|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATABASE MANAGEMENT OF SWIGGY | | | | |
|  |  | | |  |
| MySQL CASE STUDY | | | | |
|  | | By Onkar Sawant |  | |



# Introduction

Story: Imagine you're creating a system like Swiggy, where people can order food from restaurants and get it delivered to their doorstep.

**Tables:**

1. Users: These are the people using the app. They have names, email addresses, and passwords to log in.
2. Restaurants: These are the places where people can order food from. Each restaurant has a name and serves a particular type of cuisine.
3. Orders: Whenever someone places an order, it's recorded here. It keeps track of who ordered, from which restaurant, what was ordered, how much it cost, and when it was ordered.
4. Delivery Partners: These are the folks who pick up the food from the restaurant and deliver it to the customer. Each delivery partner has a name and an ID.
5. Order Details: This keeps track of the items in each order. For example, if someone orders a pizza and a drink, this table notes down those details.
6. Food Items: This is like a menu of all the food available. It has the names of items like "Non-veg Pizza" or "Choco Lava Cake" and specifies if they're vegetarian or non-vegetarian.
7. Menu: Each restaurant has a menu that lists all the food items they offer along with their prices.

**What Happens:**

* Users log in and place orders from Restaurants.
* Each Order is assigned a Delivery Partner who delivers the food.
* The Order Details table records what exactly was ordered in each order.
* The Menu table links restaurants with their respective food items and prices.

**Why It Matters:**

* This system helps manage orders efficiently, ensuring that the right food gets to the right people at the right time.
* It keeps track of what's available on the menu, making it easy for users to browse and order their favorite dishes.
* By recording feedback and ratings, it helps improve the overall experience for users, restaurants, and delivery partners alike.

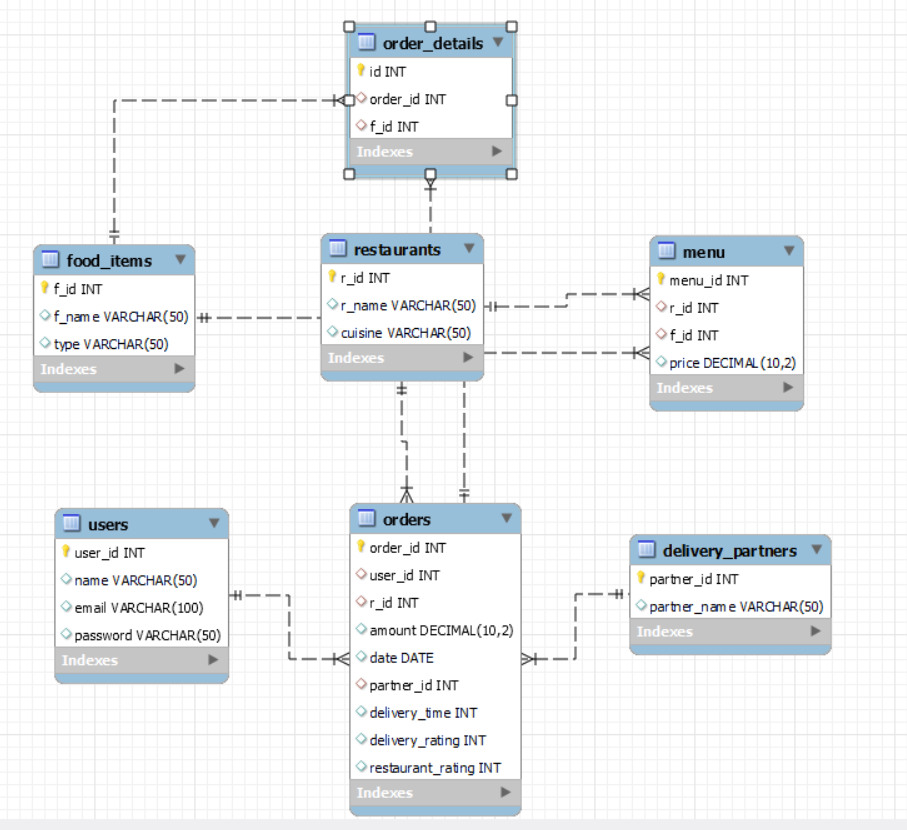
**Entities:**

1. Users: Attributes include user\_id (PK), name, email, and password.
2. Restaurants: Attributes include r\_id (PK), r\_name, and cuisine.
3. Orders: Attributes include order\_id (PK), user\_id (FK), r\_id (FK), amount, date, partner\_id (FK), delivery\_time, delivery\_rating, and restaurant\_rating.
4. Delivery\_Partners: Attributes include partner\_id (PK) and partner\_name.
5. Order\_Details: Attributes include id (PK), order\_id (FK), and f\_id (FK).
6. Food\_Items: Attributes include f\_id (PK), f\_name, and type.
7. Menu: Attributes include menu\_id (PK), r\_id (FK), f\_id (FK), and price.

**Relationships:**

1. User-Order: One-to-many relationship between Users and Orders based on user\_id.
2. Restaurant-Order: One-to-many relationship between Restaurants and Orders based on r\_id.
3. Order-Delivery\_Partner: One-to-many relationship between Orders and Delivery\_Partners based on partner\_id.
4. Order-Order\_Details: One-to-many relationship between Orders and Order\_Details based on order\_id.
5. Order\_Detail-Food\_Item: One-to-one relationship between Order\_Details and Food\_Items based on f\_id.
6. Restaurant-Menu: One-to-many relationship between Restaurants and Menu based on r\_id.
7. Food\_Item-Menu: One-to-many relationship between Food\_Items and Menu based on f\_id

|  |
| --- |
| E-R DIAGRAM FOR CASE STUDY |



|  |
| --- |
| COMMANDS |

* create database swiggy;
* use swiggy;

CREATE TABLE users (

* user\_id INT PRIMARY KEY,
* name VARCHAR(50),
* email VARCHAR(100),
* password VARCHAR(50)
* );

INSERT INTO users (user\_id, name, email, password) VALUES

* (1, 'Nitish', 'nitish@gmail.com', 'p252h'),
* (2, 'Khushboo', 'khushboo@gmail.com', 'hxn9b'),
* (3, 'Vartika', 'vartika@gmail.com', '9hu7j'),
* (4, 'Ankit', 'ankit@gmail.com', 'lkko3'),
* (5, 'Neha', 'neha@gmail.com', '3i7qm'),
* (6, 'Anupama', 'anupama@gmail.com', '46rdw2'),
* (7, 'Rishabh', 'rishabh@gmail.com', '4sw123');

CREATE TABLE restaurants (

* r\_id INT PRIMARY KEY,
* r\_name VARCHAR(50),
* cuisine VARCHAR(50)
* );

INSERT INTO restaurants (r\_id, r\_name, cuisine) VALUES

* (1, 'dominos', 'Italian'),
* (2, 'kfc', 'American'),
* (3, 'box8', 'North Indian'),
* (4, 'Dosa Plaza', 'South Indian'),
* (5, 'China Town', 'Chinese');

CREATE TABLE orders (

* order\_id INT PRIMARY KEY,
* user\_id INT,
* r\_id INT,
* foreign key(user\_id) references users(user\_id),
* foreign key(r\_id) references restaurants(r\_id),
* amount DECIMAL(10, 2),
* date DATE,
* partner\_id INT,
* foreign key(partner\_id) references delivery\_partners(partner\_id),
* delivery\_time INT,
* delivery\_rating INT,
* restaurant\_rating INT
* );

INSERT INTO orders (order\_id, user\_id, r\_id, amount, date, partner\_id, delivery\_time, delivery\_rating, restaurant\_rating)

* VALUES
* (1001, 1, 1, 550.00, '2022-05-10', 1, 25, 5, 3),
* (1002, 1, 2, 415.00, '2022-05-26', 1, 19, 5, 2),
* (1003, 1, 3, 240.00, '2022-06-15', 5, 29, 4, NULL),
* (1004, 1, 3, 240.00, '2022-06-29', 4, 42, 3, 5),
* (1005, 1, 3, 220.00, '2022-07-10', 1, 58, 1, 4),
* (1006, 2, 1, 950.00, '2022-06-10', 2, 16, 5, NULL),
* (1007, 2, 2, 530.00, '2022-06-23', 3, 60, 1, 5),
* (1008, 2, 3, 240.00, '2022-07-07', 5, 33, 4, 5),
* (1009, 2, 4, 300.00, '2022-07-17', 4, 41, 1, NULL),
* (1010, 2, 5, 650.00, '2022-07-31', 1, 67, 1, 4),
* (1011, 3, 1, 450.00, '2022-05-10', 2, 25, 3, 1),
* (1012, 3, 4, 180.00, '2022-05-20', 5, 33, 4, 1),
* (1013, 3, 2, 230.00, '2022-05-30', 4, 45, 3, NULL),
* (1014, 3, 2, 230.00, '2022-06-11', 2, 55, 1, 2),
* (1015, 3, 2, 230.00, '2022-06-22', 3, 21, 5, NULL),
* (1016, 4, 4, 300.00, '2022-05-15', 3, 31, 5, 5),
* (1017, 4, 4, 300.00, '2022-05-30', 1, 50, 1, NULL),
* (1018, 4, 4, 400.00, '2022-06-15', 2, 40, 3, 5),
* (1019, 4, 5, 400.00, '2022-06-30', 1, 70, 2, 4),
* (1020, 4, 5, 400.00, '2022-07-15', 3, 26, 5, 3),
* (1021, 5, 1, 550.00, '2022-07-01', 5, 22, 2, NULL),
* (1022, 5, 1, 550.00, '2022-07-08', 1, 34, 5, 1),
* (1023, 5, 2, 645.00, '2022-07-15', 4, 38, 5, 1),
* (1024, 5, 2, 645.00, '2022-07-21', 2, 58, 2, 1),
* (1025, 5, 2, 645.00, '2022-07-28', 2, 44, 4, NULL);

CREATE TABLE delivery\_partners (

* partner\_id INT PRIMARY KEY,
* partner\_name VARCHAR(50)
* );

INSERT INTO delivery\_partners (partner\_id, partner\_name)

VALUES

* (1, 'Suresh'),
* (2, 'Amit'),
* (3, 'Lokesh'),
* (4, 'Kartik'),
* (5, 'Gyandeep');

CREATE TABLE order\_details (

* id INT PRIMARY KEY,
* order\_id INT,
* f\_id INT,
* FOREIGN KEY (order\_id) REFERENCES orders(order\_id),
* FOREIGN KEY (f\_id) REFERENCES food\_items(f\_id)
* );

INSERT INTO order\_details (id, order\_id, f\_id) VALUES

* (1, 1001, 1),
* (2, 1001, 3),
* (3, 1002, 4),
* (4, 1002, 3),
* (5, 1003, 6),
* (6, 1003, 3),
* (7, 1004, 6),
* (8, 1004, 3),
* (9, 1005, 7),
* (10, 1005, 3),
* (11, 1006, 1),
* (12, 1006, 2),
* (13, 1006, 3),
* (14, 1007, 4),
* (15, 1007, 3),
* (16, 1008, 6),
* (17, 1008, 3),
* (18, 1009, 8),
* (19, 1009, 9),
* (20, 1010, 10),
* (21, 1010, 11),
* (22, 1010, 6),
* (23, 1011, 1),
* (24, 1012, 8),
* (25, 1013, 4),
* (26, 1014, 4),
* (27, 1015, 4),
* (28, 1016, 8),
* (29, 1016, 9),
* (30, 1017, 8),
* (31, 1017, 9),
* (32, 1018, 10),
* (33, 1018, 11),
* (34, 1019, 10),
* (35, 1019, 11),
* (36, 1020, 10),
* (37, 1020, 11),
* (38, 1021, 1),
* (39, 1021, 3),
* (40, 1022, 1),
* (41, 1022, 3),
* (42, 1023, 3),
* (43, 1023, 4),
* (44, 1023, 5),
* (45, 1024, 3),
* (46, 1024, 4),
* (47, 1024, 5),
* (48, 1025, 3),
* (49, 1025, 4),
* (50, 1025, 5);

CREATE TABLE food\_items (

* f\_id INT PRIMARY KEY,
* f\_name VARCHAR(50),
* type VARCHAR(50));

INSERT INTO food\_items (f\_id, f\_name, type)

VALUES

* (1, 'Non-veg Pizza', 'Non-veg'),
* (2, 'Veg Pizza', 'Veg'),
* (3, 'Choco Lava cake', 'Veg'),
* (4, 'Chicken Wings', 'Non-veg'),
* (5, 'Chicken Popcorn', 'Non-veg'),
* (6, 'Rice Meal', 'Veg'),
* (7, 'Roti meal', 'Veg'),
* (8, 'Masala Dosa', 'Veg'),
* (9, 'Rava Idli', 'Veg'),
* (10, 'Schezwan Noodles', 'Veg'),
* (11, 'Veg Manchurian', 'Veg');

CREATE TABLE menu (

* menu\_id INT PRIMARY KEY,
* r\_id INT,
* f\_id INT,
* price DECIMAL(10, 2),
* FOREIGN KEY (r\_id) REFERENCES restaurants(r\_id),
* FOREIGN KEY (f\_id) REFERENCES food\_items(f\_id));

INSERT INTO menu (menu\_id, r\_id, f\_id, price)

VALUES

* (1, 1, 1, 450.00),
* (2, 1, 2, 400.00),
* (3, 1, 3, 100.00),
* (4, 2, 3, 115.00),
* (5, 2, 4, 230.00),
* (6, 2, 5, 300.00),
* (7, 3, 3, 80.00),
* (8, 3, 6, 160.00),
* (9, 3, 7, 140.00),
* (10, 4, 6, 230.00),
* (11, 4, 8, 180.00),
* (12, 4, 9, 120.00),
* (13, 5, 6, 250.00),
* (14, 5, 10, 220.00),
* (15, 5, 11, 180.00);

select \* from users;

select \* from restaurants;

select \* from orders;

select \* from order\_details;

select \* from food\_items;

select \* from delivery\_partners;

select \* from menu;

-- 1) find customers that have never ordered?

* select user\_id, name from users where user\_id not in (select user\_id from orders);

-- 2) average price of all the dishes?

* select food\_items.f\_id, food\_items.f\_name, avg(price) from menu inner join food\_items on food\_items.f\_id = menu.f\_id group by menu.f\_id ;
* select f.f\_id, f.f\_name, avg(price) from menu as m inner join food\_items as f on f.f\_id = m.f\_id group by m.f\_id ;

-- 3) Find top restaurant in terms of no of orders for a given month?

* select r.r\_name, r.r\_id, count(r.r\_id) as total\_orders from orders as o inner join restaurants as r on r.r\_id = o.r\_id where date>='2022-07-01' and date<='2022-07-31' group by r\_id order by total\_orders desc limit 1;

-- 4) find restaurants with monthly sales > 500 rs

* select r.r\_name, r.r\_id, sum(amount) as monthly\_income from orders as o inner join restaurants as r on r.r\_id = o.r\_id where date>='2022-06-01' and date<='2022-06-30' group by r.r\_id having sum(amount)>500 order by monthly\_income desc;

-- 5) show all orders with order details for a particular customer in a particular date range

* select u.name, o.user\_id, o.date, f.f\_name, r.r\_name, o.amount from orders as o inner join users as u on o.user\_id = u.user\_id inner join order\_details as od on o.order\_id = od.order\_id inner join food\_items as f on od.f\_id = f.f\_id inner join restaurants as r on r.r\_id = o.r\_id where o.user\_id <= 1 and date>='2022-05-01' and date<='2022-07-3