**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,

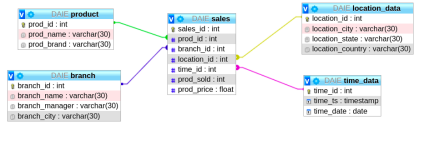
time\_ts timestamp with time zone not null,

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);

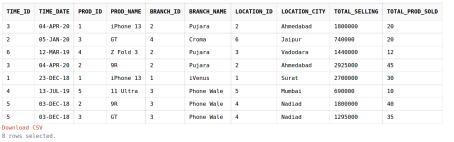
**4.4**) **View the measures for example. Total units sold and total dollars sold using Select SQL query like mentioned in the lab manual.**

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;



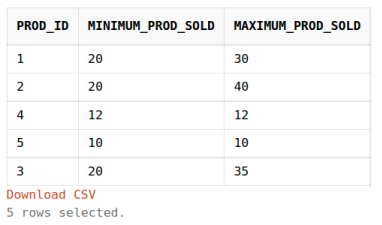
**4.5) Write down the queries to perform slice. In which one should keep one of the dimensions as constant and other dimensions should range from min to max.**

SELECT s.prod\_id, MIN(s.prod\_sold) as minimum\_prod\_sold, MAX(s.prod\_sold) as

maximum\_prod\_sold

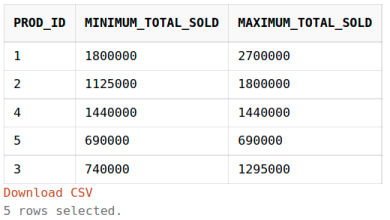
FROM sales s

GROUP BY s.prod\_id;



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;

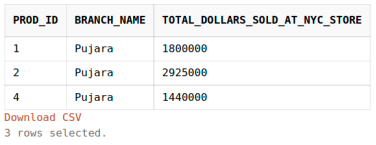


**4.6) Write down the queries to perform the dice. In which one has to keep two of the dimension’s constant**.

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;



**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.**

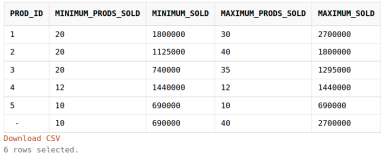
SELECT prod\_id, MIN(prod\_sold) AS minimum\_prods\_sold, MIN(prod\_sold\*prod\_price)

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)

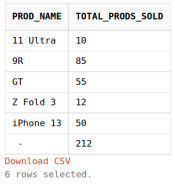
ORDER BY prod\_id



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)



**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