

Population

Visuals of the table creation codes are working. It visualizes the population separately for each table.

1

select *

2

from BILL ;

3

Results

Explain

Describe

Saved SQL

History

BILL_ID	BILL_DATE	BILL_STATUS	BILL_PAYMENT_METHOD	ORDER_PRODUCT_ID
1	12/25/2022	unpaid	cash	1
2	12/25/2022	unpaid	cash	3
3	12/25/2022	unpaid	cash	5
4	12/25/2022	unpaid	cash	7
5	12/25/2022	unpaid	cash	9
6	12/25/2022	unpaid	cash	11
7	12/25/2022	unpaid	cash	13
8	12/25/2022	unpaid	cash	15
9	12/25/2022	unpaid	cash	17
10	12/25/2022	unpaid	cash	19

10 rows returned in 0.00 secondsDownload

1

select *

2

from FOOD_INGREDIENT ;

3

Results

Explain

Describe

Saved SQL

History

FOOD_INGREDIENT_ID	FOOD_ID	INGREDIENT_ID
1	1	1
2	1	11
3	1	3
4	1	4
5	1	5
6	2	6
7	2	6
8	2	7
9	2	8
10	2	9
11	4	5

SQL Commands

SchemaWKSP_NCR

LanguageSQL

Rows30

Clear Command

Find Tables

Save

Run

1

select *

2

from ORDER_PRODUCT ;

3

Results

Explain

Describe

Saved SQL

History

ORDER_PRODUCT_ID	ORDER_ID	PRODUCT_ID	ORDER_PRODUCT_STATUS
1	1	1	prepared
2	1	2	unprepared
3	2	3	prepared
4	3	4	unprepared
5	4	5	prepared
6	5	6	unprepared
7	6	7	prepared
8	7	8	unprepared
9	8	9	prepared
10	9	10	unprepared
11	10	11	prepared

LanguageSQLRows30Clear CommandFind TablesSaveRun

1select *
2from PRODUCT ;
3

Results

ExplainDescribeSaved SQLHistory

PRODUCT_ID	PRODUCT_COST	PRODUCT_PRICE	PRODUCT_DESCRIPTION	PRODUCT_TYPE
1	50	60	Fish and chips: Fried fish and chips served with tartar sauce.	food
2	25	30	Roast beef: Roasted beef served with roast potatoes and vegetables.	food
3	10	15	Tea: A hot beverage made by infusing the dried leaves of the tea plant in hot water.	beverage
4	30	35	Bangers and mash: Sausages served with mashed potatoes.	food
5	45	50	Toad in the hole: Sausages baked in a batter mixture and served with gravy.	food
6	20	25	Coffee: A hot beverage made by brewing roasted and ground coffee beans.	beverage
7	35	40	Shepherd's pie: Mince meat topped with mashed potatoes and baked.	food
8	40	45	Ploughmans lunch: A British cold meal consisting of cheese, pickles, and bread.	food
9	15	20	Soda: A sweet, carbonated drink.	beverage
10	55	60	Beef Wellington: Beef fillet coated in pâté and duxelles, wrapped in puff pastry, and baked.	food
11	30	35	Eton mess: A dessert made with meringue, whipped cream, and fruit.	food

1select *
2from ORDER ;
3

Results

ExplainDescribeSaved SQLHistory

ORDER_ID	TABLE_ID	ORDER_DATE
1	1	01/01/2022
2	2	01/01/2022
3	3	01/01/2022
4	4	01/01/2022
5	5	01/01/2022
6	6	01/01/2022
7	7	01/01/2022
8	8	01/01/2022
9	9	01/01/2022
10	10	01/01/2022
11	11	01/01/2022

1select *
2from RESERVATION ;
3

Results

ExplainDescribeSaved SQLHistory

RESERVATION_ID	RESERVATION_CUSTOMER_NAME	RESERVATION_TIME	TABLE_ID	RESERVATION_PHONE_NUMBER
1	Ramiz Karaaoki	01/01/2022	1	555-555-55-55
2	Ezai Bayraktar	01/01/2022	2	554-554-55-55
3	Ali Kırıcı	01/01/2022	3	578-578-57-57
4	Cengiz Atay	01/01/2022	4	543-543-54-54
5	Mehmet Yılmaz	01/01/2022	5	555-888-70-60
6	Ahmet Öztürk	01/01/2022	6	555-888-70-67
7	Mehmet Güneş	01/01/2022	7	555-888-70-68
8	Ali Yılmaz	01/01/2022	8	555-888-70-69
9	Cemal Kaya	01/01/2022	9	555-888-70-70
10	Elif Demir	01/01/2022	10	555-888-70-71
11	Can Yılmaz	01/01/2022	11	555-888-70-72

Language
SQL
Rows
30
Clear Command
Find Tables

Save
Run

```

1 select *
2 from SUPPLIER ;
3

```

Results

Explain

Describe

Saved SQL

History

SUPPLIER_ID	SUPPLIER_NAME	SUPPLIER_TELEPHONE_NUMBER
1	Gallata Greengrocer	0531-430-23-22
2	Limnan Butcher	0521-123-12-34
3	Miran Market	0523-342-23-25
4	John Butcher	0531-233-23-12
5	Divan Greengrocer	0532-234-23-12
6	Last Butcher	0521-231-34-23
7	Lokal Market	0521-212-23-21
8	Hamedan Greengrocer	0521-212-23-34
9	Garden Fisherman	0512-232-34-35
10	Ehli Keyf Butcher	0521-231-21-21
11	Mozart Market	0521-231-23-33

Queries

QUERY 1 : All the unpaid bills

QUERY 2 : Suppliers and Ingredients

```
1 select INGREDIENT.INGREDIENT_NAME as INGREDIENT_NAME,
2        SUPPLIER.SUPPLIER_NAME as SUPPLIER_NAME
3        from INGREDIENT INGREDIENT,
4        SUPPLIER SUPPLIER
5        where INGREDIENT.SUPPLIER_ID=SUPPLIER.SUPPLIER_ID
```

Results Explain Describe Saved SQL History

INGREDIENT_NAME	SUPPLIER_NAME
Fish	Garden Fisherman
Olive	Miran Market
Salt	Lokal Market
Oil	Harap Grocer
Potatoes	Hanedan Greengrocer
Beef	Last Butcher
Carrots	Düdübn Greengrocer
Onions	Divan Greengrocer
Parsnips	Galata Greengrocer
Sausages	Ehli Keyfi Butcher

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.02 seconds [Download](#)

QUERY 3 : Profit per table

Q

As

```
1 select TABLE_TABLE_ID as TABLE_ID,
2 count(BILL_BILL_ID) as BILLS,
3 sum(PRODUCT_PRODUCT_PRICE) - sum(PRODUCT_PRODUCT_COST) as TOTAL_PROFIT
4 from PRODUCT_PRODUCT,
5 BILL_BILL,
6 ORDER_PRODUCT_ORDER_PRODUCT,
7 ORDER_ORDER,
8 TABLE_TABLE,
9 where ORDER_TABLE_ID=TABLE_TABLE_ID
10 and ORDER_ORDER_ID=ORDER_PRODUCT_ORDER_ID
11 and BILL_ORDER_PRODUCT_ID=ORDER_PRODUCT_ORDER_PRODUCT_ID
12 and ORDER_PRODUCT_PRODUCT_ID=PRODUCT_PRODUCT_ID
13 group by TABLE_ID
```

Results

Explain

Describe

Saved SQL

History

TABLE_ID	BILLS	TOTAL_PROFIT
6	1	5
14	1	5
1	1	10
2	1	5
8	1	5
12	1	5
4	1	5
10	1	5
18	1	5
16	1	5

1 rows returned in 0.00 seconds

QUERY 4 : All bills for a given table

Language SQL Rows 10 Clear Command Find Tables Save Run

Q

As

```
1 SELECT B.*
2 FROM BILL B
3 JOIN ORDER_PRODUCT OP ON B.ORDER_PRODUCT_ID = OP.ORDER_PRODUCT_ID
4 JOIN ORDER_O ON OP.ORDER_ID = O.ORDER_ID
5 WHERE O.TABLE_ID = :chosen_table_id;
```

Results

Explain

Describe

Saved SQL

History

BILL_ID	BILL_DATE	BILL_STATUS	BILL_PAYMENT_METHOD	ORDER_PRODUCT_ID
1	12/25/2022	unpaid	cash	1

1 rows returned in 0.02 seconds Download

QUERY 5 : Total cost price profit

Q

As

```
1 select count(BILL_BILL_ID) as "TOTAL PAID BILLS",
2 sum(PRODUCT_PRODUCT_COST) as "TOTAL COST",
3 sum(PRODUCT_PRODUCT_PRICE) as "TOTAL PRICE",
4 sum(PRODUCT_PRODUCT_PRICE) - sum(PRODUCT_PRODUCT_COST) as "TOTAL PROFIT"
5 from BILL_BILL,
6 PRODUCT_PRODUCT
7 where BILL_ORDER_PRODUCT_ID=PRODUCT_PRODUCT_ID
```

Results

Explain

Describe

Saved SQL

History

TOTAL PAID BILLS	TOTAL COST	TOTAL PRICE	TOTAL PROFIT
10	290	345	55

1 rows returned in 0.02 seconds Download

QUERY 6 : Tables for which a waiter is responsible

SQL Commands

Schema: WKSP_NCR

Language: SQL Rows: 10 Clear Command Find Tables Save Run

SQL Editor

```
1 SELECT t.*
2 FROM TABLE t
3 JOIN WAITER w ON w.WAITER_ID = t.WAITER_ID
4 WHERE w.WAITER_ID = :waiter_id
```

Results

Explain Describe Saved SQL History

TABLE_ID	TABLE_SIZE	TABLE_LOCATION	TABLE_FLOOR	WAITER_ID
1	4	inside	1	1
5	4	terrace	1	1
9	4	garden	2	1

3 rows returned in 0.04 seconds Download

QUERY 7 : Tables per floor

SQL Editor

```
1 select TABLE_TABLE_FLOOR as FLOOR,
2 count(TABLE_1.TABLE_FLOOR) as TABLES
3 from TABLE_TABLE_1,
4 TABLE_TABLE
5 where TABLE_1.TABLE_FLOOR=TABLE_1.TABLE_FLOOR
6 group by TABLE_TABLE_FLOOR
```

Results

Explain Describe Saved SQL History

FLOOR	TABLES
1	100
2	25
4	1
5	1
8	1
0	4

6 rows returned in 0.01 seconds Download

QUERY 8 : All prepared order products and their table

SQL Editor

```
1 SELECT op.ORDER_PRODUCT_ID, op.ORDER_ID, op.PRODUCT_ID, op.ORDER_PRODUCT_STATUS, o.TABLE_ID
2 FROM ORDER_PRODUCT op
3 INNER JOIN ORDER o ON op.ORDER_ID = o.ORDER_ID
4 WHERE op.ORDER_PRODUCT_STATUS = 'prepared'
```

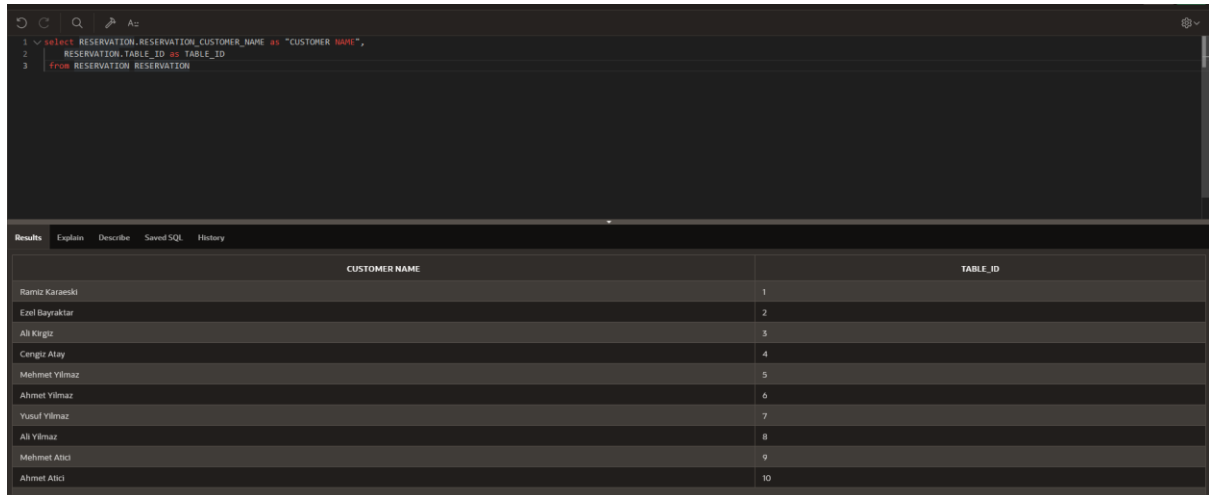
Results

Explain Describe Saved SQL History

ORDER_PRODUCT_ID	ORDER_ID	PRODUCT_ID	ORDER_PRODUCT_STATUS	TABLE_ID
1	1	1	prepared	1
3	2	3	prepared	2
5	4	5	prepared	4
7	6	7	prepared	6
9	8	9	prepared	8
11	10	11	prepared	10
13	12	13	prepared	12
15	14	15	prepared	14
17	16	17	prepared	16
19	18	19	prepared	18

10 rows returned in 0.04 seconds Download

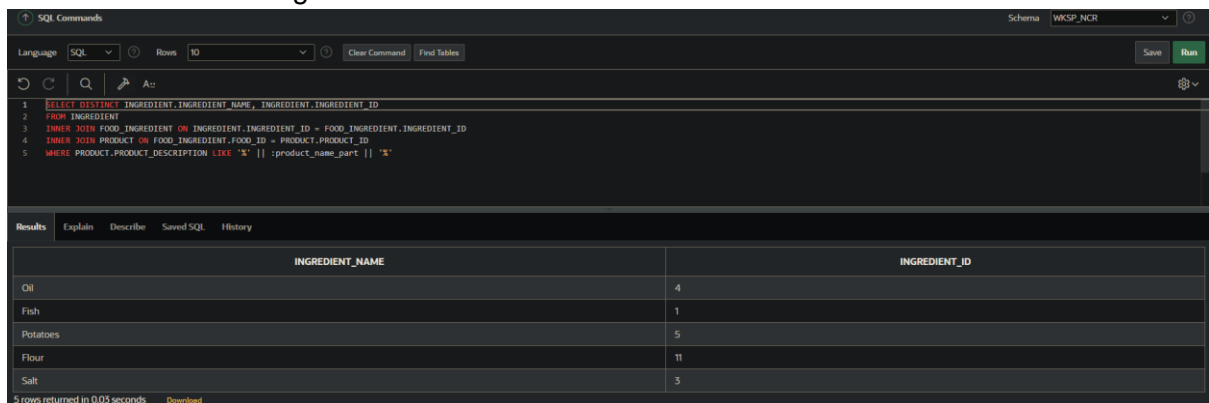
QUERY 9 : Reservations and their tables



The screenshot shows a SQL IDE interface. The top panel contains a query: `1 select RESERVATION.RESERVATION_CUSTOMER_NAME as "CUSTOMER NAME",`
`2 RESERVATION.TABLE_ID as TABLE_ID`
`3 from RESERVATION RESERVATION`. The bottom panel shows the results of the query in a table with two columns: **CUSTOMER NAME** and **TABLE_ID**. The results are as follows:

CUSTOMER NAME	TABLE_ID
Ramiz Karaeski	1
Ezel Bayraktar	2
Ali Kirge	3
Cengiz Atay	4
Mehmet Yilmaz	5
Ahmet Yilmaz	6
Yusuf Yilmaz	7
Ali Yilmaz	8
Mehmet Atici	9
Ahmet Atici	10

QUERY 10 : Get all ingredients for a chosen food

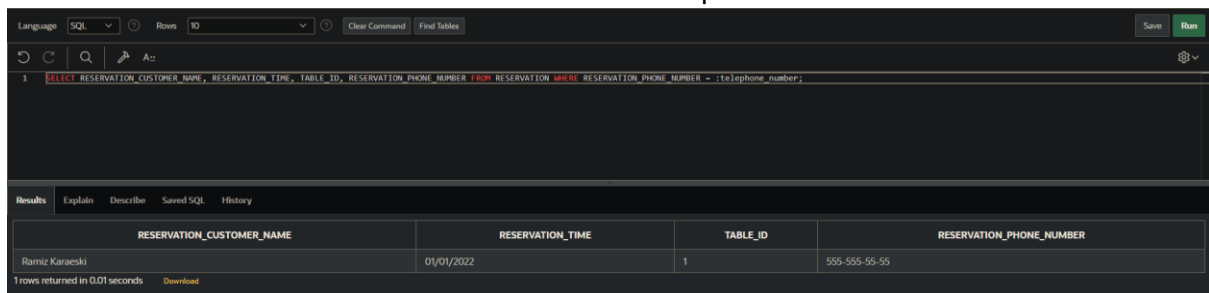


The screenshot shows a SQL IDE interface. The top panel contains a query: `1 SELECT DISTINCT INGREDIENT.INGREDIENT_NAME, INGREDIENT.INGREDIENT_ID`
`2 FROM INGREDIENT`
`3 INNER JOIN FOOD_INGREDIENT ON INGREDIENT.INGREDIENT_ID = FOOD_INGREDIENT.INGREDIENT_ID`
`4 INNER JOIN PRODUCT ON FOOD_INGREDIENT.FOOD_ID = PRODUCT.PRODUCT_ID`
`5 WHERE PRODUCT.PRODUCT_DESCRIPTION LIKE '%':product_name_part || '%'`. The bottom panel shows the results of the query in a table with two columns: **INGREDIENT_NAME** and **INGREDIENT_ID**. The results are as follows:

INGREDIENT_NAME	INGREDIENT_ID
Oil	4
Fish	1
Potatoes	5
Flour	11
Salt	3

5 rows returned in 0.05 seconds [Download](#)

QUERY 11 : Get reservation information from the telephone number



The screenshot shows a SQL IDE interface. The top panel contains a query: `1 SELECT RESERVATION_CUSTOMER_NAME, RESERVATION_TIME, TABLE_ID, RESERVATION_PHONE_NUMBER FROM RESERVATION WHERE RESERVATION_PHONE_NUMBER = :telephone_number;`. The bottom panel shows the results of the query in a table with four columns: **RESERVATION_CUSTOMER_NAME**, **RESERVATION_TIME**, **TABLE_ID**, and **RESERVATION_PHONE_NUMBER**. The results are as follows:

RESERVATION_CUSTOMER_NAME	RESERVATION_TIME	TABLE_ID	RESERVATION_PHONE_NUMBER
Ramiz Karaeski	01/01/2022	1	555-555-55-55

1 rows returned in 0.01 seconds [Download](#)

QUERY 12 : Get the bills for each table

Language: SQL Rows: 10

```
1 select TABLE_ID as "TABLE NUMBER",
2 BILL_DATE as "BILL DATE",
3 BILL_PAYMENT_METHOD as "PAYMENT METHOD",
4 PRODUCT_PRICE as "BILL PRICE"
5 from PRODUCT_PRODUCT,
6 ORDER_PRODUCT_ORDER_PRODUCT,
7 ORDER_ORDER,
8 BILL_BILL,
9 TABLE_TABLE
10 where ORDER_ID=TABLE_ID, TABLE_ID
11 and ORDER_ORDER_ID=ORDER_PRODUCT_ORDER_ID
12 and ORDER_PRODUCT_ORDER_PRODUCT_ID=BILL_ORDER_PRODUCT_ID
13 and PRODUCT_PRODUCT_ID=ORDER_PRODUCT_PRODUCT_ID
```

Results Explain Describe Saved SQL History

TABLE NUMBER	BILL DATE	PAYMENT METHOD	BILL PRICE
1	12/25/2022	cash	60
2	12/25/2022	cash	15
4	12/25/2022	cash	50
6	12/25/2022	cash	40
8	12/25/2022	cash	20
10	12/25/2022	cash	35
12	12/25/2022	cash	30
14	12/25/2022	cash	40
16	12/25/2022	cash	20
18	12/25/2022	cash	35

QUERY 13 :List of all products in ascending order of price

Language: SQL Rows: 30

```
1 SELECT * FROM PRODUCT ORDER BY PRODUCT_PRICE ASC;
```

Results Explain Describe Saved SQL History

PRODUCT_ID	PRODUCT_COST	PRODUCT_PRICE	PRODUCT_DESCRIPTION	PRODUCT_TYPE
18	10	15	Strawberry Shortcake: A sweet, fluffy cake topped.	food
14	10	15	Water: A clear, tasteless, odorless, and nearly colorless liquid.	beverage
3	10	15	Tea: A hot beverage made by infusing the dried leaves of the tea plant in hot water.	beverage
20	10	20	Chocolate Cake - a sweet, moist cake made with cocoa powder and chocolate chips, and topped with a rich chocolate frosting.	food
9	15	20	Soda: A sweet, carbonated drink.	beverage
17	15	20	Cola: A sweet, carbonated drink flavored with cola nuts.	beverage
6	20	25	Coffee: A hot beverage made by brewing roasted and ground coffee beans.	beverage
13	25	30	Yorkshire pudding: A savory puff pastry made from a batter of eggs, flour, and milk.	food
2	25	30	Roast beef: Roasted beef served with roast potatoes and vegetables.	food
11	30	35	Eton mess: A dessert made with meringue, whipped cream, and fruit.	food
4	30	35	Burgers and mash: Sausages served with mashed potatoes.	food

QUERY 14 : Most used supplier

```
1 select SUPPLIER.SUPPLIER_NAME as "SUPPLIER NAME",
2 count(INGREDIENT.INGREDIENT_NAME) as "NUMBER OF INGREDIENTS"
3 from INGREDIENT_INGREDIENT,
4 SUPPLIER_SUPPLIER
5 where SUPPLIER.SUPPLIER_ID=INGREDIENT.SUPPLIER_ID
6 group by SUPPLIER.SUPPLIER_NAME
7 order by count(INGREDIENT.INGREDIENT_NAME) DESC;
```

Results Explain Describe Saved SQL History

SUPPLIER NAME	NUMBER OF INGREDIENTS
Harap Grocer	9
Last Butcher	5
Lokal Market	4
Mundo Grocer	4
Durban Green Grocer	2
Ervela Market	2
Elli Keyf Butcher	2
Divan Green Grocer	2
Galata Green Grocer	1
Rainbow Market	1

QUERY 15 : Reservation in given date

Language
SQL
Rows
50
Clear Command
Find Tables
Save
Run

```

1 SELECT RESERVATION_ID, RESERVATION_TIME, SUBSTR(TO_CHAR(RESERVATION_TIME, 'DD-MON-YYYY HH24:MI'), 12, 6) AS RESERVATION_HOUR
2 FROM RESERVATION;

```

Results

Explain

Describe

Saved SQL

History

	RESERVATION_ID	RESERVATION_TIME	RESERVATION_HOUR
1		01/01/2022	21:00
2		01/01/2022	20:00
3		01/01/2022	25:00
4		01/01/2022	00:00
5		01/01/2022	22:00
6		01/01/2022	22:00
7		01/01/2022	22:00
8		01/01/2022	22:00
9		01/01/2022	22:00
10		01/01/2022	22:00
11		01/01/2022	22:00

QUERY 16 -1: Trigger Test Script

```
INSERT INTO WAITER (WAITER_ID, WAITER_NAME, WAITER_PHONE_NUMBER, WAITER_DATE_OF_BIRTH) VALUES(21, 'Niyazi KABA', '0501-234-56-79', TO_DATE('12/15/2009', 'MM/DD/YYYY'));

INSERT INTO WAITER (WAITER_ID, WAITER_NAME, WAITER_PHONE_NUMBER, WAITER_DATE_OF_BIRTH) VALUES(16, 'Fatma Gul', '0503-333-33-32', TO_DATE('04/01/1995', 'MM/DD/YYYY'));

INSERT INTO TABLE (TABLE_ID, TABLE_SIZE, TABLE_FLOOR, TABLE_LOCATION, WAITER_ID) VALUES(16, 4, 3, 'Inside', 16);

INSERT INTO RESERVATION (RESERVATION_ID, RESERVATION_CUSTOMER_NAME, RESERVATION_TIME, TABLE_ID, RESERVATION_PHONE_NUMBER) VALUES(21, 'Ahmet Bulut', TO_TIMESTAMP('01-JAN-2022 22:00', 'DD-MM-YYYY HH24:MI'), 16, '532-234-32-34');

INSERT INTO RESERVATION (RESERVATION_ID, RESERVATION_CUSTOMER_NAME, RESERVATION_TIME, TABLE_ID, RESERVATION_PHONE_NUMBER) VALUES(16, 'Salih Bayraktar', TO_TIMESTAMP('01-JAN-2022 22:00', 'DD-MM-YYYY HH24:MI'), 16, '555-888-70-77');

INSERT INTO PRODUCT (PRODUCT_COST, PRODUCT_PRICE, PRODUCT_DESCRIPTION, PRODUCT_TYPE) VALUES(25, 30, 'Cream of Mushroom Soup: Brimming with rich, savory flavor, it's meant to be enjoyed all on its own.', 'soup');

INSERT INTO FOOD_INGREDIENT (FOOD_ID, INGREDIENT_ID) VALUES (3, 1);

INSERT INTO TABLE (TABLE_ID, TABLE_SIZE, TABLE_FLOOR, TABLE_LOCATION, WAITER_ID) VALUES(16, 4, 3, 'pool', 16);

INSERT INTO PRODUCT (PRODUCT_COST, PRODUCT_PRICE, PRODUCT_DESCRIPTION, PRODUCT_TYPE) VALUES(-20, 40, 'Shepherd's pie: Mince meat topped with mashed potatoes and baked.', 'food');

INSERT INTO PRODUCT (PRODUCT_COST, PRODUCT_PRICE, PRODUCT_DESCRIPTION, PRODUCT_TYPE) VALUES(11, -20, 'Soda: A sweet, carbonated drink.', 'beverage');
```

QUERY 16-2 : Trigger Test Result

Number ↑↓	Elapsed	Statement	Feedback	Rows
1	0.04	INSERT INTO WAITER (WAITER_ID, WAITER_NAME, WAITER_PHONE_NUM	ORA-20001: Waiters must be at least 18 years old. ORA-06512: at "WKSP_NCR.WAITER_AGE_CHECK", line 5 ORA-04088: error during execution of trigger 'WKSP_NCR.WAITER_AGE_CHECK' ORA-06512: at "SYS.DBMS_SQL", line 1721	-
2	0.00	INSERT INTO WAITER (WAITER_ID, WAITER_NAME, WAITER_PHONE_NUM	1 row(s) inserted.	1
3	0.03	INSERT INTO TABLE (TABLE_ID, TABLE_SIZE, TABLE_FLOOR, TABLE	1 row(s) inserted.	1
4	0.05	INSERT INTO RESERVATION (RESERVATION_ID, RESERVATION_CUSTOME	1 row(s) inserted.	1
5	0.01	INSERT INTO RESERVATION (RESERVATION_ID, RESERVATION_CUSTOME	ORA-20000: Table is already reserved at this time ORA-06512: at "WKSP_NCR.PREVENT_OVERLAPPING_RESERVATIONS", line 10 ORA-04088: error during execution of trigger 'WKSP_NCR.PREVENT_OVERLAPPING_RESERVATIONS' ORA-06512: at "SYS.DBMS_SQL", line 1721	-
6	0.02	INSERT INTO PRODUCT (PRODUCT_COST, PRODUCT_PRICE, PRODUCT_DE	ORA-20008: Product type must be "beverage" or "food". ORA-06512: at "WKSP_NCR.PRODUCT_TYPE_CHECK", line 3 ORA-04088: error during execution of trigger 'WKSP_NCR.PRODUCT_TYPE_CHECK' ORA-06512: at "SYS.DBMS_SQL", line 1721	-
7	0.05	INSERT INTO FOOD_INGREDIENT (FOOD_ID, INGREDIENT_ID) VALUES	ORA-20000: The product with the food_id specified in the food_ingredient table must be of type "food". ORA-06512: at "WKSP_NCR.CHECK_FOOD_TYPE_IN_FOOD_INGREDIENT", line 8 ORA-04088: error during execution of trigger 'WKSP_NCR.CHECK_FOOD_TYPE_IN_FOOD_INGREDIENT' ORA-06512: at "SYS.DBMS_SQL", line 1721	-
8	0.00	INSERT INTO TABLE (TABLE_ID, TABLE_SIZE, TABLE_FLOOR, TABLE	ORA-02290: check constraint (WKSP_NCR.TABLE_LOCATION_CHECK) violated ORA-06512: at "SYS.DBMS_SQL", line 1721	-
9	0.01	INSERT INTO PRODUCT (PRODUCT_COST, PRODUCT_PRICE, PRODUCT_DE	ORA-02290: check constraint (WKSP_NCR.PRODUCT_COST_CHECK) violated ORA-06512: at "SYS.DBMS_SQL", line 1721	-
10	0.01	INSERT INTO PRODUCT (PRODUCT_COST, PRODUCT_PRICE, PRODUCT_DE	ORA-02290: check constraint (WKSP_NCR.PRODUCT_PRICE_CHECK) violated ORA-06512: at "SYS.DBMS_SQL", line 1721	-
Download				

QUERY 17 : Bills in a given time period

BILL_ID	BILL_DATE	BILL_STATUS	BILL_PAYMENT_METHOD	ORDER_PRODUCT_ID
1	12/25/2022	unpaid	cash	1
2	12/25/2022	unpaid	cash	3
3	12/25/2022	unpaid	cash	5
4	12/25/2022	unpaid	cash	7
5	12/25/2022	unpaid	cash	9
6	12/25/2022	unpaid	cash	11
7	12/25/2022	unpaid	cash	13
8	12/25/2022	unpaid	cash	15
9	12/25/2022	unpaid	cash	17
10	12/25/2022	unpaid	cash	19

QUERY 18 : Table's ID in given location

TABLE_IDS
9
10
11
12
15
18

QUERY 19 : Unpaid Bills Per Table

TABLE_ID	BILL_ID	BILL_DATE	BILL_STATUS	BILL PRICE
1	1	12/25/2022	unpaid	60
2	2	12/25/2022	unpaid	15
4	3	12/25/2022	unpaid	50
6	4	12/25/2022	unpaid	40
8	5	12/25/2022	unpaid	20
10	6	12/25/2022	unpaid	35
12	7	12/25/2022	unpaid	30
14	8	12/25/2022	unpaid	40
16	9	12/25/2022	unpaid	20