Dharmsinh Desai University, Nadiad
Department of Information Technology
DAIE, IT704
B.Tech. IT, Sem: VII

Experiment – 04-05 (Star Schema Diagram)
Submitted By
Roll No.: IT076
Name: DISHANT MODH

Aim: 1. Draw Star Schema diagram for Sales All Electronics Store.
2. Make Schema Table in Oracle Live.
3. Perform DMQL Query like Rollup, Drill down ..etc on it.

Tools/Apparatus: Oracle Live Online

4.1) Write down SQL query for Dimension tables Item, Branch, Location and Time with proper syntax and primary key.

1) Product Table

```
create table branch (
branch_id number not null,
branch_name varchar2(30) not null,
branch_manager varchar2(30) not null,
branch_city varchar2(30) not null,
primary key(branch_id) );
```

2)Branch Table

```
create table branch (
branch_id number not null,
branch_name varchar2(30) not null,
branch_manager varchar2(30) not null,
branch_city varchar2(30) not null,
primary key(branch_id) );
```

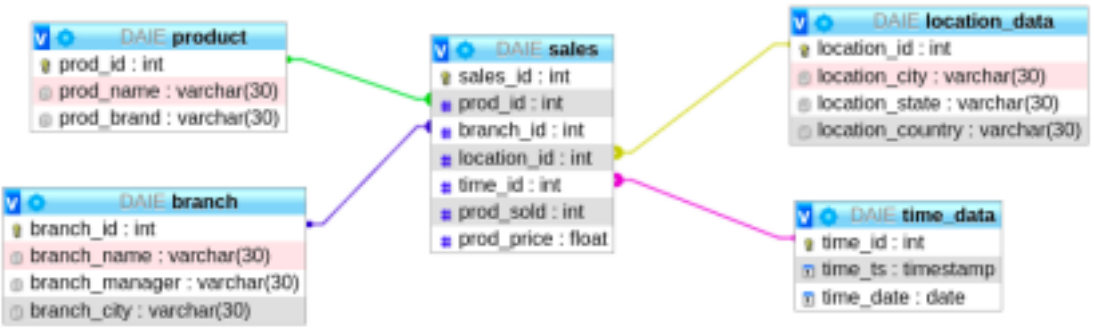
3) Location Table

```
create table location_data (
location_id number not null,
location_city varchar2(30) not null,
location_state varchar2(30) not null,
location_country varchar2(30) not null,
primary key(location_id)
);
```

4)Time Table

```
create table time_data (
time_id number not null,
time_ts timestamp with time zone not null,
time_date date not null,
primary key(time_id)
);
```

Star Schema



4.2) Write down SQL query for Sales fact table using primary keys of all dimension tables as a foreign key.

```
create table sales (
sales_id number not null,
prod_id number not null,
branch_id number not null,
location_id number not null,
time_id number not null,
prod_sold number not null,
prod_price float not null,
foreign key(prod_id) references product(prod_id),
foreign key(branch_id) references branch(branch_id), foreign
key(location_id) references location_data(location_id), foreign
key(time_id) references time_data(time_id), primary
key(sales_id)
);
```

4.3) Sample SQL query to Insert data in dimension tables.

1) Product Data

```
insert into product values(1,'iPhone 13','Apple');
insert into product values(2,'9R','OnePlus');
insert into product values(3,'GT','Realme');
insert into product values(4,'Z Fold 3','Samsung');
insert into product values (5,'11 Ultra', 'MI');
```

2)Branch Data

```
insert into branch values(1,'iVenus','Deep Shah', 'Surat'); insert into
branch values(2,'Pujara','Meet Patel',' Nadiad'); insert into branch
values(3,'Phone Wale','Jay Desai', 'Ahmedabad'); insert into branch
values(4,'Croma','Manan Shah', 'Vadodara');
```

3)Location Data

```
insert into location_data values(1,'Surat','Gujarat','India'); insert
into location_data values(2,'Ahmedabad','Gujarat','India'); insert
into location_data values(3,'Vadodara','Gujarat','India'); insert into
location_data values(4,'Nadiad','Gujarat','India'); insert into
location_data values(5,'Mumbai','Maharashtra','India'); insert into
location_data values(6,'Jaipur','Rajasthan','India');
```

4) Time Data

```
insert into time_data values(1,timestamp '2018-12-23 11:00:00',date
'2018-12-23');
insert into time_data values(2,timestamp '2020-01-05 17:00:00',date
'2020-01-05');
insert into time_data values(3,timestamp '2020-04-04 15:00:00',date
'2020-04-04');
insert into time_data values(4,timestamp '2019-07-13 14:00:00',date
'2019-07-13');
insert into time_data values(5,timestamp '2018-12-03 20:00:00',date
'2018-12-03');
```

5) Sales Data

```
insert into sales values(1,1,1,1,1,30,90000.00);
insert into sales values(2,2,2,2,3,25,45000.00);
insert into sales values(3,3,3,4,5,35,37000.00);
insert into sales values(4,4,2,3,6,12,120000.00);
insert into sales values(5,5,3,5,4,10,69000.00);
insert into sales values(6,3,4,6,2,20,37000.00);
insert into sales values(7,2,3,4,5,40,45000.00);
insert into sales values(8,2,2,2,3,20,90000.00);
insert into sales values(9,1,2,2,3,20,90000.00);
```

4.4) **View the measures for example. Total units sold and total**

dollars sold using Select SQL query like mentioned in the lab manual.

```
select s.time_id, t.time_date, s.prod_id, c.prod_name, s.branch_id,
b.branch_name,s.location_id, l.location_city, sum (s.prod_sold*s.prod_price)
as
total_selling, sum (s.prod_sold) as total_prod_sold
from time_data t, product c, branch b, location_data l, sales s where
s.time_id = t.time_id and s.prod_id = c.prod_id and s.branch_id =
b.branch_id and s.location_id = l.location_id
group by s.time_id, t.time_date, s.prod_id, c.prod_name, s.branch_id,
b.branch_name, s.location_id, l.location_city;
```

TIME_ID	TIME_DATE	PROD_ID	PROD_NAME	BRANCH_ID	BRANCH_NAME	LOCATION_ID	LOCATION_CITY	TOTAL_SELLING	TOTAL_PROD_SOLD
3	04-APR-20	1	iPhone 13	2	Pujara	2	Ahmedabad	1000000	20
2	05-JAN-20	3	GT	4	Croma	6	Jaipur	740000	20
6	12-MAR-19	4	Z Fold 3	2	Pujara	3	Vadodara	1440000	12
3	04-APR-20	2	9R	2	Pujara	2	Ahmedabad	2925000	45
1	23-DEC-18	1	iPhone 13	1	iVenus	1	Surat	2700000	30
4	13-JUL-19	5	11 Ultra	3	Phone Wale	5	Mumbai	690000	10
5	03-DEC-18	2	9R	3	Phone Wale	4	Nadiad	1800000	40
5	03-DEC-18	3	GT	3	Phone Wale	4	Nadiad	1295000	35

Download CSV
8 rows selected.

4.5) Write down the queries to perform slice. In which one should keep one of the dimensions as constant and other dimensions should range from min to max.

```
SELECT s.prod_id, MIN(s.prod_sold) as minimum_prod_sold, MAX(s.prod_sold)
as
maximum_prod_sold
FROM sales s
GROUP BY s.prod_id;
```

PROD_ID	MINIMUM_PROD_SOLD	MAXIMUM_PROD_SOLD
1	20	30
2	20	40
4	12	12
5	10	10
3	20	35

Download CSV
5 rows selected.

```
SELECT s.prod_id, MIN(s.prod_sold*s.prod_price) as
minimum_total_sold, MAX(s.prod_sold*s.prod_price) as
maximum_total_sold FROM sales s
GROUP BY s.prod_id;
```

PROD_ID	MINIMUM_TOTAL_SOLD	MAXIMUM_TOTAL_SOLD
1	1800000	2700000
2	1125000	1800000
4	1440000	1440000
5	690000	690000
3	740000	1295000

Download CSV
5 rows selected.

4.6) Write down the queries to perform the dice. In which one has to keep two of the dimension's constant.

```
SELECT s.prod_id, b.branch_name, SUM(s.prod_sold*s.prod_price)
as total_dollars_sold_at_NYC_store
```

```
FROM sales s, branch b
WHERE s.branch_id = b.branch_id and b.branch_name = 'Pujara'
GROUP BY s.prod_id, b.branch_name;
```

PROD_ID	BRANCH_NAME	TOTAL_DOLLARS_SOLD_AT_NYC_STORE
1	Pujara	1800000
2	Pujara	2925000
4	Pujara	1440000

Download CSV
3 rows selected.

4.7) Write down the queries to perform roll-up by keeping one dimension constant and others should range from min to max. It is more like a generalization.

```
SELECT prod_id, MIN(prod_sold) AS minimum_prods_sold,
MIN(prod_sold*prod_price)
AS minimum_sold, MAX(prod_sold) AS
maximum_prods_sold, MAX(prod_sold*prod_price) AS
maximum_sold
FROM sales
GROUP BY ROLLUP(prod_id)
ORDER BY prod_id
```

PROD_ID	MINIMUM_PRODS_SOLD	MINIMUM_SOLD	MAXIMUM_PRODS_SOLD	MAXIMUM_SOLD
1	20	1800000	30	2700000
2	20	1125000	40	1800000
3	20	740000	35	1295000
4	12	1440000	12	1440000
5	10	690000	10	690000
-	10	690000	40	2700000

Download CSV
6 rows selected.

```
SELECT c.prod_name, SUM(s.prod_sold) as
total_prods_sold FROM sales s, product c
```

```
WHERE s.prod_id = c.prod_id
GROUP BY ROLLUP(c.prod_name)
```

PROD_NAME	TOTAL_PRODS_SOLD
11 Ultra	10
9R	85
GT	55
Z Fold 3	12
iPhone 13	50
-	212

Download CSV

6 rows selected.

4.8) Write down the queries to perform roll-up by keeping one dimension constant and others should range from min to max. It is more like a specialization Screen

```
SELECT c.prod_name, SUM(s.prod_sold) as
total_prods_sold FROM sales s, product c
WHERE s.prod_id = c.prod_id
GROUP BY c.prod_name
```