


I Didn't Know You Could Do that with SQL! *The Easy Stuff*

Skip Marchesani
Custom Systems Corp
Sparta, NJ 07871

973-579-1340
smarches@warwick.net



Copyright Custom Systems Corp 2005

I Didn't Know You Could Do that with SQL: The Easy Stuff

Disclaimer:

This presentation may contain examples of code and names of companies or persons. The code is given for presentation purposes and has not been tested by IBM and/or Custom Systems Corp. Therefore IBM and/or Custom Systems Corp does not guarantee the reliability, serviceability, or function of the code and the code is provided "AS IS". IBM AND/OR CUSTOM SYSTEMS CORP EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND MERCHANTABILITY. Any names appearing in this presentation are designed to be fictitious and IBM and Custom Systems Corp makes no representations as to the accuracy of the names or data presented in accordance therewith.

Reproduction:

This presentation is the property of Skip Marchesani and Custom Systems Corporation. Permission is granted to make a limited number of copies of this material for non-commercial purposes, providing this page is included with all copies. Express written permission is required for making copies for other purposes.

iSeries, AS/400, OS/400, DB2/400, DB2 UDB are registered trademarks of the IBM Corporation

Copyright Custom Systems Corp 2005

Overview

An Important Goal for this Presentation

- An important goal for this presentation is to show you that **SQL is just another programming language**. The more you work and play with it, the more you realize the power of SQL and what it can do for you as an application development or database manipulation tool. With a little thought and creativity you will find you can use SQL for things that at first glance you did not think possible.
- This presentation is based on the SQL function available in V5R2 of OS/400, assumes you have a basic understanding of relational database concepts and SQL, and that you are familiar with using the **SELECT, INSERT, UPDATE, and DELETE** statements.



Copyright Custom Systems Corp 2005

Overview...

iSeries 400 Experts Journal

- This presentation is based in large part on a series articles that were published in *iSeries 400 Experts Journal* from September 2002 thru March 2004
- ***iSeries 400 Experts Journal***
 - Published by iSeries 400 Experts (formerly the 400 Group)
 - www.iseries400experts.com
- **To subscribe and access these articles on the web contact:**
Sarah J. Kimmel, Editor
781-320-8909 ext 210
sarahk@iseries400experts.com



Copyright Custom Systems Corp 2005

Overview...

I Didn't Know You Could Do that with SQL!

- **COUNT, SUM, and AVG** for data analysis
- **DEC** to format numeric columns
- **AS** to name a derived column
- **SUBSTR** and **CONCAT** for character columns
- **CAST** and **DIGITS** to change data type
- **Summarizing Data with SELECT**
- **Other Interesting Stuff**
- **Summary**
- **V5R2 SQL Information Sources**



Copyright Custom Systems Corp 2005

Sample Tables Used in Examples

- **Employee Table - EMP**

NBR	NAM	CLS	SEX	DPT	SAL
10	Ed	5	M	911	7,000
20	Heikki	2	M	901	6,000
30	John	5	M	977	3,200
40	Mike	4	M	977	6,500
50	Marcela	3	F	911	7,500
60	Frank	2	M	990	6,500

- **Department Table - DEP**

DPT	DNM
901	Accounts
977	Manufact
911	Sales
990	Spares



Copyright Custom Systems Corp 2005

SQL Syntax Used in Examples

- **All UPPER CASE - SQL required syntax**
- **All lower case - SQL parameter data supplied by end user**
- **One or more characters enclosed in single quotes ('S') is a literal used for comparison purposes and is supplied by the end user.**
- **Anything enclosed in [] is optional SQL syntax or optional parameter data**



Copyright Custom Systems Corp 2005

COUNT, SUM, and AVG for Data Analysis



Copyright Custom Systems Corp 2005

Counting with SQL

Types of Counting with SQL

- **Total number of rows in a table**
 - How many rows are in the table
- **Subset of rows in a table**
 - Based on selection criterion in WHERE clause
- **Distinct or unique occurrences of rows**
 - How many different (distinct or unique) values exist
- **Groups of rows**
 - Summarization of data using GROUP BY and HAVING clauses



Copyright Custom Systems Corp 2005

COUNT

Two Types of COUNT

- **Column Function**
- **COUNT**
 - Result set is a large integer with precision (length) of 10
 - Max result set size is 10 digits with a limit of 2,147,483,647
- **COUNT BIG**
 - Max result set size is 32 digits
 - 9,999,999,999,999,999,999,999,999,999,999,999
 - aka COUNT REALLY REALLY BIG



Copyright Custom Systems Corp 2005

COUNT...

COUNT Syntax

- **COUNT(*)**
 - Includes rows with null values in the count
- **COUNT(column_name) or COUNT(ALL column_name)**
 - Omits rows with null values from the count
- **COUNT(DISTINCT column_name)**
 - Omits rows with duplicate or null values from the count

Note: If database tables are not defined with null capable columns, COUNT(*) and COUNT(column_name) or COUNT(ALL column_name) will return the same result set



Copyright Custom Systems Corp 2005

COUNT...

COUNT Examples

SELECT COUNT(*) FROM emp

COUNT(*) 6

**SELECT COUNT(*) FROM emp
WHERE dpt = 977**

COUNT(*) 2



Copyright Custom Systems Corp 2005

COUNT...

COUNT Examples...

SELECT COUNT(DISTINCT class) FROM emp

COUNT(*)
4

**SELECT dpt, COUNT(*) FROM emp
GROUP BY dpt
ORDER BY dpt**

DPT	COUNT(*)
901	1
911	2
977	2
990	1



Copyright Custom Systems Corp 2005

Summing or Adding with SQL

SUM Column Function

- **Result set is the sum of a set of numbers with 1 to many numbers in the set**
 - Precision (length) is 31 with a scale (number of decimal positions) the same as the column being summed
- **Data type of column must be numeric**
 - Decimal - packed
 - Numeric - zoned
 - Smallint
 - Integer
 - Bigint
 - Float
 - Real
 - Double



Copyright Custom Systems Corp 2005

SUM

SUM Syntax

- **SUM(column_name) or SUM(ALL column_name)**
 - Sum the value found in the column for all rows selected
- **SUM(DISTINCT column_name)**
 - Duplicate values are not included in the sum



Copyright Custom Systems Corp 2005

SUM...

SUM Examples

SELECT SUM(sal) FROM EMP

SUM(sal) 36,700

**SELECT SUM(sal) FROM emp
WHERE dpt = 911**

SUM(sal) 14,500



Copyright Custom Systems Corp 2005

Averaging Data with SQL

AVG Column Function

- **Result set is the average of a set of numbers with 1 to many numbers in the set**
 - Precision (length) is 31 with a scale (number of decimal positions) equal to 31 minus defined column length
 - Translation: Lots of decimal positions
- **Data type of column must be numeric**
 - Decimal - packed
 - Numeric - zoned
 - Smallint
 - Integer
 - Bigint
 - Float
 - Real
 - Double

Copyright Custom Systems Corp 2005



AVG

AVG Syntax

- **AVG(column_name) or AVG(ALL column_name)**
 - Sum the value found in the column for all rows selected
- **AVG(DISTINCT column_name)**
 - Duplicate values are not included in the sum

Copyright Custom Systems Corp 2005



DEC to Format Numeric Results

DEC or DECIMAL Scalar Column Function

- **Result set is the representation of a number in decimal format (1234.56)**
 - Precision (length) and scale (number of decimal positions) are defined as part of the DEC function
- **Data type of column must be numeric**
 - Decimal - packed
 - Numeric - zoned
 - Smallint
 - Integer
 - Bigint
 - Float
 - Real
 - Double



Copyright Custom Systems Corp 2005

DEC

DEC or DECIMAL Syntax

- **DEC(expression, precision, scale, ['decimal-character'])**



Copyright Custom Systems Corp 2005

What's the Difference?

Column Headings Reflect the Function

```
SELECT AVG(sal) FROM EMP
```

AVG(sal) 6,116.6666666666666666666666666666
--

```
SELECT AVG(sal) FROM emp  
WHERE dpt = 911
```

AVG(sal) 7,250.0000000000000000000000000000
--

- Easy to fix



Copyright Custom Systems Corp 2005

AS to Name a Derived Column



Copyright Custom Systems Corp 2005

AS to Name a Derived Column

AS Clause

- **Used to**
 - Rename an existing column
 - Name a derived column
- **Considerations**
 - Name cannot be qualified
 - Name does not have to be unique



Copyright Custom Systems Corp 2005

AS

AS Syntax

- **AS column_name**



Copyright Custom Systems Corp 2005

AS...

DEC or DECIMAL Examples

```
SELECT DEC(AVG(sal),7,2) FROM EMP
```

DEC 6,116.67

```
SELECT DECIMAL(AVG(sal),7,2) FROM emp  
WHERE dpt = 911
```

DEC 7,250.00



Copyright Custom Systems Corp 2005

AS...

AS Examples

```
SELECT DEC(AVG(sal),7,2) AS avgsal  
FROM EMP
```

AVGSAL 6,116.67

```
SELECT DECIMAL(AVG(sal),7,2) AS avg911sal  
FROM emp  
WHERE dpt = 911
```

AVG911SAL 7,250.00



Copyright Custom Systems Corp 2005

SUBSTR and CONCAT for Character Columns



Copyright Custom Systems Corp 2005

SUBSTR for a Character Column

SUBSTR or SUBSTRING Scalar Function

- **Result is a subset or substring of a character column**
 - Start position and length of resulting string is defined in the function
- **Data type must be character**
 - Fixed or variable length - including large object (BLOB)
 - Single or double byte



Copyright Custom Systems Corp 2005

SUBSTR...

SUBSTR or SUBSTRING Syntax

- **SUBSTR(string_expression, start_position, length)**
- **SUBSTRING(string_expression FROM start_position FOR length)**



Copyright Custom Systems Corp 2005

SUBSTR...

SUBSTR or SUBSTRING Examples

```
SELECT dpt, SUBSTR(dnm,1,4) AS sname  
FROM dep
```

DPT	SNAME
901	Acco
977	Manu
911	Sale
990	Spar

DPT	DNM
901	Accounts
977	Manufact
911	Sales
990	Spares



Copyright Custom Systems Corp 2005

CONCAT for a Character Column

CONCAT Scalar Function

- **Combines or concatenates two string expressions together**
- **Result is a single string consisting of the first string expression, followed by the second string expression**
- **Data type must be character**
 - Fixed or variable length - including large object (BLOB)
 - Single or double byte



Copyright Custom Systems Corp 2005

CONCAT...

CONCAT Syntax

- **CONCAT(string_expression_01, string_expression_02)**



Copyright Custom Systems Corp 2005

CONCAT...

CONCAT Examples - Simple CONCAT

```
SELECT nbr, nam,
       CONCAT(sex, nam) AS sexnam
FROM emp
```

NBR	NAM	SEXNAM
10	Ed	MEd
20	Heikki	MHeikki
30	John	MJohn
40	Mike	MMike
50	Marcela	FMarcela
60	Frank	MFrank



Copyright Custom Systems Corp 2005

CONCAT...

CONCAT Examples - CONCAT with SUBSTR

```
SELECT dpt,
       CONCAT(SUBSTR(dnm,1,4), SUBSTR(dnm,1,4))
       AS double
FROM dep
```

DPT	DOUBLE
901	AccoAcco
977	ManuManu
911	SaleSale
990	SparSpar



Copyright Custom Systems Corp 2005

CONCAT...

CONCAT Examples - CONCAT with a Literal

```
SELECT dpt,  
       CONCAT(SUBSTR(dnm,1,4), 'Stuff')  
       AS stuff  
FROM dep
```

DPT	STUFF
901	AccoStuff
977	ManuStuff
911	SaleStuff
990	SparStuff



Copyright Custom Systems Corp 2005

CONCAT...

CONCAT Examples - Concat with 3 Strings

```
SELECT dpt,  
       CONCAT(CONCAT(SUBSTR(dnm,1,4), '-'), 'Stuff')  
       AS hyphen  
FROM dep
```

DPT	HYPHEN
901	Acco-Stuff
977	Manu-Stuff
911	Sale-Stuff
990	Spar-Stuff



Copyright Custom Systems Corp 2005

SUBSTR & CONCAT

SUBSTR & CONCAT with Numeric Columns

- **Can be used with numeric columns but data type must be changed from numeric to character**



Copyright Custom Systems Corp 2005

CAST and DIGITS to Change Data Type



Copyright Custom Systems Corp 2005

CAST to Change Data Type

CAST Operation

- **Converts the existing data type of the cast expression to the specified data type**
 - Numeric to character
 - Character to numeric
 - Numeric to numeric
 - Character to character
 - See SQL Reference for supported CAST from/to data types
- **When converting numeric to character**
 - Leading zeros are truncated
 - Decimal point ignored and digits to the right are part of result string
 - Digits are left justified

Copyright Custom Systems Corp 2005

DIGITS to Change Data Type

DIGITS Scalar Function

- **Only converts numeric to character data type**
- **When converting numeric to character**
 - Leading zeros are NOT truncated
 - Decimal point ignored and digits to the right are part of result string
 - Digits are left justified



Copyright Custom Systems Corp 2005

CAST and DIGITS

CAST Syntax

- **CAST(expression AS data_type)**

DIGITS Syntax

- **DIGITS(numeric_expression)**



Copyright Custom Systems Corp 2005

CAST and DIGITS...

CAST and DIGITS Example

```
SELECT nbr, nam,  
       SUBSTR(CAST(dpt AS CHAR(3)),1,1) AS chardpt  
FROM emp
```

NBR	NAM	CHARDPT
10	Ed	9
20	Heikki	9
30	John	9
40	Mike	9
50	Marcela	9
60	Frank	9

```
SELECT nbr, nam,  
       SUBSTR(DIGITS(dpt),1,1) AS chardpt  
FROM emp
```



Copyright Custom Systems Corp 2005

Summarizing Data with SELECT



Copyright Custom Systems Corp 2005

Summarizing Data with SELECT

The **SELECT** statement is an *Either Or* Proposition

- ***Either*** return detail rows in the result set
- ***Or*** return summarized data in the result set
- **A single SQL statement has no capability to do**
 - Detail
 - Detail
 - Level break
 - Detail
 - Level break
- **Query Manager CAN do above**
 - SQL based query product



Copyright Custom Systems Corp 2005

Summarizing Data with SELECT...

SELECT Statement Clauses

SELECT (columns, *, or expressions)
FROM(tables or views)
WHERE(row selection criteria)
GROUP BY. . . (row summarization criteria)
HAVING. . . . (GROUP BY selection criteria)
ORDER BY. . . (column ordering criteria)



Copyright Custom Systems Corp 2005

Summarizing Data with SELECT...

GROUP BY Clause

- **Defines grouping or summary criteria for SELECT**
- **Format**

```
SELECT FLDA, FLDE, FUNCTION(FLDG), FUNCTION(FLDH)
FROM TABLE-NAME
WHERE SELECTION-CRITERIA
GROUP BY FLDA, FLDE
ORDER BY FLDA, FLDE
```

- **If a column referenced in the column list of the SELECT statement is not operated on by a function, that column must be referenced as part of the grouping criteria in the GROUP BY clause**

Copyright Custom Systems Corp 2005

Summarizing Data with SELECT...

GROUP BY Clause...

- For each department list the total salary and total number of employees

```
SELECT DPT, SUM(SAL), COUNT (*)  
FROM EMP  
GROUP BY DPT  
ORDER BY DPT
```

DPT	SUM(SAL)	COUNT(*)
901	6,000	1
911	14,500	2
977	9,700	2
990	6,500	1



Copyright Custom Systems Corp 2005

Summarizing Data with SELECT...

HAVING Clause...

- Defines **GROUP BY** selection criteria
- **Format**

```
SELECT FLDA, FLDE, FUNCTION(FLDG), FUNCTION(FLDH)  
FROM TABLE-NAME  
WHERE SELECTION-CRITERIA  
GROUP BY FLDA, FLDE  
HAVING GROUP-BY-SELECTION-CRITERIA  
ORDER BY FLDA, FLDE
```

Copyright Custom Systems Corp 2005

Summarizing Data with SELECT...

HAVING Clause...

- **For those departments that have more than one employee, list the total salary and total number of employees**

```
SELECT DPT, SUM(SAL) AS SALTOT,  
        COUNT(*) AS EMPTOT  
FROM EMP  
GROUP BY DPT  
HAVING COUNT(*) > 1  
ORDER BY DPT
```

DPT	SALTOT	EMPTOT
911	14,500	2
977	9,700	2



Copyright Custom Systems Corp 2005

Other Interesting Stuff



Copyright Custom Systems Corp 2005

Other Interesting Stuff

Some Other Interesting SQL Scalar Functions

- **RRN**
 - Returns the relative record number of a row in a table
- **HEX**
 - Returns the hexadecimal representation of a value
- **RTRIM (Right Trim)**
 - Removes blanks or hexadecimal zeros from the end, or the trailing or right side, of a string expression
- **LENGTH**
 - Returns the length of a number or character value



Copyright Custom Systems Corp 2005

Other Interesting Stuff...

New V5R3 Scalar Functions

- **INSERT**
 - Inserts a number or character string into an existing string and optionally overlays specific positions in the existing string with the insert string
- **REPLACE**
 - Replaces all occurrences of a search string in a number or character string with a replacement string



Copyright Custom Systems Corp 2005

RRN to Find Relative Record Number

RRN Scalar Function

- **Result set is the relative record number of a row**
 - Length is 15 with zero decimal positions

RRN Syntax

- **RRN(table_name)**



Copyright Custom Systems Corp 2005

RRN

RRN Example

```
SELECT nbr, nam, RRN(ax) AS rec  
FROM emp ax ORDER BY nbr
```

NBR	NAM	REC
10	Ed	2
20	Heikki	1
30	John	5
40	Mike	4
50	Marcela	3
60	Frank	6



Copyright Custom Systems Corp 2005

HEX to Find Hexadecimal Value

HEX Scalar Function

- **Result set is the hexadecimal representation of a value**
 - Length is twice the defined length with a max length of approx 32K

HEX Syntax

- **HEX(expression)**



Copyright Custom Systems Corp 2005

HEX

HEX Example

```
SELECT nbr, nam, HEX(nam) AS hex_nam  
FROM emp ORDER BY nbr
```

NBR	NAM	HEX_NAM
10	Ed	C58440404040404040
20	Heikki	C88589929289404040
30	John	D19688954040404040
40	Mike	D48992854040404040
50	Marcela	D48199838593814040
60	Frank	C69981959240404040



Copyright Custom Systems Corp 2005

RTRIM to Remove Trailing Blanks

RTRIM Scalar Function

- **Result set is the RTRIM expression with trailing blanks or hexadecimal zeros removed**
 - Length of the result set is the length of the expression minus the number of blanks or hex zeros removed

RTRIM Syntax

- **RTRIM(expression)**



Copyright Custom Systems Corp 2005

RTRIM

RTRIM Example

```
SELECT nbr, nam, TRIM(nam) AS tnam
FROM emp ORDER BY nbr
```

NBR	NAM	TNAM
10	Ed	Ed
20	Heikki	Heikki
30	John	John
40	Mike	Mike
50	Marcela	Marcela
60	Frank	Frank

- **What's the difference between NAM and TNAM?**



Copyright Custom Systems Corp 2005

LENGTH to Determine the Length of a Value

LENGTH Scalar Function

- **Result set is the length of the string in the expression including any blanks or zeros**
 - Result set length is 10 digits with zero decimals

LENGTH Syntax

- **LENGTH(expression)**



Copyright Custom Systems Corp 2005

LENGTH

LENGTH Example

```
SELECT nbr, nam, LENGTH(nam) AS len1  
FROM emp ORDER BY nbr
```

NBR	NAM	LEN1
10	Ed	10
20	Heikki	10
30	John	10
40	Mike	10
50	Marcela	10
60	Frank	10

- **Why is the length for each NAM string = 10?**



Copyright Custom Systems Corp 2005

Combining RTRIM and LENGTH

How Many Non Blank Characters are in Each Name?

```
SELECT nbr, nam, LENGTH(RTRIM(nam)) AS len2
FROM emp ORDER BY nbr
```

NBR	NAM	LEN2
10	Ed	2
20	Heikki	6
30	John	4
40	Mike	4
50	Marcela	7
60	Frank	5



Copyright Custom Systems Corp 2005

INSERT to Overlay Positions In a String

INSERT Scalar Function - New for V5R3

- **Result set is the source string with the insert string placed into the source string at the specified start position, and optionally specified positions in the source string overlaid with the insert string**
 - Result set length is length of the source string plus length of the insert string minus the number of positions overlaid (if any)
 - Result set length cannot exceed the maximum length of the data type
 - Data type of the source string and insert string must be compatible

INSERT Syntax

- **INSERT(source_string, start, length, insert_string)**

Copyright Custom Systems Corp 2005

INSERT

INSERT Example with No Overlay

```
SELECT nbr, nam, INSERT(nam, 3, 0, 'xy') AS insrt1
FROM emp ORDER BY nbr
```

NBR	NAM	INSRT1
10	Ed	Edxy
20	Heikki	Hexyikki
30	John	Joxyhn
40	Mike	Mixyke
50	Marcela	Maxyrcela
60	Frank	Frxyank



Copyright Custom Systems Corp 2005

INSERT...

INSERT Example Overlaying Positions 3 and 4

```
SELECT nbr, nam, INSERT(nam, 3, 2, 'xy') AS insrt2
FROM emp ORDER BY nbr
```

NBR	NAM	INSRT2
10	Ed	Edxy
20	Heikki	Hexyki
30	John	Joxy
40	Mike	Mixy
50	Marcela	Maxyela
60	Frank	Frxyk

- Easy way to update part of a character string!



Copyright Custom Systems Corp 2005

REPLACE - Based on a Search String

REPLACE Scalar Function - New for V5R3

- **Result set is the source string with all occurrences of the search string replaced with the replace string**
 - Result set length is length of the source string plus length of the replace string minus the length of the search string
 - Result set length cannot exceed the maximum length of the data type
 - Data type of the source string and replace string must be compatible
 - If no match with the search string, the source string is returned unchanged as the result set

REPLACE Syntax

- **INSERT(source_string, search_string, replace_string)**

Copyright Custom Systems Corp 2005

REPLACE

REPLACE Example

```
SELECT nbr, nam, REPLACE(nam, 'ik', 'qqq') AS replc1
FROM emp ORDER BY nbr
```

NBR	NAM	REPLC1
10	Ed	Ed
20	Heikki	Heqqqki
30	John	Joxyhn
40	Mike	Miqqqe
50	Marcela	Macela
60	Frank	Frank

- **Easy way to update part of a character string!**



Copyright Custom Systems Corp 2005

Summary

I Didn't Know You Could Do that with SQL!

- **COUNT, SUM, and AVG** for data analysis
- **DEC** to format numeric columns
- **AS** to name a derived column
- **SUBSTR** and **CONCAT** for character columns
- **CAST** and **DIGITS** to change data tyoe
- **Summarizing Data with SELECT**
- **HEX** and **RRN**
- **RTRIM** and **LENGTH**
- **INSERT** amd **REPLACE** - New for V5R3



Copyright Custom Systems Corp 2005

Summary...

I Didn't Know You Could Do that with SQL...

- **From this presentation you should have a better understanding of SQL as a programming language.**
- **The more you work and play with it, the more you realize the power of SQL and what it can do for you as an application development or database manipulation tool.**
- **With a little thought and creativity you will find you can use SQL for things that at first glance you did not think possible.**



Copyright Custom Systems Corp 2005

V5R2 SQL Information Sources

- **iSeries Information Center Publications - Web or CD**

- SQL Reference
- SQL Programming Concepts
- SQL Programming with Host Languages
- Query Manager Use
- SQL Messages and Codes

- **To access Info Center on the Web**

- <http://publib.boulder.ibm.com/series/v5r2/ic2924/index.htm>
 - In left scroll bar
 - Click on Database - 3rd line from top
 - Click on Manuals
 - Use right scroll bar to scroll down to above SQL publication

- **DB2 UDB for iSeries on the Web**

- <http://www.ibm.com/servers/eserver/series/db2/>

- **iSeries 400 Experts Journal**

- <http://www.iseries400experts.com>

Copyright Custom Systems Corp 2005