

# Sistemi per la Gestione di Basi di Dati

## Esercitazione di laboratorio n. 3

### Data warehouse: SQL esteso e viste materializzate in Oracle

La finalità di questa esercitazione consiste nella realizzazione di un data warehouse conforme alle specifiche riportate nei punti seguenti, utilizzando Oracle. Oltre alla progettazione del data warehouse, sarà necessario scrivere le interrogazioni in SQL esteso e prevedere le viste materializzate ritenute utili per rispondere velocemente alle interrogazioni frequenti.

#### 1. Descrizione del problema

Una società di telefonia mobile è interessata ad analizzare i dati a sua disposizione per fornire un servizio mirato ai suoi clienti e per migliorare la distribuzione delle proprie apparecchiature sul territorio. Attualmente la società telefonica dispone di basi di dati contenenti tutte le informazioni relative alle chiamate effettuate dai suoi utenti. In particolare, per ogni chiamata sono noti il numero di telefono del chiamante, il numero di telefono del chiamato, la durata della telefonata, la tariffa applicata e l'istante di inizio della chiamata (data, ora, minuto, secondi).

La dirigenza della società vuole poter ottenere velocemente delle informazioni sul traffico telefonico effettuato sulle linee telefoniche dell'azienda e sui guadagni effettuati su base giornaliera in funzione della località del chiamante, del giorno e della tariffa.

In particolare alcune delle informazioni a cui i dirigenti sono interessati sono:

- incassi e numero di chiamate effettuate su base mensile in funzione della città nella quale si trova l'apparecchio del chiamante
- incassi e numero di chiamate effettuate su base mensile in funzione della città nella quale si trova l'apparecchio del chiamato
- incassi e numero di chiamate effettuate su base mensile in funzione della provincia e della regione nella quale si trova l'apparecchio del chiamante
- incassi e numero di chiamate effettuate su base mensile in funzione della provincia e della regione nella quale si trova l'apparecchio del chiamato
- incassi e numero di chiamate effettuate in base alla data nella quale si effettua la chiamata e della provincia nella quale si trova l'apparecchio del chiamante
- incassi e numero di chiamate effettuate su base annua in funzione della provincia e della regione nella quale si trova l'apparecchio del chiamante
- incassi e numero di chiamate effettuate su base mensile in funzione della tipologia di tariffa utilizzata
- incassi e numero di chiamate effettuate in funzione del giorno della settimana e della tipologia di tariffa utilizzata
- numero di chiamate effettuate in funzione della data e della regione nella quale si trova l'apparecchio del chiamante
- numero di chiamate effettuate in funzione della data e della regione nella quale si trova l'apparecchio del chiamato

La base di dati (OLTP) della società telefonica in cui vengono memorizzate le singole telefonate è riportata in Figura 1.

#### 2. Progettazione del data warehouse

- Progettare un data warehouse per la gestione delle problematiche illustrate in precedenza (progettazione concettuale e logica).

### 3. Interrogazione del data warehouse

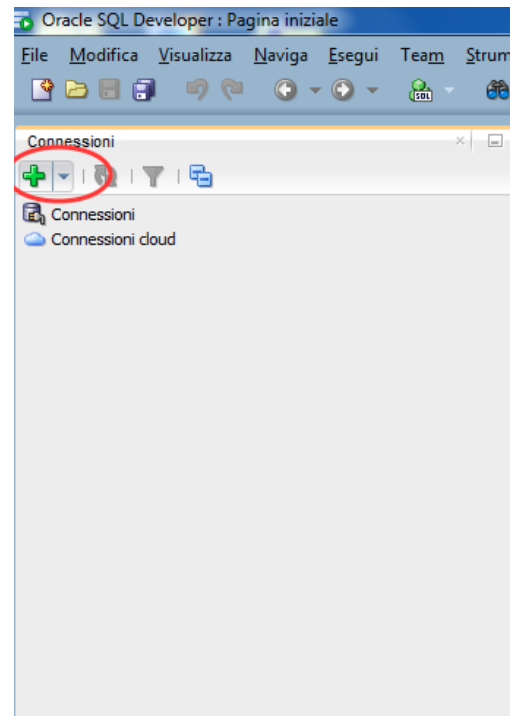
In Figura 2 è riportata una possibile soluzione per il problema proposto. Le tabelle della soluzione proposta sono già state create nel database di Laboratorio e contengono al loro interno dei dati d'esempio. Usare tale base di dati come sorgente per i punti successivi. Si noti l'utilizzo del prefisso DWABD per accedere correttamente ai nomi delle tabelle nel database di Laboratorio.

Fare riferimento al punto 4 per ciò che riguarda la connessione ad oracle e l'esecuzione di script e interrogazioni SQL.

- Risolvere usando il linguaggio SQL (versione estesa) le seguenti interrogazioni:
  1. Selezionare l'incasso totale per ogni tipo tariffa e per ogni anno. Selezionare inoltre l'incasso totale complessivo, l'incasso totale per ogni tipo di tariffa indifferentemente dall'anno, e l'incasso totale per ogni anno indifferentemente dal tipo di tariffa.
  2. Per ogni mese, selezionare il numero di chiamate totali e l'incasso totale. Utilizzando la funzione RANK(), associare ad ogni mese un numero che identifica la posizione del mese all'interno dei mesi in funzione dell'incasso totale effettuato (1 per il mese che ha incassato di più, 2 per il secondo mese, ecc.)
  3. Selezionare per ogni mese dell'anno 2003 il numero di chiamate totali. Utilizzando la funzione RANK(), associare ad ogni mese un numero che identifica la posizione del mese all'interno dei vari mesi dell'anno 2003 in funzione del numero di chiamate totali (1 per il mese con più telefonate, 2 per il secondo mese, ecc.)
  4. Selezionare per ogni data del mese di luglio 2003 l'incasso totale, e la media giornaliera degli incassi delle chiamate effettuate negli ultimi tre giorni.
  5. Selezionare per ogni mese, l'incasso del mese e l'incasso cumulativo dall'inizio dell'anno.
- Basandosi sulle cardinalità delle tabelle del data warehouse (tabelle dei fatti e tabelle delle dimensioni), decidere quali viste materializzate possono essere definite al fine di ridurre i tempi di risposta per le interrogazioni appena risolte.
- Provare a creare alcune delle viste materializzate ritenute utili e verificare se le viste create vengono effettivamente utilizzate dal DBMS analizzando il piano di esecuzione generato per ogni interrogazione.

## 4. Connessione ad Oracle

Aprire il programma Oracle SQL Developer  
Cliccare su crea nuova connessione:



### Login

Per autenticarsi inserire i seguenti parametri.

- Nome utente: bdati[scegliere un valore compreso tra 1-100]
- Password: orac[scegliere un valore compreso tra 1-100]
- Nome host: 130.192.27.4
- Porta: 1521
- SID: xe

Ad esempio, collegandosi dalla macchina numero 23 del laboratorio, usare come username **bdati23** e come password **orac23**.

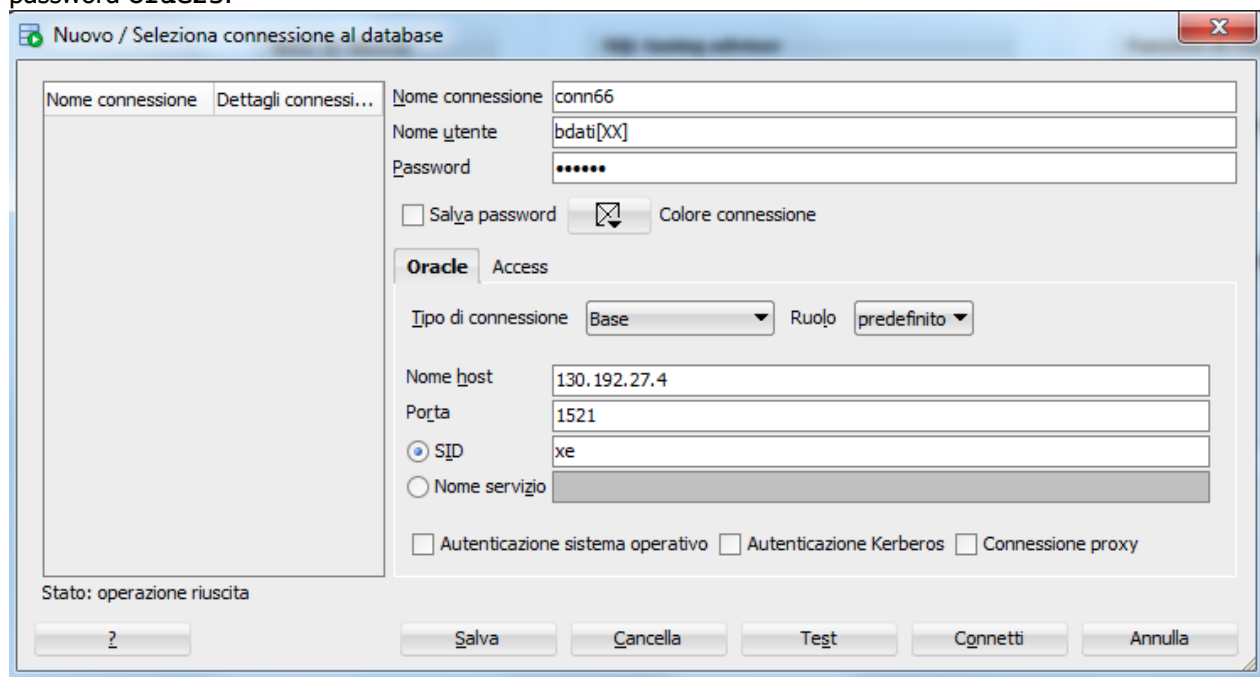


Tabelle	Descrizione
<b>TARIFFE</b> ( TipologiaTariffa INT NOT NULL, NomeTariffa VARCHAR(20) NOT NULL, CostoAlSecondo FLOAT NOT NULL, PRIMARY KEY(TipologiaTariffa) );	Tipologie (categorie) di tariffe esistenti  Cardinalità: 7 tuple
<b>LOCALITA</b> ( CodLocalita INT NOT NULL, Citta VARCHAR(20) NOT NULL, Provincia VARCHAR(20) NOT NULL, Regione VARCHAR(20) NOT NULL, PRIMARY KEY(CodLocalita) );	Località  Cardinalità: 1500 tuple
<b>TELEFONATE</b> ( TelChiamante VARCHAR(20) NOT NULL, TelChiamato VARCHAR(20) NOT NULL, LocalitaChiamante INT NOT NULL, LocalitaChiamato INT NOT NULL, Data DATE NOT NULL, Ora INT NOT NULL, Minuti INT NOT NULL, Secondi INT NOT NULL, DurataTelefonataSecondi FLOAT NOT NULL, TipologiaTariffa INT NOT NULL, PRIMARY KEY(TelChiamante,TelChiamato,Data,Ora,Minuti,Secondi), FOREIGN KEY(TipologiaTariffa) REFERENCES TARIFFE(TipologiaTariffa) , FOREIGN KEY(LocalitaChiamante) REFERENCES LOCALITA(CodLocalita), FOREIGN KEY(LocalitaChiamato) REFERENCES LOCALITA(CodLocalita) );	Telefonate effettuate negli anni 2003, 2004  Cardinalità: 422416 tuple

**Figura 1 – Base di dati sorgente contenente le informazioni sulle singole telefonate**

Tabelle	Descrizione
<b>DWABD.TEMPO</b> (ID_TEMPO            INT            NOT NULL, DATA            DATE           NOT NULL, GIORNO            CHAR(15)      NOT NULL, MESE            INT            NOT NULL, ANNO            INT            NOT NULL, PRIMARY KEY(ID_TEMPO) );	Dimensione tempo  Cardinalità: 30 tuple
<b>DWABD.TARIFFA</b> (ID_TAR            INT            NOT NULL, TIPO_TARIFFA      CHAR(20)      NOT NULL, PRIMARY KEY(ID_TAR) );	Dimensione tariffa  Cardinalità: 7 tuple
<b>DWABD.LUOGO</b> (ID_LUOGO          INT            NOT NULL, CITTA            CHAR(20)      NOT NULL, PROVINCIA        CHAR(20)      NOT NULL, REGIONE          CHAR(20)      NOT NULL, PRIMARY KEY(ID_LUOGO) );	Dimensione luogo (località)  Cardinalità: 1500 tuple
<b>DWABD.FATTI</b> (ID_TEMPO            INT            NOT NULL, ID_TAR            INT            NOT NULL, ID_LUOGO_CHIAMANTE INT            NOT NULL, ID_LUOGO_CHIAMATO INT            NOT NULL, PREZZO            FLOAT           NOT NULL, CHIAMATE          INT            NOT NULL, PRIMARY KEY(ID_TEMPO, ID_TAR, ID_LUOGO_CHIAMANTE, ID_LUOGO_CHIAMATO), FOREIGN KEY(ID_TEMPO) REFERENCES TEMPO(ID_TEMPO), FOREIGN KEY(ID_TAR) REFERENCES TARIFFA(ID_TAR), FOREIGN KEY(ID_LUOGO_CHIAMANTE) REFERENCES LUOGO(ID_LUOGO), FOREIGN KEY(ID_LUOGO_CHIAMATO) REFERENCES LUOGO(ID_LUOGO) );	Tabella dei fatti  Cardinalità: 7809 tuple

Figura 2 –Possibile soluzione del punto 2 - Tabelle del data warehouse