**Issue:** When importing Sales Price Agreement (PriceDiscTable) data from excel below error was coming:

“Insert not allowed for PriceDiscTable.InventDimId”.

**Resolution:** We have extended “postTargetProcess” method of “SalesPriceAgreementEntity” entity and written code to update/insert of InventDimId in PriceDiscTable table as below:

[ExtensionOf(tableStr(SalesPriceAgreementEntity))]

final public class SalesPriceAgreement\_Extension

{

public static void postTargetProcess(DMFDefinitionGroupExecution \_dmfDefinitionGroupExecution)

{

*//do stuff*

if (\_dmfDefinitionGroupExecution.StagingStatus == DMFBatchJobStatus::Finished)

{

SalesPriceAgreementStaging \_SalesPriceAgreementStaging;

*//select all staging records that were processed by current execution job without errors.*

while select \_SalesPriceAgreementStaging

where \_SalesPriceAgreementStaging.DefinitionGroup == \_dmfDefinitionGroupExecution.DefinitionGroup

&& \_SalesPriceAgreementStaging.ExecutionId == \_dmfDefinitionGroupExecution.ExecutionId

&& \_SalesPriceAgreementStaging.TransferStatus == DMFTransferStatus::Completed

{

str inventlocaitonid = \_SalesPriceAgreementStaging.PriceWarehouseId;

str inventsite = \_SalesPriceAgreementStaging.PriceSiteId;

date dtfromdate = \_SalesPriceAgreementStaging.PriceApplicableFromDate;

date dttodate = \_SalesPriceAgreementStaging.PriceApplicableToDate;

str 20 AccountRelation = "";

str 20 ItemNumber = "";

InventDim inventDimConfig;

inventDimConfig.InventLocationId = inventlocaitonid;

inventDimConfig.InventSiteId = inventsite;

inventDimConfig = InventDim::findOrCreate(inventDimConfig);

str inventdimid = inventDimConfig.inventDimId;

PriceDiscTable PriceDiscTable\_updt;

PriceDiscProductCodeType \_ItemCode;

PriceDiscPartyCodeType \_AccountCode;

\_ItemCode = PriceDiscProductCodeType::All;

if(\_SalesPriceAgreementStaging.ItemNumber)

{

\_ItemCode = PriceDiscProductCodeType::Table;

ItemNumber = \_SalesPriceAgreementStaging.ItemNumber;

}

\_AccountCode = PriceDiscPartyCodeType::All;

if(\_SalesPriceAgreementStaging.CustomerAccountNumber)

{

\_AccountCode = PriceDiscPartyCodeType::Table;

AccountRelation = \_SalesPriceAgreementStaging.CustomerAccountNumber;

}

else if(\_SalesPriceAgreementStaging.PriceCustomerGroupCode)

{

\_AccountCode = PriceDiscPartyCodeType::GroupId;

AccountRelation = \_SalesPriceAgreementStaging.PriceCustomerGroupCode;

}

select forupdate firstonly \* from PriceDiscTable\_updt

order by RecId asc

where

PriceDiscTable\_updt.relation == PriceType::PriceSales

&& PriceDiscTable\_updt.ItemCode == \_ItemCode

&& PriceDiscTable\_updt.ItemRelation == ItemNumber

&& PriceDiscTable\_updt.AccountCode == \_AccountCode

&& PriceDiscTable\_updt.AccountRelation == AccountRelation

&& PriceDiscTable\_updt.Currency == \_SalesPriceAgreementStaging.PriceCurrencyCode

&& PriceDiscTable\_updt.InventDimId == ""

&& PriceDiscTable\_updt.UnitId == \_SalesPriceAgreementStaging.QuantityUnitySymbol

&& PriceDiscTable\_updt.QuantityAmountFrom == \_SalesPriceAgreementStaging.FromQuantity

&& PriceDiscTable\_updt.FromDate == dtfromdate;

if(PriceDiscTable\_updt.RecId != 0)

{

ttsbegin;

PriceDiscTable\_updt.InventDimId = inventdimid;

PriceDiscTable\_updt.update();

ttscommit;

}

}

}

}

}