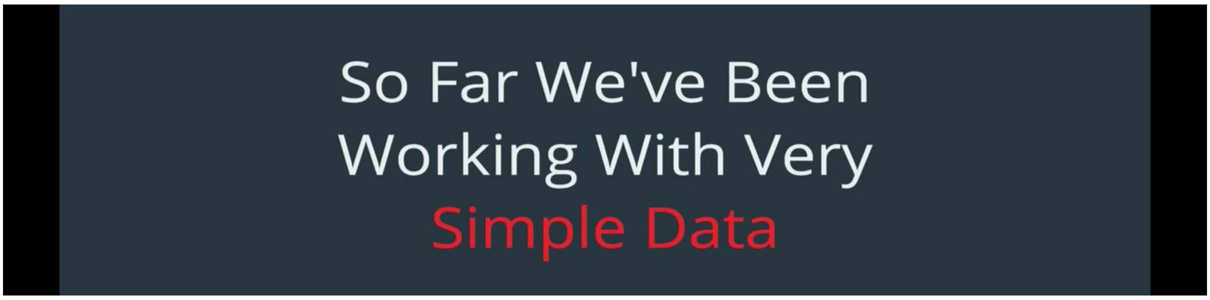
Lesson 12- (Relationship) one to many

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

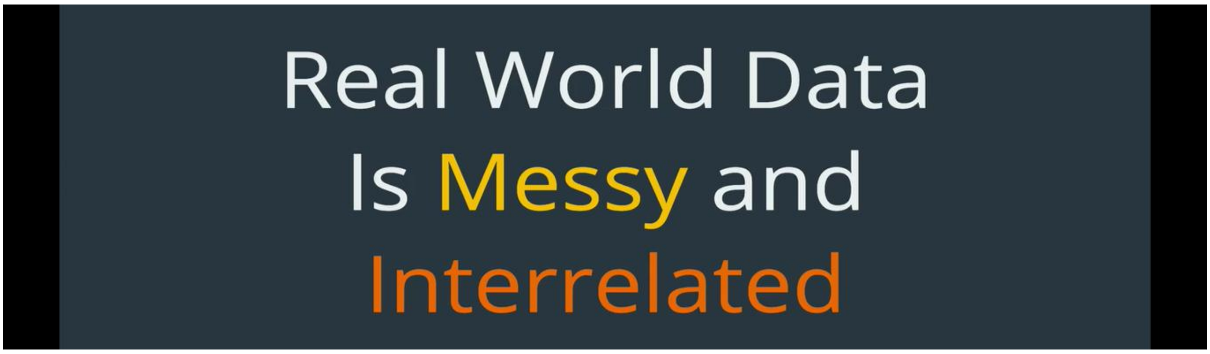
Real world data is messy

RELATIONSHIPS
AND JOINS

Lesson 12



So Far We've Been
Working With Very
Simple Data



Real World Data Is Messy and Interrelated

Books

Versions

Genres

Orders

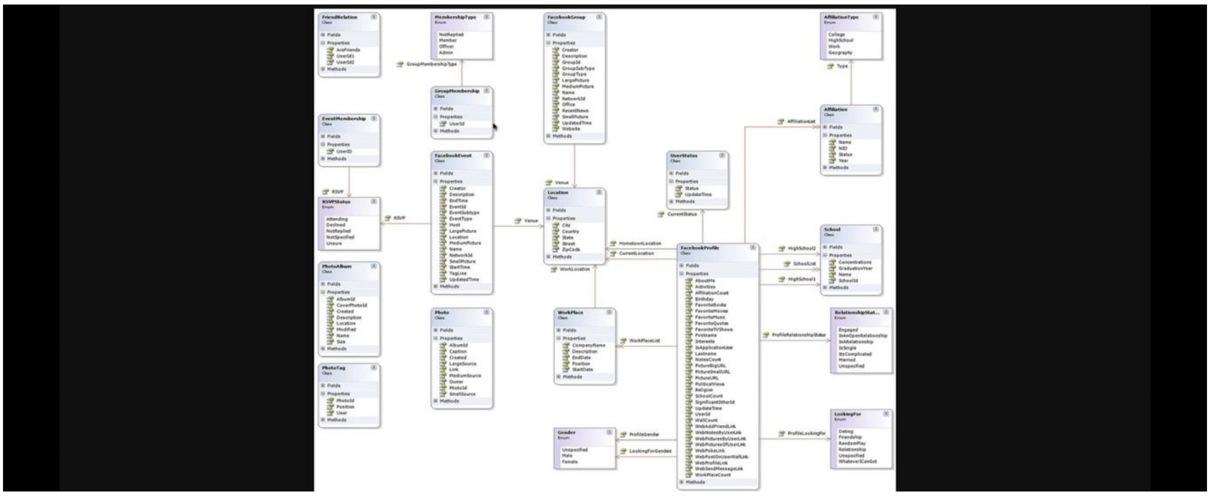
Reviews

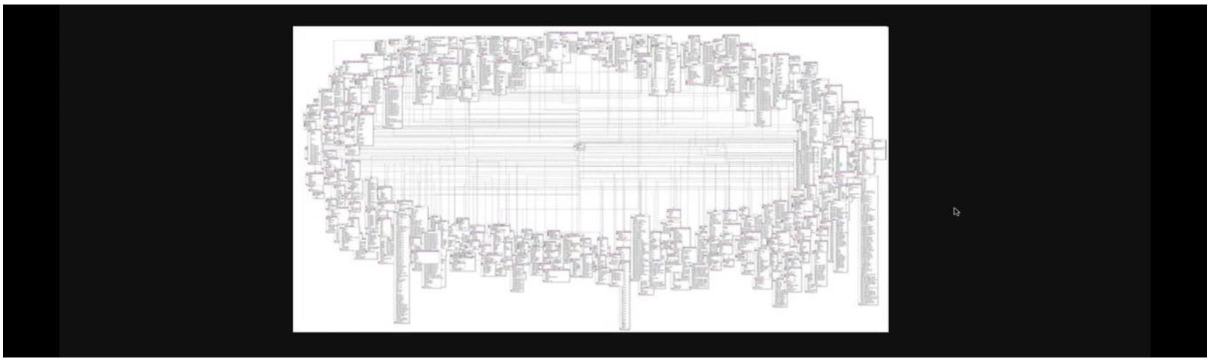
Books

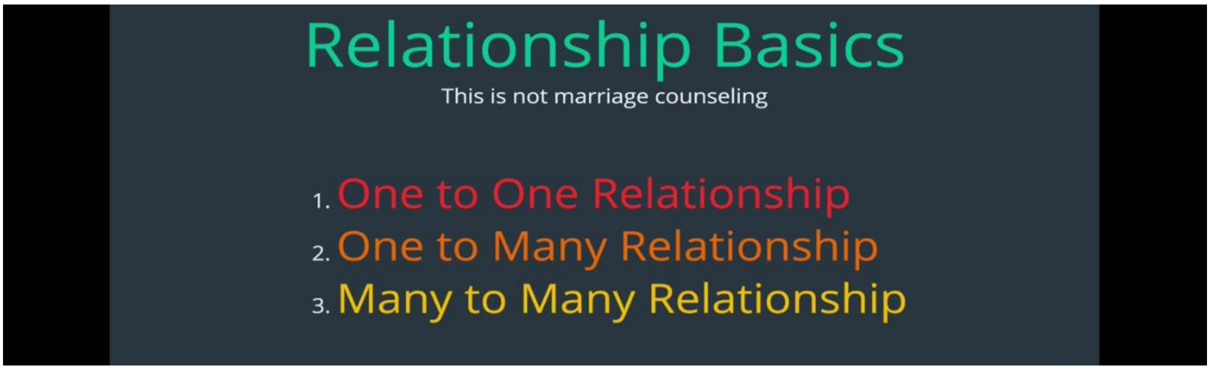
Customers

Authors

Versions





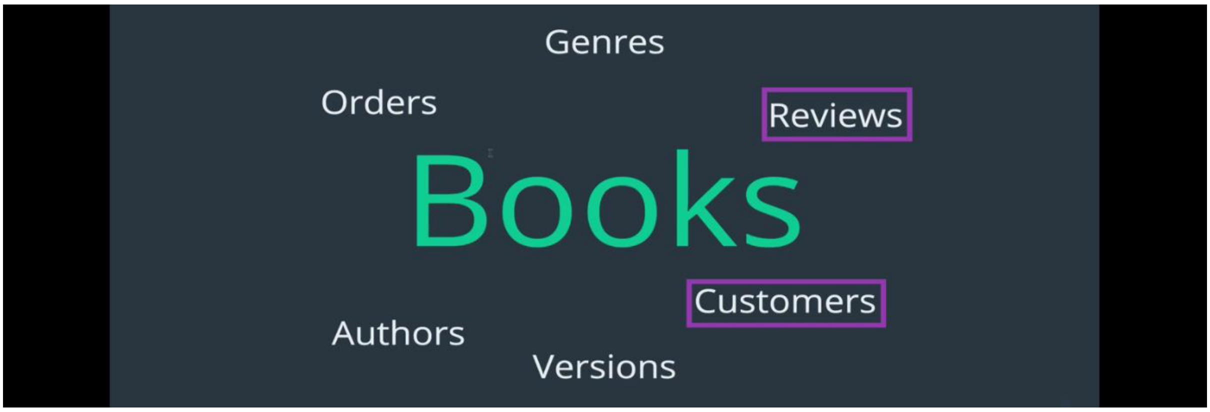


Relationship Basics

This is not marriage counseling

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

One to one



Genres

Orders

Reviews

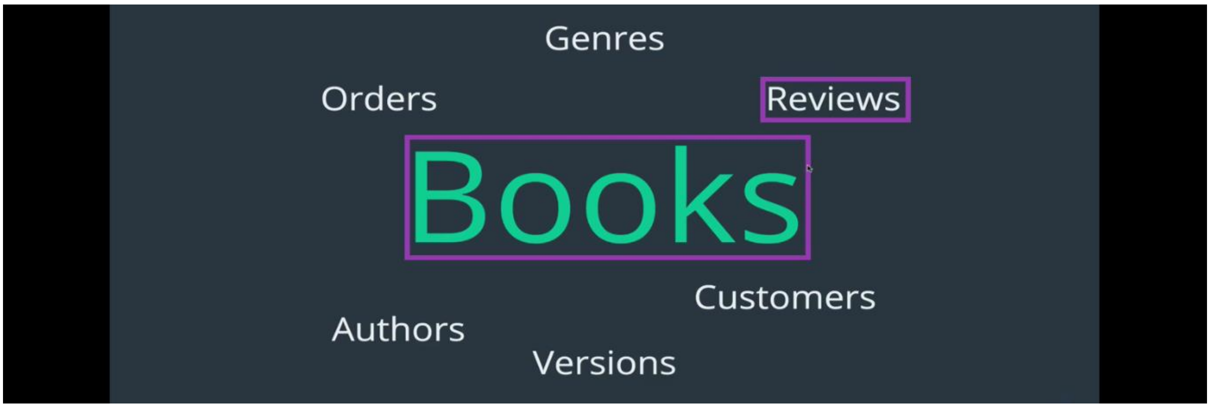
Books

Customers

Authors

Versions

One to many



Genres

Orders

Reviews

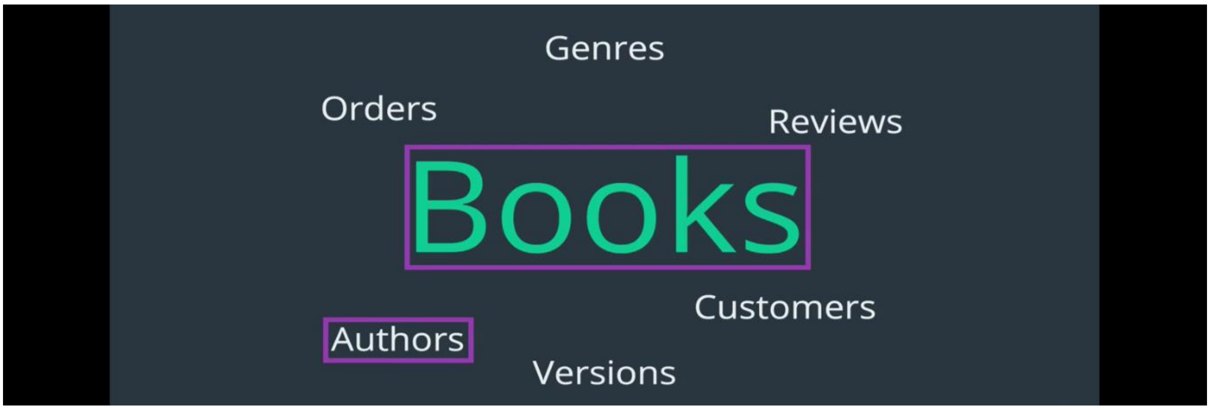
Books

Customers

Authors

Versions

Many to many



Genres

Orders

Reviews

Books

Customers

Authors

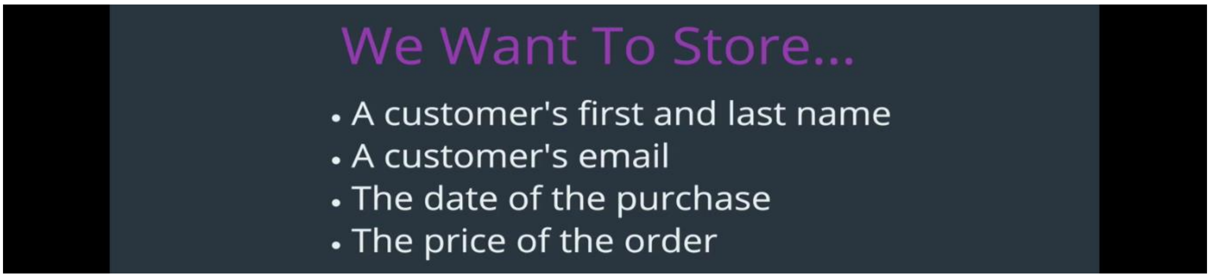
Versions

One to many

1:MANY

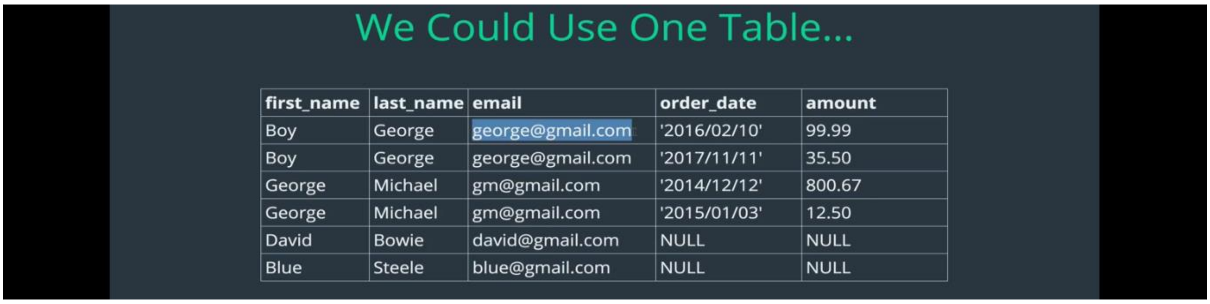
The Most Common Relationship

CUSTOMERS & ORDERS



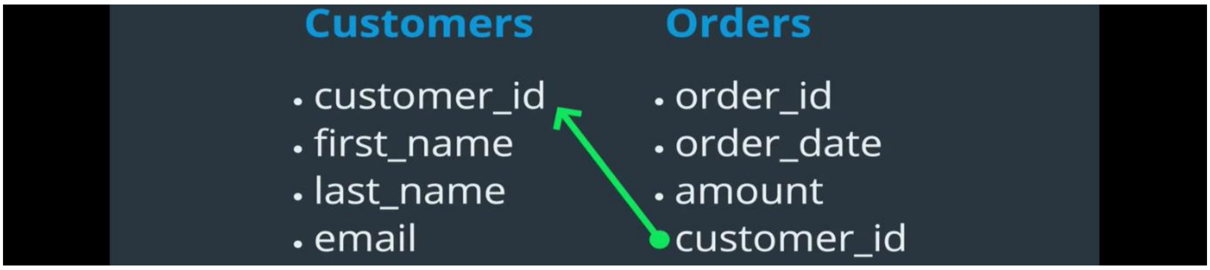
We Want To Store...

- 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

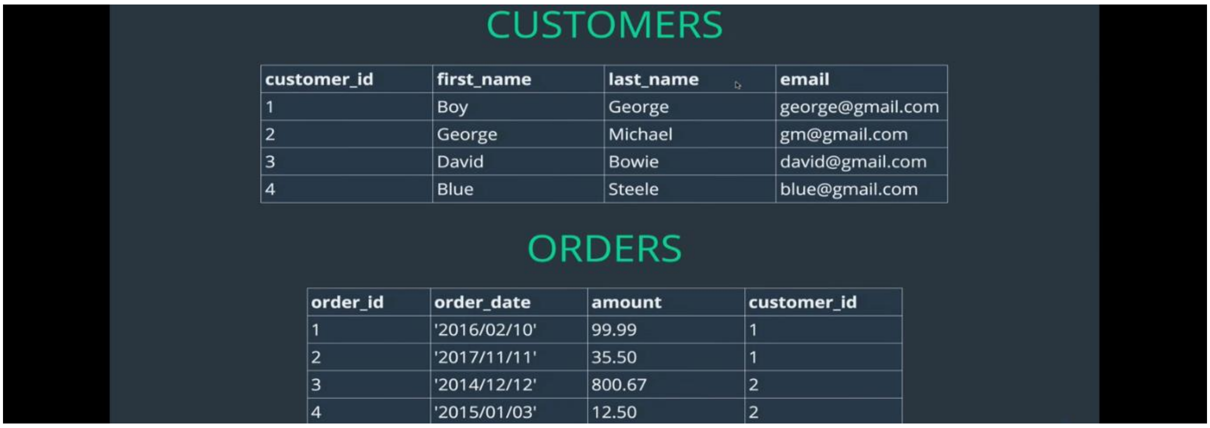


Customers

- customer_id
- first_name
- last_name
- email

Orders

- order_id
- order_date
- amount
- customer_id

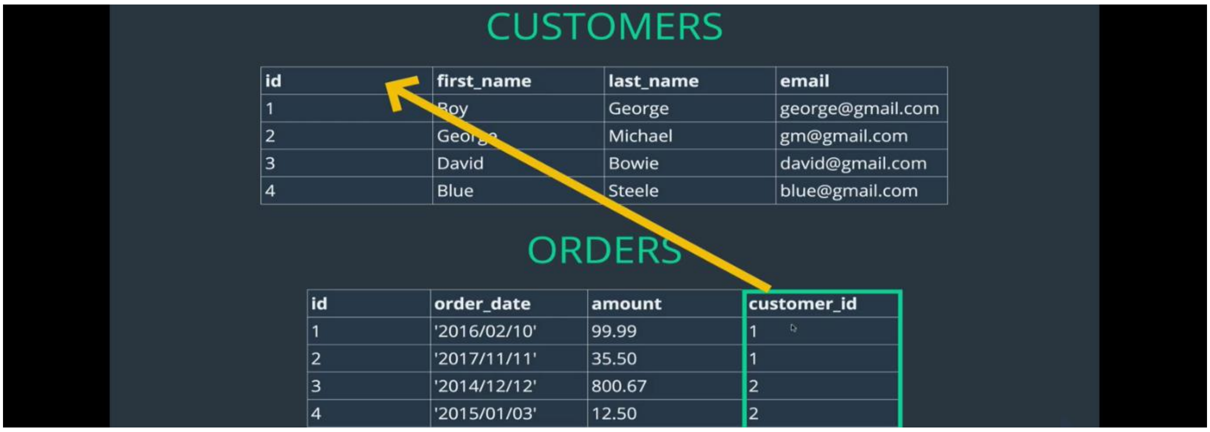


CUSTOMERS

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

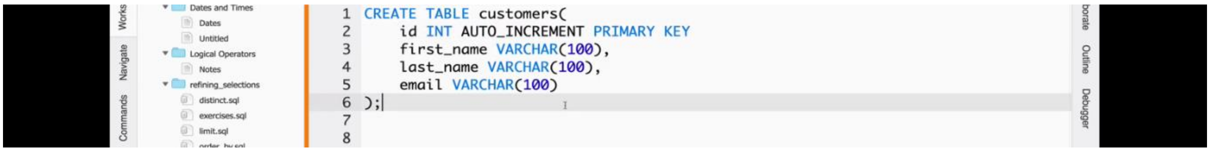


CUSTOMERS

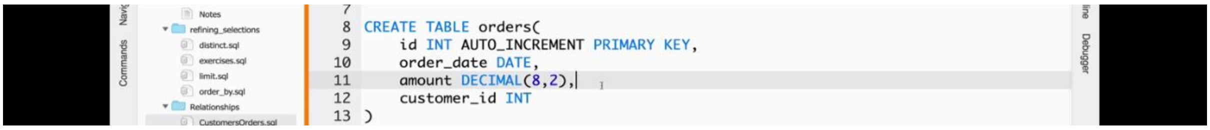
id	first_name	last_name	email
1	Boy	George	george@gmail.com
2	George	Michael	gm@gmail.com
3	David	Bowie	david@gmail.com
4	Blue	Steele	blue@gmail.com

ORDERS

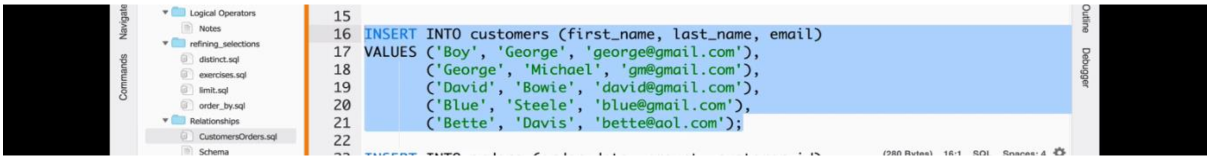
id	order_date	amount	customer_id
1	'2016/02/10'	99.99	1
2	'2017/11/11'	35.50	1
3	'2014/12/12'	800.67	2
4	'2015/01/03'	12.50	2



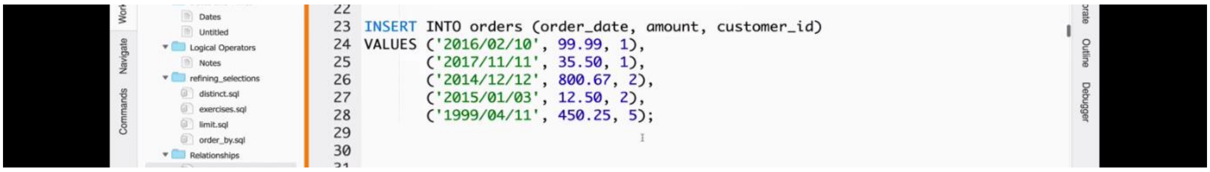
```
CREATE TABLE customers(
    id INT AUTO_INCREMENT PRIMARY KEY
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    email VARCHAR(100)
);
```



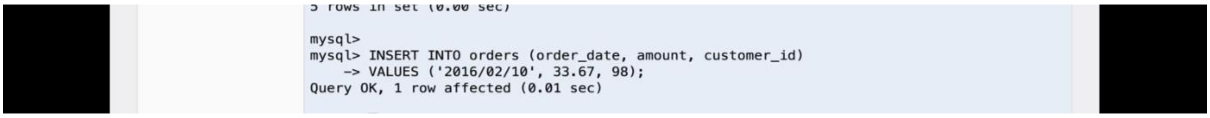
```
CREATE TABLE orders(
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_date DATE,
    amount DECIMAL(8,2),
    customer_id INT
)
```



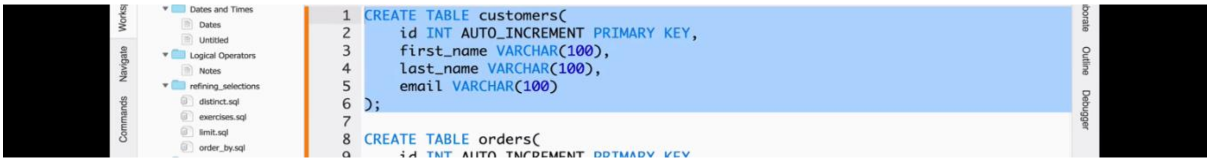
```
INSERT INTO customers (first_name, last_name, email)
VALUES ('Boy', 'George', 'george@gmail.com'),
       ('George', 'Michael', 'gm@gmail.com'),
       ('David', 'Bowie', 'david@gmail.com'),
       ('Blue', 'Steele', 'blue@gmail.com'),
       ('Bette', 'Davis', 'bette@aol.com');
```



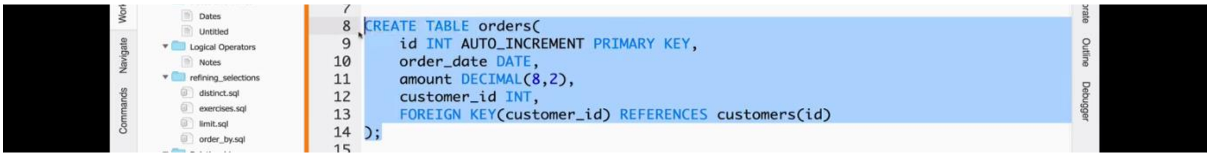
```
INSERT INTO orders (order_date, amount, customer_id)
VALUES ('2016/02/10', 99.99, 1),
       ('2017/11/11', 35.50, 1),
       ('2014/12/12', 800.67, 2),
       ('2015/01/03', 12.50, 2),
       ('1999/04/11', 450.25, 5);
```



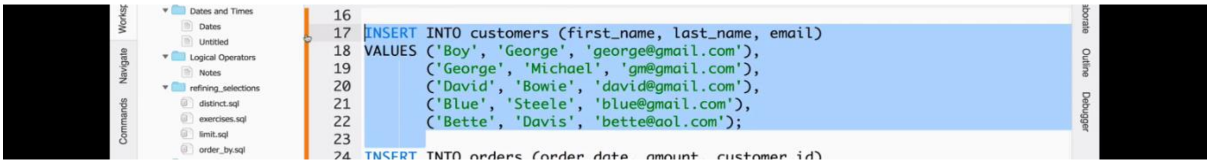
```
5 rows in set (0.00 sec)

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)
```



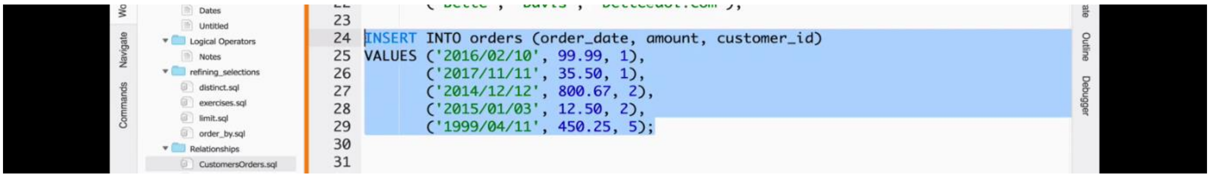
```
CREATE TABLE customers(
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    email VARCHAR(100)
);

CREATE TABLE orders(
    id INT AUTO_INCREMENT PRIMARY KEY
```



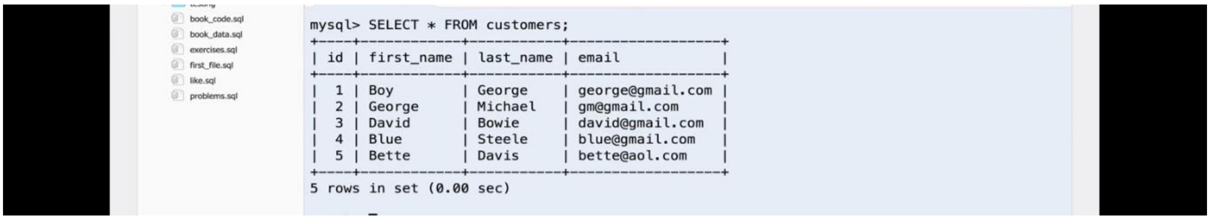
```
CREATE TABLE orders(
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_date DATE,
    amount DECIMAL(8,2),
    customer_id INT,
    FOREIGN KEY(customer_id) REFERENCES customers(id)
);
```



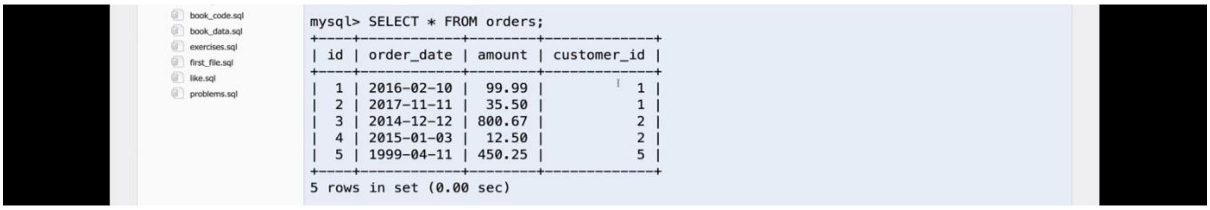
```
INSERT INTO customers (first_name, last_name, email)
VALUES ('Boy', 'George', 'george@gmail.com'),
       ('George', 'Michael', 'gm@gmail.com'),
       ('David', 'Bowie', 'david@gmail.com'),
       ('Blue', 'Steele', 'blue@gmail.com'),
       ('Bette', 'Davis', 'bette@aol.com');

INSERT INTO orders (order_date, amount, customer_id)
```



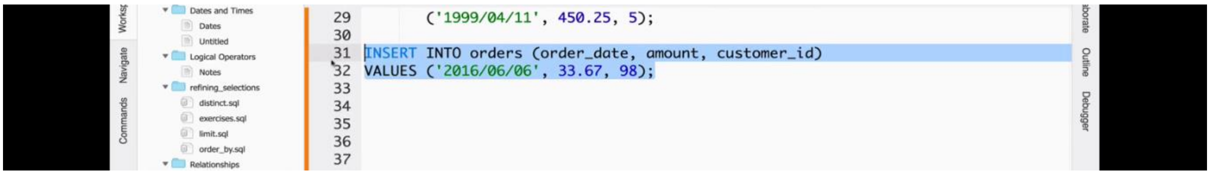
```
INSERT INTO orders (order_date, amount, customer_id)
VALUES ('2016/02/10', 99.99, 1),
       ('2017/11/11', 35.50, 1),
       ('2014/12/12', 800.67, 2),
       ('2015/01/03', 12.50, 2),
       ('1999/04/11', 450.25, 5);
```



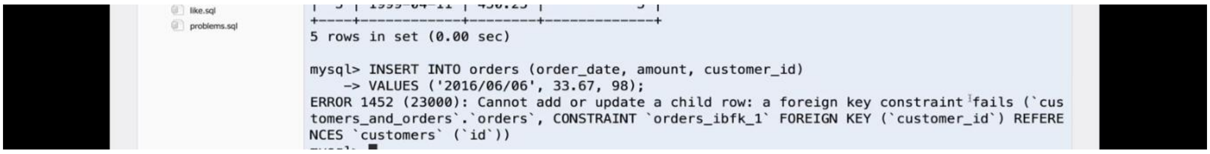
```
mysql> SELECT * FROM customers;
+----+------------+-----------+------------------+
| id | first_name | last_name | email            |
+----+------------+-----------+------------------+
|  1 | Boy        | George    | george@gmail.com |
|  2 | George     | Michael   | gm@gmail.com     |
|  3 | David      | 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)
```

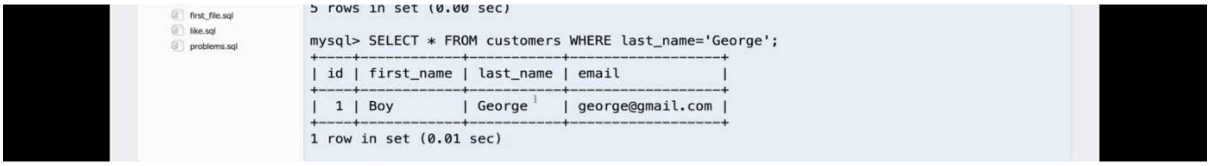


```
       ('1999/04/11', 450.25, 5);

INSERT INTO orders (order_date, amount, customer_id)
VALUES ('2016/06/06', 33.67, 98);
```

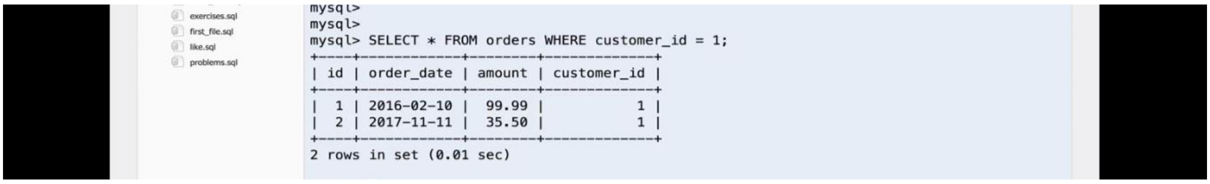


```
+----+------------+--------+-------------+
5 rows in set (0.00 sec)

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 (`cus
tomers_and_orders`.`orders`, CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer_id`) REFERE
NCES `customers` (`id`))
```

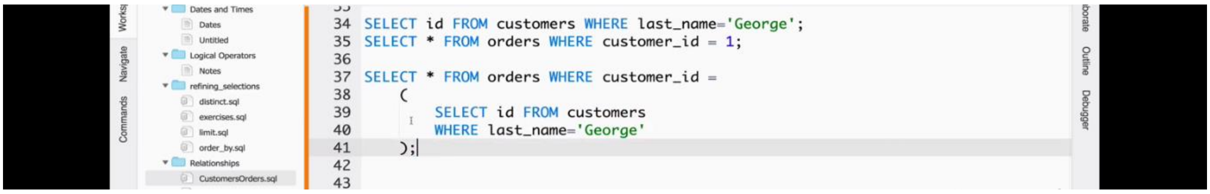
Cross join



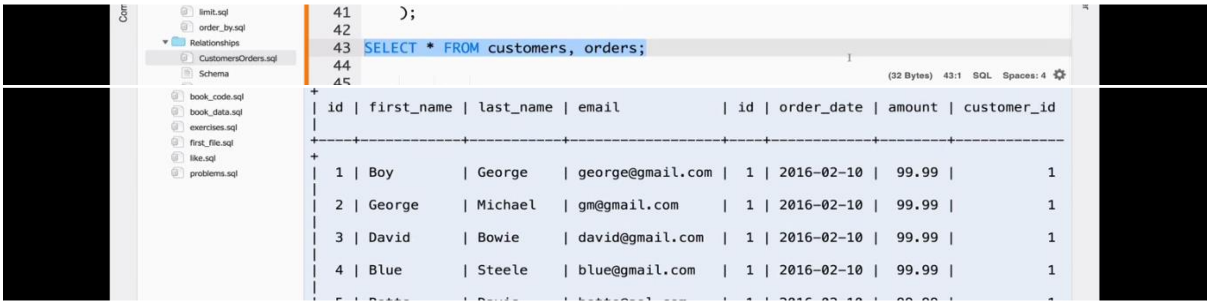
```
5 rows in set (0.00 sec)

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)
```



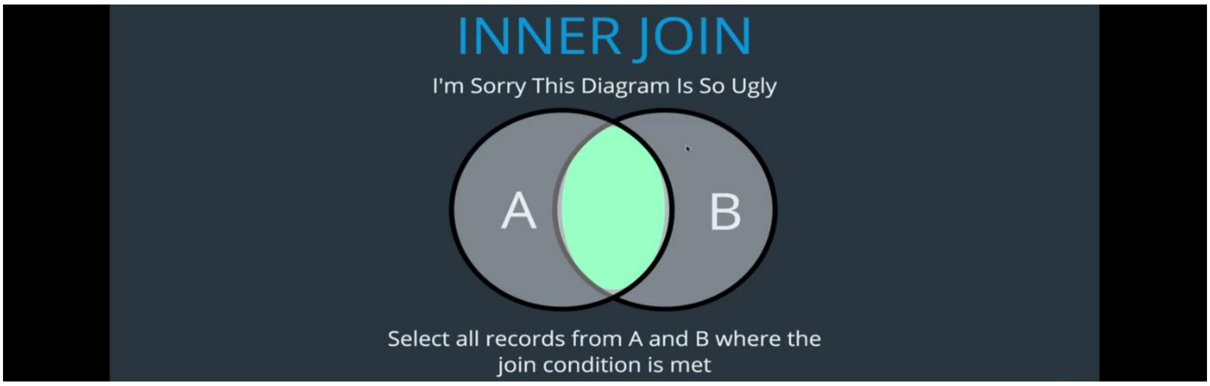
```
SELECT id FROM customers WHERE last_name='George';
SELECT * FROM orders WHERE customer_id = 1;

SELECT * FROM orders WHERE customer_id =
    (
        SELECT id FROM customers
        WHERE last_name='George'
    );
```



```
    );

SELECT * FROM customers, orders;
```

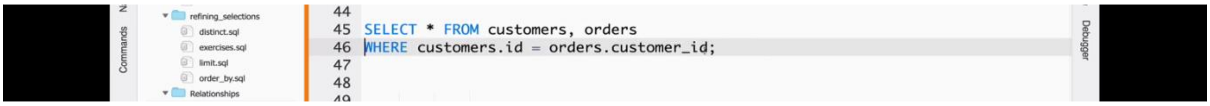
```
+
| id | first_name | last_name | email            | id | order_date | amount | customer_id
|
+----+------------+-----------+------------------+----+------------+--------+------------
+
|  1 | Boy        | George    | george@gmail.com |  1 | 2016-02-10 |  99.99 |           1
|
|  2 | George     | 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)


INNER JOIN

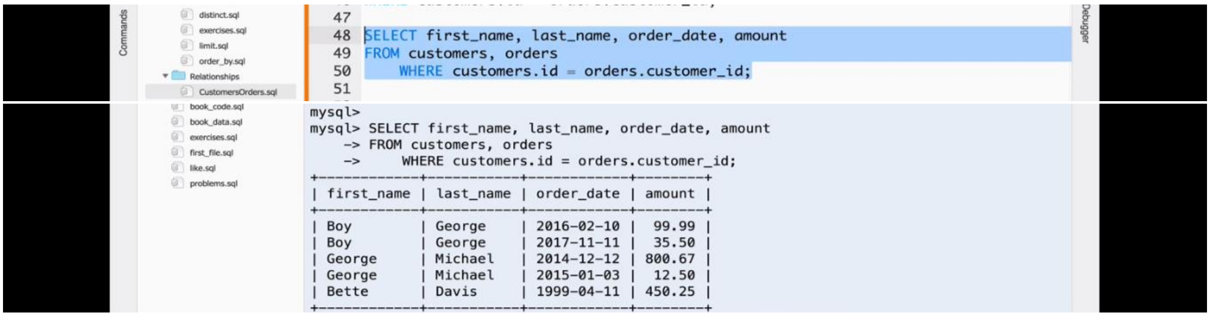
I'm Sorry This Diagram Is So Ugly

Select all records from A and B where the join condition is met



```
SELECT * FROM customers, orders
WHERE customers.id = orders.customer_id;
```

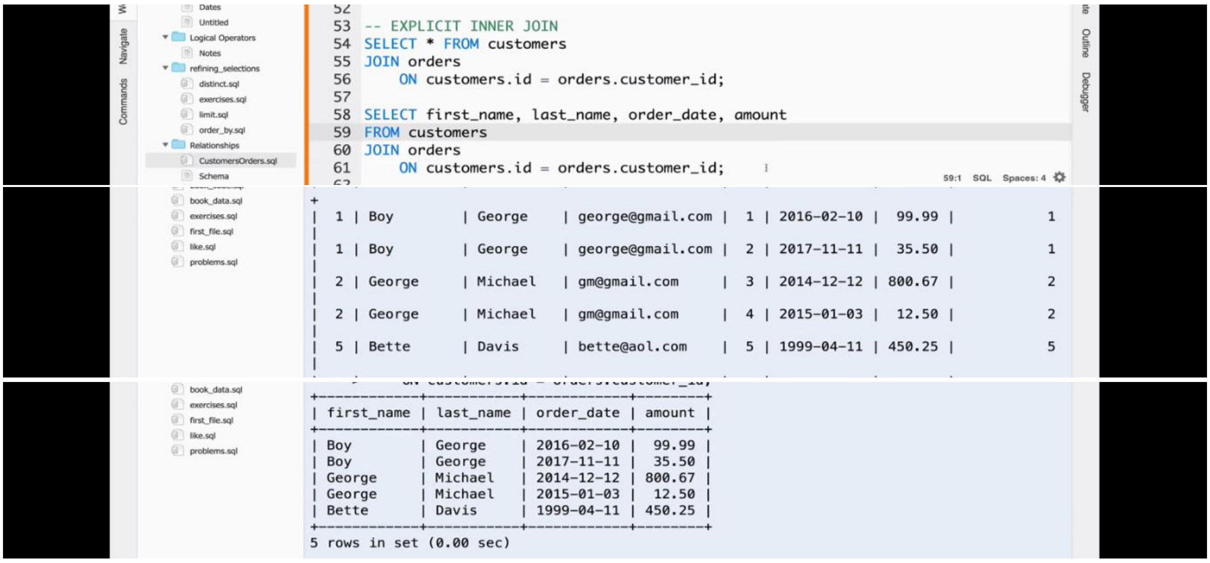
(Inner join) Implicit join



```
SELECT first_name, last_name, order_date, amount
FROM customers, orders
    WHERE customers.id = orders.customer_id;
```

```
mysql>
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 |
+------------+-----------+------------+--------+
```

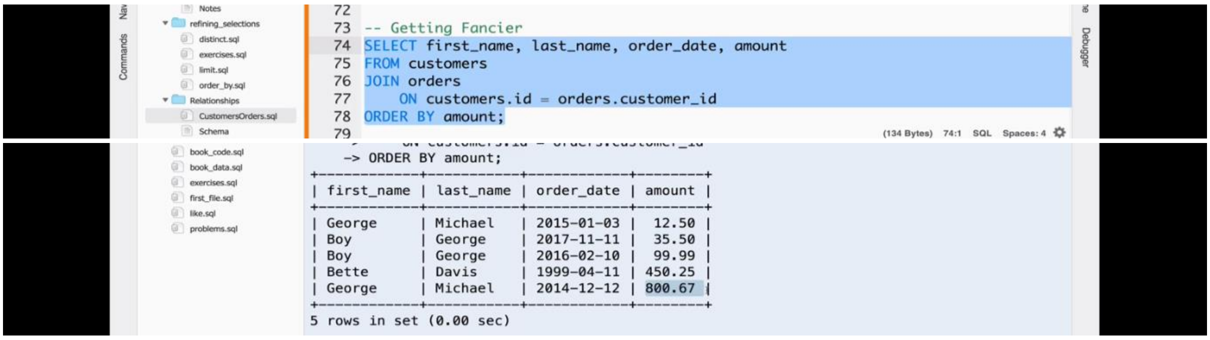
(Inner join) Explicit join (important)



```sql
-- EXPLICIT INNER JOIN
SELECT * FROM customers
JOIN orders
    ON customers.id = orders.customer_id;

SELECT first_name, last_name, order_date, amount
FROM customers
JOIN orders
    ON customers.id = orders.customer_id;
```

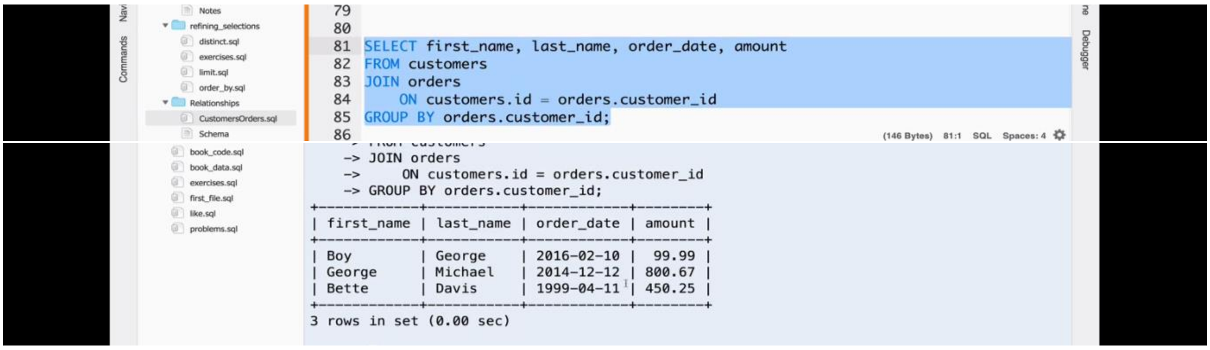
```
+
|  1 | Boy        | George     | george@gmail.com |  1 | 2016-02-10 |  99.99 |           1
|
|  1 | Boy        | George     | george@gmail.com |  2 | 2017-11-11 |  35.50 |           1
|
|  2 | George     | Michael    | gm@gmail.com     |  3 | 2014-12-12 | 800.67 |           2
|
|  2 | George  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

```
+------------+-----------+------------+--------+
| 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)
```



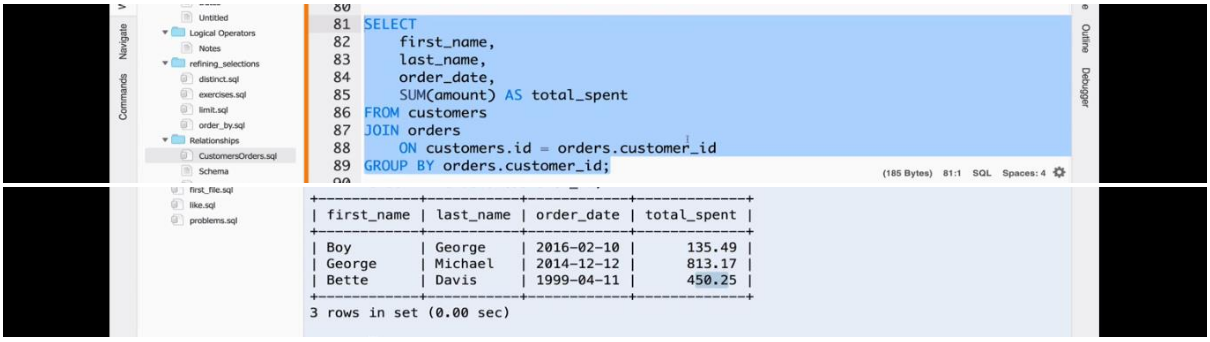
```sql
-- Getting Fancier
SELECT first_name, last_name, order_date, amount
FROM customers
JOIN orders
    ON customers.id = orders.customer_id
ORDER BY amount;
```

```
    -> ORDER BY amount;
+------------+-----------+------------+--------+
| 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)
```



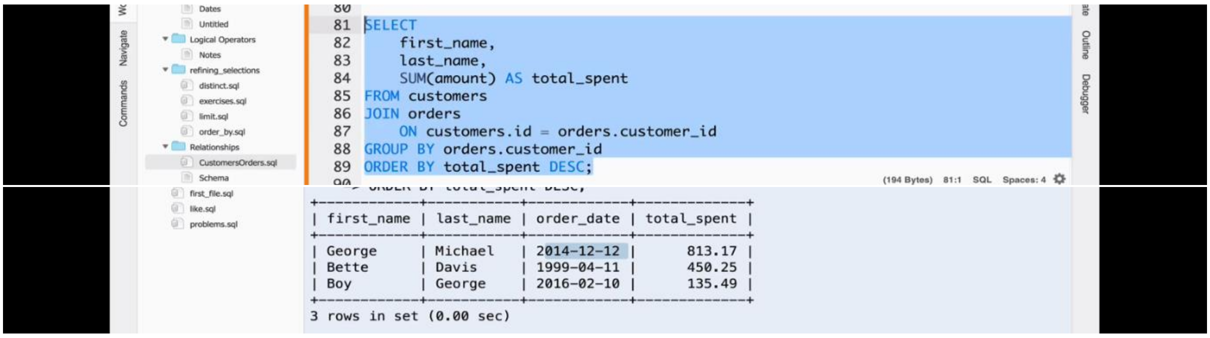
```sql
SELECT first_name, last_name, order_date, amount
FROM customers
JOIN orders
    ON customers.id = orders.customer_id
GROUP BY orders.customer_id;
```

```
    -> 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)
```



```sql
SELECT
    first_name,
    last_name,
    order_date,
    SUM(amount) AS total_spent
FROM customers
JOIN orders
    ON customers.id = orders.customer_id
GROUP BY orders.customer_id;
```

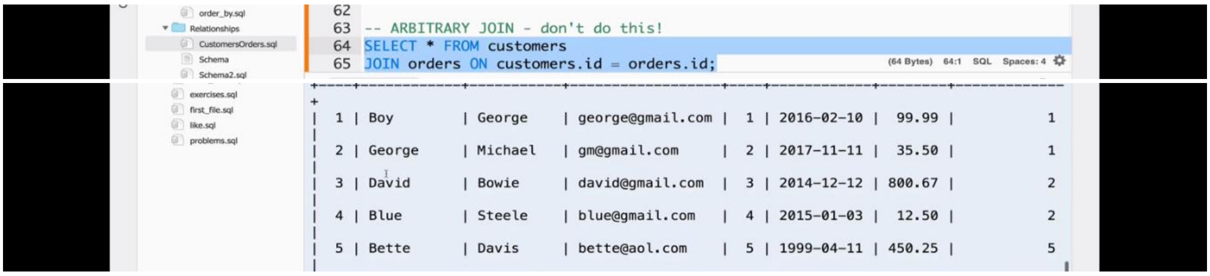
```
+------------+-----------+------------+-------------+
| first_name | last_name | order_date | total_spent |
+------------+-----------+------------+-------------+
| Boy        | George    | 2016-02-10 |      135.49 |
| George     | Michael   | 2014-12-12 |      813.17 |
| Bette      | Davis     | 1999-04-11 |      450.25 |
+------------+-----------+------------+-------------+
3 rows in set (0.00 sec)
```



```sql
SELECT
    first_name,
    last_name,
    SUM(amount) AS total_spent
FROM customers
JOIN orders
    ON customers.id = orders.customer_id
GROUP BY orders.customer_id
ORDER BY total_spent DESC;
```

```
+------------+-----------+------------+-------------+
| first_name | last_name | order_date | total_spent |
+------------+-----------+------------+-------------+
| George     | Michael   | 2014-12-12 |      813.17 |
| Bette      | Davis     | 1999-04-11 |      450.25 |
| Boy        | George    | 2016-02-10 |      135.49 |
+------------+-----------+------------+-------------+
3 rows in set (0.00 sec)
```

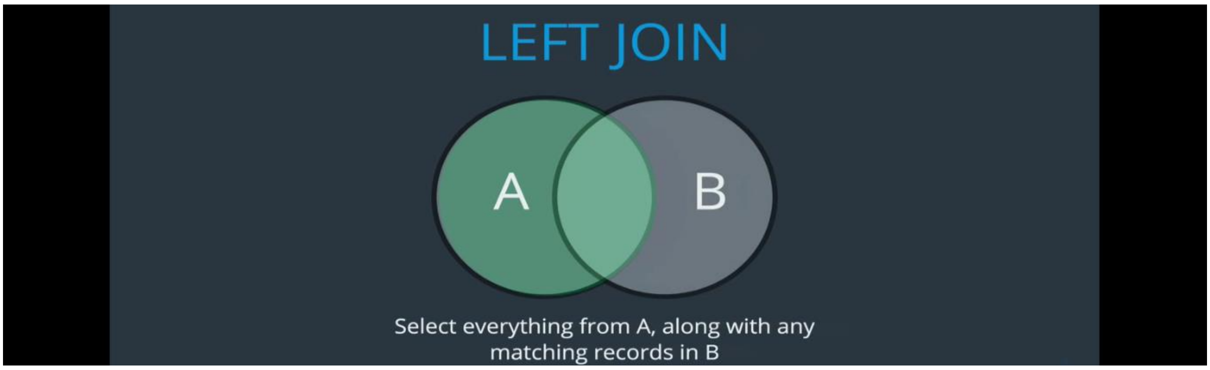
(inner join) arbitrary join



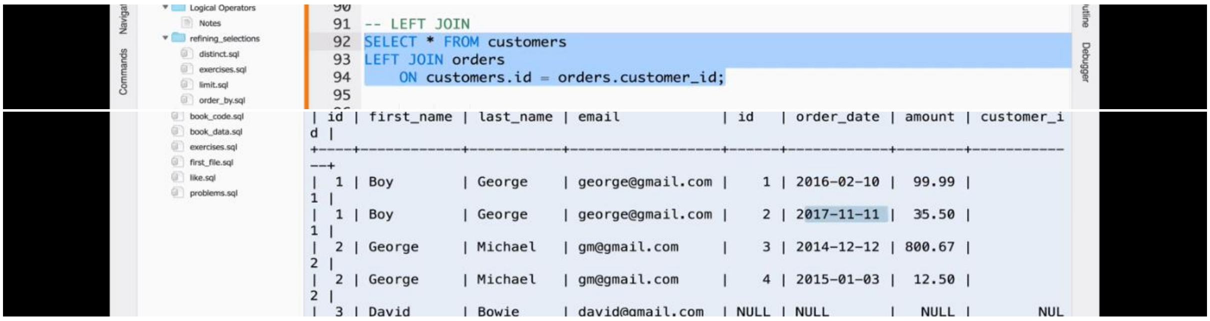
```sql
-- ARBITRARY JOIN - don't do this!
SELECT * FROM customers
JOIN orders ON customers.id = orders.id;
```

```
+----+------------+-----------+------------------+----+------------+--------+-------------
+
|  1 | Boy        | George    | george@gmail.com |  1 | 2016-02-10 |  99.99 |            1
|
|  2 | George     | Michael   | gm@gmail.com     |  2 | 2017-11-11 |  35.50 |            1
|
|  3 | David      | Bowie     | david@gmail.com  |  3 | 2014-12-12 | 800.67 |            2
|
|  4 | Blue       | Steele    | blue@gmail.com   |  4 | 2015-01-03 |  12.50 |            2
|
|  5 | Bette      | Davis     | bette@aol.com    |  5 | 1999-04-11 | 450.25 |            5
|
```

Left join (take highest and null spend user to send copne)

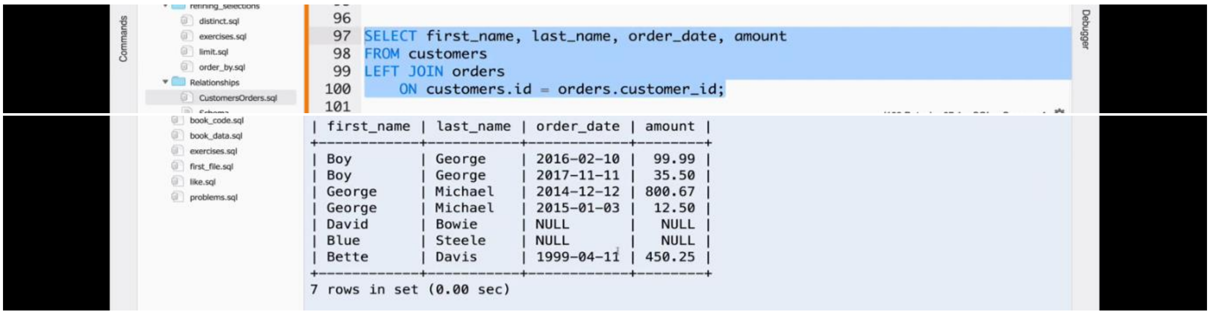

LEFT JOIN

Select everything from A, along with any matching records in B



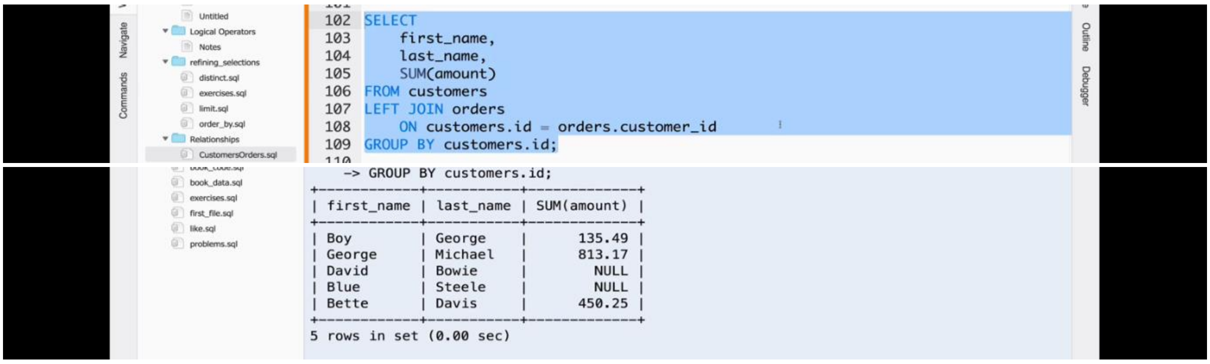
```sql
-- LEFT JOIN
SELECT * FROM customers
LEFT JOIN orders
    ON customers.id = orders.customer_id;
```

```
| id | first_name | last_name | email            | id   | order_date | amount | customer_i
d |
+----+------------+-----------+------------------+------+------------+--------+------------
--+
|  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 |
|  3 | David      | Bowie     | david@gmail.com  | NULL | NULL       |   NULL |         NUL
```



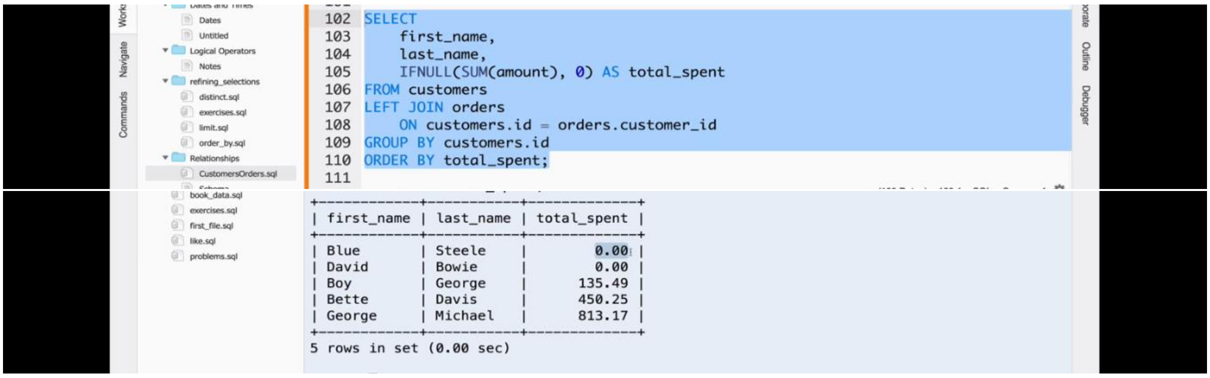
```sql
SELECT first_name, last_name, order_date, amount
FROM customers
LEFT JOIN orders
    ON customers.id = orders.customer_id;
```

```
| 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 |
| David      | Bowie     | NULL       |   NULL |
| Blue       | Steele    | NULL       |   NULL |
| Bette      | Davis     | 1999-04-11 | 450.25 |
+------------+-----------+------------+--------+
7 rows in set (0.00 sec)
```



```sql
SELECT
    first_name,
    last_name,
    SUM(amount)
FROM customers
LEFT JOIN orders
    ON customers.id = orders.customer_id
GROUP BY customers.id;
```

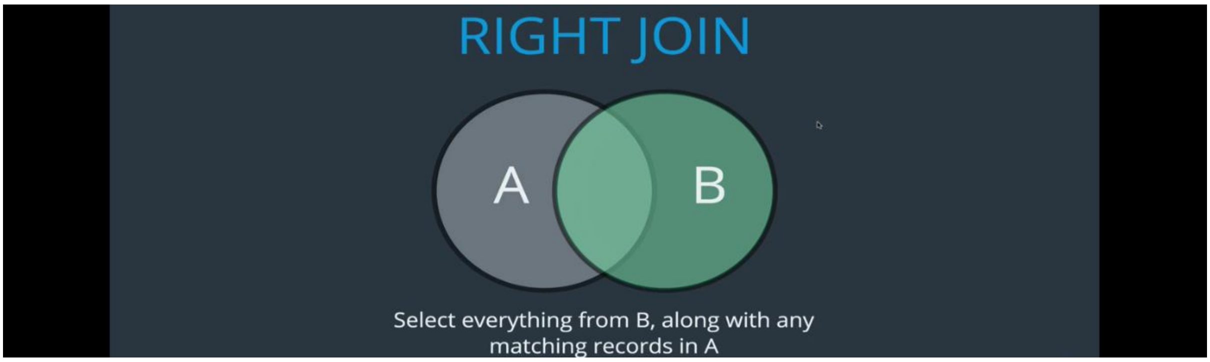
```
    -> GROUP BY customers.id;
+------------+-----------+-------------+
| first_name | last_name | SUM(amount) |
+------------+-----------+-------------+
| Boy        | George    |      135.49 |
| George     | Michael   |      813.17 |
| David      | Bowie     |        NULL |
| Blue       | Steele    |        NULL |
| Bette      | Davis     |      450.25 |
+------------+-----------+-------------+
5 rows in set (0.00 sec)
```



```sql
SELECT
    first_name,
    last_name,
    IFNULL(SUM(amount), 0) AS total_spent
FROM customers
LEFT JOIN orders
    ON customers.id = orders.customer_id
GROUP BY customers.id
ORDER BY total_spent;
```

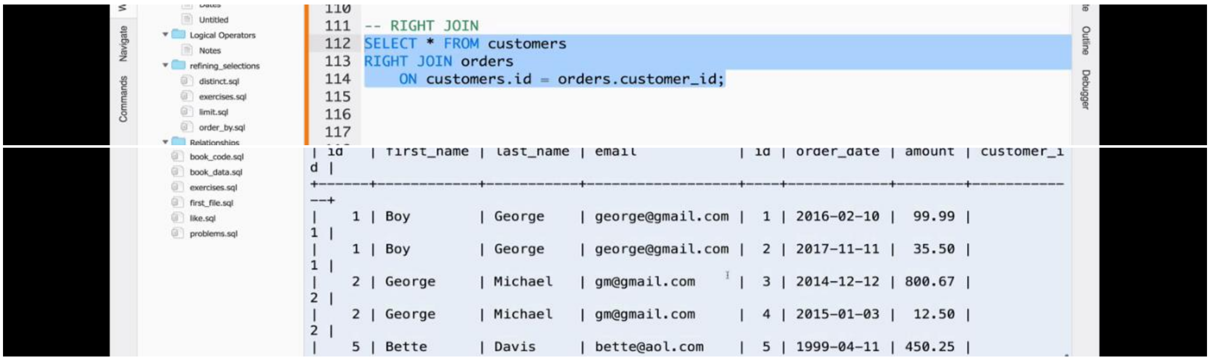
```
+------------+-----------+-------------+
| first_name | last_name | total_spent |
+------------+-----------+-------------+
| Blue       | Steele    |        0.00 |
| David      | Bowie     |        0.00 |
| Boy        | George    |      135.49 |
| Bette      | Davis     |      450.25 |
| George     | Michael   |      813.17 |
+------------+-----------+-------------+
5 rows in set (0.00 sec)
```

Right join (help to find wrong id from left)



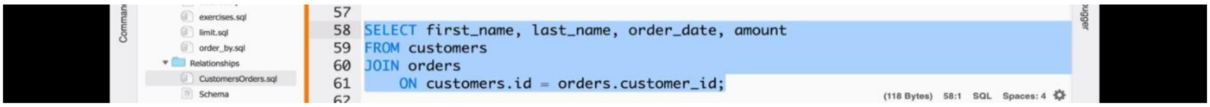
RIGHT JOIN

Select everything from B, along with any matching records in A

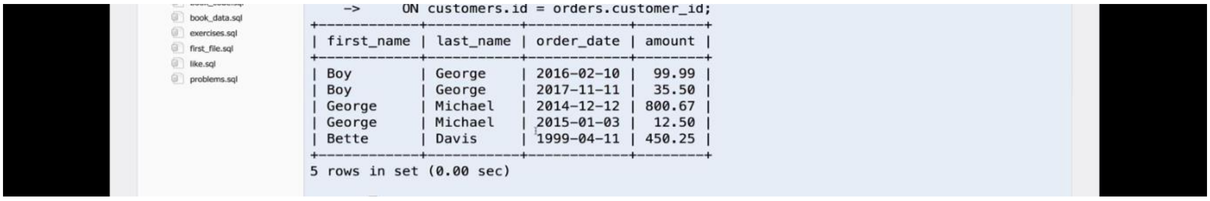


```sql
-- RIGHT JOIN
SELECT * FROM customers
RIGHT JOIN orders
    ON customers.id = orders.customer_id;
```

```
| id   | first_name | last_name | email            | id | order_date | amount | customer_i
d |
+------+------------+-----------+------------------+----+------------+--------+-----------
--+
|    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 |
```

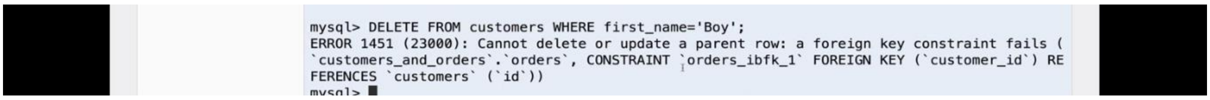


```sql
SELECT first_name, last_name, order_date, amount
FROM customers
JOIN orders
    ON customers.id = orders.customer_id;
```

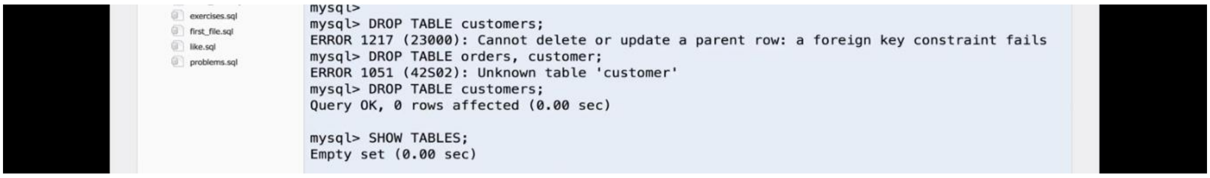


```
    -> ON customers.id = orders.customer_id;
+------------+-----------+------------+--------+
| first_name | last_name | order_date | amount |
+------------+-----------+------------+--------+
| Boy        | George    | 2016-02-10 |  99.99 |
| Boy        | George    | 2017-11-11 |  35.50 |
| George     | 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> 
```



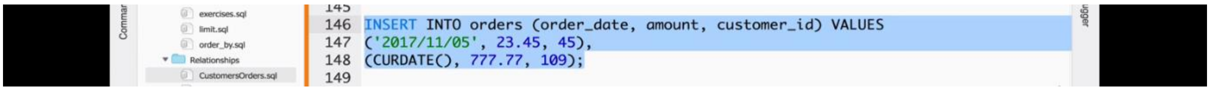
```
mysql>
mysql> DROP TABLE customers;
ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constraint fails
mysql> DROP TABLE orders, customer;
ERROR 1051 (42S02): Unknown table 'customer'
mysql> DROP TABLE customers;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW TABLES;
Empty set (0.00 sec)
```

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

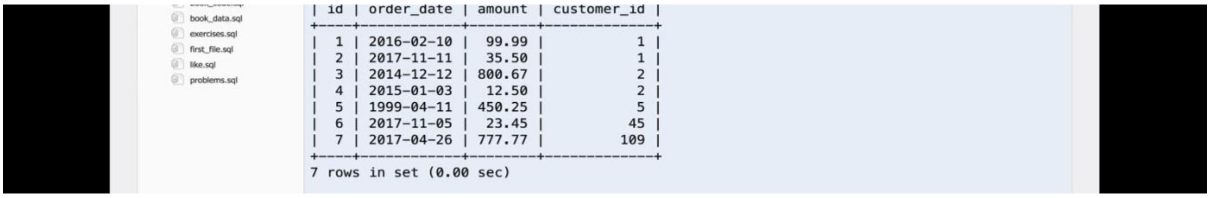
1-Delete customer and order table

2- create customer and order table again

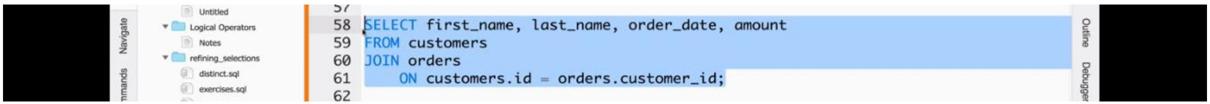
3- and add this data



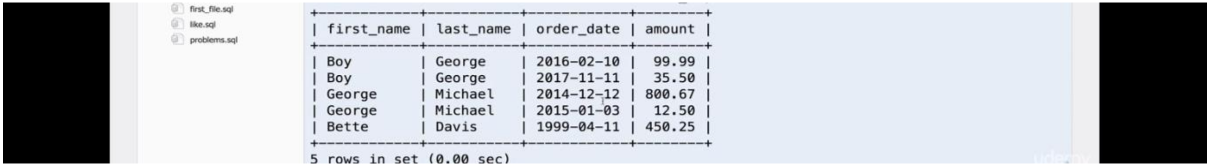
```sql
INSERT INTO orders (order_date, amount, customer_id) VALUES
('2017/11/05', 23.45, 45),
(CURDATE(), 777.77, 109);
```



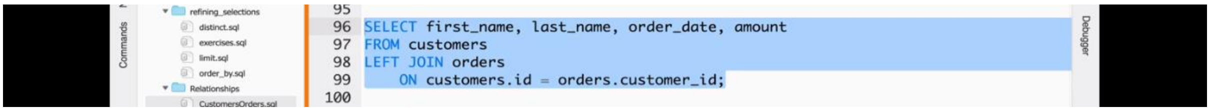
```
| id | order_date | amount | customer_id |
+----+------------+--------+-------------+
|  1 | 2016-02-10 |  99.99 |           1 |
|  2 | 2017-11-11 |  35.50 |           1 |
|  3 | 2014-12-12 | 800.67 |           2 |
|  4 | 2015-01-03 |  12.50 |           2 |
|  5 | 1999-04-11 | 450.25 |           5 |
|  6 | 2017-11-05 |  23.45 |          45 |
|  7 | 2017-04-26 | 777.77 |         109 |
+----+------------+--------+-------------+
7 rows in set (0.00 sec)
```



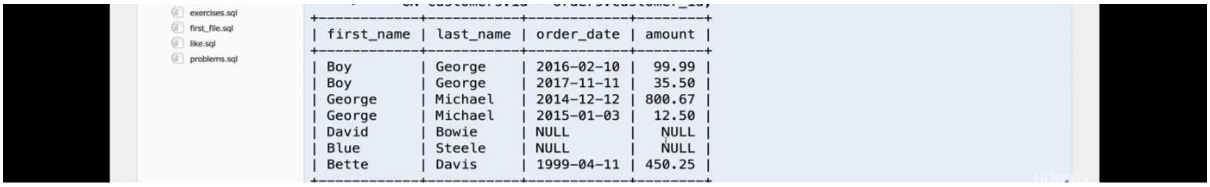
```sql
SELECT first_name, last_name, order_date, amount
FROM customers
JOIN orders
    ON customers.id = orders.customer_id;
```



```
+------------+-----------+------------+--------+
| 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)
```

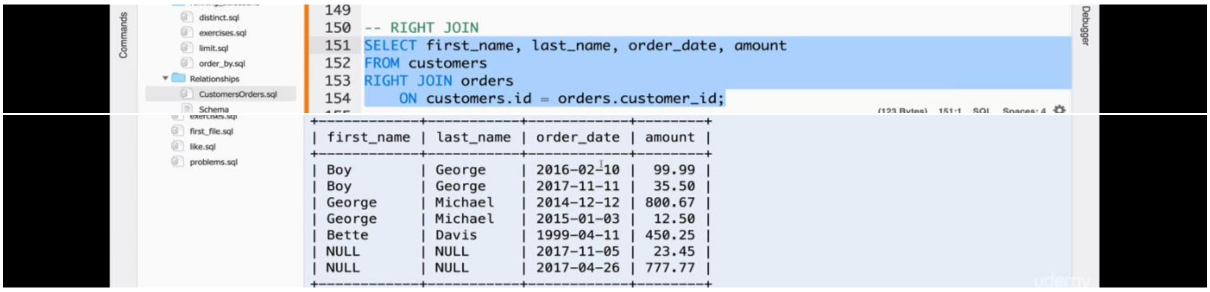


```sql
SELECT first_name, last_name, order_date, amount
FROM customers
LEFT JOIN orders
    ON customers.id = orders.customer_id;
```



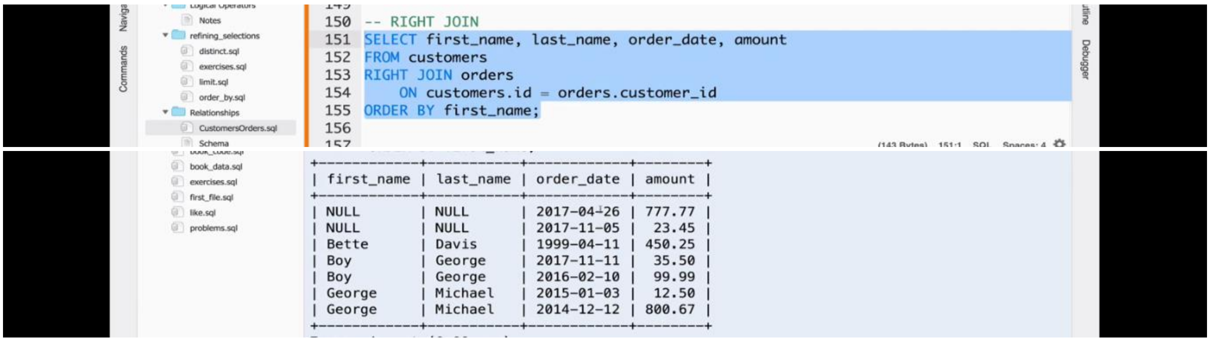
```
    -> ON customers.id = orders.customer_id;
+------------+-----------+------------+--------+
| first_name | last_name | order_date | amount |
+------------+-----------+------------+--------+
| Boy        | George    | 2016-02-10 |  99.99 |
| Boy        | George    | 2017-11-11 |  35.50 |
| George     | Michael   | 2014-12-12 | 800.67 |
| George     | Michael   | 2015-01-03 |  12.50 |
| David      | Bowie     | NULL       |   NULL |
| Blue       | Steele    | NULL       |   NULL |
| Bette      | Davis     | 1999-04-11 | 450.25 |
+------------+-----------+------------+--------+
```

Right join main



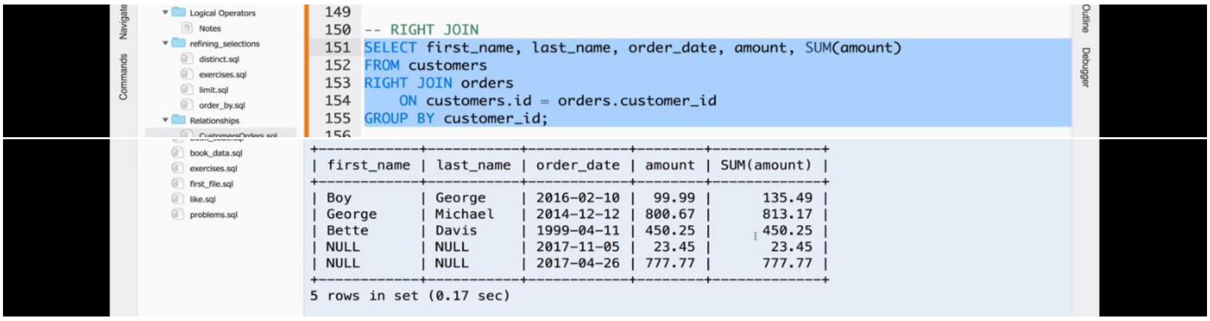
```
-- RIGHT JOIN
SELECT first_name, last_name, order_date, amount
FROM customers
RIGHT JOIN orders
    ON customers.id = orders.customer_id;
```

```
+------------+-----------+------------+--------+
| 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 |
+------------+-----------+------------+--------+
```



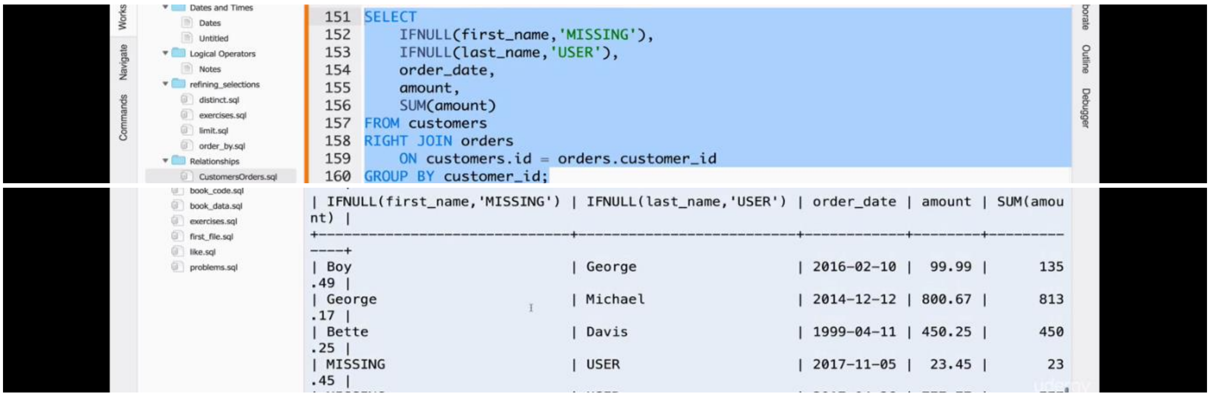
```
-- RIGHT JOIN
SELECT first_name, last_name, order_date, amount
FROM customers
RIGHT JOIN orders
    ON customers.id = orders.customer_id
ORDER BY first_name;
```

```
+------------+-----------+------------+--------+
| 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 |
+------------+-----------+------------+--------+
```



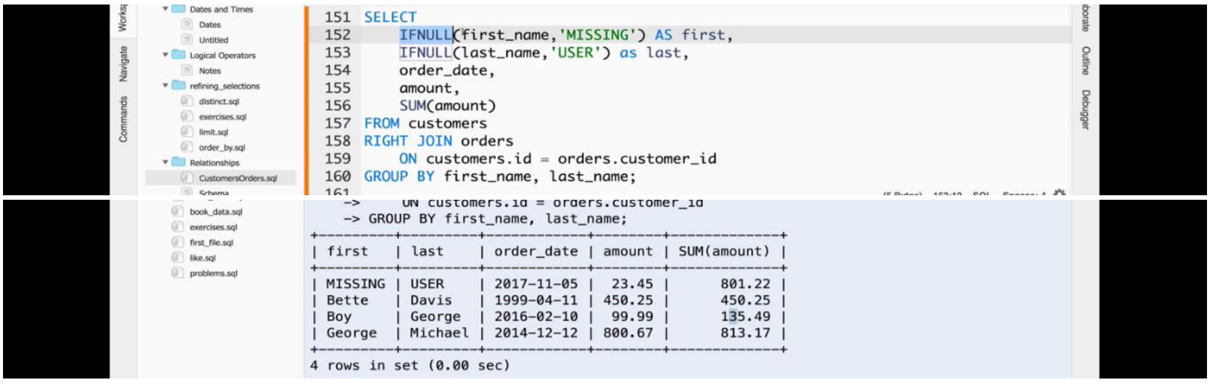
```
-- RIGHT JOIN
SELECT first_name, last_name, order_date, amount, SUM(amount)
FROM customers
RIGHT JOIN orders
    ON customers.id = orders.customer_id
GROUP BY customer_id;
```

```
+------------+-----------+------------+--------+-------------+
| 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)
```



```
SELECT
    IFNULL(first_name,'MISSING'),
    IFNULL(last_name,'USER'),
    order_date,
    amount,
    SUM(amount)
FROM customers
RIGHT JOIN orders
    ON customers.id = orders.customer_id
GROUP BY customer_id;
```

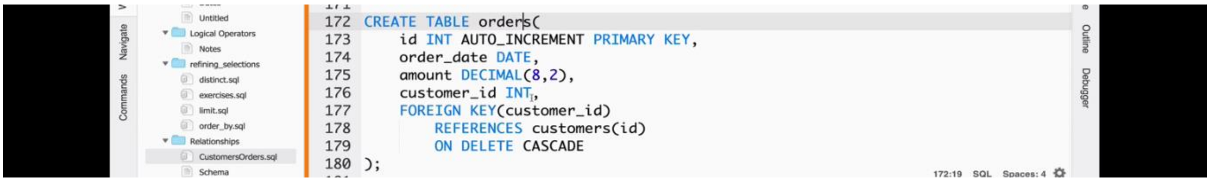
```
| IFNULL(first_name,'MISSING') | IFNULL(last_name,'USER') | order_date | amount | SUM(amou
nt) |
+------------------------------+--------------------------+------------+--------+---------
----+
| Boy                          | George                   | 2016-02-10 |  99.99 |      135
.49 |
| George                       | 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 |
```



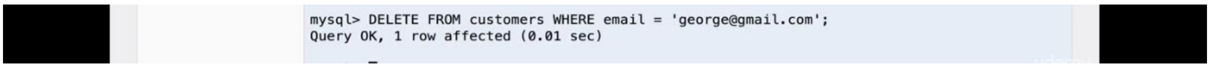
```
SELECT
    IFNULL(first_name,'MISSING') AS first,
    IFNULL(last_name,'USER') as last,
    order_date,
    amount,
    SUM(amount)
FROM customers
RIGHT JOIN orders
    ON customers.id = orders.customer_id
GROUP BY first_name, last_name;
```

```
    -> ON customers.id = orders.customer_id
    -> 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)
```

On delete cascade

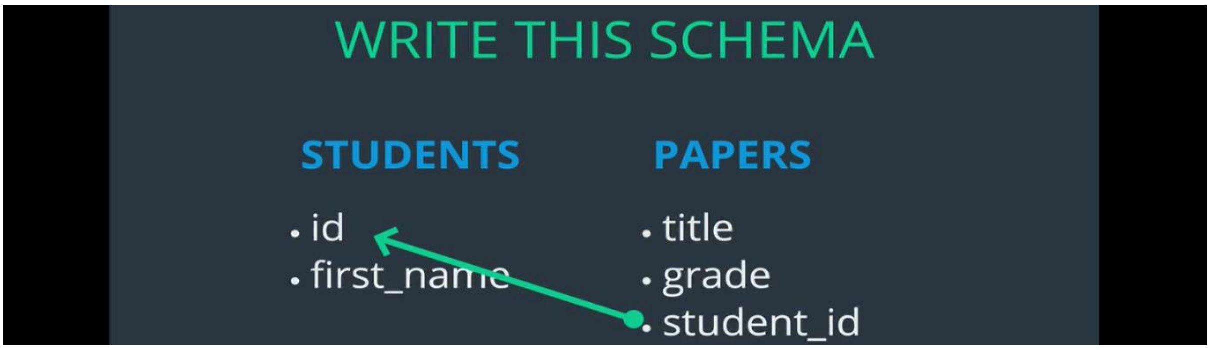


```
CREATE TABLE orders(
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_date DATE,
    amount DECIMAL(8,2),
    customer_id INT,
    FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE CASCADE
);
```



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

Exercises



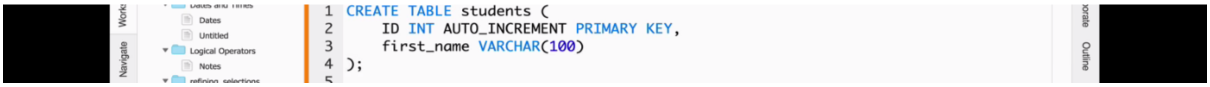
WRITE THIS SCHEMA

STUDENTS

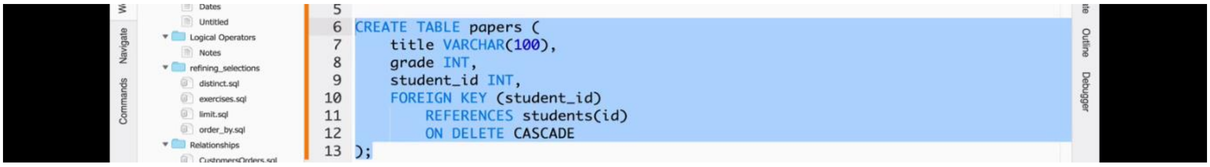
- id
- first_name

PAPERS

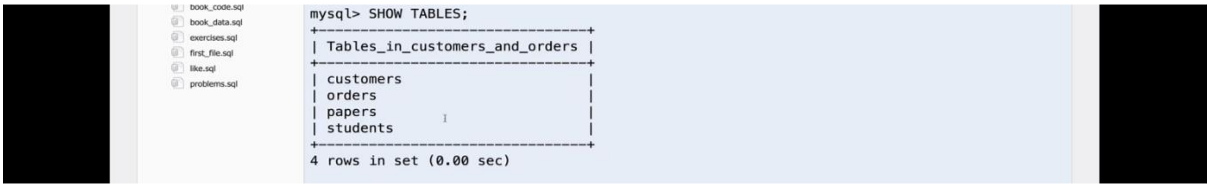
- title
- grade
- student_id



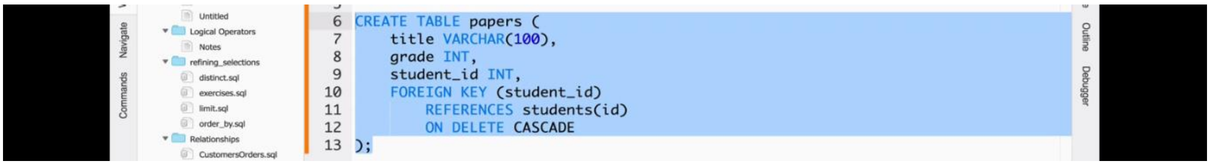
```
CREATE TABLE students (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100)
);
```



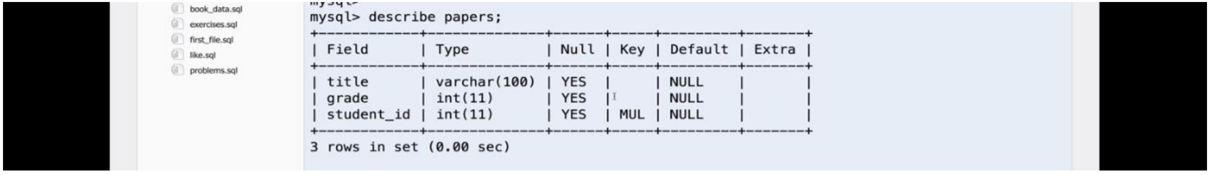
```
CREATE TABLE papers (
    title VARCHAR(100),
    grade INT,
    student_id INT,
    FOREIGN KEY (student_id)
        REFERENCES students(id)
        ON DELETE CASCADE
);
```



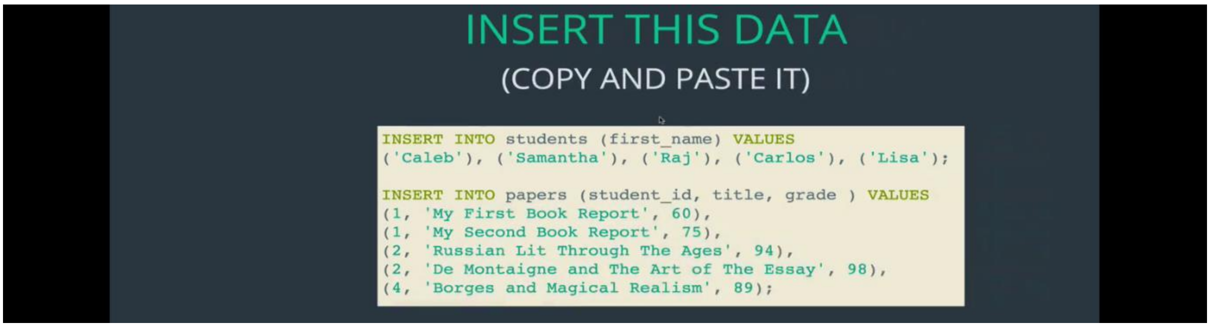
```
mysql> SHOW TABLES;
+-------------------------------+
| Tables_in_customers_and_orders |
+-------------------------------+
| customers                     |
| orders                        |
| papers                        |
| students                      |
+-------------------------------+
4 rows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



```
CREATE TABLE papers (
    title VARCHAR(100),
    grade INT,
    student_id INT,
    FOREIGN KEY (student_id)
        REFERENCES students(id)
        ON DELETE CASCADE
);
```



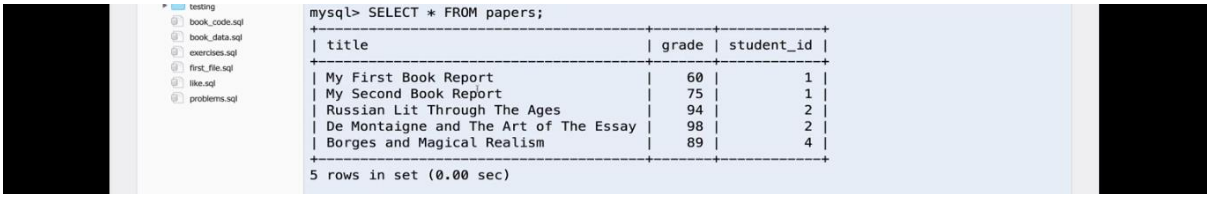
```
mysql> describe papers;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| title      | varchar(100) | YES  |     | NULL    |       |
| grade      | int(11)      |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



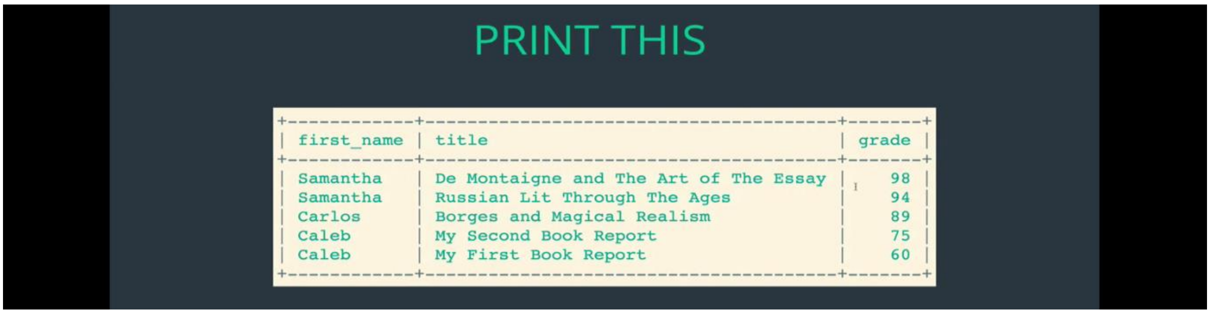
INSERT THIS DATA

(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);
```

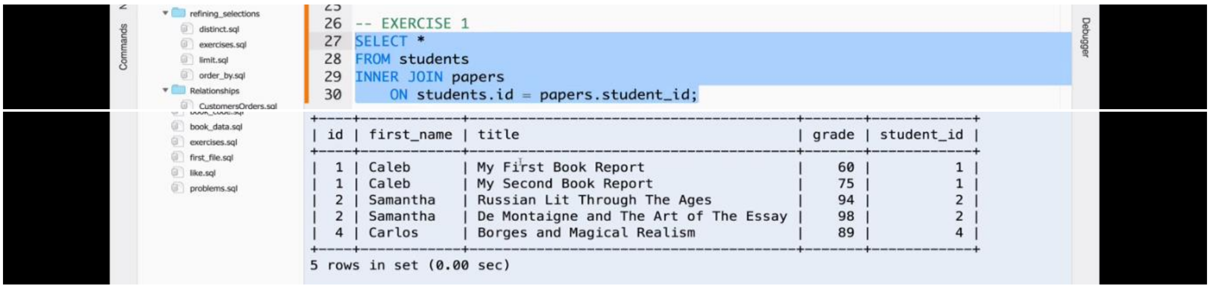


```
mysql> SELECT * FROM papers;
+---------------------------------------+-------+------------+
| title                                 | grade | student_id |
+---------------------------------------+-------+------------+
| My First Book Report                  |    60 |          1 |
| My Second Book 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)
```



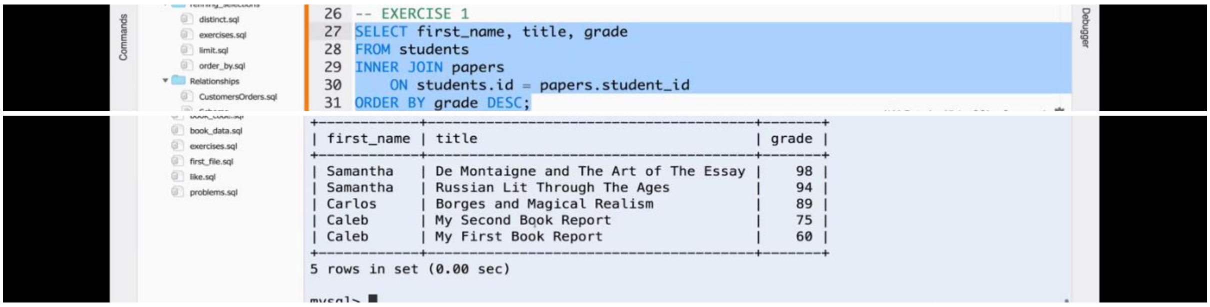
PRINT THIS

```
+------------+---------------------------------------+-------+
| 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 |
+------------+---------------------------------------+-------+
```



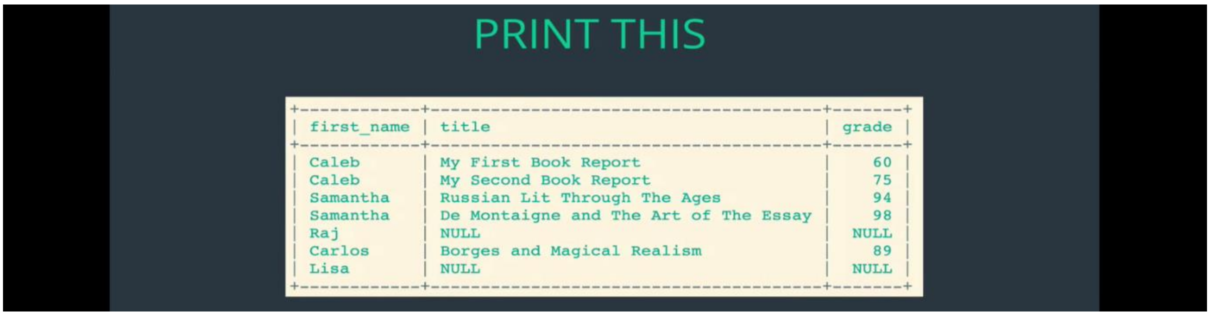
```
-- EXERCISE 1
SELECT *
FROM students
INNER JOIN papers
    ON students.id = papers.student_id;
```

```
+----+------------+---------------------------------------+-------+------------+
| 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)
```



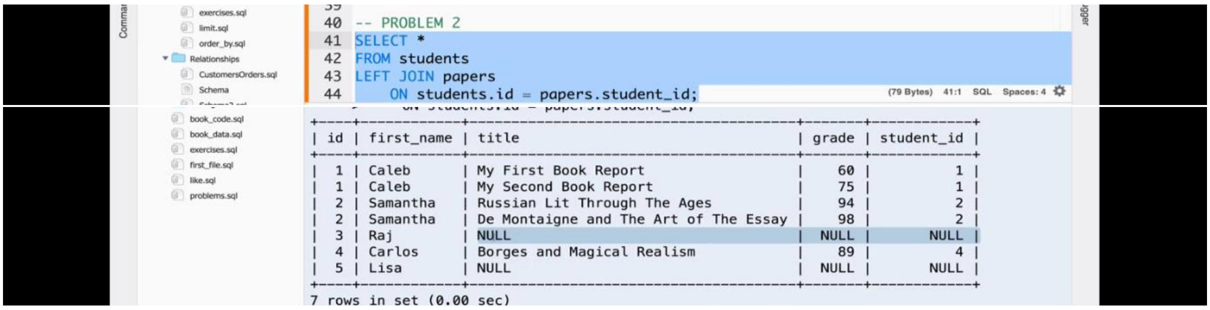
```
-- EXERCISE 1
SELECT first_name, title, grade
FROM students
INNER JOIN papers
    ON students.id = papers.student_id
ORDER BY grade DESC;
```

```
+------------+---------------------------------------+-------+
| 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)

mysql>
```



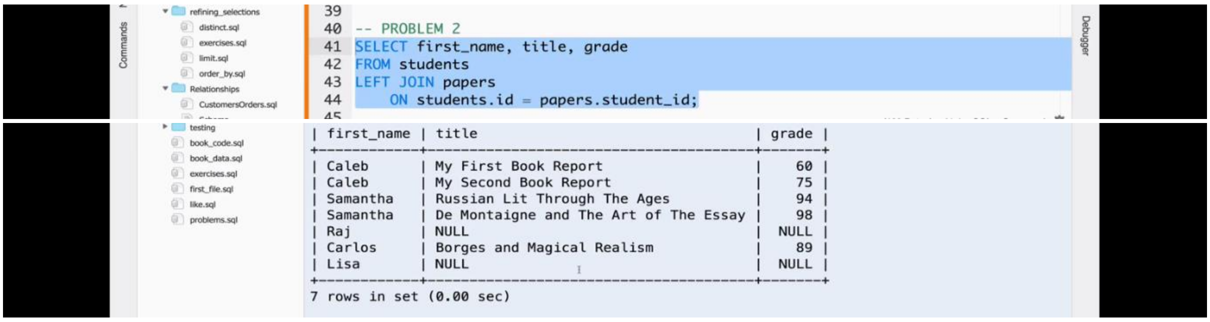
PRINT THIS

```
+------------+---------------------------------------+-------+
| 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 |
+------------+---------------------------------------+-------+
```



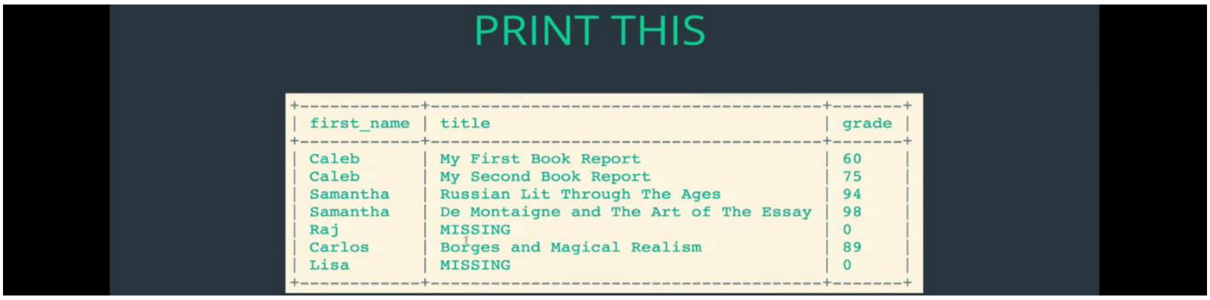
```sql
-- PROBLEM 2
SELECT *
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id;
```

```
+----+------------+---------------------------------------+-------+------------+
| 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)
```



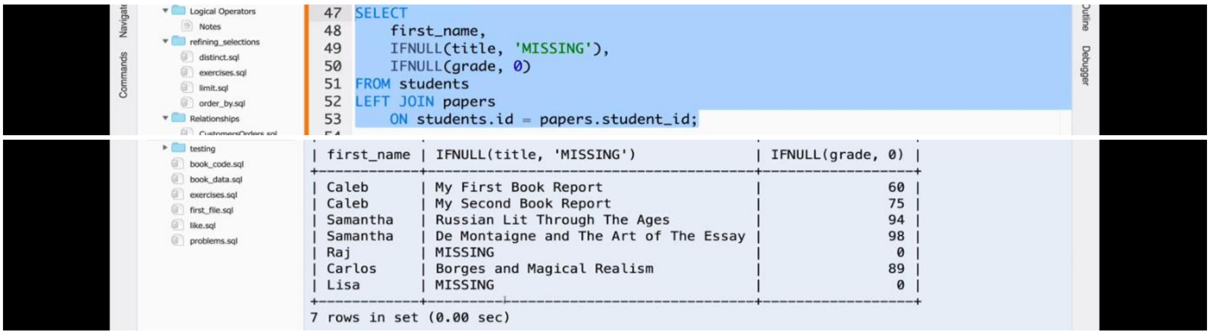
```sql
-- PROBLEM 2
SELECT first_name, title, grade
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id;
```

```
| 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)
```



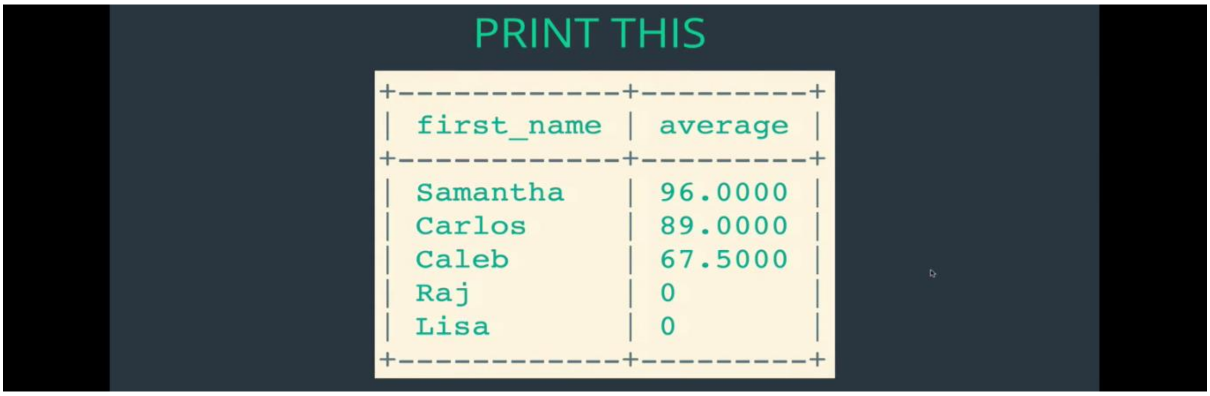
PRINT THIS

```
+------------+---------------------------------------+-------+
| 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     |
+------------+---------------------------------------+-------+
```



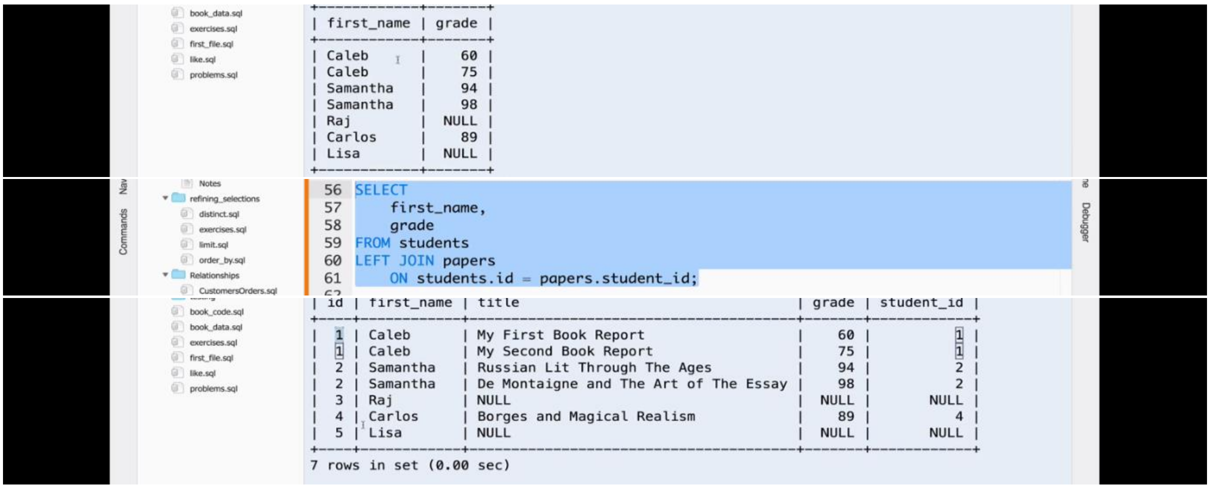
```sql
SELECT
    first_name,
    IFNULL(title, 'MISSING'),
    IFNULL(grade, 0)
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id;
```

```
| 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)
```



PRINT THIS

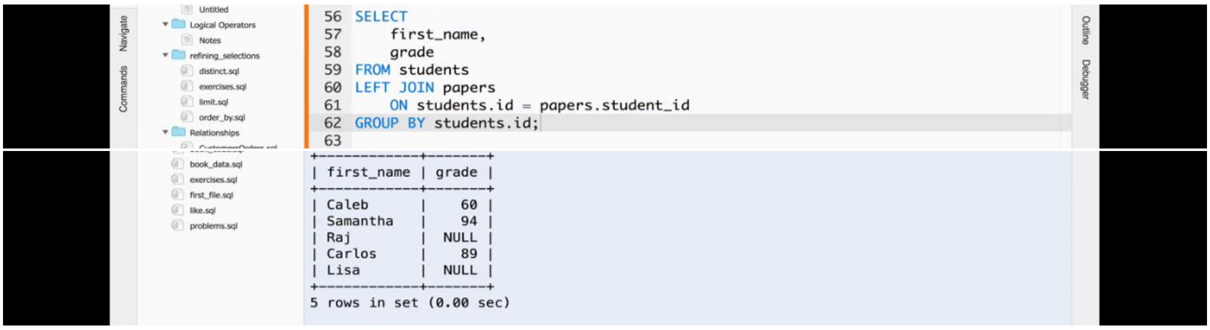
```
+------------+---------+
| first_name | average |
+------------+---------+
| Samantha   | 96.0000 |
| Carlos     | 89.0000 |
| Caleb      | 67.5000 |
| Raj        | 0       |
| Lisa       | 0       |
+------------+---------+
```



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

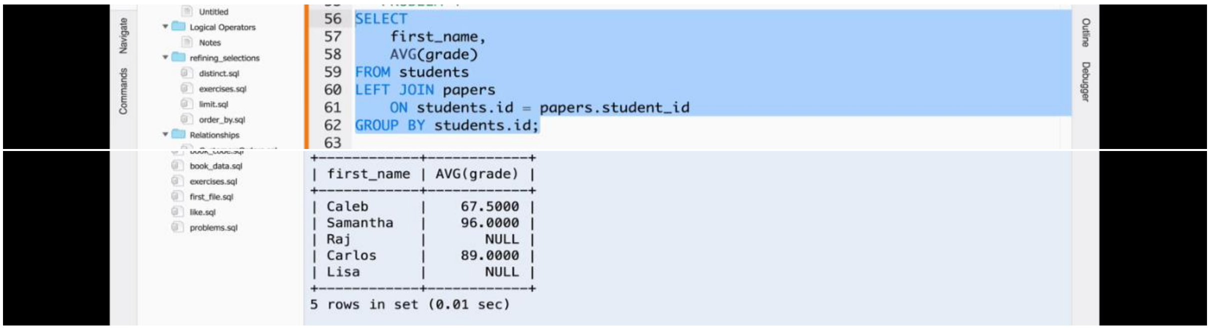
```
SELECT
    first_name,
    grade
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id;
```

```
| 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)
```



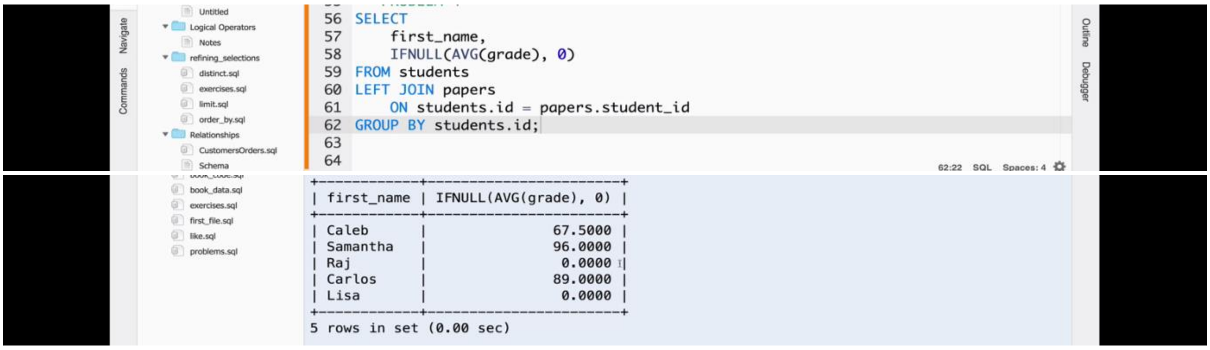
```
SELECT
    first_name,
    grade
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id
GROUP BY students.id;
```

```
+------------+-------+
| first_name | grade |
+------------+-------+
| Caleb      |    60 |
| Samantha   |    94 |
| Raj        |  NULL |
| Carlos     |    89 |
| Lisa       |  NULL |
+------------+-------+
5 rows in set (0.00 sec)
```



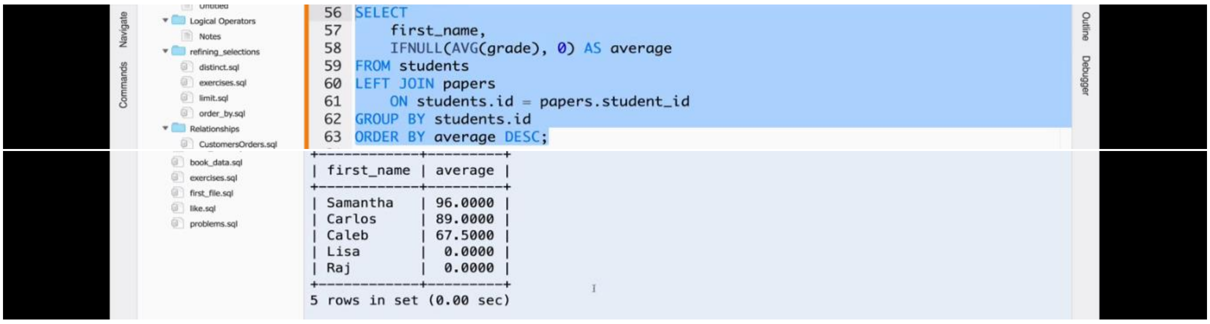
```
SELECT
    first_name,
    AVG(grade)
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id
GROUP BY students.id;
```

```
+------------+------------+
| first_name | AVG(grade) |
+------------+------------+
| Caleb      |    67.5000 |
| Samantha   |    96.0000 |
| Raj        |       NULL |
| Carlos     |    89.0000 |
| Lisa       |       NULL |
+------------+------------+
5 rows in set (0.01 sec)
```



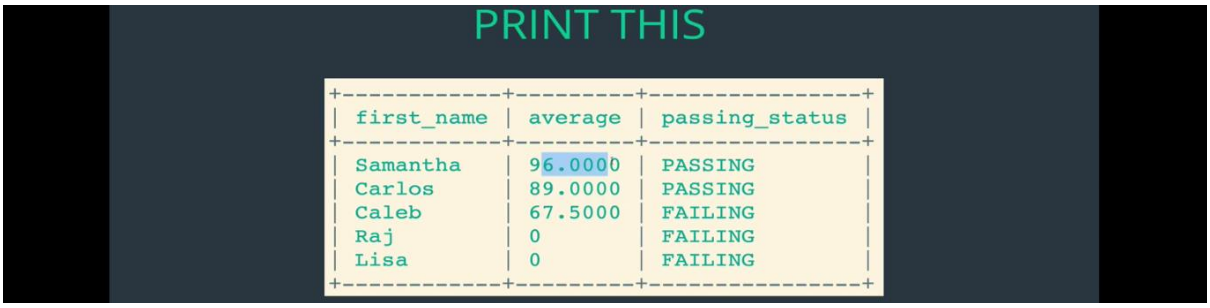
```
SELECT
    first_name,
    IFNULL(AVG(grade), 0)
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id
GROUP BY students.id;
```

```
+------------+-----------------------+
| 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)
```



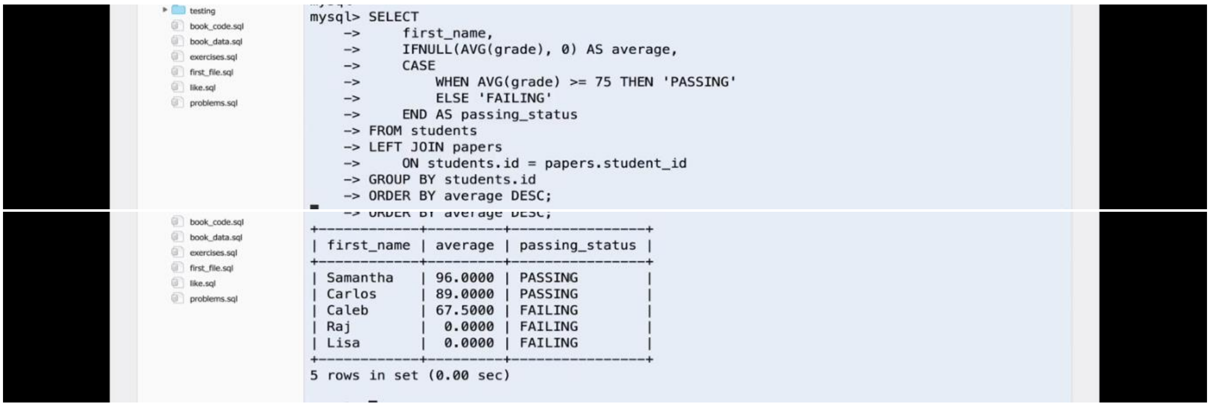
```
SELECT
    first_name,
    IFNULL(AVG(grade), 0) AS average
FROM students
LEFT JOIN papers
    ON students.id = papers.student_id
GROUP BY students.id
ORDER BY average DESC;
```

```
+------------+---------+
| 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)
```



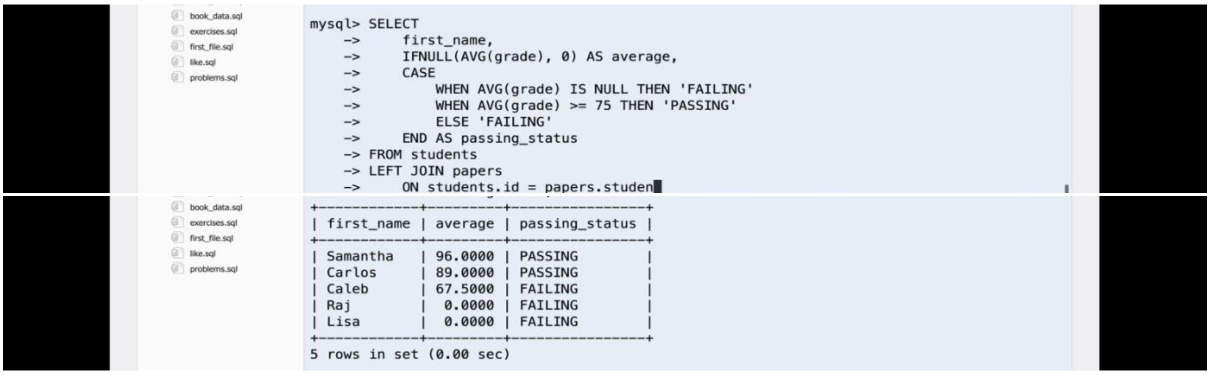
PRINT THIS

```
+--------------+---------+----------------+
| 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;
```

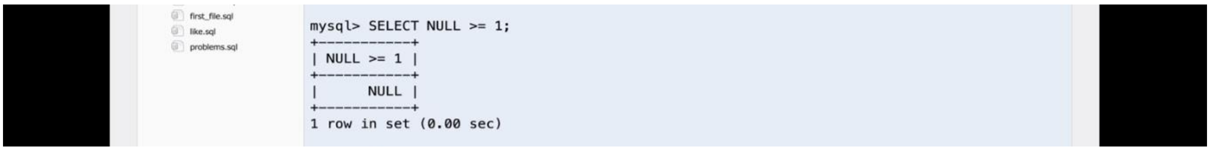
```
+------------+---------+----------------+
| first_name | average | passing_status |
+------------+---------+----------------+
| Samantha   | 96.0000 | PASSING        |
| Carlos     | 89.0000 | PASSING        |
| Caleb      | 67.5000 | FAILING        |
| Raj        |  0.0000 | FAILING        |
| Lisa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

```
+------------+---------+----------------+
| 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 |
+-----------+
|      NULL |
+-----------+
1 row in set (0.00 sec)
```