Project – Milestone #3
Query Analysis and Tuning for University Library Management System

Introduction

As the final step of the term project in this class, you should analyze and tune slow queries in your application. We have learnt a way to estimate the query costs of some evaluation plans. In this project, you will check the query plans and costs for the SQL queries in your implementation.

Data generator

To analyze the query, you need large enough data for the tables in your application; accordingly, you should implement a data generator program to generate large data for your tables. You can implement in any language such as C, Java, Perl, Python, and so on.

See *tableGen.java* in our textbook homepage for your reference.
(http://codex.cs.yale.edu/avi/db-book/db6/lab-dir/sample_tables-dir/index.html)

Additional commands

Add the following commands to your implementation. It is expected that you can see various types of query plans from these commands.

Ordinary user menu
7. display monthly check-out history.
- input: year(number), month(number)
- action: list list id, title, check-out date, return date of resources which were borrowed by the current user at that month
- error: year <= current year, 1 <= month <= 12

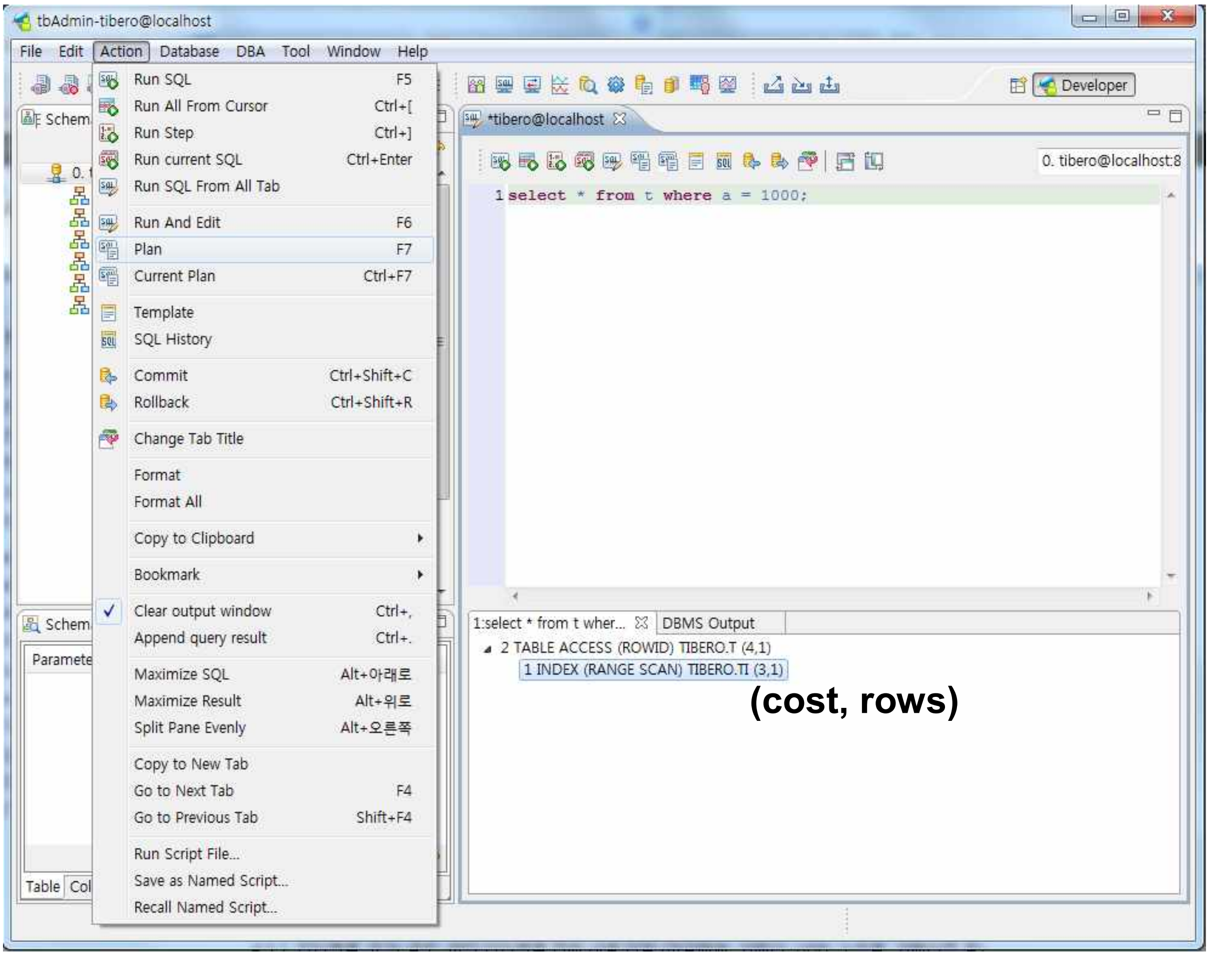
Administrative user
5. list top 100 most borrowed resources of all time.
- input: no input
- action: list id, type, title, first author, publisher, year of the top 100 most borrowed resources of all time.

6. list top 20 most borrowed resources of the specified period.
- input: start year(number), start month(number), end year(number), end month(number)
- action: list id, type, title, first author, publisher, year of the top 20 most borrowed resources during the specified period.
- error: start year <= end year, end year <= current year,
start month <= end month, 1 <= start/end month <= 12

If you did not implement the application in Milestone #2, just write and run the queries that corresponds to the features. Then, you will receive partial credits.

Query Analysis and Optimization

1. Load generated large enough data into Tibero RDBMS, and see query evaluation plans of all your SQL queries in your implementation. You can see the plans with the following ways:
- tbAdmin: Action > Plan(F7) or Current Plan(Ctrl+F7)

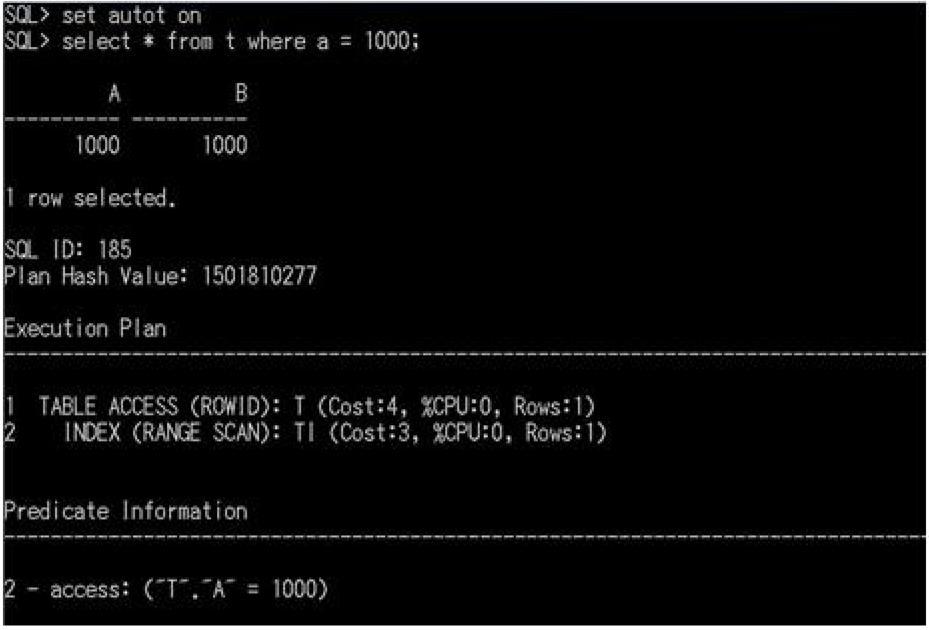


```
select * from t where a = 1000;
```

```
2 TABLE ACCESS (ROWID) TIBERO.T (4,1)
  1 INDEX (RANGE SCAN) TIBERO.TI (3,1)
```

(cost, rows)

- tbsql: set autot(race) on/traceonly/off, set timing on
(traceonly: omits query results and displays trace information only)



```
SQL> set autot on
SQL> select * from t where a = 1000;

         A          B
---------- ----------
      1000       1000

1 row selected.

SQL ID: 185
Plan Hash Value: 1501810277

Execution Plan
--------------------------------------------------------------------------------

1  TABLE ACCESS (ROWID): T (Cost:4, %CPU:0, Rows:1)
2    INDEX (RANGE SCAN): TI (Cost:3, %CPU:0, Rows:1)


Predicate Information
--------------------------------------------------------------------------------

2 - access: ("T"."A" = 1000)
```

2. Check the plans and actual execution time, and tune slow queries by creating indices or revising the queries. (Note: Tibero RDBMS automatically creates an index on primary key columns to check for unique values.)

3. Classify your queries according to the execution plan types, and explain the execution plans and the tuning process.

Submit

1. Application and data generator program files
- source files, binary files, build files(makefile, ant, or batch script)
- A SQL file to create tables, integrity constraints, and indices in Tibero RDBMS
- Generated data files to insert sample data for your test

2. A document file containing the following:
- Explanation of the different types of the execution plans for your SQL queries
- Description on your query tuning (before & after)
- Discussion or thoughts on your query analysis and optimization
- Development environment
- How to compile and run

Please submit the files in .zip format with the filename [proj3_student-id].zip via e-mail:
- To: twlee@idb.snu.ac.kr
- Title: [DB2013s_proj3] your student-id, name
- Due date: 23:59pm, 17th June (late penalty: 10% per a day, no credit after a week)