MANAGEMENT CAFE

Presented by: Group 9

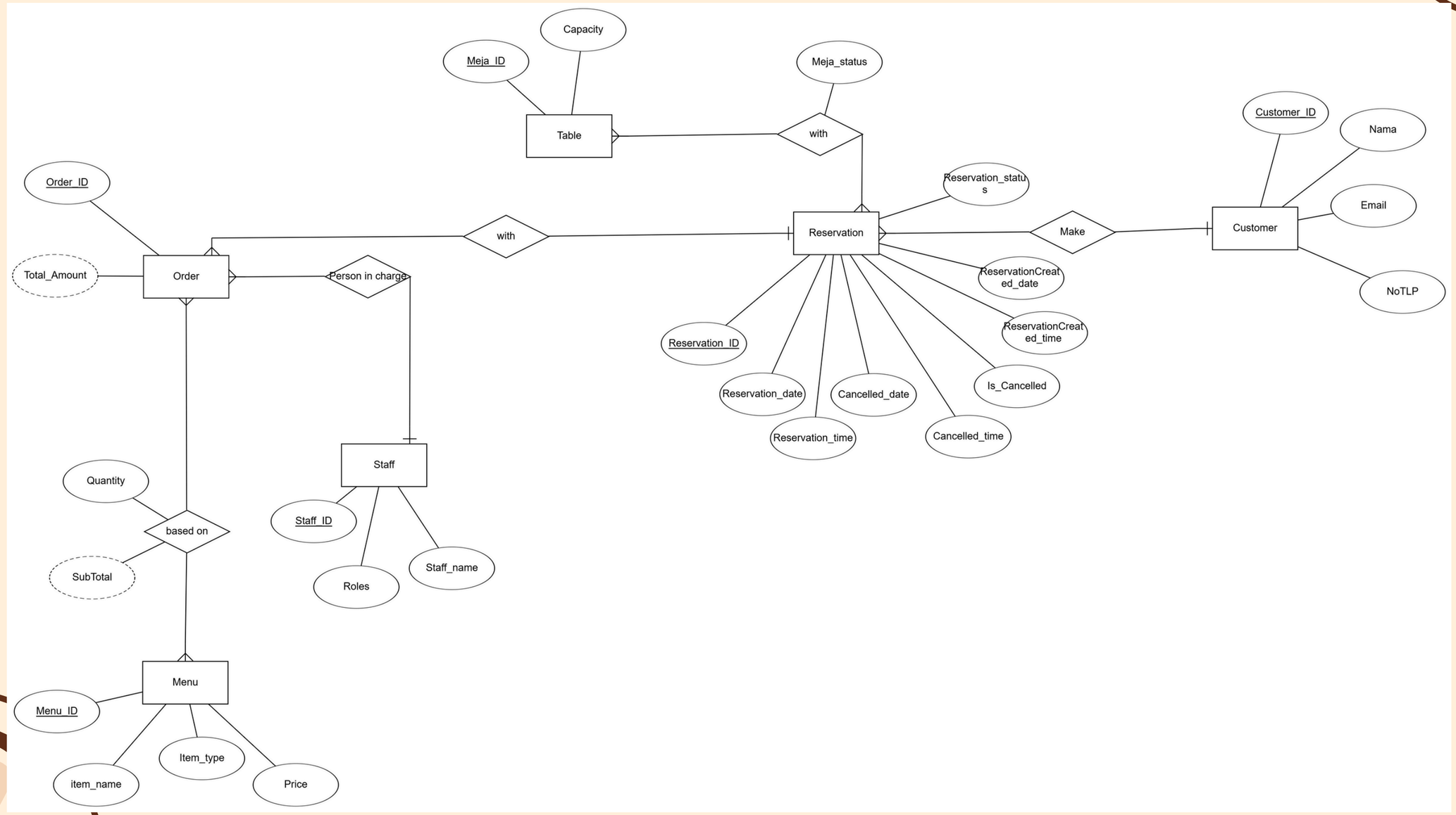
GROUP MEMBER

1. 2702320210 - Natasha Dian Mahardita
2. 2702331763 - Lavinia Nataniela Novyandi
3. 2702343523- Pazella Mutia Reflin
4. 2702208581- Nicholas Sinclair Alfianto
5. 2702360750 - Syauqina Zhafirah Nur Fakhrana
6. 2702375866 - Putri Maharani Setiawan
7. 2702235084 - Vanessa Nayla Putri

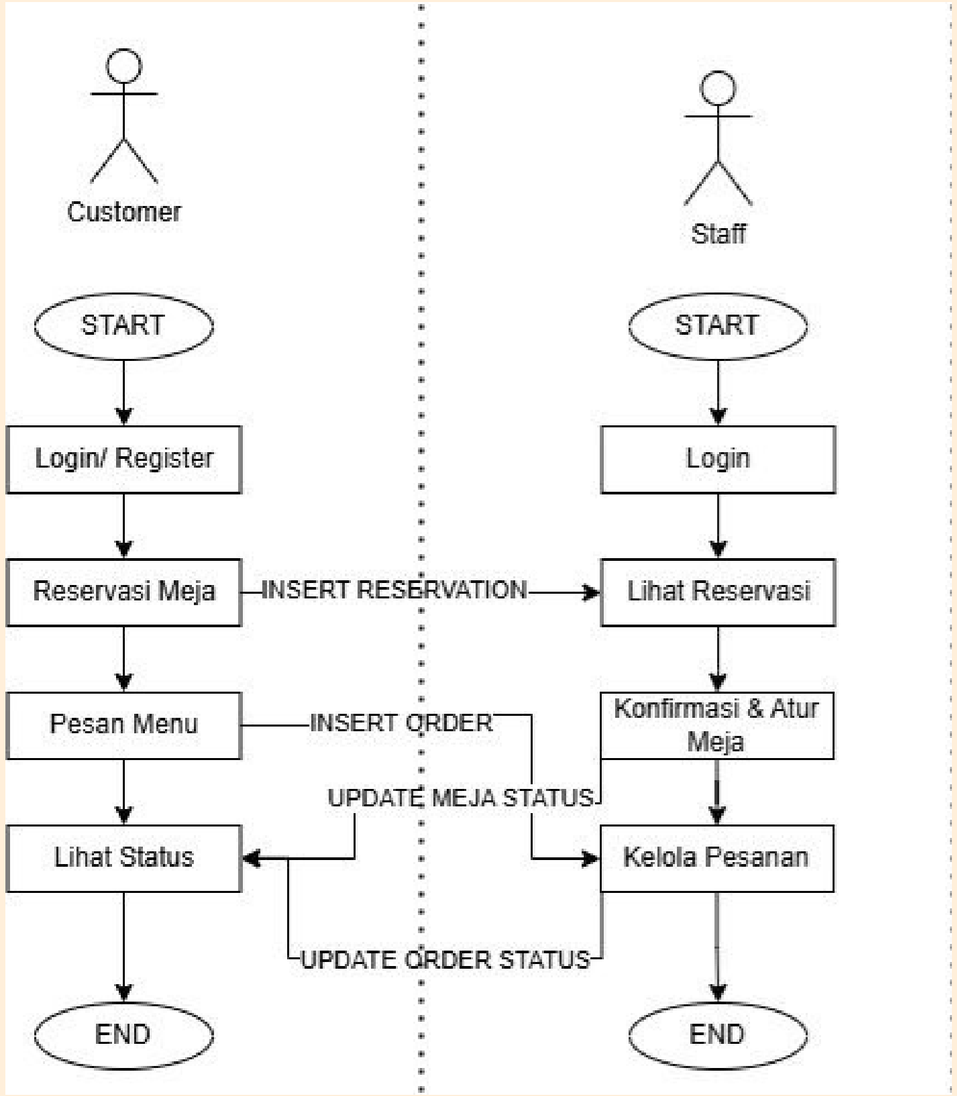
LATAR BELAKANG

Di era digital saat ini, yang mengedepankan efisiensi dan kenyamanan, banyak restoran yang masih mengandalkan cara manual dalam mengelola reservasi, pemesanan menu, dan pembayaran. Hal ini sering kali menyebabkan kesalahan dalam pencatatan, memakan waktu lama dalam proses pelayanan, serta menyulitkan pemantauan kinerja bisnis. Oleh karena itu, dibutuhkan aplikasi terintegrasi yang memungkinkan pelanggan melakukan reservasi dan pemesanan secara digital. Di sisi lain, aplikasi ini juga membantu staf restoran dalam mengoptimalkan operasional dengan sistem real-time. Fitur dashboard analitik pada aplikasi ini memungkinkan admin untuk memantau laporan dan statistik kinerja restoran, yang pada gilirannya dapat meningkatkan kepuasan pelanggan, efisiensi operasional, serta pengambilan keputusan yang lebih berbasis data.

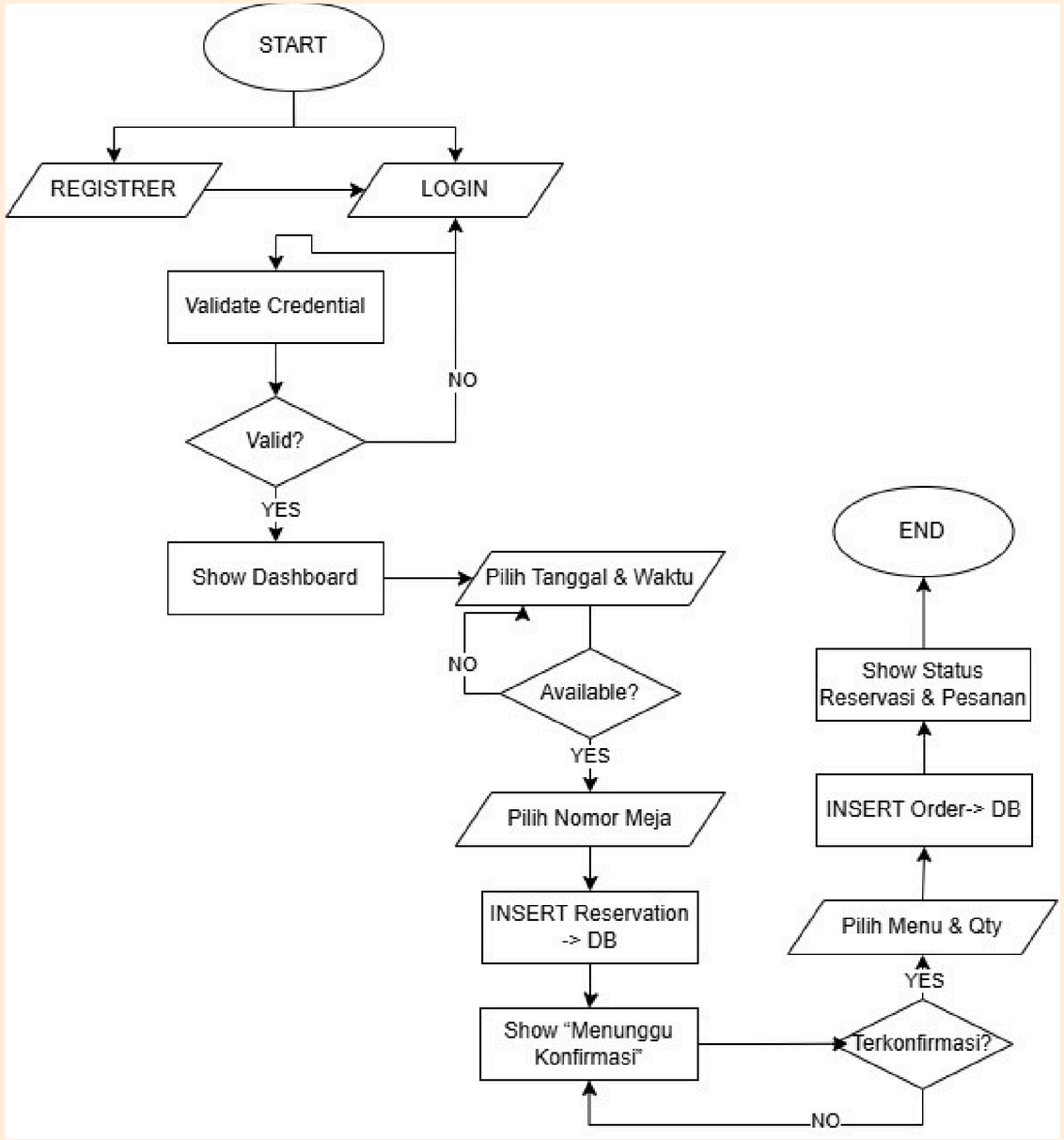
ERD


Capacity
Meja_ID
Meja_status
Table
with
Customer_ID
Nama
Order_ID
Reservation_status
Email
Reservation
Make
Customer
with
Total_Amount
Order
Person in charge
ReservationCreated_date
NoTLP
ReservationCreated_time
Reservation_ID
Reservation_date
Cancelled_date
Is_Cancelled
Reservation_time
Cancelled_time
Staff
Quantity
Staff_ID
based on
Staff_name
SubTotal
Roles
Menu
Menu_ID
Item_type
item_name
Price

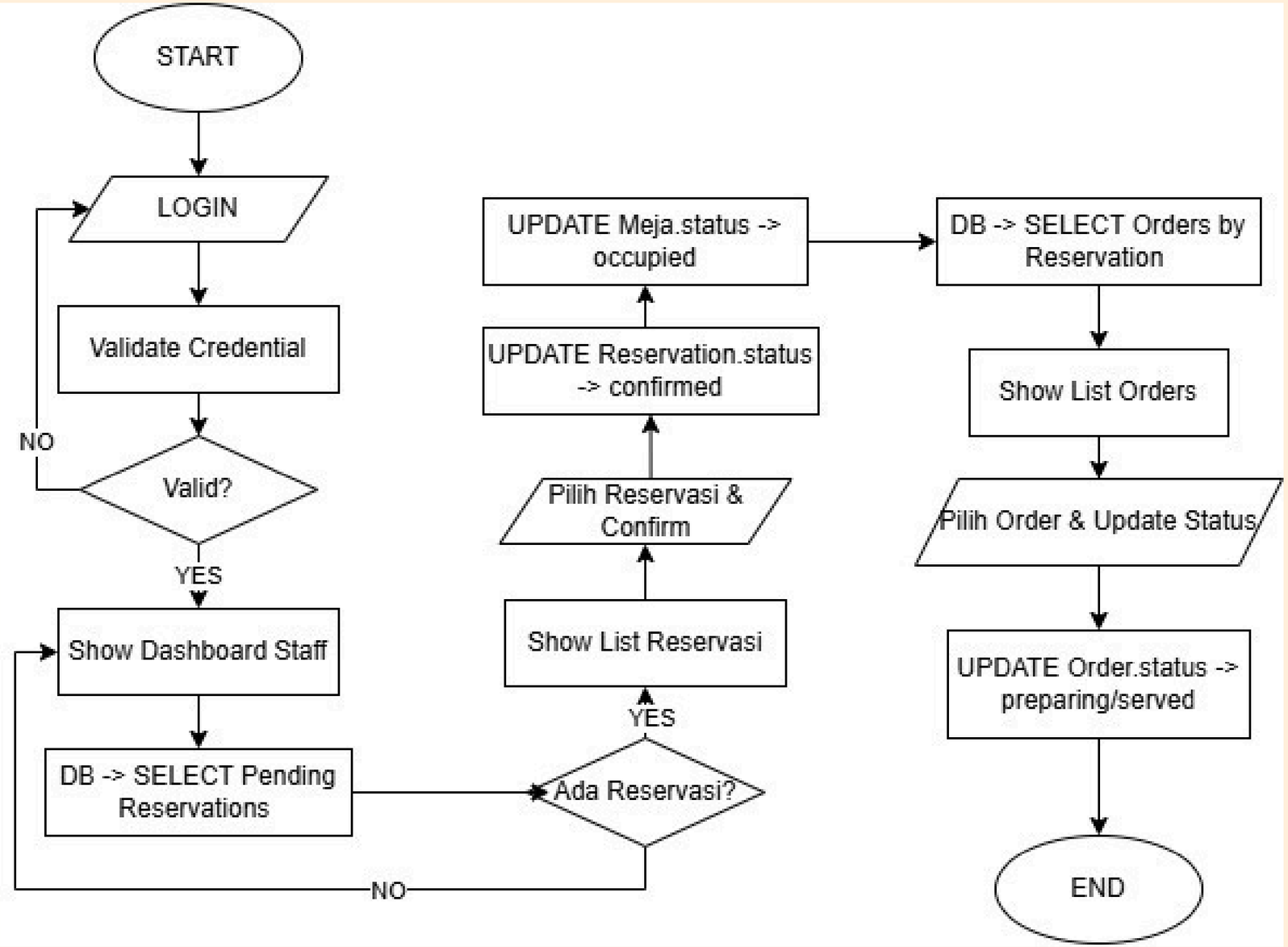
FLOWCHART GABUNGAN



FLOWCHART CUSTOMER



FLOWCHART STAFF



SCHEMA

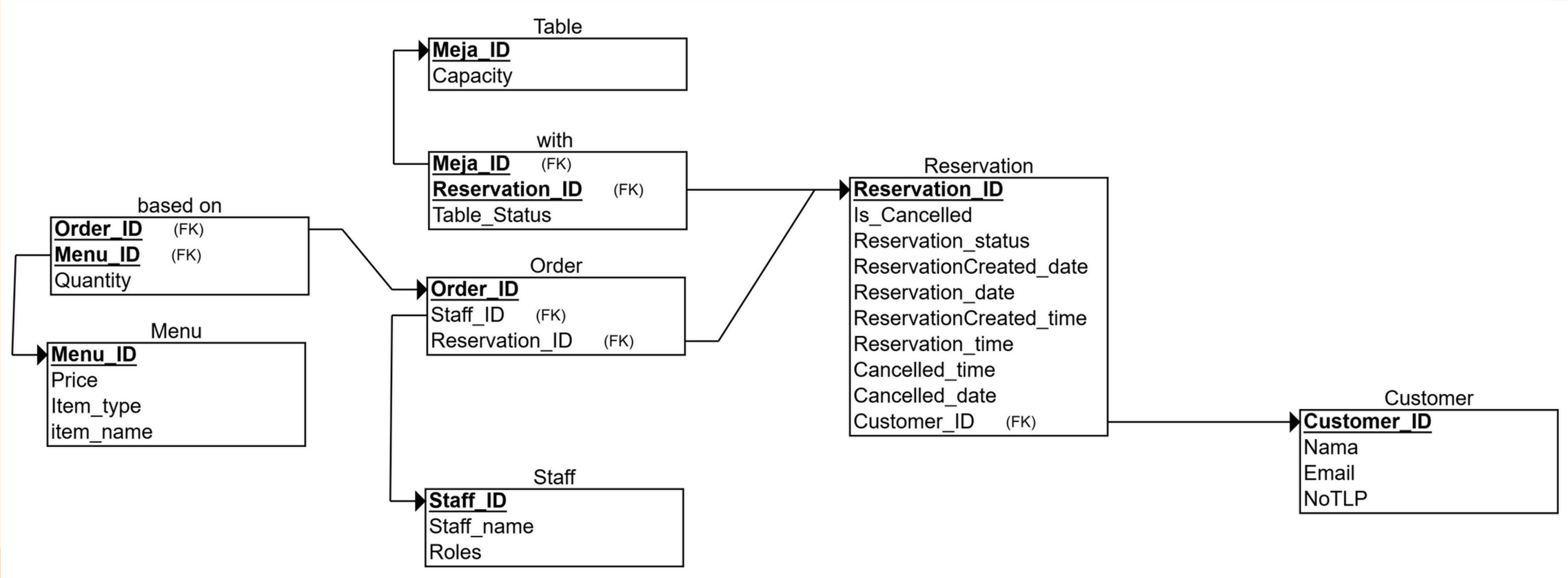
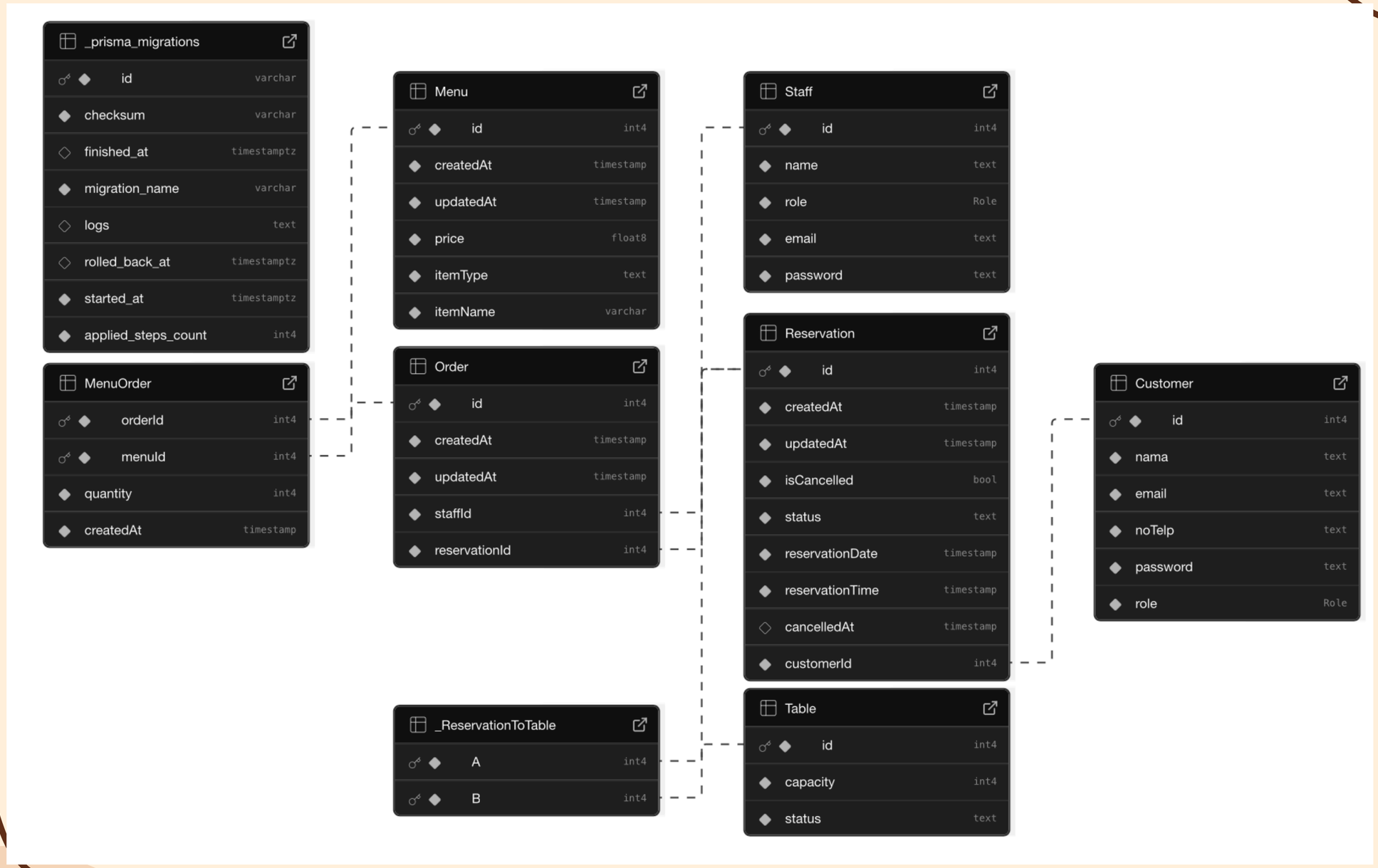


Table
Meja_ID
Capacity
with
Meja_ID (FK)
Reservation_ID (FK)
Table_Status
Reservation
Reservation_ID
Is_Cancelled
Reservation_status
ReservationCreated_date
Reservation_date
ReservationCreated_time
Reservation_time
Cancelled_time
Cancelled_date
Customer_ID (FK)
based on
Order_ID (FK)
Menu_ID (FK)
Quantity
Order
Order_ID
Staff_ID (FK)
Reservation_ID (FK)
Menu
Menu_ID
Price
Item_type
item_name
Staff
Staff_ID
Staff_name
Roles
Customer
Customer_ID
Nama
Email
NoTLP

SCHEMA FROM SUPABASE



IMPLEMENTASI DATABASE SQL

CUSTOMER

Customer_ID	Name	Email	Mobile Phone
C001	Tristan Kelley	kyleandrews@mcdaniel.com	+62 567-9078-095
C002	Joseph Morris DDS	jonathantate@moon.info	+62 259-088-443
C003	Glenn Beck	jasondavis@webster.com	+62 88-779-577
C004	Elijah Carter	ijones@hotmail.com	+62 518-5846-870
C005	Karen Lynch	sweeneyalexis@gmail.com	+62 292-1597-149

MEJA_DETAIL

Reservation_ID	Meja_ID	Meja_status
R001	MEJA00001	assigned
R002	MEJA00002	assigned
R003	MEJA00003	assigned
R004	MEJA00004	assigned
R005	MEJA00005	assigned

IMPLEMENTASI DATABASE SQL

MEJA

Meja_ID	Capacity
MEJA00001	6
MEJA00002	2
MEJA00003	2
MEJA00004	4
MEJA00005	6
MEJA00006	4
MEJA00007	6

MENU

Menu_Name	Category	Price	Menu_ID
Tubruk Coffee	Beverage	15000	M001
Iced Milk Coffee	Beverage	20000	M002
Black Coffee	Beverage	18000	M003
Pulled Tea (Teh Tarik)	Beverage	18000	M004
Sweet Iced Tea	Beverage	10000	M005

ORDER

Order_ID	Staff_ID	Reservation_ID
ORDER001	S015	R368
ORDER002	S007	R381
ORDER003	S023	R059
ORDER004	S026	R578
ORDER005	S005	R620

IMPLEMENTASI DATABASE SQL

STAFF

ID_Staff	Staff_Name	Role
S001	James Bates	Cashier
S002	Christopher Black	Waiter
S003	Nicholas Friedman	Bartender
S004	Alisha May	Dishwasher
S005	Brian Sanchez	Bartender

ORDER_DETAIL

Quantity	Menu_ID	Order_ID
2	M002	ORDER001
4	M013	ORDER001
5	M005	ORDER002
5	M018	ORDER002
1	M015	ORDER003

IMPLEMENTASI DATABASE SQL

RESERVASI

Reservation_ID	Reservation Status	IsCancelled	ReservationCreated_Date	ReservationCreated_Time	ReservationCancelled_Date	ReservationCancelled_Time	Reservation_Date	Reservation_Time	Customer_ID
R001	3	0	05/12/2024	48:56.8			5/17/2024	22:00:00	C117
R002	3	0	08/10/2024	31:59.7			8/13/2024	23:30:00	C566
R003	4	1	9/30/2024	46:41.7	10/01/2024	34:22.7	10/01/2024	06:20:00	C228
R670	2	0	10/18/2024	23:52.9			10/19/2024	20:15:00	C026
R004	4	1	09/01/2024	03:09.8	09/01/2024	10:49.0	09/08/2024	06:20:00	C367

DDL
[DOCUMENTATION QUERY]

```sql
CREATE TYPE "Role" AS ENUM ('USER', 'ADMIN');

CREATE TABLE "Customer" (
    "id" SERIAL PRIMARY KEY,
    "nama" TEXT NOT NULL,
    "email" TEXT NOT NULL UNIQUE,
    "noTelp" TEXT NOT NULL,
    "password" TEXT NOT NULL,
    "role" "Role" NOT NULL DEFAULT 'USER'
);

CREATE TABLE "Table" (
    "id" SERIAL PRIMARY KEY,
    "capacity" INTEGER NOT NULL,
    "status" TEXT NOT NULL
);

CREATE TABLE "Staff" (
    "id" SERIAL PRIMARY KEY,
    "name" TEXT NOT NULL,
    "email" TEXT NOT NULL UNIQUE,
    "password" TEXT NOT NULL,
    "role" "Role" NOT NULL DEFAULT 'ADMIN'
);

```

```sql
CREATE TABLE "Reservation" (
    "id" SERIAL PRIMARY KEY,
    "createdAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    "updatedAt" TIMESTAMP(3) NOT NULL,
    "isCancelled" BOOLEAN NOT NULL DEFAULT false,
    "status" TEXT NOT NULL,
    "reservationDate" TIMESTAMP(3) NOT NULL,
    "reservationTime" TIMESTAMP(3) NOT NULL,
    "cancelledAt" TIMESTAMP(3),
    "customerId" INTEGER NOT NULL,
    FOREIGN KEY ("customerId") REFERENCES "Customer"("id")
);

CREATE TABLE "Order" (
    "id" SERIAL PRIMARY KEY,
    "createdAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    "updatedAt" TIMESTAMP(3) NOT NULL,
    "staffId" INTEGER NOT NULL,
    "reservationId" INTEGER NOT NULL,
    FOREIGN KEY ("staffId") REFERENCES "Staff"("id"),
    FOREIGN KEY ("reservationId") REFERENCES "Reservation"("id")
);

CREATE TABLE "Menu" (
    "id" SERIAL PRIMARY KEY,
    "createdAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    "updatedAt" TIMESTAMP(3) NOT NULL,
    "price" DOUBLE PRECISION NOT NULL,
    "itemType" TEXT NOT NULL,
    "itemName" VARCHAR(255) NOT NULL
);
```

DDL
[DOCUMENTATION QUERY]

```sql
CREATE TABLE "MenuOrder" (
    "orderId" INTEGER NOT NULL,
    "menuId" INTEGER NOT NULL,
    "quantity" INTEGER NOT NULL,
    "createdAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY ("orderId", "menuId"),
    FOREIGN KEY ("orderId") REFERENCES "Order"("id"),
    FOREIGN KEY ("menuId") REFERENCES "Menu"("id")
);
```

```sql
CREATE TABLE "_ReservationToTable" (
    "A" INTEGER NOT NULL,
    "B" INTEGER NOT NULL,
    FOREIGN KEY ("A") REFERENCES "Reservation"("id"),
    FOREIGN KEY ("B") REFERENCES "Table"("id"),
    UNIQUE ("A", "B")
);
```

```sql
CREATE INDEX "Reservation_customerId_idx" ON "Reservation"("customerId");
CREATE INDEX "Order_staffId_idx" ON "Order"("staffId");
CREATE INDEX "Order_reservationId_idx" ON "Order"("reservationId");
CREATE INDEX "MenuOrder_orderId_idx" ON "MenuOrder"("orderId");
CREATE INDEX "MenuOrder_menuId_idx" ON "MenuOrder"("menuId");
```

DML
[QUERY SQL]

1.Get all customers with their reservations

```
SELECT c.id, c.nama, c.email, c.no_telp,
r.id as reservation_id, r.reservation_date, r.reservation_time,
r.status, r.is_cancelled
FROM "Customer" c
LEFT JOIN "Reservation" r ON c.id = r.customer_id
ORDER BY c.id, r.reservation_date DESC;
```

2. Get active reservations with customer details

```
SELECT r.id as reservation_id, r.reservation_date, r.reservation_time,
    r.status, c.id as customer_id, c.nama, c.email, c.no_telp
FROM "Reservation" r
JOIN "Customer" c ON r.customer_id = c.id
WHERE r.is_cancelled = false
ORDER BY r.reservation_date, r.reservation_time;
```

DML
[QUERY SQL]

3. Get reservations with assigned tables

```
SELECT r.id as reservation_id, r.reservation_date, r.reservation_time,
    r.status, c.nama as customer_name, c.email,
    t.id as table_id, t.capacity, t.status as table_status
FROM "Reservation" r
JOIN "Customer" c ON r.customer_id = c.id
JOIN "_ReservationToTable" rt ON r.id = rt."A"
JOIN "Table" t ON t.id = rt."B"
ORDER BY r.reservation_date, r.reservation_time;
```

4. Get all orders with staff and reservation details

```
SELECT o.id as order_id, o.created_at,
    s.id as staff_id, s.name as staff_name, s.role,
    r.id as reservation_id, r.reservation_date, r.reservation_time,
    c.nama as customer_name
FROM "Order" o
JOIN "Staff" s ON o.staff_id = s.id
JOIN "Reservation" r ON o.reservation_id = r.id
JOIN "Customer" c ON r.customer_id = c.id
ORDER BY o.created_at DESC;
```

DML
[QUERY SQL]

5. Get order details with menu items

```
SELECT o.id as order_id, o.created_at,
    s.name as staff_name,
    c.nama as customer_name,
    m.item_name, m.item_type, m.price,
    mo.quantity, (m.price * mo.quantity) as item_total
FROM "Order" o
JOIN "Staff" s ON o.staff_id = s.id
JOIN "Reservation" r ON o.reservation_id = r.id
JOIN "Customer" c ON r.customer_id = c.id
JOIN "MenuOrder" mo ON o.id = mo.order_id
JOIN "Menu" m ON mo.menu_id = m.id
ORDER BY o.id, m.item_type, m.item_name;
```

6. Get total sales by menu item

```
SELECT m.id, m.item_name, m.item_type, m.price,
    SUM(mo.quantity) as total_ordered,
    SUM(m.price * mo.quantity) as total_sales
FROM "Menu" m
JOIN "MenuOrder" mo ON m.id = mo.menu_id
GROUP BY m.id, m.item_name, m.item_type, m.price
ORDER BY total_sales DESC;
```

DML
[QUERY SQL]

7. Get total sales by day

```
SELECT DATE(o.created_at) as order_date,
    COUNT(DISTINCT o.id) as total_orders,
    SUM(m.price * mo.quantity) as daily_sales
FROM "Order" o
JOIN "MenuOrder" mo ON o.id = mo.order_id
JOIN "Menu" m ON mo.menu_id = m.id
GROUP BY DATE(o.created_at)
ORDER BY order_date DESC;
```

8. Find available tables for a specific time

```
SELECT t.id, t.capacity, t.status
FROM "Table" t
WHERE t.id NOT IN (
  SELECT rt."B"
  FROM "_ReservationToTable" rt
  JOIN "Reservation" r ON rt."A" = r.id
  WHERE r.reservation_date = '2025-04-26'
  AND r.reservation_time BETWEEN '18:00:00'::time AND '20:00:00'::time
  AND r.is_cancelled = false
)
ORDER BY t.capacity;
```

DML
[QUERY SQL]

9. Get staff performance (orders handled)

```
SELECT s.id, s.name, s.role,
    COUNT(o.id) as total_orders,
    COUNT(DISTINCT r.customer_id) as customers_served
FROM "Staff" s
LEFT JOIN "Order" o ON s.id = o.staff_id
LEFT JOIN "Reservation" r ON o.reservation_id = r.id
GROUP BY s.id, s.name, s.role
ORDER BY total_orders DESC;
```

10. Get customer reservation history with orders

```
SELECT c.id, c.nama, c.email,
    COUNT(DISTINCT r.id) as total_reservations,
    COUNT(DISTINCT o.id) as total_orders,
    SUM(m.price * mo.quantity) as total_spent
FROM "Customer" c
LEFT JOIN "Reservation" r ON c.id = r.customer_id
LEFT JOIN "Order" o ON r.id = o.reservation_id
LEFT JOIN "MenuOrder" mo ON o.id = mo.order_id
LEFT JOIN "Menu" m ON mo.menu_id = m.id
GROUP BY c.id, c.nama, c.email
ORDER BY total_spent DESC NULLS LAST;
```

DML
[QUERY SQL]

11. Get popular menu items by quantity ordered

```
SELECT m.item_name, m.item_type, m.price,
    SUM(mo.quantity) as times_ordered
FROM "Menu" m
JOIN "MenuOrder" mo ON m.id = mo.menu_id
GROUP BY m.id, m.item_name, m.item_type, m.price
ORDER BY times_ordered DESC
LIMIT 10;
```

12. Get cancelled reservations with reason

```
SELECT r.id, r.reservation_date, r.reservation_time,
    r.cancelled_at, c.nama as customer_name, c.email
FROM "Reservation" r
JOIN "Customer" c ON r.customer_id = c.id
WHERE r.is_cancelled = true
ORDER BY r.cancelled_at DESC;
```

DATA DICTIONARY

Customer

Stores information about restaurant customers.

Field	Type	Description	Constraints
id	Int	Unique customer identifier	Primary key, auto-increment
nama	String	Customer's full name	Required
email	String	Customer's email address	Required, unique
noTelp	String	Customer's phone number	Required
password	String	Customer's encrypted password	Required
role	Role enum	User access level	Default: USER

Table

Represents restaurant tables available for reservation.

Field	Type	Description	Constraints
id	Int	Unique table identifier	Primary key, auto-increment
capacity	Int	Number of seats at table	Required
status	String	Current status of table	Required

Reservation

Tracks customer reservations at the restaurant.

Field	Type	Description	Constraints
id	Int	Unique reservation identifier	Primary key, auto-increment
createdAt	DateTime	When reservation was created	Default: current time
updatedAt	DateTime	When reservation was last updated	Auto-updated
isCancelled	Boolean	Reservation cancellation status	Default: false
status	String	Current reservation status	Required
reservationDate	DateTime	Date of reservation	Required
reservationTime	DateTime	Time of reservation	Required
cancelledAt	DateTime	When reservation was cancelled	Optional
customerId	Int	ID of customer who made reservation	Foreign key to Customer

DATA DICTIONARY

Staff

Stores information about restaurant staff members.

Field	Type	Description	Constraints
id	Int	Unique staff identifier	Primary key, auto-increment
name	String	Staff member's name	Required
email	String	Staff email address	Required, unique
password	String	Staff encrypted password	Required
role	Role enum	Staff access level	Default: ADMIN

Menu

Contains details of items available on the restaurant menu.

Field	Type	Description	Constraints
id	Int	Unique menu item identifier	Primary key, auto-increment
createdAt	DateTime	When menu item was added	Default: current time
updatedAt	DateTime	When menu item was last updated	Auto-updated
price	Float	Price of menu item	Required
itemType	String	Category of menu item	Required
itemName	String	Name of menu item	Required, max 255 chars

Order

Tracks food and drink orders associated with reservations.

Field	Type	Description	Constraints
id	Int	Unique order identifier	Primary key, auto-increment
createdAt	DateTime	When order was created	Default: current time
updatedAt	DateTime	When order was last updated	Auto-updated
staffId	Int	ID of staff member who processed order	Foreign key to Staff
reservationId	Int	ID of associated reservation	Foreign key to Reservation

DATA DICTIONARY

MenuOrder

Junction table for many-to-many relationship between Orders and Menu items.

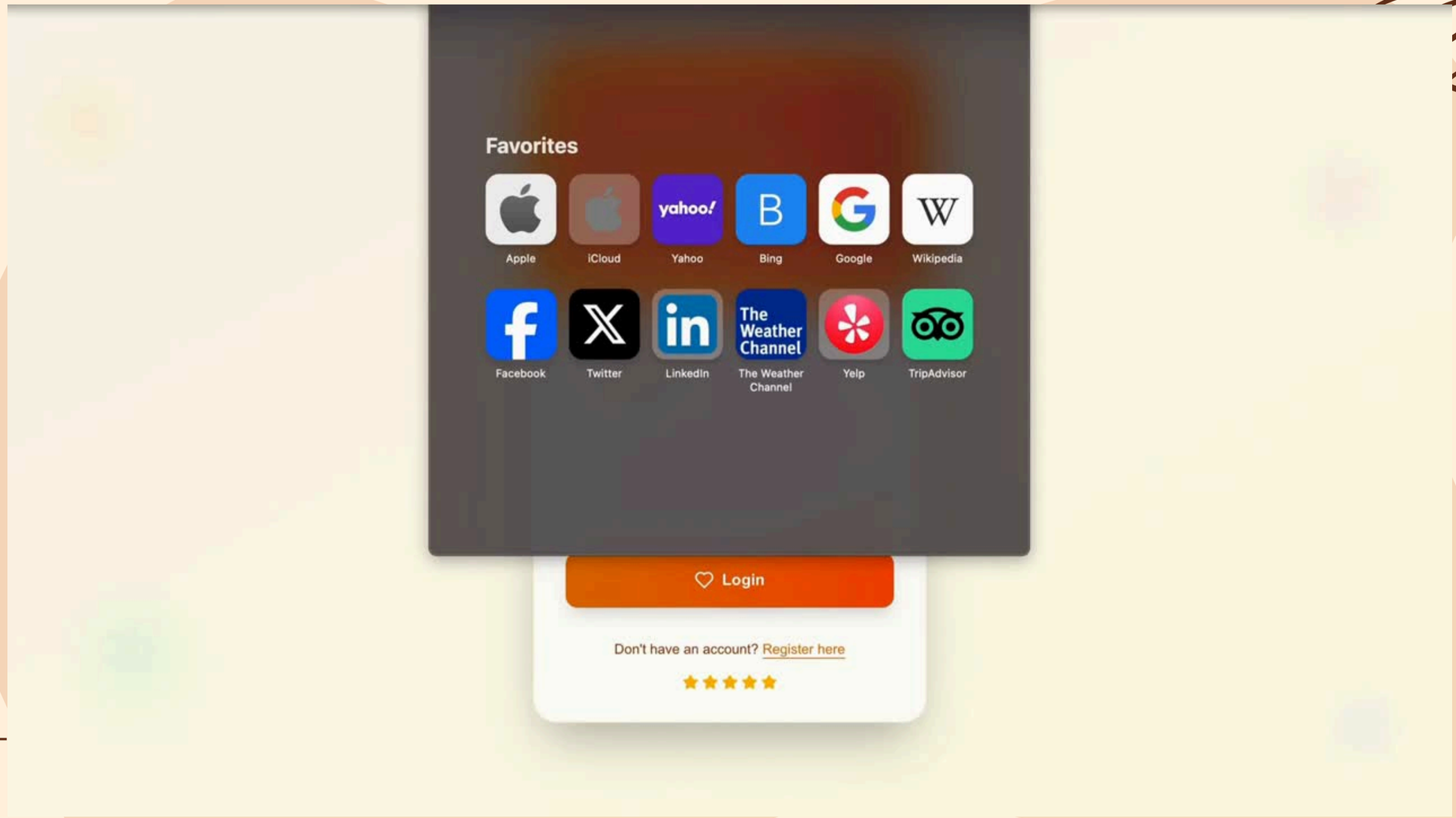
Field	Type	Description	Constraints
orderId	Int	Order identifier	Primary key component, foreign key to Order
menuId	Int	Menu item identifier	Primary key component, foreign key to Menu
quantity	Int	Number of this menu item ordered	Required
createdAt	DateTime	When the menu item was ordered	Default: current time

Role (Enum)

Defines user access levels in the system.

Value	Description
USER	Regular customer access
ADMIN	Administrative staff access

WEB INTERFACE



LINK GITHUB: HTTPS://GITHUB.COM/LAVINIANATANIELA05/CAFEMANAGEMENT
HTTPS://CAFEMANAGEMENTSYSTEM-NU.VERCEL.APP

THANK YOU