

Assignment 2: Retrieving Data

Write a SELECT statement that joins the Categories table to the Products table and returns these columns: category_name, product_name, list_price. Sort the result set by the category_name column and then by the product_name column in ascending sequence.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```
1 SELECT c.category_name, p.product_name, p.list_price
2 FROM categories c
3 JOIN products p ON c.category_id = p.category_id
4 ORDER BY c.category_name ASC, p.product_name ASC;
```

The Results grid displays the following data:

category_name	product_name	list_price
Basses	Fender Precision	799.99
Basses	Hofner Icon	499.99
Drums	Ludwig 5-Piece Drum Set with Cymbals	699.99
Drums	Tama 5-Piece Drum Set with Cymbals	799.99
Guitars	Fender Stratocaster	699.00
Guitars	Gibson Les Paul	1199.00
Guitars	Gibson SG	2517.00
Guitars	Rodriguez Caballero 11	415.00
Guitars	Washburn D10S	299.00
Guitars	Yamaha FG700S	489.99

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:07:45	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	18:08:04	USE my_guitar_shop	0 row(s) affected	0.000 sec
3	18:08:07	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id ...	10 row(s) returned	0.000 sec / 0.000 sec

Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first_name, last_name, line1, city, state, zip_code. Return one row for each address for the customer with an email address of allan.sherwood@yahoo.com.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```
1 SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code
2 FROM customers cu
3 JOIN addresses a ON cu.customer_id = a.customer_id
4 WHERE cu.email_address = 'allan.sherwood@yahoo.com';
```

The Results grid displays the following data:

first_name	last_name	line1	city	state	zip_code
Allan	Sherwood	100 East Ridgewood Ave.	Paramus	NJ	07652
Allan	Sherwood	21 Rosewood Rd.	Woodcliff Lake	NJ	07677

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:07:45	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	18:08:04	USE my_guitar_shop	0 row(s) affected	0.000 sec
3	18:08:07	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id ...	10 row(s) returned	0.000 sec / 0.000 sec
4	18:13:27	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ...	2 row(s) returned	0.000 sec / 0.000 sec

Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first_name, last_name, line1, city, state, zip_code. Return one row for each customer, but only return addresses that are the shipping address for a customer.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```
1 SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code
2 FROM customers cu
3 JOIN addresses a ON cu.customer_id = a.customer_id
4 WHERE cu.shipping_address_id = a.address_id; -- Assuming shipping_address_id links to addresses
```

The result grid displays the following data:

first_name	last_name	line1	city	state	zip_code
Alan	Sherwood	100 East Ridgewood Ave.	Paramus	NJ	07652
Barry	Zimmer	16383 Wendell St.	Omaha	NE	68135
Christine	Brown	19270 NW Cornell Rd.	Beaverton	OR	97006
David	Goldstein	186 Vermont St.	San Francisco	CA	94110
Erin	Valentino	6982 Palm Ave.	Fresno	CA	93711
Frank Lee	Wilson	23 Mountain View St.	Denver	CO	80208
Gary	Hernandez	7361 N. 41st St.	New York	NY	10012
Heather	Essay	2381 Buena Vista St.	Los Angeles	CA	90023

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:07:45	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id = ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking to name in the SCH...	0.000 sec
2	18:08:04	USE my_guitar_shop	0 row(s) affected	0.000 sec
3	18:08:07	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id = ...	10 row(s) returned	0.000 sec / 0.000 sec
4	18:13:27	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ...	2 row(s) returned	0.000 sec / 0.000 sec
5	18:14:46	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ...	8 row(s) returned	0.016 sec / 0.000 sec

Write a SELECT statement that joins the Customers, Orders, Order_Items, and Products tables. This statement should return these columns: last_name, first_name, order_date, product_name, item_price, discount_amount, and quantity. Use aliases for the tables. Sort the final result set by the last_name, order_date, and product_name columns.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```
1 SELECT cu.last_name, cu.first_name, o.order_date, p.product_name, oi.item_price, oi.discount_amount, oi.quantity
2 FROM customers cu
3 JOIN orders o ON cu.customer_id = o.customer_id
4 JOIN order_items oi ON o.order_id = oi.order_id
5 JOIN products p ON oi.product_id = p.product_id
6 ORDER BY cu.last_name ASC, o.order_date ASC, p.product_name ASC;
```

The result grid displays the following data:

last_name	first_name	order_date	product_name	item_price	discount_amount	quantity
Brown	Christine	2022-03-30 15:22:31	Gibson Les Paul	1199.00	359.70	2
Goldstein	David	2022-03-31 08:43:11	Washburn D105	299.00	0.00	1
Goldstein	David	2022-04-03 12:22:31	Fender Stratocaster	699.00	209.70	1
Hernandez	Gary	2022-04-02 11:26:38	Tama 5-Piece Drum Set with Cymbals	799.99	120.00	1
Sherwood	Alan	2022-03-28 09:40:28	Gibson Les Paul	1199.00	359.70	1
Sherwood	Alan	2022-03-29 09:44:58	Gibson SG	2517.00	1308.84	1
Sherwood	Alan	2022-03-29 09:44:58	Rodriguez Caballero 11	415.00	161.85	1
Valentino	Erin	2022-03-31 18:37:22	Washburn D105	299.00	0.00	1
Wilson	Frank Lee	2022-04-01 23:11:12	Fender Precision	799.99	240.00	1
Wilson	Frank Lee	2022-04-01 23:11:12	Fender Stratocaster	699.00	209.70	1
Wilson	Frank Lee	2022-04-01 23:11:12	Ludwig 5-piece Drum Set with Cymbals	699.99	210.00	1
Zimmer	Barry	2022-03-28 11:23:20	Yamaha FG7005	489.99	186.20	1

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:07:45	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id = ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking to name in the SCH...	0.000 sec
2	18:08:04	USE my_guitar_shop	0 row(s) affected	0.000 sec
3	18:08:07	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id = ...	10 row(s) returned	0.000 sec / 0.000 sec
4	18:13:27	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ...	2 row(s) returned	0.000 sec / 0.000 sec
5	18:14:46	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ...	8 row(s) returned	0.016 sec / 0.000 sec
6	18:15:32	SELECT cu.last_name, cu.first_name, o.order_date, p.product_name, oi.item_price, oi.discount_amount, oi.quan...	12 row(s) returned	0.000 sec / 0.000 sec

Write a SELECT statement that returns the product_name and list_price columns from the Products table. Return one row for each product that has the same list price as another product. Hint: Use a self-join to check that the product_id columns aren't equal but the list_price columns are equal. Sort the result set by the product_name column

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```

1 SELECT p1.product_name, p1.list_price
2 FROM products p1
3 JOIN products p2 ON p1.product_id <> p2.product_id AND p1.list_price = p2.list_price
4 ORDER BY p1.product_name ASC;

```

The Results tab shows the output of the query:

product_name	list_price
Fender Precision	799.99
Tama 5-Piece Drum Set with Cymbals	799.99

The bottom pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
2	18:08:04	USE my_guitar_shop	0 row(s) affected	0.000 sec
3	18:08:07	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id = p.category_id	10 row(s) returned	0.000 sec / 0.000 sec
4	18:13:27	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ON cu.customer_id = a.customer_id	2 row(s) returned	0.000 sec / 0.000 sec
5	18:14:46	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ON cu.customer_id = a.customer_id	8 row(s) returned	0.016 sec / 0.000 sec
6	18:15:32	SELECT cu.last_name, cu.first_name, o.order_date, p.product_name, oi.item_price, oi.discount_amount, oi.quantity FROM customers cu JOIN orders o ON cu.customer_id = o.customer_id JOIN products p ON o.product_id = p.product_id JOIN order_items oi ON o.order_id = oi.order_id	12 row(s) returned	0.000 sec / 0.000 sec
7	18:16:28	SELECT p1.product_name, p1.list_price FROM products p1 JOIN products p2 ON p1.product_id <> p2.product_id AND p1.list_price = p2.list_price	2 row(s) returned	0.031 sec / 0.000 sec

Write a SELECT statement that returns these two columns: category_name The category_name column from the Categories table product_id The product_id column from the Products table. Return one row for each category that has never been used. Hint: Use an outer join and only return rows where the product_id column contains a null value.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```

1 SELECT c.category_name, p.product_id
2 FROM categories c
3 LEFT JOIN products p ON c.category_id = p.category_id
4 WHERE p.product_id IS NULL;

```

The Results tab shows the output of the query:

category_name	product_id
Keyboards	

The bottom pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
3	18:08:07	SELECT c.category_name, p.product_name, p.list_price FROM categories c JOIN products p ON c.category_id = p.category_id	10 row(s) returned	0.000 sec / 0.000 sec
4	18:13:27	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ON cu.customer_id = a.customer_id	2 row(s) returned	0.000 sec / 0.000 sec
5	18:14:46	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses a ON cu.customer_id = a.customer_id	8 row(s) returned	0.016 sec / 0.000 sec
6	18:15:32	SELECT cu.last_name, cu.first_name, o.order_date, p.product_name, oi.item_price, oi.discount_amount, oi.quantity FROM customers cu JOIN orders o ON cu.customer_id = o.customer_id JOIN products p ON o.product_id = p.product_id JOIN order_items oi ON o.order_id = oi.order_id	12 row(s) returned	0.000 sec / 0.000 sec
7	18:16:28	SELECT p1.product_name, p1.list_price FROM products p1 JOIN products p2 ON p1.product_id <> p2.product_id AND p1.list_price = p2.list_price	2 row(s) returned	0.031 sec / 0.000 sec
8	18:17:52	SELECT c.category_name, p.product_id FROM categories c LEFT JOIN products p ON c.category_id = p.category_id	1 row(s) returned	0.016 sec / 0.000 sec

Use the UNION operator to generate a result set consisting of three columns from the Orders table: ship_status A column that contains a value of SHIPPED or NOT SHIPPED
order_id The order_id column
order_date The order_date column. If the order has a value in the ship_date column, the ship_status column should contain a value of SHIPPED. Otherwise, it should contain a value of NOT SHIPPED. Sort the final result set by the order_date column.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 SELECT
2   CASE
3     WHEN ship_date IS NOT NULL THEN 'SHIPPED'
4     ELSE 'NOT SHIPPED'
5   END AS ship_status,
6   order_id,
7   order_date
8 FROM orders
9 ORDER BY order_date ASC;

```

The Results tab displays the following data:

ship_status	order_id	order_date
SHIPPED	1	2022-03-28 09:40:28
SHIPPED	2	2022-03-28 11:23:20
SHIPPED	3	2022-03-29 09:44:58
SHIPPED	4	2022-03-30 15:22:31
SHIPPED	5	2022-03-31 05:43:11
NOT SHIPPED	6	2022-03-31 18:37:22
SHIPPED	7	2022-04-01 23:11:12
NOT SHIPPED	8	2022-04-02 11:26:38

The Output tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
4	18:13:27	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses ...	2 row(s) returned	0.000 sec / 0.000 sec
5	18:14:46	SELECT cu.first_name, cu.last_name, a.line1, a.city, a.state, a.zip_code FROM customers cu JOIN addresses ...	8 row(s) returned	0.016 sec / 0.000 sec
6	18:15:32	SELECT cu.last_name, cu.first_name, o.order_date, p.product_name, oi.item_price, oi.discount_amount, oi.qu...	12 row(s) returned	0.000 sec / 0.000 sec
7	18:16:28	SELECT p1.product_name, p1.list_price FROM products p1 JOIN products p2 ON p1.product_id <-> p2.produc...	2 row(s) returned	0.031 sec / 0.000 sec
8	18:17:52	SELECT c.category_name, p.product_id FROM categories c LEFT JOIN products p ON c.category_id = p.cat...	1 row(s) returned	0.016 sec / 0.000 sec
9	18:19:21	SELECT CASE WHEN ship_date IS NOT NULL THEN 'SHIPPED' ELSE 'NOT SHIPPED' E...	9 row(s) returned	0.016 sec / 0.000 sec

