PROG8080
Aggregate Functions in SQL Server

Glenn Paulley

Fall 2015

Aggregate Functions

- Aggregate functions are used to count rows and summarize (typically) numeric data in a database
- Aggregate functions can appear in:
    - the column list of a SELECT statement
    - A HAVING clause (later in this unit)
    - An ORDER BY clause
- Aggregate functions differ from scalar functions
    - Scalar functions (e.g. ABS or SUBSTRING): one value in, one value returned
    - Aggregate functions: many values in, one value returned

The five basic aggregate functions

- Basic aggregate functions:
  - COUNT
  - AVG
  - SUM
  - MIN
  - MAX
- Each supports the ALL or DISTINCT keyword
  - ALL – compute the function over all non-NULL input values
  - DISTINCT – compute the result considering only unique non-NULL values
    - COUNT(DISTINCT X)
    - AVG(ALL X)

The five basic aggregate functions

- COUNT always returns a value
  - COUNT(*) will return 0 (zero) if the input is empty
- All aggregate functions except COUNT(*) ignore NULL values in their input
  - For example, AVG(X) considers only non-null values of X
  - Aggregation functions other than COUNT will return NULL if they have all NULL values in their input

COUNT()

- COUNT(*) returns a count of the rows in a table
  - Result not related to nullability of any expression – merely a count of the rows in the input
- COUNT() <u>always</u> returns a value
  - If the input is empty, the result of COUNT(*) is 0 (zero)
- COUNT(expression)
  - Counts the number of non-null values in the input
- COUNT(DISTINCT expression)
  - Counts the number of distinct (duplicate-free) values

COUNT()

- COUNT(*) counts all rows unconditionally

```
SELECT COUNT( * ) AS [COUNT( * )]
               FROM Person
```

```
COUNT( * )
-----------
434
```

COUNT()

- COUNT(expression) counts all rows with non-null values of expression

```
SELECT COUNT( fax) AS [COUNT( fax )]
                FROM Employee
```

```
COUNT( fax )
---------------
4
```

COUNT()

- COUNT(ALL column) counts all rows with non-null values for column; ALL is the default and is usually omitted

```
SELECT COUNT( ALL fax ) AS [COUNT( ALL fax )]
               FROM Employee
```

```
COUNT( ALL fax )
-------------------
4
```

COUNT()

- COUNT(DISTINCT *expression*) counts all rows with distinct, non-null values of *expression*

```
SELECT COUNT( DISTINCT fax ) AS
                    [COUNT( DISTINCT fax )]
          FROM Employee
```

```
COUNT( DISTINCT fax )
-------------------------
3
```

COUNT()

- Naturally, you can use a WHERE clause to restrict the rows being counted

```
SELECT COUNT( DISTINCT fax ) AS
        [COUNT( DISTINCT fax ) with WHERE]
           FROM Employee
           WHERE schoolCode = 'TAP' ;
```

```
COUNT( DISTINCT fax ) with WHERE
-----------------------------------
1
```

AVG() – Compute the average

- AVG() is used to compute the average value of an expression

```
SELECT '$' + CONVERT( CHAR(7),
          CAST( AVG(amount) AS money ), 1 )
          AS 'Average amount'
         FROM SIS.dbo.InvoiceItem
```

```
Average amount
-------------
$  457.94
```

MIN()

- The MIN() aggregate function is used to find the minimum value from a set of values

```
SELECT '$' + CONVERT( CHAR(7),
             CAST( MIN(amount) AS money ), 1 )
             AS [Minimum Price]
            FROM SIS.dbo.InvoiceItem
```

```
Minimum Price
-------------
$  9.00
```

MAX()

- Similar to MIN(), MAX() is used to find the maximum value

```
SELECT '$' + CONVERT( CHAR(9),
           CAST( MAX(amount) AS money ), 1 )
           AS [Maximum Price]
          FROM SIS.dbo.InvoiceItem
```

```
Maximum Price
-------------
$ 3,380.00
```

SUM()

- SUM(*expression*) is used to compute the sum of all of the non-null values of *expression*

```
SELECT '$' + CONVERT( CHAR(10),
             CAST( SUM(amount) AS money ), 1 )
             AS [Sum of all items]
FROM SIS.dbo.InvoiceItem
```

```
Sum of all items
-----------------
$  5,037.32
```

AVG/MIN/MAX/SUM and WHERE

- You can use a WHERE clause to restrict the input rows to the computation of AVG(), MIN(), MAX(), and SUM() as well

```
SELECT '$' + CONVERT( CHAR(10),
            CAST( AVG(amount) AS money ), 1 )
            AS [Avg of all items]
FROM SIS.dbo.InvoiceItem
WHERE item LIKE '%CSI%‘
```

```
Avg of all items
-----------------
$  210.63
```