

Hidden Gems of IBM i

Scott Forstie – Business Architect, Db2 for i
forstie@us.ibm.com
@Forstie_IBMi



Finding a “Hidden Gem”



- Every release = 100s of new functions
 - Some big
 - Some small
- How to know them all?

Finding a “Hidden Gem”



- Every release = 100s of new functions
 - Some big
 - Some small
- How to know them all?
- All IBM users have favourites
- These are some of our...

“Hidden Gems”



Database – Create or Replace Tables



Create OR REPLACE Table

Data Definition Language (DDL) SQL statements that support the optional 'OR REPLACE' clause:



- ❑ CREATE OR REPLACE ALIAS
- ❑ CREATE OR REPLACE FUNCTION
- ❑ CREATE OR REPLACE MASK
- ❑ CREATE OR REPLACE PERMISSION
- ❑ CREATE OR REPLACE PROCEDURE
- ❑ CREATE OR REPLACE SEQUENCE
- ❑ **CREATE OR REPLACE TABLE**
- ❑ CREATE OR REPLACE TRIGGER
- ❑ CREATE OR REPLACE VARIABLE
- ❑ CREATE OR REPLACE VIEW

Replacing a table:

- ✓ Data-Centric
- ✓ Dependent Views & MQTs preserved
- ✓ Triggers preserved
- ✓ RCAC controls preserved
- ✓ Auditing preserved
- ✓ Authorizations preserved
- ✓ Comments and Labels preserved
- ✓ Rows optionally deleted

Knowledge Center

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/db2/rbafzhctabl.htm?lang=en

Article for previous OR REPLACE statements

<http://iprodeveloper.com/database/use-sql-create-or-replace-improve-db2-i-object-management>

Create OR REPLACE Table

- CREATE OR REPLACE TABLE allows users to manage the master table source.
- The attributes specified on the CREATE OR REPLACE TABLE will be compared to the existing attributes and the corresponding alters are performed.

You Build it

```
ALTER TABLE corpdata.employee  
  
ALTER COLUMN firstnme  
    SET DATA TYPE VARCHAR(20) NOT NULL  
  
ALTER COLUMN lastname  
    SET DATA TYPE VARCHAR(30) NOT NULL  
  
ALTER COLUMN phoneno  
    SET DATA TYPE VARCHAR(13)  
  
ADD COLUMN level INT BEFORE edlevel;
```

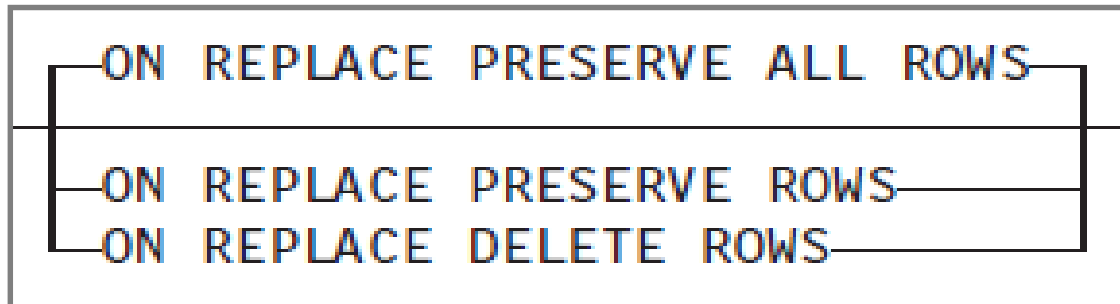
Db2 for i managed

```
CREATE OR REPLACE TABLE corpdata.employee(  
    empno CHAR(6) NOT NULL,  
    firstnme VARCHAR(20) NOT NULL,  
    midinit CHAR(1) NOT NULL,  
    lastname VARCHAR(30) NOT NULL,  
    workdept CHAR(3) DEFAULT NULL,  
    phoneno VARCHAR(13) DEFAULT NULL,  
    hiredate DATE DEFAULT NULL,  
    job CHAR(8) DEFAULT NULL,  
    level INT,  
    edlevel SMALLINT NOT NULL,  
    sex CHAR(1) DEFAULT NULL,  
    birthdate DATE DEFAULT NULL,  
    salary DECIMAL(9, 2) DEFAULT NULL,  
    bonus DECIMAL(9, 2) DEFAULT NULL,  
    comm DECIMAL(9, 2) DEFAULT NULL,  
    PRIMARY KEY( empno ) ) ;
```

Create OR REPLACE Table

Db2 for i implements table replacement using the necessary set of **ALTER** operations. If alter doesn't support the action, neither will create or replace table.

Usage Question: **Do you want to preserve the data?**



PRESERVE ALL ROWS (default)

- Rows are always preserved
- Columns can be dropped or altered

PRESERVE ROWS

- Rows are preserved, unless a range is eliminated from a partitioned table
- If a specified range or partition name matches, the partition is preserved
- Columns can be dropped or altered

DELETE ROWS

- All rows are deleted
- No delete triggers are fired

Create OR REPLACE Table

What about **CREATE TABLE AS** or **CREATE TABLE LIKE**?

```
CREATE OR REPLACE TABLE EMPLOYEE AS  
(SELECT * FROM MASTER_TABLES.EMPLOYEE)  
WITH NO DATA  
INCLUDING IDENTITY COLUMN ATTRIBUTES  
INCLUDING COLUMN DEFAULTS  
INCLUDING IMPLICITLY HIDDEN COLUMN ATTRIBUTES  
INCLUDING ROW CHANGE TIMESTAMP COLUMN ATTRIBUTES  
ON REPLACE PRESERVE ROWS;
```

```
CREATE OR REPLACE TABLE EMPLOYEE LIKE  
MASTER_TABLES.EMPLOYEE  
INCLUDING IDENTITY COLUMN ATTRIBUTES  
INCLUDING COLUMN DEFAULTS  
INCLUDING IMPLICITLY HIDDEN COLUMN ATTRIBUTES  
INCLUDING ROW CHANGE TIMESTAMP COLUMN ATTRIBUTES  
ON REPLACE PRESERVE ROWS;
```

Using **CREATE TABLE AS**

- Copy-options can be used to retain columns and attributes
- Constraints are not included
- Must use WITH NO DATA

Using **CREATE TABLE LIKE**

- Copy-options can be used to retain columns and attributes
- Constraints are not included

Create OR REPLACE Table

How does dependency management work?

```
CREATE OR REPLACE TABLE DEMO_IT (
    FRST CHAR(6) CCSID 37 NOT NULL,
    SCND INTEGER,
    THRD VARCHAR(10)
)

CREATE OR REPLACE VIEW VIEW_IT AS
    SELECT * FROM DEMO_IT
CREATE INDEX INDEX_IT ON DEMO_IT(THRD);

CREATE OR REPLACE TABLE DEMO_IT (
    FIRST_NAME FOR COLUMN FRST CLOB(1K) NOT NULL,
    SECOND_NAME FOR COLUMN SCND BIGINT DEFAULT -1,
    THIRD_NAME FOR COLUMN THRD VARCHAR(1000)
)
```

- Dependent object management:**
- Column names (SQL names), data types and attribute changes are reflected in dependent objects
 - System column names (field names) cannot be changed
 - If Db2 for i cannot gain exclusive access to all the dependent objects, the operation will fail with SQL0913
 - If the change is incompatible, the operation will fail

VIEW_IT field definitions before & after the replacing the table

Field Level Information				
Field	Data Type	Field Length	Buffer Length	
FRST	CHAR	6	6	
	Default value			
	Coded Character Set Identifier			
SCND	BINARY	18	8	
	Allows the null value			
THRD	CHAR	1000	1002	
	Variable length field			

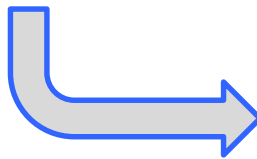
Field Level Information				
Field	Data Type	Field Length	Buffer Length	Buffer Position
FRST	CLOB	1024	32	1
	Allocated Length			
	Default value			
	Coded Character Set Identifier			
SCND	BINARY	18	8	33
	Allows the null value			
THRD	CHAR	1000	1002	41
	Variable length field			

Create OR REPLACE Table

Generating DDL for existing tables will normally produce separate statements for the table and its constraints.
 Use the **GENERATE_SQL()** procedure to produce master table source.

```
CALL qsys2.generate_sql (
  'EMPLOYEE',
  'TOYSTORE_MINNESOTA_1',
  'TABLE',
  CREATE_OR_REPLACE_OPTION => '1',
  CONSTRAINT_OPTION => '2');
```

Constraints



```
CREATE OR REPLACE TABLE TOYSTORE_MINNESOTA_1.EMPLOYEE (
EMPNO CHAR(6) CCSID 37 NOT NULL ,
FIRSTNME VARCHAR(12) CCSID 37 NOT NULL ,
MIDINIT CHAR(1) CCSID 37 NOT NULL ,
LASTNAME VARCHAR(15) CCSID 37 NOT NULL ,
WORKDEPT CHAR(3) CCSID 37 DEFAULT NULL ,
PHONENO CHAR(4) CCSID 37 DEFAULT NULL ,
HIREDATE DATE DEFAULT NULL ,
JOB CHAR(8) CCSID 37 DEFAULT NULL ,
EDLEVEL SMALLINT NOT NULL ,
SEX CHAR(1) CCSID 37 DEFAULT NULL ,
BIRTHDATE DATE DEFAULT NULL ,
SALARY DECIMAL(9, 2) DEFAULT NULL ,
BONUS DECIMAL(9, 2) DEFAULT NULL ,
COMM DECIMAL(9, 2) DEFAULT NULL ,
CONSTRAINT TOYSTORE_MINNESOTA_1.Q_TOYST00001_EMPLOYEE_EMPNO_00001 PRIMARY KEY(
EMPNO ) ,
CONSTRAINT TOYSTORE_MINNESOTA_1.RED
FOREIGN KEY( WORKDEPT )
REFERENCES TOYSTORE_MINNESOTA_1.DEPARTMENT ( DEPTNO )
ON DELETE SET NULL
ON UPDATE NO ACTION ,
CONSTRAINT TOYSTORE_MINNESOTA_1.NUMBER
CHECK( PHONENO >= '0000' AND PHONENO <= '9999' ) )
```

Managing Database Changes in Production



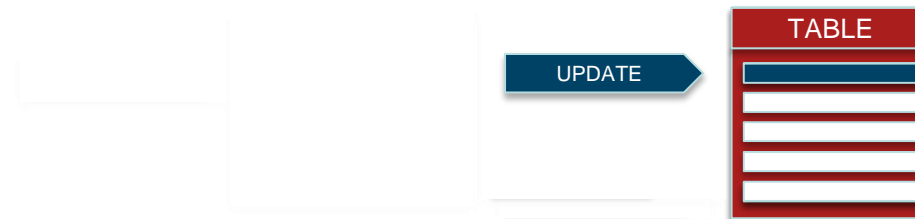
Fair Lock Option

Challenge: Frequent DML activity blocks DDL request

Response: **PREVENT_ADDITIONAL_CONFLICTING_LOCKS** QAQQINI control

Benefit: Improved ability to transform data model in production

Support: Applies to ALTER TABLE (Add, Alter or Drop Column), CREATE TRIGGER, LOCK TABLE, & RENAME TABLE



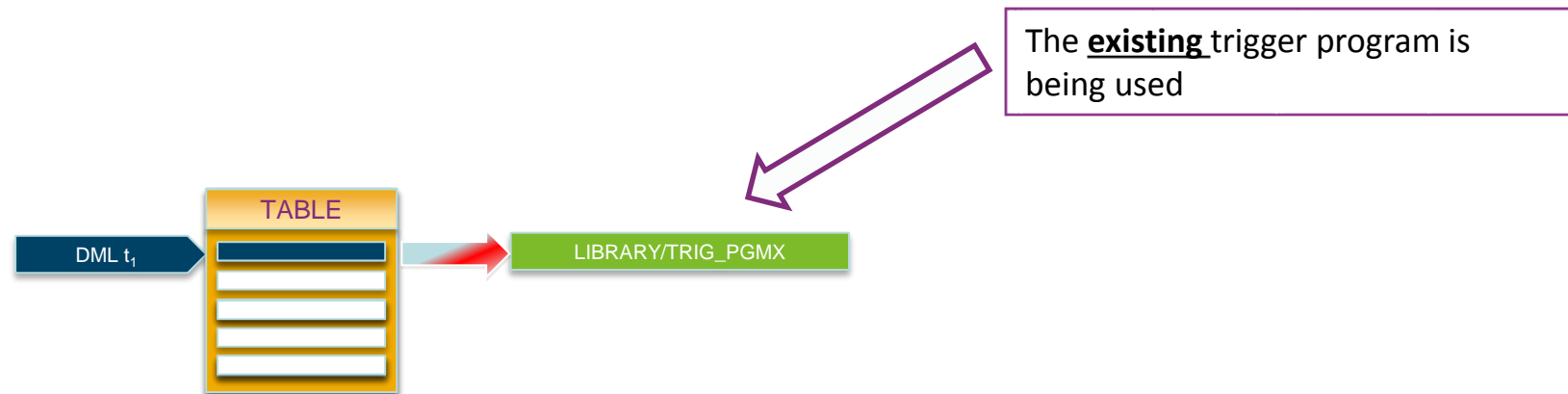
Fair Lock Option

Challenge: Seemingly impossible to make DDL changes in production

Response: **ALLOW_DDL_CHANGES_WHILE_OPEN** QAQQINI control

Benefit: Ability to deploy trigger changes without quiescing user activity

Support: Applies to CREATE TRIGGER, ALTER TRIGGER, DROP TRIGGER, COMMENT ON TRIGGER, and LABEL ON TRIGGER, ADDPFTRG, RMVPFTRG, and CHGPFTRG



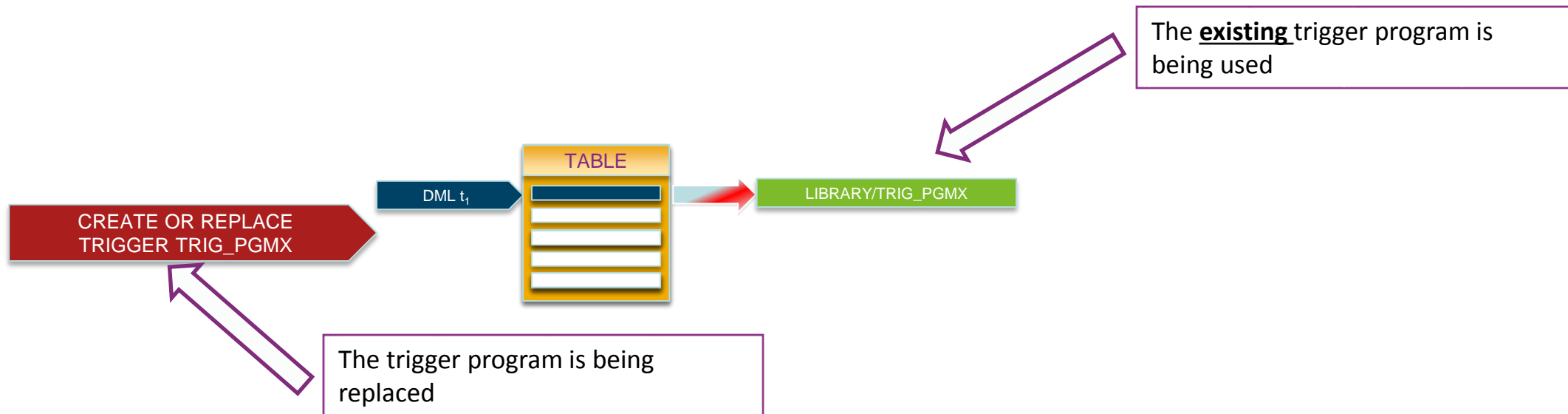
Fair Lock Option

Challenge: Seemingly impossible to make DDL changes in production

Response: **ALLOW_DDL_CHANGES_WHILE_OPEN** QAQQINI control

Benefit: Ability to deploy trigger changes without quiescing user activity

Support: Applies to CREATE TRIGGER, ALTER TRIGGER, DROP TRIGGER, COMMENT ON TRIGGER, and LABEL ON TRIGGER, ADDPFTRG, RMVPFTRG, and CHGPFTRG



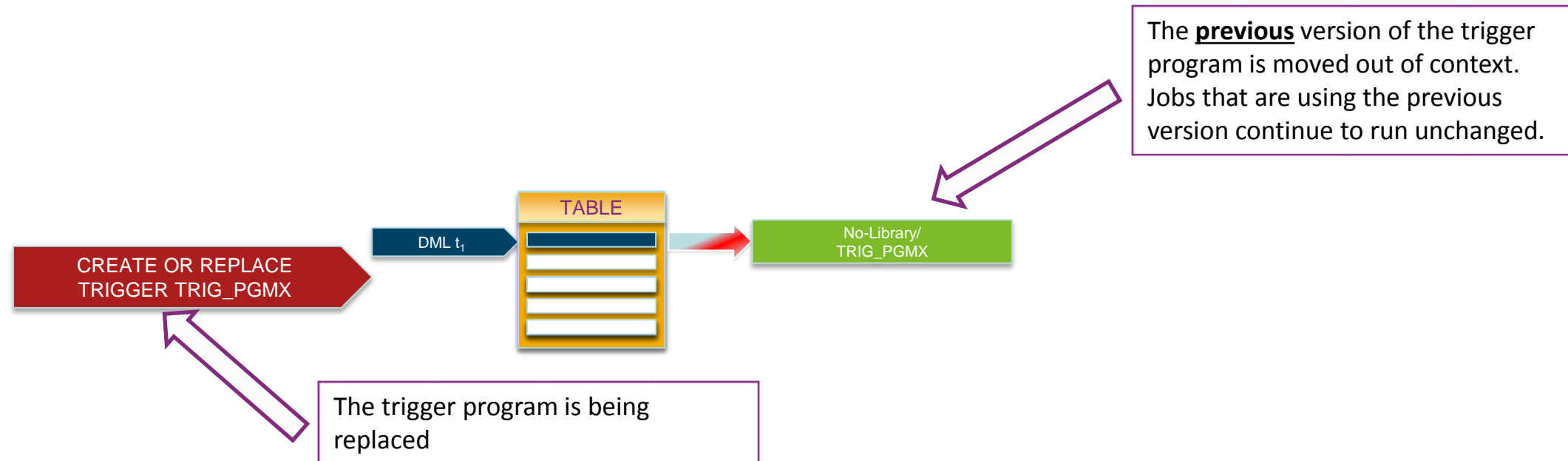
Fair Lock Option

Challenge: Seemingly impossible to make DDL changes in production

Response: **ALLOW_DDL_CHANGES_WHILE_OPEN** QAQQINI control

Benefit: Ability to deploy trigger changes without quiescing user activity

Support: Applies to CREATE TRIGGER, ALTER TRIGGER, DROP TRIGGER, COMMENT ON TRIGGER, and LABEL ON TRIGGER, ADDPFTRG, RMVPFTRG, and CHGPFTRG



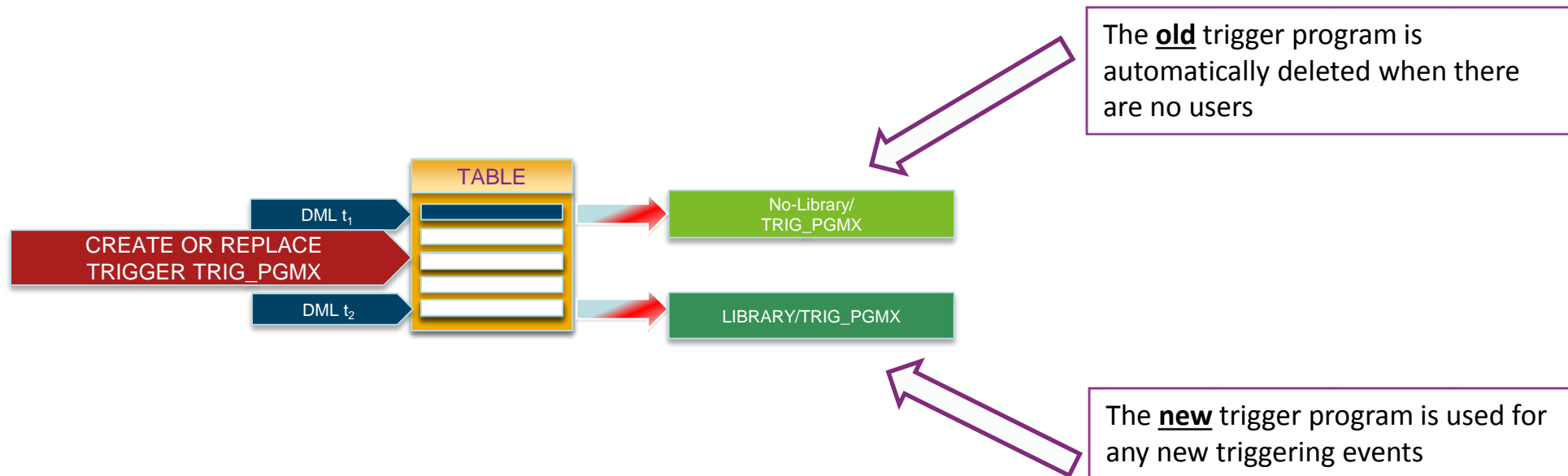
Fair Lock Option

Challenge: Seemingly impossible to make DDL changes in production

Response: **ALLOW_DDL_CHANGES_WHILE_OPEN** QAQQINI control

Benefit: Ability to deploy trigger changes without quiescing user activity

Support: Applies to CREATE TRIGGER, ALTER TRIGGER, DROP TRIGGER, COMMENT ON TRIGGER, and LABEL ON TRIGGER, ADDPFTRG, RMVPFTRG, and CHGPFTRG



Fair Lock Option

- Using the QAQQINI (Query Options) control
 - CHGQRYA
 - OVERRIDE_QAQQINI

```
call qsys2.override_qaqqini(1, '', '');  
call qsys2.override_qaqqini(2,  
                             'ALLOW_DDL_CHANGES_WHILE_OPEN',  
                             '*YES');
```

```
CREATE OR REPLACE TRIGGER toystore.new_hire  
  AFTER INSERT ON toystore.employee  
  FOR EACH ROW MODE DB2SQL  
  UPDATE toystore.company_stats  
  SET Number_of_employees = Number_of_employees + 1;
```

Dawn May – “i Can” Blog – “Managing Trigger Programs in Production”

<http://ibmsystemsmag.com/blogs/i-can/august-2017/manage-trigger-programs-in-productions/>

Implicit Remote Database Access



Implicit Remote Database Access

- A local application can run SQL statements against a local database or a remote database.
- To specify a remote database, you can use a three-part name. A three-part name consists of the RDB name, schema/library name, and object name.
 - SQL naming: <database-name>.<schema-name>.<object-name>
 - System naming: <database-name>/<schema-name>/<object-name>

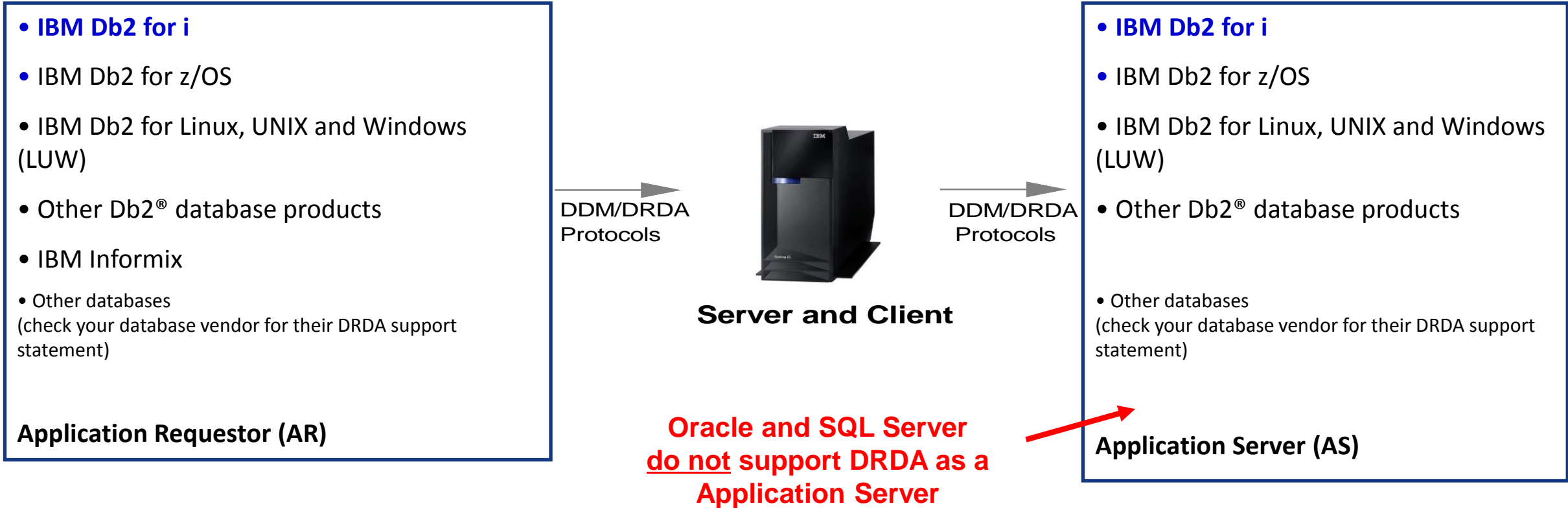
```
CL: ADDRDBDIRE RDB(X1423P2) RMTLOCNAME(X1423P2 *IP);

CREATE TABLE X1423P2.TOYSTORE.EMPLOYEE (EMPNO CHAR(6),
                                           FIRSTNME CHAR(10),
                                           LASTNAME CHAR(15));

INSERT INTO X1423P2.TOYSTORE.EMPLOYEE
VALUES ('000002', 'Michael', 'Thompson');

SELECT * FROM X1423P2.TOYSTORE.EMPLOYEE;
```

Implicit Remote Database Access



Article: Improve Your Data Center with Three-part Name Aliases
<http://iprodeveloper.com/database/improve-your-data-center-three-part-name-aliases>

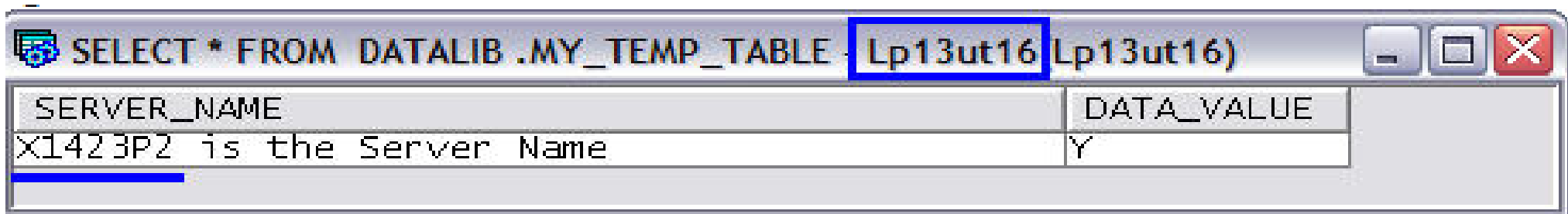
Article: Achieve improved database interoperability with SQL and RDB aliases
<http://www.ibm.com/developerworks/ibmi/library/i-improved-database-interoperability-sql-rdb/index.html>

CREATE TABLE with remote SUBSELECT

- CREATE TABLE AS allows the select to reference a single remote database
- Db2 for i recognizes the remote connection and implicitly manages the connection

```
CREATE TABLE DATALIB.MY_TEMP_TABLE (SERVER_NAME, DATA_VALUE)
  AS (SELECT CURRENT_SERVER CONCAT ' is the Server Name',
        IBMREQD
        FROM X1423P2.SYSIBM.SYSDUMMY1) WITH DATA;

SELECT * FROM DATALIB.MY_TEMP_TABLE;
```



SERVER_NAME	DATA_VALUE
X1423P2 is the Server Name	Y

CREATE TABLE with remote SUBSELECT

- Use of an ALIAS is the best practice for remote 3-part names because it shields the application. (database transparency)
- Notice how the text of the query does not change

```
CREATE OR REPLACE ALIAS DATALIB.TARGET_TABLE
FOR X1423P2.SYSIBM.SYSDUMMY1;

CREATE TABLE DATALIB.MY_TEMP_TABLE(Server_Name) AS
(SELECT CURRENT_SERVER CONCAT ' is the Server Name'
FROM DATALIB.TARGET_TABLE)
WITH DATA;

CREATE OR REPLACE ALIAS DATALIB.TARGET_TABLE
FOR LP01UT18.SYSIBM.SYSDUMMY1;

INSERT INTO DATALIB.MY_TEMP_TABLE
(SELECT CURRENT_SERVER CONCAT ' is the Server Name'
FROM DATALIB.TARGET_TABLE);

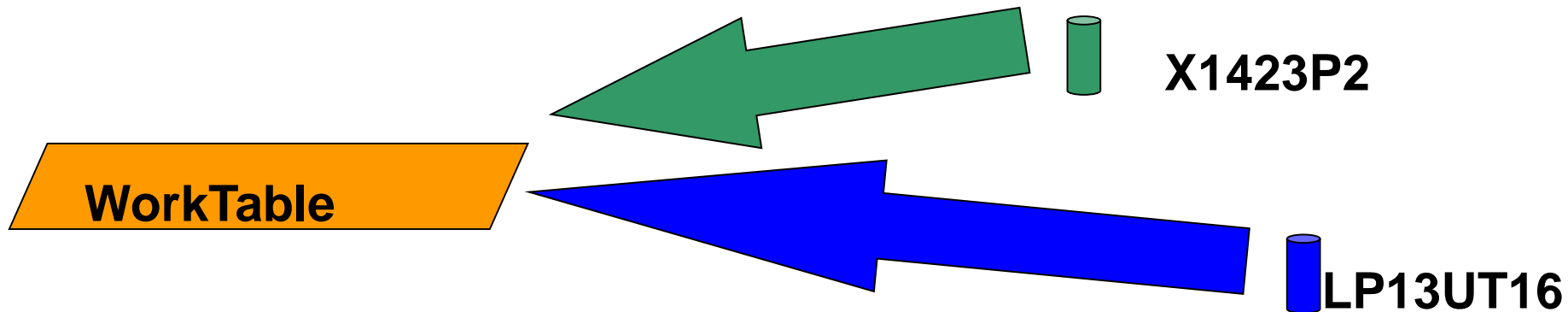
SELECT * FROM DATALIB.MY_TEMP_TABLE;
```

RDB alias support for 3-part SQL statements

- Instead of using CREATE ALIAS (SQL) to deploy database transparency, the Relational Database Directory Entry Alias name can be used.

```
ADDRDBDIRE RDB(X1423P2 MYALIAS) RMTLOCNAME(X1423P2 *IP)  
INSERT INTO WORKTABLE SELECT * FROM MYALIAS.TOYSTORE.EMPLOYEE
```

```
CHGRDBDIRE RDB(LP13UT26 MYALIAS) RMTLOCNAME(LP13UT26 *IP)  
INSERT INTO WORKTABLE SELECT * FROM MYALIAS.TOYSTORE.EMPLOYEE
```



Note:

The SQL statement text does not change

System Limits



System Limits

Customer Requirements

- We need to be **proactive** and understand our posture against important system limits
- I want to be able to **recognize** trends and run-away situations
- We need to **understand** how spikes like month-end processing affect our consumption of operating system resources.

IBM i Innovation

- **Leverage** the integrated IBM i operating system to instrument the automated recognition of resource consumption
- Accommodate **different types** of consumption (Job, Object, ASP, and System)
- **Db2 for i** is the repository
- Define the **criteria** for which limits are worthy of tracking



Patent filed March/2013 → "Integrated Limits Tracking, Trending, and Reporting"

System Limits

Added in IBM i 6.1

Customer
Requirements

- We need to be **proactive** and understand our posture against important system limits
- I want to be able to **recognize** trends and run-away situations
- We need to **understand** how spikes like month-end processing affect our consumption of operating system resources.

IBM i
Innovation

- **Leverage** the integrated IBM i operating system to instrument the automated recognition of resource consumption
- Accommodate **different types** of consumption (Job, Object, ASP, and System)
- **Db2 for i** is the repository
- Define the **criteria** for which limits are worthy of tracking



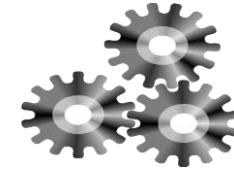
Patent filed March/2013 → "Integrated Limits Tracking, Trending, and Reporting"

System Limits Architecture

User Job – Long running data purge running with commitment control



Low priority QDBSRVnn jobs



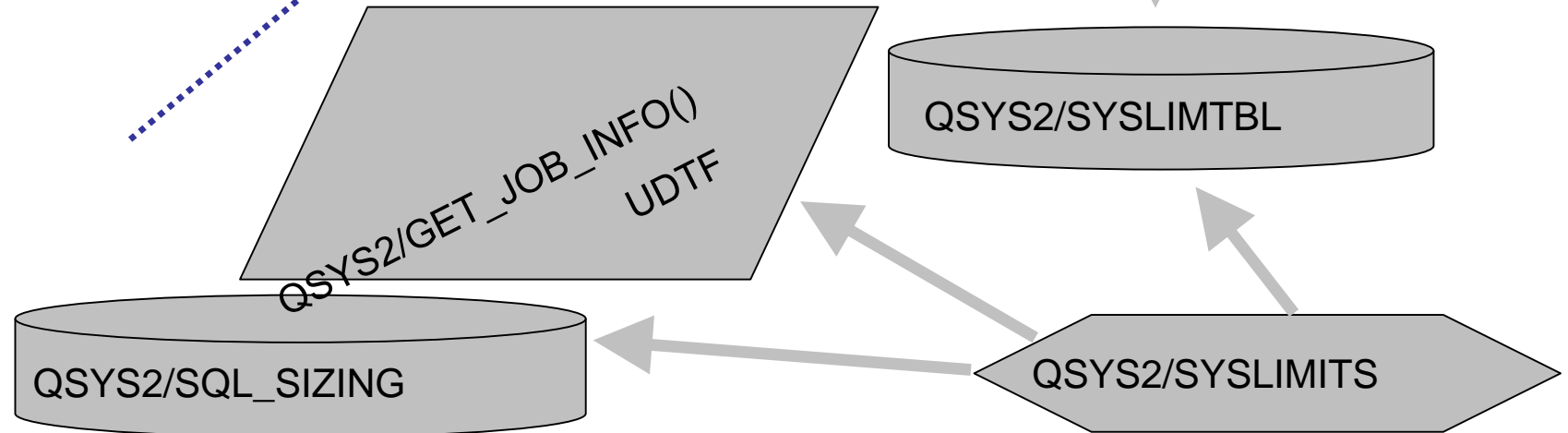
System event notification

10,000th row deleted

110,000th row deleted

210,000th row deleted

INSERT into...



System Limits – Where does the data reside

Object	Type	Purpose
QSYS2/SYSLIMTBL	*FILE SQL Table	System wide (including iASP) physical file repository for tracked System Limits. Designed to have the smallest storage footprint.
QSYS2/GET_JOB_INFO	User Defined Table Function	Accepts a job name as input and returns a single row of information about an active job.
QSYS2/SQL_SIZING	*FILE SQL Table	Table where architected limits are defined, including translated descriptions.
QSYS2/SYSLIMITS	*FILE SQL View	The external interface which joins detail from the preceding three resources.

System Limits – Documentation

www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzajq/rzajqserviceshealth.htm

Table 4. Work management limits

Limit description	Limit ID	Maximum	Floor	Increment
Maximum number of jobs	19000	970,000	1,000	400
Maximum number of spool files	19002	2,610,000	10,000	5,000
Maximum number of spooled files in each independent ASP	19003	10,000,000	10,000	5,000

System Limits – Work Management

```

WITH TT(JOB_MAXIMUM)
  AS (SELECT CURRENT_NUMERIC_VALUE
       FROM QSYS2.SYSTEM_VALUE_INFO
       WHERE SYSTEM_VALUE_NAME = 'QMAXJOB')
SELECT LAST_CHANGE_TIMESTAMP AS INCREMENT_TIME, CURRENT_VALUE AS JOB_COUNT,
TT.JOB_MAXIMUM, DEC(DEC(CURRENT_VALUE,19,2) / DEC(TT.JOB_MAXIMUM,19,2) *
100,19,2) AS PERCENT_CONSUMED
FROM QSYS2.SYSLIMITS, TT
WHERE LIMIT_ID = 19000 ORDER BY CURRENT_VALUE DESC
    
```

INCREMENT_TIME	JOB_COUNT	JOB_MAXIMUM	PERCENT_CONSUMED
2015-05-18 00:33:25.439414	71408	163520	43.66
2015-05-16 08:00:13.560947	71008	163520	43.42
2015-05-18 01:00:23.118807	70031	163520	42.82
2015-05-12 22:42:48.345298	69008	163520	42.20
2015-05-12 22:42:33.200108	68608	163520	41.95
2015-05-12 22:31:28.636105	68208	163520	41.71
2015-05-18 01:01:01.333811	68140	163520	41.67
2015-05-18 01:02:01.376725	65246	163520	39.90
2015-05-18 01:07:04.412267	54952	163520	33.60
2015-05-12 21:47:34.281314	49808	163520	30.45

Deleting data while under Commitment Control

SELECT SIZING_NAME, CURRENT_VALUE FROM QSYS2.SYSLIMITS WHERE JOB_NAME = upper('297851/Q ... - Rchaptf3.rch.s...

SIZING_NAME	CURRENT_VALUE
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	10000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	10000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	110000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	110000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	210000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	210000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	310000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	310000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	410000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	410000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	510000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	510000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	610000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	610000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	710000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	710000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	810000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	810000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	910000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	910000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	810000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	710000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	610000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	510000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	410000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	310000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	210000
MAXIMUM NUMBER OF ROW CHANGE OPERATIONS IN A UNIT OF WORK	110000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	10000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	810000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	710000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	610000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	510000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	410000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	310000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	210000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	110000
MAXIMUM NUMBER OF ROWS LOCKED IN A UNIT OF WORK	10000



Floor



Increments reflect increasing number of deleted rows

Note... we deleted 1 million rows, the high point is not recorded



Commit or Rollback releasing the locks

Protection automated with a Trigger

- Built into ACS
- Insert from Examples...

The screenshot shows the 'Run SQL Scripts' window in IBM ACS. The 'Examples' pane on the left lists 'SYSLIMITBL' and 'System Health - System Limits tracking'. The main editor displays the following SQL code:

```
-- category: IBM i Services
-- description: System Health - System Limits tracking
--
-- Description: Enable alerts for files which are growing near the maximum
--
CL: ALCOBJ OBJ((QSYS2/SYSLIMITBL *FILE *EXCL)) CONFLICT(*RQSRLS) ;
CL: DLCOBJ OBJ((QSYS2/SYSLIMITBL *FILE *EXCL));

CREATE OR REPLACE TRIGGER qsys2.system_limits_large_file
AFTER INSERT ON qsys2.syslimtbl
REFERENCING NEW AS n
FOR EACH ROW MODE db2row
SET OPTION USRPRF = *owner,
      DYNUSRPRF = *owner
BEGIN ATOMIC
  DECLARE v_cmdstmt VARCHAR(200) ;
  DECLARE v_error INTEGER;
  DECLARE EXIT HANDLER FOR sqlexception
    SET v_error = 1;
  /* -----
  /* If a table has exceeded 80% of this limit, alert the operator
  /* -----
  /* 15000 == MAXIMUM NUMBER OF ALL ROWS IN A PARTITION
  /* (max size = 4,294,967,288)
  /* -----
  IF (n.limit_id = 15000 AND
      n.current_value > ((
        SELECT supported_value
        FROM qsys2.sql_sizing
        WHERE sizing_id = 15000)
      * 0.8))
  THEN
    SET v_cmdstmt = 'SNDMSG MSG(''Table: ' concat n.system_schema_name concat
    '/' concat n.system_object_name concat ' (' concat n.system_table_member
    concat ') IS GETTING VERY LARGE - ROW COUNT = ' concat current_value
    concat ' '' TOUSR(*SYSOPR) MSGTYPE(*INFO) ' ;
    CALL qsys2.qcmdexc(v_cmdstmt);
  END IF;
END;
```

An 'Insert' button is visible at the bottom right of the editor window.

Integrated File System

Added in IBM i 7.2

Limit description	Limit ID	Maximum	Floor	Increment
Number of objects linked in a directory	18402	0	100,000	10,000
Maximum number of directories linked in a directory	18403	1,000,000	1,000	1,000
Maximum number of file system objects in *SYSBAS ASPs	18404	2,147,483,647	100,000	10,000
Maximum number of file system objects in an independent ASP	18405	2,147,483,647	100,000	10,000
Maximum number of document library objects in a folder	18406	65510	1,000	500
Number of document library objects in the system ASP	18407	0	100,000	10,000
Maximum number of document library objects in a user ASP	18408	1,000,000	100,000	10,000
Maximum number of bytes in a stream file	18409	1,099,511,627,776	16,777,216	1,048,576
Maximum number of bytes in a document	18410	2,147,483,647	16,777,216	1,048,576

Find the largest IFS files

```
SELECT LASTCHG, JOB_NAME, ASP_NUMBER, IFS_PATH_NAME, USER_NAME,
CURRENT_VALUE FROM QSYS2.SYSLIMITS WHERE LIMIT_ID = 18409 ORDER BY
CURRENT_VALUE DESC;
```

LASTCHG	JOB_NAME	ASP_NUMBER	IFS_PATH_NAME	USER_NAME	CURRENT_VALUE
2015-01-03 23:...	337465/VCPDTA/QJVACMDSRV	1	/orbtrc.18122014.0929.20.txt	VCPDTA	1099511535858
2015-01-03 23:...	337465/VCPDTA/QJVACMDSRV	1	/orbtrc.18122014.0929.20.txt	VCPDTA	1099510485672
2015-01-03 23:...	337465/VCPDTA/QJVACMDSRV	1	/orbtrc.18122014.0929.20.txt	VCPDTA	1099509435486
2015-01-03 23:...	337465/VCPDTA/QJVACMDSRV	1	/orbtrc.18122014.0929.20.txt	VCPDTA	1099508385300
2015-01-03 23:...	337465/VCPDTA/QJVACMDSRV	1	/orbtrc.18122014.0929.20.txt	VCPDTA	1099507335114
2015-02-26 15:...	407956/QACE/QP0ZSPWP	1	/QIBM/UserData/ACE/log/server.log	QACE	61870255
2015-02-27 12:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	49286416
2015-02-27 10:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	48237784
2015-02-27 08:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	47189088
2015-02-27 06:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	46140361
2015-02-27 04:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	45091718
2015-02-27 02:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	44042651
2015-02-27 00:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	42993987
2015-02-26 22:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	41945337
2015-02-26 20:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	40896606
2015-02-26 18:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	39848021
2015-02-26 16:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	38799357
2015-02-26 14:...	405803/QBRMS/Q1ACPDST	1	/tmp/brms/qbrms	QBRMS	37750700
2015-02-26 13:...	413714/QBRMS/QBRMSYNC	1	/tmp/brms/qbrms	QBRMS	36702048
2015-02-26 13:...	413707/HERBST/QPADEV09K6	1	/tmp/brms/flightrec	HERBST	22021074
2015-02-27 04:...	407982/EBANK/QJVACMDSRV	1	/ebank/logs/EBANK00052.log	EBANK	22020395
2015-02-27 00:...	407982/EBANK/QJVACMDSRV	1	/ebank/logs/EBANK00052.log	EBANK	20971806
2015-02-26 21:...	407982/EBANK/QJVACMDSRV	1	/ebank/logs/EBANK00052.log	EBANK	19923136
2015-02-26 18:...	407982/EBANK/QJVACMDSRV	1	/ebank/logs/EBANK00052.log	EBANK	18874543
2015-02-26 14:...	407982/EBANK/QJVACMDSRV	1	/ebank/logs/EBANK00052.log	EBANK	17825926

Set Server Subsystem Routing



QSYS2.SET_SERVER_SBS_ROUTING() – Procedure

This procedure can be used to configure alternate subsystems by user and IBM i server name. This allows an IBM i administrator to relocate users into subsystems that are configured to meet user expectations or to protect overall system resources.

❑ Procedure QSYS2.SET_SERVER_SBS_ROUTING()

Procedure Parameters:

1. **Authorization Name**

The user profile name

2. **Server Name**

QZDASOINIT, QRWTSRVR, and many others or *ALL

3. **Alternate Subsystem Name**

The name of the subsystem to use

4. **Allow Rollover (YES or NO)**

If the alternate subsystem cannot be used, should the default subsystem be used or should the connect fail?

Authorization name can be:

- ✓ User name
- ✓ Group name
- ✓ Supplemental Group name

Example...

- Construct a subsystem that will constrain the amount of system resources available to users who are known to execute ad hoc queries.

```
CL: CRTSBS SBSD(QGPL/ADHOCSBS) POOLS((1 *BASE))
    TEXT('Ad hoc users SBS');

CL: CRTJOBQ QGPL/ADHOCJOBQ TEXT('Ad hoc users job queue');

CL: ADDJOBQE SBSD(QGPL/ADHOCSBS) JOBQ(QGPL/ADHOCJOBQ)
    MAXACT(100) SEQNBR(40);

CL: CRTCLS CLS(QGPL/ADHOCCLS) RUNPTY(55) TIMESLICE(100)
    TEXT('Ad hoc class');
-- Repeat the ADDPJE for each server name
CL: ADDPJE SBSD(QGPL/ADHOCSBS) PGM(QSYS/QZDASOINIT)
    JOBD(QGPL/QDFTSVR) CLS(QGPL/ADHOCCLS);

CL: STRSBS SBSD(QGPL/ADHOCSBS);

CL: CALL QSYS2.SET_SERVER_SBS_ROUTING(
    'JOEUSER', '*ALL', 'ADHOCSBS', 'NO');
```

QSYS2.SERVER_SBS_ROUTING – View

- QSYS2.SERVER_SBS_ROUTING is used to access the alternative subsystem user configuration
- The configuration detail is stored within the ***USRPRF** objects
- **Authorization requirements** to change the configuration:
 - *SECADM user special authority
 - *OBJMGT and *USE to the target *USRPRF

```
SELECT * FROM QSYS2.SERVER_SBS_ROUTING;
```

<i>AUTHORIZATION_NAME</i>	<i>QRWTSRVR_SUBSYSTEM</i>	<i>QZDASOINIT_SUBSYSTEM</i>
<i>JOEUSER</i>	<i>ADHOC SBS</i>	<i>ADHOC SBS</i>
<i>QRWTSRVR_ROLLOVER</i>	<i>QZDASOINIT_ROLLOVER</i>	<i>QZRCSRVS_ROLLOVER</i>
<i>NO</i>	<i>NO</i>	<i>NO</i>

Configurable Servers

www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzajq/rzajqprocsetrouting.htm

Table 1. Servers and default subsystems

Server Description	Server Name	Default subsystem
Central server	QZSCSRVS	QUSRWRK
Database server	QZDASOINIT	QUSRWRK
Data queue server	QZHQSSRV	QUSRWRK
DDM	QRWTSRVR	QUSRWRK
DRDA	QRWTSRVR	QUSRWRK
File server	QPWFSEVSO	QSERVER
Network print server	QNPSEVSO	QUSRWRK
Remote command server	QZRCSRVS	QUSRWRK

Grouping similar Navigator users

Manage ACS users

- Avoid having all users run in QUSRWRK, with the same priority
- Setup once and manage the Group Profile

```
-- Description: Reposition all Navigator users into a
--              controlled subsystem and do not allow
--              connections to fall-over into the default
--              subsystem (QUSRWRK or QSERVER) if the
--              INAVGRP subsystem cannot be used
```

```
CALL QSYS2.SET_SERVER_SBS_ROUTING(
    AUTHORIZATION_NAME => 'INAVGRP',
    SERVER_NAME        => '*ALL',
    SUBSYSTEM_NAME     => 'INAVSBS',
    ALLOW_ROLLOVER     => 'NO');
```

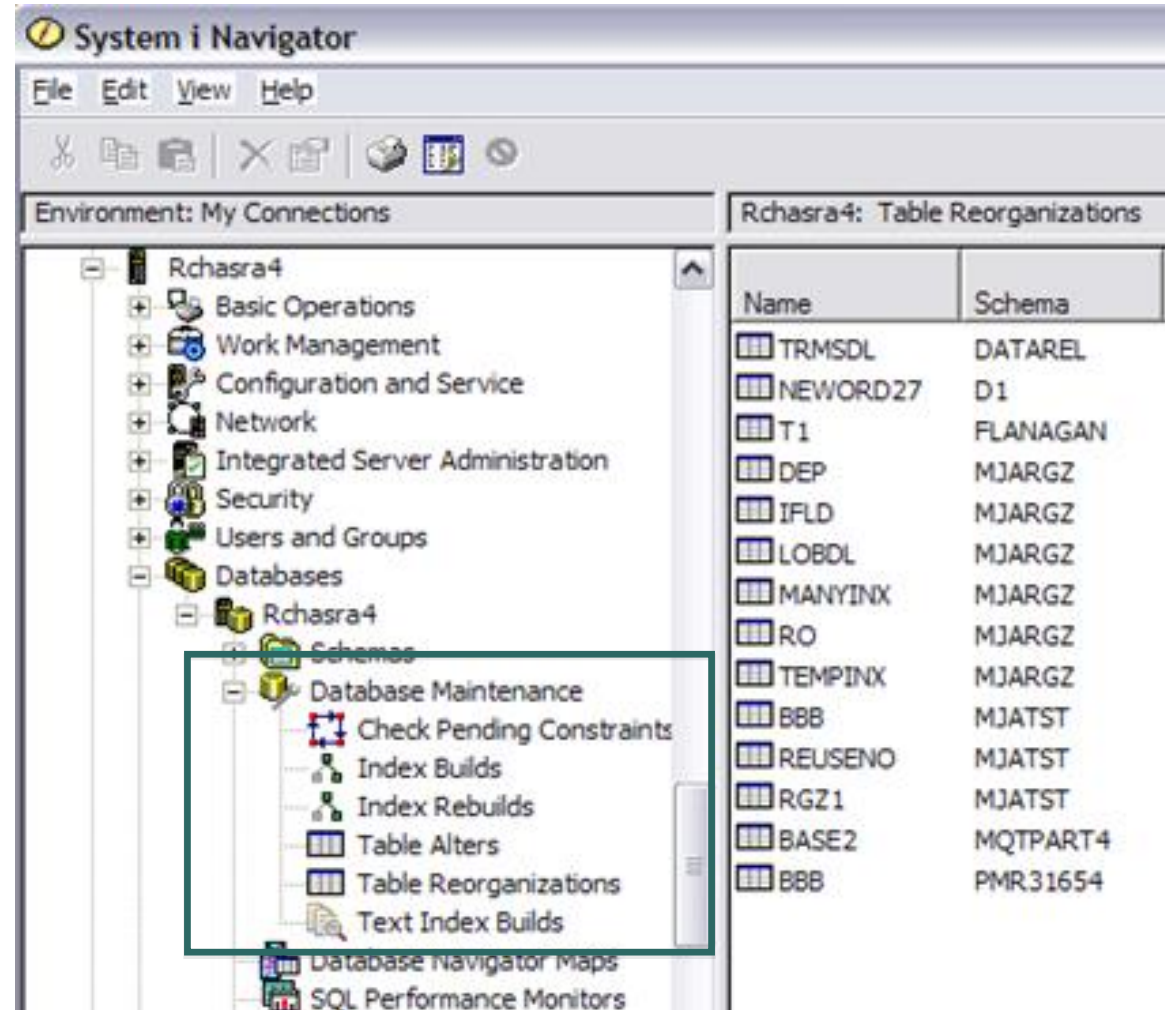
Database Maintenance in System i Navigator



Database Maintenance in System i Navigator

Two primary use cases:

1. Examine history of long running database maintenance operations
2. Monitor active database maintenance operations



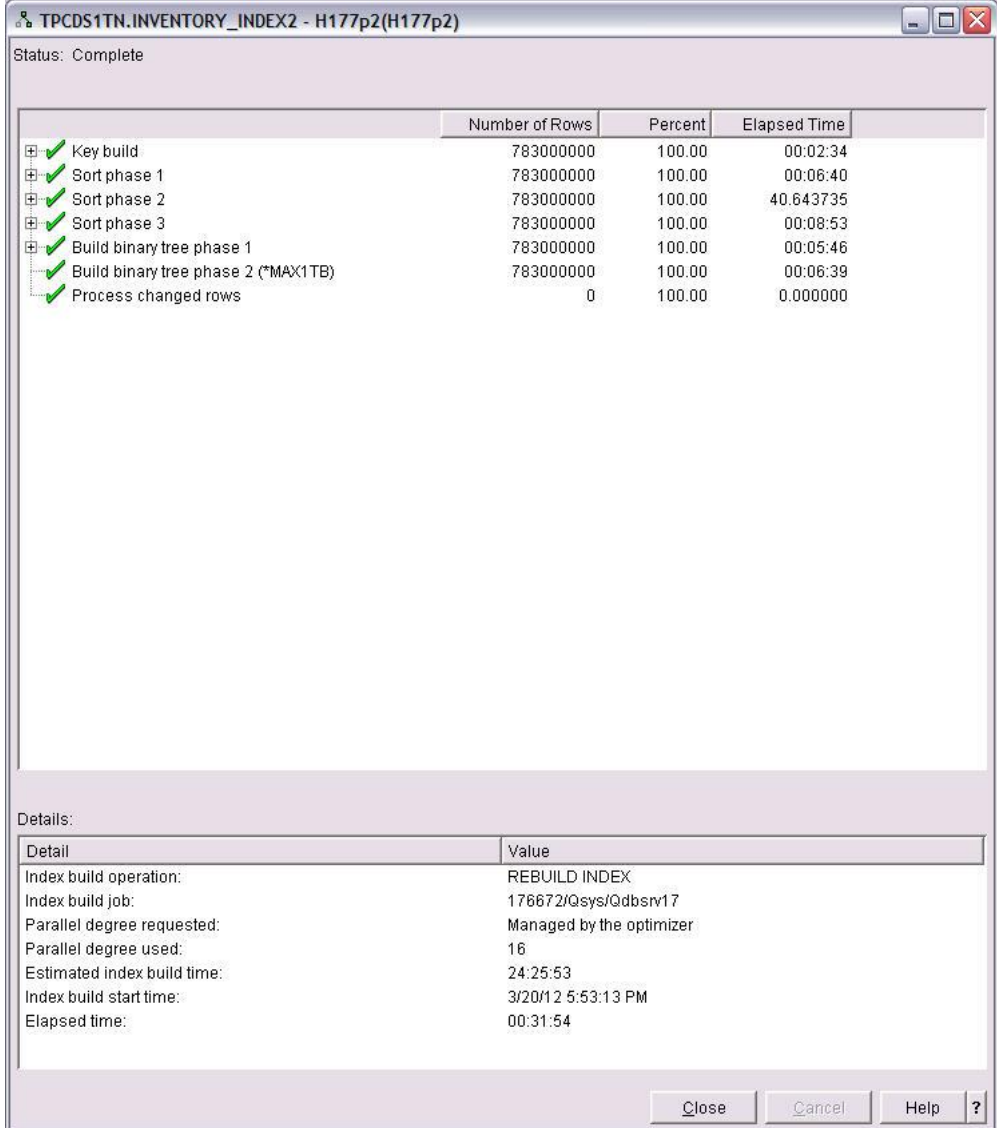
Database Maintenance in System i Navigator

- **Index builds**
 - **Alters with unique index(es)**
 - **Reorganize**
 - **New index create**

- **Index rebuilds**
 - **Restored base table without restoring the index**
 - **Alters with non-unique index(es)**

See the progress...ETA is a guess

If an anomaly occurred...
check the status file before repeating



TPCDS1TN.INVENTORY_INDEX2 - H177p2(H177p2)

Status: Complete

	Number of Rows	Percent	Elapsed Time
✓ Key build	783000000	100.00	00:02:34
✓ Sort phase 1	783000000	100.00	00:06:40
✓ Sort phase 2	783000000	100.00	40.643735
✓ Sort phase 3	783000000	100.00	00:08:53
✓ Build binary tree phase 1	783000000	100.00	00:05:46
✓ Build binary tree phase 2 (*MAX1TB)	783000000	100.00	00:06:39
✓ Process changed rows	0	100.00	0.000000

Details:

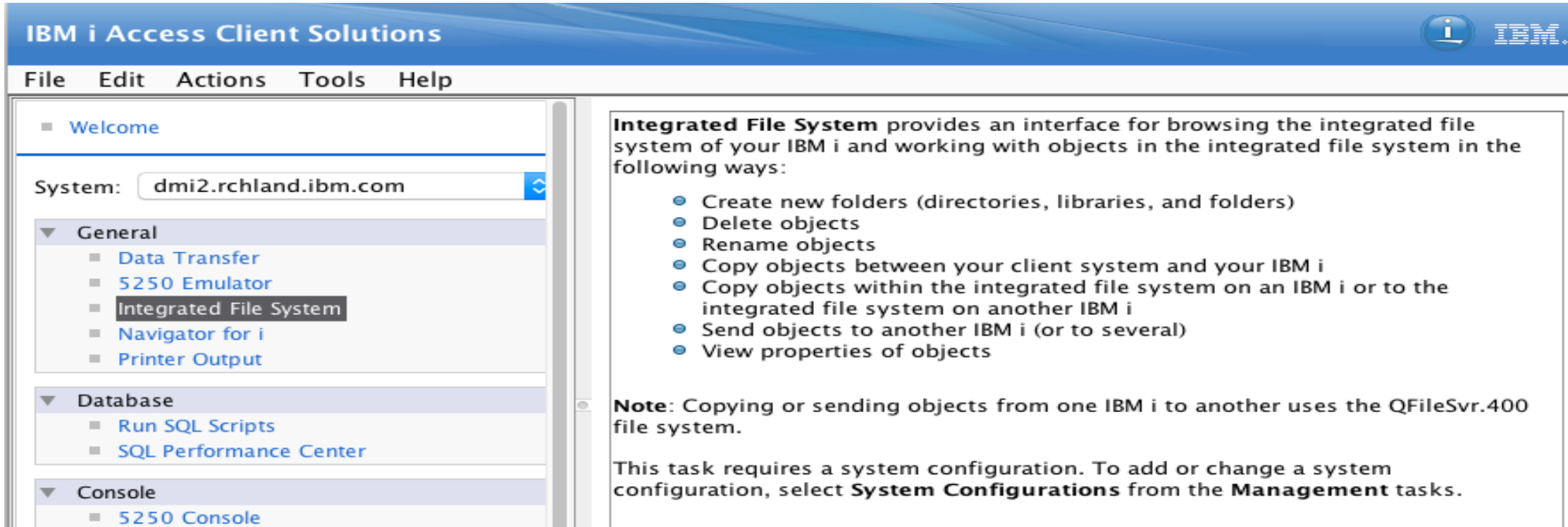
Detail	Value
Index build operation:	REBUILD INDEX
Index build job:	176672/Qsys/Qdbsrv17
Parallel degree requested:	Managed by the optimizer
Parallel degree used:	16
Estimated index build time:	24:25:53
Index build start time:	3/20/12 5:53:13 PM
Elapsed time:	00:31:54

Close Cancel Help ?

Access Client Solutions – Integrated File System



IBM i Access Client Solutions – Integrated File System



The screenshot shows the IBM i Access Client Solutions application window. The title bar reads "IBM i Access Client Solutions" and includes an information icon and the IBM logo. The menu bar contains "File", "Edit", "Actions", "Tools", and "Help". The main window is divided into a left sidebar and a right pane. The sidebar shows a "Welcome" section and a "System:" dropdown menu set to "dmi2.rchland.ibm.com". Below this are three expandable sections: "General", "Database", and "Console". Under "General", the "Integrated File System" option is highlighted with a dark background. The right pane contains text describing the Integrated File System and a list of supported actions.

IBM i Access Client Solutions

File Edit Actions Tools Help

Welcome

System: dmi2.rchland.ibm.com

General

- Data Transfer
- 5250 Emulator
- Integrated File System**
- Navigator for i
- Printer Output

Database

- Run SQL Scripts
- SQL Performance Center

Console

- 5250 Console

Integrated File System provides an interface for browsing the integrated file system of your IBM i and working with objects in the integrated file system in the following ways:

- Create new folders (directories, libraries, and folders)
- Delete objects
- Rename objects
- Copy objects between your client system and your IBM i
- Copy objects within the integrated file system on an IBM i or to the integrated file system on another IBM i
- Send objects to another IBM i (or to several)
- View properties of objects

Note: Copying or sending objects from one IBM i to another uses the QFileSvr.400 file system.

This task requires a system configuration. To add or change a system configuration, select **System Configurations** from the **Management** tasks.

IBM i Access Client Solutions – Integrated File System

File Edit View Actions

dmi2.rchland.ibm.com Filter

Directory: /

Name	Size (KB)	Last Modified
bin	98	May 4, 2016 9:17:01 AM EDT
bin.prv		8 April 29, 2016 7:02:04 AM EDT
core	25435	May 4, 2016 5:22:57 AM EDT
dev	40	April 29, 2016 8:18:00 AM EDT
etc		8 May 4, 2016 5:59:09 AM EDT
home		8 May 6, 2016 9:04:05 AM EDT
lib	73	May 4, 2016 5:18:31 AM EDT
lib64		8 May 4, 2016 5:19:46 AM EDT
QDLS		8 April 29, 2016 6:47:29 AM EDT
QFileSvr.400		8 April 29, 2016 6:47:30 AM EDT
QIBM		8 April 29, 2016 8:16:01 AM EDT
QNTC		8 April 29, 2016 6:47:30 AM EDT
QOpenSys		8 April 29, 2016 6:47:29 AM EDT
QOPT		8 April 29, 2016 6:47:30 AM EDT
QSR		12 May 6, 2016 9:14:03 AM EDT

21 objects listed

Messages

Send

Source

System: dmi2.rchland.ibm.com

Directory: /QIBM/UserData/OS400/Navigator

Objects:

Name
isc

Destination

Systems:

Host Name
common1.frankeni.com
common1.idevcloud.com
common1.iinthecloud.com
dmi2.rchland.ibm.com
etc3t1.rchland.ibm.com

Options

*ALL – All objects are copied and replaced

*NEW – Only new objects are copied

*OLD – Only existing objects are copied and replaced

OK Cancel

Thank You



Reorganize Physical File Member

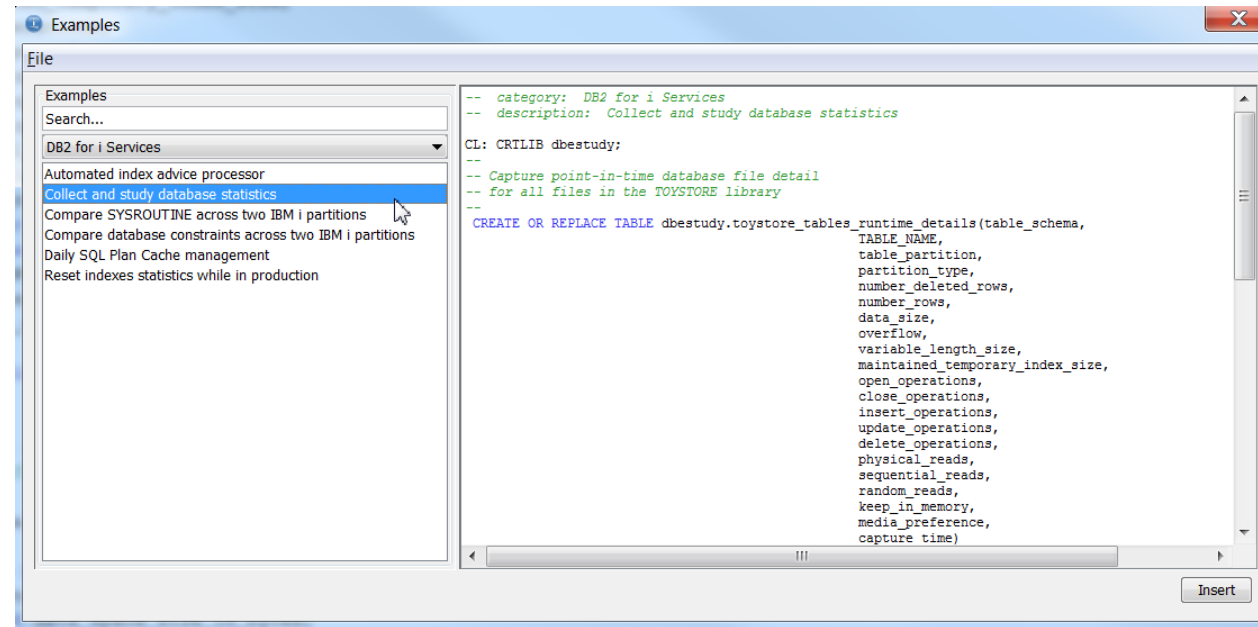
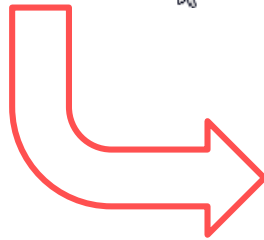
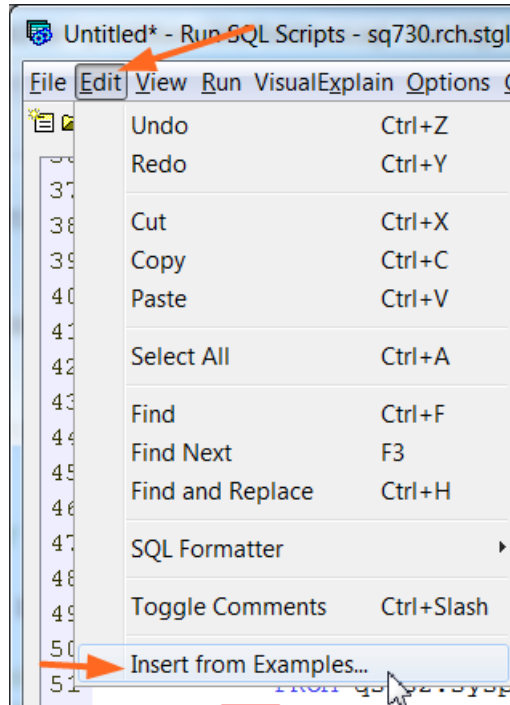
Tips and tricks for expediting reorganizations

- DB2 Symmetric Multiprocessing (SMP) – Parallel reorganize and index builds
- Use database catalogs to assess the need for reorganizes and the best strategy

Relevant enhancements in IBM i 7.2, 7.3 and future

- [Database Reorganization – User specified starting point](#)
- [Honor priority change for parallel index build](#)
- [Enhanced index build logic for highly concurrent environments](#)
- [QSYS2.SYSLIMITS](#)
- **System i Navigator's Database Maintenance support**

Reorganize Physical File Member



Reorganize Physical File Member

```
--  
-- Identify candidates for physical file reorganization  
-- Examine files with more than a million rows deleted  
--  
    SELECT table_schema,  
           TABLE_NAME,  
           number_rows AS valid_rows,  
           number_deleted_rows AS deleted_rows,  
           data_size AS data_space_size_in_bytes,  
           DEC(DEC(number_deleted_rows, 19, 2) /  
DEC(number_rows + number_deleted_rows, 19, 2) * 100, 19, 2) AS  
           deleted_row_percentage  
    FROM dbestudy.toystore_tables_runtime_details a  
   WHERE number_deleted_rows > 1000000  
   ORDER BY deleted_row_percentage DESC;
```

Reorganize Physical File Member

```
--
-- Review the distribution of deleted records
--
SELECT 1000000 - COUNT(*) AS DELETEDCNT
FROM item_fact A
GROUP BY BIGINT(RRN(A) / 1000000)
ORDER BY BIGINT(RRN(A) / 1000000)
```



DELETEDCNT
20,664
17,499
16,208
17,498
14,876
14,930
15,904
14,882
14,994
15,462
12,842
2,661
8,300
15,632
24,731
40,018
35,307
67,094
15,521
96,449
171,772
183,208
191,215
218,160
219,675
214,143
153,749
804
6,077
704,994
994,322
997,080
996,021
995,465

Reorganize Physical File Member

	ALWCANCEL(*NO)		ALWCANCEL(*YES)		
	KEYFILE (*NONE)	KEYFILE (*FILE or keyfile)	KEYFILE (*RPLDLTRCD)	KEYFILE (*NONE)	KEYFILE (*FILE or keyfile)
Cancel and restart	No	No	Yes	Yes	Yes
Concurrent Access	No	No	Yes	Yes	Yes
Parallel processing	Only index rebuilds	Only index rebuilds	Data movement and index rebuilds	Data movement and index rebuilds	Data movement and index rebuilds
Non-parallel performance	Very fast	Fast	Very fast	Slower	Slowest
Temporary storage	Double data storage	Double data storage	Journal receiver storage	Journal receiver storage	Journal receiver storage
LIFO KEYFILE index processing	N/A	Duplicates reversed	N/A	N/A	Duplicate ordering preserved
Index processing (non-KEYFILE)	Synchronous or asynchronous rebuilds	Synchronous or asynchronous rebuilds	Maintain indexes or synchronous or asynchronous rebuilds	Maintain indexes or synchronous or asynchronous rebuilds	Maintain indexes or synchronous or asynchronous rebuilds
Final row position exact	Yes	Yes	Only if LOCK(*EXCL) and not restarted	Only if LOCK(*EXCL) and not restarted	Only if LOCK(*EXCL) and not restarted
Amount of CPU and I/O used	Smallest	Next smallest	Smallest	More	Most
Variable length segment reorganize	Good	Good	Worse	Worse	Worse
Allows referential integrity parents and FILE LINK CONTROL DataLinks	Yes	Yes	No	No	No
Allows QTEMP and Database Cross Reference Files	Yes	Yes	No	No	No
HABP replication cost	Minimal - one journal entry	Minimal - one journal entry	More - journal entires for all rows moved	Most - journal entires for all rows moved	Most - journal entires for all rows moved

Database – Constraints



Gems you've owned for decades...

- Data-Centric technologies save you time and money

Pre-2010



Long Time Gems

- SQL Views → Dawn of time
- Primary Keys → V3R1M0
- Foreign Keys → V3R1M0
- Native Triggers → V3R1M0
- Check Constraints → V3R6M0
- SQL Triggers → V5R1M0

More Recently Added...

- Data-Centric technologies save you time and money

2010

2011

2012

2013

2014

2015

2016

2017



7.1 and earlier



7.2



7.3



i next

- Field Procedures
- Implicitly hidden columns
- Range and Hash Partitioning
- Row Change Timestamp
- And more...

- Row Permissions
- Column Masks
- Media Preference
- Memory Preference
- EVI Only Access

- Temporal Tables
- Generated Columns for Auditing
- New OLAP Specifications

Constraints

- **Constraints** enforce the business rules defined by the data model

There are three types of constraints:

1. A **unique constraint** is a rule that forbids duplicate values in one or more columns within a table.
Two forms:
 - a) **Unique Key(s)** – a unique index is used
 - b) **Primary Key** – a single column with a unique, non-NULL value (sometimes an Identity value is used)
2. A **referential constraint** is a logical rule about values in one or more columns in one or more tables
3. A **check constraint** sets restrictions on data added to a specific table



Constraints Enforce the Rules

EMP_MAST
EMP_MAST_PK
EMPNO
FIRSTNME
MIDINIT
LASTNAME
WORKDEPT [FK]
PHONENO
HIREDATE
JOB
EDLEVEL
SEX
BIRTHDATE
SALARY
BONUS
COMM
EM_ROW_CHANGE_TS

DEPARTMENT
DEPTNO
DEPTNAME
MGRNO [FK]
ADMRDEPT [FK]
LOCATION

Unique Keys
Provide Single Row Retrieval

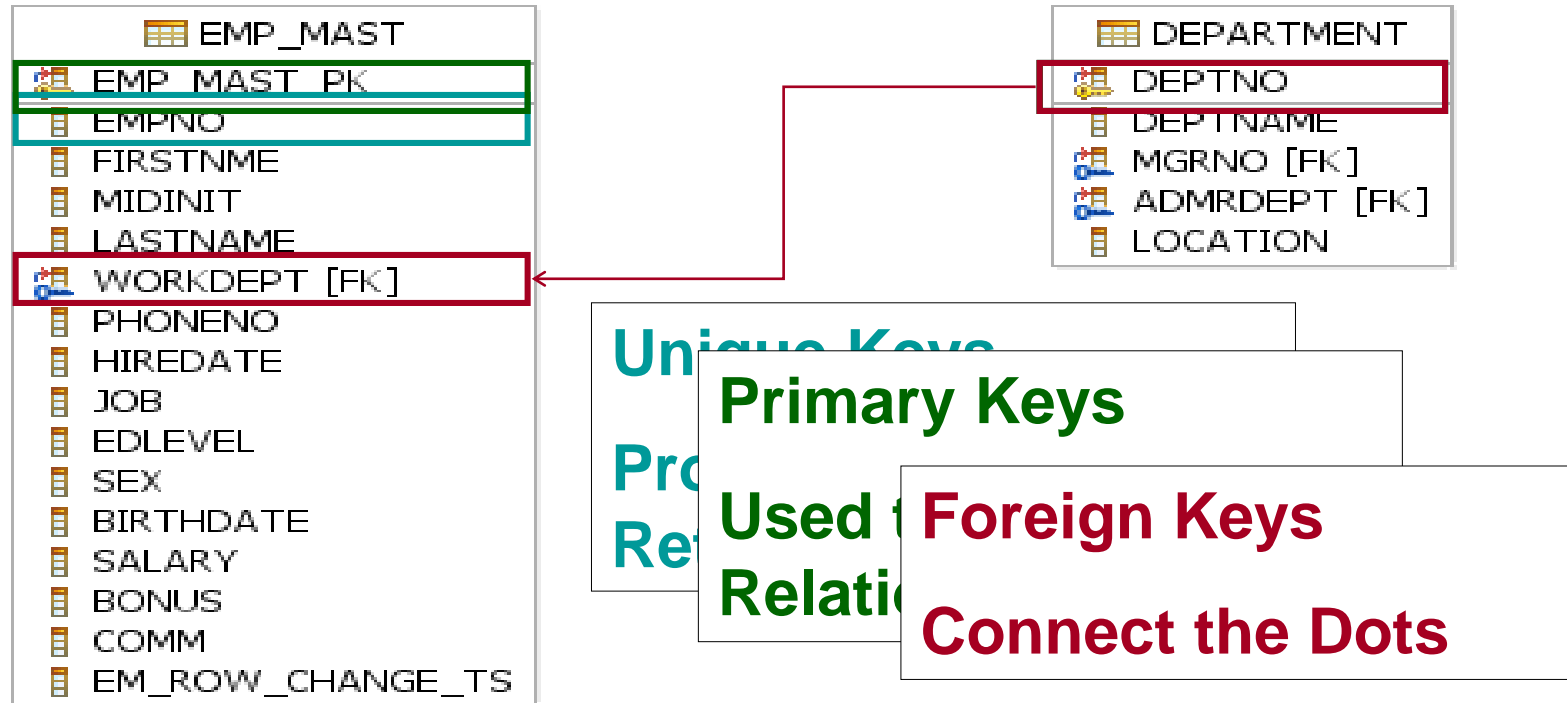
Constraints Enforce the Rules

EMP_MAST	
	EMP_MAST_PK
	EMPNO
	FIRSTNME
	MIDINIT
	LASTNAME
	WORKDEPT [FK]
	PHONENO
	HIREDATE
	JOB
	EDLEVEL
	SEX
	BIRTHDATE
	SALARY
	BONUS
	COMM
	EM_ROW_CHANGE_TS

DEPARTMENT	
	DEPTNO
	DEPTNAME
	MGRNO [FK]
	ADMRDEPT [FK]
	LOCATION

Unique Keys
Primary Keys
Used to Establish Relationships

Constraints Enforce the Rules



Constraints Enforce the Rules

EMP_MAST	
	EMP_MAST_PK
	EMPNO
	FIRSTNME
	MIDINIT
	LASTNAME
	WORKDEPT [FK]
	PHONENO
	HIREDATE
	JOB
	EDLEVEL
	SEX
	BIRTHDATE
	SALARY
	BONUS
	COMM
	EM_ROW_CHANGE_TS

DEPARTMENT	
	DEPTNO
	DEPTNAME
	MGRNO [FK]
	ADMRDEPT [FK]
	LOCATION

Check Constraint
Must be 'M' or 'F'

Unique Keys
Primary Keys
Used to Relate
Foreign Keys
Connect the Dots