FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Mestrado em Engenharia Informática

Database Technology

SQL QUERIES OPTIMIZATION

Documentation : http://www.oracle.com
See books: Performance Tuning Guide and Reference; Database Tuning with the Oracle Tuning Pack

1. Some examples:

a) Full Table Scan

```
SELECT DEPTNO FROM EMP;
```

b) Table Access by Rowid

```
SELECT * FROM EMP WHERE EMPNO = 7900;
```

c) Range Scan (create an index on column job of table emp)

```
CREATE INDEX EMP_JOB_DX ON EMP (JOB);
SELECT ENAME FROM EMP WHERE JOB='CLERK';
```

d) Merge indexes

```
SELECT *
FROM EMP
WHERE JOB='ANALYST'
AND DEPTNO = 20;
```

2. Operators IN, EXISTS, ALL, ANY

Which are the names and numbers of the departments with no employees?

a) use NOT IN
b) use NOT EXISTS
c) use ALL
d) use ANY

3. Operators NOT IN - NOT EXISTS

Change the optimizer options to FIRST_ROWS_10

```
ALTER SESSION SET OPTIMIZER_MODE = ' FIRST_ROWS_10'
```

3.1. Repeat 1a) and 1b). Did you notice any difference?

3.2. Create now an index on column deptno of table emp

Repeat again a) and b)

3.3. Did you get any differences? Which are your conclusions?

3.4. Check the plan for the other optimizer modes.

3.5. Drop the index on column Deptno of table emp.

4. Distinct operator

Using **distinct** usually requires a sort. All the data must be retrieved before returning the result.

4.1. Create two similar queries, one with distinct and the other without it.

5. Transitivity:

Select the name of the department and the number and name of the employees that belong to the department 20. Is there any difference in using condition emp.deptno = 20 instead dept.deptno = 20?

5.1. Create a table exp with a single column num number(2). Insert in the table the number 20. Answer again question 5, but now using exp.num instead of 20. Do you notice any difference when using emp.deptno or dept.deptno?

6. Transforming OR into composite queries

Create two indexes on columns deptno and job of table emp.

6.1. Select all the information on the employees of department 10 or with job='CLERK'.

6.2. Rewrite the previous query using UNION ALL.

6.4. If you delete one index, do you get any differences?

6.5 Check now changing the where condition by adding (OR 1<>1). What do you conclude?

7. Difference between union and union all and intersect

Check the differences in the execution plans of the following SQL queries.

```
a) SELECT DEPTNO FROM EMP
   UNION ALL
   SELECT DEPTNO FROM DEPT;

b) SELECT DEPTNO FROM EMP
   UNION
   SELECT DEPTNO FROM DEPT;

c) SELECT DEPTNO FROM EMP
   INTERSECT
   SELECT DEPTNO FROM DEPT;
```

Which are your conclusions?

8. Operators max, min, avg and order by

a) Which are the numbers of the employees?

b) Which is the highest salary of all employees?

c) Which are the salaries of the employees by ascending order?

d) Which is the average salary?

e) Draw your conclusions from using these operators. Which are d«the differences between a) and b)?.

9. Transporting the view into the query

9.1. Check the execution plan for the two possible ways of answering the same query:

```
a)        CREATE VIEW EMP_10
            AS SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO
                     FROM EMP
                     WHERE DEPTNO = 10;
```

```
SELECT EMPNO
FROM EMP_10
WHERE EMPNO > 7800;
```

b)
```
SELECT EMPNO
FROM EMP
WHERE DEPTNO = 10
AND EMPNO > 7800;
```

What can you conclude from a) and b)?

9.2. Check the execution plan for the two following queries:

a)
```
CREATE VIEW AVG_SALARY_VIEW AS
   SELECT DEPTNO, AVG(SAL) AS AVG_SAL_DEPT
   FROM EMP
   GROUP BY DEPTNO;

SELECT DEPT.LOC, AVG_SAL_DEPT
    FROM DEPT, AVG_SALARY_VIEW
    WHERE DEPT.DEPTNO = AVG_SALARY_VIEW. DEPTNO
          AND DEPT.LOC = 'London';
```

b)
```
SELECT DEPT.LOC, AVG(SAL)
   FROM DEPT, EMP
   WHERE DEPT.DEPTNO = EMP.DEPTNO
          AND DEPT.LOC = 'London'
    GROUP BY DEPT.ROWID, DEPT.LOC;
```

Which are your conclusions from a) and b)? And with respect to 8.1.?

10. Generating statistics

Statistics must be up-to-date in order to be useful for the optimizer. Execute the following query:

```
SELECT INDEX_NAME, DISTINCT_KEYS FROM ALL_INDEXES WHERE OWNER = '<your user>';
```

Gather statistics using

```
EXECUTE DBMS_STATS.GATHER_INDEX_STATS('<your user>','EMP_PRIMARY_KEY');
```

Execute again the previous query.