LANDSBYGGEFONDEN – DCAB STATISTICS

DD130 - DETAILED DESIGN

Version:	1.0
Status:	Approved
Approver:	
Author:	[Author]

netcompany

Document history

Version	Date	Author	Status	Comments
0.1	01-07-2022		Draft	Document creation and section selection
1.0	22-09-2022		Final	Cleanup and rewrite outstanding sections

References

Reference	Title	Author	Version

Table of contents

1 Introduction

This document describes the different layers and methods in the DCAB Statistics project. The descriptions include general descriptions of the layers but also concrete data transformations of each Dimension and Fact table in the project.

2 Extract

The SSIS packages select all columns available at the time of the making of the SSIS packages and hence does not use a star(*) select. This ensures that no errors occur when new columns are added.

The extract tables are used as a landing area for the source data. No transformations are made between the source system and the extract layer. This ensures our data loads will only minimally stress the source systems, and it gives a precise representation of the source data. Each table will have the following meta-tagging columns:

Name	Data Type	Description
Meta_Id	BIGINT	An internal unique identifier. The values are defined by an identity property.
Meta_CreateTime	DATETIME	Timestamp showing the time the row was inserted. Default is set to GETDATE().
Meta_CreateJob	BIGINT	Id of the execution that created the row.
Meta_ExternalJob	NVARCHAR(255)	If the table is loaded by an external job outside of the data warehouse, the name of the is written here. E.g. “Boomi”, “Hangfire”.

The primary key is based on the meta-tagging column [Meta_Id], since all columns accept null values and therefore cannot be part of a primary key.

```
CONSTRAINT [PK_<Source>_<Entity>] PRIMARY KEY CLUSTERED ([Meta_Id] ASC)
```

In addition an index is created on the business keys from the source and the [Meta_CreateTime] column. This index is used to find the business key from the source, which is used to merge [Extract] and [Archive]. As an example:

```
CREATE INDEX [IDX_ <Source>_<Entity>] ON [Extract].[<Source>_<Entity>] ([BusinessKey1] ASC, ..., [Meta_CreateTime] ASC)
```

3 Archive

Every table is archived with Type 2 history except Stamdata that is passed from extract to staging using fasttrack.

The Archive Layer is used to build a full history of the source data. The data is a replica of the data in the extract layer but stored with type 2 history i.e. when a row is updated, the current row is closed and a new row is inserted with the updated values. The meta-tagging therefore contains the following columns:

Name	Data Type	Description
Meta_Id	BIGINT	An internal unique identifier. The values are defined by an identity property.
Meta_VersionNo	INT	Sequence number
Meta_ValidFrom	DATETIME	Timestamp when the row arrived to the DWH, copied from Meta_CreateTime in the extract table
Meta_ValidTo	DATETIME	Timestamp when the row is closed
Meta_IsValid	BIT	Boolean indicating if all validation rules have succeeded
Meta_IsCurrent	BIT	Boolean indicating if this is the newest row
Meta_IsDeleted	BIT	Boolean indication if the row is deleted in the source
Meta_CreateTime	DATETIME	Timestamp when the row is inserted
Meta_CreateJob	BIGINT	Id of the execution that created the row.
Meta_UpdateTime	DATETIME	Timestamp when the row is updated
Meta_UpdateJob	BIGINT	Execution id linked to the execution log table in the audit framework
Meta_DeleteTime	DATETIME	Timestamp when the row is deleted
Meta_DeleteJob	BIGINT	Execution id linked to the execution log table in the audit framework
Meta_HashValue	VARBINARY(32)	Hashed value of the row values. Used for updating the archive

The primary key is defined as a composite key containing the business keys from the source and the [Meta_ValidFrom] column. As an example:

```
CONSTRAINT [PK_<Source>_<Entity>] PRIMARY KEY CLUSTERED ([<BusinessKey1>] ASC, ..., [Meta_ValidFrom] ASC)
```

In addition, a unique index is created on the [Meta_Id] column. As an example:

```
CREATE UNIQUE INDEX [IDX_<Source>_<Entity>] ON [Archive].[<Source>_<Entity>] ([Meta_Id] ASC)
```

No foreign keys are created between the tables.

3.1 UpdateArchive

UpdateArchive inserts data into the archive table using type 2 history. The procedure uses the primary key on the archive table to identify if the record in the extract table can be found in the archive. If the record is not found in the archive, then the record is inserted. If the record is found and a difference is identified in one or more of the columns, excluding the meta columns, then a new record is inserted and the meta columns of the newest and second newest records are adjusted accordingly.

The UpdateArchive procedure, can register whether a difference has been made in one of the non-primary key and non-meta columns in one of two ways. It can use a CASE WHEN statement to compare the columns in the extract and the archive. This is the default method. In a table with many columns and many rows, this method may not perform optimally. For this reason differences can also be identified using a hash value. By using a hash value, the non-primary and non-meta columns are hashed using SHA-256 and then compared with the hashed value of the record in the archive. The hashed values in the archive are located in the Meta_HashValue column.

Records that have been deleted in the source can also be marked as deleted in the archive. The UpdateArchive procedure marks deletes by first checking the Audit.DataTransferLog table to see if the corresponding extract table has been fully loaded and the last run was a success. If this is the case then records that are found in the archive but cannot be found in the extract table automatically have their Meta_IsDeleted value set to 1.

As the first and the last step, UpdateArchive executes the Audit.DataTransferStart and Audit.DataTransferEnd procedures to log the data transfers. UpdateArchive is generalized so that it can be used for all archive tables, however it only updates one table per execution. The procedure uses the following parameters:

Parameter Name	Data Type	Default Value	Output Value	Comments
TableName	NVARCHAR(255)		No	The name of the archive table
ExecutionId	BIGINT	-1	No	The id of the execution. This value is used by DataTransferStart and DataTransferEnd to log the process in the DataTransferLog.
ExtractDeleteAction	NVARCHAR(10)	Delete	No	If the value is set to “Delete” or “Truncate” the Extract tables are Truncated after each successful execution of the procedure.
PrintOnly	BIT	0	No	If the parameter is set to 1 then the procedure will not execute. Instead the query, the procedure would have executed, is printed. This is used for debugging purposes.

4 Baseline

The baseline layer provides the basis for the reporting layers, i.e. DMSA and cubes.

4.1 General Baseline Design

This section contains information on the general properties of the baseline.

4.1.1 Dimensions

The Staging layer is used to prepare data before it is moved into the baseline layer. The stored procedures for the facts and dimensions use the data in the staging tables as the source. For each fact and dimension in the baseline, there is a corresponding staging table. Except for the meta-tagging columns, the columns in the staging table should be one-to-one the same, as those in the fact or dimension. The staging tables have the following meta-tagging columns:

Name	Data Type	Description
Meta_CreateTime	DATETIME	Timestamp showing the time the row was inserted. Default is set to GETDATE().
Meta_CreateJob	BIGINT	Id of the execution that created the row.
Meta_Id	BIGINT	Internal unique identifier. Its values are defined by an identity property.
Meta_ValidFrom	DATETIME	Timestamp

On the staging tables for the dimensions, a primary key is created containing all the business keys, and the Meta_ValidFrom column if the dimension has type 2 history. All the business keys must exist in the corresponding map table. As an example:

```
CONSTRAINT [PK_Dim_<DimensionName>] PRIMARY KEY CLUSTERED ([<BusinessKey1>] ASC, ..., ([Meta_ValidFrom] ASC))
```

On the staging tables for the fact tables there is created a primary key containing all the business keys. As an example:

```
CONSTRAINT [PK_Fact_<FactName>] PRIMARY KEY CLUSTERED ([<BusinessKey1>] ASC, ...)
```

4.1.2 Facts

All facts include the following mandatory columns:

Name	Data Type	Description
Meta_CreateTime	DATETIME	Timestamp when the row is inserted.
Meta_CreateJob	BIGINT	Id of the execution that created the row.

A primary key is created on the business keys, or on a combination of all the foreign keys. As an example:

```
CONSTRAINT [PK_<FactName>] PRIMARY KEY CLUSTERED ([<BusinessKey1>] ASC, ...)
```

5 Fast Track

The Fast Track is designed for handling data by bypassing the archive and the staging layers. The data for the Fast Track does not require building of history and transformations. The Fast Track data is loaded from the extract layer directly into the baseline with a generic stored procedure.

The fast track is used for Stamdata extraction since this is already transformed and has type 2 history implemented hence we have no need to make an archive.

The fast tract tables include the following mandatory columns:

Name	Data Type	Description
Meta_CreateTime	DATETIME	Timestamp when the row is inserted.
Meta_CreateJob	NVARCHAR	Id of the execution that created the row.
Meta_Id	BIGINT	Internal unique identifier. Its values are defined by an identity property.

6 DMSA

The DMSA layer is the interface layer between the baseline and cubes decoupling the two and making it possible to e.g. edit the baseline layer without editing the cube layer.

Each table imported into the SSAS Cube is based on the corresponding view in the SQL database. Each view only handles nullreferences in columns and converts these into relevant nullhandlers:

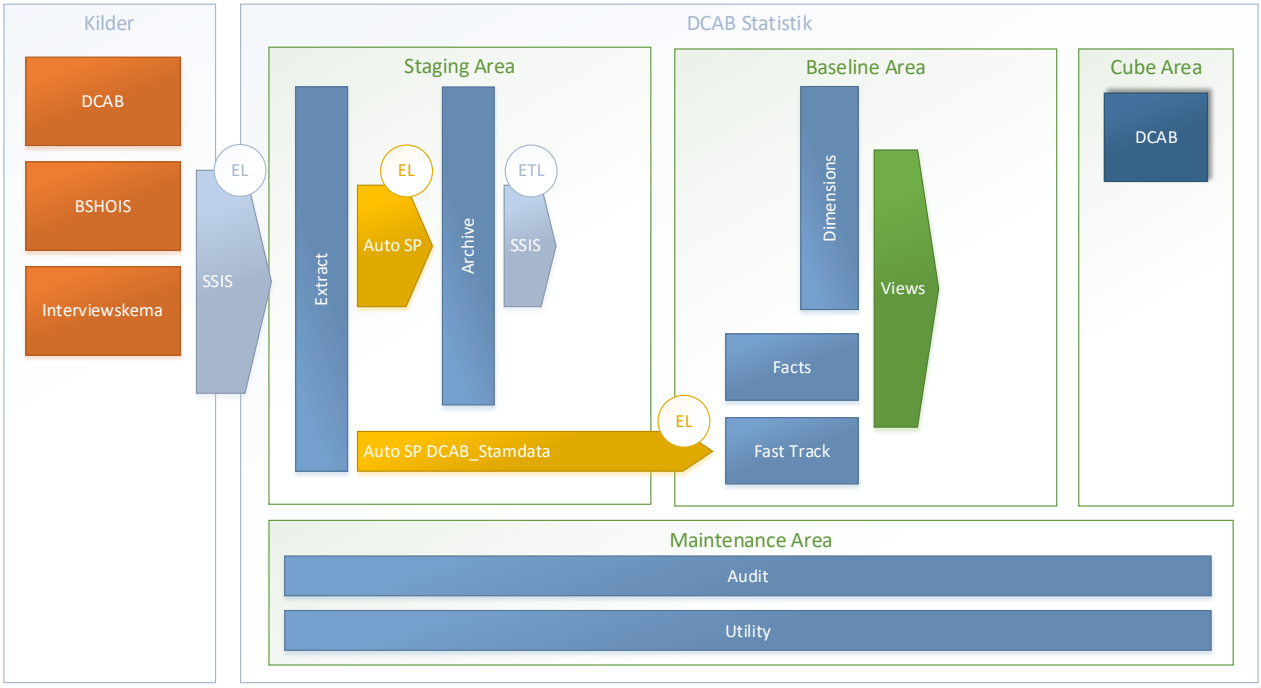
Type	Nullhandling
String	“Empty”
Int	-1
Guid	00000000-0000-0000-0000-000000000000
Datetime	1900-00-00

7 Tabular Models

A Tabular cube of compabitility level of 1500 is installed and used, this model is preferred over a multidimensional model because of the simplicity that fits this solution.

8 Data Flows

This section describes the data flow for the Data Warehouse, how data flows into the extract layer and propagates up to the Data Marts and Cubes. The overall data flow is depicted in the figure below:



The data flows are handled internal in the Data Warehouse using SSIS and stored procedures.

8.1 General

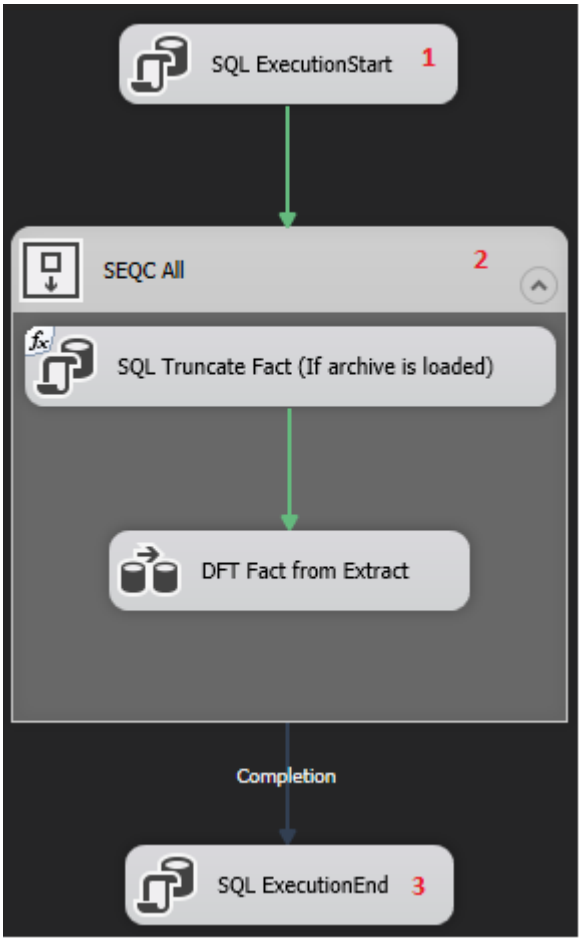
All SSIS packages have as a minimum the following parameters and variables:

Type	Parameter	Description
Parameter	ParentExecutionId	The Id of the parent package executing this one. Default is -1 indicating no parent package.
Variable	ExecutionId	Id of the job entry in the table Audit.ExecutionLog.

Control flow:

All SSIS packages have the following general control flow (depicted in the picture below):

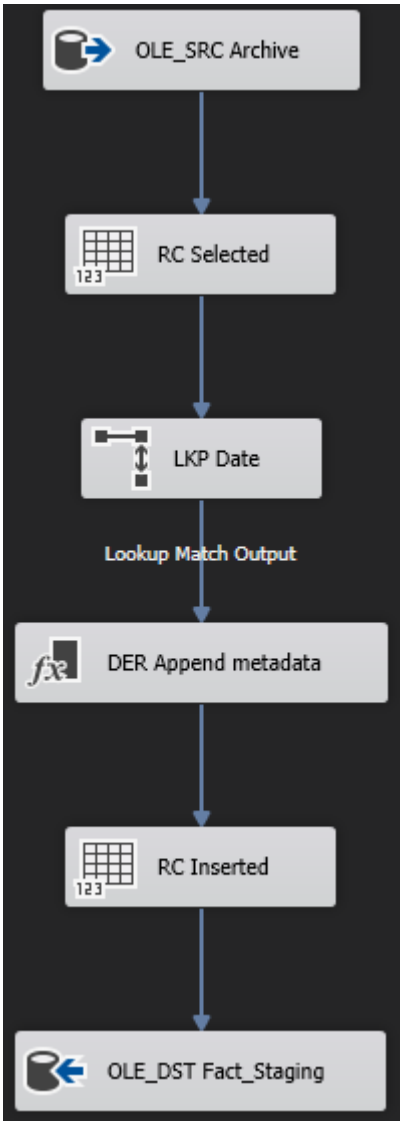
1. Start with an Execute SQL Task, that executes the stored procedure [Audit].[ExecutionStart] to log the start of the package.
2. All customized tasks are wrapped in a sequence container in order to contain any failures within and ensure that the remainder of the package will run.
3. The sequence container is connected to the Execute SQL Tasks, which updates the log by executing the stored procedure [Audit].[ExecutionEnd] conditioned



Data flow:

If a Data flow Task is used some of its tasks will be general to all SSIS packages. You will find the following similarities within all data flow tasks:

1. There will always be some form of a source component and some form of a destination component.
2. “RC Selected” and “RC Inserted” will always be included, because they are used for our auditing procedures. “RC Selected” counts the number of rows that the source component outputs, and “RC Inserted” counts the number of rows inserted into the destination table. These two values can be different in SSIS packages where faulty records are discarded or redirected to an error table.
3. DER Append Metadata is also included in the data flow for all the SSIS packages. It appends the data for the Meta_CreateJob and Meta_CreateDate columns.



8.2 Source to Extract

The source task loads all data from the external data sources without any filters. The select SQL statements uses column names instead of a star(*) to increase stability of the extract layer. This means we only get errors when columns are removed and not when columns are added.

8.3 Extract to Archive

All extract data are migrated to Archive tables using the included Stored procedure that handles Type 2 history that is also known as Slowly Changing Dimensions.

8.4 Archive to Baseline

The load of dimensions and facts to baseline are in general divided into three steps to reduce complexity of the SSIS packages and make debugging easier.

1. Truncate Dimension/Facts
2. Transformation – Archive to Dimension/Facts (SSIS Data Flow Task)

a. This step simply loads all the data from the Archive layer to a staging area.

b. During the load, the data is also transformed, and new columns are derived.

c. This is also where errors in the data can be caught.

Each dimension and fact have its own SSIS package, which in addition to the system parameters must have the following parameters and variables. The facts have the following:

Type	Parameter	Description
Parameters	FactName	The name of the fact.
	ParentExecutionId	The Id of the package that executed the SSIS package, e.g. a master package.
	Source	The primary source table.
Variables	ExecutionId	The Id of the execution.
	ExecutionStatus	The execution status, which could be either “Success” or “Failed”
	TableName	The name of the staging table
	TableSchemaName	The schema name for the staging table

The dimensions have the following:

Type	Parameter	Description
Parameters	DimensionName	The name of the dimension.
	MarkDeletes	If it is true it marks deletes in the dimension.
	ParentExecutionId	The Id of the package that executed the SSIS package, e.g. a master package.
	Source	The primary source table.
	FullLoad	If it is True, the dimension is full loaded
Variables	ExecutionId	The Id of the execution.
	ExecutionStatus	The execution status, which could be either “Success” or “Failed”
	TableName	The name of the staging table
	TableSchemaName	The schema name for the staging table
	SQL_Truncate	SQL query to truncate the staging table

8.5 Scheduling

The data is scheduled with a SQL Agent job that executes every night at 02:00 and executes the Extract Transform Load layers sequentially.

8.6 English to Danish Translations

The columns have been translated from English to Danish to provide a clearer understanding of what data they contain.

Kolonne1	Kolonne2
BfeNumberBfeNr	BfeNummer
BuildingNumber	BygningsNr
BuildingUnitId	BygningsEnhedsId
ComplexId	KompleksId
CvrNumber	CvrNr
NumberOfBuildings	AntalBygninger
OrganizationId	OrganisationsId
ReportDate	RapportDato
BuildingTypeId	BygningsTypeId
BfeNumberString	BfeNummerString
Description	Beskrivelse
AnswerType	SvarType
Comment	Kommentar
Value	Værdi
ArchivedReportId	ArkiveretRapportId
HousingOrganizationLbfNo	BoligorganisationsNummer
BuildingUnitLbfNo	BygningsEnhedsLbfNummer
BuildingUnitName	BygningsEnhedsNavn
BuildingType	BygningsType
Address	Adresse
PostalCode	Postnummer
City	By

YearOfConstruction	Indbygningsår
HousingOrganizationId	BoligorganisationsId
BuildingComplexId	BygningsKompleksId
BuildingComplexLbfNo	BygningsKompleksLbfNummer
BuildingComplexName	BygningsKompleksNavn
BuildingTypeId	BygningsTypeId
UnitsOfMeasure	Enhed
Amount	Mængde
Price	Pris
Text	Beskrivelse
NextMaintenanceYear	NæsteVedligeholdelse
MaintenanceIntervalInYear	IntervalIÅr
HasMolioPrice	HarMolioPris
Factor	Faktor
MolioPriceNumber	MolioPrisNummer
IsMaintenanceExpense	ErVedligeholdelsesudgifter
FixedPrice	FastPris
PurchasePriceMultiplier	KøbsprisMultiplikator
MaintenanceIntervalInYears	VedligeholdelsesintervalIÅr
MolioPriceNumberOld	MolioPrisNummerGammel
ExpectedBygningTyper	ForventetBygningTyper
FeeBygningTyper	GebyrBygningTyper
FeeBygningTyperTotal	GebyrBygningTyperTotal
ValidEnergyLabelM2HeatArea	GyldigEnergimærkeM2VarmeOmråde
ValidEnergyLabelsHourlyFee	GyldigeEnergimærkerTimeGebyr
ValidEnergyLabelsTotalFee	GyldigeEnergimærkerSamletGebyr
InValidEnergyLabelM2HeatArea	UgyldigEnergimærkeM2VarmeOmråde
InValidEnergyLabelsHourlyFee	UgyldigeEnergimærkerTimeGebyr

InValidEnergyLabelsTotalFee	UgyldigeEnergimærkerSamletGebyr
MissingEnergyLabelM2HeatArea	ManglendeEnergimærkeM2Varmeområde
MissingEnergyLabelsHourlyFee	ManglendeEnergimærkerTimegebyr
MissingEnergyLabelsTotalFee	ManglendeEnergimærkerTotalgebyr
IsTotal	ErTotal
Ignored	Ignoreret
Note	Note
SharedOwnership	FællesEjerskab
AccountNumber	KontoNummer
AccountNumberDescription	KontoNummerBeskrivelse
InstallationYear	IndbygningsÅr
RemainingLifeTimeYear	RestlevetidÅr
AnnualContractorExpensesExclVAT	ÅrligHåndværkerUdgifterExMoms
AnnualExpenseVAT	ÅrligMomsUdgift
AnnualMaintenanceWorkExclVat	VedligeholdelsePrÅrExMoms
AnnualMaintenanceWorkInclVAT	VedligeholdelsePrÅrInclMoms
AnnualMaintenanceWorkVAT	VedligeholdelsePrÅrMoms
ConstructionSiteExpenses	ByggepladsOmkostninger
ContractorExpensesInclVAT	SumHåndværkerUdgifterInclMoms
ProjectionArchitectEngineer	ProjekteringArkitektIngeniør
RestablishmentAroundConstructionSite	ReetableringerOmkringBygning
Comment	Kommentarfelt
BuildingPartLifeTimeNumberOfYears	LevetidBygningsdel
FloorSizeM2	GulvStørrelseM2
NumberOfFloors	AntalEtager
HasBasement	HarKælder
AsphaltPavementM2	AsfaltBelægningMængdeM2
ConcretePavingStoneM2	BetonBrostenMængdeM2

ConcretePavementCurbLbm	BetonFortovsKantMængdeLbm
LooseCoatingTypeM2	LøsBelægningsTypeMængdeM²
Wire110Value	Ø110LedningMængdeLbm
Wire160Value	Ø160LedningMængdeLbm
Wire200Value	Ø200LedningMængdeLbm
PpWellD3_0Value	PPBrøndMængdeStk
PpWellD2_5Value	PPBrøndMængdeStkk
BreakingUpAndRestoringCoatings	OpbrydningRetableringBelægningerM2
ExternalPumpWells	UdvendigePumpebrøndeStk
FloodExtinguisher	HøjvandslukkerStk
GreaseOilSeparators	FedtOlieUdskillereStk
LeakageSystems	NedsivningsAnlægInklLarStk
FacadeSizeM2	FacadeM2
RemainingLifeTimeYearFacadeType1	RestlevetidÅrFacadeType1
RemainingLifeTimeYearFacadeType2	RestlevetidÅrFacadeType2
BuildingPartLifeTimeNumberOfYears	LevetidBygningsdel
ExteriorWallValue	YdervægMængdeM2
Selection2Value	YdervægMængdeM22
Selection3Value	PudsMængdeM2
Selection4Value	IsoleringMængdeM2
AsbestosOrHazardousMaterialsInFacadeValue	AsbestMiljøskadeligeStofferFacadeVærdi
AsbestosOrHazardousMaterialsInFacadeUnits	AsbestMiljøskadeligeStofferFacadeDeler
PcbInFacadeValue	PCBIFacadeVærdi
PcbInFacadeUnits	PCBIFacadeDeler
DilationJoints	Dilitationsfuger
FacadeGrout	OmfugningAfFacader
HasPcbInFacade	HarPCBIFacade
HasHazardousMaterialsInFacade	AsbestMiljøskadeligeStofferFacade

NumberOfDormers	AntalKviste
HazardousSubstancesInRoofingMaterial	FarligeStoffer
DormerHasPaintedWoodTrim	MaletTræbeklædning
RoofTerraceSizeM2	TagterrasseStørrelseM2
TerraceRailings	TerrasseRækværk
HazardousSubstancesInTerracePaintAmount	FarligeStofferTerrasseMalingsMængde
HazardousSubstancesInRailingPaintAmount	FarligeStofferGelænderMalingMængde
PcbInGroutAmount	PcbFugeMængde
FloorLength	Gulvlængde
FloorWidth	Gulvbredde
RoofSlope	TagHældning
Gables	Gavle
RoofingPlusAccessories	TagdækningPlusTilbehør
ReplacedWindEdging	UdskiftetVindkant
WoodWindEdging	TræVindkant
NumberOfSkylights	AntalOvenlysvinduer
RoofValley	RoofValley
NumberOfRoofDrainsForInspection	AntalTagafløbEftersyn
ExistingRoofWithoutAsbestos	EksisterendeTagUdenAsbest
ExistingRoofWithAsbestos	EksisterendeTagMedAsbest
NumberOfBalconies	AntalAltaner
AverageBalconySizeM2	GennemsnitligAltanstørrelseM2
HasPcbInJoints	HarPcbLed
PaintContainsHazardousMaterials	FarligeStofferMaling
TotalLengthOfRailingForAllBalconies	SamletLængdeRækværkAlleAltaner
NumberOfExposedWalkways	AntalUdsatteGangbroer
ExposedWalkwayM2	UdsatGangbroM2
TotalLenghtOfRailngsForWalkways	SamletLængdeRækværkGangbroer

PaintContainsHazardousMaterials	MalingFarligeMaterialer
StairTypeSelectionAmount	TrappeTypeValgBeløb
StairCaseMaterialTypeSelectionAmount	TrappeMaterialeTypeValgMængde
PaintContainsHazardousMaterials	MalingFarligeMaterialer
PaintContainsHazardousMaterialsAmount	MalingFarligeMaterialerMængde
DoorSelection1Amount	Dørvalg1Beløb
DoorSelection2Amount	Dørvalg2Beløb
DoorSelection3Amount	Dørvalg3Beløb
DoorSelection1RemainingLife	Dørvalg1BeløbResterendeLiv
DoorSelection1LifeTime	Dørvalg1BeløbLivstid
DoorSelection2RemainingLife	Dørvalg2BeløbResterendeLiv
DoorSelection2LifeTime	Dørvalg2BeløbLivstid
DoorSelection3RemainingLife	Dørvalg3BeløbResterendeLiv
DoorSelection3LifeTime	Dørvalg3BeløbLivstid
DoorsContainingPcbSealant	DøreIndeholderPcbTætningsmiddel
DoorsPaintedWithHazardousMaterialPaint	DøreFarligtMaterialemaling
ElectricDoorLocks1560WithLowSecurity	ElektriskeDørlåse1560LavSikkerhed
NumberOfWindows	AntalVindues
HasPcbInGlazing	HarPcbRuder
HasHazardousMaterialsInPaint	HarFarligeStofferFarve
HasWindowStoolBeenReplaced	HarVindueskarmStollBlevetByttet
HasWindowFrameBeenChanged	HarVinduesrammeBlevetByttet
HasWindowSillBeenreplaced	HarVindueskarmSillBlevetByttet
SumOfAreaType1	SummeAreaType1
SumOfAreaType2	SummeAreaType2
TotalCircumferenceType1	SamletOmkredsType1
TotalCircumferenceType2	SamletOmkredsType2
WindowPanesTotalType1	VinduesglasTotalType1

WindowPanesTotalType2	VinduesglasTotalType2
WindowStoolSillTotalType1	VindueSkarmTotalType1
WindowStoolSillTotalType2	VindueSkarmTotalType2
TiledAreaM2	FlisebelagtArealM2
PaintedAreaM2	MaletOmrådeM2
TileAdhesiveAreaM2	FliseKlæbeArealM2
GroutedAreaLbm	FugedeArealLbm
StairsPaintedWithHazardousSubstances	TrapperMaletFarligeStoffer
StairDescription1Amount	Trappebeskrivelse1Mængde
StairDescription2Amount	Trappebeskrivelse2Mængde
HasStairPaintHazardousSubstances	HarTrappemalingFarligeStoffer
ReplacementOfGutterAmount	UdskiftningTagrendeMængde
ReplacementOfDownSpoutAmount	UdskiftningDuntudMængde
SupplementOfDownSpoutsAmount	TillægDuntudeMængde
DownSpoutTypeAmount	NedtudTypeMængde
CreatingManholeCovers	OpretteKloakdæksel
ReplacementManholeCoverRoofDrainWell	UdskiftningKloakdækselTagbrønden
ReplacementManholeCoverRoadWell	UdskiftningKloakdækselVejbrønd
ReplacementManholeCoverDownhole	UdskiftningKloakdækselHullet
GutterAlignmentWithIncorrectSlope	TagrendejusteringForkertHældning
DrainageLifetime	DræningLevetid
SanitationLifetime	SanitetsLevetid
InternalPumpWellsAmount	InternePumpebrøndeMængde
NewElectricalSystemSelectionAmount	NytElsystemValgMængde
SolarCellSystemAmount	SolcellesystemMængde
ElectricalSwitchboardSystemAmount	ElTavleSystemMængde
GasSystemAmount	GasSystemMængde
ReplacementofCirculationPumpAmount	UdskiftningCirkulationspumpeMængde

CleaningOfHotWaterTankAmount	RengøringVarmtvandsbeholderMængde
WaterTreatmentServiceAmount	VandbehandlingServiceBeløb
InsulationOfWaterPipesInNewPlant	IsoleringVandrørNytAnlæg
AsbestosInExistingInsulation	AsbestEksisterendeIsolering
ReplacementOfCirculationPump	UdskiftningCirkulationspumpe
ReplacementOfThermostatValve	UdskiftningTermostatventil
ServiceAutomationSystem	ServiceAutomationSystem
HeatPumpAmount	Varmepumpemængde
InsulationOfHeatingPipes	IsoleringVarmerør
AsbestosInExistingInsulation	AsbestEksisterendeIsolering
NumberOfVentilationSystems	AntalVentilationssystemer
VerntilationPipes	Verntilationsrør
InsulationOfVentilationPipes	IsoleringVentilationsrør
AsbestosInExistingInsulation	AsbestEksisterendeIsolering
FromYear	FraÅr
ToYear	TilÅr
Part	Del
To	Til
SendDate	SendDato
MunuiciplaityCode	Kommunekode
IsSuccessful	ErVellykket
ErrorMessage	FejlBesked
EntityIdentifier	EnhedsIdentifikator
LabelStatusCode	EtiketStatusKode
HasPdf	HarPdf
HasXml	HarXml
LabelStatus	EtiketStatus
EnergyLabelSerialIdentifier	EnergiMærkeSerielIdentifikator

EnergyLabelClassification	EnergiMærkeKlassificering
ValidFrom	GældendeFra
ValidTo	GyldigTil
StreetName	GadeNavn
HouseNumber	HusNummer
ZipCode	PostNummer
CityName	ByNavn
MunicipalityNumber	KommuneNummer
PropertyNumber	EjendomsNummer
BuildingNumber	BygningsNummer
BbruseCode	BbruseKode
HeatSupply	VarmeForsyning
EnergyLabelTypeUsage	BrugAfEnergiMærkeType
EnergyLabelTypeBasedOn	EnergiMærkeTypeBaseretPå
IsMixedUsage	ErBlandetBrug
SubmitterConsultantName	IndsenderKonsulentNavn
SubmitterCompanyName	IndsenderFirmaNavn
SchemaVersion	SkemaVersion
Wgs84Latitude	Wgs84Breddegrad
Wgs84Longitude	Wgs84Længdegrad
IsHidden	ErSkjult
ProcessedResultXmldownload	BehandletResultatXmlDownload
ProcessedResultPdfdownload	BehandletResultatPdfDownload
ProcessedResultXmlparse	BehandletResultatXmlparse
ProcessedResultPdfinsert	BehandletResultatPdfIndsæt
IsDeleted	ErSlettet/Slettet
PeriodYear	PeriodeÅr
IsCurrentPeriod	NuværendePeriode

BuildingNumberBBR	BygningsNummerBBR
LbfNumber	LbfNummer
EnergyClassification	EnergiKlassificering
MotherPropertyNumber	ModerEjendomsNummer
MotherMunicipalityNumber	ModerKommuneNummer
ConsumptionKwhPerM2	ForbrugKwhPerM2
Co2EmissionPerYear	Co2EmissionPerÅr
Locked	Låst
DirectEletricHeatingOfRoomKwhPerM2	DirekteElektriskOpvarmningRumKwhPerM2
VentilationHeatingSurfaceKwhPerM2	VentilationVarmeOverfladeKwhPerM2
HeatPumpsForRoomHeatingKwhPerM2	VarmepumperRumOpvarmningKwhPerM2
HeatPumpsForVBHHeatingKwhPerM2	VarmepumperVBHVarmeKwhPerM2
PipeTracingKwhPerM2	RørSporingKwhPerM2
ElectricHeatingOfCentralVVBKwhPerM2	ElvarmeCentralVVBKwhPerM2
DecentralEletricWaterHeaterKwhPerM2	DecentralElektriskVandVarmerKwhPerM2
TotalElectricityNeedsForHeatingKwhPerM2	SamletElektricitetsBehovOpvarmningKwhPerM2
ActualPowerOfSolarCellsKwhPerM2	FaktiskEffektSolcellerKwhPerM2
ActualPerformanceOfWindTurbinesKwhPerM2	FaktiskYdeevneVindmøllerKwhPerM2
TotalActualElectricityProductionKwPerM2	SamletFaktiskElproduktionKwPerM2
SolarCellsKwhPerM2Heat	SolCellerKwhPerM2Varme
SolarCellsKwhPerM2Electricity	SolCellerKwhPerM2Elektricitet
WindTurbinesKwhPerM2Heat	VindMøllerKwhPerM2Varme
WindTurbinesKwhPerM2Electricity	VindMøllerKwhPerM2Elektricitet
HeatPumpsKwhPerM2	VarmePumpeKwhPerM2
SolarHeatingSystemKwhPerM2	SolVarmeSystemKwhPerM2
PowerConsumptionInDistrictHeatingExchangerOrBoilerKwhPerM2	StrømForbrugFjernVarmeVekslerEllerKedelKwhPrM2
HeatDistributionPumpsKwhPerM2	VarmeFordelingsPumperKwhPerM2

LightingKwhPerM2	BelysningKwhPerM2
VBVPumpsKwhPerM2	VBVPumperKwhPerM2
StandbyConsumptionInHeatPumpsKwhPerM2	StandbyForbrugVarmepumperKwhPerM2
CoolingKwhPerM2	KølingKwhPerM2
OtherLightingKwhPerM2	AndenBelysningKwhPerM2
SpecialEquipmentKwhPerM2	SpecialUdstyrKwhPerM2
ConsumptionEquipmentUnderInternalHeatSupplementKwhPerM2	ForbrugUdstyrUnderInterntVarmeSupplementKwhPerM2
TotalElectricityNeedsForOtherThanHeatingKwhPerM2	SamletElektricitetsBehovAndetEndOpvarmningKwhPerM2
DistrictHeatingOrBoilerKwhPerM2	FjernvarmeEllerKedelKwhPerM2
DirectElectricHeatingKwhPerM2	DirekteElVarmeKwhPerM2
HeatersKwhPerM2	VarmeApparaterKwhPerM2
VentilationHeatingSurfaceKwhPerM2	VentilationVarmeOverfladeKwhPerM2
TotalCoverageOfHeatingNeedsKwhPerM2	SamletDækningVarmebehovKwhPerM2
ElectricHeatingOfCentralVVBKwhM2	ElvarmeCentralVVBKwhM2
DecentralElectricWaterHeaterKwhPerM2	DecentralElektriskVandVarmerKwhPerM2
DecentralGasWaterHeaterKwhPerM2	DecentralGasVandVarmerKwhPerM2
TotalCoverageOfHeatingNeedsInVBVSystemsKwhPerM2	SamletDækningVarmeBehovVBVSystemerKwhPerM2
RegistrationYear	RegistreringsÅr
ActivitiesTotalCostInclVAT	AktiviteterTotalOmkostningerInklMoms
ActivitiesTotalCostInclVATOriginal	AktiviteterTotalOmkostningerInklMomsOriginal
CreatedWith	OprettetMed

9 Maintenance

The maintenance area contains the auditing and the utility schema.

9.1 Utility

The utility schema contains a list of tables and stored procedures that are used for general maintenance of the data warehouse. The components in the utility schema does not add business value, but they assist in running the data warehouse.

9.1.1 Utility.GenerateCalendar

This procedure generates the calendar data used by the calendar dimension. Once the procedure is executed it inserts a record for each date from the “Date” to the “EndDate” into the Utility.Calendar table. By default, the procedure is executed automatically by the post deploy script with the Date value set to 2000-01-01 and the EndDate value set to 2099-12-31. The procedure uses the following parameters:

Parameter Name	Data Type	Default Value	Output Value	Comments
Date	DATETIME		No	The start date for the procedure
EndDate	DATETIME		No	The end date for the procedure

9.1.2 TruncateTables

TruncateTables truncates all tables within a specific schema. The procedure takes the following parameters:

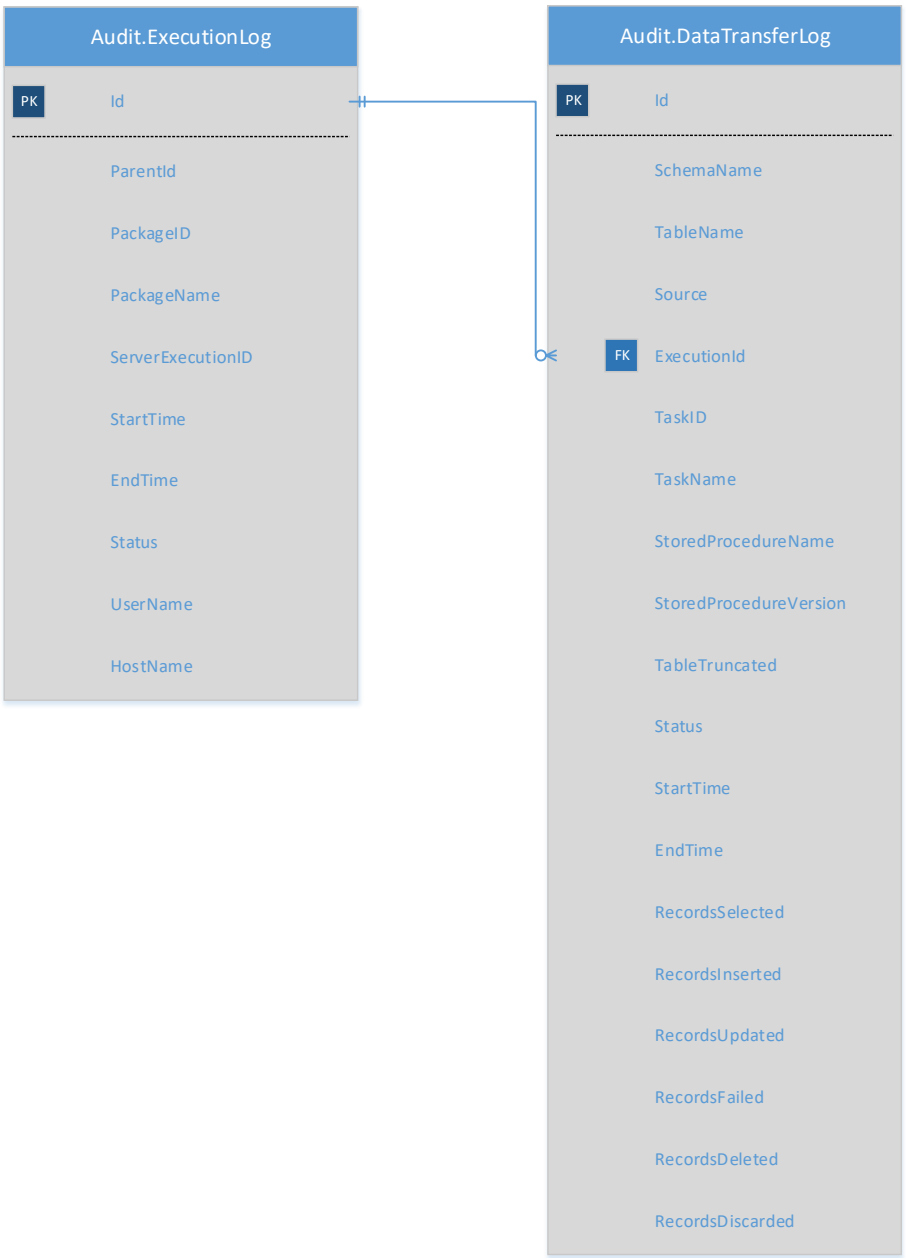
Parameter Name	Data Type	Default Value	Output Value	Comments
SchemaName	SYSNAME		No	The name of the schema that stores the tables, which have to be truncated.
TableNamePattern	SYSNAME	%	No	The name pattern of the tables in the schema, that have to be truncated.

9.2 Audit

The Audit functionality provides an overview of what has happened during the ETL processes. Auditing takes place every time data is moved from one table to another. The auditing is done by two set of procedures DataTransferStart/DataTransferEnd and ExecutionStart/ExecutionEnd. The Execution stored procedures record data in the [Audit].[ExecutionLog] table which contains data on the overall SSIS package, such as start time, package name who executed the package etc. The data transfer procedures are called for every data flow task in a SSIS package, or every time data is transferred by a stored procedure. These procedures insert data into the Audit.DataTransferLog table with information on the number of records that were selected, updated, deleted etc. The Audit functionality logs the status of the processes all the way from the Extract layer through Archive, Staging and Baseline. Note that for the Extract and Archive tables separate procedures and auditing tables have been created. These procedures and table however are functionally the same as DataTransferStart/DataTransferEnd and the DataTransferLog table. The only difference is that the Extract table and procedures does not contain the columns that are specific to the data transfers

by stored procedures and the Archive table and procedures does not contain the columns that are specific to data transfers by SSIS packages.

The data model between the two tables is illustrated below:



9.2.1 ExecutionLog

This table contains information on the execution process itself. The ExecutionLog table contains the following columns:

Name	Data Type	Description
Id	BIGINT	The execution id of the record. This column uses an identity property.
ParentId	BIGINT	The execution id of a parent package, e.g. if the SSIS package was executed by a master package, the parent id would

Name	Data Type	Description
		be the execution id of the master package.
PackageId	NVARCHAR(50)	The Id of the SSIS package
PackageName	NVARCHAR(50)	The name of the SSIS package
ServerExecutionID	BIGINT	The server execution id of the execution
StartTime	DATETIME2	The start time of the execution.
EndTime	DATETIME2	The end time of the execution.
Status	NVARCHAR(20)	The status of the execution.
UserName	NVARCHAR(255)	The username of the account that executed the procedure.
HostName	NVARCHAR(255)	The server on which the procedure was executed.

9.2.2 DataTransferLog

This table contains data on the number of rows that were used in the execution. The ExecutionLog table contains the following columns:

Name	Data Type	Description
Id	BIGINT	The DataTansferLog id. This value is created by an identity property.
SchemaName	NVARCHAR(255)	The schema name of the destination table
TableName	NVARCHAR(255)	The name of the destination table
Source	NVARCHAR(1000)	The primary source table
ExecutionId	BIGINT	A foreign key reference to the [Audit].[ExecutionLog] table.
TaskId	NVARCHAR(50)	The SourceID of the SSIS data flow task. This value is only used when logging the data transfer of a SSIS data flow task.
TaskName	NVARCHAR(260)	The name of the SSIS data flow task. This value is only used when logging the data transfer of a SSIS data flow task.

Name	Data Type	Description
StoredProcedureName	NVARCHAR(255)	The name of the stored procedure. This value is only used when logging the data transfer of a stored procedure.
StoredProcedureVersion	NVARCHAR(255)	The version of the stored procedure. This value is only used when logging the data transfer of a stored procedure.
TableTruncated	BIT	The value is 1 when the table was truncated before inserting data. Otherwise it is 0. This value is only used when logging the data transfer of a stored procedure.
Status	NVARCHAR(20)	The status of the data transfer process.
StartTime	DATETIME2	The start time of the data transfer process.
EndTime	DATETIME2	The end time of the data transfer process.
RecordsSelected	INT	The numbers of records selected by the process.
RecordsInserted	INT	The numbers of records inserted by the process.
RecordsUpdated	INT	The numbers of records updated by the process.
RecordsFailed	INT	The numbers of records that failed in the process.
RecordsDeleted	INT	The numbers of records deleted by the process.
RecordsDiscarded	INT	The numbers of records that were discarded during the process.

9.2.3 Audit.DataTransferStart

The DataTransferStart procedure is used for recording information on the data transferring process it is auditing. The record is stored in the DataTransferLog table. For the purpose of our data warehouse the data transfer processes could be either a SSIS data flow task or a stored procedure. The record created by the DataTransferStart procedure contains information on the data transformation process itself e.g. the name of the stored procedure, the table the data is inserted to etc. By default the procedure is executed by the OnPreExecutables Event Handler, for every SSIS data flow task. The procedure uses the following parameters:

Parameter Name	Data Type	Default Value	Output Value	Comments
SchemaName	NVARCHAR(255)		No	The schema name of the table the data is transferred to.
TableName	NVARCHAR(255)		No	The name of the table the data is transferred to.
DataTramsferId	BIGINT	-1	Yes	The data transfer record id.
ExecutionId	BIGINT	NULL	No	The id of the execution in the Audit.ExecutionLog table.
Source	NVARCHAR(1000)	NULL	No	The primary source of the data
TaskId	NVARCHAR(50)	NULL	No	The SourceID of the SSIS data flow task. This value is only used when logging the data transfer of a SSIS data flow task.
TaskName	NVARCHAR(260)	NULL	No	The name of the SSIS data flow task. This value is only used when logging the data transfer of a SSIS data flow task.
StoredProcedureName	NVARCHAR(255)	NULL	No	The name of the stored procedure. This value is only used when logging the data transfer of a stored procedure.
StoredProcedureVersion	NVARCHAR(255)	NULL	No	The version of the stored procedure. This value is only used when logging the data transfer of a stored procedure.
TableTruncated	BIT	NULL	No	The value is 1 when the table was truncated before inserting data. Otherwise it is 0. This

Parameter Name	Data Type	Default Value	Output Value	Comments
				value is only used when logging the data transfer of a stored procedure.
Status	NVARCHAR(10)	Started	No	The status of the data transfer process.

9.2.4 Audit.DataTransferEnd

The DataTransferEnd procedure is used for updating the already created record in the DataTransferLog table. The procedure adds information on the number of records that were transferred, as well as a status on the data transfer. This is done by first identifying the previously created record in the DataTransferLog table. The procedure does this by filtering the table on the Id column using the DataTransferId value. Once the record has been identified it is updated with information on the number of records that were selected, updated, deleted etc. as well as a status on the process. By default, the procedure is executed by the OnPostExecutables Event Handler, for every SSIS data flow task. The procedure uses the following parameters:

Parameter Name	Data Type	Default Value	Output Value	Comments
DataTransferId	BIGINT		No	The Id of the active process in the DataTransferLog table.
Status	NVARCHAR(20)	Succeeded	No	The status of the data transfer process
RecordsSelected	INT	0	No	The numbers of records selected by the process.
RecordsInserted	INT	0	No	The numbers of records inserted by the process.
RecordsUpdated	INT	0	No	The numbers of records updated by the process.
RecordsDeleted	INT	0	No	The numbers of records deleted by the process.
RecordsFailed	INT	0	No	The numbers of records that failed in the process.
RecordsDiscarded	INT	0	No	The numbers of records that were discarded during the process.

9.2.5 Audit.ExecutionStart

This procedure is executed at the start of every SSIS package. It creates a record in the ExecutionLog table with information on the time the package started, the name of the SSIS package, the user that executed it etc. The procedure uses the following parameters:

Parameter Name	Data Type	Default Value	Output Value	Comments
PackageName	NVARCHAR(260)		No	The name of the SSIS package
PackageId	NVARCHAR(50)		No	The Id of the SSIS package
ServerExecutionID	BIGINT	0	No	The server execution id of the execution
Id	BIGINT	0	Yes	The execution id of the record
ParentId	BIGINT	-1	No	The execution id of a parent package, e.g. if the SSIS package was executed by a master package, the parent id would be the execution id of the master package.
Status	NVARCHAR(10)	Started	No	The status of the execution.

9.2.6 Audit.ExecutionEnd

ExecutionEnd is used to give the final information on the execution. It will receive the Execution ID of the already created record and update it with a status and the end time of the execution. The procedure uses the following parameters:

Parameter Name	Data Type	Default Value	Output Value	Comments
Id	BIGINT		No	The execution id
Status	NVARCHAR(10)	Succeeded	No	The status of the execution.