

---

# SQL92 for a Progress DBA

**Bob Brennan**  
Integrated Manufacturing Systems, Inc.

# Agenda:

Goal: *Make you successful with SQL applications!*

- OpenEdge SQL
  - Component overview
  - Initial connection
- OpenEdge database
  - Setup and maintenance
    - Control and performance
- OpenEdge SQL specifics
  - Tools
  - Applications

# Use Case

- Open Edge Application
- Users want to use ODBC Compliant Tools
- Crystal Report, EXCEL
- How to care and tend to this added responsibility

**ABL:**  
(ABL works with relational DBs)

Open Clients:  
Java™  
.NET™  
Web services

**OpenEdge SQL**

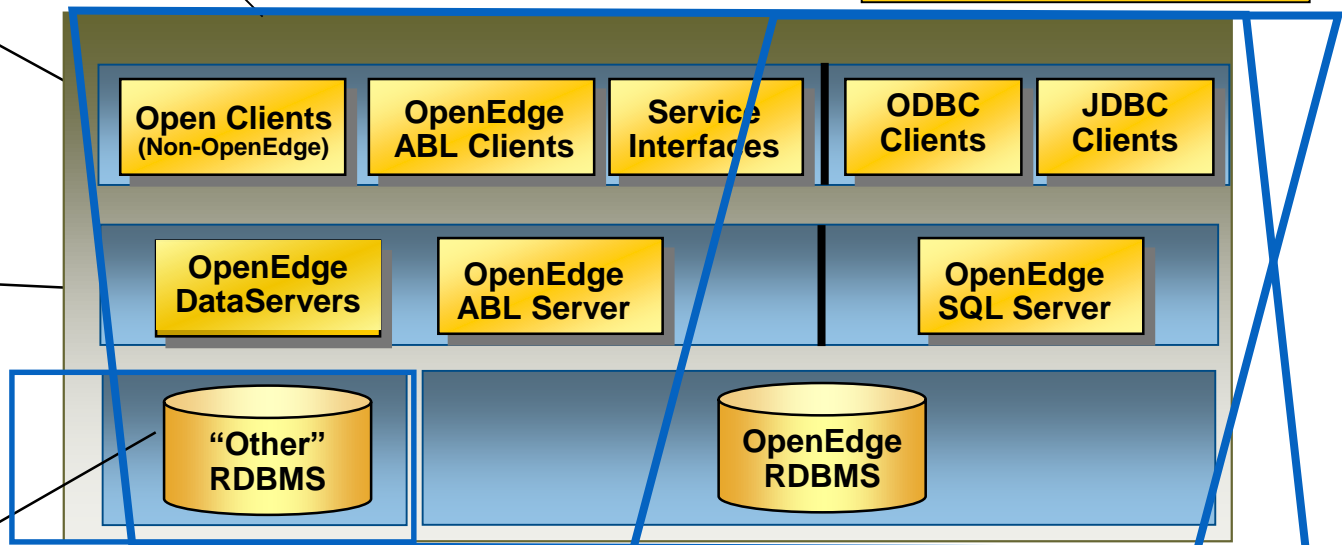
(works with OpenEdge RDBMS)

Crystal Reports  
WebSphere®  
JBOSS / JRun  
Java / JDBC apps  
J2EE™ / JTA  
.NET / ODBC apps  
ADO.NET / VB

.NET  
Java  
HTML

SSL  
HTTP  
HTTP/S  
HTML

Oracle®  
MSSQL  
ODBC



Data is fully interoperable: ABL & SQL

# Client Side Drivers v10+

%DLC%\install\odbc\sql-odbc-setup.exe (ODBC)  
ds-setup.exe (Data Server Bits)

NetSetup on a Client

SQL Client Access also available from ESD

Client does not need to me on server.

DB = AIX and Client on Windows

# Platforms and Keys

Platform	Serial Number	Control Number
Windows 32-bit	006013399	Z8CS9 TQE6D 5MGYP
Windows 64-bit	006013408	Z8DRH TPM6D N4GYM
HP-UX Itanium 64-bit	006013363	X8HRS 2QM2C 5MDYR
IBM AIX 64-bit	006013370	X9CSH 2PM6C 54CYM
Sun Solaris 64-bit	006013391	X9GR? 2QP6D 54GYR
Linux 32-bit	006013377	X8CRH TPP6C N4DYM
Linux 64-bit	006013384	X9HRS TQP6C N4G?M

# ODBC Manager

32 Bit Manager

C:\Windows\SysWOW64\odbcad32.exe

64 Bit Manager

C:\Windows\System\odbcad32.exe

# Client and the ODBC Connection

The Client application and the ODBC DSN must match bit sizes to work.

32 Bit Client + 32 Bit ODBC = Works

64 Bit Client + 64 Bit ODBC = Works

32 Bit Client + 64 Bit ODBC = Mismatch Failure

64 Bit Client + 32 Bit ODBC = Mismatch Failure



# ODBC: Multi-Database configuration

**See kbase article 2767 for details**

1. Only Primary DB can change. All Other DBs are Read Only.
2. Must specify the full path for the databases when the databases are started via the AdminServer / Progress Explorer.
3. All databases need to be of the same codepage

# ODBC DSN – single connection

ODBC Progress OpenEdge Wire Protocol Driver Setup ? X

General Advanced Security Failover About

Data Source Name: PUG Help

Description: Pug Demo DB

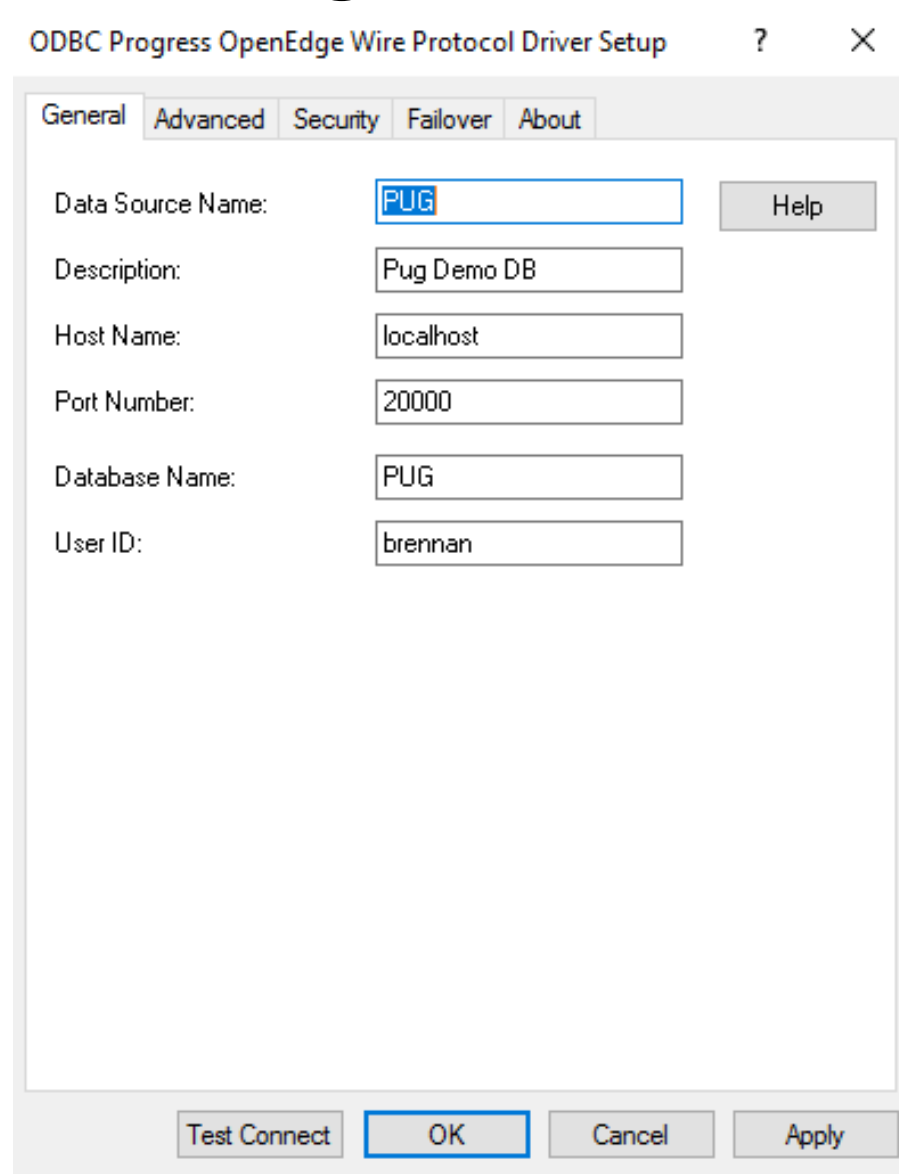
Host Name: localhost

Port Number: 20000

Database Name: PUG

User ID: brennan

Test Connect OK Cancel Apply



# ODBC DSN - Advanced Tab

The image shows a Windows dialog box titled "ODBC Progress OpenEdge Wire Protocol Driver Setup". The "Advanced" tab is selected, showing various configuration options for the driver. The "Default Isolation Level" is set to "1 - READ COMMITTED". Other settings include "Fetch Array Size" (50), "Login Timeout" (15), "Query Timeout" (0), and "Lock Wait Timeout" (empty). There are checkboxes for "Enable Timestamp with Time Zone" (checked), "TCP Keep Alive" (unchecked), and "Use Wide Character Types" (unchecked). An "Extended Options" text box is empty. At the bottom, there are buttons for "Test Connect", "OK", "Cancel", and "Apply".

ODBC Progress OpenEdge Wire Protocol Driver Setup

General Advanced Security Failover About

Default Isolation Level: 1 - READ COMMITTED Help

Fetch Array Size: 50

Login Timeout: 15

Query Timeout: 0

Lock Wait Timeout:

Enable Timestamp with Time Zone  TCP Keep Alive

Use Wide Character Types

Extended Options:

Test Connect OK Cancel Apply

# Dirty Read

A dirty read occurs when one user is updating or inserting a record while a different user is reading it, but the work is not yet committed to the database.

**User A executes:**

```
INSERT INTO State (state, state_name, region)
VALUES ('ME', 'Maine', 'Northeast');
```

**User B executes:** SELECT \* FROM State;

**User B sees:** state 'ME'

**User A executes:** ROLLBACK WORK; **User B has seen data that really did not exist.**

# Non-Repeatable Read

Occurs when one user is repeating a read operation on the same records but has updated values.

```
User A executes: SELECT * FROM State;
```

```
User B executes:
```

```
UPDATE State
```

```
SET state_name = 'Arkansas'
```

```
WHERE state = 'AK';
```

```
COMMIT WORK;
```

```
SELECT * FROM pub.State
```

```
User A re-executes:
```

```
SELECT * FROM State;
```

```
User A has now updated records in the result set.
```

# Phantom Read

Occurs when one user is repeating a read operation on the same records, but has new records in the results set.

**User A executes:**

```
SELECT * FROM State;
```

**User B executes:**

```
INSERT INTO pub.State (state, state_name, region)
VALUES ('CT', 'Connecticut', 'Northeast');
COMMIT WORK;
```

**User A re-executes:**

```
SELECT * FROM pub.State;
```

**User A has new records in the results set.**

# Default Isolation Settings

Isolation	Dirty read	Nonrepeatable read	Phantom read
READ UNCOMMITTED	Permitted	Permitted	Permitted
READ COMMITTED	Prevented	Permitted	Permitted
REPEATABLE READ	Prevented	Prevented	Permitted
SERIALIZABLE	Prevented	Prevented	Prevented

# Default Isolation Levels

- `READ_UNCOMMITTED`, other processes can be read from the database. Only modified data is locked and is not released until the transaction ends. [FOR EACH NO-LOCK]
- `READ_COMMITTED`, other processes can change a row that your application has read if the cursor is not on the row you want to change. This level prevents other processes from changing records that your application has changed until your application commits them or ends the transaction.



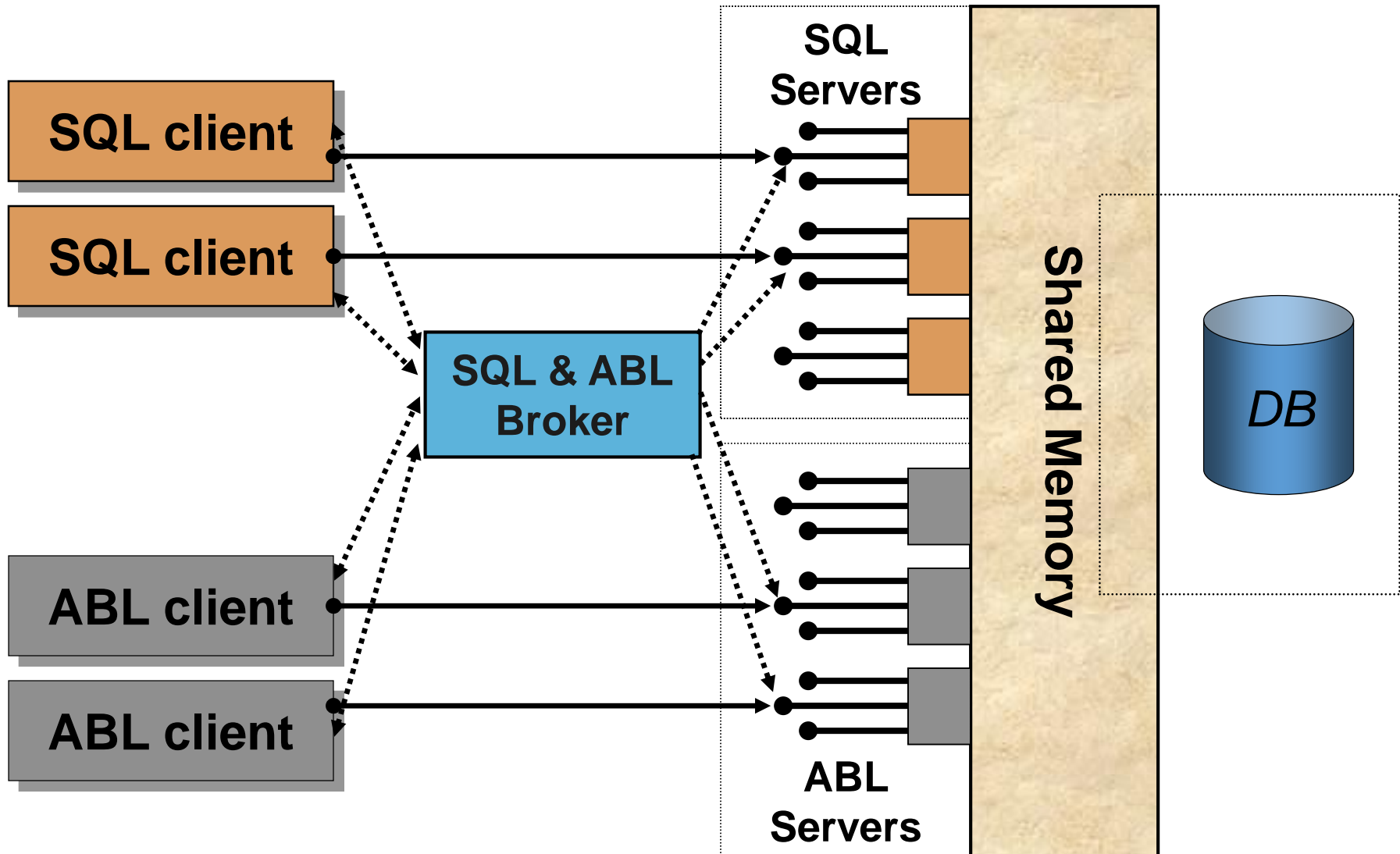
# Default Isolation Levels

- REPEATABLE\_READ, other processes are prevented from accessing data that your application has read or modified. All read or modified data is locked until transaction ends. [FOR EACH EXCLUSIVE-LOCK]
- SERIALIZABLE, other processes are prevented from changing records that are read or changed by your application (including phantom records) until your program commits them or ends the transaction. This level prevents the application from reading modified records that have not been committed by another process.

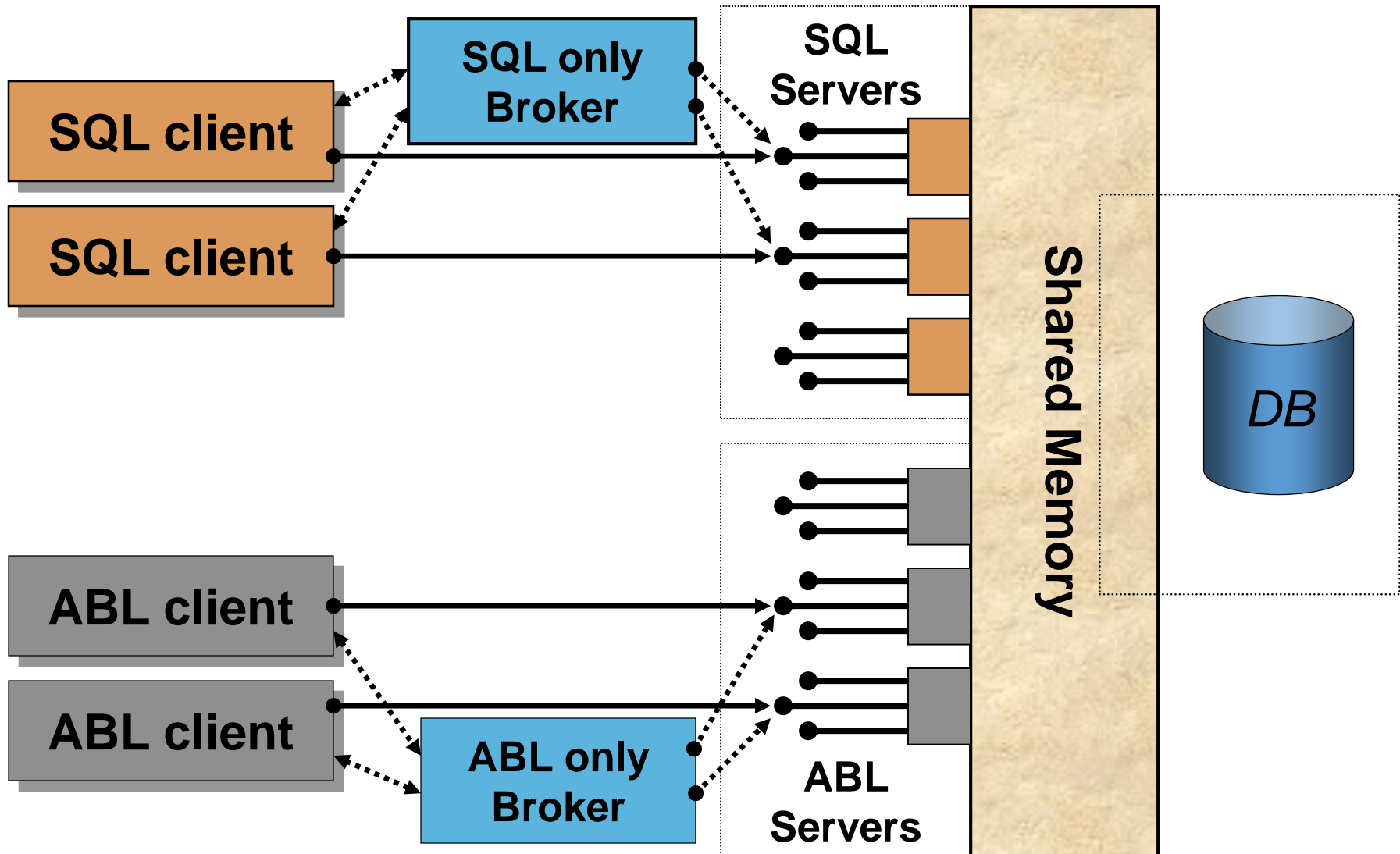
# Connection – server side



# Default server setup



# “Recommended” server setup



# Recommended parameters example

## *Separating ABL and SQL brokers/servers example*

Start a primary ABL broker

```
proserve myDB  
-S 6000 -H localhost  
-ServerType 4GL  
-Mi 1 -Ma 5  
-minport 6100  
-maxport 6300  
-Mpb 4  
-Mn 8 -B<n> -L<n> . . .
```

Start a secondary SQL broker

```
proserve myDB  
-S 5000 -H localhost  
-ServerType SQL  
-Mi 5 -Ma 10  
-minport 5100  
-maxport 5300  
-Mpb 2 -m3
```



Secondary broker uses 1 -Mn for itself

# Verify the Math

- [SQL92 ODBC Parameter Tool](#)

# Security

*Authenticate then authorize*

## *Authentication*

- Identify who I am
- Validate I am who I say I am

## *Authorization*

- Being I am who I say I am...
- What am I allowed to do



# Security

## *ID and passwords*

- Database authentication
  - Performed at connection
  - SQL
    - User ID and password required to authenticate
  - ABL
    - User ID and password NOT required





# SQL Authentication - Am I who I say I am?

## *ID and passwords scenarios*

### Case 1: *Users have not been created* *(no rows in \_User table)*

- No password validation at connection
  - You are not “authenticated”
  - You can pretend to be someone you are not
- Unable to do much (not authorized) **UNLESS**
  - You created the database
  - You know an authorized user
    - Database creator
    - Someone who was granted privileges

# SQL Authentication - Am I who I say I am?

*ID and passwords scenarios...cont'd*

Case 2: ***Users have been created***

***(rows exist in `_User` table)***

- Password validation at connection
- Valid users defined by a DBA or SA
  - Can be created by SQL **or** ABL
  - Make sure existing SQL DBA has userid/pswd
- Invalid login/password error message:
  - "Access denied (Authorisation failed). (8933)

# Authorization – What am I allowed to do?

- SQL follows GRANT security model
  - By default, a connected userid is not *authorized* to do anything.
  - *Exceptions:*
    - the DBA account (*full operations*)
    - the TABLE owner
- DBA controls operation privileges with GRANT / REVOKE syntax

# Encountering data access errors

```
select count(*) from customer;
```

Access denied (Authorization failed) (7512)

- Possible reasons for this:
  - ***No authorization privileges***
  - *Schema scope*



# Authorization – What can I do?

*Privileges syntax: GRANT (2 types)*

- Database wide (system admin or general creation)

```
GRANT { DBA, RESOURCE }  
      TO ser_name [, user_name ] , ...;
```

*Can do anything*

*Can CREATE stuff*

# Authorization – What can I do?

## *Privileges syntax: GRANT (2 types)*

- Database wide (system admin or general creation)

```
GRANT { DBA, RESOURCE }  
      TO user_name [, user_name ] , ...;
```

- For specified Tables or Views

```
GRANT { privilege [, privilege ], ... | ALL }  
      ON table_name  
      TO { user_name [, user_name ], ... | PUBLIC }  
      [ WITH GRANT OPTION ];
```

- Where ‘privilege’ is:

```
{ SELECT | INSERT | DELETE | INDEX |  
  UPDATE [ ( column , column , ... ) ] |  
  REFERENCES [ ( column , column , ... ) ] }
```

# Authorization – What can I do?

## *Example Syntax: GRANT*

- For specified Tables or Views - Example

```
GRANT select ON PUB.Customer  
          TO bob;
```

```
GRANT select ON PUB.Order-line  
          TO PUBLIC;
```

```
COMMIT WORK;  
ROLLBACK WORK;
```

- See PSDN whitepaper on authorization for additional details.

# Demo Database Grant Routine

```
proenv>sqlexp -db PUG -H localhost
          -S 20000      -user sysprogress
          -password sysprogress
          -infile grantall.sql
          -outfile sqlout.txt
```

```
create user 'sqluser', 'readonly123';
```

```
///// grantall.sql  READONLY  User
```

```
grant select on pub.Benefits          to sqluser;
```

```
grant select on pub.BillTo           to sqluser;
```

```
grant select on pub.Bin              to sqluser;
```

```
Commit work;
```



# Encountering data access errors

```
select count(*) from customer;
```

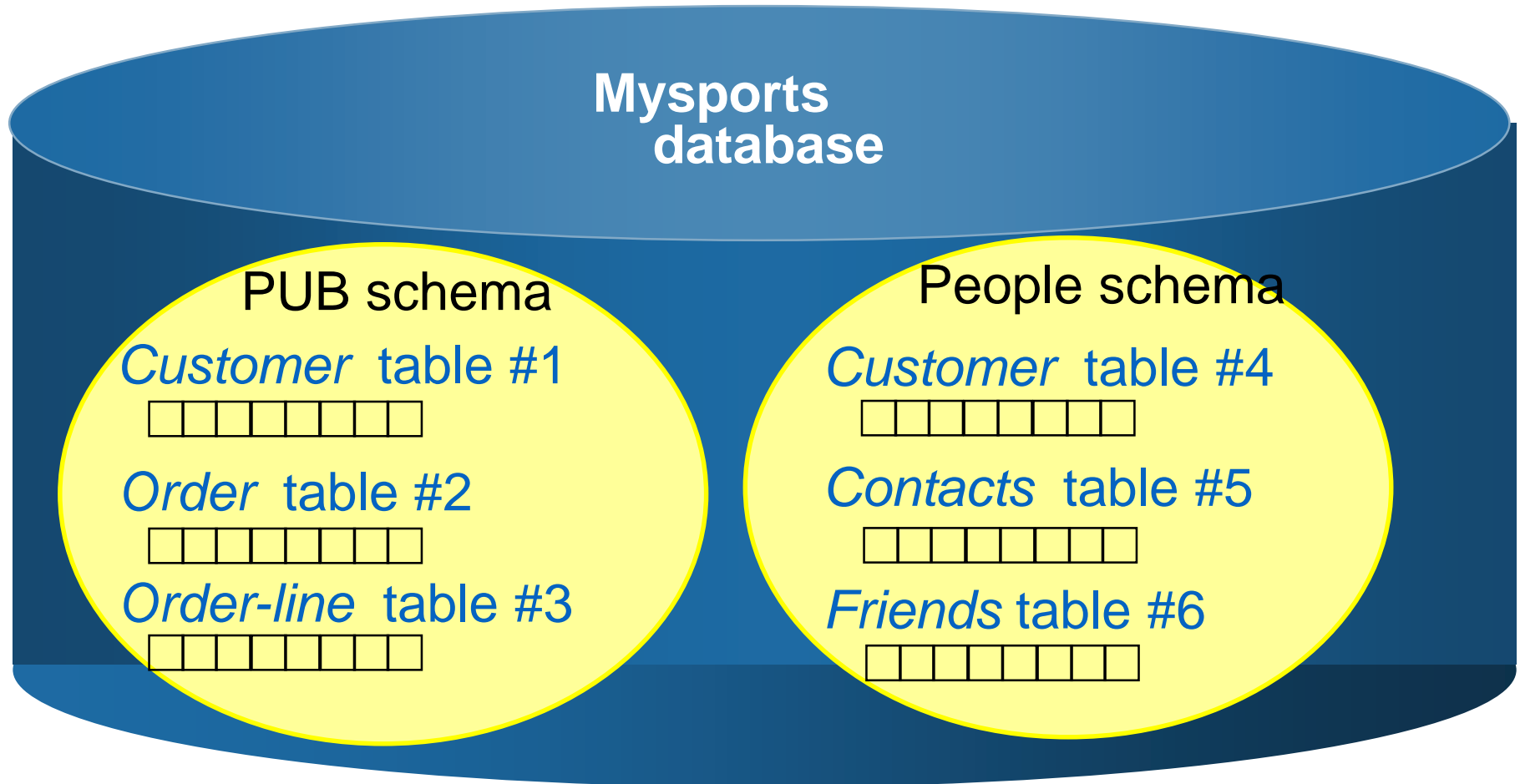
Access denied (Authorization failed) (7512)

- Possible reasons for this:
  - *No authorization privileges*
  - ***Schema scope***



# Schema: a logical grouping

In the “SQL world” **schema** is NOT meta data nor is it “Area 6”



# Schemas

## *What is a default schema?*

- Users have a default schema attached to their ID

<userid>.<table>      oeuser.customer

- ABL uses one “hidden” schema – ‘PUB’
  - Use PUB.customer for access from SQL
- Avoiding schema qualification in SQL:

```
CREATE PUBLIC SYNONYM customer FOR  
pub.customer;
```

- Can set it as a registry entry in ODBC dsn definition

# Solving data access errors

```
select count(*) from customer;
```

Access denied (Authorization failed) (7512)

*Solution #1:*

```
SELECT count(*) FROM myschema.customer;
```

*Solution #2:*

```
SELECT count(*) FROM pub.customer;
```

*Solution #3:*

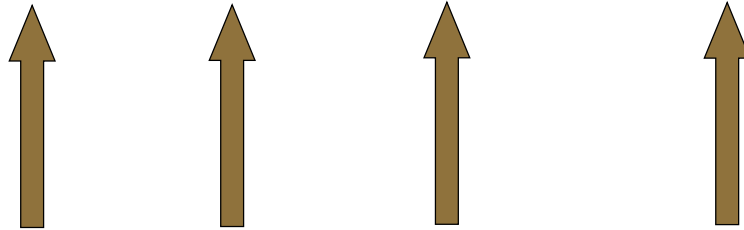
```
SET SCHEMA 'pub' ;  
SELECT count(*) FROM customer;
```

# 4 Part Naming – Multi-Database Query

*Fully Qualified Names – catalog is database name*

- Four level naming convention

*catalog.schema.table.column-name*



# 4 Part Naming – Multi-Database Query

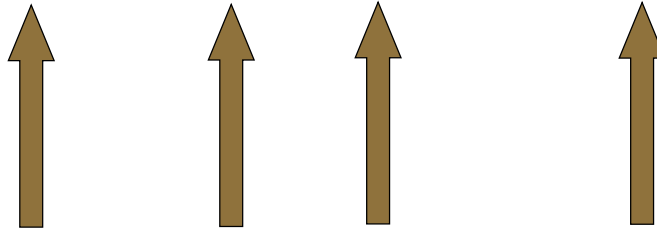
*Fully Qualified Names – catalog is database name*

- Four level naming convention

```
catalog.schema.table.column-name
```

- Example

```
SELECT Pub.Customer.CustNum,  
SportsPrimary.Pub.Customer.Name,  
SportsAux1.Pub.Order.OrderNum ...
```



# 4 Part Naming – Multi-Database Query

*Fully Qualified Names – catalog is database name*

- Four level naming convention

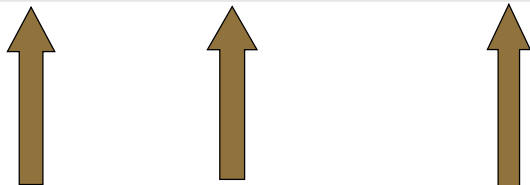
*catalog.schema.table.column-name*

- Example

```
SELECT Pub.Customer.CustNum,  
       SportsPrimary.Pub.Customer.Name,  
       SportsAux1.Pub.Order.OrderNum ...
```

- ABL has 3 level naming convention

*catalog.table.column-name*



# OpenEdge Specifics

SQL is a standard,

but each vendor has it's own dialect





# OpenEdge SQL Specifics - Quoting

## *Non-SQL standard names*

- Hyphenated names:

```
SELECT cust-num FROM PUB.Customer;
```

Column CUST cannot be found (13865)

- Solution: quoting (double quotes)

```
SELECT "cust-num" FROM PUB.Customer;
```

*Most reporting applications will do this automatically.*

# Overstuffed fields - error

- ABL allows more data than column definition
- SQL restricted to `_field._sql-width` value

```
SELECT name from PUB.customer;
```

Column name in table PUB.customer has value exceeding it's max length.

## ■ Solution:

*Fix `_sql-width` via SQL "ALTER TABLE or Data Dictionary*

```
ALTER table... ALTER column... SET PRO_SQL_WIDTH <value>;
```

# OpenEdge Specifics - Overstuffed fields

Strategies for managing:

- Dbtool : percentage option (\$DLC/bin/dbtool)

1. SQL Width & Date Scan w/Report Option

2. SQL Width Scan w/Fix Option

Choice: 2

<connect>: (0=single-user 1=self-service >1=#threads)? 3

Padding % above current max: 25

<table>: (Table number or all)? all

<area>: (Area number or all)? all

# OpenEdge Specifics - Overstuffed fields

Strategies for managing

- ABL client startup parameter

```
<progress-client> -checkwidth n
```

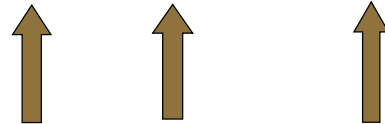
where "n" can be one of the following:

0 — Ignore: Default is to ignore `_width` value.

1 — WARNING: Store the data and generate a warning.

2 — ERROR: Do not store data and generate an error.

.lg and screen: *Width of data is greater than x.customer.Name \_width.*



# Yet More Solutions

- **Authorized Data Truncation (ADT) 11.5.1**
  - A SQL client parameter that returns data values truncated to the size noted in the meta schema.
  - Alternative is an error gets thrown and the record is not included
  - Both a Server and Client Parameter
  - Client takes precedence.
- **Autonomous Schema Update (ASU) 11.6**
  - Builds on ADT
  - Record expanded but truncated data returned first pass
  - On second read, full data returned.

# OpenEdge Specifics – Arrays / Extents

- Selecting array columns as a whole

```
SELECT quarterlySales from PUB.MySales;
```

Result: semi-colon separated **varchar** value

102332.67;330002.77;443434.55;333376.50



- Selecting array column individually – SQL99 (10.1a)

```
SELECT quarterlySales[1] from PUB.MySales;
```

Result: numeric value

102332.67

# OpenEdge Specifics – Arrays / Extents

Using views to break out array element

```
CREATE VIEW pubView.QuarterSalesView
      (qS1, qS2, qS3, qS4) AS
SELECT quarterlySales[1], quarterlySales[2],
       quarterlySales[3], quarterlySales[4]
FROM PUB.MySales;
```

```
GRANT select ON pubView.QuarterSalesView
TO PUBLIC;
```

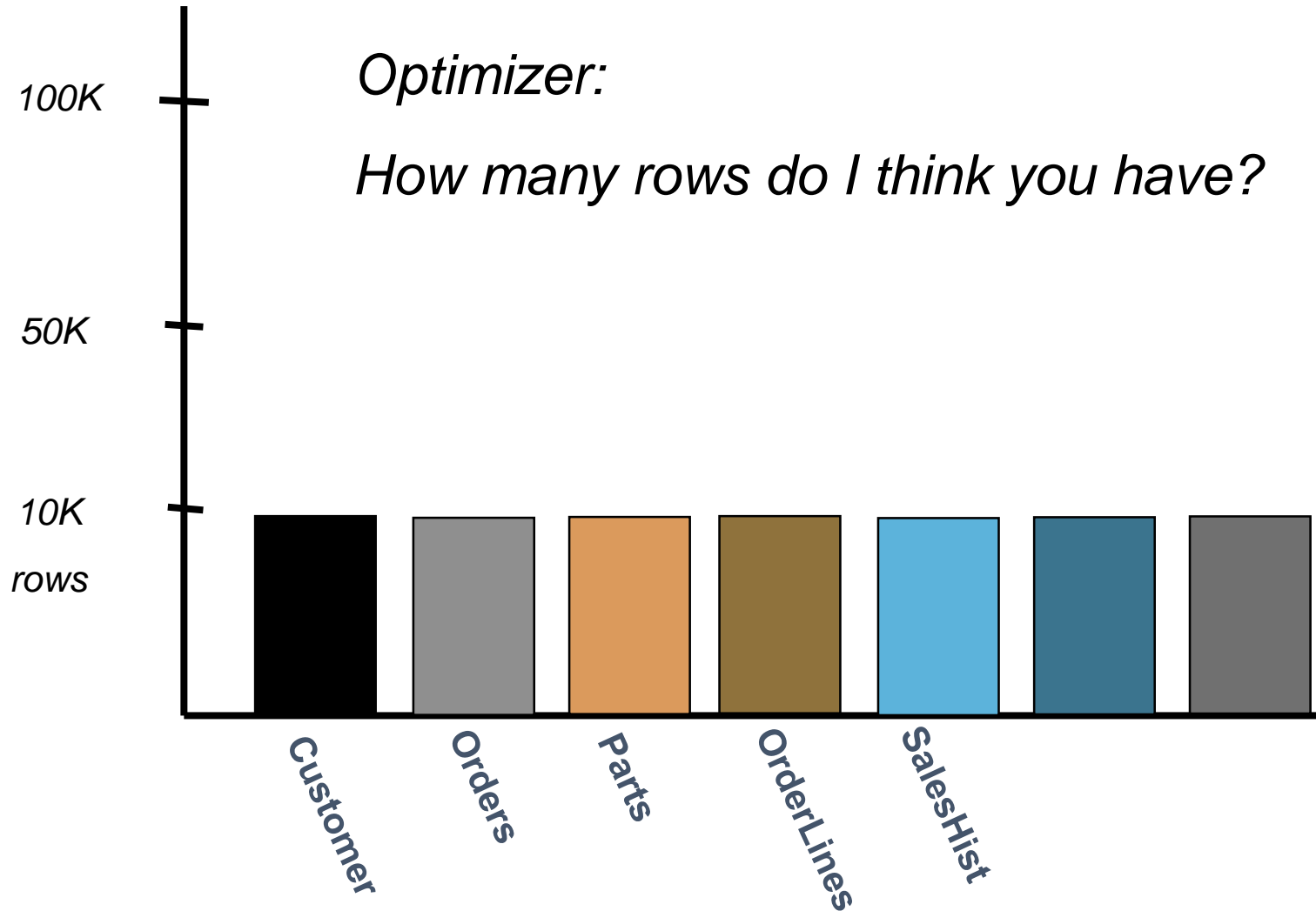
```
SELECT qS1, qS2, qS3, qS4
FROM pubView.QuarterSalesView;
```

Result: numeric values

102332.67 330002.77 443434.55 333376.50

# Basic Performance: What is the cost?

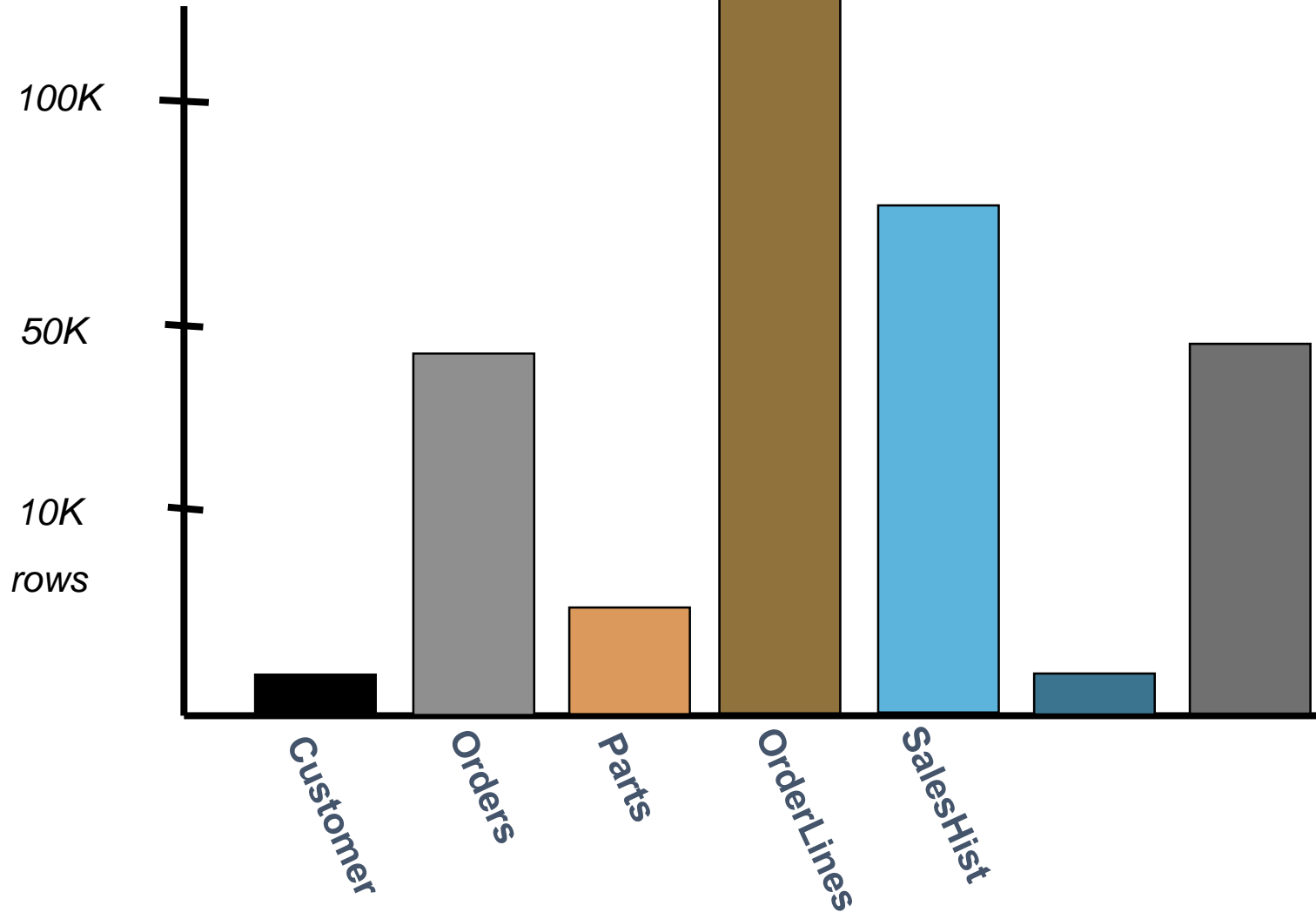
*Database without statistics*





# Basic Performance: Here's the cost.

## *Database with Update Statistics*



# Query Performance: Update Statistics

## *UPDATE STATISTICS syntax*

- All Statistics: Table Cardinality, indexes and all columns

```
UPDATE TABLE STATISTICS AND  
INDEX STATISTICS AND  
[ALL] COLUMN STATISTICS;
```

- Statistics - particular table

```
UPDATE TABLE STATISTICS AND  
INDEX STATISTICS AND  
[ALL] COLUMN STATISTICS  
FOR pub.customer;
```

# Some Questions Answered

- Is there a way to grant a user select privilege for all tables in one statement?
  - There is no way to grant privileges for all tables in one statement. Other than grant dba ;)
- Does the field level extent (array fields) reporting where members are separated by ";" work for ALL data type?
  - yes
- If so, then how is a ";" handled within a character field? Is it quoted when encountered to differentiate between an array entry separator?
  - An embedded semi-colon in a varchar array will be preceded by a ~

# Extra Topics

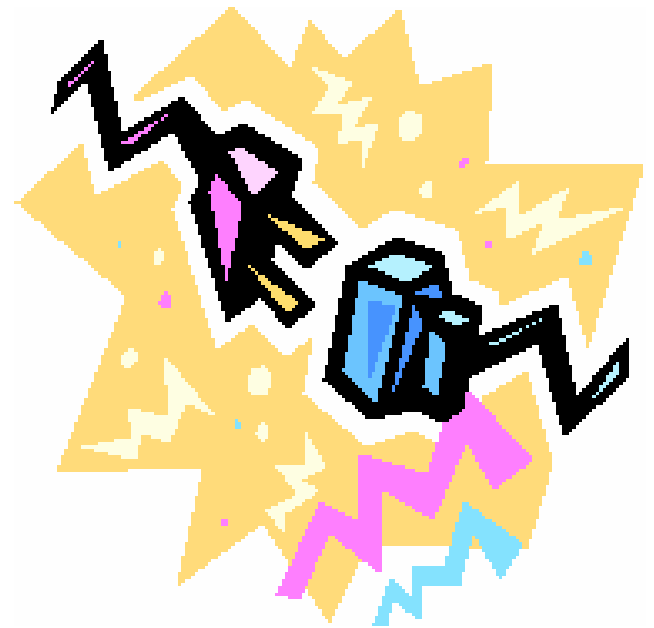
- SQLDUMP /SQLLOAD – Moves data in and out
- SQLSCHEMA – Meta data including table definitions, views, stored procedures including related privileges, and triggers

Use Case, some one set up ODBC for reporting a long time ago. New server or upgrade needing a D&L.

Go Live and all of the reporting breaks.

# In Summary

- Initial SQL connection
- Setup and maintenance in OpenEdge database for security and performance
- Specifics of OpenEdge with SQL applications



**Thank You**