

## Esercizio sulle viste materializzate

È dato il seguente schema relazionale di un data mart

```
INCASSO(IdSede, IdServizio, IdCategoriaAzienda, IdTempo, #Consulenze, Incasso)
SERVIZIO(IdServizio, Servizio, TipologiaServizio)
TEMPO(IdTempo, Data, Mese, Bimestre, Trimestre, Quadrimestre, Semestre, Anno)
SEDE-CONSULENTI(IdSede, Sede, Città, Regione, AreaGeografica, #Consulenti)
AZIENDA(IdCategoriaAzienda, CategoriaAzienda, SettoreCommerciale, Nazionalità)
```

### Punto 1

Dato lo schema logico precedente, considerare le seguenti query di interesse:

- Per ogni coppia (tipologia dei servizi, semestre), visualizzare l'incasso totale e il numero totale di consulenze, effettuate dai consulenti delle sedi site in Lombardia.
- Considerando solo le aziende italiane e tedesche, selezionare per ogni coppia (Regione delle sedi dei consulenti, Servizio), l'incasso totale e il numero di consulenze effettuate, separatamente per ogni anno.
- Considerando solo gli incassi del 2017, 2018, e 2019, per ogni coppia (Tipologia dei servizi, nazionalità dell'azienda), visualizzare l'incasso semestrale e l'incasso semestrale medio per consulenza.

Considerando le informazioni precedenti, definire una vista materializzata con CREATE MATERIALIZED VIEW, in modo da ridurre il tempo di risposta delle query di interesse da (a) a (c) sopra riportate. In particolare si specifichi la query SQL associata al Blocco A nella seguente istruzione:

```
CREATE MATERIALIZED VIEW ViewIncassi
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS
    Blocco A
```

### Punto 2

Si individui la combinazione minimale di attributi che costituisce un identificatore per la vista materializzata ViewIncassi.

### Punto 3

Si ipotizzi che la vista materializzata ViewIncassi sia gestita **senza utilizzare** l'istruzione CREATE MATERIALIZED VIEW. La tabella derivata che realizza la vista materializzata è definita dalla seguente istruzione SQL

```
CREATE TABLE ViewIncassi
(
```

```

Servizio      VARCHAR(20),
TipologiaServizio  VARCHAR(20),
Nazionalità   VARCHAR(20),
Semestre     VARCHAR(20),
Anno         VARCHAR(20),
Regione      VARCHAR(20),
IncassoTot   INTEGER
              CHECK (IncassoTot IS NOT NULL and IncassoTot >0),
NumConsulenzeTot  INTEGER
              CHECK (NumConsulenzeTot IS NOT NULL and NumConsulenzeTot >0),
PRIMARY KEY (Nazionalità, Semestre, Regione, Servizio)
)

```

Si scriva l'istruzione INSERT per il caricamento iniziale dei dati nella vista materializzata ViewIncassi, utilizzando l'istruzione INSERT(SELECT...).

#### **Punto 4**

Si ipotizzi che la gestione dell'aggiornamento della vista materializzata sia svolta mediante trigger. Si scriva il trigger per propagare le modifiche alla vista materializzata ViewIncassi in caso di inserimento di un nuovo record nella tabella dei fatti INCASSO.

#### **Punto 5**

Si ipotizzi che la gestione dell'aggiornamento della vista materializzata sia svolta mediante trigger. Si scriva il trigger per propagare l'aggiornamento (considerando la modalità di gestione del tempo di tipo 1) del valore dell'attributo TipologiaServizio della tabella SERVIZIO alla vista materializzata ViewIncassi.

#### **Punto 6**

La vista materializzata VM1 è creata con la seguente istruzione SQL

```

CREATE MATERIALIZED VIEW
BUILD IMMEDIATE
REFRESH FAST ON DEMAND
ENABLE QUERY REWRITE
AS <Blocco A>

```

Per la definizione del Blocco A, si veda il punto 1.

Scrivere le istruzioni che definiscono i MATERIALIZED VIEW LOG in Oracle necessari per l'aggiornamento automatico FAST della vista materializzata ViewIncassi. Si indichino **tutti e soli** i log necessari e all'interno di ogni definizione di log **tutti e soli** gli attributi necessari.

## BOZZA DI SOLUZIONE

### Punto 1

A: GROUP BY: TipologiaServizi, Semestre  
WHERE regione = Lombardia  
Aggregati: SUM(incasso), SUM(#Consulenze)  
FROM: INCASSO, SERVIZIO, TEMPO, SEDE-CONSULENTI

#### Query A

```
SELECT Tiposervizio, Semestre, SUM(Incasso), SUM(#Consulenze)
FROM INCASSO, SEDE, SERVIZIO, TEMPO
WHERE <join>
AND Regione = 'Lombardia'
GROUP BY TipoServizio, Semestre
```

B: GROUP BY: Regione, Servizio, Anno  
Aggregati: SUM(Incasso), SUM(#Consulenze)  
WHERE Nazionalità= 'Italiana' OR Nazionalità ='Tedesca'  
FROM: INCASSO, SERVIZIO, TEMPO, AZIENDA, SEDE-CONSULENTI

#### Query B

```
SELECT Regione, Servizio, Anno, SUM(Incasso), SUM(#Consulenze)
FROM INCASSO, SEDE, SERVIZIO, TEMPO
WHERE <join>
AND (Nazionalità= 'Italiana' OR Nazionalità ='Tedesca')
GROUP BY Regione, Servizio, Anno
```

C: GROUP BY TipologiaServizio, Nazionalità, Semestre  
Aggregati: SUM(Incasso), SUM(#Consulenze)  
WHERE Anno >= 2017 AND Anno <= 2019  
FROM: INCASSO, SERVIZIO, TEMPO, AZIENDA

#### Query C

```
SELECT TipologiaServizio, Nazionalità, Semestre, SUM(Incasso), SUM(Incasso)/SUM(#Consulenze)
FROM INCASSO, SERVIZIO, TEMPO, AZIENDA
WHERE <join>
AND Anno >= 2017 AND Anno <= 2019
GROUP BY TipologiaServizio, Nazionalità, Semestre
```

### **Blocco A**

```
SELECT Servizio, TipologiaServizio, Nazionalità, Semestre, Anno, Regione,  
       SUM(Incasso) AS IncassoTot,  
       SUM(#Consulenze) AS NumConsulenzeTot  
FROM INCASSO I, SERVIZIO S, TEMPO T, AZIENDA A, SEDE-CONSULENTI SC  
WHERE I.IdServizio=S.IdServizio AND I.IdTempo=T.IdTempo AND  
A.IdCategoriaAzienda=I.IdCategoriaAzienda AND SC.IdSede=I.IdSede  
GROUP BY Servizio, Nazionalità, Semestre, Regione, TipologiaServizio, Anno
```

### **Punto 2**

Identificatore minimale: Servizio, Nazionalità, Semestre, Regione

Note:

Semestre → Anno

Servizio → TipologiaServizio

### **Punto 3**

```
INSERT INTO ViewIncassi (Servizio, TipologiaServizio, Nazionalità, Semestre, Anno, Regione,  
                        IncassoTot, NumConsulenzeTot)  
  (SELECT Servizio, TipologiaServizio, Nazionalità, Semestre, Anno, Regione,  
   SUM(Incasso), SUM(#Consulenze)  
  FROM INCASSO I, SERVIZIO S, TEMPO T, AZIENDA A, SEDE-CONSULENTI SC  
  WHERE I.IdServizio=S.IdServizio AND I.IdTempo=T.IdTempo AND  
  A.IdCategoriaAzienda=I.IdCategoriaAzienda AND SC.IdSede=I.IdSede  
  GROUP BY Servizio, Nazionalità, Semestre, Regione, TipologiaServizio, Anno);
```

### **Punto 4**

```
CREATE TRIGGER RefreshViewIncassi  
AFTER INSERT ON INCASSO  
FOR EACH ROW  
DECLARE  
N number;  
VarServizio, VarTipoS VARCHAR(20); VarNazionalità VARCHAR(20);  
VarSemestre, VarAnno VARCHAR(20); VarRegione VARCHAR(20);
```

BEGIN

--- Leggo nelle diverse tabelle dimensionali i valori di Servizio, Nazionalità, Semestre, Regione corrispondenti ai nuovi valori inseriti di :NEW.IdServizio, :NEW.IdCategoriaAzienda, :NEW.IdTempo, :NEW.IdSede

```
SELECT Servizio, TipologiaServizio INTO VarServizio, VarTipoS
FROM SERVIZIO
WHERE IdServizio = :NEW.IdServizio;
```

```
SELECT Regione INTO VarRegione
FROM SEDE-CONSULENTI
WHERE IdSede = :NEW.IdSede;
```

```
SELECT Semestre, Anno INTO VarSemestre, VarAnno
FROM TEMPO
WHERE IdTempo = :NEW.IdTempo;
```

```
SELECT Nazionalità INTO VarNazionalità
FROM AZIENDA
WHERE IdCategoriaAzienda = :NEW. IdCategoriaAzienda;
```

--- Verifico se esiste una tupla in ViewIncassi associata ai valori di Servizio, Nazionalità, Semestre, Regione appena letti (identificatore minimale della tupla di ViewIncassi).

```
SELECT Count(*) INTO N
FROM ViewIncassi
WHERE Servizio=VarServizio AND Nazionalità = VarNazionalità AND Semestre= VarSemestre
AND Regione=VarRegione;
```

IF (N > 0) THEN

--- Se esiste la tupla occorre aggiornare il valore degli aggregati

```
UPDATE ViewIncassi
SET IncassoTot = IncassoTot + :NEW.Incasso
    NumConsulenzeTot = NumConsulenzeTot + :NEW. #Consulenze
WHERE Servizio=VarServizio AND Nazionalità = VarNazionalità
AND Semestre= VarSemestre AND Regione=VarRegione;
```

ELSE

--- Se non esiste la tupla, occorre inserire un nuovo record

```
INSERT INTO ViewIncassi (Servizio, TipologiaServizio, Nazionalità, Semestre, Anno,
    Regione, IncassoTot, NumConsulenzeTot)
VALUES (VarServizio, VarTipoS, VarNazionalità, VarSemestre, VarAnno, VarRegione,
    :NEW.Incasso, :NEW. #Consulenze);
```

END IF;

END;

### **Punto 5**

```
CREATE TRIGGER AggiornamentoTipologiaServizio
AFTER UPDATE OF TipologiaServizio ON SERVIZIO
```

```

FOR EACH ROW
DECLARE
N number;
BEGIN

--- Verifico se quella TipologiaServizio è presente nella vista materializzata
SELECT Count(*) INTO N
FROM ViewIncassi
WHERE TipologiaServizio=:OLD.TipologiaServizio;

IF (N > 0) THEN
--- E' presente quel valore di TipologiaServizio
--- Aggiorno tutti i record associati al precedente valore di TipologiaServizio
    UPDATE ViewIncassi
    SET TipologiaServizio=:NEW.TipologiaServizio
    WHERE TipologiaServizio= :OLD.TipologiaServizio;

END IF;
END;

```

### **Soluzione alternativa**

```

CREATE TRIGGER AggiornamentoTipologiaServizio
AFTER UPDATE OF TipologiaServizio ON SERVIZIO
FOR EACH ROW
DECLARE
N number;
BEGIN

--- Operazione di aggiornamento
    UPDATE ViewIncassi
    SET TipologiaServizio=:NEW.TipologiaServizio
    WHERE TipologiaServizio= :OLD.TipologiaServizio;

END IF;
END;

```

### ***Punto 6***

```

CREATE MATERIALIZED VIEW LOG ON INCASSO
WITH SEQUENCE, ROWID
(IdSede, IdServizio, IdTempo, IdCategoriaAzienda, #Consulenze, Incasso)
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON SERVIZIO
WITH SEQUENCE, ROWID
(IdServizio, Servizio, TipologiaServizio)
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON TEMPO
WITH SEQUENCE, ROWID

```

(IdTempo, Semestre, Anno)  
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON SEDE-CONSULENTI  
WITH SEQUENCE, ROWID  
(IdSede, Regione)  
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON AZIENDA  
WITH SEQUENCE, ROWID  
(IdCategoriaAzienda, Nazionalità)  
INCLUDING NEW VALUES;