Restaurant Application
Sean Onyskiw

DOCUMENT CONTROL

Work carried out by:

Name	Email Address	Other
Sean Onyskiw	**smo5107@psu.edu**	

Revision Sheet

Release No.	Date	Revision Description
1	**10/30/23**	Added Core Requirements
2	**11/05/23**	Added Conceptual Design Diagram and Updated Core Requirements
3	**11/12/23**	Added Logical Design and Updated Conceptual Design Diagram
4	**11/26/23**	Added Normalization and Updated Logical and Conceptual Design
5	**12/03/23**	Added Physical Design, code to Appendix, updated Normalization
6	**12/10/23**	Added SQL Queries and Updated Physical Design

DATABASE SPECIFICATIONS

TABLE OF CONTENTS

MILESTONE 1: DATA REQUIREMENTS

System Name or Title

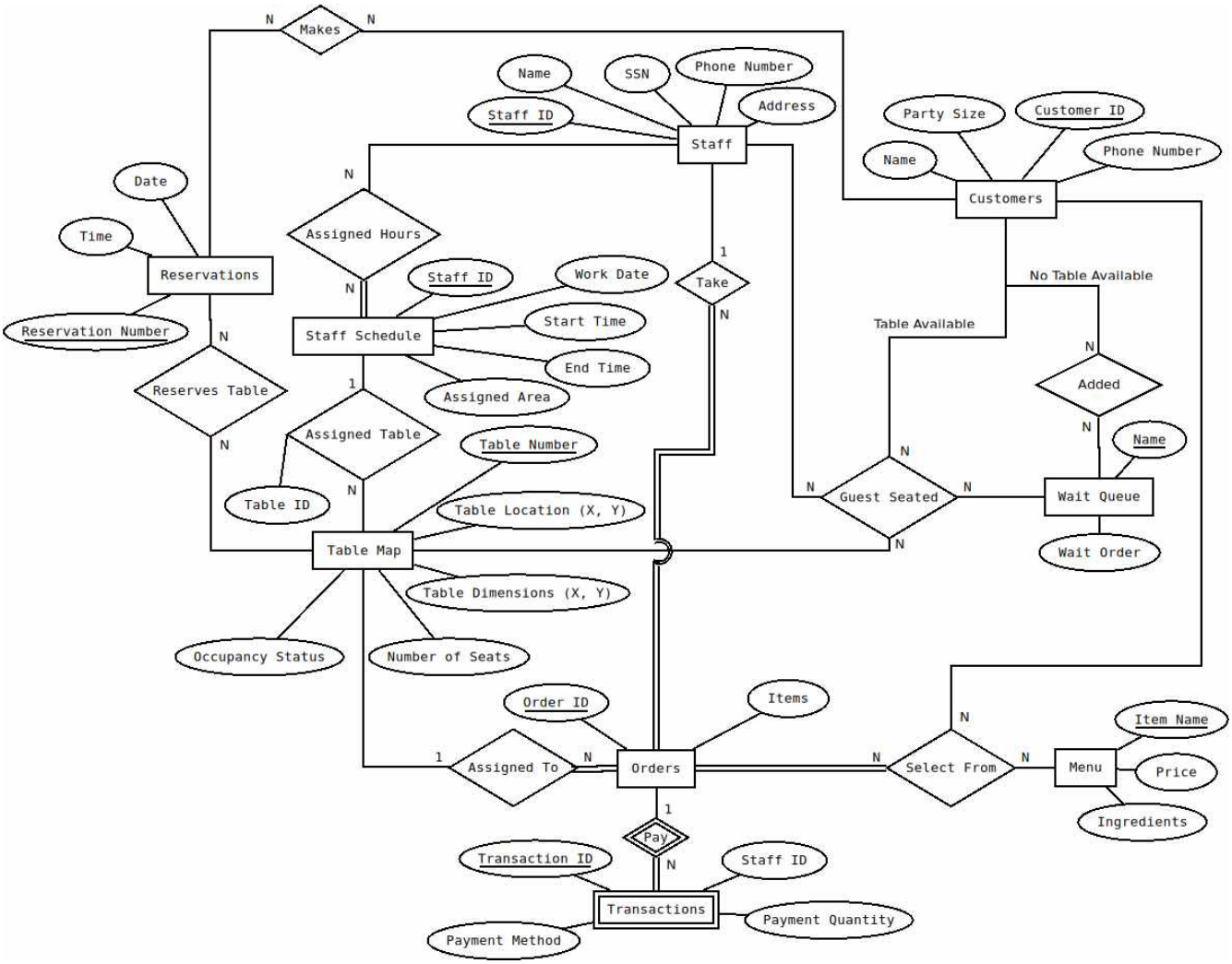
Next-Gen Restaurant Application

Core requirements

No.	Requirement	Referenced page in SRS	Referenced Section in SRS	Referenced Paragraph in Section
1	The system needs to store the information for each table, including number of chairs at each table, and the location of each table.	3 10	1.2 3.5	Item 1 3.5.2
2	The system needs to store customer reservations, including name, time and number of guests.	3 12	1.2 3.5	Item 2 3.5.6.1
3	The system needs to store walk-up customers, including name and party size.	3 10	1.2 3.5	Item 2 3.5.3.1
4	The system needs to store the wait queue.	10	3.5	3.5.3.1
5	The system needs to store table occupancy.	10	3.5	3.5.2.10
6	The system needs to store the staff schedule.	5	2.2	Bullet 6
7	The system needs to be able to store each order, including added items such as desserts or bar tab.	3 9	1.2 3.5	Item 4 3.5.1.1
8	The system needs to store the menu including ingredients.	9 19	3.5 8.1	3.5.1.1 8.1.1
9	The system needs to store the status of each order.	9	3.5	3.5.1.3
10	The system needs to store multiple payments for each order and which payment was used for each item in an order.	9	3.5	3.5.1.4
11	The system needs to store gratuity information for each payment made with a credit card and the server assigned to the table.	9	3.5	3.5.1.5
12	The system needs to store transactions.	9	3.5	3.5.1.9
13	The system can store customer phone numbers.	11	3.5	3.5.3.3
14	The system shall store encrypted payment information	13	5.3	5.3.1

MILESTONE 2: CONCEPTUAL DESIGN

Diagram

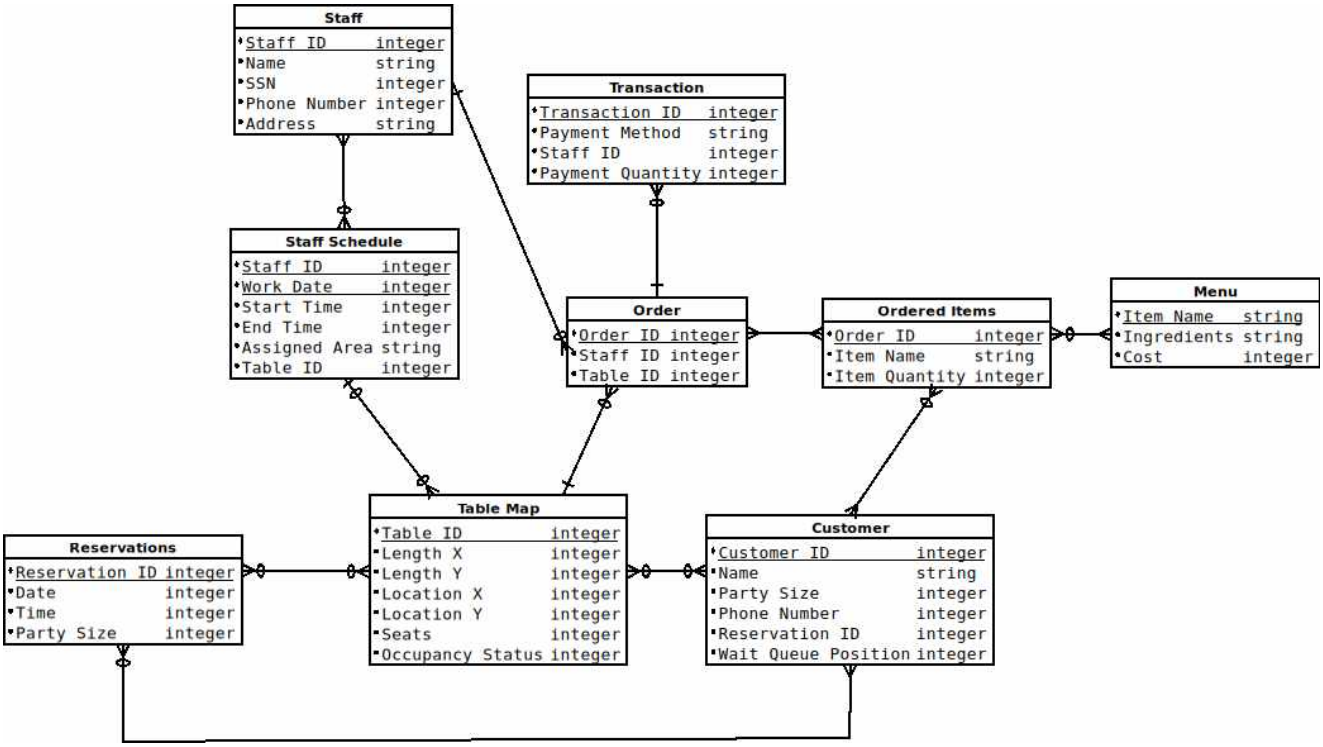


Assumptions and Constraints

- A staff ID is created in order to improve security so that SSN is not used throughout the system to represent employees.
- Staff are assigned to an area on the schedule. Servers are specifically assigned to the tables they will be serving.
- Only one server is assigned to an order. All gratuities are assigned to the server for a table. If multiple servers assist an order, only the server that is assigned the order will receive the gratuities.
- The system places walk-ups in a wait queue if there are not enough seats available. Guests with reservations are not placed on the wait queue. The system will check for any reservations first when determining table occupancy. If there is an open table not taken by a reservation, then the first guest on the wait queue will be able to be seated, assuming the table has enough seats for the party.
- Orders are assigned to a table. When the guests leave, the order can be closed after payment is received and a new order for the table can be started.
- Bar tabs are treated the same as table orders. Orders can be modified to add items as needed, such as when a tab is opened at the bar or additional items such as dessert are ordered.
- Transactions can use multiple payment methods, and items in an order can be assigned to different payments, allowing guests to split the bill.
- Transactions cannot occur without an order.
- Table occupancy uses table IDs from the table map.

MILESTONE 3: LOGICAL DESIGN

Entity Relationship Diagram



Entity name: Staff

Attributes:

Staff ID, Name, SSN, Phone Number, Address

Functional dependencies:

SSN → Staff ID, Name, Phone Number, Address
Staff ID→ SSN, Name, Phone Number, Address

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
		SSN, Staff ID	Name, Phone Number, Address

Attribute closures:

SSN+ = SSN, Staff ID, Name, Phone Number, Address
Staff ID+ = Staff ID, SSN, Name, Phone Number, Address
SSN, Staff ID are candidate keys

Unique keys: the keys for this table are

SSN
Staff ID

Staff ID would be the primary key for consistency as it is used in other tables.

Entity name: Staff Schedule

Attributes:

Staff ID, Work Date, Start Time, End Time, Assigned Area, Table ID

Functional dependencies:

Staff ID, Work Date → Start Time, End Time, Assigned Area, Table ID

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Staff ID, Work Date		Start Time, End Time, Assigned Area, Table ID

Attribute closures:

Staff ID, Work Date+ =

Staff ID, Work Date, Start Time, End Time, Assigned Area, Table ID

(Staff ID, Work Date) is the candidate key. Both are required as just Staff ID will give multiple work days.

Unique keys: the key for this table is

(Staff ID, Work Date)

Entity name: Order

Attributes:

Order ID, Staff ID, Table ID

Functional dependencies:

Order ID → Item Name, Staff ID, Table ID

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Order ID		Staff ID, Table ID

Attribute closures:

Order ID+ = Order ID, Staff ID, Table ID

Unique keys: the key for this table is

Order ID

Entity name: Menu

Attributes:

Item Name, Ingredients, Price

Functional dependencies:

Order ID → Item Name, Staff ID, Table ID

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Item Name		Ingredients, Price

Attribute closures:

Item Name+ = Item Name, Ingredients, Price

Unique keys: the key for this table is

Item Name

Entity name: Ordered Items

Attributes:

Order ID, Item Name

Functional dependencies:

Order ID → Item Name

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Order ID		Item Name, Item Quantity

Attribute closures:

Order ID+ = Order ID, Item Name, Item Quantity

Unique keys: the key for this table is

Order ID

Entity name: Transaction

Attributes:

Transaction ID, Payment Method, Staff ID, Payment Quantity

Functional dependencies:

Transaction ID→ Payment Method, Staff ID, Payment Quantity

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Transaction ID		Payment Method, Staff ID, Payment Quantity

Attribute closures:

Transaction ID+ = Transaction ID, Payment Method, Staff ID, Payment Quantity

Unique keys: the key for this table is

Transaction ID

Entity name: Table Map

Attributes:

Table ID, Length X, Length Y, Location X, Location Y, Seats, Occupancy Status

Functional dependencies:

Table ID → Length X, Length Y, Location X, Location Y, Seats, Occupancy Status
Location X, Location Y → Table ID, Length X, Length Y, Seats, Occupancy Status

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
		Table ID, Location X, Location Y	Length X, Length Y, Seats, Occupancy Status

Attribute closures:

Table ID+ = Table Id, Length X, Length Y, Location X, Location Y, Seats, Occupancy Status

Location X, Location Y+ = Table Id, Length X, Length Y, Location X, Location Y, Seats, Occupancy Status

Table ID and (Location X, Location Y) are candidate keys. Table ID would be the primary key as it only requires one attribute. Location X and Y could be used to uniquely identify a table, but requires both attributes.

Unique keys: the keys for this table are

Table ID
(Location X, Location Y)

Table ID will be used as the primary key.

Entity name: Customer

Attributes:

Customer ID, Name, Party Size, Phone Number, Reservation ID, Wait Queue Position

Functional dependencies:

Customer ID → Name, Party Size, Phone Number, Reservation ID, Wait Queue Order

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Customer ID		Name, Party Size, Phone Number, Reservation ID, Wait Queue Position

Attribute closures:

Customer ID+ = Customer ID, Name, Party Size, Phone Number, Reservation ID, Wait Queue Order

Customer ID is a candidate key. Reservation ID and Wait Queue could each function as a key, however these attributes will not always have a value, such as when a customer is a walk-up customer without a reservation, or when there are available tables and a wait queue is not needed.

Unique keys: the key for this table is

Customer ID

Entity name: Reservations

Attributes:

Reservation ID, Date, Time, Party Size

Functional dependencies:

Reservation ID → Date, Time, Party Size

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Reservation ID		Date, Time, Party Size

Attribute closures:

Reservation ID+ = Reservation ID, Date, Time, Party Size

Unique keys: the key for this table is

Reservation ID

Assumptions and Constraints

- Walk up customers could be seated at a table if there is no wait without needing to be entered into the system.
- Customers with a reservation will not have a wait queue value. Walk-up customers in the wait queue will not have a reservation number.
- A separate table will be created to store the items for each order. This will prevent redundancy when also storing the staff and table ID for an order. The total cost will be calculated using prices in the menu and the list of ordered items.
- Staff ID are connected to an order and then to a transaction to handle gratuities.
- The staff schedule assumes that each worker has one start and end time each day. If there are multiple shifts in a day, staff ID and work day would not be enough to act as a key and would require start time as well.
- Staff do not need to be assigned a table, as they could be assigned to other areas such as the kitchen or bar.

MILESTONE 4: NORMALIZATION

Naming Conventions

Tables are generally named to provide a brief description of the purpose of the table. Some tables have a written description before the structure of the table to explain the design choices made in constructing the table. Often this will also include the reasoning for why tables were split in order to meet 3NF requirements.

Tables are organized in the document below so that tables with similar functions are together, such as staff information and staff schedules in the staff subsection. Any tables that were split have the sub tables located directly after the main table. Functional dependencies (FD) may be listed at the start of a section, especially in cases where splitting tables may cause a functional dependency to be lost. In addition, if multiple candidate keys exist for a table, bold denotes primary key.

Tables

Staff

Staff table created in logical design has FDs below. Staff ID was selected as primary key as it is used throughout the remaining tables instead of SSN as an identifier for staff. Since the fundamental dependency SSN → Staff ID, Name, Phone Number, Address is a trivial dependency as it contains the key for the table, the staff table is already in 3NF.

FDs:
Staff ID→ SSN, Name, Phone Number, Address
SSN → Staff ID, Name, Phone Number, Address

	Name of the table	***Staff***			
	Description	Contains the staff that work for the restaurant and personal information for each staff member.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	
	Name	Name of employee	string	John Doe	Includes both first and last names as part of string
	SSN	Social Security Number		10000000 to 999999999	Must be a 9-digit number
	Phone Number	Phone number of employee	integer	1000000000 to 9999999999	Number stored without hyphens or other formatting, assumes US numbers only (10 digits)
	Address	Address of employee	string	123 Main St. Anytown USA, 00000	Entire address is stored as one string, parts of address are not separated into different columns/attributes
	Functional Dependencies and Keys				
	Functional dependencies	Staff ID → SSN, Name, Phone Number, Address SSN → Staff ID, Name, Phone Number, Address			
	Candidate keys	**Staff ID**, SSN			
	Normalization				
	1NF	**Yes**	All data contains single values. (Note: This assumes that it is acceptable in the design to have name and address represented by a single string. If the name and address need to be separated, then multiple attributes would need to be created to make table 1NF compliant.)		
	2NF	**Yes**	Key is single attribute (Staff ID)		
	3NF	**Yes**	SSN → Staff ID, Name, Phone Number, Address is a trivial FD since it returns the key of the table – Staff ID. No other non-prime attributes can be determined from any other non-prime attributes. There are no other nontrivial FDs. The table is in 3NF since a trivial FD that returns the table key does not prevent a table from being in 3NF.		
	BCNF	**Yes**	No transitive functional dependencies exist.		

Staff Schedule originally had attributes Staff ID, Work Date, Start Time, End Time, Assigned Area, Table ID. Two issues existed with this attribute list:

1. Assigned area was originally a string. This should be an integer to save space and make it easier to use the database. The name should exist as a separate column. Since the name is then dependent on the area id, this would create a functional dependency (Area ID → Area Name) not using the key of the table {Staff ID, Work Date}, resulting in the table not in 3NF, and would require the area name table to be split.
2. The Table ID is the assigned table. This attribute could also have a value of NULL, if the employee were assigned to an area other than the dining room, such as the kitchen. To avoid having nulls in the database, the Assigned Table attribute is split into a separate table. Since a staff member could be assigned to multiple tables, the key required for this table needs Staff ID, Work Date, and Table ID.

The FDs for the staff schedule are:
Staff ID, Work Date → Start Time, End Time, Area.
Area → Area Name
Note: No FD exists that returns Table ID, requiring a key of Staff ID, Work Date, and Table ID.

	Name of the table	***Staff Schedule***			
	Description	Contains the staff schedule and assigned area.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	
	Work Date	Work Date	integer	Between 1 to 999999999	Represented as days since a first date
	Start Time	Start time on specified work date	integer	Between 1 to 9999999999	Seconds since reference time
	End Time	End time on specified work date	integer	Between 1 to 9999999999	Seconds since reference time
	Area ID	Assigned Area	integer	1 to 99	
	Functional Dependencies and Keys				
	Functional dependencies	Staff ID, Work Date → Start Time, End Time, Area ID.			
	Candidate keys	Staff ID, Work Date			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (either Staff ID or Work Date)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

	Name of the table	***Area Names***			
	Description	Contains area IDs and names.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Area ID	ID number of area	integer	Between 1 and 99	
	Area Name	Name of area	string	Kitchen	
	Functional Dependencies and Keys				
	Functional dependencies	Area ID→ Area Name			
	Candidate keys	Area			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Area)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

Key for the Assigned Tables table is all attributes in the table since no FD exists. This table is assumed to only be used when a staff member is assigned to the dining room (assigned area matches dining room id) and is not needed for other staff members. It is also assumed that only one staff member is assigned to a table at a time. This is to make it possible for the system to assign gratuities so that the staff member assigned to the table can be looked up at the time of order. The system would then need to check when scheduling staff that no times overlap with multiple staff members assigned to the same table at the same time. If the system is not set up to prevent overlapping table assignments, then manual intervention from the user would be required to properly assign gratuities to the correct server.

	Name of the table	***Assigned Tables***			
	Description	Contains area IDs and names			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	
	Work Date	Work Date	integer	Between 1 to 999999999	Represented as days since a first date
	Table ID	Assigned table ID number	integer	1 to 999	
	Functional Dependencies and Keys				
	Functional dependencies	None			
	Candidate keys	{Staff ID, Work Date, Table ID}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (all attributes are part of the key, no FDs for this table)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute. (No non-prime attributes in the table)		
	BCNF	**Yes**	Yes, no FDs for this table.		

Table Map

	Name of the table	***Table Map***			
	Description	Contains the table map information including dimensions and locations. Occupancy status is also recorded in this table.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Table ID	ID number of an employee	integer	Between 1 and 999	
	Length X	Horizontal length of table	integer	1 to 999999	Added to table origin point to create table shape
	Length Y	Vertical length of table	integer	1 to 999999	Added to table origin point to create table shape
	Location X	Horizontal location of table	integer	1 to 999999	X-axis origin point of table
	Location Y	Vertical location of state	integer	1 to 999999	Y-axis origin point of table
	Seats	Number of seats at table	integer	1 to 99	
	Occupancy Status	Status of table	Boolean	Occupied (TRUE) or unoccupied (FALSE)	
	Functional Dependencies and Keys				
	Functional dependencies	Table ID→ Length X, Length Y, Location X, Location Y, Seats, Occupancy Status Location X, Location Y → Table ID, Length X, Length Y, Seats, Occupancy Status			
	Candidate keys	**Table ID**, {Location X, Location Y}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Table ID)		
	3NF	**Yes**	Location X, Location Y → Table ID, Length X, Length Y, Seats, Occupancy Status is a trivial FD since it returns the key of the table - Table ID. No other non-prime attributes can be determined from any other non-prime attributes. There are no other nontrivial FDs. The table is in 3NF since a trivial FD that returns the table key does not prevent a table from being in 3NF.		
	BCNF	**Yes**	No transitive functional dependencies exist.		

Orders

The order and ordered items tables were already normalized as part of the logical design step, with orders and the items ordered being split into separate tables.

	Name of the table	***Order***			
	Description	Contains the order ID and table and server.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Order ID	ID number of order	integer	1 to 999999999	
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	Needed to determine server that gratuities are assigned to
	Table ID	ID number of table order is assigned to	integer	1 to 999	
	Functional Dependencies and Keys				
	Functional dependencies	Order ID→ Staff ID, Table ID			
	Candidate keys	Order ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Order ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

	Name of the table	***Ordered Items***			
	Description	Contains the order ID and table and server.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Order ID	ID number of order	integer	1 to 999999999	
	Item Name	Name of item from menu	integer	Filet Mignon	Items restricted to available items on the menu.
	Quantity	Quantity of named item	integer	1 to 999	
	Functional Dependencies and Keys				
	Functional dependencies	Order ID, Ordered Item → Quantity			
	Candidate keys	{Order ID, Ordered Item}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (Quantity requires both Order ID and Ordered Item)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute (Only one non-prime attribute in table).		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

The menu contains the item name, ingredients, and cost of the item. Since a menu item can have multiple ingredients, both Item Name and Ingredient must be part of the key. A functional dependency also exists, Item Name → Cost. Since only the name is required for the cost, this would violate 2NF rules since only part of the key is needed to determine the non-prime attribute. Two tables are needed to satisfy 2NF conditions, one table with items and ingredients, and another with items and costs.

	Name of the table	***Menu***			
	Description	Contains the menu items and cost			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Item Name	Name of item from menu	string	Filet Mignon	
	Cost	Cost of the item	integer	1 to 999999999	Cost stored in cents, for example an item that is $20.00 is stored as 2000.
	Functional Dependencies and Keys				
	Functional dependencies	Item Name → Cost			
	Candidate keys	Item Name			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Item Name)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

	Name of the table	***Ingredients***			
	Description	Contains the ingredients for each item on the menu			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Item Name	Name of item from menu	string	Filet Mignon	
	Ingredient	Name of ingredient	string	Pepper	
	Functional Dependencies and Keys				
	Functional dependencies	None			
	Candidate keys	{Item Name, Ingredient}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (all attributes are part of the key, no FDs for this table)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute. (No non-prime attributes in the table)		
	BCNF	**Yes**	Yes, no FDs for this table.		

	Name of the table	***Transactions***			
	Description	Contains the transactions and payment methods			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Transaction ID	ID number of transaction	integer	1 to 999999999	
	Payment Method	Method of Payment	string	Credit Card	
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	Used for assigning gratuities
	Payment Quantity	Payment amount	integer	1 to 999999999	Cost stored in cents, for example a transaction that is $20.00 is stored as 2000.
	Functional Dependencies and Keys				
	Functional dependencies	Transaction ID→ Payment Method, Staff ID, Payment Quantity			
	Candidate keys	Transaction ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Transaction ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

Customers

The customer table contains two optional attributes that only apply to certain conditions: Reservation ID, and Wait Queue Position. As a result, it is possible to have NULLs in either of these attributes if the customer does not have a reservation or is waiting for a table. To avoid nulls, the customer table will be split into a Customers table, Wait Queue table, and Reservations table. The reservation table was already specified in the logical design step. Since the customer table includes party size, it was removed from the reservation table where it was originally specified in the logical design. Creating a new reservation will automatically add the customer to the Customers table.

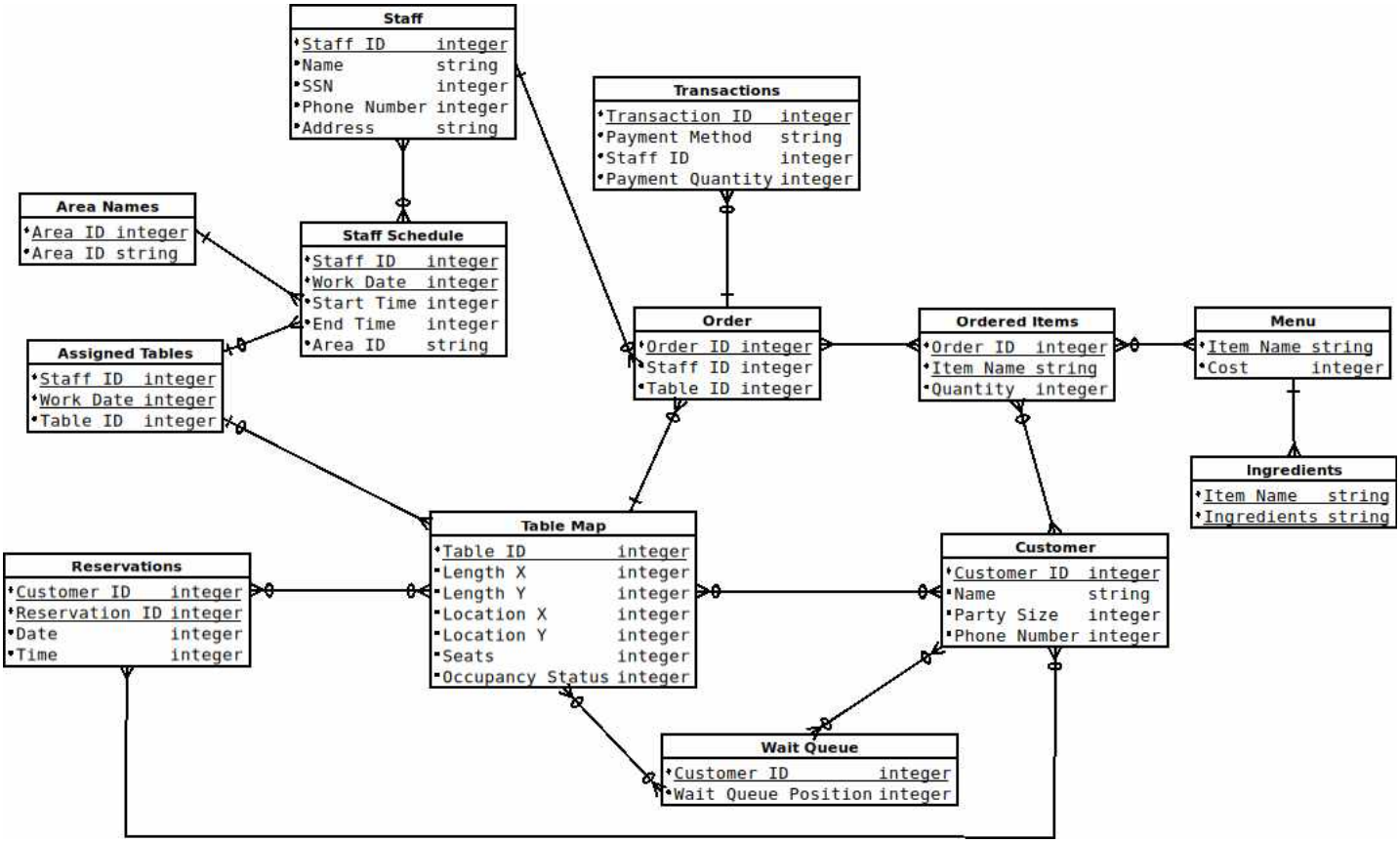
	Name of the table	***Customers***			
	Description	Contains the customer information			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Customer ID	ID number of customer	integer	1 to 999999999	Index value, increases by one for each new customer
	Name	Name of the customer	string	John Doe	
	Party Size	Number of total guests with customer	integer	Between 1 and 999	
	Phone Number	Phone number of customer	integer	1000000000 to 9999999999	Number stored without hyphens or other formatting, assumes US numbers only (10 digits)
	Functional Dependencies and Keys				
	Functional dependencies	Customer ID → Name, Party Size, Phone Number			
	Candidate keys	Customer ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Customer ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

	Name of the table	***Reservations***			
	Description	Contains the customer information			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Customer ID	ID number of customer	integer	1 to 999999999	Index value, increases by one for each new customer
	Reservation ID	ID number of reservation	integer	1 to 999999999	Index value, increases by one for each new reservation
	Date	Reservation date	integer	Between 1 to 999999999	Represented as days since a first date
	Time	Reservation time	integer	Between 1 to 9999999999	Seconds since reference time
	Functional Dependencies and Keys				
	Functional dependencies	Customer ID → Reservation ID, Date, Time Reservation ID → Customer ID, Date, Time			
	Candidate keys	**Customer ID**, Reservation ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Customer ID)		
	3NF	**Yes**	Reservation ID → Customer ID, Date, Time is a trivial FD since it returns the key of the table - Customer ID. No other non-prime attributes can be determined from any other non-prime attributes. There are no other nontrivial FDs. The table is in 3NF since a trivial FD that returns the table key does not prevent a table from being in 3NF.		
	BCNF	**No**	Trivial FD exists in the table.		

	Name of the table	***Wait Queue***			
	Description	Contains the wait queue position			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Customer ID	ID number of customer	integer	1 to 999999999	Index value, increases by one for each new customer
	Wait Queue Position	Current wait position	integer	1 to 999	Updated as customer is moved up the queue
	Functional Dependencies and Keys				
	Functional dependencies	Customer ID → Wait Queue Position			
	Candidate keys	Customer ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Customer ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		

Revised Entity Relationship Diagram

Revised entity relationship diagram based on the newly created tables created during the normailization process.



MILESTONE 5: PHYSICAL DESIGN

Naming Conventions:

Oracle supports both date and time in the DATE data type. This allows for any tables that stored a date and a time to have both attributes stored as single attribute. Day and time attributes in the staff schedule and reservations tables were combined into one attribute in the physical design step. Boolean types were changed to Number(1) type with 0 as False and 1 as True since Boolean is not supported in Oracle.

Tables in the physical design section are arranged in the order needed for table creation to satisfy foreign key constraints.

Tables:

	Name of the table	***Staff***			
	Description	Contains the staff that work for the restaurant and personal information for each staff member.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	
	Name	Name of employee	string	John Doe	Includes both first and last names as part of string
	SSN	Social Security Number		10000000 to 999999999	Must be a 9-digit number
	Phone Number	Phone number of employee	integer	1000000000 to 9999999999	Number stored without hyphens or other formatting, assumes US numbers only (10 digits)
	Address	Address of employee	string	123 Main St. Anytown USA, 00000	Entire address is stored as one string, parts of address are not separated into different columns/attributes
	Functional Dependencies and Keys				
	Functional dependencies	Staff ID → SSN, Name, Phone Number, Address SSN → Staff ID, Name, Phone Number, Address			
	Candidate keys	**Staff ID**, SSN			
	Normalization				
	1NF	**Yes**	All data contains single values. (Note: This assumes that it is acceptable in the design to have name and address represented by a single string. If the name and address need to be separated, then multiple attributes would need to be created to make table 1NF compliant.)		

	2NF	**Yes**	Key is single attribute (Staff ID)
	3NF	**Yes**	SSN → Staff ID, Name, Phone Number, Address is a trivial FD since it returns the key of the table – Staff ID. No other non-prime attributes can be determined from any other non-prime attributes. There are no other nontrivial FDs. The table is in 3NF since a trivial FD that returns the table key does not prevent a table from being in 3NF.
	BCNF	**Yes**	No transitive functional dependencies exist.
	Physical Design		
	Physical Design		staff
	Primary Key		staffId
	Foreign Keys		-
	SQL Code		CREATE TABLE staff (staffId INTEGER, name VARCHAR(50), ssn NUMBER(9) UNIQUE, phoneNumber NUMBER(10), address VARCHAR(50), PRIMARY KEY (staffId));
	Count of records in the table		20

	Name of the table	***Area Names***			
	Description	Contains area IDs and names.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Area ID	ID number of area	integer	Between 1 and 99	
	Area Name	Name of area	string	Kitchen	
	Functional Dependencies and Keys				
	Functional dependencies	Area ID→ Area Name			
	Candidate keys	Area ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Area)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	areaNames			
	Primary Key	areaId			
	Foreign Keys	-			
	SQL Code	CREATE TABLE areaNames (areaId INTEGER, areaName VARCHAR(20), PRIMARY KEY (areaId));			
	Count of records in the table	7			

	Name of the table	***Staff Schedule***			
	Description	Contains the staff schedule and assigned area.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	
	Start Time	Start time on specified work date	integer	Between 1 to 9999999999	Seconds since reference time
	End Time	End time on specified work date	integer	Between 1 to 9999999999	Seconds since reference time
	Area ID	Assigned Area	integer	1 to 99	
	Functional Dependencies and Keys				
	Functional dependencies	Staff ID, Start Time → End Time, Area ID.			
	Candidate keys	{Staff ID, Start Time}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (either Staff ID or Start Time)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	staffSchedule			
	Primary Key	{staffId, startTime}			
	Foreign Keys	staffId from staff areaId from areaNames			
	SQL Code	CREATE TABLE staffSchedule (staffId INTEGER, startTime DATE, endTime DATE, areaId INTEGER, PRIMARY KEY (staffId, startTime), FOREIGN KEY (staffId) REFERENCES staff(staffId), FOREIGN KEY (areaId) REFERENCES areaNames(areaId));			
	Count of records in the table	20			

Work date was removed since both day and time are stored using the DATE type in Oracle. Key was changed to use start time instead.

	Name of the table	***Table Map***			
	Description	Contains the table map information including dimensions and locations. Occupancy status is also recorded in this table.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Table ID	ID number of an employee	integer	Between 1 and 999	
	Length X	Horizontal length of table	integer	1 to 999999	Added to table origin point to create table shape
	Length Y	Vertical length of table	integer	1 to 999999	Added to table origin point to create table shape
	Location X	Horizontal location of table	integer	1 to 999999	X-axis origin point of table
	Location Y	Vertical location of state	integer	1 to 999999	Y-axis origin point of table
	Seats	Number of seats at table	integer	1 to 99	
	Occupancy Status	Status of table	integer	Occupied (1) or unoccupied (0)	
	Functional Dependencies and Keys				
	Functional dependencies	Table ID→ Length X, Length Y, Location X, Location Y, Seats, Occupancy Status Location X, Location Y → Table ID, Length X, Length Y, Seats, Occupancy Status			
	Candidate keys	**Table ID**, {Location X, Location Y}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Table ID)		
	3NF	**Yes**	Location X, Location Y → Table ID, Length X, Length Y, Seats, Occupancy Status is a trivial FD since it returns the key of the table - Table ID. No other non-prime attributes can be determined from any other non-prime attributes. There are no other nontrivial FDs. The table is in 3NF since a trivial FD that returns the table key does not prevent a table from being in 3NF.		
	BCNF	**Yes**	No transitive functional dependencies exist.		
	Physical Design				
	Physical Design	tableMap			
	Primary Key	tableId			
	Foreign Keys	-			
	SQL Code	CREATE TABLE tableMap (tableId INTEGER, lengthX INTEGER, lengthY INTEGER, locationX INTEGER,			

		locationY INTEGER, seats INTEGER, occupancyStatus NUMBER(1), PRIMARY KEY (tableId), CONSTRAINT loc UNIQUE (locationX, locationY));
	Count of records in the table	30

Occupancy status was changed to integer, with 0 as FALSE and 1 as TRUE since Oracle does not support Boolean type.

	Name of the table	***Assigned Tables***			
	Description	Contains area IDs and names			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	
	Start Time	Start time on specified work date	integer	Between 1 to 9999999999	Seconds since reference time
	Table ID	Assigned table ID number	integer	1 to 999	
	Functional Dependencies and Keys				
	Functional dependencies	None			
	Candidate keys	{Staff ID, Start Time, Table ID}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (all attributes are part of the key, no FDs for this table)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute. (No non-prime attributes in the table)		
	BCNF	**Yes**	Yes, no FDs for this table.		
	Physical Design				
	Physical Design	assignedTables			
	Primary Key	{staffId, startTime, tableId}			
	Foreign Keys	{staffID, startTime} from staffSchedule TableId from tableMap			
	SQL Code	CREATE TABLE assignedTables (staffId INTEGER, startTime DATE, tableId INTEGER, PRIMARY KEY (staffId, startTime, tableId), FOREIGN KEY (staffID, startTime) REFERENCES staffSchedule(staffId, startTime), FOREIGN KEY (tableId) REFERENCES tableMap(tableId));			
	Count of records in the table	50			

Work date was removed. Key was changed to use start time instead.

	Name of the table	***Orders***			
	Description	Contains the order ID and table and server.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Order ID	ID number of order	integer	1 to 999999999	
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	Needed to determine server that gratuities are assigned to
	Table ID	ID number of table order is assigned to	integer	1 to 999	
	Functional Dependencies and Keys				
	Functional dependencies	Order ID→ Staff ID, Table ID			
	Candidate keys	Order ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Order ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	orders			
	Primary Key	orderId			
	Foreign Keys	staffId from staff tableId from tableMap			
	SQL Code	CREATE TABLE orders (orderId INTEGER, staffId INTEGER, tableId INTEGER, PRIMARY KEY (orderId), FOREIGN KEY (staffId) REFERENCES staff(staffId), FOREIGN KEY (tableId) REFERENCES tableMap(tableId));			
	Count of records in the table	50			

	Name of the table	***Menu***			
	Description	Contains the menu items and cost			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Item Name	Name of item from menu	string	Filet Mignon	
	Cost	Cost of the item	integer	1 to 999999999	Cost stored in cents, for example an item that is $20.00 is stored as 2000.
	Functional Dependencies and Keys				
	Functional dependencies	Item Name → Cost			
	Candidate keys	Item Name			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Item Name)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	menu			
	Primary Key	itemName			
	Foreign Keys	-			
	SQL Code	CREATE TABLE menu (itemName VARCHAR(50), cost INTEGER, PRIMARY KEY (itemName));			
	Count of records in the table	30			

	Name of the table	***Ordered Items***			
	Description	Contains the order ID and table and server.			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Order ID	ID number of order	integer	1 to 999999999	
	Item Name	Name of item from menu	integer	Filet Mignon	Items restricted to available items on the menu.
	Quantity	Quantity of named item	integer	1 to 999	
	Functional Dependencies and Keys				
	Functional dependencies	Order ID, Item Name→ Quantity			
	Candidate keys	{Order ID, Item Name}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (Quantity requires both Order ID and Item Name)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute (Only one non-prime attribute in table).		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	orderedItems			
	Primary Key	{orderId, itemName}			
	Foreign Keys	orderId from orders itemName from menu			
	SQL Code	CREATE TABLE orderedItems (orderId INTEGER, itemName VARCHAR(50), quantity INTEGER, PRIMARY KEY (orderId, itemName), FOREIGN KEY (orderId) REFERENCES orders(orderId), FOREIGN KEY (itemName) REFERENCES menu(itemName));			
	Count of records in the table	174			

	Name of the table	***Ingredients***			
	Description	Contains the ingredients for each item on the menu			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Item Name	Name of item from menu	string	Filet Mignon	
	Ingredient	Name of ingredient	string	Pepper	
	Functional Dependencies and Keys				
	Functional dependencies	None			
	Candidate keys	{Item Name, Ingredient}			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	No attribute is based on part of the key (all attributes are part of the key, no FDs for this table)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute. (No non-prime attributes in the table)		
	BCNF	**Yes**	Yes, no FDs for this table.		
	Physical Design				
	Physical Design	ingredients			
	Primary Key	{itemName, ingredient}			
	Foreign Keys	-			
	SQL Code	CREATE TABLE ingredients (itemName VARCHAR(50), ingredient VARCHAR(50), PRIMARY KEY (itemName, ingredient));			
	Count of records in the table	189			

	Name of the table	***Transactions***			
	Description	Contains the transactions and payment methods			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Transaction ID	ID number of transaction	integer	1 to 999999999	
	Payment Method	Method of Payment	string	Credit Card	
	Staff ID	ID number of an employee	integer	Between 1 and 9999999	Used for assigning gratuities
	Payment Quantity	Payment amount	integer	1 to 999999999	Cost stored in cents, for example a transaction that is $20.00 is stored as 2000.
	Functional Dependencies and Keys				
	Functional dependencies	Transaction ID→ Payment Method, Staff ID, Payment Quantity			
	Candidate keys	Transaction ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Transaction ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	transactions			
	Primary Key	transactionId			
	Foreign Keys	staffId from staff			
	SQL Code	CREATE TABLE transactions (transactionId INTEGER, paymentMethod VARCHAR(20), staffID INTEGER, paymentQuantity INTEGER, PRIMARY KEY (transactionId), FOREIGN KEY (staffId) REFERENCES staff(staffId));			
	Count of records in the table	75			

	Name of the table	***Customers***			
	Description	Contains the customer information			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Customer ID	ID number of customer	integer	1 to 999999999	Index value, increases by one for each new customer
	Name	Name of the customer	string	John Doe	
	Party Size	Number of total guests with customer	integer	Between 1 and 999	
	Phone Number	Phone number of customer	integer	1000000000 to 9999999999	Number stored without hyphens or other formatting, assumes US numbers only (10 digits)
	Functional Dependencies and Keys				
	Functional dependencies	Customer ID → Name, Party Size, Phone Number			
	Candidate keys	Customer ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Customer ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	customers			
	Primary Key	customerId			
	Foreign Keys	-			
	SQL Code	CREATE TABLE customers (customerId INTEGER, name VARCHAR(50), partySize INTEGER, phoneNumber NUMBER(10), PRIMARY KEY (customerId));			
	Count of records in the table	50			

	Name of the table	***Reservations***			
	Description	Contains the customer information			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Customer ID	ID number of customer	integer	1 to 999999999	Index value, increases by one for each new customer
	Reservation ID	ID number of reservation	integer	1 to 999999999	Index value, increases by one for each new reservation
	Time	Reservation time	integer	Between 1 to 9999999999	Seconds since reference time
	Functional Dependencies and Keys				
	Functional dependencies	Customer ID → Reservation ID, Time Reservation ID → Customer ID, Time			
	Candidate keys	**Customer ID**, Reservation ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Customer ID)		
	3NF	**Yes**	Reservation ID → Customer ID, Time is a trivial FD since it returns the key of the table - Customer ID. No other non-prime attributes can be determined from any other non-prime attributes. There are no other nontrivial FDs. The table is in 3NF since a trivial FD that returns the table key does not prevent a table from being in 3NF.		
	BCNF	**No**	Trivial FD exists in the table.		
	Physical Design				
	Physical Design	reservations			
	Primary Key	customerId			
	Foreign Keys	customerId from customers			
	SQL Code	CREATE TABLE reservations (customerId INTEGER, reservationId INTEGER, time DATE, PRIMARY KEY (customerId), FOREIGN KEY (customerId) REFERENCES customers(customerId));			
	Count of records in the table	40			

Date attribute was removed from table since Oracle DATE type supports day and time.

	Name of the table	***Wait Queue***			
	Description	Contains the wait queue position			
	Attribute	**Description**	**Type**	*Examples of values*	*Notes*
	Customer ID	ID number of customer	integer	1 to 999999999	Index value, increases by one for each new customer
	Wait Queue Position	Current wait position	integer	1 to 999	Updated as customer is moved up the queue
	Functional Dependencies and Keys				
	Functional dependencies	Customer ID → Wait Queue Position			
	Candidate keys	Customer ID			
	Normalization				
	1NF	**Yes**	All data contains single values.		
	2NF	**Yes**	Key is single attribute (Customer ID)		
	3NF	**Yes**	No non-prime attributes are determined by another non-prime attribute.		
	BCNF	**Yes**	Yes, only attributes in FD in table.		
	Physical Design				
	Physical Design	waitQueue			
	Primary Key	customerId			
	Foreign Keys	customerId from customers			
	SQL Code	CREATE TABLE waitQueue (customerId INTEGER, position INTEGER, PRIMARY KEY (customerId), FOREIGN KEY (customerId) REFERENCES customers(customerId));			
	Count of records in the table	10			

MILESTONE 6: SQL QUERIES

The first query performs data entry for creating a new order. Subsequent queries provide information that can be used for data analytics and staffing/inventory management. Several of the queries use variables, that will need to be specified when running the code.

Query 1	New Order
English version	Create a new order and add items. Order id is generated from the max value in the table. Add new item to ordered item table if it does not exist. Increment the quantity by one for the order if the item has already been ordered.
Source for the query need in the SRS document	SRS document, page 19, Section 8.1, Test 8.1.1

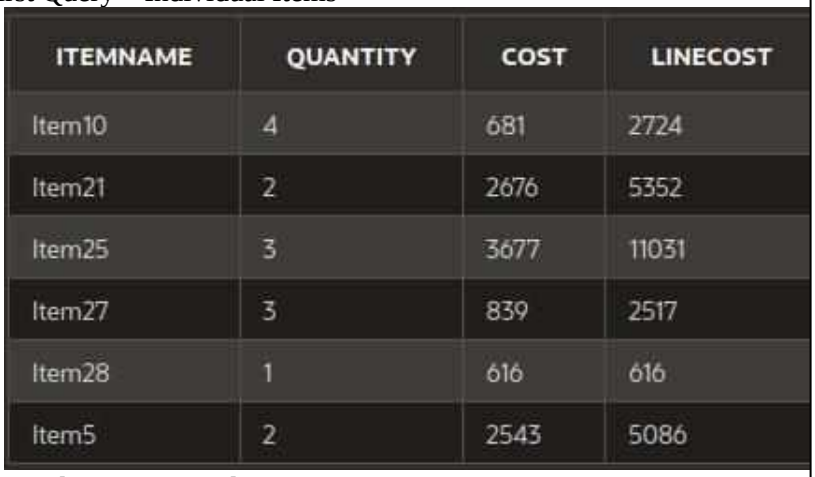
SQL sentence	--Create the order, assigning the staff and table
	DECLARE
	orderNum INTEGER;
	BEGIN
	--First statement returns the next available number for order ID, used when creating a new order.
	SELECT MAX(orderId) + 1 INTO orderNum FROM orders;
	INSERT INTO orders (orderId, staffId, tableId) values
	(orderNum, :staffNum, :tableNum);
	END;
	/
	--Add an item to an order. First checks to see if item is already in an order and adds one to quantity if it is, otherwise add a new row to the table with quantity = 1. In this example sequence, values for orderNum and item are preset, but would normally use values from the sales system.
	
	DECLARE
	orderNum integer :=51;
	item VARCHAR(50) :='Item21';
	quant integer;
	BEGIN
	SELECT quantity INTO quant FROM orderedItems
	WHERE orderId = orderNum AND itemName = item;
	quant := quant + 1;
	UPDATE orderedItems
	SET quantity = quant
	WHERE orderId = orderNum AND itemName = item;
	EXCEPTION
	WHEN NO_DATA_FOUND THEN
	quant := 1;
	IF quant = 1 THEN
	INSERT INTO orderedItems (orderId, itemName, quantity)
	VALUES (orderNum, item, quant);
	END IF;

	END; /
Example of returned rows (cropped screen caption)	 The only returned row from this query is the order ID to be used for a new order. The remainder of the SQL statements insert data into the database.

Query 2	Get ingredients for items ordered.
English version	Join ordered items in order with ingredients to create a list of ingredients for a specified order.
Source for the query need in the SRS document	SRS document, Page 7, Section 2.3, Kitchen staff stakeholder class – inventory management
SQL sentence	SELECT orderid, ordereditems.itemname, ingredient FROM ordereditems RIGHT JOIN ingredients ON ingredients.itemname=ordereditems.itemname WHERE ordereditems.orderid= :orderNum;
Example of returned rows (cropped screen caption)	

ITEMNAME	INGREDIENT
Item20	Ingredient14
Item20	Ingredient16
Item20	Ingredient48
Item20	Ingredient5
Item21	Ingredient15
Item21	Ingredient16
Item21	Ingredient18
Item21	Ingredient19
Item21	Ingredient2
Item21	Ingredient20
Item21	Ingredient23
Item21	Ingredient26
Item21	Ingredient5
Item21	Ingredient7

14 rows returned in 0.00 seconds Download

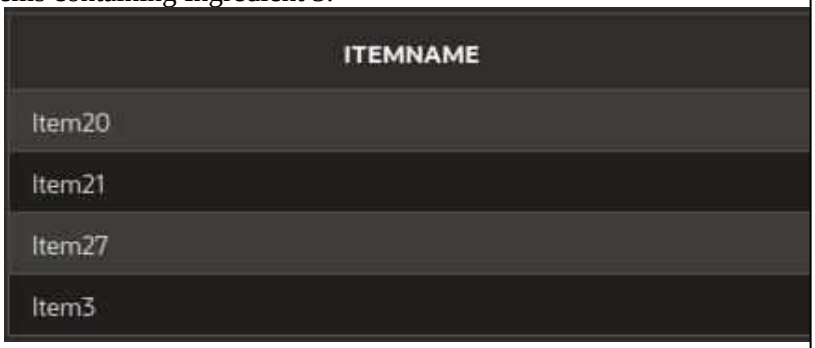
Query 3	Create bill
English version	Combine cost of items with items ordered. Calculate the cost for each item based on quantity ordered. Add up the combined cost for each item to determine a total bill amount.
Source for the query need in the SRS document	SRS document, page 57, Section 9.8, Closeout table order
SQL sentence	--This first query gives cost for each line SELECT orderedItems.itemName, quantity, cost, quantity * cost AS lineCost FROM orderedItems RIGHT JOIN menu ON orderedItems.itemName=menu.itemName WHERE orderid = :orderNum; --Gives the bill total SELECT SUM(newtable. lineCost) FROM (SELECT orderedItems.itemName, quantity, cost, quantity * cost AS lineCost FROM orderedItems RIGHT JOIN menu ON orderedItems.itemName=menu.itemName WHERE orderid = :orderNum) newtable;
Example of returned rows (cropped screen caption)	First Query – Individual Items  Second Query – Total Cost 

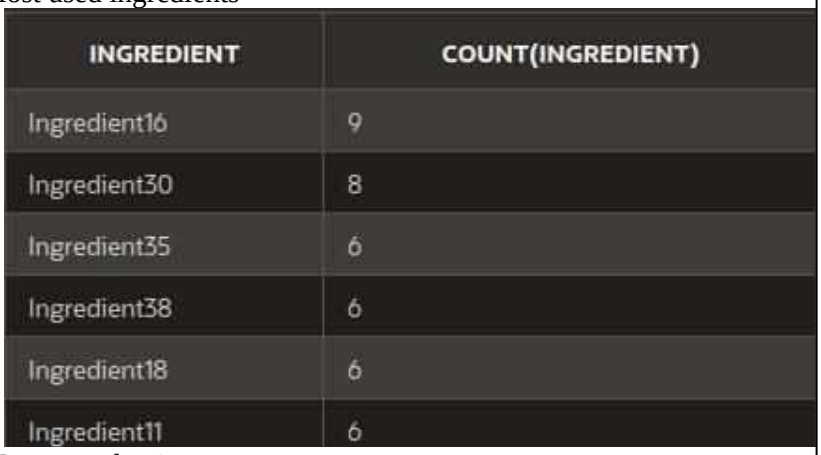
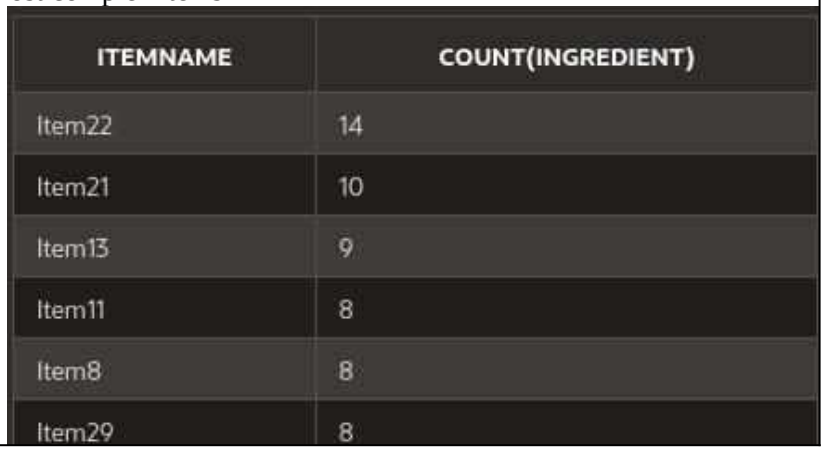
ITEMNAME	QUANTITY	COST	LINECOST
Item10	4	681	2724
Item21	2	2676	5352
Item25	3	3677	11031
Item27	3	839	2517
Item28	1	616	616
Item5	2	2543	5086

SUM(NEWTABLE.LINECOST)
27326

Query 4	Get reservations totals for specified date
English version	Combine the party size with reservations. Show number of reservations grouped by party size for a specified date.
Source for the query need in the SRS document	SRS document, Page 11, Section 3.5.4, 3.5.4.3. System Customer Tracking Protocol
SQL sentence	SELECT customers.partysize, COUNT(*) FROM reservations RIGHT JOIN customers ON reservations.customerId=customers.customerId WHERE TRUNC(reservations.time) = TO_DATE(:resDate, 'DD-MM-YYYY') GROUP BY partySize ORDER BY partySize;
Example of returned rows (cropped screen caption)	

PARTYSIZE	COUNT(*)
1	1
2	2
3	1
5	2
7	1
9	2

Query 5	Allergy list
English version	Create a list from the menu of items with specified ingredients, used for determining items a customer may need to avoid. The second query takes the items containing the specified ingredients and removes those items from the menu, giving a list of items without that ingredient.
Source for the query need in the SRS document	Not specified but a good safety feature.
SQL sentence	--For items with ingredient SELECT itemName FROM ingredients WHERE ingredient = :allergyIng; --For items without ingredient SELECT UNIQUE itemName FROM ingredients MINUS SELECT itemName FROM ingredients WHERE ingredient = :allergyIng;
Example of returned rows (cropped screen caption)	Items containing Ingredient 5.  Items not containing Ingredient 5. (List longer than shown) 

Query 6	Most used ingredients / Most complex items
English version	Count the number of ingredients in each item and show the most used ingredients. The second query returns those with more than five ingredients to show the most complex items for the kitchen to make.
Source for the query need in the SRS document	SRS document, Page 7, Section 2.3, Kitchen staff stakeholder class – inventory management
SQL sentence	--Gives most used ingredients SELECT ingredient, count(ingredient) FROM ingredients GROUP BY ingredient HAVING count(ingredient) > 5 ORDER BY count(ingredient) DESC; --Gives most complex menu items SELECT itemName, count(ingredient) FROM ingredients GROUP BY itemName HAVING count(ingredient) > 5 ORDER BY count(ingredient) DESC;
Example of returned rows (cropped screen caption)	Most used ingredients  Most complex items 

Query 7	Most sold items for a server
English version	Combines the ordered items with orders table and sums each item sold for a server. Provides the bestselling items for the specified server.
Source for the query need in the SRS document	SRS document, Page 39, Section 8.1.20, Track Customer Ordered Items
SQL sentence	SELECT ordereditems.itemName, SUM(quantity) FROM orders RIGHT JOIN ordereditems ON orders.orderId=ordereditems.orderId WHERE staffID=:staffNum GROUP BY ordereditems.itemname ORDER BY SUM(quantity) DESC;
Example of returned rows (cropped screen caption)	Most sold items for staff ID #9.

ITEMNAME	SUM(QUANTITY)
Item17	7
Item22	7
Item27	7
Item1	6
Item21	6
Item30	6
Item11	6
Item28	5

Query 8	Large order quantity - times server had order size greater than 10.
English version	Combines the orders table with ordered items and gives the number of times a staff member had to serve an order greater than 10 items. Can be used to track if large orders are causeing staff to be overburdened.
Source for the query need in the SRS document	SRS document, Page 39, Section 8.1.20, Track Customer Ordered Items
SQL sentence	SELECT newtable.staffID, count(*) FROM (SELECT orders.orderId, staffID, sum(quantity) FROM orders RIGHT JOIN ordereditems ON orders.orderId=ordereditems.orderId GROUP BY orders.orderid, staffID HAVING sum(quantity) > 10) newtable GROUP BY staffid;
Example of returned rows (cropped screen caption)	

STAFFID	COUNT(*)
11	6
12	5
10	3
9	5
13	5

APPENDIX

Table Creation Code

```
CREATE TABLE staff (
  staffId  INTEGER,
  name  VARCHAR(50),
  ssn  NUMBER(9),
  phoneNumber  NUMBER(10),
  address VARCHAR(50),
  PRIMARY KEY (staffId)
);

CREATE TABLE areaNames (
  areaId  INTEGER,
  areaName  VARCHAR(20),
  PRIMARY KEY (areaId)
);

CREATE TABLE staffSchedule (
  staffId  INTEGER,
  startTime  DATE,
  endTime  DATE,
  areaId  INTEGER,
  PRIMARY KEY (staffId, startTime),
    FOREIGN KEY (staffId) REFERENCES staff(staffId),
    FOREIGN KEY (areaID) REFERENCES areaNames(areaId)
);

CREATE TABLE tableMap (
  tableId  INTEGER,
  lengthX  INTEGER,
  lengthY  INTEGER,
  locationX  INTEGER,
  locationY  INTEGER,
  seats  INTEGER,
  occupancyStatus NUMBER(1),
  PRIMARY KEY (tableId)
);

CREATE TABLE assignedTables (
  staffId  INTEGER,
  startTime  DATE,
  tableId  INTEGER,
  PRIMARY KEY (staffId, startTime, tableId),
    FOREIGN KEY (staffID, startTime) REFERENCES staffSchedule(staffId,
startTime),
    FOREIGN KEY (tableId) REFERENCES tableMap(tableId)
```

```
);

CREATE TABLE orders (
  orderId  INTEGER,
  staffId  INTEGER,
  tableId  INTEGER,
  PRIMARY KEY (orderId),
    FOREIGN KEY (staffId) REFERENCES staff(staffId),
    FOREIGN KEY (tableId) REFERENCES tableMap(tableId)
);

CREATE TABLE menu (
  itemName  VARCHAR(50),
  cost  INTEGER,
  PRIMARY KEY (itemName)
);

CREATE TABLE orderedItems (
  orderId  INTEGER,
  itemName  VARCHAR(50),
  quantity INTEGER,
  PRIMARY KEY (orderId, itemName),
    FOREIGN KEY (orderId) REFERENCES orders(orderId),
    FOREIGN KEY (itemName) REFERENCES menu(itemName)
);

CREATE TABLE ingredients (
  itemName  VARCHAR(50),
  ingredient  VARCHAR(50),
  PRIMARY KEY (itemName, ingredient)
);

CREATE TABLE transactions (
  transactionId  INTEGER,
  paymentMethod  VARCHAR(20),
  staffID  INTEGER,
  paymentQuantity  INTEGER,
  PRIMARY KEY (transactionId),
    FOREIGN KEY (staffId) REFERENCES staff(staffId)
);

CREATE TABLE customers (
  customerId  INTEGER,
  name  VARCHAR(50),
  partySize  INTEGER,
  phoneNumber  NUMBER(10),
  PRIMARY KEY (customerId)
);

CREATE TABLE reservations (
```

```
  customerId  INTEGER,
  reservationId  INTEGER,
  time  DATE,
  PRIMARY KEY (customerId),
    FOREIGN KEY (customerId) REFERENCES customers(customerId)
);

CREATE TABLE waitQueue (
  customerId  INTEGER,
  position  INTEGER,
  PRIMARY KEY (customerId),
    FOREIGN KEY (customerId) REFERENCES customers(customerId)
);
```

Data

```
--Staff Data
insert into staff (staffId, name, ssn, phoneNumber, address) values (1, 'Bobbette Lindbergh', '266340192', '9339732579', '86 High Crossing Court');
insert into staff (staffId, name, ssn, phoneNumber, address) values (2, 'Marika Drogan', '679735355', '7375235370', '843 Grasskamp Point');
insert into staff (staffId, name, ssn, phoneNumber, address) values (3, 'Maye Breacher', '551168857', '5015522446', '9 Myrtle Parkway');
insert into staff (staffId, name, ssn, phoneNumber, address) values (4, 'Kerry Howsin', '062575602', '1058499438', '89010 Sherman Drive');
insert into staff (staffId, name, ssn, phoneNumber, address) values (5, 'Garek Puttergill', '665654435', '6338502004', '4915 Ohio Court');
insert into staff (staffId, name, ssn, phoneNumber, address) values (6, 'Rex Duddell', '671208513', '9770168820', '11105 Mendota Point');
insert into staff (staffId, name, ssn, phoneNumber, address) values (7, 'Dorie Bourgaize', '297929664', '2636159522', '637 Vera Crossing');
insert into staff (staffId, name, ssn, phoneNumber, address) values (8, 'Casey Bickford', '932292556', '0690064247', '60200 Spohn Crossing');
insert into staff (staffId, name, ssn, phoneNumber, address) values (9, 'Hymie Corder', '429085200', '2902126373', '71624 Saint Paul Lane');
insert into staff (staffId, name, ssn, phoneNumber, address) values (10, 'Rosalind Wadsworth', '609176095', '6083699388', '43 Londonderry Plaza');
insert into staff (staffId, name, ssn, phoneNumber, address) values (11, 'Shannen Allsup', '128197464', '9646074633', '0506 Rusk Way');
insert into staff (staffId, name, ssn, phoneNumber, address) values (12, 'Florina Binham', '815845040', '3666539016', '7268 Derek Street');
insert into staff (staffId, name, ssn, phoneNumber, address) values (13, 'Helen-elizabeth Saberton', '015772798', '0773550690', '940 Nova Plaza');
insert into staff (staffId, name, ssn, phoneNumber, address) values (14, 'Charline Brockherst', '529053057', '4159553289', '7 Transport Hill');
insert into staff (staffId, name, ssn, phoneNumber, address) values (15, 'Jilleen Bollam', '969099539', '2376347209', '92033 Manley Road');
insert into staff (staffId, name, ssn, phoneNumber, address) values (16, 'Cassie Rubenov', '708762912', '3563781823', '9 Surrey Court');
insert into staff (staffId, name, ssn, phoneNumber, address) values (17, 'Gibbie Joyson', '957711550', '3045940971', '1 Stephen Terrace');
insert into staff (staffId, name, ssn, phoneNumber, address) values (18, 'Kandace Fairbrace', '686064296', '9107277339', '1 Weeping Birch Circle');
insert into staff (staffId, name, ssn, phoneNumber, address) values (19, 'Georgie Ladloe', '063058153', '7466394202', '61 Everett Point');
insert into staff (staffId, name, ssn, phoneNumber, address) values (20, 'Laurene Latta', '079514534', '4435446966', '71 Talmadge Lane');

--Area Names Data
insert into areaNames (areaId, areaName) values (1, 'Host');
insert into areaNames (areaId, areaName) values (2, 'Dining Room');
insert into areaNames (areaId, areaName) values (3, 'Bar');
insert into areaNames (areaId, areaName) values (4, 'Kitchen');
```

```
insert into areaNames (areaId, areaName) values (5, 'Dishwasher');
insert into areaNames (areaId, areaName) values (6, 'Busser');
insert into areaNames (areaId, areaName) values (7, 'Management');

--Staff Schedule
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (1, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 7);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (2, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 7);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (3, TO_DATE('2023-12-10
12:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 20:00:00', 'yyyy-mm-dd hh24:mi:ss'), 7);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (4, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 4);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (5, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 4);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (6, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 4);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (7, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 4);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (8, TO_DATE('2023-12-10
12:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 20:00:00', 'yyyy-mm-dd hh24:mi:ss'), 4);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (9, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 2);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (10, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 2);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (11, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 2);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (12, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 2);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (13, TO_DATE('2023-12-10
12:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 20:00:00', 'yyyy-mm-dd hh24:mi:ss'), 2);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (14, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 3);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (15, TO_DATE('2023-12-10
12:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 20:00:00', 'yyyy-mm-dd hh24:mi:ss'), 3);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (16, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 5);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (17, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 5);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (18, TO_DATE('2023-12-10
06:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 14:00:00', 'yyyy-mm-dd hh24:mi:ss'), 1);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (19, TO_DATE('2023-12-10
14:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 22:00:00', 'yyyy-mm-dd hh24:mi:ss'), 1);
insert into staffSchedule (staffId, startTime, endTime, AreaId) values (20, TO_DATE('2023-12-10
12:00:00', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2023-12-10 20:00:00', 'yyyy-mm-dd hh24:mi:ss'), 6);

--Table Map
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (1,
40, 72, 480, 565, 5, 0);
```

```
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (2, 71, 93, 197, 273, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (3, 73, 61, 463, 465, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (4, 88, 109, 152, 568, 9, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (5, 93, 65, 489, 103, 4, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (6, 106, 58, 157, 370, 2, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (7, 92, 70, 0, 366, 9, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (8, 112, 120, 363, 123, 6, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (9, 55, 104, 170, 389, 7, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (10, 120, 81, 63, 223, 5, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (11, 60, 111, 466, 506, 5, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (12, 119, 66, 543, 434, 7, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (13, 88, 109, 340, 11, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (14, 94, 80, 345, 361, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (15, 94, 62, 311, 185, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (16, 42, 76, 402, 66, 4, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (17, 108, 113, 343, 590, 7, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (18, 81, 63, 382, 30, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (19, 74, 104, 347, 461, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (20, 120, 92, 146, 233, 6, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (21, 92, 111, 346, 182, 7, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (22, 100, 96, 386, 579, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (23, 46, 114, 503, 407, 6, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (24, 88, 65, 400, 411, 8, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (25, 67, 59, 156, 481, 8, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (26, 99, 71, 252, 281, 9, 0);
```

```
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (27, 103, 89, 192, 225, 10, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (28, 90, 38, 20, 156, 3, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (29, 44, 56, 10, 253, 8, 0);
insert into tableMap (tableId, lengthX, lengthY, locationX, locationY, seats, occupancyStatus) values (30, 97, 71, 491, 459, 3, 0);

--Assigned Tables
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 1);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 2);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 3);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 4);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 5);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 6);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 7);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 8);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 9);
insert into assignedTables (staffId, startTime, tableID) values (9, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 10);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 11);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 12);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 13);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 14);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 15);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 16);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 17);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 18);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 19);
insert into assignedTables (staffId, startTime, tableID) values (10, TO_DATE('2023-12-10 06:00:00', 'yyyy-mm-dd hh24:mi:ss'), 20);
```

```
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 21);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 22);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 23);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 24);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 25);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 26);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 27);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 28);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 29);
insert into assignedTables (staffId, startTime, tableID) values (13, TO_DATE('2023-12-10 12:00:00',
'yyyy-mm-dd hh24:mi:ss'), 30);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 1);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 2);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 3);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 4);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 5);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 6);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 7);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 8);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 9);
insert into assignedTables (staffId, startTime, tableID) values (11, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 10);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 11);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 12);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 13);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 14);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 15);
```

```
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 16);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 17);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 18);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 19);
insert into assignedTables (staffId, startTime, tableID) values (12, TO_DATE('2023-12-10 14:00:00',
'yyyy-mm-dd hh24:mi:ss'), 20);

--Orders
insert into orders (orderId, staffId, tableId) values (1, 12, 25);
insert into orders (orderId, staffId, tableId) values (2, 11, 15);
insert into orders (orderId, staffId, tableId) values (3, 10, 25);
insert into orders (orderId, staffId, tableId) values (4, 13, 28);
insert into orders (orderId, staffId, tableId) values (5, 13, 23);
insert into orders (orderId, staffId, tableId) values (6, 10, 17);
insert into orders (orderId, staffId, tableId) values (7, 9, 25);
insert into orders (orderId, staffId, tableId) values (8, 11, 6);
insert into orders (orderId, staffId, tableId) values (9, 11, 4);
insert into orders (orderId, staffId, tableId) values (10, 13, 20);
insert into orders (orderId, staffId, tableId) values (11, 11, 1);
insert into orders (orderId, staffId, tableId) values (12, 9, 28);
insert into orders (orderId, staffId, tableId) values (13, 13, 14);
insert into orders (orderId, staffId, tableId) values (14, 12, 18);
insert into orders (orderId, staffId, tableId) values (15, 11, 22);
insert into orders (orderId, staffId, tableId) values (16, 9, 14);
insert into orders (orderId, staffId, tableId) values (17, 13, 13);
insert into orders (orderId, staffId, tableId) values (18, 12, 28);
insert into orders (orderId, staffId, tableId) values (19, 12, 11);
insert into orders (orderId, staffId, tableId) values (20, 10, 30);
insert into orders (orderId, staffId, tableId) values (21, 9, 26);
insert into orders (orderId, staffId, tableId) values (22, 11, 10);
insert into orders (orderId, staffId, tableId) values (23, 9, 14);
insert into orders (orderId, staffId, tableId) values (24, 9, 16);
insert into orders (orderId, staffId, tableId) values (25, 9, 26);
insert into orders (orderId, staffId, tableId) values (26, 13, 14);
insert into orders (orderId, staffId, tableId) values (27, 10, 6);
insert into orders (orderId, staffId, tableId) values (28, 9, 2);
insert into orders (orderId, staffId, tableId) values (29, 11, 28);
insert into orders (orderId, staffId, tableId) values (30, 12, 8);
insert into orders (orderId, staffId, tableId) values (31, 12, 11);
insert into orders (orderId, staffId, tableId) values (32, 9, 18);
insert into orders (orderId, staffId, tableId) values (33, 13, 15);
insert into orders (orderId, staffId, tableId) values (34, 9, 29);
insert into orders (orderId, staffId, tableId) values (35, 9, 2);
insert into orders (orderId, staffId, tableId) values (36, 11, 21);
insert into orders (orderId, staffId, tableId) values (37, 9, 7);
insert into orders (orderId, staffId, tableId) values (38, 9, 28);
```

```
insert into orders (orderId, staffId, tableId) values (39, 9, 14);
insert into orders (orderId, staffId, tableId) values (40, 13, 21);
insert into orders (orderId, staffId, tableId) values (41, 9, 17);
insert into orders (orderId, staffId, tableId) values (42, 9, 29);
insert into orders (orderId, staffId, tableId) values (43, 12, 16);
insert into orders (orderId, staffId, tableId) values (44, 13, 8);
insert into orders (orderId, staffId, tableId) values (45, 13, 6);
insert into orders (orderId, staffId, tableId) values (46, 10, 11);
insert into orders (orderId, staffId, tableId) values (47, 9, 18);
insert into orders (orderId, staffId, tableId) values (48, 9, 23);
insert into orders (orderId, staffId, tableId) values (49, 11, 25);
insert into orders (orderId, staffId, tableId) values (50, 9, 1);

--Menu
insert into menu (itemName, cost) values ('Item1', 757);
insert into menu (itemName, cost) values ('Item2', 3741);
insert into menu (itemName, cost) values ('Item3', 4143);
insert into menu (itemName, cost) values ('Item4', 3969);
insert into menu (itemName, cost) values ('Item5', 2543);
insert into menu (itemName, cost) values ('Item6', 3658);
insert into menu (itemName, cost) values ('Item7', 3624);
insert into menu (itemName, cost) values ('Item8', 4301);
insert into menu (itemName, cost) values ('Item9', 2506);
insert into menu (itemName, cost) values ('Item10', 681);
insert into menu (itemName, cost) values ('Item11', 1409);
insert into menu (itemName, cost) values ('Item12', 3004);
insert into menu (itemName, cost) values ('Item13', 1906);
insert into menu (itemName, cost) values ('Item14', 483);
insert into menu (itemName, cost) values ('Item15', 2808);
insert into menu (itemName, cost) values ('Item16', 3066);
insert into menu (itemName, cost) values ('Item17', 4728);
insert into menu (itemName, cost) values ('Item18', 1013);
insert into menu (itemName, cost) values ('Item19', 2696);
insert into menu (itemName, cost) values ('Item20', 3862);
insert into menu (itemName, cost) values ('Item21', 2676);
insert into menu (itemName, cost) values ('Item22', 2521);
insert into menu (itemName, cost) values ('Item23', 2890);
insert into menu (itemName, cost) values ('Item24', 1503);
insert into menu (itemName, cost) values ('Item25', 3677);
insert into menu (itemName, cost) values ('Item26', 2266);
insert into menu (itemName, cost) values ('Item27', 839);
insert into menu (itemName, cost) values ('Item28', 616);
insert into menu (itemName, cost) values ('Item29', 374);
insert into menu (itemName, cost) values ('Item30', 3475);

--Ordered Items
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item21', 3);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item5', 1);
insert into orderedItems (orderId, itemName, quantity) values (18, 'Item19', 4);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item4', 1);
```

```
insert into orderedItems (orderId, itemName, quantity) values (21, 'Item27', 3);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item11', 4);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item30', 2);
insert into orderedItems (orderId, itemName, quantity) values (28, 'Item28', 1);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item5', 2);
insert into orderedItems (orderId, itemName, quantity) values (30, 'Item26', 4);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item12', 2);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item11', 2);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item27', 1);
insert into orderedItems (orderId, itemName, quantity) values (11, 'Item26', 4);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item28', 2);
insert into orderedItems (orderId, itemName, quantity) values (20, 'Item29', 1);
insert into orderedItems (orderId, itemName, quantity) values (21, 'Item21', 2);
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item3', 2);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item3', 3);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item22', 3);
insert into orderedItems (orderId, itemName, quantity) values (13, 'Item28', 4);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item7', 3);
insert into orderedItems (orderId, itemName, quantity) values (11, 'Item18', 1);
insert into orderedItems (orderId, itemName, quantity) values (13, 'Item20', 3);
insert into orderedItems (orderId, itemName, quantity) values (20, 'Item6', 1);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item5', 4);
insert into orderedItems (orderId, itemName, quantity) values (3, 'Item1', 1);
insert into orderedItems (orderId, itemName, quantity) values (22, 'Item29', 1);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item20', 1);
insert into orderedItems (orderId, itemName, quantity) values (20, 'Item25', 1);
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item17', 3);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item17', 1);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item9', 3);
insert into orderedItems (orderId, itemName, quantity) values (18, 'Item20', 4);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item1', 1);
insert into orderedItems (orderId, itemName, quantity) values (16, 'Item27', 3);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item9', 2);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item18', 4);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item24', 2);
insert into orderedItems (orderId, itemName, quantity) values (5, 'Item3', 3);
insert into orderedItems (orderId, itemName, quantity) values (11, 'Item16', 3);
insert into orderedItems (orderId, itemName, quantity) values (16, 'Item30', 1);
insert into orderedItems (orderId, itemName, quantity) values (4, 'Item30', 3);
insert into orderedItems (orderId, itemName, quantity) values (22, 'Item18', 2);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item29', 3);
insert into orderedItems (orderId, itemName, quantity) values (25, 'Item1', 2);
insert into orderedItems (orderId, itemName, quantity) values (19, 'Item5', 4);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item25', 4);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item19', 1);
insert into orderedItems (orderId, itemName, quantity) values (17, 'Item29', 1);
insert into orderedItems (orderId, itemName, quantity) values (12, 'Item12', 3);
insert into orderedItems (orderId, itemName, quantity) values (14, 'Item9', 2);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item15', 2);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item3', 1);
```

```
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item8', 2);
insert into orderedItems (orderId, itemName, quantity) values (4, 'Item13', 2);
insert into orderedItems (orderId, itemName, quantity) values (28, 'Item30', 3);
insert into orderedItems (orderId, itemName, quantity) values (28, 'Item22', 3);
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item2', 1);
insert into orderedItems (orderId, itemName, quantity) values (18, 'Item14', 2);
insert into orderedItems (orderId, itemName, quantity) values (20, 'Item13', 3);
insert into orderedItems (orderId, itemName, quantity) values (30, 'Item5', 3);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item4', 3);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item28', 3);
insert into orderedItems (orderId, itemName, quantity) values (13, 'Item7', 2);
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item28', 3);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item10', 3);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item2', 3);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item27', 4);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item12', 3);
insert into orderedItems (orderId, itemName, quantity) values (21, 'Item28', 1);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item10', 4);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item18', 2);
insert into orderedItems (orderId, itemName, quantity) values (6, 'Item30', 4);
insert into orderedItems (orderId, itemName, quantity) values (30, 'Item7', 1);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item23', 4);
insert into orderedItems (orderId, itemName, quantity) values (18, 'Item4', 2);
insert into orderedItems (orderId, itemName, quantity) values (22, 'Item8', 2);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item1', 2);
insert into orderedItems (orderId, itemName, quantity) values (6, 'Item26', 1);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item19', 1);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item16', 3);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item16', 1);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item28', 1);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item17', 1);
insert into orderedItems (orderId, itemName, quantity) values (17, 'Item6', 2);
insert into orderedItems (orderId, itemName, quantity) values (14, 'Item11', 4);
insert into orderedItems (orderId, itemName, quantity) values (4, 'Item24', 2);
insert into orderedItems (orderId, itemName, quantity) values (20, 'Item18', 4);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item7', 1);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item11', 3);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item30', 4);
insert into orderedItems (orderId, itemName, quantity) values (18, 'Item15', 3);
insert into orderedItems (orderId, itemName, quantity) values (21, 'Item25', 3);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item22', 4);
insert into orderedItems (orderId, itemName, quantity) values (5, 'Item16', 3);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item3', 1);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item6', 2);
insert into orderedItems (orderId, itemName, quantity) values (12, 'Item1', 2);
insert into orderedItems (orderId, itemName, quantity) values (5, 'Item14', 1);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item7', 3);
insert into orderedItems (orderId, itemName, quantity) values (29, 'Item4', 3);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item29', 1);
insert into orderedItems (orderId, itemName, quantity) values (19, 'Item7', 3);
```

```
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item22', 3);
insert into orderedItems (orderId, itemName, quantity) values (22, 'Item6', 4);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item6', 2);
insert into orderedItems (orderId, itemName, quantity) values (14, 'Item26', 4);
insert into orderedItems (orderId, itemName, quantity) values (4, 'Item15', 3);
insert into orderedItems (orderId, itemName, quantity) values (20, 'Item2', 2);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item15', 2);
insert into orderedItems (orderId, itemName, quantity) values (29, 'Item7', 3);
insert into orderedItems (orderId, itemName, quantity) values (19, 'Item10', 4);
insert into orderedItems (orderId, itemName, quantity) values (12, 'Item21', 2);
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item26', 1);
insert into orderedItems (orderId, itemName, quantity) values (11, 'Item29', 3);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item6', 1);
insert into orderedItems (orderId, itemName, quantity) values (7, 'Item11', 3);
insert into orderedItems (orderId, itemName, quantity) values (6, 'Item8', 3);
insert into orderedItems (orderId, itemName, quantity) values (11, 'Item19', 1);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item17', 2);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item10', 4);
insert into orderedItems (orderId, itemName, quantity) values (30, 'Item20', 1);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item27', 1);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item24', 4);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item1', 2);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item13', 3);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item1', 2);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item26', 1);
insert into orderedItems (orderId, itemName, quantity) values (28, 'Item23', 1);
insert into orderedItems (orderId, itemName, quantity) values (13, 'Item25', 3);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item19', 2);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item17', 3);
insert into orderedItems (orderId, itemName, quantity) values (27, 'Item9', 3);
insert into orderedItems (orderId, itemName, quantity) values (30, 'Item21', 2);
insert into orderedItems (orderId, itemName, quantity) values (21, 'Item10', 4);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item26', 4);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item28', 2);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item19', 3);
insert into orderedItems (orderId, itemName, quantity) values (13, 'Item3', 2);
insert into orderedItems (orderId, itemName, quantity) values (5, 'Item8', 4);
insert into orderedItems (orderId, itemName, quantity) values (17, 'Item16', 4);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item23', 2);
insert into orderedItems (orderId, itemName, quantity) values (14, 'Item2', 3);
insert into orderedItems (orderId, itemName, quantity) values (28, 'Item17', 2);
insert into orderedItems (orderId, itemName, quantity) values (26, 'Item20', 1);
insert into orderedItems (orderId, itemName, quantity) values (21, 'Item5', 2);
insert into orderedItems (orderId, itemName, quantity) values (12, 'Item3', 3);
insert into orderedItems (orderId, itemName, quantity) values (11, 'Item17', 4);
insert into orderedItems (orderId, itemName, quantity) values (3, 'Item28', 4);
insert into orderedItems (orderId, itemName, quantity) values (30, 'Item28', 1);
insert into orderedItems (orderId, itemName, quantity) values (17, 'Item21', 1);
insert into orderedItems (orderId, itemName, quantity) values (15, 'Item3', 2);
insert into orderedItems (orderId, itemName, quantity) values (12, 'Item11', 3);
```

```
insert into orderedItems (orderId, itemName, quantity) values (22, 'Item2', 2);
insert into orderedItems (orderId, itemName, quantity) values (17, 'Item13', 1);
insert into orderedItems (orderId, itemName, quantity) values (1, 'Item20', 2);
insert into orderedItems (orderId, itemName, quantity) values (16, 'Item22', 1);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item20', 2);
insert into orderedItems (orderId, itemName, quantity) values (25, 'Item20', 2);
insert into orderedItems (orderId, itemName, quantity) values (22, 'Item14', 3);
insert into orderedItems (orderId, itemName, quantity) values (6, 'Item9', 3);
insert into orderedItems (orderId, itemName, quantity) values (13, 'Item13', 2);
insert into orderedItems (orderId, itemName, quantity) values (14, 'Item6', 4);
insert into orderedItems (orderId, itemName, quantity) values (12, 'Item4', 3);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item21', 2);
insert into orderedItems (orderId, itemName, quantity) values (8, 'Item7', 1);
insert into orderedItems (orderId, itemName, quantity) values (4, 'Item8', 3);
insert into orderedItems (orderId, itemName, quantity) values (10, 'Item27', 4);
insert into orderedItems (orderId, itemName, quantity) values (2, 'Item16', 2);
insert into orderedItems (orderId, itemName, quantity) values (25, 'Item9', 1);
insert into orderedItems (orderId, itemName, quantity) values (9, 'Item17', 4);
insert into orderedItems (orderId, itemName, quantity) values (23, 'Item13', 4);
insert into orderedItems (orderId, itemName, quantity) values (24, 'Item30', 2);

--Ingredients
insert into ingredients (itemName, ingredient) values ('Item27', 'Ingredient29');
insert into ingredients (itemName, ingredient) values ('Item3', 'Ingredient5');
insert into ingredients (itemName, ingredient) values ('Item6', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item17', 'Ingredient50');
insert into ingredients (itemName, ingredient) values ('Item27', 'Ingredient32');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient35');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient26');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient18');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient10');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient4');
insert into ingredients (itemName, ingredient) values ('Item25', 'Ingredient38');
insert into ingredients (itemName, ingredient) values ('Item27', 'Ingredient18');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient35');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient29');
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient28');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient31');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient27');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient48');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient46');
insert into ingredients (itemName, ingredient) values ('Item3', 'Ingredient12');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient31');
insert into ingredients (itemName, ingredient) values ('Item9', 'Ingredient17');
insert into ingredients (itemName, ingredient) values ('Item14', 'Ingredient4');
insert into ingredients (itemName, ingredient) values ('Item6', 'Ingredient20');
insert into ingredients (itemName, ingredient) values ('Item2', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item9', 'Ingredient20');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient11');
```

```
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient12');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient38');
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient35');
insert into ingredients (itemName, ingredient) values ('Item9', 'Ingredient35');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient7');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient3');
insert into ingredients (itemName, ingredient) values ('Item6', 'Ingredient19');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient46');
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient32');
insert into ingredients (itemName, ingredient) values ('Item2', 'Ingredient11');
insert into ingredients (itemName, ingredient) values ('Item20', 'Ingredient5');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient34');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient7');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient21');
insert into ingredients (itemName, ingredient) values ('Item2', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient50');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient4');
insert into ingredients (itemName, ingredient) values ('Item28', 'Ingredient42');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient18');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient5');
insert into ingredients (itemName, ingredient) values ('Item9', 'Ingredient3');
insert into ingredients (itemName, ingredient) values ('Item10', 'Ingredient42');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient17');
insert into ingredients (itemName, ingredient) values ('Item3', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient42');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient47');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient18');
insert into ingredients (itemName, ingredient) values ('Item17', 'Ingredient38');
insert into ingredients (itemName, ingredient) values ('Item6', 'Ingredient48');
insert into ingredients (itemName, ingredient) values ('Item12', 'Ingredient9');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient35');
insert into ingredients (itemName, ingredient) values ('Item20', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item12', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient33');
insert into ingredients (itemName, ingredient) values ('Item23', 'Ingredient2');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient37');
insert into ingredients (itemName, ingredient) values ('Item14', 'Ingredient43');
insert into ingredients (itemName, ingredient) values ('Item4', 'Ingredient28');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item14', 'Ingredient32');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient40');
insert into ingredients (itemName, ingredient) values ('Item19', 'Ingredient21');
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient1');
insert into ingredients (itemName, ingredient) values ('Item4', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item28', 'Ingredient44');
insert into ingredients (itemName, ingredient) values ('Item19', 'Ingredient46');
insert into ingredients (itemName, ingredient) values ('Item4', 'Ingredient22');
insert into ingredients (itemName, ingredient) values ('Item28', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient18');
```

```
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient6');
insert into ingredients (itemName, ingredient) values ('Item7', 'Ingredient23');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient11');
insert into ingredients (itemName, ingredient) values ('Item7', 'Ingredient6');
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient25');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item14', 'Ingredient34');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient33');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient20');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient27');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient12');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient28');
insert into ingredients (itemName, ingredient) values ('Item17', 'Ingredient7');
insert into ingredients (itemName, ingredient) values ('Item17', 'Ingredient35');
insert into ingredients (itemName, ingredient) values ('Item6', 'Ingredient29');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient2');
insert into ingredients (itemName, ingredient) values ('Item6', 'Ingredient39');
insert into ingredients (itemName, ingredient) values ('Item23', 'Ingredient1');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient7');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient47');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient26');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient41');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient50');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient40');
insert into ingredients (itemName, ingredient) values ('Item19', 'Ingredient24');
insert into ingredients (itemName, ingredient) values ('Item5', 'Ingredient45');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient15');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient50');
insert into ingredients (itemName, ingredient) values ('Item5', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient45');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient37');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient33');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient13');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient20');
insert into ingredients (itemName, ingredient) values ('Item20', 'Ingredient14');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient48');
insert into ingredients (itemName, ingredient) values ('Item28', 'Ingredient45');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient39');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient43');
insert into ingredients (itemName, ingredient) values ('Item12', 'Ingredient49');
insert into ingredients (itemName, ingredient) values ('Item20', 'Ingredient48');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient29');
insert into ingredients (itemName, ingredient) values ('Item10', 'Ingredient38');
insert into ingredients (itemName, ingredient) values ('Item25', 'Ingredient22');
insert into ingredients (itemName, ingredient) values ('Item5', 'Ingredient33');
insert into ingredients (itemName, ingredient) values ('Item2', 'Ingredient24');
insert into ingredients (itemName, ingredient) values ('Item9', 'Ingredient10');
insert into ingredients (itemName, ingredient) values ('Item28', 'Ingredient22');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient13');
```

```
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient39');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient25');
insert into ingredients (itemName, ingredient) values ('Item18', 'Ingredient21');
insert into ingredients (itemName, ingredient) values ('Item9', 'Ingredient40');
insert into ingredients (itemName, ingredient) values ('Item7', 'Ingredient28');
insert into ingredients (itemName, ingredient) values ('Item25', 'Ingredient47');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient43');
insert into ingredients (itemName, ingredient) values ('Item5', 'Ingredient11');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient19');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient25');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient10');
insert into ingredients (itemName, ingredient) values ('Item23', 'Ingredient39');
insert into ingredients (itemName, ingredient) values ('Item23', 'Ingredient41');
insert into ingredients (itemName, ingredient) values ('Item23', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item14', 'Ingredient6');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient11');
insert into ingredients (itemName, ingredient) values ('Item25', 'Ingredient15');
insert into ingredients (itemName, ingredient) values ('Item14', 'Ingredient24');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient12');
insert into ingredients (itemName, ingredient) values ('Item1', 'Ingredient50');
insert into ingredients (itemName, ingredient) values ('Item7', 'Ingredient27');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient38');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient14');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient21');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient48');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient44');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient38');
insert into ingredients (itemName, ingredient) values ('Item8', 'Ingredient44');
insert into ingredients (itemName, ingredient) values ('Item5', 'Ingredient2');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient16');
insert into ingredients (itemName, ingredient) values ('Item4', 'Ingredient7');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient33');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient25');
insert into ingredients (itemName, ingredient) values ('Item23', 'Ingredient26');
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient47');
insert into ingredients (itemName, ingredient) values ('Item27', 'Ingredient5');
insert into ingredients (itemName, ingredient) values ('Item26', 'Ingredient42');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient43');
insert into ingredients (itemName, ingredient) values ('Item12', 'Ingredient26');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient19');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient49');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient10');
insert into ingredients (itemName, ingredient) values ('Item16', 'Ingredient30');
insert into ingredients (itemName, ingredient) values ('Item4', 'Ingredient2');
insert into ingredients (itemName, ingredient) values ('Item30', 'Ingredient43');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient9');
insert into ingredients (itemName, ingredient) values ('Item22', 'Ingredient31');
insert into ingredients (itemName, ingredient) values ('Item2', 'Ingredient34');
insert into ingredients (itemName, ingredient) values ('Item24', 'Ingredient49');
insert into ingredients (itemName, ingredient) values ('Item2', 'Ingredient18');
```

```
insert into ingredients (itemName, ingredient) values ('Item13', 'Ingredient14');
insert into ingredients (itemName, ingredient) values ('Item27', 'Ingredient36');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient26');
insert into ingredients (itemName, ingredient) values ('Item25', 'Ingredient20');
insert into ingredients (itemName, ingredient) values ('Item10', 'Ingredient13');
insert into ingredients (itemName, ingredient) values ('Item11', 'Ingredient32');
insert into ingredients (itemName, ingredient) values ('Item15', 'Ingredient11');
insert into ingredients (itemName, ingredient) values ('Item19', 'Ingredient1');
insert into ingredients (itemName, ingredient) values ('Item29', 'Ingredient2');
insert into ingredients (itemName, ingredient) values ('Item3', 'Ingredient8');
insert into ingredients (itemName, ingredient) values ('Item21', 'Ingredient23');

--Transactions
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (1, 'Cash', 13,
5279);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (2, 'Credit Card',
9, 14214);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (3, 'Credit Card',
9, 17664);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (4, 'Cash', 13,
1180);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (5, 'Cash', 12,
368);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (6, 'Cash', 13,
15656);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (7, 'Cash', 13,
2659);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (8, 'Credit Card',
9, 17815);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (9, 'Cash', 10,
531);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (10, 'Credit
Card', 12, 1379);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (11, 'Cash', 10,
16970);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (12, 'Cash', 12,
14172);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (13, 'Credit
Card', 13, 6789);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (14, 'Cash', 13,
7314);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (15, 'Credit
Card', 11, 2799);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (16, 'Credit
Card', 12, 3725);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (17, 'Cash', 11,
846);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (18, 'Credit
Card', 10, 4947);
```

```
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (19, 'Credit Card', 10, 16994);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (20, 'Cash', 12, 16323);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (21, 'Credit Card', 9, 18121);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (22, 'Credit Card', 9, 755);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (23, 'Credit Card', 9, 14940);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (24, 'Credit Card', 13, 15806);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (25, 'Credit Card', 9, 14662);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (26, 'Credit Card', 12, 19063);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (27, 'Cash', 13, 17257);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (28, 'Credit Card', 9, 10129);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (29, 'Credit Card', 11, 8752);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (30, 'Credit Card', 13, 7296);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (31, 'Credit Card', 12, 17965);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (32, 'Cash', 11, 2533);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (33, 'Credit Card', 13, 13327);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (34, 'Credit Card', 11, 3225);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (35, 'Credit Card', 11, 5048);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (36, 'Cash', 10, 7299);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (37, 'Cash', 9, 10223);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (38, 'Cash', 13, 6391);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (39, 'Cash', 9, 13063);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (40, 'Credit Card', 10, 8229);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (41, 'Cash', 12, 14696);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (42, 'Credit Card', 11, 14851);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (43, 'Credit Card', 10, 12445);
```

```
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (44, 'Credit Card', 10, 8368);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (45, 'Credit Card', 13, 16793);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (46, 'Cash', 12, 9450);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (47, 'Cash', 13, 9705);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (48, 'Credit Card', 13, 3950);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (49, 'Cash', 13, 17541);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (50, 'Cash', 9, 13514);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (51, 'Cash', 12, 574);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (52, 'Credit Card', 9, 14519);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (53, 'Credit Card', 11, 15237);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (54, 'Cash', 10, 19241);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (55, 'Cash', 11, 14934);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (56, 'Cash', 12, 12730);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (57, 'Credit Card', 9, 12745);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (58, 'Cash', 11, 3693);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (59, 'Credit Card', 11, 7177);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (60, 'Credit Card', 12, 2585);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (61, 'Credit Card', 12, 13645);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (62, 'Cash', 13, 658);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (63, 'Cash', 13, 6384);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (64, 'Cash', 12, 2337);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (65, 'Cash', 12, 115);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (66, 'Credit Card', 13, 15948);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (67, 'Cash', 11, 16818);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (68, 'Credit Card', 11, 6194);
```

```
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (69, 'Credit Card', 11, 8711);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (70, 'Cash', 13, 15602);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (71, 'Credit Card', 9, 8152);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (72, 'Credit Card', 11, 14089);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (73, 'Cash', 10, 18975);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (74, 'Credit Card', 9, 6186);
insert into transactions (transactionId, paymentMethod, staffId, paymentQuantity) values (75, 'Cash', 13, 18203);

--Customers
insert into customers (customerId, name, partySize, phoneNumber) values (1, 'Anna-diane Heeks', 5, '1694957585');
insert into customers (customerId, name, partySize, phoneNumber) values (2, 'Midge Exposito', 10, '7571673619');
insert into customers (customerId, name, partySize, phoneNumber) values (3, 'Filippa Dener', 8, '0708835516');
insert into customers (customerId, name, partySize, phoneNumber) values (4, 'Corly Woodland', 2, '0088325819');
insert into customers (customerId, name, partySize, phoneNumber) values (5, 'Gertrude Simonett', 7, '2897281974');
insert into customers (customerId, name, partySize, phoneNumber) values (6, 'Alexa Ferneyhough', 7, '6687627464');
insert into customers (customerId, name, partySize, phoneNumber) values (7, 'Claresta Dunford', 9, '2086143031');
insert into customers (customerId, name, partySize, phoneNumber) values (8, 'Gennie Branscomb', 6, '0879381044');
insert into customers (customerId, name, partySize, phoneNumber) values (9, 'Coral Upshall', 6, '7890155380');
insert into customers (customerId, name, partySize, phoneNumber) values (10, 'Salvatore Copcutt', 1, '6334353008');
insert into customers (customerId, name, partySize, phoneNumber) values (11, 'Sherlock Meachan', 8, '3488779062');
insert into customers (customerId, name, partySize, phoneNumber) values (12, 'Nanny Maddison', 5, '7427708530');
insert into customers (customerId, name, partySize, phoneNumber) values (13, 'Forbes Alastair', 7, '0138448029');
insert into customers (customerId, name, partySize, phoneNumber) values (14, 'Almeda Furmage', 1, '3771109248');
insert into customers (customerId, name, partySize, phoneNumber) values (15, 'Dallas Ponceford', 2, '1359267239');
insert into customers (customerId, name, partySize, phoneNumber) values (16, 'Blair Trusse', 7, '9871772974');
insert into customers (customerId, name, partySize, phoneNumber) values (17, 'Stuart Broxis', 5, '8435134094');
```

```
insert into customers (customerId, name, partySize, phoneNumber) values (18, 'Hunt Haire', 3,
'9202500906');
insert into customers (customerId, name, partySize, phoneNumber) values (19, 'Hillary Edlyne', 7,
'7519090427');
insert into customers (customerId, name, partySize, phoneNumber) values (20, 'Cherey Uc', 5,
'9534451491');
insert into customers (customerId, name, partySize, phoneNumber) values (21, 'Cirstoforo Shankland', 3,
'0044564597');
insert into customers (customerId, name, partySize, phoneNumber) values (22, 'Orlando Drewett', 9,
'2790104776');
insert into customers (customerId, name, partySize, phoneNumber) values (23, 'Marylee Rubinsky', 2,
'6922648948');
insert into customers (customerId, name, partySize, phoneNumber) values (24, 'Theresita Phlipon', 9,
'1929752185');
insert into customers (customerId, name, partySize, phoneNumber) values (25, 'Brendis Yakobovitz', 4,
'3599747536');
insert into customers (customerId, name, partySize, phoneNumber) values (26, 'Shirline Maggiore', 2,
'9970085012');
insert into customers (customerId, name, partySize, phoneNumber) values (27, 'Teri Holehouse', 1,
'9175054808');
insert into customers (customerId, name, partySize, phoneNumber) values (28, 'Timi Tallent', 7,
'8834668816');
insert into customers (customerId, name, partySize, phoneNumber) values (29, 'Aridatha Noble', 8,
'4279464274');
insert into customers (customerId, name, partySize, phoneNumber) values (30, 'Brandon Beautyman', 2,
'5507843714');
insert into customers (customerId, name, partySize, phoneNumber) values (31, 'Tandi Halfhead', 5,
'6164188894');
insert into customers (customerId, name, partySize, phoneNumber) values (32, 'Davon Mosco', 4,
'8978758492');
insert into customers (customerId, name, partySize, phoneNumber) values (33, 'Everett Thurman', 3,
'1602812700');
insert into customers (customerId, name, partySize, phoneNumber) values (34, 'Arnuad Rose', 1,
'3836309234');
insert into customers (customerId, name, partySize, phoneNumber) values (35, 'Kalila Shemwell', 9,
'7633835247');
insert into customers (customerId, name, partySize, phoneNumber) values (36, 'Yorker Allcock', 2,
'2142993839');
insert into customers (customerId, name, partySize, phoneNumber) values (37, 'Stephan Prangle', 1,
'7182619249');
insert into customers (customerId, name, partySize, phoneNumber) values (38, 'Faulkner Coppledike', 6,
'4273554079');
insert into customers (customerId, name, partySize, phoneNumber) values (39, 'Cobbie Gilbride', 5,
'4484530862');
insert into customers (customerId, name, partySize, phoneNumber) values (40, 'Ingrid Mulhill', 9,
'9945084120');
insert into customers (customerId, name, partySize, phoneNumber) values (41, 'Kennith Nilges', 4,
'2663468710');
insert into customers (customerId, name, partySize, phoneNumber) values (42, 'Felike Hardistry', 7,
'9797968802');
```

```
insert into customers (customerId, name, partySize, phoneNumber) values (43, 'Allister Gozzard', 6,
'3402898049');
insert into customers (customerId, name, partySize, phoneNumber) values (44, 'Amity Eagland', 2,
'7296617769');
insert into customers (customerId, name, partySize, phoneNumber) values (45, 'Anthia Castelyn', 3,
'9906529508');
insert into customers (customerId, name, partySize, phoneNumber) values (46, 'Isidro Guinane', 4,
'5600450515');
insert into customers (customerId, name, partySize, phoneNumber) values (47, 'Tallie Dyzart', 8,
'5383042246');
insert into customers (customerId, name, partySize, phoneNumber) values (48, 'Norris Catonnet', 8,
'7262920019');
insert into customers (customerId, name, partySize, phoneNumber) values (49, 'Jeanelle Scollan', 9,
'2135182212');
insert into customers (customerId, name, partySize, phoneNumber) values (50, 'Bobbie Giddens', 2,
'4598060231');

--Reservations
insert into reservations (customerId, reservationId, time) values (1, 1, TO_DATE('2023-12-05 11:45:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (2, 3, TO_DATE('2023-12-09 19:30:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (3, 5, TO_DATE('2023-12-04 17:00:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (4, 7, TO_DATE('2023-12-04 19:30:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (5, 9, TO_DATE('2023-12-01 11:45:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (6, 11, TO_DATE('2023-12-01 12:45:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (7, 13, TO_DATE('2023-12-05 13:30:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (8, 15, TO_DATE('2023-12-02 18:15:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (9, 17, TO_DATE('2023-12-06 14:30:00',
'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (10, 19, TO_DATE('2023-12-03
18:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (11, 21, TO_DATE('2023-12-06
15:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (12, 23, TO_DATE('2023-12-08
12:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (13, 25, TO_DATE('2023-12-05
13:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (14, 27, TO_DATE('2023-12-05
20:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (15, 29, TO_DATE('2023-12-05
18:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (16, 31, TO_DATE('2023-12-08
10:45:00', 'yyyy-mm-dd hh24:mi:ss'));
```

```
insert into reservations (customerId, reservationId, time) values (17, 33, TO_DATE('2023-12-06
20:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (18, 35, TO_DATE('2023-12-05
13:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (19, 37, TO_DATE('2023-12-07
13:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (20, 39, TO_DATE('2023-12-09
16:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (21, 41, TO_DATE('2023-12-07
10:15:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (22, 43, TO_DATE('2023-12-09
20:15:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (23, 45, TO_DATE('2023-12-03
12:15:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (24, 47, TO_DATE('2023-12-01
15:15:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (25, 49, TO_DATE('2023-12-07
21:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (26, 51, TO_DATE('2023-12-09
14:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (27, 53, TO_DATE('2023-12-04
21:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (28, 55, TO_DATE('2023-12-08
12:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (29, 57, TO_DATE('2023-12-01
15:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (30, 59, TO_DATE('2023-12-01
15:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (31, 61, TO_DATE('2023-12-05
15:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (32, 63, TO_DATE('2023-12-02
16:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (33, 65, TO_DATE('2023-12-02
12:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (34, 67, TO_DATE('2023-12-04
17:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (35, 69, TO_DATE('2023-12-05
12:45:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (36, 71, TO_DATE('2023-12-05
15:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (37, 73, TO_DATE('2023-12-04
12:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (38, 75, TO_DATE('2023-12-03
13:15:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (39, 77, TO_DATE('2023-12-03
18:30:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into reservations (customerId, reservationId, time) values (40, 79, TO_DATE('2023-12-07
16:15:00', 'yyyy-mm-dd hh24:mi:ss'));

--Wait Queue
```

```
insert into waitQueue (customerId, position) values (41, 1);
insert into waitQueue (customerId, position) values (42, 2);
insert into waitQueue (customerId, position) values (43, 3);
insert into waitQueue (customerId, position) values (44, 4);
insert into waitQueue (customerId, position) values (45, 5);
insert into waitQueue (customerId, position) values (46, 6);
insert into waitQueue (customerId, position) values (47, 7);
insert into waitQueue (customerId, position) values (48, 8);
insert into waitQueue (customerId, position) values (49, 9);
insert into waitQueue (customerId, position) values (50, 10);
```