Query Performance Analysis Using Explain Plan and Indexes

Instructions
You will write a SQL script analyzes the queries below and adds appropriate indexes.

Pre-requisite:

1. You must be able to develop your script with access to an Oracle instance, preferably from data.cs.purdue.edu.
2. Transfer the following files to your folder on data.cs.purdue.edu using scp:
    1. create_employee.sql
    2. drop_employee.sql
    3. add_more_data.sql

Objective:

For each query listed below, you must assess the existing SQL query and explain plan, apply an index which alters the plan, and explain your decision and the outcome.

Tasks:

1. Run the create_employee.sql script to populate the necessary tables.
    1. Once you complete the assignment or you need to clean up, you can run drop_employee.sql
2. Run the add_more_data.sql scrip to add data to employee and employee_project.
3. Demonstrate your understanding of SQL query plans and indexes by completing the following steps for each query documented later in the

assignment:

1. Paste the original explain plan for the query.
2. Create an index which alters the explain plan.
    1. You won't know whether the explain plan was altered unless you test. Create your new index, re-evaluate the explain plan. If your explain plan contains the created index, move on to the next step. If it is not included, then drop that index and create a new one. There is a limited number of columns to include in an index, so keep trying until you find an index which is used by the query.
    2. **All queries are expected to have at least one useful index. If you do not find an index which improves the performance, you may receive partial credit by documenting what you did try an hypothesize why it didn't help.**
3. Using your understanding of indexes and query plans, provide a short 3-5 sentence explanation of why the index helps. You should use knowledge gained from the reading and the videos.
4. Drop the index you created.

For each query and index, your submission should look something like this in your SQL script file.

```
-- QUERY 0
EXPLAIN PLAN FOR SELECT * FROM employee e where
employee_name = 'Daisy';
SELECT plan_table_output FROM
TABLE(DBMS_XPLAN.DISPLAY('plan_table',null,''));
-----------------------------------------------------
---------------------------
--| Id  | Operation         | Name     | Rows  |
Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------
---------------------------
```

```
--|   0 | SELECT STATEMENT  |          |     1 |
 30 |    15   (0)| 00:00:01 |
--|*  1 |  TABLE ACCESS FULL| EMPLOYEE |     1 |
 30 |    15   (0)| 00:00:01 |
-----------------------------------------------------
---------------------------
CREATE INDEX idx_employee_name ON employee
(employee_name);
EXPLAIN PLAN FOR SELECT * FROM employee e where
employee_name = 'Daisy';
SELECT plan_table_output FROM
TABLE(DBMS_XPLAN.DISPLAY('plan_table',null,''));
-----------------------------------------------------
-----------------------------------------------------
-
--| Id  | Operation                           | Name
             | Rows  | Bytes | Cost (%CPU)| Time
|
-----------------------------------------------------
-----------------------------------------------------
-
--|   0 | SELECT STATEMENT                    |
            |     1 |    30 |     2   (0)| 00:00:01 |
--|   1 |  TABLE ACCESS BY INDEX ROWID BATCHED|
EMPLOYEE          |     1 |    30 |     2   (0)|
00:00:01 |
--|*  2 |   INDEX RANGE SCAN                  |
IDX_EMPLOYEE_NAME |     1 |       |     1   (0)|
00:00:01 |
-----------------------------------------------------
-----------------------------------------------------
-
--In the example above, the original explain plan was
```

```
accessing every record in the employee table, as
--evidenced by the TABLE ACCESS FULL. Because there
was a filter on employee_name, I added an index for
--that column. This resulted in a new plan which uses
the index to find the value 'Daisy' in the employee
--table. Because it is an INDEX RANGE SCAN, it finds
'Daisy' (and any subsequent 'Daisy's if they exist),
--ultimately looking them up again in EMPLOYEE by
ROWID.
DROP INDEX idx_employee_name;
```

Queries:

--QUERY 1
EXPLAIN PLAN FOR SELECT project_id FROM employee_project where employee_id = 738;
SELECT plan_table_output FROM TABLE(DBMS_XPLAN.DISPLAY('plan_table',null,''));

--QUERY 2
EXPLAIN PLAN FOR SELECT e.employee_name, d.department_name FROM employee e inner join department d on e.department_id = d.department_id WHERE e.employee_name = 'Daisy';
SELECT plan_table_output FROM TABLE(DBMS_XPLAN.DISPLAY('plan_table',null,''));

--QUERY 3
EXPLAIN PLAN FOR SELECT MIN(hired_date) FROM employee e inner join department d on e.department_id = d.department_id WHERE d.department_name = 'Finance';
SELECT plan_table_output FROM TABLE(DBMS_XPLAN.DISPLAY('plan_table',null,''));

Submitting your work:

1. All work must be submitted via a single SQL script. The file name should be in the format "CSCI311-Project4-last_name-first-name.sql"

where last_name and first_name are your actual names.
2. Your script should run without error from beginning to end and produce the desired result sets. It is only required to include content like the example above (you may omit the original DDL and DML).
3. Upload your script to this assignment in Brightspace.

Grading:

Criteria	Points
Adheres to submission instructions and template	10
Queries (3 queries, 30 points each)	90
TOTAL	100

IMPORTANT ADVICE:

- Work to format your code nicely. Code which is poorly formatted may run, but is definitely harder to read and graders may not be able to identify all of your hard work.
- Even if you don't have a full solution, submit what you do have. Some points are better than nonw.
- Upload your submission before the due date. You don't want to miss points due to an issue with Oracle.

Submissions

CSCI311-Project4-Ali-Raja.... (3.11 KB) Oct 20, 2024 7:01 PM

Drop files here, or click below!

Upload Record Choose Existing

You can upload files up to a maximum of 2.86 GB.

Comments

Paragraph

Feedback

90 / 100 **90 %**

-10(query plan results not included in submission)

Last Modified Sunday, October 27, 2024 8:49 PM EDT

Activity Details

Well done! You have completed the assignment

Due October 21 at 11:59 PM

Assessment

90 / 100 **90 %**

Last Visited Nov 12, 2024 10:26 AM