```
CREATE TABLE dim_date (
    date_id INT COMMENT 'Date in YYYYMMDD format',
    full_date DATE COMMENT 'Full date',
    day TINYINT COMMENT 'Day of the month',
    month TINYINT COMMENT 'Month',
    year INT COMMENT 'Year',
    quarter TINYINT COMMENT 'Quarter',
    day_of_week TINYINT COMMENT 'Day of the week (1=Monday)',
    is_weekend BOOLEAN COMMENT 'Is weekend (true if Saturday or
Sunday)'
)
COMMENT 'Date dimension table'
STORED AS PARQUET;

CREATE TABLE dim_customer (
    customer_id BIGINT COMMENT 'Unique customer identifier',
    first_name STRING COMMENT 'First name of the customer',
    last_name STRING COMMENT 'Last name of the customer',
    email STRING COMMENT 'Email address',
    phone_number STRING COMMENT 'Phone number',
    gender STRING COMMENT 'Gender',
    date_of_birth DATE COMMENT 'Date of birth',
    registration_date DATE COMMENT 'Date the customer registered',
    city STRING COMMENT 'City of residence',
    loyalty_level STRING COMMENT 'Loyalty program level'
)
COMMENT 'Customer dimension table'
CLUSTERED BY (city) INTO 8 BUCKETS
STORED AS PARQUET;

CREATE TABLE dim_restaurant (
    restaurant_id BIGINT COMMENT 'Unique restaurant identifier',
    restaurant_name STRING COMMENT 'Name of the restaurant',
    cuisine_type STRING COMMENT 'Type of cuisine',
    city STRING COMMENT 'City where the restaurant is located',
    rating FLOAT COMMENT 'Average customer rating',
    average_delivery_time INT COMMENT 'Average delivery time in
minutes',
    operational_status STRING COMMENT 'Current operational status'
)
COMMENT 'Restaurant dimension table'
STORED AS PARQUET;


CREATE TABLE dim_delivery_partner (
    delivery_partner_id BIGINT COMMENT 'Unique delivery partner
identifier',
    first_name STRING COMMENT 'First name of the delivery partner',
    last_name STRING COMMENT 'Last name of the delivery partner',
    vehicle_type STRING COMMENT 'Type of vehicle used',
    registration_date DATE COMMENT 'Date the partner registered',
    city STRING COMMENT 'City of operation',
    average_rating FLOAT COMMENT 'Average customer rating',
    total_deliveries INT COMMENT 'Total number of deliveries
completed'
)
COMMENT 'Delivery Partner dimension table'
STORED AS PARQUET;


CREATE TABLE dim_promotion (
    promo_code STRING COMMENT 'Promotion code',
    description STRING COMMENT 'Description of the promotion',
    start_date DATE COMMENT 'Promotion start date',
    end_date DATE COMMENT 'Promotion end date',
    discount_type STRING COMMENT 'Type of discount (e.g., percentage,
fixed amount)',
    discount_value FLOAT COMMENT 'Value of the discount'
)
COMMENT 'Promotion dimension table'
STORED AS PARQUET;


CREATE TABLE dim_menu_item (
    menu_item_id BIGINT COMMENT 'Unique menu item identifier',
    restaurant_id BIGINT COMMENT 'Restaurant ID',
    item_name STRING COMMENT 'Name of the menu item',
    item_category STRING COMMENT 'Category of the menu item',
    price FLOAT COMMENT 'Price of the item',
    availability_status STRING COMMENT 'Availability status of the item'
)
COMMENT 'Menu Item dimension table'
STORED AS PARQUET

CREATE TABLE fact_orders (
    order_id BIGINT COMMENT 'Unique order identifier',
    customer_id BIGINT COMMENT 'Customer ID',
    restaurant_id BIGINT COMMENT 'Restaurant ID',
    delivery_partner_id BIGINT COMMENT 'Delivery Partner ID',
    promo_code STRING COMMENT 'Promotion code used',
    order_amount FLOAT COMMENT 'Total amount of the order',
    delivery_fee FLOAT COMMENT 'Delivery fee',
    tip_amount FLOAT COMMENT 'Tip amount given',
    delivery_time INT COMMENT 'Delivery time in minutes',
    order_status STRING COMMENT 'Status of the order',
    total_items INT COMMENT 'Total number of distinct items in the order',
    total_quantity INT COMMENT 'Total quantity of all items in the order'
)
PARTITIONED BY (date_id INT)
STORED AS PARQUET

CREATE TABLE fact_order_items (
    order_id BIGINT COMMENT 'Order ID',
    menu_item_id BIGINT COMMENT 'Menu Item ID',
    quantity INT COMMENT 'Quantity ordered',
    item_price FLOAT COMMENT 'Price of the item at time of order',
    special_instructions STRING COMMENT 'Special instructions for the item'
)
COMMENT 'Fact table for order items'
STORED AS PARQUET
```

```
CREATE TABLE fact_customer_feedback (
    feedback_id BIGINT COMMENT 'Unique feedback identifier',
    order_id BIGINT COMMENT 'Order ID',
    customer_id BIGINT COMMENT 'Customer ID',
    rating INT COMMENT 'Rating given by the customer',
    comments STRING COMMENT 'Feedback comments'
)
PARTITIONED BY (feedback_date)
COMMENT 'Customer Feedback fact table'
STORED AS PARQUET

CREATE TABLE fact_restaurant_feedback (
    feedback_id BIGINT COMMENT 'Unique feedback identifier',
    order_id BIGINT COMMENT 'Order ID',
    restaurant_id BIGINT COMMENT 'Restaurant ID',
    comments STRING COMMENT 'Feedback comments from the restaurant',
    resolution_status STRING COMMENT 'Status of feedback resolution'
)
PARTITIONED BY (feedback_date)
COMMENT 'Restaurant Feedback fact table'
STORED AS PARQUET
```

1. Populate **`dim_date`**

```
INSERT INTO dim_date VALUES

(20220101, '2022-01-01', 1, 1, 2022, 1, 6, true),

(20220102, '2022-01-02', 2, 1, 2022, 1, 7, true),

(20220103, '2022-01-03', 3, 1, 2022, 1, 1, false),

(20220104, '2022-01-04', 4, 1, 2022, 1, 2, false),

(20220105, '2022-01-05', 5, 1, 2022, 1, 3, false),

(20220106, '2022-01-06', 6, 1, 2022, 1, 4, false),

(20220107, '2022-01-07', 7, 1, 2022, 1, 5, false),

(20220108, '2022-01-08', 8, 1, 2022, 1, 6, true),

(20220109, '2022-01-09', 9, 1, 2022, 1, 7, true);
```

Explanation:

- Provides date entries for the first nine days of January 2022.
- Covers weekdays and weekends to support time-based queries.

---

2. Populate **`dim_customer`**

```
INSERT INTO dim_customer VALUES

(1001, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', 'Male',
'1990-05-15', '2021-01-10', 'New York', 'Gold'),

(1002, 'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901',
'Female', '1985-08-22', '2021-02-20', 'Los Angeles', 'Silver'),

(1003, 'Mike', 'Johnson', 'mike.johnson@example.com', '345-678-9012',
'Male', '1992-11-30', '2021-03-15', 'Chicago', 'Bronze'),

(1004, 'Emily', 'Davis', 'emily.davis@example.com', '456-789-0123',
'Female', '1988-07-07', '2021-04-25', 'Houston', 'Gold'),
(1005, 'David', 'Wilson', 'david.wilson@example.com', '567-890-1234',
'Male', '1995-02-28', '2021-05-30', 'Phoenix', 'Silver');
```

Explanation:

- Adds five customers from different cities with varying loyalty levels.
- Supports customer segmentation and loyalty program analysis.

---

3. Populate **dim_restaurant**

```
INSERT INTO dim_restaurant VALUES

(2001, 'Pizza Palace', 'Italian', 'New York', 4.5, 30, 'Open'),

(2002, 'Sushi World', 'Japanese', 'Los Angeles', 4.7, 25, 'Open'),

(2003, 'Burger Barn', 'American', 'Chicago', 4.2, 20, 'Open'),

(2004, 'Curry House', 'Indian', 'Houston', 4.6, 35, 'Open'),

(2005, 'Taco Town', 'Mexican', 'Phoenix', 4.3, 15, 'Open');
```

Explanation:

- Introduces five restaurants with different cuisines and locations.
- Enables analysis of restaurant performance and customer preferences.

---

4. Populate **dim_delivery_partner**

```
INSERT INTO dim_delivery_partner VALUES

(3001, 'Alex', 'Brown', 'Bike', '2020-06-15', 'New York', 4.8, 500),

(3002, 'Sam', 'Green', 'Car', '2020-07-20', 'Los Angeles', 4.6, 450),
(3003, 'Chris', 'White', 'Scooter', '2020-08-25', 'Chicago', 4.7,
400),

(3004, 'Pat', 'Black', 'Bike', '2020-09-30', 'Houston', 4.5, 550),

(3005, 'Taylor', 'Gray', 'Car', '2020-10-05', 'Phoenix', 4.9, 600);
```

Explanation:

- Provides delivery partners with various vehicle types and cities.
- Allows for performance analysis and comparisons.

---

5. Populate **dim_promotion**

```
INSERT INTO dim_promotion VALUES

('PROMO2021', 'New Year Promotion', '2022-01-01', '2022-01-10',
'Percentage', 10),

('SAVE5', 'Save $5 on orders over $50', '2022-01-05', '2022-01-15',
'Fixed', 5),

('FREEDelivery', 'Free Delivery Promo', '2022-01-01', '2022-01-31',
'Fixed', 0),

('WELCOME10', 'Welcome Discount', '2022-01-01', '2022-12-31',
'Percentage', 10),

('SUMMER20', 'Summer Special 20% Off', '2022-06-01', '2022-08-31',
'Percentage', 20);
```

Explanation:

- Adds various promotions with different discount types and durations.
- Supports analysis of promotion effectiveness.

---

7. Populate **dim_menu_item**

```
INSERT INTO dim_menu_item VALUES
(4001, 2001, 'Margherita Pizza', 'Pizza', 12.99, 'Available'),
(4002, 2001, 'Pepperoni Pizza', 'Pizza', 14.99, 'Available'),
(4003, 2002, 'California Roll', 'Sushi', 8.99, 'Available'),
(4004, 2002, 'Spicy Tuna Roll', 'Sushi', 9.99, 'Available'),
(4005, 2003, 'Classic Burger', 'Burger', 10.99, 'Available'),
(4006, 2003, 'Cheeseburger', 'Burger', 11.99, 'Available'),
(4007, 2004, 'Chicken Curry', 'Curry', 13.99, 'Available'),
(4008, 2004, 'Vegetable Curry', 'Curry', 11.99, 'Available'),
(4009, 2005, 'Beef Taco', 'Taco', 3.99, 'Available'),
(4010, 2005, 'Chicken Taco', 'Taco', 3.49, 'Available');
```

Explanation:

- Lists menu items for each restaurant.
- Enables detailed order item analysis.

---

8. Populate **fact_orders**

```
INSERT INTO fact_orders PARTITION (date_id) VALUES
-- Orders on 2022-01-01
(5001, 1001, 2001, 3001, 'PROMO2021', 25.98, 2.99, 3.00, 30,
'Completed', 2, 2, 20220101),
(5002, 1002, 2002, 3002, 'WELCOME10', 18.98, 2.99, 2.00, 25,
'Completed', 2, 2, 20220101),
-- Orders on 2022-01-02

(5003, 1003, 2003, 3003, NULL, 10.99, 2.99, 1.50, 20, 'Completed', 1,
1, 20220102),

(5004, 1004, 2004, 3004, 'FREEDelivery', 13.99, 0.00, 2.00, 35,
'Completed', 1, 1, 20220102),

-- Orders on 2022-01-03

(5005, 1005, 2005, 3005, 'SAVE5', 50.00, 2.99, 5.00, 15, 'Completed',
10, 10, 20220103),

(5006, 1001, 2001, 3001, 'PROMO2021', 25.98, 2.99, 3.00, 28,
'Completed', 2, 2, 20220103),

-- Orders on 2022-01-04

(5007, 1002, 2002, 3002, 'SUMMER20', 37.96, 2.99, 4.00, 24,
'Completed', 4, 4, 20220104),

(5008, 1003, 2003, 3003, NULL, 21.98, 2.99, 2.00, 22, 'Completed', 2,
2, 20220104),

-- Orders on 2022-01-05

(5009, 1004, 2004, 3004, 'FREEDelivery', 27.98, 0.00, 3.00, 33,
'Completed', 2, 2, 20220105),

(5010, 1005, 2005, 3005, NULL, 7.98, 2.99, 1.00, 16, 'Completed', 2,
2, 20220105);
```

Explanation:

- Inserts ten orders across five days.
- Includes a mix of promotions, customers, restaurants, and delivery partners.
- Provides data for testing various queries and use cases.

---

9. Populate **`fact_order_items`**

```
INSERT INTO fact_order_items VALUES
-- Order 5001 Items
(5001, 4001, 1, 12.99, 'Extra cheese'),
(5001, 4002, 1, 12.99, ''),
-- Order 5002 Items
(5002, 4003, 1, 8.99, 'No wasabi'),
(5002, 4004, 1, 9.99, ''),
-- Order 5003 Items
(5003, 4005, 1, 10.99, 'No onions'),
-- Order 5004 Items
(5004, 4007, 1, 13.99, 'Spicy level 3'),
-- Order 5005 Items
(5005, 4009, 5, 3.99, ''),
(5005, 4010, 5, 3.49, ''),
-- Order 5006 Items
(5006, 4001, 1, 12.99, ''),
(5006, 4002, 1, 12.99, ''),
-- Order 5007 Items
(5007, 4003, 2, 8.99, ''),
(5007, 4004, 2, 9.99, ''),
-- Order 5008 Items
(5008, 4005, 1, 10.99, ''),
(5008, 4006, 1, 11.99, ''),
-- Order 5009 Items

(5009, 4007, 1, 13.99, ''),

(5009, 4008, 1, 11.99, ''),

-- Order 5010 Items

(5010, 4009, 1, 3.99, ''),

(5010, 4010, 1, 3.49, '');
```

Explanation:

- Details the items for each order.
- Includes quantities, prices, and special instructions.
- Supports detailed order analysis and item-level queries.

10. Populate **fact_customer_feedback**

```
INSERT INTO fact_customer_feedback PARTITION (feedback_date) VALUES

-- Feedback on 2022-01-02

(6001, 5001, 1001, 5, 'Great food and quick delivery!', '2022-01-02'),

(6002, 5002, 1002, 4, 'Good sushi but delivery was a bit slow.',
'2022-01-02'),

-- Feedback on 2022-01-03

(6003, 5003, 1003, 3, 'Burger was okay, nothing special.',
'2022-01-03'),

-- Feedback on 2022-01-04

(6004, 5004, 1004, 5, 'Delicious curry and arrived hot!',
'2022-01-04'),
(6005, 5005, 1005, 4, 'Tacos were tasty but could be warmer.',
'2022-01-04'),

-- Feedback on 2022-01-05

(6006, 5006, 1001, 5, 'Consistently good pizza!', '2022-01-05');
```

Explanation:

- Captures customer feedback for several orders.
- Includes ratings and comments.
- Supports customer satisfaction analysis.

---

11. Populate **`fact_restaurant_feedback`**

```
INSERT INTO fact_restaurant_feedback PARTITION (feedback_date) VALUES

-- Feedback on 2022-01-02

(7001, 5001, 2001, 'Order prepared on time.', 'Resolved',
'2022-01-02'),

(7002, 5002, 2002, 'Customer requested extra ginger.', 'Resolved',
'2022-01-02'),

-- Feedback on 2022-01-03

(7003, 5003, 2003, 'No issues.', 'Resolved', '2022-01-03'),

-- Feedback on 2022-01-04

(7004, 5004, 2004, 'Delivery partner arrived late.', 'Pending',
'2022-01-04'),

(7005, 5005, 2005, 'Order delayed due to high volume.', 'Resolved',
'2022-01-04'),

-- Feedback on 2022-01-05
```

```
(7006, 5006, 2001, 'Repeat customer, prioritize order.', 'Resolved',
'2022-01-05');
```

Explanation:

- Provides feedback from restaurants regarding orders.
- Includes resolution status for tracking.
- Useful for operational analysis.