

1.

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 4\*'. The query is as follows:

```
1 • SELECT
2     product_code,
3     product_name,
4     list_price,
5     discount_percent
6 FROM products
7 ORDER BY list_price desc
```

Below the query window, the 'Result Grid' is displayed, showing the results of the query. The grid has four columns: product\_code, product\_name, list\_price, and discount\_percent. The results are sorted by list\_price in descending order.

product_code	product_name	list_price	discount_percent
sg	Gibson SG	2517.00	52.00
les_paul	Gibson Les Paul	1199.00	30.00
precision	Fender Precision	799.99	30.00
tama	Tama 5-Piece Drum Set with Cymbals	799.99	15.00
ludwig	Ludwig 5-piece Drum Set with Cymbals	699.99	30.00
strat	Fender Stratocaster	699.00	30.00
hofner	Hofner Icon	499.99	25.00
fg700s	Yamaha FG700S	489.99	38.00
rodriguez	Rodriguez Caballero 11	415.00	39.00
washburn	Washburn D10S	299.00	0.00

2.

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 4\*'. The query is as follows:

```
1 • SELECT concat(last_name, ", ", first_name) as full_name
2 FROM customers
3 where last_name >= 'M'
4 ORDER BY last_name ASC;
```

Below the query window, the 'Result Grid' is displayed, showing the results of the query. The grid has one column: full\_name. The results are sorted by last\_name in ascending order.

full_name
Sherwood, Allan
Valentino, Erin
Wilson, Frank Lee
Zimmer, Barry

3.

create\_db\_ap create\_my\_guitar\_shop SQL File 4\* SQL File 5\* x

Limit to 1000 rows

```

1 • select product_name as "The product_name column",
2     list_price as "The list_price column",
3     date_added as "The date_added column"
4 from products
5 where list_price > 500 and list_price < 2000
6 order by date_added desc

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	The product_name column	The list_price column	The date_added column
▶	Tama 5-Piece Drum Set with Cymbals	799.99	2018-07-30 13:14:15
	Ludwig 5-piece Drum Set with Cymbals	699.99	2018-07-30 12:46:40
	Fender Precision	799.99	2018-06-01 11:29:35
	Gibson Les Paul	1199.00	2017-12-05 16:33:13
	Fender Stratocaster	699.00	2017-10-30 09:32:40

4.

create\_db\_ap create\_my\_guitar\_shop SQL File 4\* SQL File 5\* SQL File 6\* x

Limit to 1000 rows

```

1 • Select product_name as "The product_name column",
2     list_price as "The list_price column",
3     discount_percent as "The discount_person column",
4     round((discount_percent/100)*list_price,2) as discount_amount,
5     round(list_price-(discount_percent/100)*list_price,2) as discount_price
6 from products
7 order by round(list_price-(discount_percent/100)*list_price,2) desc
8 limit 5

```

Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:

	The product_name column	The list_price column	The discount_person column	discount_amount	discount_price
▶	Gibson SG	2517.00	52.00	1308.84	1208.16
	Gibson Les Paul	1199.00	30.00	359.70	839.30
	Tama 5-Piece Drum Set with Cymbals	799.99	15.00	120.00	679.99
	Fender Precision	799.99	30.00	240.00	559.99
	Ludwig 5-piece Drum Set with Cymbals	699.99	30.00	210.00	489.99

Continue to next page

5.

The screenshot shows a SQL query in a text editor. The query selects columns from the 'order\_items' table, calculates 'price\_total', 'discount\_total', and 'item\_total' based on item price, discount amount, and quantity. It filters for items where the total value is greater than 500 and orders them by 'item\_total' in descending order.

```

1 • select item_id as " The Item_id column",
2   item_price as " The item_price column",
3   discount_amount as "The discount_amount column",
4   quantity as "The quantity column",
5   (item_price*quantity) as price_total,
6   (discount_amount*quantity) as discount_total,
7   ((item_price-discount_amount)*quantity) as item_total
8 from order_items
9 where ((item_price-discount_amount)*quantity) > 500
10 order by item_total desc

```

Below the query, the 'Result Grid' shows the following data:

	The Item_id column	The item_price column	The discount_amount column	The quantity column	price_total	discount_total	item_total
▶	5	1199.00	359.70	2	2398.00	719.40	1678.60
	3	2517.00	1308.84	1	2517.00	1308.84	1208.16
	1	1199.00	359.70	1	1199.00	359.70	839.30
	11	799.99	120.00	1	799.99	120.00	679.99
	9	799.99	240.00	1	799.99	240.00	559.99

6.

The screenshot shows a SQL query in a text editor. The query selects columns from the 'orders' table, filtering for orders where the 'ship\_date' is NULL.

```

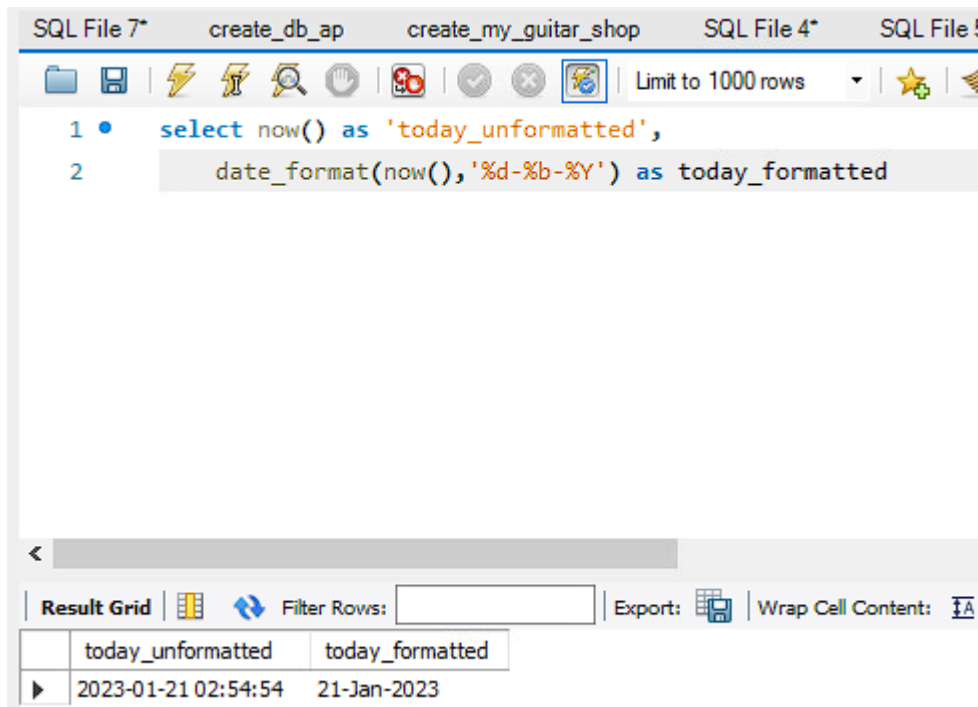
1 • select
2   order_id as "the order id column",
3   order_date as "the orderdate column",
4   ship_date as "The ship date column"
5 from orders
6 where ship_date is NULL

```

Below the query, the 'Result Grid' shows the following data:

	the order id column	the orderdate column	The ship date column
▶	6	2018-03-31 18:37:22	NULL
	8	2018-04-02 11:26:38	NULL
	9	2018-04-03 12:22:31	NULL

7.Citation: <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>



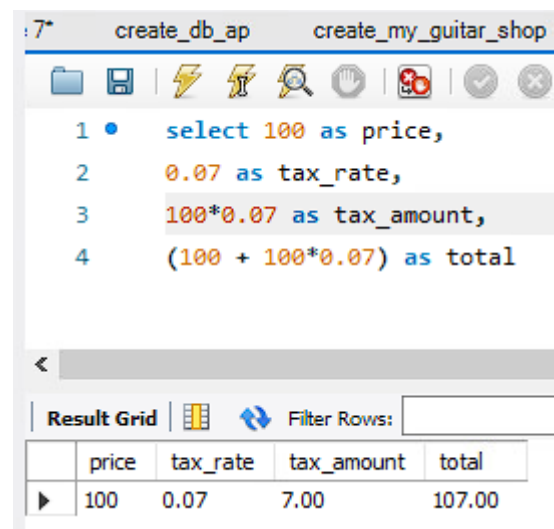
The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • select now() as 'today_unformatted',  
2    date_format(now(), '%d-%b-%Y') as today_formatted
```

The result grid displays the output of the query:

	today_unformatted	today_formatted
▶	2023-01-21 02:54:54	21-Jan-2023

8.



The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • select 100 as price,  
2    0.07 as tax_rate,  
3    100*0.07 as tax_amount,  
4    (100 + 100*0.07) as total
```

The result grid displays the output of the query:

	price	tax_rate	tax_amount	total
▶	100	0.07	7.00	107.00

Continue to next page

9.

The screenshot shows a SQL IDE window titled 'create\_my\_guitar\_shop' with a toolbar and a query editor. The query is as follows:

```
1 • select customer_id,  
2     city,  
3     state,  
4     zip_code  
5     from addresses  
6     where state in('NJ','CA','NY')  
7     order by state,city asc
```

Below the query editor, the 'Result Grid' is displayed with the following data:

	customer_id	city	state	zip_code
▶	5	Fresno	CA	93711
	8	Los Angeles	CA	90023
	8	Los Angeles	CA	90024
	4	San Francisco	CA	94110
	4	San Francisco	CA	94129
	1	Paramus	NJ	07652
	1	Woodcliff Lake	NJ	07677
	7	New York	NY	10012
	7	New York	NY	10012

10. citation: [https://www.w3schools.com/sql/func\\_mysql\\_substring.asp](https://www.w3schools.com/sql/func_mysql_substring.asp)

The screenshot shows a SQL IDE window titled 'create\_my\_guitar\_shop' with a toolbar and a query editor. The query is as follows:

```
1 • select  
2     email_address,  
3     concat(substring(first_name,1,1), ", ", last_name) as Customer  
4     from customers  
5     where email_address like "%yahoo.com"  
6     order by email_address
```

Below the query editor, the 'Result Grid' is displayed with the following data:

	email_address	Customer
▶	allan.sherwood@yahoo.com	A, Sherwood
	gary_hernandez@yahoo.com	G, Hernandez