Draft solution

Materialized views and Triggers - Practice 3

Creation and updating of the materialized view with the use CREATE MATERIALIZED VIEW in ORACLE

CREATE MATERIALIZED VIEW

1.

```
CREATE MATERIALIZED VIEW GroupByMonthYear
BUILD IMMEDIATE
AS
SELECT DateMonth, DateYear, SUM(NumberOfCalls) as NumCalls, SUM(Price) as TotPrice
FROM FACTS F, TIMEDIM T
WHERE F.ID_time = T.ID_time
GROUP BY DateMonth, DateYear;
```

2.

```
CREATE MATERIALIZED VIEW GroupByMonthPhoneRateType
BUILD IMMEDIATE
AS
SELECT DateMonth, f. id_phonerate, p.phoneratetype, SUM(NumberOfCalls) as NumCalls, SUM(Price)
as TotPrice
FROM PHONERATE P, TIMEDIM T, FACTS F
WHERE F.ID_time = T.ID_time AND f.id_phonerate = p.id_phonerate
GROUP BY  DateMonth, f.id_phonerate, p.phoneratetype
```

Update and management of materialized view with CREATE MATERIALIZED VIEW LOG – **for Students who practice with their own PC on Oracle SQL**

CREATE MATERIALIZED VIEW LOG

```
CREATE MATERIALIZED VIEW LOG ON FACTS
WITH SEQUENCE, ROWID (ID_time, id_phonerate, Id_location_receiver, Id_Location_Caller,
NumberOfCalls, Price)
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON TIMEDIM
WITH SEQUENCE, ROWID (ID_time, DateMonth, DateYear)
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON phonerate
WITH SEQUENCE, ROWID (id_phonerate, phoneratetype)
INCLUDING NEW VALUES;
```

Update of MATERIALIZE VIEW.

STEP 2.3 Try to modify the FACTS table as follows:

INSERT INTO FACTS(id_time, id_phonerate, ID_location_Caller, ID_location_receiver, price,numberofcalls) values(8,1,558,752,40000,150)

INSERT INTO FACTS(Id_time, ID_phoneRate, ID_location_Caller, ID_location_Receiver, Price,NumberOfCalls) values(1,6,558,752,100,100)

GroupByMonthYear

Before

DATEMONTH	DATEYEAR	NUMCALLS	TOTPRICE
10-2003	2003	5148	17827740
8-2003	2003	5544	19144722
7-2003	2003	5628	22829174
8-2004	2004	5496	20692410
10-2004	2004	5004	13996476
7-2004	2004	5544	15168319

After

DATEMONTH	DATEYEAR	NUMCALLS	TOTPRICE
10-2003	2003	5148	17827740
8-2003	2003	5694	19184722
7-2003	2003	5728	22829274
8-2004	2004	5496	20692410
10-2004	2004	5004	13996476
7-2004	2004	5544	15168319

GroupByMonthPhoneRateType

Before

DATEMONTH	ID_PHONERATE	PHONERATETYPE	NUMCALLS	TOTPRICE
10-2003	3	Mattino	1296	14813280
10-2003	4	24 ore su 24	540	540540
7-2003	2	Giorno	1104	1232150
8-2003	3	Mattino	1536	15961530
8-2003	5	CartaNatale	1608	1121320
8-2003	2	Giorno	888	999800
10-2003	5	CartaNatale	1800	1260000
10-2003	1	Notte	648	220320
10-2003	2	Giorno	864	993600
7-2003	5	CartaNatale	1416	982640
7-2003	1	Notte	576	189420
8-2003	1	Notte	648	213900
8-2003	4	24 ore su 24	864	848172
7-2003	4	24 ore su 24	588	583024
7-2003	3	Mattino	1944	19841940

After

DATEMONTH	ID_PHONERATE	PHONERATETYPE	NUMCALLS	TOTPRICE
10-2003	3	Mattino	1296	14813280
10-2003	4	24 ore su 24	540	540540
7-2003	2	Giorno	1104	1232150
8-2003	3	Mattino	1536	15961530
8-2003	5	CartaNatale	1608	1121320
8-2003	2	Giorno	888	999800
10-2003	1	Notte	648	220320
10-2003	5	CartaNatale	1800	1260000
7-2003	6	Festivi	100	100
7-2003	5	CartaNatale	1416	982640
10-2003	2	Giorno	864	993600
7-2003	1	Notte	576	189420
8-2003	1	Notte	798	253900
7-2003	3	Mattino	1944	19841940
7-2003	4	24 ore su 24	588	583024
8-2003	4	24 ore su 24	864	848172

Update and management of views via Trigger

1. VM1 Table

```
CREATE TABLE VM1 (
DateMonth VARCHAR(15) CHECK (DateMonth IS NOT NULL),
DateYear INTEGER CHECK (DateYear IS NOT NULL),
TOT_NumberOfCalls INTEGER,
TOT_Price INTEGER);

INSERT INTO VM1 (DateMonth, DateYear, TOT_NumberOfCalls, TOT_Price)
(SELECT DateMonth, DateYear, SUM(NumberOfCalls), SUM(Price)
FROM FACTS F, TIMEDIM T
WHERE F.ID_time = T.ID_time
GROUP BY DateMonth, DateYear);
```

2. VM2 Table

```
CREATE TABLE VM2 (
DateMonth VARCHAR(15) CHECK (DateMonth IS NOT NULL),
phoneRate VARCHAR(20) CHECK (phoneRate IS NOT NULL),
TOT_NumberOfCalls INTEGER,
TOT_Price INTEGER);

INSERT INTO VM2 (DateMonth, phoneRate, TOT_NumberOfCalls, TOT_Price)
(SELECT DateMonth, phoneRateType, sum(NumberOfCalls), sum(Price)
FROM phoneRate t, TIMEDIM te, FACTS f
WHERE f.ID_time = te.ID_time and f.id_phoneRate = t.id_phoneRate and te.DateYear=2003
GROUP BY DateMonth, phoneRateType);
```

VM1 - Output

DATEMONTH	DATEYEAR	TOT_NUMBEROFCALLS	TOT_PRICE
10-2003	2003	5148	17827740
7-2004	2004	5544	15168319
8-2003	2003	5544	19144722
7-2003	2003	5628	22829174
8-2004	2004	5496	20692410
10-2004	2004	5004	13996476

VM2 - Output

DATEMONTH	PHONERATE	TOT_NUMBEROFCALLS	TOT_PRICE
7-2003	24 ore su 24	588	583024
10-2003	Giorno	864	993600
7-2003	Notte	576	189420
7-2003	Giorno	1104	1232150
8-2003	CartaNatale	1608	1121320
10-2003	Notte	648	220320
8-2003	Giorno	888	999800
7-2003	Mattino	1944	19841940
7-2003	CartaNatale	1416	982640
8-2003	Notte	648	213900
10-2003	Mattino	1296	14813280
10-2003	24 ore su 24	540	540540
10-2003	CartaNatale	1800	1260000
8-2003	24 ore su 24	864	848172
8-2003	Mattino	1536	15961530

VM1 and VM2 outputs are obtained by considering table FACTS without any INSERT of the previous points. To reproduce these values, drop FACTS and run again the FACTS script.

Triggers

1. TRIGGER 1

```
CREATE TRIGGER Trigger1
AFTER INSERT ON FACTS
FOR EACH ROW DECLARE
N NUMBER;
VAR_DateMonth VARCHAR(15);
VAR_DateYear NUMBER;
BEGIN

SELECT DateMonth INTO VAR_DateMonth
FROM TIMEDIM
WHERE ID_time = :NEW.ID_time;


SELECT DateYear INTO VAR_DateYear
FROM TIMEDIM
WHERE ID_time = :NEW.ID_time;


SELECT Count(*) INTO N
FROM VM1
WHERE DateMonth=Var_DateMonth AND DateYear = Var_DateYear;


if (N > 0) then
        update VM1
        set TOT_NumberOfCalls = TOT_NumberOfCalls + :NEW.NumberOfCalls, TOT_Price =
        TOT_Price + :NEW.Price
        where DateMonth= Var_DateMonth AND DateYear = Var_DateYear;
else
        insert into VM1 (DateMonth, DateYear, Tot_NumberOfCalls, Tot_Price)
        values (Var_DateMonth, Var_DateYear, :NEW.NumberOfCalls, :NEW.Price);
end if;
END;
```

2. TRIGGER 2

```
CREATE TRIGGER Trigger2
AFTER INSERT ON FACTS
FOR EACH ROW
DECLARE
N NUMBER;
VAR_DateMonth VARCHAR(15);
VAR_phoneRate VARCHAR(20);
BEGIN


SELECT DateMonth INTO VAR_DateMonth
FROM TIMEDIM
WHERE ID_time = :NEW.ID_time;



SELECT phoneRateType INTO VAR_phoneRate
FROM PHONERATE
WHERE Id_phonerate = :NEW.Id_phonerate;



SELECT Count(*) INTO N
FROM VM2
WHERE DateMonth=Var_DateMonth AND phoneRate = Var_phoneRate;



if (N > 0) then
        update VM2
        set TOT_NumberOfCalls = TOT_NumberOfCalls + :NEW.NumberOfCalls, TOT_Price =
        TOT_Price + :NEW.Price
        where DateMonth= Var_DateMonth AND phoneRate = Var_phoneRate;

else
        insert into VM2 (DateMonth, phoneRate, Tot_NumberOfCalls, Tot_Price)
        values (Var_DateMonth, Var_phoneRate, :NEW.NumberOfCalls, :NEW.Price);

end if;
END;
```

Output VM1 after update via trigger.

DATEMONTH	DATEYEAR	TOT_NUMBEROFCALLS	TOT_PRICE
10-2003	2003	5148	17827740
7-2004	2004	5544	15168319
8-2003	2003	5694	19184722
7-2003	2003	5728	22829274
8-2004	2004	5496	20692410
10-2004	2004	5004	13996476

Output VM2 after update via trigger.

DATEMONTH	PHONERATE	TOT_NUMBEROFCALLS	TOT_PRICE
7-2003	24 ore su 24	588	583024
10-2003	Giorno	864	993600
7-2003	Notte	576	189420
7-2003	Giorno	1104	1232150
8-2003	CartaNatale	1608	1121320
10-2003	Notte	648	220320
8-2003	Giorno	1038	1039800
7-2003	Mattino	1944	19841940
7-2003	CartaNatale	1416	982640
8-2003	Notte	648	213900
10-2003	Mattino	1296	14813280
10-2003	24 ore su 24	540	540540
10-2003	CartaNatale	1800	1260000
8-2003	24 ore su 24	864	848172
8-2003	Mattino	1536	15961530
7-2003	Business	100	100