

Lesson 12- (Relationship) one to many

- 1- Real word data is messy
- 2- Types of data relationships
- 3- One to many: the basics
- 4- Working with foreign keys
- 5- Cross join
- 6- Inner join
- 7- Left join
- 8- Right join: part 1
- 9- Right join: part 2
- 10- Right and left join: a common question
- 11- Our first join exercise e
- 12- Our first join exercise solution
- 13- Our first join exercise 2
- 14- Our first join exercise solution 2

Real world data is messy

RELATIONSHIPS AND JOINS

Lesson 12

So Far We've Been
Working With Very
Simple Data

Real World Data Is Messy and Interrelated

Books

Versions

Books

Versions

Genres

Orders

Reviews

Books

Authors

Customers

Versions

Genres

Orders

Reviews

Books

Authors

Customers

Versions

Genres

Orders

Reviews

Books

Authors

Customers

Versions

Genres

Orders

Reviews

Books

Authors

Customers

Versions

Genres

Orders

Reviews

Books

Authors

Customers

Versions

Genres

Orders

Reviews

Books

Authors

Customers

Versions

Genres

Orders

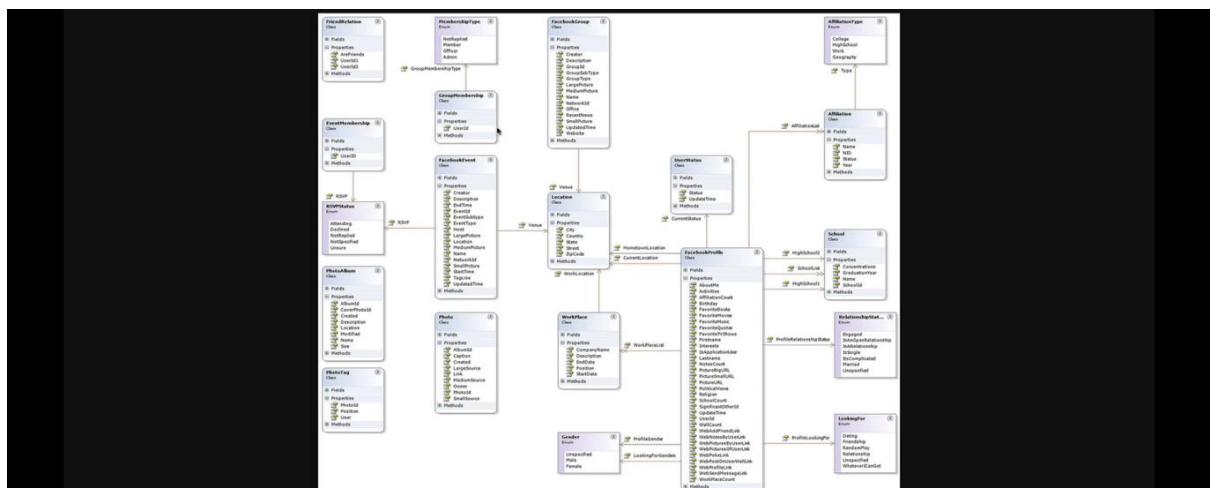
Reviews

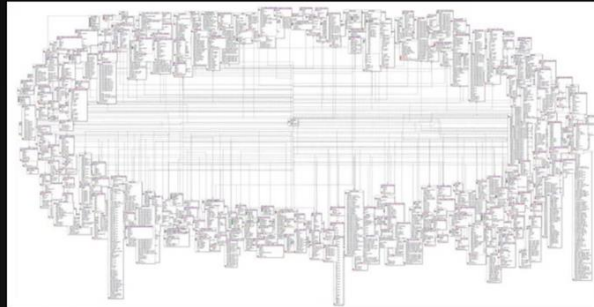
Books

Authors

Customers

Versions



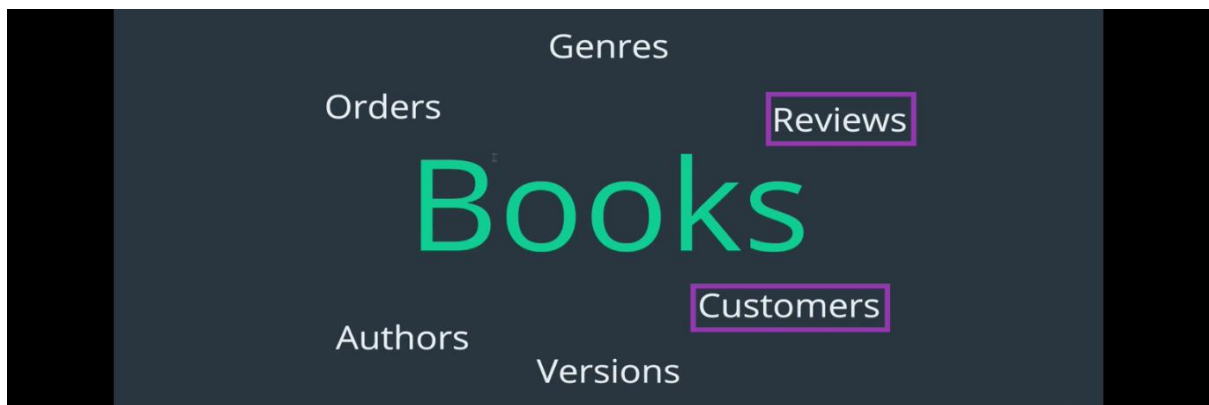


Relationship Basics

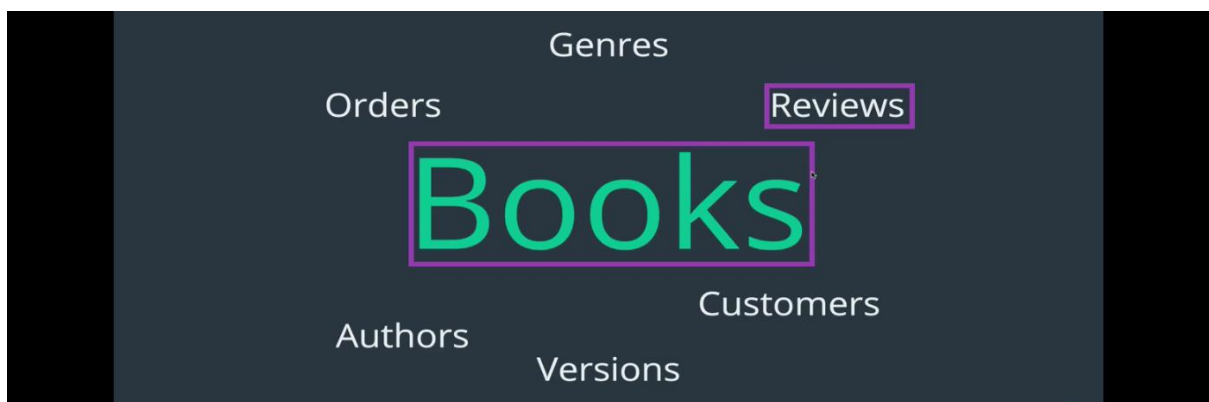
This is not marriage counseling

1. One to One Relationship
2. One to Many Relationship
3. Many to Many Relationship

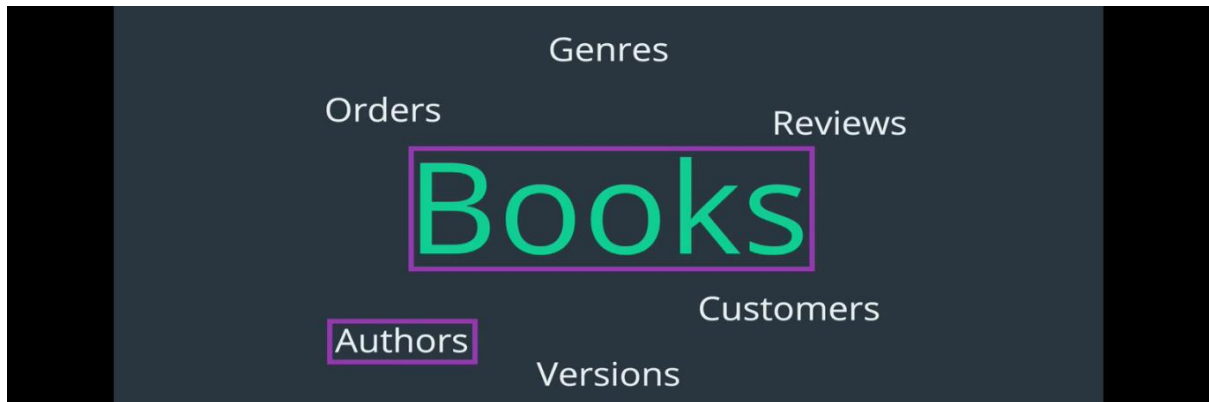
One to one



One to many



Many to many



One to many

1:MANY
The Most Common Relationship

**CUSTOMERS
& ORDERS**

We Want To Store...

- A customer's first and last name
- A customer's email
- The date of the purchase
- The price of the order

We Could Use One Table...

| first_name | last_name | email | order_date | amount |
|------------|-----------|------------------|--------------|--------|
| Boy | George | george@gmail.com | '2016/02/10' | 99.99 |
| Boy | George | george@gmail.com | '2017/11/11' | 35.50 |
| George | Michael | gm@gmail.com | '2014/12/12' | 800.67 |
| George | Michael | gm@gmail.com | '2015/01/03' | 12.50 |
| David | Bowie | david@gmail.com | NULL | NULL |
| Blue | Steele | blue@gmail.com | NULL | NULL |

Customers

- customer_id
- first_name
- last_name
- email

Orders

- order_id
- order_date
- amount
- customer_id

CUSTOMERS

| customer_id | first_name | last_name | email |
|-------------|------------|-----------|------------------|
| 1 | Boy | George | george@gmail.com |
| 2 | George | Michael | gm@gmail.com |
| 3 | David | Bowie | david@gmail.com |
| 4 | Blue | Steele | blue@gmail.com |

ORDERS

| order_id | order_date | amount | customer_id |
|----------|--------------|--------|-------------|
| 1 | '2016/02/10' | 99.99 | 1 |
| 2 | '2017/11/11' | 35.50 | 1 |
| 3 | '2014/12/12' | 800.67 | 2 |
| 4 | '2015/01/03' | 12.50 | 2 |

CUSTOMERS

| id | first_name | last_name | email |
|----|------------|-----------|------------------|
| 1 | Boy | George | george@gmail.com |
| 2 | George | Michael | gm@gmail.com |
| 3 | David | Bowie | david@gmail.com |
| 4 | Blue | Steele | blue@gmail.com |

ORDERS

| id | order_date | amount | customer_id |
|----|--------------|--------|-------------|
| 1 | '2016/02/10' | 99.99 | 1 |
| 2 | '2017/11/11' | 35.50 | 1 |
| 3 | '2014/12/12' | 800.67 | 2 |
| 4 | '2015/01/03' | 12.50 | 2 |

```
1 CREATE TABLE customers(  
2   id INT AUTO_INCREMENT PRIMARY KEY  
3   first_name VARCHAR(100),  
4   last_name VARCHAR(100),  
5   email VARCHAR(100)  
6 );
```

```
7  
8 CREATE TABLE orders(  
9   id INT AUTO_INCREMENT PRIMARY KEY,  
10  order_date DATE,  
11  amount DECIMAL(8,2),  
12  customer_id INT  
13 )
```

```
15  
16 INSERT INTO customers (first_name, last_name, email)  
17 VALUES ('Boy', 'George', 'george@gmail.com'),  
18 ('George', 'Michael', 'gm@gmail.com'),  
19 ('David', 'Bowie', 'david@gmail.com'),  
20 ('Blue', 'Steele', 'blue@gmail.com'),  
21 ('Bette', 'Davis', 'bette@aol.com');
```

```
22  
23 INSERT INTO orders (order_date, amount, customer_id)  
24 VALUES ('2016/02/10', 99.99, 1),  
25 ('2017/11/11', 35.50, 1),  
26 ('2014/12/12', 800.67, 2),  
27 ('2015/01/03', 12.50, 2),  
28 ('1999/04/11', 450.25, 5);
```

```

5 rows in set (0.00 sec)

mysql>
mysql> INSERT INTO orders (order_date, amount, customer_id)
-> VALUES ('2016/02/10', 33.67, 98);
Query OK, 1 row affected (0.01 sec)

```

```

1 CREATE TABLE customers(
2   id INT AUTO_INCREMENT PRIMARY KEY,
3   first_name VARCHAR(100),
4   last_name VARCHAR(100),
5   email VARCHAR(100)
6 );
7
8 CREATE TABLE orders(
9   id INT AUTO_INCREMENT PRIMARY KEY,
10  order_date DATE,
11  amount DECIMAL(8,2),
12  customer_id INT,
13  FOREIGN KEY(customer_id) REFERENCES customers(id)
14 );
15

```

```

7
8 CREATE TABLE orders(
9   id INT AUTO_INCREMENT PRIMARY KEY,
10  order_date DATE,
11  amount DECIMAL(8,2),
12  customer_id INT,
13  FOREIGN KEY(customer_id) REFERENCES customers(id)
14 );
15

```

```

16
17 INSERT INTO customers (first_name, last_name, email)
18 VALUES ('Boy', 'George', 'george@gmail.com'),
19 ('George', 'Michael', 'gm@gmail.com'),
20 ('David', 'Bowie', 'david@gmail.com'),
21 ('Blue', 'Steele', 'blue@gmail.com'),
22 ('Bette', 'Davis', 'bette@aol.com');
23
24 INSERT INTO orders (order_date, amount, customer_id)
25 VALUES ('2016/02/10', 33.67, 98);
26

```

```

23
24 INSERT INTO orders (order_date, amount, customer_id)
25 VALUES ('2016/02/10', 99.99, 1),
26 ('2017/11/11', 35.50, 1),
27 ('2014/12/12', 800.67, 2),
28 ('2015/01/03', 12.50, 2),
29 ('1999/04/11', 450.25, 5);
30
31

```

```

mysql> SELECT * FROM customers;
+----+-----+-----+-----+
| id | first_name | last_name | email |
+----+-----+-----+-----+
| 1 | Boy | George | george@gmail.com |
| 2 | George | Michael | gm@gmail.com |
| 3 | David | Bowie | david@gmail.com |
| 4 | Blue | Steele | blue@gmail.com |
| 5 | Bette | Davis | bette@aol.com |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> SELECT * FROM orders;
+----+-----+-----+-----+
| id | order_date | amount | customer_id |
+----+-----+-----+-----+
| 1 | 2016-02-10 | 99.99 | 1 |
| 2 | 2017-11-11 | 35.50 | 1 |
| 3 | 2014-12-12 | 800.67 | 2 |
| 4 | 2015-01-03 | 12.50 | 2 |
| 5 | 1999-04-11 | 450.25 | 5 |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

29 ('1999/04/11', 450.25, 5);
30
31 INSERT INTO orders (order_date, amount, customer_id)
32 VALUES ('2016/06/06', 33.67, 98);
33
34
35
36
37

```

```

5 rows in set (0.00 sec)

mysql> INSERT INTO orders (order_date, amount, customer_id)
-> VALUES ('2016/06/06', 33.67, 98);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint 'fails' fails ('customers_and_orders', 'orders', CONSTRAINT 'orders_ibfk_1' FOREIGN KEY ('customer_id') REFERENCES 'customers' ('id'))

```

Cross join

```

mysql> SELECT * FROM customers WHERE last_name='George';
+----+-----+-----+-----+
| id | first_name | last_name | email |
+----+-----+-----+-----+
| 1 | Boy | George | george@gmail.com |
+----+-----+-----+-----+
1 row in set (0.01 sec)

```



```

mysql>
mysql>
mysql> SELECT * FROM orders WHERE customer_id = 1;
+----+-----+-----+-----+
| id | order_date | amount | customer_id |
+----+-----+-----+-----+
| 1  | 2016-02-10 | 99.99 | 1           |
| 2  | 2017-11-11 | 35.50 | 1           |
+----+-----+-----+-----+
2 rows in set (0.01 sec)

```

```

34 SELECT id FROM customers WHERE last_name='George';
35 SELECT * FROM orders WHERE customer_id = 1;
36
37 SELECT * FROM orders WHERE customer_id =
38 (
39     SELECT id FROM customers
40     WHERE last_name='George'
41 );
42
43

```

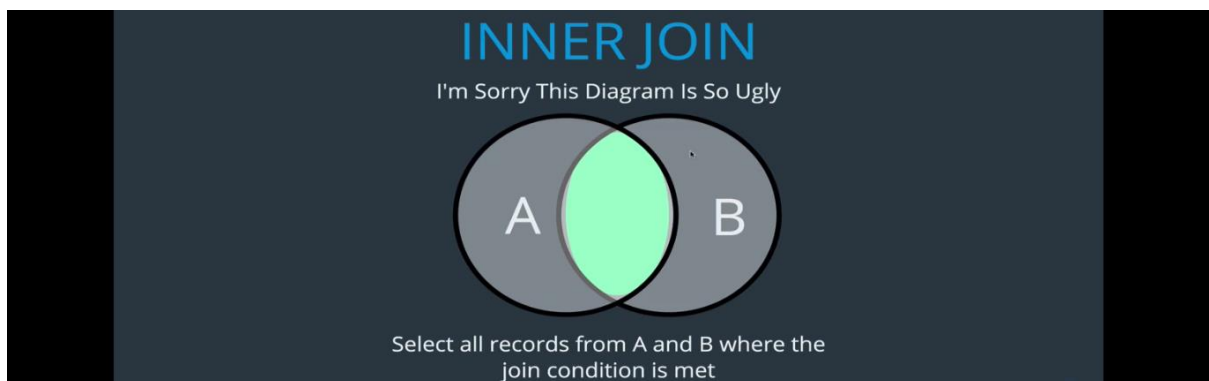
```

41 );
42
43 SELECT * FROM customers, orders;
44
45

```

| id | first_name | last_name | email | id | order_date | amount | customer_id |
|----|------------|-----------|------------------|----|------------|--------|-------------|
| 1 | Boy | George | george@gmail.com | 1 | 2016-02-10 | 99.99 | 1 |
| 2 | George | Michael | gm@gmail.com | 1 | 2016-02-10 | 99.99 | 1 |
| 3 | David | Bowie | david@gmail.com | 1 | 2016-02-10 | 99.99 | 1 |
| 4 | Blue | Steele | blue@gmail.com | 1 | 2016-02-10 | 99.99 | 1 |

Inner join (important)



```

44
45 SELECT * FROM customers, orders
46 WHERE customers.id = orders.customer_id;
47
48
49

```

(Inner join) Implicit join

```

47
48 SELECT first_name, last_name, order_date, amount
49 FROM customers, orders
50 WHERE customers.id = orders.customer_id;
51

```

```

mysql> SELECT first_name, last_name, order_date, amount
-> FROM customers, orders
-> WHERE customers.id = orders.customer_id;

```

| first_name | last_name | order_date | amount |
|------------|-----------|------------|--------|
| Boy | George | 2016-02-10 | 99.99 |
| Boy | George | 2017-11-11 | 35.50 |
| George | Michael | 2014-12-12 | 800.67 |
| George | Michael | 2015-01-03 | 12.50 |
| Bette | Davis | 1999-04-11 | 450.25 |

(Inner join) Explicit join (important)

The screenshot shows the SQL Studio interface. On the left, there's a sidebar with 'Commands' and 'Navigator'. The 'Navigator' pane shows a tree view of files: 'Dates', 'Untitled', 'Logical Operators', 'Notes', 'refining_selections' (containing 'distinct.sql', 'exercises.sql', 'limit.sql', 'order_by.sql'), 'Relationships' (containing 'CustomersOrders.sql' which is selected), and 'Schema'. Below this are several .sql files like 'book_data.sql', 'exercises.sql', etc.

The main area is the 'SQL Editor' where a query is written:

```

52
53 -- EXPLICIT INNER JOIN
54 SELECT * FROM customers
55 JOIN orders
56     ON customers.id = orders.customer_id;
57
58 SELECT first_name, last_name, order_date, amount
59 FROM customers
60 JOIN orders
61     ON customers.id = orders.customer_id;

```

Below the query, the results are displayed as a table with 8 columns: id, first_name, last_name, email, order_id, order_date, amount, and rownum. There are 5 rows of data.

| id | first_name | last_name | email | order_id | order_date | amount | rownum |
|----|------------|-----------|------------------|----------|------------|--------|--------|
| 1 | Boy | George | george@gmail.com | 1 | 2016-02-10 | 99.99 | 1 |
| 1 | Boy | George | george@gmail.com | 2 | 2017-11-11 | 35.50 | 1 |
| 2 | George | Michael | gm@gmail.com | 3 | 2014-12-12 | 800.67 | 2 |
| 2 | George | Michael | gm@gmail.com | 4 | 2015-01-03 | 12.50 | 2 |
| 5 | Bette | Davis | bette@aol.com | 5 | 1999-04-11 | 450.25 | 5 |

At the bottom, it says '5 rows in set (0.00 sec)'.

Notes

refining_selections

distinct.sql

exercises.sql

limit.sql

order_by.sql

Relationships

CustomersOrders.sql

Schema

book_code.sql

book_data.sql

exercises.sql

first_file.sql

like.sql

problems.sql

Commands

-- Getting Fancier

SELECT first_name, last_name, order_date, amount

FROM customers

JOIN orders

ON customers.id = orders.customer_id

ORDER BY amount;

(134 Bytes) 74:1 SQL Spaces: 4

| first_name | last_name | order_date | amount |
|------------|-----------|------------|--------|
| George | Michael | 2015-01-03 | 12.50 |
| Boy | George | 2017-11-11 | 35.50 |
| Boy | George | 2016-02-10 | 99.99 |
| Bette | Davis | 1999-04-11 | 450.25 |
| George | Michael | 2014-12-12 | 800.67 |

5 rows in set (0.00 sec)

Commands

- Notes
 - refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
 - Relationships
 - CustomersOrders.sql
 - Schema
- book_data.sql
- book_stats.sql
- exercises.sql
- first_file.sql
- likes.sql
- problems.sql

```

79
80
81 SELECT first_name, last_name, order_date, amount
82 FROM customers
83 JOIN orders
84   ON customers.id = orders.customer_id
85 GROUP BY orders.customer_id;
86
--
--
-- JOIN orders
--   JOIN orders
--     ON customers.id = orders.customer_id
--   GROUP BY orders.customer_id;
--
+-----+-----+-----+-----+
| first_name | last_name | order_date | amount |
+-----+-----+-----+-----+
| Boy        | George   | 2016-02-10 | 99.99  |
| George     | Michael  | 2014-12-12 | 800.67 |
| Bette      | Davis    | 1999-04-11 | 450.25 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

(146 Bytes) 81:1 SQL Spaces: 4

The screenshot shows a SQL IDE interface. On the left, there's a sidebar with a file explorer showing folders like 'Logical Operators', 'Notes', 'refining_selections', and 'Relationships'. Below it, a 'Commands' panel lists various actions like 'Untitled', 'distinct.sql', 'averages.sql', 'limit.sql', 'order_by.sql', 'CustomersOrders.sql', 'Schema', 'First_file.sql', 'like.sql', and 'problems.sql'. The main area is a code editor with a blue background, displaying a SQL query. To the right of the code editor, there's a 'Results' pane showing the output of the query as a table. At the bottom right, there's a status bar indicating '(185 Bytes) 81:1 SQL Spaces: 4'.

```

80
81 SELECT
82     first_name,
83     last_name,
84     order_date,
85     SUM(amount) AS total_spent
86 FROM customers
87 JOIN orders
88     ON customers.id = orders.customer_id
89 GROUP BY orders.customer_id;

```

| first_name | last_name | order_date | total_spent |
|------------|-----------|------------|-------------|
| Boy | George | 2016-02-10 | 135.49 |
| George | Michael | 2014-12-12 | 813.17 |
| Bette | Davis | 1999-04-11 | 450.25 |

3 rows in set (0.00 sec)

SQL Editor View (Left Panel):

- Commands: Navigate
- Logical Operators:
 - Logical Operators
 - Notes
 - refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
 - Relationships
 - CustomersOrders.sql

SQL Query (Main Panel):

```

102 SELECT
103     first_name,
104     last_name,
105     SUM(amount)
106 FROM customers
107 LEFT JOIN orders
108     ON customers.id = orders.customer_id
109 GROUP BY customers.id;
110
--> GROUP BY customers.id;

```

Result Set (Right Panel):

| first_name | last_name | SUM(amount) |
|------------|-----------|-------------|
| Boy | George | 135.49 |
| George | Michael | 813.17 |
| David | Bowie | NULL |
| Blue | Steele | NULL |
| Bette | Davis | 450.25 |

5 rows in set (0.00 sec)

SQL Editor View (Left Panel):

- Commands: Navigate
- Logical Operators:
 - Logical Operators
 - Notes
 - refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
 - Relationships
 - CustomersOrders.sql

SQL Query (Main Panel):

```

102 SELECT
103     first_name,
104     last_name,
105     IFNULL(SUM(amount), 0) AS total_spent
106 FROM customers
107 LEFT JOIN orders
108     ON customers.id = orders.customer_id
109 GROUP BY customers.id
110 ORDER BY total_spent;
111

```

Result Set (Right Panel):

| first_name | last_name | total_spent |
|------------|-----------|-------------|
| Blue | Steele | 0.00 |
| David | Bowie | 0.00 |
| Boy | George | 135.49 |
| Bette | Davis | 450.25 |
| George | Michael | 813.17 |

5 rows in set (0.00 sec)

Right join (help to find wrong id from left)



SQL Editor View (Left Panel):

- Commands: Navigate
- Logical Operators:
 - Logical Operators
 - Notes
 - refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
 - Relationships
 - CustomersOrders.sql

SQL Query (Main Panel):

```

110
111 -- RIGHT JOIN
112 SELECT * FROM customers
113 RIGHT JOIN orders
114     ON customers.id = orders.customer_id;
115
116
117

```

Result Set (Right Panel):

| | id | first_name | last_name | email | id | order_date | amount | customer_1 |
|---|----|------------|-----------|------------------|----|------------|--------|------------|
| 1 | 1 | Boy | George | george@gmail.com | 1 | 2016-02-10 | 99.99 | |
| 1 | 1 | Boy | George | george@gmail.com | 2 | 2017-11-11 | 35.50 | |
| 1 | 2 | George | Michael | gm@gmail.com | 3 | 2014-12-12 | 800.67 | |
| 2 | 2 | George | Michael | gm@gmail.com | 4 | 2015-01-03 | 12.50 | |
| 2 | 5 | Bette | Davis | bette@aol.com | 5 | 1999-04-11 | 450.25 | |

```

57
58 SELECT first_name, last_name, order_date, amount
59 FROM customers
60 JOIN orders
61 ON customers.id = orders.customer_id;
62
--> ON customers.id = orders.customer_id;
+-----+-----+-----+-----+
| first_name | last_name | order_date | amount |
+-----+-----+-----+-----+
| Boy        | George   | 2016-02-10 | 99.99  |
| Boy        | George   | 2017-11-11 | 35.50  |
| George     | Michael  | 2014-12-12 | 800.67 |
| George     | Michael  | 2015-01-03 | 12.50  |
| Bette      | Davis    | 1999-04-11 | 450.25 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Foreign key wont allow to add or delete

```

mysql> DELETE FROM customers WHERE first_name='Boy';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (
`customers_and_orders`.`orders`, CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer_id`) RE
FERENCES `customers` (`id`))
mysql>

mysql>
mysql> DROP TABLE customers;
ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constraint fails
mysql> DROP TABLE orders, customer;
ERROR 1051 (42S02): Unknown table 'customer'
mysql> DROP TABLE customers;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW TABLES;
Empty set (0.00 sec)

```

If you delete foreign key, you will be able to add or delete data

1-Delete customer and order table

2- create customer and order table again

3- and add this data

```

146 INSERT INTO orders (order_date, amount, customer_id) VALUES
147 ('2017/11/05', 23.45, 45);
148 (CURDATE(), 777.77, 109);
149
+-----+-----+-----+-----+
| id | order_date | amount | customer_id |
+-----+-----+-----+-----+
| 1  | 2016-02-10 | 99.99  | 1           |
| 2  | 2017-11-11 | 35.50  | 1           |
| 3  | 2014-12-12 | 800.67 | 2           |
| 4  | 2015-01-03 | 12.50  | 2           |
| 5  | 1999-04-11 | 450.25 | 5           |
| 6  | 2017-11-05 | 23.45  | 45          |
| 7  | 2017-04-26 | 777.77 | 109         |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

```

58 SELECT first_name, last_name, order_date, amount
59 FROM customers
60 JOIN orders
61 ON customers.id = orders.customer_id;
62
+-----+-----+-----+-----+
| first_name | last_name | order_date | amount |
+-----+-----+-----+-----+
| Boy        | George   | 2016-02-10 | 99.99  |
| Boy        | George   | 2017-11-11 | 35.50  |
| George     | Michael  | 2014-12-12 | 800.67 |
| George     | Michael  | 2015-01-03 | 12.50  |
| Bette      | Davis    | 1999-04-11 | 450.25 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

95
96 SELECT first_name, last_name, order_date, amount
97 FROM customers
98 LEFT JOIN orders
99 ON customers.id = orders.customer_id;
100
+-----+-----+-----+-----+
| first_name | last_name | order_date | amount |
+-----+-----+-----+-----+
| Boy        | George   | 2016-02-10 | 99.99  |
| Boy        | George   | 2017-11-11 | 35.50  |
| George     | Michael  | 2014-12-12 | 800.67 |
| George     | Michael  | 2015-01-03 | 12.50  |
| David      | Bowie    | NULL       | NULL   |
| Blue       | Steele   | NULL       | NULL   |
| Bette      | Davis    | 1999-04-11 | 450.25 |
+-----+-----+-----+-----+

```

Right join main

Commands

```

149
150 -- RIGHT JOIN
151 SELECT first_name, last_name, order_date, amount
152 FROM customers
153 RIGHT JOIN orders
154 ON customers.id = orders.customer_id;

```

| first_name | last_name | order_date | amount |
|------------|-----------|------------|--------|
| Boy | George | 2016-02-10 | 99.99 |
| Boy | George | 2017-11-11 | 35.50 |
| George | Michael | 2014-12-12 | 800.67 |
| George | Michael | 2015-01-03 | 12.50 |
| Bette | Davis | 1999-04-11 | 450.25 |
| NULL | NULL | 2017-11-05 | 23.45 |
| NULL | NULL | 2017-04-26 | 777.77 |

Debugger

Commands

```

150 -- RIGHT JOIN
151 SELECT first_name, last_name, order_date, amount
152 FROM customers
153 RIGHT JOIN orders
154 ON customers.id = orders.customer_id
155 ORDER BY first_name;

```

| first_name | last_name | order_date | amount |
|------------|-----------|------------|--------|
| NULL | NULL | 2017-04-26 | 777.77 |
| NULL | NULL | 2017-11-05 | 23.45 |
| Bette | Davis | 1999-04-11 | 450.25 |
| Boy | George | 2017-11-11 | 35.50 |
| Boy | George | 2016-02-10 | 99.99 |
| George | Michael | 2015-01-03 | 12.50 |
| George | Michael | 2014-12-12 | 800.67 |

Debugger

Commands

```

149
150 -- RIGHT JOIN
151 SELECT first_name, last_name, order_date, amount, SUM(amount)
152 FROM customers
153 RIGHT JOIN orders
154 ON customers.id = orders.customer_id
155 GROUP BY customer_id;

```

| first_name | last_name | order_date | amount | SUM(amount) |
|------------|-----------|------------|--------|-------------|
| Boy | George | 2016-02-10 | 99.99 | 135.49 |
| George | Michael | 2014-12-12 | 800.67 | 813.17 |
| Bette | Davis | 1999-04-11 | 450.25 | 450.25 |
| NULL | NULL | 2017-11-05 | 23.45 | 23.45 |
| NULL | NULL | 2017-04-26 | 777.77 | 777.77 |

5 rows in set (0.17 sec)

Debugger

Commands

```

151 SELECT
152 IFNULL(first_name, 'MISSING'),
153 IFNULL(last_name, 'USER'),
154 order_date,
155 amount,
156 SUM(amount)
157 FROM customers
158 RIGHT JOIN orders
159 ON customers.id = orders.customer_id
160 GROUP BY customer_id;

```

| IFNULL(first_name, 'MISSING') | IFNULL(last_name, 'USER') | order_date | amount | SUM(amount) |
|-------------------------------|---------------------------|------------|--------|-------------|
| Boy | George | 2016-02-10 | 99.99 | 135.49 |
| George | Michael | 2014-12-12 | 800.67 | 813.17 |
| Bette | Davis | 1999-04-11 | 450.25 | 450.25 |
| MISSING | USER | 2017-11-05 | 23.45 | 23.45 |

Debugger

Commands

```

151 SELECT
152 IFNULL(first_name, 'MISSING') AS first,
153 IFNULL(last_name, 'USER') AS last,
154 order_date,
155 amount,
156 SUM(amount)
157 FROM customers
158 RIGHT JOIN orders
159 ON customers.id = orders.customer_id
160 GROUP BY first_name, last_name;

```

| first | last | order_date | amount | SUM(amount) |
|---------|---------|------------|--------|-------------|
| MISSING | USER | 2017-11-05 | 23.45 | 801.22 |
| Bette | Davis | 1999-04-11 | 450.25 | 450.25 |
| Boy | George | 2016-02-10 | 99.99 | 135.49 |
| George | Michael | 2014-12-12 | 800.67 | 813.17 |

4 rows in set (0.00 sec)

Debugger

On delete cascade


```

172 CREATE TABLE orders(
173     id INT AUTO_INCREMENT PRIMARY KEY,
174     order_date DATE,
175     amount DECIMAL(8,2),
176     customer_id INT,
177     FOREIGN KEY(customer_id)
178         REFERENCES customers(id)
179         ON DELETE CASCADE
180 );

```

```

mysql> DELETE FROM customers WHERE email = 'george@gmail.com';
Query OK, 1 row affected (0.01 sec)

```

Exercises

WRITE THIS SCHEMA

STUDENTS

- id
- first_name

PAPERS

- title
- grade
- student_id

```

1 CREATE TABLE students (
2     ID INT AUTO_INCREMENT PRIMARY KEY,
3     first_name VARCHAR(100)
4 );
5
6 CREATE TABLE papers (
7     title VARCHAR(100),
8     grade INT,
9     student_id INT,
10    FOREIGN KEY (student_id)
11        REFERENCES students(id)
12        ON DELETE CASCADE
13 );

```

```

mysql> SHOW TABLES;
+-----+
| Tables_in_customers_and_orders |
+-----+
| customers                       |
| orders                         |
| papers                         |
| students                       |
+-----+
4 rows in set (0.00 sec)

```

```

6 CREATE TABLE papers (
7     title VARCHAR(100),
8     grade INT,
9     student_id INT,
10    FOREIGN KEY (student_id)
11        REFERENCES students(id)
12        ON DELETE CASCADE
13 );

```

```

mysql> describe papers;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| title      | varchar(100)  | YES  |     | NULL    |      |
| grade      | int(11)       | YES  |     | NULL    |      |
| student_id | int(11)       | YES  | MUL | NULL    |      |
+-----+
3 rows in set (0.00 sec)

```


INSERT THIS DATA (COPY AND PASTE IT)

```
INSERT INTO students (first_name) VALUES  
( 'Caleb'), ( 'Samantha'), ( 'Raj'), ( 'Carlos'), ( 'Lisa');  
  
INSERT INTO papers (student_id, title, grade ) VALUES  
(1, 'My First Book Report', 60),  
(1, 'My Second Book Report', 75),  
(2, 'Russian Lit Through The Ages', 94),  
(2, 'De Montaigne and The Art of The Essay', 98),  
(4, 'Borges and Magical Realism', 89);
```

```
mysql> SELECT * FROM papers;
```

| title | grade | student_id |
|---------------------------------------|-------|------------|
| My First Book Report | 60 | 1 |
| My Second Book Report | 75 | 1 |
| Russian Lit Through The Ages | 94 | 2 |
| De Montaigne and The Art of The Essay | 98 | 2 |
| Borges and Magical Realism | 89 | 4 |

5 rows in set (0.00 sec)

PRINT THIS

| first_name | title | grade |
|------------|---------------------------------------|-------|
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Samantha | Russian Lit Through The Ages | 94 |
| Carlos | Borges and Magical Realism | 89 |
| Caleb | My Second Book Report | 75 |
| Caleb | My First Book Report | 60 |

```
25 -- EXERCISE 1  
26  
27 SELECT *  
28 FROM students  
29 INNER JOIN papers  
30 ON students.id = papers.student_id;
```

| id | first_name | title | grade | student_id |
|----|------------|---------------------------------------|-------|------------|
| 1 | Caleb | My First Book Report | 60 | 1 |
| 1 | Caleb | My Second Book Report | 75 | 1 |
| 2 | Samantha | Russian Lit Through The Ages | 94 | 2 |
| 2 | Samantha | De Montaigne and The Art of The Essay | 98 | 2 |
| 4 | Carlos | Borges and Magical Realism | 89 | 4 |

5 rows in set (0.00 sec)

```
26 -- EXERCISE 1  
27 SELECT first_name, title, grade  
28 FROM students  
29 INNER JOIN papers  
30 ON students.id = papers.student_id  
31 ORDER BY grade DESC;
```

| first_name | title | grade |
|------------|---------------------------------------|-------|
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Samantha | Russian Lit Through The Ages | 94 |
| Carlos | Borges and Magical Realism | 89 |
| Caleb | My Second Book Report | 75 |
| Caleb | My First Book Report | 60 |

5 rows in set (0.00 sec)

PRINT THIS

| first_name | title | grade |
|------------|---------------------------------------|-------|
| Caleb | My First Book Report | 60 |
| Caleb | My Second Book Report | 75 |
| Samantha | Russian Lit Through The Ages | 94 |
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Raj | NULL | NULL |
| Carlos | Borges and Magical Realism | 89 |
| Lisa | NULL | NULL |

```

40 -- PROBLEM 2
41 SELECT *
42 FROM students
43 LEFT JOIN papers
44 ON students.id = papers.student_id;

```

| id | first_name | title | grade | student_id |
|----|------------|---------------------------------------|-------|------------|
| 1 | Caleb | My First Book Report | 60 | 1 |
| 1 | Caleb | My Second Book Report | 75 | 1 |
| 2 | Samantha | Russian Lit Through The Ages | 94 | 2 |
| 2 | Samantha | De Montaigne and The Art of The Essay | 98 | 2 |
| 3 | Raj | NULL | NULL | NULL |
| 4 | Carlos | Borges and Magical Realism | 89 | 4 |
| 5 | Lisa | NULL | NULL | NULL |

7 rows in set (0.00 sec)

Commands

- refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
- Relationships
 - CustomersOrders.sql
- testing
 - book_code.sql
 - book_data.sql
 - exercises.sql
 - first_file.sql
 - like.sql
 - problems.sql

```

29
40 -- PROBLEM 2
41 SELECT first_name, title, grade
42 FROM students
43 LEFT JOIN papers
44     ON students.id = papers.student_id;
45

```

| first_name | title | grade |
|------------|---------------------------------------|-------|
| Caleb | My First Book Report | 60 |
| Caleb | My Second Book Report | 75 |
| Samantha | Russian Lit Through The Ages | 94 |
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Raj | NULL | NULL |
| Carlos | Borges and Magical Realism | 89 |
| Lisa | NULL | NULL |

7 rows in set (0.00 sec)

Database

PRINT THIS

| first_name | title | grade |
|------------|---------------------------------------|-------|
| Caleb | My First Book Report | 60 |
| Caleb | My Second Book Report | 75 |
| Samantha | Russian Lit Through The Ages | 94 |
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Raj | MISSING | 0 |
| Carlos | Borges and Magical Realism | 89 |
| Lisa | MISSING | 0 |

Commands

Navigation

- Logical Operators
 - Notes
- refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
- Relationships
 - Advanced SQL Syntax and

```

27 SELECT
28     first_name,
29     IFNULL(title, 'MISSING'),
30     IFNULL(grade, 0)
31 FROM students
32 LEFT JOIN papers
33     ON students.id = papers.student_id;
  
```

Schema Designer

Commands

Navigation

- testing
 - book_code.sql
 - book_data.sql
 - exercises.sql
 - first_run.sql
 - llak.sql
 - problems.sql

| first_name | IFNULL(title, 'MISSING') | IFNULL(grade, 0) |
|------------|---------------------------------------|------------------|
| Caleb | My First Book Report | 60 |
| Caleb | My Second Book Report | 75 |
| Samantha | Russian Lit Through The Ages | 94 |
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Raj | MISSING | 0 |
| Carlos | Borges and Magical Realism | 89 |
| Lisa | MISSING | 0 |

7 rows in set (0.00 sec)

Schema Designer

```

PRINT THIS
+-----+-----+
| first_name | average |
+-----+-----+
| Samantha  | 96.0000 |
| Carlos    | 89.0000 |
| Caleb     | 67.5000 |
| Raj       | 0        |
| Lisa      | 0        |
+-----+-----+

```

- book_data.sql
- exercises.sql
- first_file.sql
- like.sql
- problems.sql

| first_name | grade |
|------------|-------|
| Caleb | 60 |
| Caleb | 75 |
| Samantha | 94 |
| Samantha | 98 |
| Raj | NULL |
| Carlos | 89 |
| Lisa | NULL |

- Notes
- refining_selections
- distinct.sql
- exercises.sql
- limit.sql
- order_by.sql
- Relationships
- CustomersOrders.sql

```

56 SELECT
57     first_name,
58     grade
59 FROM students
60 LEFT JOIN papers
61     ON students.id = papers.student_id;

```

| id | first_name | title | grade | student_id |
|----|------------|---------------------------------------|-------|------------|
| 1 | Caleb | My First Book Report | 60 | 1 |
| 1 | Caleb | My Second Book Report | 75 | 1 |
| 2 | Samantha | Russian Lit Through The Ages | 94 | 2 |
| 2 | Samantha | De Montaigne and The Art of The Essay | 98 | 2 |
| 3 | Raj | NULL | NULL | NULL |
| 4 | Carlos | Borges and Magical Realism | 89 | 4 |
| 5 | Lisa | NULL | NULL | NULL |

7 rows in set (0.00 sec)

- Untitled
- Logical Operators
- Notes
- refining_selections
- distinct.sql
- exercises.sql
- limit.sql
- order_by.sql
- Relationships
- CustomersOrders.sql

```

56 SELECT
57     first_name,
58     grade
59 FROM students
60 LEFT JOIN papers
61     ON students.id = papers.student_id
62 GROUP BY students.id;
63

```

| first_name | grade |
|------------|-------|
| Caleb | 60 |
| Samantha | 94 |
| Raj | NULL |
| Carlos | 89 |
| Lisa | NULL |

5 rows in set (0.00 sec)

- Untitled
- Logical Operators
- Notes
- refining_selections
- distinct.sql
- exercises.sql
- limit.sql
- order_by.sql
- Relationships
- CustomersOrders.sql

```

56 SELECT
57     first_name,
58     AVG(grade)
59 FROM students
60 LEFT JOIN papers
61     ON students.id = papers.student_id
62 GROUP BY students.id;
63

```

| first_name | AVG(grade) |
|------------|------------|
| Caleb | 67.5000 |
| Samantha | 96.0000 |
| Raj | NULL |
| Carlos | 89.0000 |
| Lisa | NULL |

5 rows in set (0.01 sec)

- Untitled
- Logical Operators
- Notes
- refining_selections
- distinct.sql
- exercises.sql
- limit.sql
- order_by.sql
- Relationships
- CustomersOrders.sql
- Schema

```

56 SELECT
57     first_name,
58     IFNULL(AVG(grade), 0)
59 FROM students
60 LEFT JOIN papers
61     ON students.id = papers.student_id
62 GROUP BY students.id;
63
64

```

| first_name | IFNULL(AVG(grade), 0) |
|------------|-----------------------|
| Caleb | 67.5000 |
| Samantha | 96.0000 |
| Raj | 0.0000 |
| Carlos | 89.0000 |
| Lisa | 0.0000 |

5 rows in set (0.00 sec)

Commands

Navigation

- Logical Operators
- Notes
- refining_selections
 - distinct.sql
 - exercises.sql
 - limit.sql
 - order_by.sql
- Relationships
 - CustomersOrders.sql
 - book_data.sql
 - exercises.sql
 - first_file.sql
 - like.sql
 - problems.sql

```

36 SELECT
37     first_name,
38     IFNULL(AVG(grade), 0) AS average
39 FROM students
40 LEFT JOIN papers
41     ON students.id = papers.student_id
42 GROUP BY students.id
43 ORDER BY average DESC;

```

| first_name | average |
|------------|---------|
| Samantha | 96.0000 |
| Carlos | 89.0000 |
| Caleb | 67.5000 |
| Lisa | 0.0000 |
| Raj | 0.0000 |

5 rows in set (0.00 sec)

Online Debugger

PRINT THIS

| first_name | average | passing_status |
|------------|---------|----------------|
| Samantha | 96.0000 | PASSING |
| Carlos | 89.0000 | PASSING |
| Caleb | 67.5000 | FAILING |
| Raj | 0 | FAILING |
| Lisa | 0 | FAILING |

```
mysql> SELECT
->     first_name,
->     IFNULL(AVG(grade), 0) AS average,
->     CASE
->         WHEN AVG(grade) >= 75 THEN 'PASSING'
->         ELSE 'FAILING'
->     END AS passing_status
-> FROM students
-> LEFT JOIN papers
->     ON students.id = papers.student_id
-> GROUP BY students.id
-> ORDER BY average DESC;
+-----+-----+-----+
| first_name | average | passing_status |
+-----+-----+-----+
| Samantha  | 96.0000 | PASSING        |
| Carlos    | 89.0000 | PASSING        |
| Caleb     | 67.5000 | FAILING        |
| Raj       | 0.0000  | FAILING        |
| Lisa      | 0.0000  | FAILING        |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT
  ->     first_name,
  ->     IFNULL(AVG(grade), 0) AS average,
  ->     CASE
  ->         WHEN AVG(grade) IS NULL THEN 'FAILING'
  ->         WHEN AVG(grade) >= 75 THEN 'PASSING'
  ->         ELSE 'FAILING'
  ->     END AS passing_status
  -> FROM students
  -> LEFT JOIN papers
  ->     ON students.id = papers.student_id
```

| first_name | average | passing_status |
|------------|---------|----------------|
| Samantha | 96.0000 | PASSING |
| Carlos | 89.0000 | PASSING |
| Caleb | 67.5000 | FAILING |
| Raj | 0.0000 | FAILING |
| Lisa | 0.0000 | FAILING |

5 rows in set (0.00 sec)

Below problem solve by above slide

```
mysql> SELECT NULL >= 1;
```

| |
|------|
| NULL |
|------|

1 row in set (0.00 sec)