



DATABASE SPECIFICATIONS

Next Gen-Restaurant Application
Rudraksh Mishra- rjm7016@psu.edu

October 2021 – December 2021

DOCUMENT CONTROL

Work carried out by:

Name	Email Address	Other
Rudraksh Mishra	rjm7016@psu.edu	Rudraksh.mishra04@gmail.com

Revision Sheet

Release No.	Date	Revision Description
1	Nov 1 st 2021	Developed a core requirements table
2	Nov 8 th 2021	Developed a conceptual design and made updates to the first revision
3	Nov 15 th 2021	Developed a logical design and made updates to the second revision
4	Nov 21 st 2021	Normalized the tables and made updates to third revision
5	Dec 5 th 2021	Developed physical design and made the updates to fourth revision
6	Dec 9 th 2021	Developed SQL Queries and made the updates to the fifth revision

DATABASE SPECIFICATIONS

TABLE OF CONTENTS

Sr.No	Title	Page Number
1	Document Control	i
2	Milestone 1: Data Requirements	2
3	Milestone 2: Conceptual Design	4
4	Milestone 3: Logical Design	10
5	Milestone 4: Normalization	15
6	Milestone 5: Physical Design	16
7	Milestone 6: SQL queries	27

MILESTONE 1: DATA REQUIREMENTS

System Name or Title

Next Gen-Restaurant Application

Core requirements

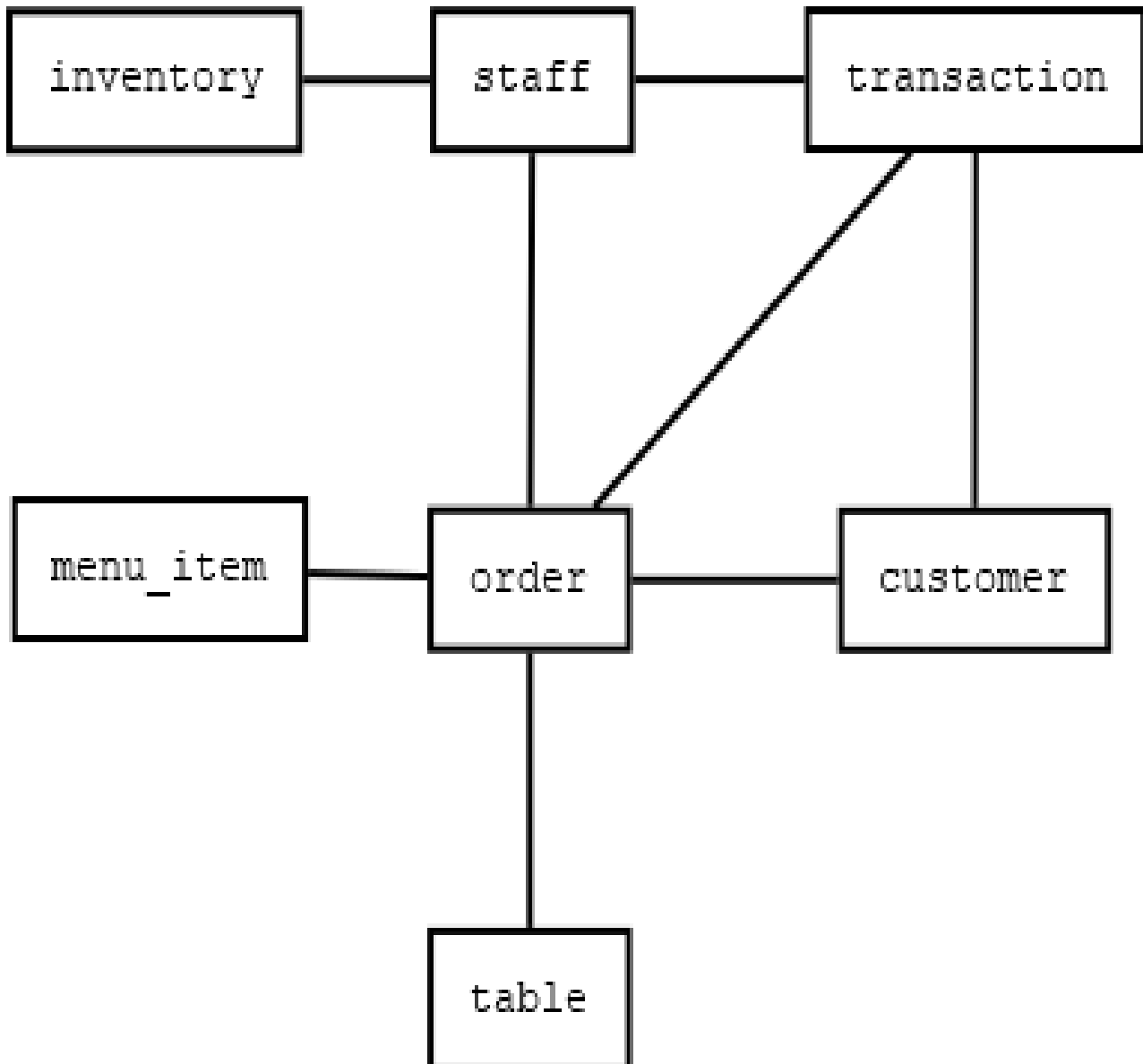
No	Requirement	Referenced page in SRS	Referenced Section in SRS	Referenced Paragraph in Section
1	The database should store information about customer. It needs to store information like customer name, customer id, phone number, DOB.	3	1.2	1
		6	2.3	2
		11	3	3.5.3.3
2	The party size, reservation time, reservation date, table number should also be stored when assigned. When there a no tables available the customer will be assigned a waiting list number.	10	3	3.5.3.2
		11	3	3.5.3.7
		11	3	3.5.3.8
		8	3	3.1.1
3	The database should store unique id and user role to the users of the system. The user's roles include: management members, servers, host/hostess, kitchen staff. Different type of users are given different authorizations. Each of the user's information is stored using, username, password, name, DOB.	6	2.3	2
		8	3	3.1.2
		10	3	3.5.3.1
		13	5	1, 2
4	The database should store the menu using item name, item id, cost per serving, item category. The item categories include entrée, side order, beverages, bar.	19	8	8.1.1
5	The order should be stored using order id, table number, customer id, item id, and item quantity. The status of the order should also be stored under order status. The statuses include order submitted, cancelled, order confirmation, preparing, ready for pick up and served.	19	8	8.1.1
		19	8	8.1.2
6	The payment information should be stored using, ordered, payment id, amount paid, payment mode card details. We also need to store the id of the server assigned to the table, so that tip should be added to his payroll.	9	3	3.5.1.5
		9	3	3.5.1.9
		13	5	5.3

7	To edit the table map, we should store the properties of the table like, table id, table name, table capacity, status of the table, table shape, and table type	10 32	3 8	3.5.2 8.1.14
8	The database should store staff schedule.	5	2	2
9	Information about bar order should be stored.	10	3	3.5.1.10
10	Information about the ingredients should be stored.	7	2.3	table

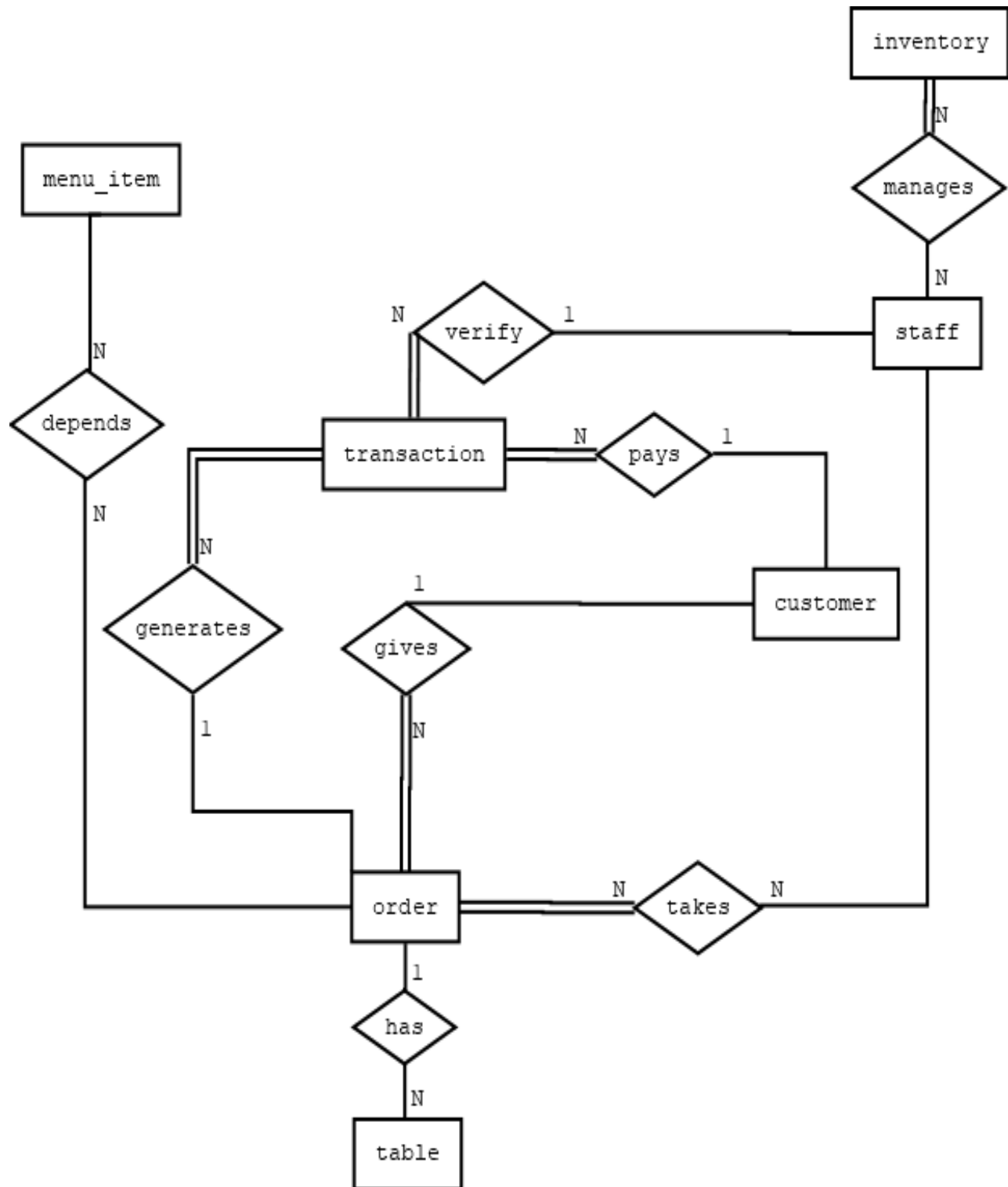
MILESTONE 2: CONCEPTUAL DESIGN

Diagram

The below diagram will give a basic understanding on which entities are related. This will allow us to get a summarized view of the design:

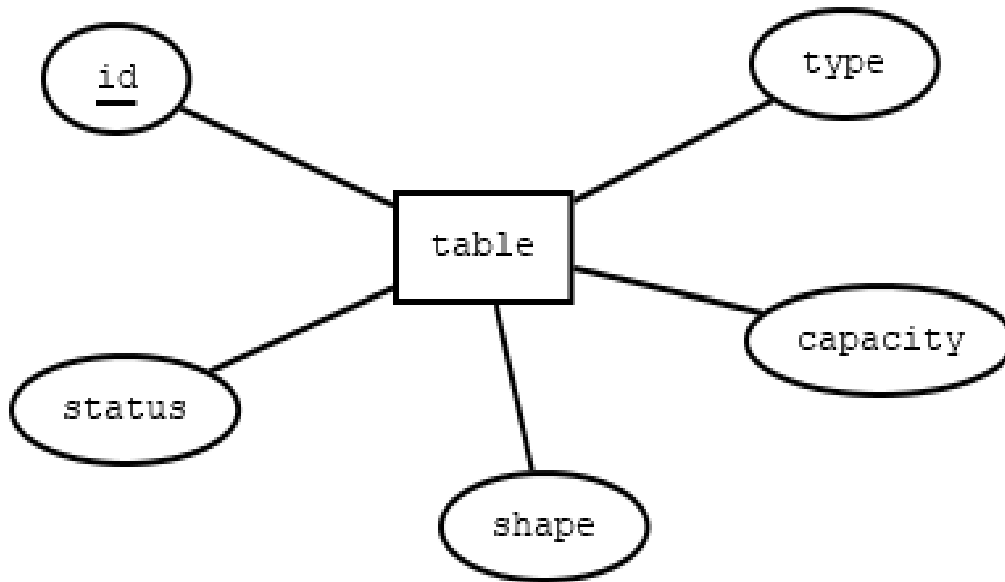


The below diagram will allow us to see how the entities are related. This gives us a detailed view of the systems design:

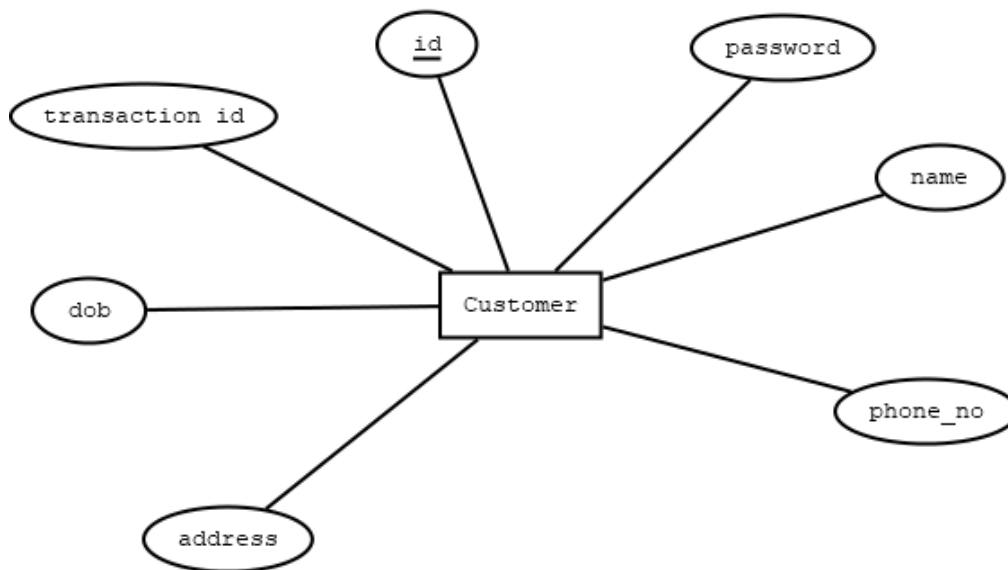


Below ER diagrams will give us the information about attributes in each of the entity.

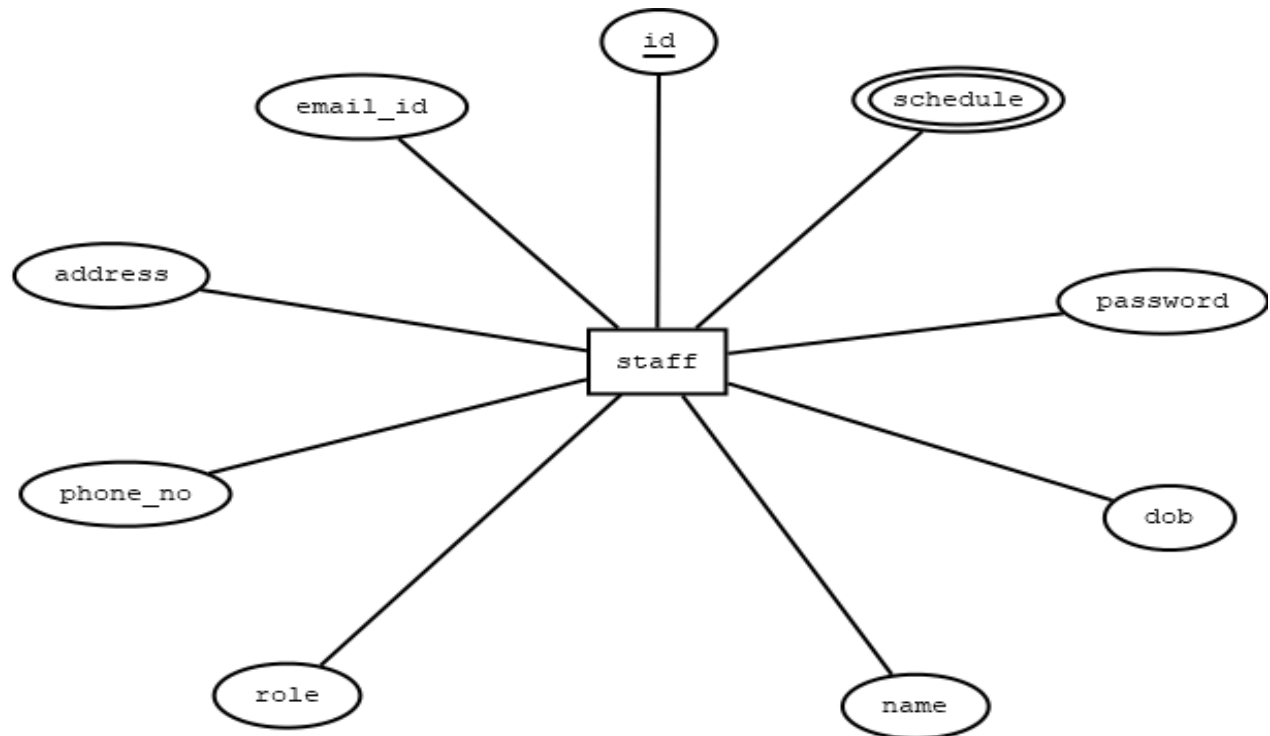
Table:



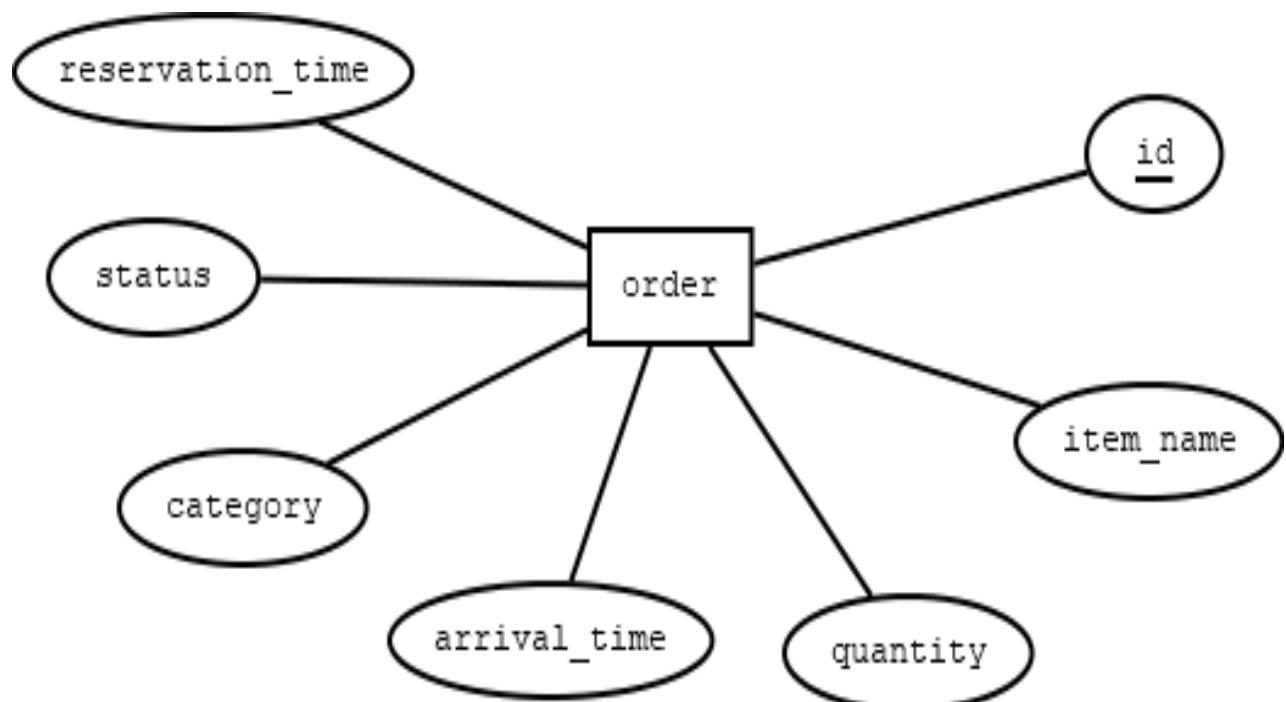
Customer:



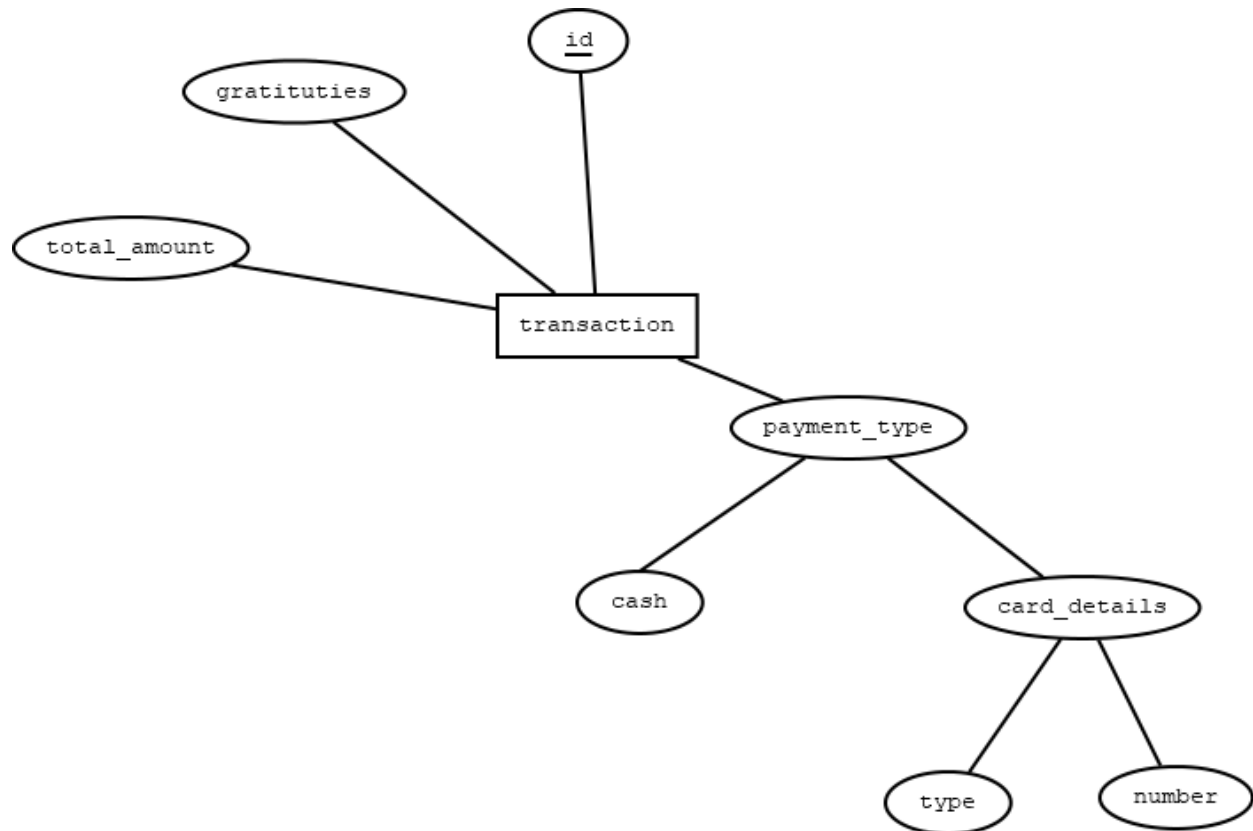
Staff:



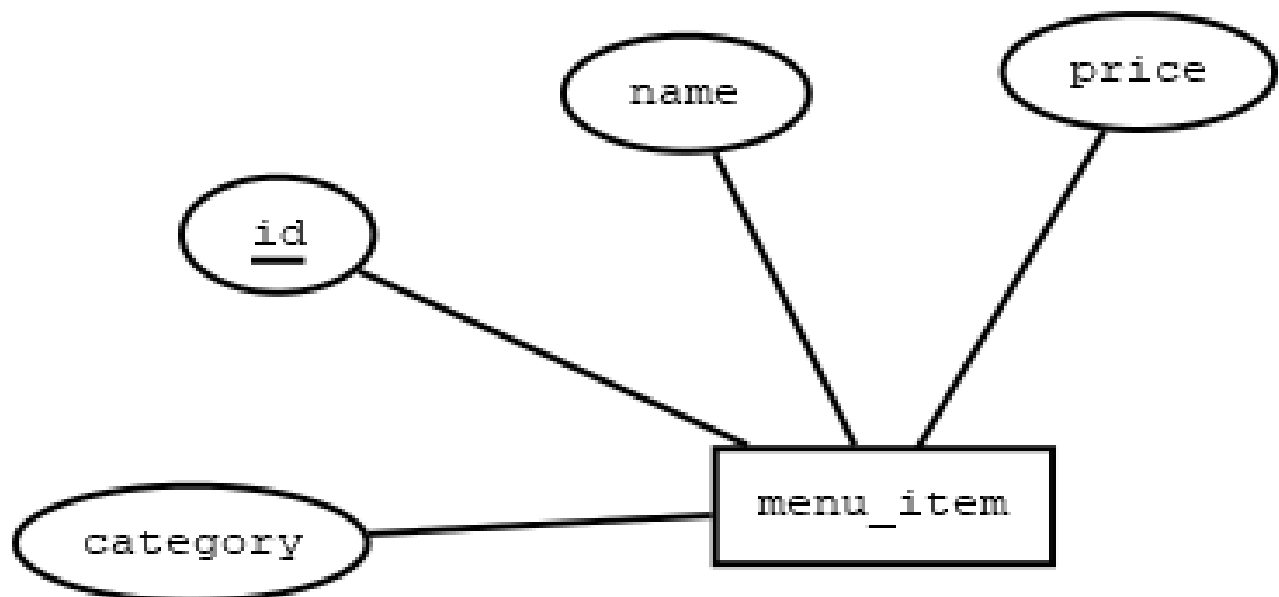
Order:



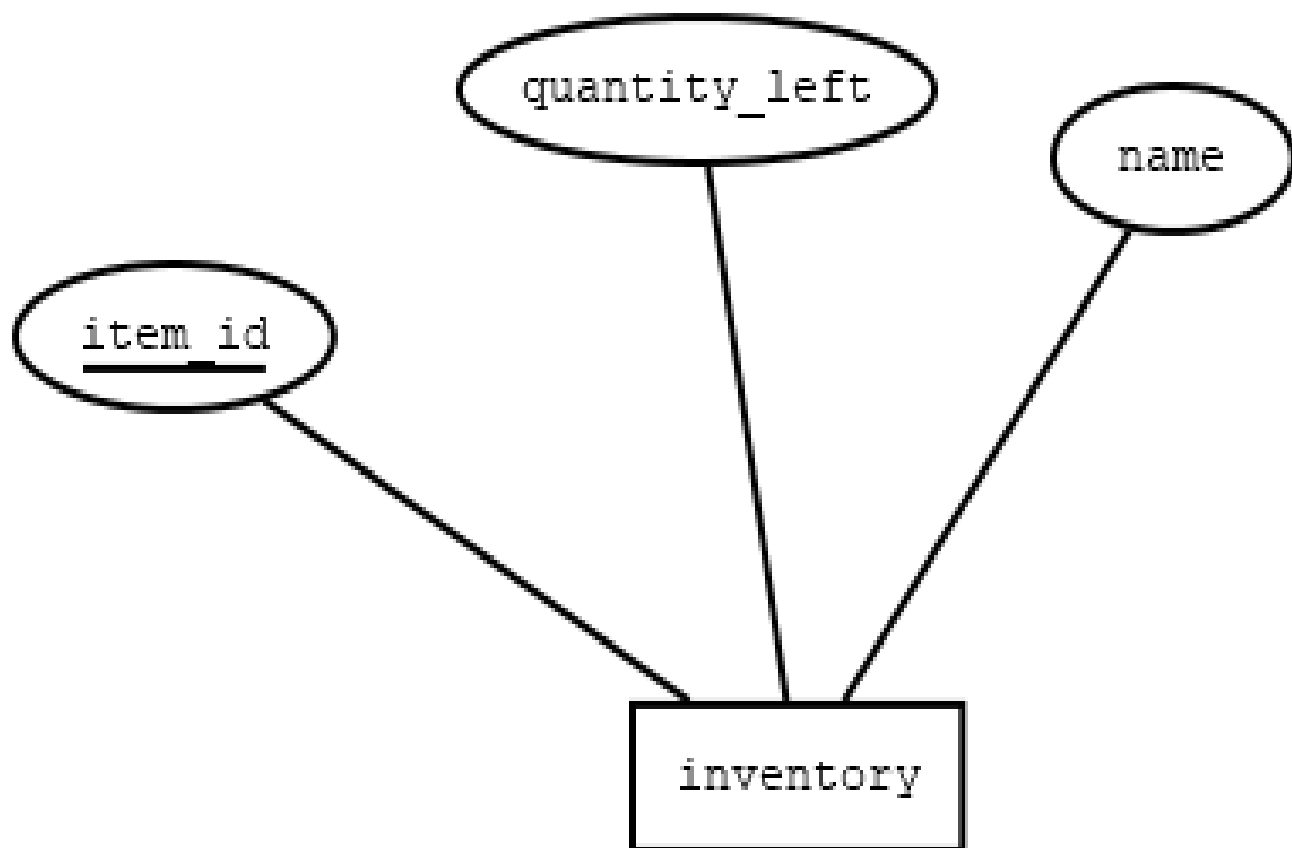
Transaction:



Menu:



Inventory



Assumptions and Constraints

- One customer represents the entire group
- Menu depends on the inventory through the staff
- Order includes food orders and bar orders
- Staff with different roles have different authorizations
- Customers can book multiple tables if the table capacity is less than group size
- A bill can be split between multiple people in a group
- Menu has both food menu and bar menu
- One order is taken by multiple staff members, namely kitchen staff and server
- The stored password should be encrypted
- The item categories include entrée, side order, beverages, bar
- Only 2 staff members are authorized to edit the inventory, one person deals with the food and the other deals with the bar. Category id in the inventory decides who deals with what.

MILESTONE 3: LOGICAL DESIGN

Entity Relationship Diagram

Entity name: Table

Attributes:

Id, type, capacity, shape, status

Functional dependencies:

Id \rightarrow name, type, capacity, shape, status

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	id		name, type, capacity, shape, status

Attribute closures (if any):

(Id) $^+=$ name, type, capacity, shape, status

Unique keys: the key for this table are

id

Entity name: Customer

Attributes:

id, password, name, phone no, address, dob

Functional dependencies:

id \rightarrow password, name, phone no, address, dob

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	id		password, name, phone no, address, dob

Attribute closures (if any):

(id) $^+=$ password, name, phone no, address, dob

Unique keys: the key for this table is

id

Entity name: staff

Attributes:

id, password, dob, name, role, phone number, address, email_id,

Functional dependencies:

id \rightarrow password, dob, name, role, phone number, address, email_id,

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	id		password, dob, name, role, phone number, address, email_id,

Attribute closures (if any):

(Id) \rightarrow password, dob, name, role, phone number, address,

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

Entity name: Schedule

Attributes:

staff_id, date, start_time, out_time

Functional dependencies:

staff_id ,date \rightarrow start_time, out_time

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Date Staff_id		start_time, out_time

Attribute closures (if any):

(staff_id, date) \rightarrow start_time, out_time

Unique keys: the key for this table is/are
staff_id, date (composite keys)

Entity name: order

Attributes:

id, item_name, quantity, arrival_time, type, status, reservation_time,

Functional dependencies:

id → item_name, quantity, arrival_time, type, status, reservation_time

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	id		item_name, quantity, arrival_time, type, status, reservation_time

Attribute closures (if any):

(Id)+= item_name, quantity, arrival_time, type, status, reservation_time

Unique keys: the key for this table are
id

Entity name: transaction

Attributes:

Id, gratuities, total_amount, pay_cash, card_type, card_number

Functional dependencies:

id → gratuities, total_amount, pay_cash, card_type, card_number

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	id		gratuities, total_amount, pay_cash, card_type, card_number

Attribute closures (if any):

(Id)+= gratuities, total_amount, pay_cash, card_type, card_number

Unique keys: the key for this table are
Id

Entity name: menu_item

Attributes:

Id, name, category, price

Functional dependencies:

$id \rightarrow name, category, price$

$name \rightarrow id, category, price$

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Id		category, price, name

Attribute closures (if any):

$(id)^+ = name, category, price$

Unique keys: the key for this table is/are

id

Entity name: inventory

Attributes:

Id, name, quatity,

Functional dependencies:

$id \rightarrow name, quatity,$

Attributes not in FD	Attributes on the left	Attributes on both sides	Attributes on the right side
	Id		quatity, category_id, name

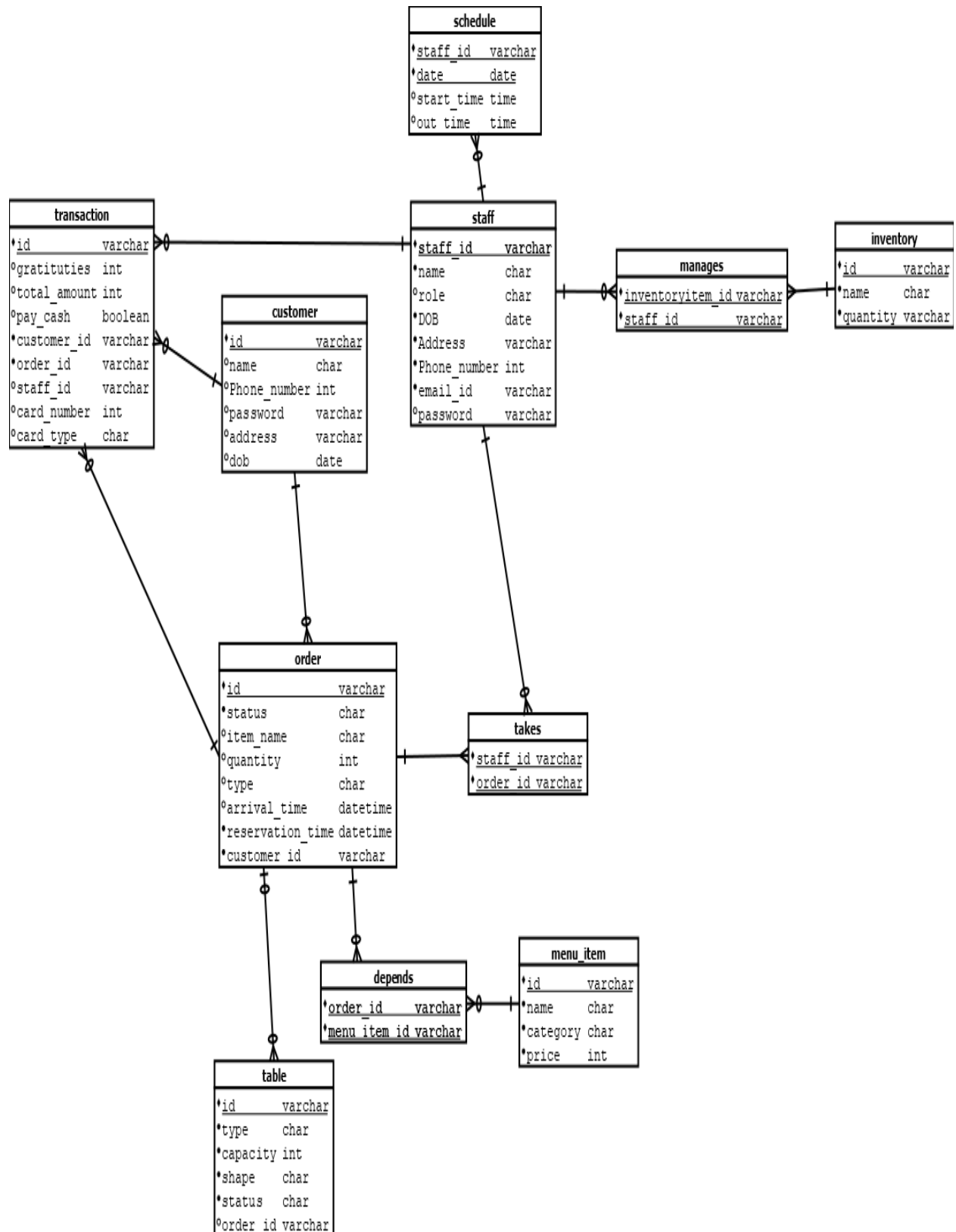
Attribute closures (if any):

$(id)^+ = name, quatity, category_id$

Unique keys: the key for this table is/are

Id

Design



MILESTONE 4: NORMALIZATION AND

MILESTONE 5: PHYSICAL DESIGN

Assumptions and Constraints

Naming Conventions

Discuss the naming standards and conventions that you have used for table creation.

Tables

	<i>Name of the table</i>	<i>Transactions</i>			
	Description	An transaction is the way the customer pays bill for his/her order at restaurant			
	Attribute	Description	Type	Examples of values	Notes
	transaction_id	Unique transaction number	Varchar	T10000, T10001...	
	gratitude	Tip from customer to staff	Integer	10\$,19\$...	
	total_amount	Total cost of order including taxes	Integer	20\$,100\$	
	pay_cash	Payment done by customer using cash	Char	Yes, No	
	Card_number	Number on the card which was used to pay the amount	Integer	64649279472, 52429346923..	
	Card_type	Type of the card used for the transactiopn	Char	Credit, debit, ...	
	Functional Dependencies and Keys				
	Functional dependencies	transaction_id -> gratitude, total_amount, pay_cash, card_number, card_type			
	Candidate keys	transaction_id			
	Normalization				
	1NF	Yes	No arrays and repeated values nor any repeated column names		
	2NF	Yes	No partial dependencies included		
	3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key		

BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation $X \rightarrow Y$, X is a super key of relation R)
Physical Design		
Primary Key	transaction_id	
Foreign Keys	order_id, staff_id, customer_id	
SQL Code	CREATE TABLE transactions (transaction_id VARCHAR2(40) NOT NULL, gratuities INTEGER, total_amount INTEGER NOT NULL, pay_cash VARCHAR(40), card_number INTEGER, card_type VARCHAR(40), customer_id VARCHAR2(40) NOT NULL, order_id VARCHAR2(40) NOT NULL, staff_id VARCHAR2(40) NOT NULL, PRIMARY KEY (transaction_id), CONSTRAINT FK_customertransaction FOREIGN KEY (customer_id) REFERENCES customer(customer_id), CONSTRAINT FK_ordertransaction FOREIGN KEY (order_id) REFERENCES orders(order_id), CONSTRAINT FK_stafftransaction FOREIGN KEY (staff_id) REFERENCES staff(staff_id));	
Count of records in the table	30	

<i>Name of the table</i>	<i>Customer</i>				
Description	A customer is the user who orders at the restaurant.				
Attribute	Description	Type	Examples of values	Notes	
customer_id	Unique customer id	Varchar	C10000, C10001...		
name	Name of the customer	Char	Jhon doe. Jane Mari		
phone_number	Phone number of the customer	Integer	7892445712, 9423885801		
password	A secret word or phrase that must be used to gain access to restaurant system	Varchar	Hello@123, Bie_xyz, 11nov1996.....		
address	Address of the customer	Varchar	Apt no: r3, Xyz street, 190412		
dob	Date of birth of the customer	Date	11/01/1972, 03/24/1994		

Functional Dependencies and Keys				
Functional dependencies	customer_id -> name, phone_number, password, address, dob			
Candidate keys	customer_id			
Normalization				
1NF	Yes	No arrays and repeated values nor any repeated column names		
2NF	Yes	No partial dependencies included		
3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key		
BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation X->Y, X is a super key of relation R)		
Physical Design				
Primary Key	customer_id			
Foreign Keys	-			
SQL Code	CREATE TABLE customer (customer_id VARCHAR2(20)NOT NULL, cname VARCHAR2(40) NOT NULL, phone_number INTEGER, cpassword VARCHAR2(40), address VARCHAR2(40), dob DATE, PRIMARY KEY(customer_id));			
Count of records in the table	30			

<i>Name of the table</i>	<i>Orders</i>				
Description	Order is what the customer requests to be served				
Attribute	Description	Type	Examples of values	Notes	
order_id	Unique order number	Varchar	O10001, O10002..		
status	Status of the order	Char	confirmed, preparing, served		
item_name	Name of the item ordered	Char	Pasta, pizza		
quantity	Quantity of the item ordered	Integer	1, 2, 3...		
type	Category of the order item	Char	Food, Bar		
arrival_time	Time when the customer arrived at the restaurent	Datetime	2008-11-11 13:23:44		
reservation_time	Reservation time of table of the cutomer	Datetime	2008-11-11 13:23:44		

	Functional Dependencies and Keys				
Functional dependencies	order_id -> status, item_name, quatity, type, arrival_time, reservation_time				
Candidate keys	order_id				
Normalization					
1NF	Yes	No arrays and repeated values nor any repeated column names			
2NF	Yes	No partial dependencies included			
3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key			
BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation X->Y, X is a super key of relation R)			
	Physical Design				
Primary Key	order_id				
Foreign Keys	customer_id				
SQL Code	CREATE TABLE orders (order_id VARCHAR2(40) PRIMARY KEY, status VARCHAR2(40) , item_name VARCHAR2(40), quantity INTEGER, arrival_time TIMESTAMP, reservation_time TIMESTAMP NOT NULL, item_type VARCHAR2(40), customer_id VARCHAR2(40) NOT NULL, CONSTRAINT FK_customerorders FOREIGN KEY(customer_id) REFERENCES customer(customer_id));				
Count of records in the table	30				

Name of the table	<i>Sit_table</i>				
Description	An individual table at the restaurant				
Attribute	Description	Type	Examples of values	Notes	
table_id	Unique table number	Varchar	T1, T2 ...		
table_type	Type of the table	Char	Large, Small..		
table_capacity	Maximum number of people the table can hold	Integer	2,4,8,12..		
table_shape	Shape of the table	char	Square, Rectangle..		
table_status	The status of the table	Char	Reserved, Open, Occupied..		

Functional Dependencies and Keys				
Functional dependencies	table_id -> table_type, table_capacity, table_shape, table_status			
Candidate keys	id			
Normalization				
1NF	Yes	No arrays and repeated values nor any repeated column names		
2NF	Yes	No partial dependencies included		
3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key		
BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation X->Y, X is a super key of relation R)		
Physical Design				
Primary Key	table_id			
Foreign Keys	order_id			
SQL Code	CREATE TABLE sit_table (table_id VARCHAR2(40) NOT NULL, table_type VARCHAR2(40) NOT NULL, table_capacity INTEGER NOT NULL, table_shape VARCHAR2(40) NOT NULL, table_status VARCHAR2(40) NOT NULL, order_id VARCHAR2(40), PRIMARY KEY (table_id), CONSTRAINT FK_orderstable FOREIGN KEY (order_id) REFERENCES orders(order_id));			
Count of records in the table	30			

	<i>Name of the table</i>	<i>Staff</i>			
	Description	The authorized individuals who work at the restaurant			
	Attribute	Description	Type	Examples of values	Notes
	staff_id	Unique staff number	Varchar	S1001, S1002..	
	sname	Name of the staff member	Char	Jacob tyler, Jane Doe...	
	srole	The role of the customer in the restaurant	Char	Chef, host, server, manger,...	
	address	Address of the staff	Varchar	Apt no: r3, Xyz street, 190412	
	phone_number	Contact number of the staff	Integer	2453286293, 3467812649	
	email_id	Email id of the staff	Varchar	Jane.d@gmail.com	
	dob	date of birth of the staff	Date	11/04/1997	

	password	A secret word or phrase that must be used to gain access to restaurant system	Varchar	Hello@123, Bie_xyz, 11nov1996.....	
	Functional Dependencies and Keys				
	Functional dependencies	staff_id -> sname, srole, address, phone_number, email_id, password email_id -> staff_id, sname, srole, address, phone_number, password			
	Unique Key	email_id			
	Candidate keys	staff_id			
	Normalization				
	1NF	Yes	No arrays and repeated values nor any repeated column names		
	2NF	Yes	partial dependencies included		
	3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key		
	BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation X->Y, X is a super key of relation R)		
	Physical Design				
	Primary Key	staff_id			
	Foreign Keys	-			
	SQL Code	CREATE TABLE staff (staff_id VARCHAR2(20) NOT NULL, sname VARCHAR2(40) NOT NULL, srole VARCHAR2(40) NOT NULL, email_id VARCHAR2(40), phone_number INTEGER NOT NULL, spassword VARCHAR2(40) NOT NULL, address VARCHAR2(40)NOT NULL, dob DATE NOT NULL, PRIMARY KEY(staff_id));			
	Count of records in the table	30			

	Name of the table	Menu_item			
	Description	Items which are served in the restaurant upon request			
	Attribute	Description	Type	Examples of values	Notes
	menu_item_id	Unique menu item number	Varchar	M10000, M10001	
	menu_item_name	Name of the item which is available	Char	Pasta, pizza	
	menu_item_category	The category of the item	Char	Food, Bar	
	menu_item_price	Price if the item per serving	Integer	10\$, 20\$	

	Functional dependencies	inventory_item_id -> inventory_item_name, inventory_item_quantity	
	Candidate keys	inventory_item_id	
	Normalization		
	1NF	Yes	No arrays and repeated values nor any repeated column names
	2NF	Yes	No partial dependencies included
	3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key
	BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation X->Y, X is a super key of relation R)
	Physical Design		
	Primary Key	inventory_item_id	
	Foreign Keys	-	
	SQL Code	CREATE TABLE inventory (inventory_item_id VARCHAR2(40) NOT NULL, iname VARCHAR2(20) NOT NULL, quantity INTEGER, PRIMARY KEY(inventory_item_id));	
	Count of records in the table	20	

	<i>Name of the table</i>	<i>Schedule</i>			
	Description	The date and timing when the staff works			
	Attribute	Description	Type	Examples of values	Notes
	staff_id	Unique Staff id	Varchar	S10000,S10001 ...	
	schedule_date	Date associated with the staff working days	Date	11/02/2021	
	start_time	The intime when the staff should start working	Time	13:23:44	
	out_time	The out time when the staff can stop working	Time	13:23:44	
	Functional Dependencies and Keys				
	Functional dependencies	staff_id, date -> date, start_time, out_time			
	Candidate keys	staff_id, date			
	Normalization				

1NF	Yes	No arrays and repeated values nor any repeated column names
2NF	Yes	No partial dependencies included
3NF	Yes	No non-prime attribute is transitively dependent on the Candidate key
BCNF	Yes	because all attributes depend only on the key (For trivial dependency in the relation $X \rightarrow Y$, X is a super key of relation R)
Physical Design		
Primary Key	staff_id, date	
Foreign Keys	staff_id	
SQL Code	CREATE TABLE schedule (staff_id VARCHAR2(40) NOT NULL, schedule_date DATE NOT NULL, start_time TIMESTAMP NOT NULL, out_time TIMESTAMP NOT NULL, PRIMARY KEY (staff_id, schedule_date), CONSTRAINT FK_staffschedule FOREIGN KEY (staff_id) REFERENCES staff(staff_id));	
Count of records in the table	30	

	<i>Name of the table</i>	<i>Manages</i>			
	Description	Relationship between staff and inventory			
	Attribute	Description	Type	Examples of values	Notes
	staff_id	Unique Staff id	Varchar	S10000,S10001 ...	
	inventory_item_id	Unique inventory item id	Varchar	I10000,I10001 ...	
	Functional Dependencies and Keys				
	Functional dependencies				
	Candidate keys	staff_id, inventory_item_id			
	Normalization				
	1NF				
	2NF				
	3NF				
	BCNF				
	Physical Design				
	Primary Key	staff_id, inventory_item_id			

Foreign Keys	staff_id, inventory_item_id
SQL Code	CREATE TABLE manages (inventory_item_id VARCHAR2(40) NOT NULL, staff_id VARCHAR2(40) NOT NULL, PRIMARY KEY(inventory_item_id, staff_id), CONSTRAINT FK_inventorymanages FOREIGN KEY (inventory_item_id) REFERENCES inventory(inventory_item_id), CONSTRAINT FK_staffmanages FOREIGN KEY (staff_id) REFERENCES staff(staff_id));
Count of records in the table	30

	Name of the table	Takes			
	Description	Relationship between order and staff			
	Attribute	Description	Type	Examples of values	Notes
	staff_id	Unique Staff id	Varchar	S10000,S10001 ...	
	order_id	Unique order id	Varchar	O10000,O1000 1...	
	Functional Dependencies and Keys				
	Functional dependencies				
	Candidate keys	staff_id, order_id			
	Normalization				
	1NF				
	2NF				
	3NF				
	BCNF				
	Physical Design				
	Primary Key	staff_id, order_id			
	Foreign Keys	staff_id, order_id			

SQL Code	CREATE TABLE takes (staff_id VARCHAR2(40) NOT NULL, order_id VARCHAR2(40) NOT NULL, PRIMARY KEY(staff_id, order_id), CONSTRAINT FK_stafftakes FOREIGN KEY (staff_id) REFERENCES staff(staff_id), CONSTRAINT FK_ordertakes FOREIGN KEY (order_id) REFERENCES orders(order_id));
Count of records in the table	30

	<i>Name of the table</i>	<i>Depends</i>			
	Description	Relationship between order and menu items			
	Attribute	Description	Type	Examples of values	Notes
	order_id	Unique order id	Varchar	O10000,O10001...	
	menu_item_id	Unique menu item id	Varchar	M10000,M10001...	
	Functional Dependencies and Keys				
	Functional dependencies				
	Candidate keys	order_id, menu_item_id			
	Normalization				
	1NF				
	2NF				
	3NF				
	BCNF				
	Physical Design				
	Primary Key	order_id, menu_item_id			
	Foreign Keys	order_id, menu_item_id			




SQL Code	CREATE TABLE depends (menu_item_id VARCHAR2(40) NOT NULL, order_id VARCHAR2(40) NOT NULL, PRIMARY KEY(menu_item_id, order_id), CONSTRAINT FK_menuitemdepends FOREIGN KEY (menu_item_id) REFERENCES menu_item(menu_item_id), CONSTRAINT FK_ordersdepends FOREIGN KEY (order_id) REFERENCES orders(order_id));
Count of records in the table	30

MILESTONE 6: SQL QUERIES AND

Query 1									
English version	Display all the staff who came to work on 1st Nov 2021(Staff schedule support)								
Source for the query need in the SRS document	SRS document, Page 5, 2.2, 6th point								
SQL sentence	SELECT staff.sname FROM schedule INNER JOIN staff ON schedule.staff_id=staff.staff_id WHERE schedule.schedule_date = '01-Nov-21' ;								
Example of returned rows (cropped screen caption)	<table><thead><tr><th></th><th>SNAME</th></tr></thead><tbody><tr><td>1</td><td>Chris Ibbotson</td></tr><tr><td>2</td><td>Mario Willimont</td></tr><tr><td>3</td><td>Charlton Treske</td></tr></tbody></table>		SNAME	1	Chris Ibbotson	2	Mario Willimont	3	Charlton Treske
	SNAME								
1	Chris Ibbotson								
2	Mario Willimont								
3	Charlton Treske								

Query 2													
English version	Display all the names and phone numbers of customers to whom text message be sent regarding their reservation												
Source for the query need in the SRS document	SRS document, Page 33, Test case ID: 8.1.15												
SQL sentence	SELECT cname, phone_number FROM customer WHERE phone_number IS NOT NULL ;												
Example of returned rows (cropped screen caption)	<table><tr><th></th><th>↕ CNAME</th><th>🔍 ↕ PHONE_NUMBER</th></tr><tr><td>1</td><td>Hortensia Piercy</td><td>361413130</td></tr><tr><td>2</td><td>Marven Quigley</td><td>4804208461</td></tr><tr><td>3</td><td>Clywd Preist</td><td>6102869737</td></tr></table>		↕ CNAME	🔍 ↕ PHONE_NUMBER	1	Hortensia Piercy	361413130	2	Marven Quigley	4804208461	3	Clywd Preist	6102869737
	↕ CNAME	🔍 ↕ PHONE_NUMBER											
1	Hortensia Piercy	361413130											
2	Marven Quigley	4804208461											
3	Clywd Preist	6102869737											







Query 3	
English version	Display the most ordered items and which category do they belong to
Source for the query need in the SRS document	
SQL sentence	SELECT item_name, COUNT(*), item_type FROM orders GROUP BY item_name, item_type HAVING COUNT(*) > 2

Example of returned rows (cropped screen caption)			
	 ITEM_N...	 COUNT(*)	 ITEM_TYPE
	1 Hot Wings	3 Sides	
	2 Jug of Beer	3 Bar	

Query 4													
English version	Display the names of the customers who gave gratuities used a card to do the payment												
Source for the query need in the SRS document													
SQL sentence	SELECT DISTINCT C.cname FROM transactions T RIGHT JOIN customer C ON T.customer_id=C.customer_id WHERE gratuities IS NOT NULL AND pay_cash IS NULL;												
Example of returned rows (cropped screen caption)	<table><tr><th></th><th>CNAME</th><th></th></tr><tr><td>1</td><td>Seline Oldknow</td><td></td></tr><tr><td>2</td><td>Findlay Laidler</td><td></td></tr><tr><td>3</td><td>Raffaello Hebburn</td><td></td></tr></table>		CNAME		1	Seline Oldknow		2	Findlay Laidler		3	Raffaello Hebburn	
	CNAME												
1	Seline Oldknow												
2	Findlay Laidler												
3	Raffaello Hebburn												

Query 5	
English version	Display the names and dob of the servers who took Bar orders (To check staff age is above 18)and verified the card payments done by jbc cards
Source for the query need in the SRS document	
SQL sentence	SELECT DISTINCT S.sname, S.dob FROM staff S, transactions T, orders O WHERE S.staff_id=T.staff_id AND O.order_id=T.order_id AND T.card_type = 'jbc' UNION SELECT DISTINCT S.sname, S.dob FROM staff S, transactions T, orders O WHERE S.staff_id=T.staff_id AND O.order_id=T.order_id AND O.item_type = 'Bar';

Example of returned rows (cropped screen caption)		⚡ SNAME	⚡ DOB
	1	Amalie Jagger	25-JAN-79
	2	Conant Casillas	27-AUG-73
	3	Urbano Pulman	09-DEC-82

Query 6													
English version	Display the roles and the names of the staff members who can edit the inventory												
Source for the query need in the SRS document													
SQL sentence	SELECT DISTINCT S.srole, S.sname FROM staff S, inventory I, manages M WHERE S.staff_id=M.staff_id AND I.inventory_item_id=M.inventory_item_id;												
Example of returned rows (cropped screen caption)													
	<table><tr><td></td><td> SROLE</td><td> SNAME</td></tr><tr><td>1</td><td>Chief</td><td>Fulton Shepperd</td></tr><tr><td>2</td><td>Kitchen staff</td><td>Eleen Hullyer</td></tr><tr><td>3</td><td>Chief</td><td>Beulah Tilson</td></tr></table>		 SROLE	 SNAME	1	Chief	Fulton Shepperd	2	Kitchen staff	Eleen Hullyer	3	Chief	Beulah Tilson
		 SROLE	 SNAME										
	1	Chief	Fulton Shepperd										
	2	Kitchen staff	Eleen Hullyer										
3	Chief	Beulah Tilson											

Query 7									
English version	Display the names of the customers who canceled they orders								
Source for the query need in the SRS document									
SQL sentence	SELECT DISTINCT C.cname FROM customer C WHERE C.customer_id IN (SELECT O.customer_id FROM orders O WHERE O.status='Cancelled');								
Example of returned rows (cropped screen caption)	<table><tr><td></td><td>CNAME</td></tr><tr><td>1</td><td>Findlay Laidler</td></tr><tr><td>2</td><td>Raffaello Hebburn</td></tr><tr><td>3</td><td>Markos Westfield</td></tr></table>		CNAME	1	Findlay Laidler	2	Raffaello Hebburn	3	Markos Westfield
	CNAME								
1	Findlay Laidler								
2	Raffaello Hebburn								
3	Markos Westfield								

Query 8													
English version	Display the id of the table whose table status is occupied, and the order status is served. So that we can process their billing												
Source for the query need in the SRS document													
SQL sentence	SELECT T.table_id, O.status, t.table_status FROM sit_table T INNER JOIN orders O ON T.order_id=o.order_id WHERE t.table_status LIKE 'O%' AND O.status LIKE 'S%';												
Example of returned rows (cropped screen caption)	<table><tr><th></th><th>TABLE_ID</th><th>STATUS</th><th>TABLE_STATUS</th></tr><tr><td>1</td><td>T43-167-0767</td><td>Served</td><td>Occupied</td></tr><tr><td>2</td><td>T48-324-4126</td><td>Served</td><td>Occupied</td></tr></table>		TABLE_ID	STATUS	TABLE_STATUS	1	T43-167-0767	Served	Occupied	2	T48-324-4126	Served	Occupied
	TABLE_ID	STATUS	TABLE_STATUS										
1	T43-167-0767	Served	Occupied										
2	T48-324-4126	Served	Occupied										

Query 9																	
English version	Group and display the table based on their type and shape																
Source for the query need in the SRS document																	
SQL sentence	SELECT table_type, table_shape, COUNT(*) FROM sit_table GROUP BY table_type, table_shape;																
Example of returned rows (cropped screen caption)	<table><tr><th></th><th>TABLE_TYPE</th><th>TABLE_SHAPE</th><th>COUNT(*)</th></tr><tr><td>1</td><td>Large</td><td>Square</td><td>1</td></tr><tr><td>2</td><td>Large</td><td>Rectangle</td><td>6</td></tr><tr><td>3</td><td>Small</td><td>Rectangle</td><td>3</td></tr></table>		TABLE_TYPE	TABLE_SHAPE	COUNT(*)	1	Large	Square	1	2	Large	Rectangle	6	3	Small	Rectangle	3
	TABLE_TYPE	TABLE_SHAPE	COUNT(*)														
1	Large	Square	1														
2	Large	Rectangle	6														
3	Small	Rectangle	3														