

Esercizio di [Materialized View and triggers](#)

a)SELECT ServiceType,Semester,Sum(income),Sum(#consultancies)

FROM Income i, Service s,Time t,Consultants\_Branch c

WHERE and Region='Lombardia'

GROUP BY ServiceType,Semester,i

b)SELECT

sum(income),

sum(consultancies)

FROM Income,Company,Service,Consultants\_Branch c,Time t

WHERE Nationality = 'Italy' or Nationality='Germany'

GROUP BY Region,Service,ServiceID,Year

c)SELECT

sum(income),

sum(income)/sum(#consultancies)

FROM Income i, Company c, Service s, Time t

WHERE Year = '2017' OR Year='2018 OR Year='2019'

GROUP BY ServiceType,Nationality,Semester

## 1. MatViewQuery:

Come farla:

Nel group by tutti gli attributi di tutti i group by+where

Nel select tutte le measure tranne i ratio

Nel from tutte le tabelle

SELECT ServiceType,Semester,Region,Service,Year,Nationality,

SUM(Income),Sum(#consultancies)

```
FROM Income i,Service s,Time t,Company c,Consultants_Branch cb
```

```
WHERE i.serviceid=s.serviceid and i.timeid=t.timeid and i.companyid=c.companyid  
and i.branchid=cb.branchid
```

```
GROUP BY ServiceType,Semester,Region,Service,Year,Nationality
```

## 2. Identifier della MatView:

Selezionare il livello più basso di ogni dimension  
quindi: Service,Region,Nationality,Semester

## 3.Popolare la mat view creata invece come create statement:

Insert into matview(attributi)(query per creare la mat view)

## 4. Trigger per aggiornare la mat view dalla fact table:

```
CREATE TRIGGER RefreshViewIncome # Creazione
AFTER INSERT ON Income -- AFTER + evento(INSERT ON/UPDATE ON)
FOR EACH ROW --Granularità, sempre for each row
DECLARE -- variables
N number; -- c'è sempre N s!
--tutte le variabili degli attributi da salvare nella mat view
VarService,VarServiceType,VarNationality,VarSemester,VarYear,VarRegion VARCHAR(20);

--Per ogni dimension, faccio una query per ripescarmi i singoli attributi
--In un trigger, ho l'oggetto NEW per accedere al record appena inserito
BEGIN
--Service Dimension
SELECT Service INTO VarService,ServiceType INTO VarServiceType
FROM Service
WHERE ServiceID = :NEW.ServiceID;;
--Consultants_branch
SELECT Region INTO VarRegion
FROM Consultants_branch
WHERE BranchID=:NEW.BranchID;
--.. stessa cosa per le altre dimensioni
--Controllo se c'è un record nella MatView con tutte queste informazioni
SELECT COUNT(*) INTO N
From ViewIncome
--soltanto per gli attributi più bassi delle dimensioni
WHERE Service = VarService AND Nationality=VarNationality AND Semester=VarSemester
AND
        Region=VarRegion
IF(N>0) THEN
    --aggiorno le measure
    UPDATE ViewIncome
    --Aggiorno gli attributi con le measure delle fact table
```

```

    SET TotIncome= TotIncome + :NEW.Income
    TotConsultancies=TotConsultancies+ :NEW.#Consultancies
    WHERE Service = VarService AND Nationality=VarNationality AND
Semester=VarSemester AND
    Region=VarRegion
ELSE
    INSERT INTO
ViewIncome(Service,ServiceType,Nationality,Semester,Year,Region,
    TotIncome,TotConsultancies(VarService,VarServiceType,#.#)
END;

```

## 5. Trigger 2(aggiornamento di un singolo attributo e propagarlo alla matview)

```

CREATE TRIGGER UpdateServiceType
AFTER UPDATE OF ServiceType ON SERVICE
FOR EACH ROW
DECLARE
N Number;
BEGIN

SELECT COUNT(*) INTO N
FROM ViewIncome
WHERE ServiceType = :OLD.ServiceType

IF(N>0) THEN
    UPDATE ViewIncome
    SET ServiceType= :NEW.ServiceType
    WHERE ServiceType = :OLD.ServiceType
ENDIF;
END;

```

## 6. Materialized View Log per l'automatic fast update

```

CREATE MATERIALIZED VIEW LOG ON Income
WITH SEQUENCE,ROWID
(BranchID,ServiceID,TimeID,CompanyCategoryID,#Consultancies,Income)
INCLUDING NEW VALUES

CREATE MATERIALIZED VIEW LOG ON Service
WITH SEQUENCE,ROWID
(ServiceID,Service,ServiceType)
INCLUDING NEW VALUES

```

```
--Replicare per tutte le tabelle di tutti gli attributi da cui derivano quelli  
presenti nella mat view
```

## Conclusioni

Negli scorsi anni, solo i punti 1,2 e 4 sono richiesti!!!!!!