DBMS Project (UCS310)



Restaurant Management System

Submitted to:

|  |  |
| --- | --- |
| Dr. Jhilik Bhattacharya | 2CS1 |

Submitted By:

|  |  |
| --- | --- |
| Sudansh Rana  Saksham Dhiman  Avneesh Jarangal  Karan Bagadi | 102217005  102217026  102217029  102217031 |

# INDEX

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Topic** | **Page No.** |
| 1. | Problem Statement | 3 |
| 2. | Overview | 3 |
| 3. | Requirement Analysis | 4 |
| 4. | ER - Diagram | 5 |
| 5. | ER To Table | 6 |
| 6. | Normalization | 7 |
| 7. | SQL Code and Output | 9 |
| 8. | PL/SQL Code and Output | 17 |
| 9. | Conclusion | 23 |
| 10. | References | 24 |

**Problem Statement**

The current manual restaurant management system leads to inefficiencies in ordering, inventory tracking, and customer service. A database management system is needed to streamline these processes and provide accurate and timely information for effective decision-making

# Overview

The Chef's Track is an innovative restaurant management system that leverages PL/SQL for its backend. The Diagrams and Flowcharts shown are made with the help of different designing tools.

The system offers a wide range of features, including a user-friendly menu display, streamlined order management, personalized customer profiles, tipping options, and much more. By automating processes such as inventory tracking and order management, the system allows for more accurate and timely decision-making.

The creation of customer profiles enables the restaurant to better understand its customers, leading to more personalized service. With its polished user interface and comprehensive features, The Chef's Track is the perfect solution for any restaurant looking to improve the customer experience and operational efficiency.

# Requirements Analysis

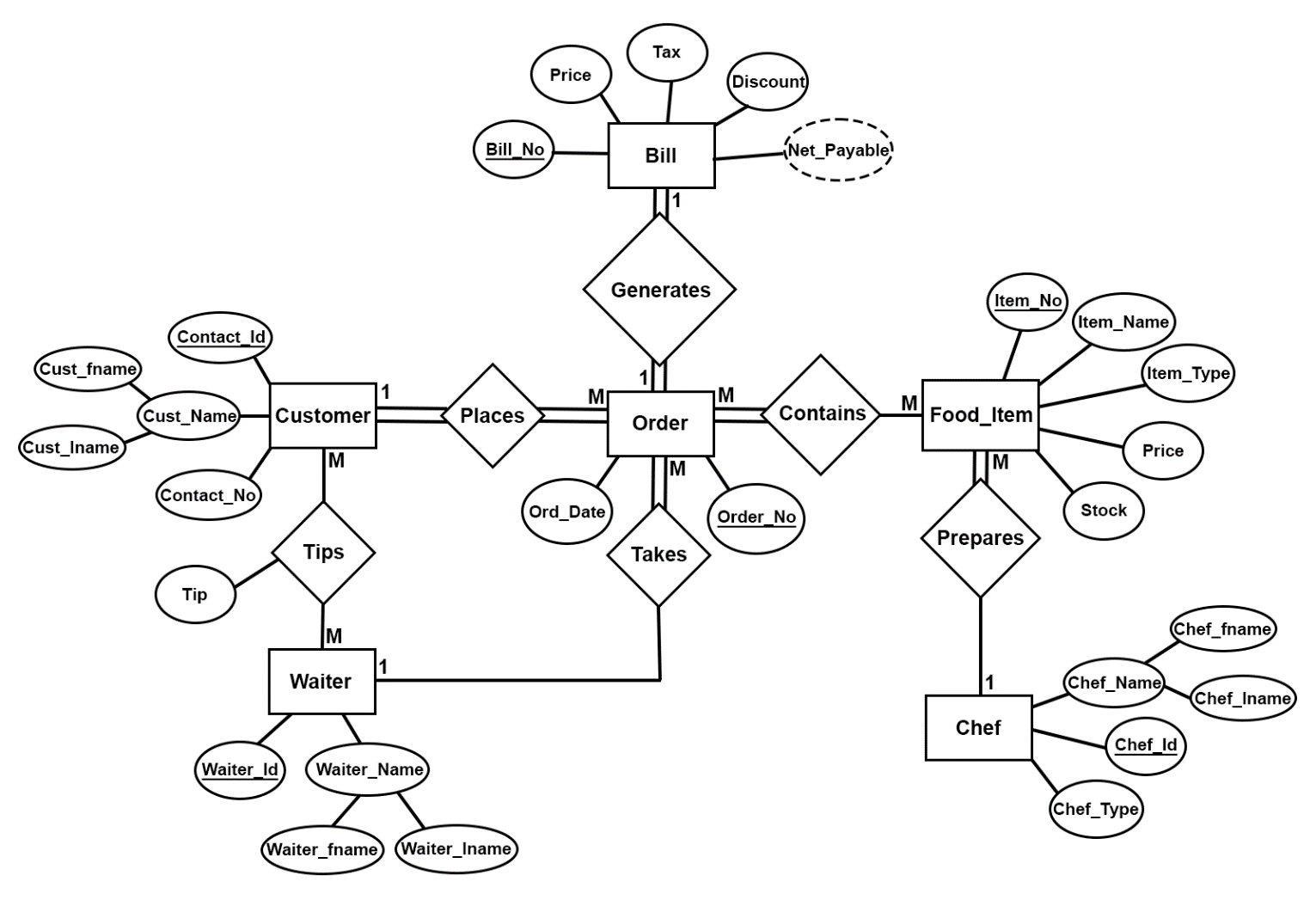
To develop a successful Restaurant Management System, we need to analyse the requirements of the system. The following are the primary requirements for the system:

1. **Menu Management:** The system must have the ability to display the menu, including the items, their prices, and descriptions.
2. **Customer Management:** The system should allow the staff to create customer profiles and store their information, including their name, contact details, etc.
3. **Order Management:** The system should provide features to place orders and add more items to order.
4. **Bill and Tips Management:** The system should be able to generate bills and also allow customers to tip the waiter.

## Software Requirements:

* 1. SQL
  2. PL/SQL
  3. Oracle Live SQL
  4. Combination of different designing tools to create ER-Diagram

# ER Diagram



# ER Diagram To Table

## Relation ‘Places’

Customer:- cust\_id , cust\_fname , cust\_lname, contact\_no Orders:- ord\_no , ord\_date , cust\_id (FK)

## Relation ‘Takes’

Orders:- ord\_no , ord\_date , waiter\_id (FK)

Waiter:- waiter\_id , waiter\_name , waiter\_lname

## Relation ‘Tips’

Customer:- cust\_id , cust\_fname , cust\_lname , contact\_no Waiter:- waiter\_id , waiter\_fname , waiter\_lname

Tips:- cust\_id (FK) , waiter\_id (FK) , tip

## Relation ‘Prepares’

Food:- item\_no , item\_name , item\_type , item\_price , item\_stock ,chef\_id (FK) Chef:- chef\_id , chef\_fname , chef\_lname , chef\_type

## Relation ‘Generates’

Orders:- ord\_no , ord\_date

Bill:- bill\_no , tot\_price , tax , discount , net\_payable , ord\_no (FK)

## Relation ‘Contains’

Food:- item\_no , item\_name , item\_type , item\_price , item\_stock Contains:- item\_no (FK) , ord\_no (FK)

Orders:- ord\_no , ord\_date

\_ \_ \_

# Normalization

Customer Table (Already in 3NF):

|  |  |  |  |
| --- | --- | --- | --- |
| cust\_id | cust\_fname | cust\_lname | contact\_no |
|  |  |  |  |

Waiter Table (Already in 3NF)

|  |  |  |
| --- | --- | --- |
| waiter\_id | waiter\_fname | waiter\_lname |
|  |  |  |

Tips Table (Already in 3NF)

|  |  |  |
| --- | --- | --- |
| waiter\_id (FK) | cust\_id (FK) | tip |
|  |  |  |

Orders Table (Already in 3NF)

|  |  |  |  |
| --- | --- | --- | --- |
| ord\_no | ord\_date | cust\_id (FK) | waiter\_id (FK) |
|  |  |  |  |

Chef Table (Already in 3NF)

|  |  |  |  |
| --- | --- | --- | --- |
| chef\_id | chef\_fname | chef\_lname | chef\_type |
|  |  |  |  |

Food Table (in 2NF)

As item\_no → item\_type and item\_type → chef\_id

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| item\_no | item\_name | item\_type | item\_price | item\_stock | chef\_id (FK) |
|  |  |  |  |  |  |

Breaking it into further tables Food Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| item\_no | item\_name | item\_type | item\_price | item\_stock |
|  |  |  |  |  |

Prepares Table:

|  |  |
| --- | --- |
| item\_type | chef\_id (FK) |
|  |  |

Contains Table (Already in 3NF)

|  |  |
| --- | --- |
| ord\_no (FK) | item\_no (FK) |
|  |  |

Bill Table (Already in 3NF)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bill\_no | tot\_price | tax | discount | net\_payable | ord\_no (FK) |
|  |  |  |  |  |  |

# SQL Code

## Creation of Tables

create table waiter(

waiter\_id integer primary key,

waiter\_fname varchar(50) not null,

waiter\_lname varchar(50)

);

create table customer(

cust\_id integer primary key,

cust\_fname varchar(50) not null,

cust\_lname varchar(50),

contact\_no integer

);

create table tips(

waiter\_id integer references waiter(waiter\_id),

cust\_id integer references customer(cust\_id),

tips integer

);

create table orders(

ord\_no integer primary key,

rd\_date date not null,

cust\_id integer references customer(cust\_id),

waiter\_id integer references waiter(waiter\_id)

);

create table chef(

chef\_id integer primary key,

chef\_fname varchar(50) not null,

chef\_lname varchar(50),

chef\_type varchar(50) not null

);

create table food(

item\_no integer primary key,

item\_name varchar(50) not null,

item\_type varchar(50) not null,

item\_price integer not null,

item\_stock integer

);

create table contains(

ord\_no integer references orders(ord\_no),

item\_no integer references food(item\_no)

);

create table prepares(

item\_type varchar(50) primary key,

chef\_id integer references chef(chef\_id)

);

create table bill(

bill\_no integer primary key,

total\_price integer not null,

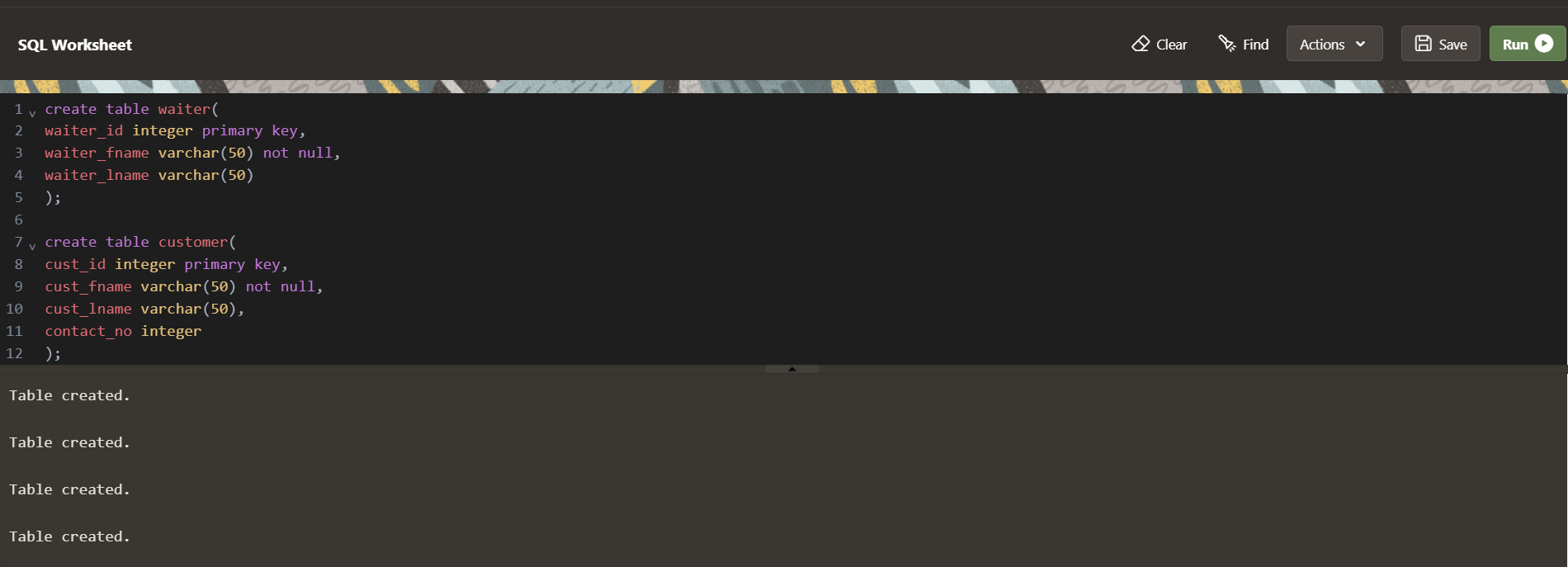
tax float default 5,

discount integer default 0,

net\_payable float as (total\_price+(total\_price\*tax/100)-(total\_price\*discount/100)),

ord\_no integer references orders(ord\_no)

);



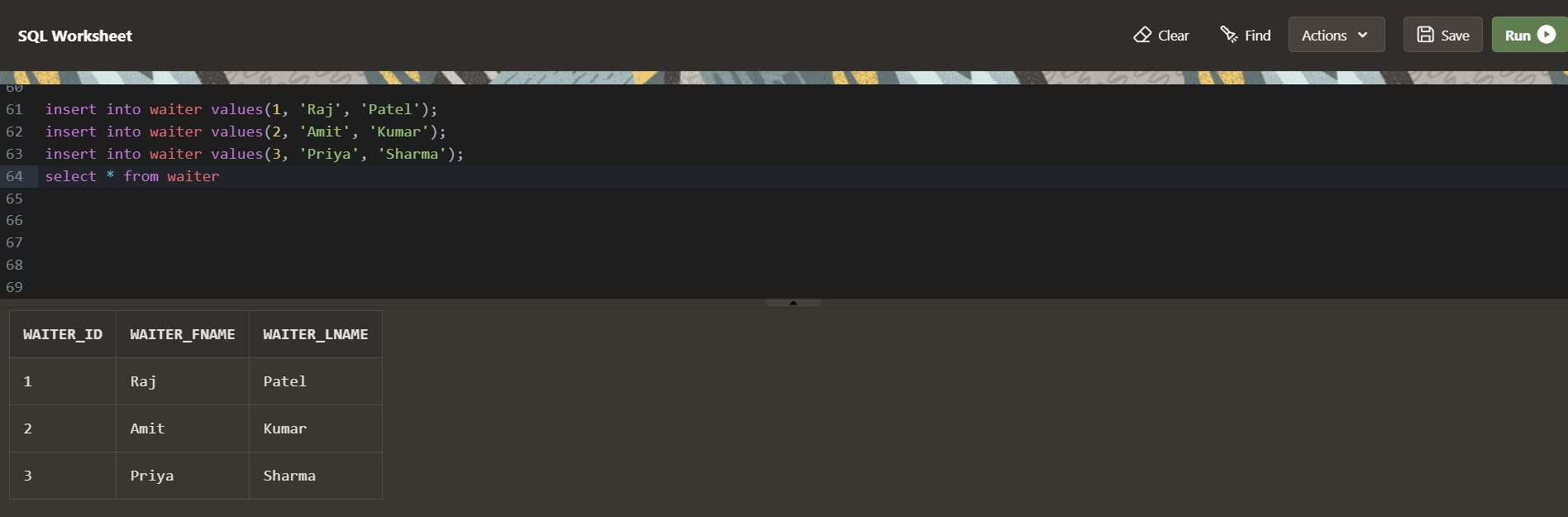
## Insertion Statements:

insert into waiter values(1, 'Raj', 'Patel');

insert into waiter values(2, 'Amit', 'Kumar');

insert into waiter values(3, 'Priya', 'Sharma');

select \* from waiter



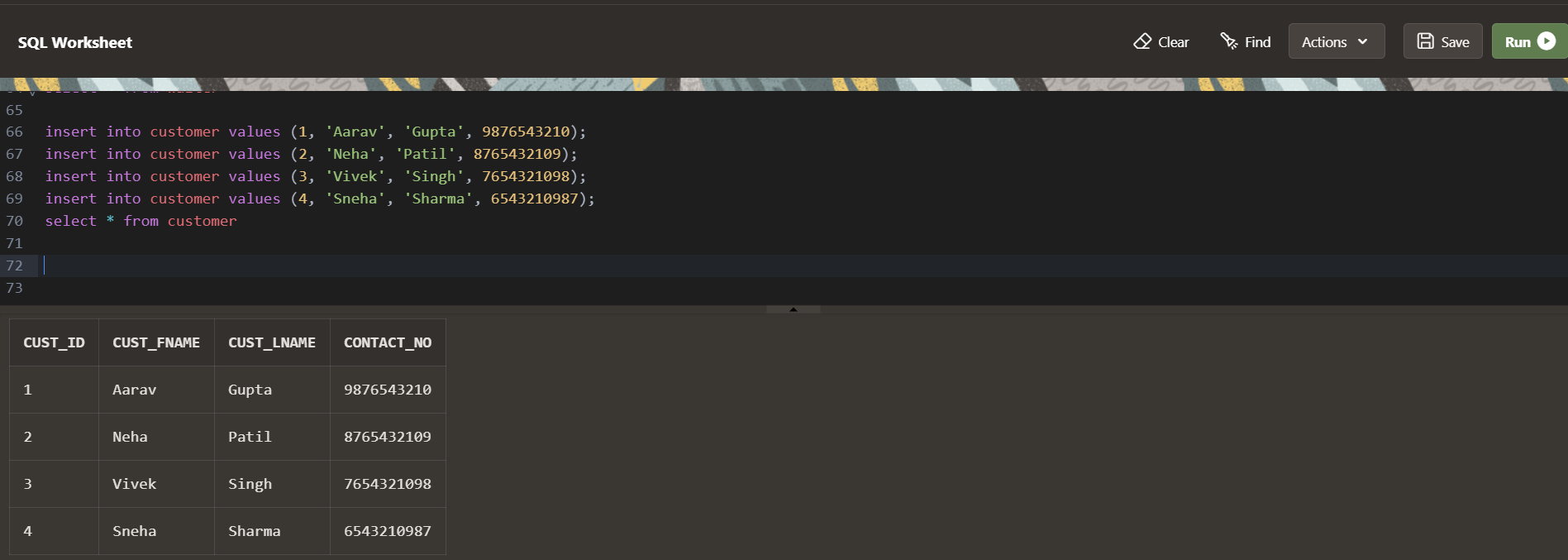
insert into customer values (1, 'Aarav', 'Gupta', 9876543210);

insert into customer values (2, 'Neha', 'Patil', 8765432109);

insert into customer values (3, 'Vivek', 'Singh', 7654321098);

insert into customer values (4, 'Sneha', 'Sharma', 6543210987);

select \* from customer

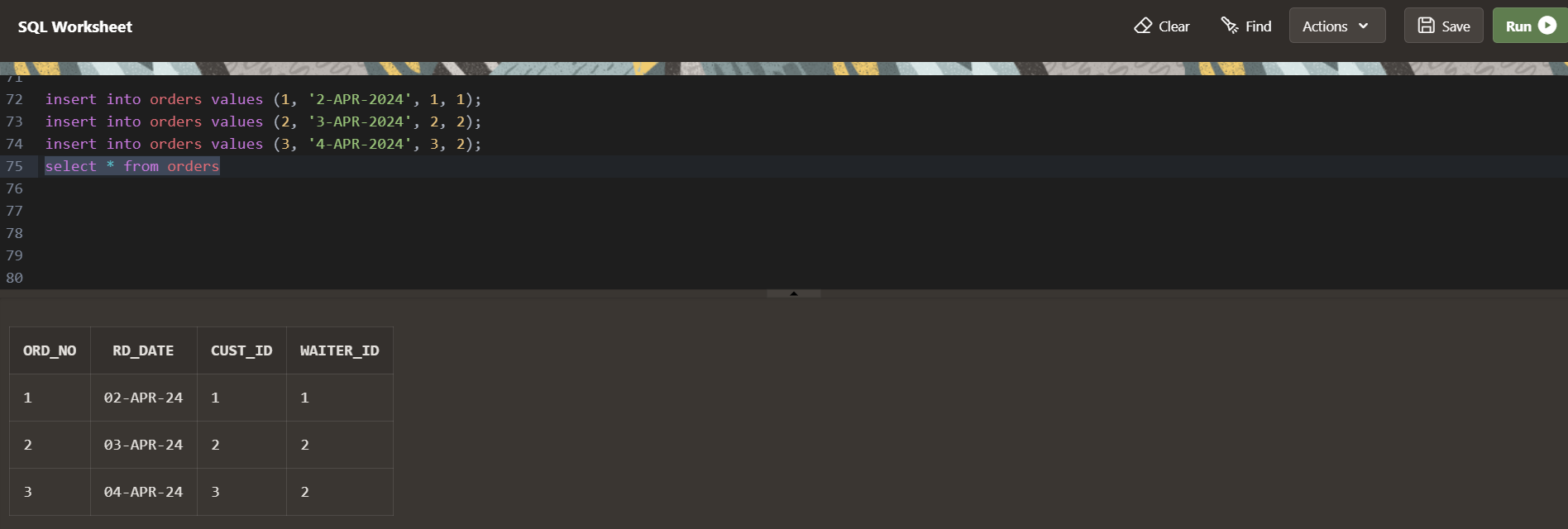


insert into orders values (1, '2-APR-2024', 1, 1);

insert into orders values (2, '3-APR-2024', 2, 2);

insert into orders values (3, '4-APR-2024', 3, 2);

select \* from orders

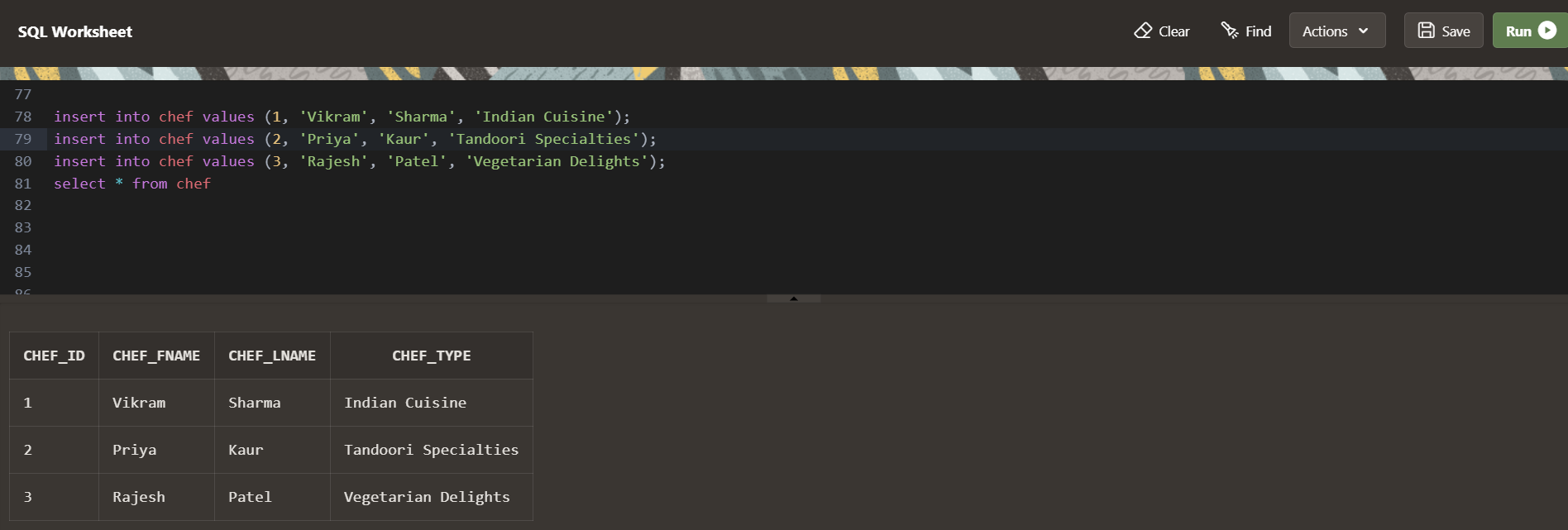


insert into chef values (1, 'Vikram', 'Sharma', 'Indian Cuisine');

insert into chef values (2, 'Priya', 'Kaur', 'Tandoori Specialties');

insert into chef values (3, 'Rajesh', 'Patel', 'Vegetarian Delights');

select \* from chef



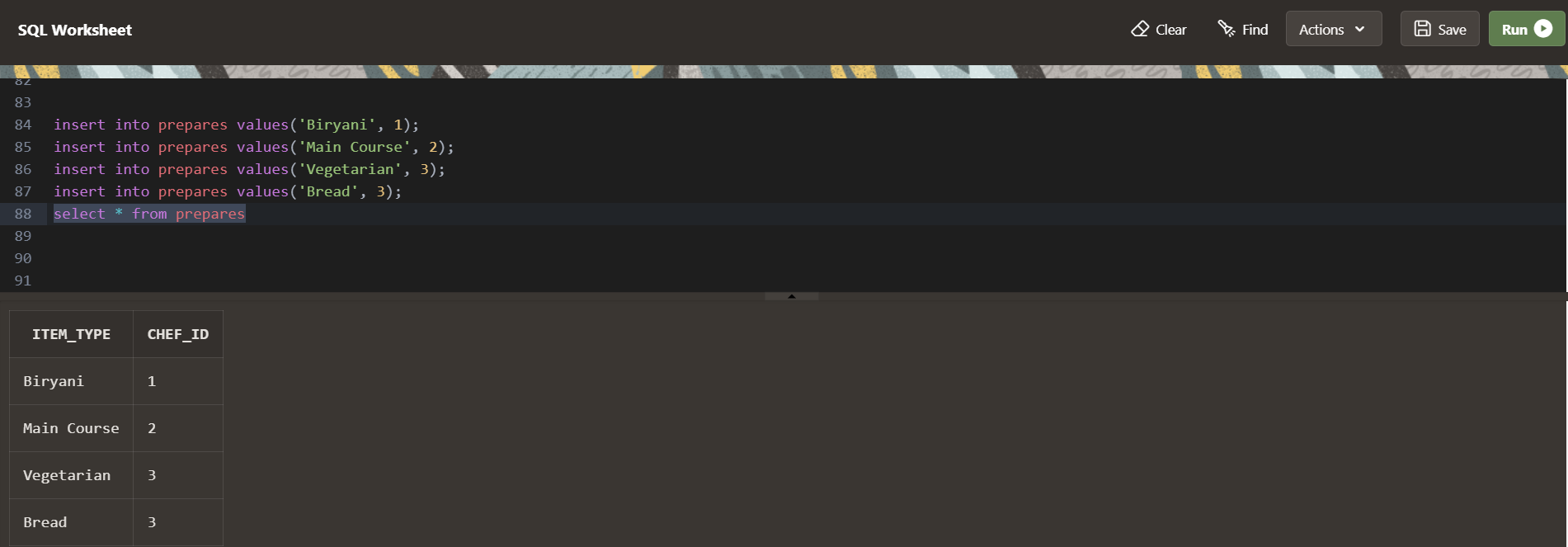
insert into prepares values('Biryani', 1);

insert into prepares values('Main Course', 2);

insert into prepares values('Vegetarian', 3);

insert into prepares values('Bread', 3);

select \* from prepares



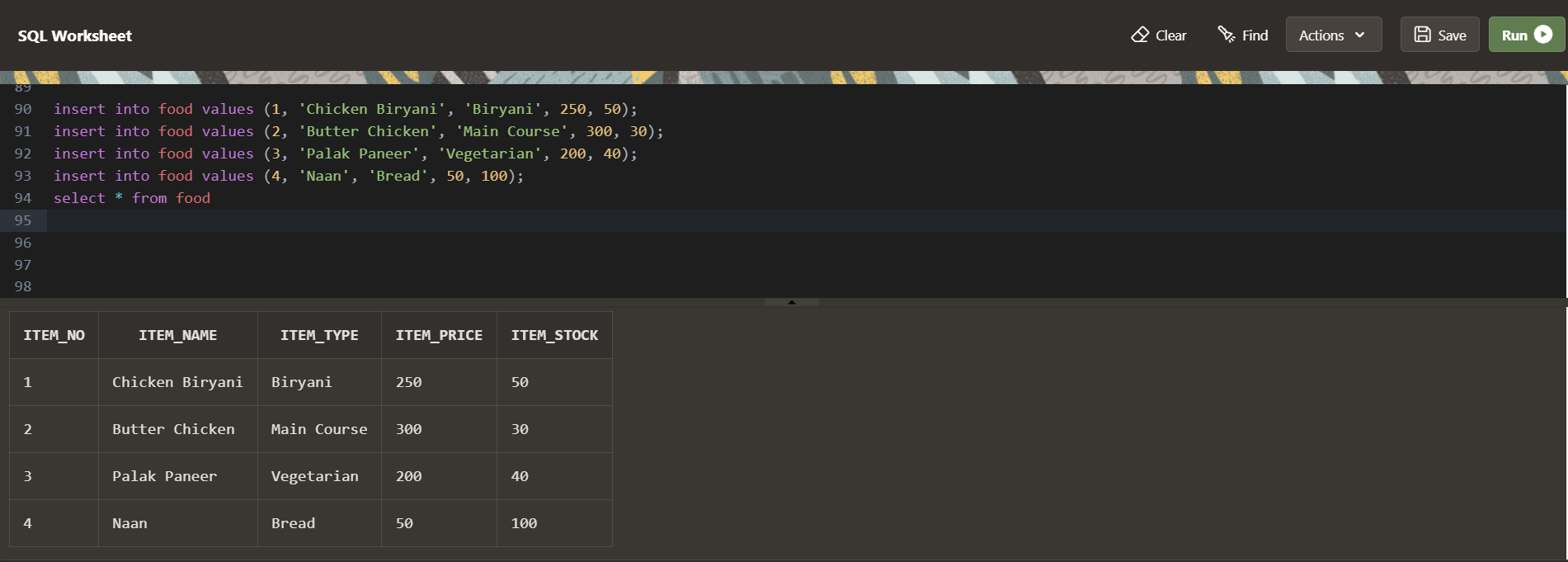
insert into food values (1, 'Chicken Biryani', 'Biryani', 250, 50);

insert into food values (2, 'Butter Chicken', 'Main Course', 300, 30);

insert into food values (3, 'Palak Paneer', 'Vegetarian', 200, 40);

insert into food values (4, 'Naan', 'Bread', 50, 100);

select \* from food



insert into contains values(1,1);

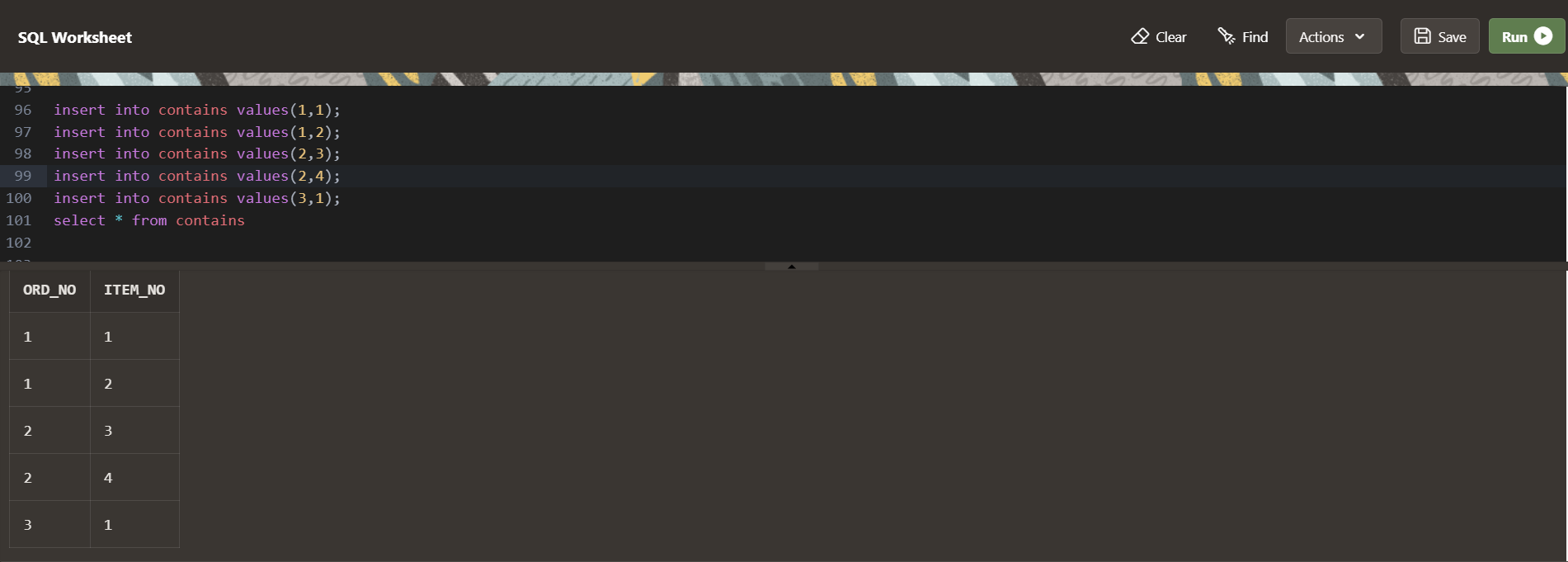
insert into contains values(1,2);

insert into contains values(2,3);

insert into contains values(2,4);

insert into contains values(3,1);

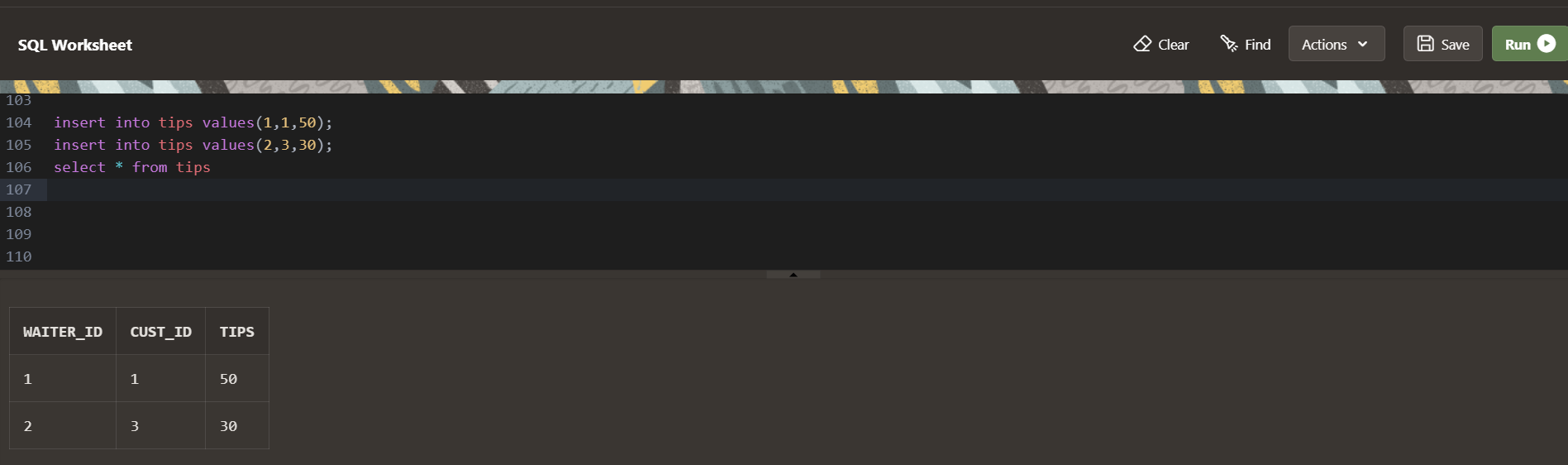
select \* from contains



insert into tips values(1,1,50);

insert into tips values(2,3,30);

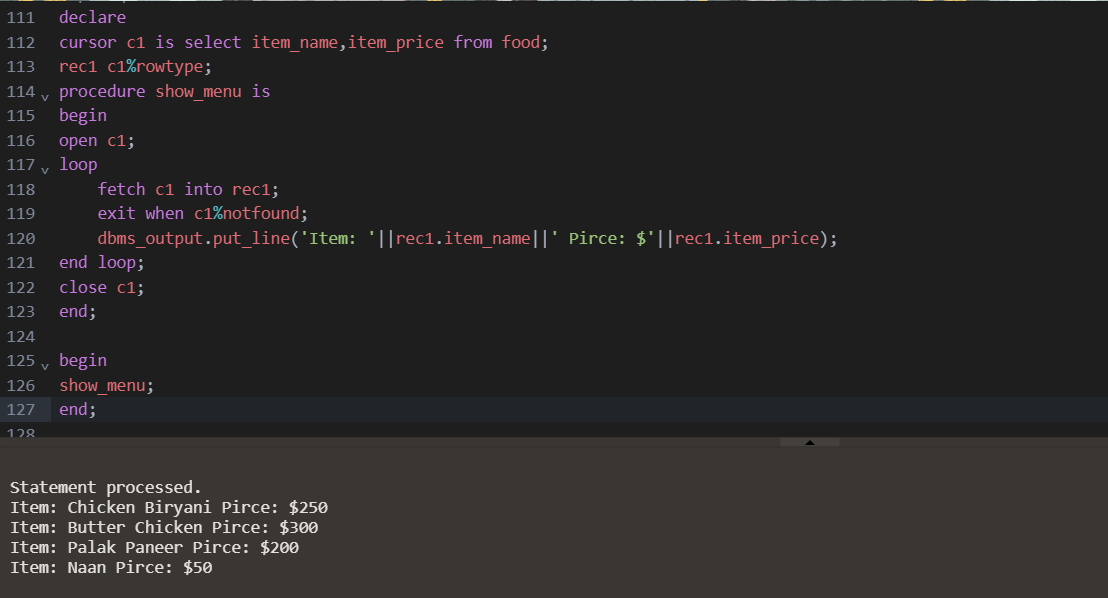
select \* from tips



**PL/SQL**

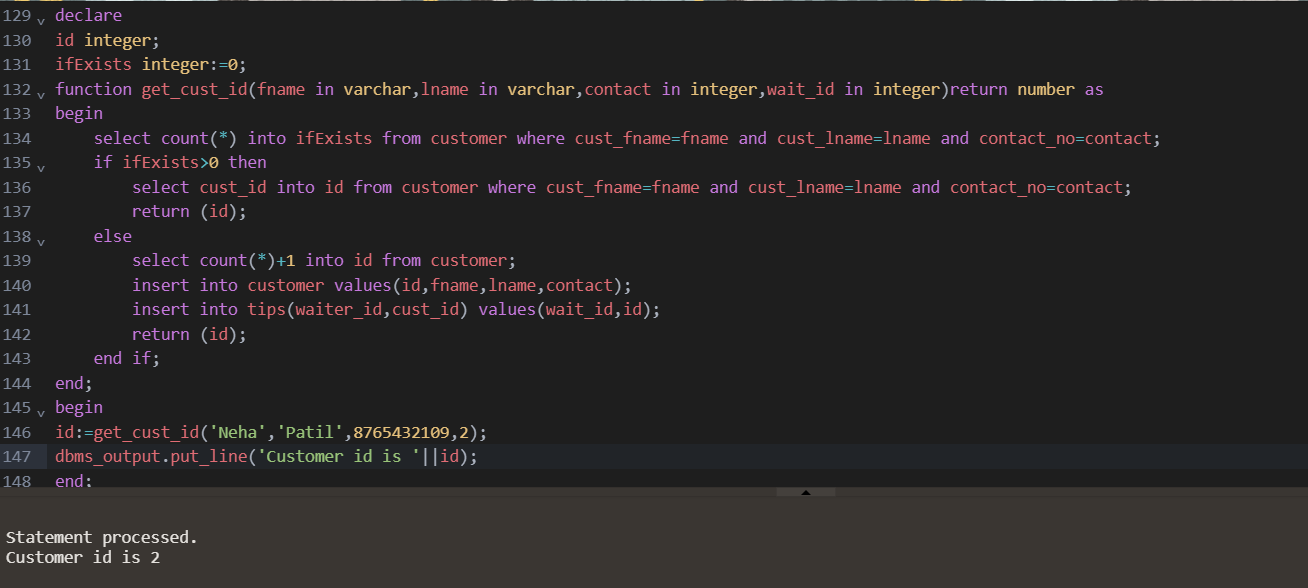
1. Show\_Menu Procedure:

Displays the items and their Prices Using a Cursor.



1. Get\_Cust\_Id Function:

If a customer already exists then returns the existing ID else create a new customer and return the new ID



1. Placing Order
   1. In\_Stock Trigger:

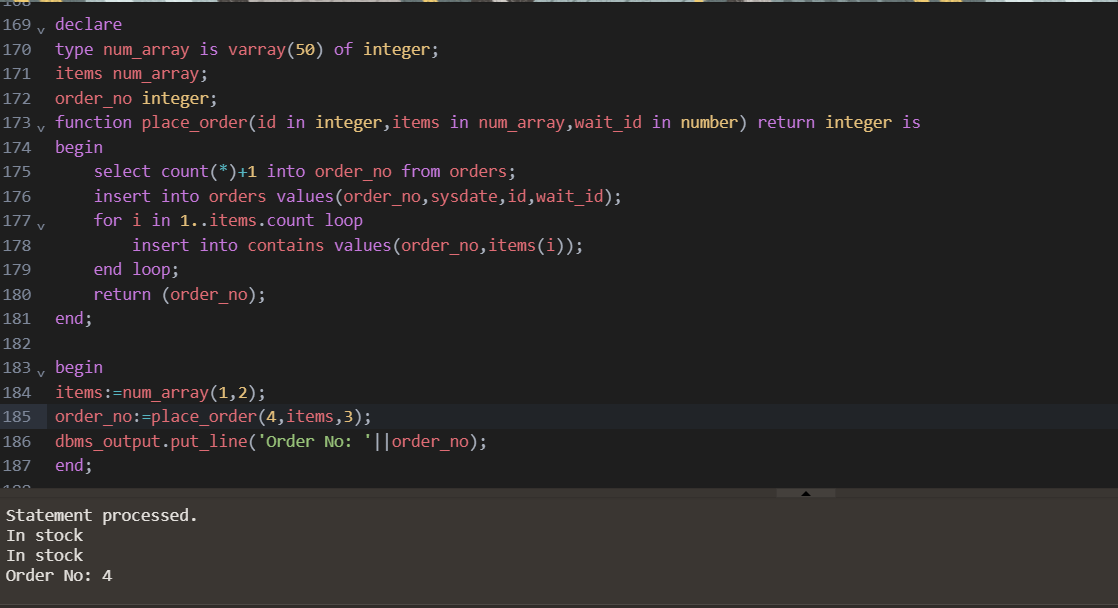
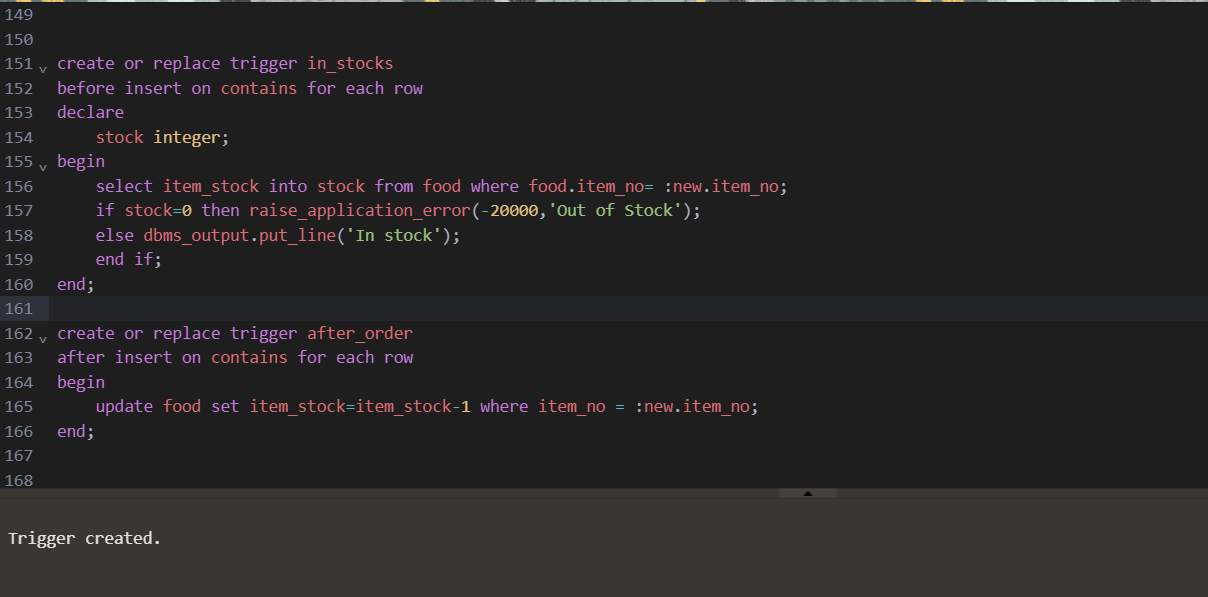
Before inserting the food items in the order, it checks if the items are in stock if they are not in stock it will raise an error.

* 1. After\_Order Trigger:

After inserting the food items in the order, it updates the stock of the items and decreases them accordingly.

* 1. Place\_Order Function:

This function inserts the items in the form of an array of item\_no in the order and returns the order\_no to the customer.

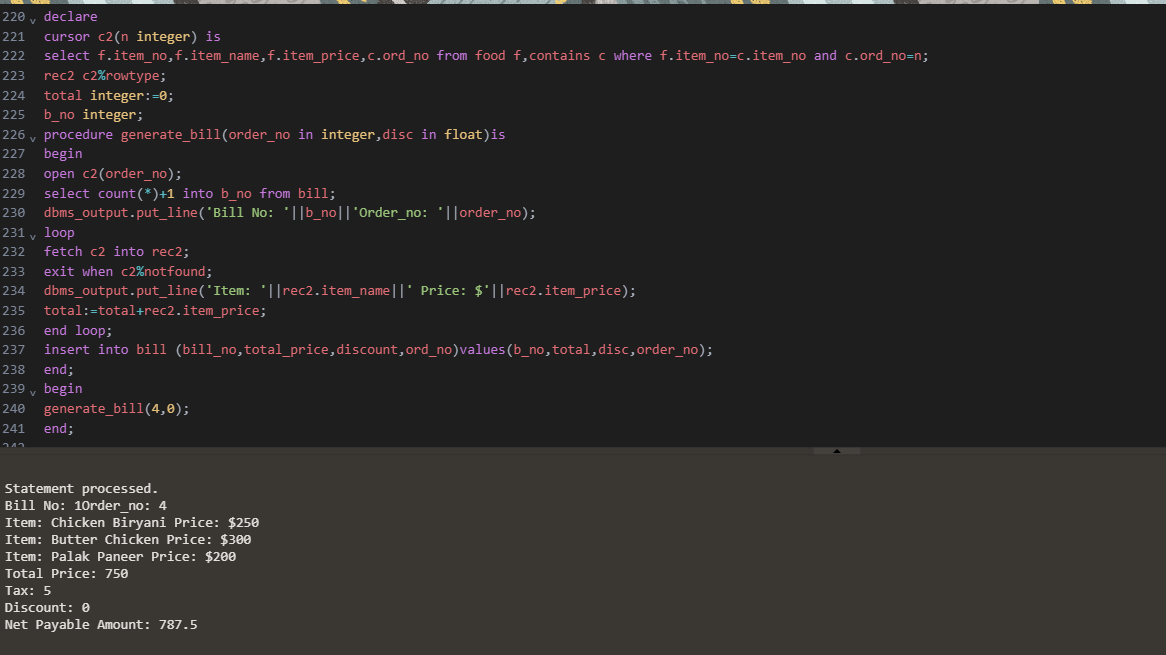
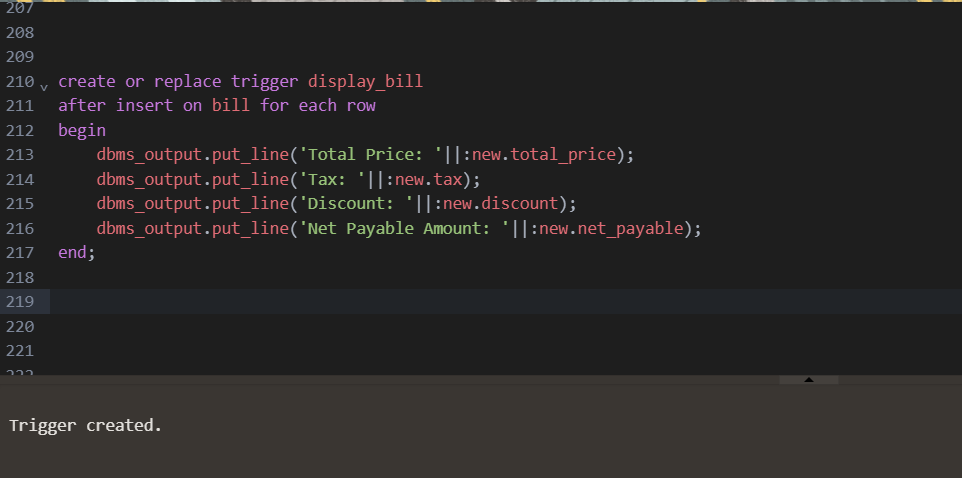


1. Generating Bill
   1. Display\_Bill Trigger:

It displays the bill along with the bill\_no, ord\_no, items, price, total price, discount, tax, and the net\_payable amount to the customer.

* 1. Generate\_Bill Procedure:

It takes in the values order\_no and any discount value and inserts the given values into the bill table.

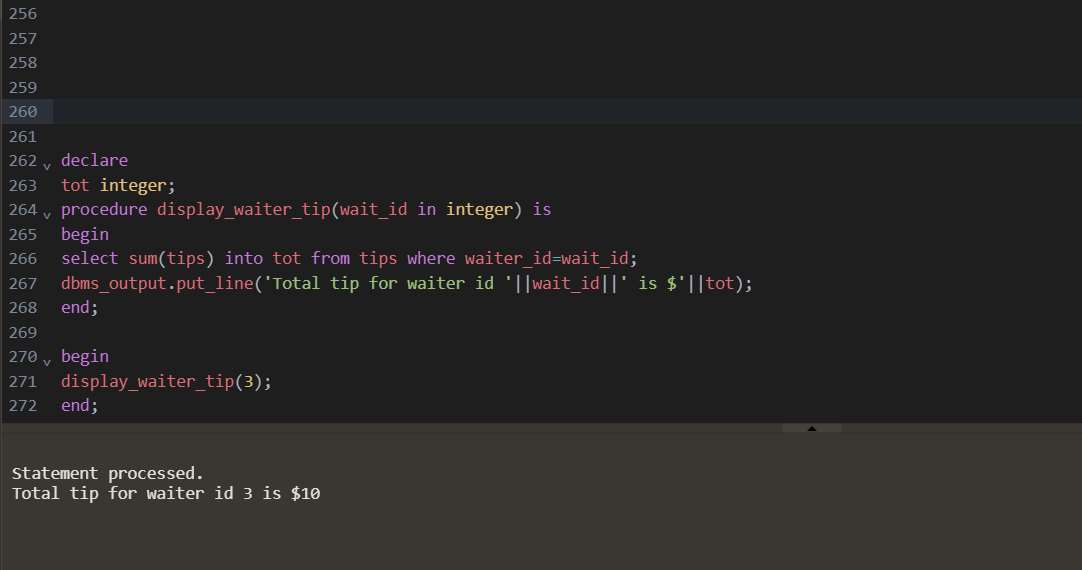


1. Tips
   1. Give\_Tip Procedure:

It takes in the value of the tip given by the customer to the denoted waiter.

* 1. Display\_Waiter\_Tip Procedure:

It displays the total tips collected by a specific waiter.



To conclude,

# Conclusion

The development of a Restaurant Management System using SQL and PL/SQL offers numerous benefits to the hospitality industry. With its robust and scalable features, the system can handle a high volume of transactions and users, integrate with third-party systems, and comply with relevant regulations and standards.

The user-friendly interface makes it easy for restaurant staff to navigate and perform their tasks, reducing errors and providing better services to customers. By considering system requirements, external requirements, and hardware requirements, the Restaurant Management System can be optimized to perform optimally and securely. Overall, investing in a Restaurant Management System using SQL and PL/SQL is a wise decision for restaurants looking to streamline their operations, improve efficiency, and enhance the customer experience.

# References

Websites –

[www.youtube.com/parteekBhatia](http://www.youtube.com/parteekBhatia) <https://app.creately.com/d/start/dashboard>

Books –

1. Fundamentals of database systems (Ramez Elmsari, Shamkant B.Navathe)
2. Database System Concepts (Avi Silberschatz · Henry F.Korth · S. Sudarshan)