

Exercise 14 : Data Warehousing and Decision Support

Figure 1: Sales

pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26
12	2	2	45
12	3	2	20
13	1	2	20
13	2	2	40
13	3	2	5

Figure 2: Locations

locid	city	state	country
1	Madison	WI	USA
2	Fresno	CA	USA
5	Chennai	TN	India

Figure 3: Products

pid	pname	category	price
11	Lee Jeans	Apparel	25
12	Zord	Toys	18
13	Biro Pen	Stationery	2

Figure 4: Times

timeid	date	month	quarter	year
1	1995-May-23	May	1995-May	1995
2	1996-Feb-02	February	1996-Jan	1996
3	1997-Jan-30	January	1997-Jan	1997

1. Consider the relation **Sales**.

a. Show the result of pivoting the relation on **Products** (**pname**) and **Times**(**year**).

Solution:

	1995	1996	1997	Total
Lee Jeans	60	30	25	115
Zord	56	65	70	191
Biro Pen	28	50	15	93
Total	144	145	110	399

b. Write a collection of SQL queries to obtain the same result as in the previous part.

Solution: The main body of the table is generated by:

```

SELECT P.pname, T.year, SUM(S.sales)
FROM Sales S, Products P, Times T
WHERE S.pid = P.pid AND S.timeid = T.timeid
GROUP BY P.pname, T.year

```

The summary column on the right is generated by:

```

SELECT P.pname, SUM(S.sales)
FROM Sales S, Products P
WHERE S.pid = P.pid
GROUP BY P.pname

```

The summary row at the bottom is generated by:

```

SELECT T.year, SUM(S.sales)
FROM Sales S, Times T
WHERE S.timeid = T.timeid
GROUP BY T.year

```

The cumulative sum in the bottom right corner is produced by the query:

```

SELECT SUM(sales)
FROM Sales

```

- c. Show the result of pivoting the relation on `Products(pname)` and `Locations(state)`.

Solution:

	WI	CA	Total
Lee Jeans	48	67	115
Zord	100	91	191
Biro Pen	28	65	93
Total	176	223	399

- d. Write the equivalent SQL query using the CUBE operator

Solution:

```

SELECT P.pname, L.state, SUM(S.sales)
FROM Sales S, Products P, Locations L
WHERE S.locid = L.locid AND S.pid = P.pid
GROUP BY CUBE(P.pname, L.state)

```

- e. Write the output of executing the SQL query above on the instances of the relations shown above. **Solution:**

P.pname	L.state	SUM(S.sales)
Lee Jeans	WI	48
Lee Jeans	CA	67
Lee Jeans	null	115
Zord	WI	100
Zord	CA	92
Zord	null	191
Biro Pen	WI	28
Biro Pen	CA	65
Biro Pen	null	93
null	WI	176
null	CA	223
null	null	399

2. Consider the cross-tabulation of the `Sales` relation as shown below

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	399

- a. Show the result of roll-up on `locid` (i.e., state).

Solution:

	Total
1995	144
1996	145
1997	110
Total	399

- b. Write a collection of SQL queries to obtain the same result as in the previous part.

Solution:

The summary column on the right is generated by:

```
SELECT T.year, SUM(sales)
FROM Sales S, Times T
WHERE S.timeid = T.timeid
GROUP BY T.year
```

The cumulative sum in the bottom right corner is produced by the query:

```
SELECT SUM(sales)
FROM Sales
```

- c. Show the result of roll-up on `locid` followed by drill-down on `pname`.

Solution:

	1995	1996	1997	Total
Lee Jeans	60	30	25	115
Zord	56	65	70	191
Biro Pen	28	50	15	93
Total	144	145	110	399

- d. Find the slice of the table where `pid=12`

Solution:

	WI	CA	Total
1995	30	26	56
1996	20	45	65
1997	50	20	70
Total	100	91	191

3. Given a data cube with fact table `Sales` and suitable dimension tables relating the ids to product categories, cities, months, quarters, and years. Suppose a data cube system currently displays the data corresponding to SQL query

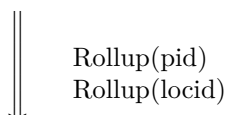
```
SELECT pid, locid, SUM(sales) from Sales S GROUP BY pid, locid
```

Starting from this, which data cube commands (of the form `rollup`, `drilldown`, `slice`, `dice`, etc., with suitable parameters) does the user need to issue to get a display corresponding to the query below. Write down the intermediate-stages' query representation.

```
SELECT P.category, T.Quarter, SUM(S.Sales)
FROM Sales S, Products P, Times T
WHERE S.pid = P.pid AND S.timeid = T.timeid
AND T.quarter >= "2013-Jan" AND T.Quarter <= "2013-Jul"
GROUP BY P.category, T.quarter
```

Solution:

```
SELECT pid, locid, SUM(sales) FROM Sales S GROUP BY pid, locid
```



```
SELECT SUM(sales) FROM Sales S group
```

\Downarrow
 Drilldown(P.category)
 Drilldown(T.quarter)

```
SELECT P.category, T.quarter, SUM(sales)
FROM Sales S , Products P, Times T
WHERE S.pid = P.pid AND S.timeid = T.timeid
GROUP BY P.category, T.quarter
```

\Downarrow
 Dice(T.quarter >= "2013-Jan" AND T.quarter <= "2013-Jul")

```
SELECT P.category, T.quarter, SUM(sales)
FROM Sales S , Products P, Times T
WHERE S.pid = P.pid AND S.timeid = T.timeid
AND T.quarter >= "2013-Jan" AND T.quarter <= "2013-Jul"
GROUP BY P.category, T.quarter
```

4. Consider the **Customers** relation as shown below

custid	name	gender	rating
356	Jack	M	5
784	Sue	F	4
334	Olivia	F	6
239	Daniel	M	2
562	Sam	M	3
834	Diana	F	5

Construct bitmap indexes for the columns **ratings** and **gender** assuming that ratings range from 1 to 6.

Solution:

1	2	3	4	5	6	M	F
0	0	0	0	1	0	1	0
0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	1