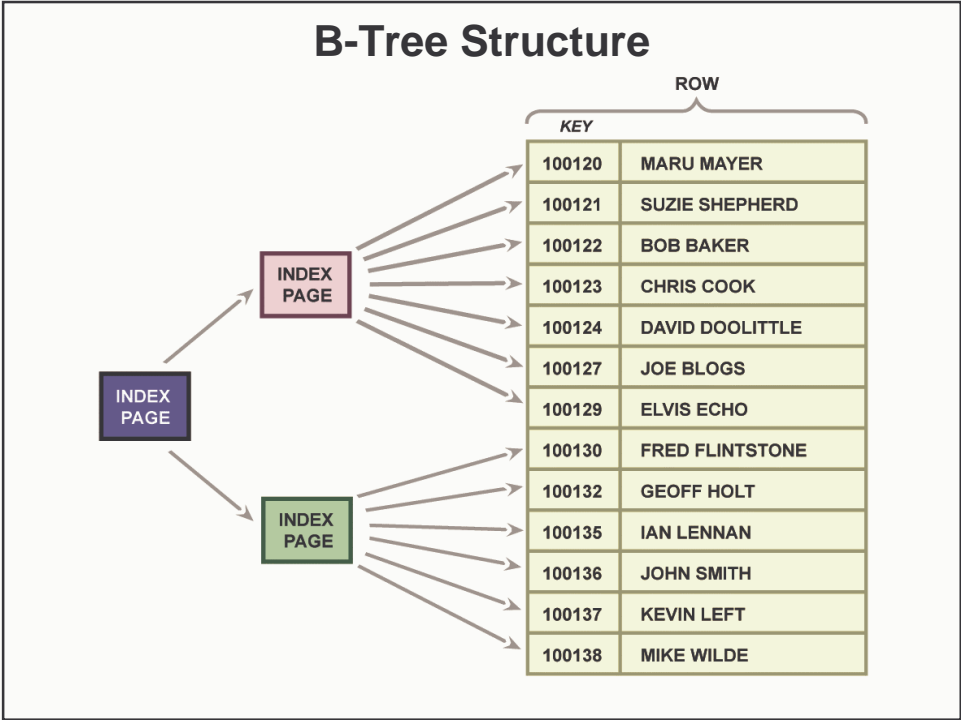
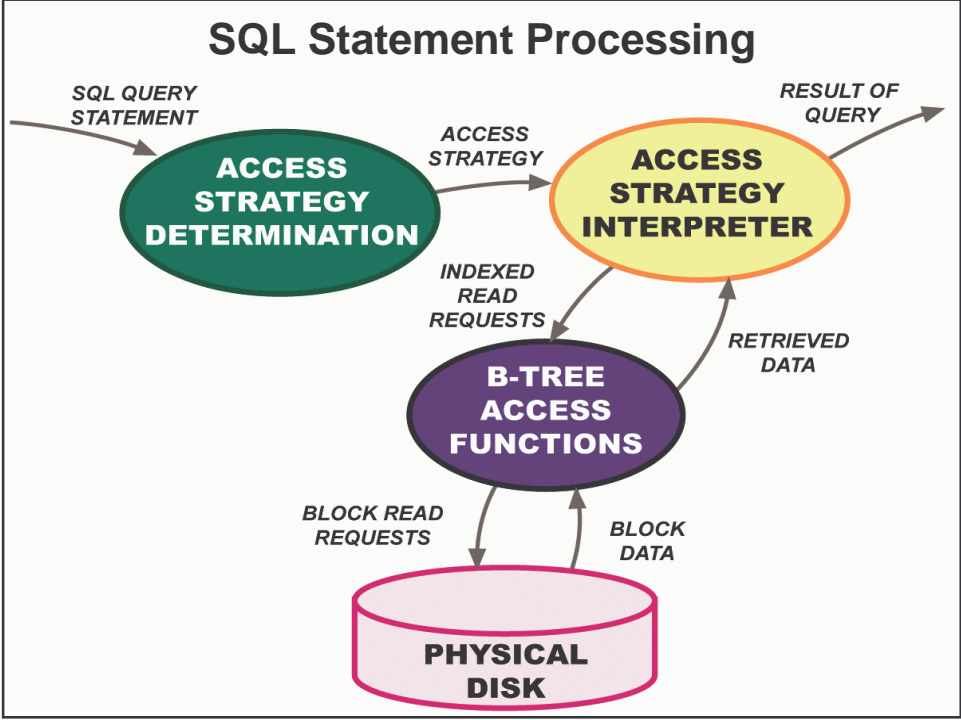


E-Genting Programming Competition 2004

Pre-Competition Workshop, Week 2
28 September 2004

Languages and Compilers

- A. A high-level language defines a virtual machine. A programmer should be able to write in a high-level language without regard to the code that might be generated by the compiler of that language.***
- B. A high-level language is a kind of shorthand for the instructions that will be executed by the computer. A programmer should always be mindful of the machine-level consequences of a particular high-level construct.***



B-Tree Operations

```
template <typename Key_t, typename Row_t> class BTree_c {
public:
    void BtReset () { /*...*/ }
        // Position the row pointer at the beginning of
        // the tree
    bool BtFind (const Key_t *key) { /*...*/ }
        // Position the row pointer at the row with key
        // 'key', or if no such row exists, the row with
        // the next highest key. Return 1 if the key
        // was found, otherwise 0.
    bool BtRead (Row_t *row) { /*...*/ }
        // Read the row that the row pointer is pointing
        // to and load it into the location addressed by
        // 'row'. Move the row pointer to the next row
        // in the tree. Return 1 if a row was read.
        // Return 0 if the row pointer is pointing to the
        // end of the tree.
};
```

Example Schema

```
// Customer file
create table custFile (
    custId    integer not null, // Customer identifier
    custName  char(40) not null // Customer name
);
create index custIdInd on custFile (custId);

// Purchases file
create table purFile (
    purId    integer not null, // Customer identifier
    purNo    integer not null, // Purchase number
    purVal   integer not null  // Purchase value
);
create index purIdNoInd on purFile (purId, purNo);
```

Query Translation

Select statement	<pre>select custName, purVal from custFile, purFile where custId >= 100127 and purId = custId;</pre>
Access strategy	<pre>cfKey.custId = 100127; custFile.BtFind (&cfKey); while (custFile.BtRead(&cfRec)) { pfKey.purId = cfRec.custId; pfKey.purNo = MIN_PUR_NO; purFile.BtFind(&pfKey); while (purFile.BtRead(&pfRec) && pfRec.purId == cfRec.custId) emit (cfRec.custName, pfRec.purVal); }</pre>

Optimisation

Select statement	<pre>select custName, purVal from custFile, purFile where custId >= 100127 and purId = custId;</pre>
Access strategy	<pre>cfKey.custId = 100127; custFile.BtFind (&cfKey); pfKey.purId = 100127; pfKey.purNo = MIN_PUR_NO; purFile.BtFind (&pfKey); pfOK = purFile.BtRead (&pfRec); while (custFile.BtRead(&cfRec)) { while (pfOK && pfRec.purId <= cfRec.custId) { if (pfRec.purId == cfRec.custId) emit (cfRec.custName, pfRec.purVal); pfOK = purFile.BtRead (&pfRec); } }</pre>