Tables and Columns

1. **Users**
   - user_id (INT, Primary Key)
   - user_name (VARCHAR(100), NOT NULL)
   - email (VARCHAR(100), UNIQUE, NOT NULL)
   - phone (VARCHAR(15), UNIQUE, NOT NULL)
   - address (TEXT)
   - created_at (DATETIME, DEFAULT CURRENT_TIMESTAMP)

user_id	user_name	email	phone	address	created_at
1	Aarav Sharma	aarav.sharma@gmail.com	9876543210	123 MG Road, Bengaluru, Karnataka	01-01-2024
2	Vivaan Gupta	vivaan.gupta@icloud.com	9876543211	456 Brigade Rd, Bengaluru, Karnataka	02-01-2023
3	Aditya Iyer	aditya.iyer@yahoo.com	9876543212	789 Church St, Mumbai, Maharashtra	06-07-2023
4	Anaya Patel	anaya.patel@yahoo.com	9876543213	321 Linking Rd, Mumbai, Maharashtra	04-07-2022
5	Sai Kumar	sai.kumar@gmail.com	9876543214	654 Jayanagar, Bengaluru, Karnataka	05-01-2023
6	Diya Singh	diya.singh@gmail.com	9876543215	987 Andheri West, Mumbai, Maharashtra	06-01-2023
7	Rohan Mehta	rohan.mehta@outlook.com	9876543216	135 HSR Layout, Bengaluru, Karnataka	07-10-2024
8	Tia Nair	tia.nair@icloud.com	9876543217	246 Bandra East, Mumbai, Maharashtra	08-01-2023
9	Krishna Rao	krishna.rao@icloud.com	9876543218	357 Koramangala, Bengaluru, Karnataka	09-01-2023
10	Pooja Joshi	pooja.joshi@outlook.com	9876543219	468 Powai, Mumbai, Maharashtra	10-05-2024

2. **Restaurants**
   - restaurant_id (INT, Primary Key)
   - restaurant_name (VARCHAR(100), NOT NULL)
   - address (TEXT, NOT NULL)
   - cuisine (VARCHAR(50), NOT NULL)
   - rating (DECIMAL(2, 1))
   - created_at (DATETIME, DEFAULT CURRENT_TIMESTAMP)

restaurant_id	restaurant_name	address	cuisine	rating	created_at
1	Spice Route	123 Curry Lane, Bengaluru, Karnataka	Indian	4.5	01-01-2023
2	Sushi Paradise	456 Sushi Street, Mumbai, Maharashtra	Japanese	3.7	02-01-2023
3	Pizza House	789 Pizza Avenue, Delhi	Italian	4.2	03-01-2023
4	Tandoori Nights	321 Grill Road, Chennai, Tamil Nadu	Mexican	4.6	04-01-2023
5	Burger Shack	654 Burger Boulevard, Pune	American	3	05-01-2023
6	Vegan Delight	987 Green St, Hyderabad	Vegan	3.8	06-01-2023
7	Mexican Fiesta	135 Taco Lane, Jaipur	Italian	4.3	07-01-2023
8	The Great Dine	246 Dine Way, Kolkata	Continental	4.1	08-01-2023
9	Barbeque Nation	357 Grill St, Ahmedabad	Barbecue	3.4	09-01-2023
10	Dim Sum House	468 Dumpling Rd, Surat	Chinese	4	10-01-2023

3. **Menus**

- menu_id (INT, Primary Key)
- restaurant_id (INT, Foreign Key referencing Restaurants(restaurant_id))
- dish_name (VARCHAR(100), NOT NULL)
- price (DECIMAL(10, 2), NOT NULL)
- availability (BOOLEAN, DEFAULT TRUE)
- created_at (DATETIME, DEFAULT CURRENT_TIMESTAMP)

menu_id	restaurant_id	dish_name	price	availability	created_at
1	1	Butter Chicken	550	1	20-09-2024
2	1	Paneer Tikka	200	1	20-09-2024
3	1	Biryani	300	1	20-09-2024
4	2	Dosa	80	1	20-09-2024
5	2	Idli	60	1	20-09-2024
6	2	Sambar	50	1	20-09-2024
7	3	Chaat	350	1	20-09-2024
8	3	Pav Bhaji	120	1	20-09-2024
9	3	Palak Paneer	520	1	20-09-2024
10	4	Vada Pav	40	1	20-09-2024

4. **Orders**

- order_id (INT, Primary Key)
- user_id (INT, Foreign Key referencing Users(user_id))
- restaurant_id (INT, Foreign Key referencing Restaurants(restaurant_id))
- order_status (VARCHAR(20), NOT NULL, Default 'Pending')

- total_amount (DECIMAL(10, 2), NOT NULL)
- order_date (DATETIME, DEFAULT CURRENT_TIMESTAMP)
- delivery_date (DATETIME)

order_id	order_status	total_amount	order_date	delivery_date	user_id	restaurant_id
1	Pending	1160	20-05-2024	29-10-2024	1	1
2	Completed	320	20-04-2024	18-02-2024	2	1
3	Cancelled	200	20-03-2024	20-04-2024	4	2
4	Pending	450	20-12-2024	26-09-2024	4	3
5	Completed	180	20-01-2024	21-10-2024	5	2
6	Pending	1250	20-06-2024	30-04-2024	4	4
7	Completed	670	20-07-2024	10-08-2024	4	9
8	Pending	320	20-08-2024	25-05-2024	8	7
9	Pending	550	20-09-2024	10-06-2024	4	3
10	Completed	680	20-11-2024	23-07-2024	5	8

5. **Order_Items**
   - order_item_id (INT, Primary Key)
   - order_id (INT, Foreign Key referencing Orders(order_id))
   - menu_id (INT, Foreign Key referencing Menus(menu_id))
   - quantity (INT, NOT NULL)
   - price (DECIMAL(10, 2), NOT NULL)

order_item_id	order_id	menu_id	quantity	price
1	1	1	2	1200
2	1	2	1	200
3	2	3	3	900
4	2	1	1	550
5	3	4	2	160
6	3	5	4	240
7	4	2	1	200
8	4	6	2	100
9	5	3	1	300
10	5	7	3	150

6. **Payments**
   - payment_id (INT, Primary Key)
   - order_id (INT, Foreign Key referencing Orders(order_id))
   - payment_type (VARCHAR(50), NOT NULL, e.g., 'Credit Card', 'Cash')
   - amount (DECIMAL(10, 2), NOT NULL)

- payment_status (VARCHAR(20), NOT NULL, e.g., 'Completed', 'Pending')
- payment_date (DATETIME, DEFAULT CURRENT_TIMESTAMP)

payment_id	order_id	payment_type	amount	payment_status	payment_date
1	1	Credit Card	1200	Completed	20-09-2024
2	2	UPI	200	Completed	20-09-2024
3	3	Cash	900	Pending	20-09-2024
4	4	Debit Card	550	Completed	20-09-2024
5	5	Net Banking	160	Failed	20-09-2024
6	1	Paytm	1200	Completed	20-09-2024
7	2	Credit Card	200	Completed	20-09-2024
8	9	UPI	150	Pending	20-09-2024
9	7	Cash	150	Completed	20-09-2024
10	8	Debit Card	130	Failed	20-09-2024

7. **Delivery_person**

- delivery_personid (INT,PRIMARY KEY)
- persons_name (VARCHAR NOT NULL)
- contact_no (VARCHAR UNIQUE NOT NULL)
- order_id (INT, Foreign Key referencing Orders(order_id))
- restaurant_id (INT, Foreign Key referencing Restaurants(restaurant_id))
- user_id (INT, Foreign Key referencing Users(user_id))

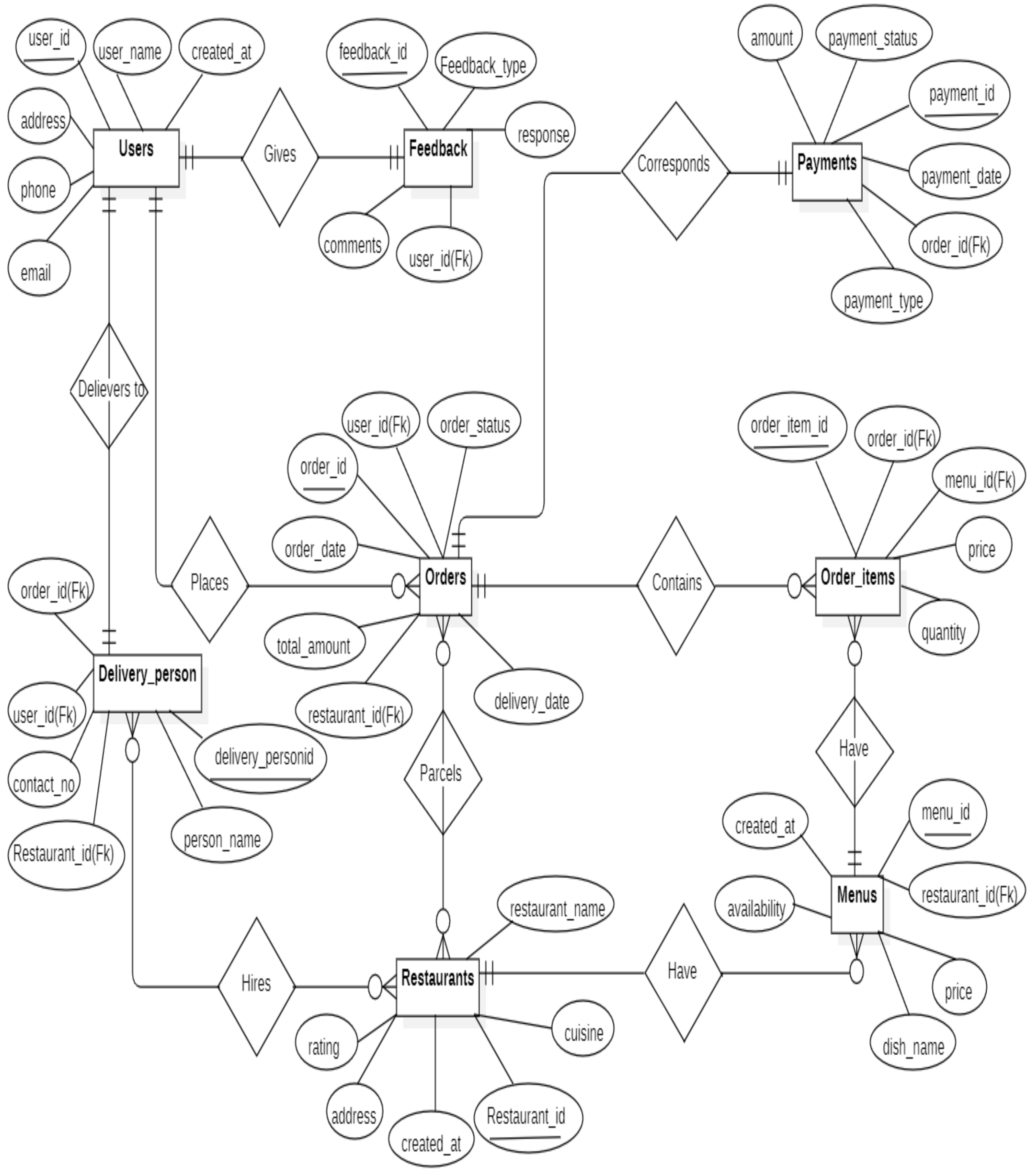
delivery_personid	contact_no	order_id	restaurant_id	user_id	persons_name
1	9876543210	1	1	1	Rahul Sharma
2	9876543211	2	2	2	Anita Singh
3	9876543212	3	3	3	Vikram Mehta
4	9876543213	4	1	4	Priya Joshi
5	9876543214	5	2	5	Suresh Patel
6	9876543215	7	4	6	Riya Verma
7	9876543216	1	5	7	Karan Gupta
8	9876543217	2	6	8	Nisha Reddy
9	9876543218	3	7	9	Amit Bansal
10	9876543219	4	8	10	Neha Iyer

8. **feedback**

- feedback_id (INT PRIMARY KEY)
- feedback_type (VARCHAR(50) NOT NULL)
- user_id (INT foreign key referencing users(user_id))
- comments (TEXT)
- response (TEXT)

feedback_id	feedback_type	user_id	comments	response
1	Positive	1	The food was excellent and delivered on time!	NULL
2	Negative	2	The order arrived late and was cold.	We apologize for the delay and are looking into it.
3	Suggestion	3	Add more vegetarian options to the menu.	NULL
4	Positive	4	Great service and delicious food!	NULL
5	Negative	5	Very disappointing experience.	We are sorry to hear that.
6	Suggestion	6	Consider a loyalty program for regular customers.	NULL
7	Positive	7	Best biryani I have ever had!	NULL
8	Negative	8	Not as spicy as I expected.	NULL
9	Positive	9	Good quality food but a bit pricey.	NULL
10	Suggestion	10	Add a feedback form in the app.	NULL

ER DIGRAM


user_id
user_name
created_at
address
Users
phone
email
Gives
feedback_id
Feedback_type
Feedback
response
comments
user_id(Fk)
amount
payment_status
payment_id
Corresponds
Payments
payment_date
order_id(Fk)
payment_type
Delievers to
user_id(Fk)
order_status
order_id
order_date
Places
Orders
Contains
total_amount
restaurant_id(Fk)
delivery_date
order_item_id
order_id(Fk)
menu_id(Fk)
price
Order_items
quantity
order_id(Fk)
Delivery_person
user_id(Fk)
delivery_personid
contact_no
Restaurant_id(Fk)
person_name
Parcels
Have
created_at
menu_id
availability
Menus
restaurant_id(Fk)
restaurant_name
Hires
Restaurants
Have
price
dish_name
rating
cuisine
address
created_at
Restaurant_id

Questions:

1. Get all users from the Users table.

   ```
   SELECT * FROM users;
   ```

2. Get the names and addresses of all restaurants.

   ```
   SELECT restaurant_name,address FROM restaurants;
   ```

3. Find all users who signed up after 2023-01-01.

   ```
   SELECT * FROM users
   WHERE created_at>'2023-01-01';
   ```

4. Get all restaurants with a rating of 4.0 or higher.

   ```
   SELECT * FROM restaurants
   WHERE rating >=4.0;
   ```

5. Find all orders placed between 2024-01-01 and 2024-09-01.

   ```
   SELECT * FROM orders
   WHERE order_date BETWEEN '2024-01-01' AND '2024-09-01';
   ```

6. Get details of users from specific email domains.

   ```
   SELECT * FROM users
   WHERE email LIKE '%@gmail.com'
   OR email LIKE '%@yahoo.com';
   ```

7. Find all restaurants serving Italian cuisine.

   ```
   SELECT * FROM restaurants
   WHERE cuisine="Italian";
   ```

8. Get the total number of orders placed by each user.

```
SELECT orders.user_id,users.user_name,count(orders.order_id) AS total_orders FROM orders
INNER JOIN users ON orders.user_id=users.user_id
 GROUP BY orders.user_id;
```

9. Get the average rating of restaurants by cuisine type.

```
SELECT cuisine,avg(rating) AS average_rating FROM restaurants
GROUP BY cuisine;
```

10. Get all order details along with user names and restaurant names.

```
SELECT users.user_name,restaurants.restaurant_name,orders.* FROM orders
INNER JOIN users ON orders.user_id=users.user_id
INNER JOIN restaurants ON orders.restaurant_id=restaurants.restaurant_id;
```

11. Get all items from an order along with the dish name and price.

```
SELECT order_item_id,menus.dish_name,menus.price FROM order_items
INNER JOIN menus ON menus.menu_id = order_items.menu_id;
```

12. Get the most expensive dish from each restaurant.

```
SELECT menus.restaurant_id,menus.dish_name,restaurants.restaurant_name,menus.price FROM menus
INNER JOIN restaurants ON menus.restaurant_id=restaurants.restaurant_id
GROUP BY menus.restaurant_id;
```

13. Get the users who have placed more than 5 orders.

```
SELECT orders.user_id,users.user_name,orders.order_id FROM orders
INNER JOIN users ON users.user_id=orders.user_id
GROUP BY orders.user_id
HAVING COUNT(orders.user_id)>=5;
```

14. Get the first 5 characters of each user’s name.

```
SELECT user_name,SUBSTRING(user_name,1,5)AS short_name FROM users;
```

15. Convert all restaurant names to uppercase.

```
SELECT restaurant_name,UPPER(restaurant_name) AS upper_names FROM restaurants;
```

16. Get the current date and time.

```
SELECT NOW() AS `current time & date`;
```

17. Calculate the time difference (in hours) between order placement and delivery.

```
SELECT order_date,delivery_date,HOUR(TIMEDIFF(order_date,delivery_date))AS time_diff
FROM orders;
```

18. Get the top 3 highest-rated restaurants.

```
SELECT * FROM restaurants

WHERE rating >= 4

LIMIT 3;
```

19. Find all restaurants where the price of the most expensive item is greater than 500.

```
SELECT restaurants.restaurant_name,menus.restaurant_id,menus.price FROM menus

INNER JOIN restaurants ON menus.restaurant_id=restaurants.restaurant_id

WHERE menus.price>500;
```

20. Show the order status for each order as 'Delivered' if order_status is 'Completed', 'In Progress' if order_status is 'Pending', and 'Cancelled' otherwise.

```
SELECT order_id,

CASE

WHEN order_status='Completed' THEN 'Delivered'

WHEN order_status='Pending' THEN 'In Progress'

ELSE 'Cancelled'

END AS `order status` FROM orders;
```

21. Display all orders with a calculated field that shows total_amount as 'High' if the total is greater than 1000, otherwise 'Low'.

```
SELECT order_id,total_amount,
CASE
WHEN total_amount>1000 THEN 'High'
ELSE 'Low'
END AS amount_type FROM orders;
```

22. Which users orders get delivered by delivery person.

```
SELECT
delivery_person.delivery_personid,delivery_person.persons_name,orders.order_status,users.user_name FROM delivery_person
INNER JOIN orders ON delivery_person.order_id=orders.order_id
INNER JOIN users ON delivery_person.user_id=users.user_id
WHERE orders.order_status='Completed';
```

23. How many dishes are available in each restaurants.

```
SELECT
restaurants.restaurant_id,restaurants.restaurant_name,menus.dish_name,COUNT(menus.availability) AS currently_available FROM menus
INNER JOIN restaurants ON restaurants.restaurant_id=menus.restaurant_id
GROUP BY menus.restaurant_id;
```

24. Find out the users who placed more than one order.

```
SELECT user_id,user_name FROM users
WHERE(SELECT count(user_id) FROM orders WHERE orders.user_id=users.user_id)>1;
```

25. Find out those users who gave the feedback but have not placed order.

```
SELECT users.user_id,users.user_name FROM users
INNER JOIN feedback ON users.user_id=feedback.user_id
LEFT JOIN orders ON users.user_id=orders.user_id
WHERE orders.order_id IS NULL;
```

26. Get restaurants that have positive feedback and average rating.

```
SELECT restaurants.restaurant_id,restaurants.restaurant_name,feedback.feedback_type
FROM restaurants

INNER JOIN orders ON orders.restaurant_id=restaurants.restaurant_id

INNER JOIN feedback ON feedback.user_id=orders.user_id

WHERE feedback_type="Positive";
```