

SQL QUERY STATEMENTS

1-Insert

```
5
4 • INSERT INTO CUSTOMERS VALUES (505999172, 'Fahad', 'Visa');
```

100% 42:1

Result Grid

Filter Rows: Search

Edit:

Export/Import:

Customer_phone_no	Customer_Name	Card_Info
504573824	Sara	MasterCard
505617368	Nour	American Express
505717129	Nora	Visa
505999172	Fahad	Visa
555777666	Bader	Mada
555837592	Reema	Visa
555847845	Tala	Mastercard
555877887	Fahad	Mada
556635288	Fahad	Mada
582749888	Rand	Visa
585559321	Khalid	MasterCard
NULL	NULL	NULL

2-Delete

1 • SELECT * FROM the_eclipse_cafe.Customers;

2 • DELETE FROM Customers WHERE Customer_phone_no=505999172;

3

100%

1:3

Result Grid

Filter Rows:

Search

Edit:

Export/Import:

Customer_phone_no	Customer_Name	Card_Info
501719000	Tala	Mastercard
505666777	Khalid	MasterCard
505717345	Sara	MasterCard
508983920	Nour	American Express
534259999	Reema	Visa
550700070	Fahad	Mada
555834678	Nora	Visa
555845847	Bader	Mada
555881878	Fahad	Mada
582749888	Rand	Visa

Customers 2

Result Grid

Form Editor

Apply

Revert

3-Update

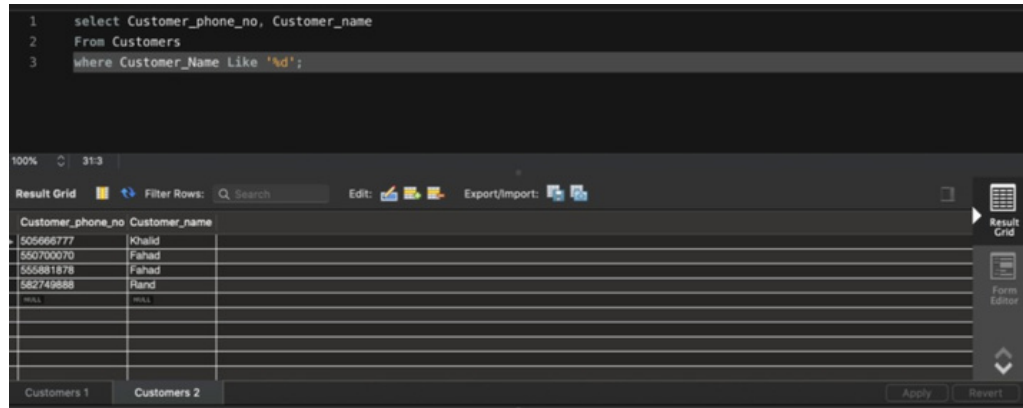
```
3 UPDATE suppliers
4 SET Supplier_Name='illy'
5 WHERE Supplier_ID=1;
```

100% 25:4

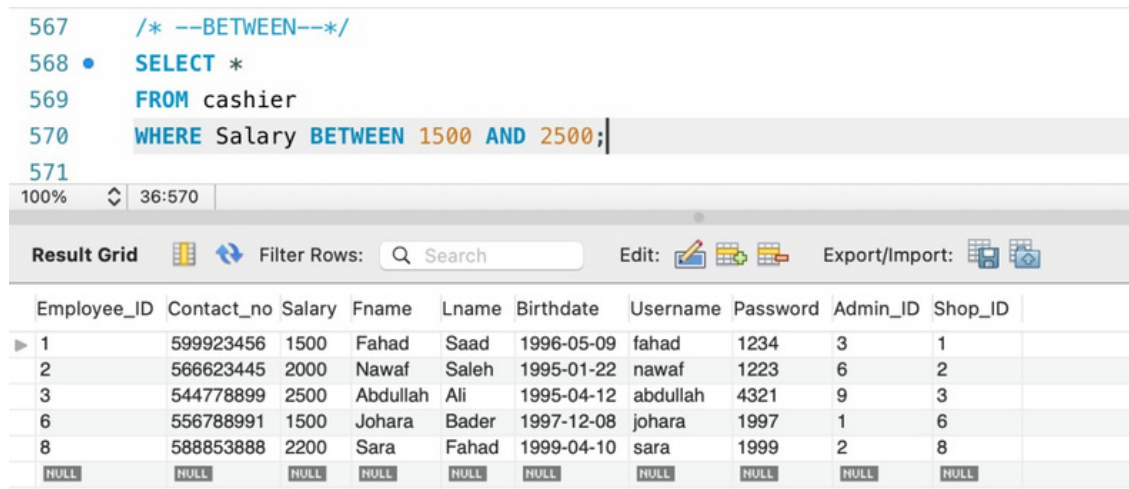
Result Grid Filter Rows: Search

Supplier_ID	Supplier_Name	Supplier_contract_No
1	illy	203
2	Mokaflor	189
3	Arabica	129
4	Camel step	989
5	Starbucks	675
6	Drip	122
7	Roasty	232
8	Robusta	212
9	Beany	356
10	Coffea	778

4- Like / NOT Like










5- Between



6- IN

```
573 /* --IN--*/
574 • SELECT *
575 FROM cashier_address
576 WHERE Cashier_address IN ('Jeddah','Jubail','Riyadh');
577 |
```

100% 1:577

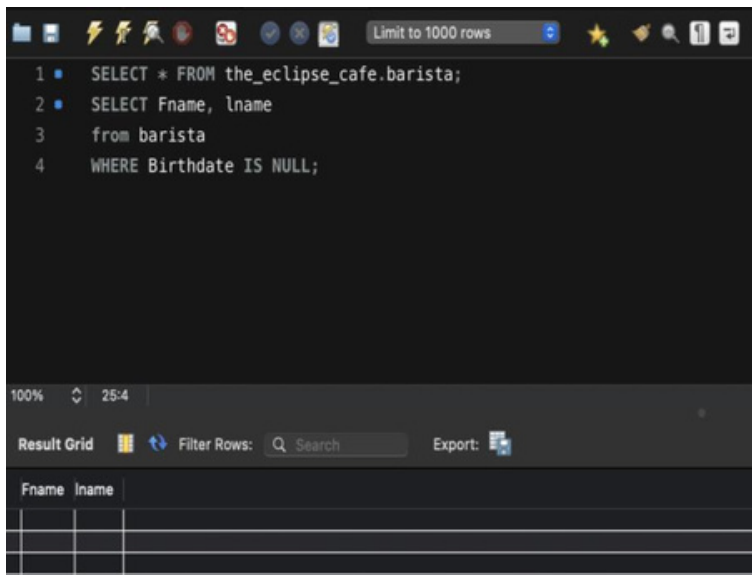
Result Grid   Filter Rows: Edit:    Export/Import:  

Employee_ID	Cashier_ad...
2	Jeddah
9	Jeddah
3	Jubail
10	Jubail
5	Riyadh
8	Riyadh
NULL	NULL

7- Order By

[illegible]

8- Is Null



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, a 'Limit to 1000 rows' dropdown, and other utility icons. The SQL editor contains the following query:

```
1 SELECT * FROM the_eclipse_cafe.barista;  
2 SELECT Fname, lname  
3   from barista  
4   WHERE Birthdate IS NULL;
```

Below the editor, the 'Result Grid' section shows a table with two columns: 'Fname' and 'lname'. The table is currently empty, with only the header row visible.

Fname	lname
-------	-------

9- Intersect-Except-Union

```
(select Fname, Lname  
from admin
```

```
Where Salary<3000)
```

```
union
```

```
(select Fname, Lname
```

```
from barista
```

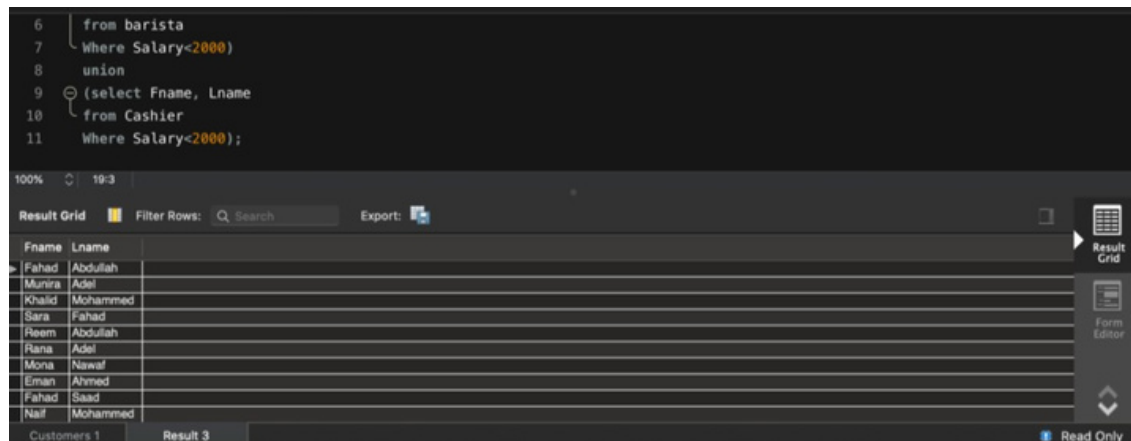
```
Where Salary<2000)
```

```
union
```

```
(select Fname, Lname
```

```
from Cashier
```

```
Where Salary<2000);
```



```
6  from barista
7  Where Salary<2000)
8  union
9  (select Fname, Lname
10 from Cashier
11 Where Salary<2000);
```

100% 19:3

Result Grid Filter Rows: Search Export:

Fname	Lname
Fahad	Abdullah
Munira	Adel
Khalid	Mohammed
Sara	Fahad
Reem	Abdullah
Rana	Adel
Mona	Nawal
Eman	Ahmed
Fahad	Saad
Nail	Mohammed

Customers 1 Result 3 Read Only

10- Group By & Having

```
580 • SELECT Fname, LName
581 FROM barista
582 GROUP BY FName, LName
583 HAVING FName > 'L' ORDER BY FName;
584
```

100% 1:579

Result Grid



Filter Rows:

Search

Export:



	Fname	LName	
►	Maha	Khalid	
	Sadeem	Abdullah	
	Saud	Fahad	
	Sulaiman	Issa	

11- Exists

The screenshot shows a SQL query in SQL Enterprise Manager. The query is:

```
1 select Fname,Lname,Order_no,Status
2 from Cashier,Orders
3 where Exists (select Cashier_ID from Orders WHERE Status="Completed");
4
5
6
7
```

The results are displayed in a grid with columns: Fname, Lname, Order_no, Status. The data is as follows:

Fname	Lname	Order_no	Status
Misral	Khalid	1	Completed
Nail	Mohammed	1	Completed
Abdullah	Ali	1	Completed
Nawaf	Saleh	1	Completed
Fahad	Saad	1	Completed
Nouf	Fahad	2	Completed
Lulu	Sulaiman	2	Completed
Sara	Fahad	2	Completed
Yara	Khalid	2	Completed
Johara	Bader	2	Completed

The interface also shows a 'Table: Orders' with columns: Order_no (int PK), Status (enum('Completed','In progress','Cancelled')), Item_Name (varchar(45)), Barista_ID (int), Date_taken (date), Cashier_ID (int), and customer_phone_no (int). The bottom panel shows the 'Action Output' with a list of actions and their durations.

12- comparison queries (ALL/ANY)

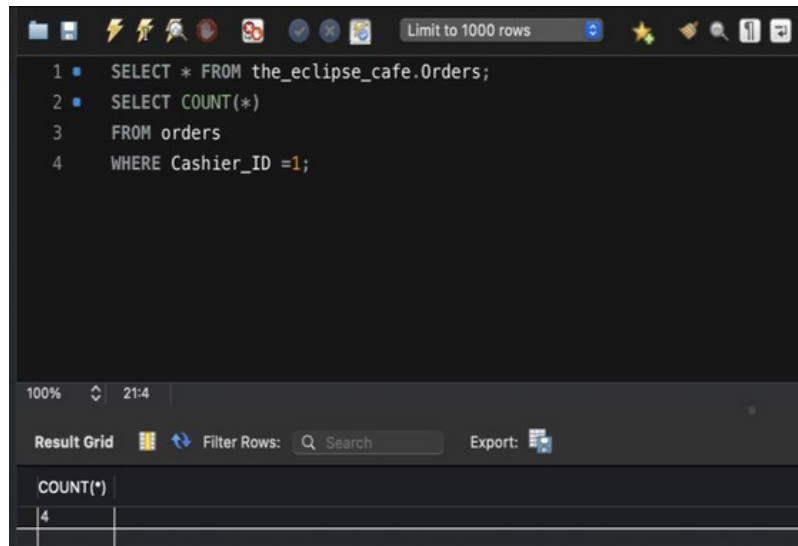
```
586 /* --ALL/ANY--*/
587 • SELECT barista.Fname,barista.LName
588 FROM barista
589 WHERE barista.Salary >ALL ( SELECT cashier.Salary From cashier );
590
```

100% 1:585

Result Grid Filter Rows: Search Export:

Fname	LName
Khalid	Abdullah
Abdullah	Saad
Jude	Saleh
Sulaiman	Issa

13- Aggregation



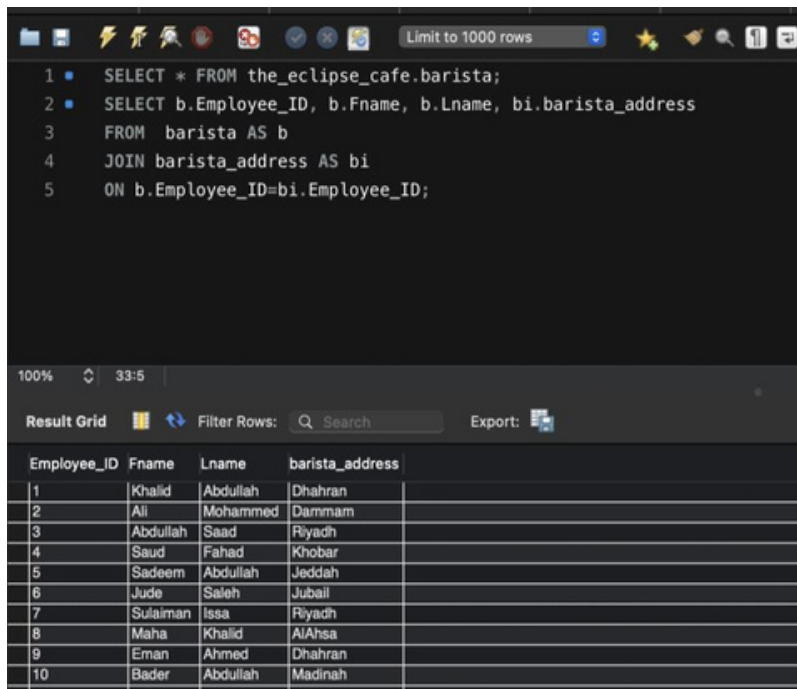
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, a 'Limit to 1000 rows' dropdown, and other utility icons. The SQL editor contains the following query:

```
1 SELECT * FROM the_eclipse_cafe.Orders;  
2 SELECT COUNT(*)  
3 FROM orders  
4 WHERE Cashier_ID =1;
```

Below the editor, the 'Result Grid' is displayed. It shows a single row with the value '4' under the column header 'COUNT(*)'. The interface also includes a 'Filter Rows' search bar and an 'Export' button.

COUNT(*)
4

14- Join + nested Query.



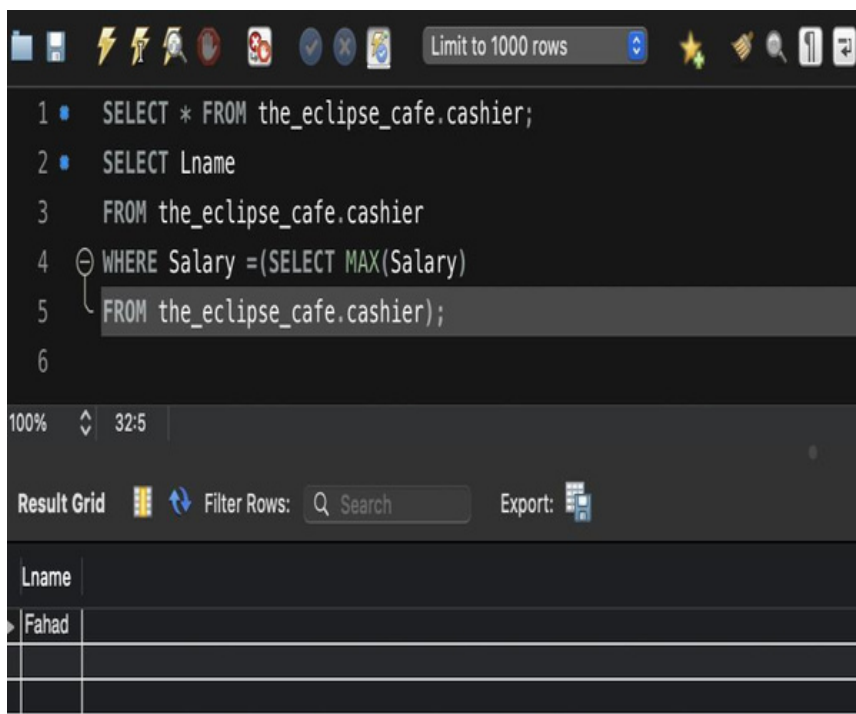
The screenshot shows a SQL IDE with a query editor and a result grid. The query is a JOIN between the `barista` and `barista_address` tables. The result grid displays 10 rows of data.

```
1 SELECT * FROM the_eclipse_cafe.barista;  
2 SELECT b.Employee_ID, b.Fname, b.Lname, bi.barista_address  
3 FROM barista AS b  
4 JOIN barista_address AS bi  
5 ON b.Employee_ID=bi.Employee_ID;
```

100% 33:5

Result Grid Filter Rows: Search Export:

Employee_ID	Fname	Lname	barista_address
1	Khalid	Abdullah	Dhahran
2	Ali	Mohammed	Dammam
3	Abdullah	Saad	Riyadh
4	Saud	Fahad	Khobar
5	Sadeem	Abdullah	Jeddah
6	Jude	Salih	Jubail
7	Sulaiman	Issa	Riyadh
8	Maha	Khalid	AlAhsa
9	Eman	Ahmed	Dhahran
10	Bader	Abdullah	Madinah



The screenshot shows a SQL IDE with a query editor and a result grid. The query is a nested query that finds the cashier with the maximum salary. The result grid displays 1 row of data.

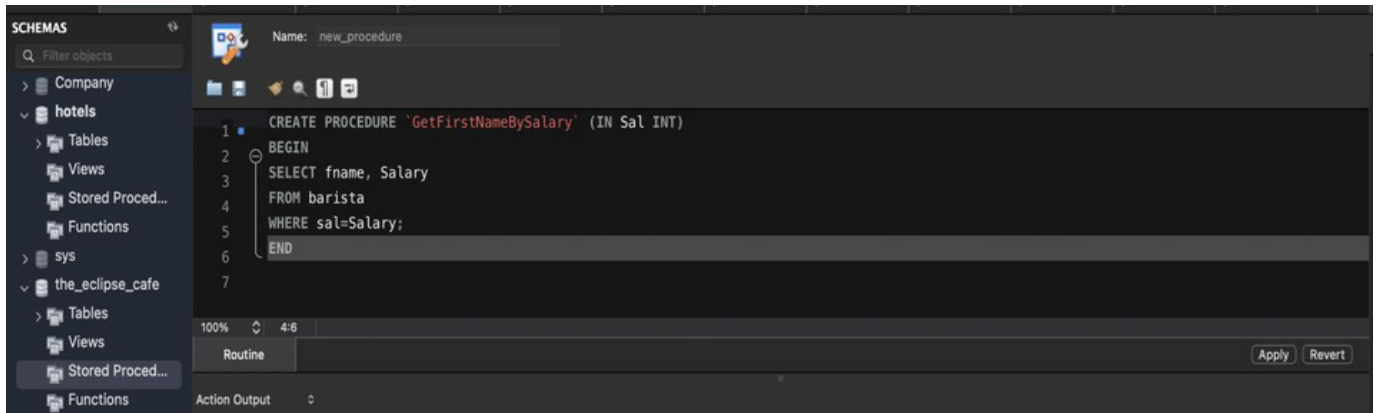
```
1 SELECT * FROM the_eclipse_cafe.cashier;  
2 SELECT Lname  
3 FROM the_eclipse_cafe.cashier  
4 WHERE Salary =(SELECT MAX(Salary)  
5 FROM the_eclipse_cafe.cashier);  
6
```

100% 32:5

Result Grid Filter Rows: Search Export:

Lname
Fahad

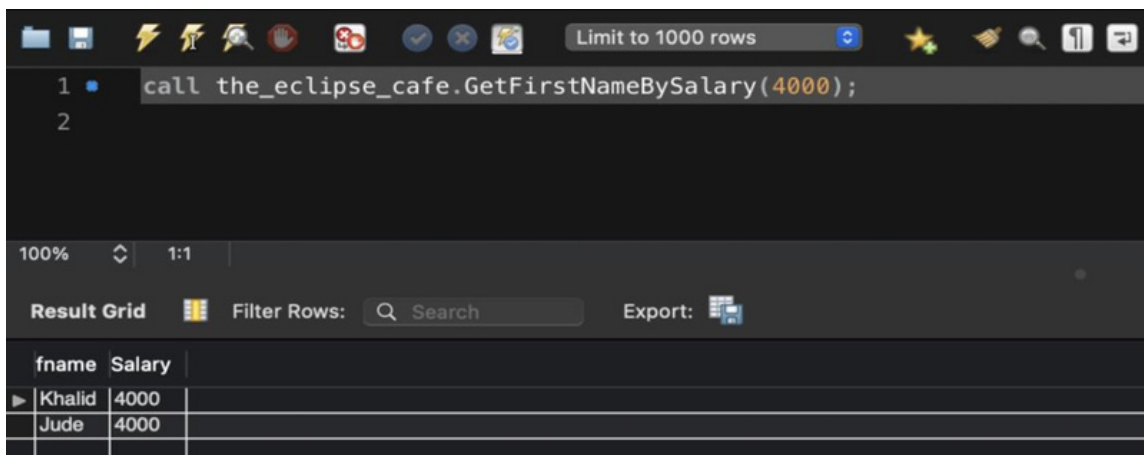
15- Function and Stored Procedure



The screenshot shows a SQL IDE interface. On the left, a 'SCHEMAS' pane lists a database named 'the_eclipse_cafe' with tables, views, and stored procedures. The main editor is titled 'Name: new_procedure' and contains the following SQL code:

```
1 CREATE PROCEDURE `GetFirstNameBySalary` (IN Sal INT)
2 BEGIN
3     SELECT fname, Salary
4     FROM barista
5     WHERE sal=Salary;
6 END
```

Below the code editor, there are zoom controls (100%, 4:6), a 'Routine' tab, and 'Apply' and 'Revert' buttons.

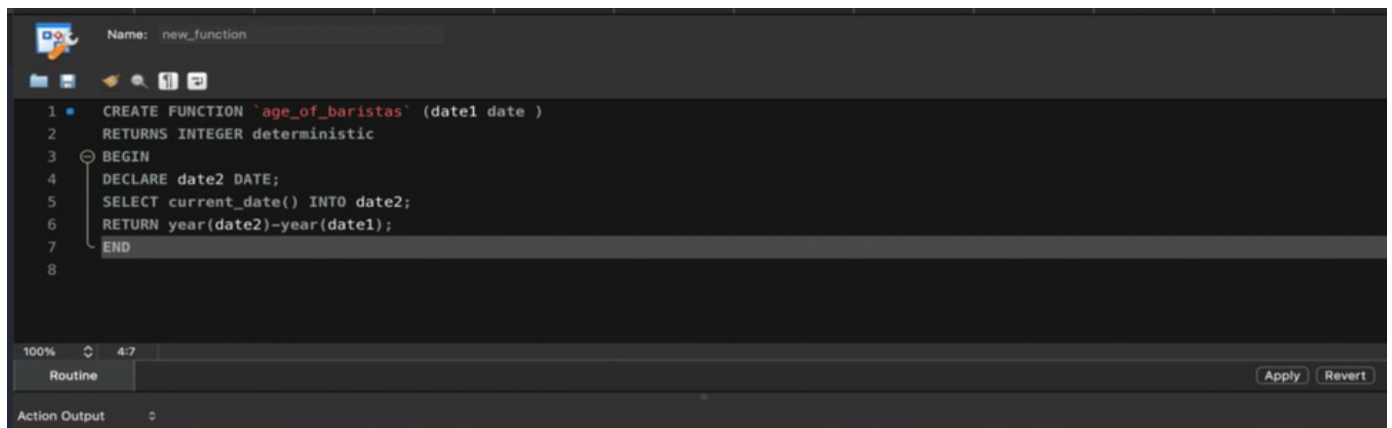


The screenshot shows the SQL IDE with the following SQL code entered:

```
1 call the_eclipse_cafe.GetFirstNameBySalary(4000);
2
```

Below the code editor, there are zoom controls (100%, 1:1), a 'Result Grid' tab, and a search bar. The results are displayed in a table:

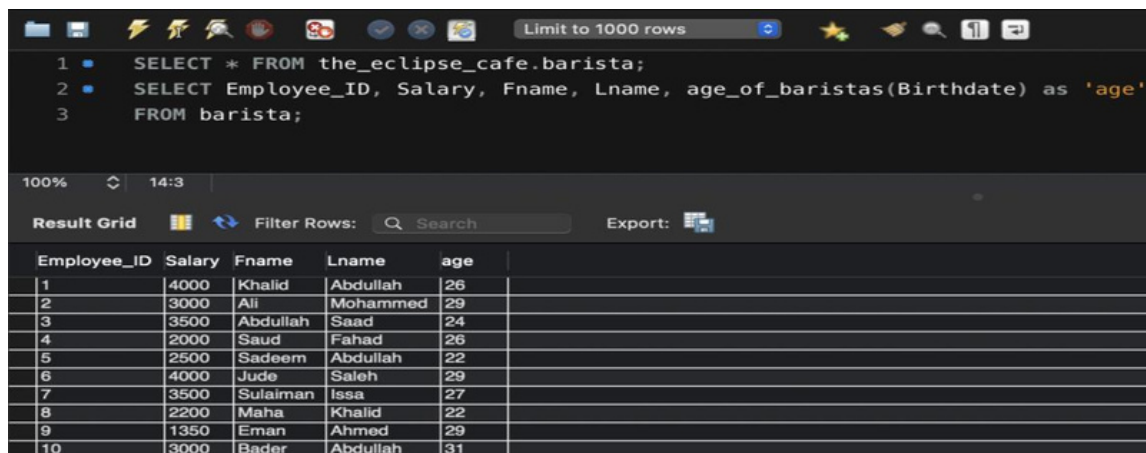
fname	Salary
Khalid	4000
Jude	4000



The screenshot shows a SQL IDE interface. On the left, a 'SCHEMAS' pane lists a database named 'the_eclipse_cafe' with tables, views, and functions. The main editor is titled 'Name: new_function' and contains the following SQL code:

```
1 CREATE FUNCTION `age_of_baristas` (date1 date )
2 RETURNS INTEGER deterministic
3 BEGIN
4     DECLARE date2 DATE;
5     SELECT current_date() INTO date2;
6     RETURN year(date2)-year(date1);
7 END
```

Below the code editor, there are zoom controls (100%, 4:7), a 'Routine' tab, and 'Apply' and 'Revert' buttons.



The screenshot shows the SQL IDE with the following SQL code entered:

```
1 SELECT * FROM the_eclipse_cafe.barista;
2 SELECT Employee_ID, Salary, Fname, Lname, age_of_baristas(Birthdate) as 'age'
3 FROM barista;
```

Below the code editor, there are zoom controls (100%, 14:3), a 'Result Grid' tab, and a search bar. The results are displayed in a table:

Employee_ID	Salary	Fname	Lname	age
1	4000	Khalid	Abdullah	26
2	3000	Ali	Mohammed	29
3	3500	Abdullah	Saad	24
4	2000	Saud	Fahad	26
5	2500	Sadeem	Abdullah	22
6	4000	Jude	Saleh	29
7	3500	Sulaiman	Issa	27
8	2200	Maha	Khalid	22
9	1350	Eman	Ahmed	29
10	3000	Bader	Abdullah	31

16- Trigger

```
2
3   delimiter //
4   create trigger Product_quantity
5   before insert on Products
6   for each row
7   begin
8   if new.Product_price>5 then
9   set new.Product_quantity=20;
10  end if;
11  end //
```

100% 7:11

Result Grid Filter Rows: Search Edit: Export/Import:

Product_ID	Product_name	Product_quantity	Product_price	Product_availability	Supplier_ID
1	Coffee	1000	5	available	1
2	Cups	500	3.5	available	2
3	Syrup	25	4.5	available	3
4	Tea	200	5	available	4
5	Sugar	400	2.5	available	5
6	masks	250	2	available	6
7	gloves	500	2	available	7
8	Tissue	1000	0.25	available	8
9	soap	50	3	available	9
10	sweets	45	5	not available	10
NULL	NULL	NULL	NULL	NULL	NULL

We add a product and its price 6

```
3   insert into Products values('11','milk', '50', '6', 'available','11');
4   Select *
5   from Products
6   where Product_price>4;
```

00% 23:6

Result Grid Filter Rows: Search Edit: Export/Import:

Product_ID	Product_name	Product_quantity	Product_price	Product_availability	Supplier_ID
1	Coffee	1000	5	available	1
3	Syrup	25	4.5	available	3
4	Tea	200	5	available	4
10	sweets	45	5	not available	10
11	milk	20	6	available	11
NULL	NULL	NULL	NULL	NULL	NULL

17- View

```
3 • create view max_price
4   as
5   select Item_name, Availability
6   from Items
7   where Item_Price>250
```

item_name	availability
beans	available
milk	available
syrup	not available
tea	available