

# Il database Oracle: concetti generali ed utilizzazione nel sistema di calibrazione del rivelatore di muoni di ATLAS

Elisabetta Vilucchi

INFN-LNF

23/04/07

# Sommario

1. Database relazionali
2. Architettura di un database Oracle
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Sommario

## 1. Database relazionali

### 1. Cos'è un RDBMS

### 2. Il linguaggio SQL

## 2. Architettura di un database Oracle

## 3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

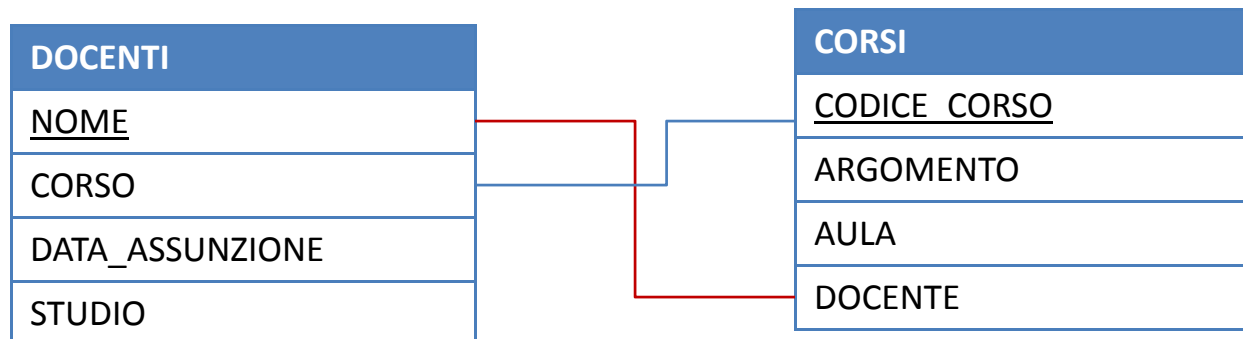
# Cos'è un RDBMS

- Un database è una collezione coerente di dati collegati tra loro, descrive qualche aspetto del mondo reale ed è disegnato per supportare delle applicazioni specifiche
- Un Database Management System (DBMS) è un software disegnato per memorizzare e gestire delle basi di dati
- Un Relational DBMS è un DBMS basato sul modello relazionale

# Organizzazione dei dati in un RDBMS

- Elemento base di un RDBMS e' la tabella bidimensionale, costituita da celle che formano righe e colonne; a ciascuna colonna vengono associati un nome ed un tipo di dati
- Le tabelle possono essere usate in modo indipendentemente o unite tramite *join*, usando una o piu' colonne in comune
- E' possibile definire una tabella in modo tale che una colonna, che verra' utilizzata per stabilire una relazione con una seconda tabella, contenga valori gia' esistenti in tale tabella: *integrita' referenziale*.
  - Il campo contenuto in questa colonna e' una *chiave esterna* che fa riferimento ad un campo di un'altra tabella (*chiave primaria=unique + not null*, della tabella esterna)

I vincoli di integrita' referenziale, dunque definiscono alcune regole nel database che garantiscano la memorizzazione di dati validi (le relazioni tra gli attributi diventano vincoli di integrita')



# Cosa offre un moderno RDBMS

- possibilità di condividere i dati tra utenti e applicazioni diversi
- possibilità di evitare ridondanza nell'archiviazione dei dati
- garanzia della consistenza dei dati
- indipendenza dalla struttura dei dati (possibilità di inserire nuove informazioni, modificare la rappresentazione dei dati, alterare la struttura senza impatto sulle applicazioni)
- flessibilità nel modificare la base di dati
- Possibilità di eseguire in parallelo diverse operazioni
- Esecuzione efficiente delle operazioni (ottimizzatore, indici, ecc.)
- Supporto per volumi "enormi" di dati
- Utility per effettuare backup/recovery e altre operazioni di gestione
- Supporto per nuovi tipi di dato e metodi
- Possibilità di sviluppare applicazioni con diverse tecnologie
- Interfacce evolute (grafiche) per l'accesso e la gestione
- Funzionalità di monitoring, auditing, analisi delle prestazioni
- ... ecc.

# Sommario

## 1. Database relazionali

1. Cos'è un RDBMS

## 2. Il linguaggio SQL

2. Architettura di un database Oracle

3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Il linguaggio SQL

- SQL Structured Query Language: un linguaggio standard di programmazione e interattivo per l'acquisizione e l'aggiornamento delle informazioni in un database
- Diversi tipi di comandi, o *statement*
  - Data Manipulation Language (DML)
  - Data Definition Language (DDL)
  - Data Control Language (DCL)
  - Transaction Control Language (TCL)
- Tutte le istruzioni SQL utilizzano l'ottimizzatore di Oracle per determinare i metodi piu' efficienti di accesso al database



# Il linguaggio SQL

- Data Manipulation Language (DML)
  - Interrogazioni, “query” sui dati
    - Simile all’algebra relazionale
  - Aggiornamenti (inserimenti, cancellazioni, ecc...)
- Data Definition Language (DDL)
  - Creazione/modifica/cancellazione delle strutture
- Data Control Language (DCL)
  - Definizione delle politiche di controllo nell’accesso agli oggetti del database
- Transaction Control Language (TCL)
  - COMMIT, ROLLBACK, SAVEPOINT

# SQL esempi

- **DML:**

```
SELECT colonne
  FROM tabella
  [WHERE condizione]
```

– **select** argomento, docente **from** corso  
**where** aula='CONVERSI'

- **DDL:**

```
CREATE TABLE <nome tabella> (
  attributo      tipo,
  ...
  attributo      tipo))
```

- **DCL:**

```
GRANT privilegio ON oggetto TO utente
```

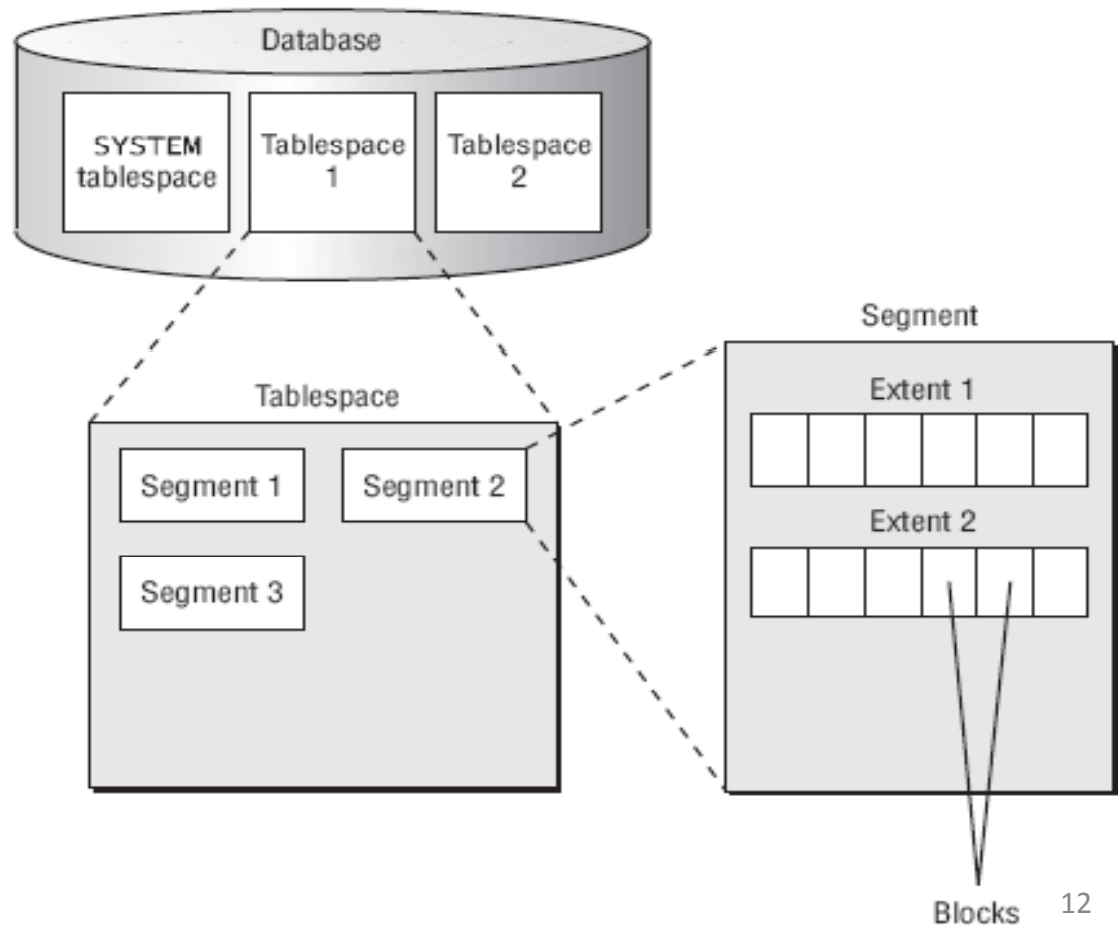
# Sommario

1. Database relazionali
2. Architettura di un database Oracle (Oracle 9i)
  1. Strutture logiche
  2. Utente e schema
  3. Struttura fisica
  4. Struttura della memoria
  5. Background Processes
  6. Server Oracle
  7. Interfacce del database
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Strutture Logiche

- Per gestire, immagazzinare e recuperare i dati in maniera piu' efficientemente Oracle divide il database in unita' piu' piccole:

- *Tablespaces*
- *Blocks*
- *Extent*
- *Segments*



# Tablespaces

- Un database e' logicamente suddiviso in unita' piu' piccole chiamate *tablespaces*. A sua volta un tablespace raggruppa altre strutture logiche correlate. Ad es., si possono raggruppare nello stesso tablespace i dati relativi alla stessa applicazione. Quando si crea un database Oracle crea almeno il tablespace *SYSTEM*. La dimensione di un database e' data dalla somma di tutti i tablespaces

# Blocks

- Un *block* e' la piu' piccola unita' di storage in Oracle ed e' generalmente pari ad un multiplo della misura del blocco di sistema operativo. La misura del block e' data dal parametro *DB\_BLOCK SIZE* ed e' determinata alla creazione del database.

# Extents

- Un *extent* e' un raggruppamento di blocchi contigui allocati insieme

# Segments

- Un *segment* e' costituito da un insieme di *extents* allocati per creare le strutture logiche come: tabelle, indici, ecc...; ne esistono 4 tipi:
  - *Data Segments*: allocati alla creazione di ogni tabella
  - *Index Segments*: allocati alla creazione di ogni indice
  - *Temporary Segments*: creati quando occorre un'area di lavoro temporanea e rilasciati in seguito (nel *Temporary tablespace*)
  - *Undo Segments*: creati per memorizzare tutte le modifiche eseguite sul database (*undo informations*); vengono utilizzati per tornare indietro - *roll back* – rispetto alle modifiche eseguite (nell'*UNDO tablespace*)
- Ogni volta che si crea una struttura logica Oracle alloca un *segment* che contiene almeno un *extent* il quale e' costituito almeno da un blocco



# Sommario

1. Database relazionali
2. **Architettura di un database Oracle**
  1. Strutture logiche
  2. **Utente e schema**
  3. Struttura fisica
  4. Struttura della memoria
  5. Background Processes
  6. Server Oracle
  7. Interfacce del database
3. Requisiti per ospitare un server Oracle
4. Altre funzionalita' di Oracle
5. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Utenti di database

- Per accedere al database un utente deve specificare un account utente del database ed eseguire l'autenticazione
  - Deve avere un tablespace definito
  - Autenticazione:
    - Password
    - External: da sistema operativo, niente password al login al DB
    - Global: consente all'utente di essere identificato tramite: uso della biometrica, certificati x509, dispositivi token (Oracle Advanced Security)
  - Deve avere un profilo (limitazione delle risorse: CPU, spazio sul tablespace, ecc...)
    - Si possono assegnare ruoli, cioè insiemi di privilegi

# Utenti e schemi del database

- La raccolta di oggetti di cui un utente e' proprietario e' definita: *schema*
- un utente puo' essere associato ad un solo *schema*
- Lo schema non e' collegato ai tablespace in cui le strutture logiche sono state create, poiche' gli oggetti di uno stesso *schema* possono risiedere su piu' tablespace ed un tablespace puo' contenere oggetti di diversi schema

## Oggetti dello schema:

- Tabelle
- Trigger
- Indici
- Viste
- Sequence
- Procedure
- Sinonimi
- Tipi di dati (definiti dall'utente)
- Database link

# Alcuni Oggetti dello Schema

- VISTA: rappresentazione “personalizzata” dei dati di una o piu’ tabelle o altre viste: *stored query*. Le viste non contengono dei dati ma li ottengono dalle tabelle su cui si basano
- *Materialized view*: e’ un oggetto del database che contiene il risultato id una query: copie locali di dati remoti (snapshots). La query puo’ essere fatta su tabelle, viste e altre *materialized view*.

# Alcuni Oggetti dello Schema

- *Sequence*: un oggetto di database dal quale piu' utenti possono generare numeri interi univoci (*primary key*)
- *Trigger*: oggetti di codice PL/SQL memorizzati nel database, vengono eseguiti o attivati automaticamente in seguito ad un evento
- **INDICI**: strutture opzionali associate alle tabelle per migliorare le prestazioni del recupero dei dati, forniscono un accesso diretto ai dati della tabella
- *DB Link*: un database link e' un oggetto dello schema che permette l'accesso ad oggetti di un altro database

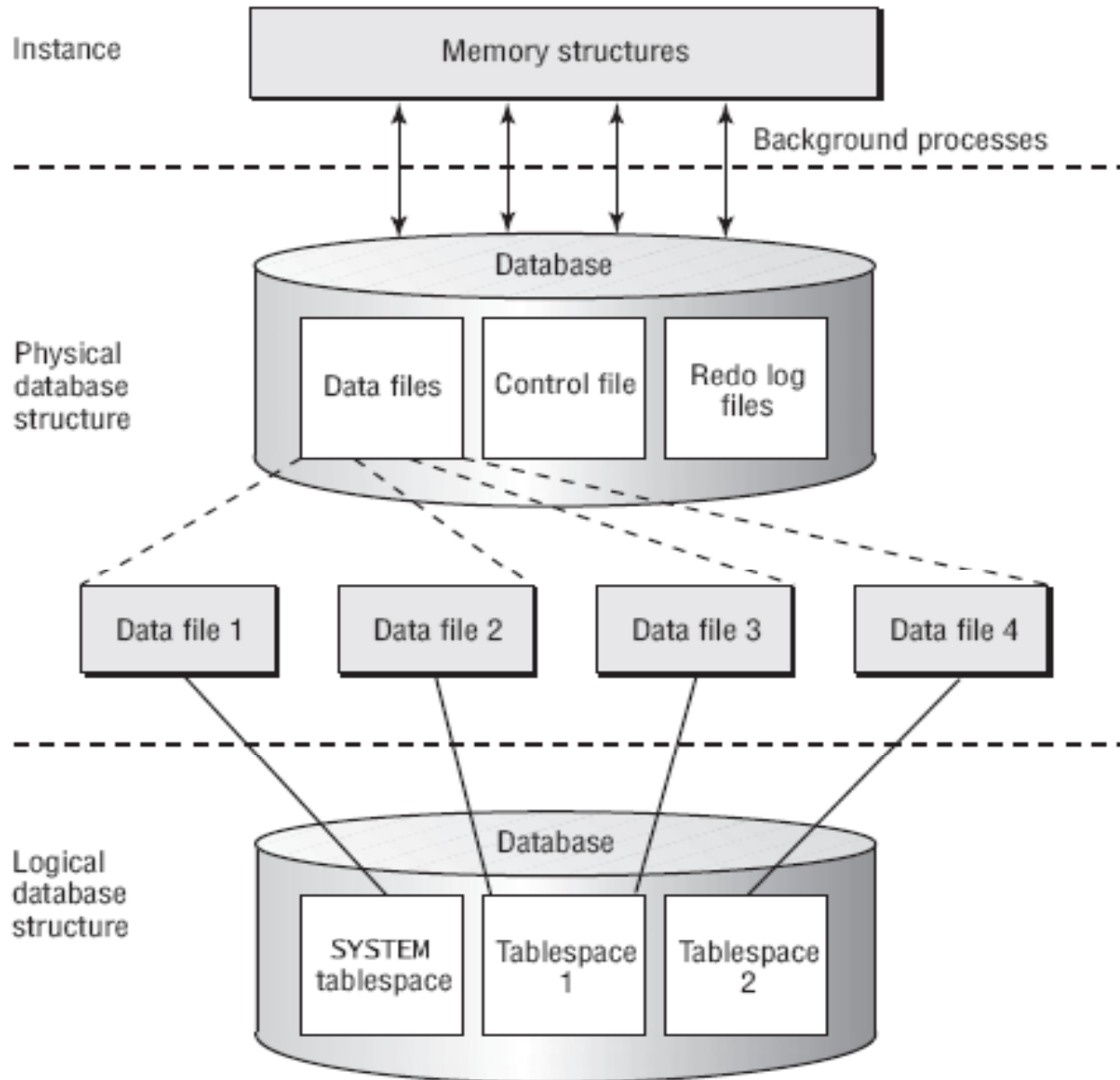
# Sommario

1. Database relazionali
2. **Architettura di un database Oracle**
  1. Strutture logiche
  2. Utente e schema
  3. **Struttura fisica**
  4. Struttura della memoria
  5. Background Processes
  6. Server Oracle
  7. Interfacce del database
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Struttura Fisica

- La struttura fisica del database e' costituita da tre tipi di file:
  - *Data Files*
  - *Redo Log Files*
  - *Control Files*

# Struttura fisica di un Server Oracle





# Data Files

- Contengono tutti i dati del database. Ciascun *data file* e' associato ad un solo *tablespace*, ogni *tablespace* e' costituito da uno o piu' *data file*.

# Redo Log Files

- Nei *redo log files* vengono memorizzati tutte le modifiche fatte al database. Ogni DB possiede almeno due file di redo log, perché Oracle scrive in questi file in maniera circolare: quando un file di redo log è pieno allora Oracle scrive in quello successivo, quando l'ultimo file di redo log è pieno allora Oracle ricomincia dal primo, assicurandosi però di memorizzare le informazioni nei data file prima di sovrascriverle. Sono critici per il database, quindi se ne possono avere piu' copie - *multiplexing* - e si parla di *redo log group*.
- Nei *Redo Log* viene scritto il numero di *check point*, all'avvio il processo *SPMON* esegue il *recovery* dell'Istanza: legge nel *Redo Log* l'ultimo numero di *check point* e lo confronta con l'*scn* nei *Control File* e nei *Data Files* e se non e' lo stesso, fa il *roll back* di tutte le transazioni fino ad arrivare ad un numero coerente.

# Control Files

- Mantengono le informazioni sulla struttura fisica del database, sono critici per il database per cui Oracle puo' mantenerne piu' copie – *multiplexed*. Contiene il nome del database, il *timestamp* della creazione e la locazione fisica di tutti i *data files* e *redo log files*.

# Altri file importanti per il corretto funzionamento del database

- *Parameter file, init file, pfile, spfile*: definisce la configurazione dell'istanza all'avvio
- *Password file*: consente agli utenti di connettersi in remoto
- *Archived log files*: log file archiviati, con un backup e questi file si puo' recuperare un data file perso (DB in modalita' *archived log*)

# Sommario

1. Database relazionali

**2. Architettura di un database Oracle**

1. Strutture logiche

2. Utente e schema

3. Strutture fisiche

**4. Struttura della memoria**

**1. System Global Area**

2. Program Global Area

5. Background Processes

6. Server Oracle

7. Interfacce del database

3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

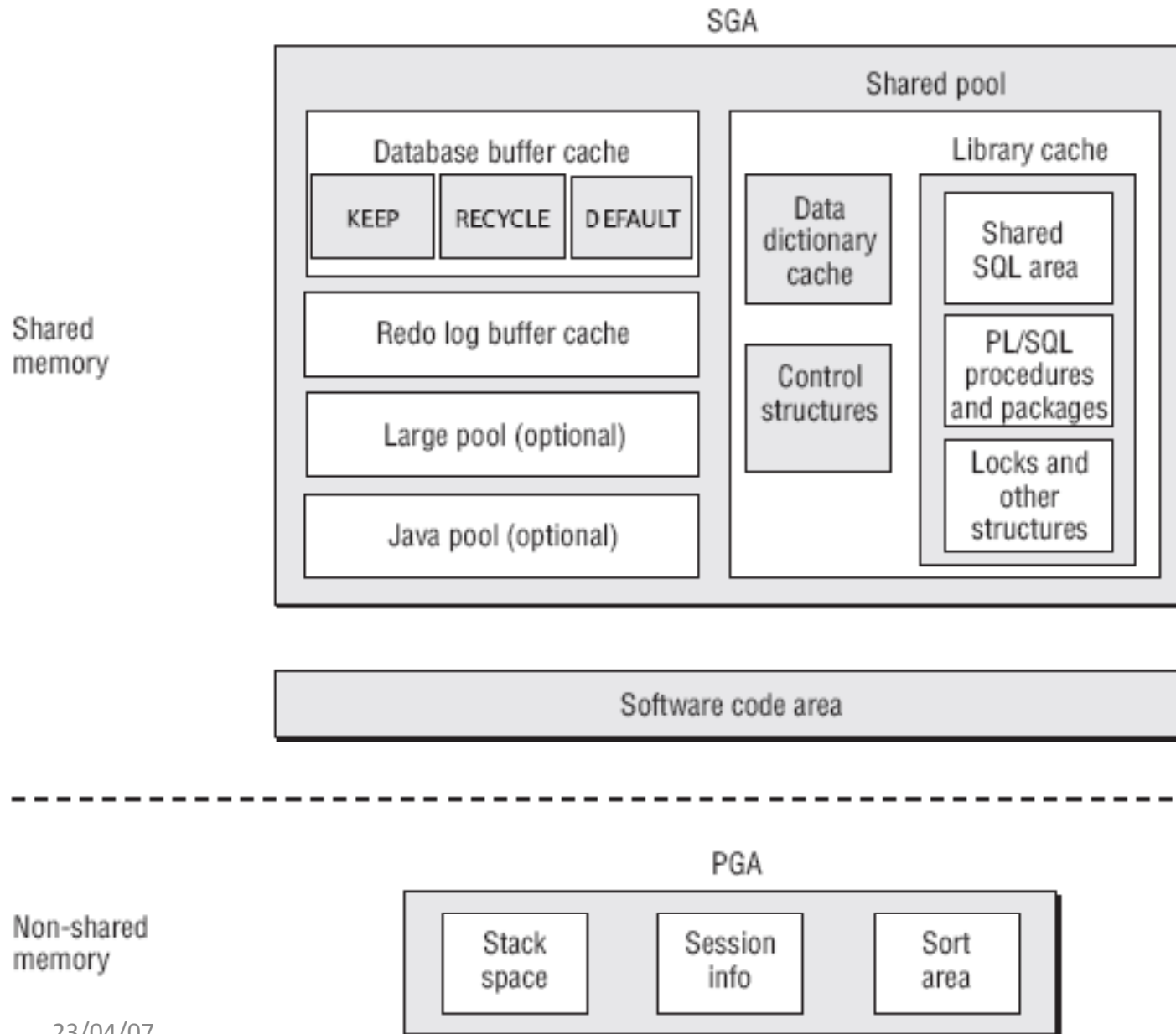
# Struttura della memoria

- Le strutture di memoria vengono utilizzate per “mettere in cache”:
  - Dati per le applicazioni
  - Informazioni del *data dictionary* (metadati degli oggetti, delle strutture logiche, schemi, privilegi, ecc...)
  - Comandi SQL (*Structured Query Language*)
  - Program Unit PL/SQL e Java
  - Informazioni sulle transazioni
  - Dati per singole richieste al database
  - .....

# Struttura della Memoria

- Le strutture di memoria vengono allocate dall'istanza Oracle all'avvio. Le due maggiori strutture sono:
  - *System Global Area – SGA (Shared Global Area)*
  - *Program Global Area – PGA (Private Global Area, Process Global Area)*

# Struttura della Memoria





# System Global Area

- La *SGA* e' un'area di memoria condivisa da tutti gli utenti del database, viene allocata all'avvio dell'istanza Oracle e rilasciata alla chiusura - *shut down*.
- Le informazioni immagazzinate nella *SGA* sono suddivise in piu' strutture di memoria dinamiche, la cui dimensione totale pero' non puo' eccedere il parametro di inizializzazione *SGA\_MAX\_SIZE*.
- La memoria nella *SGA* viene allocata in unita' di memoria contigue: *granules* (minimo 3 *granules*).

# System Global Area - SGA

- La *System Global Area* e' costituita dalle seguenti aree di memoria:
  - *Database Buffer Cache: keep, recycle e default buffer pools*
  - *Shared Pool: Library Cache, Data Dictionary Cache*
  - *Redo Log Buffer Cache*
  - *Large Pool (opzionale)*
  - *Java Pool (opzionale)*
- ✓ *v\$buffer\_pool*, vista dinamica delle performance che mostra la quantita' di *granules* allocati per: *default, keep e recycle buffer pools*

# System Global Area: Database Buffer Cache

- La *Database Buffer Cache* e' l'area di memoria che contiene i dati del database: i blocchi dei *data files* letti recentemente; ne esistono 3 tipi:
  - *Dirty Buffers*: *buffer blocks* non ancora scritti su disco (nei *data files*)
  - *Free Buffers*: non contengono dati o contengono dati che possono essere sovrascritti (es. i dati letti da disco)
  - *Pinned Buffers*: *buffers* acceduti al momento o mantenuti per uso futuro (es. *Keep buffer pool*)
- Oracle mantiene due liste per gestire la *DB Buffer Cache*: *write list (dirty buffers)* e *last recently used list (LRU: free buffer, pinned buffer, dirty buffer non ancora spostati nella write list)*.

# System Global Area: Redo Log Buffer

- La *Redo Log Buffer* e' una memoria circolare che mantiene le informazioni sulle modifiche apportate al database tramite i comandi: INSERT, UPDATE, DELETE, CREATE, ALTER o DROP.
- Queste informazioni vengono usate per ripetere le modifiche fatte in caso di *failure* e procedere al *recovery*: se il database viene chiuso in maniera anomala e non viene perso alcun file, l' stanza puo' seguire il recupero del database con le informazioni che da questa area di memoria vengono scritte nei *Redo Log file*.

# System Global Area: Shared Pool

- La *Shared Pool* area mantiene informazioni quali: *statement SQL, procedurea e packages PL/SQL, data dictionary, locks, ecc....*:
  - *Library Cache*: e' composta dalle: *shared SQL area* (comandi SQL eseguiti e piani di esecuzione), *private SQL area (bind variables)*, *procedure e packages PL/SQL* (il codice dei programmi *PL/SQL* eseguiti), strutture di controllo (*locks, ecc...*)
  - *Data Dictionary Cache*: il ***Data Dictionary*** e' un insieme di tabelle e viste contenenti metadati sul database, sulle sue strutture, privilegi e utenti. La *Data Dictionary Cache* contiene le informazioni del *Data Dictionary* usate piu' recentemente.

# System Global Area: Large Pool, Java Pool

- Sono aree della *SGA* opzionali che il *DBA* puo' scegliere di configurare:
  - *Large Pool Area*: fornisce un'alteriore allocazione di memoria per particolari operazioni quali *backup* e *Restore* (no *LRU list*, puo' essere modificata dinamicamente)
  - *Java Pool Area*: fornisce memoria per le operazioni Java (puo' essere modificata dinamicamente)

# Sommario

1. Database relazionali

2. Architettura di un database Oracle

1. Strutture logiche

2. Utente e schema

3. Strutture fisiche

4. Struttura della memoria

1. System Global Area

2. Program Global Area

5. Background Processes

6. Server Oracle

7. Interfacce del database

3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Program Global Area - PGA

- Una *Program Global Area* contiene dati ed informazioni di UN solo processo : NON e' condivisa tra piu' processi; viene allocata per OGNI *server process* e rilasciata quando il processo e' terminato.
- Il contenuto dipende dalla configurazione del server:
  - *Dedicated Server*: per ogni connessione al database la *PGA* contiene lo *stack space* e le informazioni sulla sessione
  - *Shared Server*: contiene solo l'informazione sullo *stack space* (le informazioni sulla sessione sono nella *SGA*)
- *Dedicated Server*: un *server process* per ogni connessione
- *Shared Server*: le connessioni sono raggruppate per mezzo di un *dispatcher*



# Program Global Area

- La *PGA* contiene anche:
  - *Sort Area*: in una configurazione *Dedicated Server*, Oracle usa la *Sort Area* per le operazioni di ordinamento (nel caso di *Shared Server* la *Sort Area* si trova nella *SGA*).
  - *Software Area*: area in cui viene immagazzinato il codice che deve essere eseguito.

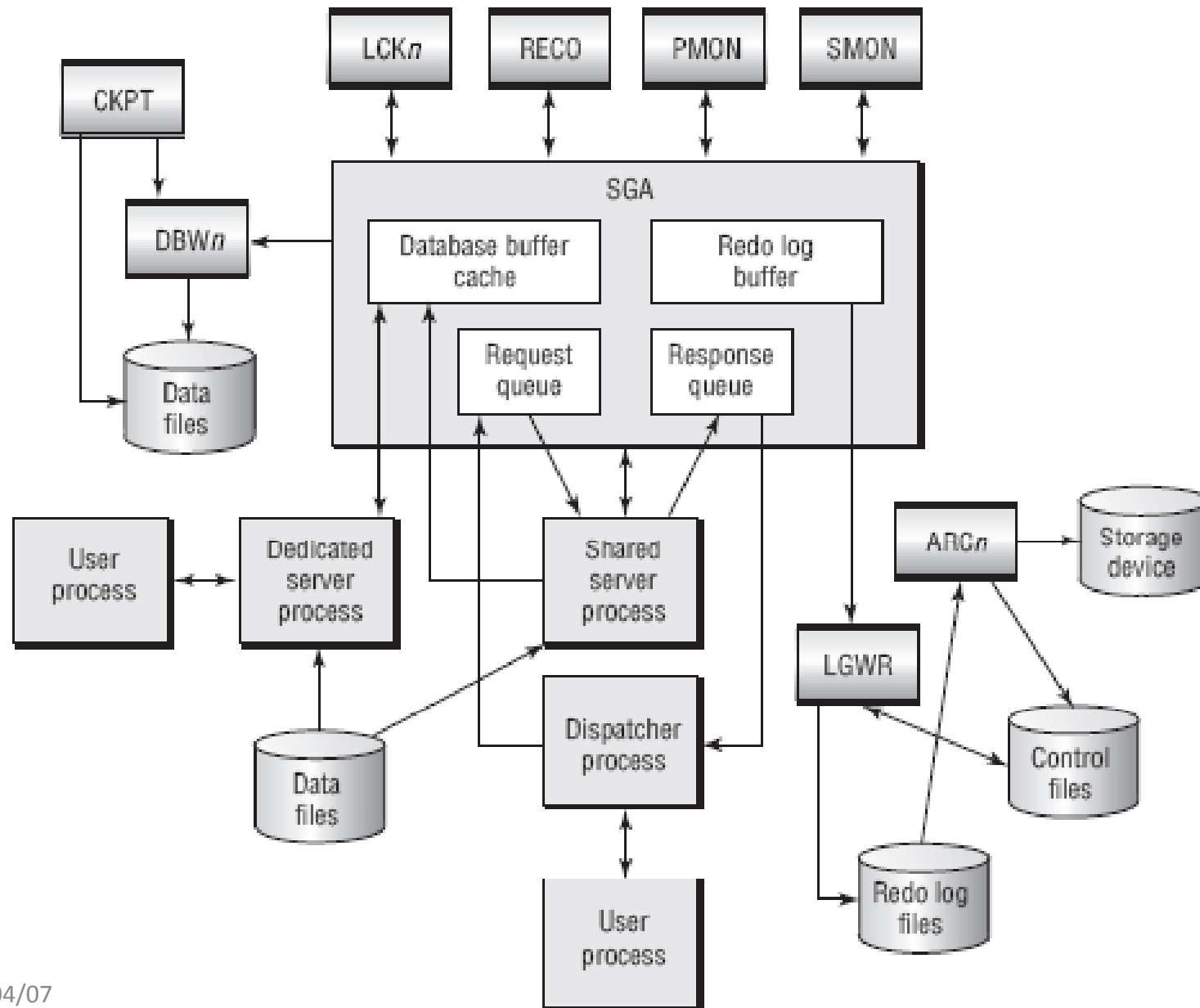
# Sommario

1. Database relazionali
2. **Architettura di un database Oracle**
  1. Struttura logiche
  2. Utente e schema
  3. Struttura fisiche
  4. Struttura della memoria
  5. **Background Processes**
  6. Server Oracle
  7. Interfacce del database
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Background Processes

- All'avvio Oracle lancia numerosi processi in background:
  - *Database Writer (DBWn)*
  - *Log Writer (LGWR)*
  - *Checkpoint (CKPT)*
  - *System Monitor (SMON)*
  - *Process Monitor (PMON)*
  - *Archiver (ARCn)*
  - *Recoverer (RECO)*
  - *Lock (LCKn)*
  - *Queue Monitor (QMn)*
  - *Dispatcher (Dnnn)*
  - *Shared Server (Snnn)*

# Background Processes



# Default Background Processes

I processi associati ad ogni istanza per default:

- *Database Writer (DBWn)*: scrive nei *data files* il contenuto del *dirty buffer* (agisce quando: non ci sono piu' *clean buffer*, avviene un *checkpoint*, un *tbs* viene posto in *backup mode*, ecc...; non dipende dal COMMIT)
- *Log Writer (LGWR)*: scrive i blocchi del *Redo Log Buffer* nei *Redo Log Files* online (agisce quando viene lanciato COMMIT, il *Redo Log Buffer* e' pieno per 1/3, il *DBWn* scrive il *dirty buffer* su disco, ogni 3 sec, ecc...)
- *Checkpoint (CKPT)*: e' un evento che scarica le modifiche dalla *Buffer Cache* al disco e aggiornando *Control File* e *Data Files*, percio' riduce il tempo necessario per il recovery dell'istanza (si verifica automaticamente quando il *Redo Log* online e' pieno – *Log Switch*)

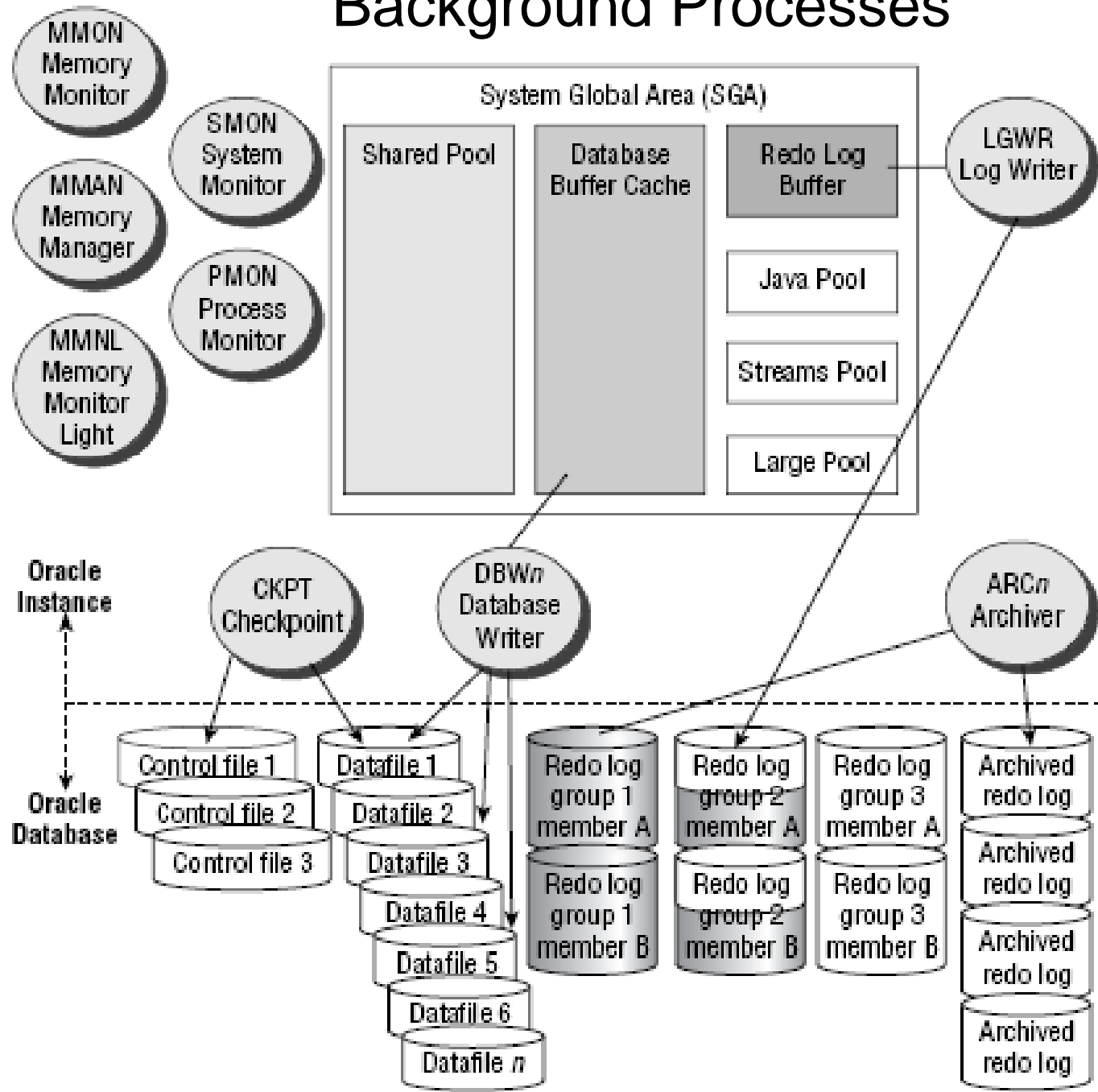
# Default Background Processes

- *System Monitor (SMON)*: realizza l'istanza o il *crash recovery* all'avvio del database usando i *Redo Log* online, libera i segmenti temporanei nei tablespace e realizza il *coalescing* degli spazi liberi contigui (si attiva regolarmente o può essere richiamato da altri processi)
  - *Process Monitor (PMON)*: ripulisce dagli *user processes* falliti (aggiornando la tabella delle transazioni attive, ecc..) e libera le risorse usate (si attiva periodicamente)
- Se uno di questi processi fallisce, l'istanza fallisce

# Altri Processi Oracle

- A livello utente, l'interazione con l'istanza, e perciò con il database, avviene tramite lo *user process* ed il *server process*. Le applicazioni accedono al database tramite lo *user process*: quando un utente lancia un'applicazione Oracle avvia uno *user process* per supportare la connessione all'istanza; stabilita la connessione viene avviato un *server process* sul server, il quale esegue gli stament SQL e restituisce il risultato allo *user process*
- *Archiver (ARCn)*: se il database e' in modalita' *ARCHIVELOG*, i *Redo Log files* online vengono copiati su disco prima di essere sovrascritti. Così si puo' eseguire il *recovery* del database fino al momento del fallimento.

# Background Processes





# Sommario

1. Database relazionali
- 2. Architettura di un database Oracle**
  1. Strutture logiche
  2. Utente e schema
  3. Strutture fisiche
  4. Struttura della memoria
  5. Background Processes
  - 6. Server Oracle**
  7. Interfacce del database
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Server Oracle

- Un **Database Oracle** e' la struttura fisica che contiene i dati:  
*Control Files, Log Files e Data Files*
  - Un'**Istanza** Oracle e' data dalle strutture della memoria (*SGA* e *PGA*) e i Processi di Background
  - **Un Server Oracle e' dato dal Database e l'Istanza**
  - Ogni Database e' associato (almeno) ad una Istanza (Nella configurazione *RAC* un database e' associato a piu' Istanze)
- Stato di un'Istanza:
- *SHUTDOWN*: nessun processo di background attivo
  - *STARTUP NOMOUNT*: l'Istanza deve poter accedere all'*init file* (*PFILE* o *SPFILE*)
  - *MOUNT*: l'Istanza controlla che **TUTTI** i *Control File* elencati nell'*init file* siano identici
  - *OPEN*: tutti i *Redo Log Groups* devono avere un file disponibile e i *data files* devono essere disponibili

# Sommario

1. Database relazionali
- 2. Architettura di un database Oracle**
  1. Strutture logiche
  2. Utente e schema
  3. Strutture fisiche
  4. Struttura della memoria
  5. Background Processes
  6. Server Oracle
  - 7. Interfacce al database**
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# Interfacce al database

- Il linguaggio SQL: SQL\*Plus
- *i*SQL\*Plus
- OEM: Oracle Enterprise Manager
- .....

# SQL\*Plus

- PL/SQL: Procedural Language/Structured Query Language
- E' un linguaggio di programmazione che estende il linguaggio SQL

```
[oracle@fuji ~]$ sqlplus elisabetta
```

```
SQL*Plus: Release 10.2.0.1.0 - Production on Thu Apr 12  
10:22:24 2007
```

```
Connected to:
```

```
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0  
- Production
```

```
With the Partitioning, OLAP and Data Mining options
```

```
SQL>
```

- Tutte le istruzioni SQL utilizzano l'ottimizzatore di Oracle per determinare i metodi piu' efficienti di accesso al database

# SQL\*Plus

- PL/SQL fornisce:
  - Estensioni procedurali all'SQL: e' possibile manipolare i dati con istruzioni SQL e controllarne il flusso con costrutti procedurali quali: IF-THEN, CASE, LOOP. E' possibile dichiarare costanti, definire procedure e funzioni
  - Supporto per programmazione orientata ad oggetti: puo' richiamare altri programmi scritti in altri linguaggi quali: C, C++ e Java
  - Viene eseguito all'interno del database, per cui e' particolarmente efficace per operazioni con uso intensivo di dati e riduce al minimo il traffico di rete nelle applicazioni

# Oggetti PL/SQL

- Sono disponibili molti tipi di oggetti di database PL/SQL:
  - **Package**: raccolta di procedure e funzioni correlate in modo logico, rappresenta l'interfaccia alle applicazioni (vengono dichiarati i tipi, le variabili, le costanti, i cursori, i sottoprogrammi, ...)
  - **Package body**: il corpo procedurale che definisce i cursori e i sottoprogrammi, contiene i dettagli dell'implementazione e le dichiarazioni private nascoste all'applicazione
  - **Type body**: raccolta di metodi (procedure e funzioni) associati ai tipi di dati definiti dall'utente.
  - **Procedura**: un blocco PL/SQL che esegue una determinata azione
  - **Funzione**: un blocco PL/SQL che restituisce un solo valore
  - **Trigger**: un blocco PL/SQL eseguito quando si verifica un determinato evento nel database

# *iSQL\*Plus*

- *iSQL\*Plus* e' un'interfaccia al database basata su un browser
- Prevede che sul server sia avviato un processo listener

`http://nomehost:porta/isqlplus`

➤ Mantiene un'history



# iSQL\*Plus

iSQL\*Plus Release 10.2.0.1.0 Production - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://fuji.roma3.infn.it:5560/isqlplus/workspace.uix

ORACLE  
iSQL\*Plus

Logout Preferences Help

Workspace History

Connected as ELISABETTA@orcl

Workspace

Enter SQL, PL/SQL and SQL\*Plus statements.

```
select * from mdt_tube
```

Execute Load Script Save Script Cancel

SEOID	HEADID	TUBEID	LOWRUN	UPRUN	RUNFLAG	CALIBFLAG	VALIDFLAG	ENTRIES	CHISQUARE	P1	P2	P3	P4	P5
76	4	241116	600779	600779	0	0	0	2.0E+004	1.2E+003	7.34E-003	9.391E+000	6.708E+000	2.22E+002	6.535E+002
77	4	241117	600779	600779	0	0	0	2.0E+004	1.2E+003	1.563E-002	5.661E+000	5.723E+000	1.546E+002	6.493E+002
78	4	241214	600779	600779	0	0	0	2.0E+004	1.2E+003	8.94E-003	6.292E+000	4.876E+000	1.838E+002	6.54E+002
79	4	241215	600779	600779	0	0	0	2.0E+004	1.2E+003	5.06E-003	1.098E+001	6.43E+000	1.951E+002	6.539E+002
80	4	241216	600779	600779	0	0	0	2.0E+004	1.2E+003	5.83E-003	1.275E+001	5.597E+000	1.987E+002	6.515E+002
81	4	241217	600779	600779	0	0	0	2.0E+004	1.2E+003	9.44E-003	6.034E+000	8.86E+000	2.215E+002	6.514E+002
82	4	241313	600779	600779	0	0	0	2.0E+004	1.2E+003	1.841E-002	2.643E+000	5.501E+000	1.974E+002	6.511E+002
83	4	241314	600779	600779	0	0	0	2.0E+004	1.2E+003	3.46E-003	9.415E+000	5.143E+000	1.968E+002	6.555E+002
84	4	241315	600779	600779	0	0	0	2.0E+004	1.2E+003	5.2E-003	1.412E+001	6.044E+000	1.086E+002	6.540E+002

23/04/07

Done

Idle

# OEM: Oracle Enterprise Manager

- Interfaccia grafica per la manipolazione degli oggetti e delle strutture del database
- Non occorre ricordarsi la sintassi
- Non occorre installare un client per avere un'interfaccia grafica verso il database
- Disponibile a partire dalla versione 10g

# OEM: Oracle Enterprise Manager

Oracle Enterprise Manager (ELISABETTA) - Database Instance: orcl - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://fuji.roma3.infn.it:1158/em/console/database/instance/sitemap?event=doLoad&target=orcl&type=oracle\_databas

ORACLE Enterprise Manager 10g Database Control

Help Logout Database

Logged in As ELISABETTA

## Database Instance: orcl

Home Performance Administration Maintenance

Page Refreshed Apr 12, 2007 10:13:51 AM Refresh View Data Automatically (60 sec)

### General

Shutdown

Status **Up**

Up Since **Apr 3, 2007 1:30:48 PM CEST**

Instance Name **orcl**

Version **10.2.0.1.0**

Host **fuji.roma3.infn.it**

Listener **LISTENER\_fuji.roma3.infn.it**

[View All Properties](#)

### Host CPU

Load **0.16** Paging **0.00**

### Active Sessions

Maximum CPU **2**

### SQL Response Time

Baseline is empty.

Reset Baseline

### Diagnostic Summary

ADDM Findings **0**

All Policy Violations **10**

Alert Log **Apr 12, 2007 8:01:09 AM**

### Space Summary

Database Size (GB)	45.786
Problem Tablespace	0
Segment Advisor Recommendations	0
Space Violations	0
Dump Area Used (%)	36

### High Availability

Instance Recovery Time (sec)	10
Last Backup	Apr 12, 2007 8:01:09 AM
Usable Flash Recovery Area (%)	12.2
Flashback Logging	Disabled

### Alerts

Category All Critical 0 Warning 5

Severity	Category	Name	Message	Alert Triggered
Warning	Invalid Objects by	Owner's Invalid Object	4 object(s) are invalid in the ROME3_ATLAS schema.	Feb 6, 2007 2:28:43

Done | Idle

# OEM: Oracle Enterprise Manager

The screenshot shows the Oracle Enterprise Manager 10g console interface. The browser window title is "Oracle Enterprise Manager (ELISABETTA) - Database Instance: orcl - Mozilla Firefox". The address bar shows the URL: [http://fuji.roma3.infn.it:1158/em/console/database/instance/sitemap?event=doLoad&target=orcl&type=oracle\\_databases](http://fuji.roma3.infn.it:1158/em/console/database/instance/sitemap?event=doLoad&target=orcl&type=oracle_databases). The page header includes the Oracle logo, "Enterprise Manager 10g", and "Database Control". A navigation bar at the top right contains "Help" and "Logout" links, and a "Database" tab is active. The user is logged in as "ELISABETTA".

## Database Instance: orcl

Home Performance **Administration** Maintenance

The Administration tab displays links that allow you to administer database objects and initiate database operations inside an Oracle database. The Maintenance tab displays links that provide functions that control the flow of data between or outside Oracle databases.

### Database Administration

<b>Storage</b> <ul style="list-style-type: none"><li><a href="#">Control Files</a></li><li><a href="#">Tablespaces</a></li><li><a href="#">Temporary Tablespace Groups</a></li><li><a href="#">Datafiles</a></li><li><a href="#">Rollback Segments</a></li><li><a href="#">Redo Log Groups</a></li><li><a href="#">Archive Logs</a></li></ul>	<b>Database Configuration</b> <ul style="list-style-type: none"><li><a href="#">Memory Parameters</a></li><li><a href="#">Undo Management</a></li><li><a href="#">All Initialization Parameters</a></li><li><a href="#">Database Feature Usage</a></li></ul>	<b>Database Scheduler</b> <ul style="list-style-type: none"><li><a href="#">Jobs</a></li><li><a href="#">Chains</a></li><li><a href="#">Schedules</a></li><li><a href="#">Programs</a></li><li><a href="#">Job Classes</a></li><li><a href="#">Windows</a></li><li><a href="#">Window Groups</a></li><li><a href="#">Global Attributes</a></li></ul>
<b>Statistics Management</b> <ul style="list-style-type: none"><li><a href="#">Automatic Workload Repository</a></li><li><a href="#">Manage Optimizer Statistics</a></li></ul>	<b>Change Database</b> <ul style="list-style-type: none"><li><a href="#">Migrate to ASM</a></li><li><a href="#">Make Tablespace Locally Managed</a></li></ul>	<b>Resource Manager</b> <ul style="list-style-type: none"><li><a href="#">Monitors</a></li><li><a href="#">Consumer Groups</a></li><li><a href="#">Consumer Group Mappings</a></li><li><a href="#">Plans</a></li></ul>
<b>Policies</b> <ul style="list-style-type: none"><li><a href="#">Policy Library</a></li><li><a href="#">Policy Violations</a></li></ul>		

### Schema

<b>Database Objects</b> <ul style="list-style-type: none"><li><a href="#">Tables</a></li><li><a href="#">Indexes</a></li><li><a href="#">Views</a></li><li><a href="#">Synonyms</a></li><li><a href="#">XML Schemas</a></li><li><a href="#">XML Type Tables</a></li></ul>	<b>Programs</b> <ul style="list-style-type: none"><li><a href="#">Packages</a></li><li><a href="#">Package Bodies</a></li><li><a href="#">Procedures</a></li><li><a href="#">Functions</a></li><li><a href="#">Triggers</a></li></ul>	<b>XML Database</b> <ul style="list-style-type: none"><li><a href="#">Configuration</a></li><li><a href="#">Resources</a></li><li><a href="#">Access Control Lists</a></li><li><a href="#">XML Schemas</a></li><li><a href="#">XML Type Tables</a></li></ul>
---	---	--

23/04/2017 14:07:00

Done | Idle | 60

# OEM: Oracle Enterprise Manager

Oracle Enterprise Manager (ELISABETTA) - Tables - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://fuji.roma3.infn.it:1158/em/console/database/databaseObjectsSearch?target=orcl&type=oracle\_database&otype=

ORACLE Enterprise Manager 10g Database Control

Help Logout Database

Database Instance: orcl > Tables Logged in As ELISABETTA

Tables Object Type Table

**Search**

Select an object type and optionally enter a schema name and an object name to filter the data that is displayed in your results set.

Schema

Object Name

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode

Select	Schema	Table Name	Tablespace	Partitioned	Rows	Last Analyzed
<input checked="" type="radio"/>	ELISABETTA	<a href="#">END_INSERT_TUBE</a>	EXAMPLE	NO	14	Feb 1, 2007 1:11:26 PM CET
<input type="radio"/>	ELISABETTA	<a href="#">END_INSERT_TUBE_HEAD</a>	EXAMPLE	NO	6	Feb 1, 2007 1:11:26 PM CET
<input type="radio"/>	ELISABETTA	<a href="#">MDT_HEAD_RT</a>	USERS	NO	0	Apr 11, 2007 4:00:06 AM CEST
<input type="radio"/>	ELISABETTA	<a href="#">MDT_RT</a>	USERS	NO	0	Apr 11, 2007 4:00:06 AM CEST
<input type="radio"/>	ELISABETTA	<a href="#">MDT_RT_CHEBY</a>	USERS	NO	0	Apr 12, 2007 4:00:08 AM CEST
<input type="radio"/>	ELISABETTA	<a href="#">MDT_RT_MAP_R</a>	USERS	NO	0	Apr 11, 2007 4:00:08 AM CEST
<input type="radio"/>	ELISABETTA	<a href="#">MDT_RT_MAP_S</a>	USERS	NO	0	Apr 11, 2007 4:00:07 AM CEST
<input type="radio"/>	ELISABETTA	<a href="#">MDT_RT_MAP_T</a>	USERS	NO	0	Apr 11, 2007 4:00:07 AM CEST
<input type="radio"/>	ELISABETTA	<a href="#">MDT_TUBE</a>	USERS	NO	2155	Feb 1, 2007 1:11:26 PM CET
<input type="radio"/>	ELISABETTA	<a href="#">MDT_TUBE_HEAD</a>	USERS	NO	123	Feb 1, 2007 1:11:27 PM CET
<input type="radio"/>	ELISABETTA	<a href="#">SITES</a>	EXAMPLE	NO	3	Feb 1, 2007 1:11:27 PM CET
<input type="radio"/>	ELISABETTA	<a href="#">TEMPORANEA</a>	EXAMPLE	NO	26	Apr 12, 2007 4:00:07 AM CEST

23/04/07 61

# Sommario

1. Database relazionali
2. Architettura di un database Oracle
3. Un esempio: il database delle calibrazioni delle camere MDT dell'esperimento Atlas

# L'architettura del database per la calibrazione delle Monitored Drift Tube Chambers di Atlas

- Dal secondo livello di trigger vengono estratte delle stream di dati opportuni per la calibrazione delle MDT.
- Questi dati vengono inviati nei Tier2 centri di calibrazione che, utilizzando anche i dati immagazzinati nel Conditions DB (replicati nei centri via Oracle Streams) potranno effettuare le calibrazioni.
- Il processo di calibrazione e' composto di diversi step, compresi dei controlli di validazione.
- Tutte le informazioni ottenute in tutti gli step intermedi verranno immagazzinate nel Calibration DB e replicate nel Calibration DB del Cern (Oracle streams).
- Le costanti di calibrazione che hanno passato il processo di validazione verranno immagazzinate nel Conditions Db del Cern.

# MDT Calibration Database Schemas

- MICHIGAN\_ATLAS in the MICHIGAN DB
- MUNICH\_ATLAS in the MUNICH DB
- ROMA1\_ATLAS in the ROMA1 DB
- CERN\_ATLAS in the CERN DB

The three schemas will merge into the CERN\_ATLAS schema of CERN DB. Updates will be replicated with the Oracle Streams technology:

- Oracle Streams enables the propagation and management of data, transactions and events in a data stream from one database to another



# MDT Calibration Database Tables

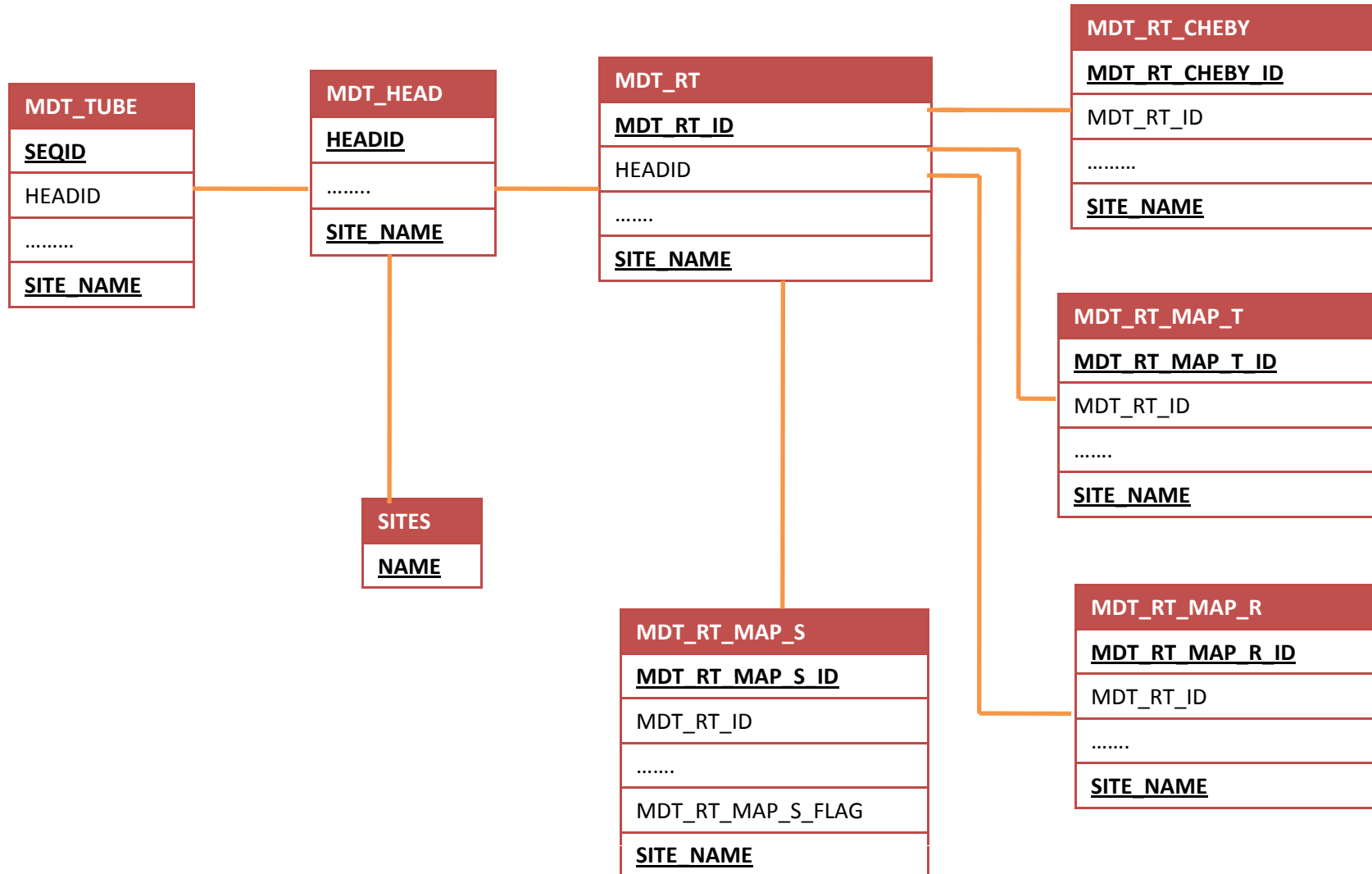
Each schema will hold the following tables

- MDT\_HEAD: table of the calibrations.
  - It contains the calibration date, the site name (Roma, Michigan, Munich), the total amount of data used, the link to the histogram (and ntuple) files. It is used as a header, to identify a single "calibration job", i.e. about a day of normal data-taking.
- MDT\_TUBE: table of the calibration constants relative to a single tube.
  - It contains the " $t_0$ " (measured time for a muon which hits the tube wire), the parameters of the fit to the single tube time spectrum, useful for DQ estimates and efficiencies, and the parameter of the ADC (charge collected by the wire) spectrum, necessary to equalize the ADC of different tubes. It contains also the full correlation matrix on these quantities.

# MDT Calibration Database Tables

- MDT\_RT: this table contains information on the "r-t relation" (from measured drift time to position in space) of one calibration region, roughly equivalent to one (or half) single chamber.
  - It contains the amount of events used, some parameter of the calib. region (temperature, mag. field, hv, ...) and the data quality of the calibration region. The rt function itself, together with its ancillary functions (its inverse  $t \rightarrow r$ , the resolution as a function of  $r$ ), may be stored in different formats and resides in the following tables. Therefore, for each calibration region, at least one of the following two tables must be present.
- MDT\_RT\_CHEBY: the  $rt$ ,  $tr$  and resolution, parametrized as a sum of Chebychev polynomials.
- MDT\_RT\_MAP\_R, MDT\_RT\_MAP\_S, MDT\_RT\_MAP\_T: the  $rt$ ,  $tr$  and resolution, parametrized as a map of points.

# MDT Calibration Database Tables



# How to realize the merge

Merging the three databases in one DB at Cern

- Using the SITES table, i.e. the initials of the calibration sites
- Using the local DBs tables primary key, composed by the fields
- SITE\_NAME referencing the field NAME in the SITES table
  - it guarantees the uniqueness of the primary key when tables merge into the Cern DB schema tables
- *primary\_key\_id* is automatically generated by a sequence at any insert.
  - it guarantee the uniqueness of the primary key in the tables of the local DB

# MDT\_HEAD Table

MDT_HEAD
<u>HEADID</u>
IMPLEMENTATION
LOWRUN
UPRUN
LOWTIME
UPTIME
LUMINOSITY
ROOTFILE
INSERT_TIME
<u>SITE_NAME</u>

## MDT\_HEAD table constraints

- **Primary Key:** (HEADID,SITE\_NAME)
- **Foreign KEY:** SITE\_NAME References SITES.NAME
- **CHECK:** UPRUN>=LOWRUN
- **CHECK:** UPTIME>=LOWTIME
- HEADID **Unique**, filled with a **SEQUENCE**

# INSERT into MDT\_HEAD

- INSERT into MDT\_HEAD values of fields: IMPLEMENTATION, LOWRUN, UPRUN, LOWTIME, UPTIME, LUMINOSITY, ROOTFILE, SITE\_NAME
- MDT\_TUBE\_HEAD\_TRIGGER trigger: is executed BEFORE an insert into MDT\_TUBE\_HEAD and passes to HEADID the NEXT-VALUE of the sequence MDT\_TUBE\_HEAD\_SEQUENCE;
- MDT\_TUBE\_HEAD\_SEQUENCE sequence: is used to generate unique values for MDT\_TUBE\_HEAD.SEQID
  - In this way the uniqueness of HEADID in the table is guaranteed.
  - Moreover the trigger insert in the insert\_time the value:  
TO\_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH:MI AM')

The same procedure is used to insert values in the other tables

# MDT\_RT table

MDT_RT
<u>MDT_RT_ID</u>
REGION_ID
MDT_HEAD_RT_ID
TUBE_ID
LOWRUN
UPRUN
HISTOGRAM
N_SEGS
AVER_ANGLE
.....
CURRENT
GAS
CALIBFLAG
VALIDFLAG
INSERT_TIME.
<u>SITE_NAME</u>

## MDT\_RT table constraints

- Primary Key: (MDT\_RT\_ID, SITE\_NAME)
- Foreign Key:  
(MDT\_HEAD\_RT\_ID, SITE\_NAME) References  
MDT\_HEAD.(MDT\_HEAD\_RT\_ID, SITE\_NAME)  
1 row in MDT\_TUBE\_HEAD -> MANY rows in  
MDT\_RT
- CHECK: UPRUN >= LOWRUN
- MDT\_RT\_ID filled with a SEQUENCE

# Data Size

- First (pessimistic) estimate of data size per day in the calibration centers:
  - 60 words/tube (→ 100 MB if no compression)
  - 400 words/rt (→ 20 MB if no compression)
  - Will be easily reduced once we know what we are doing...
- More data in the Cern Calibration database (long history)



# Queries and Views

- Materialized views have been created to improve the most common queries execution.
- These materialized views are refreshed at the end of any calibration running a procedure.
- Query example: select all fields of mdt\_tube table and the rootfile from the matching mdt\_head table for the last insert time into the mdt\_tube

```
select A.*, B.ROOTFILE from  
(select * from roma1_atlas.mdt_tube where insert_time in (select  
insert_time from (select distinct insert_time from roma1_atlas.mdt_tube  
order by insert_time desc) where rownum = '1')) A  
INNER JOIN roma1_atlas.mdt_head B  
ON A.HEADID = B.HEADID;
```