

Recordset: Obtaining SUMs and Other Aggregate Results (ODBC)

Article • 08/02/2021

ⓘ Note

The MFC ODBC Consumer wizard is not available in Visual Studio 2019 and later. You can still create a consumer manually.

This topic applies to the MFC ODBC classes.

This topic explains how to obtain aggregate results using the following [SQL](#) keywords:

- **SUM** Calculates the total of the values in a column with a numeric data type.
- **MIN** Extracts the smallest value in a column with a numeric data type.
- **MAX** Extracts the largest value in a column with a numeric data type.
- **AVG** Calculates an average value of all the values in a column with a numeric data type.
- **COUNT** Counts the number of records in a column of any data type.

You use these SQL functions to obtain statistical information about the records in a data source rather than to extract records from the data source. The recordset that is created usually consists of a single record (if all columns are aggregates) that contains a value. (There might be more than one record if you used a **GROUP BY** clause.) This value is the result of the calculation or extraction performed by the SQL function.

💡 Tip

To add a SQL **GROUP BY** clause (and possibly a **HAVING** clause) to your SQL statement, append it to the end of `m_strFilter`. For example:

```
m_strFilter = "sales > 10 GROUP BY SALESPERSON_ID";
```

You can limit the number of records you use to obtain aggregate results by filtering and sorting the columns.

⊗ Caution

Some aggregation operators return a different data type from the columns over which they are aggregating.

- **SUM** and **AVG** might return the next larger data type (for example, calling with `int` returns `LONG` or `double`).
- **COUNT** usually returns `LONG` regardless of target column type.
- **MAX** and **MIN** return the same data type as the columns they calculate.

For example, the **Add Class** wizard creates `long m_lSales` to accommodate a Sales column, but you need to replace this with a `double m_dblSumSales` data member to accommodate the aggregate result. See the following example.

To obtain an aggregate result for a recordset

1. Create a recordset as described in [Adding an MFC ODBC Consumer](#) containing the columns from which you want to obtain aggregate results.
2. Modify the [DoFieldExchange](#) function for the recordset. Replace the string representing the column name (the second argument of the [RFX](#) function calls) with a string representing the aggregation function on the column. For example, replace:

```
RFX_Long(pFX, "Sales", m_lSales);
```

with:

```
RFX_Double(pFX, "Sum(Sales)", m_dblSumSales)
```

3. Open the recordset. The result of the aggregation operation is left in `m_dblSumSales`.

Note

The wizard actually assigns data member names without Hungarian prefixes. For example, the wizard would produce `m_Sales` for a Sales column, rather than the `m_lSales` name used earlier for illustration.

If you are using a [CRecordView](#) class to view the data, you have to change the DDX function call to display the new data member value; in this case, changing it from:

```
DDX_FieldText(pDX, IDC_SUMSALES, m_pSet->m_lSales, m_pSet);
```

To:

```
DDX_FieldText(pDX, IDC_SUMSALES, m_pSet->m_dblSumSales, m_pSet);
```

See also

[Recordset \(ODBC\)](#)

[Recordset: How Recordsets Select Records \(ODBC\)](#)