Schema Design:

Schema Design for Hotel Management System

Users Table
_id: ObjectId, Primary Key
name: String, User's full name
email: String, User's email address (Unique)
passwordHash: String, Hashed password
role: String, Role of the user (e.g., 'admin', 'user')
membership: String, Membership status (e.g., 'active', 'inactive')
createdAt: Date, Account creation timestamp
updatedAt: Date, Timestamp of last update

Meals Table
_id: ObjectId, Primary Key
name: String, Name of the meal
details: String, Detailed description of the meal
image: String, URL of the meal image
category: String, Meal category (e.g., breakfast, lunch, dinner)
price: Number, Price of the meal
mealType: String, Type of meal (e.g., vegetarian, non-vegetarian)
distributorName: String, Name of the distributor
distributorEmail: String, Distributor's email

ingredients: Array, List of ingredients (array of strings)
postTime: Date, Time when the meal was posted
reactionCount: Number, Number of likes/reactions
reviews_count: Number, Number of reviews
createdAt: Date, Meal creation timestamp
updatedAt: Date, Last update timestamp

Reviews Table
_id: ObjectId, Primary Key
meal_id: ObjectId, Foreign Key referencing the meal being reviewed
user_email: String, Email of the user who posted the review
reviewText: String, Text content of the review
rating: Number, Rating out of 5 (Optional)
createdAt: Date, Date when the review was posted
updatedAt: Date, Last updated timestamp of the review

Upcoming Meals Table
_id: ObjectId, Primary Key
meal_id: ObjectId, Foreign Key referencing the meal
mealDate: Date, Date when the meal will be available
createdAt: Date, Date when the upcoming meal entry was created
updatedAt: Date, Last update timestamp of the upcoming meal entry

Payments Table

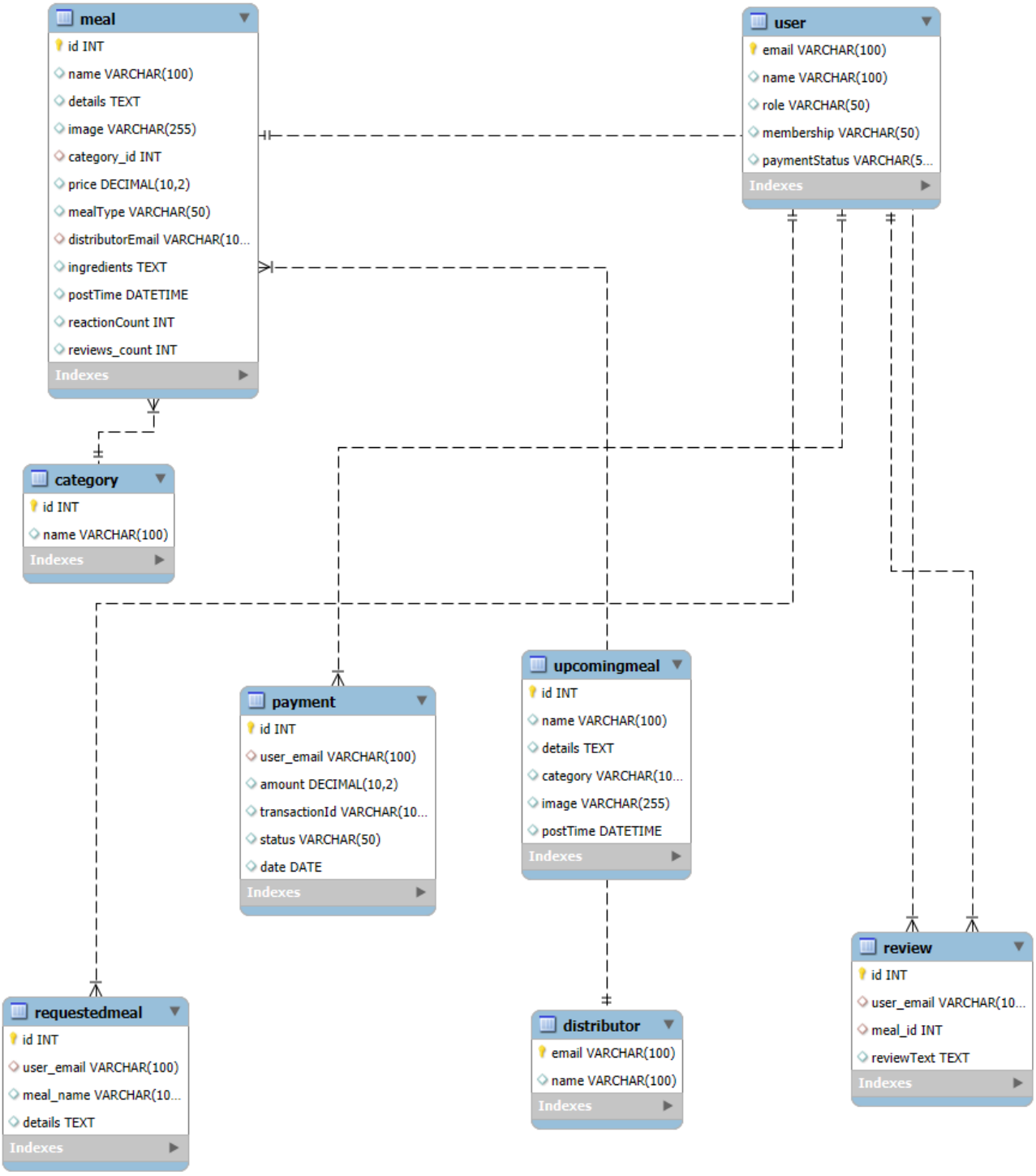
_id: ObjectId, Primary Key
user_email: String, Email of the user making the payment
amount: Number, Total amount paid
paymentMethod: String, Payment method used (e.g., 'Stripe', 'Credit Card', etc.)
status: String, Payment status (e.g., 'completed', 'pending')
paymentDate: Date, Date when the payment was made
createdAt: Date, Timestamp when the payment record was created
updatedAt: Date, Last update timestamp

Requested Meals Table
_id: ObjectId, Primary Key
user_email: String, Email of the user requesting the meal
meal_id: ObjectId, Foreign Key referencing the requested meal
quantity: Number, Quantity of the requested meal
status: String, Status of the request (e.g., 'pending', 'approved')
requestedAt: Date, Date when the meal was requested
createdAt: Date, Date when the request record was created
updatedAt: Date, Last update timestamp of the request

Relationships:

Users ↔ Meals: Review, Request, Distributor
Meals ↔ Reviews: Multiple reviews
Meals ↔ Upcoming Meals: Multiple availability dates
Users ↔ Payments: Payment for meal orders
Users ↔ Requested Meals: Request for meals

ER Diagram:


meal
id INT
name VARCHAR(100)
details TEXT
image VARCHAR(255)
category_id INT
price DECIMAL(10,2)
mealType VARCHAR(50)
distributorEmail VARCHAR(10...
ingredients TEXT
postTime DATETIME
reactionCount INT
reviews_count INT
Indexes
user
email VARCHAR(100)
name VARCHAR(100)
role VARCHAR(50)
membership VARCHAR(50)
paymentStatus VARCHAR(5...
Indexes
category
id INT
name VARCHAR(100)
Indexes
payment
id INT
user_email VARCHAR(100)
amount DECIMAL(10,2)
transactionId VARCHAR(10...
status VARCHAR(50)
date DATE
Indexes
upcomingmeal
id INT
name VARCHAR(100)
details TEXT
category VARCHAR(10...
image VARCHAR(255)
postTime DATETIME
Indexes
review
id INT
user_email VARCHAR(10...
meal_id INT
reviewText TEXT
Indexes
requestedmeal
id INT
user_email VARCHAR(100)
meal_name VARCHAR(10...
details TEXT
Indexes
distributor
email VARCHAR(100)
name VARCHAR(100)
Indexes

Normal Froms:

1. First Normal Form (1NF)

1NF Requirements:

- Every column must contain atomic values (no multi-valued attributes).
- The table must have a unique identifier (primary key).

User Table (1NF)

email (PK)	name	role	membership	paymentStatus
user1@example.com	John Doe	Admin	Gold	Paid
user2@example.com	Jane Smith	User	Silver	Pending

- **1NF**: The User table satisfies 1NF because all columns contain atomic values, and each row is uniquely identified by the primary key (email).

Distributor Table (1NF)

email (PK)	name
distributor1@example.com	PizzaCo
distributor2@example.com	SaladKing

- **1NF**: The Distributor table satisfies 1NF as well, with atomic values and a unique primary key (email).

Category Table (1NF)

id (PK)	name
1	Fast Food
2	Salad

- **1NF**: The Category table satisfies 1NF with atomic values and a primary key (id).

Meal Table (1NF)

id (PK)	name	details	image	category_id (FK)	price	mealType	distributorEmail (FK)	ingredients	postTime	reactionCount	reviews_count
1	Veg Pizza	Delicious	pizza.jpg	1	15.00	Veg	distributor1@example.com	Tomato, Cheese	2025-05-15 12:00	20	5
2	Chicken Salad	Healthy	salad.jpg	2	12.50	Non-Veg	distributor2@example.com	Chicken, Lettuce	2025-05-16 14:00	15	8

- **1NF**: The Meal table contains atomic values, but the ingredients column is a multi-valued field (Tomato, Cheese). This violates **1NF** because it contains multiple values in one cell. To resolve this, we will move the ingredients column to a separate table.

Review Table (1NF)

id (PK)	user_email (FK)	meal_id (FK)	reviewText
1	user1@example.com	1	Tastes amazing!
2	user2@example.com	2	Too spicy for me.

- **1NF**: The Review table satisfies 1NF, with atomic values and foreign keys to User and Meal.

RequestedMeal Table (1NF)

id (PK)	user_email (FK)	meal_name	details
1	user1@example.com	Veg Pizza	Add extra cheese
2	user2@example.com	Chicken Salad	No dressing

- **1NF**: The RequestedMeal table satisfies 1NF, with atomic values and a foreign key to User.

Payment Table (1NF)

id (PK)	user_email (FK)	amount	transactionId	status	date
1	user1@example.com	15.00	txn123	Paid	2025-05-15
2	user2@example.com	12.50	txn124	Pending	2025-05-16

- **1NF**: The Payment table satisfies 1NF with atomic values and a foreign key to User.

UpcomingMeal Table (1NF)

id (PK)	name	details	category	image	postTime	meal_id (FK)
1	Veg Pizza	Delicious	Fast Food	pizza.jpg	2025-05-15 12:00	1
2	Chicken Salad	Healthy	Salad	salad.jpg	2025-05-16 14:00	2

- **1NF**: The UpcomingMeal table satisfies 1NF with atomic values and a foreign key to Meal.

---

2. Second Normal Form (2NF)

2NF Requirements:

- The table must meet the requirements of **1NF**.
- Every non-key attribute must be fully functionally dependent on the primary key.

We will evaluate each table to check for **partial dependencies**.

User Table (2NF)

- Already in 2NF because all non-key attributes (name, role, membership, paymentStatus) depend on the primary key (email).

Distributor Table (2NF)

- Already in 2NF because name depends on the primary key (email).

Category Table (2NF)

- Already in 2NF because name depends on the primary key (id).

Meal Table (2NF)

- **Partial dependency**: The Meal table contains columns like **distributorEmail** and **category_id**, which are related to the **distributor** and **category** entities respectively. Therefore, we will normalize further by creating a Distributor and Category table separately, which we already have.

Review Table (2NF)

- Already in 2NF because all non-key attributes (reviewText) depend on the primary key (id).

RequestedMeal Table (2NF)

- Already in 2NF because meal_name and details depend on the primary key (id).

Payment Table (2NF)

- Already in 2NF because amount, transactionId, status, and date depend on the primary key (id).

UpcomingMeal Table (2NF)

- Already in 2NF because name, details, category, image, and postTime depend on the primary key (id), and meal_id is a foreign key that points to the Meal table.

---

3. Third Normal Form (3NF)

3NF Requirements:

- The table must meet the requirements of **2NF**.
- It must have no transitive dependencies (i.e., non-key attributes must not depend on other non-key attributes).

We'll evaluate each table for **transitive dependencies**.

User Table (3NF)

- Already in 3NF because all non-key attributes depend directly on the primary key (email).

Distributor Table (3NF)

- Already in 3NF because name depends directly on the primary key (email).

Category Table (3NF)

- Already in 3NF because name depends directly on the primary key (id).

Meal Table (3NF)

- **Transitive dependency**: The ingredients column in the **Meal** table should be moved to a separate table. This was handled earlier by creating a MealIngredients table.

Review Table (3NF)

- Already in 3NF because reviewText depends on the primary key (id).

RequestedMeal Table (3NF)

- Already in 3NF because all non-key attributes (meal_name, details) depend on the primary key (id).

Payment Table (3NF)

- Already in 3NF because all non-key attributes depend directly on the primary key (id).

UpcomingMeal Table (3NF)

- Already in 3NF because all non-key attributes depend directly on the primary key (id), and meal_id is a foreign key pointing to the Meal table.

Final Database Schema (Normalized to 3NF)

User Table

email (PK)	name	role	membership	paymentStatus
user1@example.com	John Doe	Admin	Gold	Paid

Distributor Table

email (PK)	name
distributor1@example.com	PizzaCo

Category Table

id (PK)	name
1	Fast Food

Meal Table

id (PK)	name	details	image	category_id (FK)	price	mealType	distributorEmail (FK)	postTime	reactionCount	reviews_count
1	Veg Pizza	Delicious	pizza.jpg	1	15.00	Veg	distributor1@example.com	2025-05-15 12:00	20	5

MealIngredients Table (New)

meal_id (FK)	ingredient
1	Tomato
1	Cheese

Review Table

id (PK)	user_email (FK)	meal_id (FK)	reviewText
1	user1@example.com	1	Tastes amazing!

RequestedMeal Table

id (PK)	user_email (FK)	meal_name	details
1	user1@example.com	Veg Pizza	Add extra cheese

Payment Table

id (PK)	user_email (FK)	amount	transactionId	status	date
1	user1@example.com	15.00	txn123	Paid	2025-05-15

UpcomingMeal Table

id (PK)	name	details	category	image	postTime	meal_id (FK)
1	Veg Pizza	Delicious	Fast Food	pizza.jpg	2025-05-15 12:00	1