ASSIGNMENT 2 : *To design and implement a data warehouse for a customer order processing system in a company.*

DATE : *17-09-2020* **SUBMITTED BY** : *DWDM20G05*

INTRODUCTION (OBJECTIVE & SCOPE):

A data warehouse is a system that pulls together data from many different sources within an organization for reporting and analysis. The reports created from complex queries within a data warehouse are used to make business decisions.

In more comprehensive terms, a data warehouse is a consolidated view of either a physical or logical data repository collected from various systems. The primary focus of a data warehouse is to provide a correlation between data from existing systems, i.e., product inventory stored in one system purchase orders for a specific customer, stored in another system. Data warehouses are used for online analytical processing (OLAP), which uses complex queries to analyze rather than process transactions.

BUSINESS REQUIREMENT:

In order to meet users' demand, the data warehouse system extracts data from the existing two databases into a data warehouse, and provides online analytical processing with roll up, drill down, slice and dice features according to users selections based on dimension tables to meet the user requirements.

FUNCTIONAL SPECIFICATION:

The target of a data warehouse system is an enterprise that consists of a number of stores located in different cities and states. Each store holds a variety of items in various quantities. In addition, the enterprise keeps the information of the customers. There are two kinds of customers: walk-in led by a tourism guide and mail-order by post address inclusive. The city location of the customer, together with the data of the customer's first order, is stored by the existing system. Each customer lives in one city only, and the enterprise will try to satisfy the customer's order items by the present stock in the city where the customer lives. Each customer order can be for any quantity of any number of items, and each order is uniquely identified by an order number.

The location of the stores is also recorded. Each store is located in one city, and there can be many stores in the city. Each city has one headquarter for coordinating all of its stores. The enterprise's goal is to meet all of the customer's requirements from stores located in the customer's city. If the requirement cannot be met, the company will turn to the other cities where the item can be found if there is any.

Some processing information is important for the enterprise. For example, the total quantity of item stored in each city. After every time an item is taken, the company needs to know the total quantities of the item in all the stores in a city.

DATA WAREHOUSE DESIGN:

Headquarter Database:

Relation Customer (Customer_id, Customer_name, City_id, First_order_date)

Relation Walk-in_customers (*Customer_id, tourism_guide, Time)

Relation Mail_order_customers (*Customer_id, post_address, Time)

Sales Databases:
Relation Headquarters (City_id, City_name, Headquarter_addr, State, Time)
Relation Stores (Store_id, *City_id, Phone, Time)
Relation Items (Item_id, Description, Size, Weight, Unit_price, Time)
Relation Stored_items (*Store_id, *Item_id, Qantity_held, Time)
Relation Order (Order_no, Order_date, Customer_id)
Relation Ordered_item (*Order_no, *Item_id, Quantity_ordered, Ordered_price, Time)

Derived & fact table (OLAP):
store_item_details(store_id,city_name,state,phone,description,size,weight,unit_price)
customer_order_details(order_no,order_date,customer_name,item_id)
customer_order_store_details(store_id,city_name,phone,item_id,customer_id,customer_name)
store_headquarter_item_details(store_id,city_name,headquarter_addr,state,item_id,qantity_held,SI_Time)
create table order_facts_table(fact_id integer primary key, item_id integer,store_id integer,city_id integer,quantity integer, _Time datetime);

OBSERVATIONS:
1.Find all the stores along with city, state, phone, description, size, weight and unit price that holds a particular item of stock.
create table store_item_details(store_id int,city_name varchar(25),state varchar(25) ,phone varchar(10)
,description varchar(100),size int,weight double,unit_price double);
insert into store_item_details(store_id,city_name,state,phone,description,size,weight,unit_price)
select st.store_id,hd.city_name,state,phone,description,size,weight,unit_price from headquarters hd
inner join stores st on hd.city_id = st.city_id
inner join stored_items sti on sti.store_id = st.store_id
inner join items it on it.item_id = sti.item_id;
select * from store_item_details;

store_id	city_name	state	phone	description	size	weight	unit_price
1	Pune	Maharastra	9921990090	Vaccume Cleaner	10	12.5	100
1	Pune	Maharastra	9921990090	Laptop	7	1.25	200
2	Pune	Maharastra	7914545645	Refrigerator	25	42	300
2	Pune	Maharastra	7914545645	Table	12	27	400
3	Pune	Maharastra	7814545645	Vaccume Cleaner	10	12.5	100
3	Pune	Maharastra	7814545645	Laptop	7	1.25	200
4	Mumbai	Mahaarastra	8714545645	Refrigerator	25	42	300
4	Mumbai	Mahaarastra	8714545645	Table	12	27	400
5	Mumbai	Mahaarastra	9714545645	Television	15	47	500
6	Sangli	Maharastra	984545645	Vaccume Cleaner	10	12.5	100

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

```
insert into customer_order_details
select od.order_no,order_date,customer_name,oi.item_id from _order od
inner join customer cus on od.customer_id = cus.customer_id
inner join ordered_item oi on od.order_no = oi.order_no
inner join stored_items sti on sti.item_id = oi.item_id;
select * from customer_order_details;
```

order_no	order_date	customer_name	item_id
1	2020-05-21	Aniket	1
1	2020-05-21	Aniket	5
2	2020-07-18	Aniket	2
2	2020-07-18	Aniket	3
2	2020-07-18	Aniket	4

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

```
insert into customer_order_Store_details
select sid.store_id,city_name,phone,si.item_id,customer_id,cod.customer_name from
store_item_details sid
inner join stored_items si on sid.store_id = si.store_id
inner join customer_order_details cod on si.item_id = cod.item_id
inner join customer cus on cus.customer_name  = cod.customer_name;
select store_id,city_name,phone from customer_order_store_details where item_id in (select
item_id from customer_order_details);
```

store_id	city_name	phone
1	Pune	9921990090
2	Pune	7914545645
3	Pune	7814545645
4	Mumbai	8714545645
5	Mumbai	9714545645
6	Sangli	984545645
7	Sangli	7614545645
8	Sangli	7514545645
9	Bangalore	7114545645
10	Bangalore	9514545645
11	Bangalore	9714545645
12	Delhi	8514545645
13	Delhi	8114545645

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

```
insert into store_headquarter_item_details
select distinct st.store_id,city_name,headquarter_addr,state,item_id,Qantity_held,si._time from
headquarters hq
inner join stores st on st.city_id = hq.city_id
```

```
inner join stored_items si on st.store_id = si.store_id;
select headquarter_addr,city_name,state,store_id,sum(qantity_held)
 from store_headquarter_item_details where item_id = 2 group by
headquarter_addr,city_name,state,store_id
with rollup having sum(qantity_held) > 50;
```

headquarter_addr	city_name	state	store_id	sum(qantity_held)
Shivaji Nagar	Pune	Maharastra	3	100
Shivaji Nagar	Pune	Maharastra	NULL	140
Shivaji Nagar	Pune	NULL	NULL	140
Shivaji Nagar	NULL	NULL	NULL	140
NULL	NULL	NULL	NULL	203

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 Order_no,item_id, Description,r.store_id,city_name from (select order_no,
items.item_id,items.Description,w.store_id from
((ordered_item join items on items.Item_id = ordered_item.Item_id ) inner join stored_items as
w)  inner join
(select store_id,city_name from stores inner join headquarters as x ) as y on w.store_id =
y.store_id ) as r join
(select store_id,city_name from stores join headquarters on headquarters.City_id =
stores.City_id) as q on q.store_id = r. store_id
order by order_no,item_id;
```

order_no	item_id	Description	store_id	city_name
1	1	Vaccume Cleaner	8	Sangli
1	1	Vaccume Cleaner	8	Sangli
1	1	Vaccume Cleaner	8	Sangli
1	1	Vaccume Cleaner	8	Sangli
1	1	Vaccume Cleaner	10	Bangalore
1	1	Vaccume Cleaner	10	Bangalore
1	1	Vaccume Cleaner	10	Bangalore
1	1	Vaccume Cleaner	10	Bangalore
1	1	Vaccume Cleaner	10	Bangalore
1	5	Television	1	Pune
1	5	Television	1	Pune
1	5	Television	1	Pune
1	5	Television	1	Pune
1	5	Television	1	Pune

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

```
select customer_id,customer_name,City_name,State from customer join headquarters on
customer.city_id = headquarters.city_id;
```

▶	1	Aniket	Pune	Maharastra
	2	Harshraj	Mumbai	Mahaarastra
	3	Neha	Sangli	Maharastra
	4	Shyam	Pune	Maharastra
	5	Shridhar	Mumbai	Mahaarastra
	6	Sagar	Bangalore	Karnataka
	7	Ajinkya	Pune	Maharastra
	8	Shreyash	Mumbai	Mahaarastra
	9	Saurabh	Sangli	Maharastra
	10	Rahul	Bangalore	Karnataka

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

select item_id,_Time,sum(Qantity_held) from stored_items where item_id =1 group by _Time;

	item_id	_Time	sum(Qantity_held)
▶	1	2020-01-01 00:00:00	230

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

select distinct Customer_name,city_id,Qantity_held,description,s.store_id
From _Order ,Customer,Ordered_Item,Items, Stored_items s
where _Order.Order_no =Ordered_Item.Order_no and
Customer.Customer_id=_Order.customer_id
and Ordered_Item.Item_id= Items.Item_id and S.item_id= Items.item_id;

Neha	3	5	Refrigerator	12
Harshraj	2	50	Vaccume Cleaner	1
Harshraj	2	50	Vaccume Cleaner	3
Harshraj	2	5	Vaccume Cleaner	6
Harshraj	2	25	Vaccume Cleaner	8
Harshraj	2	100	Vaccume Cleaner	11
Shridhar	2	50	Table	2
Shridhar	2	50	Table	4
Shridhar	2	50	Table	7
Shridhar	2	80	Table	9
Shridhar	2	5	Table	13
Shyam	1	50	Table	2
Shyam	1	50	Table	4
Shyam	1	50	Table	7

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

select customer.customer_id,customer_name, "Walk in" as Type from customer join
walk_in_customers on customer.customer_id = walk_in_customers.customer_id
union

select customer.customer_id,customer_name, "Mail order" as Type from customer join mail_order_customers on customer.customer_id = mail_order_customers.customer_id ;

customer_id	customer_name	Type
1	Aniket	Walk in
3	Neha	Walk in
5	Shridhar	Walk in
7	Ajinkya	Walk in
9	Saurabh	Walk in
2	Harshraj	Mail order
4	Shyam	Mail order
6	Sagar	Mail order
8	Shreyash	Mail order
10	Rahul	Mail order

CONCLUSION:

We have successfully designed and implemented a data warehouse for a customer order processing system in a company with mysql.