German University in Cairo

Faculty of Media Engineering and Technology

Computer Science Department

Bonus Report

Course Name : CSEN 604 Data Base II

Introduced By:-

Team Name :- (APlus Tree)

Team Members : -

- Sarah Ahmed 43-17401
- Ali Kabeel 43-1172
- Aman Allah Rafat 43-17354
- Mohamed Khaled 43-16053
- Salma Khaled 43-2735

Dr:-

Wael Abouelsaadat

The Idea Behind Paging Index Trees : -

- As we are required to improve our Database Engine Time Complexity for DDL and DML Queries and not to load useless data in the memory , so our approach will be to store in the Node its keys and instead of array of Nodes we store their directories and if needed we deserialize the needed directory to get the node .

- We store the index each node in a page and we use a naming convention for that purpose as (Table Name + Column Name + index) such that the term index is a static variable that starts from zero and it is incremented whenever a node is created. the problem that face us is that when we create an index over a column and then we close our running DB Engine and then start again that variable start with zero then when we need to insert a node that node will override an existing node so we decide that that variable is important to be stored as a separate page (Binary File) and whenever a node is created we read that page and create the node and write it back to the new index value for the next node.

  Code Wise : (BTreeNode.java Class)

  - Line 119 The Constructor.
  - Line 40 Method to get the next index for the node.
  - Line 30 Method to save the index after creating the node for the next insertion operation.
  - Also in Line 68 Method deserializeNode that deserialize that page and return back the node.
  - Line 97 Method save that save the Node as a Binary File (Serialize) and for updating purpose also such that updating the keys array.

- In the BTreeInnerNode.java Class that will represent the nodes in the tree except the leaves (Root Considered as an InnerNode) and we store in an array of Directories that represent its children so when we deserialize the node we do not serialize its children too.

Code Wise : (BtreeInnerNode.java Class)

- Line 35 Method to get the Child Node based on its index in the array of the Directories .

- BTreeLeafNode.java Class the leaf Node represent the actual data that are stored as Vector of References of Records (Records that are duplicate due to the column the Tree Built on) as a leafNode and we set for each leafNode a next node that will be helpful in the Range Queries .

  Code Wise : (BTreeLeafNode.java Class)

  - Line 14 , 19 the idea of storing References to handle duplicates .
  - Line 32 Method to get the next Node by deserialize the nextNode Directory and return the node back.
  - Line 43 Method to set the next Node for a LeafNode and then using the Method save to Serialize it again to save the data of the Node ,
  - Line 128,149,294,341,374 using save to store the updated version of the Node after the insert, split ,add duplicate Record , delete , deleteVectorAt (called in the borrow method that tries to borrow from another node).
- The Whole Idea is that the Node is deserialized once its needed to optimize the used memory space and that is DONE using the paging approach as described above .
- The Whole Team participate in the project as a one Unit with different approaches in all project stages brain storming , coding , testing and debugging . It was a great experience to work in such challenging project and hoping that we make it as it should be with the bonus part.