Introduzione alle componente spaziale di Oracle 10g

Claudio Rocchini
Istituto Geografico Militare

febbraio 2008

Indice

1 Introduzione

Questa nota ha lo scopo di introdurre la componente spaziale di Oracle 10g. La lettura di questo documento ha come prerequisito la conoscenza (anche minima) dei seguenti argomenti:

- Basi di Dati;
- linguaggio SQL;
- concetti geometrici della cartografia digitale;
- concetti di base di amministrazione di Oracle;
- conoscenza dell'interfaccia di Oracle 10g (web o SQL*PLUS).

Oracle Spatial 10g é un database spaziale, vale a dire che puó operare in modo nativo con dati cartografici. Altri esempi di db spaziali sono:

- PostgreSQL-Posgis: database Open Source (utilizzo gratuito), OGS compliant, funzioni estese, performance non paragonabili ad Oracle ma utilizzabile (es. Ministero Ambiente USA). Supportato da ArcGis 9.1.
- MySQL Spatial: Open Source (ma non gratuito, richiede in alcuni casi il pagamento per l'utilizzo). Funzionalitá spaziale minima, indicizzazione spaziale minima, non effettivamente utilizzabile.

Oracle é (ora) conforme a OGC e SQL92, anche se la sua struttura di base non si basa su queste specifiche, dato che la prima versione spaziale di Oracle é antecedente alla loro stesura: la conformitá si basa su funzioni di conversione appostite.

Questa documento ha la seguente struttura: per prima cosa descriveremo i concetti di base di Oracle Spatial, quindi faremo alcuni accenni all'amministrazione di un db Oracle, proseguendo con alcune esercitazioni sull'utilizzo effettivo di Oracle Spatial (definizione di feature, importazione di dati *shape*, esecuzione di query spaziali native di Oracle, visualizzazione di dati attraverso software ESRI).

2 Concetti di base di Oracle Spatial

Vediamo adesso quali sono i concetti di base di Oracle Spatial:

- *element*: forma geometrica di base di base;

- *geometry*: componente spaziale completa di un oggetto cartografico;
- *layer*: corrisponde a quello che di solito si chiama *feature class* (es. strade, fiumi, ferrovie);
- *coordinate system*: sistema di riferimento a cui appartengono i dati
- *tolerance*: tolleranza dei dati (es. minima distanza per cui due coordinate si considerano indicanti lo stesso punto).

2.1 Elementi di base

La Fig. 1 mostra i principali elementi geometrici di Oracle10g: tutti i tipi geometrici principali classici sono rappresentati. I toponimi non sono esplicitamente realizzati (scelta comune a

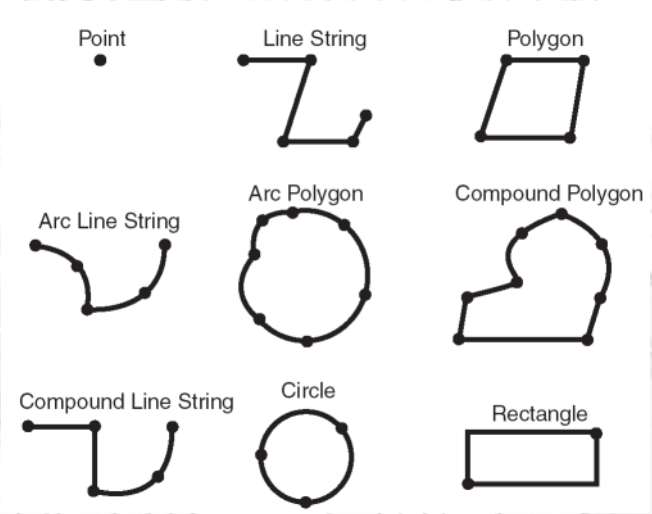


Figura 1: Principali elementi geometrici di base di Oracle.

tutti i RDBMS spaziali ed al formato shape). Oltre ai tipi geometrici classici (punti, linee, aree, etc.) si possono rappresentare punti orientati (utili per toponimi e simboli orientati come ad esempio il simbolo di ponte), direzioni di tangenza, semplici curve coniche etc.

Bisogna dire che Oracle permette di rappresentare veramente tutto: questo comporta una certa complessitá strutturale (come vedremo piú avanti), derivante anche dalla necessitá di mantenere la compatibilitá con le vecchie versioni (fondamentale per un prodotto software commerciale) e dalla volontá (probabilmente dettada da motivi di marketing) di avere una capacitá rappresentativa che supera ogni prodotto analogo.

Per quanto riguarda le dimensioni, tutte le coordinate possono essere rappresentate in 2D, 3D (o 3D piú un valore aggiuntivo, simile al campo *M* degli shape). Quasi tutte le funzioni geometriche si intendono peró *SEMPRE* in modo bidimensionale (es. intersezioni, snap, etc.), fanno eccezione alcuni filtri spaziali, che tengono conto anche della terza dimensione. Anche l'indicizzazione (quindi le ricerche spaziali veloci) si intendono sempre 2D. Inoltre

bisogna dire che molte funzionalitá avanzate (es. supporto OGS, import/export) sono per ora implementate solo in due dimensioni.

2.2 Geometry

La *geometry* é l'insieme di uno o piú elementi di base (es. compound di geometrie), corredato di una serie di dati (es. il sistema di riferimento) che completa la descrizione geometrica di un elemento cartografico. Corrisponde al tipo di dato del database relazionale: questo vuol direc che gli oggetti geometrici avranno una colonna di tipo geometry. In realtá accenneremo anche al fatto che la geoemtry é un *oggetto*.

2.3 Layer

Per definizione il *layer* é un insieme di oggetti che hanno attributi comuni. Purtroppo, come in altri casi, la terminologia Oracle non é standard: noi la chiamiamo Feature Class. Di solito, corrisponde ad una tabella di database (quindi ad una entitá in gergo DB).

Nota: il tipo di dato *geometry* é generico, questo vuol dire che non si distingue (a livello di tipo di dato) fra le geometrie puntuali, lineari o areali: vale a dire che i layer geometrici possono contenere oggetti di tipo geometrico eterogeneo. In altri termini dentro Oracle non esistono feature class solo lineari, solo puntuali o solo areali, ma é sempre possibile creare una feature class mista. Questa scelta puó complicare alcuni aspetti gestionali, anche se a nessun utente é vietato di suddividere i layer per tipo geometrico. Vedremo anche che é possibile imporre dei vincoli aggiuntivi al nostro database in modo da renderlo ordinato.

2.4 Sistema di riferimento

Ad ogni oggetto geometrico (es. una casa, una strada) é possibile associare il suo sistema di riferimento: all'interno di una stessa feature class quindi ci puó essere una casa in Roma40 ed una in WGS84. Un'altra scelta possibile é quella di lasciare non specificato il S.R. dell'oggetto singolo e di specificarlo invece nei metadati della Feature Class (o Layer): in questo modo si assegna un S.R. comune a tutti gli oggetti di una Feature Class (come nel Personal GeoDB di ArcGIS o Geomedia 5).

Per ricapitola le scelte possibili per l'assegnazione dei sistemi di riferimento sono:

- un sistema per singolo oggetto (previsto in Oracle);
- un sistema di riferimento per layer (o Feature class), come per il Personal GeoDB ESRI o Geomedia, possibile in Oracle, anche PostgreSQL utilizza questa scelta. PostGIS utilizza questa scelta.
- un sistema di riferimento per Feature Dataset o Warehouse, come per il Personal GeoDB di Geomedia4.
- Nessun sistema di riferimento specificato (come nei file *shape* base o nei raster tif+TFW).

2.5 Tolleranza

Ad ogni layer (Feature Class) é associata una tollerenza per mezzo dei metadati. Corrisponde al concetto di risoluzione o estensione spaziale dei Personal GeoDB. La tolleranza é usata dalle funzioni geometriche e topologiche per definire la coincidenza di coordinate. Inoltre viene utilizzata durante la costruzione degli indici spaziali.

3 Concetti minimi di amministrazione

Gli aspetti che riguardano l'amministrazione di un database Oracle possono essere molto complessi e sono fuori dagli scopi di questo corso. Tuttavia per introdurre Oracle Spatial é obbligatorio accennare ad alcuni concetti di base (come gli schemi). Si é cercato di ridurre al minimo indispensabile l'introduzione dei concetti di amministrazione, per brevitá di tempo, ma anche perché questi sono strettamente collegati alla particolare versione del software utilizzato (PostgreSQL, Oracle 8,9 o 10).

3.1 Gli schemi di un database

Una base di dati é suddivisa in schemi. Gli schemi sono raggruppamenti logici di oggetti di un database (come tabelle, relazioni, indici, viste). All'interno di due schemi diversi dello stesso database possono esistere oggetti con lo stesso nome (ad esempio due tabelle omonime).

In Oracle ad ogni utente corrisponde il proprio schema (che si chiama come l'utente stesso). In Oracle esistono (oltre a SYS) due utenti/schemi che gestiscono la parte spaziale:

- MDSYS : gestore tabelle spaziali;
- MDDATA : gestore dati geocoding e sistema trasporti.

3.2 Utenti e tabelle

Ogni utente avrá a che fare, oltre che con i propri dati di lavoro, con alcune tabelle globali preesistenti (*SDO_COORD_REF_SYS* e *USER_SDO_GEOM_METADATA*) ed avrá l'impressione di essere l'unico a lavorare di esse. In realtá tutte le tabelle geometriche globali sono viste di tabelle uniche contenute nello schema MDSYS. La struttura della vista nasconde i dati degli altri utenti ed il funzionamento é del tutto trasparente al singolo utente.

4 L'interfaccia SQL

Per quanto possibile, utilizzeremo l'interfaccia web di Oracle, che é molto semplice da usare. Tuttavia questa interfaccia ha alcuni inconvenienti:

- i dati complessi (come gli oggetti) e i dati geometrici non possono essere visualizzati direttamente: l'interfaccia infatti visualizza solo numeri e parole. Questo vuol dire che se il risultato di una query é un oggetto geometrico, l'interfaccia web si rifiuterá di visualizzarlo;

- l'interfaccia web di Oracle 10g notifica come utente corrente *anonymous* invece che l'utente del login (provate la query: SELECT user FROM DUAL): questo malfunzionamento distrugge il sistema spaziale, che si basa su viste personalizzate per ogni utente. Si spera che l'errore venga corretto nelle prossime versioni di Oracle.

Per quanto sopra molte delle operazioni spaziali verranno eseguite nell'interfaccia DOS SQL*PLUS.

4.1 Utilizza dell'interfaccia DOS

Dal mená **Start** di Windows, selezionare **Oracle - Esegui Riga di Comando SQL** : apparirá una finestra DOS (Fig. 2). Per connetersi al db si digita il nome dell'untente (nel

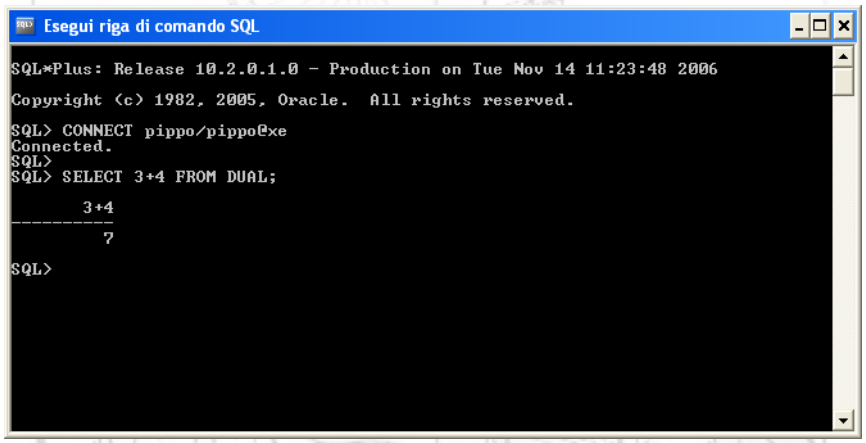


Figura 2: Interfaccia SQL DOS: ci si connette digitando il nome dell'untente, seguito da barra, seguita dalla password, seguita dalla chiocciolina, seguita dal nome del db server, che nel nostro caso é *xe*, quindi si preme *ENTER*.

nostro caso *pippo*), seguito da barra, seguita dalla password (nel nostro caso sempre *pippo*), seguita dalla chiocciolina, seguita dal nome del db server, che nel nostro caso é *xe*, quindi si preme *ENTER* (senza spazi intermedi!). Il nome del database di solito viene scelto in fase di installazione. Nel caso di installazione di Oracle Express, il database si chiama sempre *xe*.

Una volta connessi é possibile eseguire le query come nell'interfaccia web, ricordandosi di premere ENTER ad ogni fine riga. La query viene eseguita appena viene digitato il punto e virgola. Per eseguire le operazioni di copia/incolla in una finestra DOS bisogna cliccare col bottone destro sulla barra della finestra ed operare con l'apposito menú a cascata che appare.

4.2 Migliorare la visibilitá

La finestra DOS é un po' piccola ma si puó ingrandire. Cliccate col bottone destro sulla barra della finestra DOS (la fascia blu in alto alla finestra). Dal menú a discesa selezionate *proprietá* (Fig. 3), si aprirá la finestra delle propietá (Fig. 4). Adesso selezionate il tab (ovvero la pagina) *layout* e digitate 140 come come larghezza dello schermo. Premete OK. A

Figura 3:

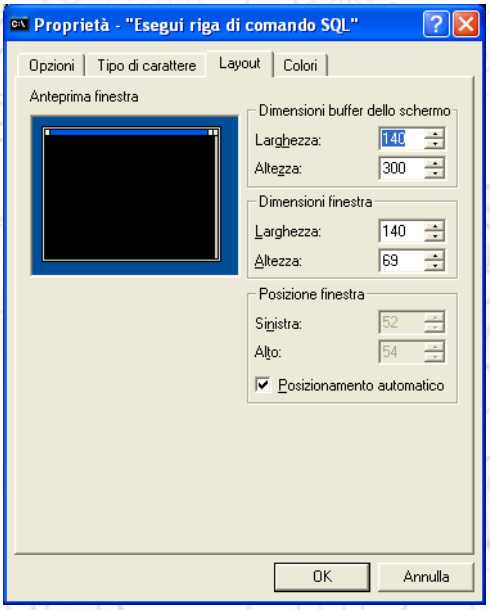


Figura 4:

questo punto cliccate sul bordo della finestra (o sull'angolo in basso) e trascinate il cursore in modo da allargare la finestra DOS fino a quando non scompare la barra di scorrimento orizzontale. Infine comunicate ad Oracle (da solo non se ne accorge) che la finestra ora é larga 140 colonne con il comando:

```
SET LINESIZE 140
```

senza punto e virgola alla fine (non é un comando SQL ma una proprietá della finestra). Adesso il risultato delle query sará molto piú leggibile.

4.3 L'opzione di spool

Un comando molto utile dell'interfaccia SQL DOS é *spool* seguito da un nome di file. Questo comando permette di salvare su di un file di testo tutto quello che compare nella finestra SQL. Scrivete ad esempio il comando (ammesso che esista una cartella work sul disco C):

```
SPOOL c:\work\log.txt
```

In questo modo potete salvarvi tutti i comandi eseguiti ed i loro risultati nel file log.txt, che potrete trattenere per avere una traccia delle esecitazioni eseguite.

4.4 La terza via

Se nel vostro sistema di lavoro avete installato Oracle Enterprise, avete anche una terza opzione, l'utilizzo di SQL*PLUS da Windows. Dal menú *START* di Windows lanciate *SQL*PLUS*: apparirá il dialogo in Fig. 5. Nel dialogo di Login inserire il nome dell'utente, la

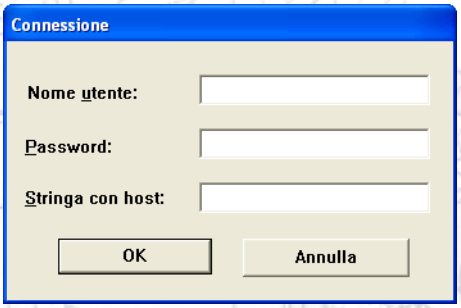


Figura 5: Finestra di Login di SQL*PLUS versione Enterprise: digitate il nome dell'utente, la password ed il nome del database.

password ed il nome del server forniti dal sistemista (nel nostro caso xe). Apparirá la finestra di Fig 6.

5 Creazione di una Feature

Vediamo adesso come sia possibile creare una feature class (o layer) in Oracle Spatial. Tutte le operazioni verranno eseguite con comandi SQL. Si ricorda che, almeno un questa prima fase, é obbligatorio utilizzare l'interfaccia DOS di Oracle (non quella web).

5.1 Creazione di una tabella spaziale

Creiamo adesso la nostra feature spaziale, ovvero la tabella associata, digitiamo quindi:

```
CREATE TABLE edifici
(
    id NUMBER PRIMARY KEY,
    descr VARCHAR2(128),
```

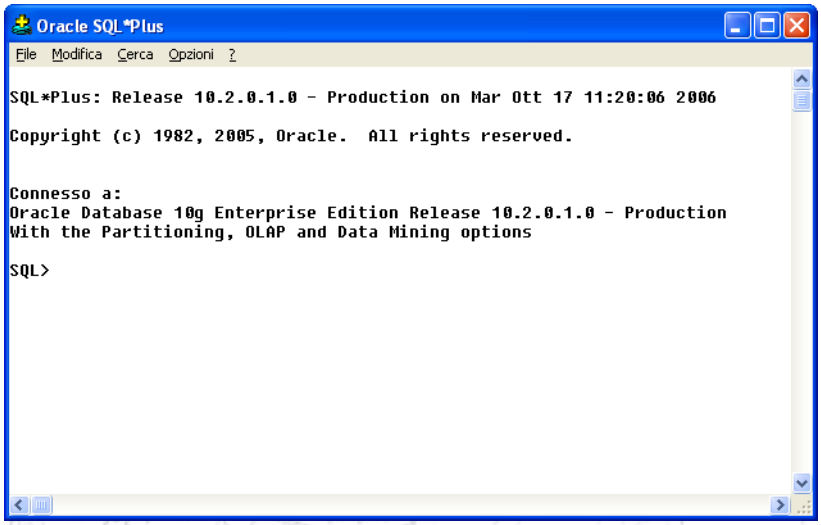


Figura 6: Finestra di SQL*PLUS versione Enterprise: il funzionamento é del tutto simile alla versione DOS.

```
    geom SDO_GEOMETRY
);
```

Dopodiché premiamo invio. Il risultato deve esse il messaggio *Tabella creata*. Analizziamo adesso cosa abbiamo fatto: abbiamo semplicemente creato una tabella di database con tre colonne, che vuole rappresentare gli edifici del nostro sistema cartografico. La prima colonna é numerica e rappresenta il codice dell'edificio: é un numero progressivo che individua univocamente ogni singolo edificio. La specifica *PRIMARY KEY* indica proprio questo fatto. In particolare non ci potranno essere due edifici con lo stesso id-entificativo. La seconda colonna é la descrizione testuale (di massimo 128 caratteri) che accompagna il nostro edificio: é un attributo generico e possiamo anche divertirci ad aggiungerne di altri. La terza colonna é di tipo *SDO_GEOMETRY*, e rappresenta la forma geometrica di ogni edificio. É proprio la presenza di questa terza colonna che rende spaziale la nostra tabella. Vedremo piú avanti in dettaglio che il tipo *SDO_GEOMETRY* é un tipo predefinito di Oracle (come *NUMBER* O *VARCHAR2*, che invece di memorizzare numeri o parole, memorizza i dati geografici degli oggetti del database.

Possiamo adesso controllare la struttura della tabella utilizzando il comando *DESCR*, digitiamo:

```
DESCR edifici;
```

Otteniamo quindi il risultato:

```
 Nome        Nullo?      Tipo
---------- --------    ---------------
ID          NOT NULL    NUMBER
DESCR                   VARCHAR2(128)
GEOM                    SDO_GEOMETRY
```

Vedere che la colonna *GEOM* risulta semplicemente di tipo *SDO_GEOMETRY*.

5.2 Popolamento di una feature

Abbiamo creato la classe degli edifici, ma per adesso la classe (tabella) é vuota. Creiamo quindi il primo edificio, inserendo una riga di dati nella tabella *edifici* con il comando (invero piuttosto oscuro):

```
INSERT INTO edifici VALUES(
 1,
 'chiesa',
  SDO_GEOMETRY(
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,3),
    SDO_ORDINATE_ARRAY(1,1, 3,7)
  )
);
```

Se non ci sono errori, il risultato sará: *Creata 1 riga.* Anche in questo caso analizziamo a posteriori il comando eseguito. I primi due valori 1 e 'chiesa' sono chiari (vanno a riempire le due colonne *id* e *descrizione*), meno chiaro é il terzo valore definito come la costruzione dell'oggetto SDO_GEOMETRY. Vediamo adesso in dettaglio le componenti dell'oggetto SDO_GEOMETRY (che a nostro avviso é leggermente piú complesso di quanto effettivamente necessario...).

6 Struttura di SDO_GEOMETRY

La componente spaziale di Oracle é principalmente formata dall'introduzione del tipo di dato *SDO_GEOMETRY*: questo tipo di dato definisce la forma geometrica di un oggetto. Come i tipi di dato *NUMBER* e *VARCHAR2* memorizzano dati numerici e testuali, cosí *SDO_GEOMETRY* memorizza la componente cartografica.

Il tipo di dato, o megli l'oggetto SDO_GEOMETRY é un tipo composto. In realtá formato di 5 sotto-parti: (come si puó vedere utilizzando il comando *DESCR*), provate ad eseguire il comando *DESCR SDO_GEOMETRY*, il risultato é il seguente:

```
NOME           TIPO
-------------- ------------------------
SDO_GTYPE      NUMBER
SDO_SRID       NUMBER
SDO_POINT      MDSYS.SDO_POINT_TYPE
SDO_ELEM_INFO  MDSYS.SDO_ELEM_INFO_ARRAY
SDO_ORDINATES  MDSYS.SDO_ORDINATE_ARRAY
```

6.1 Definizione di SDO_GTYPE

Il primo campo di *SDO_GEOMETRY* é *SDO_GTYPE*. Questo primo campo definisce il tipo di geometria contenuta nell'oggetto. Il valore contenuto é formato da un numero di quattro cifre nella forma *DLTT*, dove le quattro componenti definiscono:

- D: prima cifra, definisce il numero di dimensioni (2,3 o 4);
- L: seconda cifra, se 0 la geometria é definita da coordinate, altrimenti i valori indicano un sistema di riferimento lineare (es. distanze chilometriche lungo un percorso lineare, utile per memorizzare numeri civici, pietre chilometriche, etc.);
- TT: ultime due cifre, numero a due cifre che indica il tipo di geometria tipo di geometria, secondo lo schema seguente:
  - 00 = sconosciuto
  - 01 = punto
  - 02 = poli linea o curva
  - 03 = poligono
  - 04 = collezione eterogenea
  - 05 = punti multipli
  - 06 = linee multiple
  - 07 = poligoni multipli

Esercizio: nel nostro esempio il campo SDO_GTYPE assume il valore 2003, a cosa corrisponde? Risposta: corrisponde ad un oggetto bidimensionale (prima cifra 2), definito da coordinate (seconda cifra 0) ed il tipo geometrico é un poligono (ultime cifre: codice 03).

6.2 Definizione di SDO_SRID

Il secondo campo é un semplice numero che indica il codice del sistema di riferimento. Anche se il campo é sempre presente, é previsto che il sistema di riferimento possa avere il valore speciale *NULL* (= non definito). In questo primo esempio abbiamo preferito non indicare il sistema di riferimento, dato che le coordinate sono di fantasia (es. 2,7). Per un corretto utilizzo di dati cartografici invece é sempre necessario definire correttamente il codice del sistema di riferimento. Ad esempio, le operazioni geometriche e le ricerche possono essere eseguite solo su oggetto dello stesso sistema di riferimento: se i sistemi sono diversi occorre trasformare a priori le coordinate. Nel progettare una data warehouse cartografica bisogna tener conto che le trasformazioni *on the fly* (vale a dire durante l'utilizzo dei dati e non durante la loro memorizzazione) fra sistemi costano tempo. In un esempio con dati reali (importati da shape), selezioneremo lo SRID corretto.

La tabella *SDO_COORD_REF_SYS*, riporta l'elenco dei sistemi di riferimento supportati da Oracle, con il loro relativo codice numerico. Purtroppo i codici *SRID* non seguono (del tutto) uno standard: ad esempio PostgreSQL ha anche lui la tabella dei sistemi, ma i codici *SRID* sono diversi.

Per avere una descrizione della tabella *SDO_COORD_REF_SYS*, digitate il comando: *DESCR SDO_COORD_REF_SYS*; otterrete:

```
Nome                    Nullo?   Tipo
----------------------- -------- ------------------
SRID                    NOT NULL NUMBER(10)
COORD_REF_SYS_NAME      NOT NULL VARCHAR2(80)
COORD_REF_SYS_KIND      NOT NULL VARCHAR2(24)
COORD_SYS_ID                     NUMBER(10)
DATUM_ID                         NUMBER(10)
GEOG_CRS_DATUM_ID                NUMBER(10)
SOURCE_GEOG_SRID                 NUMBER(10)
PROJECTION_CONV_ID               NUMBER(10)
CMPD_HORIZ_SRID                  NUMBER(10)
CMPD_VERT_SRID                   NUMBER(10)
INFORMATION_SOURCE               VARCHAR2(254)
DATA_SOURCE                      VARCHAR2(40)
IS_LEGACY               NOT NULL VARCHAR2(5)
LEGACY_CODE                      NUMBER(10)
LEGACY_WKTEXT                    VARCHAR2(2046)
LEGACY_CS_BOUNDS                 MDSYS.SDO_GEOMETRY
IS_VALID                         VARCHAR2(5)
SUPPORTS_SDO_GEOMETRY            VARCHAR2(5)
```

Vedete che oltre al primo campo SRID che memorizza il codice del sistema di riferimento, la tabella riporta una serie di dati interessanti.

Non tentate di visualizzare tutta la tabella, é troppo grande (provate invece SELECT count(*) FROM SDO_COORD_REF_SYS, ricordate a cosa serve una query del genere?), piuttosto provate a digitare la seguente query, che visualizza i dati del solo sistema di riferimento numero 3064:

```
SELECT COORD_REF_SYS_NAME,
            COORD_REF_SYS_KIND
FROM SDO_COORD_REF_SYS
WHERE SRID=3064;
```

3064 infatti corrisponde al sistema UTM fuso 32, WGS84, materializzazione particolare eseguita in italia dall'Istituto Geografico Militare e denominata *IGM95*.

6.3 Definizione di SDO_POINT

Il terzo campo di *SDO_GEOMETRY* é *SDO_POINT* e contiene le coordinate di un punto. Si utilizza solo nel caso di geometrie puntuali (vale a dire formate da una sola coordianta): in questo caso il quarto e quinto campo sono nulli e vengoni ignorati. Nel caso di linea od area (come la nostra chiesa) questo campo é nullo, mentre le coordinate sono memorizzate nel quarto e quinto campo. Questa asimmetria di descrizione deriva dalla necessitá di mantenere

la comparabilitá con le vecchie versioni di Oracle Spatial che utilizzavano esclusivamente il campo *SDO_POINT* per memorizzare una singola coordinata. I campi successivi invece sono stati aggiunti nelle versioni di Oracle piú recenti.

6.4 Definizione di SDO_ELEM_INFO e SDO_ORDINATES

Gli ultimi due campi (*SDO_ELEM_INFO* e *SDO_ORDINATES*) sono rispettivamente una lista di numeri interi ed una lista di coordinate. Il primo elemento *SDO_ELEM_INFO* contiene una serie di triplette di numeri che indicano come si debba interpretare la sequenza di coordinate, contenute nel secondo campo *SDO_ORDINATES*.

La varietá di combinazioni di triplette con cui é definita la componente geoemtrica é cosí vasta da non poter essere elencata completamente in questa breve nota. Per un riferimento completo alla descrizione di questi campi si faccia riferimento alla bibliografia. Facciamo solo alcuni esempi: i valori (1,1003,3) indicano che a partire dalla prima coordinata, si descrive il bordo esteriore di un'area rettangolare, specificando solo le coordinate di due vertici opposti (alto-sinitro e basso-destro). Infine l'ultimo campo della struttura *SDO_GEOMETRY* contiene le coordiate geografiche che definiscono l'oggetto cartografico (per prima, come sempre in informatica ed in geometria cartesiana, si specifica la coordinata orizzontale, per seconda quella verticale).

7 Popolamento di una feature

Torniamo adesso al nostro popolamento, ed aggiungiamo altri edifici eseguendo la query:

```
INSERT INTO edifici VALUES(
    2,
    'scuola',
  SDO_GEOMETRY(
        2003,
        NULL,
        NULL,
        SDO_ELEM_INFO_ARRAY(1,1003,1),
        SDO_ORDINATE_ARRAY(5,1, 8,1, 8,6, 5,7, 5,1)
    )
);
```


IL codice 2003 iniziale, indica sempre un poligono piano, mentre la tripletta (1,1003,1) in INFO indica che sempre a partire dalla prima, le coordinate indicano i vertici di bordo di un poligono generico. Si noti che per ottenere un poligono chiuso si deve ripetere le coordinate del primo punto (5,1).

Per concludere, inseriamo adesso l'ultimo edificio:

```
INSERT INTO edifici VALUES(
    3,
    'ospedale',
```

```
  SDO_GEOMETRY(
        2003,
        NULL,
        NULL,
        SDO_ELEM_INFO_ARRAY(1,1003,1, 11,2003,1),
        SDO_ORDINATE_ARRAY(6,4, 10,4, 10,8, 6,8, 6,4,
                   7,5,  9,5,  9,7, 7,7, 7,5 )
    )
);
```

In questo caso, la terna (1,1003,1) indica al solito il bordo esterno di un poligono. La terna (11,2003,1) indica invece che a partire dalla coordinata 11 (ascissa ed ordinata contano per 2), inizia un nuovo bordo ma interno, il bordo quindi di un buco nel poligono (un cortile intero all'edificio). La situazione della nostra warehouse cartografica dovrebbe essere a questo punto quella di Fig. 7.

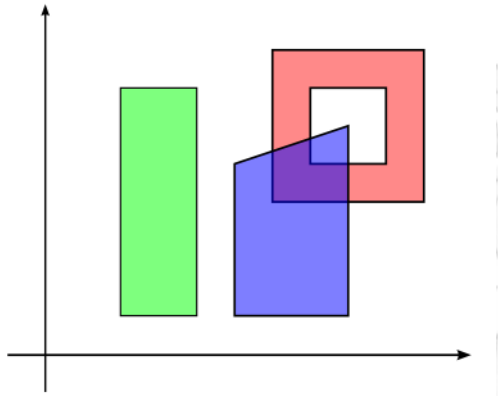


Figura 7: Visualizzazione grafica dei dati geometrici di esempio.

8 Creazione di altre feature

Creiamo adesso la feature punti quota, con la query:

```
CREATE TABLE punti_quota
(
    id NUMBER PRIMARY KEY,
    descr VARCHAR2(128),
    geom SDO_GEOMETRY
);
```

Notare che il tipo della colonna é sempre geometry. Il tipo geometrico é descritto nel contenuto degli oggetti non nella definizione della feature class. Inseriamo alcuni oggetti puntuali nella nostra feature:

```
INSERT INTO punti_quota
VALUES(1,'firenze',
    SDO_GEOMETRY(2001,NULL,
        SDO_POINT_TYPE(1,1, NULL),
        NULL,
        NULL
    ));
```

Alcune note: 2001 indica punto bidimensionale. Notare che gli ultimi due campi sono nulli e le coordinate sono descritte nel terzo campo *SDO_POINT_TYPE*. Quest'ultimo richiede sempre 3 coordinate (anche se l'oggetto é in 2 dimensioni!), quindi poniamo il valore zeta a NULL.

Esercizio: provate ad inserire altri punti quota nella feature omonima oppure provate a creare altri edifici sulla falsa riga degli esempi proposti.

9 Metadati

La creazione della tabella dati con campo geometrico é solo la prima fase della creazione di una feature. Prima di procedere ad un utilizzo effetivo dei dati, é obbligatorio eseguire altre operazioni. Prima di tutto, é obbligatorio specificare una serie di metadati riguardanti le feature create. Il dato fondamentale (su cui si basano le procedure geometriche) é il valore di tolleranza. Inoltre é necessario specificare il sistema di riferimento globale della feature e il rettangolo di ingombro (come nel personal gdb di ArcGis).

La tabella *USER_SDO_GEOM_METADATA* memorizza i metadati di tutte le feature del database. La tabella contiene quattro colonne, al solito provate ad eseguire la query:

```
DESCR user_sdo_geom_metadata;
```

Le colonne che compongono la tabella dei metadati sono TABLE_NAME, COLUMNS_NAME, DIMINFO, SRID. La prima colonna specifica il nome dalla tabella che contiene la feature. La seconda colonna specifica il nome del campo che contiene la componente spaziale (nel nostro caso é *geom*, ma si puó scegliere qualsiasi altro nome), SRID serve per specificare il sistema di riferimento se unico per tutto il Layer (Feature Class). Infine *DIMINFO* é un campo a sua volta composto, anzi addirittura é una sequenza di valori composti.

9.1 Definizione di DIMINFO

Per prima cosa *DIMINFO* é un *vettore*, vale a dire una sequenza di piú valori, con una componente per ogni dimensione: ad esempio per dati in 2 dimensioni DIMINFO ha due componenti, in 3D ne ha 3 o cosí via.

Ogni componente di *DIMINFO* descrive i metadati di una dimensione (ad esempio la prima componenete descrive la latitudine, la seconda la longitudine e cosí via). Ogni descrizione a sua volta é composta da 4 sotto attributi, provate ad eseguire la query (MDSYS.SDO_DIM_ARRAY é il tipo di dato di cui sono composte le componenti di DIMINFO):

```
DESCR MDSYS.SDO_DIM_ARRAY;
```

Le 4 componenti di *DIMINFO* sono:

- Nome coordinata
- Minimo Valore
- Massimo Valore
- Tolleranza (risoluzione dei dati)

Il primo campo definisce il nome della coordinata (es. longitudine) ed é puramente simbolico. Gli altri valori vengono usati per la creazione degli indici spaziale e per tutte le operazioni geometriche, quindi vanno definiti con cura. Il minimo e massimo valore difiniscono la zona cartografata (il rettangolo di ingombro), la tolleranza definisce invece (trascrizione letterale dal manuale di riferimento oracle):

> La massima distanza per cui due punti devono essere considerati lo stesso punto.

In effetti queste definizioni vanno prese con le molle: tanto per seminare dei dubbi provate a pensare il caso in cui la tolleranza é un metro ed io ho 10000 punti in linea, equidistanti un metro. Sono un solo punto? Se si, dove si trova questo punto? Sono due punti? Di piú?

Figura 8:

9.2 Specificare i metadati

Specifichiamo i metadati per gli edifici. Per farlo basta inserire una nuova riga nella tabella dei metadati: definiamo il dominio delle coordinate da -10 a 10 con tolleranza 0.005:

```
INSERT INTO user_sdo_geom_metadata
VALUES (
    'edifici',    -- Tabella Fature
    'geom',       -- Nome campo geometrico
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('lon', -10, 10, 0.005),
        SDO_DIM_ELEMENT('lat', -10, 10, 0.005)
    ),
    NULL        -- SRID
);
```

Quindi definiamo i metadati della feature punti quota (l'unica cosa da cambiare é il nome della tabella):

```
INSERT INTO user_sdo_geom_metadata
VALUES (
    'punti_quota',    -- Tabella Fature
    'geom',            -- Nome campo geometrico
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('lon', -10, 10, 0.005),
        SDO_DIM_ELEMENT('lat', -10, 10, 0.005)
    ),
    NULL        -- SRID
);
```

Per semplificitá abbiamo tralasciato la specifica corretta del sistema di riferimento, inoltre abbiamo battezzato le dimensioni delle nostre coordinate con i nomi *lon* e *lat*.

Ricontrolliamo la tabella dei metadati, visualizzando la relativa tabella:

```
SELECT * FROM user_sdo_geom_metadata;
```

Il risultato é qualcosa del tipo:

```
TABLE_NAME   COLUMN_NAME
-----------  -----------
EDIFICI      GEOM
PUNTI_QUOTA  GEOM

DIMINFO(SDO_DIMNAME, SDO_LB, SDO_UB, SDO_TOLERANCE)
--------------------------------------------------------------------
SDO_DIM_ARRAY(SDO_DIM_ELEMENT('lon',-10,10,.005),
    SDO_DIM_ELEMENT('lat',-10,10,.005))
SDO_DIM_ARRAY(SDO_DIM_ELEMENT('lon',-10,10,.005),
    SDO_DIM_ELEMENT('lat',-10,10,.005))
```

10 Gli indici

Per concludere la creazione delle nostre feature, é necessario creare un indice sulla colonna *geom*. In questo caso sará un indice di tipo spaziale (non convenzionale) e quindi é necessario specificarne il tipo. L'indice spaziale velocizza l'utilizzo delle geometrie: diventa quindi obbligatorio in presenza di grandi moli di dati, inoltre é obbligatorio nel caso si utilizzi ArcGIS per visualizzare i dati.

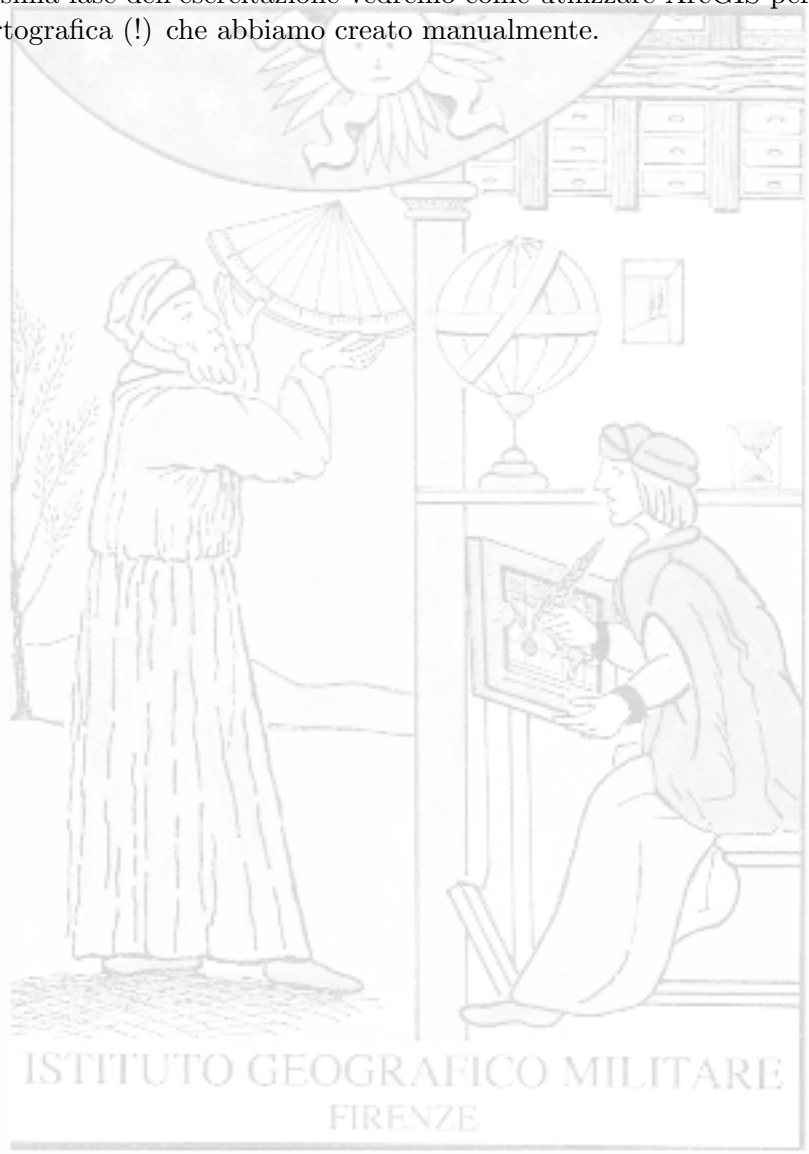
Per creare l'indice sugli edifici digitate il comando:

```
CREATE INDEX edifici_spatial_idx
ON edifici(geom)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

La parola *edifici_spatial_idx* specifica il nome dell'indice che stiamo creando. A differenza degli indici standard (su numeri e parole), é necessario specificare il tipo di indice con la riga *INDEXTYPE IS MDSYS.SPATIAL_INDEX*. Infine creiamo l'indice per i punti:

```
CREATE INDEX punti_quota_spatial_idx
ON punti_quota (geom)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Le nostre feature sono completate! É arrivato il momento di vedere qualcosa di cartografico. Nella prossima fase dell'esercitazione vedremo come utilizzare ArcGIS per visualizzare la warehouse cartografica (!) che abbiamo creato manualmente.



Riferimenti bibliografici

[1] Renzo Sprugnoli, *Libri di base: le basi di dati*, Editori Riuniti (*www.editoririuniti.it*), 1987.

[2] Oracle - Michele Cyran, *Oracle(C) Database:Concepts 10g Release 2 (10.2)*, B14220-02, (*www.oracle.com/pls/db102/homepage*), December 2005.

[3] Oracle - Simon Watt, *Oracle(C) Database:SQL*Plus User's Guide and Reference Release 10.2*, B14357-01, (*www.oracle.com/pls/db102/homepage*), June 2005.

[4] Oracle - Diana Lorentz, *Oracle(C) Database: SQL Reference 10g Release 2 (10.2)*, B14200-02, (*www.oracle.com/pls/db102/homepage*), December 2005.

[5] Oracle - Chuck Murray, *Oracle(C) Spatial: User's Guide and Reference 10g Release 2 (10.2)* , B14255-03, (*www.oracle.com/pls/db102/homepage*), October 2005.

[6] Oracle - Chuck Murray, *Oracle(C) Spatial: Topology and Network Data Models 10g Release 2 (10.2)* , B14256-02, (*www.oracle.com/pls/db102/homepage*), October 2005.

[7] Oracle - Chuck Murray, *Oracle(C) Spatial: GeoRaster 10g Release 2 (10.2)*, B14254-02, (*www.oracle.com/pls/db102/homepage*), October 2005.

[8] Oracle - Chuck Murray, *Oracle(C) Spatial: Resource Description Framework (RDF) 10g Release 2 (10.2)*, B19307-03, (*www.oracle.com/pls/db102/homepage*), October 2005.

[9] PostgreSQL Global Development Group, *PostgreSQL 8.1.0 Documentation*, (*www.postgresql.org/docs/manuals*), 2006.

Elenco delle figure