

Vultar Systems

Vultar T24 Data Access Components

Developer guide

Contents

Overview	3
Installation.....	5
Vultar T24 Data Access Server	6
TAFC Connector	7
Database DIRECT Connector	9
Oracle DIRECT Connector	9
MSSQL DIRECT Connector	10
DB2 DIRECT Connector	11
TAFJ Connector	12
Vultar JDBC Driver	12
Vultar ODBC Driver	14
Administrator tool	16
Usage.....	17
Queries	17
Simple queries	17
Where clause	18
Order by clause.....	19
Nested queries (subqueries).....	20
Group by clause	20
Multivalue (MV) support	21
No expansion.....	21
Expanded values	23
Expand subvalues	24
Values & Subvalues number	25
Extracting only one value or subvalue	26
Filter by a MV value	27
Search by MV value	27
Group by a MV value	28
Type conversion	29
Integer conversion.....	29

Float conversion	31
Date conversion.....	32
DateTime conversion.....	33
Character conversion/expansion	33
Multivalued fields conversion.....	35
Local fields.....	36
Query local fields	37
Search by local fields	37
Local fields expansion.....	38
T AFC specific features.....	39
Field joins (Linked fields)	39
User Defined Functions (UDF)	40
Stored Procedures	42
T24 Concat(index) files usage	43
EVAL() function	44
Performance tips	46
Use multiple SPU	46
Multiple CPU/Cores server	46
Fast Hard Disks	46
RAM Disk Usage.....	46
Speed test tool	46
Free version vs Commercial version	46
Technical considerations	47
Failovers	47
Performance.....	47
Logging	47
Troubleshooting	47
Internal error: unable to open database: C:\Windows\Temp\.....	47
JBC code wont compile while starting	48

Overview

This manual describes Vultar T24 Data Access Components for Temenos' T24 systems developed by VULTAR SYSTEMS. The components allow any application using ODBC/JDBC connectivity to access T24 data.

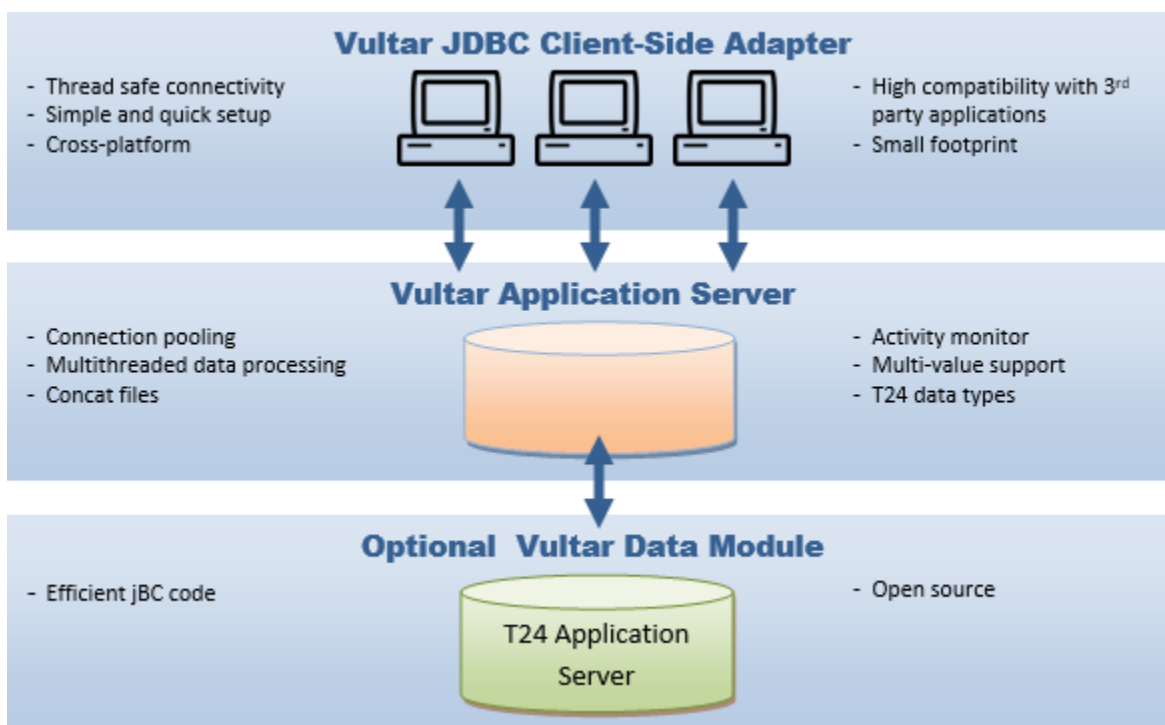
Vultar T24 Data Access Components contains:

- T24 Data Access Server – a high performance server which allows other components to access T24 data
- Vultar Data Module – a set of jBASE routines for accessing data on T24 systems with jBASE backend DB
- ODBC driver – driver for ODBC compliant applications
- JDBC driver – type 3 JDBC driver for JDBC compliant applications

The JDBC component is a type 3 driver, also known as the Net-protocol/pure-Java Driver. Written completely in Java, type 3 drivers are thus platform independent.

Vultar T24 Data Access Components is based upon 3-tier client-server architecture:

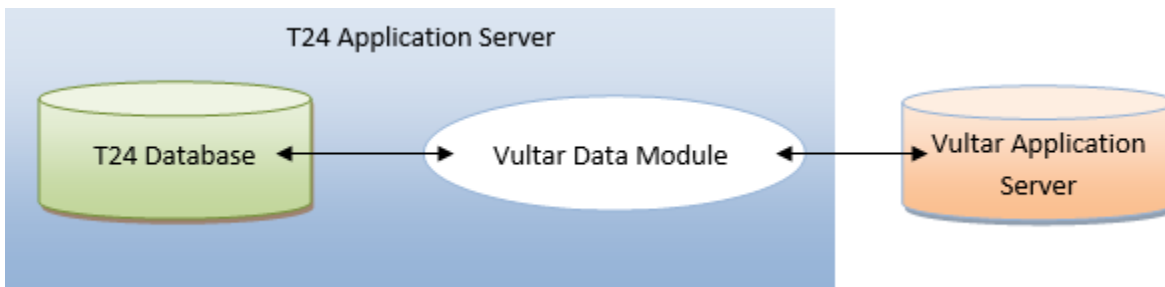
- **Tier 1:** Vultar ODBC/JDBC Client-side adapter
- **Tier 2:** Vultar T24 Data Access Server
- **Tier 3:** Optional Vultar Data Module for T24 (when using TAFC/TAFJ connector)



The driver supports two types of connectors between Vultar Data Access Server and T24:

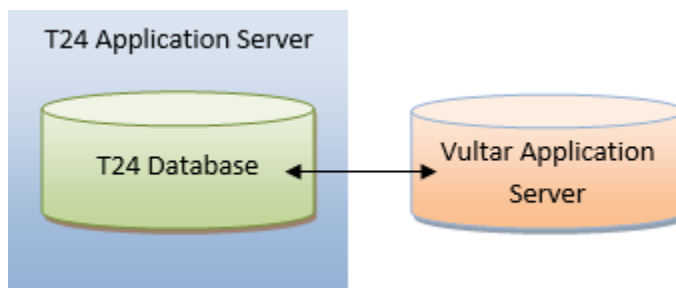
TAFC / TAFJ Connector

This is a universal T24 data access. It uses TAFC (or TAFJ) Temenos Application Framework to collect T24 data using **Vultar Data Module for T24** and send it to **Vultar T24 Data Access Server**. This connector type is compatible with all database backends of T24 - JBASE, ORACLE, MSSQL or DB2. It is written in pure jBC code and is open source.



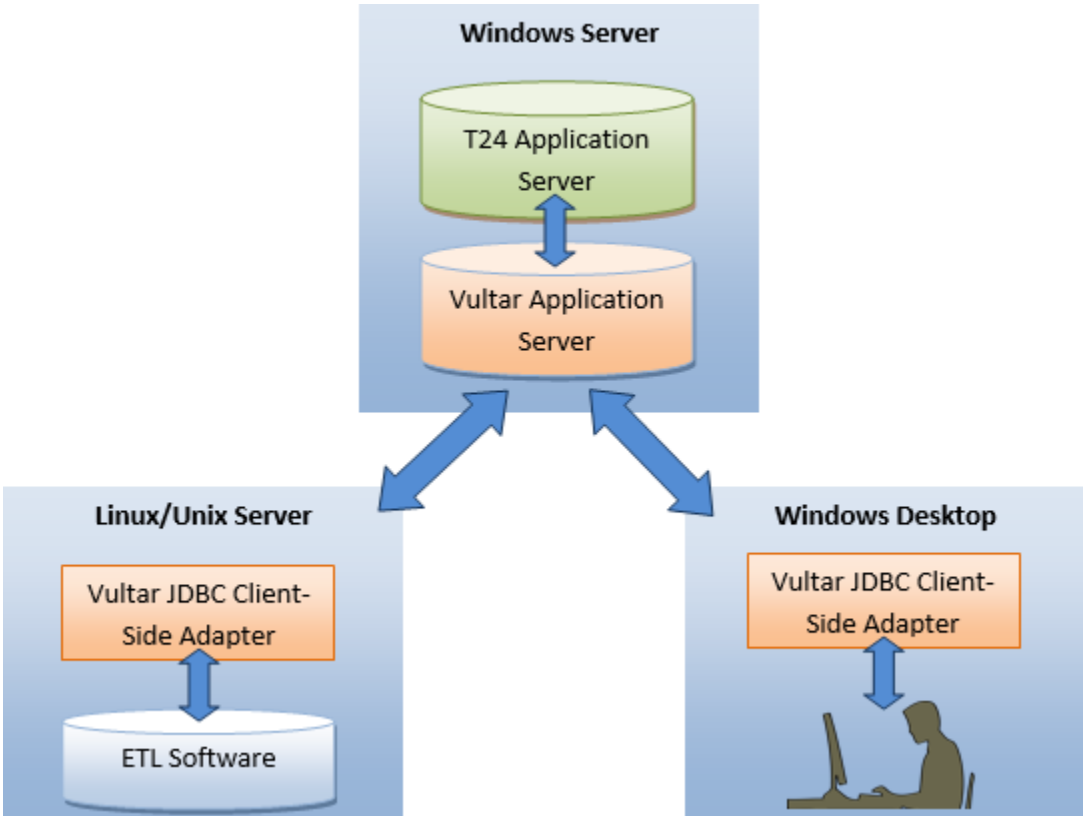
Database DIRECT Connector

With this connector, **Vultar T24 Data Access Server** can connect directly to T24 backend database and query data with much higher speeds compared with TAFC Connector. It works only with MSSQL, ORACLE and DB2 databases.

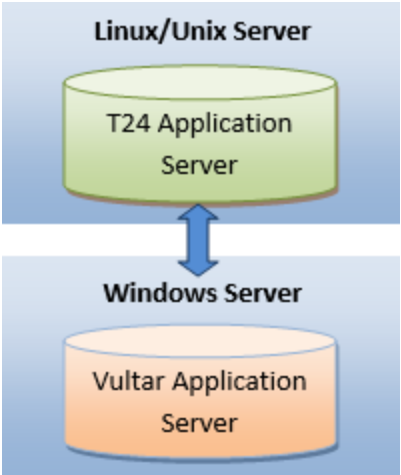


Installation

Vultar T24 Data Access Server and Vultar ODBC/JDBC Client-Side adapter may run in various configurations. The following diagram shows an example of such configuration:



In above example, T24 server is running on a Windows machine. Vultar T24 Data Access Server has been installed and working on the same machine. Vultar ODBC/JDBC Client-Side adapter is cross-platform; hence, T24 data may be served to some ETL software running on UNIX or Linux machine. At the same time, users working on Windows desktops may query T24 data as well using Vultar JDBC Client-Side adapter. In case T24 environment is running on Linux/Unix server, Vultar T24 Data Access Server should be installed on a dedicated Windows machine:



Vultar T24 Data Access Server

Vultar T24 Data Access Server is installed by running the corresponding installation kit *t24das.exe*.

Server will be installed as Windows Service. Besides that, a console version of server will be installed as well.

Vultar T24 Data Access Server installation package contains the following files:

vtconsole.exe	Vultar T24 Data Access Server - console application.
vtservice.exe	Vultar T24 Data Access Server – Windows service application.
config.ini	Vultar T24 Data Access Server parameter file. Will be installed in <i>c:\ProgramData\Vultar</i> folder.
aliases.json	Set up here connectivity details to your T24 database. Will be installed in <i>c:\ProgramData\Vultar</i> folder. More details are provided in the following chapters.

Minimal Windows Server hardware requirements:

- OS: Windows XP
 - RAM: 1GB
 -
1. After installation update the parameters in *config.ini* file as follows:

```
;Server's port number to listen the incoming requests  
;from JDBC client applications  
Port=888  
  
;Server's log level  
;0 - none, 1 - errors, 2 - info, 3 - debug, 5 - full  
LogLevel=2  
  
;Folder for log files  
LogDir=logs  
;Oracle's libraries location  
;OracleLibPath=c:\Oracle12  
  
;Temp folder for cached results  
;By default Window's %TEMP% folder will be used  
;Temp=z:\
```

2. Restart the service or run *vtconsole.exe* console application from Start menu:

```
Background server is running.  
  
Press [Enter] to close the server.
```

After doing changes in [config.ini](#) or *aliases.json* files there is no need to restart the server. The changes will be applied immediately.

TAFC Connector

Minimal requirements: Java 8.

1. Check java version. Connector supports only Java version 8 and above. From command line type:
java -version
2. Define a new alias (or update the existing one) in *aliases.json* (from [Server](#)) and set the following values:
 - a. *Name* – any desired alias name (will be used in connection URL)
 - b. *Type* – “jbase”, used for this type of connectors
 - c. *Host* – IP address/hostname[:port] of the jBASE server and port number specified in *t24client.properties* file (default is 8125)
Note: update your network/firewall configuration to allow connections on this port.
 - d. *User* – shell user, used to automatically start *t24client.jar*
 - e. *Password* – shell user’s password (or passphrase if using PPK (Putty Private Key file))
 - f. *SSH* – use plain telnet protocol or SSH
 - g. *PrivateKey* – path to PPK (Putty Private Key) file, if used to login instead of password
ex. “c:\programdata\Vultar\login.ppk”
Note: do not forget to escape backslashes, i.e. value in *aliases.json* shall look as follows:
“PrivateKey”: “c:\\programdata\\Vultar\\login.ppk”
 - h. *Timeout* – timeout in seconds. Number of seconds to wait for response from server after request. If exceeded – raises error.
3. Copy **t24client.jar**, **t24dac_startup.cmd/t24dac_startup.sh** and **t24client.properties** files to T24 environment *bnk.run* folder (using SCP, FTP or Samba) or use the installation tool **t24clientInstall.exe** included in the kit (in case you have defined shell user and password in alias configuration).
Note: for Unix environment, after copying *t24dac_startup.sh* file, execute *chmod a+x t24dac_startup.sh* command to mark script as executable.
4. Create **t24dac_profile** file by copying system’s profile:
 - a. On Windows: from *remote.cmd* (or *mbremote.cmd*) file
 - b. On Unix: from *.profile* file

Edit **t24dac_profile** file and remove from the end all lines containing calls to loginproc, other executables or exits. This file shall contain only setup for T24 environment to work correctly.

5. In case you did not defined shell user and password, then start **t24client** on T24 server manually by running the script:
 - a. On Windows: `.\t24dac_startup.cmd`
 - b. On Unix: `./t24dac_startup.sh`

```
jsh mbrl0 ~ -->./t24dac_startup.sh
Using properties file: t24client.properties
Aug 17, 2019 9:56:27 PM com.vultar.java.ClientInstall <init>
INFO: TAFC_HOME=/usr/tafcrl0/R10
Loading QuickServer v2.0.0 [PID:14243]... Done

-----
Name : T24 client v1
Port : 8125
-----

Press CTRL-C to stop the server...
█
```

Default listening port is 8125. In case the one would like to use another port; you can set it in *t24client.properties* file.

Note: Update your network/firewall configuration to allow connections on this port.

Note: Preferred way to start **t24client.jar** is to let Server to start it by specifying user and password in alias settings. **t24client.jar** file will auto-update itself when installing new version of T24 Data Access Server.

Database DIRECT Connector

In case you have Oracle/MSSQL/DB2 as T24 backend database – then you can use **direct** connector to respective database instead of [T AFC/T AFJ Data Module](#). This will allow you to avoid socket connection to jBASE, will increase speed and reduce server resources consumption.

Below you will find steps how to setup direct connectors.

Oracle DIRECT Connector

Oracle connector uses Oracle's high performance OCI (Oracle Client Interface) library to extract data directly from Oracle DB without involving T AFC/T AFJ. Download OCI v.12 from <http://vultar.md/files/Oci12.zip>, unzip it and update [config.ini](#) file by setting *OracleLibPath* parameter pointing to unzipped folder.

1. Define a new alias (or update the existing one) in *aliases.json* (from [Server](#)) and set the following values:
 - a. *Name* – any desired alias name (will be used in connection URL)
 - b. *Type* – “Ora”, used for this type of connectors
 - c. *Host* – IP address/hostname of the Oracle server/DB name (ex. “MyOracle/T24DB”)
 - d. *Schema* – DB schema (case sensitive!)
 - e. *User* – user name
Note: user specified here must have SELECT access to following system tables: DBA_TABLES, DBA_INDEXES, USER_OBJECTS, DBA_IND_EXPRESSIONS
 - f. *Password* – user password
Note: password value will be automatically encrypted.
 - g. *Compression* – enables data compression during select (default is *false*). By enabling compression the selection speed can increase significantly, depending on T24/network configuration.
Note: in order to enable compression – download <http://vultar.md/files/t24dacOraCompression.zip> file, unzip it and run the included utility program *t24dacOraCompression.exe*.

Note: If Vultar T24 Data Access Server is installed on Windows XP operating system, version 10 of OCI shall be used instead. Download it from <http://vultar.md/files/Oci10.zip>. Unpack and set the path to it in *OracleLibPath* parameter in [config.ini](#) file.

MSSQL DIRECT Connector

MSSQL Direct Connector uses Microsoft's SQL Native client library to extract data directly from MSSQL DB without involving TAFC/TAFJ.

Before using MSSQL Direct Connector you have to download & install SQL Native Client from [Microsoft SQL Server 2012 Feature Pack](#). In installation instructions you will find links to **sqlncli.msi** files for both x32 and x64 Windows. If you are using Windows x64 –install the x64 package. It will install both x64 and x32 clients.

1. Define a new alias (or update the existing one) in *aliases.json* (from [Server](#)) and set the following values:
 - a. *Name* – any desired alias name (will be used in connection URL)
 - b. *Type* – “MSSQL”, used for this type of connectors
 - c. *Host* – IP address/hostname of the MSSQL server/DB name (ex. “MyMSSQL/T24DB”).
Note: In case of multiple SQL Server instances using on same host, then instance shall be specified with double backslash, ex: “SQLHost\\SQLInstance/T24DB”.
 - d. *Schema* – DB schema (case sensitive!)
 - e. *User* – user name (set empty value “”, in case of using Windows Authentication)
Note: user specified here must have SELECT access to following system tables:
INFORMATION_SCHEMA.TABLES, SYS.INDEXES, SYS.INDEX_COLUMNS, SYS.COLUMNS, SYS.TABLES, SYS.TYPES
 - f. *Password* – user password (set empty value “”, in case of using Windows Authentication).
Note: password value will be automatically encrypted.

DB2 DIRECT Connector

To be implemented in next version.

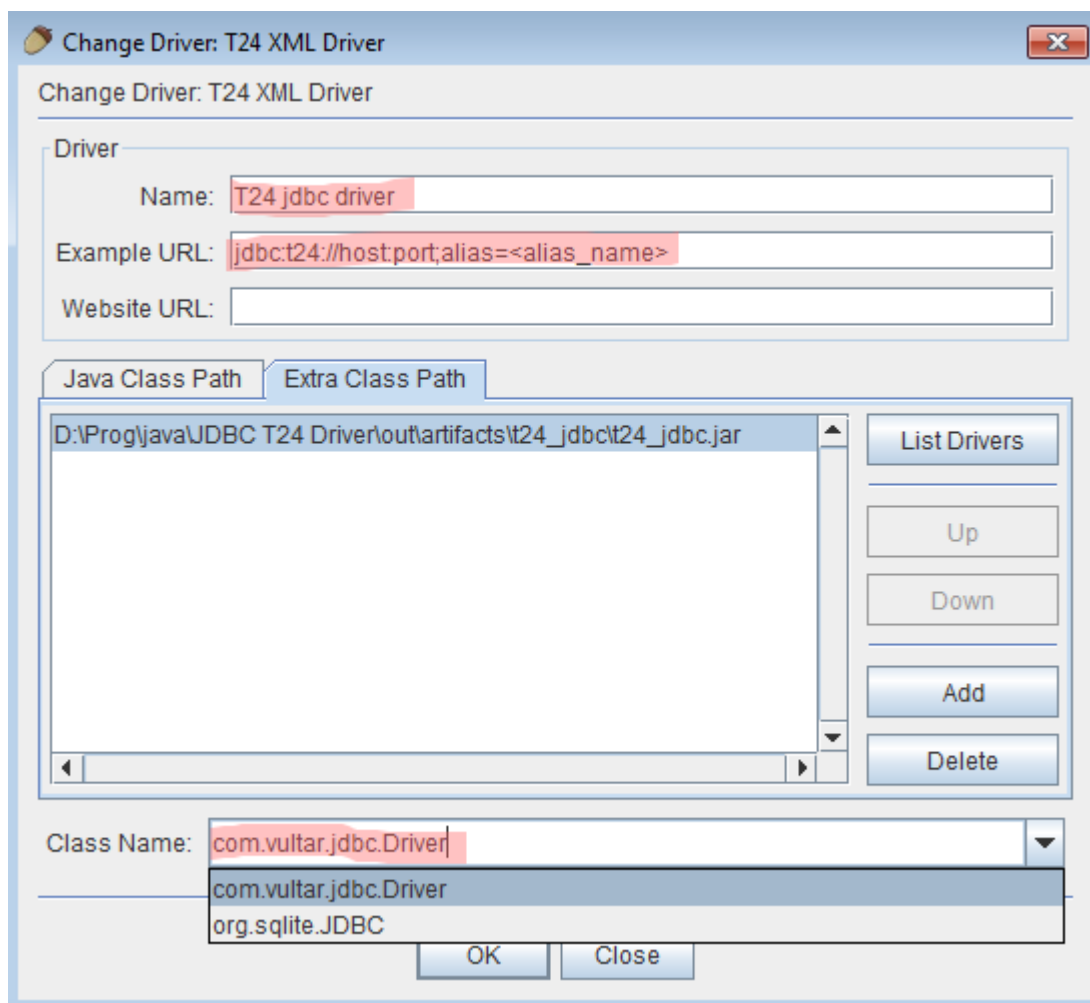
TAFJ Connector

To use Vultar T24 Data Access Components with TAFJ – just configure the corresponding direct connector.

Vultar JDBC Driver

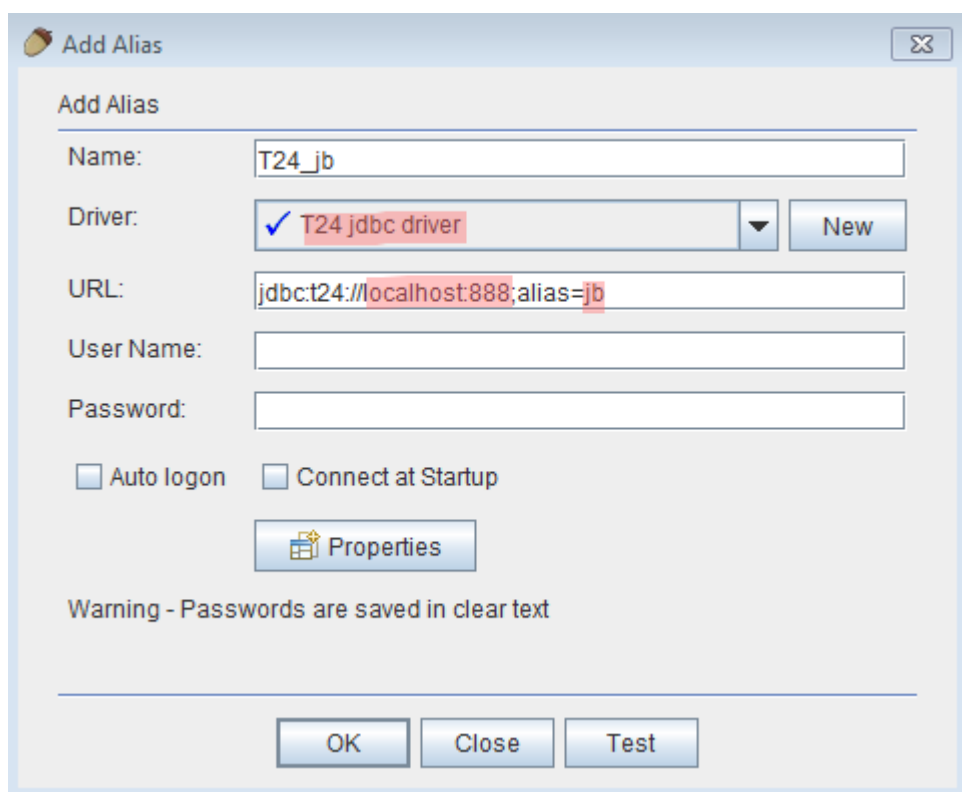
JDBC driver setup will be explained based on free **Squirrel SQL Client** (<http://www.squirrelsql.org/>)

1. Add the driver with Driver manager (Drivers -> Add -> Extra Class Path -> Select & open the **t24_jdbc.jar** file).
2. Set driver's parameters as follows:
 - a. *Name* – driver's name, ex. *T24 jdbc driver*
 - b. *Example URL* – driver's URL template, ex. *jdbc:t24://host:port;alias=<alias_name>*
 - c. *Class name* – driver's class name, *com.vultar.jdbc.Driver*.

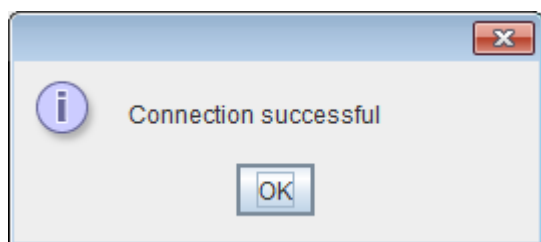


3. Create an alias with the following parameters:

- a. *Name* – any desired name;
- b. *Driver* – driver name defined in previous step
- c. *URL* – connection URL in the following format: `jdbc:t24://<host>:<port>;alias=<alias_name>`, where:
 - *Host* – IP address/hostname of the server where **Vultar T24 Data Access Server** is running;
 - *Port* – port number set in [config.ini](#) of the **Vultar T24 Data Access Server** (default is 888);
 - *Alias_name* – alias name to T24 environment (defined in *aliases.json* config file);

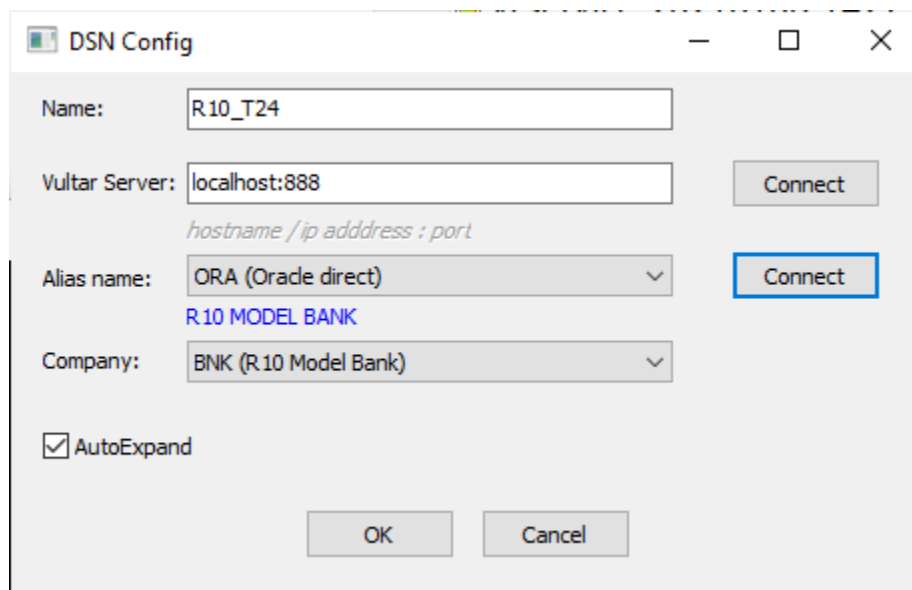


4. Click *Test* ->*Connect* button. Connection should succeed:



Vultar ODBC Driver

1. Run driver installation *t24odbc.exe* which will install both 32 and 64 bit versions.
2. Launch Windows ODBC Data Source Administrator
 - For 32 bit Windows open *c:\Windows\SysWOW64\odbcad32.exe*
 - For 64 bit Windows open *Control Panel -> Administrative Tools -> Data Sources (ODBC)*
3. Set driver's parameters as follows:
 - *Name* – any desired data source name
 - *Vultar Server* – hostname (or IP address) and port number
 - Click *Connect* button to check connectivity and get server's alias list
 - Select alias name which will be used by selected Data Source
 - Click second *Connect* button to connect to T24 environment.



4. Select default company mnemonic (used when selecting from *F_** tables, ex. *F_ACCOUNT*)
5. AutoExpand: if enabled, selected records will get automatically [expanded](#) without specifying the expansion flag.
6. Click OK.
7. You can select data from T24 using the free [WinSQL lite](#) SQL tool.

WinSQL 32bit (Lite)

File Edit View Query Tools Help

R10_T24

Available Query Pages: #1 select * from fb Execute

Query Result Catalog Details

Resultset #1 Messages

	_ID	CUSTOMER	CATEGORY	ACCOUNT_TITLE_1	ACCOUNT_TITLE_2	SHORT_TITLE
1	USD119100001	NULL	11910	Permanent Participations	NULL	Permanent Parti
2	USD120100001	NULL	12010	Commission Receivable	NULL	Comm Recvble
3	USD120110001	NULL	12011	Safekeep Fees Receivable	NULL	Safekeep Fees
4	USD120120001	NULL	12012	Advisory Fees Receivable	NULL	Advisory Fees
5	USD120130001	NULL	12013	Deffered Charges -Accounts	NULL	Deffered Chg-ac
6	USD121050001	NULL	12105	Rent	NULL	Rent
7	USD121100001	NULL	12110	Insurance	NULL	Insurance
8	USD121150001	NULL	12115	Travel	NULL	Travel
9	USD122050001	NULL	12205	Tax Receivable	NULL	Tax Receivable
10	USD122100001	NULL	12210	Sc Tax Reclaimed	NULL	Sctax Reclaimed
11	USD125050001	NULL	12505	Coupon Receivable	NULL	Coupon Recvble
12	USD126100001	NULL	12610	Accounts Receivable	NULL	Accounts Receiv
13	USD126550001	NULL	12655	Sundry Debtors	NULL	Sundry Debtors

Line 1, Pos 26 R10_T24 (R10 MODEL BANK) Not Saved Execution time: 00:06.040

Administrator tool

Installation package contains the administrator tool (vtdrvadm.exe) which allows you to create/delete/update/test connections without editing *aliases.json* file manually.

Usage

Queries

Simple queries

*SELECT * FROM FBNK_ACCOUNT*

Squirrel SQL Client Version 3.7.1

File Drivers Aliases Plugins Session Windows Help

Connect to: jBase_dummy Active Session: 2 - T24_jb ()

Aliases Drivers

SQL

select * from fbnk_account

Limit Rows: 100

select * from fbnk_account

select * from f

Limited to 100 rows; select * from fbnk_account Rows: 1, Cols: 0

Results MetaData Info Overview / Charts Rotated table Results as text

_ID	CUSTOM...	CATEG...	ACCOUNT_TITLE_1	ACCOUNT_TITLE_2	
USD119100001	<null>	11910	Permanent Participations	<null>	Permanent
USD120100001	<null>	12010	Commission Receivable	<null>	Comm Rec
USD120110001	<null>	12011	Safekeep Fees Receivable	<null>	Safekeep F
USD120120001	<null>	12012	Advisory Fees Receivable	<null>	Advisory Fe
USD120130001	<null>	12013	Deffered Charges -Accounts	<null>	Deffered Ch
USD121050001	<null>	12105	Rent	<null>	Rent
USD121100001	<null>	12110	Insurance	<null>	Insurance
USD121150001	<null>	12115	Travel	<null>	Travel
USD122050001	<null>	12205	Tax Receivable	<null>	Tax Receiva
USD122100001	<null>	12210	Sc Tax Reclaimed	<null>	Sc tax Recla
USD125050001	<null>	12505	Coupon Receivable	<null>	Coupon Re
USD126100001	<null>	12610	Accounts Receivable	<null>	Accounts R

/T24_jb 1,27 / 27

Please try out the Tools popup by hitting ctrl+t in the SQL Editor. Do it three times to stop this message.
Please try out the Tools popup by hitting ctrl+t in the SQL Editor. Do it three times to stop this message.

Logs: Errors 2, Warnings 0, Infos 14 42 of 64 MB 0 11:31:20 PM EEST

Where clause

`SELECT _ID, CUSTOMER, CATEGORY, CURRENCY FROM FBNK_ACCOUNT WHERE CURRENCY = 'EUR'`

The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the query: `SELECT _ID, CUSTOMER, CATEGORY, CURRENCY FROM FBNK_ACCOUNT WHERE CURRENCY = 'EUR'`. The results pane displays a table with 15 rows and 4 columns: `_ID`, `CUSTOM...`, `CATEG...`, and `CURR...`. The status bar at the bottom indicates that the query was executed successfully, reading 100 rows in 1.723 seconds.

_ID	CUSTOM...	CATEG...	CURR...
EUR144410001	<null>	14441	EUR
21617	100683	1001	EUR
EUR143050001	<null>	14305	EUR
15307	100357	1001	EUR
14087	100394	1001	EUR
EUR150050001	<null>	15005	EUR
13188	100397	1001	EUR
13487	100295	1001	EUR
EUR144210001	<null>	14421	EUR
17124	100351	6001	EUR
EURUSD140160001	<null>	14016	EUR
EUR100011510	<null>	10001	EUR
42358	100391	1001	EUR

Order by clause

SELECT _ID, CUSTOMER, CATEGORY, CURRENCY FROM FBNK_ACCOUNT WHERE CURRENCY = 'EUR' ORDER BY _ID

The screenshot shows the Squirrel SQL Client interface. The top menu bar includes File, Drivers, Aliases, Plugins, Session, Windows, and Help. The 'Connect to' dropdown is set to 'JBase_dummy', and the 'Active Session' is '2 - T24_jb ()'. The SQL editor contains the query: `SELECT _ID, CUSTOMER, CATEGORY, CURRENCY FROM FBNK_ACCOUNT WHERE CURRENCY = 'EUR' ORDER BY _ID`. The 'Limit Rows' checkbox is checked and set to 100. The 'Results' tab is active, displaying a table with 100 rows. The table has columns: _ID, CUSTOMER, CATEG..., and CURR... The status bar at the bottom shows 'Query 1 of 1, Rows read: 100, Elapsed time (seconds) - Total: 1.877, SQL query: 1.876, Reading results: 0.001' and '11:33:02 PM EEST'.

_ID	CUSTOMER	CATEG...	CURR...
10472	100224	6001	EUR
10561	100382	1001	EUR
10715	100318	1001	EUR
10812	100383	1001	EUR
10917	100319	1001	EUR
10968	100292	1001	EUR
11018	100282	1001	EUR
11061	100363	1001	EUR
11118	100256	1001	EUR
11169	100310	1001	EUR
11266	100296	1001	EUR
11363	100257	1001	EUR
11517	100297	1001	EUR

Nested queries (subqueries)

*SELECT _ID, CURRENCY, WORKING_BALANCE FROM (SELECT * FROM FBNK_ACCOUNT) WHERE (CURRENCY = "EUR" AND WORKING_BALANCE > 0) ORDER BY WORKING_BALANCE DESC, _ID*

Squirrel SQL Client snapshot-20181227_1847

File Drivers Aliases Plugins Session Windows Help

Connect to: R10_Ora_JB Active Session: 1 - R10_Ora_JB ()

Objects SQL

SELECT _ID, CURRENCY, WORKING_BALANCE FROM (SELECT * FROM F...

Limit Rows: 100

SELECT _ID, CURRENCY, WORKING_BALANCE FROM (SELECT * FROM FBNK_ACCOUNT WHERE CURRENCY = 'EUR' AND WORKING_BALANCE > 0 ORDER BY WORKING_BALANCE DESC, _ID

SELECT _ID, CUR SELECT _ID, CUR SELECT _ID, CUR SELECT _ID, CUR

Rows 30; SELECT _ID, CURRENCY, WORKING_BALANCE FROM (SELECT * FROM Selected Rows: 1, Cols: 1

Results MetaData Info Overview / Charts Rotated table Results as text

_ID	CURR...	WORKING_BALANCE
22632	EUR	5.0E7
15466	EUR	1.905E7
EURUSD190220001	EUR	1.095E7
14637	EUR	1769944.31
13935	EUR	1068633.52
28158	EUR	1017559.26
EUR128000002	EUR	1000000.0
EUR165100001	EUR	1000000.0
EUR141960001	EUR	829885.22
11606	EUR	500000.0
22101	EUR	484403.02
28185	EUR	433218.97
15307	EUR	393184.67
13037	EUR	351837.5
23817	EUR	202000.0
EUR143050001	EUR	175724.14
EUR141570001	EUR	142500.0

/R10_Ora_JB Current schema: <None> 4,35 / 158

ErrorCode: 0

Query 1 of 1, Rows read: 30, Elapsed time (seconds) - Total: 2.223, SQL query: 2.196, Reading results: 0.027

Query 1 of 1, Rows read: 30, Elapsed time (seconds) - Total: 1.104, SQL query: 1.104, Reading results: 0

Logs: Errors 0, Warnings 0, Infos 15 71 of 246 MB 0 15:10:32 EEST

Tip: Try to include all filter criteria in the **inner** query, it will reduce the data transfer from T24 to T24DAC Server. Inner query is executed on T24 database, the outer query is executed on T24DAC Server side.

Group by clause

Group by works only with nested queries.

SELECT CURRENCY, SUM(WORKING_BALANCE) AS WB_SUM, COUNT(WORKING_BALANCE) AS WB_CNT FROM

(SELECT CURRENCY, WORKING_BALANCE FROM FBNK_ACCOUNT) **GROUP BY CURRENCY ORDER BY WB_CNT DESC**

Squirrel SQL Client snapshot-20181227_1847

File Drivers Aliases Plugins Session Windows Help

Connect to: R10_Ora_JB Active Session: 1 - R10_Ora_JB ()

Aliases Drivers

Objects SQL

```
SELECT CURRENCY, SUM(WORKING_BALANCE) AS WB_SUM, COUNT(W...
SELECT CURRENCY, SUM(WORKING_BALANCE) AS WB_SUM, COUNT(WORKING_BALANCE) AS WB_CNT FROM
(
  SELECT CURRENCY, WORKING_BALANCE FROM FBNK_ACCOUNT
) GROUP BY CURRENCY ORDER BY WB_CNT DESC
```

Limit Rows: 100

SELECT CURRENCY

Rows 15; SELECT CURRENCY, SUM(WORKING_BALANCE) AS WB_SUM, COUNT(WORK... Selected Rows: 1, Cols: 0

Results MetaData Info Overview / Charts Rotated table Results as text

CURRENCY	WB_SUM	WB_CNT
USD	2.39994306649E9	394
GBP	-2.481991719E7	82
EUR	-4.996696031E7	62
JPY	-6.24926152E8	25
CHF	-948099.21	18
SGD	-4855.4	12
CAD	0.0	6
XAU	1000.0	4
ARS	2000000.0	1
INR	1500000.0	1
PHP	-2000000.0	1
AUD	<null>	0
HKD	<null>	0
PLN	<null>	0
ZAR	<null>	0

/R10_Ora_JB Current schema: <None> 4,41 / 184

Query 1 of 1, Rows read: 30, Elapsed time (seconds) - Total: 1.104, SQL query: 1.104, Reading results: 0
Query 1 of 1, Rows read: 30, Elapsed time (seconds) - Total: 1.1, SQL query: 1.099, Reading results: 0.001
Query 1 of 1, Rows read: 15, Elapsed time (seconds) - Total: 1.121, SQL query: 1.12, Reading results: 0.001

Logs: Errors 0, Warnings 0, Infos 15 106 of 246 MB 0 15:24:10 EEST

Multivalue (MV) support

No expansion

Multivalue fields are treated as string fields with char delimited values.

SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE, PAY_TYPE, PAY_AMT_ORIG FROM FBNK_PD_PAYMENT_DUE

Squirrel SQL Client Version 3.7.1

File Drivers Aliases Plugins Session Windows Help

Connect to: jBase_dummy Active Session: 2 - T24_jb ()

2 - T24_jb ()

Objects SQL

SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE, PAY_TYPE, PAY_AMT_ORI... Limit Rows: 100

SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE, PAY_TYPE, PAY_AMT_ORIG FROM FBK_PD_PAYMENT_DUE

SELECT _ID, FIN

Rows 36; SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE, PAY_TYPE, PAY_AMT_ORIG FROM FBK_P Rows: 1, Cols: 1

Results MetaData Info Overview / Charts Rotated table Results as text

_ID	FINAL_DUE_D...	PAYMENT_DTE_DUE	PAY_TYPE	PAY_AMT...
PDBR0833934267	2008-12-10	20081210	PRüPEüPS	67194.95ü251.98ü28
PDL0833902848	2009-12-05	20090102ý20081226ý20081219ý20081212	INýINýINýIN	1150.68ý1150.68ý1150.6
PDL0833905908	2009-12-05	20090105ý20081206	INüPRüPEüPSýPRüPEüPS	1989.04ü16500ü18.08ü0
PDL0833908726	2008-12-31	20081231ý20081205ý20081204	COüINüPRüPEüPSýCOýCH	173.66ü3562.62ü500136
PDL0833908980	2009-12-05	20090105	PRüPEüPS	200000ü109.59ü10.96
PDL0833919531	2009-12-05	20090105ý20081205	INýIN	12739.73ý12328.77
PDL0833923930	2009-12-05	20081227	IN	1481.64
PDL0833925006	2009-12-05	20090104ý20081204	INýCH	4246.58ý2000
PDL0833929031	2011-12-05	20090106	INüPRüPEüPS	364.38ü2571.42ü1.06ü0.
PDL0833934906	2009-12-05	20090105ý20081205	INýIN	34444.44ý33333.33
PDL0833940387	2009-12-05	20090107	PRüINüPEüPS	7000ü8630.14ü3.84ü0.19
PDL0833940960	2009-01-05	20090105ý20090104ý20090103ý2009010...	INüPRüPEýINýINüPRüPEüPSýINýI...	0.01ü50ü0.05ý0.01ý0.93ü
PDL0833944151	2009-01-05	20090105ý20090104ý20090103ý2009010...	INüPRüPEüPSýINýINüPRüPEüPSý...	0.03ü100ü0.11ü0.01ý0.0...
PDL0833946897	2009-12-05	20090105ý20081205	INýIN	44777.78ý43333.33
PDL0833947029	2009-12-05	20090105ý20081205	INýIN	1019.18ý986.3
PDL0833947202	2009-12-05	20090105ý20081205	INýIN	6888.89ý6666.67

/T24_jb 1,1 / 1

Query 1 of 1, Rows read: 36, Elapsed time (seconds) - Total: 1.699, SQL query: 1.694, Reading results: 0.005

Query 1 of 1, Rows read: 36, Elapsed time (seconds) - Total: 2.551, SQL query: 2.547, Reading results: 0.004

Logs: Errors 2, Warnings 0, Infos 14 53 of 64 MB 0 11:38:18 PM EEST

Expanded values

Expanding values using [EX] operand. Non multivalued fields are replicated. Subvalues are treated as string fields with char delimited subvalues.

```
SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[EX], PAY_TYPE[EX], PAY_AMT_ORIG[EX] FROM  
FBNK_PD_PAYMENT_DUE
```

The screenshot shows the Squirrel SQL Client interface. The query editor contains the following SQL statement:

```
SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[EX], PAY_TYPE[EX], PAY_AMT_ORIG[EX] FROM FBNK_PD_PAYMENT_DUE
```

The query results are displayed in a table with 5 columns: _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE, PAY_TYPE, and PAY_AMT_ORIG. The results show 131 rows, with the first 10 rows displayed. The table is titled "Results" and has a status bar at the bottom indicating "Query 1 of 1, Rows read: 131, Elapsed time (seconds) - Total: 1.9, SQL query: 1.892, Reading results: 0.008".

_ID	FINAL_DUE_DATE	PAYMENT_DTE_DUE	PAY_TYPE	PAY_AMT_ORIG
PDBR0833934267	2008-12-10	20081210	PRüPEüPS	67194.95ü251.98ü28
PDL0833902848	2009-12-05	20090102	IN	1150.68
PDL0833902848	2009-12-05	20081226	IN	1150.68
PDL0833902848	2009-12-05	20081219	IN	1150.68
PDL0833902848	2009-12-05	20081212	IN	1150.68
PDL0833905908	2009-12-05	20090105	INüPRüPEüPS	1989.04ü16500ü18.08ü0.9
PDL0833905908	2009-12-05	20081206	PRüPEüPS	14000ü86.58ü4.33
PDL0833908726	2008-12-31	20081231	COüINüPRüPEüPS	173.66ü3562.62ü500136.99ü959.16ü47.96
PDL0833908726	2008-12-31	20081205	CO	6.94
PDL0833908726	2008-12-31	20081204	CH	1000
PDL0833908980	2009-12-05	20090105	PRüPEüPS	20000ü109.59ü10.96
PDL0833919531	2009-12-05	20090105	IN	12739.73
PDL0833919531	2009-12-05	20081205	IN	12328.77
PDL0833923930	2009-12-05	20081227	IN	1481.64
PDL0833925006	2009-12-05	20090104	IN	4246.58
PDL0833925006	2009-12-05	20081204	CH	2000
PDL0833929031	2011-12-05	20090106	INüPRüPEüPS	364.38ü2571.42ü1.06ü0.11

Expand subvalues

Expanded subvalues using [EX2] operand. Non subvalued fields are replicated.

```
SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[EX], PAY_TYPE[EX2], PAY_AMT_ORIG[EX2] FROM  
FBNK_PD_PAYMENT_DUE
```

Squirrel SQL Client Version 3.7.1

File Drivers Aliases Plugins Session Windows Help

Connect to: jBase_dummy Active Session: 2 - T24_jb ()

SQL

SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[EX], PAY_TYPE[EX2], PAY_AMT_ORIG[EX2] FROM FBNK_PD_PAYMENT_DUE

Limit Rows: 100

Rows 255; SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[EX], PAY_TYPE[EX2], PAY_AMT_ORIG[EX2] FROM FBNK_PD_PAYMENT_DUE

Results Metadata Info Overview / Charts Rotated table Results as text

_ID	FINAL_DUE_DATE	PAYMENT_DTE_DUE	PAY_TYPE	PAY_AMT_ORIG
PDBR0833934267	2008-12-10	20081210	PR	67194.95
PDBR0833934267	2008-12-10	20081210	PE	251.98
PDBR0833934267	2008-12-10	20081210	PS	28
PDL0833902848	2009-12-05	20090102	IN	1150.68
PDL0833902848	2009-12-05	20081226	IN	1150.68
PDL0833902848	2009-12-05	20081219	IN	1150.68
PDL0833902848	2009-12-05	20081212	IN	1150.68
PDL0833905908	2009-12-05	20090105	IN	1989.04
PDL0833905908	2009-12-05	20090105	PR	16500
PDL0833905908	2009-12-05	20090105	PE	18.08
PDL0833905908	2009-12-05	20090105	PS	0.9
PDL0833905908	2009-12-05	20081206	PR	14000
PDL0833905908	2009-12-05	20081206	PE	86.58
PDL0833905908	2009-12-05	20081206	PS	4.33
PDL0833908726	2008-12-31	20081231	CO	173.66
PDL0833908726	2008-12-31	20081231	IN	3562.62
PDL0833908726	2008-12-31	20081231	PR	500136.99

/T24_jb 1,1 / 1

Query 1 of 1, Rows read: 131, Elapsed time (seconds) - Total: 1.9, SQL query: 1.892, Reading results: 0.008

Query 1 of 1, Rows read: 255, Elapsed time (seconds) - Total: 1.911, SQL query: 1.895, Reading results: 0.016

Logs: Errors 2, Warnings 0, Infos 14 62 of 64 MB 0 11:44:58 PM EEST

Values & Subvalues number

Selecting values and subvalues numbers using [VN] and [SN] operands;

```
SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[VN] as P_DTE_DUE_VN, PAYMENT_DTE_DUE[EX],  
PAY_TYPE[VN] AS PAY_TYPE_VN, PAY_TYPE[SN] AS PAY_TYPE_SN, PAY_TYPE[EX2], PAY_AMT_ORIG[VN] AS  
PAY_AMT_ORIG_VN, PAY_AMT_ORIG[SN] AS PAY_AMT_ORIG_SN, PAY_AMT_ORIG[EX2] FROM  
FBNK_PD_PAYMENT_DUE
```

The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the following query:

```
SELECT _ID, FINAL_DUE_DATE, PAYMENT_DTE_DUE[VN] as P_DTE_DUE_VN, PAYMENT_DTE_DUE[EX],  
PAY_TYPE[VN] AS PAY_TYPE_VN, PAY_TYPE[SN] AS PAY_TYPE_SN, PAY_TYPE[EX2],  
PAY_AMT_ORIG[VN] AS PAY_AMT_ORIG_VN, PAY_AMT_ORIG[SN] AS PAY_AMT_ORIG_SN, PAY_AMT_ORIG[EX2]  
FROM FBNK_PD_PAYMENT_DUE
```

The query results are displayed in a table with the following columns: _ID, FINAL_DUE_DATE, P_..., PAYMENT_..., PA..., PAY..., PAY..., PAY_A..., PAY..., PAY_AMT_ORIG. The table contains 255 rows of data.

_ID	FINAL_DUE_DATE	P_...	PAYMENT_...	PA...	PAY...	PAY...	PAY_A...	PAY...	PAY_AMT_ORIG
PDBR0833934267	2008-12-10	1	20081210	1	1	PR	1	1	67194.95
PDBR0833934267	2008-12-10	1	20081210	1	2	PE	1	2	251.98
PDBR0833934267	2008-12-10	1	20081210	1	3	PS	1	3	28
PDL0833902848	2009-12-05	1	20090102	1	1	IN	1	1	1150.68
PDL0833902848	2009-12-05	2	20081226	2	1	IN	2	1	1150.68
PDL0833902848	2009-12-05	3	20081219	3	1	IN	3	1	1150.68
PDL0833902848	2009-12-05	4	20081212	4	1	IN	4	1	1150.68
PDL0833905908	2009-12-05	1	20090105	1	1	IN	1	1	1989.04
PDL0833905908	2009-12-05	1	20090105	1	2	PR	1	2	16500
PDL0833905908	2009-12-05	1	20090105	1	3	PE	1	3	18.08
PDL0833905908	2009-12-05	1	20090105	1	4	PS	1	4	0.9
PDL0833905908	2009-12-05	2	20081206	2	1	PR	2	1	14000
PDL0833905908	2009-12-05	2	20081206	2	2	PE	2	2	86.58
PDL0833905908	2009-12-05	2	20081206	2	3	PS	2	3	4.33

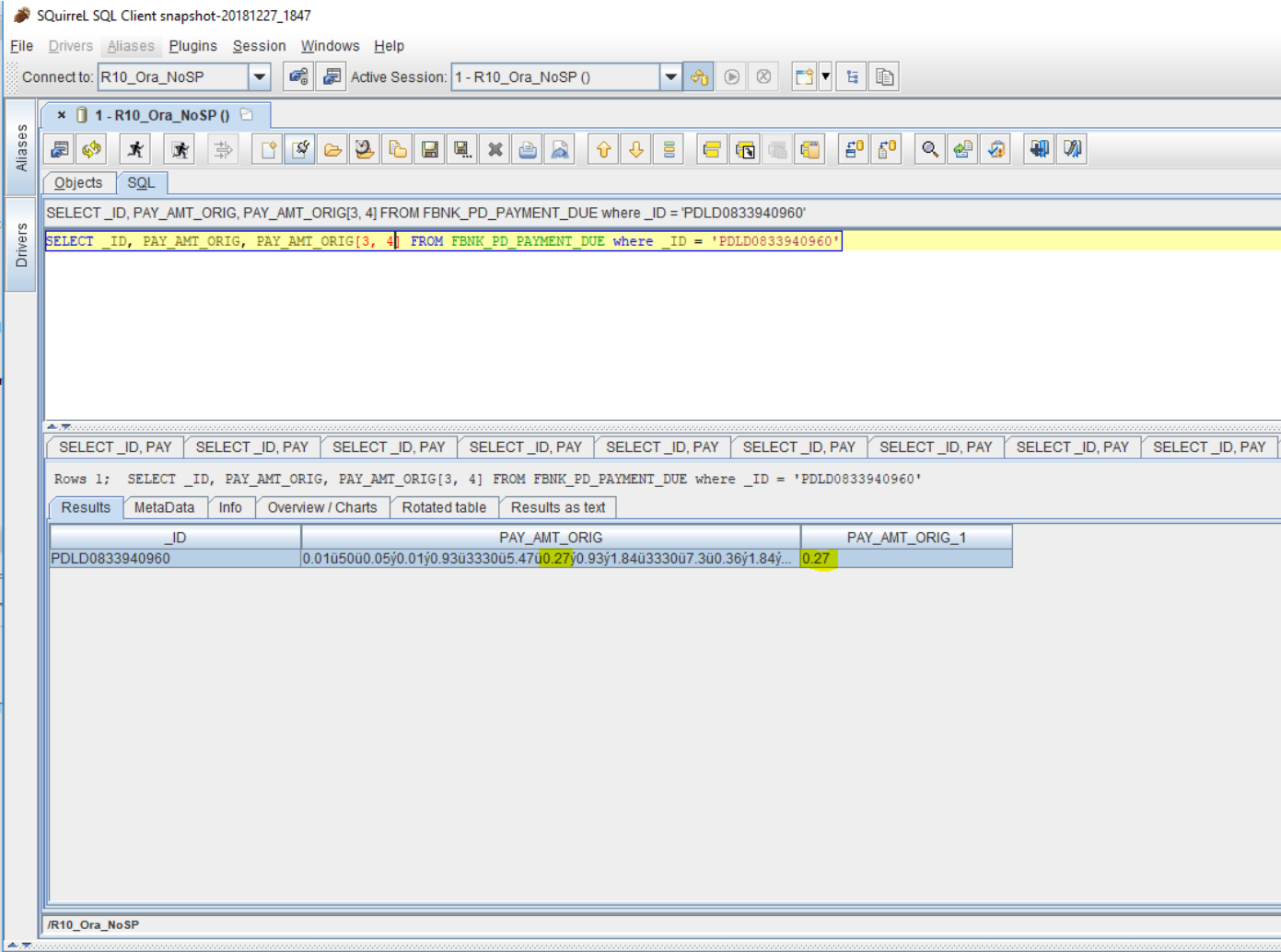
The status bar at the bottom indicates: Error Code: 0, Query 1 of 1, Rows read: 255, Elapsed time (seconds): - Total: 1.946, SQL query: 1.937, Reading results: 0.00%, Logs: Errors 2, Warnings 0, Infos 14, 33 of 68 MB, 11:50:01 PM EEST.

Extracting only one value or subvalue

You can specify the exact value number [<vn>] (in case only one value is required) or value & subvalue number [<vn>, <sn>] (in case only one specific subvalue is required).

```
SELECT _ID, PAY_AMT_ORIG, PAY_AMT_ORIG[3, 4] FROM FBNK_PD_PAYMENT_DUE where _ID = 'PDLD0833940960'
```

In this example the 4th subvalue of 3rd value from field PAY_AMT_ORIG is extracted.



Filter by a MV value

To filter by a multivalue you need to use [subquery](#). In the below example in the inner query the multivalue fields are [expanded](#) and then only PR payment types are selected having payment value greater than 10,000

```
SELECT * FROM (SELECT_ID, PAYMENT_DTE_DUE[EX, D], PAY_TYPE[EX2], PAY_AMT_ORIG[EX2,F] FROM  
FBNK_PD_PAYMENT_DUE) WHERE PAY_TYPE = 'PR' AND PAY_AMT_ORIG > 10000 ORDER BY PAY_AMT_ORIG DESC
```

The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the following query:

```
SELECT * FROM  
(  
  SELECT_ID, PAYMENT_DTE_DUE[EX, D],  
  PAY_TYPE[EX2], PAY_AMT_ORIG[EX2,F] FROM FBNK_PD_PAYMENT_DUE  
)  
WHERE PAY_TYPE = 'PR' AND PAY_AMT_ORIG > 10000 ORDER BY PAY_AMT_ORIG DESC
```

The query results are displayed in a table with the following data:

_ID	PAYMENT_DTE...	PAY_TYPE	PAY_AMT_ORIG
PDL0900678090	2009-01-07	PR	1000000.0
PDL0833908726	2008-12-31	PR	500136.99
PDL0833908980	2009-01-05	PR	200000.0
PDL0833953106	2008-12-05	PR	100000.0
PDTF083398704201	2009-01-02	PR	100000.0
PDBR0833934267	2008-12-10	PR	67194.95
PDL0833905908	2009-01-05	PR	16500.0
PDL0833905908	2008-12-06	PR	14000.0

The status bar at the bottom indicates: Query 1 of 1, Rows read: 8, Elapsed time (seconds) - Total: 1.121, SQL query: 1.121, Reading results: 0. The bottom right corner shows the time 15:45:05 EEST.

Search by MV value

You can include MV search criteria in WHERE clause of your SQL statement.

```
SELECT_ID, CURRENCY, LOCAL_REF[EX] FROM FBNK_MM_MONEY_MARKET  
WHERE LOCAL_REF = 'DCD.MM.SY0834045334.0001'
```

Will return the records, where MV field LOCAL.REF contains value of 'DCD.MM.SY0834045334.0001'

<none> Script-2

<Oracle - r10orcl> Ora scripts.sql

*<T24 R10 Ora> Script-1 ×

SELECT _ID, CURRENCY, LOCAL_REF[EX] FROM FBNK_MM_MONEY_MARKET

WHERE LOCAL_REF = 'DCD.MM.SY0834045334.0001'

FBNK_MM_MONEY_MARKET 1 ×

SELECT _ID, CURRENCY, LOCAL_REF[EX] FROM FBNK_MM

Enter a SQL expression to filter results (use Ctrl

Grid

1

MM0834000033

USD

SY0834045334

2

MM0834000033

USD

DCD.MM.SY0834045334.0001

3

MM0834000033

USD

YES

Group by a MV value

You can group by a MV value using [subqueries](#) and field [expansion](#). In below example records from PD_PAYMENT_DUE table are expanded in inner query and grouped in outer query.

```
SELECT PAY_TYPE, SUM(PAY_AMT_ORIG) AS PAY_AMT_ORIG_SUM FROM (SELECT PAY_TYPE[EX2],
PAY_AMT_ORIG[EX2,F] FROM FBNK_PD_PAYMENT_DUE) WHERE PAY_TYPE <> " GROUP BY PAY_TYPE ORDER BY
PAY_TYPE
```

28 | Page

Squirrel SQL Client snapshot-20181227_1847

File Drivers Aliases Plugins Session Windows Help

Connect to: R10_Ora_JB Active Session: 1 - R10_Ora_JB ()

Aliases Drivers

SQL

SELECT PAY_TYPE, SUM(PAY_AMT_ORIG) AS PAY_AMT_ORIG_SUM FROM ...

Limit Rows: 100

SELECT PAY_TYPE, SUM(PAY_AMT_ORIG) AS PAY_AMT_ORIG_SUM FROM
(
SELECT PAY_TYPE[EX2], PAY_AMT_ORIG[EX2,F] FROM FBK_PD_PAYMENT_DUE
)
WHERE PAY_TYPE <> '' GROUP BY PAY_TYPE ORDER BY PAY_TYPE

SELECT PAY_TYPE

Rows 6; SELECT PAY_TYPE, SUM(PAY_AMT_ORIG) AS PAY_AMT_ORIG_SUM FROM Selected Rows: 0, Cols: 0

Results MetaData Info Overview / Charts Rotated table Results as text

PAY_TYPE	PAY_AMT_ORIG_SUM
CH	3100.0
CO	180.6
IN	614914.65
PE	3332.89
PR	2164546.21
PS	303.15

/R10_Ora_JB Current schema: <None> 5,57 / 192

Query 1 of 1, Rows read: 6, Elapsed time (seconds) - Total: 1.119, SQL query: 1.119, Reading results: 0
Query 1 of 1, Rows read: 6, Elapsed time (seconds) - Total: 1.096, SQL query: 1.096, Reading results: 0
Query 1 of 1, Rows read: 6, Elapsed time (seconds) - Total: 1.263, SQL query: 1.263, Reading results: 0

Logs: Errors 0, Warnings 0, Infos 15 128 of 364 MB 16:16:39 EEST

Type conversion

Driver supports some standard T24 field types like *IN2CUS*, *IN2YM*, *IN2ACC*, *IN2CAT*, *IN2D*, *IN2AMT*, *IN2DEC*, *IN2R*. All other T24 field types are treated as *VARCHAR* type. In case a type conversion is needed for other field types – extra operands can be used.

Integer conversion

Integer conversion with [I] operand.

```
SELECT _ID, CATEGORY, CATEGORY[I] AS CATEG_INT FROM FBK_ACCOUNT
```

Squirrel SQL Client Version 3.7.1

File Drivers Aliases Plugins Session Windows Help

Connect to: jBase_dummy Active Session: 2 - T24_jb ()

Aliases Drivers

2 - T24_jb ()

Objects SQL

SELECT _ID, CATEGORY, CATEGORY[I] AS CATEG_INT FROM FBNK_ACCOUNT

Limit Rows: 100

SELECT _ID, CATEGORY, CATEGORY[I] AS CATEG_INT FROM FBNK_ACCOUNT

SELECT _ID, FIN SELECT _ID, FIN SELECT _ID, FIN SELECT _ID, FIN SELECT _ID, CAT SELECT _ID, CAT

Limited to 100 rows; SELECT _ID, CATEGORY, CATEGORY[I] AS CATEG_INT FROM FBNK_ACCOUNT Rows: 1, Cols: 1

Results MetaData Info Overview / Charts Rotated table Results as text

ColumnIndex	getColumnName	getColumnTypeName	getPrecision	getScale
1	ID	Varchar	19	0
2	CATEGORY	Varchar	6	0
3	CATEG_INT	Integer	35	0

/T24_jb 1,65 / 65

Query 1 of 1, Rows read: 100, Elapsed time (seconds) - Total: 1.781, SQL query: 1.779, Reading results: 0.002

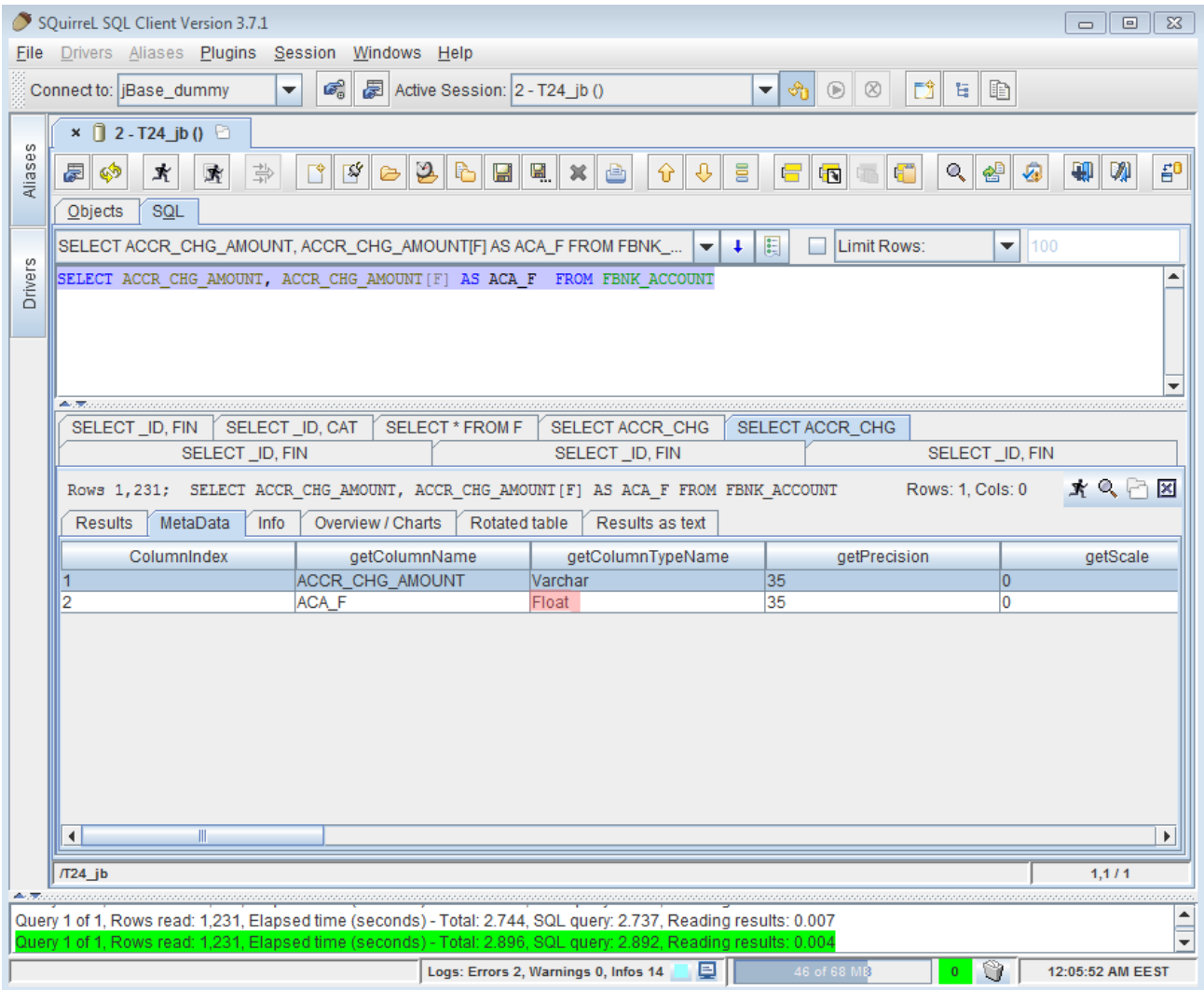
Query 1 of 1, Rows read: 100, Elapsed time (seconds) - Total: 1.688, SQL query: 1.687, Reading results: 0.001

Logs: Errors 2, Warnings 0, Infos 14 45 of 68 MB 0 11:57:07 PM EEST

Float conversion

Float conversion with [F] operand.

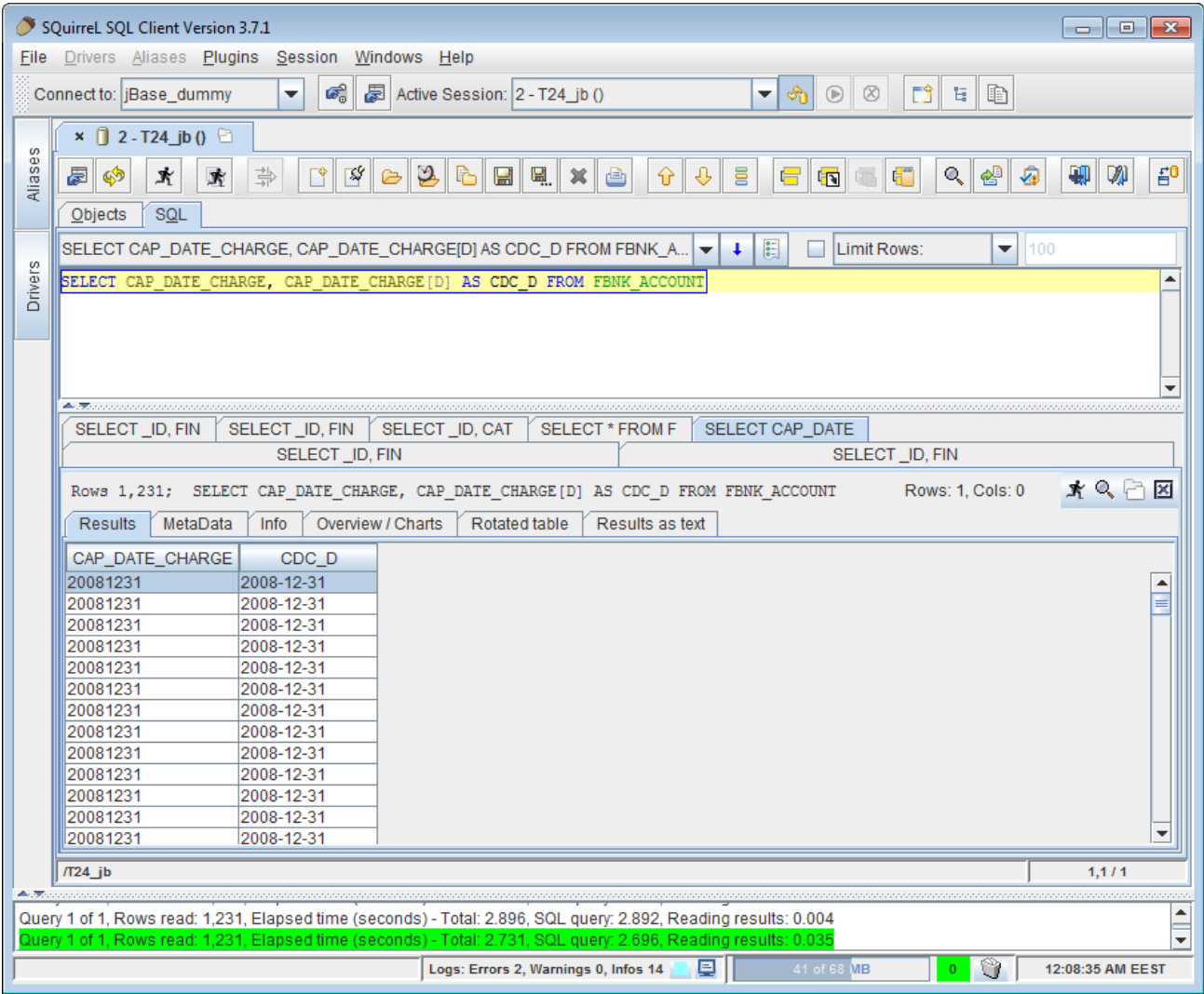
```
SELECT ACCR_CHG_AMOUNT, ACCR_CHG_AMOUNT[F] AS ACA_F FROM FBNK_ACCOUNT
```



Date conversion

Date conversion with [D] operand.

```
SELECT CAP_DATE_CHARGE, CAP_DATE_CHARGE[D] AS CDC_D FROM FBNK_ACCOUNT
```



DateTime conversion

DateTime conversion with [D.YYMMDDHHNN] operand.

```
SELECT _ID, DATE_TIME, DATE_TIME[D.YYMMDDHHNN] FROM FBNK_ACCOUNT
```

The screenshot shows the Squirrel SQL Client interface. The title bar indicates the session is 'Squirrel SQL Client snapshot-20181227_1847'. The 'Connect to' field shows 'R10_Ora_JB' and the 'Active Session' is '1 - R10_Ora_JB ()'. The 'SQL' tab is active, displaying the query: `SELECT _ID, DATE_TIME, DATE_TIME[D.YYMMDDHHNN] FROM FBNK_ACCOUNT`. The 'Limit Rows' is set to 100. The 'Results' tab shows a table with 3 columns: `_ID`, `DATE_TIME`, and `DATE_TIME_1`. The table contains 10 rows of data, all starting with 'USD' and dates from 2009-03-17. The status bar at the bottom shows 'Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 1.1, SQL query: 1.099, Reading results: 0.001'.

_ID	DATE_TIME	DATE_TIME_1
USD119100001	0903171748	2009-03-17 17:48:00.0
USD120100001	0903171747	2009-03-17 17:47:00.0
USD120110001	0903171740	2009-03-17 17:40:00.0
USD120120001	0903171748	2009-03-17 17:48:00.0
USD120130001	0903171749	2009-03-17 17:49:00.0
USD121050001	0903171749	2009-03-17 17:49:00.0
USD121100001	0903171749	2009-03-17 17:49:00.0
USD121150001	0903171747	2009-03-17 17:47:00.0
USD122050001	0903171747	2009-03-17 17:47:00.0
USD122100001	0903171747	2009-03-17 17:47:00.0

Character conversion/expansion

Sometimes field's values can have more data than the length specified in STANDARD.SELECTION table. In this case you can increase field's size with [C.<nnn>] operand.

```
SELECT MNEMONIC, MNEMONIC[C.100] AS M_100 FROM FBNK_ACCOUNT
```

Squirrel SQL Client Version 3.7.1

File Drivers Aliases Plugins Session Windows Help

Connect to: jBase_dummy Active Session: 2 - T24_jb ()

Aliases Drivers

SQL

SELECT MNEMONIC, MNEMONIC[C.100] AS M_100 FROM FBANK_ACCOUNT

Limit Rows: 100

SELECT MNEMONIC, MNEMONIC[C.100] AS M_100 FROM FBANK_ACCOUNT

SELECT_ID, FIN SELECT_ID, CAT SELECT * FROM F SELECT CAP_DATE SELECT MNEMONIC

SELECT_ID, FIN SELECT_ID, FIN SELECT_ID, FIN

Rows 1,231; SELECT MNEMONIC, MNEMONIC[C.100] AS M_100 FROM FBANK_ACCOUNT Rows: 1, Cols: 0

Results MetaData Info Overview / Charts Rotated table Results as text

ColumnIndex	getColumnName	getColumnTypeName	getPrecision	getScale
1	MNEMONIC	Varchar	10	0
2	M_100	Varchar	100	0

/T24_jb 1,60 / 60

Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 2.731, SQL query: 2.696, Reading results: 0.035

Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 2.724, SQL query: 2.718, Reading results: 0.006

Logs: Errors 2, Warnings 0, Infos 14 42 of 68 MB 0 12:12:03 AM EEST

Multivalued fields conversion

In case a multivalued field needs expansion and conversion – a combination of expansion and conversion operands can be used.

`SELECT _ID, PAYMENT_DTE_DUE, PAYMENT_DTE_DUE[EX, D] FROM FBK_PD_PAYMENT_DUE`

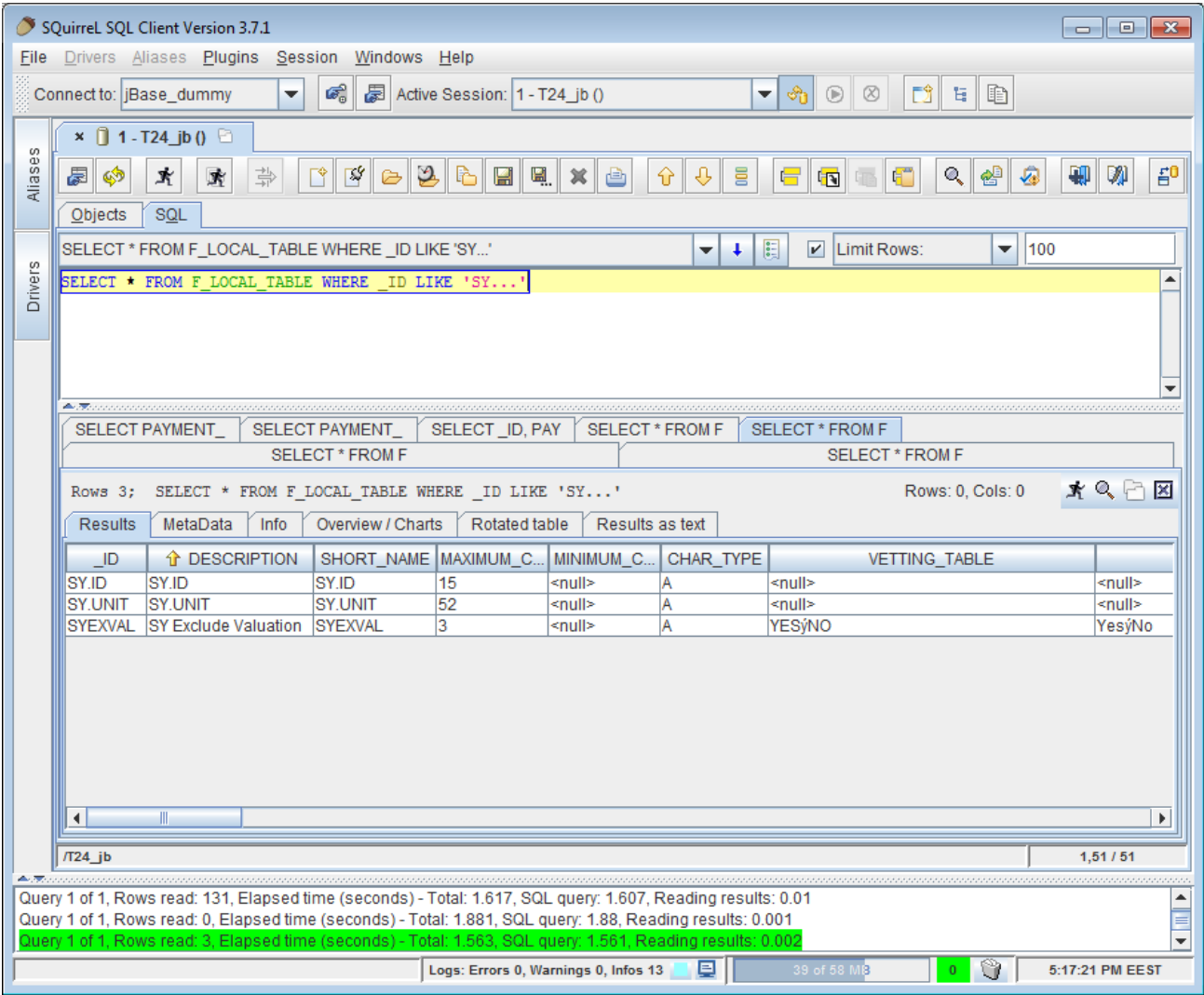
The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the query: `SELECT _ID, PAYMENT_DTE_DUE, PAYMENT_DTE_DUE[EX, D] FROM FBK_PD_PAYMENT_DUE`. The results pane displays a table with three columns: `_ID`, `PAYMENT_DTE_DUE`, and `PAYMENT_DTE_D...`. The table contains 13 rows of data. The status bar at the bottom indicates that the query was executed successfully, with 131 rows read and 0.006 seconds elapsed.

_ID	PAYMENT_DTE_DUE	PAYMENT_DTE_D...
PDBR0833934267	20081210	2008-12-10
PDL0833902848	20090102y20081226y20081219y20081212	2009-01-02
PDL0833902848	20090102y20081226y20081219y20081212	2008-12-26
PDL0833902848	20090102y20081226y20081219y20081212	2008-12-19
PDL0833902848	20090102y20081226y20081219y20081212	2008-12-12
PDL0833905908	20090105y20081206	2009-01-05
PDL0833905908	20090105y20081206	2008-12-06
PDL0833908726	20081231y20081205y20081204	2008-12-31
PDL0833908726	20081231y20081205y20081204	2008-12-05
PDL0833908726	20081231y20081205y20081204	2008-12-04
PDL0833908980	20090105	2009-01-05
PDL0833919531	20090105y20081205	2009-01-05
PDL0833919531	20090105y20081205	2008-12-05

Local fields

Driver supports local fields selection.

In the following example *FBNK_MM_MONEY_MARKET* table has 3 local fields defined in LOCAL.TABLE application.



Query local fields

Local fields are selected as normal table fields.

```
SELECT _ID, SY_ID, SY_UNIT, SYEXVAL FROM FBNK_MM_MONEY_MARKET
```

The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the query: `SELECT _ID, SY_ID, SY_UNIT, SYEXVAL FROM FBNK_MM_MONEY_MARKET`. The query has been executed, and the results are displayed in a table with 4 columns: `_ID`, `SY_ID`, `SY_UNIT`, and `SYEXVAL`. The table contains 14 rows of data. The status bar at the bottom indicates that the query was successful, with 74 rows read and an elapsed time of 1.597 seconds.

_ID	SY_ID	SY_UNIT	SYEXVAL
MM0833989610	<null>	<null>	<null>
MM0834000025	SY0834086928y...	OTCGROI.Deposit.SY0834086928.0001yOTCGROI.Deposit2...	YESyNO
MM0834000026	SY0834042600	OTCCRAN.Deposit.SY0834042600.0001	YES
MM0834000027	SY0834078080	OTCGROI.Deposit.SY0834078080.0001	YES
MM0834000028	SY0834032420	DCD.MM.SY0834032420.0001	YES
MM0834000029	SY0834092599	OTCCRAN.Deposit.SY0834092599.0001	YES
MM0834000030	SY0834094485	OTCGROI.Deposit.SY0834094485.0001	YES
MM0834000031	SY0834000077	OTCCRAN.Deposit.SY0834000077.0001	YES
MM0834000032	SY0834006023	DCD.MM.SY0834006023.0001	YES
MM0834000033	SY0834045334	DCD.MM.SY0834045334.0001	YES
MM0835300024	SY0835357803	OTCCRAN.Deposit.SY0835357803.0001	YES
MM0836100003	SY0836140786	OTCGROI.Deposit.SY0836140786.0001	YES

Search by local fields

Local fields can be used in WHERE criteria

```
SELECT _ID, CURRENCY, SY_UNIT FROM FBNK_MM_MONEY_MARKET
WHERE SY_UNIT = 'DCD.MM.SY0834045334.0001'
```

```
SELECT _ID, CURRENCY, SY_UNIT FROM FBNK_MM_MONEY_MARKET WHERE SY_UNIT = 'DCD.MM.SY0834045334.0001'
```

FBNK_MM_MONEY_MARKET 1 x

SELECT _ID, CURRENCY, SY_UNIT FROM FBNK_MM_MONI *Enter a SQL expression to filter results (use Ctrl+Space)*

	ASC _ID	ASC CURRENCY	ASC SY_UNIT
1	MM0834000033	USD	DCD.MM.SY0834045334.0001

Local fields expansion

Local fields are expanded as normal table fields

```
SELECT _ID, SY_ID[EX], SY_UNIT[EX], SYEXVAL[EX] FROM FBNK_MM_MONEY_MARKET
```

The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the query: `SELECT _ID, SY_ID[EX], SY_UNIT[EX], SYEXVAL[EX] FROM FBNK_MM_MONEY_MARKET`. The results pane displays a table with 4 columns: `_ID`, `SY_ID`, `SY_UNIT`, and `SYEX...`. The table contains 14 rows of data. The status bar at the bottom indicates 'Query 1 of 1, Rows read: 75, Elapsed time (seconds) - Total: 1.898, SQL query: 1.894, Reading results: 0.004'.

_ID	SY_ID	SY_UNIT	SYEX...
MM0834000025	SY0834086929	OTCGROI.Deposit2.TEST	NO
MM0834000026	SY0834042600	OTCCRAN.Deposit.SY0834042600.0001	YES
MM0834000027	SY0834078080	OTCGROI.Deposit.SY0834078080.0001	YES
MM0834000028	SY0834032420	DCD.MM.SY0834032420.0001	YES
MM0834000029	SY0834092599	OTCCRAN.Deposit.SY0834092599.0001	YES
MM0834000030	SY0834094485	OTCGROI.Deposit.SY0834094485.0001	YES
MM0834000031	SY0834000077	OTCCRAN.Deposit.SY0834000077.0001	YES
MM0834000032	SY0834006023	DCD.MM.SY0834006023.0001	YES
MM0834000033	SY0834045334	DCD.MM.SY0834045334.0001	YES
MM0835300024	SY0835357803	OTCCRAN.Deposit.SY0835357803.0001	YES
MM0836100003	SY0836140786	OTCGROI.Deposit.SY0836140786.0001	YES
MM0900600005	SY0900600005	DCD.MM.SY0900600005.0001	YES

TAFC specific features

Note: Features described in this section are available only for only for [TAFC](#) connectors.

Field joins (Linked fields)

Field Joins functionality allows users to retrieve data from more than one file. This can be achieved by joining a field from the selected file with a field from another file.

Syntax: source_field > dest_file > dest_field

source_field - field name of the source file, containing record ID of the target file

dest_file - target file name

dest_field - field name of the destination file

For example, the following query will return accounts from FBANK_ACCOUNT table and customer's short name from FBANK_CUSTOMER table

`SELECT _ID, CURRENCY, CUSTOMER, CUSTOMER>F_CUSTOMER>SHORT_NAME FROM FBANK_ACCOUNT`

The screenshot shows the Squirrel SQL Client interface. The top menu bar includes File, Drivers, Aliases, Plugins, Session, Windows, and Help. The 'Connect to' dropdown is set to 'R10_jb', and the 'Active Session' is '1 - R10_jb ()'. The 'Objects' tab is selected, showing a tree view with '1 - R10_jb ()'. The 'SQL' tab is active, displaying the query: `SELECT _ID, CURRENCY, CUSTOMER, CUSTOMER>F_CUSTOMER>SHORT_NAME FROM FBANK_ACCOUNT`. The 'Limit Rows' dropdown is set to '100'. The 'Results' tab is selected, showing a table with 4 columns: _ID, CURR..., CUSTOMER, and SHORT_NAME. The table contains 10 rows of data. The status bar at the bottom shows 'Logs: Errors 0, Warnings 0, Infos 10' and '118 of 256 MB'.

_ID	CURR...	CUSTOMER	SHORT_NAME
21617	EUR	100683	Rcb Warsaw
21601	USD	100595	Indust Development Corp of Sa Ltd
42706	USD	100228	Charles Cooper
43338	USD	111514	Stewart
42935	USD	1000024	Xyz Private Limited
42927	USD	1000024	Xyz Private Limited
36908	USD	100315	Boeing
15307	EUR	100357	Cargill Plc
14087	EUR	100394	Merrill Lynch

User Defined Functions (UDF)

User defined functions (UDF) can be used inside queries to transform extracted data.

Example:

```
SELECT _ID, WORKING_BALANCE, @UDF.AC.OPEN.BAL('20090101') AS BAL_YEAR FROM FBNK_ACCOUNT
```

Will return account number, working balance and open balance for a specific date.

The screenshot shows the Squirrel SQL Client interface. The top menu bar includes File, Drivers, Aliases, Plugins, Session, Windows, and Help. The 'Connect to:' field is set to 'R10_jb', and the 'Active Session' is '1 - R10_jb ()'. The left sidebar shows 'Aliases' and 'Drivers' with 'R10_jb' and 'T24 - Oracle thin client' listed. The main window displays a SQL query: `SELECT _ID, WORKING_BALANCE, @UDF.AC.OPEN.BAL('20090101') AS BAL_YEAR FROM FBNK_ACCOUNT`. The query is limited to 100 rows. The results are shown in a table with columns: `_ID`, `WORKING_BAL...`, and `BAL_YEAR`. The table contains 8 rows of data. The status bar at the bottom shows 'Query 1 of 1, Rows read: 100, Elapsed time (seconds) - Total: 1.207, SQL query: 1.205, Reading results: 0.002'.

_ID	WORKING_BAL...	BAL_YEAR
USD141970001	-2483425.34	2164128.24
USD141980001	-10337.49	4690.42
USD142050001	<null>	0.0
USD142100001	1450000.0	0.0
USD142150001	0.0	0.0
USD142150010	608141.02	15000.0
USD142200001	-138274.19	0.0
USD142200010	-1257074.54	415529.85

You will find the source code of UDF.AC.OPEN.BAL subroutine in installation folder on your T24 environment (bnk.run/T24DAC.BP).

Each UDF function shall have 4 parameters: PARAMS, RESULT, ID.REC, R.REC.

- PARAMS – a multivalue set of system parameters
 - o If PARAMS<1> has value "TYPE" then UDF shall return in RESULT variable the type of function result, like "F" for float, "I" for integer, "D" for date, "C.128" for varchar(128)

- PARAMS<2, 1..N> contains the parameters passed to UDF in query.
For example in case of *SELECT _ID, @MY.UDF(1, 'SECOND', 3, CURRENCY) FROM FBNK_ACCOUNT* query, PARAMS<2> array will contain:
 - PARAMS<2, 1> = 1
 - PARAMS<2, 2> = 'SECOND'
 - PARAMS<2, 3> = 3
 - PARAMS<2, 4> = value of CURRENCY field for the corresponding record
- RESULT – in this variable the UDF's result shall be returned
- ID.REC – in this variable is stored the ID of the currently selected record
- R.REC – this variable will contain the currently selected record

To create a new UDF, the following steps are required:

1. Create, compile & catalog the a new jBC subroutine which will have 4 required parameters (PARAMS, RESULT, ID.REC, R.REC) and will do all the needed logic
2. Restart t24client.java module, so that the newly compiled routines will be accessible by t24client.java module
3. Use newly created UDF in queries
4. t24client.java module shall be restarted after every UDF change

Stored Procedures

Stored procedures can be used to generate data which is not present in a table, so it can be considered as “virtual” tables, from which data can be selected.

Example:

```
SELECT * FROM @SP.TABLE.FIELDS('ACCOUNT') ORDER BY FIELD_NAME
```

This query will return list of fields for a specific T24 application (ACCOUNT in this case)

The screenshot shows the Squirrel SQL Client interface. The SQL editor contains the query: `SELECT * FROM @SP.TABLE.FIELDS('ACCOUNT') ORDER BY FIELD_NAME`. The query has been executed, and the results are displayed in a table with the following columns: TABLE_NAME, FIELD_NAME, FIELD_TY..., FIELD_NO, FIELD_L..., and SINGLE_MU... The results show data for the ACCOUNT table, including fields like CUSTOMER, CATEGORY, PRODCAT, ACCOUNT.TITLE.1, ACCOUNT.TITLE.2, SHORT.TITLE, MNEMONIC, and POSITION.TYPE.

TABLE_NAME	FIELD_NAME	FIELD_TY...	FIELD_NO	FIELD_L...	SINGLE_MU...
ACCOUNT	CUSTOMER	D	1	10	S
ACCOUNT	CATEGORY	D	2	6	S
ACCOUNT	PRODCAT	D	2	6	S
ACCOUNT	ACCOUNT.TITLE.1	D	3	35	S
ACCOUNT	ACCOUNT.TITLE.2	D	4	35	S
ACCOUNT	SHORT.TITLE	D	5	35	S
ACCOUNT	MNEMONIC	D	6	10	S
ACCOUNT	POSITION.TYPE	D	7	2	S

You will find the source code of @SP.TABLE.FIELDS stored procedure in installation folder on your T24 environment (bnk.run/T24DAC.BP).

Each stored procedure shall have 3 parameters: PARAMS, RESULT, ID.REC.

- PARAMS – a multivalue set of system parameters
 - o If PARAMS<1> has value “FIELDS” then SP shall return in RESULT variable the list of field names and their types in format <field name>;<type>, ex. AMOUNT;F or DESCRIPTION;C.128 or COUNT;I
 - o If PARAMS<1> has value “PREPARE” then SP shall return in RESULT variable the list of IDs to be processed

- If PARAMS<1> has value “SELECT” then SP shall process the id passed in ID.REC parameter
- PARAMS<2, 1..N> contains the parameters passed to stored procedure in query.
For example in case of *SELECT _ID, NAME FROM @SP.CUS.LIST(1, 'second', 3)* query, PARAMS<2> array will contain:

- PARAMS<2, 1> = 1
- PARAMS<2, 2> = 'second'
- PARAMS<2, 3> = 3

- RESULT – in this variable the SP will return the list of ID or the selected/generated record
- ID.REC – in this variable is stored the ID which needs processing

To create a new stored procedure, the following steps are required:

1. Create, compile & catalog the a new jBC subroutine which will have 3 required parameters (PARAMS, RESULT, ID.REC) and will do all the needed logic
2. Restart t24client.java module, so that the newly compiled routines will be accessible by t24client.java module
3. Use newly created SP in queries

t24client.java module shall be restarted after every SP change

T24 Concat(index) files usage

Concat files (T24 index files) can be used using [stored procedures](#).

You will find the source code of several stored procedures that are using T24 concat files in installation folder on your T24 environment (bnk.run/T24DAC.BP):

- *IDX.SE.BOOK.DATE* – stored procedure that will select from STMT.ENTRY table based on booking date
- *IDX.CE.BOOK.DATE* – stored procedure that will select from CATEG.ENTRY table based on booking date
- *IDX.RCSE.BOOK.DATE* – stored procedure that will select from RE.CONSOLE.SPEC.ENTRY table based on booking date

*SELECT * FROM @IDX.SE.BOOK.DATE (20090108)*

or

*SELECT * FROM @IDX.SE.BOOK.DATE (LWORK)* – use last working day instead of specific one

Squirrel SQL Client snapshot-20181227_1847

File Drivers Aliases Plugins Session Windows Help

Connect to: R10_Ora_JB Active Session: 1 - R10_Ora_JB ()

1 - R10_Ora_JB ()

Objects SQL

SELECT account_number, amount_lcy, booking_date from @idx.se.book.date(LWORK)

Limit Rows: 100

SELECT account_number, amount_lcy, booking_date from @idx.se.book.date(LWORK)

SELECT * from @ SELECT * from @ SELECT account_ SELECT account_

Rows 298; SELECT account_number, amount_lcy, booking_date from @idx.se.book.date(LWORK) Selected Rows: 1, Cols: 0

Results MetaData Info Overview / Charts Rotated table Results as text

ACCOUNT_NUMBER	AMOUNT_LCY	BOOKING_DATE
SGDUSD140160001	0.21	2009-01-08
USD140050001	-2800.0	2009-01-08
USD140050001	-14625.0	2009-01-08
USD140050014	-4013.68	2009-01-08
USD140050014	-109614.37	2009-01-08
USD140050027	-6806.07	2009-01-08
USD140050027	-102172.77	2009-01-08
USD140050027	-99276.85	2009-01-08
USD140050060	-111.02	2009-01-08
USD140100001	4875.0	2009-01-08
USD140100014	4056.95	2009-01-08
USD140100014	109635.46	2009-01-08
USD140100027	6824.51	2009-01-08
USD140100027	102244.56	2009-01-08
USD140100027	99303.01	2009-01-08
USD140100060	124.89	2009-01-08
USD140650001	17000.0	2009-01-08
USD140650001	-17000.0	2009-01-08
USD141250001	166.67	2009-01-08

/R10_Ora_JB Current schema: <None> 1,78 / 78

Query 1 of 1, Rows read: 298, Elapsed time (seconds) - Total: 1.123, SQL query: 1.095, Reading results: 0.028

Query 1 of 1, Rows read: 298, Elapsed time (seconds) - Total: 1.101, SQL query: 1.1, Reading results: 0.001

Query 1 of 1, Rows read: 298, Elapsed time (seconds) - Total: 1.096, SQL query: 1.095, Reading results: 0.001

Logs: Errors 0, Warnings 0, Infos 15 81 of 303 MB 0 18:49:04 EEST

EVAL() function

The driver supports jBASE EVAL function. The function may be used as part of *select_expression*, or inside of *where_conditions*. Please note, EVAL function works slower than direct selection of fields or [user defined functions](#). Whether data processing speed is crucial, consider using UDF instead.

Syntax: *EVAL(eval expression)*

Eval expression - an [I-TYPE](#) expression

Example: *select _ID, EVAL(IF ACCOUNT.NUMBER[1, 3] = 'EUR' THEN 'euro' ELSE 'non-euro') FROM FBK_ACCOUNT*

Squirrel SQL Client snapshot-20181227_1847

File Drivers Aliases Plugins Session Windows Help

Connect to: R10_Ora_JB Active Session: 2 - R10_Ora_JB ()

2 - R10_Ora_JB ()

Objects SQL

```
select _ID, EVAL(IF ACCOUNT.NUMBER[1, 3] = 'EUR' THEN 'euro' ELSE 'non-euro') FROM FBANK_ACCOUNT
```

Limit Rows: 100

select _ID, EVAL(IF ACCOUNT.NUMBER[1, 3] = 'EUR' THEN 'euro' ELSE 'non-euro') FROM FBANK_ACCOUNT

select _ID, OPE select _ID, EVA

Rows 1,231; select _ID, EVAL(IF ACCOUNT.NUMBER[1, 3] = 'EUR' THEN 'euro' ELSE 'non-euro') FROM FBANK_ACCOUNT Selected Rows: 1, Cols: 1

Results MetaData Info Overview / Charts Rotated table Results as text

_ID	FIELD2
EURUSD140100001	euro
EURUSD190210001	euro
EURUSD190220001	euro
GBP100010011	non-euro
GBP100011200	non-euro
GBP100011500	non-euro
GBP100011501	non-euro
GBP100011510	non-euro

/R10_Ora_JB Current schema: <None> 1,1 / 1

SQLState: null
 ErrorCode: 0
 Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 1.092, SQL query: 1.09, Reading results: 0.002

Logs: Errors 0, Warnings 0, Infos 15 165 of 339 MB 1 21:40:01 EEST

SELECT _ID, OPENING_DATE, EVAL(OPENING.DATE[1, 4]) AS YEAR, EVAL(OCONV(ICONV(OPENING.DATE, "DG"), "D")) AS LONG_DATE FROM FBANK_ACCOUNT

Squirrel SQL Client snapshot-20181227_1847

File Drivers Aliases Plugins Session Windows Help

Connect to: R10_Ora_JB Active Session: 2 - R10_Ora_JB ()

2 - R10_Ora_JB ()

Objects SQL

```
select _ID, OPENING_DATE, EVAL(OPENING.DATE[1, 4]) AS YEAR, EVAL(OCONV(ICONV(OPENING.DATE, "DG"), "D")) AS LONG_DATE FROM FBANK_ACCOUNT
```

Limit Rows: 100

select _ID, OPENING_DATE, EVAL(OPENING.DATE[1, 4]) AS YEAR, EVAL(OCONV(ICONV(OPENING.DATE, "DG"), "D")) AS LONG_DATE FROM FBANK_ACCOUNT

select _ID, OPE

Rows 1,231; select _ID, OPENING_DATE, EVAL(OPENING.DATE[1, 4]) AS YEAR, EVAL(OCONV(ICONV(OPENING.DATE, "DG"), "D")) AS LONG_DATE FROM FBANK_ACCOUNT Selected Rows: 1, Cols: 0

Results MetaData Info Overview / Charts Rotated table Results as text

ID	OPENING...	YEAR	LONG_DATE
USD119100001	2008-12-02	2008	02 DEC 2008
USD120100001	2008-12-02	2008	02 DEC 2008
USD120110001	2008-12-02	2008	02 DEC 2008
USD120120001	2008-12-02	2008	02 DEC 2008
USD120130001	2008-12-02	2008	02 DEC 2008
USD121050001	2008-12-02	2008	02 DEC 2008
USD121100001	2008-12-02	2008	02 DEC 2008
USD121150001	2008-12-02	2008	02 DEC 2008

/R10_Ora_JB Current schema: <None> 1,1 / 1

Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 2.101, SQL query: 2.097, Reading results: 0.004
 Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 2.096, SQL query: 2.092, Reading results: 0.004
 Query 1 of 1, Rows read: 1,231, Elapsed time (seconds) - Total: 2.096, SQL query: 2.093, Reading results: 0.003

Logs: Errors 0, Warnings 0, Infos 15 151 of 339 MB 1 21:39:11 EEST

Performance tips

Use multiple SPU

SPU (SQL Processing Unit) is an “agent” which selects data from T24. Having more SPU will increase the selecting speed proportional. You can find out the optimal SPU count for your environment using our included [speed test tool](#).

Multiple CPU/Cores server

T24 Data Access Server uses parallel data processing. Installing server on a PC with multiple CPU/Cores will speed up the data process a lot.

Fast Hard Disks

T24Data Access Server uses temporary files to store data extracted from T24. Using high speed HDD/SSD will accelerate the extraction process.

RAM Disk Usage

Another way to speed up the temporary files process is to use RAM Disks (ex. [ImDisk](#), an opensource project). Just create a new RAM drive and update *Temp* parameter in [configuration](#) file to point to newly created drive.

Speed test tool

In the installation package (t24dac.zip) you will find *SpeedTest.exe* tool. It will test the driver performance with different number of SPU to determine the optimal number of SPU.

Before running it, you must update several parameters in *SpeedTest.ini* file:

- *TestTable* – the table used for testing the selection speed. Table shall have enough number of records
- *TestRecordCount* – number of records to be selected from test table for testing the selection speed
- *TestMinSPU* – initial number of SPU to start tests with.
- *SPUStep* – SPU increase step
- *RecordCountStep* – increase step for record count

After running the speed test tool, you will get the running statistics with recommended number of SPU.

Free version vs Commercial version

There are some limitations in Free version comparing to a Commercial one.

1. Free version can use only 1 SPU
2. Speed limitation. Only first 10 seconds the driver will run at full speed of 1 SPU, the rest will be processed at limited speed (300 records per second)
3. Parallel execution limitation. It is not possible to run jobs in parallel. If a job is started while another is running – an exception will be raised.

Technical considerations

Failovers

If during the record selection process (in case of [Oracle Direct Connector](#)) Vultar Server receives ORA-03113 (end-of-file on communication channel) or ORA-12514 (TNS:listener does not currently know of service requested in connect descriptor) error, then it will retry to select the failed record. The retry count is specified in *RetryCount* value in configuration file. After that the job will be cancelled and error propagated up to client.

Vultar Server keeps all connections in a pool. If a connection becomes inactive – it is closed and a new one is created.

Vultar Server monitors its RAM consumption. In case it uses more than 2GB (it can happen when data is selected faster than it's processed) – then it will slow down the selection process to parse the already selected and queued records. The value of 2GB can be changed by adjusting the *MaxRAM* value in configuration file.

Performance

Each selection job in Vultar Server is executed in parallel by multiple SPU (SQL Processing Unit). The number of SPU is specified in the license key. In case there are several selection jobs running in parallel, the SPU number will be shared between the jobs. You can limit the number of SPU used by updating the value SPU in configuration file.

When record is selected by an SPU – it is sent for processing (parsing, expanding, multivalue fields processing, etc.) to a unit called Processor. By default, Vultar Server creates as many Processors, as many CPU cores are on the server. If it is needed to limit the number of CPU used – set the maximum number of Processors that can be created by updating the value ProcessorCount in configuration file.

Logging

By default, log files are stored in c:\ProgramData\Vultar\logs folder. It can be changed in configuration file by updating the LogDir value. In same configuration file (c:\ProgramData\Vultar\config.ini) you can set the log level (0-5). Greater level – more details. Default log level is 2. Log levels greater than 2 should be used only for troubleshooting because it affects the performance significantly.

Log level description:

- 0 – no logs
- 1 – only errors
- 2 – includes warnings and general job statistics
- 3 – includes more technical details about the jobs, like SELECT statement and other details
- 5 – includes all communications between Vultar Server and T24

Troubleshooting

Internal error: unable to open database: C:\Windows\Temp\...

Cause: The service does not have access full access to C:\Windows\Temp folder.

Solution: Give full access for all users to *C:\Windows\Temp* folder, or update *Temp* value in [config.ini](#) file to point to a different folder where all users have access (ex. *Temp=d:\temp*)

JBC code wont compile while starting

1. **Cause:** t24client.jar failed to detect T24 include folder

Solution: update t24client.properties and set *T24.Include.Dir* property to the proper value

Ex: *T24.Include.Dir=../T24_BP*

2. **Cause:** t24client.jar fails to compile JBC code due to unknown reason

Solution:

- a. update t24client.properties and set *Recompile=false*
- b. manually compile the JBC code located in T24DAC.BP folder