

## Practical No.2

### Implementation of Analytical queries

**1. Create tables Employee and Department as per the given schema and insert data into them.**

dept(deptno number(2,0),dname varchar2(14),loc varchar2(13),constraint pk\_dept primary key (deptno));

emp(empno number(4,0),ename varchar2(10), job varchar2(9), mgr number(4,0),hiredate date,sal number(7,2),comm number(7,2),deptno number(2,0), constraint pk\_emp primary key (empno),constraint fk\_deptno foreign key (deptno) references dept (deptno));

**Execute the following queries.**

**Create a table employee with attribute empid, name, deptid,deptname, salary and joining date.**

```
1 v create table dept(  
2   deptno number(2,0),  
3   dname  varchar2(14),  
4   loc    varchar2(13),  
5   constraint pk_dept primary key (deptno)  
6 );
```

Table created.

```
8 v create table emp(  
9     empno    number(4,0),  
10    ename    varchar2(10),  
11    job      varchar2(9),  
12    mgr      number(4,0), |  
13    hiredate date,  
14    sal      number(7,2),  
15    comm     number(7,2),  
16    deptno   number(2,0),  
17    constraint pk_emp primary key (empno),  
18    constraint fk_deptno foreign key (deptno) references dept (deptno)  
19 );  
20
```

Table created.

```
21 v insert into dept
22 values(10, 'ACCOUNTING', 'NEW YORK');
23
24 v insert into dept
25 values(20, 'RESEARCH', 'DALLAS');|
26
27 v insert into dept
28 values(30, 'SALES', 'CHICAGO');
29
30 v insert into dept
31 values(40, 'OPERATIONS', 'BOSTON');
32
33
34
```

1 row(s) inserted.

1 row(s) inserted.

```
39
40 v insert into emp
41 values(
42     7698, 'BLAKE', 'MANAGER', 7839,
43     to_date('1-5-1981','dd-mm-yyyy'),
44     2850, null, 30
45 );
46
47 v insert into emp
48 values(
49     7782, 'CLARK', 'MANAGER', 7839,
50     to_date('9-6-1981','dd-mm-yyyy'),
51     2450, null, 10
52 );
53
54 v insert into emp
55 values(
56     7566, 'JONES', 'MANAGER', 7839,
57     to_date('2-4-1981','dd-mm-yyyy'),
58     2975, null, 20
59 );
60
61 v insert into emp
62 values(
```

1 row(s) inserted.

Write the queries -

1. To return the first salary reported in each department.

131	✓	SELECT deptno, MIN(sal) AS first_salary
132		FROM emp
133		GROUP BY deptno;
134		

  

DEPTNO	FIRST_SALARY
30	950
10	1300
20	800

2. To show us how the average salary has changed over the years

135	✓	SELECT EXTRACT(YEAR FROM hiredate) AS year, AVG(sal) AS avg_salary
136		FROM emp
137		GROUP BY EXTRACT(YEAR FROM hiredate)
138		ORDER BY year;

  

YEAR	AVG_SALARY
87	2050
1980	800
1981	2282.5
1982	1300

### 3.To display the salary of each employee, along with the lowest and highest within their department

```

140 ✓ SELECT e.ename, e.sal, d.deptno,
141         (SELECT MIN(sal) FROM emp WHERE deptno = e.deptno) AS min_salary,
142         (SELECT MAX(sal) FROM emp WHERE deptno = e.deptno) AS max_salary
143 FROM emp e
144 JOIN dept d ON e.deptno = d.deptno;
145

```

ENAME	SAL	DEPTNO	MIN_SALARY	MAX_SALARY
BLAKE	2850	30	950	2850
ALLEN	1600	30	950	2850
WARD	1250	30	950	2850
MARTIN	1250	30	950	2850
TURNER	1500	30	950	2850
JAMES	950	30	950	2850
KING	5000	10	1300	5000

### 4. To divide the whole result set into five buckets based on salary

```

146 v SELECT empno,ename,deptno,sal,
147       NTILE(5) OVER (PARTITION BY deptno ORDER BY sal) AS bucket_no
148 FROM emp;
149
150

```

EMPNO	ENAME	DEPTNO	SAL	BUCKET_NO
7934	MILLER	10	1300	1
7782	CLARK	10	2450	2
7839	KING	10	5000	3
7369	SMITH	20	800	1

**5. To display for each employee in Department 30 in the employees table, the hire date of the**

**employee hired just after**

```

156 v SELECT ename,hiredate,
157       LEAD(hiredate,1)OVER(ORDER BY hiredate) AS "NEXT HIRED"
158 FROM emp
159 WHERE deptno = 30
160 ORDER BY hiredate;
161

```

ALLEN	20-FEB-81	22-FEB-81
WARD	22-FEB-81	01-MAY-81
BLAKE	01-MAY-81	08-SEP-81
TURNER	08-SEP-81	28-SEP-81
MARTIN	28-SEP-81	03-DEC-81
JAMES	03-DEC-81	-

**2. Create the table Sales and insert records as given. Write analytical queries –**

```

1  CREATE TABLE Sales (
2      year NUMBER(4),
3      country VARCHAR2(20),
4      product VARCHAR2(20),
5      profit NUMBER(10)
6  );

```

Table created.

```

8  INSERT INTO Sales VALUES (2000, 'Finland', 'Computer', 1500);
9  INSERT INTO Sales VALUES (2000, 'Finland', 'Phone', 100);
10 INSERT INTO Sales VALUES (2001, 'Finland', 'Phone', 10);
11 INSERT INTO Sales VALUES (2000, 'India', 'Calculator', 75);
12 INSERT INTO Sales VALUES (2000, 'India', 'Calculator', 75);
13 INSERT INTO Sales VALUES (2000, 'India', 'Computer', 1200);
14 INSERT INTO Sales VALUES (2000, 'USA', 'Calculator', 75);
15 INSERT INTO Sales VALUES (2000, 'USA', 'Computer', 1500);
16 INSERT INTO Sales VALUES (2001, 'USA', 'Calculator', 50);
17 INSERT INTO Sales VALUES (2001, 'USA', 'Computer', 1500);
18 INSERT INTO Sales VALUES (2001, 'USA', 'Computer', 1200);
19 INSERT INTO Sales VALUES (2001, 'USA', 'TV', 150);
20 INSERT INTO Sales VALUES (2001, 'USA', 'TV', 100);
21

```

1 row(s) inserted.

1 row(s) inserted.

### 1. To find total profit for each country.



```

22 v SELECT country, SUM(profit) AS total_profit
23 FROM Sales
24 GROUP BY country;
25

```

COUNTRY	TOTAL_PROFIT
Finland	1610
USA	4575
India	1350

## 2. Display Country with maximum profit.

```

26 v SELECT country, SUM(profit) AS total_profit
27 FROM Sales
28 GROUP BY country
29 ORDER BY total_profit DESC;
30 FETCH FIRST 1 ROWS ONLY;
31

```

COUNTRY	TOTAL_PROFIT
USA	4575

## 3. Display products with maximum and minimum profit in each country.

```

32 v SELECT country, product,
33      MAX(profit) AS max_profit,
34      MIN(profit) AS min_profit
35 FROM Sales
36 GROUP BY country, product;
37

```

COUNTRY	PRODUCT	MAX_PROFIT	MIN_PROFIT
Finland	Phone	100	10
India	Computer	1200	1200
India	Calculator	75	75
USA	TV	150	100

**4. Display average sale of each product in each country.**

```

38 v SELECT country, product, AVG(profit) AS avg_profit
39 FROM Sales
40 GROUP BY country, product;

```

COUNTRY	PRODUCT	AVG_PROFIT
Finland	Phone	55
India	Computer	1200
India	Calculator	75
USA	TV	125
USA	Computer	1400
Finland	Computer	1500
USA	Calculator	62.5

##### 5. Find total profit product wise.

```

42 v SELECT product, SUM(profit) AS total_profit
43 FROM Sales
44 GROUP BY product;
45

```

PRODUCT	TOTAL_PROFIT
Phone	110
TV	250
Computer	6900
Calculator	275

**3. The research is about to create the star schema for the sales system. The research consists of**

**all the information related to the sale's record like items, location and the time etc.**

**Create a**

**schema (database) with fact and dimension tables. Perform the OLAP operations on your**

**schema.**

- Suppose we want to record in a warehouse information about every Item sale, e.g.:

- Product number,

- location from where the product was sold,

- date of the sale, and

- Units sold.

- The fact table is thus:

Sales(item\_key, loc\_key, time\_key, units)

- The dimension tables include information about the Items, Location, and time

“dimensions”:

Loc(loc\_key, city,state,country)

items(item\_key,item\_name, item\_category, color,price)

Time(time\_key,sdate,week,month,quarter,syear),

**Step 1: Create the Fact and Dimension Tables****Fact Table: Sales**

```
1 v CREATE TABLE Sales (  
2     item_key NUMBER,  
3     loc_key NUMBER,  
4     time_key NUMBER,  
5     units NUMBER  
6 );|
```

Table created.

**Dimension Table: Loc (Location)**

```
8 v CREATE TABLE Loc (  
9     loc_key NUMBER PRIMARY KEY,  
10    city VARCHAR2(50),  
11    state VARCHAR2(50),  
12    country VARCHAR2(50)  
13 );|
```

Table created.

**Dimension Table: Items**

```

15 CREATE TABLE Items (
16     item_key NUMBER PRIMARY KEY,
17     item_name VARCHAR2(100),
18     item_category VARCHAR2(50),
19     color VARCHAR2(20),
20     price NUMBER
21 );

```

Table created.

**Dimension Table: Time**

```

23 CREATE TABLE Time (
24     time_key NUMBER PRIMARY KEY,
25     sdate DATE,
26     week NUMBER,
27     month NUMBER,
28     quarter NUMBER,
29     syyear NUMBER
30 );

```

Table created.

**Insert Data into Loc (Location) Table**

```

32 INSERT INTO Loc VALUES (1, 'Mumbai', 'Maharashtra', 'India');
33 INSERT INTO Loc VALUES (2, 'Chennai', 'Tamil Nadu', 'India');
34 INSERT INTO Loc VALUES (3, 'New York', 'New York', 'USA');
35 INSERT INTO Loc VALUES (4, 'San Francisco', 'California', 'USA');

```

1 row(s) inserted.

**Insert Data into Items Table**

```

37 INSERT INTO Items VALUES (101, 'Pen', 'Stationery', 'Blue', 10);
38 INSERT INTO Items VALUES (102, 'Jeans', 'Clothing', 'Blue', 1500);
39 INSERT INTO Items VALUES (103, 'Laptop', 'Electronics', 'Silver', 50000);
40 INSERT INTO Items VALUES (104, 'Phone', 'Electronics', 'Black', 20000);

```

1 row(s) inserted.

**Insert Data into Time Table**

```

42 INSERT INTO Time VALUES (201, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 3, 1, 1, 2024);
43 INSERT INTO Time VALUES (202, TO_DATE('2024-04-12', 'YYYY-MM-DD'), 15, 4, 2, 2024);
44 INSERT INTO Time VALUES (203, TO_DATE('2024-07-22', 'YYYY-MM-DD'), 30, 7, 3, 2024);
45 INSERT INTO Time VALUES (204, TO_DATE('2024-10-05', 'YYYY-MM-DD'), 40, 10, 4, 2024);

```

1 row(s) inserted.

**Insert Data into Sales (Fact Table)**

```

47 INSERT INTO Sales VALUES (101, 1, 201, 500);
48 INSERT INTO Sales VALUES (102, 1, 201, 150);
49 INSERT INTO Sales VALUES (102, 2, 202, 100);
50 INSERT INTO Sales VALUES (101, 2, 202, 300);
51 INSERT INTO Sales VALUES (103, 3, 203, 80);
52 INSERT INTO Sales VALUES (104, 3, 203, 120);
53 INSERT INTO Sales VALUES (103, 4, 204, 50);
54 INSERT INTO Sales VALUES (104, 4, 204, 200);

```

1 row(s) inserted.

## Step 2: OLAP Queries

### 1. Display data for quarter 1

```
56 v SELECT * FROM Sales s
57 JOIN Time t ON s.time_key = t.time_key
58 WHERE t.quarter = 1;
59
```

ITEM_KEY	LOC_KEY	TIME_KEY	UNITS	TIME_KEY	SDATE	WEEK	MONTH	QUARTER	SYEAR
101	1	201	500	201	15-JAN-24	3	1	1	2024
102	1	201	150	201	15-JAN-24	3	1	1	2024

### 2. Display total sales of pen or jeans from “mumbai” or “chennai” for quarter 1 or 2.

```
61 v SELECT SUM(s.units) AS total_sales
62 FROM Sales s
63 JOIN Loc l ON s.loc_key = l.loc_key
64 JOIN Items i ON s.item_key = i.item_key
65 JOIN Time t ON s.time_key = t.time_key
66 WHERE (i.item_name = 'Pen' OR i.item_name = 'Jeans')
67        AND (l.city = 'Mumbai' OR l.city = 'Chennai')
68        AND (t.quarter = 1 OR t.quarter = 2);
69
```

TOTAL\_SALES

1050



**3. Find the total units sales in each state.**

```

71 ✓ SELECT l.state, SUM(s.units) AS total_units
72 FROM Sales s
73 JOIN Loc l ON s.loc_key = l.loc_key
74 GROUP BY l.state;
75

```

STATE	TOTAL_UNITS
Maharashtra	650
California	250
New York	200
Tamil Nadu	400

**4. Find the total units sales in each city**

```
76 v SELECT l.city, SUM(s.units) AS total_units
77 FROM Sales s
78 JOIN Loc l ON s.loc_key = l.loc_key
79 GROUP BY l.city;
80
```

CITY	TOTAL_UNITS
Chennai	400
Mumbai	650
New York	200
San Francisco	250