL	NULL

8.) The NOW() function returns a value equal to the current date and time.

Query Results:

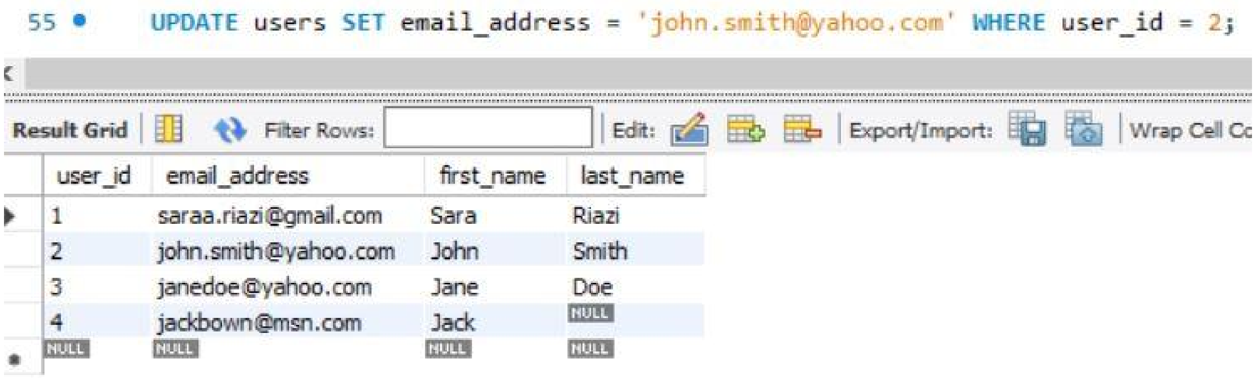


```
INSERT INTO downloads VALUES (1, 1, NOW(), 'pedals_are_falling.mp3', 1),
 (2, 2, NOW(), 'turn_signal.mp3', 1),
 (3, 2, NOW(), 'one_horse_town.mp3', 2);
```

	download_id	user_id	download_date	filename	product_id
▶	1	1	2023-03-15 14:14:40	pedals_are_falling.mp3	1
	2	2	2023-03-15 14:14:40	turn_signal.mp3	1
	3	2	2023-03-15 14:14:40	one_horse_town.mp3	2
*	NULL	NULL	NULL	NULL	NULL

9.) UPDATE users SET email_address = 'john.smith@yahoo.com' WHERE user_id = 2;

Query Results:

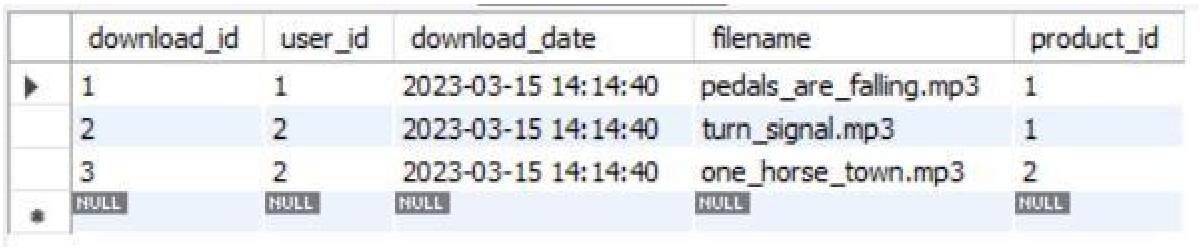


```
UPDATE users SET email_address = 'john.smith@yahoo.com' WHERE user_id = 2;
```

	user_id	email_address	first_name	last_name
▶	1	saraa.riazi@gmail.com	Sara	Riazi
	2	john.smith@yahoo.com	John	Smith
	3	janedoe@yahoo.com	Jane	Doe
	4	jackbown@msn.com	Jack	NULL
*	NULL	NULL	NULL	NULL

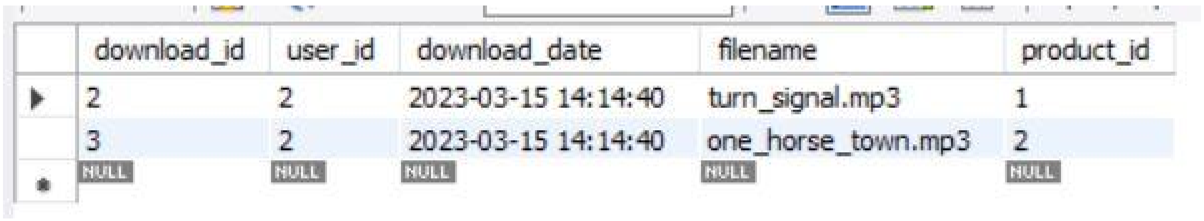
10.) Result of running statement in (1): Error
14:36:07 DELETE FROM users WHERE user_id=1 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`my_web_db`.`downloads`, CONSTRAINT `fk_downloads_users` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`)) 0.000 sec

DELETE FROM downloads WHERE user_id = 1;
Query Results:
Before delete-



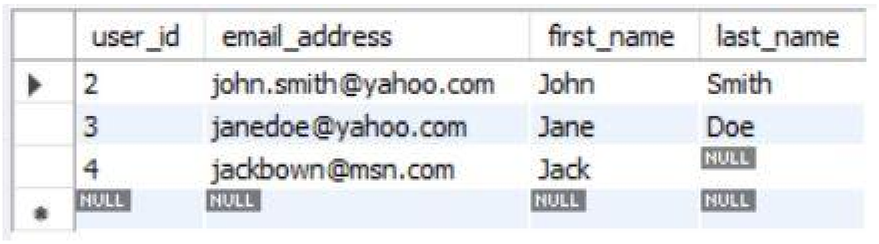
	download_id	user_id	download_date	filename	product_id
▶	1	1	2023-03-15 14:14:40	pedals_are_falling.mp3	1
	2	2	2023-03-15 14:14:40	turn_signal.mp3	1
	3	2	2023-03-15 14:14:40	one_horse_town.mp3	2
*	NULL	NULL	NULL	NULL	NULL

After delete-



	download_id	user_id	download_date	filename	product_id
▶	2	2	2023-03-15 14:14:40	turn_signal.mp3	1
	3	2	2023-03-15 14:14:40	one_horse_town.mp3	2
*	NULL	NULL	NULL	NULL	NULL

Result of running statement in (3):



	user_id	email_address	first_name	last_name
▶	2	john.smith@yahoo.com	John	Smith
	3	janedoe@yahoo.com	Jane	Doe
	4	jackbown@msn.com	Jack	NULL
*	NULL	NULL	NULL	NULL

The DELETE statement throws an error when running it in (1) because user_id is a foreign key in the downloads table. The references to user_id 1 have to be deleted in downloads first before they can be deleted in the users table, otherwise the rows in download will point to a value in the users table that does not exist. This is why running the statement in (3) works while (1) throws an error.