WebNameSQL

Alternative SQL Statement

Command List and Syntax

Jan 2022 @ Hong Kong

Introduction

WebNameSQL is a C# in-memory databending software that supports accountants using a web browser to interactive with accounting rules and tables for databending. However, this project became obsolete and it is replaced by a new project “Peaks DataFrame” to solve issues arising from real-time processing and big data. During a continuing effort in academic research, it is implemented new algorithms by using Golang which resulted in a performance gain of around 5X ~ 10X. Please refer to https://github.com/hkpeaks/peaks-framework

Commands to be re-implemented in the Peaks DataFrame will not be the same as those below. Considering there are too many commands for your learning and practice, further consolidation and improvement is necessary. The use cases are no longer restricted to accounting; for example, some use cases will cover bioinformatics.

WebNameSQL Command by Unit

Unit 1: Copy Table

- ✧ CopyTable
- ✧ CopyTable.CommonTable
- ✧ CSV2Web
- ✧ CSV2Web.CommonTable
- ✧ DataTable2Web
- ✧ Web2CSV
- ✧ Web2DataTable
- ✧ Web2JSON
- ✧ Web2HTML
- ✧ Web2XML
- ✧ Web2OneColumn
- ✧ ManyCSV2Web
- ✧ OneColumn2Web

Unit 2: Add Column

- ✧ ComputeColumn
- ✧ ConditionalJoin
- ✧ FullJoin
- ✧ InnerJoin
- ✧ JoinTable
- ✧ Number2Text
- ✧ ResidualJoin

Unit 3: Data Filter and Selection

- ✧ AndFilter
- ✧ AndFilter.ConditionTable
- ✧ AndFilter.DistinctTable
- ✧ OrFilter
- ✧ OrFilter.ConditionList
- ✧ OrFilter.DistinctList
- ✧ RemoveColumn
- ✧ SelectColumn

Unit 4: Presentation of Data

- ✧ Crosstab
- ✧ Distinct
- ✧ GroupBy
- ✧ OrderBy
- ✧ ReverseCrosstabCSV2Web
- ✧ ReverseManyCrosstabCSV2Web

Unit 5: Amendment

- ✧ AmendColumnName
- ✧ AmendDate
- ✧ ReverseNumber
- ✧ AmendDateFormat

Unit 6: Add Row from Cell / Table

- ✧ AppendRow
- ✧ ComputeCell
- ✧ CombineTableByCommonColumn
- ✧ MergeCommonTable
- ✧ MergeTable
- ✧ Table2Cell

Unit 7: Accounting

- ✧ Amortization
- ✧ CurrentBuildBalanceSetting
- ✧ BuildDailyBalance
- ✧ BuildMonthlyBalance
- ✧ BuildWeeklyBalance
- ✧ BuildDailyBalanceCrosstabPeriod

- ✧ BuildMonthyBalanceCrosstabPeriod
- ✧ BuildWeeklyBalanceCrosstabPeriod
- ✧ Date2EffectiveDate
- ✧ Date2DailyPeriod
- ✧ Date2WeeklyPeriod
- ✧ Date2YearlyPeriod
- ✧ DC2NegativePositive
- ✧ DC2PositiveNegative
- ✧ NegativePositive2DC
- ✧ PositiveNegative2DC
- ✧ ReverseDC
- ✧ ReverseDailyVoucher
- ✧ ReverseWeeklyVoucher
- ✧ ReverseMonthlyVoucher
- ✧ VoucherEntry

Unit 8: Process Control

- ✧ ContinueProcess
- ✧ CurrentTable
- ✧ Disable
- ✧ Enable
- ✧ EndProcess
- ✧ ParallelProcess
- ✧ Process
- ✧ ReplaceRule

Unit 9: Conditional Process Control

- ✧ AndCondition2Action
- ✧ AndCondition2Cell
- ✧ OrCondition2Action
- ✧ OrCondition2Cell

Unit 10: Utility

- ✧ FileList2Web
- ✧ Rule2Web

Unit 11: WebNameSQL SQL

Initial Setting

- ✧ CurrentConnectionString
- ✧ CurrentSQLServer
- ✧ CreateSQLServerDatabase

CopyTable

- ✧ DataTableClone2SQLServer
- ✧ WebClone2SQLServer
- ✧ SQLTable2Web

Data Filter and Selection

- ✧ FilterSQLRow.AndCondition
- ✧ FilterSQLRow.AndConditionTable
- ✧ FilterSQLRow.AndDistinctTable
- ✧ FilterSQLRow.OrCondition
- ✧ FilterSQLRow.OrConditionTable
- ✧ FilterSQLRow.OrDistinctTable

Presentation of Data

- ✧ CrosstabSQLTable
- ✧ DistinctSQLTable
- ✧ GroupSQLTableBy

Amendment

- ✧ DataTableAppend2SQLServer
- ✧ WebAppend2SQLServer
- ✧ WebAmend2SQLServer
- ✧ RemoveSQLRow.AndCondition
- ✧ RemoveSQLRow.AndConditionTable
- ✧ RemoveSQLRow.AndDistinctTable
- ✧ RemoveSQLRow.OrCondition
- ✧ RemoveSQLRow.OrConditionTable
- ✧ RemoveSQLRow.OrDistinctTable
- ✧ RemoveSQLDatabase
- ✧ RemoveSQLColumn
- ✧ RemoveSQLTable

Accounting

- ✧ BuildDailySQLBalance
- ✧ BuildMonthlySQLBalance
- ✧ BuildWeeklySQLBalance
- ✧ BuildDailySQLBalanceCrosstabPeriod
- ✧ BuildMonthySQLBalanceCrosstabPeriod
- ✧ BuildWeeklySQLBalanceCrosstabPeriod

Utility

- ✧ RunNonQuerySQL
- ✧ RunSQL2DataTable
- ✧ RunSQL2Web

WebNameSQL Command Syntax by Unit

Unit 1: Copy Data

Except for using the symbol *, there is no change in the final meaningful content of the table after the copying process.

- ✧ CopyTable{SourceWebName | ~ ReturnWebName}
- ✧ CopyTable.CommonTable{ | @ CommonTableName ~ ReturnWebName}
- ✧ CSV2Web{SourceFileName~ ReturnWebName}
- ✧ CSV2Web.CommonTable{SourceFileName@ CommonTableName ~ ReturnWebName }
- ✧ DataTable2Web{SourceDataTable ~ ReturnWebName}
- ✧ Web2CSV{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnFileName}
  Web2CSV{SourceWebName | * ~ ReturnFileName}
- ✧ Web2DataTable{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnWebName}
  Web2DataTable{SourceWebName | * ~ ReturnWebName}
  Note: DataTable is C# ADO.Net memory table.
- ✧ Web2JSON{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnFileName}
  Web2JSON{SourceWebName | * ~ ReturnFileName}
- ✧ Web2HTML{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnFileName}
- ✧ Web2HTML{SourceWebName | * ~ ReturnFileName}
- ✧ Web2MSSQL{SourceWebName | * ~ ReturnDataBaseName(ReturnTableName)}
- ✧ Web2OneColumn{SourceWebName | * ~ ReturnFileName}
- ✧ Web2XML{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnFileName}
- ✧ Web2XML{SourceWebName | * ~ ReturnFileName}
- ✧ ManyCSV2Web{FolderPath(Path)FileFilter(Filter)Subdirectory(Include/Exclude) ~ ReturnWebName }
- ✧ OneColumn2Web{SourceFileName~ ReturnWebName}

Rule details as indicated by blue color are optional settings.

Reserve Symbol	Description
{ }	Start/end of rule detail of a current rule type
#	Block name for a group of rule
|	Read data from SourceWebName
,	Separate different refer names i.e. cell, column, table name
" "	Indicate a text or number value (it is not column name)
*	Select all column names
[]	Include row number of a column
()	Parameter(Setting)
=	Assign new value of a column name
=>	Start mathematical or statistical calculation
@	Set relation with alternative table
.	Start indicate number of decimal places of ReturnColumnName
~	Save data to ReturnWebName

Web represents in-memory data tables. Symbols as indicated in red color are designed to work with Web tables and cells.

For all Web tables and cells created or copied by using the symbol ~ and () respectively, they will be kept within memory until the process is completed. It can be reused or overwritten using the same table/cell name with ~ and ().

Unit 2: Add Column

Column can be added after the last column. To select or reorganize column names, you can use the rule type **Select Column**.

- ComputeColumn{SourceWebName | SourceColumnName, SourceColumnName => Add(ReturnColumnName.d) ~ ReturnWebName}
- ComputeColumn{SourceWebName | SourceColumnName, SourceColumnName => Subtract(ReturnColumnName.d) ~ ReturnWebName}
- ComputeColumn{SourceWebName | SourceColumnName, SourceColumnName => Multiply(ReturnColumnName.d) ~ ReturnWebName}
- ComputeColumn{SourceWebName | SourceColumnName, SourceColumnName => Divide(ReturnColumnName.d) ~ ReturnWebName}
- ComputeColumn{SourceWebName | SourceColumnName, SourceColumnName => CombineText(ReturnColumnName.d) ~ ReturnWebName}

d represents decimal places which support >0.

For Add, Subtract, Multiply and Divide, if you apply the number instead of the cell name, please double quote "" with the number, e.g. "2".

For CombineText, if you apply the text instead of the cell name, , please double quote "" with the number, e.g. "apple".

- ConditionalJoin{TransactionTable(SourceColumnName, SourceColumnName) @ MasterTable(SourceColumnName, SourceColumnName) ~ ReturnWebName}
- FullJoin{TransactionTable(SourceColumnName, SourceColumnName) @ Master Table(SourceColumnName, SourceColumnName) ~ ReturnWebName}
- InnerJoin{TransactionTable(SourceColumnName, SourceColumnName) @ Master Table(SourceColumnName, SourceColumnName) ~ ReturnWebName}
- JoinTable{Transaction Table(SourceColumnName, SourceColumnName) @ Master Table(SourceColumnName, SourceColumnName) ~ ReturnWebName}
- Number2Text{SourceWebName | SoruceNumberColumn, SourceNumberColumn ~ ReturnWebName}
    - ResidualJoin{TransactionTable(SourceColumnName, SourceColumnName) @ Master Table(SourceColumnName, SourceColumnName) ~ ReturnWebName}

ResidualJoin also follows condition rules of **ConditionalJoin**, but ResidualJoin runs slower than ConditionalJoin because it runs each master table row after the process completed of prior master table row. If your condition rules do not have any dependency across each master table row, you can use ConditionalJoin to achieve faster processing speed.

Unit 3: Data Filter and Selection

SourceWebName will not be changed if the new table is saved as an alternative name.

- AndFilter{SourceWebName | SourceColumnName(Condition) SourceColumnName(Condition) ~ ReturnWebName}
- AndFilter.ConditionList{SourceWebName | @ ConditionTableName ~ ReturnWebName}

- AndFilter.DistinctList{SourceWebName | @ DistinctTableName ~ ReturnWebName}
- OrFilter{SourceWebName | SourceColumnName(Condition) SourceColumnName(Condition) ~ ReturnWebName}
- OrFilter.ConditionList{SourceWebName | @ ConditionTableName ~ ReturnWebName}

  Where condition can be any combination of > value, < value, >= value, <= value, = value, != Value
- OrFilter.DistinctList{SourceWebName | @ DistinctTableName ~ ReturnWebName}
- RemoveColumn{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnWebName}
- SelectColumn{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnWebName}

Unit 4: Presentation of Data

Unit 3 and unit 4 are essential to support your different scenarios of reporting.

- ✧ Crosstab{SourceWebName | X(SourceColumnName, SourceColumnName) Y(SourceColumnName, SourceColumnName)
  => Statistics 1(ReturnColumnName) Statistic n(ReturnColumnName) ~ ReturnWebName}
- ✧ Distinct{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnWebName}
- ✧ GroupBy{SourceWebName | SourceColumnName, SourceColumnName => Statistics 1(ReturnColumnName) Statistic n(ReturnColumnName)
  ~ ReturnWebName}
  Where statistics can be Count(), Sum(Column), Max(Column), Min(Column),
  if using count, no need to specify column name
- ✧ OrderBy{SourceWebName | Column 1(A Or D) Column n(A Or D) ~ ReturnWebName}
  Where A is ascending order and D is descending order
- ✧ ReverseCrosstabCSV2Web{SourceFileName~ ReturnWebName}
- ✧ ReverseManyCrosstabCSV2Web{FolderPath(Path)FileFilter(Filter)Subdirectory(Include Or Exclude) ~ ReturnWebName}

Unit 5: Amend Column

No additional column will be generated. ReturnWebName will be amended accordingly. If no setting of the ReturnWebName, the SourceWebName will be amended.

- ✧ AmendColumnName{SourceWebName | SoruceColumnName = ReturnColumnName ~ ReturnWebName}
- ✧ AmendDate{Cell Name => AddYear(n) AddMonth(n) AddDay(n) ~ ReturnWebName}
  AmendDate{Cell Name => Year(n) Month(n) Day(n) ~ ReturnWebName}
  AmendDate{SourceWebName | SourceColumnName, SourceColumnName => AddYear(n) AddMonth(n) AddDay(n) ~ ReturnWebName}
  AmendDate{SourceWebName | SourceColumnName, SourceColumnName => Year(n) AddMonth(n) AddDay(n) ~ ReturnWebName}
  AmendDate{SourceWebName | * => Year(n) AddMonth(n) AddDay(n) ~ ReturnWebName}

  To effect calculation of date, the date format of the SourceWebName must be in OLEAutomationDate format.

- ✧ AmendDateFormat{Cell Name => OLEAutomationDate = MM-dd-yyyy ~ ReturnWebName}
  AmendDateFormat{SourceWebName | SourceColumnName, SourceColumnName => MM-dd-yyyy = OLEAutomationDate
  ~ ReturnWebName}
  AmendDateFormat{SourceWebName | * => MM-dd-yyyy = OLEAutomationDate ~ ReturnWebName}
  Where n Day can be First, Last, 1,2,3
- ✧ ReverseNumber{SourceWebName | SourceNumberColumnName, SourceNumberColumnName ~ ReturnWebName}

Unit 6: Add Row from Cell / Table

Prior to understanding how to **AppendRow**, you shall practice how to manipulate cells by **ComputeCell** and **Table2Cell**.

- ✧ AppendRow{SourceWebName | SourceColumnName(Value or Cell Name) SourceColumnName(Value or

Cell Name) ~ ReturnWebName}

- ComputeCell{SourceCellName1 or "Number", SourceCellName n or "Number"=> Add(Cell Name.2)}
- ComputeCell{SourceCellName1 or "Number", SourceCellName n or "Number"=> Subtract(Cell Name.2)}
- ComputeCell{SourceCellName1 or "Number", SourceCellName n or "Number"=> Multiply(Cell Name.2)}
- ComputeCell{SourceCellName1 or "Number", SourceCellName n or "Number"=> Divide(Cell Name.2)}

  .2 represents decimal places which support >0.

  If you apply the number instead of the cell name, please double quote "" with the number, e.g. "2".

- CombineTableByCommonColumn {SourceWebName, SourceWebName ~ ReturnWebName}
- MergeCommonTable{SourceWebName, SourceWebName ~ ReturnWebName}
- MergeTable{SourceWebName, SourceWebName ~ ReturnWebName}
- Table2Cell{SourceWebName | SourceColumnName, SourceColumnName => Count(ReturnCellName)}
- Table2Cell{SourceWebName | SourceColumnName, SourceColumnName => Sum(ReturnCellName)}
- Table2Cell{SourceWebName | SourceColumnName, SourceColumnName => Max(ReturnCellName)}
- Table2Cell{SourceWebName | SourceColumnName, SourceColumnName => Min(ReturnCellName)}
- Table2Cell{SourceWebName | *Source*ColumnName[SourceRowNumber] => CellAddress(ReturnCell Name)}

  Where row can be First, Last or row number such as 1,10,99. First represents row 1, Last represents last row.

CombineTableByCommonColumn vs **MergeTable**: resulting column of MergeTable may be more than CombineTableByCommonColumn. If more columns are number type, CombineTableByCommonColumn is faster than MergeTable, If more columns are text type, MergeTable is faster than CombineTableByCommonColumn.

Unit 7: Accounting

Following rules are designed specifically for accounting practice. **VoucherEntry** offers simple settings to support general voucher preparation. **Amortization** supports more on specific voucher preparation. **BuildBalance** allows your massive volume of vouchers to be summarized in periodical balances with multiple currencies and segments.

- Amortization{SourceWebName | Cost($) StartDate(d) TotalTenor(t) Method(m) ~ ReturnWebName}

  Where m can be StraightLine, Rule78, ReducingBalance
- BuildDailyBalance{SourceWebName | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName
  @ LedgerMaster ~ ReturnWebName}
- BuildMonthlyBalance{SourceWebName | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName
  @ LedgerMaster ~ ReturnWebName}
- BuildWeeklyBalance{SourceWebName | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName
  @ LedgerMaster ~ ReturnWebName}
- BuildDailyBalanceCrosstabPeriod{SourceWebName | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName
  @ LedgerMaster ~ ReturnWebName}
- BuildWeeklyBalanceCrosstabPeriod{SourceWebName | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName
  @ LedgerMaster ~ ReturnWebName}
- BuildMonthlyBalanceCrosstabPeriod{SourceWebName | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName

@ LedgerMaster ~ ReturnWebName}

Example of LedgerMaster

Ledger	Account	Account Name	Year End:Account
BP01..BP02	1000	Investment properties	1000
BP01..BP02	1090	Fair Value Adjustment	1090
BP01..BP02	1100	Fixed assets - cost	1100
BP01..BP02	1150	Accumulate depreciation	1150
BP01..BP02	1510	Lease commitment receivable	1510
BP01..BP02	1520	Unearn lease income	1520
BP01..BP02	2100	Prepaid charges	2100
BP01..BP02	2200	Deposits	2200
BP01..BP02	2300	Accounts receivable	2300
BP01..BP02	2310	Accrued rental income	2310
BP01..BP02	2320	Rental receivable	2320
BP01..BP02	2500	Bank balances	2500
BP01..BP02	3100	Tenants deposits received	3100
BP01..BP02	3200	Accrued expenses	3200
BP01..BP02	3300	Interest payable	3300
BP01..BP02	3400	Tax Payable	3400
BP01..BP02	3500	Bank loan due within one year	3500
BP01..BP02	4500	Bank loan due after one year	4500
BP01..BP02	5100	Share capital	5100
BP01..BP02	**5200**	**Retained earning - b/f**	**5200**
BP01..BP02	6100	Rental income	5200
BP01..BP02	6500	Bank interest income	5200
BP01..BP02	6900	Sundry income	5200
BP01..BP02	7100	Building management fee	5200
BP01..BP02	7800	Exchange Variation Account	5200
BP01..BP02	8110	Audit fee	5200
BP01..BP02	8120	Bank charges	5200
BP01..BP02	8130	Bank loan interest expenses	5200
BP01..BP02	8140	Business registration fee	5200
BP01..BP02	8160	Depreciation	5200
BP01..BP02	8170	Electricity & water	5200
BP01..BP02	8180	Insurance	5200
BP01..BP02	8200	Legal & professional fees	5200
BP01..BP02	8220	Printing & stationery	5200
BP01..BP02	8230	Rates & government rent	5200
BP01..BP02	8240	Repair & maintenance	5200
BP01..BP02	8250	Staff cost	5200
BP01..BP02	8260	Sundry expenses	5200
BP01..BP02	8270	Telephone & fax	5200
BP01..BP02	8280	Travelling expenses	5200

BP01..BP02	8310	Club House Net Expenses	5200
BP01..BP02	8320	Coach Service	5200
BP01..BP02	8330	Lifts & Escalators Maintenance	5200
BP01..BP02	8340	Minor Asset Items	5200
BP01..BP02	8350	Recreation/Promotion	5200
BP01..BP02	8360	Security Service	5200
BP01..BP02	8370	Uniform	5200
BP01..BP02	8900	Taxation	5200

For each end of financial year, account balance will be carried forward to account as indicated by the column **Year End:Account.** The system support none to many retained accounts.

- CurrentBuildBalanceSetting{LedgerMasterTable}
- Date2EffectiveDate{SourceWebName | Date Column ~ ReturnWebName}
- Date2DailyPeriod{SourceWebName | Column(Date Column) StartDay(n) ~ ReturnWebName}
- Date2WeeklyPeriod{SourceWebName | Column(Date Column) StartWeek(n) ~ ReturnWebName}
- Date2MonthlyPeriod{SourceWebName | Column(Date Column) StarMonth(n) ~ ReturnWebName}
- DC2NegativePositive{SourceWebName | Number Column 1, Number Column n ~ ReturnWebName}
- DC2PositiveNegative{SourceWebName | Number Column 1, Number Column n ~ ReturnWebName}
- NegativePositive2DC{SourceWebName | Number Column 1, Number Column n ~ ReturnWebName}
- PositiveNegative2DC{SourceWebName | Number Column 1, Number Column n ~ ReturnWebName}
- ReverseDC{ SourceWebName | ~ ReturnWebName}
- ReverseDailyVoucher{SourceColumnName, SourceColumnName => AddYear(n) AddMonth(n) AddDay(n) ~ ReturnWebName}
  ReverseDailyVoucher{SourceWebName | SourceColumnName, SourceColumnName => Year(n) Month(n) Day(n) StartDay(n) }
  ReverseDailyVoucher{SourceWebName | * => Year(n) AddMonth(n) Day(n) @ LedgerMaster ~ ReturnWebName}
- ReverseWeeklyVoucher{SourceColumnName, SourceColumnName => NextPeriodAddDay(n) StartWeek(n) ~ ReturnWebName}
- ReverseMonthlyVoucher{SourceColumnName, SourceColumnName => AddYear(n) AddMonth(n) AddDay(n) ~ ReturnWebName}
  ReverseMonthlyVoucher{SourceWebName | SourceColumnName, SourceColumnName => Year(n) Month(n) Day(n) StartMonth(n)}
  ReverseMonthlyVoucher{SourceWebName | * => Year(n) AddMonth(n) Day(n) @ LedgerMaster ~ ReturnWebName}
- VoucherEntry{ SourceWebName | Debit(Column) Credit(Column) Balance(Column) ExcludeBalanceGroupBy(Column) ~ ReturnWebName}

Unit 8: Process Control

Unit 8 introduces simple process control while unit 9 introduces more advanced process control.

- ContinueProcess{BlockName}
- CurrentTable{SourceWebName}
- Disable{Message2Screen, Message2File}
- Enable{Message2Screen, Message2File}
- EndProcess{}
- ParallelProcess{BlockName}

- ✧ Process{BlockName}
  Where "Exit" is an exit function of the Command Group Name of Process{}
- ✧ ReplaceRule{ReplaceName, ReplaceName}
  Support first row to define Replace Name

Unit 9: Conditional Process Control

Actual cell value is essential to work with condition. You can use **ComputeCell** and **Table2Cell** as introduced in unit 6 to generate one or more cell value(s) to work with condition(s).

- ✧ AndCondition2Cell{CellName(Condition) CellName(Condition) => Cell Name}
- ✧ AndCondition2Action{CellName(Condition) CellName(Condition) => Process(BlockName)}
  Where Condition can be any combination of > value, < value, >= value, <= value, = value, != Value
- ✧ OrCondition2Action{CellName(Condition) CellName(Condition) => Process(BlockName)}
- ✧ OrCondition2Cell{CellName(Condition) CellName(Condition) => Cell Name}
  Where Condition can be any combination of > value, < value, >= value, <= value, = value, != Value

Unit 10: Utility

These 2 commands are used to support your management of rule files.

- ✧ FileList2Web{FolderPath(Path)FileFilter(Filter)Subdirectory(Exclude/Include) ~ ReturnWebName}
- ✧ Rule2Web{SourceWebName | ColumnName of CalcRule File Path ~ ReturnWebName}

Unit 11: WebNameSQL SQL

Startup

- ✧ CurrentConnectionString{Connection String}
  e.g. Server=localhost\SQLEXPRESS;Database=master;Trusted_Connection=True;
- ✧ CurrentSQLServer{Microsoft SQL Server, Version}
- ✧ CreateSQLDatabase{ReturnSQLDataBase}

CopyTable

- ✧ DataTableClone2SQLServer{DataTable ~ ReturnSQLTable}
- ✧ WebClone2SQLServer{SourceWebName | SourceColumnName, SourceColumnName ~ ReturnSQLTable}
  WebClone2SQLServer{SourceWebName | * ~ ReturnSQLTable }
- ✧ SQLTable2Web{SourceSQLTable | * ~ ReturnWebName}
  SQLTable2Web{SourceSQLTable | SourceColumnName, SourceColumnName ~ ReturnWebName}

Data Filter and Selection

- ✧ FilterSQLRow.AndCondition{SourceSQLTable | SourceColumnName(Condition), SourceColumnName (Condition) ~ ReturnWebName}
- ✧ FilterSQLRow.AndConditionTable{SourceSQLTable | @ SourceWebName ~ ReturnWebName}
- ✧ FilterSQLRow.AndDistinctTable{ SourceSQLTable | @ SourceWebName ~ ReturnWebName}
- ✧ FilterSQLRow.OrCondition{SourceSQLTable | Column 1(Condition), Column n(Condition) ~ ReturnWebName}
- ✧ FilterSQLRow.OrConditionTable{SourceSQLTable | @ SourceWebName ~ ReturnWebName}
- ✧ FilterSQLRow.OrDistinctTable{SourceSQLTable | @ SourceWebName ~ ReturnWebName}

Presentation of Data

- ✧ CrosstabSQLTable{SourceSQLTable | X(SourceColumnName, SourceColumnName) Y(SourceColumnName, SourceColumnName) => Statistics(ReturnColumnName) Statistic (ReturnColumnName) ~ ReturnWebName}
- ✧ DistinctSQLTable{SourceSQLTable | SourceColumnName, SourceColumnName ~ ReturnWebName}
- ✧ GroupSQLTableBy{SourceSQLTable | SourceColumnName, SourceColumnName => Statistics (ReturnColumnName) Statistics(ReturnColumnName) ~ ReturnWebName}
  Where statistics can be Count(), Sum(Column), Max(Column), Min(Column),
  if using count, no need to specify column name

Amendment

- ✧ DataTableAppend2SQLServer{DataTable ~ ReturnSQLTable}
- ✧ WebAppend2SQLServer{SourceWebName | SourceColumnName, SourceColumnName ~ TargetSQLTable}
- ✧ WebAppend2SQLServer{SourceWebName | * ~ TargetSQLTable}
- ✧ WebAmend2SQLServer{SourceWebName | AmendKey(SourceColumnName, SourceColumnName) AmendMode (Period/Fiscal Year/Calendar Year/Month/Day) TargetSQLTable(TableName)}
- ✧ RemoveSQLRow.AndCondition{TargetSQLTable | TargetColumnName(Condition), TargetColumnName(Condition)}
- ✧ RemoveSQLRow.AndConditionTable{TargetSQLTable | @ SourceWebName}
- ✧ RemoveSQLRow.AndDistinctTable{TargetSQLTable | @ SourceWebName}
- ✧ RemoveSQLRow.OrCondition{TargetSQLTable | TargetColumnName(Condition), TargetColumnName(Condition)}

- RemoveSQLRow.OrConditionTable{TargetSQLTable | @SourceWebName}
- RemoveSQLRow.OrDistinctTable{TargetSQLTable | @SourceWebName}
- RemoveSQLDatabase{TargetSQLDatabase}
- RemoveSQLColumn{TargetSQLTable | TargetColumnName, TargetColumnName}
- RemoveSQLTable{TargetSQLTable}

Accounting (Current version has not included this functions)

- BuildDailySQLBalance{SourceSQLTable | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName @ LedgerMaster ~ ReturnWebName}
- BuildWeeklySQLBalance{SourceSQLTable | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName @ LedgerMaster ~ ReturnWebName}
- BuildMonthlySQLBalance{SourceSQLTable | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName @ LedgerMaster ~ ReturnWebName}
- BuildDailySQLBalanceCrosstabPeriod{SourceSQLTable | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName @ LedgerMaster ~ ReturnWebName}
- BuildWeeklySQLBalanceCrosstabPeriod{SourceSQLTable | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName @ LedgerMaster ~ ReturnWebName}
- BuildMonthlySQLBalanceCrosstabPeriod{SourceSQLTable | SourceTextColumnName, SourceTextColumnName, SourceAmountColumnName, SourceAmountColumnName @ LedgerMaster ~ ReturnWebName}

Utility

- RunNonQuerySQL{SQL Statement}
- RunSQL2DataTable{SQL Statement ~ ReturnDataTable}
- RunSQL2Web{SQL Statement ~ ReturnWebName}

Use Case 1: Accounting for Sales of Goods

```
CSV2Web{Trading.csv ~ trading}
VoucherEntry{trading | Credit(Sales) Debit(Discount) Balance(Account Receivable)
ExcludeBalanceGroupBy(Item No) => Amount}
SelectColumn{Date, D/C, Account, Invoice No,Customer Code, Item No, Amount}
OrderBy{Date(A) Invoice No(A) D/C(D)}
Web2CSV{trading | * ~ Result-VoucherEntryTrading2.csv}
EndProcess{}
```

Use Case 2: Build Monthly Account Balance

```
CSV2Web{LedgerMasterRange.csv ~ LedgerMaster}
CSV2Web{VoucherList10X.csv ~ Voucher}

Date2MonthlyPeriod{DateColumn(Date) StartMonth(1)}
BuildMonthlyBalance{Voucher | Ledger, Account, Amount @ LedgerMaster ~ TrialBalance}
Web2CSV{TrialBalance | * ~ Result-TrialBalanceByPeriod4a.csv}

Date2WeeklyPeriod{Voucher | DateColumn(Date) CultureOption(zh-HK) StartWeek(1)}
BuildWeeklyBalance{Voucher | Ledger, Account, Amount @ LedgerMaster ~ TrialBalanceByWeek}
Web2CSV{TrialBalanceByWeek | * ~ Result-TrialBalanceByWeek4a.csv}

Date2DailyPeriod{Voucher | DateColumn(Date) CultureOption(zh-HK) StartDay(1)}
BuildDailyBalance{Voucher | Ledger, Account, Amount @ LedgerMaster ~ TrialBalanceByDay}
Web2CSV{TrialBalanceByDay | * ~ Result-TrialBalanceByDay4a.csv}
```

Use Case 3: Account Reconcilation

```
CSV2Web{1MillionRows.csv ~ Table1}
ComputeColumn{Quantity, Unit Price => Multiply(Original Amount.2)}
ComputeColumn{Original Amount, Exchange Rate => Multiply(Base Amount.2)}
ComputeColumn{"Table1" => CombineText(TableName)}
CSV2Web{1MillionRowsAmend.csv ~ Table2}
ComputeColumn{Quantity, Unit Price => Multiply(Original Amount.2)}
ComputeColumn{Original Amount, Exchange Rate => Multiply(Base Amount.2)}
ReverseNumber{Quantity, Original Amount, Base Amount}
ComputeColumn{"Table2" => CombineText(TableName)}
CopyTable.CommonTable{Table2 | @ Table1 ~ Table2b}
MergeCommonTable{Table2b, Table1 ~ Table}
GroupBy{Date,Ledger,Account,PartNo,Project,Contact,Unit Code,D/C,Currency => Sum(Quantity) Sum(Original
Amount) Sum(Base Amount) ~ Group}
OrFilter{Quantity(!=0)Original Amount(!=0)Base Amount(!=0)}
Web2CSV{Group | * ~ Result-Reconcilation-Summary.csv}
SelectColumn{Date,Ledger,Account,PartNo,Project,Contact,Unit Code,D/C,Currency}
CurrentTable{Table}
AndFilter.DistinctList{@ Group ~ Reconcilation-Detail}
Web2CSV{Reconcilation-Detail | * ~ Result-Reconcilation-Detail2.csv}
EndProcess{}
```

Use Case 4: Amortisation for Budgeting

```
Process{Import Data}
Process{Acquisition}
Process{Amortization}
Process{Disposal}
Process{Combine Voucher}
Process{Export Report}
EndProcess{}

# Import Data
CSV2Web{Cost.csv ~ Table}
Amortization{Method(MonthlyBasis,StraightLine,ProRateActualDay,Round2) ~ AmortizedTable}
Date2MonthlyPeriod{DateColumn(Date) StartMonth(1)}
Number2Text{Tenor}
AmendColumnName{Text:Tenor = TextTenor}
Web2CSV{AmortizedTable | * ~ Result-AmortizedTable.csv}

# Acquisition
AndFilter{Tenor(=0) ~ Acquisition}
SelectColumn{Date, mPeriod Change, AssetID, Tenor, TextTenor, Acquisition}
VoucherEntry{Debit(Acquisition) Credit(Payable) => Amount}
ComputeColumn{"Acquisition" => CombineText(Voucher Type) ~ Acquisition2}

# Amortization
AndFilter{AmortizedTable | Amortization(>0) ~ Amortization}
SelectColumn{Date, mPeriod Change, AssetID, Tenor, TextTenor, Amortization}
VoucherEntry{Debit(Amortization) Credit(AccAmortization) => Amount}
ComputeColumn{"Amortization" => CombineText(Voucher Type) ~ Amortization2}

# Disposal
AndFilter{AmortizedTable | Disposal(>0) ~ Disposal}
SelectColumn{Date, mPeriod Change, AssetID, Tenor, Disposal}
ComputeColumn{Tenor, "1" => Subtract(Last Tenor)}
SelectColumn{AssetID, Last Tenor}
AmendColumnName{Last Tenor = Tenor}
AndFilter.DistinctList{@ AmortizedTable | Disposal ~ DisposalOfCost2}

SelectColumn{Date, mPeriod Change, AssetID, Tenor, TextTenor, Acquisition, AccAmortization}
Web2CSV{DisposalOfCost2 | * ~ Result-DisposalOfCost2.csv}
VoucherEntry{Debit(AccAmortization) Credit(Acquisition)Balance(Receivable) => Amount}
ComputeColumn{"Disposal" => CombineText(Voucher Type) ~ Disposal2}

# Combine Voucher
CombineTableByCommonColumn{Acquisition2, Amortization2, Disposal2 ~ Voucher}
OrderBy{AssetID(A) mPeriod Change(A)}
AmendDateFormat{Date => OLEAutomationDate = dd-MMM-yyyy}
ComputeColumn{"Tenor ",TextTenor, ": ", Voucher Type => CombineText(Particular)}
```

SelectColumn{Date, mPeriod Change, Voucher Type, D/C, Account, Amount, AssetID, Particular ~ VoucherList}

Export Report
Web2CSV{VoucherList | * ~ Result-VoucherList.csv}
Crosstab{X(Voucher Type, Account, D/C) Y(AssetID, mPeriod Change) => Sum(Amount) ~ TrialBalanceByAssetIDByPeriod}
Web2CSV{TrialBalanceByAssetIDByPeriod | * ~ Result-TrialBalanceByAssetIDByPeriod.csv}
Crosstab{VoucherList | X(Voucher Type, Account, D/C) Y(mPeriod Change) => Sum(Amount) ~ TrialBalanceByPeriod}
Web2CSV{TrialBalanceByPeriod | * ~ Result-TrialBalanceByPeriod.csv}
Crosstab{VoucherList | X(Voucher Type, Account, D/C) Y(AssetID) => Sum(Amount) ~ TrialBalanceByAssetID}
Web2CSV{TrialBalanceByAssetID | * ~ Result-TrialBalanceByAssetID.csv}

Use Case 5: Accounting for Sales of Goods with Conditions

ReplaceRule{305570}
CSV2Web{Customer.csv ~ Customer}
CSV2Web{SaleInvoice.csv ~ SaleTable}
JoinTable{SaleTable(Customer Code) @ Customer(Customer Code) ~ SaleWithDiscountRate}
Table2Cell{HKD Amount => Sum(sumValue1.2)}
AndCondition2Action{sumValue1(>=305570) => Process(Sales Revenue1)}
AndCondition2Action{sumValue1(<305570) => Process(Sales Revenue2)}
Process{Sales Discount}
Process{Account Receivable}
Process{Combining Journal Entries}
EndProcess{}

Sales Revenue1
NegativePositive2DC{SaleWithDiscountRate | HKD Amount ~ SaleEntry}
ComputeColumn{"6100" => CombineText(Account)}

Sales Revenue2
NegativePositive2DC{HKD Amount ~ SaleEntry}
ComputeColumn{"6200" => CombineText(Account)}

Sales Discount
CurrentTable{SaleWithDiscountRate}
ComputeColumn{"-1", HKD Amount, Discount => Multiply(HKD Amount.2)}
NegativePositive2DC{HKD Amount}
ComputeColumn{"7100" => CombineText(Account) ~ DiscountEntry}

```
# Account Receivable
MergeTable{SaleEntry, DiscountEntry ~ AccountReceivable}
DC2NegativePositive{HKD Amount}
GroupBy{Date, Invoice No, Customer Code, Customer Name    => Sum(HKD Amount)}
PositiveNegative2DC{HKD Amount}
ComputeColumn{"2600" => CombineText(Account)    ~ AccountReceivableEntry}

# Combining Journal Entries
MergeTable{SaleEntry, AccountReceivableEntry, DiscountEntry ~ CombineEntry}
OrderBy{Invoice No(A) D/C(D)}
SelectColumn{Date, D/C, Account, Invoice No,Customer Code, Customer Name,Item No, HKD Amount}
Web2CSV{CombineEntry | * ~ Result-RunBlockVoucherList.csv}
```

WebNameSQL Command by Alphabetical Order

1. AmendColumnName
2. AmendDate
3. AmendDateFormat
4. Amortization
5. AndCondition2Action
6. AndCondition2Cell
7. AndFilter
8. AndFilter.ConditionList
9. AndFilter.DistinctList
10. AppendRow
11. BuildDailyBalance
12. BuildDailyBalanceCrosstabPeriod
13. BuildDailySQLBalance
14. BuildDailySQLBalanceCrosstabPeriod
15. BuildWeeklyBalance
16. BuildWeeklyBalanceCrosstabPeriod
17. BuildWeeklySQLBalance
18. BuildWeeklySQLBalanceCrosstabPeriod
19. BuildYearlyBalance
20. BuildYearlyBalanceCrosstabPeriod
21. BuildYearlySQLBalance
22. BuildYearlySQLBalanceCrosstabPeriod
23. CombineTableByCommonColumn
24. ComputeCell
25. ComputeColumn
26. ConditionalJoin
27. ContinueProcess
28. CopyTable
29. CopyTable.CommonTable
30. CreateSQLServerDatabase
31. Crosstab
32. CrosstabSQLTable
33. CSV2Web
34. CSV2Web.CommonTable
35. CurrentBuildBalanceSetting
36. CurrentConnectionString
37. CurrentSQLServer
38. CurrentTable
39. DataTable2Web
40. DataTableAppend2SQLServer
41. DataTableClone2SQLServer
42. Date2DailyPeriod
43. Date2EffectiveDate
44. Date2WeeklyPeriod
45. Date2YearlyPeriod
46. DC2NegativePositive
47. DC2PositiveNegative
48. Disable
49. Distinct
50. DistinctSQLTable
51. Enable
52. EndProcess
53. FileList2Web
54. FilterSQLRow.Condition
55. FilterSQLRow.AndConditionTable
56. FilterSQLRow.AndDistinctTable
57. FilterSQLRow.OrCondition
58. FilterSQLRow.OrConditionTable
59. FilterSQLRow.OrDistinctTable
60. FullJoin
61. GroupBy
62. GroupSQLTableBy
63. InnerJoin
64. JoinTable
65. Web2CSV
66. Web2DataTable
67. Web2HTML
68. Web2JSON
69. Web2OneColumn
70. Web2XML
71. WebAmend2SQLServer
72. WebAppend2SQLServer
73. WebClone2SQLServer
74. ManyCSV2Web
75. MergeCommonTable
76. MergeTable
77. NegativePositive2DC
78. Number2Text
79. OneColumn2Web
80. OrCondition2Action
81. OrCondition2Cell
82. OrderBy
83. OrFilter
84. OrFilter.ConditionList
85. OrFilter.DistinctList
86. ParallelProcess
87. PositiveNegative2DC
88. Process
89. RemoveColumn
90. RemoveSQLColumn
91. RemoveSQLDatabase
92. RemoveSQLRow.AndCondition
93. RemoveSQLRow.AndConditionTable
94. RemoveSQLRow.AndDistinctTable
95. RemoveSQLRow.OrCondition
96. RemoveSQLRow.OrConditionTable
97. RemoveSQLRow.OrDistinctTable
98. RemoveSQLTable
99. ReplaceRule
100. ResidualJoin
101. ReverseCrosstabCSV2Web
102. ReverseDailyVoucher
103. ReverseDC
104. ReverseManyCrosstabCSV2Web
105. ReverseMonthlyVoucher
106. ReverseNumber
107. ReverseWeeklyVoucher
108. RunNonQuerySQL
109. RunSQL2DataTable
110. RunSQL2Web
111. SelectColumn
112. SQLTable2Web
113. Table2Cell
114. Rule2Web
115. VoucherEntry