

Queries

/*

Query 1

*/

```
SELECT tipo_tariffa, anno, SUM(prezzo)
  FROM DWABD.Fatti F, DWABD.Tempo Te, DWABD.Tariffa Ta
 WHERE F.id_tempo=Te.id_tempo
       AND F.id_tar=Ta.id_tar
 GROUP BY CUBE(tipo_tariffa, anno);
```

/*

Query 2

*/

```
SELECT mese,anno, SUM(chiamate) as TotChiamate, SUM(prezzo) as Incasso,
       RANK() over (ORDER BY SUM(prezzo) DESC) as RankIncasso
  FROM DWABD.Fatti F, DWABD.Tempo Te
 WHERE F.id_tempo=Te.id_tempo
 GROUP BY mese,anno;
```

/*

Query 3

*/

```
SELECT mese, SUM(chiamate) as TotChiamate,
       RANK() over (ORDER BY SUM(chiamate) DESC) as RankChiamate
  FROM DWABD.Fatti F, DWABD.Tempo Te
 WHERE F.id_tempo=Te.id_tempo
       AND anno=2003
 GROUP BY mese;
```

/*

Query 4

*/

```
SELECT data, SUM(prezzo),
       AVG(SUM(prezzo)) OVER (ORDER BY data RANGE BETWEEN INTERVAL '2' day
                              preceding and current row) as MediaUltimi3Giorni
  FROM DWABD.Fatti F, DWABD.Tempo Te
 WHERE F.id_tempo=Te.id_tempo
       AND anno=2003 AND mese=7
 GROUP BY data;
```

/*

Query 5

*/

```
SELECT mese, anno, SUM(prezzo) as Incasso,
       SUM(SUM(prezzo)) over (PARTITION BY anno ORDER BY mese rows unbounded preceding)
  as IncassoCumulativo
  FROM DWABD.Fatti F, DWABD.Tempo Te
 WHERE F.id_tempo=Te.id_tempo
 GROUP BY mese, anno;
```

Materialized views

The cardinality of all queries is at least one order of magnitude lower than those of the fact table. Hence, for each query it may be potentially useful to create a materialized view.

Queries 2, 3, and 5 are pretty similar. To answer these queries efficiently we can create a single materialized view, which is reported below.

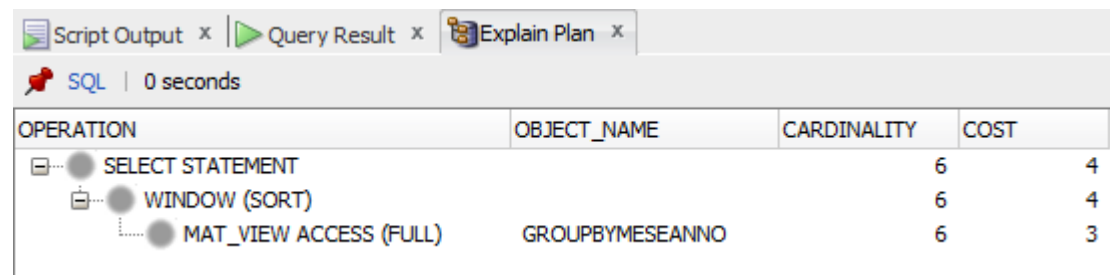
```
create materialized view groupByMeseAnno
  build immediate
  refresh on demand
  --enable query rewrite
as
select Mese, Anno, SUM(Chiamate) as NumSpese, SUM(Prezzo) as TotSpesa
  from DWABD.FATTI F, DWABD.TEMPO T
 where F.id_tempo=T.id_tempo
 group by Mese,Anno;
```

Queries using materialized view

```
/*
Query 2
*/
```

```
SELECT mese, anno, NumSpese, TotSpesa,
RANK() over (ORDER BY TotSpesa DESC) as RankIncasso
FROM GROUPBYMESEANNO;
```

```
/*
Explain plan
*/
```



OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		6	4
WINDOW (SORT)		6	4
MAT_VIEW ACCESS (FULL)	GROUPBYMESEANNO	6	3

/*

Query 3

*/

```
SELECT mese, NumSpese,  
RANK() over (ORDER BY NumSpese DESC) as RankChiamate  
FROM GROUPBYMESEANNO  
WHERE anno=2003;
```

/*

Explain plan

*/

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		3	4
WINDOW (SORT)		3	4
MAT_VIEW ACCESS (FULL)	GROUPBYMESEANNO	3	3
Filter Predicates			
ANNO=2003			

/*

Query 5

*/

```
SELECT mese, anno, TotSpesa,  
SUM(TotSpesa) over (PARTITION BY anno ORDER BY mese rows unbounded  
preceding) as  
IncassoCumulativo  
FROM GROUPBYMESEANNO;
```

/*

Explain plan

*/

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		6	4
WINDOW (SORT)		6	4
MAT_VIEW ACCESS (FULL)	GROUPBYMESEANNO	6	3