Queries

1) Select the yearly income for each phone rate, the total income for each phone rate, the total yearly income and the total income.

```
SELECT dateYear, phoneRateType, SUM(Price),
SUM(SUM(Price)) OVER (PARTITION BY phoneRateType), SUM(SUM(Price)) OVER
(PARTITION BY dateYear), SUM(SUM(Price)) OVER ()
FROM FACTS F, TIMEDIM T, PHONERATE P
WHERE F.Id_time = T.Id_time and F.Id_phoneRate = P.Id_phoneRate
GROUP BY phoneRateType, dateYear;
```

2) Select the monthly number of calls and the monthly income. Associate the RANK() to each month according to its income (1 for the month with the highest income, 2 for the second, etc., the last month is the one with the least income).

```
SELECT DateMonth, DateYear, SUM(NumberOfCalls) as TotNumOfCalls, SUM(price)
as TotalIncome, RANK() over (ORDER BY SUM(price) DESC) as RankIncome
FROM FACTS F, TIMEDIM T
WHERE F.id_time=T.id_time
GROUP BY DateMonth,DateYear;
```

3) For each month in 2003, select the total number of calls. Associate the RANK() to each month according to its total number of calls (1 for the month with the highest number of calls, 2 for the second, etc., the last month is the one with the least number of calls).

```
SELECT DateMonth, SUM(NumberOfCalls) as TotNumOfCalls,
RANK() over (ORDER BY SUM(NumberOfCalls) DESC) as RankNumOfCalls
FROM FACTS F, TIMEDIM T
WHERE F.id_time=T.id_time AND DateYear=2003
GROUP BY DateMonth;
```

4) For each day in July 2003, select the total income and the average income over the last 3 days.

```
SELECT DayDate , SUM(Price), AVG(SUM(Price)) OVER ( ORDER BY DayDate RANGE
BETWEEN INTERVAL '2' day preceding and current row) as avglast3days FROM
FACTS F, TIMEDIM T
WHERE  F.ID_time=T.ID_time  AND  DateYear=2003  AND  DateMonth= '7-2003'
GROUP BY DayDate;
```

```
SELECT DayDate, SUM(Price),
 AVG(SUM(Price)) OVER ( ORDER BY DayDate ROWS 2 preceding) as avglast3days
FROM FACTS F, TIMEDIM T
WHERE  F.ID_time=T.ID_time  AND  DateYear=2003  AND  DateMonth= '7-2003'
GROUP BY DayDate
ORDER BY DayDate;
```

5) Select the monthly income and the cumulative monthly income from the beginning of the year.

```
SELECT DateYear, DateMonth, SUM(Price) AS TOTINCOME,
SUM(SUM(PRICE)) OVER( PARTITION BY DateYear ORDER BY DateMonth ROWS UNBOUNDED
PRECEDING) AS CUMULATIVEINCOME
 FROM FACTS F, TIMEDIM T
WHERE F.ID_time=T.ID_time
GROUP BY DateMonth, DateYear;
```

6) Consider the year 2003. Separately for phone rate and month, analyze (i) the total income, (ii) the percentage of income with respect to the total revenue considering all the phone rates, (iii) the percentage of income with respect to the total revenue considering all the months.

```
SELECT DateMonth, phoneRateType, SUM(Price),
      SUM(Price)/SUM(SUM(Price)) OVER (PARTITION BY phoneRateType),
      SUM(Price)/SUM(SUM(Price)) OVER (PARTITION BY DateMonth)
FROM FACTS F, TIMEDIM T, PHONERATE P
WHERE F.Id_time = T.Id_time and F.Id_phoneRate = P.Id_phoneRate AND
DateYear=2003
GROUP BY phoneRateType, DateMonth;
```

7) For each caller region, select the monthly number of calls and the cumulative monthly number of calls from the beginning of the year.

```
SELECT DateYear, DateMonth, SUM(NumberOfCalls) AS TOTCALLS,
SUM(SUM(NumberOfCalls)) OVER( PARTITION BY DateYear, Region
                        ORDER BY DateMonth
                        ROWS UNBOUNDED PRECEDING) AS CUMULATIVECALLS_YEARs
FROM FACTS F, TIMEDIM T, LOCATION L
WHERE F.ID_time=T.ID_time and F.id_location_caller = L.Id_location
GROUP BY DateMonth, DateYear, Region;
```

8) Consider the year 2003. Analyze the total number of calls for (i) separately for each month and (ii) separately for each month, phone rate and caller region and (iii) separately for each month, phone rate and receiver region.

```
SELECT DateMonth, phoneRateType, LR.region, LC.region, SUM(NumberOfCalls)
FROM FACTS F, TIMEDIM T, PHONERATE P, LOCATION LC,  LOCATION LR
WHERE F.Id_time = T.Id_time and F.Id_phoneRate = P.Id_phoneRate and
DateYear=2003 and F.Id_location_receiver = LR.Id_location and
F.Id_location_caller = LC.Id_location
GROUP BY GROUPING SETS ((DateMonth), (DateMonth, phoneRateType,
LR.region), (DateMonth, phoneRateType, LC.region));
```