BRIGHAM YOUNG UNIVERSITY

CIT 225 Database Design and Development

Week 12 Assignment case-study

Tumwesige Nelson

Questions

1. Why indexes exist?
2. What unique and non-unique indexes do, like their upside and downside in an application?
3. What are the differences and performance characteristics of single column and multicolumn indexes?
4. What are the differences between a B-Tree and Bitmap Index?

Answers

Given indexes to be one of the effects that attacks MSQL codes, it is of importance to find rows with a specific unique values quickly and walking through the entire table.

Forexample: SELECT fname, lname

FROM students

WHERE fname LIKE (A%);

Therefore it is used to quickly locate data without having a search for every row.

For the unique index, it maintains data integrity by ensuring that no rows of data in a table have identical key values. If an email is added and it already exists, an error will be received. Therefore on the primary key columns no unique index is built. Also a unique index creates a unique table and no duplicate values allowed. In conclusion the non-unique index will act as the opposite of the unique index.

The differences and performance characteristics of single column and multi-column indexes

**Differences**

|  |  |
| --- | --- |
| **A single column** | **Multi column** |
| Based on values in one column of a table. | Based on values in multi columns of a table. |
| Rows are returned in order saved in the SQL written codes. | Rows are returned in the order they are stored on a disk. |
| It does not consume a lot of time since one row is to be sorted at a time. | It`s time consuming if there is sorting a lot of rows. |

**Performance characteristics**

|  |  |
| --- | --- |
| **A single column** | **Multi column** |
| Retrieves data slowly from the table. | Retrieves data faster from the table. |
| Less metadata is needed since it uses fewer columns. | More metadata is needed since it uses more than one column. |

The differences between B-tree and Bitmap indexes are:

|  |  |
| --- | --- |
| **B-tree** | **Bitmap** |
| Data is divided into balanced groups and organized in a tree like structure. | Data is not divided into balanced groups. |
| There is low duplication and high cardinality. | There is high duplication and low cardinality. |
| B-tree is faster when collecting small dataset. | It is slow during the collection of data. |
| B-tree delivers low quality results when looking for larger dataset. | Bitmap delivers high quality results when looking for larger dataset. |
| Inefficient for inserting and updating values in a table and easily arise an error. | Efficient for inserting and updating values in a table and does not arise an error. |