DS Customer Interaction Data Model

(Data Stitching)

# Table of Contents

[**Table of Contents 1**](#_mpqtfk6c5x60)

[**Power BI Model 2**](#_z9eusox1x5y2)

[**Data Model 3**](#_ml1wzcvc47)

[dim\_order\_dse\_channel 3](#_h0qpfrbg3y3y)

[dim\_order\_dse\_lob 3](#_ud8ge3n7s29f)

[dim\_product\_dse 4](#_tefa02jv8svh)

[fact\_order\_dse 6](#_n5cr5cl4akz4)

[fact\_login\_tels 9](#_9fszwirlbspd)

[dim\_location\_dse 11](#_h590lqyl13rq)

[dim\_contact\_dse 15](#_qcc9e0etbebg)

[dim\_date 17](#_d7m0knb86uwq)

[dim\_contact\_dssi 20](#_dal6ktiitdzb)

[dim\_location\_dssi 22](#_4bp2sxtoxk99)

[fact\_order\_dssi 24](#_mgaitrieskkw)

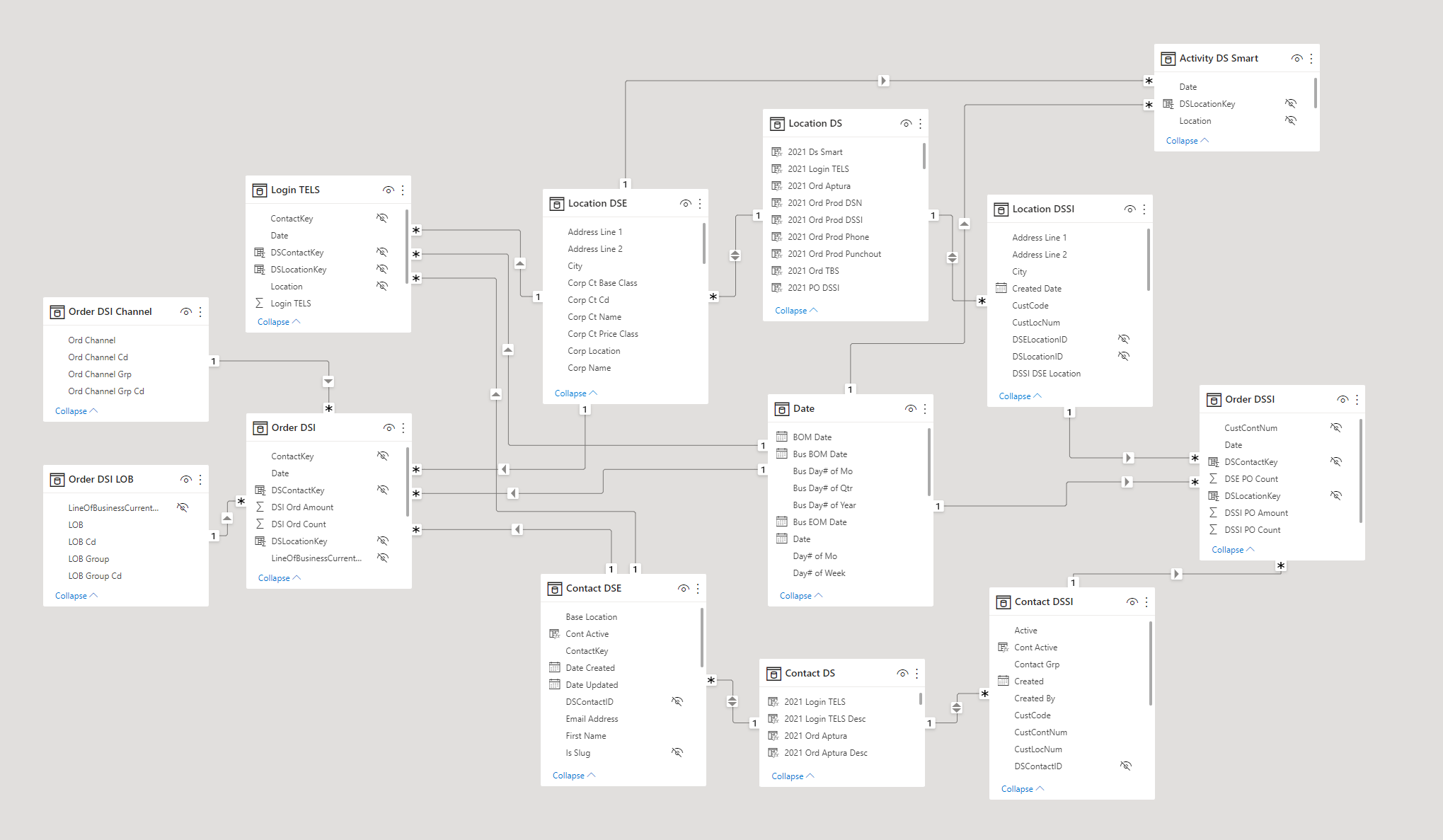
[fact\_activity\_dssmart 26](#_amc8j3obrtoc)

[dim\_location\_ds 27](#_y5a2wdy298k8)

[dim\_contact\_ds 28](#_9g0k1m5gcj2t)

# 

# Power BI Model



# Data Model

## dim\_order\_dse\_channel

List of Ordering channels used for DSE Orders. Based on FinProfit DM

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | OrderChannelCd | string | Ordering Channel Code used when placing the order |
|  | OrderChannelDesc | string | Name of Ordering Channel (Phone, DSSI, etc.) |
|  | OrderChannelGroupDesc | string | Ordering Channel Group Name (Phone/E-commerce) |
|  | OrderChannelSubGroupDesc | string | Ordering Channel Sub-group Name |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Model Name: Order DSE Channel
* Model Query:

| select distinct l.OrderChannelCd, l.OrderChannelDesc, l.OrderChannelGroupDesc, l.OrderChannelSubGroupDesc from dbo.factline l  WHERE l.OrderChannelCd NOT IN ('A', 'C', 'M', 'O') |
| --- |

* Model Steps:
  + ~~RenameColumns(Source,{{"OrderChannelCd", "Ord Channel Cd"}, {"OrderChannelDesc", "Ord Channel"}, {"OrderChannelGroupDesc", "Ord Channel Grp"}, {"OrderChannelSubGroupDesc", "Ord Channel Sub Grp"}})~~

## dim\_order\_dse\_lob

List of Direct Supply Lines of Business

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | LineOfBusinessCurrentKey | int | Generated Identifier |
|  | LOB | string | Line of Business Name (Aptura, Supply, Capital, etc.) |
| SK | LOBCd | string | Line of Business Code (01, 02, etc.) |
|  | LOBGroup | string | Line of Business Group Name (Products, Services, etc.) |
|  | LOBGroupCd | string | Line of Business Group Code (A, B, etc.) |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Model Name: Order DSE LOB
* Model Query:

| select LobCurrentKey as LineOfBusinessCurrentKey, b.LobCd, b.LobNm, b.LobGroupCd, b.LobGroupNm from DimLineOfBusinessCurrent b |
| --- |

* Model Steps:
  + RenameColumns(Source,{{"LobNm", "LOB"}, {"LobCd", "LOB Cd"}, {"LobGroupCd", "LOB Group Cd"}, {"LobGroupNm", "LOB Group"}})

## dim\_product\_dse

Dsi products available and ordered

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | ProductCurrentKey | int | Product source identifier (DimProductCurrent.ProductCurrentKey) |
|  | ProductNbr | string | DS (categorized) or Supplier (non-categorized) product number |
|  | ProductSegmentId | int | DS product segment identifier (first category tier) |
|  | ProductSegment | string | DS product segment name |
|  | ProductProgramId | int | DS product program identifier (second category tier) |
|  | ProductProgram | string | DS product program name |
|  | ProductSubGroupId | string | DS product sub group identifier (third category tier) |
|  | ProductSubGroup | string | DS product sub group name |
|  | ProductPriceFamily | string | DS product price family. Similar to product model name |
|  | ProductType | string | DS product type |
|  | ProductSubType | string | DS product sub-type |
|  | ProductPreFlatNbr | string | DS product number before prem flattening (delineates similar products) |
|  | ProductDesc | string | Line/Product description |
|  | SupplierCd | string | Supplier Code |
|  | SupplierNm | string | Supplier Name |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Query:

| select  p.ProductCurrentKey  , p.SourceNonCatProdNbr as ProductNbr  , p.SegmentId AS ProductSegmentId  , p.SegmentNm AS ProductSegment  , p.ProgramId AS ProductProgramId  , p.ProgramNm AS ProductProgram  , UPPER(RTRIM(p.SubGroupCd)) AS ProductSubGroupId  , p.SubGroupNm AS ProductSubGroup  , p.PrcFamilyNm as ProductPriceFamily  , p.TypeNm as ProductType  , p.SubTypeNm as ProductSubType  , p.PreFlatSourceCatProdNbr as ProductPreFlatNbr  , P.ProductDesc  , UPPER(RTRIM(p.SupplierCd)) as SupplierCd  , p.SupplierNm  from dbo.DimProductCurrent p |
| --- |

## fact\_order\_dse

Order transaction line detail for customer order and returns for last 7 calendar years plus current. Does not include DSSI purchase orders not ordered through E&F. Based on FinProfit DM

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | LineId | int | Line source identifier (TrLines.LineID) |
| FK | OrderDate | date | Date order was placed (ref dim\_date) |
| FK | Location | string | Order Location (ref dim\_location\_dse) |
| FK | ContactKey | string | Person/Customer who placed order (ref dim\_contact\_dse) |
| FK | LineOfBusinessCurrentKey | int | Order Line of Business (ref dim\_order\_dse\_lob) |
| FK | OrderChannelCd | string | Ordering channel used to place order (ref dim\_order\_dse\_channel) |
| FK | ProductCurrentKey | int | Product ordered (ref dim\_product\_dse) |
|  | OrderCancelled | bool | Is Order Canceled |
|  | OrderNbr | string | Order Number |
|  | OrderWhen | timestamp | Date and Time order was placed |
|  | LineQty | numeric(18,4) | Line units ordered |
|  | LineUnitSell | numeric(18,4) | Line sell price per unit |
|  | LineAmount | numeric(18,4) | Total line amount (LineQty\*LineUnitSell + line charges) |
|  | LineUMCd | string | Unit of measure (e.g. case, each…) |
|  | OrderCorpContractCd | string | Contract Code at time of order |
|  | OrderCorpContractNm | string | Contract Name at time of order |
|  | OrderGPOContractCd | string | (Optional) GPO Contract Code at time of order |
|  | OrderGPOContractNm | string | (Optional) GPO Contract Name at time of order |
|  | ContractedProduct | boolean | Contracted product (price) at time of order |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Query:

| select  l.LineId  , x.Date  , UPPER(RTRIM(f.SourceLocationId)) as Location  , UPPER(RTRIM(c.SourceContactKey)) as ContactKey  , l.LineOfBusinessCurrentKey  , l.OrderChannelCd  , l.ProductCurrentKey  , cast(case when l.LineStatusCd = 'I' then 1 else 0 end as bit) as OrderCancelled  , UPPER(RTRIM(l.OrderNbr)) as OrderNbr  , l.OrderWhen  , x.LineQty  , l.LineUnitSell  , x.LineAmount  , l.UnitMeasureCd as LineUMCd  , UPPER(RTRIM(l.OrderCorpContractCd)) as CorpContractCd  , l.OrderCorpContractNm as CorpContractNm  , UPPER(RTRIM(l.OrderGPOContractCd)) as GPOContractCd  , l.OrderGPOContractNm as GPOContractNm  , x.Contracted  from (  select  d.Date,  ISNULL(l.LineUnitQty, lo.Qty) As LineQty,  ISNULL(l.BK\_ProductSales, lo.Qty\*lo.UnitSell) AS LineAmount,  l.LineId,  Contracted = cast(max(case when mi.RefNum is not null then 1 else 0 end) as bit)  from dbo.factline l  INNER JOIN dbo.DimDate d on d.DateKey = OrderDateKey  INNER JOIN dbo.DimProductCurrent p on p.ProductCurrentKey = l.ProductCurrentKey  LEFT JOIN MDID\_TRAN.dbo.TrLines lo on lo.LineID = l.LineID  LEFT JOIN MDID\_Btrieve.dbo.clcontyp nact  inner join USER\_SUPPLY\_CHAIN.dbo.pm\_NA\_ContractProductListExclusionsHist nx  ON nx.Contract = nact.[Contract]  --check for contracted net sell  inner join USER\_SUPPLY\_CHAIN.dbo.pm\_NA\_ProductListExclusionsHist ns  on ns.CategoryID = nx.CategoryID  inner join REPORTING.dbo.All\_Products\_MainInfo mi  on ns.ProdNum = right(mi.RefNum,5)  inner join USER\_SUPPLY\_CHAIN.dbo.pm\_NA\_ProductListExclusionsCategory ce  on ns.CategoryID = ce.CategoryID  --and ce.Purpose = 3 --customer specific  inner join USER\_SUPPLY\_CHAIN.dbo.pm\_NA\_NetSellPurpose cep  ON ce.Purpose = cep.PurposeID  on l.orderCorpContractcd = nact.Contract  and nact.nationalacctrep not in ('','wjb') --not adopted by an NA rep so probably not a corp we are targeting/contracted with  and Sell\_Other1 is not null  AND L.Orderwhen >= nx.CreatedWhenUTC and L.Orderwhen < nx.RetiredWhenUTC  AND L.Orderwhen >= ns.CreatedWhenUTC and L.Orderwhen < ns.RetiredWhenUTC  and p.sourcecatprodnbr = mi.refnum  WHERE l.LineStatusCd <> 'I'  AND l.IsCancelledPreBillCreditFlag = 'N'  AND l.OrderChannelCd NOT IN ('A', 'C', 'M')  AND l.OrderSbuId <> 4 AND d.Date > dateadd(year, datediff(year, 0, getdate())-7, 0)  group by l.lineid,  d.Date,  ISNULL(l.LineUnitQty, lo.Qty),  ISNULL(l.BK\_ProductSales, lo.Qty\*lo.UnitSell)  ) as x  INNER JOIN dbo.FactLine l on x.lineid = l.lineid  INNER JOIN dbo.DimLocationCurrent f on f.LocationCurrentKey = l.ShipLocationCurrentKey  INNER JOIN dbo.DimContactCurrent c on c.ContactCurrentKey = l.ContactCurrentKey  INNER JOIN dbo.DimLineOfBusinessCurrent b on b.LobCurrentKey = l.LineOfBusinessCurrentKey |
| --- |

## fact\_login\_tels

Summary of TELS login events for last 7 calendar years plus current

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | Location | string | TELS login Location (ref dim\_location\_dse) |
| PK | ContactKey | string | Person/Customer who placed order (ref dim\_contact\_dse) |
| PK | Date | date | TELS login Date (ref dim\_date.Date) |
|  | LoginTELS | int | Count of TELS logins (any) |
|  | ~~LoginTELSWeb~~ | ~~int~~ | ~~Count of TELS Web logins~~ |
|  | ~~LoginTELSMobile~~ | ~~int~~ | ~~Count of TELS Mobile logins~~ |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Model Name: Login TELS
* Model Query:

| SELECT UPPER(RTRIM(f.FacilityPLT)) AS Location  ,UPPER(RTRIM(c.ContactKey)) AS ContactKey  ,cast(t.[When] AS DATE) AS [Date]  ,sum(LoginTELS) AS LoginTELS  FROM (  SELECT t.PersonID  ,t.LocationBusinessUnitID  ,t.[When]  ,1 AS LoginTELS  FROM CONTACT.dbo.telsSignInActivity t  ) AS T  INNER JOIN MDID\_BTRIEVE.dbo.clcont c ON c.personid = t.personid  INNER JOIN contact.dbo.telsFacility f ON f.BusinessUnitID = t.LocationBusinessUnitID  WHERE [when] > dateadd(year, datediff(year, 0, getdate()) - 7, 0)  AND EXISTS (  SELECT \*  FROM FinProfit\_DM.dbo.DimContactCurrent x  WHERE x.SourceContactKey = c.ContactKey  )  AND EXISTS (  SELECT \*  FROM FinProfit\_DM.dbo.DimLocationCurrent x  WHERE x.SourceLocationId = f.FacilityPLT  )  GROUP BY f.FacilityPLT  ,c.ContactKey  ,cast(t.[when] AS DATE) |
| --- |

* Model Steps:
  + RenameColumns(Source,{{"HadMobileActivity", "Login TELS Mobile"}, {"HadWebActivity", "Login TELS Web"}, {"LoginCount", "Login TELS"}, {"When", "Date"}})

## dim\_location\_dse

Represents a DSE location/account. Based on FinProfit DM

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | Location | string | Unique location code/identifier (ClLocn.Location) |
|  | LocStatusCd | string | Location Status (ClLocn.Status) |
| FK | CorpLocation | string | Corporate Location Code (ClLocn.CorpLocation, self ref) |
|  | OwnManageLease | string | Own/Manage/Lease Code (ClLocn.OwnManageLease) |
| FK | OwnerLocation | string | Owner Location Code (ClLocn.OwnerLocation, self ref) |
|  | DSELocName | string | Location Name (ClLocn.Name) |
|  | AddressLine1 | string | Location Address Line #1 (ClLocn.AddressLine1) |
|  | AddressLine2 | string | Location Address Line #2 (ClLocn.AddressLine2). Optional |
|  | City | string | Location City (ClLocn.City) |
|  | State | string | Location State (ClLocn.State) |
|  | Zip | string | Location Zip (ClLocn.Zip) |
|  | Country | string | Location Country (ClLocn.Country) |
|  | PhoneAreaCode | string | Phone Area Code (ClLocn.PhoneAreaCode) |
|  | Phone | string | Phone Number (ClLocn.Phone, e.g. 555-5555) |
|  | PhoneExt | string | Optional Phone Extension (ClLocn.PhoneExtension) |
|  | CreatedDate | timestamp | When Location was created. Null if unknown (ClLocn.CreateDate) |
|  | UpdatedDate | timestamp | When Location was last updated (ClLocn.DateUpdated) |
|  | LocTypeCd | string | Location Type Code (many–similar to MarketCd) |
|  | LocType | string | Location Type Name (many, similar to Market) |
|  | DivisionCd | string | Location Division Code (I,C) |
|  | Division | string | Location Division Name (Independent, Corporate) |
|  | MarketCd | string | The market code of the location. 'AL'='Assisted Living', 'SN'='Skilled Nursing', 'XN'='Special Markets'. This is Marketing's view of the market. It is a grouping of Location Type, which is assigned by the Customer Database team |
|  | Market | string | The market name of the location. Values include: 'Assisted Living', 'Skilled Nursing', 'Special Markets'. This is Marketing's view of the market. It is a grouping of Location Type, which is assigned by the Customer Database team |
|  | ForProfitCd | string | For Profit Code (F, N, C, blank) |
|  | ForProfit | string | For Profit Name (For Profit, Not For Profit, Non-Profit, Unknown) |
|  | CorpLocationNm | string | Corporation Location Name (via CorpLocation) |
|  | LocCorpContractCd | string | Corporate Contract Code |
|  | LocCorpContractNm | string | Corporate Contract Name |
|  | LocCorpContractPriceClass | string | Corporate Contract Price Class Code |
|  | LocCorpContractBaseClass | string | Corporate Contract Base Class (Level A, B, C D) |
|  | LocGPOContractCd | string | GPO Contract Code (optional) |
|  | LocGPOContractNm | string | GPO Contract Name (optional) |
|  | LocGPOContractPriceClass | string | GPO Contract Price Class Code (optional) |
|  | LocGPOContractBaseClass | string | GPO Contract Base Class (Level A, B, C D) (optional) |
| FK | DSLocationID | string | Calculated reference to combined Location (ref dim\_location\_ds) |
|  | AcuteCareBedCnt | int | Number of beds for Acute Care |
|  | AlzheimersBedCnt | int | Number of beds for Alzheimers |
|  | AssistedLivingBedCnt | int | Number of beds for Assisted Living |
|  | HospiceBedCnt | int | Number of beds for Hospice |
|  | IndependentLivingBedCnt | int | Number of beds for Independent Living |
|  | IntermediateCareBedCnt | int | Number of beds for Intermediate Care |
|  | LongTermCareSkilledNursingBedCnt | int | Number of beds for Long Term Care/Skilled Nursing |
|  | OtherBedCnt | int | Number of beds for Other purposes |
|  | SubacuteCareBedCnt | int | Number of beds for Subacute Care |
|  | TotalBedCnt | int | Number of all beds at Location |
|  | OwnerLocationNm | string | Owning Location Name |
|  | CreditStatusDesc | string | The location's Financial Credit Status |
|  | AccountManagerDivisionNm | string | The division name of the account manager currently assigned to the location by Sales Ops |
|  | AccountManagerTeamNm | string | The team name of the account manager currently assigned to the location by Sales Ops |
|  | AccountManagerNm | string | The name of the account manager currently assigned to the location by Sales Ops |
|  | CapitalSalesConsultantTeamNm | string | The team name of the capital sales consultant currently assigned to the location by Sales Ops |
|  | CapitalSalesConsultantNm | string | The capital sales consultant name currently assigned to the location by Sales Ops |
|  | NationalAccountManagerTeamNm | string | The team name of the national account manager currently assigned to the location by Sales Ops |
|  | NationalAccountManagerNm | string | The name of the national account manager currently assigned to the location by Sales Ops |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Model Name: Location DSE
* Model Query:

| SELECT UPPER(RTRIM(L.Location)) as Location,  L.Status,  UPPER(RTRIM(L.CorpLocation)) as CorpLocation,  L.OwnManageLease,  UPPER(RTRIM(L.OwnerLocation)) as OwnerLocation,  L.Name,  L.AddressLine1,  L.AddressLine2,  L.City,  L.State,  L.Zip,  L.Country,  L.PhoneAreaCode,  L.Phone,  L.PhoneExtension,  case when l.DateCreated between '1/1/1985' and getdate() then l.DateCreated end as DateCreated,  L.DateUpdated,  L2.TypeCd,  L2.TypeDesc,  L2.DivisionCd,  L2.DivisionNm,  L2.MarketCd,  L2.MarketNm,  L.ForProfit as ForProfitCd,  ISNULL(cfp.description, 'Unknown') as ForProfit,  L2.CorpLocationNm,  UPPER(RTRIM(L2.CorpContractCd)) as CorpContractCd, L2.CorpContractNm, L2.CorpContractContractedPriceClass, isnull(cpb.BaseClass, '{N/A}') as CorpContractBaseLevel,  UPPER(RTRIM(L2.GPOContractCd)) as GPOContractCd, L2.GPOContractNm, ISNULL(gc.PriceClass, '{N/A}') AS GPOContractPriceClass, ISNULL(gpb.BaseClass, '{N/A}') as GPOContractBaseLevel,  L2.AcuteCareBedCnt,  L2.AlzheimersBedCnt,  L2.AssistedLivingBedCnt,  L2.HospiceBedCnt,  L2.IndependentLivingBedCnt,  L2.IntermediateCareBedCnt,  L2.LongTermCareSkilledNursingBedCnt,  L2.OtherBedCnt,  L2.SubacuteCareBedCnt,  L2.TotalBedCnt,  L2.OwnerLocationNm,  L2.CreditStatusDesc,  L2.AccountManagerDivisionNm,  L2.AccountManagerTeamNm,  L2.AccountManagerNm,  L2.CapitalSalesConsultantTeamNm,  L2.CapitalSalesConsultantNm,  L2.NationalAccountManagerTeamNm,  L2.NationalAccountManagerNm  FROM MDID\_BTRIEVE.dbo.ClLocn L  INNER JOIN FinProfit\_DM.dbo.DimLocationCurrent as L2 on L2.SourceLocationId = l.Location  LEFT JOIN dshe\_codes.dbo.CodeTranslation cfp on cfp.CodeTypeID = 1363 and cfp.Code1 = l.ForProfit and cfp.Code1 <> ''  LEFT JOIN mdid\_btrieve.dbo.Clbasecl cpb on cpb.Class = L2.CorpContractContractedPriceClass  LEFT JOIN mdid\_btrieve.dbo.clcontyp gc on gc.Contract = L2.GPOContractCd  LEFT JOIN mdid\_btrieve.dbo.Clbasecl gpb on gpb.Class = gc.PriceClass  WHERE L.Location <> '' |
| --- |

* Model Steps:
  + #"Added Custom" = Table.AddColumn(Source, "DSLocationID", each [Location]),
  + #"Renamed Columns" = Table.RenameColumns(#"Added Custom",{{"AddressLine1", "Address Line 1"}, {"AddressLine2", "Address Line 2"}, {"CorpLocation", "Corp Location"}, {"DateCreated", "Created Date"}, {"DateUpdated", "Updated Date"}, {"OwnerLocation", "Owner Location"}, {"OwnManageLease", "Own Manage Lease"}, {"PhoneAreaCode", "Phone Area Code"}, {"PhoneExtension", "Phone Ext"}, {"Status", "Loc Status Cd"}, {"Name", "DSE Loc Name"}, {"MarketCd", "Market Cd"}, {"MarketNm", "Market"}, {"TypeCd", "Loc Type Cd"}, {"TypeDesc", "Loc Type"}, {"DivisionNm", "Division"}, {"DivisionCd", "Division Cd"}, {"ForProfitCd", "For Profit Cd"}, {"ForProfit", "For Profit"}, {"CorpContractBaseLevel", "Corp Ct Base Class"}, {"CorpContractCd", "Corp Ct Cd"}, {"CorpContractContractedPriceClass", "Corp Ct Price Class"}, {"CorpContractNm", "Corp Ct Name"}, {"CorpLocationNm", "Corp Name"}, {"GPOContractBaseLevel", "GPO Ct Base Class"}, {"GPOContractCd", "GPO Ct Cd"}, {"GPOContractNm", "GPO Ct Name"}, {"GPOContractPriceClass", "GPO Ct Price Class"}})

## dim\_contact\_dse

Represents a DSE customer contact. Based on FinProfit DM

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | ContactKey | string | Unique Contact Code/Identifier (ClCont.ContactKey) |
| FK | BaseLocation | string | Main location for Contact (ref dim\_location\_dse.Location) |
|  | IsSlug | string | Y/N indicator for a “Slug” contact. Typically “N” |
|  | PreName | string | Name Prefix/Title (e.g. Mr., Mrs., Dr.) |
|  | FirstName | string | Contact First Name (formal) |
|  | MiddleName | string | Contact Middle Name/Initial |
|  | LastName | string | Contact Last Name |
|  | Suffix | string | Contact Suffix (e.g. Jr., Sr., RN, MD) |
|  | Salutation | string | Contact First/NickName |
|  | JobCd | string | Contact Job/Occupation Code (~80 codes total) |
|  | JobCdDescription | string | Contact Job Description (for JobCode) |
|  | PhoneAccess | string | Phone Prefix/Country Code (typically 1) |
|  | PhoneAreaCode | string | Phone Area Code |
|  | Phone | string | Phone core number (7 digit no hyphen) |
|  | PhoneExt | string | Phone Extension |
|  | DateCreated | timestamp | Date and time record created |
|  | DateUpdated | timestamp | Date and time record last updated |
| SK | PersonID | int | Alternate identifier field for contact |
|  | EmailAddress | string | Contact email address (if known) |
| FK | DSContactID | string | Calculated reference to combined Contact (ref dim\_contact\_ds) |
|  | Status | string | Contact Status (Active/Inactive) |
|  | JobBucketCd | string | Job Bucket Code. Coalesce Job Code/Desc to one of 14 buckets |
|  | JobBucket | string | Job Bucket Name (e.g. Nursing, Maintenance, Administrator) |
|  | JobTitle | string | Contact Job Description (free form) |

* Source: SQL-DSHE-PI-DW, FinProfit\_DM
* Model Name: ContactDSE
* Model Query:

| SELECT UPPER(RTRIM(C.ContactKey)) AS ContactKey, UPPER(RTRIM(C.BaseLocation)) AS BaseLocation, C.IsSlug, C.PreName, C.FirstName, C.MidName, C.LastName, C.SufName, C.Salutation, C.JobCode, C.JobDescription as JobTitle, C.PhoneAccess, C.PhoneAreaCode, C.Phone, C.PhoneExtension,  C.DateCreated, C.DateUpdated, C.PersonID, ISNULL(E.EmailAddress, '') AS EmailAddress,  'D'+UPPER(RTRIM(COALESCE(NULLIF(R2.ContactKey, ''), NULLIF(R1.ContactKey, ''), C.ContactKey))) AS DSContactID  ,C2.StatusDesc, c2.JobBucketCd, c2.JobBucketDesc as JobBucket, c2.JobCdDesc as JobDescription  FROM MDID\_BTRIEVE.dbo.ClCont C  INNER JOIN FinProfit\_DM.dbo.DimContactCurrent C2 on c2.SourceContactKey = c.ContactKey  LEFT JOIN MDID\_BTRIEVE.dbo.ClCont R1 ON R1.ContactKey = C.DupPointTo AND C.DupPointTo > ''  LEFT JOIN MDID\_BTRIEVE.dbo.ClCont R2 ON R2.ContactKey = R1.DupPointTo AND R1.DupPointTo > ''  LEFT JOIN MDID\_BTRIEVE.dbo.ClEmail E ON E.Location = '' and E.ContactKey = C.ContactKey AND E.EmailCode = 'CE' |
| --- |

* Model Steps:
  + RenameColumns(Source,{{"BaseLocation", "Base Location"}, {"DateCreated", "Date Created"}, {"EmailAddress", "Email Address"}, {"FirstName", "First Name"}, {"JobCode", "Job Code"}, {"JobDescription", "Job Description"}, {"DateUpdated", "Date Updated"}, {"LastName", "Last Name"}, {"PhoneAccess", "Phone Access"}, {"PhoneAreaCode", "Phone Area Code"}, {"PhoneExtension", "Phone Ext"}, {"PreName", "Pre Name"}, {"SufName", "Suffix"}, {"IsSlug", "Is Slug"}, {"StatusDesc", "Status"}, {"JobBucketCd", "Job Bucket Cd"}, {"JobBucketDesc", "Job Bucket"}})

## dim\_date

Direct Supply date table with business days

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | Date | date | Date |
|  | BusDay#ofMo | tinyint | Business Day of Month (bus days on/before date in current month) |
|  | BusDay#ofQtr | tinyint | Business Day of Quarter (bus days on/before date in current quarter) |
|  | BusDay#ofYear | smallint | Business Day of Year (bus days on/before date in current year) |
|  | IsBOM | string | Is date the beginning of month |
|  | IsEOM | string | Is date the end of the month |
|  | IsHoliday | string | Is date a holiday |
|  | IsBudgetException | string | Is date excluded from budget (sales) |
|  | BegOfMonthDate | date | Date of the beginning of the current month (e.g. 1/1, 2/1) |
|  | EndOfMonthDate | date | Date of the end of the current month (e.g. 1/31, 2/28) |
|  | BusBegOfMonthDate | date | Date of the first business day in current month |
|  | BusEndOfMonthDate | date | Date of the last business day in current month |
|  | Day#ofWeek | tinyint | Day of week (Sunday = 1, etc.) |
|  | Day#ofMo | tinyint | Month day |
|  | Day#ofYear | smallint | Day of year (days on/before date in current year) |
|  | Week# | tinyint | Calendar week number of year (partial first week common) |
|  | Month# | tinyint | Month number of year (1-12) |
|  | Quarter# | tinyint | Quarter number of year (1-4) |
|  | Year | smallint | Year Number |
|  | Weekday | string | Day of Week |
|  | Month | string | Month Name |
|  | IsBusinessDay | string | Is date a business day for Direct Supply |

* Source: SQL-DSSI-PE-DM, BI
* Model Name: Date
* Model Query:

| select  DateKey  ,BusinessDayOfMonthNbr  ,BusinessDayOfQuarterNbr  ,BusinessDayOfYearNbr  ,IsStartOfMonth  ,IsEndOfMonth  ,IsHoliday  ,IsBudgetException  ,StartOfMonthDate  ,EndOfMonthDate  ,BusinessStartOfMonthDate  ,BusinessEndOfMonthDate  ,DayOfWeekNbr  ,DayOfMonthNbr  ,DayOfYearNbr  ,WeekOfYearNbr  ,MonthOfYearNbr  ,QuarterOfYearNbr  ,Year  ,DayName  ,MonthName  , IsBusinessDay = CASE WHEN IsHoliday = 'N' AND DayOfWeekNbr BETWEEN 2 and 6 THEN 'Y' ELSE 'N' END  FROM dbo.DimDate |
| --- |

* Model Steps:
  + Table.RenameColumns(dbo\_DimDate,{{"DateKey", "Date"}, {"BusinessDayOfMonthNbr", "Bus Day# of Mo"}, {"BusinessDayOfQuarterNbr", "Bus Day# of Qtr"}, {"BusinessDayOfYearNbr", "Bus Day# of Year"}, {"DayName", "Weekday"}, {"BusinessEndOfMonthDate", "Bus EOM Date"}, {"BusinessStartOfMonthDate", "Bus BOM Date"}, {"DayOfMonthNbr", "Day# of Mo"}, {"DayOfWeekNbr", "Day# of Week"}, {"DayOfYearNbr", "Day# of Year"}, {"EndOfMonthDate", "EOM Date"}, {"IsBudgetException", "Is Budget Exception"}, {"IsEndOfMonth", "Is EOM"}, {"IsHoliday", "Is Holiday"}, {"IsStartOfMonth", "Is BOM"}, {"MonthName", "Month"}, {"MonthOfYearNbr", "Month#"}, {"QuarterOfYearNbr", "Quarter#"}, {"StartOfMonthDate", "BOM Date"}, {"WeekOfYearNbr", "Week#"}})

## dim\_contact\_dssi

Represents a DSSI login/customer contact. Based on DSSI BI datamart

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| SK | CustCode | string | DSSI Provider Code (ref dim\_location\_dssi) |
| FK | CustLocNum | string | DSSI Customer’s Location Number (ref dim\_location\_dssi) |
| PK | CustContNum | int | Unique integer identifier for Contact |
| SK | Username | string | Unique username within Provider |
|  | FirstName | string | Person’s first name |
|  | LastName | string | Person’s last name |
| FK | DSSIDSEContactKey | string | Optional reference to DSE Contact (ref dim\_contact\_dse.ContactKey) |
|  | JobCd | string | DSSI Job Code |
|  | Active | string | Login enabled (Y/N) |
|  | TestFlag | string | Test/Production status code - test accounts cannot order (T/P) |
|  | EmailAddress | string | Person’s email address |
|  | Created | timestamp | Date and time record was created |
|  | CreatedBy | string | DS employee’s login for who created |
|  | ContactGrp | string | Calculated code to define contact groups. Accounts for DSSI master contacts. |
| FK | DSContactID | string | Calculated code: use if valid DSSIDSEContactKey, else CustContNum (ref dim\_contact\_ds.DSContactID) |

* Source: SQL-DSSI-PE-DM, BI
* Model Name: Contact DSSI
* Model Query:

| SELECT UPPER(RTRIM(C.CustCode)) AS CustCode, UPPER(RTRIM(C.CustLocNum)) AS CustLocNum  , C.CustContNum, RTRIM(C.Username) AS Username, RTRIM(ISNULL(M.FirstName, C.Firstname)) as FirstName  , RTRIM(ISNULL(M.Lastname, C.LastName)) as LastName, UPPER(RTRIM(C.DSIContId)) as DSIContId  , UPPER(RTRIM(C.JobCode)) AS JobCode, UPPER(RTRIM(C.Active)) As Active  , UPPER(RTRIM(C.TestFlag)) AS TestFlag  , RTRIM(ISNULL(M.EmailAddress, C.EmailAddress)) as EmailAddress, C.Created, RTRIM(C.CreatedBy) AS CreatedBy  , UPPER(RTRIM(ISNULL('M'+RTRIM(M.MasteruserID), 'C'+RTRIM(C.CustContNum)))) AS ContactGrp  FROM CONTACT.dbo.InCont C  LEFT JOIN CONTACT.dbo.InContMasterLink R ON R.CustContNum = C.CustContNum  LEFT JOIN CONTACT.dbo.InContMaster M ON M.MasterUserID = R.MasterUserID |
| --- |

* Model Steps:
  + #"Merged Queries" = Table.NestedJoin(Source, {"DSIContId"}, Table.SelectColumns(#"Contact DSE",{"ContactKey", "DSContactID"}), {"ContactKey"}, "DSE Contact", JoinKind.LeftOuter),
  + #"Expanded DSE Contact" = Table.ExpandTableColumn(#"Merged Queries", "DSE Contact", {"DSContactID"}, {"DSEContactID"}),
  + #"Added Custom" = Table.AddColumn(#"Expanded DSE Contact", "DSContactID", each if [DSEContactID] is null then [ContactGrp] else [DSEContactID]),
  + #"Renamed Columns" = Table.RenameColumns(#"Added Custom",{{"ContactGrp", "Contact Grp"}, {"CreatedBy", "Created By"}, {"EmailAddress", "Email Address"}, {"FirstName", "First Name"}, {"JobCode", "Job Code"}, {"LastName", "Last Name"}, {"TestFlag", "Test Flag"}, {"DSIContId", "DSSI DSE ContactKey"}, {"Active", "Active"}})

## dim\_location\_dssi

Represents a DSSI facility. Based on DSSI BI datamart

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| SK | CustCode | string | DSSI Provider Code |
| SK | CustLocNum | string | DSSI Customer’s Location Number/Code |
| PK | FacilityKey | int | DSSI Location Identifier |
|  | Status | string | Location Active Status/Type (A, I, S, T) |
| FK | DSSIDSELocation | string | Optional reference to DSE Location (ref dim\_location\_dse.Location) |
|  | DSSILocName | string | Location Name |
|  | AddressLine1 | string | Location Address Line 1 |
|  | AddressLine2 | string | Location Address Line 2 |
|  | City | string | Location City |
|  | State | ‘string | Location State |
|  | Zip | string | Location Zip |
|  | PhoneAccess | string | Phone Prefix/Country Code (typically 1) |
|  | PhoneAreaCode | string | Phone Area Code |
|  | Phone | string | Phone Number (7 digit no dashes) |
|  | PhoneExt | string | Optional Phone Extension |
|  | KeepActive | string | Y/N indicator, once used for DSE link |
|  | LiveDate | timestamp | Date location can begin ordering (go live) |
|  | LocTestFlag | string | Test flag code for prod/test (P, T) |
|  | LocType | tinyint | Code for position in location hierarchy (3=corp, 2=division,1=region,0=fac) |
|  | CreatedDate | timestamp | When record was created |
| FK | DSLocationID | string | Calculated code: use DSSIDSELocation if valid, else CustCode.CustLocNum (ref dim\_location\_ds) |
|  | TotalBedCnt | smallint | Number of all beds at Location |

* Source: SQL-DSSI-PE-DM, BI
* Model Name: Location DSSI
* Model Query:

| SELECT UPPER(RTRIM(l.CustCode)) AS CustCode, UPPER(RTRIM(l.CustLocNum)) AS CustLocNum, f.FacilityKey, UPPER(RTRIM(l.Status)) AS Status, UPPER(RTRIM(NULLIF(l.DSILocNum, ''))) AS DSILocNum, l.Name, l.AddressLine1, l.AddressLine2, l.City, l.State, l.Zip, l.PhoneAccess, l.PhoneAreaCode, l.Phone, l.PhoneExt, UPPER(RTRIM(l.KeepActive)) AS KeepActive, l.LiveDate, UPPER(RTRIM(l.LocnTestFlag)) AS LocnTestFlag, l.LocnType AS LocnType, case when l.Created between '1/1/1985' and getdate() then l.Created end as CreatedWhen, f.BedSize As TotalBedCnt  FROM CONTACT.dbo.InLocn l  INNER JOIN BI.dbo.DimFacility F ON F.CustCode = l.CustCode and f.CustLocNum = l.CustLocNum |
| --- |

* Model Steps:
  + #"Added DSSILocationID" = Table.AddColumn(Source, "DSSILocationID", each Text.Combine({[CustCode],[CustLocNum]}, ".")),
  + #"Merged Queries" = Table.NestedJoin(#"Added DSSILocationID", {"DSILocNum"}, #"Location DSE", {"Location"}, "DSELocation", JoinKind.LeftOuter),
  + #"Expanded DSELocation" = Table.ExpandTableColumn(#"Merged Queries", "DSELocation", {"Location"}, {"DSELocationID"}),
  + #"Added Custom" = Table.AddColumn(#"Expanded DSELocation", "DSLocationID", each if [DSELocationID] = null then [DSSILocationID] else [DSELocationID]),
  + #"Renamed Columns" = Table.RenameColumns(#"Added Custom",{{"AddressLine1", "Address Line 1"}, {"AddressLine2", "Address Line 2"}, {"DSILocNum", "DSSI DSE Location"}, {"KeepActive", "Keep Active"}, {"LiveDate", "Live Date"}, {"LocnTestFlag", "Loc Test Flag"}, {"LocnType", "Loc Type"}, {"Name", "DSSI Loc Name"}, {"PhoneAccess", "Phone Access"}, {"PhoneAreaCode", "Phone Area Code"}, {"PhoneExt", "Phone Ext"}, {"CreatedWhen", "Created Date"}})
* Note: FacilityKey column not yet in Power BI Model

## fact\_order\_dssi

Transaction line detail for DSSI Purchase Orders. Contains last 7 calendar years plus current. Based on DSSI BI datamart

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
|  | CustCode | string | DSSI Provider Code (ref dim\_location\_dssi) |
|  | CustLocNum | string | DSSI Customer’s Location Number/Code (ref dim\_location\_dssi) |
| FK | FacilityKey | int | DSSI Facility Key (ref dim\_location\_dssi) |
| FK | CustContNum | int | DSSI Contact Identifier (ref dim\_contact\_dssi.CustContNum) |
| FK | Date | date | Order Date (ref dim\_date.Date) |
|  | POCancelled | bool | Canceled indicator |
|  | PONbr | string | DSSI Purchase Order Number (with credit logic) |
| PK | LinePoNbr | string | DSSI Source system key/Orig PO Number (chlines.chponum) |
| PK | LineNbr | smallint | DSSI Source system key/Line Number (chlines.AccLineNum) |
|  | LineQty | smallint | Quantity Ordered |
|  | LineUnitSell | decimal(19,4) | Sell price of item |
|  | LineAmount | decimal(19,4) | Total Line Amount (Qty\*UnitSell) plus any line charges |
|  | LineUMCd | string | Unit of Measure ordered (CA for Case, EA for Each, etc.) |
|  | ProductNbr | string | Supplier’s Product Number for item |
|  | ProductDesc | string | Product/Item Description |
|  | SupplierCd | string | DSSI’s Supplier (Distribution center) Code |
|  | SupplierName | string | DSSI’s Supplier (Distribution center) Name |
|  | ParentSupplierCd | string | DSSI Corp Supplier Code |
|  | ParentSupplierName | string | DSSI Corp Supplier Name |

* Source: SQL-DSSI-PE-DM, BI
* Query:

| SELECT UPPER(RTRIM(f.CustCode)) AS CustCode  ,UPPER(RTRIM(f.CustLocNum)) AS CustLocNum  ,f.FacilityKey  ,c.CustContNum  ,l.PODateKey AS DATE --key fields  ,cast(CASE  WHEN ls.Code in ('O', 'I') AND ps.StatusCode in ('F', 'O', 'P')  THEN 0  ELSE 1  END AS BIT) AS POCancelled  ,UPPER(RTRIM(l.ChPONum)) AS PONbr  ,UPPER(RTRIM(l.LineChPoNum)) AS LinePoNbr  ,l.AccLineNum AS LineNbr  ,l.Quantity AS LineQty  ,l.UnitAmount AS LineUnitSell  ,l.ExtendedAmount AS LineAmount  ,UPPER(RTRIM(u.UMCode)) AS LineUMCd  ,p.SuppProdNum AS ProductNbr  ,isnull(nullif(l.ProductDesc, ''), p.ProductDesc) AS ProductDesc  ,UPPER(RTRIM(p.ChSupplier)) AS SupplierCd  ,p.SupplierName AS SupplierName  ,UPPER(RTRIM(p.ParentChSupplier)) AS ParentSupplierCd  ,p.ParentSupplierName AS ParentSupplierName  FROM dbo.FactPOLine l  INNER JOIN dbo.DimPOLineStatus ls ON ls.POLineStatusKey = l.POLineStatusKey  INNER JOIN dbo.DimFacility f ON f.FacilityKey = l.FacilityKey  INNER JOIN dbo.DimContact c ON c.ContactKey = l.OrderedByKey  INNER JOIN dbo.DimCustomerProduct p ON p.CustomerProductKey = l.CustomerProductKey  INNER JOIN dbo.DimUnitMeasure u ON u.UnitMeasureKey = l.UnitMeasureKey  INNER JOIN dbo.DimPOStatus ps on ps.POStatusKey = l.POStatusKey  WHERE l.PODateKey > dateadd(year, datediff(year, 0, getdate()) - 7, 0) |
| --- |

## fact\_activity\_dssmart

Summary of DS Smart Activity

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | Location | string | DSE Location Code (ref dim\_location\_dse.Location) |
| PK | Date | string | Event Date (ref dim\_date.Date) |
|  | LoginsDSSmart | bigint | Count of DSSmart Login events |
|  | VitalsDSSmart | bigint | Count of Vital Submissions taken |

* Source: dssmart-reporting-mart.cfjeeb1zlhf6.us-east-1.redshift.amazonaws.com, ds\_smart\_prod
* Model Name: Activity DS Smart
* Model Query:

| select upper(rtrim(l.sourcelocationid\_plt)) as Location, d.date as Date, sum(case when w.activityname = 'Login' then 1 else 0 end) as Logins, sum(case when w.activityname = 'VitalsSubmission' then 1 else 0 end) as Vitals  from public.fact\_usage u  inner join public.dim\_customer l on l.dimcustomersk = u.dimcustomersk  inner join public.dim\_date d on d.dimdatesk = u.dimusagedatesk  inner join public.dim\_workflow w on w.dimworkflowsk = u.dimworkflowsk  group by upper(rtrim(l.sourcelocationid\_plt)), d.date |
| --- |

* Model Steps:
  + RenameColumns(Source,{{"logins", "Logins DS Smart"}, {"date", "Date"}, {"location", "Location"}, {"vitals", "Vitals DS Smart"}})

## dim\_location\_ds

Stitched customer location/account based on existing DSE and DSSI mappings

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | DSLocationID | string | Identifier for merged Location |
|  | LocName | string | Location Name |
|  | AddressLine1 | string | Location Address Line 1 |
|  | AddressLine2 | string | Location Address Line 2 |
|  | City | string | Location City |
|  | State | string | Location State |
|  | Zip | string | Location Zip |
|  | PhoneAreaCode | string | Phone Area Code |
|  | Phone | string | Phone Number (7 digit no dashes) |
|  | PhoneExt | string | Optional Phone Extension |

* Source: Derived from prior output
* Model Name: Location DS
* Model Steps:
  + let
  + Source = #"Location DSSI",
  + #"Filtered Rows" = Table.SelectRows(Source, each [DSELocationID] = null),
  + #"Removed Columns" = Table.SelectColumns(#"Filtered Rows", {"DSSI Loc Name", "Address Line 1", "Address Line 2", "City", "State", "Zip", "Phone Area Code", "Phone", "Phone Ext", "DSLocationID"}),
  + #"Appended Query" = Table.Combine({Table.RenameColumns(#"Removed Columns", {{"DSSI Loc Name", "Loc Name"}}), Table.RenameColumns(#"Location DSE", {{"DSE Loc Name", "Loc Name"}})}, {"Loc Name", "Address Line 1", "Address Line 2", "City", "State", "Zip", "Phone Area Code", "Phone", "Phone Ext", "DSLocationID"}),
  + ~~#"Added Index" = Table.AddIndexColumn(#"Appended Query", "Location#", 1, 1, Int32.Type)~~
  + in
  + #"Added Index"
* Note: Location# column not required in output

## dim\_contact\_ds

Stitched customer contact based on existing DSE and DSSI mappings. Note: DSSI to DSE mapping takes precedence over DSSI master contact mapping

|  | **Column** | **Type** | **Desc** |
| --- | --- | --- | --- |
| PK | DSContactID | string | Identifier for merged Location |
|  | FirstName | string | Contact First Name |
|  | LastName | string | Contact Last Name |
|  | EmailAddress | string | Contact Email Address |

* Source: Derived from prior output
* Model Name: Contact DS
* Model Steps:
  + let
  + Source = Table.SelectColumns(#"Contact DSSI",{"First Name", "Last Name", "Email Address", "DSContactID", "DSEContactID"}),
  + #"Filtered Rows" = Table.SelectRows(Source, each [DSEContactID] = null),
  + #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"First Name", "Last Name", "Email Address", "DSContactID"}),
  + #"Removed Duplicates" = Table.Distinct(#"Removed Other Columns"),
  + #"Appended Query" = Table.Combine({#"Removed Duplicates", Table.SelectColumns(Table.SelectRows(#"Contact DSE", each [DSContactID] = Text.Combine({"D",[ContactKey]})),{"First Name", "Last Name", "Email Address", "DSContactID"})}),
  + ~~#"Added Index" = Table.AddIndexColumn(#"Appended Query", "Contact#", 1, 1, Int32.Type)~~
  + in
  + #"Added Index"
* Note: Contact# column not required in output