



Database Systems

B.Sc. (Hons) in IT

Laboratory Worksheet 07

In this Lab Assignment, we will use Oracle's EXPLAIN PLAN to describe the query plans selected by Oracle.

EXPLAIN PLAN:

The EXPLAIN PLAN statement displays execution plans chosen by the Oracle optimizer for SELECT, UPDATE, INSERT, and DELETE statements. This statement inserts a row describing each step of the execution plan into a specified table (default PLAN_TABLE).

Note : You must create a table called PLAN_TABLE as specified by Oracle in order to use EXPLAIN PLAN. You may use the SQL script utlxplan.sql script to create the PLAN_TABLE.

E.g. @ Z:\utlxplan

In this Lab Assignment, you will use the following relational schema of practical 1:

Client (clno: char(3), name: varchar(12), address: varchar(30))

Stock (company: char(7), price: number(6,2), dividend: number(4,2), eps: number(4,2))

Trading (company: char(7), exchange: varchar(12))

Purch (clno: char(3), company: char(7), pdate: date, qty: number(6), price: number(6,2))

Use the SQL script SampleDB.sql to create these tables with the sample data of Practical 1, is provided as a support file on the moodle.

E.g. @ Z:\SampleDB

Consider the following query for the schema in practical 1:

Display the client number and name of clients who have made large purchases. A large purchase occurs when a client purchases more than 1000 shares from a company at any given time.

Alter the session as given below before you proceed to see the CPU cost and IO costs.

```
ALTER SESSION SET OPTIMIZER_MODE = ALL_ROWS;
ALTER SESSION SET "_optimizer_cost_model"=CPU;
```

- (a.) Use EXPLAIN PLAN to find Oracle's generated query plan for the above query by executing the following statement:

```
explain plan for
select c.clno, c.name
from client c, purch p
where c.clno = p.clno and p.qty > 1000;
```

Now you will find the expected plan in the plan table.

- (b.) You can execute the script `utlxpls.sql` to view the query plan and associated costs. Describe each step of the query plan.

E.g. @ Z:\utlxpls

Note that the output contains 3 columns for costs: COST, CPU_COST and IO_COST. In Oracle, the optimizer uses both cpu costs and I/O costs. The value of column COST is a function of the CPU_COST and IO_COST.

- (c.) Next, create indexes for client and purchase tables. In Oracle you can use the CREATE INDEX statement to create indexes. Below is some background on Oracle's indexes:

- Oracle's default index is an unclustered, B + -Tree and all primary keys are enforced with indexes on them
- Example of a index:

```
CREATE INDEX index_1 ON client(name);
```

This creates an unclustered B + -Tree index on client table with search key field name.

Using CREATE INDEX statement, create the following indexes:

- Unclustered B + -Tree index on purchase (qty, clno)
- Unclustered B + -Tree index on client (clno, name)

- (d.) Re-execute the explain plan in part (a). Use `utlxpls.sql` to view the query plan and associated costs. Explain the query plan and compare it with the query plan in part (b.)

Additional Notes:

- You can drop an index with the DROP INDEX statement.
- The USER_INDEXES table contains all information about indexes. You can use find indexes on a particular table as follows:

```
select index_name
from user_indexes
where table_name = 'CLIENT';
```

- You could also obtain DDL statements used to create indexes of a particular table by using DBMS_METADATA.GET_DDL() function as follows:

```
select DBMS_METADATA.GET_DDL('INDEX',u.index_name)
from user_indexes u
where table_name = 'CLIENT';
```