DATABASE SPECIFICATIONS

Next-Gen Restaurant Application (NRA)

Eric Brown – esb191@psu.edu

School of Graduate Professional Studies
Information Science Department
IN SC 521 - Introduction to Database Concepts

November, 2022

DOCUMENT CONTROL

Work carried out by:

Name	Email Address	Other
Eric Brown	**esb191@psu.edu**	

Revision Sheet

Release No.	Date	Revision Description
1	**9/4/2022**	First Submission of Milestone 1 – General Data Requirements
2	**9/12/2022**	First Revision – Simplified, and made more direct data requirements for Milestone 1
3	**9/18/2022**	First Submission of Milestone 2 – Conceptual Design
4	**9/30/2022**	Second Revision – Addressed weak to weak entity relationship issue between stakeholder and customer entities. Also, clarified what notation I used for my ERD conceptual diagram (Chen Notation)
5	**10/2/2022**	First Submission of Milestone 3 – Logical Design
6	**10/15/2022**	Third Revision – Addressed the translation of restaurant hosts check customer waiting queue relationship (one-to many relationship)
7	**10/16/2022**	First Submission of Milestone 4 - Normalization
8	**10/30/2022**	First Submission of Milestone 5 – Physical Design
9	**11/11/2022**	Third Revision – Addressed the missing Physical Design Diagram issue & updated record count values of each table
10	**11/13/2022**	First Submission of Milestone 6 – SQL Queries

DATABASE SPECIFICATIONS

TABLE OF CONTENTS

MILESTONE 1: DATA REQUIREMENTS

System Name or Title

Next-Gen Restaurant Application (NRA) - automation system to orchestrate customer seating and reservations in a restaurant environment

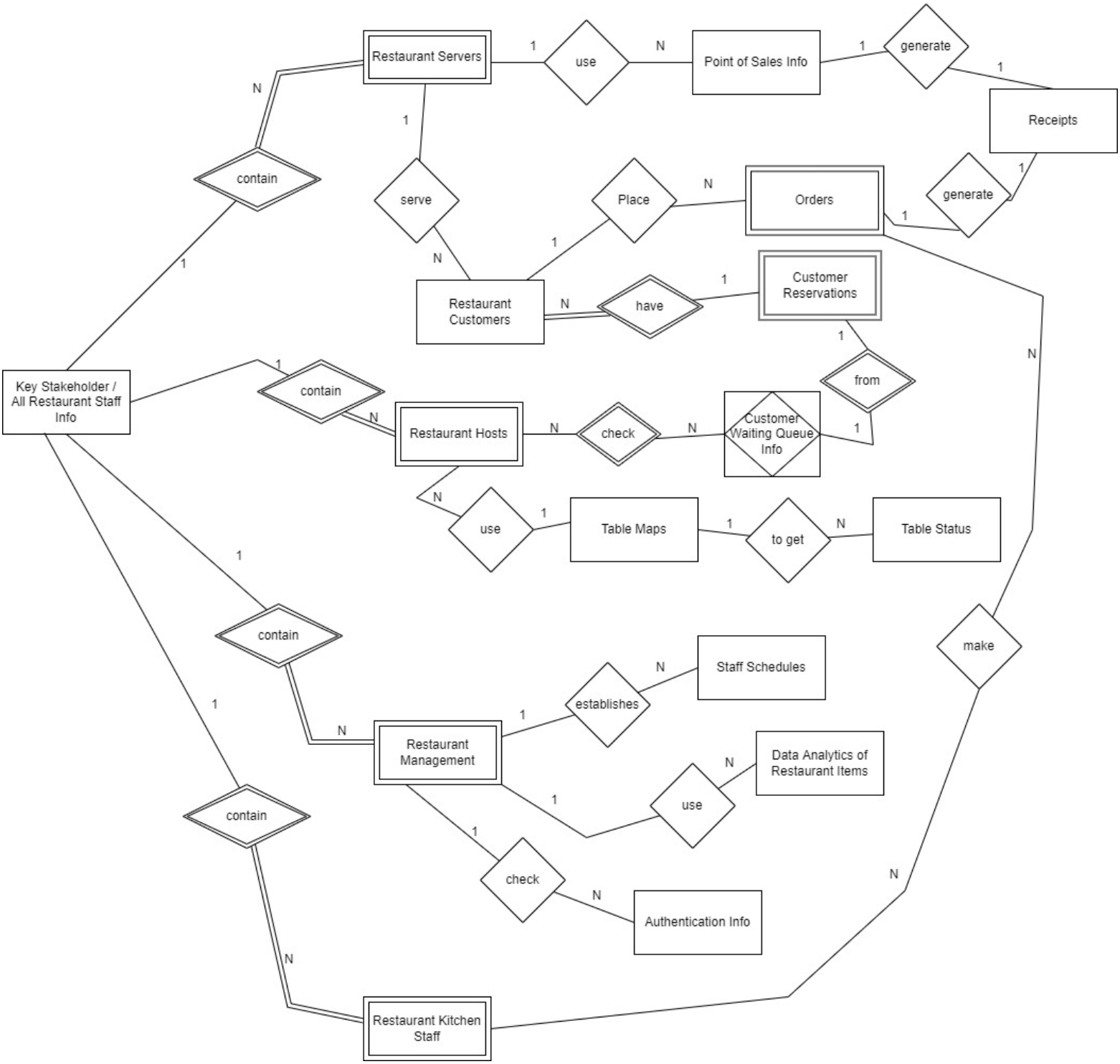
Core requirements

No.	Requirement	Referenced page in SRS	Referenced Section in SRS	Referenced Paragraph in Section
1	System should store customer reservations online and by phone	13 13	3.5.6 3.5.6	1 3
2	System should store staff scheduling information	6	2.2	6
3	System should store table status info – by managing queues of customers and sending out notifications to host/hostess and customers, when a customer's table is ready	13	3.5.6	6
4	System should store/retrieve documents such as receipts (both a customer copy and a store copy of each credit card signature slip shall be printed for every credit card order)	10	3.5.1	8
5	System should store data analytics of things such as menu items, ingredients, liquor, soft drinks, etc... (to anticipate traffic patterns and popularity of items on a menu that customers order frequently)	12 12 12	3.5.4 3.5.4 3.5.4	1 2 4
6	System should store table maps so the user can create a seating map that is reflective of the restaurant's table layout	11	3.5.2	1
7	System should store bar tabs of orders made by customers	11	3.5.1	10
8	The system should store age restriction parameters on certain orders, to prevent underage drinking by a customer (under 21) and under age serving /handling of alcoholic beverages by a staff member (under 21)	9	3.1	2
9	System should store point of sales system information, in terms of gratuities made by customers, as well as different forms of payment made by customers on	10	3.5.1	4

	orders (cash, credit cards - Visa, MasterCard, American Express, Discover, and gift cards)			
10	System should store authentication information (runtime errors, connectivity problems, performance issues, third party service error messages, and file system errors)	12	3.5.5	2
11	System should store managerial preauthorization approvals such as spending limits	11 11	3.5.1 3.5.1	11 12
12	The system should store customer information who are in the waiting queue (customer's name, party size, and telephone number)	11 12	3.5.3 3.5.3	2 3
13	The system should store table assignment data	12	3.5.3	6
14	The system should store key stakeholder information of all people who work in the restaurant (restaurant management, kitchen staff, hosts/hostesses, and servers)	7	2.3	3

MILESTONE 2: CONCEPTUAL DESIGN

CHEN Notation – ERD Diagram

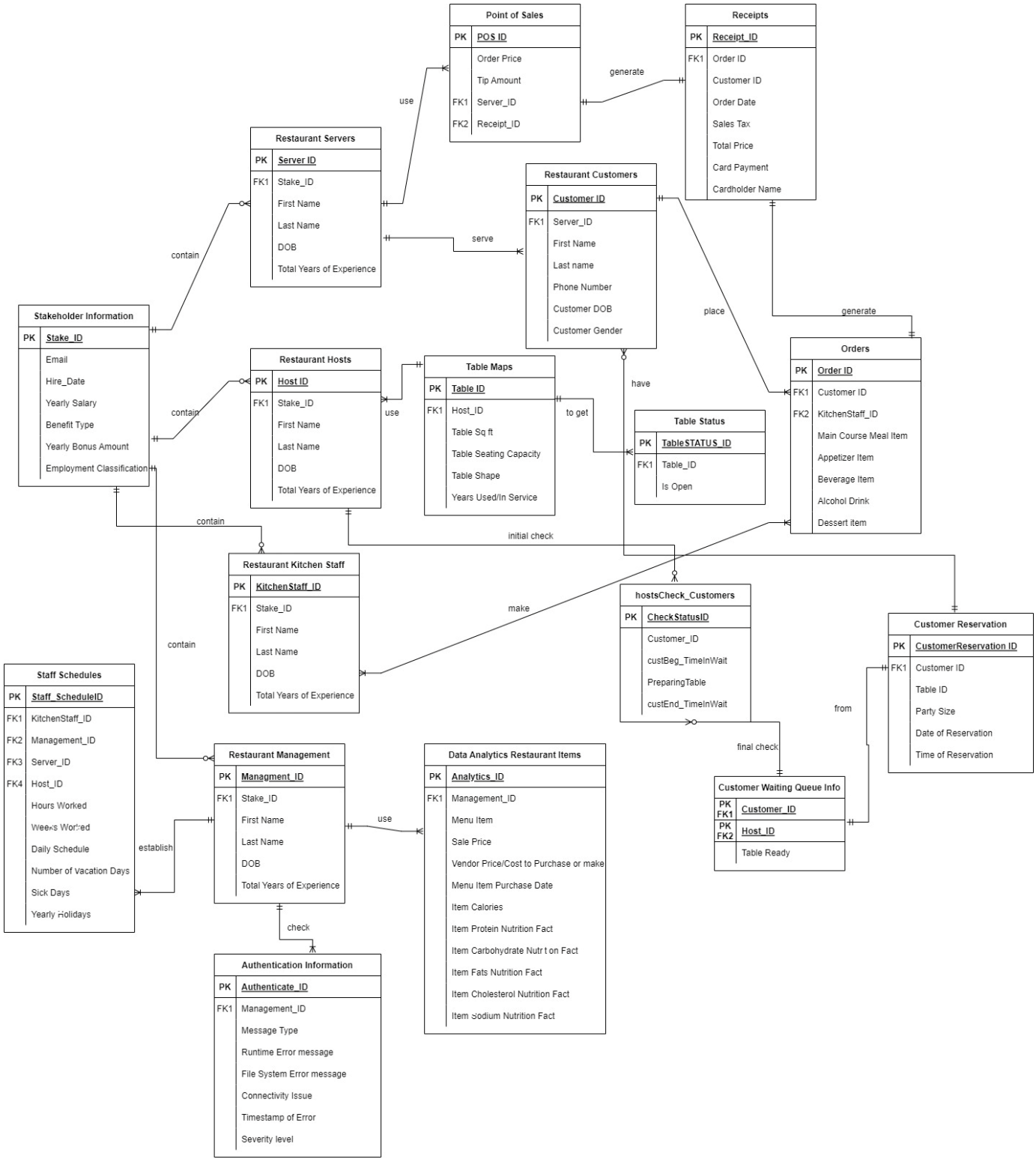


Assumptions and Constraints

- Bar tabs made by customers are always listed in the receipts of customer's total order
- Point of sales system stores age restriction parameters of certain items listed on the restaurant menu, which customers can order
- Point of sale system stores managerial preauthorization approval codes for things such as spending limits
- Table assignment information is always displayed on Table Maps of the restaurant

MILESTONE 3: LOGICAL DESIGN

ER Diagram – Crow Foot Notation



Entity name: Stakeholder Information

Attributes:

Stake_ID, Email, Hire_Date, Yearly Salary, Benefit Type, Yearly Bonus Amount, Employment Classification

Functional dependencies:

Stake_ID ⟶ Email, Hire_Date, Yearly Salary, Benefit Type, Yearly Bonus Amount, Employment Classification

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Stake_ID		Email, Hire_Date, Yearly Salary, Benefit Type, Yearly Bonus Amount, Employment Classification

Attribute closures (if any):

Stake_ID+= Email, Hire_Date, Yearly Salary, Benefit Type, Yearly Bonus Amount, Employment Classification

Unique keys: the key for this table is/are

Stake_ID

Entity name: Restaurant Servers

Attributes:

Server_ID, Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Functional dependencies:

Server_ID ⟶ Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Server_ID		Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attribute closures (if any):

Server_ID+= First Name, Last Name, DOB, Total Years of Experience

Unique keys: the key for this table is/are

Server_ID

Entity name: Point of Sales

Attributes:

POS_ID, Order Price, Tip Amount, Server_ID, Receipt_ID

Functional dependencies:

POS_ID ⟶ Order Price, Tip Amount, Server_ID, Receipt_ID

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	POS_ID		Order Price, Tip Amount, Server_ID, Receipt_ID

Attribute closures (if any):
POS_ID+= Order Price, Tip Amount, Server_ID, Receipt_ID

Unique keys: the key for this table is/are
POS_ID

Entity name: Receipts

Attributes:

Receipt_ID, Order_ID, Customer_ID, Order Date, Sales Tax, Total Price, Card Payment, Cardholder Name

Functional dependencies:

Receipt_ID ⟶ Order_ID, Customer_ID, Order Date, Sales Tax, Total Price, Card Payment, Cardholder Name

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Receipt_ID		Order_ID, Customer_ID, Order Date, Sales Tax, Total Price, Card Payment, Cardholder Name

Attribute closures (if any):
Receipt_ID+= Order_ID, Customer_ID, Order Date, Sales Tax, Total Price, Card Payment, Cardholder Name

Unique keys: the key for this table is/are
Receipt_ID

Entity name: Restaurant Customers

Attributes:

Customer_ID, Server_ID, First Name, Last Name, Phone Number, Customer DOB, Customer Gender

Functional dependencies:

Customer_ID ⟶ Server_ID, First Name, Last Name, Phone Number, Customer DOB, Customer Gender

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Customer_ID		Server_ID, First Name, Last Name, Phone Number, Customer DOB, Customer Gender

Attribute closures (if any):

Customer_ID+= Server_ID, First Name, Last Name, Phone Number, Customer DOB, Customer Gender

Unique keys: the key for this table is/are

Customer_ID

Entity name: Orders

Attributes:

Order_ID, Customer_ID, KitchenStaff_ID, Main Course Meal Item, Appetizer Item, Beverage Item, Alcohol Drink, Dessert Item

Functional dependencies:

Order_ID ⟶ Customer_ID, KitchenStaff_ID, Main Course Meal Item, Appetizer Item, Beverage Item, Alcohol Drink, Dessert Item

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Order_ID		Customer_ID, KitchenStaff_ID, Main Course Meal Item, Appetizer Item,

			Beverage Item, Alcohol Drink, Dessert Item

Attribute closures (if any):
Order_ID+= Customer_ID, KitchenStaff_ID, Main Course Meal Item, Appetizer Item, Beverage Item, Alcohol Drink, Dessert Item

Unique keys: the key for this table is/are
Order_ID

Entity name: Restaurant Hosts

Attributes:

Host_ID, Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Functional dependencies:

Host_ID ⟶ Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Host_ID		Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attribute closures (if any):
Host_ID+= Stake_ID, First Name, Last Name, DOB, Total Years of Experience
Unique keys: the key for this table is/are
Host_ID

Entity name: Table Maps

Attributes:

Table_ID, Table Sq ft, Table Seating Capacity, Table Shape, Years Used, In Service

Functional dependencies:

Table_ID ⟶ Table Sq ft, Table Seating Capacity, Table Shape, Years Used, In Service

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side

	Table_ID		Table Sq ft, Table Seating Capacity, Table Shape, Years Used, In Service

Attribute closures (if any):
Table_ID+= Table Sq ft, Table Seating Capacity, Table Shape, Years Used, In Service

Unique keys: the key for this table is/are
Table_ID

Entity name: Table Status

Attributes:

TableSTATUS_ID, Table_ID, Is Open

Functional dependencies:

TableSTATUS_ID ⟶ Table_ID, Is Open

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	TableSTATUS_ID		Table_ID, Is Open

Attribute closures (if any):
TableSTATUS_ID+= Table_ID, Is Open

Unique keys: the key for this table is/are
TableSTATUS_ID

Entity name: Restaurant Kitchen Staff

Attributes:

KitchenStaff_ID, Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Functional dependencies:

KitchenStaff_ID ⟶ Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side

	KitchenStaff_ID		Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attribute closures (if any):
KitchenStaff_ID+= Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Unique keys: the key for this table is/are
KitchenStaff_ID

Entity name: Customer Waiting Queue Info

Attributes:

Customer_ID, Host_ID, Table Ready

Functional dependencies:

Customer_ID ⟶ Host_ID, Table Ready

Host_ID ⟶ Customer_ID, Table Ready

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
		Customer_ID, Host_ID	Table Ready

Attribute closures (if any):
Customer_ID+= Host_ID, Table Ready
Host_ID+=Customer_ID, Table Ready
(Customer_ID, Host_ID) is a super key

Unique keys: the key for this table is/are
Customer_ID, Host_ID

Entity name: Customer Reservation

Attributes:

CustomerReservation_ID, Customer_ID, Table_ID, Party Size, Date of Reservation, Time of Reservation

Functional dependencies:

CustomerReservation_ID ⟶ Customer_ID, Table_ID, Party Size, Date of Reservation, Time of Reservation

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
		CustomerReservation_ID	Customer_ID, Table_ID, Party Size, Date of Reservation, Time of Reservation

Attribute closures (if any):

CustomerReservation_ID+= Customer_ID, Table_ID, Party Size, Date of Reservation, Time of Reservation

Unique keys: the key for this table is/are

CustomerReservation_ID

Entity name: Restaurant Management

Attributes:

Management_ID, Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Functional dependencies:

Management_ID ⟶ Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Management_ID		Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Attribute closures (if any):

Management_ID+= Stake_ID, First Name, Last Name, DOB, Total Years of Experience

Unique keys: the key for this table is/are

Management_ID

Entity name: Data Analytics Restaurant Items

Attributes:

Analytics_ID, Management_ID, Menu Item, Sale Price, Vendor Price/Cost to Purchase or make, Menu Item Purchase Date, Item Calories, Item Protein Nutrition Fact, Item Carbohydrate Nutrition Fact, Item Fats Nutrition Fact, Item Cholesterol Nutrition Fact, Item Sodium Nutrition Fact

Functional dependencies:

Analytics_ID ⟶ Management_ID, Menu Item, Sale Price, Vendor Price/Cost to Purchase or make, Menu Item Purchase Date, Item Calories, Item Protein Nutrition Fact, Item Carbohydrate Nutrition Fact, Item Fats Nutrition Fact, Item Cholesterol Nutrition Fact, Item Sodium Nutrition Fact

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
		Analytics_ID	Management_ID, Menu Item, Sale Price, Vendor Price/Cost to Purchase or make, Menu Item Purchase Date, Item Calories, Item Protein Nutrition Fact, Item Carbohydrate Nutrition Fact, Item Fats Nutrition Fact, Item Cholesterol Nutrition Fact, Item Sodium Nutrition Fact

Attribute closures (if any):

Analytics_ID+= Management_ID, Menu Item, Sale Price, Vendor Price/Cost to Purchase or make, Menu Item Purchase Date, Item Calories, Item Protein Nutrition Fact, Item Carbohydrate Nutrition Fact, Item Fats Nutrition Fact, Item Cholesterol Nutrition Fact, Item Sodium Nutrition Fact

Unique keys: the key for this table is/are

Analytics_ID

Entity name: Authentication Information

Attributes:

Authenticate_ID, Management_ID, Message Type, Runtime Error Message, File System Error Message, Connectivity Issue, Timestamp of error, Severity level

Functional dependencies:

Authenticate_ID ⟶ Management_ID, Message Type, Runtime Error Message, File System Error Message, Connectivity Issue, Timestamp of error, Severity level

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side

		Authenticate_ID	Management_ID, Message Type, Runtime Error Message, File System Error Message, Connectivity Issue, Timestamp of error, Severity level

Attribute closures (if any):

Authenticate_ID+= Management_ID, Message Type, Runtime Error Message, File System Error Message, Connectivity Issue, Timestamp of error, Severity level

Unique keys: the key for this table is/are

Authenticate_ID

Entity name: Staff Schedules

Attributes:

Staff_ScheduleID, Management_ID, Server_ID, Host_ID, Hours Worked, Weeks Worked, Daily Schedule, Number of Vacation Days, Sick Days, Yearly Holidays

Functional dependencies:

Staff_ScheduleID ⟶ Management_ID, Server_ID, Host_ID, Hours Worked, Weeks Worked, Daily Schedule, Number of Vacation Days, Sick Days, Yearly Holidays

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
		Staff_ScheduleID	Management_ID, Server_ID, Host_ID, Hours Worked, Weeks Worked, Daily Schedule, Number of Vacation Days, Sick Days, Yearly Holidays

Attribute closures (if any):

Staff_ScheduleID+= Management_ID, Server_ID, Host_ID, Hours Worked, Weeks Worked, Daily Schedule, Number of Vacation Days, Sick Days, Yearly Holidays

Unique keys: the key for this table is/are

Staff_ScheduleID

Entity name: hostsCheck_Customers

Attributes:

CheckStatusID, custBeg_TimeInWait, PreparingTable, custEnd_TimeInWait

Functional dependencies:

CheckStatusID ⟶ custBeg_TimeInWait, PreparingTable, custEnd_TimeInWait

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	CheckStatusID		custBeg_TimeInWait, PreparingTable, custEnd_TimeInWait

Attribute closures (if any):

CheckStatusID+= custBeg_TimeInWait, PreparingTable, custEnd_TimeInWait

Unique keys: the key for this table is/are
CheckStatusID

Assumptions and Constraints

- Bar tabs made by customers are always listed in the receipts of customer's total order
- Point of sales system stores age restriction parameters of certain items listed on the restaurant menu, which customers can order
- Point of sale system stores managerial preauthorization approval codes for things such as spending limits
- Table assignment information is always displayed on Table Maps of the restaurant
- Free form objects are always displayed on Table Maps of the restaurant
- Text messages are sent by Restaurant Hosts to customers once a table is ready for a customer

MILESTONE 4: NORMALIZATION

Tables

	Name of the table	*Staff Schedules*			
	Description	Daily schedule report of all stakeholders who work at the restaurant. Which include the kitchen staff, management staff, server staff, and host staff.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Staff_ScheduleID	ID of staff member's schedule	integer	Between 1 and 999999999	
	KitchenStaff_ID	ID of a kitchen staff member	integer	Between 1 and 999999999	
	Management_ID	ID of a management staff member	integer	Between 1 and 999999999	
	Server_ID	ID of a server staff member	integer	Between 1 and 999999999	
	Host_ID	ID of a host staff member	integer	Between 1 and 999999999	
	Hours Worked	Weekly number of hours worked by a given staff member of the restaurant	integer	Between 0 and 100	Can not be negative
	Weeks Worked	Number of weeks worked by a given staff member of the restaurant in a month	integer	Between 0 and 5	Can not be negative
	Daily Schedule	Days of the week a given staff member works at the restaurant	text	Mon, Tues, Wed	A set of abbr. must be determined
	Number of Vacation Days	Number of vacation days given to each employee of the restaurant established by upper management	integer	Between 0 and 21	Can not be negative
	Sick Days	Number of sick days given to each employee of the restaurant established by upper management	integer	Between 1 and 14	Can not be negative or zero

	Yearly Holidays	The established number of holidays the restaurant will be closed on during the year	integer	Between 1 and 12	Can not be negative or zero
	Functional Dependencies and Keys				
	Functional dependencies	Staff_ScheduleID -> KitchenStaff_ID Staff_ScheduleID -> Management_ID Staff_ScheduleID -> Server_ID Staff_ScheduleID -> Host_ID Staff_ScheduleID -> Hours Worked Staff_ScheduleID -> Weeks Worked Staff_ScheduleID -> Daily Schedule Staff_ScheduleID -> Number of Vacation Days Staff_ScheduleID -> Sick Days Staff_ScheduleID -> Yearly Holidays			
	Candidate keys	**Staff_ScheduleID**			
	Normalization				
	1NF	**Yes**	All cells contain a unique value		
	2NF	**Yes**	The key of the table is a single attribute		
	3NF	**Yes**	All the non-key attributes depend only on a key		
	BCNF	**Yes**	All the attributes depend only on a key		

	Name of the table	***Restaurant Management***			
	Description	Credentials of staff members who qualified to be included in the management staff of the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Management_ID	ID of a management staff member	integer	Between 1 and 999999999	
	Stake_ID	ID of stakeholders of the restaurant	integer	Between 1 and 999999999	
	First Name	First Name of a given restaurant management staff member	varchar(60)	John	Can not be null
	Last Name	Last Name of a given restaurant management staff member	Varchar(60)	Doe	Can not be null
	DOB	Date of Birth of a restaurant staff member	Date	1/1/1996	Must be bigger than 1/1/1900

	Total Years of Experience	The total number of years worked at the restaurant or at another job with relevant skills pertaining to the restaurant	integer	Between 1 and 90	Can not be zero or negative
	Functional Dependencies and Keys				
	Functional dependencies	Management_ID -> Stake_ID Management_ID -> First Name Management_ID -> Last Name Management_ID -> DOB Management_ID -> Total Years of Experience			
	Candidate keys	**Management_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Analytics Restaurant Items***			
	Description	Analyzation of customer ordering habits in order to determine which items are popular and which ones are not.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Analytics_ID	ID for each analytic task needed by management	integer	Between 1 and 999999999	
	Management_ID	ID of a management staff member	integer	Between 1 and 999999999	
	Menu Item	Popular menu items purchased by customers	text	French fries, burgers, pasta	One item per entry
	Sale Price	The price which the restaurant is selling a given item to customers	decimal	4.59	Can not be negative
	Vendor Price/Cost to Purchase or make	The actual price to purchase the item from a vendor	decimal	5.99	Can not be negative
	Menu Item Purchase Date	The date in which the item was purchased by the restaurant	date	1/14/2000	Must be bigger than 1/1/1900
	Item Calories	The total amount of calories a given item contains	decimal	49.0 or 55.5	

	Item Protein Nutrition Fact	The total amount of protein listed by the nutrition facts of an item	decimal	49.0 or 55.5	
	Item Carb Nutrition Fact	The total amount of carbohydrates listed by the nutrition facts of an item	decimal	49.0 or 55.5	
	Item Cholesterol Nutrition Fact	The total amount of cholesterol listed by the nutrition facts of an item	decimal	49.0 or 55.5	
	Item Sodium Nutrition Fact	The total amount of sodium listed by the nutrition facts of an item	decimal	49.0 or 55.5	
	Functional Dependencies and Keys				
	Functional dependencies	Analytics_ID -> Management_ID Analytics_ID -> Menu Item Analytics_ID -> Sales Price Analytics_ID -> Vendor Price/Cost to Purchase or make Analytics_ID -> Menu Item Purchase Date Analytics_ID -> Item Calories Analytics_ID -> Item Protein Nutrition Fact Analytics_ID -> Item Carbohydrate Nutrition Fact Analytics_ID -> Item Cholesterol Nutrition Fact Analytics_ID -> Item Sodium Nutrition Fact			
	Candidate keys	**Analytics_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Authentication Information***			
	Description	Log of software system errors and failures that happen throughout the day when the restaurant is open for business.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Authenicate_ID	ID for each authentication message that appears in the system	integer	Between 1 and 999999999	
	Management_ID	ID of a management staff member	integer	Between 1 and 999999999	
	Message Type	Each authentication error has a predetermined error	text	Control flow errors, error handling error,	Can not be null

		message type such as file not found, access denied, low disk space		calculation errors, functionality errors, communication errors	
	Runtime Error message	Specific errors that occur while the software system is running once successfully compiled	text	logical errors, Input/Output errors, undefined object errors, division by zero errors	Can be null
	File System Error message	Disk-related errors that may be due to corrupt files, bad sectors, and/or disk integrity corruption	text	Invalid format, Device capabilities error, Unable to allocate memory for file operation, Max number of available files exceeded	Can be null
	Connectivity Issue	Network related errors that happen when the system loses connection to local network servers	text	DNS resolution errors, TCP connection timeout/error, conflict with IP address	Can be null
	Timestamp of Error	Date and time of when a given error/issue occurs during the operational hours of the restaurant	Date/Time	12/2/2000/ 12:00	Can not be null
	Severity level	The level of urgency of a given type of error that occurs when the software system is in use	integer	1 to 10	A numeric scale must be determined, Can not be null
	Functional Dependencies and Keys				
	Functional dependencies	Authenicate_ID -> Management_ID Authenicate_ID -> Message Type Authenicate_ID -> Runtime Error message Authenicate_ID -> File System Error message Authenicate_ID -> Connectivity Issue Authenicate_ID -> Timestamp of Error Authenicate_ID -> Severity level			
	Candidate keys	**Authenicate_ID**			

	Normalization		
	1NF	Yes	All cells contain a unique value
	2NF	Yes	The key of the table is a single attribute
	3NF	Yes	All the non-key attributes depend only on a key
	BCNF	Yes	All the attributes depend only on a key

	Name of the table	***Restaurant Kitchen Staff***			
	Description	Credentials of staff members who qualified to be included in the kitchen staff of the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	KitchenStaff_ID	ID of a kitchen staff member	integer	Between 1 and 999999999	
	Stake_ID	ID of stakeholders of the restaurant	integer	Between 1 and 999999999	
	First Name	First Name of a given kitchen staff member	varchar(60)	John	Can not be null
	Last Name	Last Name of a given kitchen staff member	varchar(60)	Doe	Can not be null
	DOB	Date of Birth of a kitchen staff member	Date	1/1/1996	Must be bigger than 1/1/1900
	Total Years of Experience	The total number of years worked at the restaurant or at another job with relevant skills pertaining to the restaurant	integer	Between 1 and 90	Must be greater than 0, not a negative number
	Functional Dependencies and Keys				
	Functional dependencies	KitchenStaff_ID -> Stake_ID KitchenStaff_ID -> First Name KitchenStaff_ID -> Last Name KitchenStaff_ID -> DOB KitchenStaff_ID -> Total Years of Experience			
	Candidate keys	**KitchenStaff_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***HostsCheck_Customers***
	Description	Information regarding when a customer enters into the waiting queue of the restaurant. As well as when they leave the waiting queue to be seated in the restaurant.

	Attribute	Description	Type	Examples of values	Notes
	CheckStatusID	ID of each occurrence of when a customer enters the waiting queue to be seated	integer	Between 1 and 999999999	
	Customer_ID	ID of a restaurant customer	integer	Between 1 and 999999999	
	custBeg_TimeinWait	Time and date of when a customer enters the waiting queue	Date/Time	12/2/2000/ 12:00	
	PreparingTable	Binary answer of whether a table is being prepared for a given customer to seat at in the restaurant	char(4)	Yes or No	Has to be predetermined Can not be null
	custEnd_TimeinWait	Time and date of when a customer exits the waiting queue	Date/time	12/2/2000/ 12:00	
	Functional Dependencies and Keys				
	Functional dependencies	CheckStatusID -> Customer_ID CheckStatusID -> custBeg_TimeinWait CheckStatusID -> PreparingTable CheckStatusID -> custEnd_TimeinWait			
	Candidate keys	**CheckStatusID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Customer Waiting Queue Info***			
	Description	Information regarding whether a table is ready for a customer that entered into the waiting queue to be seated at the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Customer_ID	ID of a restaurant customer	integer	Between 1 and 999999999	
	Host_ID	ID of a restaurant host	integer	Between 1 and 999999999	
	Table Ready	Status of whether a table is ready for the customer to sit down	Char(4)	Yes or No	Has to be predetermined Can not be null

		at the restaurant to be served			
	Functional Dependencies and Keys				
	Functional dependencies	Customer_ID -> Host_ID Customer_ID -> Table Ready			
	Candidate keys	**Customer_ID, Host_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Customer Reservation***			
	Description	Information regarding the seating arrangements made by the customer which the restaurant will try to accommodate.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	CustomerReservation_ID	ID of each occurrence of when a customer makes a reservation online or by phone	integer	Between 1 and 999999999	
	Customer_ID	ID of a restaurant customer	integer	Between 1 and 999999999	
	Table_ID	ID of the table number	integer	Between 1 and 9999	
	Party Size	The number people that will be seated in the reservation made by the customer	integer	Between 1 and 12	Can not be a negative number
	Date of Reservation	Date of the reservation made by the customer	Date	12/2/2000	
	Time of Reservation	Time of the reservation made by the customer	Time	12:00	
	Functional Dependencies and Keys				
	Functional dependencies	CustomerReservation_ID -> Customer_ID CustomerReservation_ID -> Table_ID CustomerReservation_ID -> Party Size CustomerReservation_ID -> Date of Reservation CustomerReservation_ID -> Time of Reservation			
	Candidate keys	**CustomerReservation_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		

BCNF	Yes	All the attributes depend only on a key	

	Name of the table	***Point of Sales***			
	Description	Information generated from orders placed by customers. Mainly with regards to the cost of each order and what information will be displayed on the receipt of each order.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	POS_ID	ID of a point of sales information	integer	Between 1 and 999999999	
	Order Price	Total price of each order placed by customers	decimal	14.50	Can not be null or negative
	Tip Amount	Tip amount generate based on order amount	decimal	0.03	Can not be negative
	Server_ID	ID of a restaurant server	integer	Between 1 and 999999999	
	Receipt_ID	ID of a order receipt	integer	Between 1 and 999999999	
	Functional Dependencies and Keys				
	Functional dependencies	POS_ID -> Order Price POS_ID -> Tip Amount POS_ID -> Server_ID POS_ID -> Receipt_ID			
	Candidate keys	**POS_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Receipts***			
	Description	Information generated from POS system based on customer's orders. Acknowledging that payment was made from a customer and received by the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Receipt_ID	ID of each given receipt by the restaurant	integer	Between 1 and 999999999	
	Customer_ID	ID of a customer	integer	Between 1 and 999999999	
	Order Date	Date of order placed by a customer	Date/Time	10/14/2010 12:00	

	Sales Tax	Tax paid to a governing body for the sales of goods and services provided by the restaurant	decimal	0.10	
	Total Price	The total price of each order placed by a customer	decimal	40.00	
	Card Payment	The type of card payment made by a customer	text	Visa, Mastercard, cash	Can not be null Card text must be determined
	Cardholder Name	The name of the cardholder who made the payment of the order	text	John	
	Functional Dependencies and Keys				
	Functional dependencies	Receipt_ID -> Customer_ID Receipt_ID -> Order Date Receipt_ID -> Sales Tax Receipt_ID -> Total Price Receipt_ID -> Card Payment Receipt_ID -> Cardholder Name			
	Candidate keys	**Receipt_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Orders***			
	Description	All types of meals, beverages, soft drinks, appetizers placed by customers in their orders.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Order_ID	ID of each order placed by customers	integer	Between 1 and 999999999	
	Customer_ID	ID of a customer	integer	Between 1 and 999999999	
	KitchenStaff_ID	ID of a kitchen staff member	integer	Between 1 and 999999999	
	Main Course Meal Item	The main meal of an order placed by a customer	text	Salmon, steak, chicken alfredo, pasta bolognese	Can not be null
	Appetizer Item	The appetizer of an order placed by a customer	text	Calamari, chicken wings, curly fries	

	Beverage Item	The beverage of an order placed by a customer	text	Coke, pepsi, sweet tea, water	
	Alcohol Drink	The alcoholic drink of an order placed by a customer	text	Margarita, Cosmopolitan, Daiquiri, Manhattan	
	Dessert Item	The dessert of an order placed by a customer	text	Ice cream cake slice, cookies, key lime pie	
	Functional Dependencies and Keys				
	Functional dependencies	Order_ID -> Customer_ID Order_ID -> KitchenStaff_ID Order_ID -> Main Course Meal Item Order_ID -> Appetizer Item Order_ID -> Beverage Item Order_ID -> Alcohol Drink Order_ID -> Dessert Item			
	Candidate keys	**Order_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Table Maps***			
	Description	Information regarding the table layout of the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Table_ID	ID of each table used in the restaurant	integer	Between 1 and 999999999	
	Host_ID	ID of each restaurant host	integer	Between 1 and 999999999	
	Table Sq ft	The amount of square feet each table takes up in the restaurant	float	8, 8.6	Measured in feet
	Table Seating Capacity	The seating capacity of each table used in the restaurant	integer	1,6, 12	Maximum seating capacity of each table is 12
	Table Shape	The table's overall shape	text	Square, rectangular, circle, oval	Must be determined
	Years Used/In Service	The number of years the table was used in the restaurant	integer	1, 10, 25	Can not be negative
	Functional Dependencies and Keys				

	Functional dependencies	Table_ID -> Host_ID Table_ID -> Table Sq ft Table_ID -> Table Seating Capacity Table_ID -> Table Shape Table_ID -> Table Shape Table_ID -> Years Used/In Service	
	Candidate keys	**Table_ID**	
	Normalization		
	1NF	Yes	All cells contain a unique value
	2NF	Yes	The key of the table is a single attribute
	3NF	Yes	All the non-key attributes depend only on a key
	BCNF	Yes	All the attributes depend only on a key

	Name of the table	***Table Status***			
	Description	Information regarding whether a given table is available for use and not currently occupied by customers.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	TableSTATUS_ ID	ID of each table availability status	integer	Between 1 and 999999999	
	Table_ID	ID of a table in the restaurant	integer	Between 1 and 999999999	
	Is Open	Status of whether a table is occupied or not by customers	Char(4)	Yes or no	Can not be null
	Functional Dependencies and Keys				
	Functional dependencies	TableSTATUS_ID -> Table_ID TableSTATUS_ID -> Is Open			
	Candidate keys	**TableSTATUS_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Stakeholder Information***			
	Description	General information pertaining to each type of staff member that works at the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Stake_ID	ID of every stakeholder in the restaurant	integer	Between 1 and 999999999	

	Email	Email address of each restaurant employee	text	johndoe@yahoo.com	
	Hire_Date	Date of hire of each employee of the restaurant	Date	10/14/2018	Can not be null
	Yearly Salary	The yearly salary of each restaurant employee	float	45000.00	
	Benefit Type	The different type of benefits offered and accepted by each employee	text	medical, disability, life insurance, retirement benefits, paid time off	Must be determined
	Yearly Bonus Amount	The yearly bonuses received by each restaurant employee	float	5000.00	
	Employment Classification	The type of employment contract each employee is placed on	text	Part-time, full-time, flex hours	Must be determined
	Functional Dependencies and Keys				
	Functional dependencies	Stake_ID -> Email Stake_ID -> Hire_Date Stake_ID -> Yearly Salary Stake_ID -> Benefit Type Stake_ID -> Yearly Bonus Amount Stake_ID -> Employment Classification			
	Candidate keys	**Stake_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Restaurant Customers***			
	Description	Information regarding customers of the restaurant. Which will make the hosts and servers job easier to deliver quality service to the customers.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Customer_ID	ID of a customer	integer	Between 1 and 999999999	
	Server_ID	ID of a server			
	First Name	First Name of a given customer	varchar(60)	John	Can not be null
	Last Name	Last Name of a given customer	varchar(60)	Doe	Can not be null

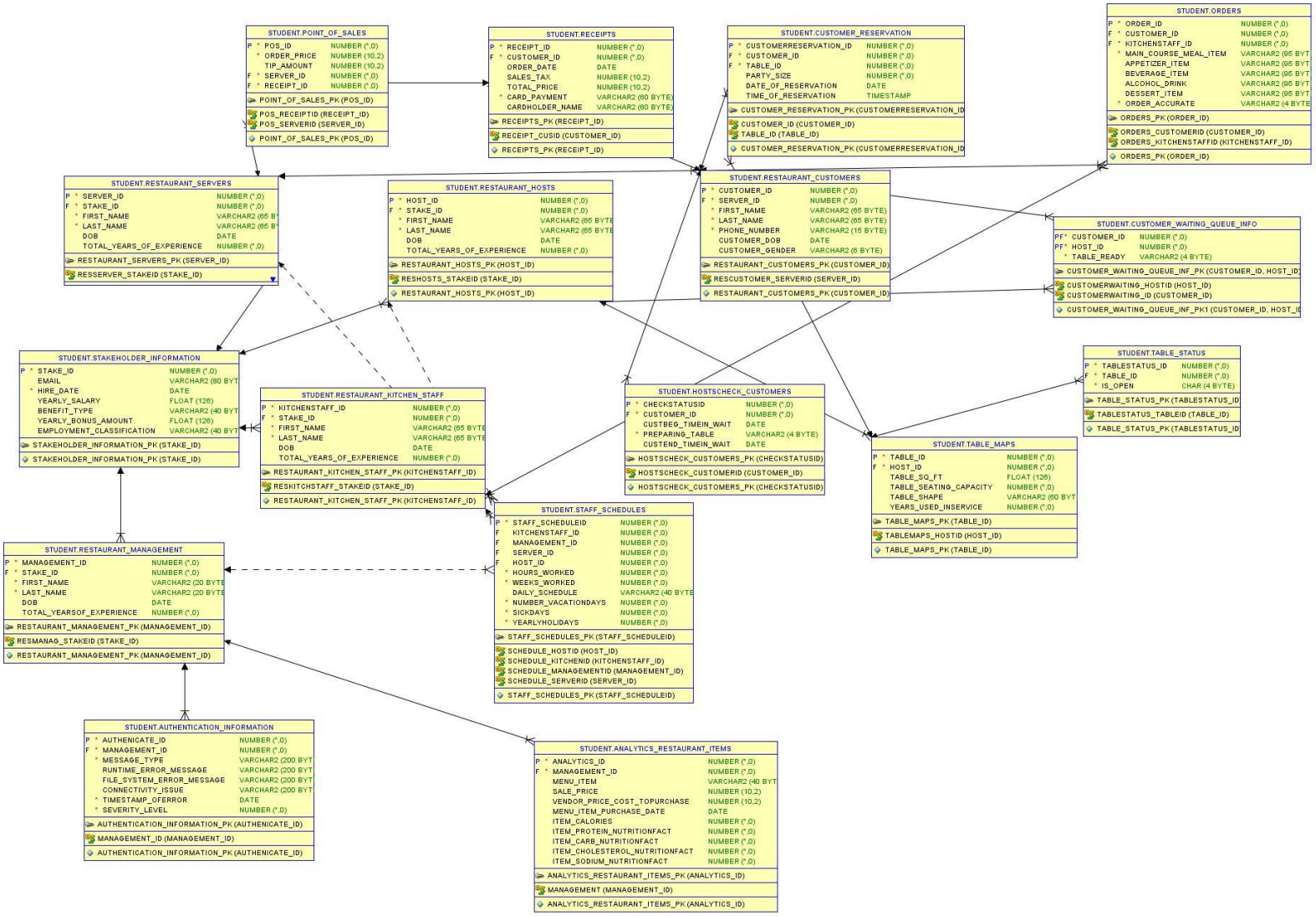
	Phone Number	Telephone number of a given customer	text	917-543-2488	Can not be null
	Customer DOB	Date of Birth of a customer	Date	1/1/1996	Must be bigger than 1/1/1900
	Customer Gender	The gender type of a given customer	Varchar(6)	Male, Female, M, F	Must be determined
	Functional Dependencies and Keys				
	Functional dependencies	Customer_ID -> Server_ID Customer_ID -> First Name Customer_ID -> Last Name Customer_ID -> Phone Number Customer_ID -> Customer DOB Customer_ID -> Customer Gender			
	Candidate keys	**Customer_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

	Name of the table	***Restaurant Servers***			
	Description	Credentials of staff members who qualified to be included in the server staff of the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Server_ID	ID of a restaurant server	integer	Between 1 and 999999999	
	Stake_ID	ID of stakeholders of the restaurant	integer	Between 1 and 999999999	
	First Name	First Name of a given restaurant server	varchar(60)	John	Can not be null
	Last Name	Last Name of a given restaurant server	varchar(60)	Doe	Can not be null
	DOB	Date of Birth of a restaurant server	Date	1/1/1996	Must be bigger than 1/1/1900
	Total Years of Experience	The total number of years worked at the restaurant or at another job with relevant skills pertaining to the restaurant	integer	Between 1 and 90	Must be greater than 0, not a negative number
	Functional Dependencies and Keys				
	Functional dependencies	Server_ID -> Stake_ID Server_ID -> First Name Server_ID -> Last Name Server_ID -> DOB Server_ID -> Total Years of Experience			

	Candidate keys	**Server_ID**	
	Normalization		
	1NF	Yes	All cells contain a unique value
	2NF	Yes	The key of the table is a single attribute
	3NF	Yes	All the non-key attributes depend only on a key
	BCNF	Yes	All the attributes depend only on a key

	Name of the table	***Restaurant Hosts***			
	Description	Credentials of staff members who qualified to be included in the host staff of the restaurant.			
	Attribute	**Description**	**Type**	**Examples of values**	**Notes**
	Host_ID	ID of each restaurant host	integer	Between 1 and 999999999	
	Stake_ID	ID of stakeholders of the restaurant	integer	Between 1 and 999999999	
	First Name	First Name of a given restaurant host	varchar(60)	John	Can not be null
	Last Name	Last Name of a given restaurant host	varchar(60)	Doe	Can not be null
	DOB	Date of Birth of a restaurant host	Date	1/1/1996	Must be bigger than 1/1/1900
	Total Years of Experience	The total number of years worked at the restaurant or at another job with relevant skills pertaining to the restaurant	integer	Between 1 and 90	Must be greater than 0, not a negative number
	Functional Dependencies and Keys				
	Functional dependencies	Host_ID -> Stake_ID Host_ID -> First Name Host_ID -> Last Name Host_ID -> DOB Host_ID -> Total Years of Experience			
	Candidate keys	**Host_ID**			
	Normalization				
	1NF	Yes	All cells contain a unique value		
	2NF	Yes	The key of the table is a single attribute		
	3NF	Yes	All the non-key attributes depend only on a key		
	BCNF	Yes	All the attributes depend only on a key		

MILESTONE 5: PHYSICAL DESIGN



ANALYTICS_RESTAURANT_ITEMS	
Primary Key	ANALYTICS_ID
Foreign Keys	MANAGEMENT_ID
SQL Code	CREATE TABLE ANALYTICS_RESTAURANT_ITEMS (ANALYTICS_ID NUMBER(*, 0) NOT NULL , MANAGEMENT_ID NUMBER(*, 0) NOT NULL , MENU_ITEM VARCHAR2(40 BYTE) , SALE_PRICE NUMBER(*, 0) , VENDOR_PRICE_COST_TOPURCHASE NUMBER(*, 0) , MENU_ITEM_PURCHASE_DATE DATE , ITEM_CALORIES NUMBER(*, 0)

	, ITEM_PROTEIN_NUTRITIONFACT NUMBER(*, 0) , ITEM_CARB_NUTRITIONFACT NUMBER(*, 0) , ITEM_CHOLESTEROL_NUTRITIONFACT NUMBER(*, 0) , ITEM_SODIUM_NUTRITIONFACT NUMBER(*, 0) , CONSTRAINT ANALYTICS_RESTAURANT_ITEMS_PK PRIMARY KEY (ANALYTICS_ID) ALTER TABLE ANALYTICS_RESTAURANT_ITEMS ADD CONSTRAINT MANAGEMENT FOREIGN KEY (MANAGEMENT_ID) REFERENCES RESTAURANT_MANAGEMENT (MANAGEMENT_ID) ENABLE;
Count of records in the table	52

AUTHENTICATION_INFORMATION	
Primary Key	AUTHENTICATE_ID
Foreign Keys	MANAGEMENT_ID
SQL Code	CREATE TABLE AUTHENTICATION_INFORMATION (AUTHENICATE_ID NUMBER(*, 0) NOT NULL , MANAGEMENT_ID NUMBER(*, 0) NOT NULL , MESSAGE_TYPE VARCHAR2(200 BYTE) NOT NULL , RUNTIME_ERROR_MESSAGE VARCHAR2(200 BYTE) , FILE_SYSTEM_ERROR_MESSAGE VARCHAR2(200 BYTE) , CONNECTIVITY_ISSUE VARCHAR2(200 BYTE) , TIMESTAMP_OFERROR DATE NOT NULL , SEVERITY_LEVEL NUMBER(*, 0) NOT NULL , CONSTRAINT AUTHENTICATION_INFORMATION_PK PRIMARY KEY (AUTHENICATE_ID) ALTER TABLE AUTHENTICATION_INFORMATION ADD CONSTRAINT MANAGEMENT_ID FOREIGN KEY (MANAGEMENT_ID) REFERENCES RESTAURANT_MANAGEMENT (MANAGEMENT_ID) ENABLE;
Count of records in the table	12

CUSTOMER_RESERVATION	
Primary Key	CUSTOMERRESERVATION_ID
Foreign Keys	CUSTOMER_ID, TABLE_ID
SQL Code	CREATE TABLE CUSTOMER_RESERVATION (CUSTOMERRESERVATION_ID NUMBER(*, 0) NOT NULL , CUSTOMER_ID NUMBER(*, 0) NOT NULL , TABLE_ID NUMBER(*, 0) NOT NULL

	, PARTY_SIZE NUMBER(*, 0) , DATE_OF_RESERVATION DATE , TIME_OF_RESERVATION TIMESTAMP(6) , CONSTRAINT CUSTOMER_RESERVATION_PK PRIMARY KEY (CUSTOMERRESERVATION_ID) ALTER TABLE CUSTOMER_RESERVATION ADD CONSTRAINT CUSTOMER_ID FOREIGN KEY (CUSTOMER_ID) REFERENCES RESTAURANT_CUSTOMERS (CUSTOMER_ID) ENABLE; ALTER TABLE CUSTOMER_RESERVATION ADD CONSTRAINT TABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES TABLE_MAPS (TABLE_ID) ENABLE;
Count of records in the table	11

CUSTOMER_WAITING_QUEUE_INFO	
Primary Key	CUSTOMER_ID, HOST_ID
Foreign Keys	CUSTOMER_ID, HOST_ID
SQL Code	CREATE TABLE CUSTOMER_WAITING_QUEUE_INFO (CUSTOMER_ID NUMBER(*, 0) NOT NULL , HOST_ID NUMBER(*, 0) NOT NULL , TABLE_READY VARCHAR2(4 BYTE) NOT NULL , CONSTRAINT CUSTOMER_WAITING_QUEUE_INF_PK PRIMARY KEY (CUSTOMER_ID , HOST_ID) ALTER TABLE CUSTOMER_WAITING_QUEUE_INFO ADD CONSTRAINT CUSTOMERWAITING_HOSTID FOREIGN KEY (HOST_ID) REFERENCES RESTAURANT_HOSTS (HOST_ID) ENABLE; ALTER TABLE CUSTOMER_WAITING_QUEUE_INFO ADD CONSTRAINT CUSTOMERWAITING_ID FOREIGN KEY (CUSTOMER_ID) REFERENCES RESTAURANT_CUSTOMERS (CUSTOMER_ID

	) ENABLE;
Count of records in the table	*9*

HOSTSCHECK_CUSTOMERS	
Primary Key	CHECKSTATUSID
Foreign Keys	CUSTOMER_ID
SQL Code	CREATE TABLE HOSTSCHECK_CUSTOMERS (CHECKSTATUSID NUMBER(*, 0) NOT NULL , CUSTOMER_ID NUMBER(*, 0) NOT NULL , CUSTBEG_TIMEIN_WAIT DATE , PREPARING_TABLE VARCHAR2(4 BYTE) NOT NULL , CUSTEND_TIMEIN_WAIT DATE , CONSTRAINT HOSTSCHECK_CUSTOMERS_PK PRIMARY KEY (CHECKSTATUSID) ALTER TABLE HOSTSCHECK_CUSTOMERS ADD CONSTRAINT HOSTSCHECK_CUSTOMERID FOREIGN KEY (CUSTOMER_ID) REFERENCES RESTAURANT_CUSTOMERS (CUSTOMER_ID) ENABLE;
Count of records in the table	11

ORDERS	
Primary Key	ORDER_ID
Foreign Keys	CUSTOMER_ID, KITCHENSTAFF_ID
SQL Code	CREATE TABLE ORDERS (ORDER_ID NUMBER(*, 0) NOT NULL , CUSTOMER_ID NUMBER(*, 0) NOT NULL , KITCHENSTAFF_ID NUMBER(*, 0) NOT NULL , MAIN_COURSE_MEAL_ITEM VARCHAR2(95 BYTE) NOT NULL , APPETIZER_ITEM VARCHAR2(95 BYTE) , BEVERAGE_ITEM VARCHAR2(95 BYTE) , ALCOHOL_DRINK VARCHAR2(95 BYTE) , DESSERT_ITEM VARCHAR2(95 BYTE) , CONSTRAINT ORDERS_PK PRIMARY KEY (ORDER_ID) ALTER TABLE ORDERS ADD CONSTRAINT ORDERS_CUSTOMERID FOREIGN KEY (CUSTOMER_ID) REFERENCES RESTAURANT_CUSTOMERS (CUSTOMER_ID

	) ENABLE; ALTER TABLE ORDERS ADD CONSTRAINT ORDERS_KITCHENSTAFFID FOREIGN KEY (KITCHENSTAFF_ID) REFERENCES RESTAURANT_KITCHEN_STAFF (KITCHENSTAFF_ID) ENABLE;
Count of records in the table	45

POINT_OF_SALES	
Primary Key	POS_ID
Foreign Keys	RECEIPT_ID, SERVER_ID
SQL Code	CREATE TABLE POINT_OF_SALES (POS_ID NUMBER(*, 0) NOT NULL , ORDER_PRICE NUMBER(*, 0) NOT NULL , TIP_AMOUNT NUMBER(*, 0) , SERVER_ID NUMBER(*, 0) NOT NULL , RECEIPT_ID NUMBER(*, 0) NOT NULL , CONSTRAINT POINT_OF_SALES_PK PRIMARY KEY (POS_ID) ALTER TABLE POINT_OF_SALES ADD CONSTRAINT POS_RECEIPTID FOREIGN KEY (RECEIPT_ID) REFERENCES RECEIPTS (RECEIPT_ID) ENABLE; ALTER TABLE POINT_OF_SALES ADD CONSTRAINT POS_SERVERID FOREIGN KEY (SERVER_ID) REFERENCES RESTAURANT_SERVERS (SERVER_ID) ENABLE;
Count of records in the table	20

RECEIPTS	
Primary Key	RECEIPT_ID
Foreign Keys	CUSTOMER_ID
SQL Code	CREATE TABLE RECEIPTS

	(RECEIPT_ID NUMBER(*, 0) NOT NULL , CUSTOMER_ID NUMBER(*, 0) NOT NULL , ORDER_DATE DATE , SALES_TAX NUMBER(*, 0) , COLUMN3 NUMBER(*, 0) , CARD_PAYMENT VARCHAR2(60 BYTE) NOT NULL , CARDHOLDER_NAME VARCHAR2(60 BYTE) , CONSTRAINT RECEIPTS_PK PRIMARY KEY (RECEIPT_ID) ALTER TABLE RECEIPTS ADD CONSTRAINT RECEIPT_CUSID FOREIGN KEY (CUSTOMER_ID) REFERENCES RESTAURANT_CUSTOMERS (CUSTOMER_ID) ENABLE;
Count of records in the table	20

RESTAURANT_CUSTOMERS	
Primary Key	CUSTOMER_ID
Foreign Keys	SERVER_ID
SQL Code	CREATE TABLE RESTAURANT_CUSTOMERS (CUSTOMER_ID NUMBER(*, 0) NOT NULL , SERVER_ID NUMBER(*, 0) NOT NULL , FIRST_NAME VARCHAR2(65 BYTE) NOT NULL , LAST_NAME VARCHAR2(65 BYTE) NOT NULL , PHONE_NUMBER VARCHAR2(15 BYTE) NOT NULL , CUSTOMER_DOB DATE , CUSTOMER_GENDER VARCHAR2(6 BYTE) , CONSTRAINT RESTAURANT_CUSTOMERS_PK PRIMARY KEY (CUSTOMER_ID) ALTER TABLE RESTAURANT_CUSTOMERS ADD CONSTRAINT RESCUSTOMER_SERVERID FOREIGN KEY (SERVER_ID) REFERENCES RESTAURANT_SERVERS (SERVER_ID) ENABLE;
Count of records in the table	20

RESTAURANT_HOSTS	
Primary Key	HOST_ID
Foreign Keys	STAKE_ID
SQL Code	CREATE TABLE RESTAURANT_HOSTS

	(HOST_ID NUMBER(*, 0) NOT NULL , STAKE_ID NUMBER(*, 0) NOT NULL , FIRST_NAME VARCHAR2(65 BYTE) NOT NULL , LAST_NAME VARCHAR2(65 BYTE) NOT NULL , DOB DATE , TOTAL_YEARS_OF_EXPERIENCE NUMBER(*, 0) , CONSTRAINT RESTAURANT_HOSTS_PK PRIMARY KEY (HOST_ID) ADD CONSTRAINT RESHOSTS_STAKEID FOREIGN KEY (STAKE_ID) REFERENCES STAKEHOLDER_INFORMATION (STAKE_ID) ENABLE;
Count of records in the table	10

RESTAURANT_KITCHEN_STAFF	
Primary Key	KITCHENSTAFF_ID
Foreign Keys	STAKE_ID
SQL Code	CREATE TABLE RESTAURANT_KITCHEN_STAFF (KITCHENSTAFF_ID NUMBER(*, 0) NOT NULL , STAKE_ID NUMBER(*, 0) NOT NULL , FIRST_NAME VARCHAR2(65 BYTE) NOT NULL , LAST_NAME VARCHAR2(65 BYTE) NOT NULL , DOB DATE , TOTAL_YEARS_OF_EXPERIENCE NUMBER(*, 0) , CONSTRAINT RESTAURANT_KITCHEN_STAFF_PK PRIMARY KEY (KITCHENSTAFF_ID) ALTER TABLE RESTAURANT_KITCHEN_STAFF ADD CONSTRAINT RESKITCHSTAFF_STAKEID FOREIGN KEY (STAKE_ID) REFERENCES STAKEHOLDER_INFORMATION (STAKE_ID) ENABLE;
Count of records in the table	10

RESTAURANT_MANAGEMENT	
Primary Key	MANAGEMENT_ID
Foreign Keys	STAKE_ID
SQL Code	CREATE TABLE RESTAURANT_MANAGEMENT (MANAGEMENT_ID NUMBER(*, 0) NOT NULL , STAKE_ID NUMBER(*, 0) NOT NULL , FIRST_NAME VARCHAR2(20 BYTE) NOT NULL

	, LAST_NAME VARCHAR2(20 BYTE) NOT NULL , DOB DATE , TOTAL_YEARSOF_EXPERIENCE NUMBER(*, 0) , CONSTRAINT RESTAURANT_MANAGEMENT_PK PRIMARY KEY (MANAGEMENT_ID) ADD CONSTRAINT RESMANAG_STAKEID FOREIGN KEY (STAKE_ID) REFERENCES STAKEHOLDER_INFORMATION (STAKE_ID) ENABLE;
Count of records in the table	10

RESTAURANT_SERVERS	
Primary Key	SERVER_ID
Foreign Keys	STAKE_ID
SQL Code	CREATE TABLE RESTAURANT_SERVERS (SERVER_ID NUMBER(*, 0) NOT NULL , STAKE_ID NUMBER(*, 0) NOT NULL , FIRST_NAME VARCHAR2(65 BYTE) NOT NULL , LAST_NAME VARCHAR2(65 BYTE) NOT NULL , DOB DATE , TOTAL_YEARS_OF_EXPERIENCE NUMBER(*, 0) , CONSTRAINT RESTAURANT_SERVERS_PK PRIMARY KEY (SERVER_ID) ALTER TABLE RESTAURANT_SERVERS ADD CONSTRAINT RESSERVER_STAKEID FOREIGN KEY (STAKE_ID) REFERENCES STAKEHOLDER_INFORMATION (STAKE_ID) ENABLE;
Count of records in the table	10

STAFF_SCHEDULES	
Primary Key	STAFF_SCHEDULEID
Foreign Keys	HOST_ID, KITCHENSTAFF_ID, MANAGEMENT_ID, SERVER_ID
SQL Code	CREATE TABLE STAFF_SCHEDULES (STAFF_SCHEDULEID NUMBER(*, 0) NOT NULL , KITCHENSTAFF_ID NUMBER(*, 0) NOT NULL , MANAGEMENT_ID NUMBER(*, 0) NOT NULL , SERVER_ID NUMBER(*, 0) NOT NULL , HOST_ID NUMBER(*, 0) NOT NULL , HOURS_WORKED NUMBER(*, 0)

	, WEEKS_WORKED NUMBER(*, 0) , DAILY_SCHEDULE VARCHAR2(40 BYTE) , NUMBER_VACATIONDAYS NUMBER(*, 0) , SICKDAYS NUMBER(*, 0) , YEARLYHOLIDAYS NUMBER(*, 0) , CONSTRAINT STAFF_SCHEDULES_PK PRIMARY KEY (STAFF_SCHEDULEID) ALTER TABLE STAFF_SCHEDULES ADD CONSTRAINT SCHEDULE_HOSTID FOREIGN KEY (HOST_ID) REFERENCES RESTAURANT_HOSTS (HOST_ID) ENABLE; ALTER TABLE STAFF_SCHEDULES ADD CONSTRAINT SCHEDULE_KITCHENID FOREIGN KEY (KITCHENSTAFF_ID) REFERENCES RESTAURANT_KITCHEN_STAFF (KITCHENSTAFF_ID) ENABLE; ALTER TABLE STAFF_SCHEDULES ADD CONSTRAINT SCHEDULE_MANAGEMENTID FOREIGN KEY (MANAGEMENT_ID) REFERENCES RESTAURANT_MANAGEMENT (MANAGEMENT_ID) ENABLE; ALTER TABLE STAFF_SCHEDULES ADD CONSTRAINT SCHEDULE_SERVERID FOREIGN KEY (SERVER_ID) REFERENCES RESTAURANT_SERVERS (SERVER_ID) ENABLE;
Count of records in the table	40

STAKEHOLDER_INFORMATION	
Primary Key	STAKE_ID
Foreign Keys	-
SQL Code	CREATE TABLE STAKEHOLDER_INFORMATION (STAKE_ID NUMBER(*, 0) NOT NULL , EMAIL VARCHAR2(80 BYTE) , HIRE_DATE DATE NOT NULL

	, YEARLY_SALARY FLOAT(126) , BENEFIT_TYPE VARCHAR2(40 BYTE) , YEARLY_BONUS_AMOUNT FLOAT(126) , EMPLOYMENT_CLASSIFICATION VARCHAR2(40 BYTE) , CONSTRAINT STAKEHOLDER_INFORMATION_PK PRIMARY KEY (STAKE_ID)
Count of records in the table	40

TABLE_MAPS	
Primary Key	TABLE_ID
Foreign Keys	HOST_ID
SQL Code	CREATE TABLE TABLE_MAPS (TABLE_ID NUMBER(*, 0) NOT NULL , HOST_ID NUMBER(*, 0) NOT NULL , TABLE_SQ_FT FLOAT(126) , TABLE_SEATING_CAPACITY NUMBER(*, 0) , TABLE_SHAPE VARCHAR2(60 BYTE) , YEARS_USED_INSERVICE NUMBER(*, 0) , CONSTRAINT TABLE_MAPS_PK PRIMARY KEY (TABLE_ID) ALTER TABLE TABLE_MAPS ADD CONSTRAINT TABLEMAPS_HOSTID FOREIGN KEY (HOST_ID) REFERENCES RESTAURANT_HOSTS (HOST_ID) ENABLE;
Count of records in the table	12

TABLE_STATUS	
Primary Key	TABLESTATUS_ID
Foreign Keys	TABLE_ID
SQL Code	CREATE TABLE TABLE_STATUS (TABLESTATUS_ID NUMBER(*, 0) NOT NULL , TABLE_ID NUMBER(*, 0) NOT NULL , IS_OPEN CHAR(4 BYTE) NOT NULL , CONSTRAINT TABLE_STATUS_PK PRIMARY KEY (TABLESTATUS_ID) ALTER TABLE TABLE_STATUS ADD CONSTRAINT TABLESTATUS_TABLEID FOREIGN KEY (TABLE_ID) REFERENCES TABLE_MAPS

	(TABLE_ID) ENABLE;
Count of records in the table	12

Assumptions and Constraints

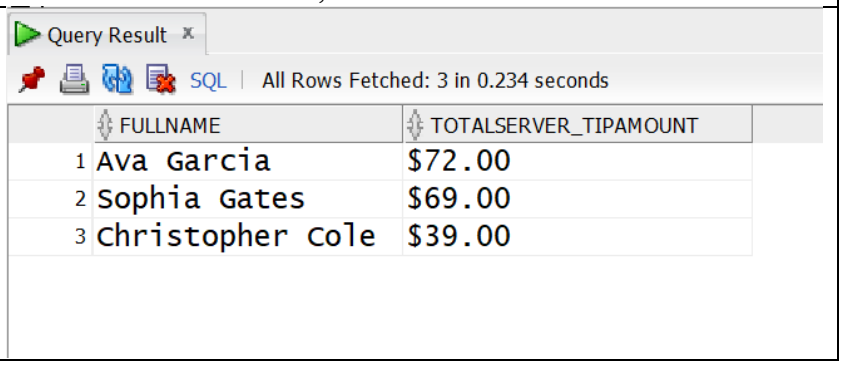
Stakeholder_Information table is the initial source where all restaurant staff members data comes from in the database. It is also the only table that does not have a foreign key associated with it.

Naming Conventions

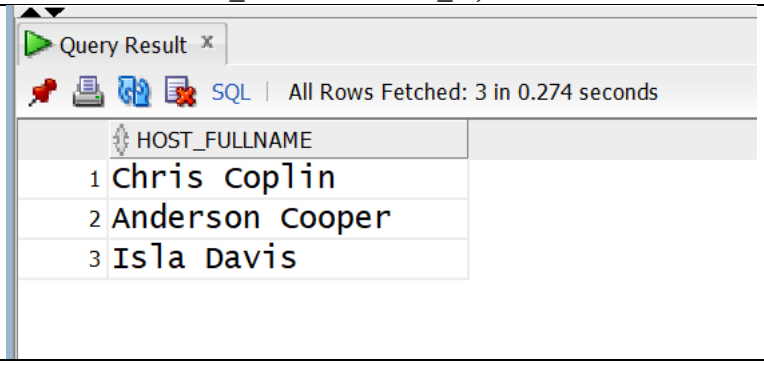
Oracle only allows for uppercase lettering, so all table names and column names are all capitalized. Oracle also does not allow for spaces to be included in table names or column names. So, all table and column names in this database have an underscore character to replace space characters. Oracle also has a character limit. So, for some column names, truncations/abbreviations were needed, to fit names under Oracle's character limit constraints.

MILESTONE 6: SQL QUERIES AND

Note: Please make sure you add/have 25 records in each table, on average.

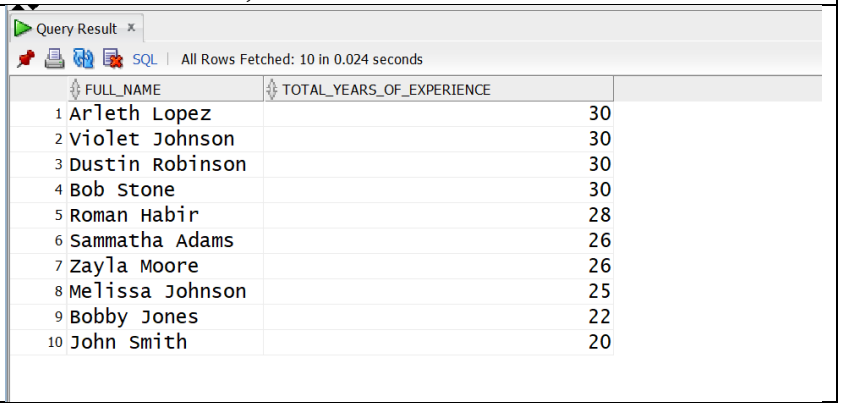
Query 1	
English version	Return the top 3 restaurant servers who received the most tips from customers.
Source for the query need in the SRS document	*SRS document, pages 10, Point-of-Sale, section 3.5.1.5*
SQL sentence	Select * from (select first_name||' '||last_name as FullName, to_char(sum(tip_amount), 'fmL99G999D00')as TotalServer_TipAmount from point_of_sales ps join restaurant_servers rs on rs.server_id=ps.server_id group by rs.server_id,rs.first_name,rs.last_name order by TotalServer_TipAmount desc) WHERE ROWNUM <= 3;
Example of returned rows (cropped screen caption)	

Query 2	
English version	Return the top 3 restaurant hosts who sat customers at their respective tables the fastest.
Source for the query need in the SRS document	*SRS document, pages 11-12, section 3.5.3.1 & 3.5.3.7*

SQL sentence	select distinct First_name||' '||Last_name as Host_FullName from restaurant_hosts rh join (select cwqi.customer_id,host_id from customer_waiting_queue_info cwqi join (select rpt.customer_id from receipts rpt join (select * from(select customer_id, hostc.custend_timein_wait - hostc.custbeg_timein_wait as difference from hostscheck_customers hostc order by difference) where rownum<=3) custWait on rpt.customer_id=custWait.customer_id) cusSer on cwqi.customer_id=cusSer.customer_id) fastHost on rh.host_id=fastHost.host_id;
Example of returned rows (cropped screen caption)	

Query 3	
English version	Return the top 10 restaurant staff members who have the most work experience.
Source for the query need in the SRS document	*SRS document, pages 7, section 2.3 - Stakeholder Characteristics*

SQL sentence	select *
	from(
	  select *
	  from(
	  select first_name||' '||last_name as Full_Name,total_years_of_experience
	  from restaurant_hosts
	  union
	  select first_name||' '||last_name,total_years_of_experience
	  from restaurant_kitchen_staff
	  union
	  select first_name||' '||last_name,total_yearsof_experience
	  from restaurant_management
	  union
	  select first_name||' '||last_name,total_years_of_experience
	  from restaurant_servers
	  )
	  order by total_years_of_experience desc
	)
	where rownum<=10;

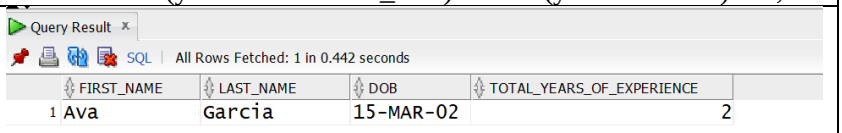
Example of returned rows (cropped screen caption)	

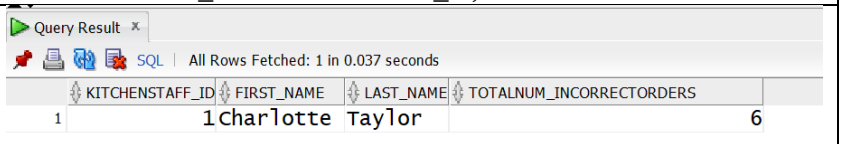
Query Result

SQL | All Rows Fetched: 10 in 0.024 seconds

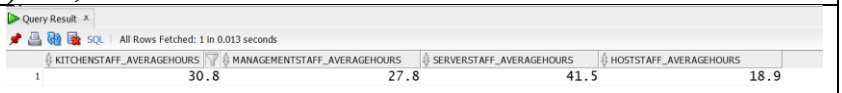
	FULL_NAME	TOTAL_YEARS_OF_EXPERIENCE
1	Arleth Lopez	30
2	Violet Johnson	30
3	Dustin Robinson	30
4	Bob Stone	30
5	Roman Habir	28
6	Sammatha Adams	26
7	Zayla Moore	26
8	Melissa Johnson	25
9	Bobby Jones	22
10	John Smith	20

Query 4	
English version	Return restaurant servers who served alcoholic drinks to customers, that are under the age of 21.
Source for the query need in the SRS document	*SRS document, pages 9, section 3.1 - Domain Requirements*

SQL sentence	select distinct First_name,Last_name,DOB,Total_years_of_experience from (select server_id from (select receipt_id from (select customer_id,alcohol_drink from orders where alcohol_drink is not null) cusAl join receipts r on cusAl.customer_id=r.customer_id) recAl join point_of_sales pos on recAl.receipt_id=pos.receipt_id) serAl join restaurant_servers resSer on serAl.server_id=resSer.server_id where extract(year from current_date)-extract(year from dob)<21;
Example of returned rows (cropped screen caption)	

Query 5	
English version	Return restaurant kitchen staff members who served more than 5 incorrect orders to customers.
Source for the query need in the SRS document	*SRS document, pages 9, Primary Stakeholder Identification and Ranking: Stakeholder Class Rank*
SQL sentence	Select ktin.kitchenstaff_id, first_name,last_name,TotalNum_IncorrectOrders from (select kitchenstaff_id, count(kitchenstaff_id) as TotalNum_IncorrectOrders from orders where lower(order_accurate)='no' group by kitchenstaff_id having count(order_accurate)>=5) ktin join restaurant_kitchen_staff rks on ktin.kitchenstaff_id=rks.kitchenstaff_id;
Example of returned rows (cropped screen caption)	

Query 6	
English version	Return the average hours worked by each department in the restaurant for scheduling purposes.
Source for the query need in the SRS document	*SRS document, pages 9, Primary Stakeholder Identification and Ranking: Stakeholder Class Rank*

SQL sentence	select * from (select avg(hours_worked) as KitchenStaff_AverageHours from staff_schedules where kitchenstaff_id is not null) ktch, (select avg(hours_worked) as ManagementStaff_AverageHours from staff_schedules where management_id is not null) mgs, (select avg(hours_worked) as ServerStaff_AverageHours from staff_schedules where server_id is not null) serv, (select avg(hours_worked) as HostStaff_AverageHours from staff_schedules where host_id is not null) host;
Example of returned rows (cropped screen caption)	

Query 7	
English version	Return the four most popular main course meal items ordered by the restaurant customers.
Source for the query need in the SRS document	*SRS document, page 6, section 2.2 – Product Functions, page 12, section 3.5.4 – Data Analytics*
SQL sentence	select main_course_meal_item,count(main_course_meal_item) as Freqeuncy_ItemOrdered from orders group by main_course_meal_item having count(main_course_meal_item)>2 order by count(main_course_meal_item) desc;
Example of returned rows (cropped screen caption)