



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

Time travel with Db2 for i - Temporal tables on IBM i 7.3





Which release should you choose?

Enhancements in IBM i 7.2?

- Database performance
 - ✓ SQE handles Native DB access
 - ✓ New I/O Costing Model
 - ✓ EVI Only Access
- Data-centric security
 - ✓ Row & Column Access Control for SQL and DDS file
- Developer productivity
 - ✓ Default parameters on functions
 - ✓ Built-in Global Variables
 - ✓ Many other improvements
- Workload insight
 - ✓ Improved SQL Plan Cache
 - ✓ Performance Data Perspectives

Enhancements in IBM i 7.3?

- **Data-centric history**
 - ✓ **System-period Temporal table support for SQL tables and DDS created physical files**
- **Data-centric accountability**
 - ✓ **Generated columns for SQL and DDS files**
 - ✓ Authority Collection to avoid excess authority
- On-Line Analytical Processing (OLAP)
 - ✓ New OLAP built-in functions
 - ✓ Improved capabilities for Db2 Web Query, Cognos Analytics and other BI tools
- Improved value from priced options
 - ✓ DB2 SMP – Parallel execution of OLAP
 - ✓ DB2 Multisystem – Attach/Detach partitions

MARTY... WE NEED A TEMPORAL TABLE



Knowledge Center and IBM i 7.3

Read about it... (live links in the pdf)

- [SQL Reference - What's New](#)
- [SQE Optimizer - What's New](#)
- [Temporal Tables - Administration](#)
- [Temporal Tables - Programming](#)
- [Generated Columns for Auditing](#)
- [On-Line Analytical Processing \(OLAP\) specifications](#)
- [OLAP specifications - Examples](#)
- [IBM i Navigator - database enhancements](#)



Generated Columns

Generated Columns

- **What you have on previous releases:**
 - When was this row last updated?
(*row-change-timestamp-clause*)

```
--  
-- Add a row change timestamp generated column  
--  
ALTER TABLE TOYSTORE.SALES  
  ADD COLUMN LAST_CHANGE FOR COLUMN LASTCHG  
  TIMESTAMP(6)  
  GENERATED ALWAYS FOR EACH ROW ON UPDATE  
  AS ROW CHANGE TIMESTAMP  
  NOT NULL  
  IMPLICITLY HIDDEN;
```

Generated Columns

- **What you have on previous releases:**
 - When was this row last updated?
(*row-change-timestamp-clause*)
- **New Generated expressions in IBM i 7.3:**
 - DATA CHANGE OPERATION (I/U/D)
 - Special register
 - Built-in Global Variable

special-register

```
|--+CURRENT CLIENT_ACCTNG-----+
+-CURRENT CLIENT_APPLNAME----+
+-CURRENT CLIENT_PROGRAMID--+
+-CURRENT CLIENT_USERID-----+
+-CURRENT CLIENT_WRKSTNNAME--+
+-CURRENT SERVER-----+
'+-SESSION_USER+-----+'
'-USER-----'
```

built-in-global-variable

```
|--+QSYS2.JOB_NAME-----+
+QSYS2.SERVER_MODE_JOB_NAME---+
+SYSIBM.CLIENT_HOST-----+
+SYSIBM.CLIENT_IPADDR-----+
+SYSIBM.CLIENT_PORT-----+
+SYSIBM.PACKAGE_NAME-----+
+SYSIBM.PACKAGE_SCHEMA-----+
+SYSIBM.PACKAGE_VERSION-----+
+SYSIBM.ROUTINE_SCHEMA-----+
+SYSIBM.ROUTINE_SPECIFIC_NAME--+
'-SYSIBM.ROUTINE_TYPE-----'
```

Generated Columns

- **Add generated columns to SQL Tables & DDS Created Physicals**
- **No need to change applications**

```
ALTER TABLE  account
ADD COLUMN  audit_type_change CHAR (1)
            GENERATED ALWAYS AS (DATA CHANGE OPERATION)
ADD COLUMN  audit_user VARCHAR(128)
            GENERATED ALWAYS AS (SESSION_USER)
ADD COLUMN  audit_client_IP VARCHAR(128)
            GENERATED ALWAYS AS (SYSIBM.CLIENT_IPADDR)
ADD COLUMN  audit_job_name VARCHAR(28)
            GENERATED ALWAYS AS (QSYS2.JOB_NAME)
```

Temporal Tables

Time based questions

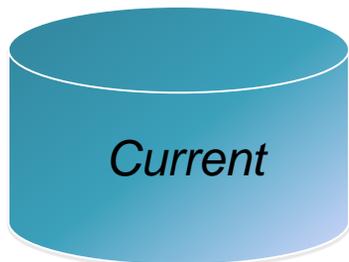
- With Temporal Table & Generated columns, you can:
 - **Show me the client reps from two years ago**
 - **Produce an inventory report using a different point in time**
 - **Who deleted that row?**
 - **Who last updated this row?**

SQL answers

- With Temporal Table & Generated columns, you can:
 - **Show me the client reps from two years ago**
SELECT CLIENT_REP FROM ACCOUNTS
FOR SYSTEM_TIME AS OF CURRENT_TIMESTAMP – 2 YEARS;
 - **Produce an inventory report using a different point in time**
SET CURRENT TEMPORAL SYSTEM_TIME
'2016-03-22 17:00:00';
CALL GENERATE_INVENTORY_REPORT();
 - **Who deleted that row?**
SELECT AUDIT_USER, AUDIT_JOB FROM SALES
FOR SYSTEM_TIME FROM CURRENT DATE – 1 MONTH TO
CURRENT DATE WHERE AUDIT_OP = 'D'
 - **Who last updated this row?**
SELECT AUDIT_USER, AUDIT_CLIENT_IP FROM ITEM_FACT
WHERE ITEM_KEY = '125A16'

History – Do It Yourself

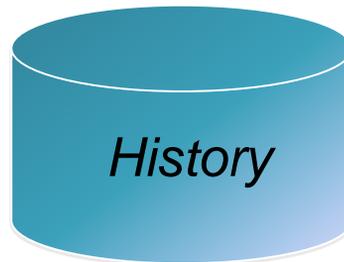
Accessing Data
• SELECT



Modifying Data
• INSERT
• UPDATE
• DELETE



Accessing Data
• SELECT



Modifying Data
• INSERT
• UPDATE
• DELETE

History – Db2 for i Managed

Accessing Data
• SELECT

~~Accessing Data~~
~~• SELECT~~



Modifying Data
• INSERT
• UPDATE
• DELETE

~~Modifying Data~~
~~• INSERT~~
~~• UPDATE~~
• DELETE
(DBE Only)

Temporal construction

```
ALTER TABLE account
  ADD COLUMN row_birth
    TIMESTAMP(12) NOT NULL
    GENERATED ALWAYS AS ROW BEGIN
  ADD COLUMN row_death
    TIMESTAMP(12) NOT NULL
    GENERATED ALWAYS AS ROW END
  ADD COLUMN transaction_time
    TIMESTAMP(12)
    GENERATED ALWAYS AS TRANSACTION START ID
  ADD PERIOD SYSTEM_TIME (row_birth, row_death);

CREATE TABLE account_hist LIKE account;

ALTER TABLE account
  ADD VERSIONING USE HISTORY TABLE account_hist;
```

Temporal construction

```
ALTER TABLE account
  ADD COLUMN row_birth
    TIMESTAMP(12) NOT NULL IMPLICITLY HIDDEN
    GENERATED ALWAYS AS ROW BEGIN
  ADD COLUMN row_death
    TIMESTAMP(12) NOT NULL IMPLICITLY HIDDEN
    GENERATED ALWAYS AS ROW END
  ADD COLUMN transaction_time
    TIMESTAMP(12) IMPLICITLY HIDDEN
    GENERATED ALWAYS AS TRANSACTION START ID
  ADD PERIOD SYSTEM_TIME (row_birth, row_death);

CREATE TABLE account_hist LIKE account;

ALTER TABLE account
  ADD VERSIONING USE HISTORY TABLE account_hist;
```

Accessing a Temporal Table

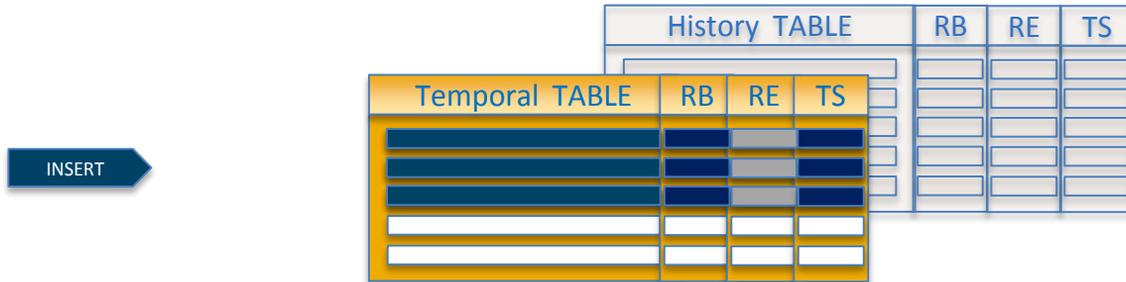
- SQL statements reference the current table, DB2 accesses the history table as needed
- New clauses on the SELECT statement
 - **FOR SYSTEM_TIME AS OF <value>**
 - **FOR SYSTEM_TIME FROM <value> TO <value>**
 - **FOR SYSTEM_TIME BETWEEN <value> AND <value>**
- New special register
 - **CURRENT TEMPORAL SYSTEM_TIME**



Temporal in motion

Inserting rows does not impact the history table

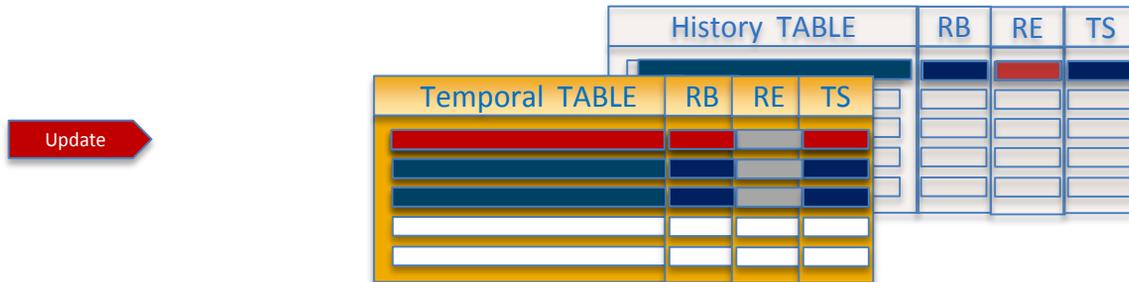
- ROW BEGIN (RB) Column – timestamp when the row was born
- ROW END (RE) Column – set to “end of time”



Temporal in motion

Updating rows causes rows to be added to the history table

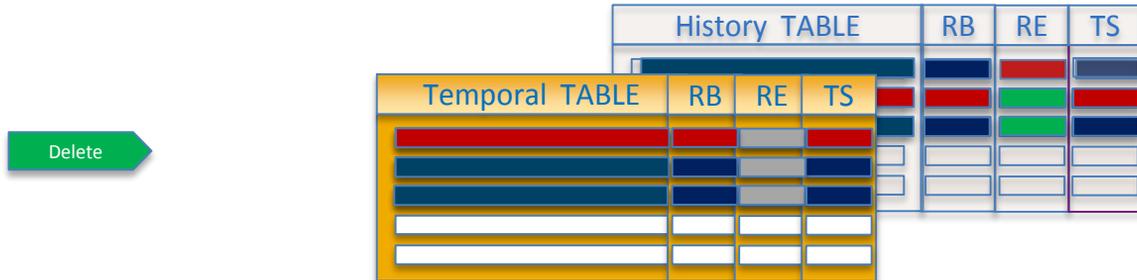
- ROW BEGIN (RB) Column – timestamp when the row was born
- ROW END (RE) Column – the death of the row results in the RE of the historical row matching the RB of the active row



Temporal in motion

Deleting rows removes them from the temporal table and adds them to history table

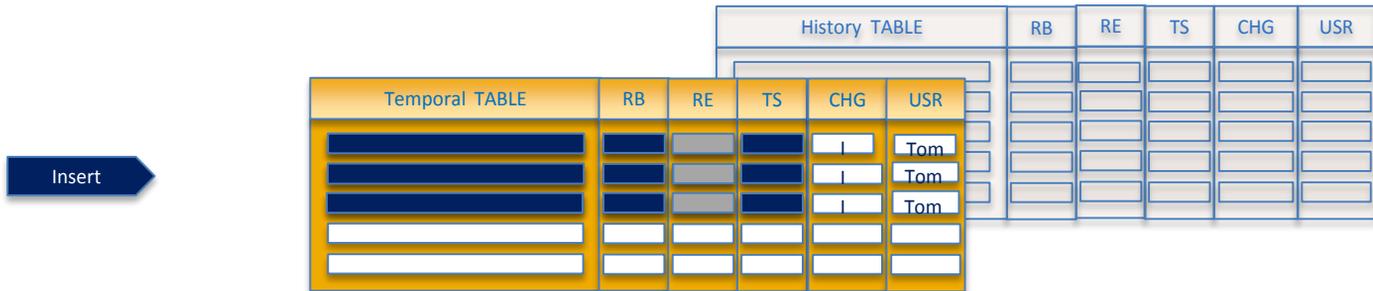
- ROW END (RE) Column – set to the death time of the row



Temporal in motion

History table stores previous versions of a system-period temporal table's rows

- ROW BEGIN (RB) Column – timestamp when the rows were born
- ROW END (RE) Column – set to “end of time”
- Data Change Operation (CHG) – ‘I’ for INSERT
- Session User (USR) – identity of inserter



Temporal in motion

History table stores previous versions of a system-period temporal table's rows

- ROW BEGIN (RB) Column – Birth
- ROW END (RE) Column – Death
- Data Change Operation (CHG) – 'U' for UPDATE
- Session User (USR) – identity of updater



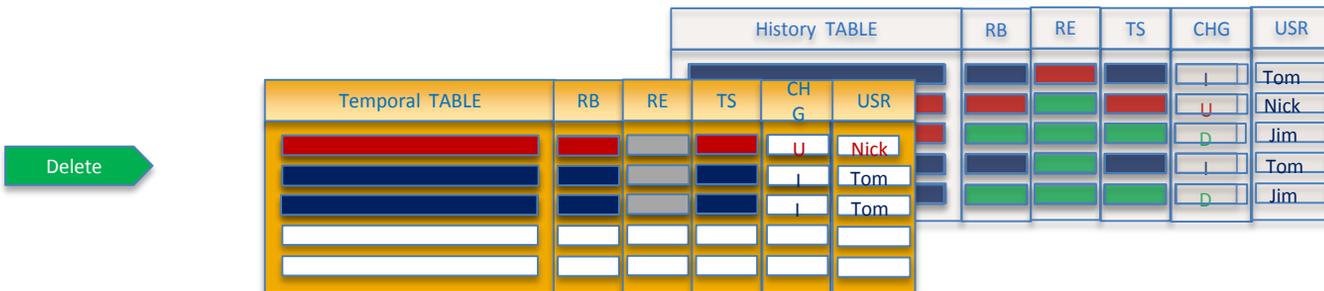
Temporal TABLE	RB	RE	TS	CHG	USR

History TABLE	RB	RE	TS	CHG	USR

ON DELETE ADD EXTRA ROW

History table stores previous versions of a system-period temporal table's rows

- ROW BEGIN (RB) Column – Birth
- ROW END (RE) Column – Death
- Data Change Operation (CHG) – 'D' for DELETE
- Session User (USR) – identity of deleter



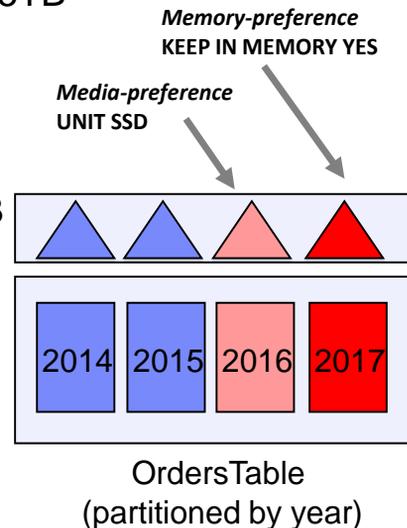
DB2 Multisystem (Option 27)

- **Provides ability to partition tables**

- Non-partitioned tables are limited to 4.2B rows or 1.7TB
- Partitioning multiplies these limits by up to 256 times
 - Limits of over one trillion rows and 435TB
- Management benefits
 - Efficient removal of old data
 - Faster save times
 - Ability to detach partitions in IBM i 7.3
 - Efficient remote journaling

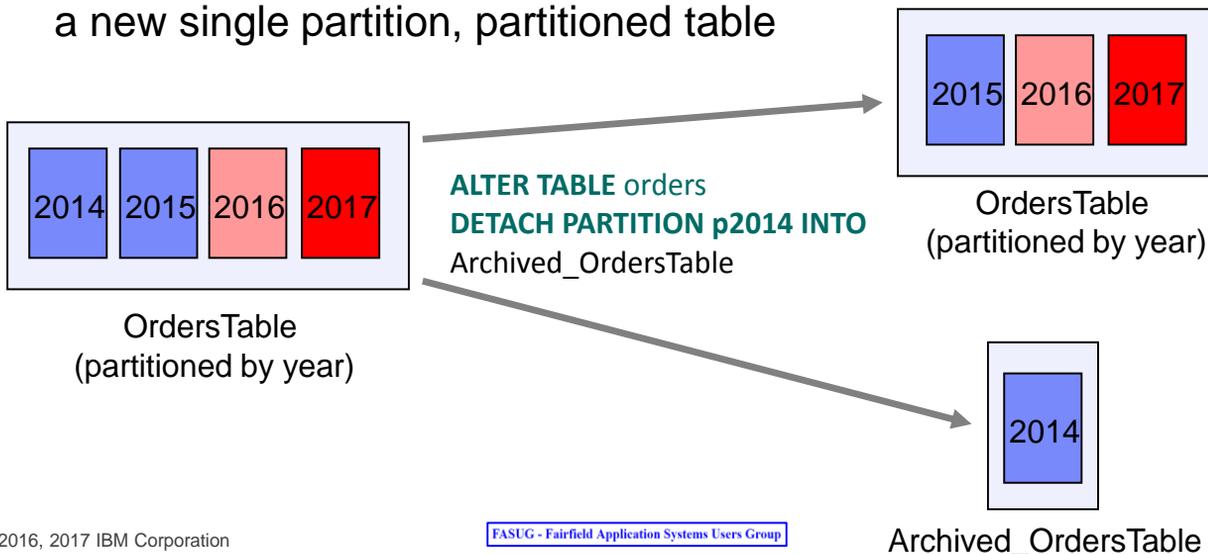
- **Planning is critical**

- White Paper:
Table Partitioning Strategies for Db2 for i
<https://ibm.biz/PartitionedTablesIBMi>
- Db2 for i VLDB Consulting Workshop
<https://ibm.biz/DB2CoEworkshops>



ATTACH and DETACH Partition

- ALTER TABLE DETACH PARTITION allows for the efficient roll-out of a partition that is no longer needed to be kept online
- ❑ ALTER TABLE DROP PARTITION – Delete the data
- ❑ ALTER TABLE DETACH PARTITION – Retain the data, creating a new single partition, partitioned table



Temporal history

- Temporal table history tables contain rows that are natural to organize by time.
- The history table can be partitioned, even if the system-time temporal table is not partitioned
- Why consider using local partitioning for your history table?
 1. Reduced index maintenance
 2. Faster save times
 3. Ease of use when data is has aged beyond relevance
 4. Better solution for remote journaling

```
CREATE TABLE account_history LIKE account
PARTITION BY RANGE ( row_death)
(PARTITION p2016 STARTING ('01/01/2016') INCLUSIVE ENDING ('01/01/2017') EXCLUSIVE,
PARTITION p2017 STARTING ('01/01/2017') INCLUSIVE ENDING ('01/01/2018') EXCLUSIVE,
PARTITION p2018 STARTING ('01/01/2018') INCLUSIVE ENDING ('01/01/2019') EXCLUSIVE,
PARTITION p2019 STARTING ('01/01/2019') INCLUSIVE ENDING ('01/01/2020') EXCLUSIVE
);
```

Db2 for i priced OS options

Try before you buy! On any IBM i 7.x release!

DB2 Symmetric Multiprocessing – Option 26

DB2 Multisystem – Option 27

The IBM Lab Services Db2 for IBM i team has the ability to allow you to evaluate either of these options for up to 70 days, for no charge.

This is a simpler, no strings attached, way to evaluate these valuable database options.

Contact...

Rob Bestgen (bestgen@us.ibm.com) or

Scott Forstie (forstie@us.ibm.com)



History behind the scenes

```
SELECT * FROM account
WHERE ACCT_ID = '88880001';
```

ACCT_ID	BALANCE	TRANSACTION_TIME	INSTANCE_BEGIN	INSTANCE_END	TRANSACTION_ID
88880001	60000.00	2014-12-20 10:05:18.617454000000	2014-12-20 10:05:18.617454000000	9999-12-30 00:00:00.000000000000	2014-12-20 10:05:18.617454000000

```
SELECT * FROM account_hist
WHERE ACCT_ID = '88880001';
```

ACCT_ID	BALANCE	TRANSACTION_TIME	INSTANCE_BEGIN	INSTANCE_END	TRANSACTION_ID
88880001	3000.00	2013-01-02 10:02:16.987139000000	2013-01-02 10:02:16.987139000000	2013-05-05 14:36:16.637149000000	2013-01-02 10:02:16.987139000000
88880001	10.00	2013-05-05 14:36:16.637149000000	2013-05-05 14:36:16.637149000000	2013-12-30 10:50:59.637124000000	2013-05-05 14:36:16.637149000000
88880001	50000.00	2013-12-30 10:50:59.637124000000	2013-12-30 10:50:59.637124000000	2014-01-05 10:50:59.611224000000	2013-12-30 10:50:59.637124000000
88880001	9000.00	2014-01-05 10:50:59.611224000000	2014-01-05 10:50:59.611224000000	2014-03-05 21:12:23.321216000000	2014-01-05 10:50:59.611224000000
88880001	1000.00	2014-03-05 21:12:23.321216000000	2014-03-05 21:12:23.321216000000	2014-09-01 14:01:11.111231000000	2014-03-05 21:12:23.321216000000
88880001	100.00	2014-09-01 14:01:11.111231000000	2014-09-01 14:01:11.111231000000	2014-12-20 10:05:18.617454000000	2014-09-01 14:01:11.111231000000

More examples...

- Compare balance **between** different points in time for account 88880001

```
SELECT T1.BALANCE AS BALANCE_2013,  
       T2.BALANCE AS BALANCE_2014  
FROM account FOR SYSTEM_TIME AS OF '2013-12-31' T1,  
     account FOR SYSTEM_TIME AS OF '2014-12-31' T2  
WHERE T1.ACCT_ID = '88880001' AND  
       T2.ACCT_ID = '88880001';
```

BALANCE_2013	BALANCE_2014
50000.00	60000.00

More examples...

- Query all versions of rows for account 88880001

```
SELECT ACCT_ID,  
       BALANCE,  
       BALANCE - LAG(BALANCE,1,0)  
       OVER(ORDER BY TRANSACTION_TIME)  
         AS CHANGES,  
       TRANSACTION_TIME,  
       ROW_DEATH  
FROM account FOR SYSTEM_TIME  
  BETWEEN '0001-01-01' AND '9999-12-30'  
WHERE ACCT_ID= '88880001'  
ORDER BY TRANSACTION_TIME ASC;
```

LAG is one of many
new OLAP
specifications added
in IBM i 7.3

ACCT_ID	BALANCE	CHANGES	TRANSACTION_TIME	INSTANCE_END
88880001	3000.00	-2990.00	2013-01-02 10:02:16.987139000000	2013-05-05 14:36:16.637149000000
88880001	10.00	49990.00	2013-05-05 14:36:16.637149000000	2013-12-30 10:50:59.637124000000
88880001	50000.00	-41000.00	2013-12-30 10:50:59.637124000000	2014-01-05 10:50:59.611224000000
88880001	9000.00	-8000.00	2014-01-05 10:50:59.611224000000	2014-03-05 21:12:23.321216000000
88880001	1000.00	-900.00	2014-03-05 21:12:23.321216000000	2014-09-01 14:01:11.111231000000
88880001	100.00	59900.00	2014-09-01 14:01:11.111231000000	2014-12-20 10:05:18.617454000000
88880001	60000.00	-60000.00	2014-12-20 10:05:18.617454000000	9999-12-30 00:00:00.000000000000

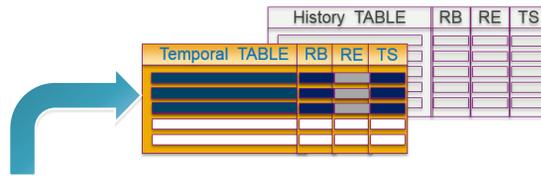
CURRENT TEMPORAL SYSTEM_TIME

- Affects any system-period temporal table in the query
 - Allows reuse of previous functions/procedures
 - Effects queries executed after setting the register
 - Works for external functions/procedures (C/C++/RPG)
 - When this register set to a non-null value:
 - Explicit time specification cannot be used within the SQL query
 - Cursors cannot be updatable

```
SET CURRENT TEMPORAL SYSTEM_TIME = '2014-09-02';  
  
SELECT * FROM account WHERE ACCT_ID = '88880001';
```

System-period temporal table

- Can be either a DDS-created physical file or an SQL table
- Associated with a single history table
- Must be journaled
- Generated columns can be IMPLICITLY HIDDEN
- Things you **can do** while versioning is enabled:
 - ❖ Add columns or expand their width
 - ❖ Attach Partitions
- Things you **can't do** while versioning is enabled:
 - ❖ Add Generated columns
 - ❖ Drop Columns or reduce their width
 - ❖ Drop or Detach Partitions
 - ❖ Cannot use DSPDBR or DSPFD to view temporal existence



History table

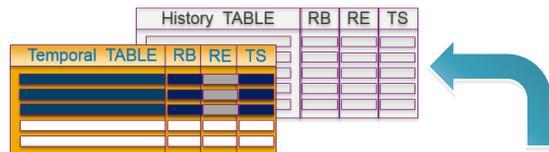
- Must be an SQL table and reside within the same library
- Must match the production table format
- Must be journaled
- Can be partitioned or non-partitioned
- Reorganize Physical File Member does not change history
- Things you **can do** with history

- ❖ Remove old history

- DELETE
- TRUNCATE
- ALTER TABLE DROP PARTITION
- ALTER TABLE DETACH PARTITION

- Things you **can't do** with history:

- ❖ Drop, alter or change the history table
- ❖ Cannot use DSPDBR or DSPFD to view temporal existence



Temporal TABLE	RB	RE	TS

History TABLE	RB	RE	TS

SYSTIME - Bind Option

System Time Sensitivity is controlled at the program level:

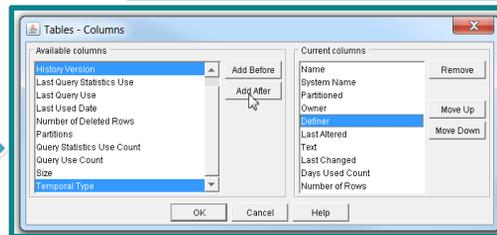
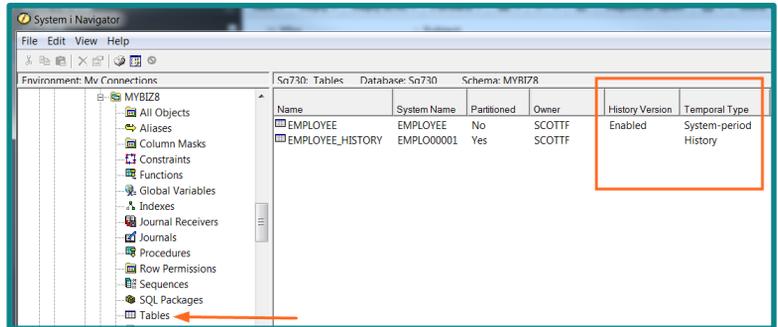
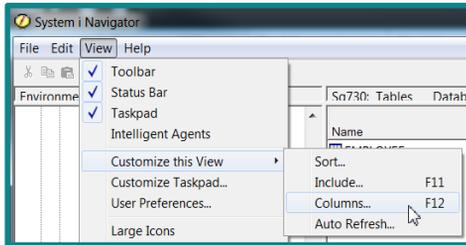
- SYSTEM_TIME_SENSITIVE column within QSYS2.SYSPROGRAMSTAT
 - NULL or 'NO' – Program is not time sensitive
 - 'YES' – Program is time sensitive
- Programs built prior to IBM i 7.3 are by default, **not time sensitive**
 - CURRENT TEMPORAL SYSTEM_TIME is ignored
- Programs re(built) on IBM i 7.3 are by default, **time sensitive**
 - CURRENT TEMPORAL SYSTEM_TIME is applied when queries reference temporal tables

Build time controls:

- Routines (SQL/External) → SET OPTION SYSTIME = *YES or *NO
- CRTSQLxxx → OPTION(*SYSTIME or *NOSYSTIME)
- RUNSQLSTM → SYSTIME(*YES or *NO)

Navigator and Temporal

Schemas → Tables ... Add Temporal columns to your Navigator view



Navigator and Temporal

Generate SQL ...

Use the Temporal versioning option to generate complete SQL

The screenshot shows the 'Generate SQL' dialog box in IBM Db2 Navigator SQL. The 'Output' tab is selected, and the 'Standards' section is expanded. The 'Temporal versioning (for table objects)' checkbox is checked, indicated by an orange arrow. The 'Output' section is also expanded, showing various formatting options, with 'OR REPLACE clause' checked. The 'Run SQL Scripts' window displays the generated SQL script, which includes a primary key constraint, a row and column access control (RCDFMT) statement, and an ALTER TABLE statement with the 'ADD VERSIONING USE HISTORY TABLE' clause, highlighted by an orange arrow. The status bar at the bottom of the SQL window indicates a successful connection to the database.

```
SQL will be generated for the following objects:
```

Name	Schema
EMPLOYEE_HISTORY	MYBIZ8
EMPLOYEE	MYBIZ8

```
File Edit View Run Visual Explain Monitor Options Connection Help
```

```
INSTANCE_END FOR COLUMN ROW_END  TIMESTAMP(12) GENERATED ALWAYS AS ROW END  
TRANSACTION_ID FOR COLUMN TRANS_TIME TIMESTAMP(12) GENERATED ALWAYS AS TRANS  
CONSTRAINT MYBIZ8.Q_MYBIZ8_EMPLOYEE_EMPID_00002 PRIMARY KEY( EMPID ) ,  
PERIOD SYSTEM_TIME ( INSTANCE_BEGIN , INSTANCE_END ) )  
  
RCDFMT EMPLOYEE  ;  
  
GRANT ALTER , DELETE , INDEX , INSERT , REFERENCES , SELECT , UPDATE  
ON MYBIZ8.EMPLOYEE TO SCOTTF WITH GRANT OPTION ;  
  
ALTER TABLE MYBIZ8.EMPLOYEE  
ADD VERSIONING USE HISTORY TABLE MYBIZ8.EMPLOYEE_HISTORY ;  
  
connected to relational database sq730 on sq730 as scottf - 134555/quser/qzdasoinit
```

Standards

- ANSI / ISO
- DB2 family
- Extensions

Output

- Statements formatted for readability
- Informational messages
- Schema qualify names for objects
- System names for objects
- OR REPLACE clause
- DROP statements
- SQL privilege statements
- Labels and comments
- Column CCSID values
- Associated constraints and triggers (for table objects)
- Associated row and column access controls (for table objects)
- Temporal versioning (for table objects)

Navigator and Temporal

Table Definition... Add the three required system generated columns

The image shows a sequence of three screenshots from Oracle SQL Developer, illustrating the process of adding system-generated columns to a table definition. A large blue arrow points from the first screenshot to the second, and a smaller orange arrow points from the 'Definition' menu item in the first screenshot to the table definition window in the second.

First Screenshot: Navigator
Database: Sq730, Schema: MYBIZ8
Table: EMPLOYEE
Menu: Definition

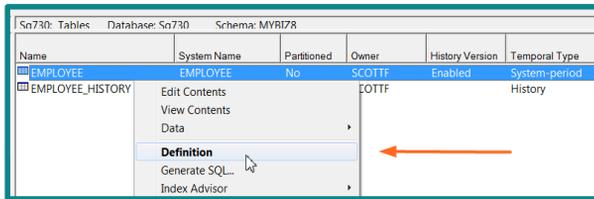
Second Screenshot: Table Definition
Table: MYBIZ8.EMPLOYEE - Sq730(Sq730)
Columns table:

Column Name	System Name	Data Type	Length	Nullable	Generated Value	Default Value
EMPID	EMPID	INTEGER		No		No default
FIRST_NAME	FIRST_NAME	VARCHAR	50	Yes		Null
LAST_NAME	LAST_NAME	VARCHAR	50	Yes		Null
JOB_TITLE	JOB_TITLE	CLOB	100	Yes		Null
INSTANCE_BEGIN	ROWV_BEGIN	TIMESTAMP	12	No	Row begin	
INSTANCE_END	ROWV_END	TIMESTAMP	12	No	Row end	
TRANSACTION_ID	TRANS_TIME	TIMESTAMP	12	Yes	Transaction start ID	

Third Screenshot: Column Definition
Column name: INSTANCE_BEGIN
System name: ROWV_BEGIN
Data type: TIMESTAMP
Precision: 12
Generated value: Row begin
Database manager generates value: Always
Heading line 1: INSTANCE_BEGIN

Navigator and Temporal

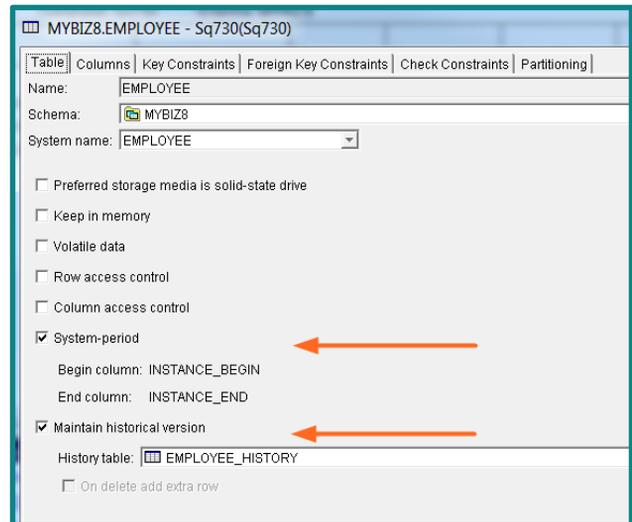
Table Definition... Establish System-period columns and declare the history table



Name	System Name	Partitioned	Owner	History Version	Temporal Type
EMPLOYEE	EMPLOYEE	No	SCOTT	Enabled	System-period
EMPLOYEE_HISTORY			COTT		History

EMPLOYEE_HISTORY context menu items:

- Edit Contents
- View Contents
- Data
- Definition** (indicated by orange arrow)
- Generate SQL...
- Index Advisor



MYBIZ8.EMPLOYEE - Sq730(Sq730)

Table Columns Key Constraints Foreign Key Constraints Check Constraints Partitioning

Name: EMPLOYEE

Schema: MYBIZ8

System name: EMPLOYEE

- Preferred storage media is solid-state drive
- Keep in memory
- Volatile data
- Row access control
- Column access control
- System-period
 - Begin column: INSTANCE_BEGIN (indicated by orange arrow)
 - End column: INSTANCE_END
- Maintain historical version
 - History table: EMPLOYEE_HISTORY (indicated by orange arrow)
 - On delete add extra row

Navigator and Temporal

Table Definition... History tables contain a reference to the system-period temporal table

Sq730: Tables Database: Sq730 Schema: MYBIZ8

Name	System Name	Partitioned	Owner	History Version	Temporal Type
EMPLOYEE	EMPLOYEE	No	SCOTT	Enabled	System-period
EMPLOYEE_HISTORY	EMP_HIST	Yes	SCOTT		History

Context menu for EMPLOYEE_HISTORY:

- Edit Contents
- View Contents
- Data
- Definition** (highlighted with an orange arrow)
- Generate SQL
- Index Advisor



MYBIZ8.EMPLOYEE_HISTORY - Sq730(Sq730)

Table Columns Check Constraints Partitioning

Name: EMPLOYEE_HISTORY

Schema: MYBIZ8

System name: EMP_HIST

- Preferred storage media is solid-state drive
- Keep in memory
- Volatile data
- Row access control
- Column access control

Related system-period temporal table: EMPLOYEE (highlighted with an orange arrow)

Text:

Navigator and Temporal

Visual Explain... shows the UNION ALL implementation and Temporal query controls

The screenshot displays the Visual Explain interface for a query. The query plan on the left shows a 'Final Select' node connected to a 'Union all' node, which in turn connects to a 'Table Scan' and a 'Table Probe' node. An 'Index Probe' node is also shown connected to the 'Table Probe'. An orange arrow points to the 'Union all' node. The right pane shows a list of attributes and their values. Two attributes, 'Current Temporal SYSTEM_TIME' and 'SYSTEMTIME bind option', are highlighted with an orange box.

Attribute	Value
Optimize for N Rows	30
Fetch First N Rows	All Rows
Commitment Control Level	WITH NC
Current Degree	Not Available
Session User	Not Available
System User	Not Available
Debug Mode	Not Available
Client Accounting Code	
Client User Identifier	FRANKOBA
Client Application Name	START SQL INTERACTIVE
Client Workstation Name	QINTER
Client Program Identifier	STRSQL
Current User	Not Available
Warm I/O Optimization Requested	Default
Warm I/O Optimization Used	Yes
Optimization Goal Used	Default
DECFLOAT Warnings	No
Allow AGP	Yes
Collate Errors	No
Field Procedure Comparison	Equal only
Allow Array Value Changes	No
Current Implicit XML Parse Option	Not Available
Current Temporal SYSTEM_TIME	NULL
SYSTEMTIME bind option	Yes
Concurrent Access Behavior	Default

select * from mybiz8.employee for system_time as of '2016-01-06-19.22.19.287442000245'

Temporal – FAQ

- **Users and applications are largely unaware that the history table exists**
 - SQL Query Engine unions in rows as needed
- **Consider using Range Partitioning for the History Table**
 - Organizing Historical rows by “Row End” is easy and has value
 - Value: Faster save times, partition avoidance, smart use of IN MEMORY and ON SSD
- **Performance**
 - Create radix indexes over “Row Begin” and “Row End” columns
- **Native I/O**
 - Native reads work against either the temporal or history table
 - Historical queries are unique to SQL
 - Generated columns are safe to add
 - Db2 for i ensures the correct values are used

Temporal – Catalogs

- **QSYS2/SYSTABLES**
Contains a column called TEMPORAL_TYPE.
 - 'S' the table is a system-period
 - 'H' the table is a history table
 - 'N' the table is neither temporal or history
- **QSYS2/SYSCOLUMNS**
The HAS_DEFAULT column indicates the type of generated column
- **QSYS2/SYSPERIODS**
Contains one row for each table with a system period and identifies temporal and versioning information
- **QSYS2/SYSHISTORYTABLES**
Contains one row for each history table

Temporal – Save and Restore

- The system-period temporal table and history table **must be explicitly saved**
- When a system-period temporal table is restored without its corresponding history table, the restored table's versioning relationship remains defined but is not established.

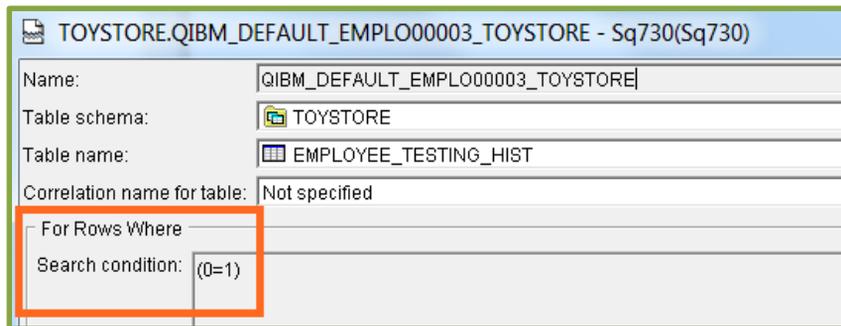
Defined state

will automatically change to versioned after both tables have been restored

- When in a defined state, the only operations that are allowed are:
 - ALTER TABLE ADD VERSIONING
 - ALTER TABLE DROP VERSIONING
 - DROP TABLE

Temporal – Row & Column Access Control

- When Row or Column Access Control (RCAC) is activated for a system-period temporal table, a default row permission is activated on the history table when versioning is added
- Time specification queries use the RCAC rule(s) of the temporal table
- The default row permission prevents any direct user access to the history table



TOYSTORE.QIBM_DEFAULT_EMPLO00003_TOYSTORE - Sq730(Sq730)

Name:	QIBM_DEFAULT_EMPLO00003_TOYSTORE
Table schema:	TOYSTORE
Table name:	EMPLOYEE_TESTING_HIST
Correlation name for table:	Not specified
For Rows Where	
Search condition:	(0=1)

Temporal – Storage Strategy

1. Analyze the volume of UPDATEs and DELETEs
2. Decide whether to use ON DELETE ADD EXTRA ROW
3. Decide whether to add other generated columns
4. Understand the record length of the file
5. Decide how long historical rows need to remain online
6. Decide whether to partition the history table
7. Decide whether to use media or memory preferences
8. Review indexing strategy
9. Review the data model to identify dimension tables that should also be made temporal (repeat steps 1-8 for those tables)
10. Reflect on your DR and/or HA strategy
 - PowerHA → Business As Usual
 - Logical Replication → Talk to your HA provider

Assessing the rate of history

```
CREATE SCHEMA DBESTUDY;  
  
CREATE OR REPLACE TABLE DBESTUDY.HISTORY_DETAIL  
(TABLE_SCHEMA VARCHAR(128),  
 TABLE_NAME VARCHAR(128),  
 POINT_IN_TIME TIMESTAMP,  
 UPDATE_OPERATIONS BIGINT,  
 DELETE_OPERATIONS BIGINT) ON REPLACE DELETE ROWS;  
  
--  
-- execute this insert once per day  
--  
INSERT INTO DBESTUDY.HISTORY_DETAIL  
  SELECT 'TOYSTORE5', 'SALES', CURRENT_TIMESTAMP,  
         UPDATE_OPERATIONS, DELETE_OPERATIONS  
  FROM QSYS2.SYSTABLESTAT  
  WHERE TABLE_SCHEMA = 'TOYSTORE5' AND  
         TABLE_NAME   = 'SALES';
```

Db2 for iBM i Lab Services

- Facilitated workshops covering current state, requirements, future state, possible solutions, implementation best practices, and formulation of a strategic roadmap:

- RCAC

- Temporal Tables

- Customized consulting workshops
 - Advanced SQL and Data-centric Programming
 - SQL Performance Best Practices, Monitoring and Tuning
- Consulting on any Db2 for i topic
- For more information, contact mcaain@us.ibm.com



ithankyou

www.ibm.com/developerworks/ibmi/techupdates/db2

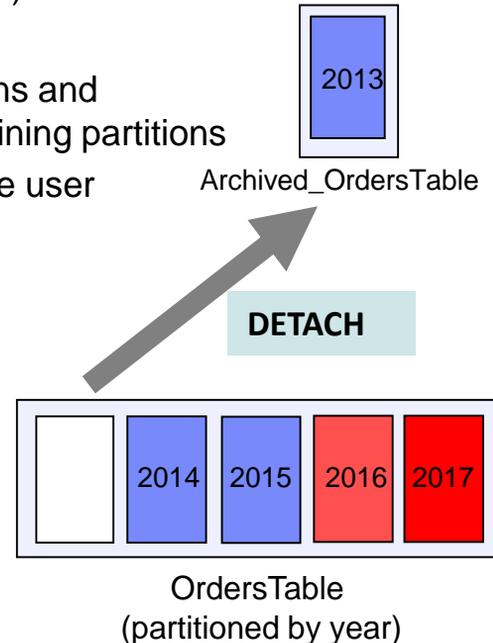
DETACH PARTITION – Dependent object rules

Dependent objects on the **source** table (OrdersTable)

- Views are rebuilt to use the remaining partitions
- DDS-created logical files that reference all partitions and Spanning SQL indexes are rebuilt to use the remaining partitions
- MQTs are retained, but need to be refreshed by the user

Usage details

- Cannot be a system-period temporal table
- Constraints are not added to the target table
- Privileges are not propagated to the target table
- When RCAC is active, a default row permission is activated on the target table
- An Identity column will not be an identity column in the target table



ATTACH PARTITION – Dependent object rules

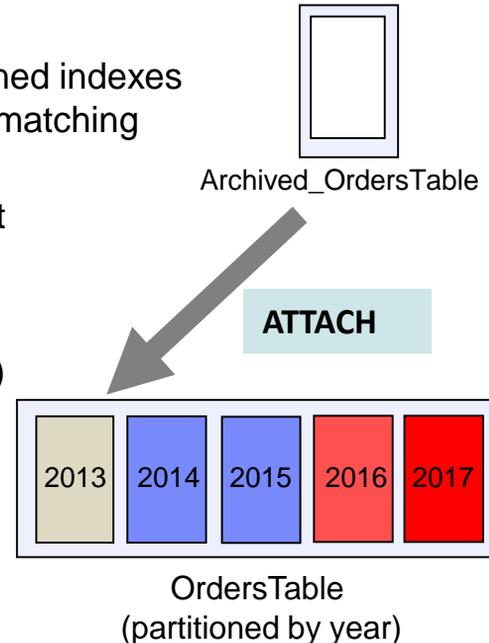
Dependent objects on the **source** table (Archived_OrdersTable)

- Views and MQTs are discarded
- Partitioned indexes which correspond with partitioned indexes on the target are retained, as long as they have a matching logical page size
- Active RCAC must match on the source and target

Usage details

Dependent objects on the **target** table (OrdersTable)

- Views are rebuilt to include the new partition
- Spanning indexes are rebuilt
- MQTs are retained, but need to be refreshed
- Partitioned indexes, with no corresponding partitioned index on the source are modified to accommodate for the new partition





Power Systems Social Media

IBM Power Systems Official Channels:



<https://facebook.com/IBMPowerSystems>



<https://twitter.com/IBMPowerSystems>



<https://www.linkedin.com/company/ibm-power-systems>



<http://www.youtube.com/c/ibmpowersystems>



<https://www.ibm.com/blogs/systems/topics/servers/power-systems/>



More to Follow:

Blogs	 Twitter	#Hashtags
<ul style="list-style-type: none"> • IBM Systems Magazine You and i (Steve Will) • IBM Systems Magazine i-Can (Dawn May) • IBM Systems Magazine: iDevelop (Jon Paris and Susan Gantner) • IBM Systems Magazine: iTalk with Tuohy • IBM Systems Magazine: Open your i (Jesse Gorzinski) • IBM Db2 for i (Mike Cain) • IBM DB2 Web Query for i (Doug Mack) 	<p> @IBMSystems @COMMONug @IBMChampions @IBMSystemsISVs @LinuxIBMMag @OpenPOWERorg @AIXMag @IBMiMag @ITJungleNews @SAPonIBMi @SiDforIBMi @IBMAIXeSupp @IBMAIXdoc @Forstie IBMi </p>	<p> #PowerSystems #IBMi #IBMAIX #POWER8 #LinuxonPower #OpenPOWER #HANAonPower #ITinfrastructure #OpenSource #HybridCloud #BigData </p>



Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.



Special notices (cont.)

IBM, the IBM logo, ibm.com AIX, AIX (logo), AIX 5L, AIX 6 (logo), AS/400, BladeCenter, Blue Gene, ClusterProven, DB2, ESCON, i5/OS, i5/OS (logo), IBM Business Partner (logo), IntelliStation, LoadLeveler, Lotus, Lotus Notes, Notes, Operating System/400, OS/400, PartnerLink, PartnerWorld, PowerPC, pSeries, Rational, RISC System/6000, RS/6000, THINK, Tivoli, Tivoli (logo), Tivoli Management Environment, WebSphere, xSeries, z/OS, zSeries, Active Memory, Balanced Warehouse, CacheFlow, Cool Blue, IBM Systems Director VMControl, pureScale, TurboCore, Chiphopper, Chiphopper, Cloudscape, DB2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Parallel File System, , GPFS, HACMP, HACMP/6000, HASM, IBM Systems Director Active Energy Manager, iSeries, Micro-Partitioning, POWER, PowerExecutive, PowerVM, PowerVM (logo), PowerHA, Power Architecture, Power Everywhere, Power Family, POWER Hypervisor, Power Systems, Power Systems (logo), Power Systems Software, Power Systems Software (logo), POWER2, POWER3, POWER4, POWER4+, POWER5+, POWER6, POWER6+, POWER7, System i, System p, System p5, System Storage, System z, TME 10, Workload Partitions Manager and X-Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A full list of U.S. trademarks owned by IBM may be found at: <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Altivec is a trademark of Freescale Semiconductor, Inc.

AMD Opteron is a trademark of Advanced Micro Devices, Inc.

InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries or both.

NetBench is a registered trademark of Ziff Davis Media in the United States, other countries or both.

SPECint, SPECfp, SPECjbb, SPECweb, SPECjAppServer, SPEC OMP, SPECviewperf, SPECcap, SPECchpc, SPECjvm, SPECmail, SPECimap and SPECsfs are trademarks of the Standard Performance Evaluation Corp (SPEC).

The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).

UNIX is a registered trademark of The Open Group in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.