

Database

Atoti FRTB

5.1

Table of Contents

Table of Contents	2
Database	5
Global Database Definition	5
FX Rates	5
Trade Booking	5
BOOK_DESK_MAPPING	5
Unique Key	6
Incoming Joins	6
BOOK_HIERARCHY	6
Unique Key	7
Incoming Joins	7
Populating From a ParentChild Table	8
DESK_DESCRIPTION	8
Unique Key	9
Incoming Joins	9
FXRATES	9
Unique Key	9
LEGAL_ENTITY_ATTRIBUTES	10
Unique Key	10
Incoming Joins	10
LEGAL_ENTITY_HIERARCHY	10
Unique Key	11
Incoming Joins	11
Populating From a ParentChild Table	11
TRADE_MAPPING	12
Unique Key	12
Incoming Joins	12
Outgoing Joins	12
Standardised Approach Database Definition	13
Trade Description	13
Risk Factor Descriptions	13

Sensitivities	13
Overrides	14
OBLIGOR_OVERRIDES	14
RISK_FACTOR_DESCRIPTION_OVERRIDES	16
RRAOOVERRIDES	18
TRANCHE_OVERRIDES	20
UNDERLYING_DESCRIPTION_OVERRIDES	22
CSRBUCKET_DESC	25
Unique Key	26
Incoming Joins	26
EQUITY_BUCKET_DESC	26
Unique Key	27
Incoming Joins	27
OBLIGOR	27
Unique Key	28
Incoming Joins	28
RISK_FACTOR_DESCRIPTION	28
Unique Key	30
Incoming Joins	30
Outgoing Joins	30
RRAO	31
Unique Key	31
Incoming Joins	31
SASENSITIVITIES	32
Unique Key	35
Outgoing Joins	36
SATRADE_DESCRIPTION	36
Unique Key	37
Incoming Joins	37
SECURITY	37
Unique Key	38
Incoming Joins	38
SENIORITY_DESCRIPTION	39
Unique Key	39
Incoming Joins	39

TRANCHE	39
Unique Key	40
Incoming Joins	41
UNDERLYING_DESCRIPTION	41
Unique Key	43
Incoming Joins	44
Outgoing Joins	44

Database

This section provides the database definitions in Atoti FRTB.

Here are a few points to note about the database descriptions:

- The documentation mentions some constraints, for example NOT NULL and UNIQUE KEY. These constraints may not be enforced by all databases and may be difficult to enforce when using views. However, Atoti FRTB will assume that the data satisfies these constraints and may behave unpredictably if they are not satisfied.
- The documentation includes the joins used between the tables/views. These are provided for informational purposes, though they may optionally be used to construct keys and indices to help maintain data integrity and improve performance.

Each cube in Atoti FRTB uses a star schema with many-to-one joins radiating out from a base table. The base tables are as follows:

Cube	Base Table
StandardisedApproachCube	SASENSITIVITIES

Additionally, there are “isolated” tables that are not part of the star schema but are still used in the cubes.

Global Database Definition

This section describes tables that are common to all cubes. This includes FX rates and trade booking.

FX Rates

The FX rates are stored in the `FX_RATES` table.

Trade Booking

The `TRADE_MAPPING` table maps trades/positions to books and legal entities by `TRADE_ID` and `AS_OF_DATE`.

The multi-level book organizational hierarchy and desk descriptions are in the `BOOK_HIERARCHY` table which is indexed by `BOOK` and `AS_OF_DATE`.

The multi-level legal entity organizational hierarchy is in the `LEGAL_ENTITY_HIERARCHY` table which is indexed by `LEGAL_ENTITY` and `AS_OF_DATE`. It is built from the table.

The `LEGAL_ENTITY_ATTRIBUTES` table provides a description of the legal entities.

BOOK_DESK_MAPPING

The BOOK_DESK_MAPPING table contains the mappings for books to desks

Column Name	Type	Not Null	Cube Field	Description
BOOK	STRING	Y		Leaf node of the book hierarchy.
DESK	STRING	Y	[Booking].[Desks]	The desk to which the book belongs.
AS_OF_DATE	DATE	Y	See field in joined table (SASENSITIVITIES)	Timestamp (at close of business) for the data.

Unique Key

Columns
AS_OF_DATE
BOOK

Incoming Joins

Source Table	Source Columns	Target Columns
TRADE_MAPPING	AS_OF_DATE BOOK	AS_OF_DATE BOOK

BOOK_HIERARCHY

The BOOK_HIERARCHY table contains the multi-level book organizational structure and the desk-level information.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y		Timestamp (at close of business) for the data.
BOOK	STRING	Y		Leaf node of the book hierarchy. This matches the last non-_DATAMEMBER_ node in levels 1 - 15.
HIERARCHY_LEVEL1	STRING		Level 1	Node at level 1 of the book hierarchy.
HIERARCHY_LEVEL2	STRING		Level 2	Node at level 2 of the book hierarchy.
HIERARCHY_LEVEL3	STRING		Level 3	Node at level 3 of the book hierarchy.

Column Name	Type	Not Null	Cube Field	Description
HIERARCHY_LEVEL4	STRING		Level 4	Node at level 4 of the book hierarchy.
HIERARCHY_LEVEL5	STRING		Level 5	Node at level 5 of the book hierarchy.
HIERARCHY_LEVEL6	STRING		Level 7	Node at level 6 of the book hierarchy.
HIERARCHY_LEVEL7	STRING		Level 6	Node at level 7 of the book hierarchy.
HIERARCHY_LEVEL8	STRING		Level 8	Node at level 8 of the book hierarchy.
HIERARCHY_LEVEL9	STRING		Level 9	Node at level 9 of the book hierarchy.
HIERARCHY_LEVEL10	STRING		Level 10	Node at level 10 of the book hierarchy.
HIERARCHY_LEVEL11	STRING		Level 11	Node at level 11 of the book hierarchy.
HIERARCHY_LEVEL12	STRING		Level 12	Node at level 12 of the book hierarchy.
HIERARCHY_LEVEL13	STRING		Level 13	Node at level 13 of the book hierarchy.
HIERARCHY_LEVEL14	STRING		Level 14	Node at level 14 of the book hierarchy.
HIERARCHY_LEVEL15	STRING		Level 15	Node at level 15 of the book hierarchy.

Unique Key

Columns

AS_OF_DATE

BOOK

Incoming Joins

Source Table	Source Columns	Target Columns
--------------	----------------	----------------

Source Table	Source Columns	Target Columns
TRADE_MAPPING	AS_OF_DATE BOOK	AS_OF_DATE BOOK

Populating From a ParentChild Table

Instead of creating and populating the BOOK_HIERARCHY table directly, you can create a BOOK_PARENT_CHILD table to store the parent-child relationships.

Column Name	Type	Not Null	Cube Field	Description
NAME	STRING	Y		Child in the parent-child relationship.
PARENT	STRING			Parent in the parent-child relationship. Null means the child is a root node.
DATE	DATE	Y		Timestamp (at close of business) for the data.

With this table and the SQL script included in the [online documentation](#), you can create a series of intermediate views to populate the multiple levels of the BOOK_HIERARCHY table.

TIP

For performance reasons, views may impact query and aggregation performance. Injecting the results of the final constructed view into a table can improve query performance. The tradeoff is that this final table will need to be manually kept up to date if the underlying parent-child tables are updated.

DESK_DESCRIPTION

The DESK_DESCRIPTION table contains the desk descriptions.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (BOOK_DESK_MAPPING)	Timestamp (at close of business) for the data.
DESK	STRING	Y	[Booking].[Desks]	The desk to which the book belongs. This will match one of the non-_DATAMEMBER_ nodes in levels 1 - 15.

Column Name	Type	Not Null	Cube Field	Description
FRTB_DESK_MODEL	STRING		[Booking].[FRTB Model]	Specifies whether the desk should be treated as IMA or SA for the capital charge calculations.
PLA_ZONE	STRING		[Booking].[PLA Zone]	Indicates which zone the desk falls into according to the PLA test metrics [MAR32.42].
IRT_DESK	STRING		[Booking].[IRT Desk]	Indicates whether the desk is an IRT desk. ACR is calculated separately for desks flagged as IRT.

Unique Key

Columns

AS_OF_DATE

DESK

Incoming Joins

Source Table	Source Columns	Target Columns
BOOK_DESK_MAPPING	AS_OF_DATE DESK	AS_OF_DATE DESK

FXRATES

The FXRATES table contains all the FX Rates. It is an isolated table and not part of any cube facts. FX Rates are looked up via the default implementation of IFXRates API.

Column Name	Type	Not Null	Description
AS_OF_DATE	DATE	Y	Timestamp (at close of business) for the data.
BASE_CCY	STRING	Y	The left side of the currency pair.
COUNTER_CCY	STRING	Y	The right side of the currency pair.
FX_RATE	DOUBLE	Y	Forex rate between the two currencies.

Unique Key

Columns

AS_OF_DATE

BASE_CCY

COUNTER_CCY

LEGAL_ENTITY_ATTRIBUTES

The LEGAL_ENTITY_ATTRIBUTES table contains a description of the legal entity.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (SASENSITIVITIES)	Timestamp (at close of business) for the data.
LEGAL_ENTITY	STRING	Y	See field in joined table (TRADE_MAPPING)	The legal entity.
NETTING_SET	STRING		Netting Set	The netting set the legal entity belongs to.

Unique Key**Columns**

AS_OF_DATE

LEGAL_ENTITY

Incoming Joins

Source Table	Source Columns	Target Columns
TRADE_MAPPING	AS_OF_DATE LEGAL_ENTITY	AS_OF_DATE LEGAL_ENTITY

LEGAL_ENTITY_HIERARCHY

The LEGAL_ENTITY_HIERARCHY table contains the multi-level legal entity organizational structure.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y		Timestamp (at close of business) for the data.

Column Name	Type	Not Null	Cube Field	Description
LEGAL_ENTITY	STRING	Y		Leaf node of the book hierarchy. This matches the last non- <code>_DATAMEMBER_</code> node in levels 1 - 15.
HIERARCHY_LEVEL1	STRING		Level 1	Node at level 1 of the book hierarchy.
HIERARCHY_LEVEL2	STRING		Level 2	Node at level 2 of the book hierarchy.
HIERARCHY_LEVEL3	STRING		Level 3	Node at level 3 of the book hierarchy.
HIERARCHY_LEVEL4	STRING		Level 4	Node at level 4 of the book hierarchy.
HIERARCHY_LEVEL5	STRING		Level 5	Node at level 5 of the book hierarchy.

Unique Key

Columns

AS_OF_DATE

LEGAL_ENTITY

Incoming Joins

Source Table	Source Columns	Target Columns
TRADE_MAPPING	AS_OF_DATE LEGAL_ENTITY	AS_OF_DATE LEGAL_ENTITY

Populating From a ParentChild Table

Instead of creating and populating the `LEGAL_ENTITY_HIERARCHY` table directly, you can create a `LEGAL_ENTITY_PARENT_CHILD` table to store the parent-child relationships.

Column Name	Type	Not Null	Cube Field	Description
NAME	STRING	Y		Child in the parent-child relationship.
PARENT	STRING			Parent in the parent-child relationship. Null means the child is a root node.
DATE	DATE	Y		Timestamp (at close of business) for the data.

With this table and the SQL script included in the [online documentation](#), you can create a series of intermediate views to populate the multiple levels of the LEGAL_ENTITY_HIERARCHY table.

💡 TIP

For performance reasons, views may impact query and aggregation performance. Injecting the results of the final constructed view into a table can improve query performance. The tradeoff is that this final table will need to be manually kept up to date if the underlying parent-child tables are updated.

TRADE_MAPPING

The TRADE_MAPPING store maps trades to books, desks and legal entities.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y		Timestamp (at close of business) for the data.
TRADE_KEY	STRING	Y		Unique Trade (or Position) ID.
BOOK	STRING	Y	[Booking]. [Books]	The book to map the trade to (must match the node in the Book Hierarchy).
LEGAL_ENTITY	STRING	Y	[Organization]. [Legal Entities]	Legal Entity to map the trade to (must match the node in the Legal Entity Hierarchy).
TRADE_DATE	DATE		[Dates]. [TradeDates]	The date on which the trade took place.

Unique Key

Columns

AS_OF_DATE

TRADE_KEY

Incoming Joins

Source Table	Source Columns	Target Columns	Cube
SASENSITIVITIES	AS_OF_DATE TRADE_KEY	AS_OF_DATE TRADE_KEY	StandardisedApproachCube

Outgoing Joins

Target Table	Source Columns	Target Columns
BOOK_DESK_MAPPING	AS_OF_DATE BOOK	AS_OF_DATE BOOK
BOOK_HIERARCHY	AS_OF_DATE BOOK	AS_OF_DATE BOOK
LEGAL_ENTITY_HIERARCHY	AS_OF_DATE LEGAL_ENTITY	AS_OF_DATE LEGAL_ENTITY
LEGAL_ENTITY_ATTRIBUTES	AS_OF_DATE LEGAL_ENTITY	AS_OF_DATE LEGAL_ENTITY

Standardised Approach Database Definition

The SA Cube Schema starts with the **SASENSITIVITIES** table, which is an index to all the facts in the SA Cube.

Trade Description

The **TRADE_MAPPING** table places each trade in the organizational hierarchy. See **Global** section for more details.

The **SA_TRADE_DESCRIPTION** table provides trade-level data.

Risk Factor Descriptions

The **SASENSITIVITIES** table references the risk-factor descriptions for all SA facts.

The risk-factor description starts with the **RISK_FACTOR_DESCRIPTION** table, which contains the description of risk-factor (independent of the underlying).

The **RISK_FACTOR_DESCRIPTION** contains references to the follow tables:

Risk Class	Underlying Store
GIRR, CSR non-Sec, CSR Sec non-CTP, CSR Sec CTP, Equity, Commodity, FX	UNDERLYING_DESCRIPTION
DRC non-Sec	OBLIGOR
DRC Sec non-CTP	TRANCHE
RRAO	RRAO

Sensitivities

The **SASENSITIVITIES** table holds all sensitivities.

Overrides

Overrides allow for changing static data for different Parameter Sets.

For example, CRR2 adds additional Buckets over BCBS, so we use overrides to move Risk Factors from one Bucket to another for the CRR2 Parameter Set.

- **OBLIGOR_OVERRIDES**
- **RISK_FACTOR_DESCRIPTION_OVERRIDES**
- **RRAO_OVERRIDES**
- **TRANCHE_OVERRIDES**
- **UNDERLYING_DESCRIPTION_OVERRIDES**

OBLIGOR_OVERRIDES

The OBLIGOR_OVERRIDES table contains the Override definitions for the Obligor table.

Column Name	Type	Not Null
OBLIGOR_ID	STRING	Y
RISK_CLASS	STRING	Y
PARAMETER_SET	STRING	Y
OBLIGOR_CATEGORY	STRING	
RATING	STRING	
RISK_WEIGHT	DOUBLE	
DRC_FUND_TREATMENT	STRING	
AS_OF_DATE	DATE	Y

Unique Key

Columns
OBLIGOR_ID
RISK_CLASS
PARAMETER_SET
AS_OF_DATE

Override Base Table

The base table for these overrides is the **Obligor** table. To define Overrides you must add facts to the base table. For details on why this is required, see [Overrides With DirectQuery](#).

Inject Base Table

For each override, you must generate multiple entries in the **Obligor** table with the following structure.

Override Parameter	Obligor Table Field
OVERRIDE KEY FIELD	OBLIGOR_ID
OVERRIDE DATE FIELD	AS_OF_DATE

Where:

- **Override Parameter:** The parameters to determine where to apply this override.
- **Obligor Table Field:** The field in the **Obligor** base table for this override.

Create Base Store Tuples

See the [Create Override Tuples](#) section for an example of how to create the override tuples for the following override fields. These are the fields we want to override in the base override table, in this case the **Obligor** Table.

Override Table
PARAMETER_SET
OBLIGOR_CATEGORY
RATING
RISK_WEIGHT
DRC_FUND_TREATMENT

These fields form an intermediate table containing the Override's base store fields and will be merged back into the Override base table: **Obligor**.

Map Override Fields to Base Table Fields

Once you create your tuples for each override field, you can then map the tuples back to the base store using the following relationship:

Override Table	Obligor Table Field	Note
OVERRIDE KEY FIELD	OBLIGOR_ID	
OVERRIDE DATE FIELD	AS_OF_DATE	
PARAMETER_SET		No mapping exists
OBLIGOR_CATEGORY	OBLIGOR_CATEGORY	

Override Table	Obligor Table Field	Note
RATING	RATING	
RISK_WEIGHT	RISK_WEIGHT	
DRC_FUND_TREATMENT	DRC_FUND_TREATMENT	
	RISK_CLASS	No mapping exists

RISK_FACTOR_DESCRIPTION_OVERRIDES

The RISK_FACTOR_DESCRIPTION_OVERRIDES table contains the Override definitions for the Risk Factor Descriptions.

Column Name	Type	Not Null
AS_OF_DATE	DATE	Y
RISK_FACTOR	STRING	Y
RISK_CLASS	STRING	Y
RISK_MEASURE	STRING	Y
PARAMETER_SET	STRING	Y
RISK_FACTOR_TYPE	STRING	
COMMODITY_LOCATION	STRING	
SENIORITY	STRING	
MATURITY	STRING	
ZERO_RISK_WEIGHT	STRING	

Unique Key

Columns
AS_OF_DATE
RISK_FACTOR
RISK_CLASS
RISK_MEASURE
PARAMETER_SET

Override Base Table

The base table for these overrides is the [Risk Factor Description](#) table. To define Overrides you must add facts to the base table. For details on why this is required, see [Overrides With DirectQuery](#).

Inject Base Table

For each override, you must generate multiple entries in the [Risk Factor Description](#) table with the following structure.

Override Parameter	Risk Factor Description Table Field
OVERRIDE KEY FIELD	RISK_FACTOR
OVERRIDE DATE FIELD	AS_OF_DATE

Where:

- **Override Parameter:** The parameters to determine where to apply this override.
- **Risk Factor Description Table Field:** The field in the [Risk Factor Description](#) base table for this override.

Create Base Store Tuples

See the [Create Override Tuples](#) section for an example of how to create the override tuples for the following override fields. These are the fields we want to override in the base override table, in this case the [Risk Factor Description](#) Table.

Override Table
PARAMETER_SET
RISK_FACTOR_TYPE
COMMODITY_LOCATION
SENIORITY
MATURITY
ZERO_RISK_WEIGHT

These fields form an intermediate table containing the Override's base store fields and will be merged back into the Override base table: [Risk Factor Description](#).

Map Override Fields to Base Table Fields

Once you create your tuples for each override field, you can then map the tuples back to the base store using the following relationship:

Override Table	Risk Factor Description Field	Note
----------------	-------------------------------	------

Override Table	Risk Factor Description Field	Note
OVERRIDE KEY FIELD	RISK_FACTOR	
OVERRIDE DATE FIELD	AS_OF_DATE	
PARAMETER_SET		No mapping exists
RISK_FACTOR_TYPE	RISK_FACTOR_TYPE	
COMMODITY_LOCATION	COMMODITY_LOCATION	
SENIORITY	SENIORITY	
MATURITY	MATURITY	
ZERO_RISK_WEIGHT	ZERO_RISK_WEIGHT	
	RISK_CLASS	No mapping exists
	RISK_MEASURE	No mapping exists
	UNDERLYING	No mapping exists
	UNDERLYING_FXRISK_CCY	No mapping exists
	UNDERLYING_MATURITY	No mapping exists

RRAOOVERRIDES

The RRAOOVERRIDES table contains the Override definitions for RRAO categories.

Column Name	Type	Not Null
RRAOCATEGORY	STRING	Y
RISK_CLASS	STRING	Y
PARAMETER_SET	STRING	Y
RESIDUAL_RISK	STRING	
EXOTIC_UNDERLYING	STRING	
OTHER_RESIDUAL_RISK_TYPE	STRING	
AS_OF_DATE	STRING	Y

Unique Key

Columns
RRAOCATEGORY

Columns

RISK_CLASS

PARAMETER_SET

AS_OF_DATE

Override Base Table

The base table for these overrides is the **RRAO** table. To define Overrides you must add facts to the base table. For details on why this is required, see [Overrides With DirectQuery](#).

Inject Base Table

For each override, you must generate multiple entries in the **RRAO** table with the following structure.

Override Parameter	RRAO Table Field
OVERRIDE KEY FIELD	RRAOCATEGORY
OVERRIDE DATE FIELD	AS_OF_DATE

Where:

- **Override Parameter:** The parameters to determine where to apply this override.
- **RRAO Table Field:** The field in the **RRAO** base table for this override.

Create Base Store Tuples

See the [Create Override Tuples](#) section for an example of how to create the override tuples for the following override fields. These are the fields we want to override in the base override table, in this case the **RRAO** Table.

Override Table

RESIDUAL_RISK

EXOTIC_UNDERLYING

OTHER_RESIDUAL_RISK_TYPE

These fields form an intermediate table containing the Override's base store fields and will be merged back into the Override base table: **RRAO**.

Map Override Fields to Base Table Fields

Once you create your tuples for each override field, you can then map the tuples back to the base store using the following relationship:

Override Table	RRAO Field	Note
OVERRIDE KEY FIELD	RRAOCATEGORY	
OVERRIDE DATE FIELD	AS_OF_DATE	
PARAMETER_SET		No mapping exists
RESIDUAL_RISK	RESIDUAL_RISK	
EXOTIC_UNDERLYING	EXOTIC_UNDERLYING	
OTHER_RESIDUAL_RISK_TYPE	OTHER_RESIDUAL_RISK_TYPE	
	RISK_CLASS	No mapping exists

TRANCHE_OVERRIDES

The TRANCHE_OVERRIDES table contains the Override definitions for the Tranches.

Column Name	Type	Not Null
TRANCHE	STRING	Y
RISK_CLASS	STRING	Y
PARAMETER_SET	STRING	Y
BUCKET	STRING	
SENIORITY	STRING	
RATING	STRING	
TYPE	STRING	
REGION	STRING	
ASSET_CLASS	STRING	
ATTACHMENT	DOUBLE	
DETACHMENT	DOUBLE	
RISK_WEIGHT	DOUBLE	
AS_OF_DATE	DATE	Y

Unique Key

Columns
TRANCHE

Columns

RISK_CLASS

PARAMETER_SET

AS_OF_DATE

Override Base Table

The base table for these overrides is the [Tranche](#) table. To define Overrides you must add facts to the base table. For details on why this is required, see [Overrides With DirectQuery](#).

Inject Base Table

For each override, you must generate multiple entries in the [Tranche](#) table with the following structure.

Override Parameter	Tranche Table Field
OVERRIDE KEY FIELD	TRANCHE
OVERRIDE DATE FIELD	AS_OF_DATE

Where:

- **Override Parameter:** The parameters to determine where to apply this override.
- **Tranche Table Field:** The field in the [Tranche](#) base table for this override.

Create Base Store Tuples

See the [Create Override Tuples](#) section for an example of how to create the override tuples for the following override fields. These are the fields we want to override in the base override table, in this case the [Tranche](#) Table.

Override Table

PARAMETER_SET

BUCKET

CSRQUALITY

CSRSECTOR

CSRRATING

EQUITY_MARKET_CAP

EQUITY_ECONOMY

EQUITY_SECTOR

POOL

Override Table

ATTACHMENT

DETACHMENT

These fields form an intermediate table containing the Override's base store fields and will be merged back into the Override base table: [Tranche](#).

Map Override Fields to Base Table Fields

Once you create your tuples for each override field, you can then map the tuples back to the base store using the following relationship:

Override Table	Tranche Field	Note
OVERRIDE KEY FIELD	TRANCHE	
OVERRIDE DATE FIELD	AS_OF_DATE	
PARAMETER_SET		No mapping exists
BUCKET	BUCKET	
SENIORITY	SENIORITY	
RATING	RATING	
TYPE	TYPE	
REGION	REGION	
ASSET_CLASS	ASSET_CLASS	
ATTACHMENT	ATTACHMENT	
DETACHMENT	DETACHMENT	
RISK_WEIGHT	RISK_WEIGHT	
	RISK_CLASS	No mapping exists

UNDERLYING_DESCRIPTION_OVERRIDES

The UNDERLYING_DESCRIPTION_OVERRIDES table contains the Override definitions for the Underlying Descriptions.

Column Name	Type	Not Null
UNDERLYING	STRING	Y
RISK_CLASS	STRING	Y

Column Name	Type	Not Null
PARAMETER_SET	STRING	Y
BUCKET	STRING	
CSRQUALITY	STRING	
CSRSECTOR	STRING	
CSRRATING	STRING	
EQUITY_MARKET_CAP	STRING	
EQUITY_ECONOMY	STRING	
EQUITY_SECTOR	STRING	
POOL	STRING	
ATTACHMENT	DOUBLE	
DETACHMENT	DOUBLE	
AS_OF_DATE	DATE	Y

Unique Key

Columns
UNDERLYING
RISK_CLASS
PARAMETER_SET
AS_OF_DATE

Override Base Table

The base table for these overrides is the [Underlying Description](#) table. To define Overrides you must add facts to the base table. For details on why this is required, see [Overrides With DirectQuery](#).

Inject Base Table

For each override, you must generate multiple entries in the [Underlying Description](#) table with the following structure.

Override Parameter	Underlying Description Table Field
OVERRIDE KEY FIELD	UNDERLYING

Override Parameter	Underlying Description Table Field
OVERRIDE DATE FIELD	AS_OF_DATE

Where:

- **Override Parameter:** The parameters to determine where to apply this override.
- **Underlying Description Table Field:** The field in the [Underlying Description](#) base table for this override.

Create Base Store Tuples

See the [Create Override Tuples](#) section for an example of how to create the override tuples for the following override fields. These are the fields we want to override in the base override table, in this case the [Underlying Description](#) Table.

Override Table
PARAMETER_SET
BUCKET
CSRQUALITY
CSRSECTOR
CSRRATING
EQUITY_MARKET_CAP
EQUITY_ECONOMY
EQUITY_SECTOR
POOL
ATTACHMENT
DETACHMENT

These fields form an intermediate table containing the Override's base store fields and will be merged back into the Override base table: [Underlying Description](#).

Map Override Fields to Base Table Fields

Once you create your tuples for each override field, you can then map the tuples back to the base store using the following relationship:

Override Table	Underlying Description Table Field	Note
OVERRIDE KEY FIELD	UNDERLYING	

Override Table	Underlying Description Table Field	Note
OVERRIDE DATE FIELD	AS_OF_DATE	
PARAMETER_SET		No mapping exists
BUCKET	BUCKET	
CSRQUALITY	CSRQUALITY	
CSRSECTOR	CSRSECTOR	
CSRRATING	CSRRATING	
EQUITY_MARKET_CAP	EQUITY_MARKET_CAP	
EQUITY_ECONOMY	EQUITY_ECONOMY	
EQUITY_SECTOR	EQUITY_SECTOR	
POOL	POOL	
ATTACHMENT	ATTACHMENT	
DETACHMENT	DETACHMENT	
	RISK_CLASS	No mapping exists
	GIRR_CURVE_TYPE	No mapping exists
	GIRR_CCY	No mapping exists
	UNDERLYING_FXORIGINAL_CCY	No mapping exists

CSRBUCKET_DESC

i NOTE

While this table is currently required in the DirectQuery database schema, it will be removed in a future version.

The CSRBUCKET_DESC table provides canonical descriptions for the CSR buckets.

Column Name	Type	Not Null	Description
AS_OF_DATE	DATE	Y	Timestamp (at close of business) for the data.
RISK_CLASS	STRING	Y	The risk-class (“CSR non-Sec”, “CSR Sec non-CTP”, “CSR Sec CTP”)

Column Name	Type	Not Null	Description
BUCKET	STRING	Y	The Bucket the Underlying belongs to.
RATING_CATEGORY	STRING		The canonical name for the bucket rating.
SECTOR_CATEGORY	STRING		The canonical name for the bucket sector.

Unique Key

Columns

AS_OF_DATE

RISK_CLASS

BUCKET

Incoming Joins

Source Table	Source Columns	Target Columns
UNDERLYING_DESCRIPTION	AS_OF_DATE RISK_CLASS BUCKET	AS_OF_DATE RISK_CLASS BUCKET

EQUITY_BUCKET_DESC

i NOTE

While this table is currently required in the DirectQuery database schema, it will be removed in a future version.

The EQUITY_BUCKET_DESC table provides canonical descriptions for the equity buckets.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (UNDERLYING_DESCRIPTION)	Timestamp (at close of business) for the data.
RISK_CLASS	STRING	Y	See field in joined table (UNDERLYING_DESCRIPTION)	The risk-class (set to "Equity")

Column Name	Type	Not Null	Cube Field	Description
BUCKET	STRING	Y	See field in joined table (UNDERLYING_DESCRIPTION)	The Bucket the Underlying belongs to.
MARKET_CAP_CATEGORY	STRING		Equity Market Cap Category	The canonical name for the bucket market cap.
ECONOMY_CATEGORY	STRING		Equity Economy Category	The canonical name for the bucket economy.
SECTOR_CATEGORY	STRING		Equity Sector Category	The canonical name for the bucket sector.

Unique Key

Columns

AS_OF_DATE

RISK_CLASS

BUCKET

Incoming Joins

Source Table	Source Columns	Target Columns
UNDERLYING_DESCRIPTION	AS_OF_DATE RISK_CLASS BUCKET	AS_OF_DATE RISK_CLASS BUCKET

OBLIGOR

The OBLIGOR table contains the description of a DRC non-Sec obligor.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (SASENSITIVITIES)	Timestamp (at close of business) for the data.

Column Name	Type	Not Null	Cube Field	Description
OBLIGOR_ID	STRING	Y	[Default Risk Charge].[DRC Obligor]	The ID of the obligor.
RISK_CLASS	STRING	Y		Set to "DRC non-Sec".
OBLIGOR_CATEGORY	STRING	Y	[Default Risk Charge].[DRC non-Sec Bucket]	the bucket to which the obligor belongs.
RATING	STRING	Y	[Default Risk Charge].[DRC non-Sec Rating]	The credit quality of the obligor.
RISK_WEIGHT	DOUBLE		DRC non-Sec JTD Weightings Override measure	Optional override for the DRC non-Sec Obligor risk-weight.
DRC_FUND_TREATMENT	STRING		[Default Risk Charge].[DRC Fund Treatment]	Flag indicating if the obligor cannot be included in offsetting or diversification with other exposures.

Unique Key

Columns

AS_OF_DATE

OBLIGOR

RISK_CLASS

Incoming Joins

Source Table	Source Columns	Target Columns
RISK_FACTOR_DESCRIPTION	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE OBLIGOR RISK_CLASS

RISK_FACTOR_DESCRIPTION

The RISK_FACTOR_DESCRIPTION table contains the description of risk-factor, independent of the underlying.

The fields used in this table and the purpose depend on the risk-class and risk-measure. See the [Implementation and Interpretation Guide](#) for details on each risk-class.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (SASENSITIVITIES)	Timestamp (at close of business) for the data.
RISK_FACTOR	STRING	Y	See field in joined table (SASENSITIVITIES)	The name of the risk factor.
RISK_CLASS	STRING	Y	See field in joined table (SASENSITIVITIES)	The risk-class (“GIRR”, “CSR non-Sec”, “CSR Sec non-CTP”, “CSR Sec CTP”, “Equity”, “Commodity”, “FX”, “DRC non-Sec”, “DRC Sec non-CTP”, “RRAO”)
RISK_MEASURE	STRING	Y	See field in joined table (SASENSITIVITIES)	The risk-measure (“Delta”, “Vega”, “Curvature”, “DRC”, “RRAO”)
UNDERLYING	STRING	Y	[Market Data]. [Underlying]	The primary component of the risk factor. See datastore references below.
RISK_FACTOR_TYPE	STRING		[Risk].[Risk Factor Types]	The type of the risk-factor CSR Delta: “Bond” or “CDS” Equity Delta: “Spot” or “Repo”
COMMODITY_LOCATION	STRING		[Risk]. [Commodity Location]	Commodity only. Commodity delivery location
UNDERLYING_FXRISK_CCY	STRING		[Risk].[FX Counter Currency]	FX only. The counter currency of the risk-factor currency pair.
SENIORITY	STRING		[Default Risk Charge].[DRC Seniority]	Seniority of the exposure
MATURITY	STRING		[Risk].[Original Maturity]	The tenor or maturity (e.g. “1D”, “2W”, “12M”, “1Y”, or date “YYYY-MM-DD”).
UNDERLYING_MATURITY	STRING		[Risk].[Original Underlying Maturity]	GIRR Vega only. Underlying residual maturity.

Column Name	Type	Not Null	Cube Field	Description
ZERO_RISK_WEIGHT	STRING		[Default Risk Charge].[DRC Zero Risk Weight]	Flag, 'Y' or 'N', indicating if the exposure qualifies for a zero risk-weight (default = N).

Unique Key

Columns
AS_OF_DATE
RISK_FACTOR
RISK_CLASS
RISK_MEASURE

Incoming Joins

Source Table	Source Columns	Target Columns
SASENSITIVITIES	RISK_FACTOR RISK_CLASS RISK_MEASURE AS_OF_DATE	RISK_FACTOR RISK_CLASS RISK_MEASURE AS_OF_DATE

Outgoing Joins

Target Table	Source Columns	Target Columns	Risk Class
UNDERLYING_DESCRIPTION	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE UNDERLYING RISK_CLASS	"GIRR", "CSR non-Sec", "CSR Sec non-CTP", "CSR Sec CTP", "Equity", "Commodity", "FX"
OBLIGOR	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE OBLIGOR RISK_CLASS	"DRC non-Sec"
TRANCHE	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE TRANCHE RISK_CLASS	"DRC Sec non-CTP"
SECURITY	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE SECURITY RISK_CLASS	"DRC Sec CTP"

Target Table	Source Columns	Target Columns	Risk Class
RRAO	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE RRAOCATEGORY RISK_CLASS	“RRAO”
SENIORITY_DESCRIPTION	AS_OF_DATE SENIORITY	AS_OF_DATE SENIORITY	“DRC non-Sec”

RRAO

The RRAO table contains the description of the RRAO category.

The RRAO category is not part of the specification, however, it is used to group trades whose RRAO may change between jurisdictions.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in (SASENSITIVITIES) table	Timestamp (at close of business) for the data.
RRAOCATEGORY	STRING	Y	RRAO Category	The ID of the RRAO Category.
RISK_CLASS	STRING	Y		Set to “RRAO”.
RESIDUAL_RISK	STRING		RRAO	Flag ‘Y’ or ‘N’ indicating if this RRAO category is subject to residual risk add-on.
EXOTIC_UNDERLYING	STRING		Exotic Underlying	Flag ‘Y’ or ‘N’ indicating an exotic underlying for the RRAO category.
OTHER_RESIDUAL_RISK_TYPE	STRING		Other Residual Risk Type	Optional data indicating the residual Risk type.

Unique Key

Columns

AS_OF_DATE

RRAOCATEGORY

RISK_CLASS

Incoming Joins

Source Table	Source Columns	Target Columns
RISK_FACTOR_DESCRIPTION	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE RRAOCATEGORY RISK_CLASS

SASENSITIVITIES

The SASENSITIVITIES table is the base in the SA Cube star schema and holds all the sensitivities. Each row in this table represents a fact in the SA Cube.

Column Name	Type	Not Null	Cube Field	Risk Measure	Description
AS_OF_DATE	DATE	Y	[Dates]. [AsOfDate]		Timestamp (at close of business) for the data.
TRADE_KEY	STRING	Y	This field is for internal usage only		Contains the tradeID for full data or Book#LegalEntity for summary data
UNDERLYING	STRING	Y	[Market Data]. [Underlying]		The primary component of the risk factor. See datastore references below.
TRADE_ID	STRING	Y	[Booking]. [TradeId]		Unique Trade (or Position) ID
RISK_FACTOR	STRING	Y	[Risk]. [RiskFactor]		Risk-factor identifier (unique per risk-class and risk-measure).
RISK_CLASS	STRING	Y	[Risk]. [RiskClass]		“Commodity”, “CSR non-Sec”, “CSR Sec non-CTP”, “CSR Sec CTP”, “Equity”, “FX”, “GIRR”, “DRC non-Sec”, “DRC Sec non-CTP”, “RAAO”
RISK_MEASURE	STRING	Y	[Risk]. [Measure]		“Delta”, “Vega”, “Curvature”, “RAAO”, “DRC”

Column Name	Type	Not Null	Cube Field	Risk Measure	Description
CCY	STRING	Y	Currency		Currency used in the Sensitivity, ShiftUpPV, ShiftDownPV, PresentValue, Notional, GrossJTD, and Adjustment fields.
SENSITIVITY	DOUBLE		This is a measure	Delta and Vega	The sensitivity.
PRESENT_VALUE	DOUBLE		This is a measure	Curvature and DRC	The unshifted PV for Curvature, or the bond-equivalent market value for DRC.
NOTIONAL	DOUBLE		This is a measure	DRC	The bond-equivalent notional for DRC.
SHIFT_UP_PV	DOUBLE		This is a measure	Curvature	PV resulting from parallel shocks up.
SHIFT_DOWN_PV	DOUBLE		This is a measure	Curvature	PV resulting from parallel shocks down.
GROSS_JTD	DOUBLE		This is a measure	DRC	(optional) Gross JTD value (alternative to calculating it from the market value and notional).
ADJUSTMENT	DOUBLE		This is a measure	DRC	The adjustment added to the Gross JTD (when sa.drc.adjustment.apply=true)

Column Name	Type	Not Null	Cube Field	Risk Measure	Description
FXCOMPLEX_TRADE	STRING			Delta	FX Only. Boolean 'Y' or 'N' to indicate if the sensitivity can be converted from one reporting currency to another.
FXOTHER_CCY	STRING			Delta	FX Only.
FX_DIVIDER_ELIGIBILITY	STRING			Curvature	FX Only. Boolean 'Y' or 'N' to indicate if the CVR qualifies for dividing by 1.5.
OPTIONALITY	STRING		Delta Optionality	Delta	Indicates whether the instrument has optionality (See BCBS 457 [MAR21.2]). It is set to 'Y' for instruments with optionality (and hence with Vega and Curvature risk); set to 'N' for trades without optionality (with no Vega and Curvature risk).
RISK_WEIGHT	DOUBLE			Curvature	Optional field to allow clients to send the risk weight to apply for curvature. If the field is null, the default value (most severe Delta weight) should be applied.
PV_APPLIED	STRING			Curvature	Boolean 'Y' or 'N' to indicate if PV has been removed from sensitivities or not.

Column Name	Type	Not Null	Cube Field	Risk Measure	Description
PV_LADDER	STRING		Present Value Ladder	Curvature	The cube leaf level (along with the RiskFactor and AsOfDate) to use when interpolating shocked PV ladders.
INSTRUMENT_LGD_TYPE	STRING		[Default Risk Charge]. [DRC Instrument LGD Type]	DRC	Instrument type for LGD (BCBS 457, [MAR22.12]) <ul style="list-style-type: none"> • equity • junior debt • senior debt • covered bond
DIRECTION	STRING		[Default Risk Charge]. [DRC Direction]	DRC	'long' or 'short'.
GROSS_JTD_OVERRIDDEN	STRING			DRC	
FXORIGINAL_DIVIDER_ELIGIBILITY	STRING			Delta	FX Only. Boolean 'Y' or 'N' to indicate if the CVR qualifies for dividing by 1.5.
ORIGINAL_OPTIONALITY	STRING			Delta	Set to same value as OPTIONALITY

Unique Key

Columns
AS_OF_DATE
TRADE_ID
TRADE_KEY
UNDERLYING
RISK_FACTOR

Columns

RISK_CLASS

RISK_MEASURE

Outgoing Joins

Target Table	Source Columns	Target Columns
SA_TRADE_DESCRIPTION	AS_OF_DATE TRADE_ID	AS_OF_DATE TRADE_ID
RISK_FACTOR_DESCRIPTION	AS_OF_DATE RISK_FACTOR RISK_CLASS RISK_MEASURE	AS_OF_DATE RISK_FACTOR RISK_CLASS RISK_MEASURE

SATRADE_DESCRIPTION

The SATRADE_DESCRIPTION table contains trade-level data used in the SA calculations.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (SASENSITIVITIES)	Timestamp (at close of business) for the data.
TRADE_ID	STRING	Y		Database key for trade/position.
SENSITIVITY_SCALE_CATEGORY	STRING		Sensitivity Scale Category	The category used to scale the SBM sensitivities. This matches the category in the SensitivityScaling configuration table.
NOTIONAL	DOUBLE		This field is a measure	The Notional of the trade/position (used for RRAO and DRC). <i>Deprecated for DRC use</i>
NOTIONAL_CCY	STRING			Currency code of the Notional. Required if Notional provided.

Column Name	Type	Not Null	Cube Field	Description
PRESENT_VALUE	DOUBLE		This field is a measure	The current present value of the trade/position (used in curvature and DRC). <i>Deprecated</i>
PVCCY	STRING			Currency code of present value. Required if present value provided.

Unique Key

Columns
AS_OF_DATE
TRADE_ID

Incoming Joins

Source Table	Source Columns	Target Columns
SASENSITIVITIES	AS_OF_DATE TRADE_ID	AS_OF_DATE TRADE_ID

SECURITY

The SECURITY table

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y		See field in joined table (SASENSITIVITIES) Timestamp (at close of business) for the data.
SECURITY	STRING	Y	[Default Risk Charge].[DRC Sec CTP Security]	The underlying product of the trade that could be an obligor or a fully qualified tranche (with the series, attachment and detachment).
RISK_CLASS	STRING	Y		Always "DRC Sec CTP"
BUCKET	STRING	Y	[Default Risk Charge].[DRC Sec CTP Bucket]	Obligor or Index the underlying relies on (see BCBS 457, MRA22.40)

Column Name	Type	Not Null	Cube Field	Description
SENIORITY	STRING		[Default Risk Charge].[DRC Sec CTP Seniority]	“Senior” or “Junior”.
RATING	STRING		[Default Risk Charge].[DRC Sec CTP Rating]	From AAA to Default.
TYPE	STRING		[Default Risk Charge].[DRC Sec CTP Rating Type]	Rating type : STC or empty for non-STC.
ATTACHMENT	DOUBLE		[Default Risk Charge].[DRC Sec CTP Attachment]	The start of the tranche or empty if non-tranched product.
DETACHMENT	DOUBLE		[Default Risk Charge].[DRC Sec CTP Detachment]	The end of the tranche or empty if non-tranched product.
INSTRUMENT_TYPE	STRING		[Default Risk Charge].[DRC Sec CTP Instrument Type]	Tranche or non-tranched, depending on the Attachment and Detachment fields.
RISK_WEIGHT	DOUBLE			Optional risk-weight, to override value.

Unique Key

Columns

AS_OF_DATE

SECURITY

RISK_CLASS

Incoming Joins

Source Table	Source Columns	Target Columns
--------------	----------------	----------------

Source Table	Source Columns	Target Columns
RISK_FACTOR_DESCRIPTION	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE SECURITY RISK_CLASS

SENIORITY_DESCRIPTION

The SENIORITY_DESCRIPTION table provides a ranking of seniorities that can be used when calculating the DRC non-Sec net JTD.

Column Name	Type	Not Null	Description
AS_OF_DATE	DATE	Y	Timestamp (at close of business) for the data.
SENIORITY	STRING	Y	Seniority of the exposure.
RANKING	INTEGER	Y	Ranking to determine relative seniorities of SENIORITY values.

Unique Key

Columns
AS_OF_DATE
SENIORITY

Incoming Joins

Source Table	Source Columns	Target Columns
RISK_FACTOR_DESCRIPTION	AS_OF_DATE SENIORITY	AS_OF_DATE SENIORITY

TRANCHE

The TRANCHE table contains the description of a DRC Sec non-CTP tranche.

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table	Timestamp (at close of business) for the data.

Column Name	Type	Not Null	Cube Field	Description
TRANCHE	STRING	Y	[Default Risk Charge]. [DRC Sec non-CTP Tranche]	ID of the tranche.
RISK_CLASS	STRING	Y		Set to "DRC Sec non-CTP".
BUCKET	STRING	Y	[Default Risk Charge]. [DRC Sec non-CTP Bucket]	The DRC bucket.
SENIORITY	STRING	Y	[Default Risk Charge]. [DRC Sec non-CTP Seniority]	Seniority of the tranche.
RATING	STRING		[Default Risk Charge]. [DRC Sec non-CTP Rating]	The credit rating of the tranche.
TYPE	STRING		[Default Risk Charge]. [DRC Sec non-CTP Rating Type]	The rating type used alongside the rating to determine the SEC-ERBA risk-weight.
REGION	STRING		[Default Risk Charge]. [DRC Sec non-CTP Region]	The region used to determine the Bucket.
ASSET_CLASS	STRING		[Default Risk Charge]. [DRC Sec non-CTP Asset Class]	The asset class used to determine the Bucket.
ATTACHMENT	DOUBLE		[Default Risk Charge]. [DRC Sec non-CTP Attachment]	Attachment point (Decimal values are expected).
DETACHMENT	DOUBLE		[Default Risk Charge]. [DRC Sec non-CTP Detachment]	Detachment point (Decimal values are expected).
RISK_WEIGHT	DOUBLE		DRC Sec non-CTP JTD Weightings Override measure	Optional override for the DRC Sec non-CTP Tranche risk-weight.

Unique Key

Columns

AS_OF_DATE

TRANCHE

Columns

RISK_CLASS

Incoming Joins

Source Table	Source Columns	Target Columns
RISK_FACTOR_DESCRIPTION	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE TRANCHE RISK_CLASS

UNDERLYING_DESCRIPTION

The UNDERLYING_DESCRIPTION table contains the description of the principal component of the SBM risk-factors.

Each row in the table describes one of the following depending on the risk class:

Risk Class	Underlying
(link to risk-class specific documentation)	(link to risk-class specific underlying)
GIRR	yield, inflation, or cross-currency basis curve
FX	FX rate
Equity	equity or equity issuer
CSR non-Sec	relevant issuer credit spread curve
CSR Sec non-CTP	tranche credit spread curves
CSR Sec CTP	underlying credit spread curves
Commodity	distinct commodity

Column Name	Type	Not Null	Cube Field	Description
AS_OF_DATE	DATE	Y	See field in joined table (SASENSITIVITIES)	Timestamp (at close of business) for the data.
UNDERLYING	STRING	Y	See field in joined table (RISK_FACTOR_DESCRIPTION)	The primary component of the SBM risk factor.

Column Name	Type	Not Null	Cube Field	Description
RISK_CLASS	STRING	Y	See field in joined table (SASENSITIVITIES)	The risk-class ("GIRR", "CSR non-Sec", "CSR Sec non-CTP", "CSR Sec CTP", "Equity", "Commodity", "FX").
BUCKET	STRING			The Bucket the Underlying belongs to.
GIRR_CURVE_TYPE	STRING		[Market Data].[GIRR Curve Types]	GIRR Delta and Vega only. The Curve type ("Yield", "Basis", or "Inflation").
GIRR_CCY	STRING		[Risk].[Currencies]	GIRR only. The currency of the curve. This is also the Bucket.
CSRQUALITY	STRING		[Market Data].[CSR Quality]	CSR only. The credit quality of the curve ("Senior IG", "IG", "HY", or "NR").
CSRSECTOR	STRING		[Market Data].[CSR Sector]	CSR only. The relevant sector of the curve.
CSRRATING	STRING		[Market Data].[CSR Rating]	CSR non-Sec only. "high" for AA- and above covered bonds.

Column Name	Type	Not Null	Cube Field	Description
EQUITY_MARKET_CAP	STRING		[Market Data].[Equity Market Cap]	Equity only. The equity issuer market cap (“Large”, “Small”, “Other”).
EQUITY_ECONOMY	STRING		[Market Data].[Equity Issuer Economy]	Equity only. The equity issuer economy (“Emerging”, “Advanced”, “Other”).
EQUITY_SECTOR	STRING		[Market Data].[Equity Sector]	Equity only. The equity issuer sector.
POOL	STRING		[Market Data].[CSR Sec non-CTP Pool]	CSR Sec non-CTP only. The underlying pool for the tranche.
ATTACHMENT	DOUBLE		[Market Data].[CSR Sec non-CTP Attachment]	CSR Sec non-CTP only. Attachment point for the tranche.
DETACHMENT	DOUBLE		[Market Data].[CSR Sec non-CTP Detachment]	CSR Sec non-CTP only. Detachment point for the tranche.
UNDERLYING_FXORIGINAL_CCY	STRING			FX only. Set to the same as UNDERLYING.

Unique Key

Columns

AS_OF_DATE

UNDERLYING

Columns

RISK_CLASS

Incoming Joins

Source Table	Source Columns	Target Columns
RISK_FACTOR_DESCRIPTION	AS_OF_DATE UNDERLYING RISK_CLASS	AS_OF_DATE UNDERLYING RISK_CLASS

Outgoing Joins

Target Table	Source Columns	Target Columns	Risk Class
CSRBUCKET_DESC	AS_OF_DATE RISK_CLASS BUCKET	AS_OF_DATE RISK_CLASS BUCKET	“CSR non-Sec”, “CSR Sec non-CTP”, “CSR Sec CTP”
EQUITY_BUCKET_DESC	AS_OF_DATE RISK_CLASS BUCKET	AS_OF_DATE RISK_CLASS BUCKET	“Equity”