

TAI QUERY REPORTER USERS GUIDE

VERSION 1.0.0.1

Introduction

2

- ❑ **TAI Query Tool provides users with ad hoc ability to generate customized reports and analytics to monitor and improve their daily operations.**
- ❑ **The tool's graphical capability enables users to visualize the results.**
- ❑ **Query tool's functionality resembles that of Microsoft SQL Query Analyzer where a user has the flexibility to manipulate data at the lowest level.**

General

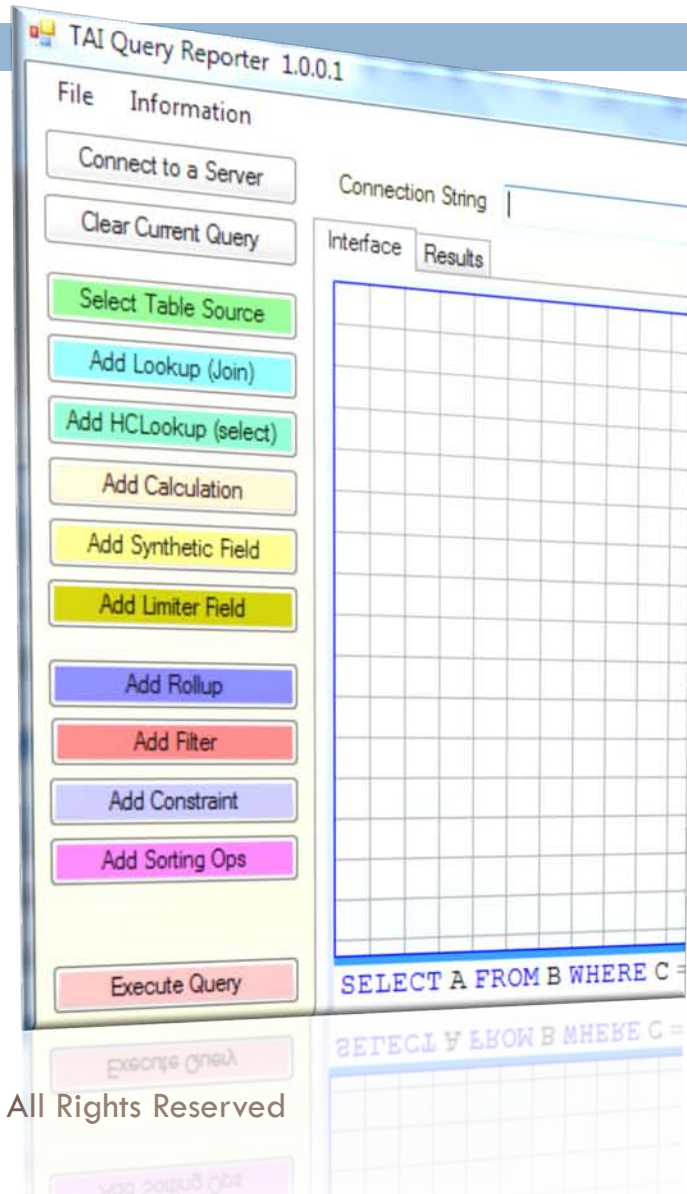
3

- ❑ **The Query tool enables users, familiar with data elements of a database and knowledge of where to find certain information, to explore the depth and wealth of data.**
- ❑ **Data dictionary document, if available, is a good reference material describing data elements in tables and databases.**
- ❑ **This version of the tool requires .NET 4 Framework to be installed on the machine on which the tool is executed/run from.**

Key Functionality

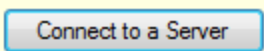
4

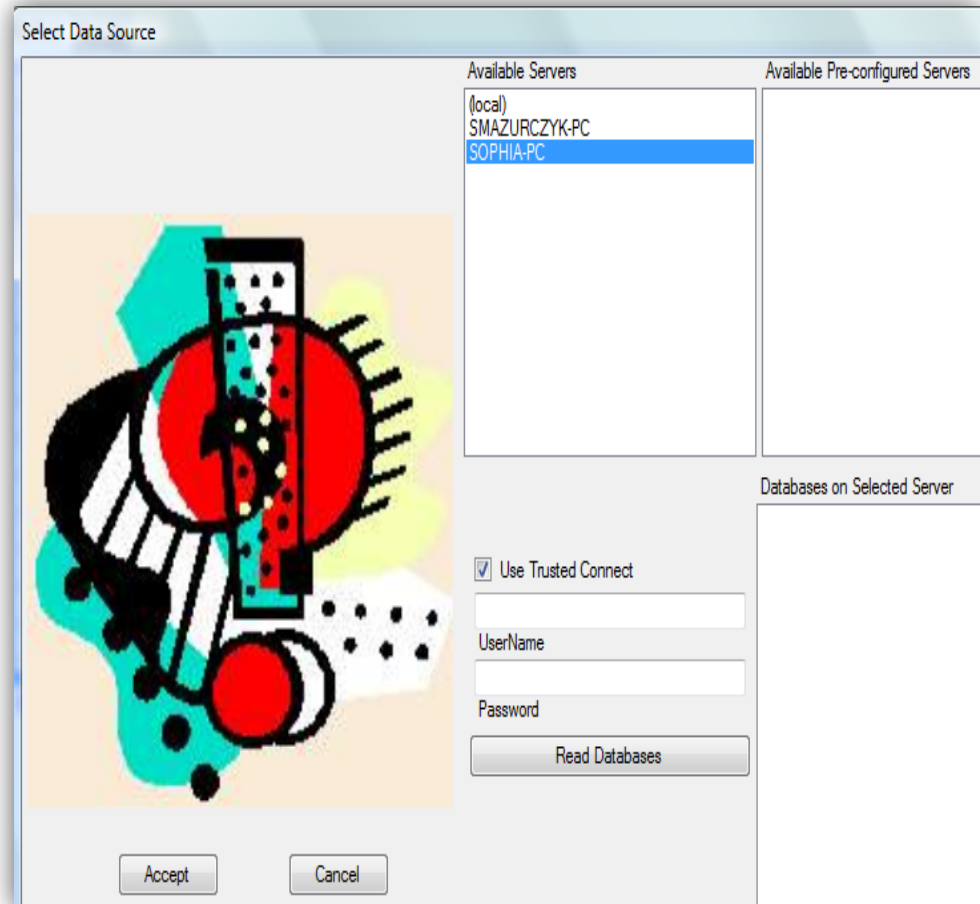
- **The following lists key functionality of the tool as depicted by the buttons to the right in a partial screen capture:**



Connect to a Server

5

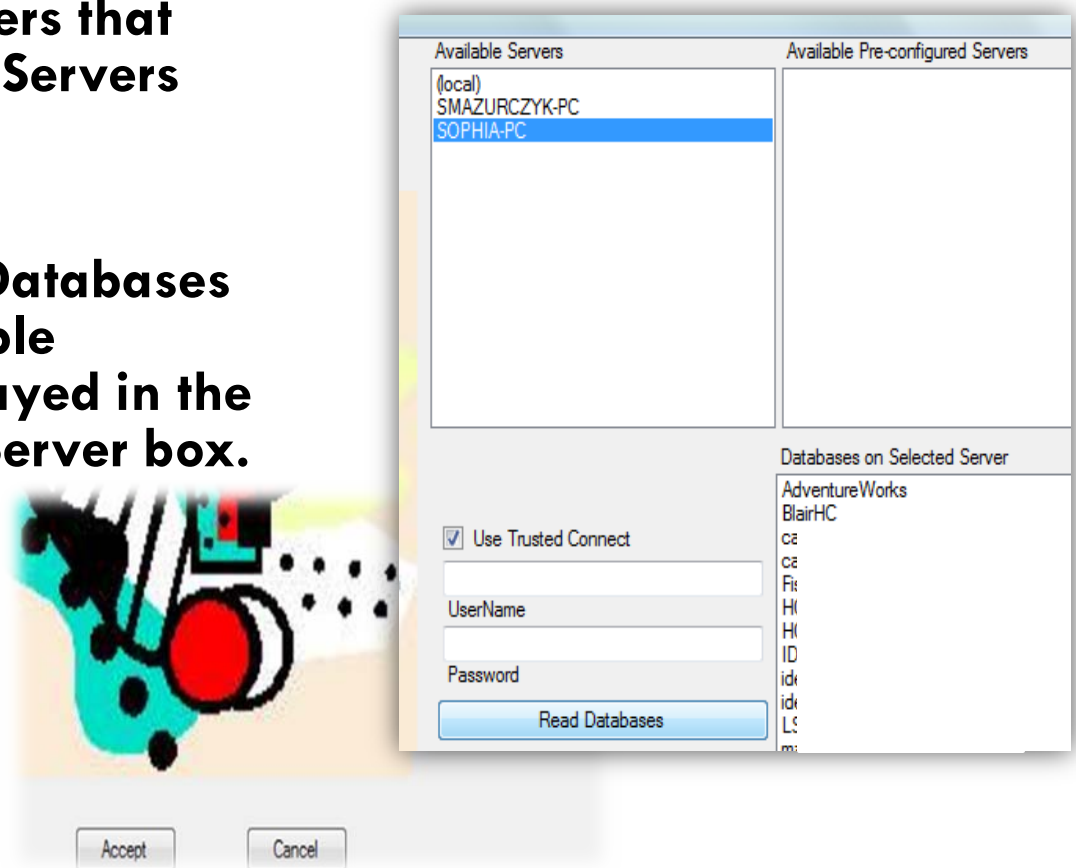
- **The first step is to find a database server. Click on the  button.**
- **The tool will automatically search and display a list of available database servers that are accessible to the tool.**
- **Users will need a SQL login to the server to which they'll be connecting.**



Select Server & dB

6

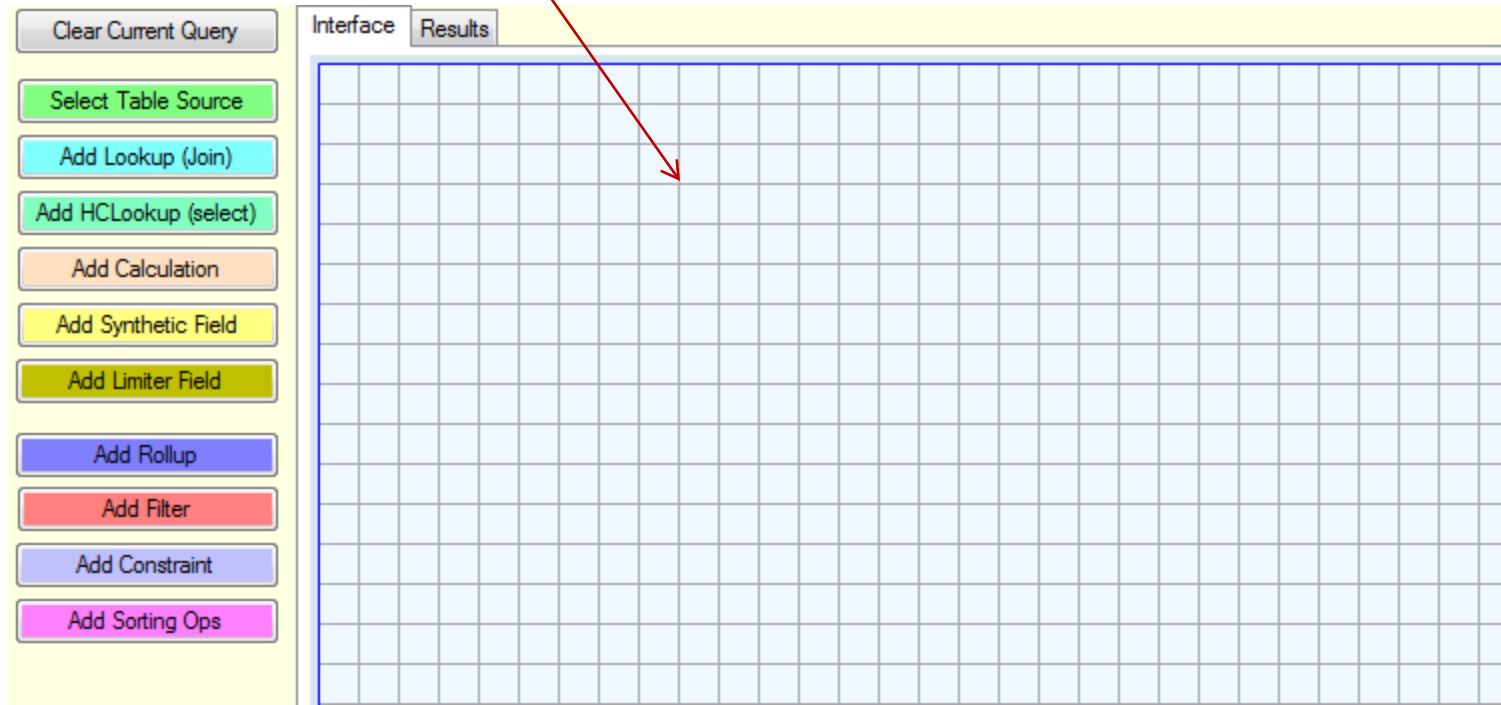
- ❑ Click on one of the servers that appear in the **Available Servers** box.
- ❑ Then click on the **Read Databases** button. A list of available databases will be displayed in the **Databases on Selected Server** box.
- ❑ Click on the database you would like to work with – i.e. **BlairHC**.
Then click on the **Accept** button



Clear Current Query

7

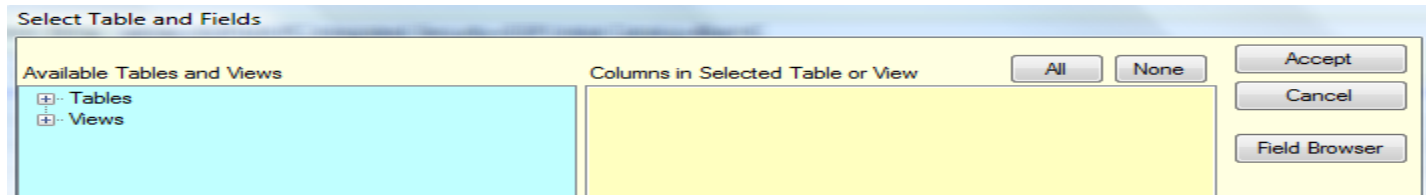
- **Clicking on the Clear Current Query button will erase the work area in preparation for a new query.**



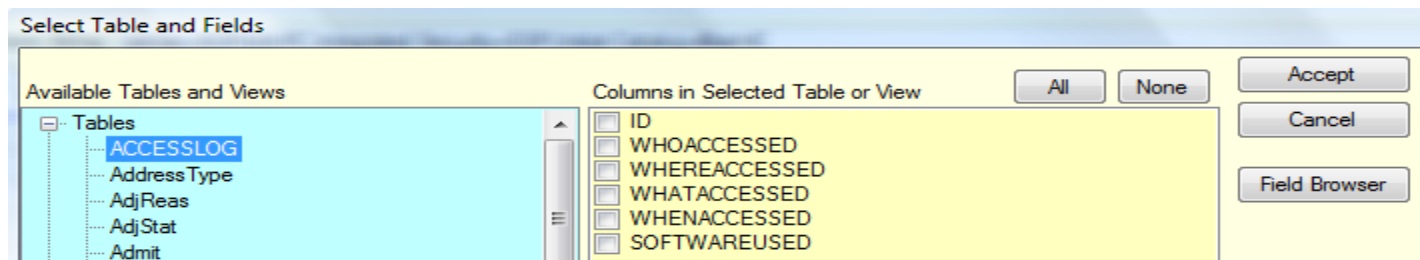
Select Table Source

8

- Before you can start creating queries, you will need to select the tables, their fields/data elements, etc.
- To obtain a list of tables, expand them by clicking on the plus sign to the left of the Tables.



- Then click on the actual table – i.e. **ACCESSLOG** and the list of its fields will be displayed in the area to the right. Select the individual fields by clicking to the left of each and then press **Accept** button.



Query Display

9

- Clicking on the Field Browser button will display table data.

Field Browser					
Sample Field Contents					
ID	WHOACCESSED	WHEREACCESSED	WHATACCESSED	WHENACCESSED	SOFTWAREUSED
129	tairemote	HCSVR2	EXECUTED	7/20/2010	CLAIMSEXPLORER

- The basic query was started when you accepted the fields. This is illustrated by the DATASOURCE image in the work area. Below the work space is the actual SQL query that is created for you.
- If you mouse over the image, you'll see the details such as the table and fields selected.

DATASOURCE																																																						
<pre>SELECT ACCESSLOG.ID, ACCESSLOG.WHOACCESSED, ACCESSLOG.WHEREACCESSED, ACCESSLOG.WHATACCESSED, ACCESSLOG.WHENACCESSED, ACCESSLOG.SOFTWAREUSED FROM ACCESSLOG</pre>																																																						

Add Lookup (Join)

10

- This functionality allows you to collect data from two different tables where both tables have a field that is common. This field will link the two tables.
- For example, if you need to pull data that spans among multiple tables (i.e. partly in Claims_all and partly in Lag_detail2), a field that is common to both in this case is “claim_num” and “claim_number” in the Lag_detail2 table.

Add / Edit Lookup (Join)

Available Fields For Index	Available Tables For Indexing Into	Fields In Table To Index Against	Fields To Return On A Match
payor claimmst_i claim_num date_rec site_id prin_diag oth1_diag oth2_diag claimdet_i date_from date_to proc_code revenuecode check_id adjud_doll pay_doll units contractrate touch_date fin_cat check_date prov_id name fin_cat_name prin_diag_code oth1_diag_code oth2_diag_code prin_diag_name oth1_diag_name oth2_diag_name descript patient_id	fee_new FilerRelation Global_Code HcfaMan HNA_PHP icm0708 icm0809 icm0910 IMH_ROLL Independent_OP Independent_prac_members inp_clm_rol INP_detox INP_MH inp_rehab Insuspan irte_det irte_hed L1ReviewFormat L2ReviewFormat lag lag_all_2 Lag_Detail lag_detail_2 lag_detail2 Language LOC LOCtoCPT LOOKUPCOUNTY LOOKUPPROVIDERSPECIALTY LOOKUPPROVIDERTYPES map_tbl_short	fin_cat_name category of aid paidamt servicemon paidmon claimmst_i claim_number dos	fin_cat_name category of aid paidamt servicemon paidmon claimmst_i claim_number dos

Step 1... Select the field you want to use as the index field

Step 2... Select the table you want to index into

Step 3... Select the field in this table that you want to index against

Step 4... Finally select the field you want the index to return

☒ Fields in Base Table and fields in Match Table or empty (NULL) values in match table where no match is found.

☐ Fields in base table and fields in match table only. If match table fields do not exist then resulting row in Base table is also not returned

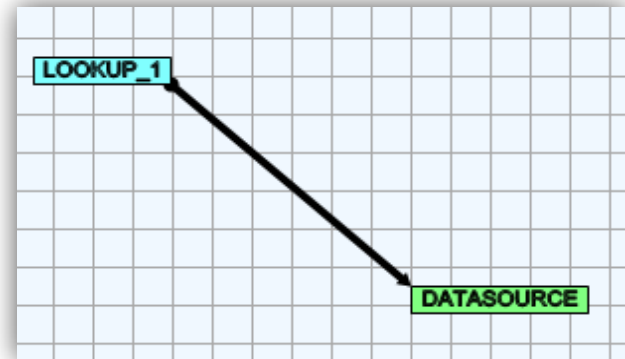
Accept

Cancel

Lookup detail

11

- This will result in a **Lookup_1** image in your work space
- It will also add to the query on the bottom, below the workspace.



```
SELECT
claims_all.claim_num,claims_all.date_from,claims_all.date_to,claims_all.proc_code,
claims_all.pay_doll,claims_all.units,claims_all.fin_cat,claims_all.check_date,
claims_all.fin_cat_name,claims_all.prin_diag_code,
LT0.PaidAmt AS 'claim_num_PaidAmt',
LT0.Units AS 'claim_num_Units'
FROM claims_all
LEFT OUTER JOIN lag_detail2 AS LT0 ON claims_all.claim_num=LT0.claim_number
```

Add HCLookup (Select)

12

- This function is similar to the other Lookup with the exception that it does a “Select TOP 1” subquery rather than a Join. The purpose is to ensure that data element from only one record is used for joining. This may be useful in an environment with duplicate records – such as duplicate members

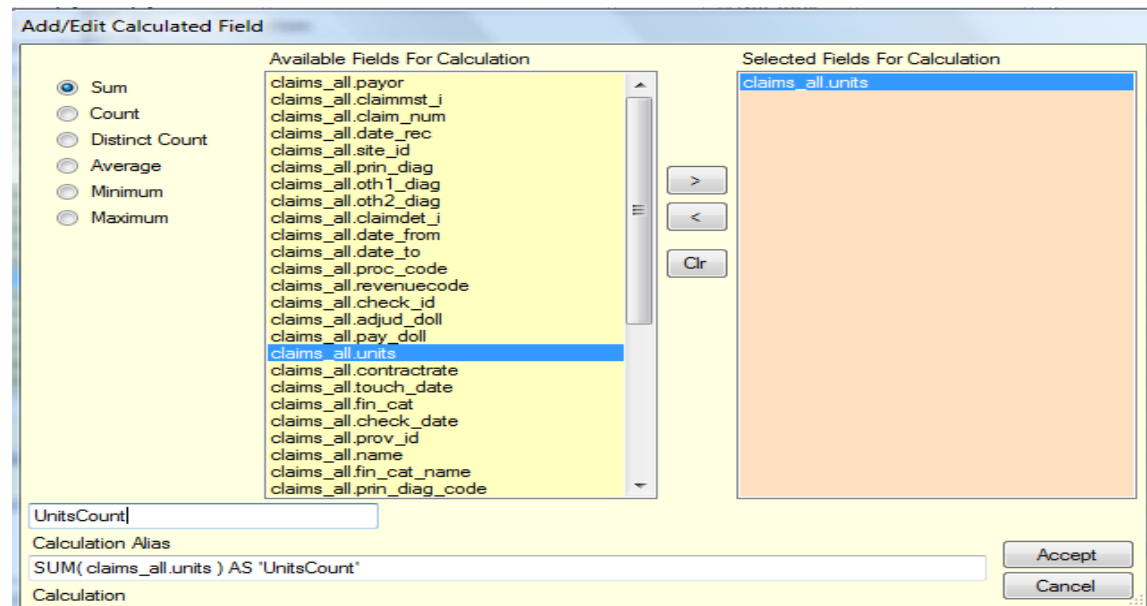
HCLOOKUP_1

```
SELECT
tblMemberServices.SSN,tblMemberServices.COSTCTR,
(SELECT TOP 1 FIRSTNAME FROM tblMemberMain WHERE tblMemberServices.SSN=tblMemberMain.SSN) AS 'SSN_FIRSTNAME',
(SELECT TOP 1 LASTNAME FROM tblMemberMain WHERE tblMemberServices.SSN=tblMemberMain.SSN) AS 'SSN_LASTNAME',
(SELECT TOP 1 DOB FROM tblMemberMain WHERE tblMemberServices.SSN=tblMemberMain.SSN) AS 'SSN_DOB'
FROM tblMemberServices
```

Add Calculation

13

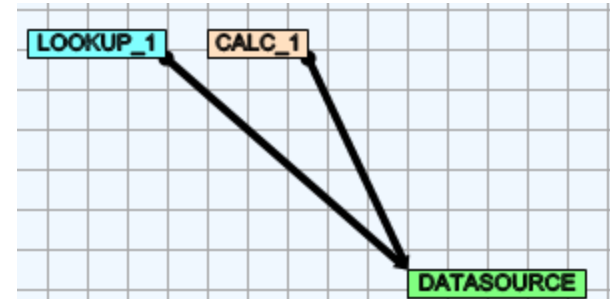
- ❑ This function allows users to run a total on a specific data field.
- ❑ If you notice, the actual query is displayed in the Calculation text box.
- ❑ You can name the resulting displayed field to something more meaningful in the Calculation Alias text box.
- ❑ Clicking Accept button will add this function into the query in your work space and it appears as SUM



Calculation – SQL SUM

14

- Image Calc is displayed in the work area.



- Sum line is added into the query.

```
SUM( claims_all.units ) AS 'UnitsCount'
```

Add Synthetic Field

15

- This function serves the purpose of doing additional calculations or manipulation of data fields. For example if you want to add two existing fields to create a new field on the report.

Add Synthetic Field

Simple Math Functions

- ☒ Add Multiple Numeric Fields Together
- ☐ Subtract Multiple Numeric Fields from each other
- ☐ Multiply Multiple Numeric Fields Together
- ☐ Divide Multiple Numeric Fields into each other

Simple Date Functions

- ☐ Get the Year of a Date
- ☐ Get the Month of a Date
- ☐ Get the Day of a Date
- ☐ Get the Day of the Year of a Date
- ☐ Get the Week of the Year of a Date
- ☐ Get the Quarter of the Year of a Date
- ☐ Calculate Age in Years from Today
- ☐ Calculate Age in Years between two selected dates

Simple String Functions

- ☐ Upper Case
- ☐ Lower Case
- ☐ Pull X Characters from Left of field
- ☐ Pull X Characters from Right of field
- ☐ Pull X Characters Starting From Y

Fields to apply the synthetic creation to:

COLUMN NAME	DATA TYPE	ORDINAL POSITION	TABLE NAME
claimmst_i	int	2	claims_all
site_id	int	5	claims_all
prin_diag	int	6	claims_all
oth1_diag	int	7	claims_all
oth2_diag	int	8	claims_all
claimdet_i	int	9	claims_all
PaidAmt	money	10	claims_all
Units	int	12	lag_detail2
check_id	int	13	lag_detail2
claimmst_i	int	14	lag_detail2
check_id	int	14	lag_detail2
adjud_doll	money	15	claims_all
pay_doll	money	16	claims_all
units	int	17	claims_all
contractrate	money	18	claims_all
prov_id	int	22	claims_all
patient_id	int	32	claims_all
cla_doll	money	33	claims_all
claimadj_i	int	37	claims_all
stat_id	int	39	claims_all

Claims_TOTALS

Synthetic Field Alias

(claims_all adjud_doll+claims_all cla_doll) AS 'Claims_TOTALS'

Synthetic Field Creation Syntax

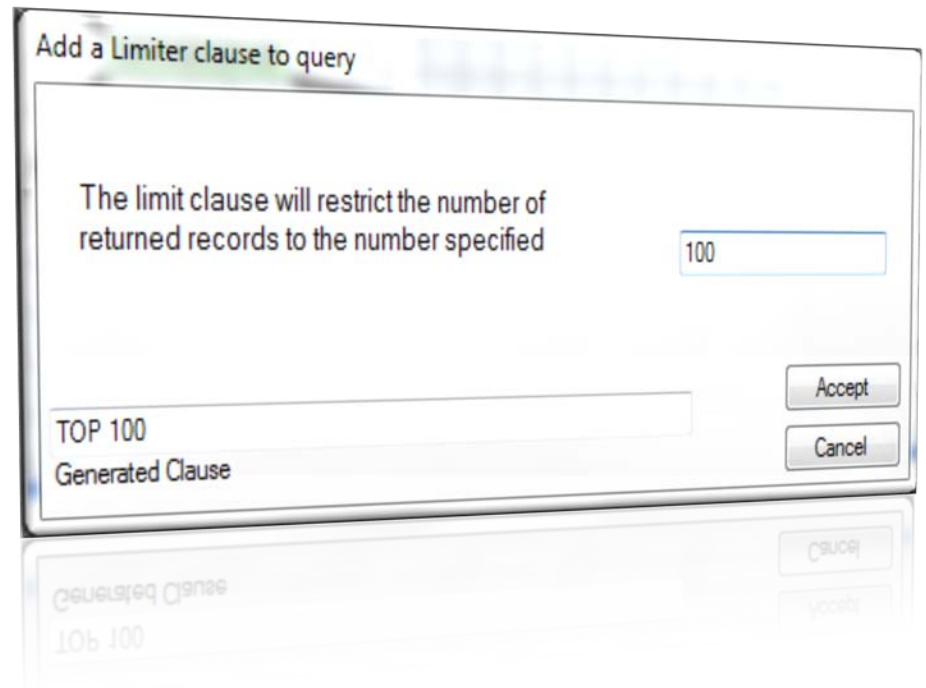
Accept

Cancel

Add Delimiter Field

16

- This function gives you the control over how many records will be returned from the query. For example, this may be useful if you're interested in the top 100 members only. This will translate to "SELECT TOP 100" in the query.



Add Rollup

17

- ❑ Rollup function lets you group or organize your records that will be displayed by selected fields you chose to be in the results set. Rollup is performed on rows of informational record set with same values.
- ❑ A Rollup image will be added to graphical portion of the work area



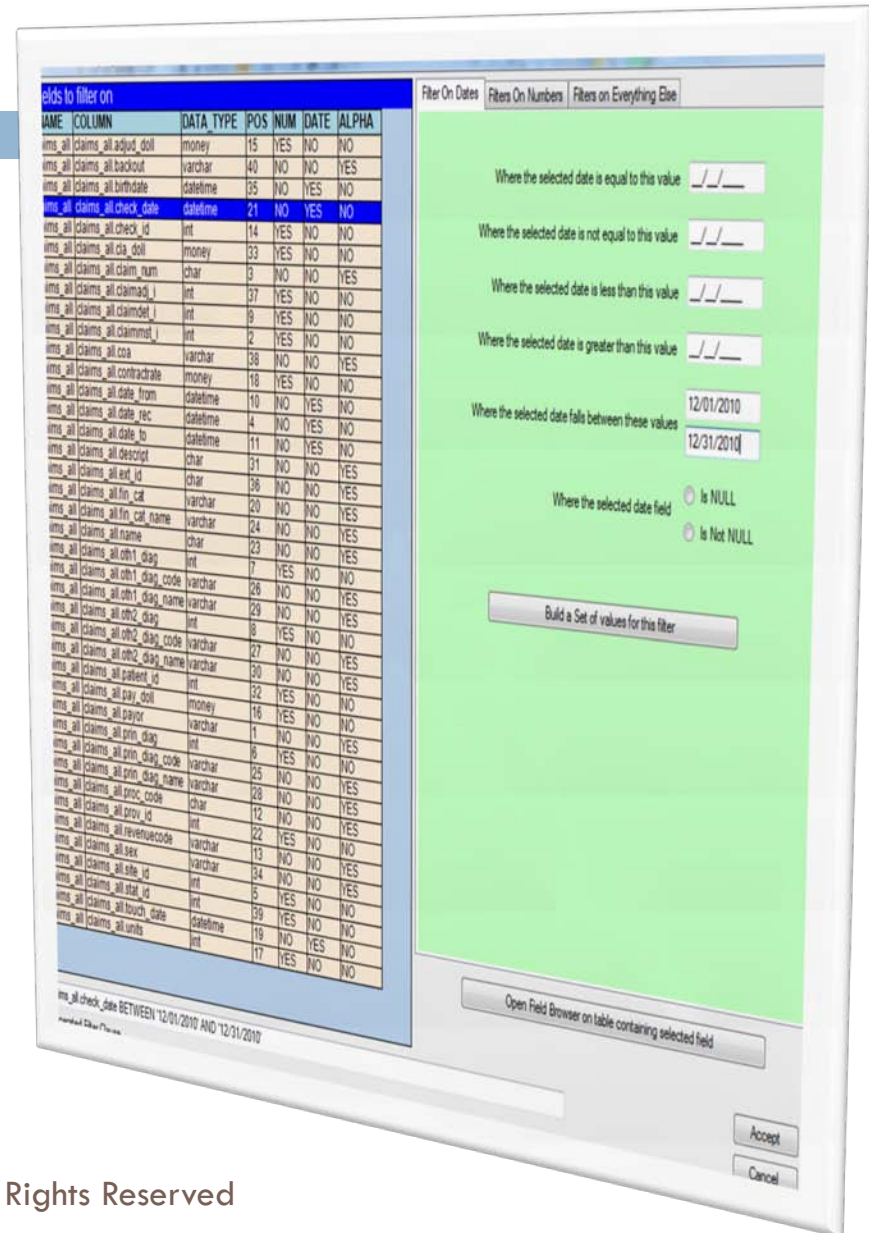
- ❑ SQL “GROUP BY” listing selected fields will be added to the query

```
GROUP BY  
claims_all.claim_num,claims_all.date_from,claims_all.date_to,
```

Add Filter

18

- Filtering allows you to narrow down your results set. For example, you may only be interested in claims that were paid in the month of December 2010.
- You can also filter on number fields or fields containing certain values.
- Filters translate to SQL “WHERE” clause.



Add Constraint

19

- This function works on the field(s) you've created with the Add Synthetic Field button.
- It lets you sets limits or constraints on the values of the created field.
- For example, if you're interested in claims whose payment is \$5000

Add a constraint to the query

Aggregation Fields to build the constraint against

Aggregation to constrain
(claims_all.adjud_doll+claims_all.pay_doll)

Where the selected number is equal to this value

Where the selected number is not equal to this value

Where the selected number is less than this value

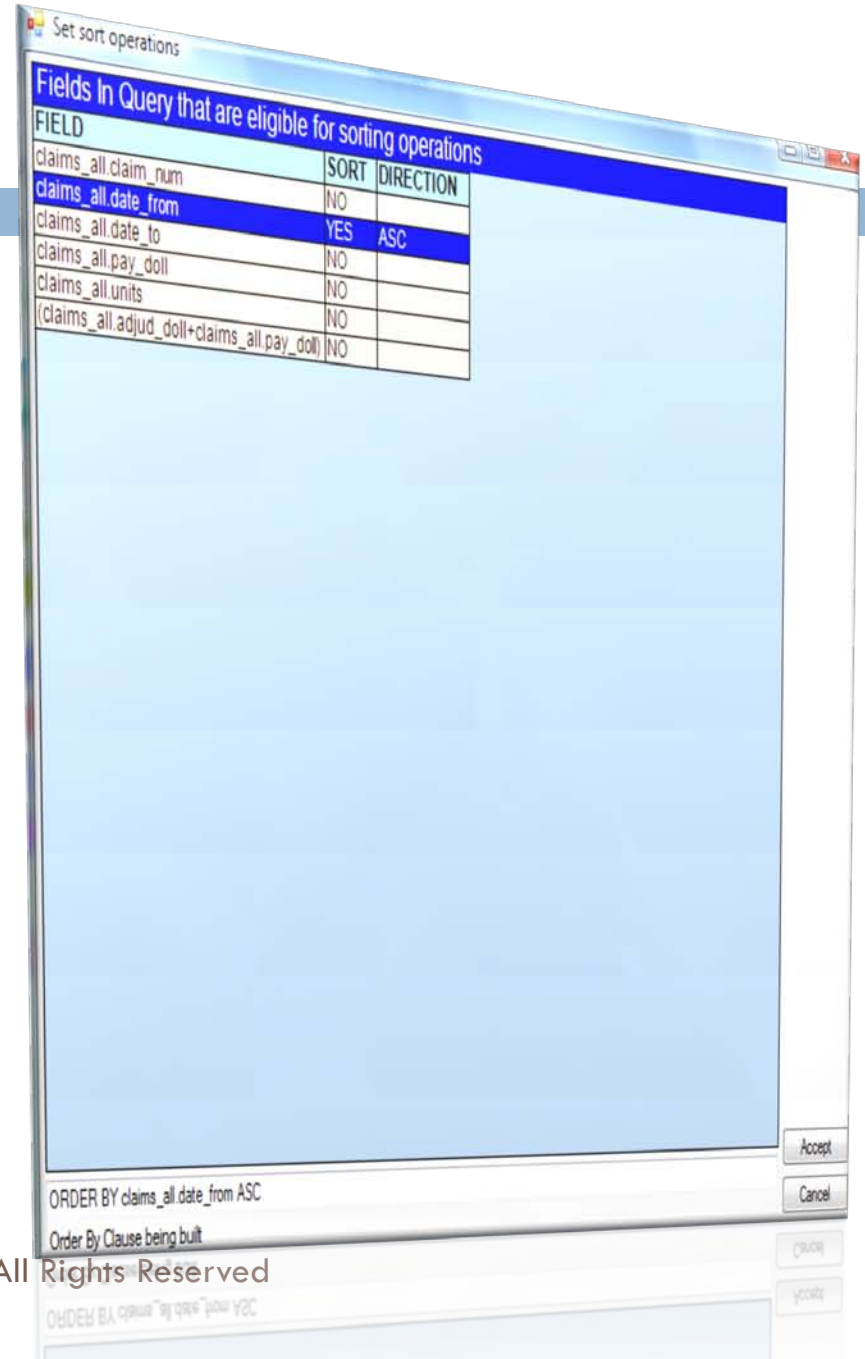
Where the selected number is greater than this value

Where the selected number falls between these values

Add Sorting Ops

20

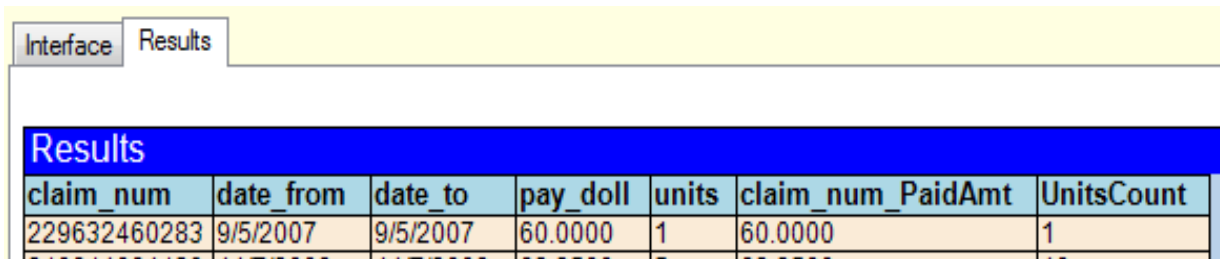
- ❑ Sorting helps you to force your results set to be arranged or ordered by certain fields and also in either ascending or descending.
- ❑ For example, you may want to display the results by member's last name in the ascending order or claims by service date.
- ❑ This function translates to SQL "ORDER BY".



Execute Query

21

- This button is self explanatory – clicking it will run the query that was created.
- The outcome/results will be displayed in the area under the Results tab.



The screenshot shows a software interface with two tabs: 'Interface' and 'Results'. The 'Results' tab is active, displaying a table with the following data:

claim_num	date_from	date_to	pay_doll	units	claim_num_PaidAmt	UnitsCount
229632460283	9/5/2007	9/5/2007	60.0000	1	60.0000	1

Saving queries

22

- Once you have created a query that works (gives you what you need), you can save it and run it as required with possibly changing some filters such as check dates, etc.
- To Save a query, click on File then select a folder that may be designated for queries.
- You can then File/Open at some later date and just Execute the saved query to produce the desired report.