DESCRIPTIONS OF DATABASES AND FILES

Version 5.1.0

1.9.2015

Bank of Finland
PAYMENT AND
SETTLEMENT SYSTEM
SIMULATOR

VERSIONS

DATE	AUTHOR	VERSION	CHANGES
28.11.2002	M. Kilvio	0.0.1	Document body, based on BoF – PSS II General Functional Specification *Version 2.0*
8.12.2002	Markus Nissilä	0.0.2	Descriptions of the entities. Some changes to chapter 4.
10.12.2002	Markus Nissilä	0.0.3	Log file added to chapter 3.
11.12.2002	Markus Nissilä	0.0.4	References added, etc.
02.01.2003	Maritta Halonen	0.0.5	Changed the document template to MSG's template, added the database SYSTEM. Added file descriptions.
10.01.2003	Maritta Halonen	0.0.6	Corrections after the review
23.01.2003	Maritta Halonen	0.0.7	Corrections after Harry Leinonen's and Kimmo Soramäki's comments. Defined more closely directories. Changed to Bank of Finland's format.
14.02.2003	Maritta Halonen	0.0.8	Corrections after Harry Leinonen's and Kimmo Soramäki's comments.
28.04.2003	Maritta Halonen	0.0.9	Added field SB_CVITYPE to the BACO table. Removed field SB_AFIELDS from the BACO table. Changed the name of the root directory.
12.05.2003	Harry Leinonen	0.0.10	Proof reading and editorial changes
12.05.2003	Harry Leinonen	0.0.11	Proof reading and editorial changes
18.06.2003	Harry Leinonen	0.0.12	Proof reading and editorial changes
18.06.2003	Leena Tyni	0.0.13	Baco table moved to the output database. Linkcodes from INT to Varchar(8), Usercodes from Varchar(8) -> Varchar(12).
06.08.2003	Leena Tyni	0.0.14	SYLS and ACST tables updated.
08.08.2003	Harry Leinonen and Kimmo Soramäki	0.0.18	Updates on statistical tables, especially descriptions. Annex 1 has been introduced.
12.09.2003	Harry Leinonen	0.0.19	Field and description updates
04.12.2003	Harry Leinonen	0.9.6	SA_SYSCODE included and new file directory structure and AVERTISE unit descripition
30.07.2004	Harry Leinonen	1.0.0	No database changes from beta version to production version 1.0.0
24.2.2005	Matti Hellqvist	1.2.0	Updates in field descriptions caused by ENFORURG-algorithm. Editorial changes and corrections in Annex 1.
27.06.2005	Harry Leinonen	2.0.0	Update for version 2.0.0, among others bilateral limits, increased user defined fields and performance improving definitions
5.6.2009	Matti Hellqvist	2.4.0	Editorial updates for the new version. No database changes.
6.8.2009	Markus Penttilä	2.4.0	More editorial updates, couple of fixes. No database changes.
12.8.2009	Kasperi Korpinen	2.4.0	Data type errors fixed.
14.8.2009	Markus Penttilä	2.4.0	Minor editorial updates.
29.11.2010	Kasperi Korpinen	3.0.0	3.0.0 DB Changes
19.1.2011	Matti Hellqvist	3.0.0	Editorial updates, added settlement status "-1"
23.3.2011	Kasperi Korpinen	3.0.1	Field value 0 for QURE queueing reason code was defined.
18.11.2011	Kasperi Korpinen and Matti Hellqvist	3.1.0	Credit cap additions Correction of documentation of AVERTISE & AVEQUELE-fields Documentation of field QUERCAU
02.07.2012	Kasperi Korpinen	3.1.0	P_ACCOTYPE, M_RSRVDSID
7.9.2012	Kasperi Korpinen	3.2.1	R_QURECODE codes 4 and 5
28.5.2013	Kasperi Korpinen		QUEIDATE QUEITIME, BEST,UNST,SUST,CCST,CTST tables not supported anymore
1.8.2013	Kasperi Korpinen	4.0.0	new queue reason code FIFO constraint
12.4.2013	Kasperi Korpinen	4.0.0	Added to ICCL and TRAN, Business day field SYLS simulation date changed to business day

			ACST: ACCOTYPE, business day BIST: business day TEST: business day AVST: business day Added the table business_day_event
19.09.2013	Kasperi Korpinen	4.0.0	Added M_BUSDESID to SIRI table description
24.9.2013	Kasperi Korpinen	4.0.0	Input data structure: added evnts
16.9.2015	Kasperi Korpinen	5.1.0	New fields for different iccl orders and the ICl execution output statistics table

Contents

1 Introduction....5
2 General....5
2.1 About the databases....5
2.2 Data sets....7
3 Relation Model....8
3.1 Notation....8
3.2 System database structure....9
3.3 Input database structure....9
3.5 Output database structure....12
4 Description of database tables....15
4.1 System database....15
4.1.1 Defaults [DEFA]....15
4.1.2 Project [PROJ]....15
4.1.3 Algorithm definition [ALDE]....15
4.1.4 Template [TEMP]....16
4.2 Input database....17
4.2.1 System control data [SYCD]....17
4.2.2 Participant data table [PART]....18
4.2.3 Daily balances table [DBAL]....19
4.2.4 Intraday changes in credit limit [ICCL]....19
4.2.5 Bilateral limit table [BLIM]....19
4.2.6 Reservations table [RSRV]....20
4.2.7 Transaction data table [TRAN]....21
4.2.8 Transaction changes [trch]....22
4.2.9 System algorithms [SALG]....24
4.3 Output database....25
4.3.1 System level statistics [SYLS]....25
4.3.2 Account statistics [ACST]....26
4.3.3 Bilateral statistics table [BIST]....28
4.3.4 Transaction event statistics [TEST]....28
4.3.5 Netting event statistics [NEST]....30
4.3.6 Account violation statistics [AVST]....31
4.3.7 Booking event statistics [BEST]....31
4.3.8 Queue reason information [QURE]....33
4.3.9 Unsettled transactions statistics [UNST]....33
4.3.10 Submitted transactions statistics [SUST]....34
4.3.11 Queued transactions statistics [QUST]....36
4.3.12 Comment transactions statistics [CTST]....37
4.3.13 Comment intraday credit statistics [CCST]....39
4.3.14 Technical output database tables....39
5 Files....42
5.1 Error list....43
5.2 Basic statistics reports....44
5.3 CSV and Excel files....44
6 Annex 1 Calculation of specific indicators....45

1 Introduction

This document describes the databases and files in the Payment and Settlement Systems Simulator, version 2 (BoF-PSS2).

The document has four main chapters:

1) The general chapter 2 explains the databases the program creates and where these databases are located. There is also some information about data sets.

2) The relation model chapter 3 describes the structure of the databases (tables and relations).

3) The next chapter 4 describes the database tables.

4) The final chapter 5 describes files, directories, and their locations.

2 General

2.1 About the databases

The simulator uses MySQL databases which are as a default setting stored on the same workstation. The simulator uses three types of databases: a system database, input databases and output databases. There is only one common system database for each simulator installation, but each project will have an own input and output database.

The simulator's system database is created during the initial installation of the program. The installation program creates the directory PSS2_SYSTEMDB in the installation directory of the program for the system database. C:\BoF-PSS2 is the default directory for the simulator (see Figure 1).

The simulator uses MyISAM storage engine of MySQL. With this setup, each database table has three separate files. These are:
- Data dictionary information (.FRM files),
- Index information (.MYI files), and
- Data files (.MYD files).

These files reside in the file directory of that particular database. The location information of the actual file directory is saved for MySQL in a symbolic link file (database_name.SYM) which is located in data folder of MySQL server installation path (c:\mysql\data).

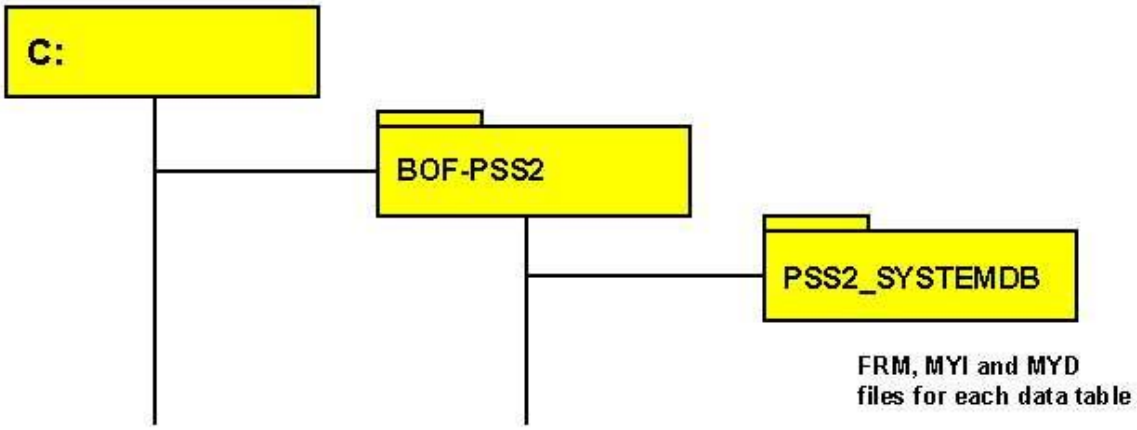

Figure 1: PSS2_SYSTEMDB directory path.

Input and output databases are created by default for each specified project in a project-specific directory. The project directory by default is identical with the project name preceded by P_. The input and output databases by default reside in different subdirectories of the project directory identified by the prefix 'I_' for input databases and 'O_' for output databases followed by the project name. These directories will be created by the initial specification view together with the .FRM, .MYI and .MYD files for each database table.

Only one input and output database at a time can be specified for each project. Often, however, the input database may be shared by several projects, e.g. the same input data may be used both for a liquidity requirement and a systemic risk simulation project. The user can also specify directory names other than the default, but *must use caution* to ensure database integrity. In certain cases, it is convenient to store large databases e.g. on a network drive.

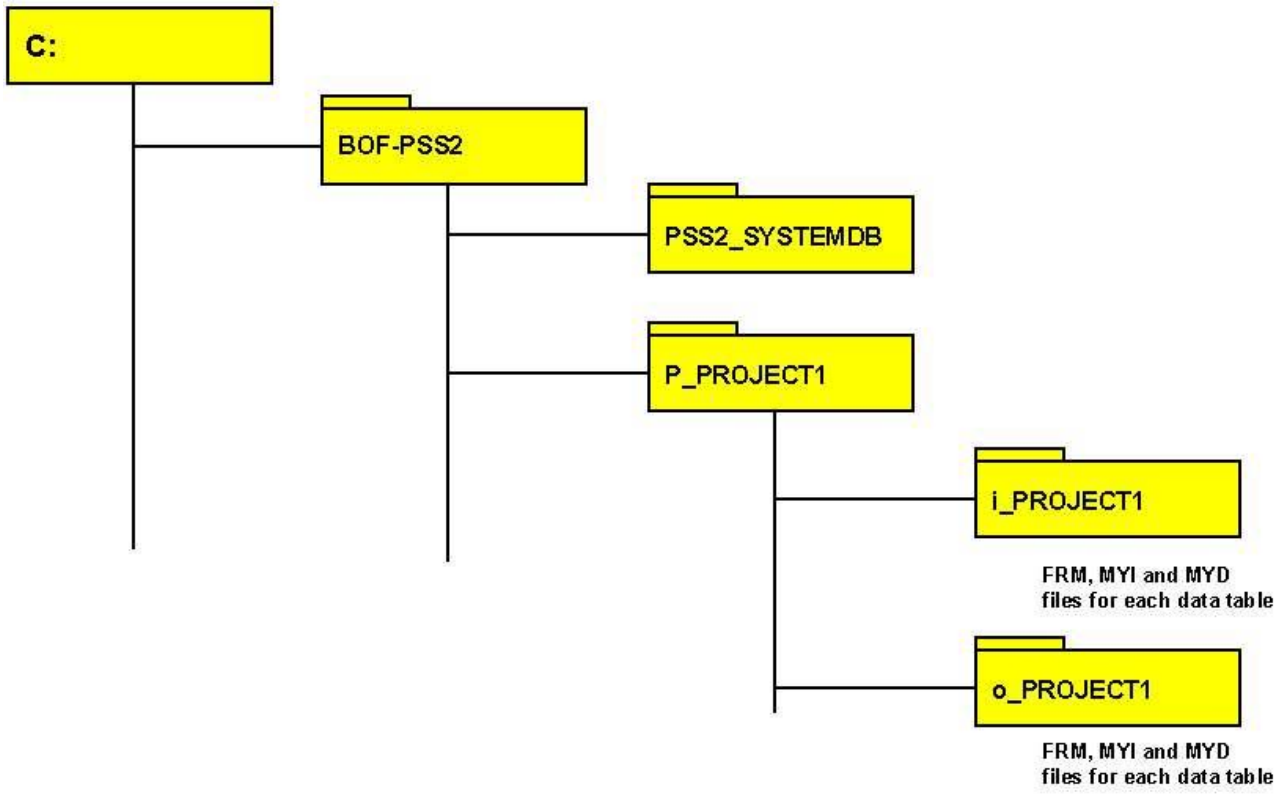

Figure 2: Project and input/output database paths.

2.2 Data sets

One input database can store many data sets for each type of input data. The different data sets are stored in the same physical database table and are distinguished by a data set ID. The user defines the data set ID separately for each data table. It has no internal database relation with any data set ID of other database tables (i.e. it is not a foreign key).

In the simulation execution phase, the user defines which specific data sets among all those stored are to be used in a specific simulation as described in the Figure 3.

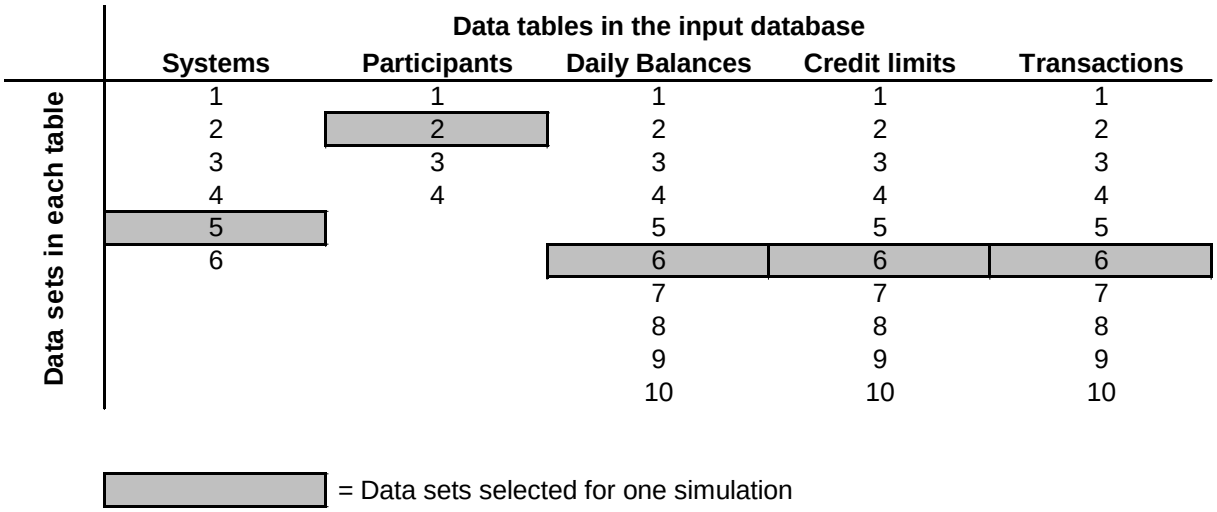

Figure 3: Parallel data sets in the same database tables.

3 Relation Model

3.1 Notation

The database tables are pictured in Figures 4, 5 and 6 below as rectangles. Each rectangle contains the acronym and full name of the table, as well as the fields of the table. For output tables only part of the fields is shown.

The letters PK and FK defines the key type. PK indicates that the field is a primary key while FK indicates that the field is a foreign key. The primary key is a field or a combination of fields in the table that uniquely identify a record in the table. Foreign keys are all the fields in a table that refer to a primary key in another table.

Keys that are not primary or foreign keys are marked with an asterisk (*), i.e. key data, which are stored as redundant information without being used as key in that particular data table.

Arrows describe relations between tables. Relations are marked two ways:
- A solid line describes a real relation maintained by the database, and
- A dashed line describes a relation that is used in the simulations and will be cross-checked in the application. For dashed line there is no real relation defined in the database.

The markings at the starting points of the arrows show how many records are related in the table. The possibilities are:

- 1:1: Each record in the table from where the arrow starts is related to the record in the table where the arrow ends.
- 1:N: Each record in the table where the arrow starts is related to one or more records in the table where the arrow ends.
- 0:N: Each record in the table where the arrow starts may, but does not necessarily, have related records in the table where the arrow ends.

The transaction table has a self-reference, i.e. that a transaction has a connection with another transaction (the DVP/PVP relation).

The data type of the field is described after the name of the field:
- char(x): The length of the field is x characters. By default MySQL uses the ISO-8859-1 (Latin1) character set with sorting according to Swedish/Finnish. This character set is also suitable for the North America and most of Europe. Instructions for changing character sets are found in MySQL manual /1/. The data type for char(x) in different database could be, for example, BINARY(x).
- varchar(x): The field can contain the maximum x characters. If the value in the field is shorter, the length of the field is the length of the string. The character set is given as in char(x) above. Data type for char(x) in different databases could be, for example, CHAR VARYING(x).
- Bigint: This integer-type field can range between -9223372036854775808 and 9223372036854775807 (signed), or between 0 and 18446744073709550615 (unsigned).
- Int: This integer-type field can range between -2147483648 and 2147483647 (signed), or between 0 and 4294967295 (unsigned).
- Mediumint: This integer-type field can range between -8388608 and 8388607 (signed), or between 0 and 16777215 (unsigned).

- Smallint: This integer-type field can range between -32768 and 32767 (signed), or between 0 and 65535 (unsigned).
- Tinyint: This integer-type field can range between -128 and 127 (signed), and between 0 and 256 (unsigned).
- Text: This field can contain up to 65536 characters. The stored length of the field is the length of the string in the field. Character set is the same as in char(x).
- Decimal: This field stores values in decimal-format. Precision of the field is defined as decimal (20,2) and thus the database can hold at most 20 significant numbers.[1] With this stup only values which are in the range (-1E19…1E19) can be stored accurately.

A consistent naming convention has been created for data fields, where each field in the same data table has the same prefix. Data fields containing the same logical data has then the same particular name, e.g. the system identifier field contained in different data tables has the name S_SYSTEMID, P_SYSTEMID, T_SYSTEMID depending on the table in which it is included. All field names are 10 characters long, including the prefix and delimiter.

3.2 System database structure

The system database contains:
- Default data (DEFA) contains default information for a current project. Field names have an SD_ prefix.
- Algorithm definition data (ALDE) contains information of user modules and algorithms. Field names have an SA_ prefix.
- Project data (PROJ) contains data of all projects. Field names have an SP_ prefix.
- Template data (TEMP) contains template data of input files. Field names have an ST_ prefix.
- Acceptable system ID:s data (ASID) contains information on acceptable systems id:s. Field names have a SY_prefix. **Note!** This is a database table is not used in version 2.4.0.

The tables have no relations with each other. The SALG table in the input database, however, relates to the ALDE table.

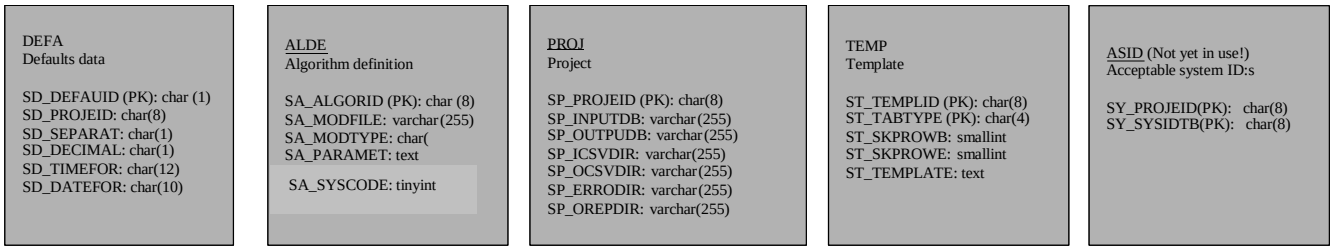

Figure 4: System database structure.

3.3 Input database structure

The input data contains

[1] Up to version 2.4.0 the output database rows were defined to hold only 15 significant numbers i.e. largest value, which is recorded exactly is 9 999 999 999 999,99 . The data field precision can be increased manually with alter table command, see MySQL documentation.

- System control data (SYCD) contains system data for specific data sets of the systems. Field names have an S_ prefix.
- Participant data (PART) contains participant data sets for specific participants set belonging to a particular system. Participants can be distinguished in this table at two levels. The participant ID can be 11 characters long and can contain a SWIFT BIC address. The account ID can be 34 characters long and can contain an IBAN. Both fields can also be used for other identifiers. The participant ID is mandatory. The account ID can be omitted. In that case the field will be empty, i.e. will contain the value "". Field names have a P_ prefix.
- Daily balances data (DBAL) contains daily opening balances for the participants in the PART table. Field names have a D_ prefix.
- Intraday credit limits data (ICCL) contains information of changes in intraday credit limits for specified participants in the PART table. Field names have an I_ prefix.
- Bilateral limits data (BLIM) contains information of changes in bilateral sending limits for pairs of participants in the PART table or towards all as a multilateral sending limit. Field names have a L_prefix.
- Reservations data (RSRV) contains liquidity reservation information. Field names have a R-prefix.
- Transaction data (TRAN) contains transaction data sets for participants set in the PART table related to a system (or systems, when multiple systems are simulated) set in the SYCD table. Field names have a T_ prefix.
- System algorithms data (SALG) contains algorithm data for the specified systems. Field names have an A_ prefix. This table relates to the Algorithm definition (ALDE) table in the system database.
- Transaction changes data (TRCH) contains change information related to transactions.
  Note! This is a database table for future development plans with no function currently.
- Simulation events [business_day_event] is used to store data on start and end of days of each business day. The table can be used to store some other tailored specific events supporter by specific eventhandlers.

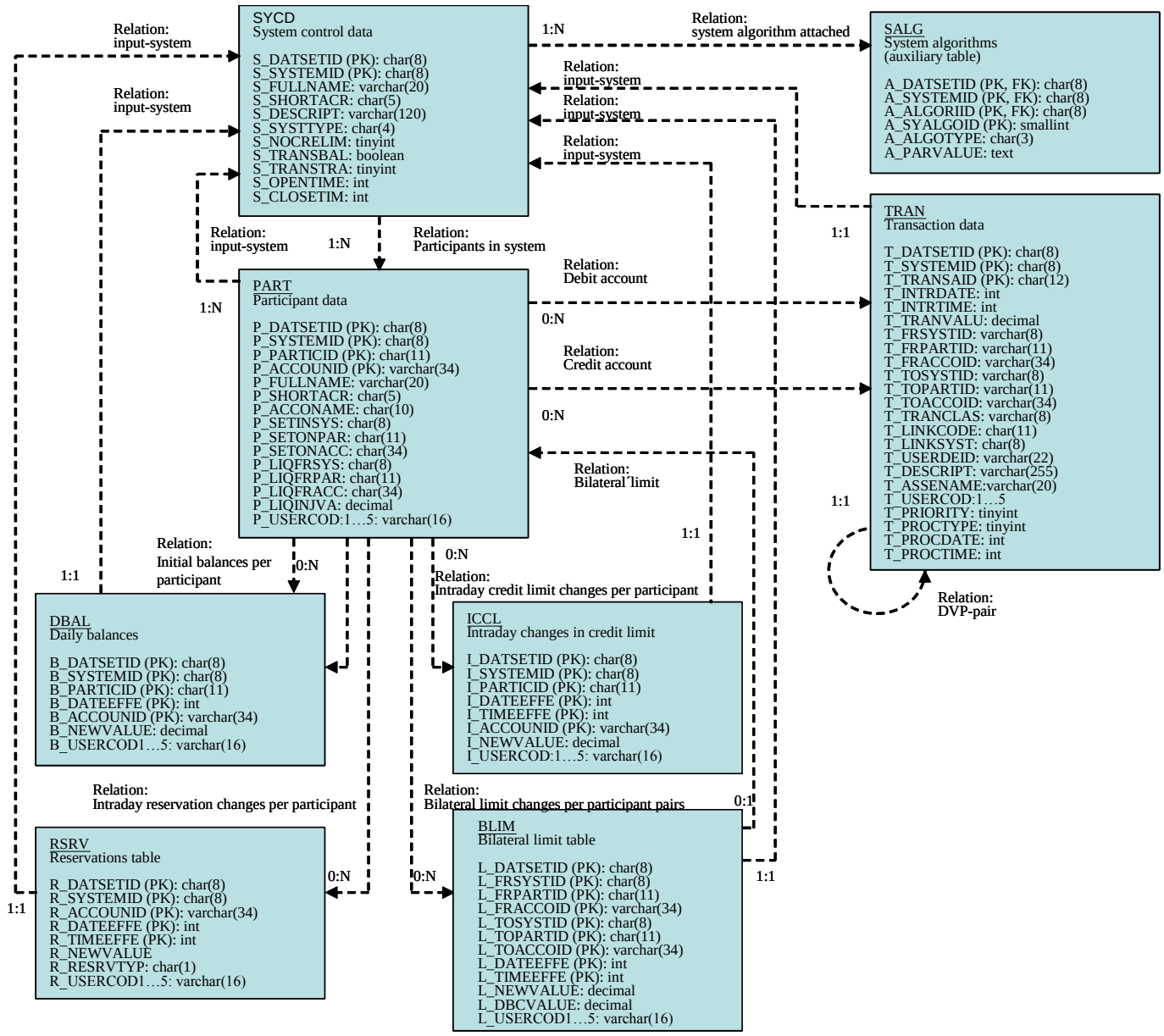

Figure 5: Input database structure.

3.5 Output database structure

The structure of the output database is described in Figure 6 (to fit the figure on a single page, only the keys of each table are included).

Each data table contains a simulation ID to separate the output data of different simulations. This can be compared to the data set ID, which separates different input data sets in the input database. The output database contains the results of a number of simulations to facilitate comparison between simulations.

The output database contains four levels of data tables: simulation, system, account (participant) and transaction.

The output database is not normalized (the same fields are in different tables), because the program handles tables as if they were separate. The user can select the desired simulation output data. The user can also delete any data from the output tables.
The output data contains:

- Batch run information (BARI) contains information of simulation batch runs. Field names have an R_ prefix.
- Simulation run information (SIRI) contains information about simulation runs. Field names have an M_ prefix.
- System level statistics (SYLS) contains system-level statistics for simulation runs. Field names have an Y_ prefix.
- Account statistics (ACST) contains general statistics per participant or account for a given day. Field names have an A_ prefix
- Bilateral limits statistics (BIST) contains statics on bilateral limits for each pair of participants, which have been assigned bilateral limits or for all assigned multilateral sending limits
- Transaction event statistics (TEST) contains the general statistics of transaction events occurring in specific simulation runs. Field names have an E_ prefix.
- Netting event statistics (NEST) contains information about netting events occurring in specific simulation runs. Field names have an N_ prefix.
- Account violation statistics (AVST) contains information of account violations occurring in simulation runs. Field names have a V_ prefix.
- Booking event statistics (BEST) contains information of bookings in simulation runs. Field names have an O_ prefix.
- Queue reason statistics (QURE) contains information on the different reasons for queuing transactions. Field names have a K_prefix.
- Unsettled transactions statistics (UNST) contains information of transactions that remained unsettled in simulations. Field names have a U_ prefix.
- Submitted transactions statistics (SUST) can be used to follow a user-made submission algorithm output. Note, this will duplicate transaction statistics! Field names have an X_ prefix.
- Queued transactions statistics (QUST) contains general information of queued transactions in specified simulation runs. Field names have a Q_ prefix.
- Comment transactions statistics (CTST) contains event information for commented transactions of specified simulation runs. Field names have a C_ prefix.
- Comment intraday credit statistics (CCST) contains information of changes in commented intraday credit limits of specified simulation runs. Field names have a D_ prefix.
- Basic comparison view data (BACO) contains template data of the comparison analysis report. Field names have an SB_ prefix.

- Applicationruns is intended to contain data on simulation runs. It is not in use for the moment
- Processlog is used to keep log of different events occurring during a simulation such as start of algorithm and end of algorithm. The table is intended to keep better track of the simulation process and is usefull for testing purposes. All algorithms do not support this function and thus cannot write to the processlog.

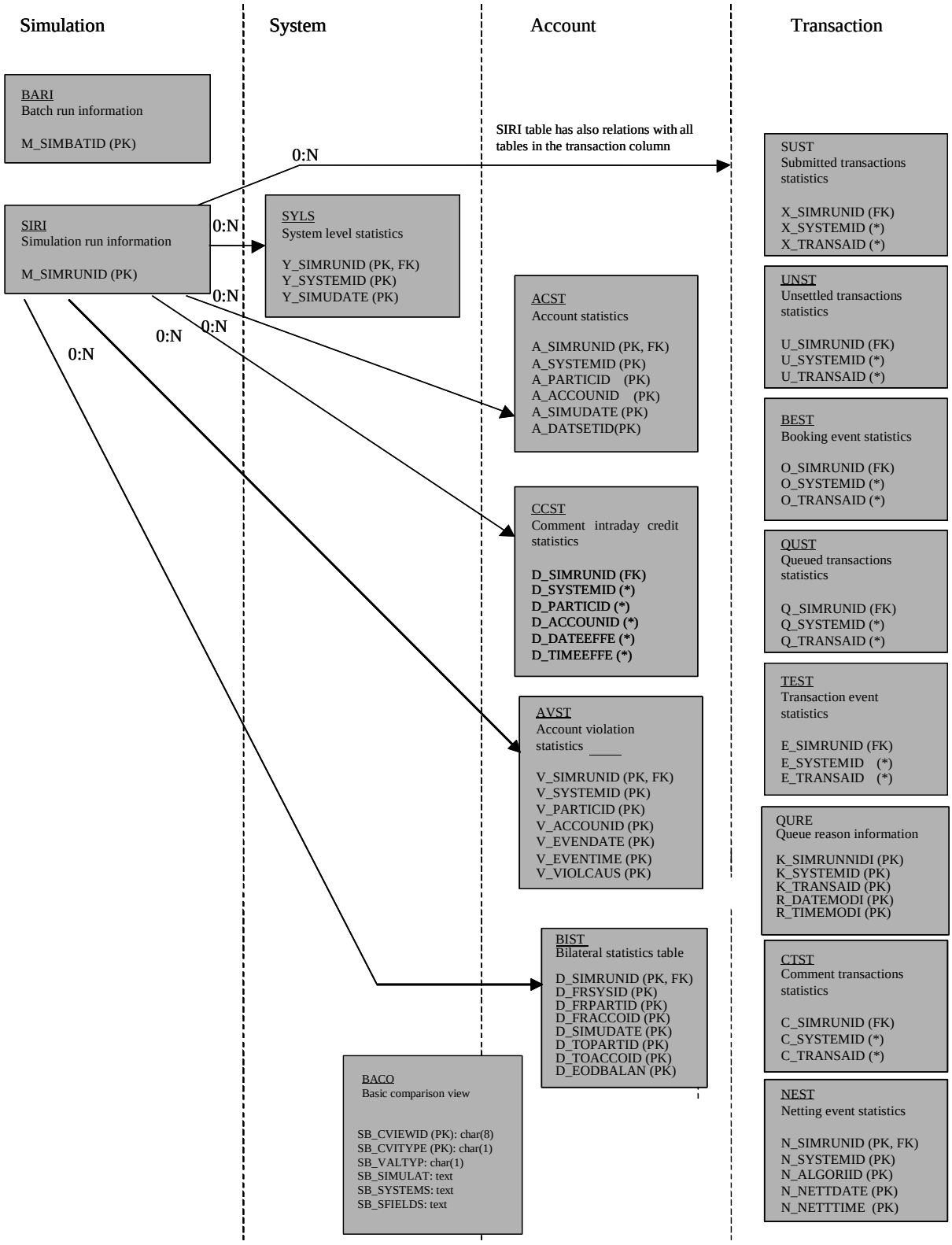

Figure 6: Output database structure.

4 Description of database tables

The first column indicates the variable name of the field. The second column indicates the type of field. The third column indicates whether the field is a foreign (F) key, a primary (P) key or just a key (*). The fourth column gives the detailed name of the field. The fifth column describes the field. The last column indicates if the data is optional (O) or mandatory (M).

4.1 System database

4.1.1 Defaults [DEFA]

Field name	Data type	Key	Detailed name	Description	
SD_DEFAUID	CHAR(1)	P		Value = 1, just to introduce the mandatory key.	M
SD_PROJEID	CHAR(8)		Project ID	Current project ID.	M
SD_SEPARAT	CHAR(1)		Separator mark	Separator used between data fields in CSV files.	M
SD_DECIMAL	CHAR(1)		Decimal point format	Decimal point format.	M
SD_TIMEFOR	CHAR(12)		Time format	Time format.	M
SD_DATEFOR	CHAR(10)		Date format	Date format.	M
SD_TRAVALU	CHAR(5)		Time transposition value	+/-HH:MM value to change input time	

4.1.2 Project [PROJ]

Field name	Data type	Key	Detailed name	Description	
SP_PROJEID	CHAR(8)	P	Project ID	Identifier for the project.	M
SP_INPUTDB	VARCHAR(255)		Input database	Input database name.	M
SP_OUTPUDB	VARCHAR(255)		Output database	Output database name.	M
SP_ICSVDIR	VARCHAR(255)		Input file directory	Directory for input file.	M
SP_OCSVDIR	VARCHAR(255)		Output file directory	Directory for output file.	M
SP_ERRORDIR	VARCHAR(255)		Error directory	Directory for error list.	M
SP_OREPDIR	VARCHAR(255)		Output report directory	Directory for output reports.	M
SP_NETDIR	VARCHAR(255		Networks directory	Default location for created networks	M
SP_NETREPORTS	VARCHAR(255		Network reports directory	Default location for generated network reports	M

4.1.3 Algorithm definition [ALDE]

Field name	Data type	Key	Detailed name	Description	
SA_ALGORID	CHAR(8)	P	Algorithm ID	Unique identifier of the algorithm.	M
SA_MODFILE	VARCHAR(255)		Module file	Name of module file.	M
SA_MODTYPE	CHAR(3)		Module type	Type of module.	M
SA_PARAMET	TEXT		Parameters and check rules	Enumerated list of parameters and checking rules used by algorithm.	O
SA_SYSCODE	TINYINT(4)		System code	Describes in which system the algorithm is available. Values are additive 8=RTGS, 4= CNS and 2=DNS e.g. 12 indicates availability in RTGS and CNS systems.	M

4.1.4 Template [TEMP]

Field name	Data type	Key	Detailed name	Description	
ST_TEMPLID	CHAR(8)	P	Template ID	Unique identifier of the template.	M
ST_TABTYPE	CHAR(4)	P	Table type	Refers to specific data set table.	M
ST_SKPROWB	SMALLINT (3)		Rows to skip in the beginning	Number of rows to skip in the beginning.	O
ST_SKPROWE	SMALLINT (3)		Rows to skip in the end	Number of rows to skip in the end.	O
ST_TEMPLAT	TEXT		Template	Enumerated list of column numbers, which describes the column structure of the CSV file. Value in form: DB-Column1 match column in CSV file, DB-Column2 match column in CSV file ,…	M

Acceptable system Ids [ASID] (Note! Not in use.)

Field name	Data type	Key	Detailed name	Description	
SY_PROJEID	CHAR(8)	P	Project ID	Unique identifier of the project.	M
SY_SYSIDTB	CHAR(8)	P	System ID	Unique identifier of acceptable system.	M

4.2 Input database

4.2.1 System control data [SYCD]

Field name	Data type	K	Detailed name	Description	
S_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of the data set to distinguish the data set from other parallel data sets used in simulations.	M
S_SYSTEMID	CHAR(8)	P	System ID	Unique identifier of system.	M
S_FULLNAME	VARCHAR(20)		Full name	Full name of system.	O
S_SHORTACR	CHAR(5)		Short acronym	Short acronym for system.	O
S_DESCRIPT	VARCHAR(120)		Description of the system	Description of system.	O
S_SYSTTYPE	CHAR(4)		System type	Possible system types are RTGS, CNS and DNS. CNS and DNS systems typically settle their end-of day net-positions. RTGS systems and sometimes CNS systems may have intraday liquidity injections from an RTGS system. The DNS system settles transactions at specified settlement occasions on a batch net basis, while CNS systems settle continuously at the transaction level. This selection only affects the list of algorithms made available in the GUI for algorithm selection. This does not affect the simulations it selves... Possible values: RTGS DNS CNS	M
S_NOCRELIM	TINYINT(4)		No credit limit in force parameter	Gives the opportunity to specify that all participants or accounts in a system have infinite credit limits. Mainly used for DNS systems, but may also be used for RTGS and CNS systems to determine maximum liquidity requirements. Possible values: 1: Credits according to limit table 2: No Credits available 3: Credits available without limits	O
S_TRANSBAL	TINYINT(1)		Transfer of balances to next day	Transfer of end-of-day balance to the next day. Mainly used in RTGS systems. Possible values: 0: Balaces are not transferred 1: Balances are transferred	O
S_TRANSTRA	TINYINT(4)		Transfer of unprocessed transactions to the next day or settlement occasion	At the end of the day, there may be unprocessed transactions in the RTGS and CNS queues. These can be eliminated or transferred to the next day. In the DNS system, the choice is to eliminate or transfer unprocessed transactions to the next settlement occasion. Possible values: 0: unsettled payments are not transferred 1: unsettled payments are transferred	O
S_OPENTIME	INT		Opening time of the system	Defines the time from which transactions will be submitted to the system in the beginning of the day. Transactions that have an earlier submission time will wait until the open time point is reached.	M
S_CLOSETIM	INT		Closing time of the system	Transactions with submission times after the closing time of the system will be submitted at the beginning of the following day.	M
S_BILIMUSE	TINYINT(4)		Bilateral limit in use	Values: 0 denotes that bilateral llimits are not in use 1 denotes that bilateral limits are in use	

4.2.2 Participant data table [PART]

Field name	Data type	Key	Detailed name	Description	
P_DATSETID	CHAR(8)	P	Data set ID	Unique data set identifier distinguishes this data set from other parallel data sets used in simulations.	M
P_SYSTEMID	CHAR(8)	P	System ID	Unique identifier for system in which account is located.	M
P_PARTICID	CHAR(11)	P	Participant ID	Unique identifier for the participant used in the transaction data. This is sufficient also as an account identifier when there is only one account per participant per system.	M
P_ACCOUNID	VARCHAR(34)	P	Account ID	Unique identifier of account in which credits and debits are made. This attribute is only relevant when the participant has more than one account. This field is mandatory because it is a primary key, but it can be an empty field ie "" to which no data has been imported.	O
P_ACCOTYPE	VARCHAR(1)		Account type	Used to distinguish different types of accounts.	O
P_FULLNAME	VARCHAR(35)		Full name	Full name of participant.	O
P_SHORTACR	CHAR(5)		Short acronym	Acronym for full name of participant. The acronym is used in the run-time view of the simulator, if available.	O
P_ACCONAME	CHAR(10)		Account name	Name of account, e.g. "Euro RTGS account."	O
P_SETINSYS	CHAR(8)		Settles in system	For DNS or CNS systems, the ID of the system where proceedings are booked. May also be used in RTGS systems for transferring end-of-day positions from sub-systems or accounts to main systems or accounts.	O
P_SETONPAR	CHAR(11)		Settles on participant	For DNS or CNS systems, the ID of the participant to whom the end-of-day proceedings are booked. May also be used in RTGS systems for transferring end-of-day positions from sub-systems or accounts to main systems or accounts.	O
P_SETONACC	CHAR(34)		Settles on account	For DNS or CNS systems, the ID of the account in which the end-of-day proceedings are booked. May also be used in RTGS systems for transferring end-of-day positions from sub-systems or accounts to main systems or accounts.	O
P_LIQFRSYS	CHAR(8)		Liquidity injection from system	For CNS systems, the ID of the system to and from which liquidity injections are booked. May also be used in RTGS systems for transferring liquidity to and from sub-systems or accounts from and to main systems or accounts.	O
P_LIQFRPAR	CHAR(11)		Liquidity injection from participant	For CNS systems, the ID of the participant to and from which liquidity injections are booked. May also be used in RTGS systems for transferring liquidity to and from sub-systems or accounts from and to main systems or accounts.	O
P_LIQFRACC	CHAR(34)		Liquidity injection from account	For CNS systems, the ID of the account to and from which liquidity injections are booked. May also be used in RTGS systems for transferring liquidity to and from sub-systems or accounts from and to main systems or accounts.	O
P_LIQINJVA	DECIMAL (20,2)		Participant/account specific liquidity injection value	When specified, the injection value overrides any system-level value.	O
P_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.	O

4.2.3 Daily balances table [DBAL]

Field name	Data type	Key	Detailed name	Description	
B_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
B_SYSTEMID	CHAR(8)	P	System ID	Identifier of system.	M
B_PARTICID	CHAR(11)	P	Participant ID	Identifier of participant.	M
B_ACCOUNID	VARCHAR(34)	P	Account ID	Identifier for the account. This field is mandatory because it is a primary key.	M
B_DATEEFFE	INT	P	Date effective	Date opening balance is effective.	M
B_NEWVALUE	DECIMAL (20,2)		New value	Value of opening balance.	M
B_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.	O

4.2.4 Intraday changes in credit limit [ICCL]

Field name	Data type	Key	Detailed name	Description	
I_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
I_SYSTEMID	CHAR(8)	P	System ID	Identifier of system.	M
I_PARTICID	CHAR(11)	P	Participant ID	Identifier of participant.	M
I_ACCOUNID	VARCHAR(34)	P	Account ID	Identifier for the account. This field is mandatory because it is a primary key.	M
I_DATEEFFE	INT (11)	P	Date effective	Date from which new credit limit is effective.	M
I_TIMEEFFE	BIGINT (20,2)	P	Time effective	Time from which new credit limit is effective.	M
I_NEWVALUE	DECIMAL (20,2)		New value	Value of new credit limit.	M
I_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.	O
business_day	CHAR(8)	P	Business day	Business day to which the limit value belongs to. This value is used to determine the ICCL events to be included to the corresponding simulation day having the same business day.	M
Transaction_link_id	VARCHAR(30)		Transaction link ID	Id of the transaction linked to this credit limit order.	
order_type	VARCHAR(10)		Type of the iccl order		O
entry_date	INTEGER		Entry date	Date when the order becomes visible to the system	
entry_time	BIGINT(12)		Entry time	Time when the order becomes visible to the system	
revocation_date	INTEGER		Revocation date	Day when the iccl order is revoked	
revocation_time	BIGINT(12)		Revocation time	Time when the iccl order is revoked	
id	BIGINT(20)		Id	Unique identifier of the iccl order. It acts as a link to the output table.	

4.2.5 Bilateral limit table [BLIM]

Field name	Data type	Key	Detailed name	Description	
L_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
L_SYSTEMID	CHAR(8)	P	System ID	Identifier of the system this dataset belongs to.	
L_FRSYSTID	CHAR(8)	P	From System ID	Identifier of system. Default value will be the current system, i.e. same as L_SYSTEMID.	M
L_FRPARTID	CHAR(11)	P	From Participant ID	Identifier of participant.	M
L_FRACCOID	VARCHAR(34)	P	From Account ID	Identifier for the account. This field is mandatory because it is a primary key, but it has a default value of space character when the Account ID level is not in use.	M

L_TOSYSTID	CHAR(8)	P	To System ID	Identifier of receiving system. Default value will be the current system	M
L_TOPARTID	CHAR(11)	P	To Participant ID	Identifier of receiving participant. Bilateral limits can also be stated as a general multilateral sending limit towards all the other participants by setting the value as "*MULTILIMIT".	M
L_TOACCOID	CHAR(34)	P	To Account ID	Identifier for the receiving account. This field is mandatory because it is a primary key, but it has a default value of space character when the Account ID level is not in use.	M
L_DATEEFFE	INT(11)	P	Date effective	Date from which the new credit limit is effective.	M
L_TIMEEFFE	BIGINT(12)	P	Time effective	Time from which the new credit limit is effective.	M
L_NEWVALUE	DECIMAL (20,2)		Debit cap	Value of the new lower limit for the bilateral balance (or multilateral balance if counterparty is *MULTILIMIT). The debit cap will constrain outgoing payments, if resulting bilateral position would go below the limit. The value can be positive or negative. A negative value is the most common case. It indicates that net outflow of liquidity is allowed while positive value indicates a request for a reception surplus. A value of .99 indicates that no limit is in force. It can be used to remove limits that have been assigned earlier during the day. Defining the debit cap will start the recording of bilateral position if no limits defined in BLIM data were in place previously for the given pair of participants.	M
L_DBCVALUE	DECIMAL (20,2)		Credit cap	Value of the new upper limit for the bilateral balance (or multilateral balance if counterparty is *MULTILIMIT). The credit cap will constrain incoming payments if the resulting bilateral balance would go above the limit. The value can be positive or negative. A positive value is the most common case. It indicates that inflow of liquidity is allowed while negative value would be a request for a sending surplus. A value of .99 indicates that no limit is in force. It can be used to remove limits that have been assigned earlier during the day. Defining the credit cap will start the recording of bilateral position if no limits defined in BLIM data were in place previously for the given pair of participants. In projects, which are created with version 3.1.0 or later credit cap value can be imported either separately on an own row of BLIM data or together in a row which also has value for the debit cap with same time label. In older database versions, values which have same time label need to be always imported in one row.	O
L-USERCOD1…5	VARCHAR(16)		User-defined codes 1…5	Five optional fields where user-defined information can be stored for user-defined algorithms.	O

4.2.6 Reservations table [RSRV]

Field name	Data type	Key	Detailed name	Description	
R_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
R_SYSTEMID	CHAR(8)	P	System ID	Identifier of system.	M
R_PARTICID	CHAR(11)	P	Participant ID	Identifier of participant.	M
R_ACCOUNID	VARCHAR(34)	P	Account ID	Identifier for the account.	(M)
R_DATEEFFE	INT(11)	P	Date effective	Date from which new reservation is effective.	M
R_TIMEEFFE	BIGINT(12)	P	Time effective	Time from which new reservation is effective.	M
R_NEWVALUE	DECIMAL (20,2)		New value	Value of the new reservation	M
R_RESRVTYP	CHAR(1)		Reservation type	Type of the new reservation (H= highly urgent, U=urgent). If both reservation types are changed	M

				at the same time two update records are needed. Only positive values or zero are accepted.	
R_USERCOD1…5	VARCHAR(16)		User-defined codes 1…5	Five optional fields where user-defined information can be stored for user-defined algorithms.	O

4.2.7 Transaction data table [TRAN]

Field name	Data type	Key	Detailed name	Description	
T_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
T_SYSTEMID	CHAR(8)	P	System ID	Identifier of the system this dataset belongs to.	M
T_TRANSAID	CHAR(12)	P	Transaction ID	Unique identifier of transaction.	M
T_INTRDATE	INT(11)		Introduction date	Day of transaction.	M
T_INTRTIME	BIGINT(12)		Introduction time	Time of transaction.	M
T_TRANVALU	DECIMAL (20,2)		Transaction value	Value of transaction.	M
T_FRSYSTID	VARCHAR(8)		From system ID	System where the account defined below can be found.	M
T_FRPARTID	VARCHAR(11)		From participant ID	Participant or account from which payment is debited.	M
T_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used.	O
T_TOSYSTID	VARCHAR(8)		To system ID	System where the account defined below can be found.	M
T_TOPARTID	VARCHAR(11)		To participant ID	Participant or account to which payment is credited.	M
T_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used.	O
T_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments eg. interbank payments, customer payments,… . This categorization can be used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.	O
T_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the transactions same way as T_TRANCLAS. For example it can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.	
T_LINKCODE	VARCHAR(30)		Link code	A code used recognize all transaction belonging to a group. Linkcode can be used to link the different legs of e.g . DVP or PVP transactions.	O
T_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.	O
T_LINKTRNN	INT(11)		Linked transaction number	Count of the transaction linked by the same T_LINKCODE. Not used.	O
T_USERDEID	VARCHAR (50)		User defined ID	User-defined transaction ID that allows transaction to be compared in internal system runs.	O
T_DESCRIPT	VARCHAR(255)		Description	Text description of transaction.	O
T_ASSENAME	VARCHAR(20)		Asset name	Name of transaction asset.	O
T_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.	O
T_PRIORITY	TINYINT(4)		Priority	Value indicating importance of payment from 0-9, with 9 the highest priority. Used to order transactions in payment queues.	O
T_PROCTYPE	TINYINT(4)		Processing type	Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use)	O

				2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.	
T_PROCDATE	INT(11)		Processing date	Day processing takes place as defined in T_PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.	O
T_PROCTIME	BIGINT (12)		Processing time	Time processing takes place as defined in T_PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.	O
T_PROCTYP2	TINYINT(4)		Processing type 2	Gives the opportunity to introduce second set of control variables to affect the settlement of the transaction. Feature is not yet in use in general version 3.0.0. Default value: 0 – Not defined. Is set to 0 automatically during import if null or empty.	O
T_PROCDAT2	INT(11)		Processing date 2	Day processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYP2 is set to 0.	O
T_PROCTIM2	BIGINT (12)		Processing time 2	Time processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYP2 is set to 0.	O
T_ASBIC	CHAR(11)		Ancillary system bic code	Bic code of the ancillary system	
business_day	CHAR(8)		business day	Business day of the transaction. The field content is used to deduct the business days of a simulation if the selection: Business day deducted from transaction data is in force.	M
Iccl_link_id	VARCHAR(30)		Iccl link id	Link to the Transaction_link_id of the iccl table.	O

4.2.8 Transaction changes [trch]

Field name	**Data type**	**Key**	**Detailed name**	**Description**	
H_DATSETID	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
H_SYSTEMID	CHAR(8)	P	System ID	Identifier of the system this dataset belongs to.	M
H_TRANCHID	CHAR(12)	P	Transaction change ID	Transaction change ID	M
T_TRANSAID	CHAR(12)	F	Transaction ID	Unique identifier of transaction.	M
H_INTRDATE	INT(11)		Introduction date	Day of change request.	M
H_INTRTIME	BIGINT(12)		Introduction time	Time of change request.	M
H_CHPRIOR	TINYINT(4)		Priority of change request	Change request priority	O
H_NEWDATE	INT(11)		New date	New value for Day of transaction.	O
H_NEWTIME	BIGINT(12)		New time	New value for Time of transaction.	O
H_TRANCLAS	VARCHAR(8)		Transaction class	New value for T_ TRANCLAS	O
H_TRANCLA2	VARCHAR(8)		Transaction class 2	New value for T_ TRANCLA2	O
H_PRIORITY	TINYINT(4)		Priority	New priority value (T_PRIORITY) for the underlying transaction.	O
T_PROCTYPE	TINYINT(4)		Processing type	New value for T_PROCTYPE	O
T_PROCDATE	INT(11)		Processing date	New value for T_PROCDATE	O
T_PROCTIME	BIGINT (12)		Processing time	New value for T_PROCTIME	O
T_PROCTYP2	TINYINT(4)		Processing type 2	New value for T_PROCTYP2	O
T_PROCDAT2	INT(11)		Processing date 2	New value for T_PROCDAT2	O
T_PROCTIM2	BIGINT (12)		Processing time 2	New value for T_PROCTIM2	O
H_NEWUDC1...5	VARCHAR(16)		User defined codes 1...5	New values to replace the old user defined codes of the underlying transaction.	O
H_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.	O

H_CANCEL	TINYINT(4)		Cancellation of whole transaction	Binary value for cancelling the transaction. 1 = cancel 0 or null, no cancellation (i.e. some other change)	O

4.2.9 Simulation events [business_day_event]

Field name	Data type	Key	Detailed name	Description	
data_set_id	CHAR(8)	P	Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.	M
system_id	CHAR(8)	P	System ID	Identifier of system.	M
business_day	CHAR(8)	P	Business day	Business day to wich the event belongs to. This value is used to determine the events to be included to the corresponding simulation day having the same business day.	M
id	SMALLINT(6)	P	Account ID	Identifier for the event. The identifier is added during the import process according to the mapping defined in the PSS.properties files located within the BOF-PSS2.jar package.	
name	VARCHAR(20)		name of the event	name of the event	M
date	INT (11)		Date effective	Date when the event takes place	M
time	BIGINT (20,2)		Time effective	Time when the event takes place	M

4.2.10 System algorithms [SALG]

Field name	Data type	Key	Detailed name	Description	
A_DATSETID	CHAR(8)	P, F	Data set ID	Same data set ID as in the SYCD table.	M
A_SYSTEMID	CHAR(8)	P, F	System ID	Identifier of system.	M
A_ALGORIID	CHAR(8)	P, F	Algorithm ID	Unique identifier of algorithm.	M
A_SYALGOID	INT	P	ID of system algorithm	Unique identifier of system algorithm in a system definition. Defines the order in which algorithms are displayed on the system definition view's algorithm table.	M
A_ALGOTYPE	CHAR(3)		Algorithm type	Type of algorithm.	M
A_PARVALUE	TEXT		Parameter values	Enumerated list of parameters values used by algorithm. Parameters in form: parameter1?value][parameter2?value2][…[parameterlast?value	O
A_TEALGOID	CHAR(8)		TEA algorithm ID	ID of the time estimation algorithm (TEA) algorithm	O
A_TEALGPAR	TEXT		TEA parameters	parameters defined in the system definition for the selected TEA -algorithm	O
A_PARALPRO	TINYINT(4)		parallel processing mode indicator	Optional code for indicating whether the algorithm is processed in parallel. Possible values: 0 or empty: not processed in parallel 1: is processed in parallel	O
A_ALGDEFID	INT(11)	P, F	Algorithm definition ID	Unique identifier for algorithm definition	M

4.3 Output database

After each simulation, the result data is stored in a defined output database. Thus, the results of all closely related simulations can be kept in a single database to facilitate comparison of results. Output database contains tables for actual statistics and additional technical tables for simulation logs and batch run information. To reduce file space, the user can define which output statistics data tables are recorded for each simulation batch. A unique simulation run identifier identifies the information belonging to the same simulation run. All tables are optional, i.e. the user must define the necessary output for each simulation run. Sometimes only a small output sample is needed to determine which alternative is preferable for deeper analysis.

4.3.1 System level statistics [SYLS]

Field name	Data type	Key	Detailed name	Description
Y_SIMRUNID	CHAR(8)	P, F	Simulation run number	ID of associated simulation run.
Y_SYSTEMID	CHAR(8)	P	System ID	ID of system.
business_day	CHAR(8)	P	Business day	Date of the Business day(format YYYYMMDD).
Y_SYSTNAME	CHAR(20)		System name	Name of system.
Y_VALUDATA	DECIMAL (22,2)		Value in data	Total value of transactions in day's transaction data.
Y_VALUCARR	DECIMAL (22,2)		Value carried over	Total value of transactions carried over from previous day(s).
Y_VALUSUBM	DECIMAL (22,2)		Value submitted	Total value of transactions submitted to system by submission algorithm.
Y_VALUSETT	DECIMAL (22,2)		Value settled	Total value of transactions settled by settlement algorithms.
Y_VALUUNST	DECIMAL (22,2)		Value unsettled	Total value of transactions remaining unsettled during the day [Y-VALUDATA] + {VALUCARR] – [Y-VALUESETT].
Y_NUMBDATA	INT(11)		Number in data	Total number of transactions in day's transaction data.
Y_NUMBCARR	INT(11)		Number carried over	Total number of transactions carried over from previous day(s).
Y_NUMBSUBM	INT(11)		Number submitted	Total number of payments submitted to the system by submission algorithm.
Y_NUMBSETT	INT(11)		Number settled	Total number of transactions settled by settlement algorithms.
Y_NUMBUNST	INT(11)		Number unsettled	Total number of transactions remaining unsettled during the day [Y-NUMBDATA] + {NUMBCARR] – [Y-NUMBSETT].
Y_BODBALAN	DECIMAL (22,2)		Beginning-of -day balances	The sum of the day's initial balances of the participants/accounts.
Y_EODBALAN	DECIMAL (22,2)		End-of-day balances	The sum of the day's ending balances of the participants/accounts.
Y_AVGCRLIM	DECIMAL (22,2)		Average credit limit	The time weighted average of the available credit limits of the participants/accounts at system level.
Y_LIQAVAIL	DECIMAL (22,2)		Liquidity available	The sum of the beginning of day balances and the time weighted average intraday credit available to the participants/accounts during the day, i.e. Y_BODBALAN + Y_AVGCRLIM.
Y_ABSCLUSA	DECIMAL (22,2)		Absolute overdrafts	The sum of average overdrafts (negative balances) for the participants/accounts during the day.
Y_RELCLUSA	DECIMAL (22,2)		Relative credit limit usage	The average overdraft divided by the average credit limit for the participants/accounts during the day.
Y_TOTLIQAV	DECIMAL (22,2)		Total liquidity available	Total liquidity available across all participants during the day.
Y_LOWBOUND	DECIMAL (22,2)		Lower bound of liquidity	The sum of net liquidity requirement for the participants/accounts in the system (see Annex 1).
Y_MAXQUEVA	DECIMAL (22,2)		Maximum queue value	Maximum (peak) queue value during the day.
Y_AVEQUEVA	DECIMAL (22,2)		Average queue value	Average queue value during the day (the average time weighted value of queue balance).
Y_AVEQUELE	BIGINT(20)		Average queue length	Average queue duration for queued payments i.e. the sum of queuing time of queued payments divided by the total number of queued payments. Directly settled payments are not taken into account. With

				the system setup "Delete unsettled transactions (exclude from statistics), Unsettled transactions are not included in the average. (format hhhhmmss000, where 000 denotes milliseconds).
Y_QUENUMBE	INT(11)		Number of queued transactions	Number of queued transactions per day.
Y_QUETOTVA	DECIMAL (22,2)		Total value of queued transactions	Total value of queued transactions per day.
Y_QUESTTIM	BIGINT(20)		Queue stop time	Total time during the day that outgoing transactions were queued and the process was blocked due to insufficient liquidity for the participants i.e. the sum of the individual participant level queue stop times (format hhhhmmss000 where 000 denotes milliseconds). If many participants have long queues this value can be longer that the open hours.
Y_AVERTISE	BIGINT(20)		Average time of settlement	Simple average of queuing times of all payments. Note that also payments that are always settled directly by definition and that cannot be queued will also affect the average. With the system setup "Delete unsettled transactions (exclude from statistics), Unsettled transactions are not included in the average. (The database storing and export format is hhhhmmss000, the value is calculated with hhhmmss precision)
Y_LIQUSAGC	DECIMAL (22,2)		Liquidity usage indicator based on continuous calculation	Liquidity usage indicator based on consumed liquidity i.e. consumed overdrafts and reserve deposits compared with submitted volume. Calculation explained in document Annex 1.
Y_LIQUSAGR	DECIMAL (22,2)		Liquidity usage indicator based on available liquidity (rigid credit limits)	Liquidity usage indicator based on available liquidity (rigid credit limits) i.e. total credit limits compared with submitted volumes. Calculation explained in Annex 1.
Y_SETDELAY	DECIMAL (22,2)		Settlement delay indicator	Indicator of settlement delay i.e. actual delay compared to theoretic maximum delay at end of day. Calculation explained in Annex 1.
Y_SETTINGS	TEXT		Currently no value.	Reserved for future needs.
Y_MAXCRUSG	DECIMAL (22,2)		Peak credit usage	Peak value of credit used during the simulation.

4.3.2 Account statistics [ACST]

Field name	Data type	Key	Detailed name	Description
A_SIMRUNID	CHAR(8)	P, F	Simulation run ID	ID of simulation run.
A_SYSTEMID	CHAR(8)	P	System ID	ID of system.
A_PARTICID	CHAR(11)	P	Participant ID	ID of participant.
A_ACCOUNID	VARCHAR(34)	P	Account ID	ID of account.
A_ACCOTYPE	VARCHAR(1)		Account type	Used to distinguish different types of accounts.
business_day	CHAR(8)	P	Business day	Date of the business day (format YYYYMMDD).
A_DATSETID	CHAR(8)	P	Data set ID	ID of data set.
A_FULLNAME	VARCHAR(35)		Full name	Full name of participant.
A_SHORTACR	CHAR(5)		Short acronym	Acronym for full name of participant.
A_ACCONAME	CHAR(10)		Account name	Name of account, e.g. "Euro RTGS account."
A_SETINSYS	CHAR(8)		Settles in system	For DNS or CNS systems, the ID of the system where proceedings are booked. May also be used in RTGS systems for transferring end-of-day positions from sub-systems or accounts to main systems or accounts.
A_SETONPAR	CHAR(11)		Settles on participant	For DNS or CNS systems, the ID of the participant to whom the end-of-day proceedings are booked. May also be used in RTGS systems for transferring end-of-day positions from sub-systems or accounts to main systems or accounts.
A_SETONACC	CHAR(34)		Settles on account	For DNS or CNS systems, the ID of the account in which the end-of-day proceedings are booked. May also be used in RTGS systems for transferring end-of-day positions from sub-systems or accounts to main systems or accounts.
A_LIQFRSYS	CHAR(8)		Liquidity injection from system	For CNS systems, the ID of the system to and from which liquidity injections are booked. May also be used in RTGS

				systems for transferring liquidity to and from sub-systems or accounts from and to main systems or accounts.
A_LIQFRPAR	CHAR(11)		Liquidity injection from participant	For CNS systems, the ID of the participant to and from which liquidity injections are booked. May also be used in RTGS systems for transferring liquidity to and from sub-systems or accounts from and to main systems or accounts.
A_LIQFRACC	CHAR(34)		Liquidity injection from account	For CNS systems, the ID of the account to and from which liquidity injections are booked. May also be used in RTGS systems for transferring liquidity to and from sub-systems or accounts from and to main systems or accounts.
A_LIQINJVA	DECIMAL (22,2)		Participant/account specific liquidity injection value	When specified, the injection value overrides any system-level value.
A_VALUDATA	DECIMAL (22,2)		Value in data	
A_VALUCARR	DECIMAL (22,2)		Value carried over	
A_VALUSUBM	DECIMAL (22,2)		Value submitted	
A_VALUSETT	DECIMAL (22,2)		Value settled	
A_VALUUNST	DECIMAL (22,2)		Value unsettled	
A_VALURECE	DECIMAL (22,2)		Value received	
A_NUMBDATA	INT(11)		Number in data	
A_NUMBCARR	INT(11)		Number carried over	
A_NUMBSUBM	INT(11)		Number submitted	
A_NUMBSETT	INT(11)		Number settled	
A_NUMBUNST	INT(11)		Number unsettled	
A_NUMBRECE	INT(11)		Number received	
A_BODBALAN	DECIMAL (22,2)		Beginning-of-day balance	The day's initial balance.
A_EODBALAN	DECIMAL (22,2)		End-of-day balance	The day's ending balance.
A_AVEBALAN	DECIMAL (22,2)		Average balance during the day.	Average balance during the day.
A_MINBALAN	DECIMAL (22,2)		Minimum balance	Minimum balance during the day.
A_MAXBALAN	DECIMAL (22,2)		Maximum balance	Maximum balance during the day.
A_AVGCRLIM	DECIMAL (22,2)		Average credit limit	Weighted (time) average credit limits. In case of extending credits without restrictions, the automatically granted limit is assumed to be in force until the end-of-day or until more credit is extended.
A_AVELIQAV	DECIMAL (22,2)		Average liquidity available	Average liquidity available during the day, i.e. average balance plus relevant credit limit.
A_CREDUSAG	DECIMAL (22,2)		Credit limit usage i.e. average overdraft	Average overdraft during the day, i.e. average of the negative balances of the day.
A_CREDUSAP	DECIMAL (22,2)		Relative credit limit usage i.e. percentual overdraft	Average overdraft percentage during the day, i.e. average of the negative balances of the day compared to relevant total credit limit.
A_LOWBOUND	DECIMAL (22,2)		Lower bound of liquidity	See Annex 1.
A_UPPBOUND	DECIMAL (22,2)		Upper bound of liquidity	Upper bound of liquidity is defined as the amount of liquidity need for immediate settlement of all transactions (i.e. no queues). This is not calculated in the simulation, because it requires a special simulation run in which there are no limits on intraday credit. This field is reserved if the user wants to include this information in the table.
A_MAXQUEVA	DECIMAL (22,2)		Maximum queue value	Maximum queue value during the day.
A_AVEQUEVA	DECIMAL (22,2)		Average queue value	Average queue value during the day (average time weighted value of queue balance).

A_AVEQUELE	BIGINT(20)		Average queue length	Average queue duration for queued payments i.e. the sum of queuing time of queued payments divided by the total number of queued payments. Directly settled payments are not taken into account. With the system setup "Delete unsettled transactions (exclude from statistics), Unsettled transactions are not included in the average. (format hhhhmmss000000, where 000000 denotes microseconds).
A_QUENUMBE	INT(11)		Number of queued transactions	Number of queued transactions per day.
A_QUETOTVA	DECIMAL (22,2)		Total value of queued transactions	Total value of queued transactions per day.
A_QUESTTIM	BIGINT(20)		Queue stop time	Total time during the day that outgoing transactions were queued and the process was blocked due to insufficient liquidity for this account (format hhhhmmss000, where 000 denotes milliseconds).
A_AVERTISE	BIGINT(20)		Average time of settlement	Simple average of queuing times of all payments. Note that also such payments that are settled directly by definition and that cannot be queued will also affect the average. With the system setup "Delete unsettled transactions (exclude from statistics), Unsettled transactions are not included in the average. (format hhhhmmss000000, where 000000 denotes microseconds. The calculation precision is in seconds)
A_LIQUSAGC	DECIMAL (22,2)		Liquidity usage indicator based on continuous calculation	Liquidity usage indicator based on consumed liquidity i.e. consumed overdrafts and reserve deposits compared with submitted volume. Calculation explained in document Annex 1.
A_LIQUSAGR	DECIMAL (22,2)		Liquidity usage indicator based on available liquidity (rigid credit limits)	Liquidity usage indicator based on available liquidity (rigid credit limits) i.e. total credit limits compared with submitted volumes. Calculation explained in Annex 1.
A_SETDELAY	DECIMAL (22,2)		Settlement delay indicator	Indicator of settlement delay i.e. actual delay compared to theoretic maximum delay at end of day. Calculation explained in Annex 1.

4.3.3 Bilateral statistics table [BIST]

Field name	Data type	Key	Detailed name	Description
D_SIMRUNID	CHAR(8)	P, F	Simulation run ID	ID of simulation run.
D_FRSYSID	CHAR(8)	P	System ID	ID of system.
D_FRPARTID	CHAR(11)	P	Participant ID	ID of participant.
D_FRACCOID	VARCHAR(34)	P	Account ID	ID of account.
business_day	CHAR(8)	P	Business day	Date of business day (format YYYYMMDD).
D_TOSYSTID	CHAR(8)	P	Receiving in system	ID of receiving system
D_TOPARTID	CHAR(11)	P	Receiving participant	ID of receiving participant
D_TOACCOID	CHAR(34)	P	Receiving account	ID of receiving account
D_EODBALAN	DECIMAL (22,2)		End-of-day balance	The day's ending bilateral balance (a sending surplus is a negative balance).

4.3.4 Transaction event statistics [TEST]

Field name	Data type	Key	Detailed name	Description
E_INDEXNUM	BIGINT(20)	P	Settlement order index	Global settlement order index number starting from 1 for all transactions settled in one simulation to be used for sorting transactions in settlement order.
E_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of simulation run.
E_SYSTEMID	CHAR(8)	*	System ID	Identifier of system.
E_TRANSAID	CHAR(12)	*	Transaction ID	Unique identifier of transaction.
E_DATSETID	CHAR(8)		Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.
E_INTRDATE	INT(11)		Introduction date	Day of transaction (format YYYYMMDD).
E_INTRTIME	BIGINT(12)		Introduction time	Time of transaction (format hhmmss000000, where 000000 denotes microseconds).

E_TRANVALU	DECIMAL (22,2)		Transaction value	Value of transaction.
E_FRSYSTID	VARCHAR(8)		From system ID	System where the account defined below can be found.
E_FRPARTID	VARCHAR(11)		From participant ID	Participant or account from which payment is debited.
E_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used.
E_TOSYSTID	VARCHAR(8)		To system ID	System where the account defined below can be found.
E_TOPARTID	VARCHAR(11)		To participant ID	Participant or account to which payment is credited.
E_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used.
E_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments eg. interbank payments, customer payments,… . This categorization can be used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.
E_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the same way as T_TRANCLAS. For example It can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.
E_LINKCODE	VARCHAR(30)		Link code	A code used to recognize all transaction belonging to a group. Lincode can be used to link the different legs of e.g . DVP or PVP transactions.
E_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.
E_USERDEID	VARCHAR(50)		User defined ID	User-defined transaction ID that allows transaction to be compared in internal system runs.
E_DESCRIPT	VARCHAR(255)		Description	Text description of transaction.
E_ASSENAME	VARCHAR(20)		Asset name	Name of transaction asset.
E_USERCOD 1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.
E_PRIORITY	TINYINT(4)		Priority	Value indicating importance of payment from 0-9, with 9 the highest priority. Used to order transactions in payment queues.
E_PROCTYPE	TINYINT(4)		Processing type	Value transferred from the input database field T_PROCTYPE. Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use) 2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.
E_PROCDATE	INT(11)		Processing date	Day processing takes place as defined in T_PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
E_PROCTIME	INT(12)		Processing time	Time processing takes place as defined in T_PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
E_PROCTYP2	TINYINT(4)		Processing type 2	Value transferred from the input database field T_PROCTYP2.
E_PROCDAT2	INT(11)		Processing date 2	Value transferred from the input database field T_PROCDAT2.
E_PROCTIM2	BIGINT (12)		Processing time 2	Value transferred from the input database field T_PROCTIM2.
E_SUBMDATE	INT(11)		Submission date	Date transaction was submitted for settlement.
E_SUBMTIME	INT(12)		Submission time	Time transaction was submitted for settlement.
E_SETTDATE	INT(11)		Settlement date	Date transaction was settled.
E_SETTTIME	INT(12)		Settlement time	Time transaction was settled.
E_SUBMORIG	CHAR(8)		Submission origin of the transaction	ID of algorithm generating an internal transaction, or 0 if from the transaction data.
E_SETTALGO	CHAR(8)		Settling algorithm	The ID of algorithm that settled the transaction.
E_ORIGVALU	DECIMAL (22,2)		Original Value	Transaction’s original value (useful when the transaction was split).
E_SUBEVENT	SMALLINT(6)		Number of sub-event	Number of sub-event (useful when the transaction was split).

E_SENDACBA	DECIMAL (22,2)		Sending account balance	Sending account balance after settlement.
E_RECEACBA	DECIMAL (22,2)		Receiving account balance	Receiving account balance after settlement.
E_SETTSTAT	TINYINT(4)		Settlement status	Value indicating if transaction was settled: -1= unsettled because of defined latest debit time, 0=unsettled 1=settled directly 2=settled via queue 3=forced end of day settlement). 4=payment replaced due to process reasons and recorded for reference.
E_BILABALA	DECIMAL (22,2)		Bilateral balance	Bilateral balance seen from the sending account after the transaction has been settled
E_ENTRDATE	INT(11)		Entry date of transaction.	Date when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.
E_ENTRTIME	BIGINT(12)		Entry time of transaction.	Time when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.
E_ASBIC	CHAR(11)		Ancillary system bic code	BIC code of the ancillary system
business_day	CHAR(8)	I	Business day	Business day the transaction belongs to
Iccl_link_id	VARCHAR(30)		Iccl link id	Link to the Transaction_link_id of the iccl table. Link to the corresponding icl order.

4.3.5 Intraday credit limit order execution statistics [iccl_order_execution_statistics]

Field name	Data type	Key	Detailed name	Description
id	BIGINT(20)	P	Id	Id of the icl order acting as a link to the input database
Iccl_id	BIGINT(20)		Iccl id	Link to the original icl order in the iccl table of the input db.
simulation_id	VARCHAR(20)		Simulation id	Link to the simulation
dataset_id	VARCHAR(20)		Dataset id	Dataset id to which the order belongs
system_id	CHAR(8)		System	ID of associated system.
business_day	CHAR(8)		Business day	Business day the order belongs to
execution_status	INTEGER		Execution status	0 = rejected, 1 = executed directly, 2 = executed after being queued, 650 = pending credit line decrease replaced by new order, -3 = removed at cut off bank
entry_date	INTEGER		Entry date	
entry_time	BIGINT(12)		Entry time	
Resolution_date	INTEGER		Resolution date	Date when order was either removed from queue or executed
Resolution_time	BIGINT(12)		Resolution time	Time when order was either removed from queue or executed

4.3.6 Netting event statistics [NEST]

Field name	Data type	Key	Detailed name	Description
N_SIMRUNID	CHAR(8)	P, F	Simulation run ID	
N_SYSTEMID	CHAR(8)	P	System ID	ID of associated system.

N_ALGORIID	CHAR(8)	P	Algorithm ID	ID of netting algorithm.
N_NETTDATE	INTEGER	P	Resolution date	Date netting (e.g. gridlock resolution) was executed.
N_NETTTIME	BIGINT(12)	P	Resolution time	Time netting (e.g. gridlock resolution) was executed.
N_NETTINID	CHAR(8)		Netting ID	ID of associated net settlement.
N_TRANSVAL	DECIMAL (22,2)		Transaction value	Value of additional transactions generated by the netting algorithm.
N_TRANSVOL	INT		Transaction volume	Number of additional transactions generated by the netting algorithm.
N_SETTLVAL	DECIMAL (22,2)		Settled value	Value of original transactions settled by the netting algorithm.
N_SETTLVOL	DECIMAL (22,2)		Settled volume	Number of original transactions settled by the netting algorithm.
N_TOTALVAL	DECIMAL (22,2)		Total value	Value of all transactions subject to netting.
N_TOTALVOL	DECIMAL (22,2)		Total volume	Volume of all transactions subject to netting.

4.3.7 Account violation statistics [AVST]

Field name	Data type	Key	Detailed name	Description
V_SIMRUNID	CHAR(8)	P, F	Simulation run ID	ID of associated simulation run.
V_SYSTEMID	CHAR(8)	P	System ID	System in which violation occurred.
V_PARTICID	CHAR(11)	P	Participant ID	ID of participant for which violation occurred.
V_ACCOUNID	VARCHAR(34)	P	Account ID	ID of account in which violation occurred.
V_EVENDATE	INT(11)	P	Event date	Date violation occurred.
V_EVENTIME	BIGINT(12)	P	Event time	Time violation occurred.
V_VIOLCAUS	VARCHAR (12)	P	Cause of violation	Reason for violation. Typically, forced end-of-day settlement or credit limit reduction. Value ICCL when depends on new lower credit limit, ANCSETTL when depends on ancillary system settlements and the value equal a transaction ID when the violation is caused by a forced end-of-day settlement.
V_VIOLVALU	DECIMAL (22,2)		Violation value	Value of violation.
business_day	CHAR(8)		Business day	Date of the business day

4.3.8 Booking event statistics [BEST] (not supported any more)

Can be filtered from the test table

Field name	Data type	Key	Detailed name	Description
O_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of simulation run.
O_SYSTEMID	CHAR(8)	*	System ID	Identifier of system.
O_TRANSAID	CHAR(12)	*	Transaction ID	Unique identifier of transaction.
O_DATSETID	CHAR(8)		Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.
O_INTRDATE	INT(11)		Introduction date	Day of transaction.
O_INTRTIME	BIGINT(12)		Introduction time	Time of transaction.
O_TRANVALU	DECIMAL (22,2)		Transaction value	Value of transaction.
O_FRSYSTID	VARCHAR(8)		From system ID	System where the account defined below can be found.
O_FRPARTID	VARCHAR(11)		From participant ID	Participant or account from which payment is debited.
O_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used.
O_TOSYSTID	VARCHAR(8)		To system ID	System where the account defined below can be found.
O_TOPARTID	VARCHAR(11)		To participant ID	Participant or account to which payment is credited.
O_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used.
O_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments eg. interbank payments, customer payments,… . This categorization can be

				used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.
O_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the same way as T_TRANCLAS.For example It can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.
O_LINKCODE	VARCHAR(30)		Link code	A code used to recognize all transaction belonging to a group. Lincode can be used to link the different legs of e.g. DVP or PVP transactions.
O_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.
O_USERDEID	VARCHAR(50)		User defined ID	User-defined transaction ID that allows transaction to be compared in internal system runs.
O_DESCRIPT	VARCHAR(255)		Description	Text description of transaction.
O_ASSENAME	VARCHAR(20)		Asset name	Name of transaction asset.
O_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.
O_PRIORITY	TINYINT(4)		Priority	Value indicating importance of payment from 0-9, with 9 the highest priority. Used to order transactions in payment queues.
O_PROCTYPE	TINYINT(4)		Processing type	Value transferred from the input database field T_PROCTYPE. Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use) 2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.
O_PROCDATE	INT(11)		Processing date	Day processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
O_PROCTIME	BIGINT (12)		Processing time	Time processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
O_PROCTYP2	TINYINT(4)		Processing type 2	Value transferred from the input database field T_PROCTYP2.
O_PROCDAT2	INT(11)		Processing date 2	Value transferred from the input database field T_PROCDAT2.
O_PROCTIM2	BIGINT (12)		Processing time	Value transferred from the input database field T_PROCTIM2.
O_SUBMDATE	INT		Submission date	Date transaction was submitted for settlement.
O_SUBMTIME	INT		Submission time	Time transaction was submitted for settlement.
O_SETTDATE	INT		Settlement date	Date transaction was settled.
O_SETTTIME	INT		Settlement time	Time transaction was settled.
O_SUBMORIG	CHAR(8)		Submission origin of the origin	ID of algorithm generating an internal transaction, or 0 if from the transaction data.
O_SETTALGO	CHAR(8)		Settling algorithm	The ID of algorithm that settled the transaction.
O_SENDACBA	DECIMAL (15,2)		Sending account balance	Sending account balance after settlement.
O_RECEACBA	DECIMAL (15,2)		Receiving account balance	Receiving account balance after settlement.
O_ENTRDATE	INT(11)		Entry date of transaction.	Date when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.
O_ENTRTIME	BIGINT(12)		Entry time of transaction.	Time when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.

4.3.9 Queue reason information [QURE]

Field name	Data type	Key	Detailed name	Description
K_SIMRUNIDI	CHAR(8)	P	Simulation run ID	ID of the simulation run
K_SYSTEMID	CHAR(8)	P	System ID	Identifier of system.
K_TRANSAID	CHAR(12)	P	Transaction ID	Identifier of transaction.
R_DATEMODI	INT(11)	P	Date modified	Date when change in queuing reason.
R_TIMEMODI	BIGINT(12)	P	Time modified	Time when change in queuing reason.
R_QURECODE	TINYINT(4)		Queue reason code	The queue reason code can take following values: 0 = queued due to process reasons. e.g. deferred system or payment is always settled via queue 1 = not enough liquidity on the account, 2 = bilateral limit exhausted and 3 = multilateral limit exhausted, when transactions are placed in queue. 4 = bilateral credit cap is limiting 5= multilateral credit cap is limiting 9 = Transaction is cleared from any queue 10 = bilateral credit cap exhausted 11 = multilateral credit cap exhausted 12 = FIFO, blocking payment in queue

4.3.10 Unsettled transactions statistics [UNST] (not supported any more)

Can be filtered from the test table

Field name	Data type	Key	Detailed name	Description
U_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of associated simulation run.
U_SYSTEMID	CHAR(8)	*	System ID	Identifier of system.
U_TRANSAID	CHAR(12)	*	Transaction ID	Unique identifier of transaction.
U_DATSETID	CHAR(8)		Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.
U_INTRDATE	INT(11)		Introduction date	Day of transaction.
U_INTRTIME	BIGINT(12)		Introduction time	Time of transaction.
U_TRANVALU	DECIMAL (22,2)		Transaction value	Value of transaction.
U_FRSYSTID	VARCHAR(8)		From system ID	System where the account defined below can be found.
U_FRPARTID	VARCHAR(11)		From participant ID	Participant or account from which payment is debited.
U_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used.
U_TOSYSTID	VARCHAR(8)		To system ID	System where the account defined below can be found.
U_TOPARTID	VARCHAR(11)		To participant ID	Participant or account to which payment is credited.
U_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used.
U_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments eg. interbank payments, customer payments,… . This categorization can be used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.
U_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the same way as T_TRANCLAS. For example It can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.
U_LINKCODE	VARCHAR(30)		Link code	A code used recognize all transaction belonging to a group. Lincode can be used to link the different legs of e.g. DVP or PVP transactions.
U_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.
U_USERDEID	VARCHAR(50)		User defined ID	User-defined transaction ID that allows transaction to be compared in internal system runs.
U_DESCRIPT	VARCHAR(255)		Description	Text description of transaction.

U_ASSENAME	VARCHAR(20)		Asset name	Name of transaction asset.
U_USERCOD 1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.
U_PRIORITY	TINYINT(4)		Priority	Value indicating importance of payment from 0-9, with 9 the highest priority. Used to order transactions in payment queues.
U_PROCTYPE	TINYINT(4)		Processing type	Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use) 2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.
U_PROCDATE	INT(11)		Processing date	Day processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
U_PROCTIME	BIGINT(12)		Processing time	Time processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
U_PROCTYP2	TINYINT(4)		Processing type 2	Value transferred from the input database field T_PROCTYP2.
U_PROCDAT2	INT(11)		Processing date 2	Value transferred from the input database field T_PROCDAT2.
U_PROCTIM2	BIGINT (12)		Processing time 2	Value transferred from the input database field T_PROCTIM2.
U_SUBMDATE	INT(11)		Submission date	Date transaction was submitted for settlement.
U_SUBMTIME	BIGINT(12)		Submission time	Time transaction was submitted for settlement.
U_SUBMORIG	CHAR(8)		Transaction origin	ID of algorithm that generated: the internal transaction or 0 if it is from transaction data.
U_ENTRDATE	INT(11)		Entry date of transaction.	Date when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.
U_ENTRTIME	BIGINT(12)		Entry time of transaction.	Time when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.

4.3.11 Submitted transactions statistics [SUST] (not supported any more)

Can be filtered from the test table

Field name	Data type	Key	Detailed name	Description
X_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of associated simulation run.
X_SYSTEMID	CHAR(8)	*	System ID	Identifier of system.
X_TRANSAID	CHAR(12)	*	Transaction ID	Unique identifier of transaction.
X_DATSETID	CHAR(8)		Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in the simulations.
X_INTRDATE	INT(11)		Introduction date	Day of transaction.
X_INTRTIME	BIGINT(12)		Introduction time	Time of transaction.
X_TRANVALU	DECIMAL (22,2)		Transaction value	Value of transaction.
X_FRSYSTID	VARCHAR(8)		From system ID	System where the account defined below can be found.
X_FRPARTID	VARCHAR(11)		From participant ID	Participant or account from which payment is debited.

X_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used.
X_TOSYSTID	VARCHAR(8)		To system ID	System where the account defined below can be found.
X_TOPARTID	VARCHAR(11)		To participant ID	Participant or account to which payment is credited.
X_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used.
X_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments e.g.. interbank payments, customer payments,… . This categorization can be used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.
X_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the same way as T_TRANCLAS. For example it can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.
X_LINKCODE	VARCHAR(30)		Link code	A code used to recognize all transaction belonging to a group. Lincode can be used to link the different legs of e.g. DVP or PVP transactions.
X_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.
X_USERDEID	VARCHAR(502)		User defined ID	User-defined transaction ID that allows transaction to be compared in internal system runs.
X_DESCRIPT	VARCHAR(255)		Description	Text description of transaction.
X_ASSENAME	VARCHAR(20)		Asset name	Name of transaction asset.
X_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields where user-defined information can be stored for use by user-defined algorithms during simulations or in analysis of simulation output.
X_PRIORITY	TINYINT(4)		Priority	Value indicating importance of payment from 0-9, with 9 the highest priority. Used to order transactions in payment queues.
X_PROCTYPE	TINYINT(4)		Processing type	Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use) 2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.
X_PROCDATE	INT(11)INT		Processing date	Day processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
X_PROCTIME	BIGINT(12)INT		Processing time	Time processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
X_PROCTYP2	TINYINT(4)		Processing type 2	Value transferred from the input database field T_PROCTYP2.
X_PROCDAT2	INT(11)		Processing date 2	Value transferred from the input database field T_PROCDAT2.
X_PROCTIM2	BIGINT (12)		Processing time 2	Value transferred from the input database field T_PROCTIM2.
X_SUBMDATE	INT(11)INT		Submission date	Date transaction submitted for settlement.
X_SUBMTIME	BIGINT(12)INT		Submission time	Time transaction submitted for settlement.
X_SUBMORIG	CHAR(8)		Transaction origin	ID of algorithm that generated internal transaction, or 0 if from the transaction data.
X_ENTRDATE	INT(11)		Entry date of transaction.	Date when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.
X_ENTRTIME	BIGINT(12)		Entry time of transaction.	Time when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.

4.3.12 Queued transactions statistics [QUST]

Field name	Data type	Key	Detailed name	Description
Q_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of associated simulation run.
Q_SYSTEMID	CHAR(8)	*	System ID	Identifier for system.
Q_TRANSAID	CHAR(12)	*	Transaction ID	Unique identifier of transaction.
Q_DATSETID	CHAR(8)		Data set ID	Unique identifier of data set, distinguishing this data set from other parallel data sets used in simulations.
Q_INTRDATE	INTEGER		Introduction date	Day of transaction.
Q_INTRTIME	BIGINT(12)		Introduction time	Time of transaction.
Q_TRANVALU	DECIMAL (22,2)		Transaction value	Value of transaction.
Q_FRSYSTID	VARCHAR(8)		From system ID	System where the account defined below can be found.
Q_FRPARTID	VARCHAR(11)		From participant ID	Participant or account from which payment is debited.
Q_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used.
Q_TOSYSTID	VARCHAR(8)		To system ID	System where the account defined below can be found.
Q_TOPARTID	VARCHAR(11)		To participant ID	Participant or account to which payment is credited.
Q_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used.
Q_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments eg. interbank payments, customer payments,… . This categorization can be used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.
Q_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the same way as T_TRANCLAS.For example It can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.
Q_LINKCODE	VARCHAR(30)		Link code	A code used recognize all transaction belonging to a group. Lincode can be used to link the different legs of e.g . DVP or PVP transactions.
Q_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.
Q_USERDEID	CHAR(11)		User defined ID	User-defined transaction ID so transaction can be compared to internal system runs.
Q_DESCRIPT	VARCHAR(255)		Description	Text description of transaction.
Q_ASSENAME	VARCHAR(20)		Asset name	Name of asset of the transaction.
Q_USERCOD1...5	VARCHAR(16)		User defined codes 1...5	Five optional fields in which user-defined information can be stored for use by user-defined algorithms during simulations or for analysis of simulation output.
Q_PRIORITY	TINYINT(4)		Priority	Value indicating the importance of the payment values from 0-9, with 9 as the highest priority. Used for ordering transactions in payment queues.
Q_PROCTYPE	TINYINT(4)		Processing type	Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use) 2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.
Q_PROCDATE	INT(11)		Processing date	Day processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
Q_PROCTIME	BIGINT(12)		Processing time	Time processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
Q_PROCTYP2	TINYINT(4)		Processing type 2	Value transferred from the input database field T_PROCTYP2.
Q_PROCDAT2	INT(11)		Processing date 2	Value transferred from the input database field T_PROCDAT2.
Q_PROCTIM2	BIGINT (12)		Processing time 2	Value transferred from the input database field T_PROCTIM2.

Q_QUEIDATE	INT(11)		Queue introduction date	Date transaction was introduced to the queuing system. Simulation date at the time when the transaction is put to queue. IF transfer unsettled transactions to next day is selected, for consecutive business days, the start of business day's calendar day is used.
Q_QUEITIME	BIGINT(12)		Queue introduction time	Time transaction was queued. Simulation time at the time when the transaction is put to queue. IF transfer unsettled transactions to next day is selected, for consecutive business days, the start of business day's time is used.
Q_QUERDATE	INT(11)		Queue release date	
Q_QUERTIME	BIGINT(12)		Queue release time	
Q_QUERCAU	VARCHAR(20)		Cause of queue release	Reason for queue release, incoming payment, increased liquidity, bilateral offsetting or other settlement/gridlock resolution algorithm. Possible values: - Name of algorithm: when the payment has been settled by a netting algorithm. - End of day removal: When payments are removed from queues at the end of calendar day as indicated by the closing time of the system. - Forced end of day settlement: when the selection "forced end of day settlement has been done in the system definition The field can contain also some other specific values.
Q_ENTRDATE	INTEGER		Entry date of transaction.	Date when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.
Q_ENTRTIME	BIGINT(12)		Entry time of transaction.	Time when transaction is finally entered into clearing process in the simulated system. The time label can be different from submission time due to delays caused by simulated parallel processing or TEA time estimation. All statistics are based on submission time, not on entry time.

4.3.13 Comment transactions statistics [CTST] (not supported any more)

Field name	**Data type**	**Key**	**Detailed name**	**Description**
C_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of associated simulation run.
C_SYSTEMID	CHAR(8)	*	System ID	Identifier for system.
C_TRANSAID	CHAR(12)	*	Transaction ID	Unique identifier of the transaction.
C_DATSETID	CHAR(8)		Data set ID	Unique identifier of the data set to distinguish this data set from other parallel data sets used in simulations.
C_INTRDATE	INT(11)		Introduction date	Day of transaction.
C_INTRTIME	BIGINT(12)		Introduction time	Time of transaction.
C_TRANVALU	DECIMAL (15,2)		Transaction value	Value of transaction.
C_FRSYSTID	VARCHAR(8)		From system ID	The system in which the account defined below can be found.
C_FRPARTID	VARCHAR(11)		From participant ID	The participant or account from which the payment is debited.
C_FRACCOID	VARCHAR(34)		From account ID	Sub-account from which payment is debited. Mandatory only when sub-accounts are used..
C_TOSYSTID	VARCHAR(8)		To system ID	The system in which the account defined below can be found.
C_TOPARTID	VARCHAR(11)		To participant ID	The participant or account to which the payment is credited.
C_TOACCOID	VARCHAR(34)		To account ID	Sub-account to which payment is credited. Mandatory only when sub-accounts are used..
C_TRANCLAS	VARCHAR(8)		Transaction class	Transaction class is used to categorize payments eg. interbank payments, customer payments,… . This categorization can be used for variable purposes in specific algorithms and some parts of the processes. The main available algorithms do not use this information.

C_TRANCLA2	VARCHAR(8)		Transaction class 2	Transaction class 2 is used to categorize the same way as T_TRANCLAS.For example It can be used to direct payments to different queues or to be settled by different algorithms. The main available algorithms do not use this information.
C_LINKCODE	VARCHAR(30)		Link code	A code used recognize all transaction belonging to a group. Lincode can be used to link the different legs of e.g . DVP or PVP transactions.
C_LINKSYST	CHAR(8)		Linked system	ID of system in which the other leg of the transaction is settled.
C_USERDEID	CHAR(22)		User defined ID	User-defined transaction ID. Allows comparison of transaction to internal system runs.
C_DESCRIPT	VARCHAR(255)		Description	Text description of transaction..
C_ASSENAME	VARCHAR(20)		Asset name	Name of asset in the transaction.
C_USERCOD 1...5	VARCHAR(12)		User defined codes 1...5	Five optional fields where user-defined information may be stored for user-defined algorithms during simulations or in the analysis of simulation output.
C_PRIORITY	TINYINT(4)		Priority	Value indicating the importance of the payment values from 0-9, with 9 the highest priority. Used for ordering transactions in payment queues.
C_PROCTYPE	TINYINT(4)		Processing type	Gives the opportunity to introduce various delayed processing options for transactions at a reference time. Possible values: 0 – Not defined. Is set to 0 automatically during import if null or empty. 1 – This transaction is settled exactly at the time described in T-PROCTIME and T-PROCDATE attributes. (Not in use) 2 – This transaction is not settled before the time described in T-PROCTIME and T-PROCDATE attributes.
C_PROCDATE	INT(11)		Processing date	Day processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
C_PROCTIME	BIGINT(12)		Processing time	Time processing takes place as defined in T-PROCTYPE. Is set to -1(not defined) during the import process if the T_PROCTYPE is set to 0.
C_PROCTYP2	TINYINT(4)		Processing type 2	Value transferred from the input database field T_PROCTYP2.
C_PROCDAT2	INT(11)		Processing date 2	Value transferred from the input database field T_PROCDAT2.
C_PROCTIM2	BIGINT (12)		Processing time 2	Value transferred from the input database field T_PROCTIM2.
C_COMIDATE	INT(11)		Comment introduction date	Date the commented transaction was introduced or changed.
C_COMITIME	BIGINT (12)		Introduction time	Time when the transaction was introduced or changed.
C_COMMENTS	VARCHAR(255)		Comment	
C_COMMCAUS	VARCHAR(50)		The cause for the comment	Reason for the comment.

4.3.14 Comment intraday credit statistics [CCST] (not supported any more)

Field name	Data type	Key	Detailed name	Description
D_SIMRUNID	CHAR(8)	F	Simulation run ID	ID of associated simulation run.
D_SYSTEMID	CHAR(8)	*	System ID	Identifier of system.
D_PARTICID	CHAR(11)	*	Participant ID	Identifier of participant.
D_DATEEFFE	INT(11)	*	Date effective	Date from which new credit limit is effective.
D_TIMEEFFE	BIGINT(12)	*	Time effective	Time from which new credit limit is effective.
D_ACCOUNID	VARCHAR(34)	*	Account ID	Identifier of account.
D_DATSETID	CHAR(8)		Data set ID	Unique identifier of data set to distinguish this data set from other parallel data sets used in simulations.
D_NEWVALUE	DECIMAL (22,2)		New value	Value of new credit limit.
D_COMIDATE	INT(11)		Comment introduction date	Date when commented intraday credit limit was changed.
D_COMITIME	BIGINT(12)		Introduction time	Time when change was made.
D_COMMENTS	VARCHAR(255)		Comment	
D_COMMCAUS	VARCHAR(50)		The cause for the comment	Reason for comment.

4.3.15 Technical output database tables

The simulator output database has three technical tables. BARI is related to simulation batches, SIRI contains defined simulation ID's and their information and BACO contains comparison view details.

Batch run information [BARI]

Field name	Data type	Key	Detailed name	Description
R_SIMBATID	CHAR(8)	P	Simulation batch name	Name of simulation batch.
R_NROFRUNS	SMALLINT (6)		Number of runs	Number of simulation runs in batch.
R_PROCTIME	BIGINT(12)		Processing time	Total processing time for simulation batch.
R_SIMRUNID	TEXT		Simulation runs included in the batch	The IDs of simulation runs included in batch.

Simulation run information [SIRI]

Field name	Data type	Key	Detailed name	Description
M_SIMRUNID	CHAR(8)	P	Simulation run ID	Simulation run ID.
M_SIMUNAME	VARCHAR(20)		Name of the simulation	
M_SIMDESCR	VARCHAR(120)		Description of the simulation	
M_PROCDATE	INT(11)		Processing date	Processing date for simulation run (format YYYYMMDD).
M_PROCTIME	BIGINT(12)		Processing time	Processing time for simulation run (format hhmmss000 where 000 stands for milliseconds).
M_DURATION	INT(11)		Simulation duration	Duration of simulation run (format hhmmsss000 where 000 stands for milliseconds, where the milliseconds can also have specific values).
M_SYSTEMID	TEXT		System Ids	System IDs belonging to the simulation. Values in form: system1,system2,…
M_OUTPTABL	TEXT		Output tables	Output data selected for the simulation. Values in form: SYLS,ACST,…
M_SYCDDSID	TEXT		System control data set IDs	Data set IDs of systems belonging to the simulation. Values in form: dataset1,dataset2,…
M_PARTDSID	TEXT		Participant data table data set Ids	Data set IDs of participants belonging to the simulation. Values in form: dataset1,dataset2,…

M_DBALDSID	TEXT		Daily balances table data set Ids	Data set IDs of balances belonging to the simulation. Values in form: dataset1,dataset2,…
M_ICCLDSID	TEXT		Intraday changes in credit limit data set Ids	Data set IDs of credit limits belonging to the simulation. Values in form: dataset1,dataset2,…
M_TRANDSID	TEXT		Transaction data table data set ids	Data set IDs of transactions belonging to the simulation. Values in form: dataset1,dataset2,…
M_BUSDESID	TEXT		Business day event table data set ids	Data set Ids indicating the events data set to be used in the simulation.
M_NUMBSYST	SMALLINT(6)		Number of systems	Number of systems belonging to the simulation.
M_NUMBPART	MEDIUMINT(9)		Number of participants	Number of participants belonging to the simulation.
M_NUMBTRAN	INT(11)		Number of transactions	Number of transactions belonging to the simulation.
M_SUBALGID	CHAR(8)		Submission algorithm ID	Identifier of submission algorithm.
M_ALGOTYPE	CHAR(8)		Algorithm type	Type of algorithm, value 'SUB'.
M_SUBPARAM	TEXT		Parameter values	Enumerated list of parameters.
M_BLIMDSID	TEXT		Intraday changes in bilateral limit data set Ids	Data set IDs of bilateral limits belonging to the simulation. Values in form: dataset1,dataset2,…
M_RSRVDSID	TEXT		Data set ID in RSRV table	Data set Ids of reservation data used in the simulation. Values in form: dataset1,dataset2,…

Basic comparison view [BACO]

Field name	Data type	Key	Detailed name	Description	
SB_CVIEWID	CHAR(8)	P	Comparison view ID	Unique identifier of comparison view template.	M
SB_CVITYPE	CHAR(1)	P	Comparison view type	Type of comparison view (system or account).	M
SB_CVALTYP	CHAR(1)		Comparison value type	Differences shown in percentage (0), absolute value (1), differences not shown (2) or differences shown in percentage and absolute value (3).	M
SB_SIMULAT	TEXT		Simulations	List of simulation IDs belonging to the system and account level comparisons. Values in form: simulation1,simulation2,…	M
SB_SYSTEMS	TEXT		Systems	List of system IDs in the comparison. Values in form: system1,system2,…	M
SB_SFIELDS	TEXT		Selected fields	List of field identifiers in the comparison. Values in form: field1,field2,…	M

Applicationruns [Applicationruns] (Not in use)

Field name	Data type	Key	Detailed name	Description	
ID	CHAR(8)	P	ID of application run		M
SetupID	CHAR(1)	P			M
SetupName	CHAR(1)				M
RunInfo	TEXT				M
StartTime	TEXT				M
EndTime	TEXT				M
USERCOD 1...5					

Process log [Processlog]

Field name	Data type	Key	Detailed name	Description	
ID	BIGINT(12)	P	ID	Technical ID for entry used as unique key	
ApplicationRunID	VARCHAR(8)	F	Application RUN ID	Link to Application run ID	
ProcessRunID	BIGINT(12)		Process run ID	All log rows associated to one algorithm run are stored with the same ProcessRunID	
ProcessName	VARCHAR (30)		Name of Process	Name of Algorithm or process	
Event	VARCHAR (30)		Event Type	Start of algorithm Start of algorithm postponed End of algorithm execution	
Description	VARCHAR (255)			Field for additional information. Content can be dynamically formed by the algorithm running to mediate information to the user. This can be used for debugging and validating the functioning of some features..	
Date	INT(11)			Current date when the row is logged	
Time	BIGINT(12)			Time when the row is logged	
SimDate	INT(11)		Simulation Date	Day in simulation	
SimTime	BIGINT(12)		Simulation Time	Time in simulation	
Info	VARCHAR(16)				
USERCOD 1...5	VARCHAR (16)				

5 Files

The simulator creates separate default directories by project for the different file types used (see Figure 7):

- Input file directory
- Output file directory
- Error list directory
- Output report directory
- Network files directory
- Network reports directory

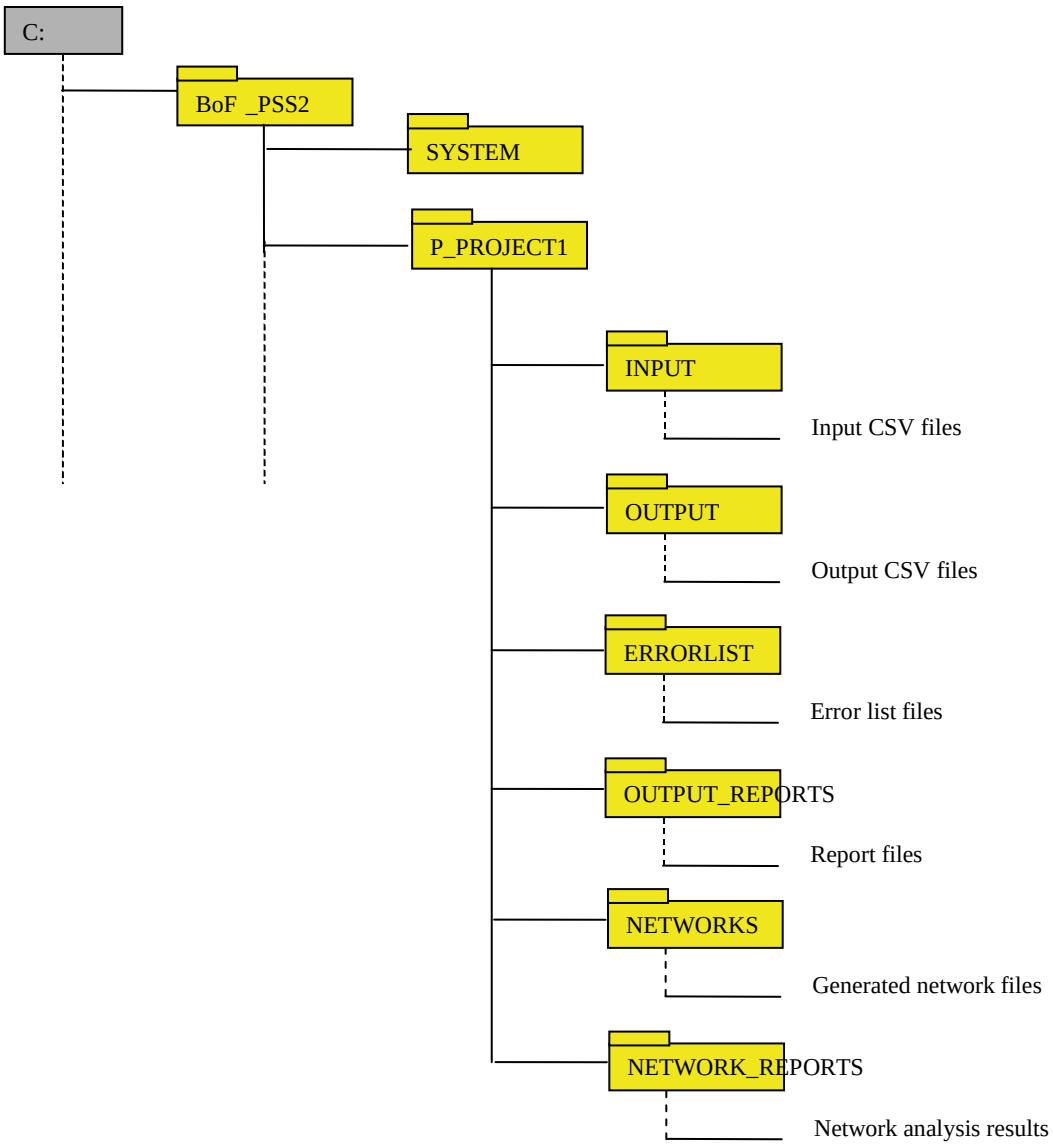

Figure 7.

5.1 Error list

When errors are found, an error list is generated. The name of the file is errorlist_date_time and the file type is plain text / comma separated values (.csv). The list contains:

Row	Description
1	The heading “Errors.”
2	Informs where errors have arisen. Values “simulation execution”, “import input data” or “cross-check input data.”
3	Empty row.
4	In this underlined row is stated - in case of import input date the faulty data type (corresponds to input data tables), the data set ID and the system ID - in case of cross-check input data simulation ID, system ID, data set ID and data type - in case of simulation execution simulation batch ID, simulation ID, system ID, data set ID
5	Empty row.
6	Faulty row in input file or in database table.
7	Error code and description of the error.
8-…	If the same data type, simulation ID, system ID and data set ID have more errors, they are listed as rows 5, 6 and 7. If the data type, simulation ID, system ID or data set ID change, rows 3 and 4 are written before rows 5, 6 and 7. These rows are written until all errors have been listed.

5.2 Basic statistics reports

The Analyser Tool allows the calculation of a number of further statistics, mainly time series and cross tables based on the statistics in the output tables. The reports can be saved as CSV and opened and viewed with a spreadsheet application, e.g. Excel. The names of the reports are systemstatistics, accountstatistics, systimeseriesstatistics, acctimeseriesstatistics.

The structure of the reports appears below.

Row	Col	Data type	Description
1	1	Varchar(50)	The heading gives the type of report. Values are "system statistics," "account statistics," "system time series statistics" or "account time series statistics."
2	1	Char(8)	Simulation ID.
2	2	Char(8)	System ID (not in system statistics).
2	3	Char(23)	Account ID (only in account time series statistics).
3			Empty row.
4	1	Varchar(7)	Headings. The first heading is "system" in system statistics, "account" in account statistics and "time" in the system and account time-series statistics.
4	2…n	Varchar(50)	The other headings are described in document /2/ Annex 2 (system statistics, account statistics, system time series, account time series: column entries).
5	1	Char(8), char(23) or char(10)	The corresponding data of the heading (system, account or time).
5	2…n	Depending on the corresponding field.	The corresponding data of the headings (document /2/ Annex 2).
6…n	1…n		The row 5 will be repeated as long as there are some data.

5.3 CSV and Excel files

The program creates CSV files from reports and export data. Data are moved to files from the saved database tables. Data in the CSV files are separated from each other by a separator, which the user can select. The extension is .csv for the CSV file. If the user has defined it, the first row in the file consists of field headings.

6 Annex 1 Calculation of specific indicators

Trough this annex text following notations are used.

$n \in N$	Number of participants (or accounts) in the system.
$i \in \{1,...,n\}$	Index number pointing to one particular participant.
$d \in N$	Total number of payments send in the system over the course of business day.
d_i	Number of payments sent by bank *i*.
i,k	Pair of index numbers pointing to one particular payment *k* of participant *i*. Here $k \in \{1,...,d_i\}$

Lower bound liquidity demand Y_LOWBOUND and A_LOWBOUND

On the low extreme all banks might have just enough liquidity to settle all the day's payments before the end of the day by using multilateral net settlement to solve gridlock situations. We shall refer to this amount of liquidity as the lower bound of liquidity *LB*. The lower bound of liquidity [A_LOWBOUND] for the *i*th participant/account LB_i can be written as

$$LB_i = \max\left(0, \sum_{k=1}^{d_i} a_{i,k} - \sum_{j=1}^{n}\sum_{k=1}^{d_j} a_{j,k}\Big|_{(r_{j,k}=i)}\right), \text{ where}$$

$a_{j,k} \in \Re_+$ = the value of payment *k* of participant *j*.

$r_{j,k} \in \{1,..., j-1, j+1,...,n\}$ = the receiver of payment.

The first sum is the value of payments send and the second sum is the value of the payments received over the course of the business day by bank *i*.

If the value of payments received during the day is larger than the value of payments sent, a participant/account only needs to use the liquidity it receives in the form of incoming payments for settling its own payments and thus the lower bound equals zero. If the value of payments sent exceeds the value of payments received, the difference has to be available at least at the end of the day.

Lower bound of liquidity in the system level [Y_LOWBOUND] is simply the sum of lower bounds of individual participants/accounts.

Settlement delay Y_SETDELAY and A_SETDELAY

The delay indicator is a relative indicator ranging from 0 to 1. If not transactions are queued the value is 0 if all transactions are queued the maximum time ie to the end of the day the value is 1. The value is calculated as the time weight queuing value for each queued transaction (transaction value times the time in queue) divided by the time weighted value if all payments were delayed to the end of the day (the transaction value times the time from submission to the end of the day). The values are calculated for each participant/account.

$$Settlement\,delay = \frac{\sum_{i=1}^{n}\sum_{k=1}^{d} q_{i,k} * a_{i,k}}{\sum_{i=1}^{n}\sum_{k=1}^{d} s_{i,k} * a_{i,k}} \text{ where}$$

q = queing time for each payment

s = maximum settlement delay ie time difference between submission and end-of -day.

The values of unsettled transactions are included in both factors.

Consumed liquidity Y_LIQUSAGC and A_LIQUSAGC

The consumed liquidity indicator measures to which extent overdrafts (ie negative balances) and reserve deposits have been used for settling payments ie the difference between the beginning of day and minimum balance during the day divided by volume of submitted transactions. It measures the consumed liquidity compared with the throughput volume or inversely to which extent the liquidity of received payments have not been able to cover the liquidity needs of outgoing payments.

$$\text{Consumed liquidity} = \frac{\sum_{t=0}^{T} L_t^d}{\sum_{t=0}^{T}\sum_{i=0}^{t} V_i^O} \text{ where}$$

L^d = the difference between daily opening and minimum balance

V^o = the average transaction volume

Rigid liquidity indicator Y_LIQUSAGR and A_LIQUSAGR

The rigid liquidity indicator gives the relation between the total available credit limit compared to the transaction volume to be settled ie the sum of transactions to be sent. It measures the credits allocated compared to the throughput volume.

$$Rigid\ liquidity\ indicator = \frac{\sum_{t=0}^{T} L_t^a}{\sum_{t=0}^{T}\sum_{i=0}^{t} V_t^O} \text{ where}$$

L^a = the average credit limit available during the day