**Data Warehousing and Data Mining.**

**Name : Antuley Aman Siraj.**

**Roll No. : 23CO25.**

**Class : TE-CO**

**Batch : 01**

**Experiment - 04**

### **Aim :**

Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot on **Online Food Delivery System** in terms of Queries.

**Theory :**OLAP stands for Online Analytical Processing Server. It is a software technology that allows users to analyze information from multiple database systems at the same time. It is based on multidimensional data model and allows the user to query on multi-dimensional data (eg. Delhi > 2018 -> Sales data). OLAP databases are divided into one or more cubes and these cubes are known as Hyper-cubes.

**Dimension Tables :**

1. **User Dimension :**

CREATE TABLE user\_dim (

user\_id INT PRIMARY KEY,

user\_name VARCHAR(100),

city VARCHAR(50)

);

1. **Restaurant Dimension :**

CREATE TABLE restaurant\_dim (

restaurant\_id INT PRIMARY KEY,

restaurant\_name VARCHAR(100),

cuisine\_type VARCHAR(50)

);

1. **Delivery Dimension :**

CREATE TABLE delivery\_dim (

delivery\_id INT PRIMARY KEY,

partner\_name VARCHAR(100),

delivery\_time\_minutes INT

);

1. **Payment Dimension :**

CREATE TABLE payment\_dim (

payment\_id INT PRIMARY KEY,

payment\_method VARCHAR(50),

status VARCHAR(20)

);

1. **Date Dimension :**

CREATE TABLE date\_dim (

date\_id INT PRIMARY KEY,

order\_date DATE,

day\_of\_week VARCHAR(20),

month VARCHAR(20),

year INT

);

**Fact Table :**

CREATE TABLE order\_fact (

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT,

restaurant\_id INT,

delivery\_id INT,

payment\_id INT,

date\_id INT,

order\_count INT,

FOREIGN KEY (user\_id) REFERENCES user\_dim(user\_id),

FOREIGN KEY (restaurant\_id) REFERENCES restaurant\_dim(restaurant\_id),

FOREIGN KEY (delivery\_id) REFERENCES delivery\_dim(delivery\_id),

FOREIGN KEY (payment\_id) REFERENCES payment\_dim(payment\_id),

FOREIGN KEY (date\_id) REFERENCES date\_dim(date\_id)

);

**Inserting the Data :**

**Users :**

INSERT INTO user\_dim (user\_name, city) VALUES

('Aman', 'Pune'),

('XYZ', 'Mumbai'),

('ABC', 'Delhi');

**Restaurants :**

INSERT INTO restaurant\_dim (restaurant\_name, cuisine\_type) VALUES

('Dominos', 'Pizza'),

('Biryani House', 'Indian'),

('KFC', 'Fast Food');

**Deliveries :**

INSERT INTO delivery\_dim (partner\_name, delivery\_time\_minutes) VALUES

('Zomato', 30),

('Swiggy', 25),

('UberEats', 40);

**Payments :**

INSERT INTO payment\_dim (payment\_method, status) VALUES

('UPI', 'Success'),

('Credit Card', 'Success'),

('Cash', 'Pending');

**Dates :**

INSERT INTO date\_dim (order\_date, day\_of\_week, month, year) VALUES

('2025-08-20', 'Wednesday', 'August', 2025),

('2025-08-21', 'Thursday', 'August', 2025),

('2025-08-22', 'Friday', 'August', 2025);

**Fact Table (Orders) :**

INSERT INTO order\_fact (user\_id, restaurant\_id, delivery\_id, payment\_id, date\_id, order\_amount, order\_count) VALUES

(1, 1, 1, 1, 1, 500.00, 2),

(2, 2, 2, 2, 2, 750.00, 3),

(3, 3, 3, 3, 3, 1200.00, 4);

**Output :**

+-----------------------+

| Tables\_in\_aman |

+-----------------------+

| date\_dim |

| delivery\_dim |

| order\_fact |

| payment\_dim |

| restaurant\_dim |

| user\_dim |

+-----------------------+

1. **User :**

| **user\_id** | **user\_name** | **city** |
| --- | --- | --- |
| 1 | Aman | Mumbai |
| 2 | XYZ | Pune |
| 3 | ABC | Delhi |

1. **Restaurant :**

| **restaurant\_id** | **restaurant\_name** | **cuisine\_type** |
| --- | --- | --- |
| 1 | Dominos | Pizza |
| 2 | Biryani House | Indian |
| 3 | KFC | Fast Food |

1. **Deliveries :**

| **delivery\_id** | **partner\_name** | **delivery\_time\_minutes** |
| --- | --- | --- |
| 1 | Zomato | 30 |
| 2 | Swiggy | 25 |
| 3 | UberEats | 40 |

1. **Payments :**

| **payment\_id** | **payment\_method** | **status** |
| --- | --- | --- |
| 1 | UPI | Success |
| 2 | Credit Card | Success |
| 3 | Cash | Pending |

1. **Dates :**

| **date\_id** | **order\_date** | **day\_of\_week** | **month** | **year** |
| --- | --- | --- | --- | --- |
| 1 | 2025-08-20 | Wednesday | August | 2025 |
| 2 | 2025-08-21 | Thursday | August | 2025 |
| 3 | 2025-08-22 | Friday | August | 2025 |

**Fact Table (Orders) :**

| **order\_id** | **user\_id** | **restaurant\_id** | **delivery\_id** | **payment\_id** | **date\_id** | **order\_count** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 4 |

**OLAP operations:**

1. **Roll up:**

It is just opposite of the drill-down operation. It performs ag gregation on the OLAP cube. It can be done by: Climbing up in the concept hierarchy Reducing the dimensions In the cube given in the overview section, the roll-up operation is performed by climbing up in the concept hierarchy of Location dimension (City -> Country)

**SELECT restaurant\_id, date\_id, SUM(order\_count) AS daily\_orders**

**FROM order\_fact**

**GROUP BY restaurant\_id, date\_id;**

| **restaurant\_id** | **date\_id** | **daily\_orders** |
| --- | --- | --- |
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |

2. **Drill down:**

In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by: Moving down in the concept hierarchy Adding a new dimension In the cube given in the overview section, the drill down operation is performed by moving down in the concept hierarchy of Time dimension.

**SELECT date\_id, payment\_id, SUM(order\_count) AS orders\_by\_payment**

**FROM order\_fact**

**GROUP BY date\_id, payment\_id;**

| **date\_id** | **payment\_id** | **orders\_by\_payment** |
| --- | --- | --- |
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |

3. **Dice:**

It selects a sub-cube from the OLAP cube by selecting two or more dimensions. In the cube given in the overview section, a sub cube is selected by selecting following dimensions with criteria:

* Location = “Delhi” or “Kolkata”
* Time = “Q1” or “Q2”
* Item = “Car” or “Bus”

**SELECT date\_id, restaurant\_id, SUM(order\_count) AS total\_orders**

**FROM order\_fact**

**WHERE restaurant\_id = 1 AND payment\_id = 1**

**GROUP BY date\_id, restaurant\_id;**

| **date\_id** | **restaurant\_id** | **total\_orders** |
| --- | --- | --- |
| 1 | 1 | 2 |

4. **Slice:**

It selects a single dimension from the OLAP cube which results in a new sub-cube creation. In the cube given in the overview section, Slice is performed on the dimension Time = “Q1”.

**SELECT date\_id, SUM(order\_count) AS total\_orders**

**FROM order\_fact**

**WHERE payment\_id = 1 -- slice on payment method (e.g., UPI)**

**GROUP BY date\_id;**

| **date\_id** | **total\_orders** |
| --- | --- |
| 1 | 2 |

5. **Pivot:**

It is also known as rotation operation as it rotates the current view to get a new view of the representation. In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.

**SELECT restaurant\_id,**

**SUM(CASE WHEN payment\_id = 1 THEN order\_count ELSE 0 END) AS UPI\_Orders,**

**SUM(CASE WHEN payment\_id = 2 THEN order\_count ELSE 0 END) AS Card\_Orders,**

**SUM(CASE WHEN payment\_id = 3 THEN order\_count ELSE 0 END) AS Cash\_Orders**

**FROM order\_fact**

**GROUP BY restaurant\_id;**

| **restaurant\_id** | **UPI\_Orders** | **Card\_Orders** | **Cash\_Orders** |
| --- | --- | --- | --- |
| 1 | 2 | 0 | 0 |
| 2 | 0 | 3 | 0 |
| 3 | 0 | 0 | 4 |

**Conclusion:** Hence we have studied the various OLAP Operations.