Aim:

To design and implement a data warehouse for a customer order processing system in a company.

Introduction

The objective of this project is to design and implement a data warehouse system to help an enterprise that operates in multiple cities and states to effectively manage their stores, customers, and inventory. The system is expected to provide analytical processing features such as roll up, drill down, slice, and dice to meet user requirements

Business Requirements

The system is expected to answer nine queries which include,

1. Find all the stores along with city, state, phone, description, size, weight and unit price that hold a particular item of stock.

2. Find all the orders along with customer name and order date that can be fulfilled by a given store.

3. Find all stores along with city name and phone that hold items ordered by given customer.

4. Find the headquarter address along with city and state of all stores that hold stocks of an item above a particular level.

5. For each customer order, show the items ordered along with description, store id and city name and the stores that hold the items.

6. Find the city and the state in which a given customer lives.

7. Find the stock level of a particular item in all stores in a particular city.

8. Find the items, quantity ordered, customer, store and city of an order.

9. Find the walk-in customers, mail order customers and dual customers (both walk-in and mail order).

Functional Specification

The system will have input and output specifications that will allow users to select dimensions and perform analysis using roll up, drill down, slice, and dice features.

Data Warehousing Design

The data warehousing design will follow a stepwise procedure methodology of designing the data warehouse, including the creation of a star schema. The schema will have fact tables and dimension tables.

Data Cube Implementation

The loading of data into data cubes will be automated using computer systems. The system will generate data cubes and populate them with data from the source databases.

Observations

a. Online analytical processing reports: The system will provide users with the ability to generate OLAP reports using commands or panels.

b. Data verification: The system will verify the data used to generate OLAP reports by comparing the results with the source relational tables' data.

Program:

DATA WAREHOUSE

To create data warehouse, we need fact table and dimension tables;

Dimension tables:

Store_dim (store_id, city_id, phone, description, size, weight, unit_price)

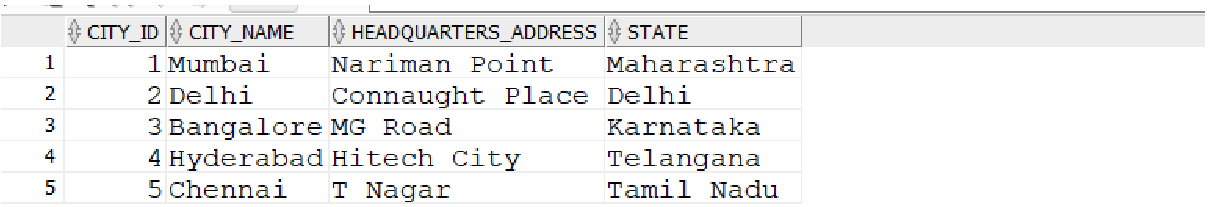
Customer_dim (customer_id, customer_name, city_id)

Item_dim (item_id, description, size, weight, unit_price)

Order_dim (order_no, order_date, customer_id)

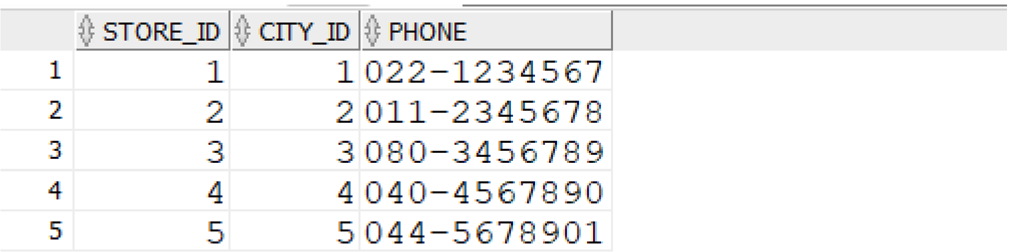
City_dim (city_id, city_name, state, headquarter_addr)

```
CREATE TABLE city_dim (
   city_id INT PRIMARY KEY,
   city_name VARCHAR(50),
   headquarters_address VARCHAR(100),
   state VARCHAR(50)
);

CREATE TABLE store_dim (
   store_id INT PRIMARY KEY,
   city_id INT,

   phone VARCHAR(20),

   FOREIGN KEY (city_id) REFERENCES city_dim(city_id)

);
```



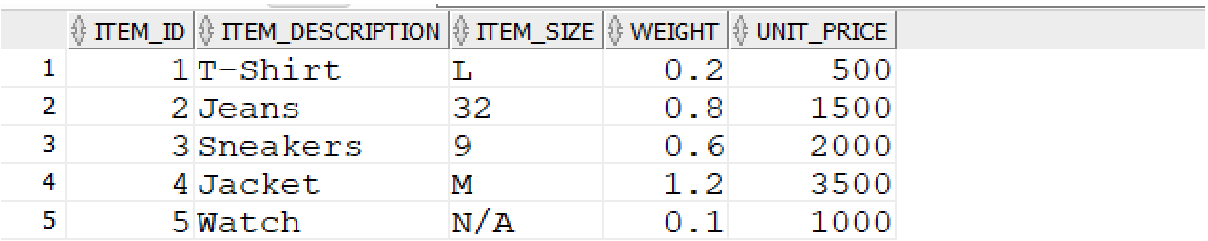
	CITY_ID	CITY_NAME	HEADQUARTERS_ADDRESS	STATE
1	1	Mumbai	Nariman Point	Maharashtra
2	2	Delhi	Connaught Place	Delhi
3	3	Bangalore	MG Road	Karnataka
4	4	Hyderabad	Hitech City	Telangana
5	5	Chennai	T Nagar	Tamil Nadu

```
CREATE TABLE item_dim (

    item_id INT PRIMARY KEY,

    item_description VARCHAR(100),

    item_size VARCHAR(20),

    weight NUMBER(10,2),

    unit_price NUMBER(10,2)

);
```



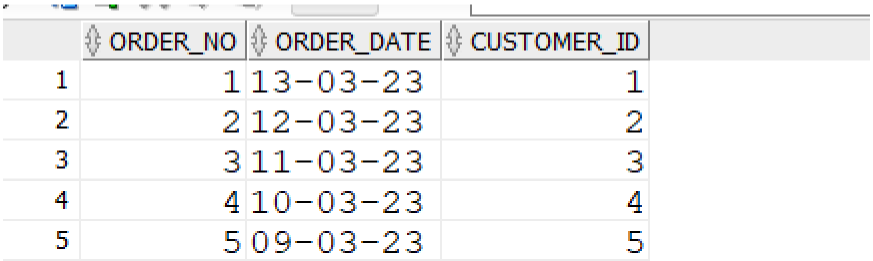
	STORE_ID	CITY_ID	PHONE
1	1	1	022-1234567
2	2	2	011-2345678
3	3	3	080-3456789
4	4	4	040-4567890
5	5	5	044-5678901

```
CREATE TABLE stored_item_dim (

    store_id INT,
    item_id INT,
    quantity_held INT,
    FOREIGN KEY (store_id) REFERENCES store_dim(store_id),
    FOREIGN KEY (item_id) REFERENCES item_dim(item_id)
);
```



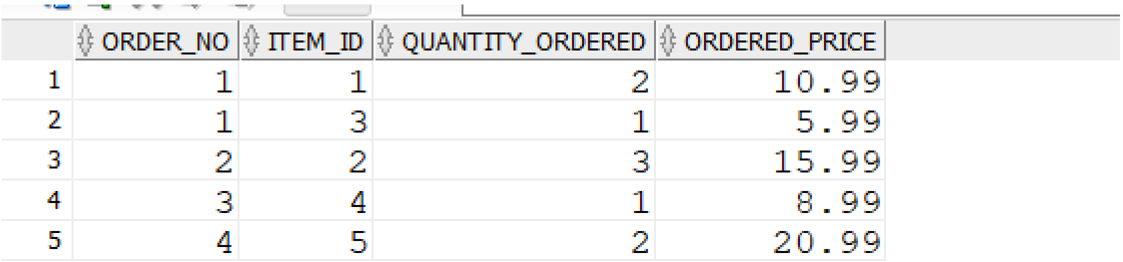
	ITEM_ID	ITEM_DESCRIPTION	ITEM_SIZE	WEIGHT	UNIT_PRICE
1	1	T-Shirt	L	0.2	500
2	2	Jeans	32	0.8	1500
3	3	Sneakers	9	0.6	2000
4	4	Jacket	M	1.2	3500
5	5	Watch	N/A	0.1	1000

```
CREATE TABLE order_dim (
    order_no INT PRIMARY KEY,
    order_date DATE,
    customer_id INT
);
```



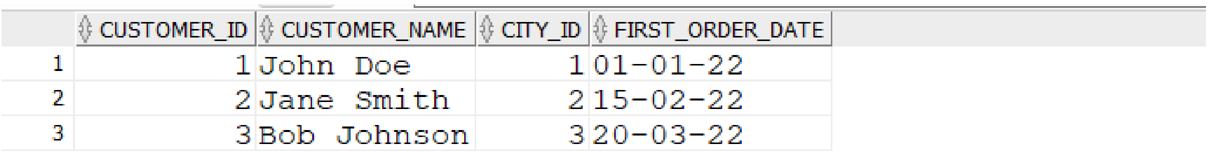
	ORDER_NO	ORDER_DATE	CUSTOMER_ID
1	1	13-03-23	1
2	2	12-03-23	2
3	3	11-03-23	3
4	4	10-03-23	4
5	5	09-03-23	5

```
CREATE TABLE ordered_item_dim (
    order_no INT,
    item_id INT,
    quantity_ordered INT,
    ordered_price DECIMAL(10,2),
    FOREIGN KEY (order_no) REFERENCES order_dim(order_no),
    FOREIGN KEY (item_id) REFERENCES item_dim(item_id)
);
```



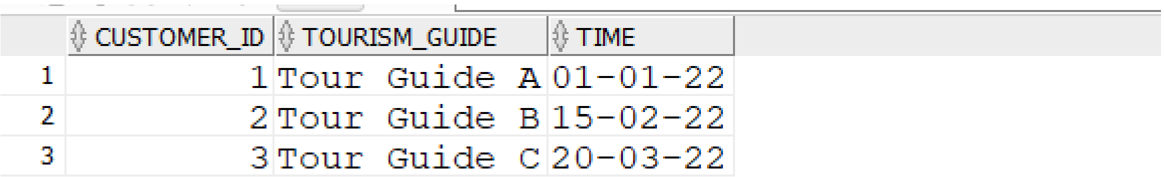
	ORDER_NO	ITEM_ID	QUANTITY_ORDERED	ORDERED_PRICE
1	1	1	2	10.99
2	1	3	1	5.99
3	2	2	3	15.99
4	3	4	1	8.99
5	4	5	2	20.99

```
drop table customer_dim;

CREATE TABLE customer_dim (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(50),
    city_id INT,
    first_order_date DATE
);
```



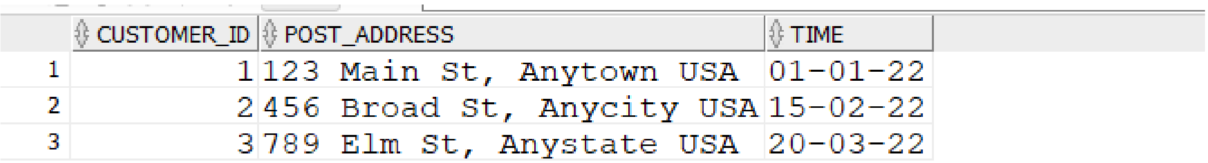
	CUSTOMER_ID	CUSTOMER_NAME	CITY_ID	FIRST_ORDER_DATE
1	1	John Doe	1	01-01-22
2	2	Jane Smith	2	15-02-22
3	3	Bob Johnson	3	20-03-22

```
CREATE TABLE walk_in_customers_dim (
   customer_id INT,
   tourism_guide VARCHAR(50),
   time DATE,
   PRIMARY KEY (customer_id)
);
```



	CUSTOMER_ID	TOURISM_GUIDE	TIME
1	1	Tour Guide A	01-01-22
2	2	Tour Guide B	15-02-22
3	3	Tour Guide C	20-03-22

```
CREATE TABLE mail_order_customers_dim (
   customer_id INT,
   post_address VARCHAR(100),
   time DATE,
   PRIMARY KEY (customer_id)
);
```



	CUSTOMER_ID	POST_ADDRESS	TIME
1	1	123 Main St, Anytown USA	01-01-22
2	2	456 Broad St, Anycity USA	15-02-22
3	3	789 Elm St, Anystate USA	20-03-22

```
INSERT INTO customer_dim VALUES (1, 'John Doe', 1, TO_DATE('2022-01-01', 'YYYY-MM-DD'));
INSERT INTO customer_dim VALUES (2, 'Jane Smith', 2, TO_DATE('2022-02-15', 'YYYY-MM-DD'));
INSERT INTO customer_dim VALUES (3, 'Bob Johnson', 3, TO_DATE('2022-03-20', 'YYYY-MM-DD'));

INSERT INTO walk_in_customers_dim VALUES (1, 'Tour Guide A', TO_DATE('2022-01-01', 'YYYY-MM-DD'));
INSERT INTO walk_in_customers_dim VALUES (2, 'Tour Guide B', TO_DATE('2022-02-15', 'YYYY-MM-DD'));
INSERT INTO walk_in_customers_dim VALUES (3, 'Tour Guide C', TO_DATE('2022-03-20', 'YYYY-MM-DD'));


INSERT INTO mail_order_customers_dim VALUES (1, '123 Main St, Anytown USA', TO_DATE('2022-01-01', 'YYYY-MM-DD'));

INSERT INTO mail_order_customers_dim VALUES (2, '456 Broad St, Anycity USA', TO_DATE('2022-02-15', 'YYYY-MM-DD'));

INSERT INTO mail_order_customers_dim VALUES (3, '789 Elm St, Anystate USA', TO_DATE('2022-03-20', 'YYYY-MM-DD'));


INSERT INTO city_dim (city_id, city_name, headquarters_address, state)

VALUES (1, 'Mumbai', 'Nariman Point', 'Maharashtra');


INSERT INTO city_dim (city_id, city_name, headquarters_address, state)

VALUES (2, 'Delhi', 'Connaught Place', 'Delhi');


INSERT INTO city_dim (city_id, city_name, headquarters_address, state)

VALUES (3, 'Bangalore', 'MG Road', 'Karnataka');


INSERT INTO city_dim (city_id, city_name, headquarters_address, state)

VALUES (4, 'Hyderabad', 'Hitech City', 'Telangana');


INSERT INTO city_dim (city_id, city_name, headquarters_address, state)

VALUES (5, 'Chennai', 'T Nagar', 'Tamil Nadu');


INSERT INTO store_dim (store_id, city_id, phone)

VALUES (1, 1, '022-1234567');


INSERT INTO store_dim (store_id, city_id, phone)

VALUES (2, 2, '011-2345678');


INSERT INTO store_dim (store_id, city_id, phone)
```

```
VALUES (3, 3, '080-3456789');

INSERT INTO store_dim (store_id, city_id, phone)
VALUES (4, 4, '040-4567890');

INSERT INTO store_dim (store_id, city_id, phone)
VALUES (5, 5, '044-5678901');

INSERT INTO item_dim (item_id, item_description, item_size, weight, unit_price)
VALUES (1, 'T-Shirt', 'L', 0.2, 500.00);

INSERT INTO item_dim (item_id, item_description, item_size, weight, unit_price)
VALUES (2, 'Jeans', '32', 0.8, 1500.00);

INSERT INTO item_dim (item_id, item_description, item_size, weight, unit_price)
VALUES (3, 'Sneakers', '9', 0.6, 2000.00);

INSERT INTO item_dim (item_id, item_description, item_size, weight, unit_price)
VALUES (4, 'Jacket', 'M', 1.2, 3500.00);

INSERT INTO item_dim (item_id, item_description, item_size, weight, unit_price)
VALUES (5, 'Watch', 'N/A', 0.1, 1000.00);



INSERT INTO stored_item_dim (store_id, item_id, quantity_held)
VALUES (1, 1, 100);

INSERT INTO stored_item_dim (store_id, item_id, quantity_held)
VALUES (1, 2, 50);

INSERT INTO stored_item_dim (store_id, item_id, quantity_held)
```

```
VALUES (2, 2, 75);

INSERT INTO stored_item_dim (store_id, item_id, quantity_held)
VALUES (3, 3, 60);

INSERT INTO stored_item_dim (store_id, item_id, quantity_held)
VALUES (4, 4, 25);

INSERT INTO order_dim (order_no, order_date, customer_id) VALUES (1, TO_DATE('2023-03-13', 'YYYY-MM-DD'), 1);
INSERT INTO order_dim (order_no, order_date, customer_id) VALUES (2, TO_DATE('2023-03-12', 'YYYY-MM-DD'), 2);
INSERT INTO order_dim (order_no, order_date, customer_id) VALUES (3, TO_DATE('2023-03-11', 'YYYY-MM-DD'), 3);
INSERT INTO order_dim (order_no, order_date, customer_id) VALUES (4, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 4);
INSERT INTO order_dim (order_no, order_date, customer_id) VALUES (5, TO_DATE('2023-03-09', 'YYYY-MM-DD'), 5);

INSERT INTO ordered_item_dim (order_no, item_id, quantity_ordered, ordered_price) VALUES (1, 1, 2, 10.99);
INSERT INTO ordered_item_dim (order_no, item_id, quantity_ordered, ordered_price) VALUES (1, 3, 1, 5.99);
INSERT INTO ordered_item_dim (order_no, item_id, quantity_ordered, ordered_price) VALUES (2, 2, 3, 15.99);
INSERT INTO ordered_item_dim (order_no, item_id, quantity_ordered, ordered_price) VALUES (3, 4, 1, 8.99);
INSERT INTO ordered_item_dim (order_no, item_id, quantity_ordered, ordered_price) VALUES (4, 5, 2, 20.99);


CREATE TABLE sales_fact (
sales_id INT PRIMARY KEY,
store_id INT,
item_id INT,
order_no INT,
customer_id INT,
customer_type VARCHAR(20),
order_date DATE,
quantity_ordered INT,
ordered_price DECIMAL(10,2),
FOREIGN KEY (store_id) REFERENCES store_dim(store_id),
FOREIGN KEY (item_id) REFERENCES item_dim(item_id),
FOREIGN KEY (order_no) REFERENCES order_dim(order_no),
FOREIGN KEY (customer_id) REFERENCES customer_dim(customer_id)
);
```

Find all the stores along with city, state, phone, description, size, weight and unit price that hold a particular item of stock.

```
SELECT s.store_id, c.city_name, c.state, s.phone, i.item_description, i.item_size, i.weight, i.unit_price
FROM store_dim s
JOIN city_dim c ON s.city_id = c.city_id
JOIN stored_item_dim si ON s.store_id = si.store_id
JOIN item_dim i ON si.item_id = i.item_id
WHERE i.item_description = 'particular item';
```

Find all the orders along with customer name and order date that can be fulfilled by a given store.

```
SELECT o.order_no, c.customer_name, o.order_date
FROM order_dim o
JOIN customer_dim c ON o.customer_id = c.customer_id
JOIN sales_fact sf ON o.order_no = sf.order_no
WHERE sf.store_id = given_store_id;
```

Find all stores along with city name and phone that hold items ordered by given customer.

```
SELECT DISTINCT s.store_id, c.city_name, s.phone
```

```
FROM store_dim s
JOIN city_dim c ON s.city_id = c.city_id
JOIN sales_fact sf ON s.store_id = sf.store_id
WHERE sf.customer_id
```

Fact table:

Order_facts (order_no, store_id, item_id, quantity_ordered, ordered_price)

```
CREATE TABLE sales_fact (
sales_id INT PRIMARY KEY,
store_id INT,
item_id INT,
order_no INT,
customer_id INT,
customer_type VARCHAR(20),
order_date DATE,
quantity_ordered INT,
ordered_price DECIMAL(10,2),
FOREIGN KEY (store_id) REFERENCES store_dim(store_id),
FOREIGN KEY (item_id) REFERENCES item_dim(item_id),
FOREIGN KEY (order_no) REFERENCES order_dim(order_no),
FOREIGN KEY (customer_id) REFERENCES customer_dim(customer_id)
);
```

Queries:

1. **Find all the stores along with city, state, phone, description, size, weight and unit price that hold a particular item of stock.**

```
SELECT s.store_id, c.city_name, c.state, s.phone, i.item_description, i.item_size, i.weight, i.unit_price
FROM store_dim s
JOIN city_dim c ON s.city_id = c.city_id
JOIN stored_item_dim si ON s.store_id = si.store_id

JOIN item_dim i ON si.item_id = i.item_id

WHERE i.item_description = 'T-Shirt';
```

2. **Find all the orders along with customer name and order date that can be fulfilled by a given store.**

```
SELECT o.order_no, c.customer_name, o.order_date

FROM order_dim o

JOIN customer_dim c ON o.customer_id = c.customer_id

JOIN sales_fact sf ON o.order_no = sf.order_no

WHERE sf.store_id = 101;
```

3. **Find all stores along with city name and phone that hold items ordered by given customer.**

```
SELECT DISTINCT s.store_id, c.city_name, s.phone

FROM store_dim s

JOIN city_dim c ON s.city_id = c.city_id

JOIN sales_fact sf ON s.store_id = sf.store_id

WHERE sf.customer_id = 1;
```

4. **Find the headquarter address along with city and state of all stores that hold stocks of an item above a particular level.**

```
SELECT DISTINCT c.headquarters_address, c.city_name, c.state

FROM city_dim c

JOIN store_dim s ON c.city_id = s.city_id

JOIN stored_item_dim si ON s.store_id = si.store_id

JOIN item_dim i ON si.item_id = i.item_id

WHERE i.item_description = 'T-Shirt' AND si.quantity_held > 1;
```

5. **For each customer order, show the items ordered along with description, store id and city name and the stores that hold the items.**

```
SELECT o.order_no, i.item_description, s.store_id, c.city_name
```

```
FROM order_dim o
JOIN sales_fact sf ON o.order_no = sf.order_no
JOIN store_dim s ON sf.store_id = s.store_id
JOIN city_dim c ON s.city_id = c.city_id
JOIN item_dim i ON sf.item_id = i.item_id
WHERE sf.customer_id = 1;
```

6. **Find the city and the state in which a given customer lives.**

```
SELECT c.customer_name, ci.city_name, ci.state
FROM customer_dim c
JOIN city_dim ci ON c.city_id = ci.city_id
WHERE c.customer_name = 'Jane Smith';
```

7. **Find the stock level of a particular item in all stores in a particular city.**

```
SELECT si.quantity_held
FROM sales_fact s
JOIN store_dim sd ON s.store_id = sd.store_id
JOIN city_dim c ON sd.city_id = c.city_id
JOIN stored_item_dim si ON sd.store_id = si.store_id
JOIN item_dim i ON si.item_id = i.item_id
WHERE c.city_name = 'Mumbai' AND i.item_description = 'Jeans';
```

8. **Find the items, quantity ordered, customer, store and city of an order.**

```
SELECT o.order_no, c.customer_name, s.store_id, s.city_id, i.item_description, si.quantity_held, oi.quantity_ordered
FROM sales_fact sf
JOIN order_dim o ON sf.order_no = o.order_no
JOIN customer_dim c ON o.customer_id = c.customer_id
JOIN store_dim s ON sf.store_id = s.store_id
JOIN city_dim cd ON s.city_id = cd.city_id

JOIN stored_item_dim si ON s.store_id = si.store_id

JOIN ordered_item_dim oi ON sf.order_no = oi.order_no AND sf.item_id = oi.item_id

JOIN item_dim i ON oi.item_id = i.item_id

WHERE cd.city_name = 'Mumbai' AND i.item_description = 'Jeans';
```

9. **Find the walk-in customers, mail order customers and dual customers (both walk-in and mail order).**

```
SELECT c.customer_id, c.customer_name,

  CASE

    WHEN w.customer_id IS NOT NULL AND m.customer_id IS NOT NULL THEN 'dual'

    WHEN w.customer_id IS NOT NULL THEN 'walk-in'

    WHEN m.customer_id IS NOT NULL THEN 'mail-order'

  END AS type

FROM customer_dim c

LEFT JOIN walk_in_customers_dim w ON c.customer_id = w.customer_id

LEFT JOIN mail_order_customers_dim m ON c.customer_id = m.customer_id;
```

Conclusion

The data warehouse system will provide an effective way for the enterprise to manage their stores, customers, and inventory. The system will provide analytical processing features to help users analyse data and make informed decisions.