

1. Write a SELECT statement that returns these columns from the Products table:
The list_price column
The discount_percent column
A column named discount_amount that uses the previous two columns to calculate the discount amount and uses the ROUND function to round the result so it has 2 decimal digits.

The screenshot shows the MySQL Workbench interface. The SQL Editor at the top contains the following query:

```
1 SELECT
2   list_price,
3   discount_percent,
4   ROUND(list_price * discount_percent / 100, 2) AS discount_amount
5 FROM
6   products;
7
```

Below the editor, the 'Result Grid' tab is active, displaying the query results in a table with three columns: list_price, discount_percent, and discount_amount. The results are as follows:

list_price	discount_percent	discount_amount
699.00	30.00	209.70
1199.00	30.00	359.70
2517.00	52.00	1308.84
489.99	38.00	186.20
299.00	0.00	0.00
415.00	39.00	161.85
799.99	30.00	240.00
499.99	25.00	125.00

At the bottom, the 'Action Output' tab shows the execution log. The last two actions are highlighted in red, indicating errors:

- 19 22:45:45 SELECT list_price, discount_percent, ROUND(list_price * (discount_percent / 100), 2) AS discount_... Error Code: 1146. Table 'my_guitar_shop.Products' doesn't exist 0.031 sec
- 20 22:46:43 SELECT list_price, discount_percent, ROUND(list_price * discount_percent / 100, 2) AS discount_amount FR... 10 row(s) returned 0.015 sec / 0.000 sec

2. Write a SELECT statement that returns these columns from the Orders table:
- The order_date column
 - A column that uses the DATE_FORMAT function to return the four-digit year that's stored in the order_date column
 - A column that uses the DATE_FORMAT function to return the order_date column in this format: Mon-DD-YYYY. In other words, use abbreviated months and separate each date component with dashes.
 - A column that uses the DATE_FORMAT function to return the order_date column with only the hours and minutes on a 12-hour clock with an am/pm indicator
 - A column that uses the DATE_FORMAT function to return the order_date column in this format: MM/DD/YY HH:SS. In other words, use two-digit months, days, and years and separate them by slashes. Use 2-digit hours and minutes on a 24-hour clock. And use leading zeros for all date/time components.

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

```
1 SELECT
2   order_date,
3   DATE_FORMAT(order_date, '%Y') AS four_digit_year,
4   DATE_FORMAT(order_date, '%b-%d-%Y') AS formatted_date,
5   DATE_FORMAT(order_date, '%h:%i %p') AS formatted_time,
6   DATE_FORMAT(order_date, '%m/%d/%y %h:%i') AS custom_format
7 FROM
8   orders;
```

The Results window displays the following data:

order_date	four_digit_year	formatted_date	formatted_time	custom_format
2015-03-28 09:40:28	2015	Mar-28-2015	09:40 AM	03/28/15 09:40
2015-03-28 11:23:20	2015	Mar-28-2015	11:23 AM	03/28/15 11:23
2015-03-29 09:44:58	2015	Mar-29-2015	09:44 AM	03/29/15 09:44
2015-03-30 15:22:31	2015	Mar-30-2015	03:22 PM	03/30/15 15:22
2015-03-31 05:43:11	2015	Mar-31-2015	05:43 AM	03/31/15 05:43
2015-03-31 18:37:22	2015	Mar-31-2015	06:37 PM	03/31/15 18:37
2015-04-01 23:11:12	2015	Apr-01-2015	11:11 PM	04/01/15 23:11

The Output window shows the following messages:

#	Time	Action	Message	Duration / Fetch
17	22:45:34	INSERT INTO administrators (admin_id, email_address, password, first_name, last_name) VALUES (1, 'admin@my...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.031 sec
18	22:45:34	GRANT SELECT, INSERT, UPDATE, DELETE ON * TO mgs_user@localhost IDENTIFIED BY 'ba55word'	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server...	0.328 sec
19	22:45:45	SELECT list_price, discount_percent, ROUND(list_price * (discount_percent / 100), 2) AS discount_ammoun...	Error Code: 1146. Table 'my_guitar_shop.Products' doesn't exist	0.031 sec
20	22:46:43	SELECT list_price, discount_percent, ROUND(list_price * discount_percent / 100, 2) AS discount_amount FR...	10 row(s) returned	0.015 sec / 0.000 sec
21	22:50:00	SELECT order_date, DATE_FORMAT(order_date, '%Y') AS four_digit_year, DATE_FORMAT(order_date, '%b...	9 row(s) returned	0.203 sec / 0.000 sec

3. Write a SELECT statement that returns these columns from the Orders table:
- The card_number column
 - The length of the card_number column
 - When you get that working right, add the columns that follow to the result set. This is more difficult because these columns require the use of functions within functions.
 - The last four digits of the card_number column
 - A column that displays an X for each digit of the card_number column except for the last four digits. If the card number contains 16 digits, it should be displayed in this format: XXXX-XXXX-XXXX-1234, where 1234 are the actual last four digits of the number. If the card number contains 15 digits, it should be displayed in this format: XXXX-XXXXXX-X1234. (Hint: Use an IF function to determine which format to use.)

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

```
1 SELECT
2   card_number,
3   LENGTH(card_number) AS card_number_length,
4   RIGHT(card_number, 4) AS last_four_digits,
5   CASE
6     WHEN LENGTH(card_number) = 16 THEN
7       CONCAT('XXXX-XXXX-XXXX-', RIGHT(card_number, 4))
8     WHEN LENGTH(card_number) = 15 THEN
```

The Results window displays the following data:

card_number	card_number_length	last_four_digits	formatted_card_number
4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
4012888888888881	16	1881	XXXX-XXXX-XXXX-1881
4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
37828246310005	15	0005	XXXX-XXXXXX-X0005
4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
6011111111111117	16	1117	XXXX-XXXX-XXXX-1117
5555555555554444	16	4444	XXXX-XXXX-XXXX-4444

The Output window shows the following actions:

#	Time	Action	Message	Duration / Fetch
18	22:45:34	GRANT SELECT, INSERT, UPDATE, DELETE ON * TO mgs_user@localhost IDENTIFIED BY pa55word	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server...	0.328 sec
19	22:45:45	SELECT list_price, discount_percent, ROUND(list_price * (discount_percent / 100), 2) AS discount_amount	Error Code: 1146. Table 'my_guitar_shop.Products' doesn't exist	0.031 sec
20	22:46:43	SELECT list_price, discount_percent, ROUND(list_price * discount_percent / 100, 2) AS discount_amount	10 row(s) returned	0.015 sec / 0.000 sec
21	22:50:00	SELECT order_date, DATE_FORMAT(order_date, '%Y') AS four_digt_year, DATE_FORMAT(order_date, '%b...	9 row(s) returned	0.203 sec / 0.000 sec
22	22:52:30	SELECT card_number, LENGTH(card_number) AS card_number_length, RIGHT(card_number, 4) AS last_four...	9 row(s) returned	0.016 sec / 0.000 sec

4. Write a SELECT statement that returns these columns from the Orders table:
The order_id column
The order_date column
A column named approx_ship_date that's calculated by adding 2 days to the order_date column
The ship_date column if it doesn't contain a null value
A column named days_to_ship that shows the number of days between the order date and the ship date
When you have this working, add a WHERE clause that retrieves just the orders for March 2018.

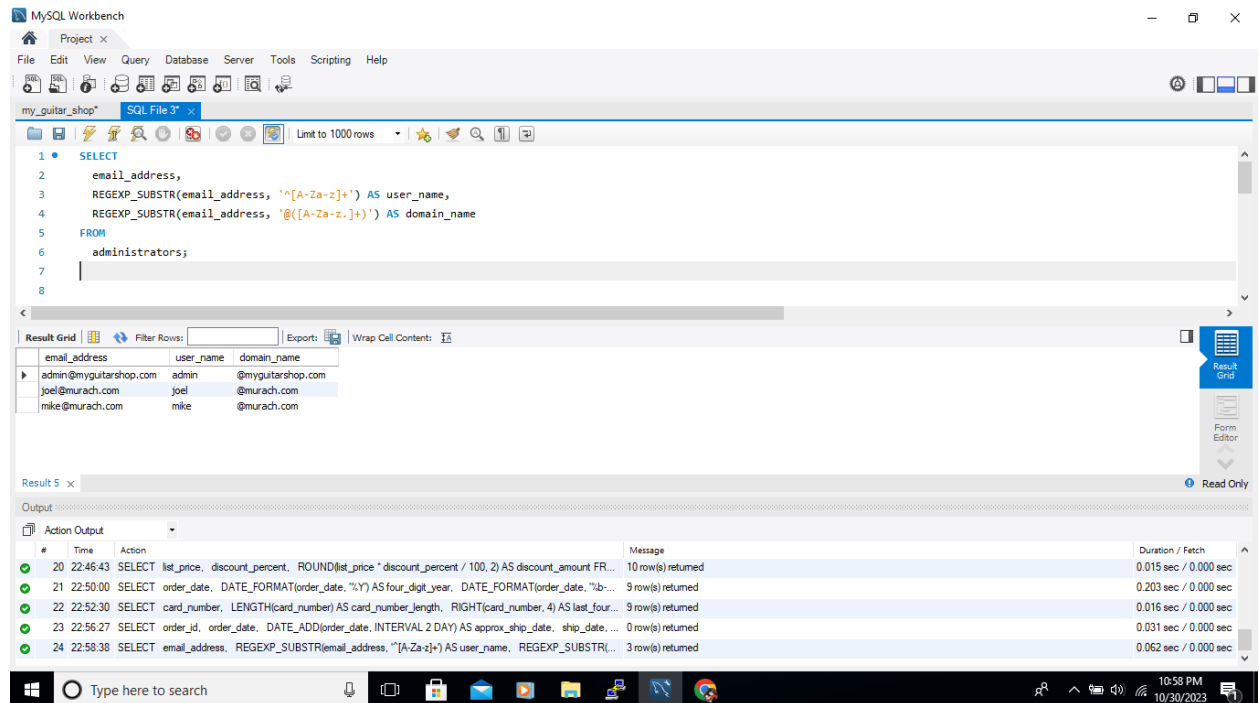
The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
4  DATE_ADD(order_date, INTERVAL 2 DAY) AS approx_ship_date,  
5  ship_date,  
6  DATEDIFF(ship_date, order_date) AS days_to_ship  
7  FROM  
8  orders  
9  WHERE  
10 order_date >= '2018-03-01' AND order_date < '2018-04-01';  
11
```

Below the editor, the Result Grid shows the columns: order_id, order_date, approx_ship_date, ship_date, and days_to_ship. The Output pane at the bottom displays the execution log:

#	Time	Action	Message	Duration / Fetch
19	22:45:45	SELECT list_price, discount_percent, ROUND(list_price * (discount_percent / 100), 2) AS discount_amount...	Error Code: 1146. Table 'my_guitar_shop.Products' doesn't exist	0.031 sec
20	22:46:43	SELECT list_price, discount_percent, ROUND(list_price * discount_percent / 100, 2) AS discount_amount FR...	10 row(s) returned	0.015 sec / 0.000 sec
21	22:50:00	SELECT order_date, DATE_FORMAT(order_date, '%Y') AS four_digit_year, DATE_FORMAT(order_date, '%b-...	9 row(s) returned	0.203 sec / 0.000 sec
22	22:52:30	SELECT card_number, LENGTH(card_number) AS card_number_length, RIGHT(card_number, 4) AS last_four...	9 row(s) returned	0.016 sec / 0.000 sec
23	22:56:27	SELECT order_id, order_date, DATE_ADD(order_date, INTERVAL 2 DAY) AS approx_ship_date, ship_date, ...	0 row(s) returned	0.031 sec / 0.000 sec

5. Write a SELECT statement that uses regular expression functions to get the username and domain name parts of the email addresses in the Administrators table. Return these columns: The email_address column A column named user_name that contains the username part of the email_address column (the part before the @ symbol) A column named domain_name that contains the domain name part of the email_address column (the part after the @ symbol) Note: The username part of the email addresses contains only letters, and the domain name part contains only letters and a period.



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

```
1 SELECT
2   email_address,
3   REGEXP_SUBSTR(email_address, '^[A-Za-z]+' ) AS user_name,
4   REGEXP_SUBSTR(email_address, '@([A-Za-z.]+)' ) AS domain_name
5 FROM
6   administrators;
7
8
```

The Results Grid shows the output of the query:

email_address	user_name	domain_name
admin@myguitarshop.com	admin	@myguitarshop.com
joel@murach.com	joel	@murach.com
mike@murach.com	mike	@murach.com

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

#	Time	Action	Message	Duration / Fetch
20	22:46:43	SELECT list_price, discount_percent, ROUND(list_price * discount_percent / 100, 2) AS discount_amount FR...	10 row(s) returned	0.015 sec / 0.000 sec
21	22:50:00	SELECT order_date, DATE_FORMAT(order_date, '%Y') AS four_digit_year, DATE_FORMAT(order_date, '%b-...	9 row(s) returned	0.203 sec / 0.000 sec
22	22:52:30	SELECT card_number, LENGTH(card_number) AS card_number_length, RIGHT(card_number, 4) AS last_four...	9 row(s) returned	0.016 sec / 0.000 sec
23	22:56:27	SELECT order_id, order_date, DATE_ADD(order_date, INTERVAL 2 DAY) AS approx_ship_date, ship_date, ...	0 row(s) returned	0.031 sec / 0.000 sec
24	22:58:38	SELECT email_address, REGEXP_SUBSTR(email_address, '^[A-Za-z]+') AS user_name, REGEXP_SUBSTR(...	3 row(s) returned	0.062 sec / 0.000 sec

6. Write a SELECT statement that uses the ranking functions to rank products by the total quantity sold. Return these columns: The product_name column from the Products table A column named total_quantity that shows the sum of the quantity for each product in the order_items table A column named rank that uses the RANK function to rank the total quantity in descending sequence A column named dense_rank that uses the DENSE_RANK function to rank the total quantity in descending sequence.

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

```
1 SELECT
2   product_name,
3   RANK() OVER (ORDER BY total_quantity DESC) AS product_rank,
4   DENSE_RANK() OVER (ORDER BY total_quantity DESC) AS dense_product_rank,
5   total_quantity
6 FROM
7 (
8   SELECT
9     product_name,
```

The Results tab displays the following data:

product_name	product_rank	dense_product_rank	total_quantity
Gibson Les Paul	1	1	3
Fender Stratocaster	2	2	2
Washburn D10S	2	2	2
Gibson SG	4	3	1
Yamaha FG700S	4	3	1

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
68	16:19:10	SELECT	product_name, ... 9 row(s) returned	0.016 sec / 0.000 sec
69	16:20:42	SELECT	product_name, ... 9 row(s) returned	0.015 sec / 0.000 sec
70	16:22:14	SELECT	product_name, ... 9 row(s) returned	0.015 sec / 0.000 sec
71	16:24:00	SELECT	product_name, ... Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'dense_rank, total_quantity FROM (SELE...	0.015 sec
72	16:25:05	SELECT	product_name, ... 9 row(s) returned	0.031 sec / 0.000 sec

7. Write a SELECT statement that uses the analytic functions to get the highest and lowest sales by product within each category. Return these columns: The category_name column from the Categories table The product_name column from the Products table A column named total_sales that shows the sum of the sales for each product with sales in the order_items table A column named highest_sales that uses the FIRST_VALUE function to show the name of the product with the highest sales within each category A column named lowest_sales that uses the LAST_VALUE function to show the name of the product with the lowest sales within each category

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

```

1  SELECT
2  c.category_name,
3  p.product_name,
4  SUM(oi.quantity) AS total_sales,
5  FIRST_VALUE(p.product_name) OVER (PARTITION BY c.category_name ORDER BY SUM(oi.quantity) DESC) AS highest_sales,
6  LAST_VALUE(p.product_name) OVER (PARTITION BY c.category_name ORDER BY SUM(oi.quantity) ASC) AS lowest_sales
7  FROM
8  categories c
9  JOIN

```

The Results tab shows a table with 5 columns: category_name, product_name, total_sales, highest_sales, and lowest_sales. The data is as follows:

category_name	product_name	total_sales	highest_sales	lowest_sales
Basses	Fender Precision	1	Fender Precision	Fender Precision
Basses	Hofner Icon	1	Fender Precision	Hofner Icon
Drums	Ludwig 5-piece Drum Set with Cymbals	1	Ludwig 5-piece Drum Set with Cymbals	Tama 5-Piece Drum Set with Cymbals
Drums	Tama 5-Piece Drum Set with Cymbals	1	Ludwig 5-piece Drum Set with Cymbals	Tama 5-Piece Drum Set with Cymbals
Guitars	Gibson Les Paul	3	Gibson Les Paul	Gibson Les Paul

The Output tab shows the following messages:

#	Time	Action	Message	Duration / Fetch
59	16:06:24	SELECT	c.category_name... Error Code: 1146. Table 'my_guitar_shop.Categories' doesn't exist	0.015 sec
60	16:06:56	SELECT	c.category_name... Error Code: 1054. Unknown column 'total_sales' in field list	0.016 sec
61	16:08:03	SELECT	c.category_name... Error Code: 1146. Table 'my_guitar_shop.Categories' doesn't exist	0.015 sec
62	16:09:09	SELECT	c.category_name... Error Code: 1054. Unknown column 'oi.sales' in field list	0.016 sec
63	16:09:55	SELECT	c.category_name... 10 row(s) returned	0.031 sec / 0.000 sec