CSCI317 – Database Performance Tuning

Tutorial – Indexing

Sionggo Japit
sjapit@uow.edu.au

17 August 2022

Question

The following SQL statement has been used to create an index on a relational table CUSTOMER owned by a user CSCI317:

```
CREATE INDEX IDX_CUSTOMER ON
CUSTOMER(C_LNAME, C_EMAIL);
```

Give 3 **different** SELECT statements such that each statement is processed only by accessing the index and **NOT** by accessing a relational table CUSTOMER. This way of query processing is commonly called "index processing only", because a database systems does not plan to access a relational table to compute a query.

Make sure the queries really are **different**, e.g. one query could be a join query another query could be a nested query with correlation variables and yet another query could be an aggregation query with GROUP BY and HAVING clauses. Each query must retrieve different information.

Solution

(1)

```
SELECT C_LNAME, C_EMAIL FROM
CUSTOMER;
```

The system will horizontally scan a leaf level of B*-Tree that implements the index IDX_CUSTOMER.

Solution

(2)

```
SELECT COUNT( DISTINCT C_LNAME)
FROM CUSTOMER;
```

The system will horizontally scan a leaf level of B*-Tree that implements the index IDX_CUSTOMER and it will count the total number of distinct values of attribute C_LNAME.

Solution

(3)

```
SELECT C_LNAME, COUNT(*) FROM
CUSTOMER GROUP BY C_LNAME;
```

The system will horizontally scan a leaf level of B*-Tree that implements the index IDX_CUSTOMER. While scanning it will group the values of an attribute C_LNAME and it will count the total number of row identifiers associated with each distinct value of C_LNAME.