

# Breve Introduzione alle Componenti Spaziali di Oracle10

Claudio Rocchini  
Istituto Geografico Militare



## Prerequisiti

- Basi di Dati
- Linguaggio SQL
- Concetti geometrici della cartografia digitale
- Concetti di base DBA di Oracle
- SQL\*PLUS



Corso di Aggiornamento Professionale in DB Topografici: Oracle Spatial

2

## Concorrenza e Conformità

- **PostgreSQL – Postgis**: Open Source, OGS compliant, funzioni estese, performance non paragonabili ad Oracle ma utilizzabile (es. Ministero Ambiente USA).
- **MySQL Spatial**: Open Source, funzionalità spaziale minima, indicizzazione spaziale minima, non effettivamente utilizzabile.
- Dove possibile faremo dei collegamenti agli altri dbms, specialmente PostgreSQL.
- Oracle è (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 è antecedente alla loro stesura.



Corso di Aggiornamento Professionale in DB Topografici: Oracle Spatial

3

## Struttura della Presentazione

1. Introduzione ai concetti di base
2. Concetti Minimi di Amministrazione
3. Introduzione Minima a SQL\*Plus
4. Esempio di utilizzo



Corso di Aggiornamento Professionale in DB Topografici: Oracle Spatial

4

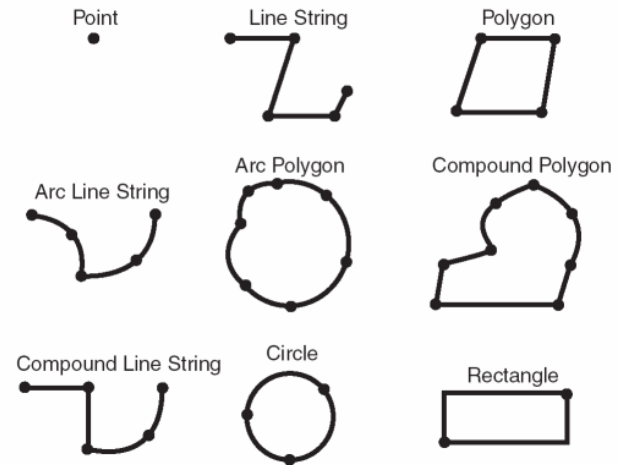
## Parte 1

### Concetti base di Oracle Spatial

- Element (elementi di base)
- Geometry (collezioni di oggetti)
- Layer (che noi chiamiamo Feature Class)
- Coordinate System
- Tolerance



### Elementi di Base



### Note sugli elementi di base

- Toponimi non esplicitamente realizzati (scelta comune a tutti i RDBMS spaziali)
- Si possono anche rappresentare punti orientati, direzioni di tangenza, semplici curve coniche etc.
- Eccessiva complessità strutturale (come vedremo più avanti) che deriva dalla necessità di mantenere la compatibilità con le vecchie versioni, e dalla volontà di poter rappresentare quasi tutto.



### Dimensioni

- Tutte le coordinate possono essere rappresentate in 2 , 3 dimensioni (o 3 + un valore aggiuntivo, simile al campo m degli shape).
- Quasi tutte le funzioni geometriche si intendono però SEMPRE in modo bidimensionale (es. intersezioni, snap, etc.), fanno eccezione alcuni filtri spaziali.
- Anche l'indicizzazione (quindi le ricerche spaziali) si intendono sempre 2D.



## Geometry

- La geometry è l'insieme di uno o più elementi di base (es. compound di geometrie).
- Ad essa è associato un sistema di riferimento.
- Corrisponde al tipo di dato del database relazionale (gli oggetti geometrici avranno una colonna di tipo geometry).



## Layer

- Definizione: un'insieme di oggetti che hanno attributi comuni.
- La terminologia non è standard: noi la chiamiamo Feature Class.
- Di solito, corrisponde ad una tabella di database.
- Nota: il campo geometry è generico, questo vuol dire che non esistono layer puntuali, lineari o areali, ma layer geometrici che possono contenere oggetti di tipo geometrico eterogeneo. Questa scelta può complicare alcuni aspetti gestionali, anche se a nessun utente è vietato di suddividere i layer per tipo geometrico.



## Sistemi di riferimento

- Ad ogni oggetto geometrico (es. una casa, una strada) è possibile associare il suo sistema di riferimento: all'interno di una stessa feature class ci può essere una casa in Roma40 ed una in WGS84. Un'altra scelta possibile è quella di lasciare non specificato il S.R. dell'oggetto singolo e di specificarlo invece nei metadati della Feature Class (o Layer).
- Le scelte possibili per i S.R. sono:
  - Un sistema per singolo oggetto (previsto in Oracle)
  - Un sistema di riferimento per layer (o Feature class), come per il Personal GeoDB (Possibile in Oracle).
  - Un sistema di riferimento per Feature Dataset o Warehouse, come per il Personal GeoDB di Geomedia4.
  - Nessun sistema di riferimento (shape base).
- La concorrenza: PostgreSQL ha optato per la stessa struttura, anche se sconsiglia di mischiare S.R. diversi sulla stessa tabella, e consiglia di aggiungere un *Check* che impone la coerenza del sistema di riferimento.



## Tolleranza

- Ad ogni layer è associata una tolleranza
- Corrisponde al concetto di risoluzione o estensione dei Personal GeoDB.
- La tolleranza è usata dalle funzioni geometriche e topologiche per definire la coincidenza di coordinate.



## Parte 2: Concetti Minimi di Amministrazione

- Gli schemi di sistema
- Nota: questa parte è volutamente ridotta al minimo, perché:
  - Riguarda i particolari di basso livello, che non comportano l'apprendimento di nessun concetto;
  - Varia molto al variare del DBMS utilizzato, ma soprattutto dalla versione di Oracle.



## Gli schemi di Sistema

- Una base di dati è suddivisa in schemi
- Oracle ha due utenti/schemi di sistema che gestiscono
- MDSYS – gestore tabelle spaziali
- MDDATA – gestore dati geocoding e sistema trasporti



## Utenti e Tabelle

- Ogni utente avrà a che fare con alcune tabelle globali (SDO\_COORD\_REF\_SYS e USER\_SDO\_GEOM\_METADATA) ed avrà l'impressione di essere l'unico a lavorare di esse.
- In realtà tutte le tabelle geometriche globali sono viste materializzate di tabelle uniche contenute nello schema MDSYS.
- La struttura della vista nasconde i dati degli altri utenti: il funzionamento è del tutto trasparente.



## Nota sull'interfaccia

- Sebbene l'interfaccia Web sia molto più semplice da utilizzare, questa ha due inconvenienti:
  - I dati complessi oggetto e geometrici non possono essere visualizzati direttamente: l'interfaccia visualizza solo numeri e parole
  - L'interfaccia notifica come utente corrente "anonymous" invece che l'utente del login (provate: SELECT user FROM DUAL): questo distrugge il sistema spaziale, che si basa su viste personalizzate per ogni utente.
- Per quanto sopra tutte le operazioni spaziali verranno eseguite nell'interfaccia DOS



## Un'altra possibilità: DOS (1)

- Oltre all'interfaccia "web" è presente anche un'interfaccia DOS
- Dal menù start, selezionare Oracle > Esegui Riga di Comando SQL
- Apparirà una finestra DOS



## DOS (2)

- Connettersi digitando utente/password@xe e premendo enter
- A questo punto è possibile eseguire comandi SQL

```
Esegui riga di comando SQL
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Nov 14 11:23:48 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
SQL> CONNECT pippo/pippo@xe
Connected.
SQL>
SQL> SELECT 3+4 FROM DUAL;
-----
3+4
   7
SQL>
```



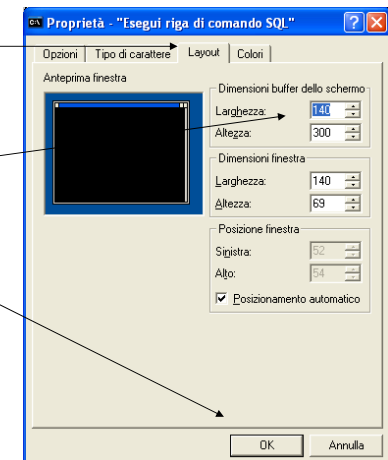
## DOS: migliorare la vista (1)

- La finestra DOS è un po' piccola ma si può ingrandire.
- Cliccate col bottone destro sulla barra della finestra DOS
- Dal menù selezionate "proprietà"



## DOS: migliorare la vista (2)

- Selezionate il tab "layout" (pagina)
- Quindi digitate 140 come larghezza dello schermo
- Premete OK
- Allargate il bordo della finestra DOS fino a quando non scompare la barra di scorrimento orizzontale.



## DOS: migliorare la vista (3)

- Infine comunicate ad Oracle che la finestra ora è larga 140 colonne con il comando (da solo non se ne accorge):

```
SET LINESIZE 140
```

- Senza ; alla fine
- Adesso il risultato delle query sarà molto più visibile.



## Nota importante su SqlPlus

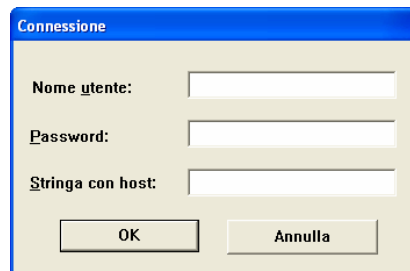
- Un comando molto utile dell'interfaccia SQL dos è spool seguito da un nome di file. Questo comando permette di salvare su di un file testo tutto quello che compare nella finestra SQL.
- Scrivete ad esempio:  

```
SPOOL c:\work\log.txt
```
- In questo modo potete salvarvi tutti i comandi eseguiti ed i loro risultati nel file log.txt



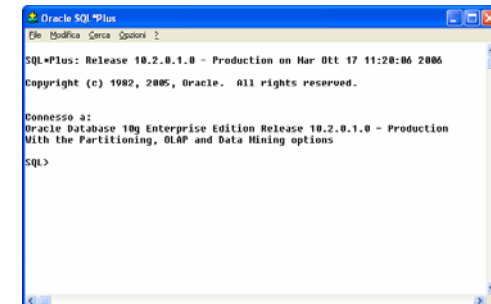
## Introduzione a SQL\*Plus (Oracle Enterprise)

- Dal menù Programmi – Oracle Lanciare SQL\*Plus
- Apparirà il seguente dialogo



## Connessione con SQL\*Plus

- Inserire il nome dell'utente, la password ed il nome del server forniti dal sistemista (nel nostro caso xe).
- Apparirà la seguente finestra:



## Utilizzo di SQL\*Plus

- Nella finestra di SQL\*Plus è possibile digitare comandi SQL e visualizzarne il risultato.
- Proviamo a scrivere la seguente query:  

```
SELECT 3+4 FROM DUAL;
```
- Per concludere premiamo INVIO
- Il risultato dovrebbe essere 7.
- (Dual è una tabella fittizia che serve per rispettare la sintassi di SELECT, il quale si aspetta sempre almeno una tabella).



## Creazione di una Feature

- Creiamo adesso la nostra feature spaziale, digitiamo:  

```
CREATE TABLE edifici  
(  
  id NUMBER PRIMARY KEY,  
  descr VARCHAR2(128),  
  geom SDO_GEOMETRY  
);
```
- Dopodichè premiamo invio. Il risultato deve esse “Tabella creata”.



## Note su Creazione Feature

- Abbiamo creato una tabella di database con tre colonne.
- La prima colonna è numerica e rappresenta il codice dell’edificio (chiave primaria).
- La seconda colonna è la descrizione testuale (di massimo 128 caratteri).
- La terza colonna è di tipo SDO\_GEOMETRY, e rappresenta la forma geometrica di ogni edificio.



## Controllo della Tabella

- Possiamo controllare la struttura della tabella utilizzando il comando descr, digitiamo:

```
DESCR edifici;
```

- Otteniamo quindi:

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



## Popoliamo la Feature

- Adesso creiamo il primo edificio, inserendo una riga nella tabella edifici:

```
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)
  )
);
```

- Il risultato sarà: "Creata 1 riga".



## Note sul popolamento

- I primi due valori 1 e 'chiesa' sono chiari (codice e descrizione), meno chiaro è il terzo valore definito come la costruzione dell'oggetto SDO\_GEOMETRY.
- Vediamo in dettaglio le componenti dell'oggetto SDO\_GEOMETRY (che a nostro avviso è leggermente più complesso di quanto effettivamente necessario...).



## Struttura di SDO\_GEOMETRY

- L'oggetto SDO\_GEOMETRY è formato di 5 parti (come si può vedere utilizzando il comando DESCR):

<i>NOME</i>	<i>TIPO</i>
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



## SDO\_GEOMETRY: SDO\_GTYPE

- Il primo campo definisce il tipo di geometria contenuta nell'oggetto.
- E' un numero di 4 cifre nella forma **DLTT**:
  - **D**: dimensioni (2,3 o 4)
  - **L**: 0=coordinate, altrimenti il valore indicano un sistema di riferimento lineare (es. distanze chilometriche)
  - **TT**: numero a due cifre che indica il tipo di geometria tipo di geometria, secondo la tabella seguente.





## SDO\_GEOMETRY: SDO\_GTYPE

### Codici dei tipi geometrici:

- 00 = sconosciuto
- 01 = punto
- 02 = line o curva
- 03 = poligono
- 04 = collezione eterogenea
- 05 = punti multipli
- 06 = linee multiple
- 07 = poligoni multipli
- Esercizio: nel nostro esempio SDO\_GTYPE era 2003, a cosa corrisponde?



## SDO\_GEOMETRY:

### SDO\_SRID(1)

- Il secondo campo è un numero che indica il codice del sistema di riferimento.
- Anche se il campo è sempre presente, è 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).
- 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 Datawarehouse bisogna tener conto che le trasformazioni on the fly fra sistemi costano tempo.



## SDO\_GEOMETRY:

### SDO\_SRID(2)

- In un esempio con dati reali, selezioneremo lo SRID corretto.
- La tabella SDO\_COORD\_REF\_SYS, riporta l'elenco dei sistemi di riferimento supportati da Oracle.
- Purtroppo i codici SRID non seguono uno standard, ad esempio PostgreSQL ha anche lui la tabella dei sistemi, ma i codici SRID sono diversi.

## SDO\_GEOMETRY:

### SDO\_SRID(3)

- Per avere una descrizione della tabella, digitate il comando: `DESCR SDO_COORD_REF_SYS;`  
Otterrete:

Nome	Null?	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)



## SDO\_GEOMETRY: SDO\_SRID(4)

- Non tentate di visualizzare tutta la tabella, è troppo grande, piuttosto provate a digitare la seguente query:  

```
SELECT COORD_REF_SYS_NAME,  
       COORD_REF_SYS_KIND  
FROM SDO_COORD_REF_SYS  
WHERE SRID=3064;
```
- 3064 corrisponde a UTM fuso 32, WGS84, materializzazione IGM95.



## SDO\_GEOMETRY: SDO\_POINT

- Il terzo campo contiene le coordinate di un punto
- Si utilizza solo nel caso di geometrie puntuali: in questo caso il quarto e quinto campo sono nulli.
- Nel caso di linea od area (come la nostra chiesa) questo campo è nullo, mentre le coordinate sono memorizzate nel quarto e quinto campo.
- Questa asimmetria di descrizione deriva dalla necessità di mantenere la comparabilità con le vecchie versioni di Oracle Spatial.



## SDO\_GEOMETRY: SDO\_ELEM\_INFO & SDO\_ORDINATES

- Gli ultimi due campi sono rispettivamente una lista di numeri interi ed una lista di coordinate.
- SDO\_ELEM\_INFO contiene una serie di triplette di numeri che indicano come si debba interpretare la sequenza di coordinate, contenute in SDO\_ORDINATES.



## SDO\_GEOMETRY: SDO\_ELEM\_INFO

- La varietà di combinazioni di triplette è così vasta da non poter essere elencata completamente.
- I valori del nostro esempio (1,1003,3) indicano che a partire dalla prima coordinata, si descrive il bordo esterno di un'area rettangolare specificando solo le coordinate di due vertici opposti.



## Popolamento (Continua 2)

- Torniamo adesso al nostro popolamento, ed aggiungiamo altri edifici:

```
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)
  )
);
```

- 2003 è sempre un poligono, mentre la tripletta (1,1003,1) indica che sempre a partire dalla prima, le coordinate indicano i vertici di bordo di un poligono generico. Si noti che di deve ripetere il primo punto per chiudere l'area.



## Popolamento (Continua 3)

- Inseriamo adesso l'ultimo edificio:

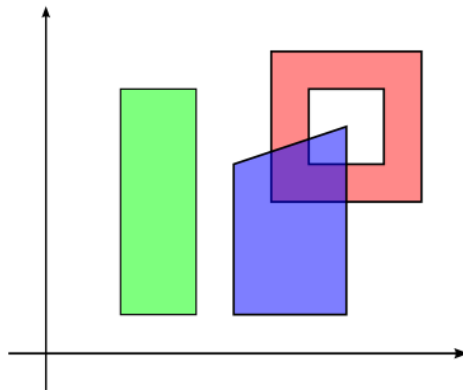
```
INSERT INTO edifici VALUES(
  3,
  'ospedale',
  SDO_GEOMETRY(
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1, 11,2003,1),
    SDO_ORDINATE_ARRAY(6,4, 10,4, 10,8, 6,8, 6,4,
    7,5, 9,5, 9,7, 7,7, 7,5 )
  )
);
```

- La terna (1,1003,1) indica al solito il bordo il bordo esterno di un poligono. La terna (11,2003,1) indica che a partire dalla coordinata 11, inizia un nuovo bordo, quindi un buco nel poligono.



## Situazione Edifici

- La situazione degli edifici assomiglia alla seguente:



## Creazione di una nuova Feature

- Creiamo adesso la feature punti quota:

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

- Notare che il tipo della colonna e' sempre geometry. Il tipo geometrico è descritto nel contenuto degli oggetti.



## Popolamento dei punti

- Inseriamo alcuni punti:

```
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, quindi poniamo la zeta a NULL.
- Esercizio: provate ad inserire altri punti quota.



## Metadati Obbligatori

- La creazione della tabella dati con campo geometrico, è solo la prima fase della creazione di una feature.
- Prima di procedere con la creazione degli indici e con l'utilizzo del db, occorre specificare una serie di metadati obbligatori.
- La tabella USER\_SDO\_GEOM\_METADATA memorizza tali dati.
- Il dato fondamentale (su cui si basano le procedure geometriche) è il valore di tolleranza.



## Struttura Metadati

- La tabella user\_sdo\_geom\_metadata contiene i metadati di tutte le colonne geometriche dello schema.
- La tabella contiene quattro colonne, provate ad eseguire la query:

```
DESCR user_sdo_geom_metadata;
```

- Le colonne sono TABLE\_NAME, COLUMNS\_NAME, DIMINFO, SRID
- Le due prima colonne sono ovvie, SRID serve per specificare il sistema di riferimento se unico per tutto il Layer (Feature Class).
- DIMINFO è invece a sua volta un oggetto composto, anzi un vettore di oggetti composti ...



## Metadati: DIMINFO (1)

- Per prima cosa DIMINFO è un vettore, con una componente per ogni dimensione: in 2D DIMINFO ha due componenti, in 3D ne ha 3 o così via.
- Ogni componente di DIMINFO descrive i metadati di una dimensione, ed è composta da 4 sotto attributi, provate ad eseguire la query:

```
DESCR MDSYS.SDO_DIM_ARRAY;
```



## Metadati: DIMINFO (2)

- Le componenti di DIMINFO sono:
  - Nome coordinata
  - Minimo Valore
  - Massimo Valore
  - Tolleranza (risoluzione dei dati)
- Il nome è puramente simbolico.
- Gli altri valori vengono usati per la creazione degli indici e per tutte le query spaziali



## Un inciso sulla tolleranza

- Nel manuale di Oracle la tolleranza è definita letteralmente come:  
*“La distanza per cui due punti devono essere considerati lo stesso punto”.*
- In effetti queste definizioni vanno prese con le molle: se la tolleranza è un metro ed io ho 10000 punti in linea, equidistanti un metro, sono un solo punto? Due? Dove si trova questo punto?



## Metadati: edifici

- Specifichiamo i metadati per gli edifici: 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
);
```



## Metadati: Punti Quota

- L'unica cosa da cambiare è 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
);
```



## Controllo dei Metadati

- Ricontrolliamo la tabella dei metadati, visualizzando la tabella:

```
SELECT * FROM user_sdo_geom_metadata;
```

- Il risultato è 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))
```



## Creazione Indici

- Per concludere la creazione delle nostre feature, sarà necessario creare almeno un indice sulla colonna geom.
- In questo caso sarà un indice di tipo spaziale (non convenzionale) e quindi sarà necessario specificarne il tipo.
- L'indice spaziale velocizza l'utilizzo delle geometrie ed è obbligatorio nel caso si utilizzi ArcGIS per visualizzare i dati.



## Creazione Indici

- Per creare l'indice sugli edifici digitate il comando:

```
CREATE INDEX edifici_spatial_idx
ON edifici(geom)
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;
```



## Vediamo Qualcosa

- E' 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.



# Oracle Spatial + ArcGIS

## Introduzione

Claudio Rocchini  
Istituto Geografico Militare

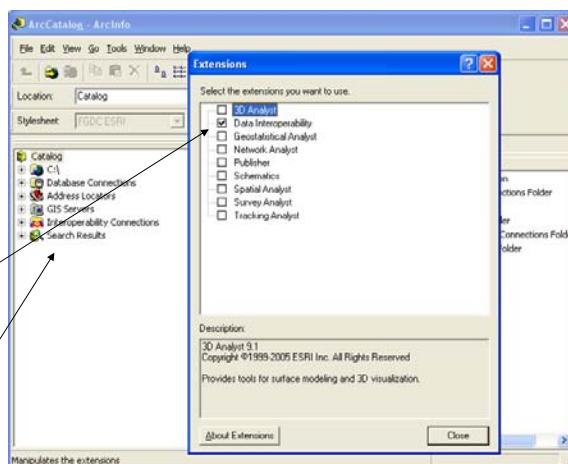
## Introduzione

- Ci sono (almeno) tre modi con cui ArcGIS si può connettere ad Oracle Spatial
  - Con una connessione OLE DB (possibile con ArcGIS 9.0), solo a colonne dati, senza interpretazione della parte spaziale (ma è possibile il geocoding)
  - Connessione tramite Interoperability Extension (possibile con ArcGIS 9.1, che noi useremo): dati spaziali in sola lettura
  - Connessione tramite ArcSDE: dati spaziali in lettura scrittura (necessita la configurazione di un server SDE).



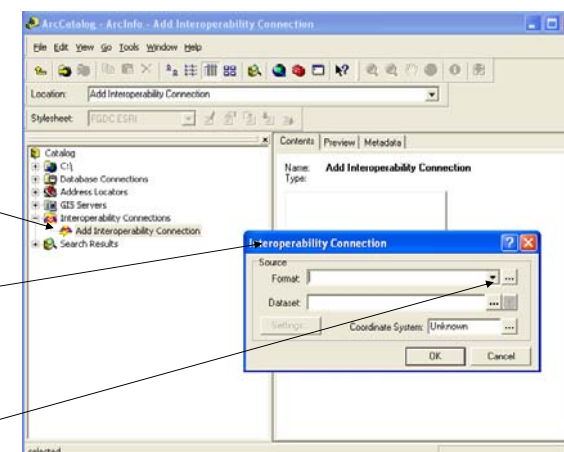
## Preparazione di ArcCatalog

- Per prima cosa attiviamo l'estensione Interoperability
- Lanciare ArcCatalog
- Selezionare Tools > Extensions
- Attivare Data Interoperability
- Nel catalogo appare il bottone Interoperability connections



## Creazione della Connessione

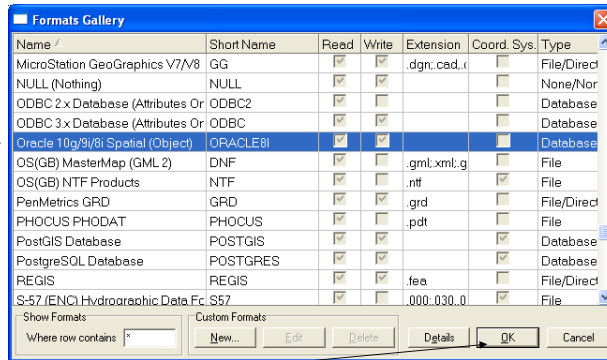
- Aprire Interoperability connections e cliccare su Add...
- Appare il dialogo della connessione
- Scegliere il formato cliccando sul bottone ...





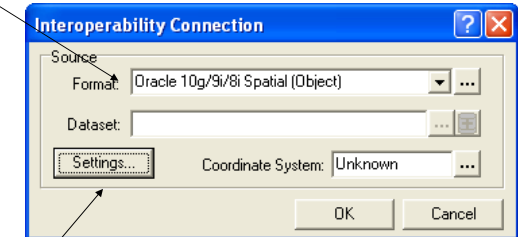
## Scelta del Formato

- Appare il dialogo dei formati
- Cercare Oracle e selezionare la riga corrispondente
- Premere il bottone OK



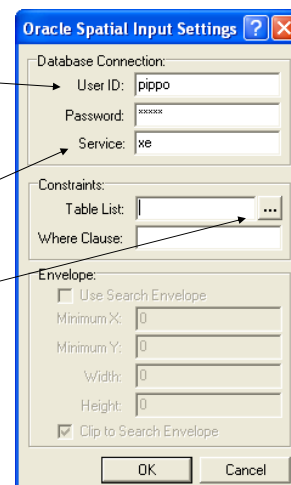
## Settaggi di Connessione

- Tornando al dialogo di connessione appare il formato selezionato
- Adesso bisogna settare i parametri di connessione
- Premiamo quindi il bottone Settings...



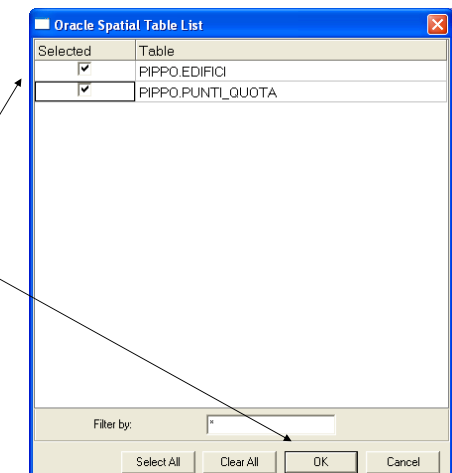
## Parametri di Connessione

- Digitare il nome dell'utente Oracle e la password
- Cos'è il service? Non ne abbiamo parlato: riguarda la configurazione di Oracle.
- In Oracle Express il service si chiama xe
- Cliccate sulla table list per far apparire la lista di feature presenti
- Se la lista non compare abbiamo sbagliato un parametro di connessione.



## Scelta delle feature

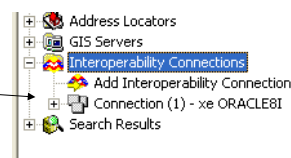
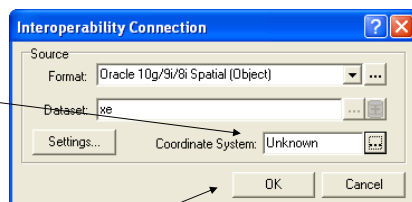
- Selezionate la lista delle feature da inserire nella connessione.
- Ceccate le feature che si desidera inserire.
- Premete il tasto OK
- E poi di nuovo OK nel dialogo dei parametri





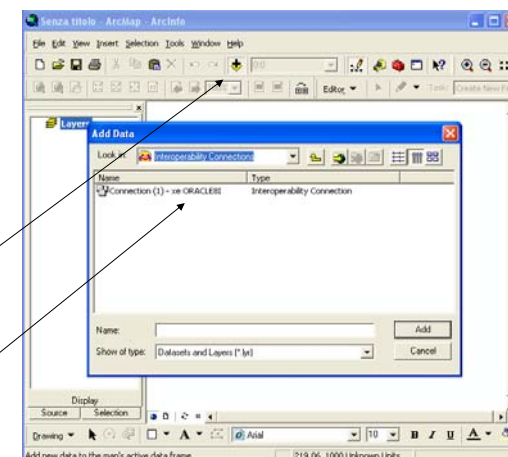
## Sistema di Riferimento

- Abbiamo quasi finito
- Bisognerebbe impostare il sistema di riferimento
- Questo perché Interoperability non è capace di recuperare il sistema di riferimento di Oracle
- In questo caso le geometrie hanno coordinate di fantasia (10,20) e non lo setteremo
- Una volta premuto il tasto OK, appare la connessione nella lista di ArcCatalog
- Nel prossimo esempio (DB25) setteremo invece il S.R. corretto.



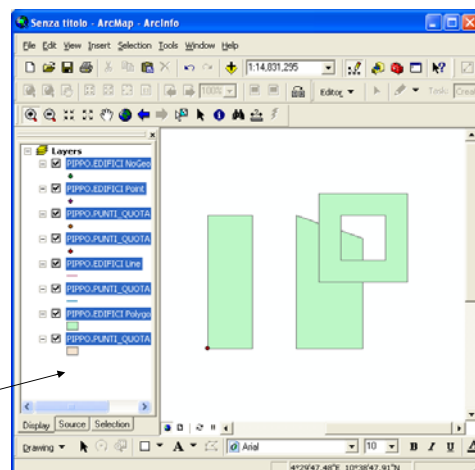
## ArcMap

- Adesso possiamo anche chiudere ArcCatalog.
- Lanciamo ArcMap
- Clicchiamo su aggiungi tema e selezioniamo la connessione appena creata.



## Visualizzazione dei dati

- Dopo che ArcGIS si è lamentato dell'assenza di sistema di riferimento, le feature sono inserite.
- La situazione è la seguente:



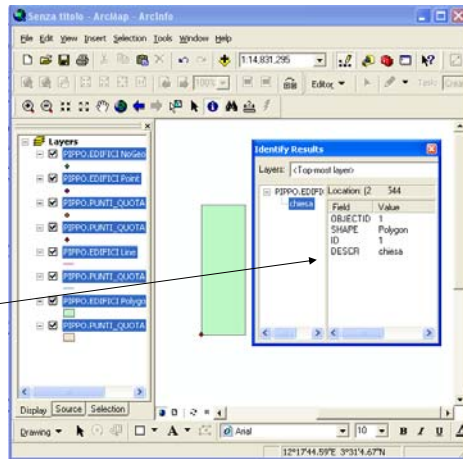
## Nota sulle feature

- Le feature class di Oracle non hanno tipo geometrico, nel senso che all'interno di una feature possono convivere, mescolati fra di loro, punti linee ed aree.
- Questo contrasta con la struttura di ArcMap.
- Per ovviare a questa divergenza ArcMap suddivide ogni feature di Oracle in tre feature separate: una che contiene gli oggetti puntuali, una lineari ed una areali.
- Ovviamente alcune (molte) di queste feature saranno vuote, se noi siamo stati ordinati nel suddividere i dati per tipo geometrico.



## Dati Associati

- Ovviamente, oltre alla geometria, è possibile visualizzare i dati associati agli oggetti nel database.



## Esempi sui dati

- Le operazioni fatte fino ad adesso servivano per avere un'idea della struttura di base di Oracle.
- Per poter passare a degli esempi di utilizzo reale, dobbiamo però avere in mano dei dati più interessanti.
- Vedremo ora di seguito come si possono importare dei dati cartografici pre-esistenti in formato Shape.



## Importazione di Dati in Oracle

Claudio Rocchini  
Istituto Geografico Militare



## Importazione di Dati

- Ci sono molti modi di importare dati pre-esistenti in Oracle.
- Ad esempio attraverso ArcSDE è possibile scrivere i dati direttamente da ArcGIS.
- Nel nostro esempio invece utilizzeremo uno strumento fornito da Oracle stesso, che permette di importare dati in formato Shape.
- I dati di esempio sono un piccolo estratto di un foglio DB25 dell'Istituto Geografico Militare.



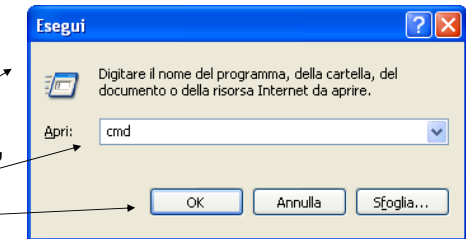
## Preparare i file shape

- Copiare i dati (file shape) in una cartella con il nome comodo (es. c:\work).
- Il nome deve essere “comodo” (vale a dire corto e senza spazi), perché lavoreremo dalla finestra DOS.
- Le feature class che useremo sono le seguenti:
  - PAQ040 ponti
  - LAP030 strade
  - LBH030 fossi
  - AAL015 edifici
  - ABH135 risaie
  - AEA010 culture generiche



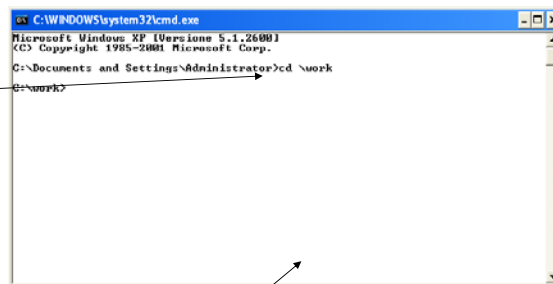
## Lanciare il DOS

- Lanciare una finestra DOS (se trovate il bottone...)
- Altrimenti premete Start > Esegui
- Nel dialogo che appare digitate “cmd”
- Premete il tasto OK
- Appairà la finestra DOS



## Andare nella cartella giusta

- Nella finestra DOS digitate il comando:  
`cd \work`
- (con la barra rovescita all’inizio) per andare nella cartella di lavoro.
- Per questo il nome deve essere semplice!



Nota: ho invertito i colori della finestra DOS (da nero a bianco) per non finire tutto l'inchiostro del Centro di Geotecnologie



## Preparazione al caricamento

- Se alcune delle impostazioni di Windows sono in italiano, si può incorrere in problemi di caricamento.
- In questo caso è necessario scrivere nella finestra DOS il seguente comando:  
`SET NLS_LANG=AMERICAN_AMERICA`
- Questo per cercare di nascondere la colpa di essere italiani.



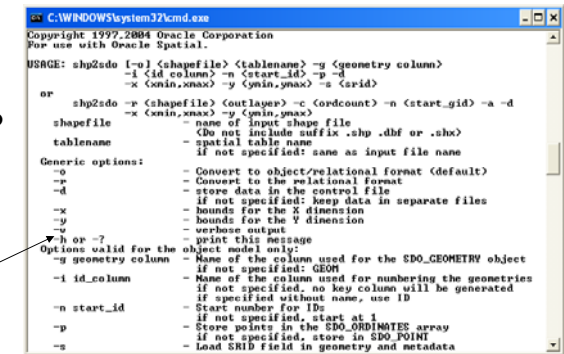
# Introduzione a Shp2sdo

- Questo semplice comando DOS fornito da Oracle, trasforma un file shp in una coppia di file testo:
  - Il primo file contiene i comandi sql per la creazione della feature class. I comandi sono del tutto simili a quelli che abbiamo scritto a mano nel nostro esempio.
  - Il secondo file contiene i dati in un formato leggibile dal caricatore di dati di Oracle



# Test di shp2sdo

- Provate a digitare il comando:  
`shp2sdo -help`
- Apparirà un breve manuale di utilizzo del comando



```
C:\WINDOWS\system32\cmd.exe
Copyright 1997, 2004 Oracle Corporation
For use with Oracle Spatial.

USAGE: shp2sdo [-o] <shapefile> <tablename> -g <geometry column>
              -i <id column> -n <start_id> -p -j
              -x <xmin,ymax> -y <ymin,ymax> -s <sruid>

or
shp2sdo -r <shapefile> <outlayer> -c <ordcount> -n <start_gid> -a -d
shapefile -x <xmin,ymax> -y <ymin,ymax>
shapefile -x <xmin,ymax> -y <ymin,ymax>
          - name of input shape file
          <Do not include suffix .shp .dbf or .shx>
tablename - spatial table name
          if not specified: same as input file name

Generic options:
-o          - Convert to object/relational format (default)
-p          - Convert to the relational format
-d          - Store data in the control file
          if not specified: keep data in separate files
-x          - bounds for the X dimension
-y          - bounds for the Y dimension
-t          - verbose output
-v          - print this message
-h or -?   - Options valid for the object model only:
-g geometry column - Name of the column used for the SDO_GEOMETRY object
                  if not specified: GEOM
-i id_column  - Name of the column used for numbering the geometries
                  if not specified, no key column will be generated
                  if specified without name, use ID
-n start_id  - Start number for IDs
                  if not specified, start at 1
-p          - Store points in the SDO_ORDINATES array
          if not specified, store in SDO_POINT
-s          - Load SRID field in geometry and metadata
```



# Parametri del comando

- La sintassi del comando è:  
`shp2sdo file_shape nome_feature [parametri]`
- I parametri principali sono:
  - -i <nome del campo chiave principale>
  - -g <nome campo geometrico>
  - -d : dati inseriti in un file separato
  - -x (minima x, massima x)
  - -y (minima y, massima y)
  - -t <risoluzione piana>
  - -s <codice sistema di riferimento oracle>
  - -f <utilizza alta precisione nelle coordinate>
  - -v : modo loquace (verbose)



# Rilevazione ingombro

- Per poter inserire i dati in Oracle è obbligatorio specificare l'ingombro della geometria
- Questo può essere rilevato con ArcGis
- Nel nostro caso la x va da 507,000 a 513,000, la y da 5004,000 a 5010,000.



## Costruzione del comando

- A questo punto digitiamo il comando per la trasformazione della feature PAQ040, il comando sarà:

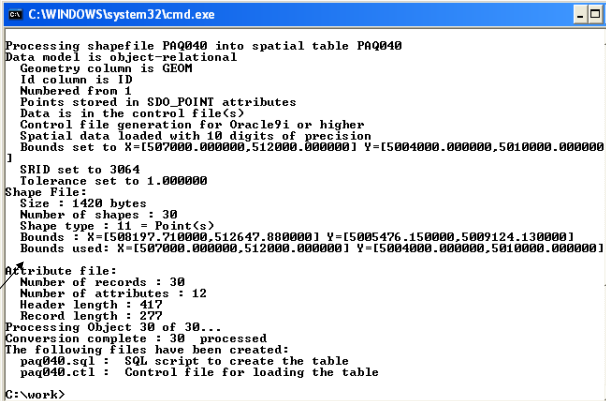
```
shp2sdo PAQ040 paq040 -i gid -g geom -d -  
x (507000,512000) -y (5004000,5010000) -  
t 1 -s 3064 -f -v
```

- Si ricorda che il sistema di riferimento giusto è il numero 3064.
- Proviamo quindi ad eseguire il comando.



## Esecuzione di shp2sdo

- Il software mostra un report dell'operazione eseguita
- Proviamo a leggerlo
- Viene anche calcolato il bbox reale



```
C:\WINDOWS\system32\cmd.exe  
Processing shapefile PAQ040 into spatial table PAQ040  
Data model is object-relational  
Geometry column is GEOM  
Id column is ID  
Numbered from 1  
Points stored in SDO_POINT attributes  
Data is in the control file(s)  
Control file generation for Oracle9i or higher  
Spatial data loaded with 10 digits of precision  
Bounds set to X=1507000.000000,512000.000000 Y=15004000.000000,5010000.000000  
1  
SRID set to 3064  
Tolerance set to 1.000000  
Shape File:  
Size: 1420 bytes  
Number of shapes: 30  
Shape type: 11 = Point(s)  
Bounds: X=[500197.710000,512647.880000] Y=[5005476.150000,5009124.130000]  
Bounds used: X=[507000.000000,512000.000000] Y=[5004000.000000,5010000.000000]  
Attribute file:  
Number of records: 30  
Number of attributes: 12  
Header length: 417  
Record length: 277  
Processing Object 30 of 30...  
Conversion complete: 30 processed  
The following files have been created:  
paq040.sql: SQL script to create the table  
paq040.ct1: Control file for loading the table  
C:\work>
```



## Analisi dei risultati

- Il primo file prodotto è un file SQL per la creazione della feature, provate ad aprirlo con il notepad:

```
DROP TABLE PAQ040;  
CREATE TABLE PAQ040 (  
  ID          NUMBER(38)  
  PRIMARY KEY,  
  info        VARCHAR2(100),  nam      VARCHAR2(80),  
  ao1         NUMBER,  ltn      NUMBER,  
  ubd         NUMBER,  wd1      NUMBER,  
  mcc         VARCHAR2(3),  bsc      VARCHAR2(3),  
  tuc         VARCHAR2(3),  lab      VARCHAR2(8),  
  id          NUMBER,  GMRotation  NUMBER,  
  GEOM        MDSYS.SDO_GEOMETRY);
```



## Analisi dei risultati (2)

- Il file prosegue con la definizione dei metadati:

```
DELETE FROM USER_SDO_GEOM_METADATA  
WHERE TABLE_NAME = 'PAQ040' AND COLUMN_NAME = 'GEOM' ;  
  
INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME,  
  COLUMN_NAME, DIMINFO, SRID)  
VALUES ('PAQ040', 'GEOM',  
  MDSYS.SDO_DIM_ARRAY  
  (MDSYS.SDO_DIM_ELEMENT('X', 507000.000000000,  
  512000.000000000, 1.000000000),  
  MDSYS.SDO_DIM_ELEMENT('Y', 5004000.000000000,  
  5010000.000000000, 1.000000000)  
  , MDSYS.SDO_DIM_ELEMENT('Z', 0, 1000000,  
  1.000000000)) ,  
  3064);
```



## Analisi dei risultati (3)

- Il secondo file prodotto contiene i dati, provate ad aprirlo con notepad:

```
1|Ponte in muratura o conglomerato cementizio armato per
autostrada (A1) , con
luc|UNK|331.638900000|4|0.000000000|200|021|014|002|P350A|59|-
28.361060000|
3001|3064|508293.140000000|5006330.010000000|86.210000000|
2|Ponte in muratura o conglomerato cementizio armato per
autostrada (A1) , con
luc|UNK|316.065800000|4|0.000000000|200|021|014|002|P350A|60|-
43.934230000|
3001|3064|508907.550000000|5005759.130000000|84.430000000|
3|Ponte, viadotto in muratura o conglomerato cementizio armato
per strade di
categ|UNK|19.804900000|1|0.000000000|25|000|014|004|P351G|68|19
.804860000|
3001|3064|510911.310000000|5008467.060000001|83.350000000|
```

...



## Esecuzione dei file prodotti

- Per eseguire il file sql utilizzeremo un altro comando DOS di Oracle: sqlplus.
- Questo comando è quello che lanciamo per aprire la finestra SQL DOS dei nostri esercizi
- Lo stesso comando può essere utilizzato per lanciare un file SQL precostituito.
- Eseguite quindi il comando:  
`sqlplus pippo/pippo@xe @paq040.sql`
- Come vedere è necessario specificare i parametri di connessione ed il file da eseguire preceduto da una @.
- Alla fine dell'esecuzione scrivete il comando quit per uscire dalla finestra sql.



## Esecuzione di SQL

- Se tutto è andato bene, il risultato deve essere qualcosa del genere.
- Ricordatevi di scrivere quit per uscire da SQL.

```
C:\WINDOWS\system32\cmd.exe
Refer to the SQL*Plus User's Guide and Reference for more information.
C:\work>sqlplus pippo/pippo@xe @paq040.sql
SQL*Plus: Release 10.2.0.1.0 - Production on Wed Nov 15 11:24:07 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

Table dropped.
Table created.
1 row deleted.
1 row created.
Commit complete.
SQL> quit
Disconnected from Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
C:\work>
```



## Caricamento Dati

- Infine per il caricamento dati utilizziamo un terzo programma Oracle: sqlldr (loader).
- La sintassi è uguale a quella di sqlplus:  
`sqlldr pippo/pippo@xe paq040.ct1`
- Se tutto va bene i dati sono caricati e il programma mostra un messaggio del tipo: *commit point reached*.
- Nota: non abbiamo mai parlato di commit e transazioni, questo argomento esula dagli scopi del corso.



## Le altre feature

- A questo punto possiamo procedere al caricamento delle altre feature.
- L'unica cosa che bisogna cambiare nei comandi sono i nomi dei file ed il nome della feature class.
- Ad esempio il prossimo comando di conversione sarà:

```
shp2sdo LBH030 LBH030 -i id -g geom -d -x  
(507000,512000) -y (5004000,5010000) -t 1 -s  
3064 -f -v
```

- Provate adesso a caricare tutte le altre feature



## Un passo manuale: gli indici

- Shp2sdo non crea nessun indice.
- Sarà necessario quindi necessario creare gli indici manualmente.
- Colleghiamoci di nuovo con l'interfaccia sql DOS.
- Lanciamo il programma "Esegui riga di comando SQL"



## Indici (2)

- Ci connettiamo con il comando:  
`connect pippo/pippo@xe`

```
Esegui riga di comando SQL  
SQL*Plus: Release 10.2.0.1.0 - Production on Wed Nov 15 13:24:08 2006  
Copyright (c) 1982, 2005, Oracle. All rights reserved.  
SQL> connect pippo/pippo@xe  
Connected.  
SQL> _
```



## Indici (3)

- A questo punto possiamo creare gli indici manualmente come sappiamo fare con i comandi:  
`CREATE INDEX aa1015_sp_idx  
ON aa1015(geom)  
INDEXTYPE IS MDSYS.SPATIAL_INDEX;`
- E poi  
`CREATE INDEX abh135_sp_idx  
ON abh135(geom)  
INDEXTYPE IS MDSYS.SPATIAL_INDEX;`
- E così via per le altre feature (cambia solo il nome dell'indice e il nome della tabella).





# Visualizzazione con ArcGis

- Per visualizzare i dati importati con ArcGis si procede come già sappiamo.
- Questa volta però impostiamo correttamente il sistema di riferimento

