

# Sistemi per la Gestione delle Basi di Dati

## Esercitazione 2 – Ottimizzatore di Oracle – indici e hint

### Obiettivo dell'esercitazione

Calcolare il piano di esecuzione per alcune query SQL analizzando i seguenti aspetti:

1. metodo di accesso alle tabelle
2. metodo di join
3. ordine in cui vengono eseguite le operazioni
4. uso di indici definiti dall'utente.

Lo studio verrà effettuato con Oracle Database 10g Express Edition (Oracle XE).

### Struttura della base di dati

La base di dati di riferimento è composta da 3 tabelle (EMP, DEPT e SALGRADE). In seguito viene riportato lo schema delle tabelle ed alcuni record di esempio. Il contenuto delle tabelle riportato è solo di esempio, in quanto nella realtà la base di dati è composta da un numero elevato di record.

Tabella **EMP**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT NO
7369	LUIGI	SEGRETARIO	7902	17-DEC-80	800		20
7839	MONICA	PRESIDENTE		17-NOV-81	5000		10
7844	SANTE	SALESMAN	7698	08-SEP-81	1500	0	30
7876	DIEGO	DOTTORANDO	7788	23-MAY-87	1100		20
7900	ELENA	INGEGNERE	7698	03-DEC-81	950		30
7902	SILVIA	AVVOCATO	7566	03-DEC-81	3000		20
7934	PAOLO	SICUREZZA	7782	23-JAN-82	1300		10

Tabella **DEPT**

DEPTNO	DNAME	LOC
10	ACCOUNTING	TARANTO
20	RESEARCH	TORINO
30	SALES	CHIETI
40	OPERATIONS	ROMA

Tabella **SALGRADE**

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000

Passi preliminari per lo svolgimento dell'esercitazione

### Connessione alla base di dati

Connessione via browser Web all'indirizzo <http://130.192.37.58:8080/apex>

### Login

Per autenticarsi collegandosi all'interfaccia Web, inserire i seguenti parametri.

- Nome utente: bdati[scegliere un valore compreso tra 1-100]
- Password: orac[scegliere un valore compreso tra 1-100]

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



Una volta eseguito il login verrà visualizzata una finestra indicante il seguente menu.



## Materiale disponibile

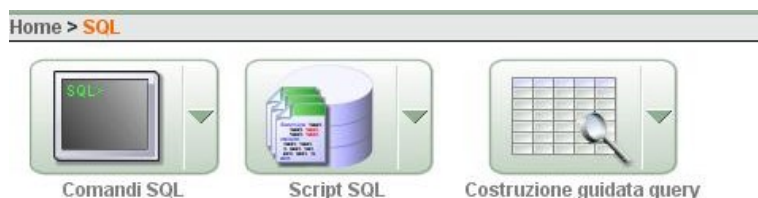
Sono disponibili alcuni script contenenti istruzioni SQL per svolgere le seguenti operazioni:

1. creare un indice su un campo della base di dati
2. calcolare le statistiche per la base di dati

Gli script sono disponibili:

- sul sito web del corso, nell'archivio Script1.zip
- <http://dbdmq.polito.it/twiki/bin/view/Public/SistemiDiGestioneDiBasiDati>

Gli script possono essere caricati nell'interfaccia Web selezionando l'icona "SQL" dal menu principale e successivamente la freccia dell'icona "Script SQL" indicata nella seguente figura.



Cliccare sulla freccia a fianco l'icona di "Script SQL" e selezionare la voce carica per caricare lo script. Per eseguire gli script con **Firefox**, impostare la vista "Dettagli" e cliccare sul pulsante "Esegui" relativo allo script da eseguire.



## Definizione dell'ambiente di analisi

All'inizio della sessione di lavoro è necessario svolgere i seguenti passi:

1. calcolare le statistiche delle tabelle tramite l'interfaccia Web o usando lo script `comp_statistics_tables.sql`
2. controllare che non siano presenti altri indici oltre a quelli di sistema (il cui nome inizia con `sys`), eseguendo la seguente query  
`select INDEX_NAME from USER_INDEXES;`  
ed eventualmente cancellare gli indici non di sistema che influenzano le tabelle EMP, DEPT e SALGRADE eseguendo il seguente comando  
`DROP INDEX nomeindice;`

## Calcolo del piano di esecuzione per una query

Per ottenere il piano di esecuzione di una data query nell'interfaccia **Web** è sufficiente accedere alla sezione Home>SQL>Comandi SQL, indicata in Fig.3:

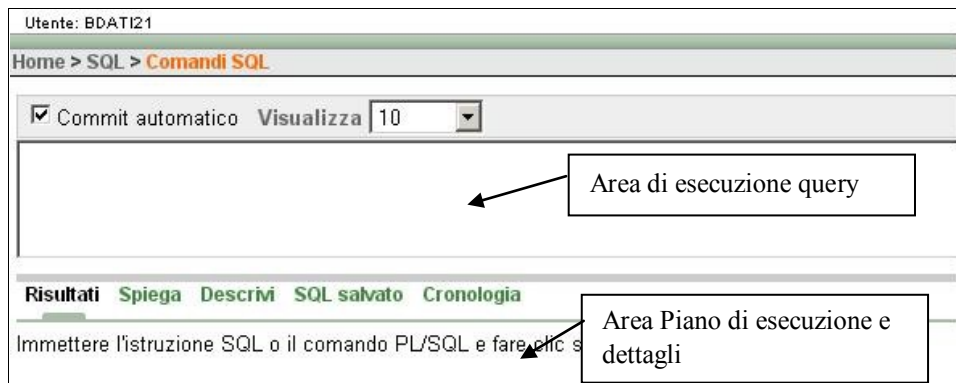


Fig.3

Eseguire la query e selezionare nella barra in basso la scheda "Spiega" per visualizzare il piano di esecuzione della query.

## Comandi utili

- Per leggere quali campi compongono una tabella:  
`DESCRIBE NomeTabella;`
- Per creare un indice su un campo di una tabella:  
`CREATE INDEX NomeIndice ON NomeTabella (NomeCampo);`
- Per aggiornare le statistiche relative ad un indice esistente:  
`ANALYZE INDEX NomeIndice COMPUTE STATISTICS;`
- Per rimuovere un indice:  
`DROP INDEX NomeIndice;`
- Per visualizzare l'elenco degli indici relativi ad una tabella:  
`SELECT INDEX_NAME FROM USER_INDEXES  
WHERE table_name='nome tabella in maiuscolo';`
- Per visualizzare le statistiche relative agli indici:  
`SELECT USER_INDEXES.INDEX_NAME as INDEX_NAME, INDEX_TYPE,  
USER_INDEXES.TABLE_NAME, COLUMN_NAME||'('||COLUMN_POSITION||')' as  
COLUMN_NAME, BLEVEL, LEAF_BLOCKS, DISTINCT_KEYS, AVG_LEAF_BLOCKS_PER_KEY,  
AVG_DATA_BLOCKS_PER_KEY, CLUSTERING_FACTOR  
FROM user_indexes, user_ind_columns  
WHERE user_indexes.index_name=user_ind_columns.index_name and  
user_indexes.table_name=user_ind_columns.table_name;`

- Per visualizzare le statistiche relative alle tabelle:  

```
SELECT TABLE_NAME, NUM_ROWS, BLOCKS, EMPTY_BLOCKS, AVG_SPACE, CHAIN_CNT,
AVG_ROW_LEN
FROM USER_TABLES;
```
- Per visualizzare le statistiche sugli attributi delle tabelle:  

```
SELECT COLUMN_NAME, NUM_DISTINCT, NUM_NULLS, NUM_BUCKETS, DENSITY
FROM USER_TAB_COL_STATISTICS
WHERE TABLE_NAME = 'NomeTabella' ORDER BY COLUMN_NAME;
```
- Per visualizzare gli istogrammi:
  - Istogrammi di tipo Height-balanced  

```
SELECT ENDPOINT_NUMBER, ENDPOINT_VALUE
FROM USER_HISTOGRAMS
WHERE TABLE_NAME='NomeTabella' AND COLUMN_NAME='NomeColonna'
ORDER BY endpoint_number;
```
  - Istogrammi di tipo Frequency  

```
SELECT ENDPOINT_ACTUAL_VALUE, ENDPOINT_NUMBER
FROM USER_HISTOGRAMS
WHERE table_name='NomeTabella' and column_name='NomeColonna'
ORDER BY endpoint_number;
```

Esempio per visualizzare gli istogrammi per il campo JOB della tabella EMP:

```
SELECT ENDPOINT_ACTUAL_VALUE, ENDPOINT_NUMBER
FROM USER_HISTOGRAMS
WHERE TABLE_NAME='EMP' and COLUMN_NAME='JOB'
ORDER BY ENDPOINT_NUMBER;
```

## Query da analizzare

Le seguenti query dovranno essere analizzate durante l'esercitazione.

Per ogni query analizzata, svolgere i seguenti passi:

1. scrivere l'espressione algebrica ad albero e il piano di esecuzione senza strutture secondarie
2. creare degli indici e valutare degli hint per ottimizzare l'esecuzione della query
3. analizzare i piani di esecuzione di Oracle, sia senza sia con le strutture secondarie create

## Query #1 and #2

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione contemporanea di **entrambe** le seguenti query:

```
select avg(e.sal)
from emp e
where e.deptno < 10 and
e.sal > 100 and e.sal < 200;
```

```
select e.sal, s.grade
from emp e, salgrade s
where e.sal > s.losal;
```

### Query #3

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query:

```
select dname
from dept
where deptno in (select deptno
                  from emp
                  where job = 'FILOSOFO');
```

### Query #4

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query:

```
select e1.ename, e1.empno, e1.sal, e2.ename, e2.empno, e2.sal
from emp e1, emp e2
where e1.ename <> e2.ename and e1.sal < e2.sal
and e1.job = 'FILOSOFO' and e2.job = 'INGEGNERE';
```

### Query #5

Oltre alla creazione di una o più strutture secondarie (indici), si provino a valutare degli hint per la seguente query:

```
select e.ename, e.sal, s.grade
from emp e, salgrade s
where e.sal > s.losal and e.deptno > 506 and s.hisal < 900;
```

### Query #6

Oltre alla creazione di una o più strutture secondarie (indici), si provino a valutare degli hint per la seguente query:

```
select *
from emp e, dept d
where e.deptno = d.deptno
and not exists ( select * from salgrade s
                 where e.sal = s.hisal and hisal > 500 and hisal < 1900);
```