ABL Code Performance Workshop: Take 3

You don't know *beep* about indexes
compile-listing ain’t enough

- Intro

- Getting Connected
  - Laptop, wifi, remote desktop, credentials
  - Lab 0

- Useful Tools:
  - LOG-MANAGER, ProTop, -zqil, XREF
  - Lab 1

- Index selection basics: AND
  - Lab 2

- (( Index ) AND (selection) OR (maybe not))
  - OR branches and parentheses
  - Lab 3

- Some subtleties: casing, unknowns, temp-tables
  - Lab 4

- Conclusion
who are we?

- One named Peter
- One named Paul
one named Peter (so-called Judge)

pjudge@progress.com
Software Architect*
@ Progress since 2003
Integration-y stuff – Authentication Gateway, HTTP-Out, Corticon et al
OE Best Practices / OERA / AutoEdge / CCS
4GL since 1996

* Aka programmer who knows PowerPoint
Progress DBA and UNIX admin since 1994
Expert OpenEdge technical consulting
Wide range of experience
  - Small 10 person offices to 3500+ concurrent users
  - AIX, HPUX, Linux, Windows…if Progress runs on it, I’ve worked on it
Father to these two monkeys

pk@wss.com
▪ The oldest and most respected independent DBA consulting firm in the world

▪ Five of the world’s top OpenEdge DBAs

▪ Author of ProTop, the #1 FREE OpenEdge Database Monitoring Tool

http://protop.wss.com
The Compiler constructs a logical tree from a query and evaluates both sides of each AND or OR, looking for index criteria. ABL counts equality, range, and sort matches (for OR) and uses them to select and bracket indexes.

The precise rules are numerous and complex, and it is not important to fully understand their details.
Help! I need somebody!
test and verify? how !?!

Given a simple query

```sql
for each order no-lock where salesrep begins 'd'
```

- ProTop of course. It’s free.

<table>
<thead>
<tr>
<th>Idx#</th>
<th>Area#</th>
<th>Index Name</th>
<th>LvlS</th>
<th>Index Activity</th>
<th>Blocks</th>
<th>Util Idx</th>
<th>Root</th>
<th>Create</th>
<th>Read v</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>8</td>
<td>Order.SalesRep</td>
<td>2</td>
<td></td>
<td>358</td>
<td>51%</td>
<td>77952</td>
<td>0</td>
<td>182915</td>
</tr>
</tbody>
</table>

- Query object handle

- Log-manager - see next slides
A terribly underused but awesomely amazing tool

- Allows you to leave debug messages in your code
  - No more /* Message here *Value. */

- Create some secret hotkey sequence to activate
  - I.e. you can turn it on in production for one user

- Writes detailed info to a log file

```plaintext
assign log-manager:logfile-name = "c:\temp\wshop.log"
log-manager:logging-level = 3 // There are more levels than you think
log-manager:log-entry-types = "4GLTrace,4GLTrans,QryInfo".
```
test and verify with log-manager

- Still with our for each order no-lock where salesrep begins 'd' example:

  Type: Dynamically Opened Query
  PREPARE-STRING: for each order no-lock where salesrep begins 'd'
  Prepared at Runtime
  Client Sort: N
  Scrolling: Y
  Table: sports2000.Order
    Indexes: SalesRep
  Times prepared: 1
  Time to prepare (ms): 0
  DB Blocks accessed:
    sports2000 : 366074
  DB Reads:
    Table: sports2000.Order : 182913
    Index: Order.SalesRep : UNAVAILABLE
test and verify using protop
About ProTop

- Download free version at [http://protop.wss.com](http://protop.wss.com)
- Local ChUI for real-time data
- Web portal for trending data
  - Free and $$
- Monitoring and Alerts ($$)
- Free version is open source (GNU GPL). Take a look under the hood if you like

- Hit “h” to get the help screen - Options are CaSE-sENSiTiVe!!
These graphs are from the $$$ part of the ProTop web dashboard
-zqil

- Unsupported and undocumented startup parameter
  - Aren't those the best!?!?
- Writes detailed run-time index usage information to db.lg - yes db.lg
  - Do NOT use in PRODUCTION

- Tells you which index is used and how many fields deep
- Presented as number of lower brackets and upper brackets
  - GT and GE are lower brackets: they define a lower boundary for the index field
  - LT and LE are upper brackets: they define an upper boundary for the index field
Index basics
but before we get to the rules…

- This only applies to ABL and not SQL
- Rules are applied in hierarchical order to filter indexes
  - This is important: Each rule is applied and the result is one or more **remaining** indexes
  - Use a worksheet approach to make the rules "easy" to apply

- The first 7 rules only apply to a subset of indexes
  - Compiler scans all fields in the query and selects all indexes that have leading components with those fields

- Field match rules must be contiguous
  - “Equality” on fields 1 and 3 of the index counts as 1, not 2

**Rules continue to be applied until there is only one index left**
but before we get to the rules…

Table: Customer

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sRepStateCity</td>
<td>8</td>
<td>3 + SalesRep + State + City</td>
</tr>
<tr>
<td>Comments</td>
<td>8</td>
<td>1 + Comments</td>
</tr>
<tr>
<td>CountryPost</td>
<td>8</td>
<td>2 + Country + PostalCode</td>
</tr>
<tr>
<td>CustNum</td>
<td>8</td>
<td>1 + CustNum</td>
</tr>
<tr>
<td>Name</td>
<td>8</td>
<td>1 + Name</td>
</tr>
<tr>
<td>SalesRep</td>
<td>8</td>
<td>1 + SalesRep</td>
</tr>
<tr>
<td>sRepState</td>
<td>8</td>
<td>2 + SalesRep + State</td>
</tr>
<tr>
<td>CustNumUseless</td>
<td>8</td>
<td>2 + CustNum + SalesRep</td>
</tr>
</tbody>
</table>

FOR EACH Customer WHERE State = 'Leinster' AND City = 'Dublin'

// The sRepStateCity index is NOT eligible for selection
// NO indices are pre-selected for this query

Field match rules must be contiguous

- “Equality” on fields 1 and 3 of the index counts as 1, not 2

Rules continue to be applied until there is only one index left
For the first-pass set of indexes, filter using the following rules:

1. Pre-select only indexes with leading components in the where clause
2. If CONTAINS use word-index
3. Unique index with all components involved in the equality matches
4. Complete equality matches in > 1 index: all index components involved in the equality matches
5. Most active equality matches
6. Most active range matches
7. Most active sort matches

If multiple indexes remain, select one from

8. The primary index
9. First index alphabetically by name
For the first-pass set of indexes, filter using the following rules:

1. Pre-select only indexes with leading components in the where clause
2. If CONTAINS use word-index
3. Unique index with all components involved in the equality matches
4. Complete equality matches in > 1 index: all index components involved in the equality matches
5. Most active equality matches
6. Most active range matches
7. Most active sort matches

If multiple indexes remain, select one from
8. The primary index
9. First index alphabetically by name

Superset Selector
If there are indexes that will select supersets of records that are selected by other indexes, then we eliminate those.

Sort Selector Skip
Index records are already sorted by the index fields, so we don't need to reevaluate
### Order Table Indexes

<table>
<thead>
<tr>
<th>Table</th>
<th>Records</th>
<th>Size</th>
<th>Size</th>
<th>Count</th>
<th>LOBs</th>
<th>Size</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUB.Order</td>
<td>727285</td>
<td>63.9M</td>
<td>92</td>
<td>727285</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

### Indexes

<table>
<thead>
<tr>
<th>Flags</th>
<th>Index Name</th>
<th>Cnt</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>CustOrder</td>
<td>2</td>
<td>CustNum + Ordernum</td>
</tr>
<tr>
<td></td>
<td>OrderDate</td>
<td>1</td>
<td>OrderDate</td>
</tr>
<tr>
<td>pu</td>
<td>OrderNum</td>
<td>1</td>
<td>Ordernum</td>
</tr>
<tr>
<td></td>
<td>OrderStatus</td>
<td>1</td>
<td>OrderStatus</td>
</tr>
<tr>
<td></td>
<td>SalesRep</td>
<td>1</td>
<td>SalesRep</td>
</tr>
<tr>
<td>w</td>
<td>SRepW</td>
<td>1</td>
<td>SalesRep</td>
</tr>
<tr>
<td></td>
<td>SRepDate</td>
<td>2</td>
<td>SalesRep + OrderDate</td>
</tr>
<tr>
<td></td>
<td>DateSRep</td>
<td>2</td>
<td>OrderDate + SalesRep</td>
</tr>
<tr>
<td></td>
<td>SDateOstat</td>
<td>2</td>
<td>ShipDate + OrderStatus</td>
</tr>
</tbody>
</table>
for each order

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>CustOrder</th>
<th>OrderDate</th>
<th>OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>wSRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Complete equality match</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Most active equality matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Most active range matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Most active sort matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>The primary index</td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
</tr>
</tbody>
</table>

First index alphabetically by name
where salesrep = "BBB"

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>U CustOrder</th>
<th>OrderDate</th>
<th>OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>W sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If &quot;CONTAINS&quot;, use word-index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

First index alphabetically by name
where salesrep = "DKP" and orderdate = 09/05/2011

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>U CustOrder</th>
<th>OrderDate</th>
<th>PU OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>W sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If &quot;CONTAINS&quot;, use word-index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First index alphabetically by name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
where salesrep = "DKP" and orderdate = 09/05/2011 by salesrep

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>U CustOrder</th>
<th>OrderDate</th>
<th>PU OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>W sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First index alphabetically by name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Where `salesrep` = "BBB" and `orderStatus` = "Shipped".
where salesrep >= "DKP" and orderdate >= 09/05/2011 by salesrep

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>Index</th>
<th>U CustOrder</th>
<th>OrderDate</th>
<th>PU OrderNum</th>
<th>OrderStatus</th>
<th>W SalesRep</th>
<th>sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First index alphabetically by name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Selection Rule</td>
<td>CustOrder</td>
<td>OrderDate</td>
<td>OrderNum</td>
<td>OrderStatus</td>
<td>SalesRep</td>
<td>sRepW</td>
<td>SRepDate</td>
<td>DateSRep</td>
<td>SDateOstat</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

First index alphabetically by name
where orderdate = 09/05/2011 and shipdate = 10/01/2011

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>CustOrder</th>
<th>OrderDate</th>
<th>OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

First index alphabetically by name
where `ordorderate = 09/05/2011` and `shipdate = 10/01/2011`

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>U CustOrder</th>
<th>OrderDate</th>
<th>PU OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>W sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td><img src="checkmark.png" alt="Checkmark" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td><img src="exclamation.png" alt="Exclamation" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First index alphabetically by name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Query Code:**
```
src\ lab_2.p src\lab_2.p 13 SEARCH bigsports.Order OrderDate
```
Expressions break bracketing

\[ \text{FOR EACH } \text{order } \text{NO-LOCK WHERE MONTH(\text{orderDate})} = 1 \ldots \]

BEGINS does NOT break bracketing

- Considered a range bracket

\[ \text{FOR EACH } \text{order } \text{NO-LOCK WHERE } \text{salesRep BEGINS "D"} \]

- Uses the \text{order.salesRep} index

MATCHES breaks bracketing

When in doubt, test and verify
where \( \text{month(orderDate)} > 6 \)

<table>
<thead>
<tr>
<th>Selection Rule</th>
<th>U</th>
<th>CustOrder</th>
<th>OrderDate</th>
<th>PU</th>
<th>OrderNum</th>
<th>OrderStatus</th>
<th>SalesRep</th>
<th>W</th>
<th>sRepW</th>
<th>SRepDate</th>
<th>DateSRep</th>
<th>SDateOstat</th>
</tr>
</thead>
<tbody>
<tr>
<td>If “CONTAINS”, use word-index</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Unique index with all components involved in the equality matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Complete equality match</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Most active equality matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Most active range matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Most active sort matches</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The primary index

First index alphabetically by name
That's all ... OR?
### customer table indexes

<table>
<thead>
<tr>
<th>Table</th>
<th>Records</th>
<th>Size</th>
<th>Size</th>
<th>Count</th>
<th>LOBs</th>
<th>Size</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUB.Customer</td>
<td>201120</td>
<td>31.1M</td>
<td>162</td>
<td>201120</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

### Indexes

<table>
<thead>
<tr>
<th>Flags</th>
<th>Index Name</th>
<th>Cnt</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>Comments</td>
<td>10</td>
<td>1 + Comments</td>
</tr>
<tr>
<td></td>
<td>Country</td>
<td>9</td>
<td>1 + Country</td>
</tr>
<tr>
<td></td>
<td>CountryPost</td>
<td>10</td>
<td>2 + Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ PostalCode</td>
</tr>
<tr>
<td>pu</td>
<td>CustNum</td>
<td>10</td>
<td>1 + CustNum</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>10</td>
<td>1 + Name</td>
</tr>
<tr>
<td></td>
<td>SalesRep</td>
<td>10</td>
<td>1 + SalesRep</td>
</tr>
<tr>
<td></td>
<td>SrepCountryCLimit</td>
<td>9</td>
<td>3 + SalesRep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ CreditLimit</td>
</tr>
</tbody>
</table>
for each Customer where Customer.Name = "Koufalis" or Customer.SalesRep = "BBB"

What indexes are used?
for each Customer where Customer.Name = "Koufalis" or Customer.SalesRep = "BBB"

What indexes are used?

- If you guessed **Name**, bravo.
- If you guessed **SalesRep**, bravo.
- Validate with COMPILE ... XREF

```plaintext
custom.p 45 ACCESS sports2000.Customer Name
custom.p 45 ACCESS sports2000.Customer Name
custom.p 45 STRING "Koufalis" 8 NONE TRANSLATABLE
custom.p 45 STRING "BBB" 5 NONE TRANSLATABLE
```
```plaintext
custom.p 45 SEARCH sports2000.Customer Name
```
```plaintext
```
for each Customer where Customer.Name = "Koufalis" or Customer.Name = "Judge"

What indexes are used?
for each Customer where Customer.Name = "Koufalis" or Customer.Name = "Judge"

What indexes are used?

- If you guessed Name, bravo.

- Validate with COMPILE ... XREF

```plaintext
custom.p 45 ACCESS sports2000.Customer Name
custom.p 45 ACCESS sports2000.Customer Name
custom.p 45 STRING "Koufalis" 8 NONE TRANSLATABLE
custom.p 45 STRING "Judge" 5 NONE TRANSLATABLE
custom.p 45 SEARCH sports2000.Customer Name
```
Which indexes will be used?

- If you guessed Name, bravo
- Validate with COMPILE ... XREF for each customer where Customer.Name = "Koufalis" or Customer.Name = "Judge"

custom.p 45 ACCESS sports2000.Customer Name

custom.p 45 SEARCH sports2000.Customer Name

custom.p 45 STRING "Koufalis" 8 NONE TRANSLATABLE

custom.p 45 STRING "Judge" 5 NONE TRANSLATABLE
the OR operator makes a difference

Each side of an OR is its own, distinct index selection operation

for each customer where Customer.Name = "Koufalis" or Customer.SalesRep = "BBB"
  • 2 distinct index selection operations

for each customer where Customer.Name = "Koufalis" or Customer.Name = "Judge"
  • This is also 2 distinct index selection operations

• A "logic tree" is built until we hit an AND. Break down the query until we hit ANDs
• This implies that we can force multiple indexes to be used via ORs and parentheses
• Now we apply our rules to the AND part of the logic tree, as above
simple OR clause

for each customer where

\[
\text{customer.Name} = "Koufalis" \quad \text{or} \quad \text{customer.SalesRep} = "BBB"
\]
### Simple OR Clause

For each customer where

- `customer.Name = "Koufalis"`
- or
- `customer.Name = "Judge"`

---

<table>
<thead>
<tr>
<th>Index</th>
<th>Selection Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PU</td>
</tr>
<tr>
<td>-------</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete equality match</td>
<td></td>
</tr>
<tr>
<td>Most active equality matches</td>
<td></td>
</tr>
<tr>
<td>Most active range matches</td>
<td></td>
</tr>
<tr>
<td>Most active sort matches</td>
<td></td>
</tr>
<tr>
<td>The primary index</td>
<td></td>
</tr>
</tbody>
</table>

---

First index alphabetically by name

---

**Customize**

- `custom.p 45 ACCESS sports2000.Customer Name`
- `custom.p 45 ACCESS sports2000.Customer Name`
- `custom.p 45 STRING "Koufalis" 8 NONE TRANSLATABLE`
- `custom.p 45 STRING "Judge" 5 NONE TRANSLATABLE`
- `custom.p 45 SEARCH sports2000.Customer Name`
- `custom.p 45 SEARCH sports2000.Customer Name`
complex OR with parentheses (1)

How many "simple clauses" are here?

for each Customer where

( Customer.Name = 'Koufalis' or
  ( Customer.Country = 'AAA' or
    ( Customer.Name = 'Koufalis' and Customer.Country = 'DEN' )
  )
)

or

( Customer.Salesrep = 'BBB' and Customer.Salesrep = 'XXX' )
How many "simple clauses" are here?

for each Customer where

( Customer.Name = 'Koufalis' or
  ( Customer.Country = 'AAA' or
    ( Customer.Name = 'Koufalis' and Customer.Country = 'DEN' )
  )
)

or

( Customer.Salesrep = 'BBB' and Customer.Salesrep = 'XXX' )

custom.p 45 SEARCH sports2000.Customer Name
custom.p 45 SEARCH sports2000.Customer Name
How many "simple clauses" are here?

for each Customer where

( Customer.Name = 'Koufalis' or
  ( Customer.Country = 'AAA' or
    ( Customer.Name = 'Koufalis' and
      ( Customer.Country = 'DEN' or Customer.Country = 'USA' )
    )
  )
)

or

( Customer.Salesrep = 'BBB' and Customer.Salesrep = 'XXX' )
How many "simple clauses" are here?

```
for each Customer where
  ( Customer.Name = 'Koufalis' or
    ( Customer.Country = 'AAA' or
      ( Customer.Name = 'Koufalis' and
        (Customer.Country = 'DEN' or Customer.Country = 'USA' )
      )
    )
  )

or

( Customer.Salesrep = 'BBB' and Customer.Salesrep = 'XXX' )
```
But wait! There's more!
indexes and unknown values

▪ Unique indexes enforce a constraint of a one value per index-field per table …

BUT multiple unknown values in ABL are allowed

▪ If you want truly unique values, mark the field(s) in the index as mandatory.

▪ Index field(s) containing unknown values sort **higher** than any other value

create Customer.
assign Customer.custnum = 1
    Customer.name  = 'Peter'.

create Customer.
assign Customer.custnum = 2
    Customer.name  = 'Paul'.

create Customer.
assign Customer.custnum = ?
    Customer.name  = 'Mary'.

for each Customer by CustNum desc:

<table>
<thead>
<tr>
<th>CustNum</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Mary</td>
</tr>
<tr>
<td>2</td>
<td>Paul</td>
</tr>
<tr>
<td>1</td>
<td>Peter</td>
</tr>
</tbody>
</table>
indexes and case sensitivity

- By default, the AVM doesn't care about case for fields (whether indexed or not)
  - Field (data) values are stored as entered
  - Index field values are stored in UPPER CASE

  "Paul" = "PAUL" = "paul"

- If the fields are case sensitive
  - Field (data) values are stored as entered
  - Index field values are stored as entered

  "Paul" <> "PAUL" <> "paul"

  • Sorting on these fields may differ
  • Word indexed fields are always treated as case-insensitive
when no index is needed

find Customer where rowid(Customer) eq 0xDEADBEEF

- Fast, non-indexed record access
- ROWID is the new RECID
  - The value can change - only use for temporary/short-lived use

for each Customer table-scan

- Fast, non-indexed record access for temp-tables or db tables in Type 2 storage
- If the AVM could select an index with the first 5 rules, it will warn you at COMPILe time
- TABLE-SCAN > WHOLE-INDEX since only record blocks are read
- The first-alphabetical rule is replaced by a first-defined rule

- Fields cannot be marked as mandatory so uniqueness must be guaranteed by application code
The precise rules are numerous and complex, and it is not important to fully understand their details.

- Understanding the index "selection" rules is vital
  - There are subtleties and oddities though

- But so is being able to validate that your expected index is actually being used
  - COMPILE XREF
  - LOG-MANAGER
  - INDEX-INFORMATION()

OpenEdge Web Paper: ABL Database Triggers and Indexes
Enter our raffle to win Apple AirPods

https://prgress.co/PUGChallenge
fin