



Introduzione a Oracle

Prof. Matteo Golfarelli

Alma Mater Studiorum - Università di Bologna

Per approfondimenti:

➤ ORACLE 11g Rel. 2 - Concepts

I DBMS (richiami)

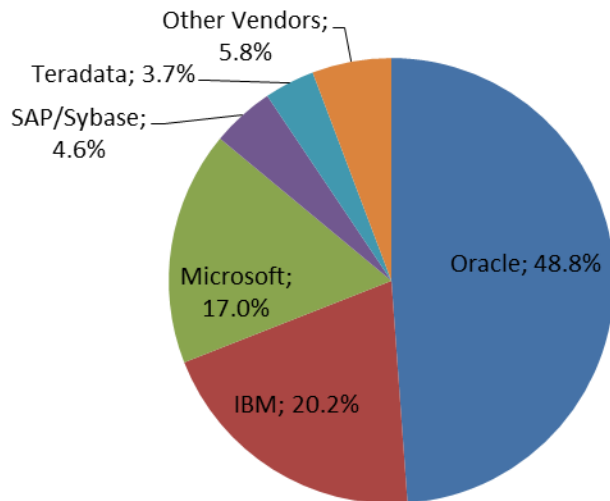
- ❑ Un *DBMS (Data Base Management System)* è un sistema software in grado di gestire efficientemente le informazioni necessarie a un sistema informativo, rappresentandone i dati in forma integrata e garantendone la persistenza
 - La gestione di grandi quantità di dati richiede particolare attenzione ai problemi di efficienza
 - La persistenza e la condivisione richiedono che un DBMS fornisca meccanismi per garantire l'affidabilità dei dati (*fault tolerance*), per il controllo degli accessi e per il controllo della concorrenza
 - Un modello dei dati consente agli utenti di disporre di un'astrazione di alto livello attraverso cui interagire con il DB
 - Diverse altre funzionalità vengono messe a disposizione per motivi di efficacia, ovvero per semplificare la descrizione delle informazioni, lo sviluppo delle applicazioni, l'amministrazione del DB, ecc.
- ❑ Un *RDBMS* è un DBMS che offre come modello logico dei dati il modello relazionale

Oracle

- ❑ Oracle è il DBMS più diffuso sul mercato

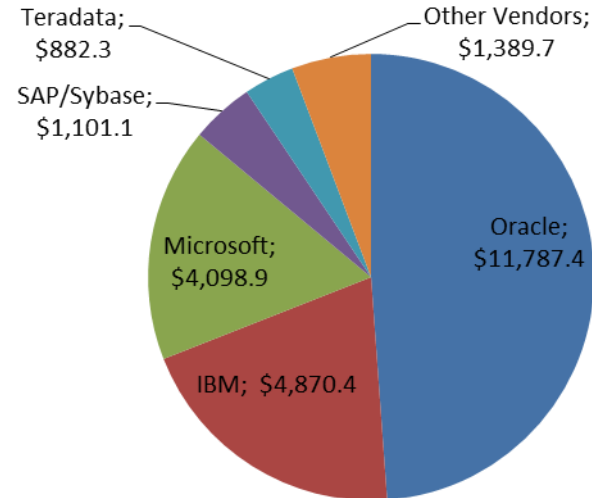
Market Share 2011

(fonte Gartner Group)



Market Value 2011 (in milioni di dollari)

(fonte Gartner Group)



- ❑ Attorno al DBMS è cresciuta negli anni una complessa architettura per la gestione dei dati non solo relazionali
- ❑ Suddetta architettura rappresenta la base tecnologica per lo sviluppo di applicazioni nell'ambito dei sistemi informativi (es. ERP, CRM, BI)

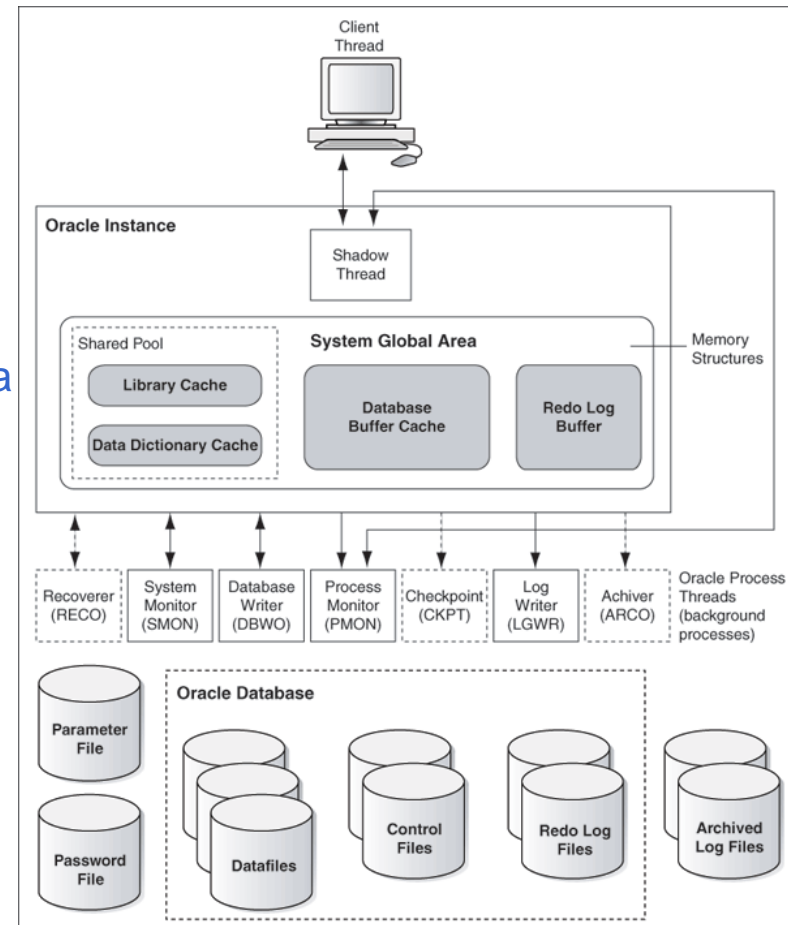
Un po' di storia

- ❑ 1979 prima release (Larry Ellison, Bob Miner, Ed Oates)
 - SQL language e modello relazionale
- ❑ 1999 Oracle 8i
 - Supporto nativo ai protocolli internet e supporto server-side per Java
- ❑ 2001 Oracle 9i
 - Più istanze possono accedere a un singolo database
 - Gestione nativa di dati XML
- ❑ 2003 Oracle 10g
 - Supporto al grid computing
- ❑ 2007 Oracle 11g
 - Maggiore flessibilità nella modifica degli schemi
- ❑ 2013 Oracle 12c
 - ❑ Supporto per il cloud computing
- ❑ 2013 Oracle difende la Coppa America



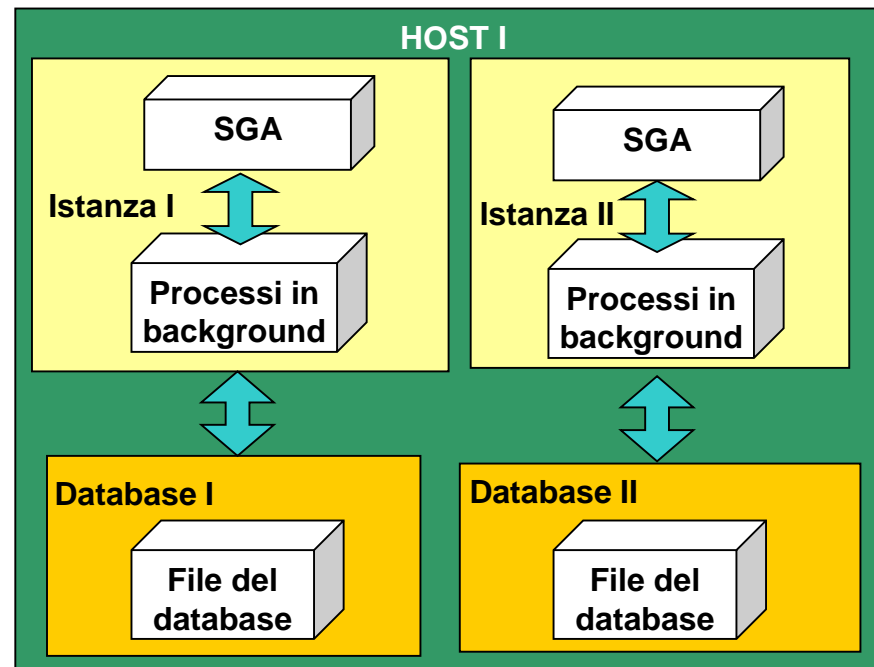
Architettura I

- ❑ Un **server** Oracle 11g è formato da un'istanza e da un database.
 - Il **database** è composto da un insieme di file in cui vengono memorizzati i dati. I diversi tipi di file memorizzano diversi tipi di informazioni
 - L'**istanza**, è costituita da una serie di strutture in memoria e di processi eseguiti in background che accedono ai file del database. I diversi processi svolgono funzioni diverse e permettono di realizzare le funzionalità avanzate che differenziano i DBMS dai file system.
 - Per ogni connessione utente all'istanza viene creato un processo separato dotato di una propria area di memoria denominata PGA – Program Global Area



Architettura II

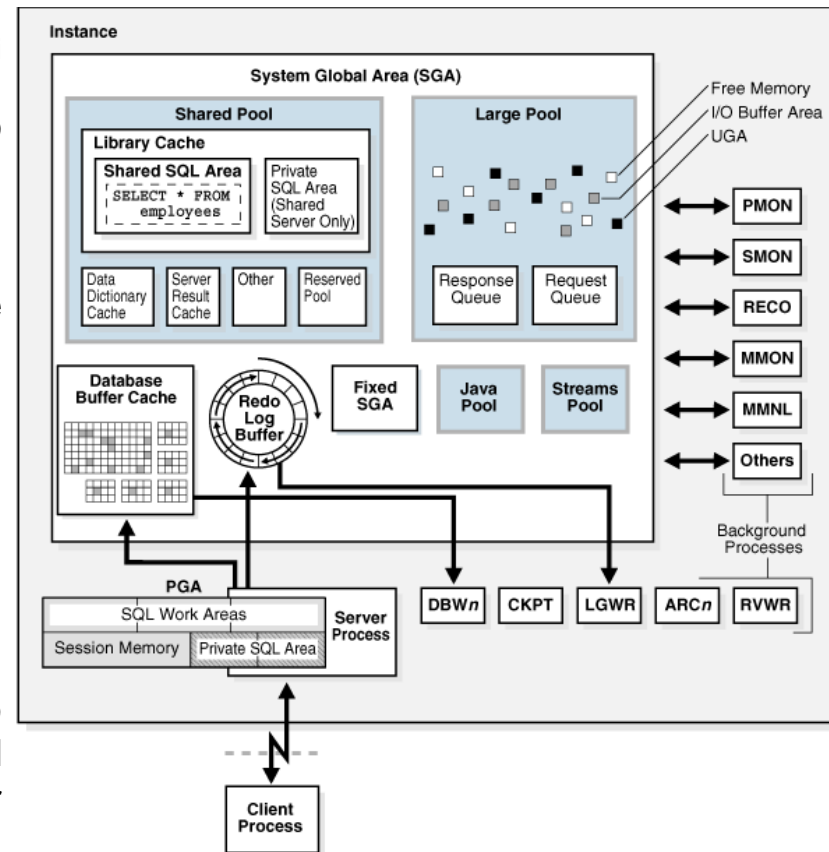
- Una macchina fisica (host) può ospitare più server Oracle
 - Le due istanze saranno percepite come completamente indipendenti e potranno comunicare mediante i medesimi protocolli utilizzati da istanze residenti su host separati.



Oracle Instance: memory I

- ❑ La System Global Area è la principale area di memoria dell'istanza. Essa contiene un insieme di strutture di memorizzazione condivise da tutti i processi del server

- **Database buffer cache:** contiene copia dei blocchi dati letti (frequentemente) dai data files (vedi funzioni avanzate DBMS).
- **Redo Log buffer:** è un buffer circolare necessario a mantenere la coerenza del database in caso di errori (vedi funzioni avanzate DBMS).
- **Fixed SGA:** contiene informazioni di servizio quali, per esempio, lo stato del database e delle istanze a cui i processi di background devono accedere.
- **Shared Pool:** contiene diversi tipi di informazioni tra cui:
 - Shared SQL Area
 - Parsed SQL query
 - Codice PL/SQL
 - Parametri di sistema
 - Porzioni del data dictionary
- **Large pool:** area di memoria opzionale che può contenere blocchi di informazioni della Shared pool la cui dimensione supera quella prevista per la Shared pool stessa



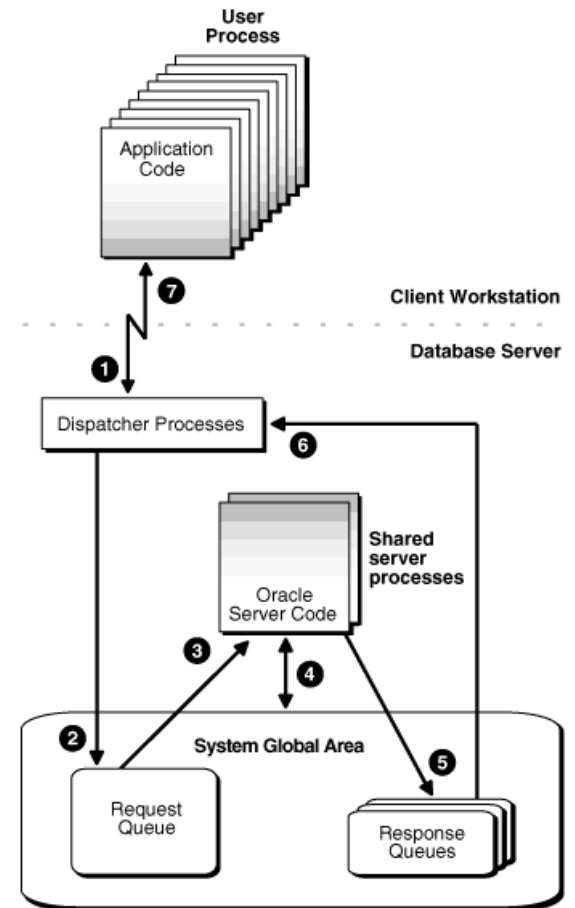
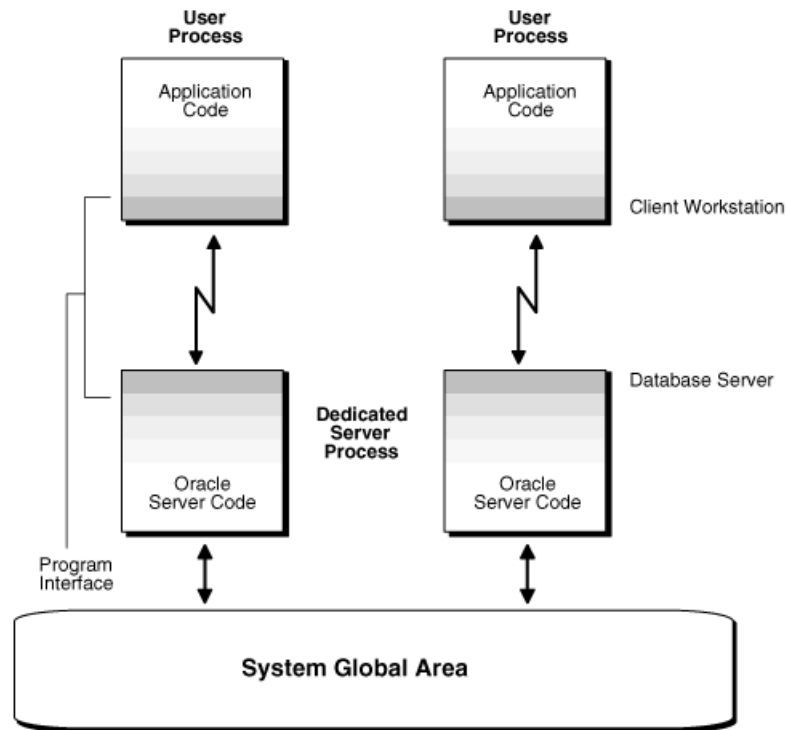
Oracle Instance: memory II

- ❑ La Program Global Area è l'area di memoria allocata a uno specifico processo. Le principali informazioni memorizzate sono:
 - **Private SQL Area:** contiene informazioni specifiche del processo circa le query e i programmi eseguiti : bind variable, stato dell'esecuzione
 - **SQL Work Area:** spazi di memoria necessari all'esecuzione della query
 - Sort area: spazio di memoria utilizzato dagli algoritmi di sort (vedi sort-merge Z-vie)
 - Hash area (vedi ottimizzazione performance)
 - Bitmap Merge area (vedi ottimizzazione performance)

Oracle Instance: process I

- ❑ Un'istanza contiene o interagisce con i seguenti tipi di processi:
 - **Processi client:** eseguono le applicazioni utente. Sono esterni all'istanza, non possono leggere l'SGA, possono essere eseguiti su host diversi da quello dell'istanza
 - **Processi Oracle:** eseguono il codice dell'istanza stessa, possono leggere l'SGA e devono essere eseguiti sull'host dell'istanza
 - **Processi di background:** sono inizializzati assieme all'istanza ed eseguono operazioni di servizio quali recovery, lettura e scrittura dei buffer, ecc.
 - **Processi server:** eseguono le operazioni richieste dagli utenti quali query SQL e codice PL/SQL.
- ❑ A seconda del tipo di connessione i processi server possono servire uno o più processi client
 - **Dedicated Server Process:** ogni processo server dialoga con uno e un solo processo client
 - **Shared Server Process:** le operazioni di molti processi client sono eseguite da un numero limitato di processi server. Le richieste sono inserite in una coda e sono quindi assegnate, da un processo dispatcher, al primo processo server disponibile.
 - Le connessioni shared assicurano una migliore gestione delle risorse (memoria) e di conseguenza permettono a un numero maggiore di processi client di connettersi contemporaneamente all'istanza

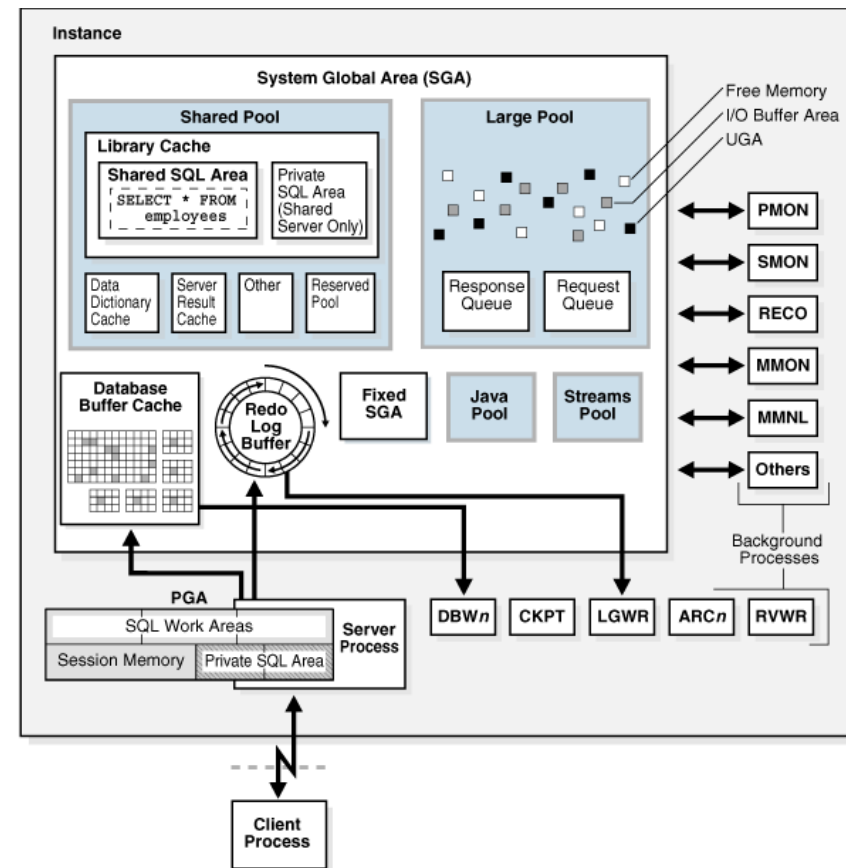
Oracle Instance: process II



Oracle Instance: process III

❑ I principali processi di background sono:

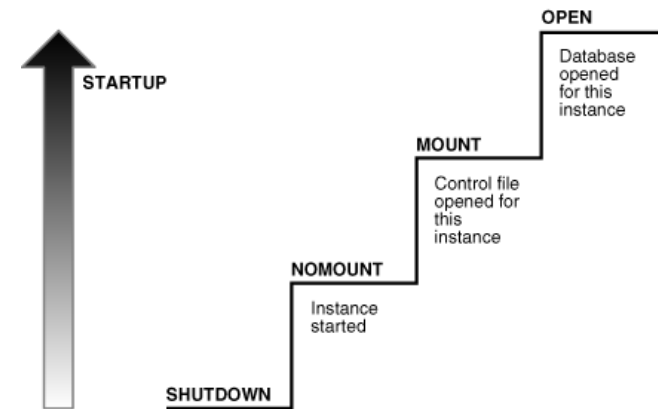
- **Process Monitor (PMON):** libera le risorse occupate dagli utenti quando un loro processo fallisce
- **System Monitor (SMON):** all'avvio dell'istanza esegue il ripristino dell'istanza mediante il registro di REDO. Al processo è inoltre demandato il compito di deframmentare le aree di disco occupate dal database. (vedi funzioni avanzata DBMS).
- **Database Writer (DBWn):** gestisce la cache dei blocchi e del dizionario. Più processi possono essere istanziati al fine di accelerare questa operazione.
- **Log Writer (LGWR):** gestisce la scrittura sul file in linea del registro di REDO del relativo buffer in memoria
- **Check point (CKPT):** si occupa di inserire nei file dati e nel dizionario dati i checkpoint: marcatori che garantiscono la sicurezza degli aggiornamenti effettuati prima di un dato istante. In questo modo il ripristino di un database sarà necessario solo per le operazioni eseguite dopo l'ultimo checkpoint.



Stati di start-up

□ Un database Oracle può essere in uno dei seguenti stati:

- **Shutdown**
- **Nomount:** i processi di background e l'SGA sono allocati ma non sono connessi a nessun DB.
- **Mount:** l'istanza è associata a un database tramite la lettura dei suoi control file. Il server è accessibile solo agli amministratori per operazioni di manutenzione
- **Open:** il server è accessibile a tutti gli utenti



Configurazione DB - Instance

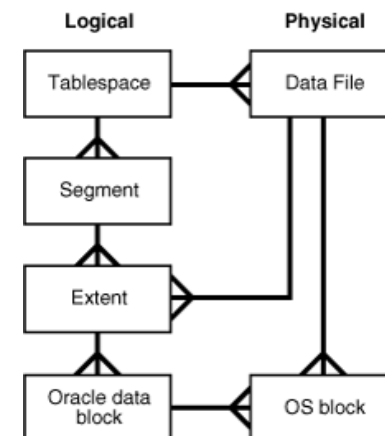
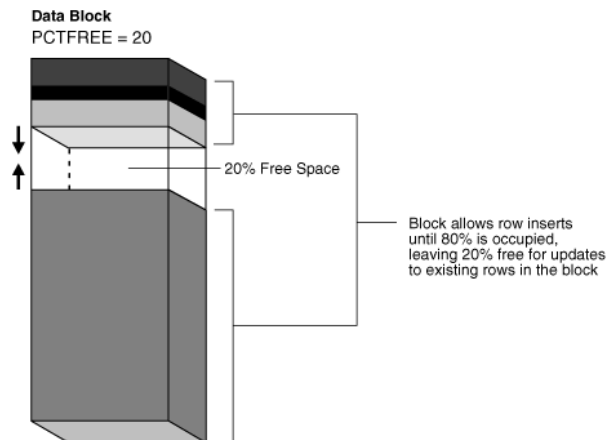
- ❑ Il collegamento tra un database e un'istanza può essere di due tipi:
 - **Single-instance:** a un DB è associata una sola istanza
 - **Oracle Real Application Cluster (RAC):** a un DB possono essere associate più istanze
 - Servizio gestito in modo trasparente con pool di server per una gestione automatica del carico di lavoro. Miglioramento delle performance.
 - Esegue qualsiasi applicazione pacchettizzata o personalizzata senza cambiamenti
 - Maggiore scalabilità: supporta fino a 100 nodi
 - Maggiore fault tolerance
 - E' la base per l'architettura cloud di Oracle
- ❑ In entrambe i casi un'istanza è sempre associata a uno e un solo DB

Organizzazione fisica dei dati

- ❑ I principali file in cui sono mantenute informazioni relativamente a uno specifico database (DB) sono:
 - **I file di controllo:** sono file binari che gestiscono l'architettura fisica del db. Memorizzano le informazioni di controllo sui file del db e vengono utilizzati per gestirne la coerenza interna. Vista l'importanza di questi file ne vengono conservate più copie che dovrebbero essere memorizzate in dischi distinti.
 - Sono indispensabili allo start-up per poter montare il database.
 - Contengono informazioni necessarie in fase di recover (check points)
 - **I file dati (data file):** i dati del db vengono memorizzati in appositi file con estensione .dbf. I nomi dei file riportano il nome del tablespace a cui appartengono.
 - **I file di log on line:** contengono le informazioni relative a tutte le transazioni eseguite e determinano la possibilità di annullare operazioni già eseguite e di evitare perdite di informazioni causate da guasti.
 - **Il file di configurazione:** contiene i parametri di configurazione del db e viene letto in fase di inizializzazione dei processi ad esso correlati.
- ❑ La minima unità fisica letta e scritta da un sistema operativo è quella dell'OS-Block

Organizzazione logica dei dati

- ❑ Al fine di semplificare l'organizzazione dei dati e di migliorare le performance, all'organizzazione fisica viene associata una organizzazione logica
 - **DB-Block:** è la quantità minima di informazioni letta e scritta dal DBMS. Se differisce dal OS-Block deve esserne un multiplo. Il sistema operativo non conosce la dimensione del DB-Block (definita dal parametro DB_BLOCK_SIZE). Parte dello spazio occupato da un DB-Block viene utilizzato per memorizzare meta-dati, una percentuale (parametro PCTFREE) viene lasciato libero per le operazioni di aggiornamento
 - **Extent:** è costituito da più DB-BLOCK contigui. Un extent è sempre contenuto in un solo data file.
 - **Segment:** è un insieme di extent che contengono tutte le informazioni relative a un singolo oggetto (non partizionato) dello schema (es. tabella, indice). Un segmento può essere memorizzato in uno o più data file



Organizzazione logica dei dati

- **Tablespace:** L'elemento principale di suddivisione logica dei dati memorizzati all'interno di un database è il tablespace.
 - Ogni database ha almeno il tablespace SYSTEM
 - Ogni tablespace è costituito da uno o più segmenti ed è quindi memorizzabile in uno o più data file.
 - Ogni utente è associato di default a un tablespace in cui vengono memorizzate le informazioni di sua proprietà

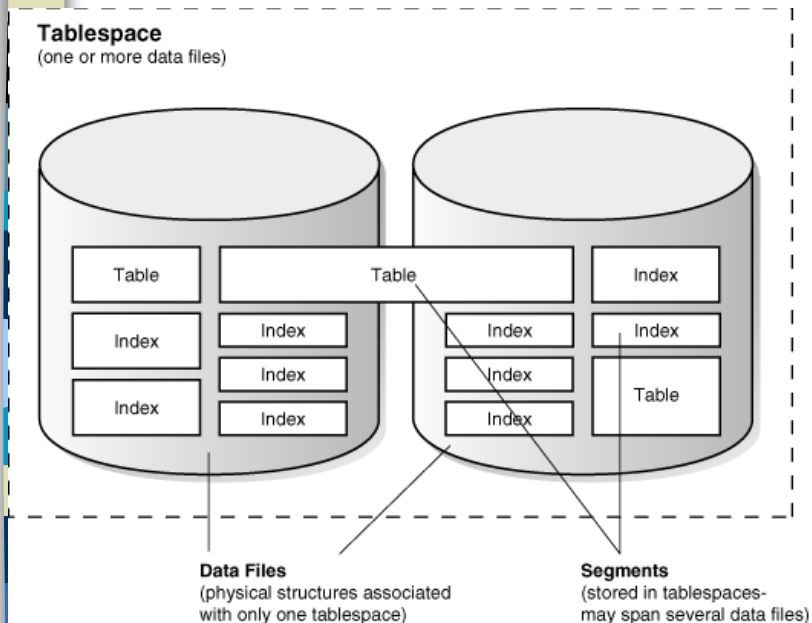
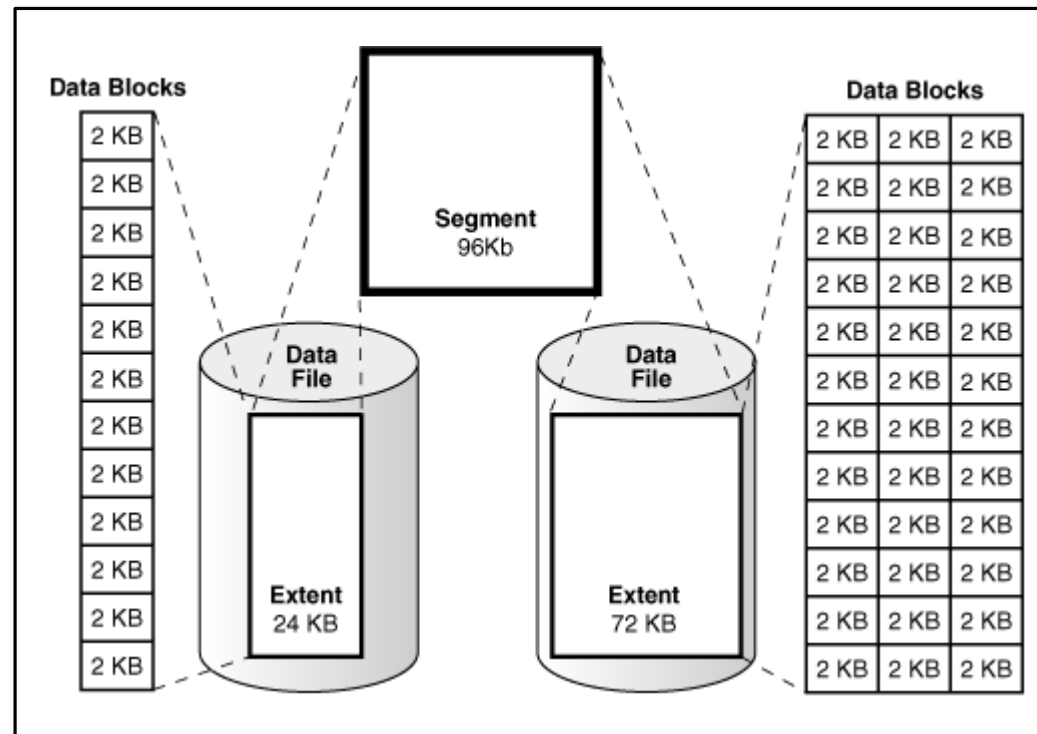


Table space



Database Schema I

- ❑ L'insieme degli oggetti che appartengono a un utente si chiama schema:
 - **Tabelle:** sono i principali contenitori dei dati del database.
 - **Viste:** forniscono una visione parziale dei dati contenuti in una o più tabelle e possono essere pensate come delle interrogazioni sulle tabelle stesse.
 - **Viste materializzate:** memorizzano dati ridondanti ottenuti dai dati presenti nel database stesso. Vengono utilizzate principalmente in applicazioni di data warehousing, database distribuiti e mobile computing per memorizzare i dati maggiormente richiesti dalle applicazioni.
 - **Indici:** sono lo strumento di base per il miglioramento delle performance. Tra quelli messi a disposizione da ORACLE i più comunemente utilizzati sono i B+-Tree e i Bitmap.
 - **Dimensioni:** codificano delle relazioni gerarchie tra coppie di attributi.
 - **Sequenze:** vengono utilizzate per semplificare il lavoro del programmatore e forniscono un elenco sequenziale di valori univoci.
 - **Procedure e Funzioni:** sono blocchi di istruzioni PL/SQL memorizzato nel dizionario dei dati e richiamabili dalle applicazioni. Le procedure permettono di memorizzare in un database le logiche applicative utilizzate frequentemente.
 - **Package:** riuniscono le funzioni e le procedure in raggruppamenti logici.

Database Schema II

- ❑ L'insieme degli oggetti che appartengono a un utente si chiama schema:
 - **Cluster:** indicano insiemi di tabelle memorizzate assieme per motivi prestazionali.
 - **Trigger:** sono procedure che si eseguono al verificarsi di un evento nel database.
 - **Sinonimi:** semplificano l'identificazione degli oggetti (es. nei database distribuiti) e vengono utilizzati per motivi di sicurezza (rendono trasparente la locazione di un oggetto o il suo proprietario).
 - **Privilegi e ruoli:** definiscono le modalità di accesso di un utente a un oggetto del db.
- ❑ Alla creazione sono automaticamente creati due schemi:
 - **Schema SYS:** contiene il data dictionary ed è di proprietà dell'omonimo utente amministratore. Lo schema SYS è manipolato solo dal server stesso
 - **Schema SYSTEM:** contiene altre informazioni necessarie all'amministrazione, è di proprietà dell'omonimo utente amministratore.

II Data Dictionary I

- ❑ Il cuore di un database Oracle è il **data dictionary** che descrive completamente la struttura del database tramite un insieme di tabelle. Il data dictionary contiene:
 - La definizione di tutti gli oggetti dello schema
 - Lo spazio allocato per ogni elemento dagli oggetti dello schema
 - I valori di default per le colonne
 - I vincoli di integrità
 - I nomi degli utenti ORACLE
 - Privilegi e ruoli ricoperti dagli utenti
 - Informazioni di auditing
 - Le statistiche sui dati
- ❑ Il data dictionary è memorizzato nel tablespace SYSTEM, le sue tabelle, di proprietà di SYS, sono in sola lettura per tutti gli altri utenti del database.
- ❑ Un elenco dettagliato delle informazioni contenute nel Data Dictionary è contenuto nel manuale Oracle11g Reference

II Data Dictionary II

- ❑ La struttura del data dictionary è composta da:
 - **Tabelle di base:** sono le tabelle che memorizzano le informazioni vere e proprie in forma normalizzata. Vista la loro complessa struttura sono normalmente lette e scritte solo da ORACLE.
 - **Viste:** riassumono le informazioni memorizzate nelle tabelle di base in modo da renderle più facilmente fruibili agli utilizzatori.

- ❑ Il data dictionary ha tre utilizzi primari
 - **Oracle accede** al data dictionary per trovare le informazioni relative agli utenti, agli schemi e alle strutture di memorizzazione.
 - **Oracle modifica** il data dictionary ogni volta che un'istruzione del DDL viene eseguita
 - **Ogni utente può accedere** al data dictionary (con modalità read-only) per recuperare le informazioni relative al database.

II Data Dictionary III

- ❑ Le viste utilizzate dagli utenti per accedere al data dictionary si suddividono in:
 - **ALL_**: sono viste create nella prospettiva di tutti gli utenti. Queste viste contengono informazioni relative a oggetti, diversi dai propri, a cui gli utenti hanno accesso mediante GRANT espliciti, privilegi e ruoli.
 - Esempio: *“Tutti gli oggetti a cui ho accesso”*
SELECT owner, object_name, object_type FROM ALL_OBJECTS;
 - **USER_**: sono le viste create nella prospettiva del singolo utente. Fanno quindi riferimento agli oggetti propri dell'utente (es. oggetti dello schema, grant, ecc.) e quindi mostrano un sottoinsieme dei dati delle viste ALL_.
 - Esempio: *“Tutti gli oggetti del mio schema”*:
SELECT object_name, object_type FROM USER_OBJECTS;
 - **DBA_**: possono essere interrogate solo dal DBA o da chi possiede il privilegio SELECT ANY TABLE. Forniscono una visione globale dell'intero database. Alcune viste contengono attributi aggiuntivi

II Data Dictionary IV

- ❑ Le viste ALL_, DBA_ e USER_ sono dette statiche poiché il loro contenuto cambia solo a fronte di un cambiamento dei dati del data dictionary.
- ❑ Nel seguito sono elencate alcune delle tabelle su cui lavoreremo. Per una descrizione completa del loro contenuto si veda il manuale Oracle 11g Reference. La sintassi ??? indica che le tabelle sono disponibili sia in versione USER che ALL.
 - **???_IND_COLUMNS:** mantiene le informazioni sulle colonne delle tabelle in cui sono costruiti indici
 - **???_IND_TABLES:** mantiene le informazioni sulle tabelle in cui sono costruiti indici
 - **???_TAB_COLUMNS:** mantiene le informazioni sulle colonne delle tabelle
 - **???_TAB_COL_STATISTICS:** mantiene le statistiche sulle colonne delle tabelle
 - **???_TABLESPACES:** descrive i tablespace accedibili dall'/dagli utente/i
 - **???_ROLE_PRIVS:** descrive i ruoli disponibili all'/agli utente/i
 - **???_COL_PRIVS:** descrive i privilegi disponibili all'/agli utente/i

Dynamic Performance Tables

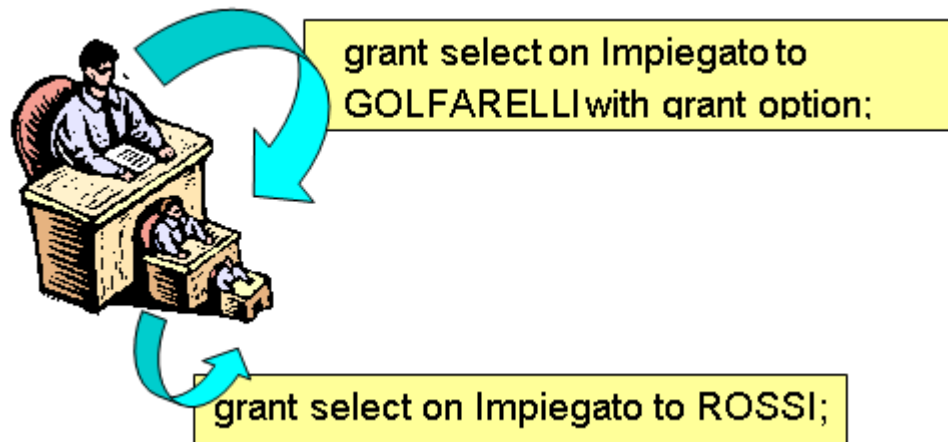
- ❑ Sono continuamente aggiornate mentre il sistema è in funzionamento e contengono principalmente informazioni relative alle performance.
- ❑ Sebbene siano percepite come tali, non sono memorizzate come tabelle nell'accezione relazionale del termine e non possono essere interrogate mediante operazioni di raggruppamento, join e ordinamento
- ❑ Sono identificate dal prefisso V_\$ ma il loro sinonimo pubblico (interrogabile) è V\$
- ❑ Nel seguito sono elencate alcune delle tabelle. Per una descrizione completa del loro contenuto si veda il manuale Oracle Reference.
 - **V\$TRANSACTION**: lista delle transazioni attive
 - **V\$SESSION**: lista delle sessioni attive
 - **V\$LOCK**: lista dei lock in atto
 - **V\$SGA**: descrizione della dimensione dell'area SGA
 - **V\$SQL**: informazioni sull'area condivisa SQL

Gestione degli utenti

- ❑ L'accesso a un database ORACLE è basato sul concetto di **utente** identificato da un **username** e da una **password**.
- ❑ Per ogni utente è definito un insieme di parametri che ne definiscono le caratteristiche:
 - **Username:** definisce univocamente l'utente all'interno del sistema.
 - **Password:** realizza il principale meccanismo di verifica dell'identità dell'utente.
 - **Tablespace di default:** è il tablespace in cui vengono salvati gli oggetti creati nello schema.
 - **Tablespace temporaneo:** è il tablespace in cui vengono salvati i segmenti temporanei utilizzati durante le transazioni.
 - **Quota:** limita la dimensione massima occupabile dagli oggetti creati dall'utente.
 - **Profilo:** specifica il profilo dell'utente.
 - **Ruolo:** definisce i ruoli dell'utente.

I privilegi utente I

- ❑ Avere un account (essere un utente) su un database non è di per se sufficiente a eseguirvi operazioni poiché questa possibilità varia in base all'insieme di *privilegi* che l'utente possiede.
- ❑ Un **privilegio** definisce la possibilità di eseguire un certo tipo di operazione. I privilegi possono essere definiti:
 - **A livello di sistema:** vale per tutti gli oggetti del tipo indicato. Per concedere un privilegio di sistema è necessario avere abilitato il privilegio WITH ADMIN OPTION.
 - **Su uno specifico oggetto:** vale per lo specifico oggetto indicato. Per concedere un privilegio di sistema è necessario avere abilitato il privilegio WITH GRANT OPTION.



I privilegi utente II

- La tabella mostra alcuni dei privilegi di sistema assegnabili in base al tipo di oggetto (per una lista completa si veda il manuale SQL Language Reference).

Privilegio	Consente di...
Indici	
CREATE INDEX	Creare un indice su una tabella nel proprio schema di colui che assegna il privilegio un indice sulle proprie tabelle.
CREATE ANY INDEX	Creare in ogni schema, ad eccezione di quello di SYS un indice di dominio o un indice su qualsiasi tabella.
DROP ANY INDEX	Cancellare gli indici di ogni schema ad eccezione di SYS.
Procedure	
CREATE PROCEDURE	Creare una procedura o funzione nel proprio schema
CREATE ANY PROCEDURE	Creare una procedura o funzione in qualsiasi schema
ALTER ANY PROCEDURE	Modificare procedure e funzioni in qualsiasi schema a esclusione di quello di SYS.
Profili	
CREATE PROFILE	Creare dei profili.
ALTER ANY ROLE	Alterare i profili.
Ruoli	
CREATE ROLE	Creare i ruoli.
ALTER ANY ROLE	Modificare i ruoli.
Tabelle	
CREATE ANY TABLE	Creare tabelle in ogni schema ad eccezione di SYS.
ALTER ANY TABLE	Modificare una tabella o una vista dello schema.
SELECT ANY TABLE	Eseguire interrogazioni su tabelle o viste in qualsiasi schema ad eccezione di SYS.
SYSDBA	Eseguire le operazioni di STARTUP e SHUTDOWN, CREATE DATABASE ecc.

I ruoli

- ❑ Per semplificare la gestione dei privilegi è possibile definire dei **ruoli** ossia degli insiemi di privilegi che definiscono un tipo di utente.
- ❑ I ruoli sono definibili in modo personalizzato ma ORACLE ne fornisce alcuni di default.

RUOLO	SCOPO
CONNECT	Profilo di base per la connessione, include il privilegio di sistema CREATE SESSION
RESOURCE	Fornisce un insieme di privilegi che consentono di svolgere attività di sviluppo, include i seguenti privilegi di sistema: CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE)
DBA	Fornisce tutti i privilegi di sistema WITH ADMIN OPTION.
EXP_FULL_DATABASE IMP_FULL_DATABASE	Consentono di fruire delle funzionalità di importazione ed esportazione.
GATHER_SYSTEM_STATISTICS	Permette all'utente di aggiornare le statistiche di sistema

SQL Data Type I

- ❑ Ogni valore manipolato da ORACLE deve essere associato a un tipo di dato che ne specifica le caratteristiche (es. valori ammessi, valori massimi, minimi, ecc.)
- ❑ Tra i diversi tipi di dati che utilizzeremo più di frequente (vedi Oracle11g SQL Reference) distinguiamo:
 - **Tipi di dati built-in**
 - **CHAR(size)** stringa a lunghezza fissa definita da *size* < 2000.
 - **VARCHAR2(size)** stringa a lunghezza variabile definita da *size* < 4000
 - **NUMBER(p,s)** a virgola fissa con *scale* cifre decimali e *precision-scale* cifre intere (es. *NUMBER(3,2)* → 1.24; *NUMBER(3)*=*NUMBER(3,0)* → 122; *NUMBER* → *floating point*)
 - **FLOAT(p)** *p* indica il numero di cifre binarie significative (utilizzare *NUMBER* per definire numeri *floating point*)
 - **DATE**
 - **ROWID** stringa esadecimale che contiene l'indirizzo univoco di una tupla all'interno di una tabella.
 - **Tipi di dati ANSI:** molti tipi di dati standard ANSI sono supportati e convertiti in tipi di dati ORACLE
 - **Tipi di dati utente:** sono tipi di dati personalizzabili dagli utenti per mezzo del comando *CREATE TYPE*. Sono basati sui tipi di dati precedenti

SQL Data Type: Date

- ❑ Il tipo DATE memorizza le informazioni relative a data e orario fino al secondo (**vedi 3.17 Oracle11g SQL Reference**):
 - Se l'orario non viene specificata il default è 12:00:00 AM
 - Se la data non specificata il default è SYSDATE (la data corrente)
 - Il formato standard utilizzato per la data è specificato nella variabile di sistema NLS_DATE_FORMAT
- ❑ Funzioni per la gestione di valori di tipo DATE
 - **TO_DATE(stringa[,formato])**: converte un campo di testo (CHAR o VARCHAR2) contenente una data espressa nel formato formato in un dato di tipo DATE
 - VARCHAR2(50) v_string:= 'January 15, 1989, 11:00 A.M.';
 - DATE v_date:= TO_DATE(p_date,'Month dd, YYYY, HH:MI A.M.')
 - **TO_CHAR(data[,formato])**: converte un campo data contenente una data espressa nel formato formato in una stringa di tipo VARCHAR2

```
SELECT TO_CHAR(SYSDATE, 'Month DD, YYYY') FROM dual;
```

Si veda pagina 3.61 del manuale SQL Reference per i possibili formati di data
 - **SYSDATE**: restituisce la data corrente

SQL Data Type: Conversioni di tipi

- Una espressione ORACLE deve essere svolta su operatori con lo stesso tipo di dato. Nel caso in cui i tipi di dati siano diversi è necessario effettuare una conversione che può essere implicita o esplicita.

➤ Conversioni implicite

- Quando nelle operazioni di INSERT e UPDATE il dato da inserire viene convertito nel tipo di dato dell'attributo corrispondente.
- Quando viene utilizzata una funzione SQL o un operatore con un parametro di tipo diverso da quello del valore che viene fornito in input, Oracle converte il tipo di dato del valore in quello del parametro.
- Quando viene utilizzato un operatore di comparazione tra due valori di tipi diversi. Oracle converte il tipo di uno dei due operandi.

```
SELECT sal + '10' FROM emp;
```

```
SELECT ename FROM emp WHERE empno = '7936';
```

```
SELECT ename FROM emp WHERE hiredate = '12-MAR-1993';
```

➤ Conversioni esplicite

- CHAR→DATE: TO_DATE
- CHAR→NUMBER: TO_NUMBER(stringa[,formato]) converte stringa tipo CHAR o VARCHAR2 in un numero di tipo NUMBER
- NUMBER→CHAR: TO_CHAR(val[,formato]) converte val di tipo NUMBER in una stringa di tipo VARCHAR2.

SQL Data Definition Language

- ❑ E' la porzione di SQL che permette la creazione/eliminazione degli oggetti dello schema, i principali costrutti che utilizzeremo sono:

➤ **Creazione di tabelle (vedi 16.6 SQL Language Reference)**

```
CREATE TABLE nometabella (  
  nomecol DATATYPE,  
  nomecol DATATYPE,  
  PRIMARY KEY (nomecol,...,nomecol)  
  [,FOREIGN KEY (nomecol,...,nomecol)  
  REFERENCES nometabella (nomecol,...,nomecol)]:
```

dove DATATYPE è uno dei tipi di dati previsti da ORACLE

➤ **Eliminazione di tabelle** (vedi 18.5 SQL Language Reference)

DROP TABLE *nometabella*;

➤ **Creazione di sequenze (vedi 15.67 SQL Language Reference)**

CREATE SEQUENCE *nomesequenza* START WITH *primovalore*
INCREMENT BY *incremento*:

➤ Utilizzo di sequenze

nomesequenza. NEXTVAL

nomesequenza. NEXTVAL

➤ **Eliminazione di sequenze (vedi 18.2 SQL Language Reference)**

DROP SEQUENCE *nomesequenza*;

(ATTENZIONE: per reinizializzare una sequenza a un valore diverso da quello corrente è necessario e ricrearla ex-novo)

SQL Data Definition Language

- E' la porzione di SQL che permette la creazione/eliminazione degli oggetti dello schema, i principali costrutti che utilizzeremo sono:

- **Creazione di indici (vedi 14.61 SQL Language Reference)**

CREATE INDEX [UNIQUE,BITMAP] *nomeindice*
ON *nometabella* (*nomecol*,...,*nomecol*);

- **Eliminazione di indici (vedi 17.51 SQL Language Reference)**

DROP INDEX *nomeindice*;

- **Creazione di procedure (vedi 15.48 SQL Language Reference)**

CREATE [OR REPLACE] PROCEDURE *nomeprocedura*
[(*nomepar* DATATYPE,..., *nomepar* DATATYPE)] IS
Blocco codice PL-SQL;

Import & Export dei dati

- ❑ Oracle 11g mette a disposizione diversi metodi:
 - Strumenti da linea di comando: **SQL*Loader**, Data Pump Import/Export
 - Procedure guidate tramite SQL Developer
- ❑ Lo strumento SQL*Loader (*sqlldr.exe*) consente di importare in maniera selettiva i dati contenuti in file esterni
 - Diversi formati supportati, tra cui TXT, CSV e **RDF**
 - Richiede la definizione di un file di controllo per specificare i parametri di importazione
 - I dati possono essere caricati da più file, manipolati e scritti in più tabelle
 - **NON** esegue DDL: i dati devono essere caricati in tabelle preesistenti
- ❑ Esempio:

```
C:\> %ORACLE_HOME%\bin\sqlldr  
userid = username/password@sid  
control = control_file.ctl
```

```
load data  
infile 'c:\data\mydata.csv'  
into table emp  
fields terminated by "," optionally enclosed by '"'  
( empno, empname, sal, deptno )
```

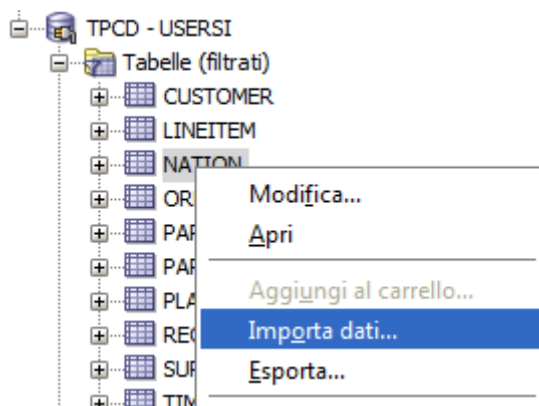
Import & Export dei dati

- ❑ Oracle 11g mette a disposizione diversi metodi:
 - Strumenti da linea di comando: SQL*Loader, **Data Pump Import/Export**
 - Procedure guidate tramite SQL Developer

- ❑ L'import/export dei dati in formato binario è gestito tramite gli strumenti di Data Pump (*impdp.exe*, *expdp.exe*)
 - Tutti gli oggetti di Oracle (tabelle, tablespace, schema, schema objects, database) possono essere esportati e importati
 - Prestazioni elevate e possibilità di filtrare i dati
 - Alternativi ai vecchi strumenti di import/export (*imp.exe*, *exp.exe*), i quali non supportano i tipi di dati Float e Double, ma supportano XMLType (a differenza di Data Pump)

Import & Export dei dati

- ❑ Oracle 11g mette a disposizione diversi metodi:
 - Strumenti da linea di comando: SQL*Loader, Data Pump Import/Export
 - **Procedure guidate tramite SQL Developer**
- ❑ Le procedure guidate automatizzano i processi di import/export
 - Rispetto a SQL*Loader, non supportano il formato RDF, ma supportano il formato Excel, script SQL e PL/SQL
 - L'importazione dei dati non prevede selettività
 - I parametri di import/export possono essere personalizzati:
Strumenti > Preferenze > Database > Utility > Esporta/Importazione



Il software

- ❑ Oracle 11g è liberamente scaricabile dal sito Oracle per usi non commerciali
- ❑ Il 30% del fatturato Oracle proviene da sanzioni per l'utilizzo di software non licenziato
- ❑ Attenzione le soluzioni dei compiti presenti sul sito del corso e relative agli AA antecedenti il 13-14 fanno riferimento alla versione 8i e quindi potrebbero presentare delle incoerenze rispetto alla nuova versione