

Pick An Index, Any Index...

Michael Lonski

Allegro

AllegroConsultants.com

Allegro

Who I Am...

- ❖ Started working with Progress® v3 (1986)
- ❖ Founded Allegro in 1993
- ❖ Internationally recognized speaker
- ❖ Author of “Coding Smart” book on ADM2
- ❖ Punster and Frequent Talker

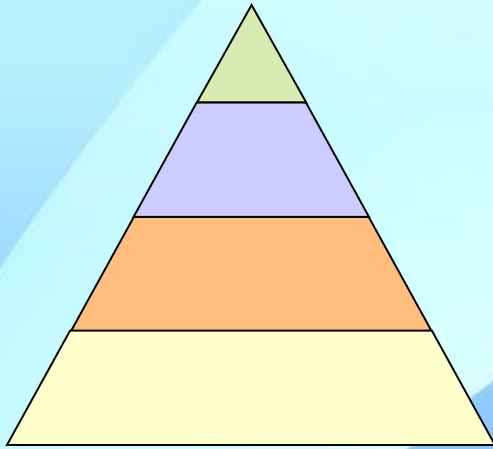
...And Why I Am Here

- ❖ Overview index rules
- ❖ Practical tools, not bookish theory
- ❖ Focus on fact, not guesswork
- ❖ Compile time information
- ❖ Run-time information
- ❖ Show off skill at alliteration

What's Next...

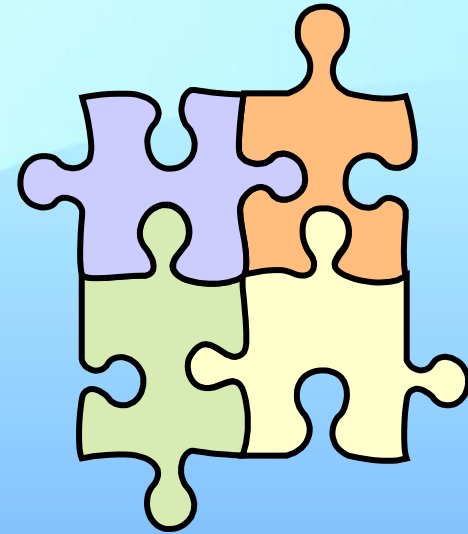
❖ Prepare

Index Rules Engines



“Rules based”

VS



“Context based”

Creating Index Entries

- ❖ At first occurrence of any of:
 - EOS which assigns all values to all index components
 - End of closest iterating transaction block where record is created
 - VALIDATE
 - Record is released from buffer
 - End of transaction scoping record create

Updating Index Entries

- ❖ Immediately at end of any statement changing one or more index elements
- ❖ Record is re-found using new entry
- ❖ Data not changed until released or end of scope

Bracketing

LastName LT "H"



Baker	Anita
Baker	Daniel
Baker	Thomas
Drew	Nancy
Gaston	Daniel
Gaston	Sherri
Marcus	Anita
Smith	Betty
Smith	Bob
Smith	Nancy


**LastName EQ "Smith" AND
FirstName BEGINS "B"**



More Bracketing

**LastName LT "H" AND
FirstName EQ "Daniel"**

~~**FirstName BEGINS "B"**~~



Baker	Anita
Baker	Daniel
Baker	Thomas
Drew	Nancy
Gaston	Daniel
Gaston	Sherri
Marcus	Anita
Smith	Betty
Smith	Bob
Smith	Nancy

What's Next...

- ❖ Prepare

- ❖ Predict

Hierarchy For A Single Index

1. If “CONTAINS”, use word-index
2. Unique index with all components involved in the equality matches
3. Most active equality matches
4. Most active range matches
5. Most active sort matches
6. The primary index
7. First index alphabetically by index name*

Multiple Index Usage

- ❖ If indexes are available for both sides of *WHERE... AND/OR*, more than one index can be used
- ❖ Multiple indexes will **only** be used to assist in bracketing* records
- ❖ **Can still be a bracket of 1 record*
- ❖ Return order **not** guaranteed

WHERE...AND...

- ❖ *WHERE* clause includes the use of *AND*
- ❖ All components of each index are involved in equality matches
- ❖ No unique indexes are involved

WHERE...OR...

- ❖ **WHERE** clause includes the use of **OR**
- ❖ Both the left and right side of the OR contain at least the lead component of an index
- ❖ These lead components are involved in either equality or range matches

What's Next...

- ❖ Prepare
- ❖ Predict
- ❖ Prevent

Breaking The Engine

USE-INDEX ...

FIND ... WHERE ROWID () = ...

Breaking The Rules

WHERE NOT ...

WHERE <non-indexed field> = ...

WHERE ... MATCHES ...

WHERE IF ...

THEN ... = vclInput

WHERE SUBSTRING (...) = "A"

ELSE TRUE

Breaking The Engine

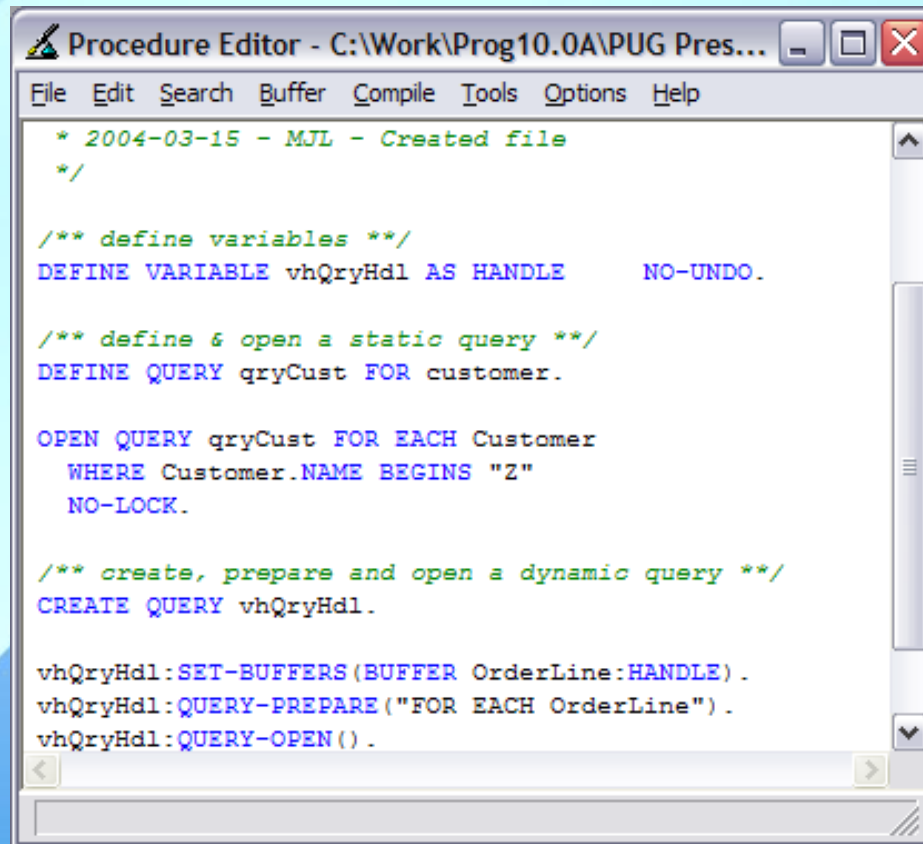
FOR EACH table WHERE unique-field EQ ?
BY non-unique-field:

- ❖ Equality queries on unique indices ignore sorting (since only 1 record should be returned)
- ❖ Use “unique-field GE ?” to change to range match

Breaking Data

- ❖ Ability to have multiple records with “?” unknown value in unique indices is a side effect
- ❖ Sorts differently when in an index field vs. non-index field.
- ❖ Use only EQ and NE in comparisons or face frustration
- ❖ See KB 15969 and P4130 for more details

Demo - UniqueProblems.p



```
Procedure Editor - C:\Work\Prog10.0A\PUG Pres...
File Edit Search Buffer Compile Tools Options Help

* 2004-03-15 - MJL - Created file
*/

/** define variables **/
DEFINE VARIABLE vhQryHdl AS HANDLE      NO-UNDO.

/** define & open a static query **/
DEFINE QUERY qryCust FOR customer.

OPEN QUERY qryCust FOR EACH Customer
  WHERE Customer.NAME BEGINS "Z"
  NO-LOCK.

/** create, prepare and open a dynamic query **/
CREATE QUERY vhQryHdl.

vhQryHdl:SET-BUFFERS(BUFFER OrderLine:HANDLE).
vhQryHdl:QUERY-PREPARE("FOR EACH OrderLine").
vhQryHdl:QUERY-OPEN().
```

What's Next...

- ❖ Prepare
- ❖ Predict
- ❖ Prevent
- ❖ Plan

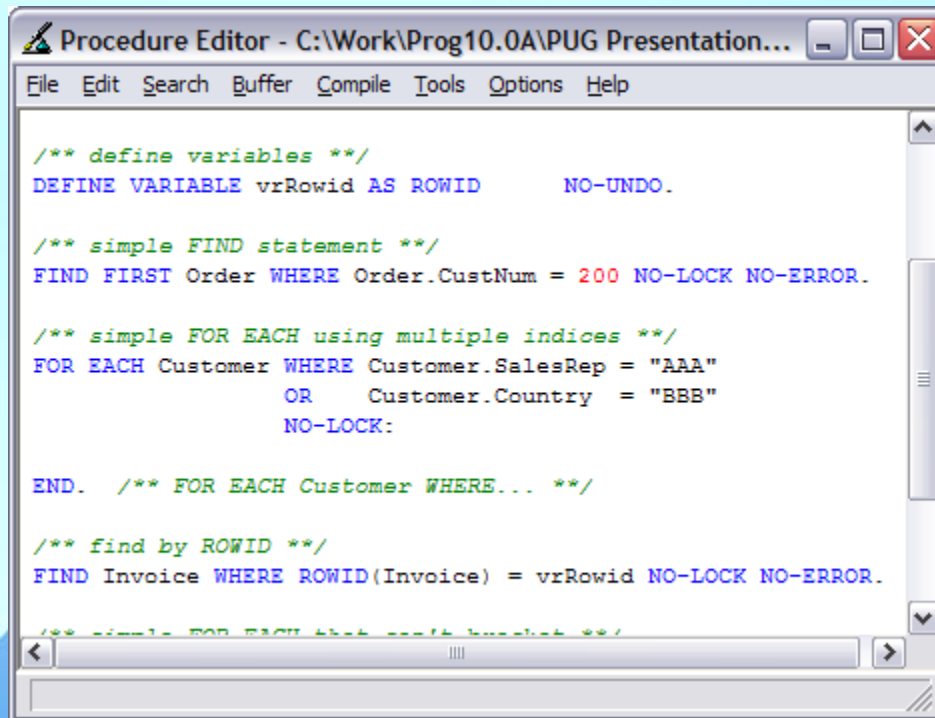
XREF Information

COMPILE prog.p XREF prog.xref

FIND

FOR EACH

Demo - XREF-demo.p



```
Procedure Editor - C:\Work\Prog10.0A\PUG Presentation...
File Edit Search Buffer Compile Tools Options Help

/** define variables **/
DEFINE VARIABLE vrRowid AS ROWID      NO-UNDO.

/** simple FIND statement **/
FIND FIRST Order WHERE Order.CustNum = 200 NO-LOCK NO-ERROR.

/** simple FOR EACH using multiple indices **/
FOR EACH Customer WHERE Customer.SalesRep = "AAA"
                    OR Customer.Country = "BBB"
                    NO-LOCK:

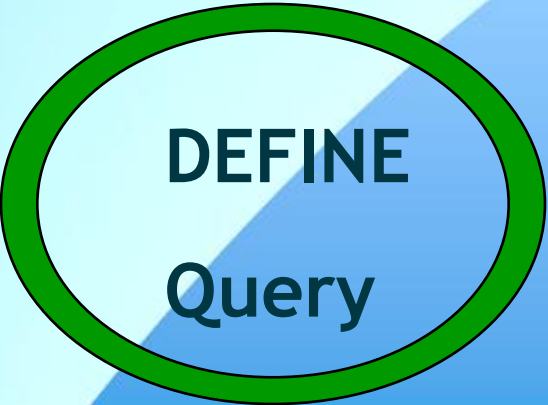
END. /** FOR EACH Customer WHERE... **/

/** find by ROWID **/
FIND Invoice WHERE ROWID(Invoice) = vrRowid NO-LOCK NO-ERROR.


/** simple FOR EACH that uses a bucket **/
```

XREF Information

COMPILE prog.p XREF prog.xref

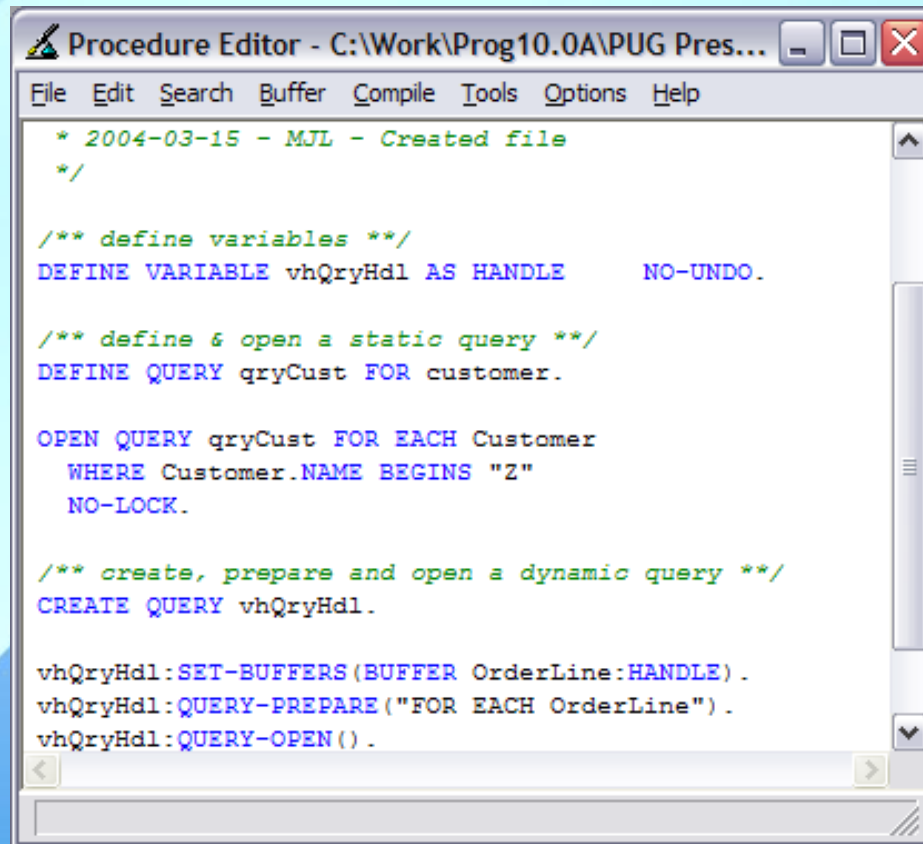


DEFINE
Query



CREATE
Query

Demo - Query-demo.p



The image shows a screenshot of a 'Procedure Editor' window. The title bar reads 'Procedure Editor - C:\Work\Prog10.0A\PUG Pres...'. The menu bar includes 'File', 'Edit', 'Search', 'Buffer', 'Compile', 'Tools', 'Options', and 'Help'. The main text area contains the following code:

```
* 2004-03-15 - MJL - Created file
*/

/** define variables **/
DEFINE VARIABLE vhQryHdl AS HANDLE      NO-UNDO.

/** define & open a static query **/
DEFINE QUERY qryCust FOR customer.

OPEN QUERY qryCust FOR EACH Customer
  WHERE Customer.NAME BEGINS "Z"
  NO-LOCK.

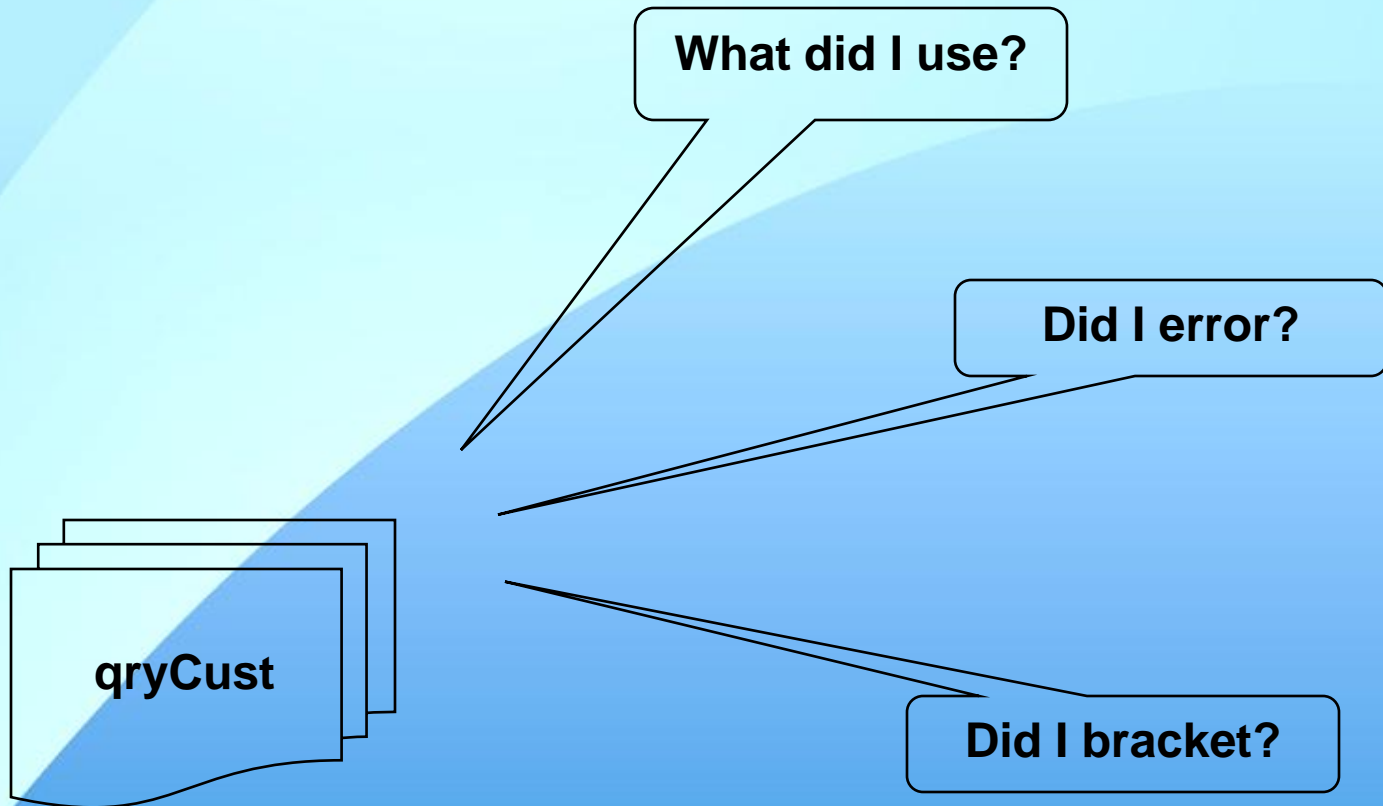
/** create, prepare and open a dynamic query **/
CREATE QUERY vhQryHdl.

vhQryHdl:SET-BUFFERS(BUFFER OrderLine:HANDLE).
vhQryHdl:QUERY-PREPARE("FOR EACH OrderLine").
vhQryHdl:QUERY-OPEN().
```

What's Next...

- ❖ Prepare
- ❖ Predict
- ❖ Prevent
- ❖ Plan
- ❖ Prove

If A Query Could Talk...



How To Ask

```
DEFINE VARIABLE vhQryCust AS HANDLE  
CREATE QUERY vhQryCust
```



```
DEFINE QUERY qCust  
QUERY qCust:
```

```
vhQryCust = QUERY qCust:HANDLE
```

What to Ask

What did I use?

`vhQry:INDEX-INFORMATION()`

Did I error?

`ERROR-STATUS:NUM-MESSAGES`

Did I bracket?

`ENTRY(1, vhQry:I-I) = "WHOLE-INDEX"`



qryCust

Demo - wQueryTest.w

The screenshot shows the 'Query Test Station' application window. It is divided into four main sections: Database / Table Information, Index Information, Query Information, and Query Used.

Database / Table Information

DataBase: sports
Table: Customer

Field Name	Type
Name	character
Phone	character
PostalCode	character
SalesRep	character
State	character
Terms	character

Index Information

Indices: Comments

Primary Unique Word Index

Field Name	Asc?
Comments	Yes

Query Information

FOR EACH Customer WHERE...

Name = "AAA" and SalesRep = "BBB"

Build Query **Exit**

Query Used

FOR EACH Customer WHERE Name = "AAA" and SalesRep = "BBB"

Indices Chosen

Name
SalesRep

What's Next...

- ❖ Prepare
- ❖ Predict
- ❖ Prevent
- ❖ Plan
- ❖ Prove
- ❖ Process

OpenEdge® 10 - Log Manager

- ❖ New features in OpenEdge 10
- ❖ Both command-line and run-time controls
- ❖ Highly configurable

Log Manager Controls

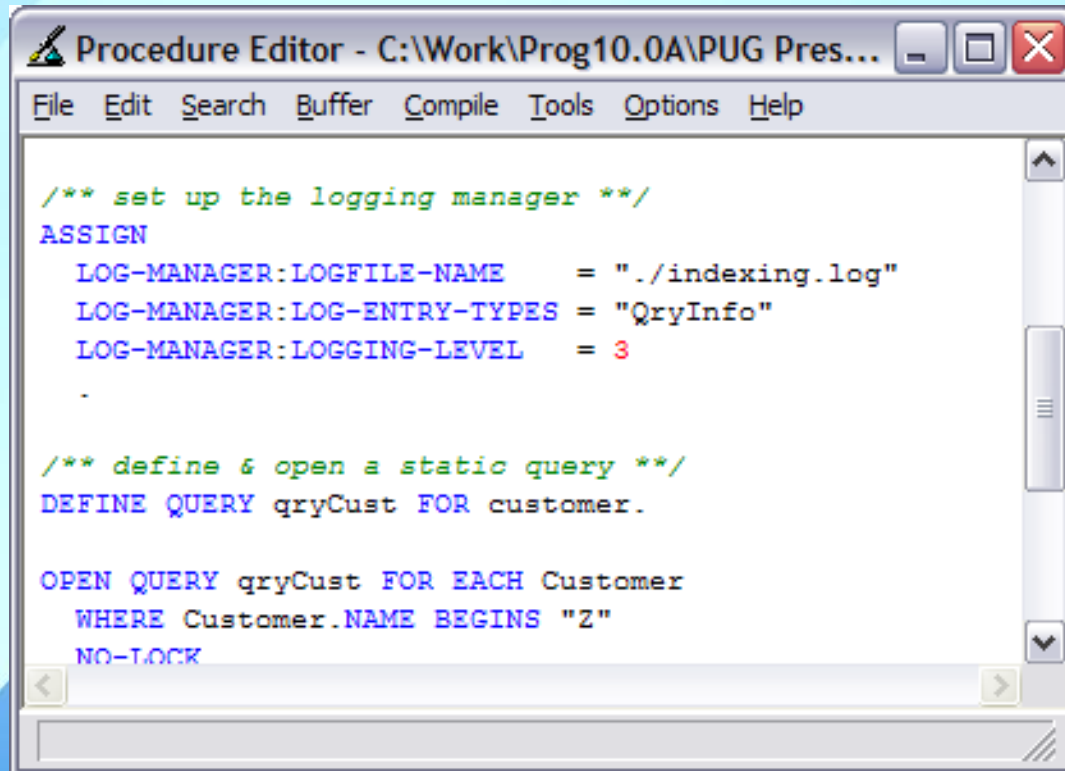
Command Line

- ❖ -clientlog
- ❖ -logentrytypes
- ❖ -logginglevel
- ❖ -numlogfiles
- ❖ -logthreshold

LOG-MANAGER

- ❖ :LOGFILE-NAME
- ❖ :LOG-ENTRY-TYPES
- ❖ :LOGGING-LEVEL
- ❖ :NUM-LOG-FILES
- ❖ :LOG-THRESHOLD

Demo - Logging-demo.p



```
Procedure Editor - C:\Work\Prog10.0A\PUG Pres...
File Edit Search Buffer Compile Tools Options Help

/** set up the logging manager */
ASSIGN
  LOG-MANAGER:LOGFILE-NAME      = "./indexing.log"
  LOG-MANAGER:LOG-ENTRY-TYPES  = "QryInfo"
  LOG-MANAGER:LOGGING-LEVEL    = 3
.

/** define & open a static query */
DEFINE QUERY qryCust FOR customer.

OPEN QUERY qryCust FOR EACH Customer
  WHERE Customer.NAME BEGINS "Z"
  NO-LOCK
```

**Now that I've rambled on,
are there any questions?**



PUGCentral.org for files

Thanks for attending...