

# Business Intelligence – Lab 9

Data Warehouse – Società di Telefonia Mobile

Schema Logico e Interrogazioni SQL

## 1. Schema Concettuale

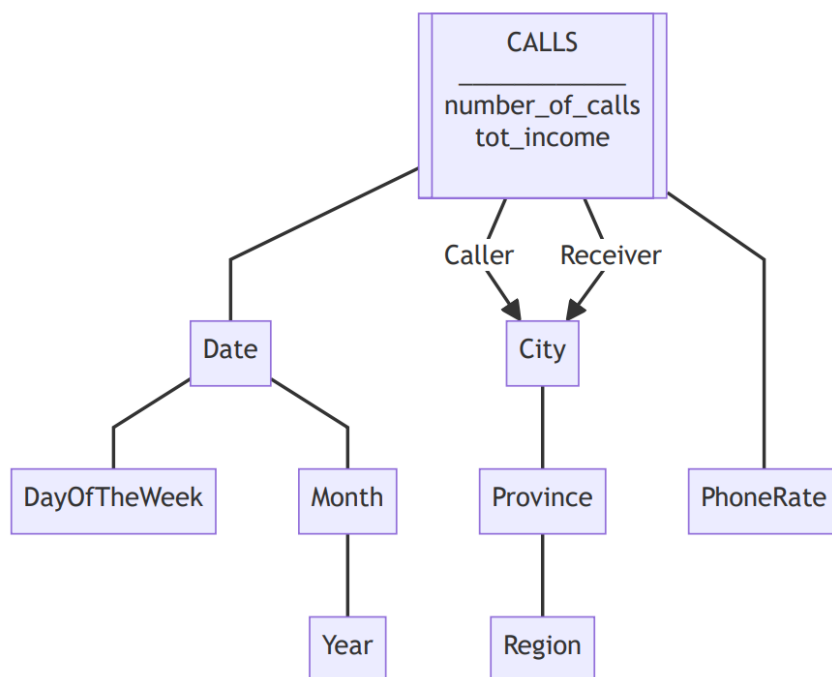
---

Il data warehouse adotta uno Star Schema con una tabella dei fatti centrale e tre dimensioni:

- TEMPO – granularità giornaliera (Data, GiornoDellaSettimana, Mese, Anno)
- TARIFFA (PhoneRate) – tipologia di tariffa applicata alla chiamata
- LUOGO – città, provincia e regione (per chiamante e chiamato)

**Le misure nella tabella dei fatti sono:**

- INCASSO (FLOAT) – incasso totale aggregato per la combinazione chiave
- NUM\_CHIAMATE (INT) – numero di chiamate aggregate per la combinazione chiave



## 2. Schema Logico

Le tabelle del data warehouse sono nel schema DWABD. Di seguito la struttura dettagliata.

### 2.1 DWABD.TEMPO

Cardinalità: 30 tuple (una per ogni giorno distinto presente nei dati)

Colonna	Tipo	Vincolo	Descrizione
ID_TEMPO	INT	PK	Chiave surrogata
DATA	DATE	NOT NULL	Data della chiamata
GIORNODellaSettimana	CHAR(15)	NOT NULL	Nome giorno (Lunedì...)
MESE	CHAR(15)	NOT NULL	Nome mese (Gennaio...)
ANNO	INT	NOT NULL	Anno della chiamata

```
CREATE TABLE DWABD.TEMPO (  
  ID_TEMPO INT NOT NULL,  
  DATA DATE NOT NULL,  
  GIORNODellaSettimana CHAR(15) NOT NULL,  
  MESE CHAR(15) NOT NULL,  
  ANNO INT NOT NULL,  
  PRIMARY KEY (ID_TEMPO)  
);
```

### 2.2 DWABD.TARIFFA

Cardinalità: 7 tuple (una per tipologia di tariffa).

Colonna	Tipo	Vincolo	Descrizione
ID_TAR	INT	PK	Chiave surrogata
TIPO_TARIFFA	CHAR(20)	NOT NULL	Nome tariffa (Mattino...)

```
CREATE TABLE DWABD.TARIFFA (  
  ID_TAR INT NOT NULL,  
  TIPO_TARIFFA CHAR(20) NOT NULL,  
  PRIMARY KEY (ID_TAR)  
);
```

### 2.3 DWABD.LUOGO

Cardinalità: 1500 tuple. Usata con role-playing: rappresenta sia il luogo del chiamante che del chiamato.

Colonna	Tipo	Vincolo	Descrizione
ID_LUOGO	INT	PK	Chiave surrogata
CITTA	CHAR(20)	NOT NULL	Nome città
PROVINCIA	CHAR(20)	NOT NULL	Nome provincia
REGIONE	CHAR(20)	NOT NULL	Nome regione

```
CREATE TABLE DWABD.LUOGO (
  ID_LUOGO INT NOT NULL,
  CITTA CHAR(20) NOT NULL,
  PROVINCIA CHAR(20) NOT NULL,
  REGIONE CHAR(20) NOT NULL,
  PRIMARY KEY (ID_LUOGO)
);
```

## 2.4 DWABD.FATTI

Cardinalità: 7809 tuple. La chiave primaria è composta da tutte e quattro le chiavi esterne.  
Granularità: una riga per ogni combinazione (giorno, tariffa, luogo chiamante, luogo chiamato).

Colonna	Tipo	Vincolo	Descrizione
ID_TEMPO	INT	PK, FK→TEMPO	Riferimento dimensione Tempo
ID_TAR	INT	PK, FK→TARIFFA	Riferimento dimensione Tariffa
ID_LUOGO_CHIAMANTE	INT	PK, FK→LUOGO	Luogo apparecchio chiamante
ID_LUOGO_CHIAMATO	INT	PK, FK→LUOGO	Luogo apparecchio chiamato
INCASSO	FLOAT	NOT NULL	Misura: incasso in euro
NUM_CHIAMATE	INT	NOT NULL	Misura: numero di chiamate

```
CREATE TABLE DWABD.FATTI (
  ID_TEMPO INT NOT NULL,
  ID_TAR INT NOT NULL,
  ID_LUOGO_CHIAMANTE INT NOT NULL,
  ID_LUOGO_CHIAMATO INT NOT NULL,
  INCASSO FLOAT NOT NULL,
  NUM_CHIAMATE INT NOT NULL,
  PRIMARY KEY (ID_TEMPO, ID_TAR,
  ID_LUOGO_CHIAMANTE, ID_LUOGO_CHIAMATO),
  FOREIGN KEY (ID_TEMPO) REFERENCES DWABD.TEMPO (ID_TEMPO),
  FOREIGN KEY (ID_TAR) REFERENCES DWABD.TARIFFA (ID_TAR),
  FOREIGN KEY (ID_LUOGO_CHIAMANTE) REFERENCES DWABD.LUOGO (ID_LUOGO),
  FOREIGN KEY (ID_LUOGO_CHIAMATO) REFERENCES DWABD.LUOGO (ID_LUOGO)
);
```



### 3. Interrogazioni SQL

---

Le query seguenti utilizzano SQL esteso con funzioni analitiche Oracle (RANK, SUM OVER, AVG OVER, ROLLUP).

#### Query 1 – Incasso per (tipo tariffa, anno) con subtotali ROLLUP

Seleziona l'incasso totale per ogni coppia (tipo tariffa, anno), il subtotale per ogni tipo di tariffa (indipendentemente dall'anno), il subtotale per ogni anno (indipendentemente dalla tariffa) e il totale complessivo. Si utilizza ROLLUP che genera automaticamente tutti i livelli di aggregazione.

```
SELECT
    t.TIPO_TARIFFA,
    tp.ANNO,
    SUM(f.INCASSO) AS INCASSO_TOTALE
FROM
    DWABD.FATTI f
    JOIN DWABD.TARIFFA t ON f.ID_TAR = t.ID_TAR
    JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
GROUP BY ROLLUP(t.TIPO_TARIFFA, tp.ANNO)
ORDER BY t.TIPO_TARIFFA, tp.ANNO;
```

*Nota: ROLLUP(A,B) produce le aggregazioni (A,B), (A,NULL) e (NULL,NULL), ovvero il dettaglio, il subtotale per tariffa e il totale assoluto.*

#### Query 2 – Chiamate e incasso per mese con RANK sull'incasso

Per ogni mese calcola il numero totale di chiamate e l'incasso totale, e assegna un rank crescente dove 1 corrisponde al mese con il maggior incasso.

```
SELECT
    tp.MESE,
    SUM(f.NUM_CHIAMATE) AS TOT_CHIAMATE,
    SUM(f.INCASSO) AS INCASSO_TOTALE,
    RANK() OVER (ORDER BY SUM(f.INCASSO) DESC) AS RANK_INCASSO
FROM
    DWABD.FATTI f
    JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
GROUP BY tp.MESE
ORDER BY RANK_INCASSO;
```

#### Query 3 – Chiamate per mese del 2003 con RANK sul numero di chiamate

Considera solo le chiamate del 2003. Per ogni mese calcola il totale di chiamate e assegna un rank dove 1 è il mese con più chiamate.

```
SELECT
    tp.MESE,
    SUM(f.NUM_CHIAMATE) AS TOT_CHIAMATE,
    RANK() OVER (ORDER BY SUM(f.NUM_CHIAMATE) DESC) AS RANK_CHIAMATE
FROM
```

```
DWABD.FATTI f
JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
WHERE tp.ANNO = 2003
GROUP BY tp.MESE
ORDER BY RANK_CHIAMATE;
```

## Query 4 – Incasso giornaliero luglio 2003 e media mobile 3 giorni

Per ogni giorno di luglio 2003 mostra l'incasso totale e la media degli incassi degli ultimi tre giorni (giorno corrente + 2 precedenti) usando una finestra scorrevole.

```
SELECT
    tp.DATA,
    SUM(f.INCASSO) AS INCASSO_GIORNO,
    AVG(SUM(f.INCASSO)) OVER (
        ORDER BY tp.DATA
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
    ) AS MEDIA_3GG
FROM
    DWABD.FATTI f
    JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
WHERE tp.ANNO = 2003
    AND tp.MESE = 'Luglio'
GROUP BY tp.DATA
ORDER BY tp.DATA;
```

*Nota: ROWS BETWEEN 2 PRECEDING AND CURRENT ROW definisce una finestra di 3 righe (la riga corrente e le 2 precedenti per data).*

## Query 5 – Incasso mensile e incasso cumulativo dall'inizio dell'anno

Per ogni mese mostra l'incasso del mese e la somma cumulativa dall'inizio dell'anno. La PARTITION BY ANNO fa ripartire il cumulato ad ogni anno.

```
SELECT
    tp.ANNO,
    tp.MESE,
    SUM(f.INCASSO) AS INCASSO_MESE,
    SUM(SUM(f.INCASSO)) OVER (
        PARTITION BY tp.ANNO
        ORDER BY MIN(tp.DATA)
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS INCASSO_CUMULATIVO
FROM
    DWABD.FATTI f
    JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
GROUP BY tp.ANNO, tp.MESE
ORDER BY tp.ANNO, MIN(tp.DATA);
```

*Nota: MIN(tp.DATA) viene usato nell'ORDER BY della funzione analitica per garantire l'ordine cronologico dei mesi (il nome del mese non è ordinabile alfabeticamente in modo corretto).*

## Query 6 – Anno 2004, tariffa Mattino: chiamate, incasso, cumulativo per regione

Considera solo le chiamate del 2004 con tariffa 'Mattino'. Per ogni regione del chiamante e per ogni mese, mostra il numero di chiamate, l'incasso totale e l'incasso cumulativo dall'inizio dell'anno, separatamente per regione.

```
SELECT
  lc.REGIONE,
  tp.MESE,
  SUM(f.NUM_CHIAMATE) AS TOT_CHIAMATE,
  SUM(f.INCASSO)      AS INCASSO_MESE,
  SUM(SUM(f.INCASSO)) OVER (
    PARTITION BY lc.REGIONE, tp.ANNO
    ORDER BY MIN(tp.DATA)
    ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
  ) AS INCASSO_CUMULATIVO
FROM
  DWABD.FATTI f
  JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
  JOIN DWABD.TARIFFA t ON f.ID_TAR = t.ID_TAR
  JOIN DWABD.LUOGO lc ON f.ID_LUOGO_CHIAMANTE = lc.ID_LUOGO
WHERE tp.ANNO = 2004
  AND TRIM(t.TIPO_TARIFFA) = 'Mattino'
GROUP BY lc.REGIONE, tp.ANNO, tp.MESE
ORDER BY lc.REGIONE, tp.ANNO, MIN(tp.DATA);
```

## Query 7 – Città chiamato = 'Citta6': incasso medio giornaliero e subtotali per città chiamante

Considera solo le chiamate verso la Citta6. Per ogni coppia (città del chiamante, mese) mostra l'incasso medio giornaliero, il numero totale di chiamate e l'incasso totale. Tramite ROLLUP aggiunge anche il subtotale per città chiamante (indipendentemente dal mese).

```
SELECT
  lc.CITTA          AS CITTA_CHIAMANTE,
  tp.MESE,
  SUM(f.INCASSO) / COUNT(DISTINCT tp.DATA) AS INCASSO_MEDIO_GG,
  SUM(f.NUM_CHIAMATE) AS TOT_CHIAMATE,
  SUM(f.INCASSO) AS INCASSO_TOTALE
FROM
  DWABD.FATTI f
  JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
  JOIN DWABD.LUOGO lc ON f.ID_LUOGO_CHIAMANTE = lc.ID_LUOGO
  JOIN DWABD.LUOGO lt ON f.ID_LUOGO_CHIAMATO = lt.ID_LUOGO
WHERE TRIM(lt.CITTA) = 'Citta6'
GROUP BY ROLLUP(lc.CITTA, tp.MESE)
ORDER BY lc.CITTA, tp.MESE;
```

*Nota: ROLLUP(lc.CITTA, tp.MESE) genera tre livelli: (città, mese), (città, NULL) ovvero subtotale per città, e (NULL, NULL) ovvero il totale assoluto.*

## 4. Viste Materializzate

---

Le viste materializzate pre-aggregano i dati per ridurre i tempi di risposta delle query più frequenti. Si identificano due viste principali.

### 4.1 MV\_MENSILE\_TARIFFA\_REGIONE

Utile per: Query 1, 2, 5, 6. Aggrega i fatti a livello mensile per tipo tariffa e regione del chiamante.

```
CREATE MATERIALIZED VIEW DWABD.MV_MENSILE_TARIFFA_REGIONE
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS
SELECT
  tp.ANNO,
  tp.MESE,
  MIN(tp.DATA)          AS PRIMA_DATA,
  t.TIPO_TARIFFA,
  lc.REGIONE            AS REGIONE_CHIAMANTE,
  SUM(f.INCASSO)        AS INCASSO_TOTALE,
  SUM(f.NUM_CHIAMATE)  AS TOT_CHIAMATE
FROM
  DWABD.FATTI f
  JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
  JOIN DWABD.TARIFFA t ON f.ID_TAR = t.ID_TAR
  JOIN DWABD.LUOGO lc ON f.ID_LUOGO_CHIAMANTE = lc.ID_LUOGO
GROUP BY tp.ANNO, tp.MESE, t.TIPO_TARIFFA, lc.REGIONE;
```

### 4.2 MV\_GIORNALIERO\_CITTA

Utile per: Query 4, 7. Aggrega i fatti a livello giornaliero per città del chiamante e città del chiamato.

```
CREATE MATERIALIZED VIEW DWABD.MV_GIORNALIERO_CITTA
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS
SELECT
  tp.DATA,
  tp.ANNO,
  tp.MESE,
  lc.CITTA AS CITTA_CHIAMANTE,
  lt.CITTA AS CITTA_CHIAMATO,
  SUM(f.INCASSO) AS INCASSO_TOTALE,
  SUM(f.NUM_CHIAMATE) AS TOT_CHIAMATE
FROM
  DWABD.FATTI f
  JOIN DWABD.TEMPO tp ON f.ID_TEMPO = tp.ID_TEMPO
  JOIN DWABD.LUOGO lc ON f.ID_LUOGO_CHIAMANTE = lc.ID_LUOGO
  JOIN DWABD.LUOGO lt ON f.ID_LUOGO_CHIAMATO = lt.ID_LUOGO
GROUP BY tp.DATA, tp.ANNO, tp.MESE, lc.CITTA, lt.CITTA;
```

### 4.3 Riscrittura Query 4 con MV\_GIORNALIERO\_CITTA

```

SELECT
    DATA,
    SUM(INCASSO_TOTALE) AS INCASSO_GIORNO,
    AVG(SUM(INCASSO_TOTALE)) OVER (
        ORDER BY DATA
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
    ) AS MEDIA_3GG
FROM
    DWABD.MV_GIORNALIERO_CITTA
WHERE ANNO = 2003
    AND MESE = 'Luglio'
GROUP BY DATA
ORDER BY DATA;

```

#### 4.4 Riscrittura Query 7 con MV\_GIORNALIERO\_CITTA

```

SELECT
    CITTA_CHIAMANTE,
    MESE,
    SUM(INCASSO_TOTALE) / COUNT(DISTINCT DATA) AS INCASSO_MEDIO_GG,
    SUM(TOT_CHIAMATE) AS TOT_CHIAMATE,
    SUM(INCASSO_TOTALE) AS INCASSO_TOTALE
FROM
    DWABD.MV_GIORNALIERO_CITTA
WHERE TRIM(CITTA_CHIAMATO) = 'Citta6'
GROUP BY ROLLUP(CITTA_CHIAMANTE, MESE)
ORDER BY CITTA_CHIAMANTE, MESE;

```