Indexing in database

The index on the database is a data structure technique which is used to quickly locate and access the data in a database by providing a method to quickly lookup the requested data without scan the whole table/all data to find out the result (full scan).

Indexes are a powerful tool used in the background of a database to speed up querying.

Structure of an Index in Database :

- **Search key** that contains a copy of the primary key or candidate key of the table. These values are stored in *sorted order* so that the corresponding data can be accessed quickly.
- **Data Reference** or **Pointer** which contains a set of pointers holding the address of the disk block where that particular key value can be found (the address on the memory disk of the row with the rest of the information).

I had experience using indexes in my previous company, for example on a Cargo Ready Container System project. Because the data in the database used to display reports which containers and how many containers can be ready is very large, and queries required joining several tables, indexing is created to speed up querying. Indexes are created on column filters and joins that are most frequently used in queries. And did testing several times to find the most optimal indexing, with the fastest query time.

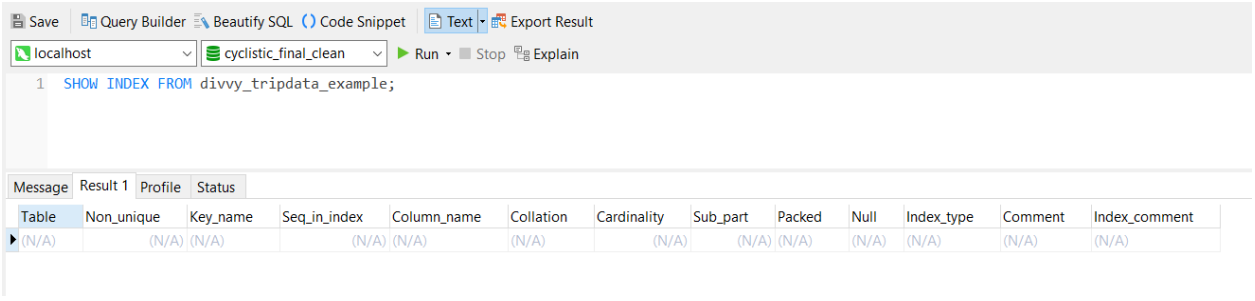
I'm using a cyclistic database to give an example of an index.

Table name : divvy_tripdata_example

Total records of divvy_tripdata_example : 4.560.047

I have not created any index till now on this table. Let's verify this by the command: SHOW INDEX. It returns 0 results.

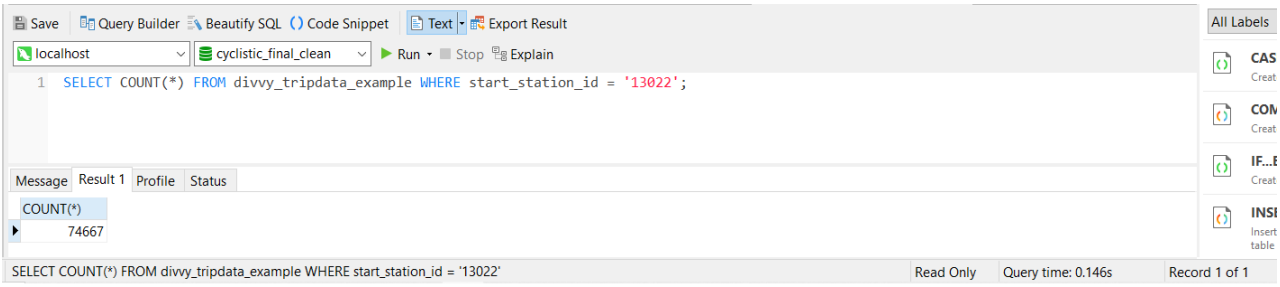
```
SHOW INDEX FROM divvy_tripdata;
```



At this moment, we will run a simple SELECT query with a where statement in column start_station_id. Since there is no user defined index, the query will scan the whole table to find out the result :

Query time : 6.774 s

Then I create an index on column start_station_id and re-run that simple SELECT query :



Query time : 0.146 s

As seen in the image above, a table that has an index in the start_station_id column has a much faster query time than the previous query time without an index.

Summary :

- Indexing adds a data structure with columns for the search conditions and a pointer
- The pointer is the address on the memory disk of the row with the rest of the information
- The index data structure is sorted to optimize query efficiency
- The query looks for the specific row in the index; the index refers to the pointer which will find the rest of the information
- The index reduces the number of rows the query has to search.