DB2 and PHP

In Depth on IBM i

SEIDEN GROUP

seidengroup.com

Seiden Group and Club Seiden

Alan is a leader and expert in PHP on IBM i; leader, Zend's PHP Toolkit for IBM i: and "Performance guru of PHP on IBM i"

SEIDEN GROUP

Seiden Group is a team of experts available for mentoring/troubleshooting/project advice/development.

seidengroup.com, alan@seidengroup.com

Contact

Alan Seiden

alan@seidengroup.com

201-447-2437

www.SeidenGroup.com
twitter: @alanseiden

Today's discussion will include:

- **Uniqueness of DB2 for IBM i**
- **Which DB2-enabled middleware to use with PHP**
- **Securing your SQL with prepared queries**
- **Connection options for speed and reliability**
  - Persistent connections
  - Library lists
- **What's NEW in ibm_db2**
- **Connecting from "off the box"**
- **Many other tips**

Why learn DB2 best practices?

- **As chief database on IBM i, DB2 runs these:**
  - Most transaction processing systems
  - Stored procedures
  - "XMLSERVICE" Toolkit
    - Accessible with db2 stored procedures from PHP
- **DB2 knowledge will help you:**
  - Maximize speed
  - Reduce CPU usage
  - Maximize reliability
    - Avoid unexpected locking and other operational problems

Heart of IBM i is DB2

- **DB2 built in**
  - Transaction processing workhorse
  - Database implemented below operating system level!
    - IBM i's "Machine Interface (MI)" between OS and hardware
  - Journaling, auditing, commitment control very commonly used
  - Never corrupted
    - Doesn't lose data even if knock out power plug
- **Database often taken for granted**
  - So self-managing, DBAs are rare

Reliable

Nondisruptive business growth

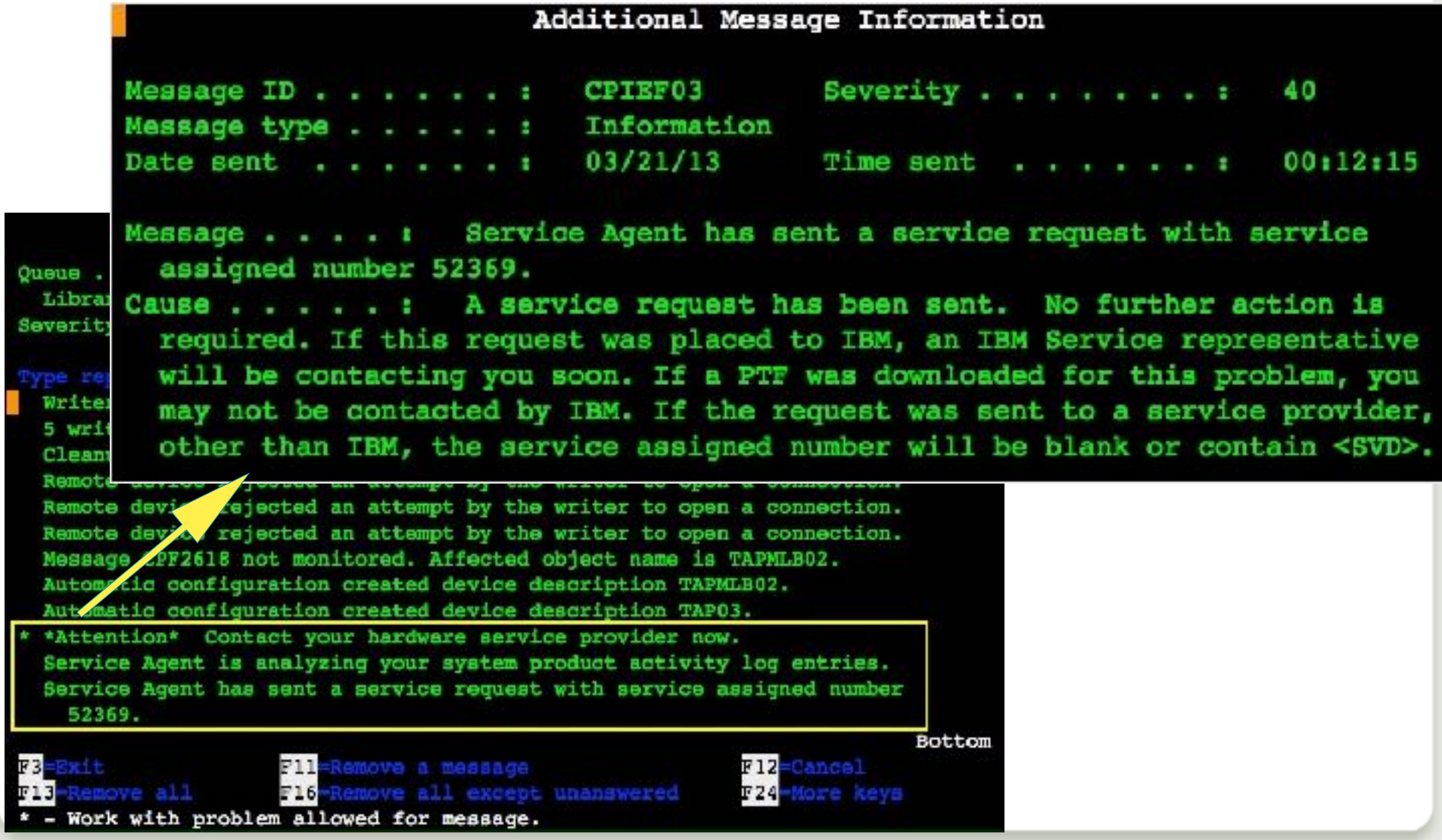
- **Scales vertically**
  - One system can handle large and diverse workloads
    - Total Cost of Ownership (TCO), including reduced operator costs, is said to be competitive or cheaper than assembling server farms
  - Can activate additional processors without restarting system

- **Dependable**

Steve Pitcher @steve_cpitcher
#IDMI is impervious to viruses, malware, hackers, mother-in-laws and the zombie apocalypse. #powersystems

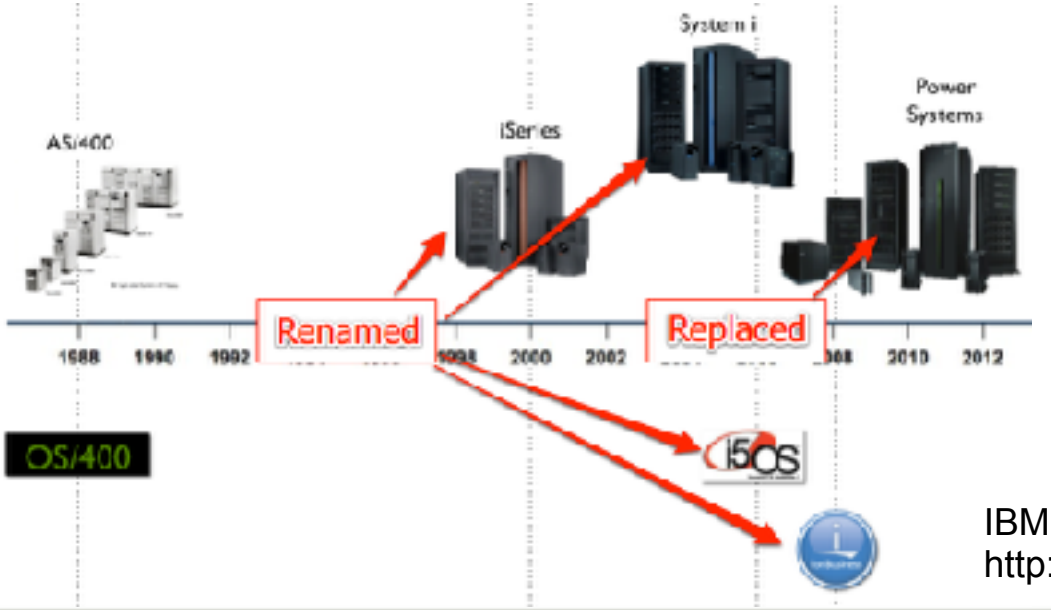
  - Resistant to viruses
    - Object-based system since the 1970s
  - Journaling, commitment control, replication, high availability
  - Security features galore
  - Keeps on running
    - You will sleep soundly at night

IBM i can “phone home”



Data and programs last forever

- **IBM has been dedicated to legacy app longevity and data longevity**
  - RPG (and occasionally COBOL) running for 30-40 years
  - DB2 data evolving 30-40 years
  - Middleware insulates applications from hardware changes



IBM i Heritage chart from Trevor Perry
http://blog.angustheitchap.com/?p=415

Long-serving back ends, new front ends

- **What does this mean to you?**
  - RPG and DB2, mature and evolving for years, can be part of your data model, accessed by PHP
  - Create web GUI interfaces and web services around these venerable resources
    - Business logic is encapsulated in RPG/COBOL/DB2
    - You can keep your hands somewhat clean of business details

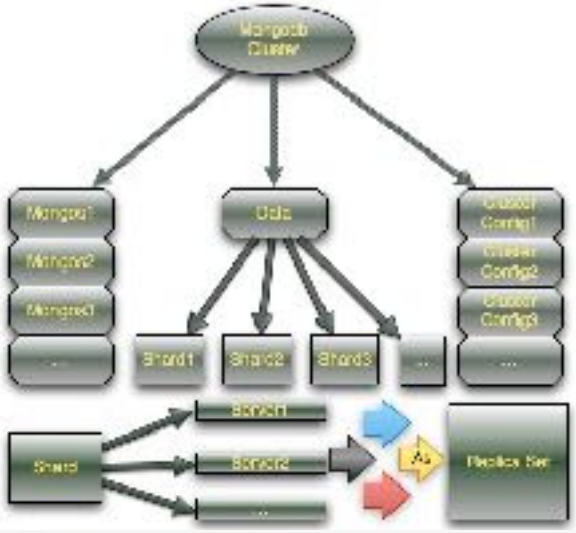
DB2 on IBM i is…

- **Fully integrated business database**
  - Coded at the kernel level (below the OS)
    - Other database systems are .exe files
  - User profiles = real IBM i user profiles
    - Security, logging, auditing consistent throughout the system
    - Other databases have “pretend” users whose security is enforced only by database code
- Always present and available
  - No daemon that can be ended or hacked
  - “Who shut down the database?” Not on IBM i!

DB2 on IBM i is…

- **Indestructible, low maintenance (cost-effective)**
  - Data corruption almost unheard of
  - 30+ years of work for reliability and throughput
  - DBA-less operation. Scale vertically. Add disk and go!

- DB2 on i is **not** this:



Disclaimer: Not singling out other databases as “bad.” A NoSQL database such as Mongo prizes flexibility more than management ease

MongoDB @MongoDB

Sharding Pitfalls Part III: Chunk Balancing and Collection Limits buff.ly/1rgktMc

Database jargon

Modern term	Traditional term or phrase
Schema	Library
Table	File or Physical File
Index	Logical File*
Row	Record
Column	Field

* a logical file resembles an “index + view”

Modern views, triggers, etc. are supported by DB2.

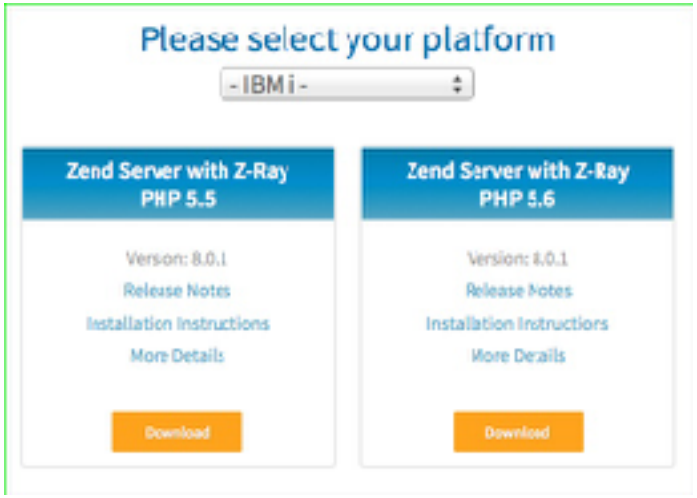
Tip: LF indexes used automatically

- **Logical file keys treated as indexes by SQL**
- **To take advantage of LF "indexes," use ORDER BY, WHERE, and JOIN in your SQL as usual**
  - Indexes from LFs will be chosen automatically as appropriate
  - No need to specify LF in SQL: use physical file/table
- **If your preferred indexes aren't selected by the optimizer, try Visual Explain to learn why.**
  - http://www.ibmsystemsmag.com/ibmi/developer/general/visual_explain/
  - Visual Explain runs in Access Client Solutions (ACS) and the older IBM i Navigator.

Prerequsites for DB2 with PHP

Zend Server for IBM i (8.x)

- **Download Zend Server 8.x**
  - http://www.zend.com/en/products/server/downloads-ibmi
    - Easy upgrade from 6.x
    - Includes ibm_db2 version 1.9.7 with many updates



- **Editions**
  - Basic (free), Professional, Enterprise
  - http://www.zend.com/en/products/server/editions
  - Same download, different license

Zend Server for IBM i (9.x)

- **Download Zend Server 9.x**
  - http://www.zend.com/en/products/server/downloads-ibmi
    - Includes PHP 7.1

- **Editions**
  - Basic (free), Professional, Enterprise
  - http://www.zend.com/en/products/server/editions
  - Same download, different license

Current PTFs within your release

- **Latest DB2 for IBM i group PTF level for your release**
  - 7.1: WRKPTFGRP SF99701
  - 7.2: WRKPTFGRP SF99702
  - 7.3: WRKPTFGRP SF99703
  - For the latest levels: http://www-01.ibm.com/support/docview.wss?uid=nas4PSPbyNum

DB2 is an exciting growth zone for IBM i

- **Behooves you to stay on top of it**
- **Example of update:**
  - http://www.mcpressonline.com/ibm-i-os/400-i5/os/step-right-up-and-hear-about-db2-and-tr9.html
    - Regular expressions in WHERE clause
    - System info available via SQL…joblog, liblist, more

Alan’s current favorite features

A select list of features (1 of 2)

- **SQL Views**
  - Act like a table (SELECT) but implements “virtual” logic
- **User defined functions (UDF)**
  - Create your own scalar function
- **Stored procedures**
  - Efficient, flexible
  - Good place for business logic
  - Run multiple queries if desired

more…

A select list of features (2 of 2)

- **VARCHAR/Trim**
  - Avoid extra spaces caused by fixed-length strings
- **Web service support**
  - DB2 can retrieve/send HTTP data and parse XML

Views

- **Like old "join logical" file and much more**

```
create view presllmstj1
(programcode, programdesc, item, orderbydate, orderbydate_mmddyy, expired) as
select  trim(pccode), trim(pcdesc), pmprod, pmobdate,
mmddyy_slashes(pmobdate),
CASE WHEN CURRENT_DATE > PMOBDATE THEN 1 ELSE 0 END as expired
from PRESLLMST left join PRESLLCDE on PMCODE = PCCODE
```

select * from presllmstj1

PROGRAMCODE	PROGRAMDESC	ITEM	ORDERBYDATE	ORDERBYDATE_MMDDYY	EXPIRED
HNRK	HENDRICKS GIN PRE-SELL	1306920	2015-06-26	6/26/15	1
HNRK	HENDRICKS GIN PRE-SELL	1306930	2015-06-30	6/30/15	1
HNRK	HENDRICKS GIN PRE-SELL	1306900	2015-06-26	6/26/15	1
BJNV	BEAUJOLAIS NOUVEAU15	6641041	2015-06-21	6/21/15	1
HH15	HEAVEN HILL 2015	1544340	2015-07-05	7/5/15	0
HH15	HEAVEN HILL 2015	1544540	2015-07-04	7/4/15	1
HH15	HEAVEN HILL 2015	1544240	2015-07-05	7/5/15	0
JPIN	JP INSIGNIA 15	8218050	2015-07-02	7/2/15	1

User-defined functions (UDF)

- **Create your own function in DB2**

e.g. 6 digit numeric date mmddyy to real date

old way:

```
select date(substr(digits(phdate),1,2) || '/' ||  substr(digits(phdate),3,2) || '/' ||
substr(digits(phdate),5,2)) as podate from podet inner join pohead on
pipo=phpo where piprod = 4317140;


CREATE FUNCTION  mmddyy_to_date (thedate numeric(6,0))
  RETURNS DATE LANGUAGE SQL DETERMINISTIC
  BEGIN
      RETURN  date(substr(digits(thedate),1,2) || '/' ||  substr(digits(thedate),
3,2) || '/' || substr(digits(thedate),5,2));
  END
```

New way:

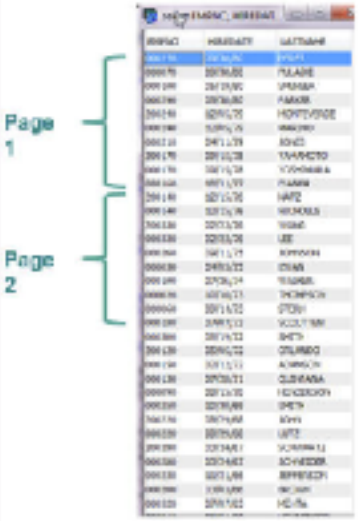
```
select MMDDYY_TO_DATE(phdate) as podate from podet
 inner join pohead on pipo=phpo where piprod = 4317140
```

Stored procedures

- **Most flexible.**
- **Multiple queries, resultsets, call RPG, SQL, all sorts of logic, parameters**

OFFSET and LIMIT for Stateless Pagination

```
CREATE OR REPLACE PROCEDURE
TOYSTORE.FIND_EMPLOYEES
  (IN P_PAGESIZE BIGINT, IN P_OFFSET BIGINT)
    DYNAMIC RESULT SETS 1
    LANGUAGE SQL
BEGIN
  DECLARE V_PREP_STMT1 VARCHAR(4096)  ;
  DECLARE CEMP_RESULT_SET1 CURSOR
     WITH RETURN FOR PREP_STMT1;
  SET V_PREP_STMT1 =
  'SELECT EMPNO, HIREDATE, LASTNAME FROM
     TOYSTORE.EMPLOYEE
     ORDER BY HIREDATE DESC
     LIMIT ? OFFSET ?';
  PREPARE PREP_STMT1 FROM V_PREP_STMT1 ;
  OPEN CEMP_RESULT_SET1 USING P_PAGESIZE,
                              P_OFFSET;
END;

CALL TOYSTORE.FIND_EMPLOYEES(10, 0);
CALL TOYSTORE.FIND_EMPLOYEES(10, 10);
```



Varchar/trim

- **VARCHAR/Trim**
  - Avoid extra spaces caused by fixed-length strings
  - <input type=text name='srvczip' size=9 value=**"43031 "** id="srvczip" maxlength="9">

- **Extra spaces caused by CHAR (fixed length strings)**
- **Use VARCHAR instead for automatic trimming or trim(MyField) in SQL**

Web services

- DB2 can GET/POST HTTP
- Parse XML
- So DB2 can be a web service engine

- See my presentation on “PHP Tricks for RPG developers”

DB2 drivers

Choice of middleware to access DB2

- **Zend Server includes at least three such extensions:**

- **odbc**
  - Less functionality than the others (generic)
  - “Free” connections from other platforms

- **IBM_PDO**
  - PDO = PHP Data Objects
  - Generic DB2. Experimental “/” separator and library lists

- **ibm_db2**
  - Provides IBM i-specific features such as library list support

ibm_db2 documentation

- **Manual page**
  - http://php.net/ibm_db2
- **Source code and additional documentation at the "PECL" PHP extension repository**
  - http://pecl.php.net/package/ibm_db2
  - Read the "C" source sometime—it's educational
- **We will examine ibm_db2 in detail today**

Connect to DB2

Connect with db2_connect()

- **db2_connect() creates a database connection**
  - Accepts four parameters that you should master
    - Three string parameters
    - One array of optional options
- **resource db2_connect (string $database,**
  string $username,
  string $password
  [, array $options])
- **db2_pconnect() is similar**
  - pconnect creates persistent connections (more on that later)

db2_connect() string parameters

- **$database**
  - Leave blank ('') for default local database
  - Use a db name from WRKRDBDIRE for a choice of databases
    - Database name can be *LOCAL or that of an LPAR, IASP (Independent auxiliary storage pool), or another machine
- **$username**
  - Leave blank ('') for default Apache user (QTMHHTTP)
    - Not recommended
  - Use any valid user profile to associate queries with that user
- **$password**
  - Leave blank ('') if $database and $username were blank
  - Otherwise, provide password corresponding to $username

db2_connect() basic examples

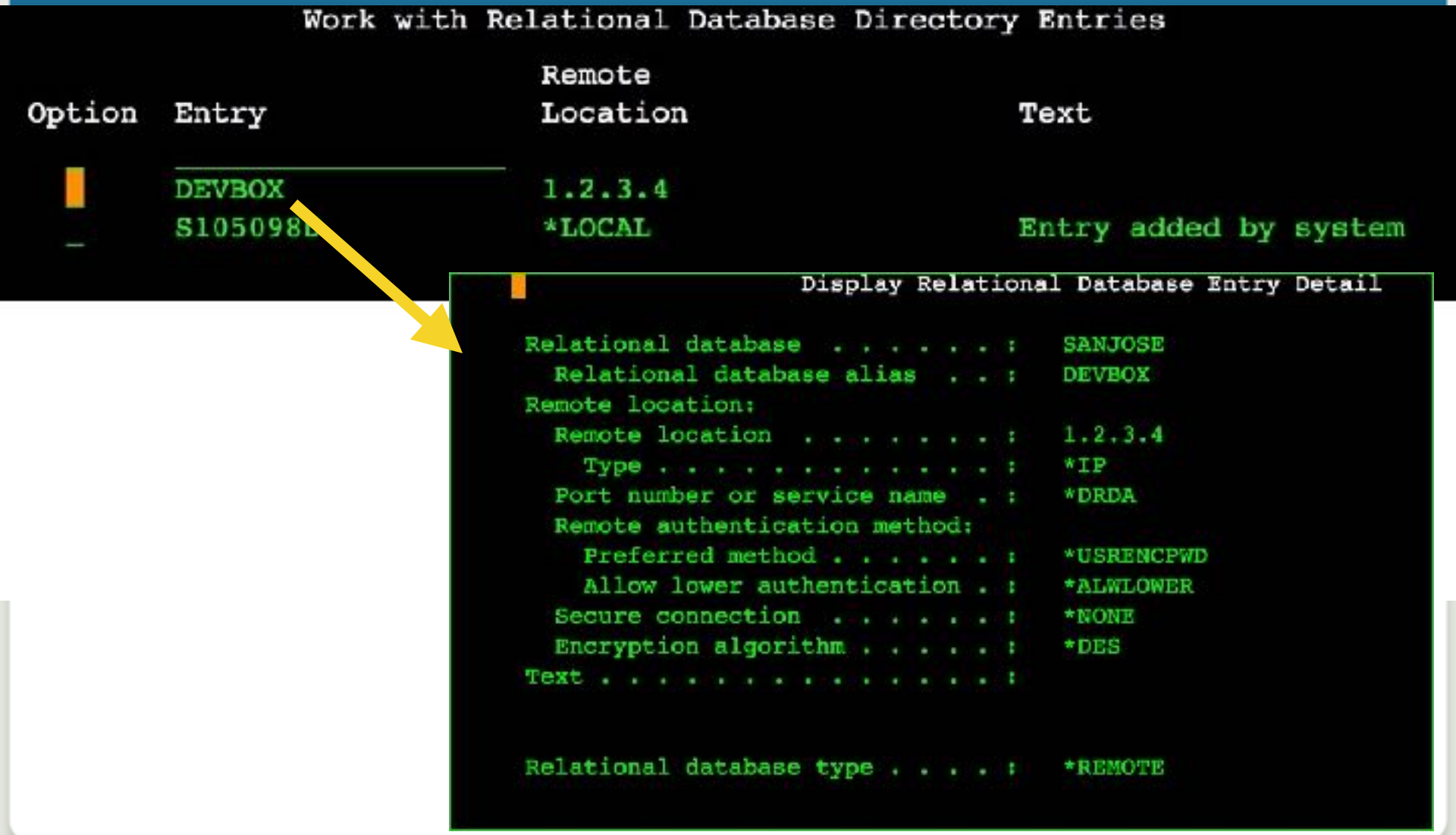
- **Empty params**
  - `$conn = db2_connect('', '', '');`
  - Connects to local database with web user QTMHHTTP
  - Not recommended: may be disallowed in future release
- **Better: use specific values**
  - `$conn = db2_connect('MYDB', 'MYUSER', 'MYPASS');`
  - Connects to MYDB database (must be configured in WRKRDBDIRE) with user MYUSER

How to connect to remote DB or iASP

- Connect to another partition, server, or iASP as if local
- Useful for testing PHP scripts against databases of multiple partitions (dev/test/production)

- Example: remote server is at IP 1.2.3.4 and has database named SANJOSE
- We will refer to it on our box with alias “DEVBOX”
- Use WRKRDBDIRE (Work with Relational Database Directory Entries) to create local alias to remote database

WRKRDBDIRE to set up alias



Now we can access “DEVBOX” database

- **User and password must be correct for the remote database**

```
$db = db2_connect('DEVBOX', 'DEVUSER', 'DEVPWD');
```

Fourth parameter (array $options)

- **Optional array to fine-tune the connection**
- **Below are choices that are most relevant for IBM i**
- **Details as we go**
  - i5_lib
    - Set a single default library
  - i5_naming
    - Choose “system” or SQL naming
  - i5_libl
    - Set a library **list** (be sure to set i5_naming on)
  - i5_commit
    - Commitment control options
  - autocommit
    - DB2_AUTOCOMMIT_ON (default) or _OFF

i5_lib

- **Specify one library as default**
  - 'i5_lib'=>'MYLIB'
- **Any unqualified files/tables will be assumed to use this library**

i5_naming for library lists

- **DB2_I5_NAMING_ON**
  - A constant equal to 1 that turns on “system naming” mode
  - Table files qualified using the slash (/) delimiter
  - Unqualified files are resolved using the library list for the job

- **DB2_I5_NAMING_OFF**
  - A constant equal to 0 (default) that enables “SQL naming” mode
  - Table files qualified using the period (.) delimiter
  - Unqualified files are resolved using either the default library (i5_lib) or the user profile name specified on db2_connect() (could be QTMHHTTP)
    - Message to watch for:
      MYTABLE in YOURNAME type *FILE not found.
      SQLCODE=-204”

User profile strategy

Three techniques (1 of 3)

- Small number of "system" user profiles. One user per library list. When "real" user signs in, authenticate against a database, LDAP, etc. Not against user profile list

Three techniques (2 of 3)

- db2_connect with real user profile
  - user can get disabled when wrong password is entered
  - revealing an actual user profile to end users (OK for internal. security problem if external)
  - benefit: authority, program logic, journal entries will work with "real" IBM i user profile
- db2_pconnect works the same except that the jobs stay semi-active, visible to operator

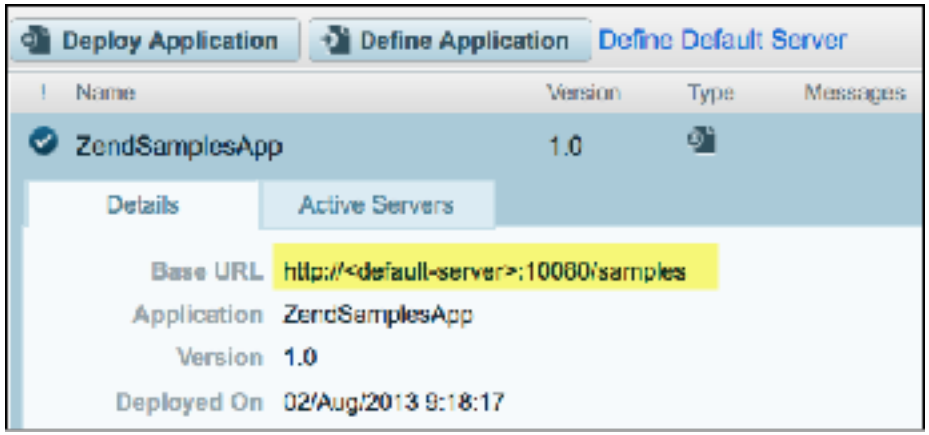
Three techniques (3 of 3)

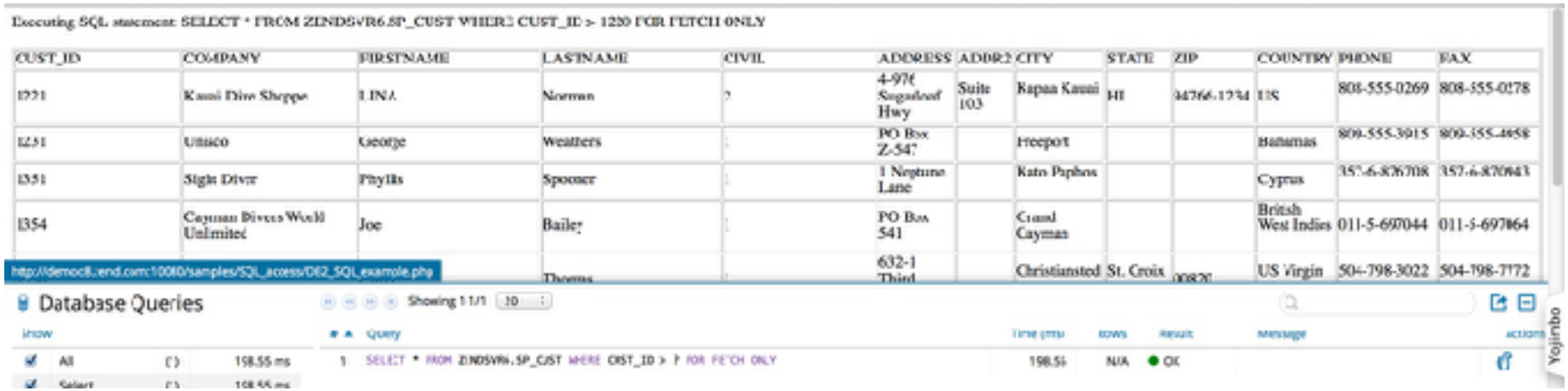
- db2_pconnect with generic "system" profile but then switch to "real" profile after authentication
  - fast connection
  - limits number of active jobs
  - performance cost of the "switch"
  - security concerns about mixed generic/specific user profile attributes/authority in a job
  - need to "switch back" afterward

Included sample script

Included with Zend Server

"SQL Access" sample script illustrates several techniques from this talk





Snippet of sample script

```
/* Construct the SQL statement */
$sql = "SELECT * FROM ZENDSVR.SP_CUST WHERE CUST_ID > ? FOR FETCH ONLY";

/* Prepare, bind and execute the DB2 SQL statement */
$stmt= db2_prepare($conn_resource, $sql);
$lower_limit = 1220; //from the CUST_ID value
$flds = db2_num_fields($stmt);
if($flds > 0 )
{
//show Table Header (analyze result set)
      echo "<table border=1>";
      echo "<tr>";
      for($i=0; $i<$flds; $i++)
      {
            echo '<td width="20%">';
            $name = db2_field_name($stmt, $i);
            echo $name;
            echo "</td>";
      }

      echo "</tr>";

//Execute statement , uses a binding of parameters
db2_bind_param($stmt, 1, "lower_limit", DB2_PARAM_IN);
$result = db2_execute($stmt);
```

Commitment control

Assure data integrity

- Commitment control allows “all or none” logic when running multiple update/insert/delete queries
- If one query fails, roll back previous related queries
- Example: If a detail record update fails, roll back the header update
- Requires that journaling be enabled on files/tables being written to

Use commitment control for data integrity

- **Example: “all or none” for two INSERTS**
  - with ibm_db2.i5_allow_commit = 1
  - and autocommit = DB2_AUTOCOMMIT_OFF

```
$conn = db2_pconnect('', '', '', array('autocommit'=>DB2_AUTOCOMMIT_OFF));
$stmt=db2_prepare($conn,"INSERT INTO MYTABLE (IDNUM, NAME) VALUES(?, ?)");
$result1 = db2_execute($stmt, array(1, 'jane')); // should insert OK
$result2 = db2_execute($stmt, array('x', 'bob')); // not numeric!

// check if both INSERTs succeeded
if ($result1 && $result2) {
  // Success. Commit both inserts
  db2_commit($conn);
} else {
  // *** Error with one of the inserts; roll them both back ***
  db2_rollback($conn);
}
// Neither record will be in the table. We rolled back.
```

i5_commit

- **i5_commit**
  - Options for isolation level
  - Gives you fine-grained control (and ability to turn off altogether)
  - Before Zend Server 8.5, must also enable commitment control system-wide
    - ibm_db2.i5_allow_commit = 1 in INI file
  - Choices:
    - *DB2_I5_TXN_NO_COMMIT* – turns off commitment control for this connection
    - *DB2_I5_TXN_READ_UNCOMMITTED*
    - *DB2_I5_TXN_READ_COMMITTED*
    - *DB2_I5_TXN_REPEATABLE_READ*
    - *DB2_I5_TXN_SERIALIZABLE*

autocommit

- **DB2_AUTOCOMMIT_ON**
  - A constant equal to 1 (default)
  - Turns autocommit on
    - End of script causes commit
  - Only relevant when commitment control is used
  - Convenient: insert/update/delete will work without db2_commit()
- **DB2_AUTOCOMMIT_OFF**
  - A constant equal to 0
  - Turns autocommit off
  - Only relevant when commitment control is used
  - Provides flexibility to ensure data integrity in multi-step transactions by using db2_commit()/db2_rollback() around groups of insert/update/delete queries

Commitment control tips

- **In php.ini:**
  - To use commitment control before 1.9.6-sg25, set ibm_db2.i5_allow_commit = 1
- **In db2_connect() option array:**
  - Modify default settings with ‘i5_commit’ option
  - Choose ‘autocommit’ on or off
- **Turn on journaling for schemas (libraries)**
  - Already on if schema created via “CREATE SCHEMA” (SQL/DDL)
  - Extra step needed for libraries created via CRTLIB
    - Start Journal Library (STRJRNLIB, v6.1+) makes a library a journaled object. Any objects eligible to be journaled that are added to the library can be automatically journaled
    - http://www.redbooks.ibm.com/abstracts/tips0662.html

More about db2_connect, db2_pconnect

- **Manual pages**
  - http://www.php.net/manual/en/function.db2-connect.php
  - http://www.php.net/manual/en/function.db2-pconnect.php
  - http://www.php.net/manual/en/features.persistent-connections.php

Security

Prepare queries

“Prepared” = safe and fast

- **Prepared queries help in several ways**
  - Eliminate errors due to un-escaped single quotes
  - Protect your data from SQL Injection attacks
  - Speed up repeated queries
- **They are also known as prepared statements**
- **Here’s an example of the mischief they prevent**

Apostrophes confuse query parsers

```
// mysite.com?name=whatever
$name = $_GET['name'];
$sql = "select custno from custfile
  where name = '$name' and status = 'ACTIVE' ";
```

- **Do you see any potential problems?**

- **What if the name is “O’Shea” ? Error!**

```
$sql = "select custno from custfile
  where name = 'O'Shea' and status = 'ACTIVE' ";
```

- **Single quotes confuse query parser when they serve two purposes**
  - Used as apostrophe in data
  - Delimiter of string literals in the SQL syntax

Malicious users can try “SQL Injection”

```
// mysite.com?name=whatever
$name = $_GET['name'];
$sql = "select custno from custfile
  where name = '$name' and status = 'ACTIVE' ";
```

- **What if the name is the weird-looking “`x' OR 1=1--`”**

(That is, a user typed: mysite.com?name=x' OR 1=1--)

```
$sql = "select custno from custfile
where name = 'x' OR 1=1--' and status = 'ACTIVE' ";
```

- **Every record in the table will be selected**
  - OR 1=1 will always be true
  - -- turns subsequent ‘where’ criteria into a comment (ignored!)

Safeguard data with prepared queries

```
// mysite.com?name=whatever
$name = $_GET['name'];
$sql = "select custno from custfile
 where name = ? and status = 'ACTIVE' ";
```

- **Represent parameters with question marks (?) instead of literal values**
- **It's fine to retain hard-coded values in the query**
  - Such as 'ACTIVE' in the example above
- **Supply parameters in an array**
  - $params = array("O'Shea");
- **Full example on next slide**

db2_prepare() with db2_execute()

```
$name = $_GET['name'];
$conn = db2_connect('','','');
$sql = "select custno from custfile
  where name = ? and status = 'ACTIVE' ";
$params = array($name); // can be "O'Shea" for all we care
$stmt = db2_prepare($conn, $sql);
if ($stmt) { // prepared OK
    $result = db2_execute($stmt, $params);
    if ($result) { // ran query OK with parameters
        while ($row = db2_fetch_assoc($stmt)) {
            echo "$row['custno']\n";
        }
    }
}
```

Ordinary db2_exec() re-calcs plan

- **Ex. of non-prepared SQL repeated with different params**

```
$values = array('acme', 'shoprite', 'stop n shop');
foreach ($values as $value) {
    $sql = "select custno from custfile
            where name = '$value' and status = 'ACTIVE' ";
    // query gets re-optimized in each iteration
    $stmt = db2_exec($conn, $sql);
    if ($stmt) { // do something with $stmt }
}
```

- **The query plan will re-optimize on each db2_exec() because a new SQL string was supplied each time**
- **OK for one-off queries but not when repeated**

Prepared statement allows re-use of plan

- **Ex. of prepared SQL; execution with different params**

```
// prepare the query ONCE
$sql = "select custno from custfile
            where name = ? and status = 'ACTIVE' ";
$stmt = db2_prepare($conn, $sql);
// now execute with values only
$values = array('acme', 'shoprite', 'stop n shop');
foreach ($values as $value) {
    $result = db2_execute($stmt, array($value));
    if $result {// do something with $stmt }
}
```

- **The query plan is calculated ONCE and reused with each db2_execute(), saving time and CPU**

Prepared statements/queries are best

- **Replace db2_exec() with db2_prepare() and db2_execute()**
- **Benefits**
  - Queries will run as intended, with fewer surprises
  - Protection from a common form of hacking (SQL injection)
  - Performance will improve for repeated queries

RCAC

Db2 security technique

- **Row and Column Access Control (RCAC)**
  - In IBM i 7.2+
  - Implemented via additional SQL “WHERE” clauses and more
- **Row control: limit what rows can be selected, at the database level, depending on the user or any other criteria**
- **Column Access: mask or manipulate columns at the database level**
- **Will restrict in all applications**
- **More info**
  - http://www.ibmsystemsmag.com/Blogs/i-Can/September-2014/IBM-i-7-2---Protect-Data-With-RCAC/

Debug/diagnostics

Functions to get error codes/messages

- **db2_conn_error()** connection error code
- **db2_conn_errormsg()** connection error text
- **db2_stmt_error()** prepare/execute error code
- **db2_stmt_errormsg()** prepare/execute error text

Connections return a resource or false

```
$conn = db2_connect("*LOCAL", "MYUSER", "BADPASS");

    // test for false
    if (!$conn) {
        echo "Connection failed. SQL Err: ";
        echo db2_conn_error() . "<br>";
        echo db2_conn_errormsg();

        die();

    } else {
        // use the connection....
    }
```

An incorrect password will generate this output:

```
Connection failed. SQL Err: 08001
Authorization failure on distributed database
connection attempt. SQLCODE=-30082
```

Prepare/execute: resource or false

```
$sql = "SELECT * FROM BADLIB.SP_CUST WHERE CUST_ID > ?";

$stmt= db2_prepare($conn, $sql);

if (!$stmt) {
     echo 'The db2 prepare failed. ';
     echo 'SQLSTATE value: ' . db2_stmt_error() . '<BR>';
     echo ' Message: ' .   db2_stmt_errormsg();
}
```

The error code and message might resemble:

```
SQLSTATE value: 42704
Message: SP_CUST in BADLIB type *FILE not found.
SQLCODE=-204
```

Other trace/debug ideas

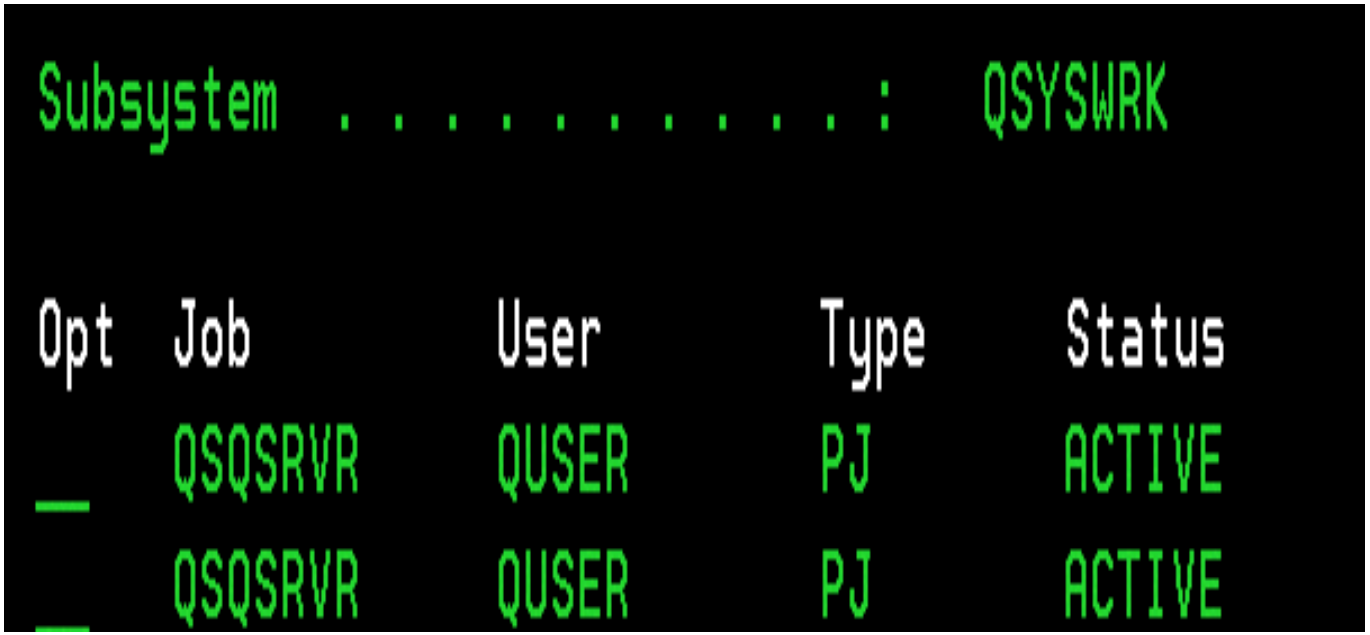
Advanced use only: special libdb400 tracing driver:

http://yips.idevcloud.com/wiki/index.php/PASE/Service

Configuration

Prestart jobs in QSYSWRK by default

DB2 queries run in separate prestart jobs



Prestart job optimization

- **QSQSRVR prestart jobs run in QSYSWRK**
- **Or, if remote DRDA, QRWTSRVR in QUSRWRK**
- **Configurable pool of jobs**

```
CHGPJE SBSD(QSYS/QSYSWRK) PGM(QSYS/QSQSRVR)
  STRJOBS(*YES) INLJOBS(xx) THRESHOLD(xx)
  ADLJOBS(xx) MAXUSE(xx or *NOMAX)
```

- **More on prestart db2 jobs and "server mode"**
  - http://www.redbooks.ibm.com/abstracts/tips0658.html
  - http://www.mcpressonline.com/tips-techniques/database/techtip-grab-control-of-the-db2-qsqsrvr-jobs.html
  - http://www.mcpressonline.com/database/db2/finding-sql-server-mode-connecting-jobs.html

Tip on shared read locks

$stmt = ‘’ releases resources

- **End-of-script normally releases resources/memory**
- **For longer scripts, consider releasing these sooner**
  - Set statement to ‘’ (empty string)
    - Releases memory, resources; closes cursor
    - Old way: db2_free_stmt(), now deprecated
  - Ensure that you don’t need the $stmt anymore

```
$stmt=db2_exec($conn,"SELECT * FROM MYTABLE");
while($row=db2_fetch_array($stmt)) {
  echo "\n<br>";
  var_dump($row);
}
// free statement resources
$stmt = '';
```

db2_free_result() clears “pseudo locks”

- **In persistent mode, SELECT statements can cause *SHRRD (shared read) pseudo locks**
  - See them in QSQSRVR jobs via the WRKOBJLCK command
    - Pseudo locks help retain performance-enhancing cursors
    - Normally, CLRPFM and other exclusive ops will clear the locks
- **If exclusive operations on your ‘i’ occur while your script is active, use db2_free_result() to release cursors**

```
$stmt=db2_exec($conn,"SELECT * FROM MYTABLE");
while($row=db2_fetch_array($stmt)) {
  echo "\n<br>";
  var_dump($row);
}
db2_free_result ($stmt); // allow exclusive ops
```

Another way to clear “pseudo locks”

- **If persistent connections create “shared read locks”**

  Run this command in your CL job stream before nightly exclusive lock attempts.

```
ALCOBJ OBJ((MYLIB/MYFILE *FILE *EXCL *N))
CONFLICT(*RQSRLS)
```

  if a member:

```
ALCOBJ OBJ((MYLIB/MYFILE *FILE *EXCL MYMEMBER))
CONFLICT(*RQSRLS)
```

  Make sure you add a MONMSG immediately after the ALCOBJ command to handle any messages such as "cannot allocate..." which may arise normally.

Library list jubilee (3 ways to set)

1. Most efficient way to set library list

- **Let IBM i set library list based on user profile**
  - No extra program calls required
- **Use a user profile that has an "initial library list" in its job description (JOBD)**
- **Specify the user profile and i5_naming=ON**
  - `$conn = db2_connect('*LOCAL', 'LIBLUSER', 'PASS', array('i5_naming' => DB2_I5_NAMING_ON));`

“JOBD” technique in detail

Create jobd:

```
CRTJOBD JOBD(QGPL/APPROD)
        INLLIBL(LIB1 LIB2 LIB3 QGPL)
```

Set jobd in user profile (or create new profile):

```
CHGUSRPRF USRPRF(APPRODUSR)
          JOBD(QGPL/APPROD)
```

- **In PHP, specify the user profile and i5_naming=ON**
  - `$conn = db2_connect('*LOCAL', 'LIBLUSER', 'PASS',`
    `array('i5_naming' => DB2_I5_NAMING_ON));`

2. Middle ground: i5_libl

- $options array accepts 'i5_libl'
  - i5_libl is a space-delimited library list
    - `'i5_libl'=>'MYLIB YOURLIB ANYLIB'`
- Causes ibm_db2 to run CHGLIBL at the middleware level
- Example:

```
$conn = db2_connect('*LOCAL' ,'MYUSER', 'MYPASS',
                    array('i5_naming' => DB2_I5_NAMING_ON,
                          'i5_libl'   => 'MYLIB1 MYLIB2'));
```

- Note: with persistent connections, CHGLIBL only run the first time in, so be sure to distinguish different jobs by user profile

3. Program call (not recommended)

- **If maximum flexibility required:**
  - Run a command or program that sets up library list

```
$db = db2_connect(‘*LOCAL', 'MYUSER', 'MYPASS',
            array('i5_naming' => DB2_I5_NAMING_ON));

$libl = ‘CNXSRTQA QGPL QTEMP’;
$sql = "call qsys2/qcmdexc('CHGLIBL LIBL($libl)')";
$stmt = db2_exec($conn,$sql);
```

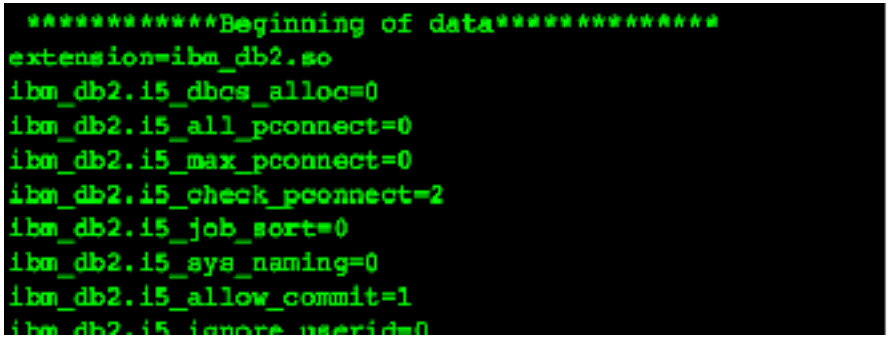
db2_connect() example with $options

```
$database = 'MYDB';
$user = 'MYUSER';
$password = 'MYPASS';

$options = array('i5_naming'  => DB2_I5_NAMING_ON,
                 'i5_libl'    => 'MYLIB1 MYLIB2'
);
$conn = db2_connect($database, $user, $password, $options);

if ($conn) {
    echo "Connection succeeded.";
} else {
    echo "Connection failed.";
}
// MYTABLE will be found, if in library MYLIB1 or MYLIB2
$stmt=db2_exec($conn,"SELECT * FROM MYTABLE");
```

New
Global settings

Config file(s) for ibm_db2

- **ibm_db2.ini**
  - Main location for these settings
  - /usr/local/zendsvr6/etc/conf.d/ibm_db2.ini
  - A small file containing only ibm_db2 settings
  - Initial contents:
    `extension=ibm_db2.so`



- **php.ini**
  - Less common location
  - /usr/local/zendsvr6/etc/php.ini
  - Large file containing hundreds of settings
  - Add or modify settings under the section `[ibm_db2]`

New ibm_db2 config options

- Shipped since Zend Server 7
- See change log for news
  - http://www.youngiprofessionals.com/wiki/index.php/XMLSERVICE/PHPDB2ChangeLog
- Highlights coming right up

Highlights new (slide 1 of 2)

Setting	Default	What it does
ibm_db2.**i5_sys_naming**	**0**,1	1 enables library lists by default (even in LUW DB2 Connect 10.5)
ibm_db2.**i5_blank_userid**	0,**1**	When 0, blank user id/ password become invalid
ibm_db2.**i5_max_pconnect**	**0**-n	Cleans up persistent QSQSRVR jobs every so many connections
ibm_db2.**i5_check_pconnect**	**0**-4	Checks pconnect job. 0 means no test; 1-4 progressively more robust to test for a valid connection. If connection invalid, reconnect fresh

Highlights new (slide 2 of 2)

Setting	Values	What it does
ibm_db2.**i5_log_verbose**	**0**,1	Log DB2 errors in /usr/local/zendsvr(6)/var/logs/php.log **Alan recommends 1**
ibm_db2. **i5_servermode_subsystem**	**[empty]**, *SAME, subsystem (QSYSWRK)	Exposes connection attribute SQL_ATTR_SERVERMODE_SUBSYSTEM to specify subsystem for QSQSRVR jobs
ibm_db2.**i5_guard_profile**	**0**,1	Restore job's original "current user" at end of PHP request (this feature subject to change)

Alan's recommended settings

/usr/local/zendsvr6/etc/conf.d/ibm_db2.ini

```
extension=ibm_db2.so
; log db2 errors in PHP error log
ibm_db2.i5_log_verbose=1

; Reset persistent connection after every 200
connection requests
ibm_db2.i5_max_pconnect=200

; Quick check on each pconnect (conn. alive)
ibm_db2.i5_check_pconnect=1
```

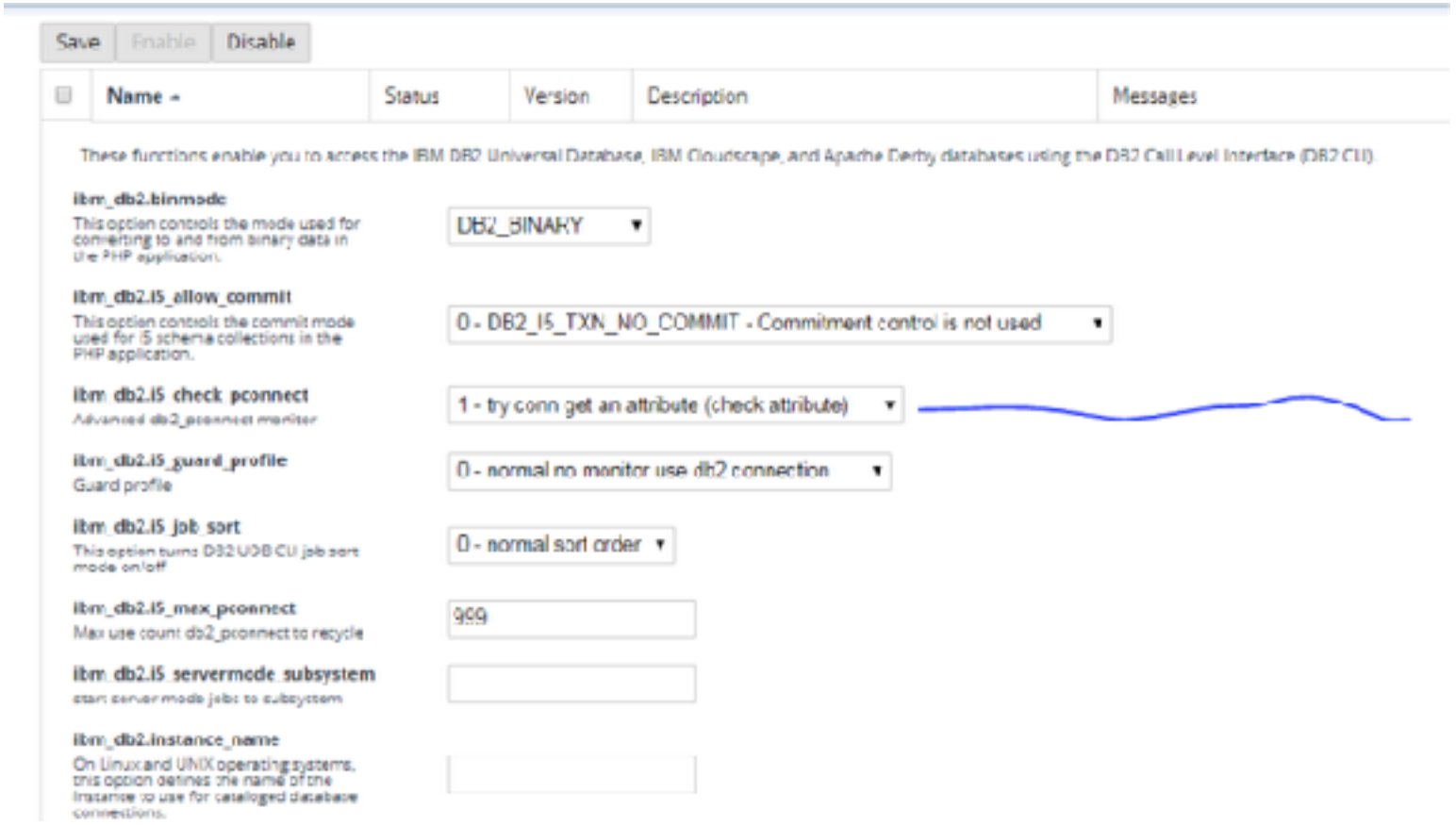
Default values to examine

/usr/local/zendsvr6/etc/conf.d/ibm_db2.ini

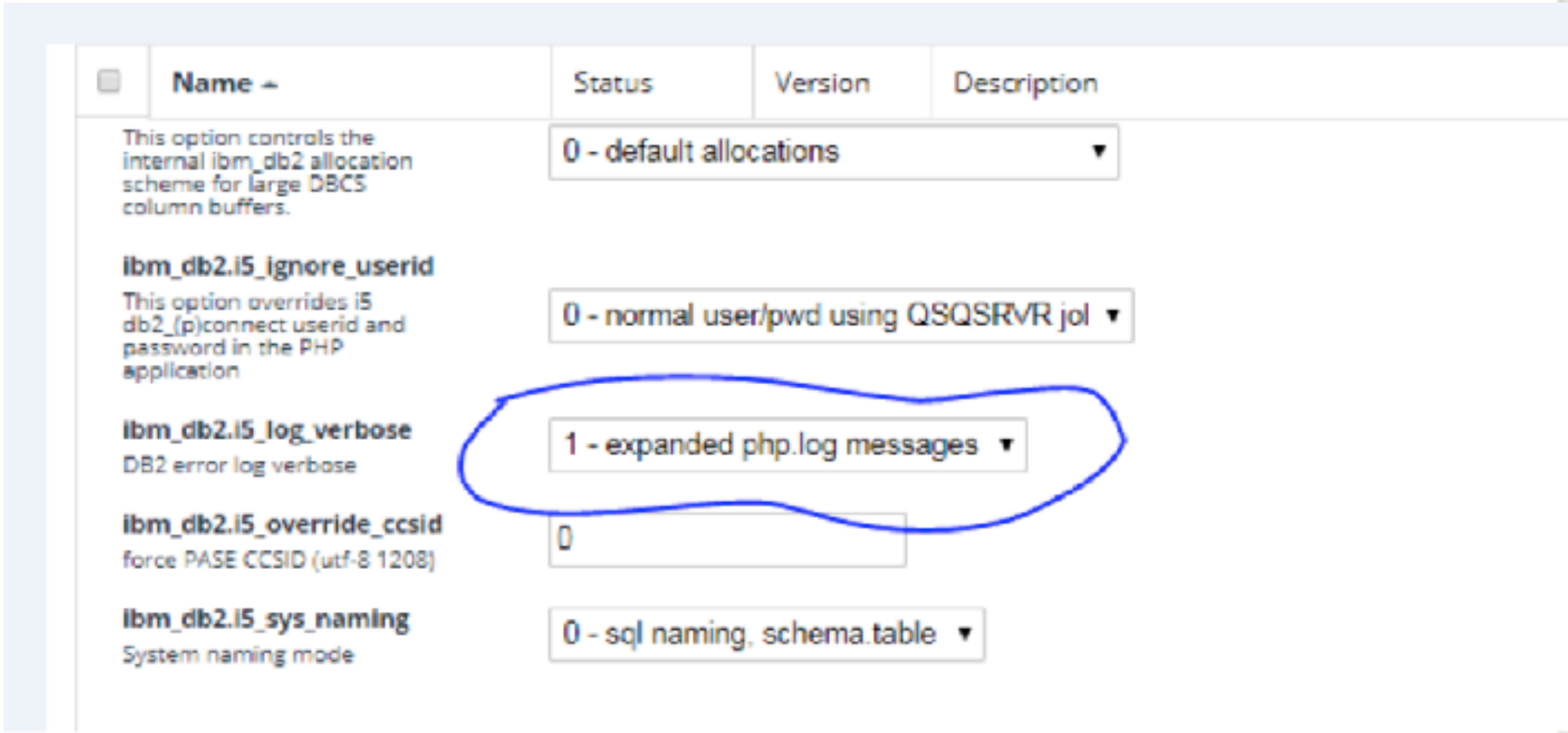
```
; no commitment control
ibm_db2.i5_allow_commit=0

; allow blank user/pw (change to 0 if you can)
ibm_db2.i5_blank_userid=1
```

Note: Do not update these values directly in INI files. Go to PHP section of Zend Server admin web interface to update (see next slides)

i5_check_pconnect in Zend Server admin



i5_log_verbose in Zend Server admin



Set up DB2/i in Zend Framework

IBM i configuration change

In application.config.php, add __DIR__

```
 'config_glob_paths' => array(
                __DIR__ . '/autoload/{,*.}
{global,local}.php',
 ),
```

This is because IBM i normally starts searching at the IFS root. Must tell it to start at __DIR__.

Set up IBM i DB2 connection

In local.php (or combination global and local)

```
return array(
    'db' => array(
        'driver' => 'IbmDb2',
        'database' => '*LOCAL',
        'username' => 'MYUSER',
        'password' => 'MYPASS',
        'persistent' => true, // true or false. Update your ZF2 for this
        'driver_options' => array(
            'autocommit' => 'DB2_AUTOCOMMIT_OFF',
            'i5_lib' => 'MYLIB', // for example. Could also specify i5_libl
        ),
    ),
    'service_manager' => array(
        'factories' => array(
            'db-adapter' => 'Zend\Db\Adapter\AdapterServiceFactory',
        ),
    ),
);
```

Local development (on your PC)

“How do I develop on non-i, deploy on i?”

- **Developer goal:**
  - Develop IBM i-based PHP code on a non-i machine (Linux, Windows)
  - Connect to DB2 on IBM i using code that will also work on ‘i’ later in production
- **Requires a separate product from IBM**

IBM DB2 Connect from Windows/Linux

- **Purchase IBM's "DB2 Connect"**
  - https://www.ibm.com/ms-en/marketplace/db2-connect
  - Not free. We'll see how this develops (no more Personal Edition)
- **Your non-i computer will host a local "dummy" DB2 database that actually accesses DB2 on your IBM i**
  - Use the same ibm_db2 functions that you normally do
  - Configuration tips: http://yips.idevcloud.com/wiki/index.php/Tier2/DB2Connect
- **Library lists (system naming) available in DB2 Connect 10.5 on IBM i 7.1+**

ODBC alternative to DB2 Connect

- ODBC: no charge
- Not optimized for IBM i, minimally maintained/ supported
- Some have had success with it
- I don't recommend for mission-critical systems but you are welcome to try it, especially if your software already supports ODBC
- Less portable than DB2 Connect in moving app to/ from IBM i

Future options

- **db2sock project**

Pagination (LIMIT/OFFSET)

Email from a PHP fan

"I've run into a little problem with DB2. I'm used to being able to use the **LIMIT option in MySQL** to set a **range of records** I want to view, perhaps the 20-29 records for instance.

"Could you tell me if there is or is not a way to do that on IBM i? My boss and I are beginning to think that such an option does not exist for use on the AS400 IBM i."

- Yes, it can be done via a choice of two techniques
- Useful for pagination (page-at-a-time logic)
- LIMIT and OFFSET are non-standard, not in DB2

1. ibm_db2 middleware technique

```
// specify DB2_SCROLLABLE in db2_exec or db2_execute
$startingRow = 20;
$endingRow = 29;
$stmt = db2_execute($stmt,
                    array('cursor' => DB2_SCROLLABLE));
$currentRow = $startingRow;
while (($currentRow <= $endingRow) &&
           $row = db2_fetch_array($stmt, $currentRow)) {
    print "$row[0]\n";
    $currentRow++;
}
```

Note additional param for row number:

array db2_fetch_array (resource $stmt [, int $row_number = -1])

Disadvantage: middleware-dependent (only works with ibm_db2)

2. Old SQL technique

- DB2's row_number() and over() functions can select records by number in a recordset
- This simulates LIMIT and OFFSET

- Example coming up

Record range selection in DB2

Start with this:

```
$queryString = "SELECT CUST_ID, COMPANY FROM SP_CUST
order by CUST_ID”
```

Use row_number() and over() to limit the record selection:

```
$queryString = "SELECT CUST_ID, COMPANY FROM
  (select row_number() over (order by CUST_ID) as rowid,
CUST_ID, COMPANY
   from SP_CUST) as t
   where t.rowid between 20 and 29";
```

Or let Zend Framework handle it

- **Use ZF's Zend Db component from regular PHP**
- **Builds SQL, hiding complexity of LIMIT/OFFSET implementation, with cross-database code**
- **Full script:** https://github.com/alanseiden/Code-Examples/blob/master/IBMi/DB2/ZF2LimitOffset.php

```
// works with v2.3.2 of ZF2.
$sql = new Sql($adapter);
$select = $sql->select();
$select->from('SP_CUST')
       ->where('CUST_ID > 1220')
       ->order('CUST_ID ASC')
       ->limit(10)
       ->offset(20);
```

Sample DB2 on i pagination script

coded by Zend's Clark Everetts

https://github.com/clarkphp/Code-Examples/tree/master/IBMi/pagination

Showing ALL the THINGZ! Page 1 of 3 (54 total records)

CUST_ID	COMPANY	FIRSTNAME	LASTNAME	ADDRESS	ADDR2	CITY	STATE	ZIP	COUNTRY	PHONE	FAX
1221	Kauai Dive Shoppe	LINA	Norman	4-976 Sugarloaf Hwy	Suite 103	Kapaa Kauai	HI	94766-1234	US	808-555-0259	808-555-0273
1231	Unisco	George	Weathers	PO Box Z-547		Freeport			Bahamas	809-555-3915	809-555-4953
1351	Sight Diver	Phyllis	Spooner	1 Neptune Lane		Kato Paphos			Cyprus	357-6-876708	357-6-870943
1354	Cayman Divers World Unlimited	Joe	Bailey	PO Box 541		Grand Cayman			British West Indies	011-5-697044	011-5-697064
1356	Tom Sawyer Diving Centre	Chris	Thomas	632-1 Third Frydenhoj		Christiansted	St. Croix	00820	US Virgin Islands	504-798-3022	504-798-7772
1380	Blue Jack Aqua Center	Ernest	Barratt	23-738 Paddington Lane	Suite 310	Waipahu	HI	99776	US	401-609-7623	401-609-9403
1384	VIP Divers Club	Russell	Christopher	32 Main St.		Christiansted	St. Croix	02800	US Virgin Islands	809-453-5976	809-453-5932
1510	Ocean Paradise	Paul	Gardner	PO Box 8745		Kailua-Kona	HI	94756	US	808-555-8231	808-555-8459
1513	Fantastique Aquatica	Susan	Wong	Z32 999 #12A-77 A.A.		Bogota			Columbia	057-1-773434	057-1-773421
1551	Marmot Divers Club	Joyce	Marsh	872 Queen St		Kitchener	Ontario	G3N 2E1	Canada	416-698-0339	426-698-0399
1560	The Depth Charge	Sam	Witherspoon	15243 Underwater Fwy.		Marathon	FL	35003	US	800-555-3798	800-555-0353
1563	Blue Sports	Theresa	Kunec	203 12th Ave. Box 746		Giribaldi	OR	91187	US	610-772-6704	610-772-6893
1624	Makai SCUBA Club	Donna	Sisus	PO Box 8534		Kailua-Kona	HI	94756	US	317-649-9098	317-649-6787
1645	Action Club	Michael	Spurling	PO Box 5451-F		Sarasota	FL	32274	US	813-870-0239	813-870-0282
1651	Jamaica SCUBA Centre	Barbara	Harvey	PO Box 68		Negril	Jamaica		West Indies	011-3-697043	011-3-697043
1680	Island Finders	Desmond	Ortega	6133 1/3 Stone Avenue		St Simons Isle	GA	32521	US	713-423-5675	713-423-5675
1984	Adventure Undersea	Gloria	Gonzales	PO Box 744		Belize City			Belize	011-34-09054	011-34-09064
2118	Blue Sports Club	Harry	Bathbone	63365 Nez Perce Street		Largo	FL	34684	US	612 897 0342	612 897 0343
2135	Frank's Divers Supply	Lloyd	Fellows	1455 North 44th St.		Eugene	OR	90427	US	503-555-2778	503-555-2769
2156	Davy Jones' Locker	Tanya	Wagner	246 South 16th Place		Vancouver	BC	K8V 9P1	Canada	803-509-0112	803-509-0553

« First Page « < Prev Page < > Next Page > » Last Page »

3. “Real” LIMIT and OFFSET

- Recent: LIMIT and OFFSET support in TR11 (7.1) and TR3 (7.2)

- https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/OFFSET%20and%20LIMIT
- From Scott Forstie’s wiki page:
  - LIMIT: alternative to FETCH FIRST x ROWS ONLY
  - OFFSET: skip rows in the query result
- Addition to Zend Framework to use real LIMIT and OFFSET:
  - https://github.com/zendframework/zend-db/pull/275

Stored procedure from Scott Forstie

OFFSET and LIMIT for Stateless Pagination

```
CREATE OR REPLACE PROCEDURE
TOYSTORE.FIND_EMPLOYEES
  (IN P_PAGESIZE BIGINT, IN P_OFFSET BIGINT)
    DYNAMIC RESULT SETS 1
    LANGUAGE SQL
BEGIN
  DECLARE V_PREP_STMT1 VARCHAR(4096)  ;
  DECLARE CEMP_RESULT_SET1 CURSOR
     WITH RETURN FOR PREP_STMT1;
  SET V_PREP_STMT1 =
  'SELECT EMPNO, HIREDATE, LASTNAME FROM
     TOYSTORE.EMPLOYEE
     ORDER BY HIREDATE DESC
     LIMIT ? OFFSET ?';
  PREPARE PREP_STMT1 FROM V_PREP_STMT1 ;
  OPEN CEMP_RESULT_SET1 USING P_PAGESIZE,
                              P_OFFSET;
END;

CALL TOYSTORE.FIND_EMPLOYEES(10, 0);
CALL TOYSTORE.FIND_EMPLOYEES(10, 10);
```

Page 1

Page 2

Performance tips

Performance tips

- **Use persistent connections (see next slides)**
- **Optimize SQL**
  - A few specific tips (there are so many ways):
    - Avoid scalar functions (such as UPPER) in WHERE clause
    - OPTIMIZE FOR n ROWS (when know how many rows)
    - Check out IBM Rochester's DB2 SQL performance class
- **Index properly**
  - Refer to Plan Cache and Index Advisor
  - Try Encoded Vector Indexes when appropriate
- **System configuration**
  - More memory, more memory!
  - Consider separate memory pool to keep data together

System specs

- Memory and CPU make a big difference
- So will more disk arms, write cache, SSD

- Old story: earlier in the decade, a friend of mine upgraded his customer from a Power 5 to a Power 6. A process that had taken 30 seconds, requiring a complex programming workaround, now took 0.5 seconds.

- Power 8 machines even better

More about specs

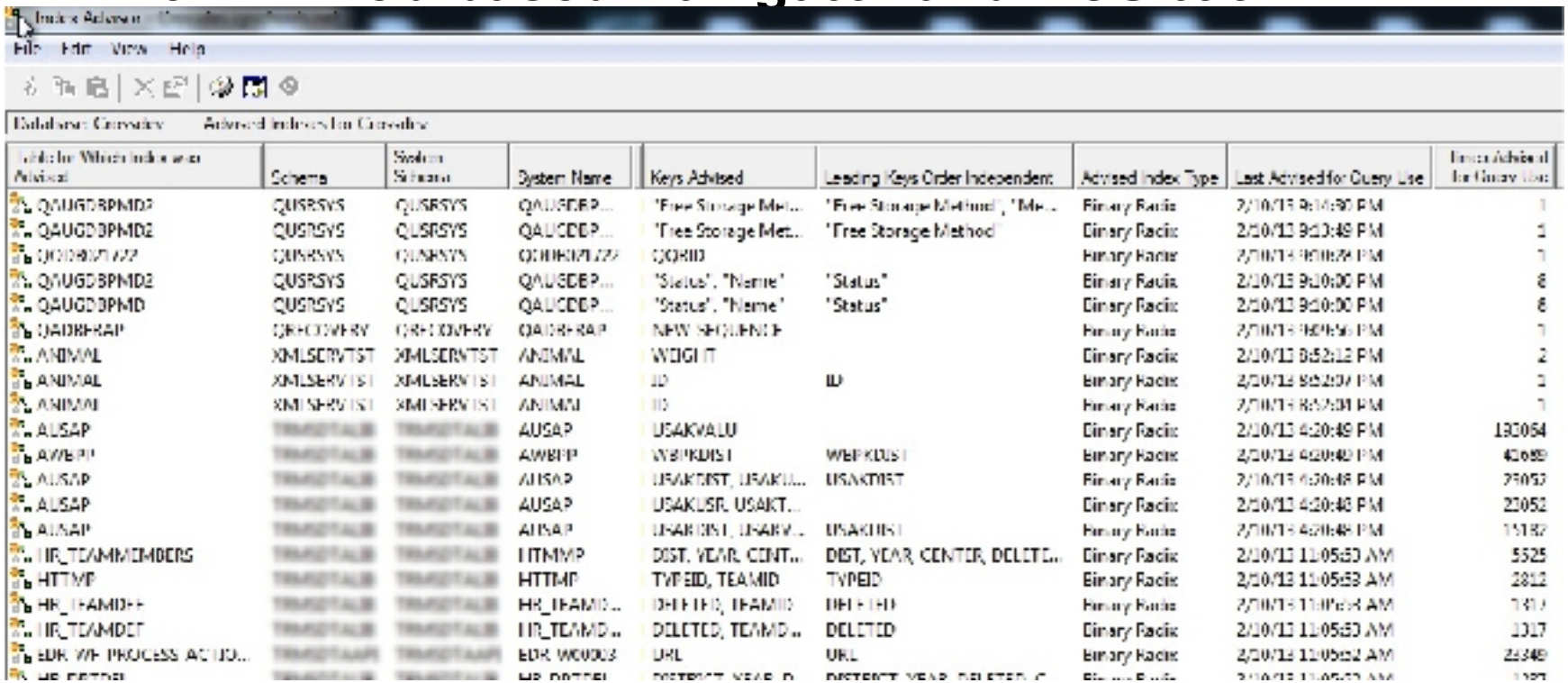
- **OS matters: newer releases speed up complex DB2 queries with a smarter query optimizer**
- **IBM i is built to scale as large as its resources**
  - Example: IBM i's query optimizer will choose the best plan it can, given the amount of memory available to it. More memory, a more intelligent plan.

DB2 query optimization

- **IBM i has great tools**
- **I'll share a couple of favorites**
  - Index Advisor
  - SQL Plan Cache
- **See IBM's book**
  - IBM i Database Performance and Query Optimization
  - https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzatd/rzatdprintable.htm?view=kc

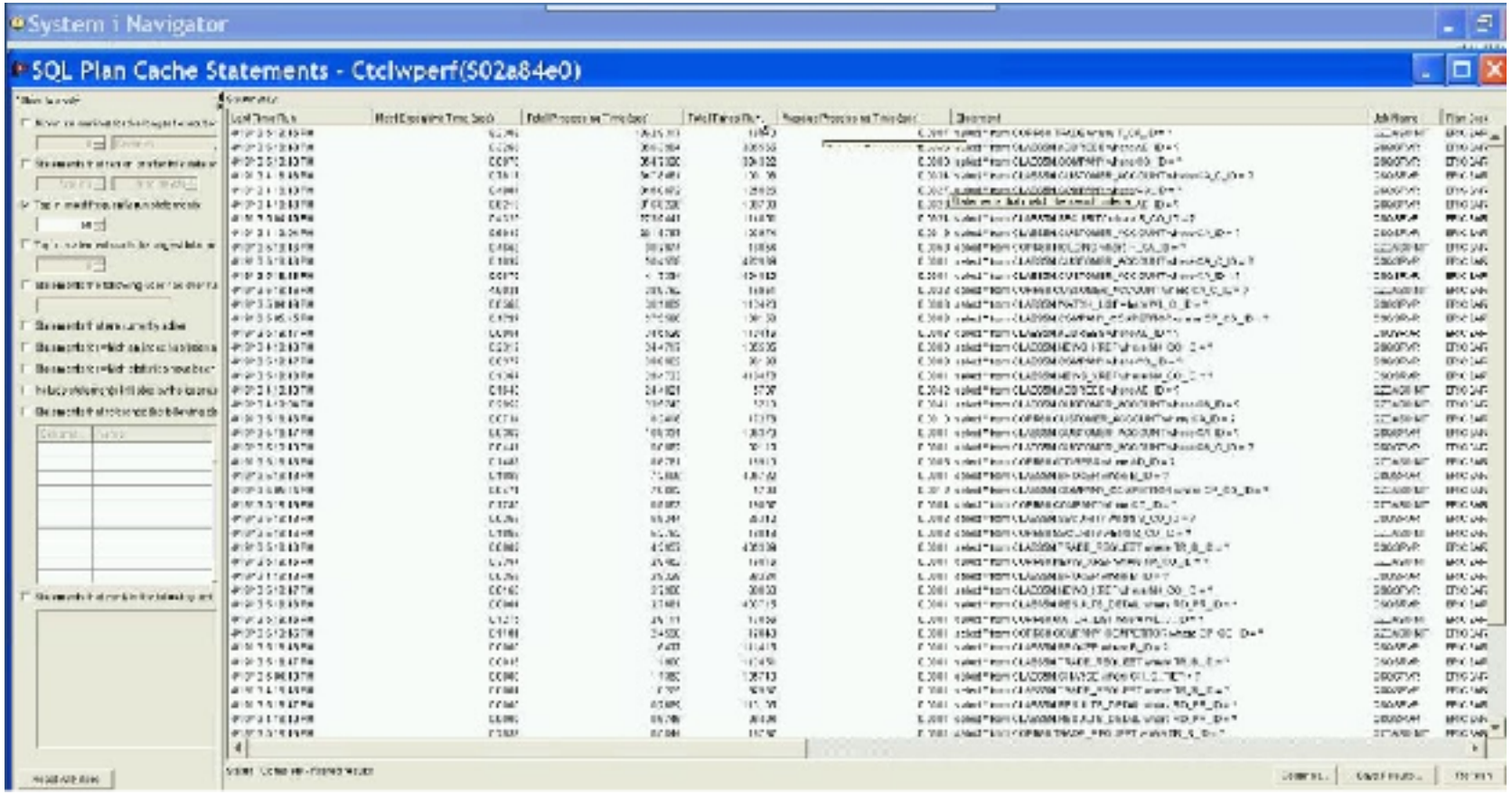
Index Advisor

- Recommends indexes across all queries
- Now in web-based Navigator and ACS too



SQL Plan Cache

- **Lets you see what queries are REALLY running, who's running them, and how long they take**



Persistent connections

- **Create a pool of database jobs**
  - Known as a “connection pool”
  - You will connect quickly because a job is waiting for you
  - DB2 will...
    - Choose a QSQSRVR job when your PHP job first connects
    - Create new jobs to handle high workload

```
Subsystem  . . . . . :   QSYSWRK

Opt  Job         User        Type      Status
__   QSQSRVR     QUSER       PJ        ACTIVE
__   QSQSRVR     QUSER       PJ        ACTIVE
```

- **The word “persistent” may be misleading**
  - No guarantee that a browser session reconnects to same job
  - Between requests, cannot rely on maintaining state (QTEMP, library lists). OK within a request, though

db2_pconnect() to connect persistently

- `resource db2_pconnect ( string $database , string $username , string $password [, array $options ] )`

- **Persistent is much faster than non-persistent**
  - db2_pconnect can reuse connections, reducing the time needed to connect (after the first time) to almost zero
  - SQL statement objects can also be reused, speeding queries

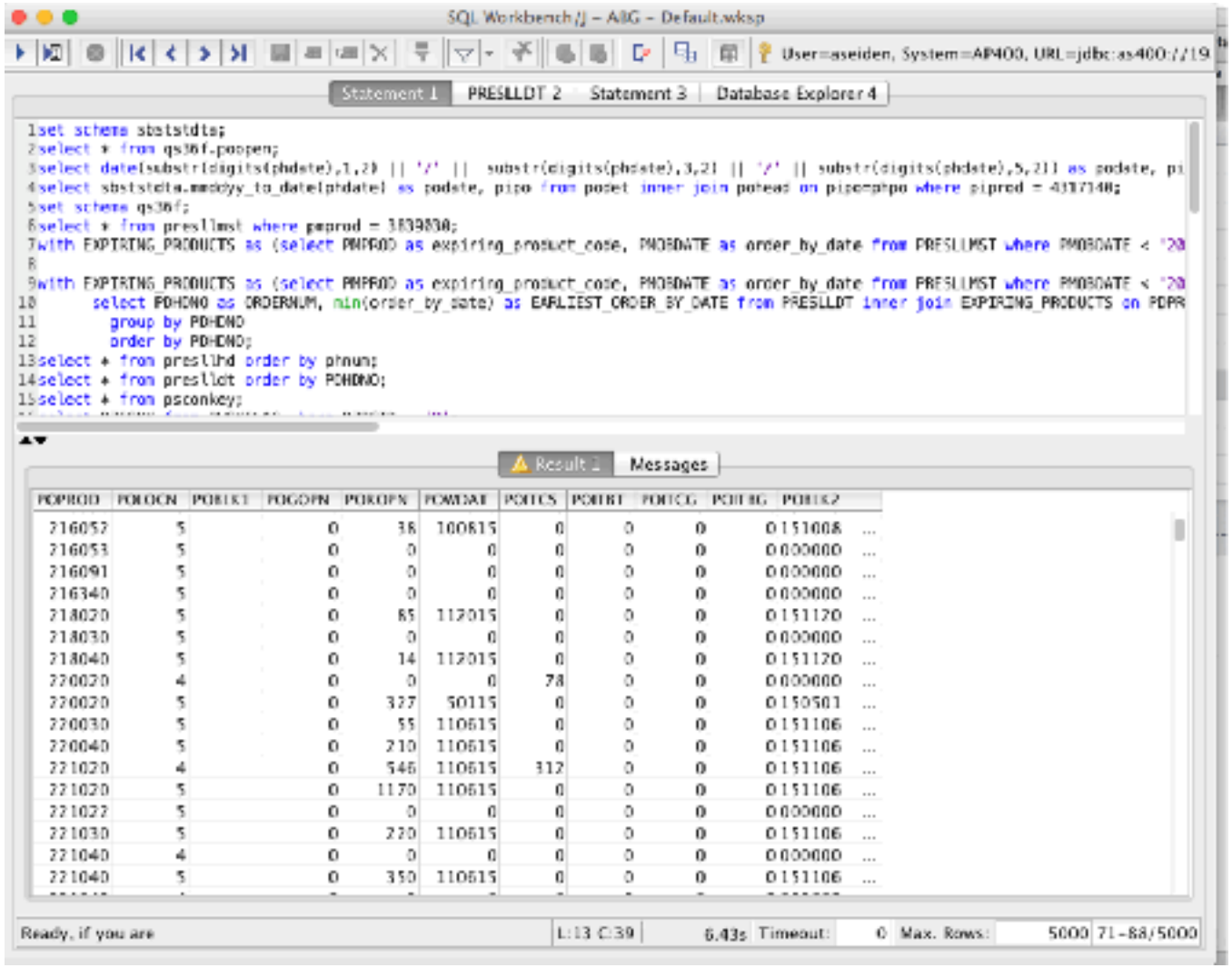
- **How db2_pconnect() reuses connections**
  - Connections defined by *database*, *username*, and *password*
  - Tries to reuse an existing connection matching these 3 params
  - db2_close() on a persistent connection does nothing
  - db2_pclose() forces the conn to close

Tools for testing/ development

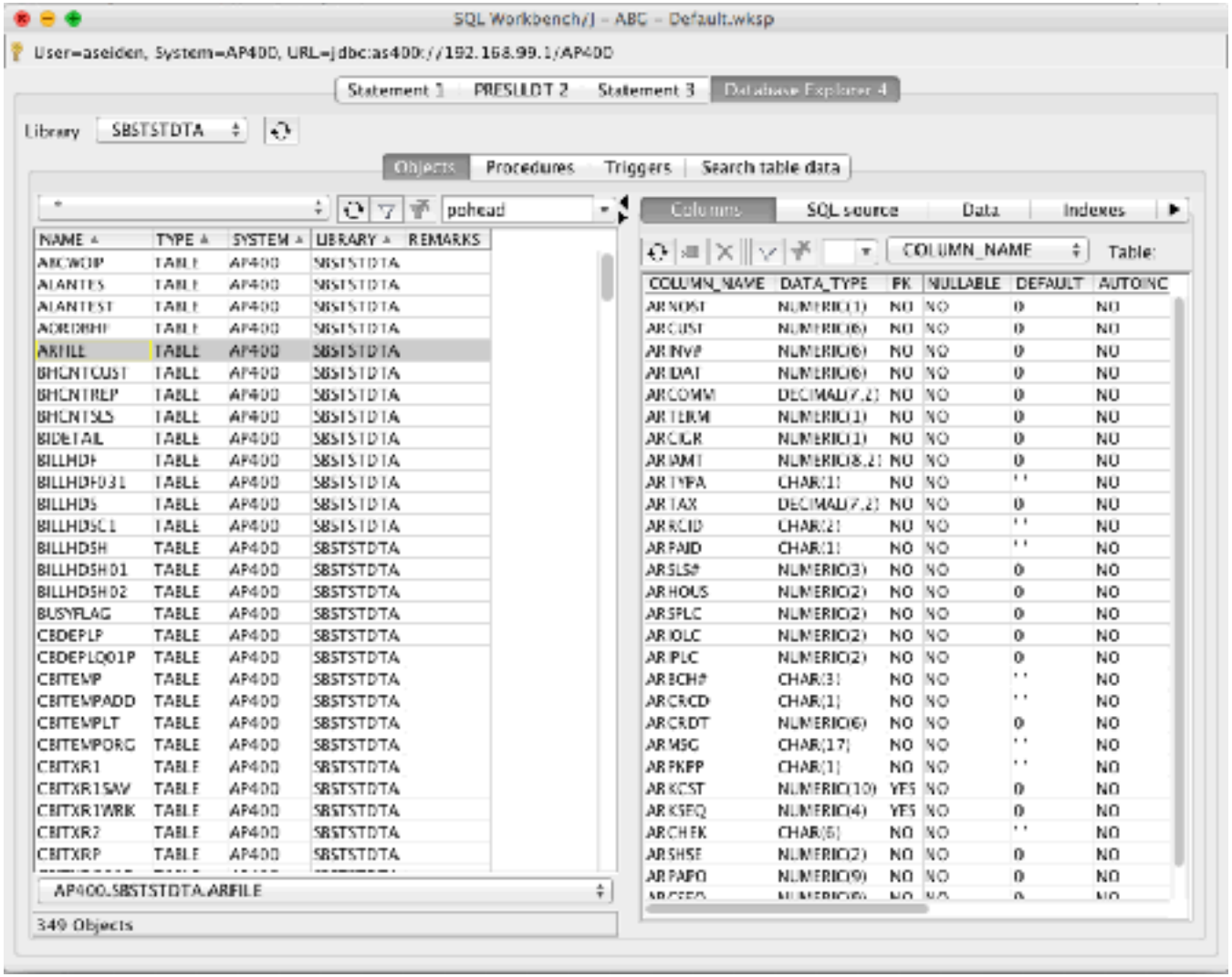
There's more than STRSQL

- **Operations Navigator (old reliable, but Windows only)**
- **Web-based navigator**
  - Starting to get quite good, with Run SQL window just added
  - http://myibmi:2001
- **IBM i Access Client Solutions**
  - New client for Mac, Linux, Windows 10+
  - http://www-03.ibm.com/systems/power/software/i/access/
- **Other tools such as SQL Workbench/J, Data Studio**
  - http://www.sql-workbench.net/, https://www.ibm.com/developerworks/ibmi/library/i-debugger-db2-i/

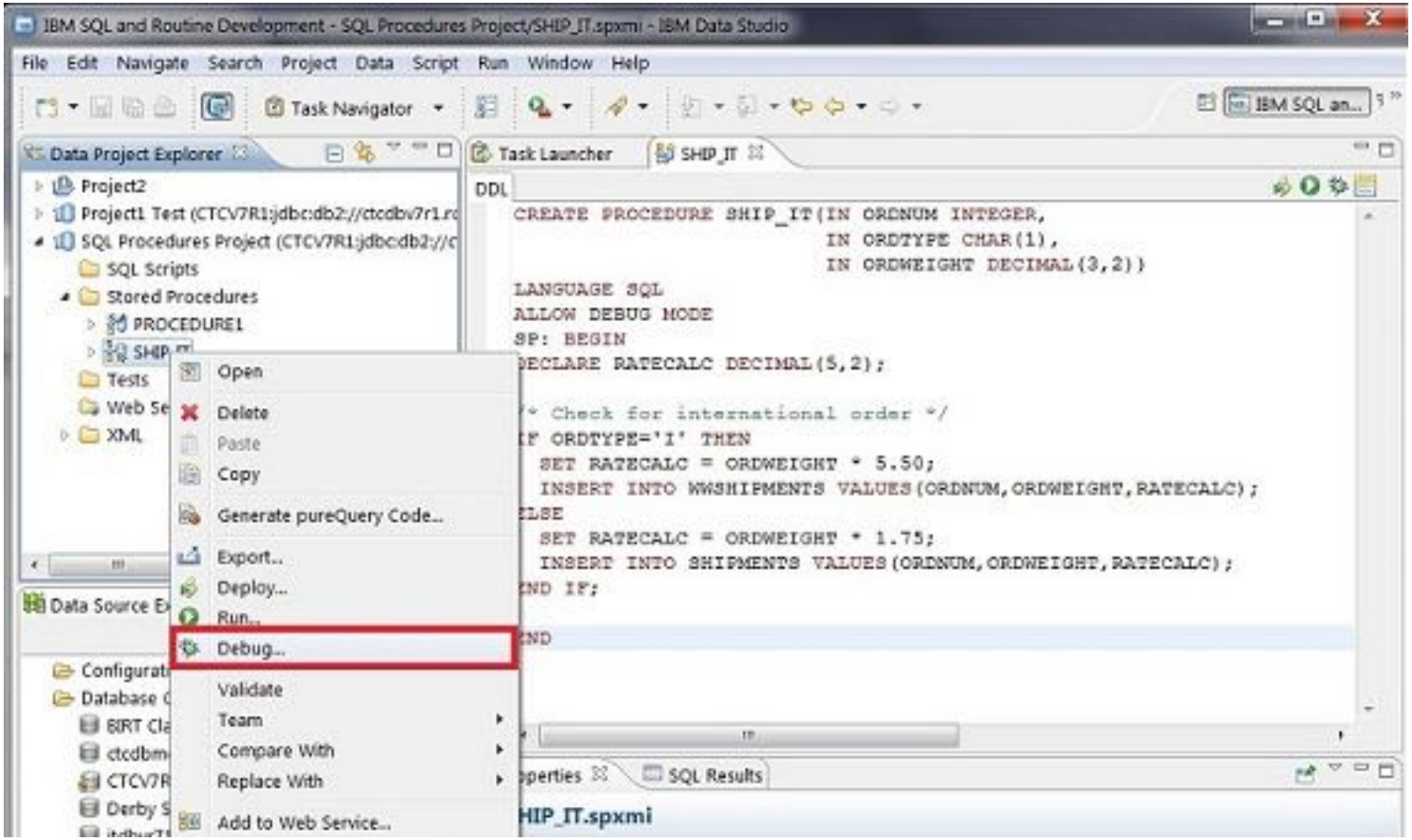
SQL Workbench/J



SQL Workbench/J



Data Studio


IBM SQL and Routine Development - SQL Procedures Project/SHIP_IT.spxmi - IBM Data Studio
File Edit Navigate Search Project Data Script Run Window Help
Task Navigator
Data Project Explorer
Project2
Project1 Test (CTCV7R1:jdbc:db2://ctcdbv7r1.r
SQL Procedures Project (CTCV7R1:jdbc:db2://
SQL Scripts
Stored Procedures
PROCEDURE1
SHIP_IT
Tests
Web Se
XML
Task Launcher
SHIP_IT
DDL
CREATE PROCEDURE SHIP_IT(IN ORDNUM INTEGER,
IN ORDTYPE CHAR(1),
IN ORDWEIGHT DECIMAL(3,2))
LANGUAGE SQL
ALLOW DEBUG MODE
SP: BEGIN
DECLARE RATECALC DECIMAL(5,2);
/* Check for international order */
IF ORDTYPE='I' THEN
SET RATECALC = ORDWEIGHT * 5.50;
INSERT INTO WWSHIPMENTS VALUES(ORDNUM,ORDWEIGHT,RATECALC);
ELSE
SET RATECALC = ORDWEIGHT * 1.75;
INSERT INTO SHIPMENTS VALUES(ORDNUM,ORDWEIGHT,RATECALC);
END IF;
END
Open
Delete
Paste
Copy
Generate pureQuery Code...
Export...
Deploy...
Run...
Debug...
Validate
Team
Compare With
Replace With
Add to Web Service...
Data Source E
Configurat
Database
BIRT Cla
ctcdbm
CTCV7R
Derby S
SQL Results
HIP_IT.spxmi

DB2 and PHP Resources

- **IBM**
  - IBM_DB2 manual and open source repository
    - http://php.net/ibm_db2, http://pecl.php.net/package/ibm_db2
  - DeveloperWorks wiki
    - http://ibm.com/developerworks/ibmi
  - Many details about PHP and DB2 connections
    - http://www.youngiprofessionals.com/wiki/index.php/PHP/DB2Connection
- **Zend**
  - Zend Server for IBM i
    - http://www.zend.com/en/products/server/zend-server-ibm-i
  - Forums for PHP on IBM i
    - http://forums.zend.com/viewforum.php?f=67

Resources

For more info and learning

- **DB2 for i SQL reference**
  - http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/db2/rbafzintro.htm?lang=en
- **Young i PHP/Db2 page**
  - http://www.youngiprofessionals.com/wiki/index.php/PHP/DB2Documents
- **IBM DB2 forum**
  - https://www.ibm.com/developerworks/community/forums/html/forum?id=11111111-0000-0000-0000-000000000292&ps=50

Questions

Contact and tips

Alan Seiden
Seiden Group
Ho-Ho-Kus, NJ

Free newsletter:
http://seidengroup.com/tips

SEIDEN GROUP

alan@seidengroup.com • 201-447-2437 • twitter: @alanseiden

Contact and tips

Alan Seiden

Seiden Group

Ho-Ho-Kus, NJ

Free newsletter:
http://seidengroup.com/tips

SEIDEN GROUP

alan@seidengroup.com • 201-447-2437 • twitter: @alanseiden