

Sistemi per la Gestione delle Basi di Dati

Esercitazione 1 – Ottimizzatore di Oracle

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.

Table **EMP**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT NO
1	COZZA MARIA	PROFESSOR	0	09-JUN-81	1181		126
2	ECO LUIGI	PHDSTUDENT	0	09-JUN-81	1360		189
3	CORONA CLARA	PHDSTUDENT	2	09-JUN-81	624		15

Table **DEPT**

DEPTNO	DNAME	LOC
1	INFORMATION	BARI
2	CHAIRMANSHIP	FOGGIA
3	ENVIRONMENT	BRINDISI
4	PHYSICS	FOGGIA

Table **SALGRADE**

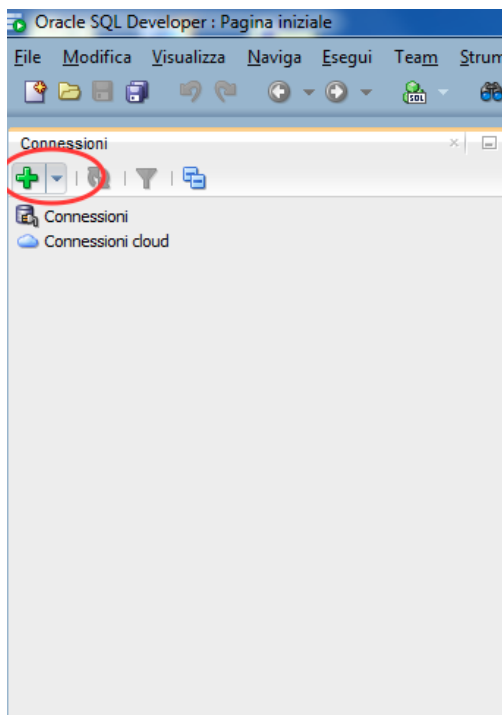
GRADE	LOSAL	HISAL
1	478	1503
2	661	1346
3	489	1358
4	942	1320

Passi preliminari per lo svolgimento dell'esercitazione

Connessione alla base di dati

Aprire il programma Oracle SQL Developer

Cliccare su crea nuova connessione:

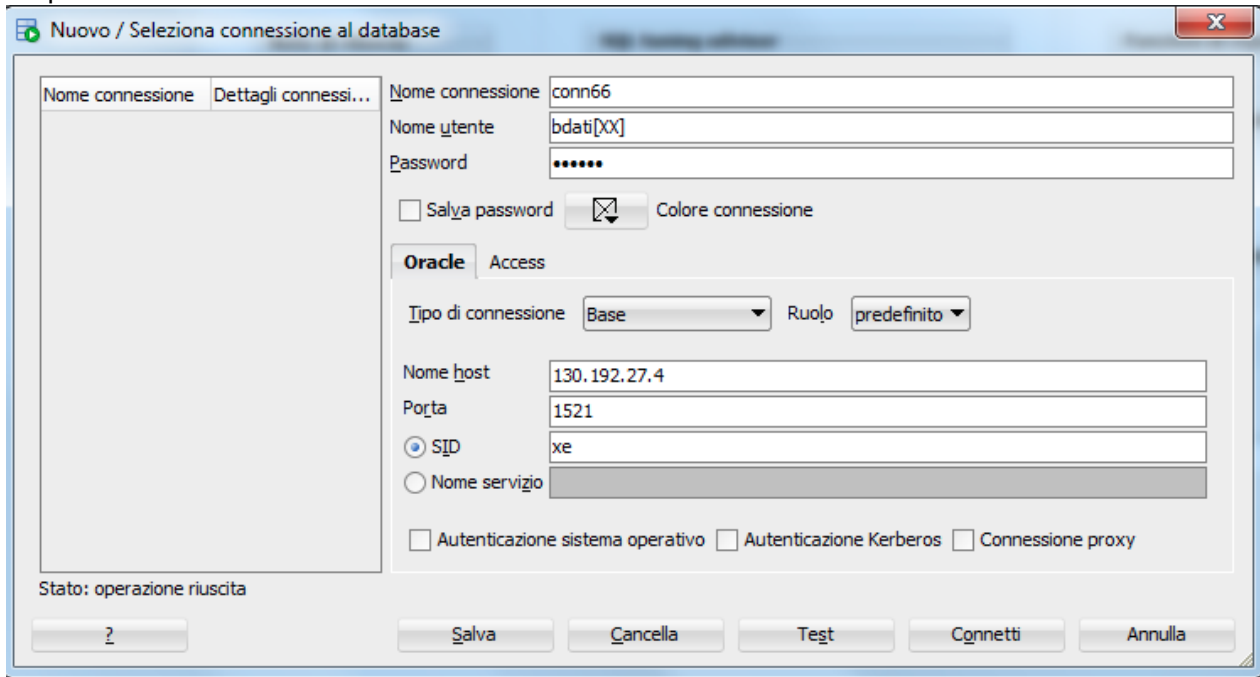


Login

Per autenticarsi inserire i seguenti parametri.

- Nome utente: bdati[scegliere un valore compreso tra 1-100]
- Password: orac[scegliere un valore compreso tra 1-100]
- Nome host: 130.192.27.4
- Porta: 1521
- SID: xe

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



Materiale disponibile

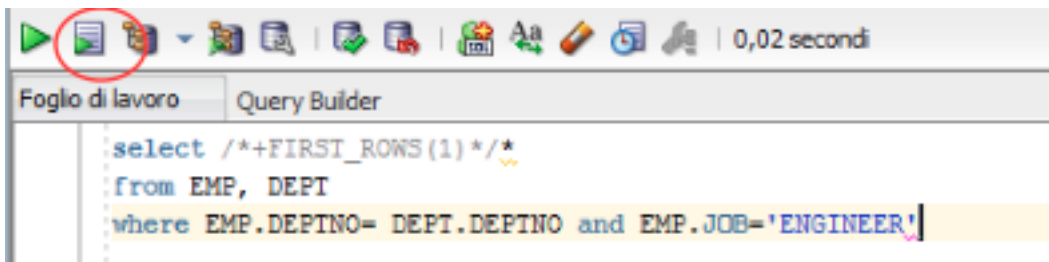
Sono disponibili alcuni script contenenti istruzioni SQL per svolgere le seguenti operazioni:

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://dbdmg.polito.it/wordpress/teaching/sistemi-per-la-gestione-di-basi-di-dati/>

Gli script possono essere caricati aprendo File->Apri e selezionando il file `.sql` e successivamente cliccando il pulsante "Esegui Script"



Per visualizzare le statistiche sugli indici, eseguire lo script `show_indexes.sql` (oppure copiare il contenuto dello script e incollarlo come comando SQL).

Definizione dell'ambiente di analisi

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

1. calcolare le statistiche delle tabelle tramite interfaccia sql 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 è necessario **eseguire la query e successivamente** cliccare il pulsante "Piano di esecuzione" indicato in figura



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:
`SELECT *
FROM USER_HISTOGRAMS;`

Query da analizzare

Le seguenti query dovranno essere analizzate durante l'esercitazione, eseguendo i passi:

1. espressione algebrica ad albero della query
2. piano di esecuzione di Oracle della query originale senza strutture secondarie
3. Per le query da #4 a #7, scegliere una o più strutture fisiche accessorie per migliorare le prestazioni dell'interrogazione.

Richiamo alla struttura delle tabelle

EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)

DEPT (DEPTNO, DNAME, LOC)

SALGRADE (GRADE, LOSAL, HISAL)

Query #1

```
SELECT *
FROM emp, dept
WHERE emp.deptno = dept.deptno AND emp.job = 'ENGINEER';
```

Cambiare l'obiettivo di ottimizzazione dalla modalità ALL ROWS (best throughput) alla modalità FIRST_ROWS (best response time) attraverso l'uso di hint (/**+ FIRST_ROWS (n) */). n è una variabile numerica intera che può assumere valori maggiori o uguali a 1. Assegnare diversi valori ad n e verificare come variano il piano d'esecuzione e i costi delle diverse operazioni.

```
SELECT /**+ FIRST_ROWS (n) */ *
FROM emp, dept
WHERE emp.deptno = dept.deptno AND emp.job = 'ENGINEER';
```

Query #2

Confrontare i costi di hash join e nested loop usando l'hint USE / NO_USE_HASH

```
SELECT /**+ NO_USE_HASH(e d) */ d.deptno, AVG(e.sal)
FROM emp e, dept d
WHERE d.deptno = e.deptno
GROUP BY d.deptno;
```

Query #3

Disabilitare il metodo hash join mediante l'uso di hint (/**+ NO_USE_HASH(e d) */)

```
SELECT /**+ NO_USE_HASH(e d) */ ename, job, sal, dname
FROM emp e, dept d
WHERE e.deptno = d.deptno
AND NOT EXISTS
    (SELECT * FROM salgrade WHERE e.sal = hisal);
```

Query #4

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query. Si analizzi con particolare attenzione il cambiamento nel piano di esecuzione creando due indici sugli attributi interessati dall'interrogazione.

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

Query #5

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 = 'PHILOSOPHER');
```

Query #6

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query (rimuovere eventuali indici già esistenti per confrontare le performance della query con e senza indici):

```
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 = 'PHILOSOPHER' and e2.job = 'ENGINEER';
```