



ABL Code Performance Workshop: Take 3

You don't know *beep* about indexes



compile-listing ain't enough

- Intro
- Useful Tools:
 - LOG-MANAGER, ProTop, -zqil, XREF
- Index selection basics: AND
 - (((Index) AND (selection) OR (maybe not))
 - OR branches and parentheses
 - Some subtleties: casing, unknowns, temp-tables
- 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

one named Paul (Koufalis)

- 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





White Star Software

- 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

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

- ProTop of course. It's free.

Idx#	Area#	Index Name	Lvls	Index Blocks	Activity Util	Idx Root	Create	Read v
45	8	Order.SalesRep	2	358	51%	77952	0	182915

- Query object handle

```
q:index-information(1) = SalesRep
```

- Log-manager - see next slides



log-manager



- A terribly underused but awesomely amazing tool
- Allows you to leave debug messages in your code
 - No more `/* Message here vValue. */`
- 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

```
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

```

..... Table Activity .....
.  Tbl# Area# Table Name          RM Chain  #Records  Turns  Create  Read v  Update  Delete  OS Read .
.....
.  >  790   20 s_crm-valid-queue        60      1193966  0.16    0    194453    0      0      0      0 .
.    936   18 wm-send                    16      5368656  0.00    0      341      0      0      0      0 .
.    782   20 s_crm-crm-field                60         48  0.52    0      25      0      0      0      0 .
.    787   20 s_crm-lat-field                 60         48  0.36    0      17      0      0      0      0 .
.    849   22 wb_dept-user                   60      7088    0.00    0      12      0      0      0      0 .
.    798   20 s_param                         39      1979    0.00    0      8       0      0      0      0 .
.    169   20 cost-factor                     60         18  0.39    0      7       0      0      0      0 .
.    528   18 prod-exp                        37     217216  0.00    0      7       0      0      0      0 .
.    557   18 product                          1355    214159  0.00    0      7       0      0      0      0 .
.    654  142 so-pick-d                      383    28455997  0.00    0      7       0      0      0      0 .
.      1   22 alternate                       58        488  0.00    0      0       0      0      0      0 .
.      2   18 am-list                          60         2  0.00    0      0       0      0      0      0 .
.      3   8  am-list-d                          4         0  0.00    0      0       0      0      0      0 .
.....
..... Index Activity .....
.  Idx# Area# Index Name          Lvl#  Blocks Util Idx Root  Create  Read v  Split  Delete BlkDl Note .
.....

```

Index Activity												
Idx#	Area#	Index Name	Lvl#	Blocks Util	Idx Root	Create	Read v	Split	Delete	BlkDl	Note	
45	8	Order.SalesRep	2	358 51%	77952	0	182915	0	0	0		
. 1456	. 19	. so-pick.ctrl-machine	. 3	. 969 90%	. 9535	. 0	. 68	. 0	. 0	. 0	. .	
. 1463	. 19	. so-pick.shipped	. 2	. 292 98%	. 9983	. 0	. 65	. 0	. 0	. 0	. .	
. 4	. 6	. _Field._Field-Position	. 0	. 0 0%	. 68416	. 0	. 51	. 0	. 0	. 0	. U .	
. 6	. 6	. _Index-Field._Index/Number	. 0	. 0 0%	. 68544	. 0	. 36	. 0	. 0	. 0	. PU .	
. 1734	. 21	. s_crm-crm-field.s_crm-crm-field	. 1	. 1 6%	. 35263	. 0	. 26	. 0	. 0	. 0	. PU .	
. 1816	. 23	. wb_dept-user.wb_user	. 2	. 36 50%	. 13439	. 0	. 23	. 0	. 0	. 0	. .	
. 5	. 6	. _Index._File/Index	. 0	. 0 0%	. 68480	. 0	. 11	. 0	. 0	. 0	. PU .	
. 2	. 6	. _Field._File/Field	. 0	. 0 0%	. 68288	. 0	. 2	. 0	. 0	. 0	. PU .	
. 1196	. 121	. prod-exp-loc-d.prod-exp-loc-d	. 4	. 147,490 70%	. 127	. 0	. 0	. 0	. 0	. 0	. PU .	
User IO Activity												
Usr#	Tenant	Name	PID	Flags	Blk Ac v	OS Rd	OS Wr	Hit%	Rec Lck	Rec Wts	Line#	Program Name
. > 773	. 0	. trarm	. 48139	. SXB*	. 390366	. 0	. 0	. 100.00%	. 194492	. 0	. 582	. crmim/apply.p
. 778	. 0	. xuycata2	. 13742	. SX	. 1411	. 0	. 0	. 99.99%	. 1	. 0	. 305	. so/bmonaut1.p
. 772	. 0	. xc11wt2	. 21791	. SXB	. 1291	. 0	. 0	. 100.00%	. 169	. 0	. -1	. wm/senddata.p
. 783	. 0	. mlwt2	. 6278	. SXB	. 200	. 0	. 0	. 100.00%	. 0	. 0	. -1	. wm/senddata.p

About ProTop

- Download free version at <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

```
ProTop Help
Command Keys are Case-Sensitive

d = Performance Dashboard
t = Table Activity
i = Index Activity
u = User IO Activity
x = Active Transactions
L = Login Brokers
a = Storage Areas
^b = Buffer Pools

U = User Information Viewer

T = Show Table and Index Range Info
A = Generate DBAnalys Output

^p = programmer mode
^d = dump & load scripts

I = Set Refresh Interval

R = Switch between Raw and Rate Sampling
p = Pause to allow manual Screen Capture
Z = Set Top X Row Limit

c = Configuration Viewer
l = Latches and Resources
k = Checkpoints
b = Blocked Sessions
r = OE Replication Agents
s = Server Activity
f = File IO
B = BigB Guesstimator
@ = App Servers

# = Set Usr#
* = Client Statement Cache

6 = Set Table Name
7 = Set Index Name

z = Modify Sort Columns

S = Switch between Auto and On-Demand Sampling
M = Send Screen Capture as an e-mail
^ = Modify _Lock Scan Limit

O = OS Info (if avail)
Q = SQL activity
D = Disk Free Space
/ = Sequences
N = Network Traffic
m = Multi-Tenant Info
w = Who is Connected?
^a = After-Imaging Info
e = Application Specific

P = Set Process Id
C = Clear Client Statement Cache

8 = Monitor Users of a Table
9 = Monitor Users of an Index

^t = show session temp-tables
Y = show session user*stat
y = run profiler on this session

& = Show Global Properties

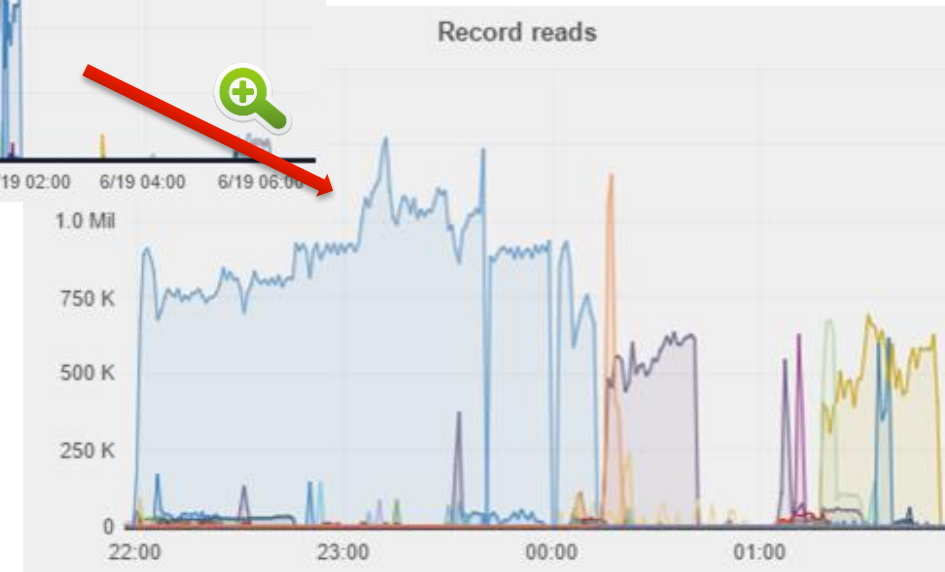
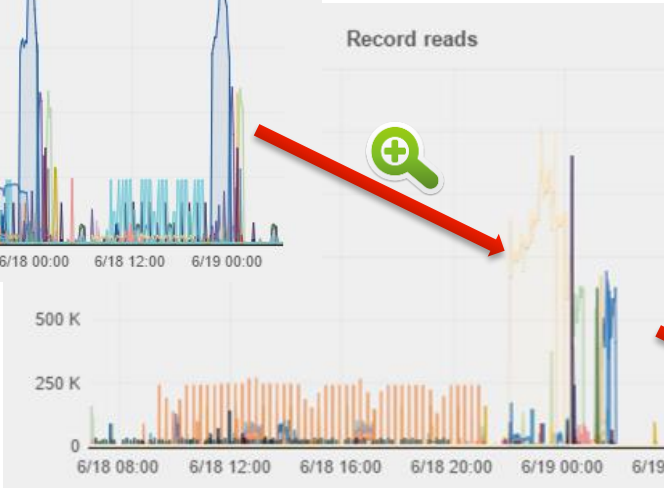
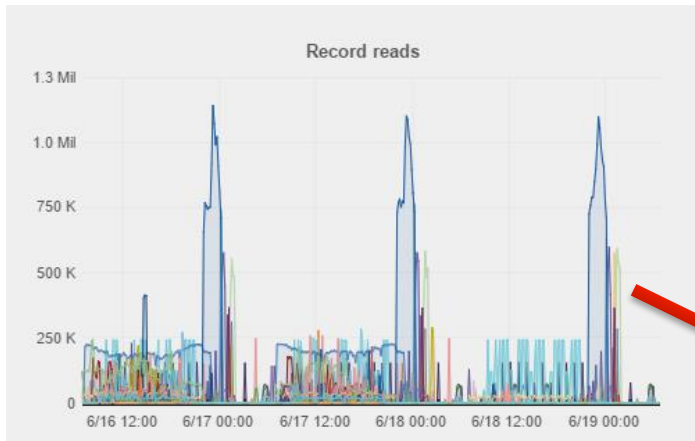
<space> = Refresh Sample
h, ? = Help
q = Quit
```

- Hit “h” to get the help screen - Options are CaSE-sENSiTiVe!!



ProTop Web Dashboard

Graphical view of `_TableStat/_Indexstat VST`



These graphs are from the \$\$ part of the ProTop web dashboard



- 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
      sRepStateCity      8      3 + SalesRep + State + City
w     Comments          8      1 + Comments
      CountryPost       8      2 + Country  + PostalCode
pu    CustNum           8      1 + CustNum
      Name              8      1 + Name
      SalesRep          8      1 + SalesRep
      sRepState         8      2 + SalesRep + State
      CustNumUseless    8      2 + CustNum  + SalesRep
```

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
```

- “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

the rules

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	Records	Size	Size	Count	LOBs	Size	Mean
PUB.Order	727285	63.9M	92	727285	--	--	--

Indexes





Flags	Index Name	Cnt	Field Name
u	CustOrder	2	+ CustNum + Ordernum
	OrderDate	1	+ OrderDate
pu	OrderNum	1	+ Ordernum
	OrderStatus	1	+ OrderStatus
	SalesRep	1	+ SalesRep
w	SRepW	1	+ SalesRep
	SRepDate	2	+ SalesRep + OrderDate
	DateSRep	2	+ OrderDate + SalesRep
	SDateOstat	2	+ ShipDate + OrderStatus

where salesrep = "BBB"






Index	U		PU		W				
	CustOrder	OrderDate	OrderNum	OrderStatus	SalesRep	sRepW	SRepDate	DateSRep	SDateOstat
Selection Rule									
If "CONTAINS", use word-index					X		X		
Unique index with all components involved in the equality matches					X		X		
Complete equality match									
Most active equality matches									
Most active range matches									
Most active sort matches									
The primary index									
First index alphabetically by name									

2


where salesrep = "DKP" and orderdate = 09/05/2011

Index	U CustOrder	OrderDate	PU OrderNum	OrderStatus	W SalesRep	sRepW	SRepDate	DateSRep	SDateOstat
Selection Rule									
If "CONTAINS", use word-index		X			X		X	X	
Unique index with all components involved in the equality matches		X			X		X	X	
Complete equality match							X	X	 
Most active equality matches							X	X	
Most active range matches							X	X	
Most active sort matches							X	X	
The primary index							X	X	
First index alphabetically by name									

where salesrep = "DKP" and orderdate = 09/05/2011 by salesrep

Index	U	OrderDate	PU	OrderStatus	SalesRep	W	sRepW	SRepDate	DateSRep	SDateOstat
Selection Rule										
If "CONTAINS", use word-index		X			X			X	X	
Unique index with all components involved in the equality matches		X			X			X	X	
Complete equality match								X	X	 
Most active equality matches								X	X	
Most active range matches								X	X	
Most active sort matches								X	X	
The primary index								X	X	
First index alphabetically by name										

where salesrep >= "DKP" and orderdate >= 09/05/2011 by salesrep

Index	U CustOrder	OrderDate	PU OrderNum OrderStatus	W SalesRep sRepW SRepDate DateSRep	SDateOstat	
Selection Rule						
If "CONTAINS", use word-index		X		X	X X	4
Unique index with all components involved in the equality matches		X		X	X X	
Complete equality match		X		X	X X	
Most active equality matches		X		X	X X	
Most active range matches				X	X X	3
Most active sort matches				X	X	
The primary index						
First index alphabetically by name						

where orderdate = 09/05/2011

Index	U	OrderDate	PU	OrderStatus	SalesRep	W	SRepDate	DateSRep	SDateOstat
Selection Rule	CustOrder		OrderNum			sRepW			
If "CONTAINS", use word-index		X						X	
Unique index with all components involved in the equality matches		X						X	
Complete equality match		X						X	
Most active equality matches		X						X	
Most active range matches		X						X	
Most active sort matches		X						X	
The primary index		X						X	
First index alphabetically by name									



where orderdate = 09/05/2011 and shipdate = 10/01/2011

Index	U	OrderDate	PU	OrderStatus	SalesRep	W	sRepW	SRepDate	DateSRep	SDateOstat
Selection Rule										
If "CONTAINS", use word-index		X							X	X
Unique index with all components involved in the equality matches		X							X	X
Complete equality match		X							X	X
Most active equality matches		X							X	X
Most active range matches		X							X	X
Most active sort matches		X							X	X
The primary index		X							X	X
First index alphabetically by name										

3



- Expressions break bracketing

FOR EACH order **NO-LOCK** **WHERE** MONTH(orderDate) = 1 ...

- BEGINS does NOT break bracketing

- Considered a range bracket

FOR EACH order **NO-LOCK** **WHERE** salesRep **BEGINS** "D"

- Uses the order.salesRep index

- MATCHES breaks bracketing

- When in doubt, test and verify

That's all ... OR?

customer table indexes

Table	Records	Size	Size	Count	LOBs	Size	Mean
PUB.Customer	201120	31.1M	162	201120	--	--	--

Indexes

Flags	Index Name	Cnt	Field Name
w	Comments	10	1 + Comments
	Country	9	1 + Country
	CountryPost	10	2 + Country + PostalCode
pu	CustNum	10	1 + CustNum
	Name	10	1 + Name
	SalesRep	10	1 + SalesRep
	SrepCountryCLimit	9	3 + SalesRep + Country + CreditLimit

oh really?

for each Customer where Customer.Name = "Koufalis" or Customer.SalesRep = "BBB"

What indexes are used?

oh really?

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

```
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  
custom.p 45 SEARCH sports2000.Customer Name  
custom.p 45 SEARCH sports2000.Customer SalesRep
```

oh really?

for each Customer where Customer.Name = "Koufalis" or Customer.Name = "Judge"

What indexes are used?

oh really?

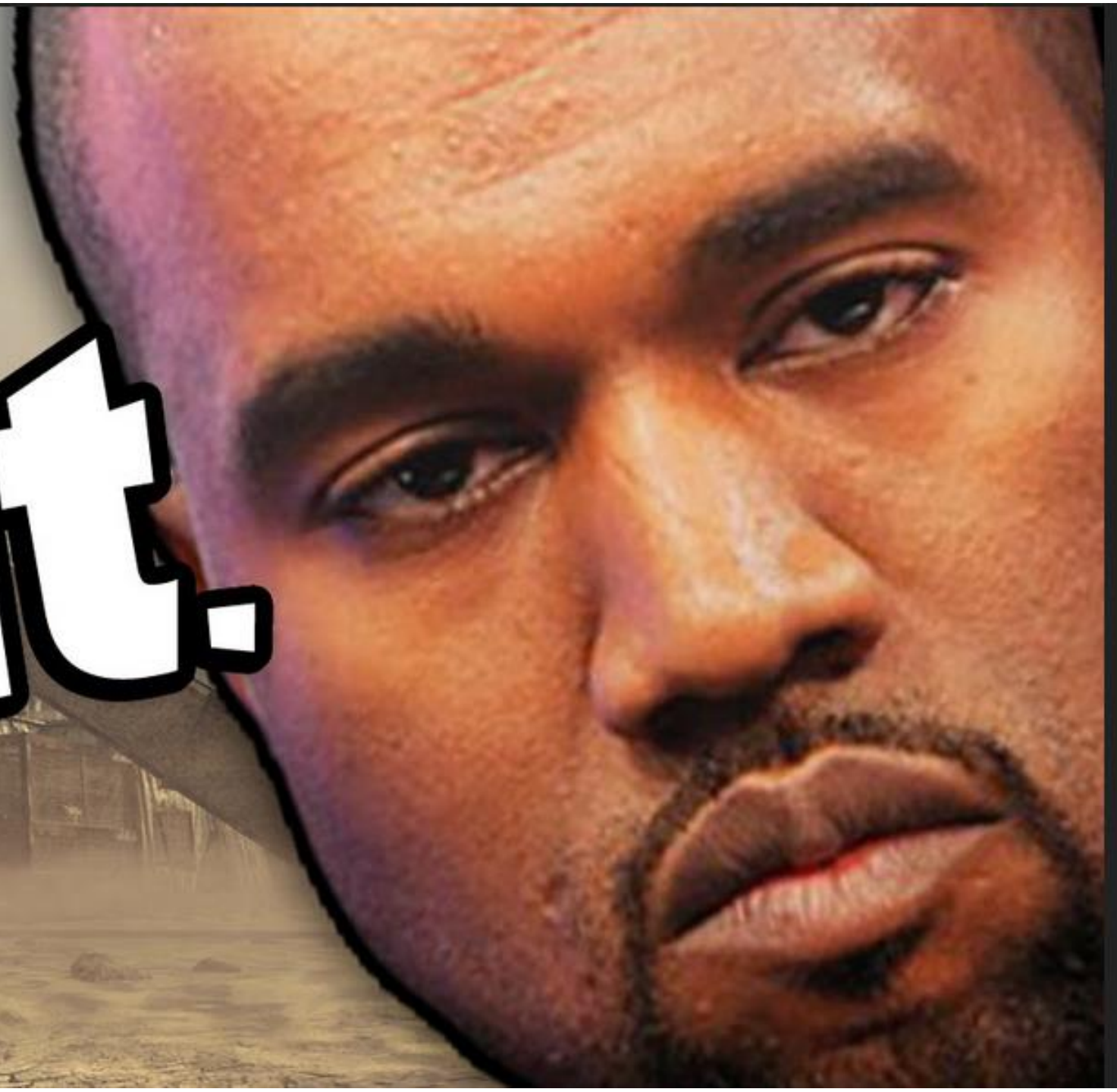
for each Customer where Customer.Name = "Koufalis" or Customer.Name = "Judge"

What indexes are used?

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

```
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
```

Wut.



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

customer.Name = "Koufalis"

for each customer where
or

customer.SalesRep = "BBB"

Index	PU CustNum	CountryPost	Name	SalesRep	W Comments	SRepDate	sRepState
Selection Rule							
If "CONTAINS", use word-index			X				1
Unique index with all components involved in the equality matches			X				
Complete equality match			✓				
Most active equality matches							
Most active range matches							
Most active sort matches							

The primary index							
First index alphabetically by name							

Index	PU CustNum	CountryPost	Name	SalesRep	W Comments	SRepDate	sRepState
Selection Rule							
If "CONTAINS", use word-index				X		X	X
Unique index with all components involved in the equality matches				X		X	X
Complete equality match				✓			
Most active equality matches							
Most active range matches							
Most active sort matches							

The primary index							
First index alphabetically by name							

```

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
custom.p 45 SEARCH sports2000.Customer Name
custom.p 45 SEARCH sports2000.Customer Salesrep
    
```

simple OR clause

customer.Name = "Koufalıs" for each customer where or customer.Name = "Judge"

Index	PU CustNum	CountryPost	Name	SalesRep	W Comments	SRepDate	sRepState
Selection Rule							
If "CONTAINS", use word-index			X				1
Unique index with all components involved in the equality matches			X				
Complete equality match			✓				
Most active equality matches							
Most active range matches							
Most active sort matches							

The primary index							
First index alphabetically by name							

Index	PU CustNum	CountryPost	Name	SalesRep	W Comments	SRepDate	sRepState
Selection Rule							
If "CONTAINS", use word-index			X				1
Unique index with all components involved in the equality matches			X				
Complete equality match			✓				
Most active equality matches							
Most active range matches							
Most active sort matches							

The primary index							
First index alphabetically by name							

```

custom.p 45 ACCESS sports2000.Customer Name
custom.p 45 ACCESS sports2000.Customer Name
custom.p 45 STRING "Koufalıs" 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' )
```

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' )
```

The diagram consists of five green arrows pointing left towards the SQL query. Arrow 1 points to the clause 'Customer.Name = 'Koufalis''. Arrow 2 points to the clause 'Customer.Country = 'AAA''. Arrow 3, 4 points to the clause 'Customer.Name = 'Koufalis' and Customer.Country = 'DEN''. Arrow 5 points to the clause 'Customer.Salesrep = 'BBB' and Customer.Salesrep = 'XXX''.

1	Name
2	Country
3, 4	Country, Name
5	Srep

```
custom.p 45 SEARCH sports2000.Customer Name  
custom.p 45 SEARCH sports2000.Customer Country  
custom.p 45 SEARCH sports2000.Customer Country  
custom.p 45 SEARCH sports2000.Customer Name  
custom.p 45 SEARCH sports2000.Customer Srep
```

complex OR with parentheses (2)

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' )
```

complex OR with parentheses (2)

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' )
```



- 46 SEARCH sports2000.Customer Name
- 46 SEARCH sports2000.Customer Country
- 46 SEARCH sports2000.Customer Name
- 46 SEARCH sports2000.Customer SalesRep

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:

CustNum	Name
-----	-----
?	Mary
2	Paul
1	Peter



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

```
-----  
Error (Press HELP to view stack trace)  
-----  
** Customer already exists with "Paul".  
(132)  
-----  
OK   Help  
-----
```



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 COMPILER time
- TABLE-SCAN > WHOLE-INDEX since only record blocks are read



temp-table anomalies

- 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
 - COMPILER XREF
 - LOG-MANAGER
 - INDEX-INFORMATION()

OpenEdge Web Paper: ABL Database Triggers and Indexes

https://documentation.progress.com/output/ua/OpenEdge_latest/wp-abl-triggers/openedge-web-paper-3a-abl-database-triggers-and-in.html





Enter our **raffle** to win Apple AirPods



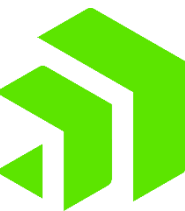
<https://pgress.co/PUGChallenge>





PUGCHALLENGE▶ AMERICAS

Paul Koufalis pk@wss.com
Peter Judge pjudge@progress.com





fin