```
USE MaxiDev;


GO

/*

CREATE TABLE [dbo].[Transfer] ( IdTransfer int NOT NULL PRIMARY KEY, IdCustomer int NOT NULL,
IdBeneficiary int NOT NULL, IdPaymentType int NOT NULL, IdBranch int NULL, IdPayer int NOT NULL,
IdGateway int NULL, GatewayBranchCode nvarchar(-1) NOT NULL, IdAgentPaymentSchema int NOT
NULL, IdAgent int NOT NULL, IdAgentSchema int NULL, IdCountryCurrency int NOT NULL, IdStatus int
NOT NULL, ClaimCode nvarchar(50) NOT NULL, ConfirmationCode nvarchar(50) NOT NULL,
AmountInDollars money NOT NULL, Fee money NOT NULL, AgentCommission money NOT NULL,
CorporateCommission money NOT NULL, DateOfTransfer datetime NOT NULL, ExRate money NOT NULL,
ReferenceExRate money NOT NULL, AmountInMN money NOT NULL, Folio int NOT NULL,
DepositAccountNumber nvarchar(-1) NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser
int NOT NULL, TotalAmountToCorporate money NOT NULL, BeneficiaryName nvarchar(-1) NOT NULL,
BeneficiaryFirstLastName nvarchar(-1) NOT NULL, BeneficiarySecondLastName nvarchar(-1) NOT NULL,
BeneficiaryAddress nvarchar(-1) NOT NULL, BeneficiaryCity nvarchar(-1) NOT NULL, BeneficiaryState
nvarchar(-1) NOT NULL, BeneficiaryCountry nvarchar(-1) NOT NULL, BeneficiaryZipcode nvarchar(-1) NOT
NULL, BeneficiaryPhoneNumber nvarchar(-1) NOT NULL, BeneficiaryCelularNumber nvarchar(-1) NOT
NULL, BeneficiarySSNumber nvarchar(-1) NULL, BeneficiaryBornDate datetime NULL,
BeneficiaryOccupation nvarchar(-1) NULL, BeneficiaryNote nvarchar(-1) NOT NULL, CustomerName
nvarchar(-1) NOT NULL, CustomerIdAgentCreatedBy int NOT NULL,
CustomerIdCustomerIdentificationType int NULL, CustomerFirstLastName nvarchar(-1) NOT NULL,
CustomerSecondLastName nvarchar(-1) NOT NULL, CustomerAddress nvarchar(-1) NOT NULL,
CustomerCity nvarchar(-1) NOT NULL, CustomerState nvarchar(-1) NOT NULL, CustomerCountry
nvarchar(-1) NOT NULL, CustomerZipcode nvarchar(-1) NOT NULL, CustomerPhoneNumber nvarchar(-1)
NOT NULL, CustomerCelullarNumber nvarchar(-1) NOT NULL, CustomerSSNumber nvarchar(-1) NOT
NULL, CustomerBornDate datetime NULL, CustomerOccupation nvarchar(-1) NOT NULL,
CustomerIdentificationNumber nvarchar(-1) NOT NULL, CustomerExpirationIdentification datetime
NULL, IdOnWhoseBehalf int NULL, Purpose nvarchar(200) NOT NULL, Relationship nvarchar(200) NOT
NULL, MoneySource nvarchar(200) NOT NULL, AgentCommissionExtra money NOT NULL,
AgentCommissionOriginal money NOT NULL, ModifierCommissionSlider money NOT NULL,
ModifierExchangeRateSlider money NOT NULL, CustomerIdCarrier int NULL, IdSeller int NOT NULL,
ReviewDenyList bit NULL, ReviewOfac bit NULL, ReviewKYC bit NULL, ReviewRejected bit NULL,
ReviewGateway bit NULL, ReviewReturned bit NULL, OriginExRate money NOT NULL, OriginAmountInMN
money NOT NULL, DateStatusChange datetime NULL, NoteAdditional nvarchar(-1) NULL,
CustomerIdentificationIdCountry int NULL, CustomerIdentificationIdState int NULL, IdReasonForCancel
int NULL, IdBeneficiaryIdentificationType int NULL, BeneficiaryIdentificationNumber nvarchar(-1) NULL,
AgentNotificationSent bit NOT NULL, EmailByJobSent bit NOT NULL, FromStandByToKYC bit NULL,
CustomerIdCountryOfBirth int NULL, BeneficiaryIdCountryOfBirth int NULL, AccountTypeId int NULL,
ReviewId bit NULL, CustomerOccupationDetail nvarchar(-1) NULL, TransferIdCity int NULL,
BeneficiaryIdCarrier int NULL, FeeSecondary money NULL, NumModify int NULL, CustomerIdOccupation
int NULL, CustomerIdSubOccupation int NULL, CustomerSubOccupationOther varchar(50) NULL,
```

BranchCodePontual varchar(10) NULL, IdTransferResend int NULL, StateTax money NULL, IdPaymentMethod int NOT NULL, Discount money NOT NULL, DateOfTransferUTC datetime NULL, OperationFee money NOT NULL, IsValidCustomerPhoneNumber bit NULL, IdDialingCodePhoneNumber int NULL, IdDialingCodeBeneficiaryPhoneNumber int NULL, IsRequiredCustomerPhoneNumber bit NULL, IsRefunded bit NULL, ReviewCancelled bit NULL) , FOREIGN KEY (IdBranch) REFERENCES Branch(IdBranch), FOREIGN KEY (IdCountryCurrency) REFERENCES CountryCurrency(IdCountryCurrency), FOREIGN KEY (IdCustomer) REFERENCES Customer(IdCustomer), FOREIGN KEY (IdGateway) REFERENCES Gateway(IdGateway), FOREIGN KEY (CustomerIdentificationIdCountry) REFERENCES Country(IdCountry), FOREIGN KEY (IdPayer) REFERENCES Payer(IdPayer), FOREIGN KEY (CustomerIdentificationIdState) REFERENCES State(IdState), FOREIGN KEY (IdPaymentType) REFERENCES PaymentType(IdPaymentType), FOREIGN KEY (IdStatus) REFERENCES Status(IdStatus), FOREIGN KEY (AccountTypeId) REFERENCES AccountType(AccountTypeId), FOREIGN KEY (IdPaymentMethod) REFERENCES PaymentMethod(IdPaymentMethod), FOREIGN KEY (IdOnWhoseBehalf) REFERENCES OnWhoseBehalf(IdOnWhoseBehalf), FOREIGN KEY (TransferIdCity) REFERENCES City(IdCity), FOREIGN KEY (IdReasonForCancel) REFERENCES ReasonForCancel(IdReasonForCancel), FOREIGN KEY (IdDialingCodeBeneficiaryPhoneNumber) REFERENCES DialingCodePhoneNumber(IdDialingCodePhoneNumber), FOREIGN KEY (IdDialingCodePhoneNumber) REFERENCES DialingCodePhoneNumber(IdDialingCodePhoneNumber), FOREIGN KEY (CustomerIdCountryOfBirth) REFERENCES CountryBirth(IdCountryBirth), FOREIGN KEY (IdBeneficiaryIdentificationType) REFERENCES BeneficiaryIdentificationType(IdBeneficiaryIdentificationType), FOREIGN KEY (BeneficiaryIdCountryOfBirth) REFERENCES CountryBirth(IdCountryBirth), FOREIGN KEY (IdAgent) REFERENCES Agent(IdAgent), FOREIGN KEY (IdBeneficiary) REFERENCES Beneficiary(IdBeneficiary)

CREATE TABLE [dbo].[TransferClosed] (IdTransferClosed int NOT NULL PRIMARY KEY, IdCustomer int NOT NULL, IdBeneficiary int NOT NULL, IdPaymentType int NOT NULL, PaymentTypeName nvarchar(-1) NOT NULL, IdBranch int NULL, BranchName nvarchar(-1) NULL, IdPayer int NOT NULL, PayerName nvarchar(-1) NOT NULL, IdGateway int NULL, GatewayName nvarchar(-1) NULL, GatewayBranchCode nvarchar(-1) NULL, IdAgentPaymentSchema int NOT NULL, AgentPaymentSchema nvarchar(-1) NOT NULL, IdAgent int NOT NULL, AgentName nvarchar(-1) NOT NULL, IdAgentSchema int NULL, SchemaName nvarchar(-1) NULL, IdCountryCurrency int NOT NULL, IdCountry int NOT NULL, CountryName nvarchar(-1) NOT NULL, IdCurrency int NOT NULL, CurrencyName nvarchar(-1) NOT NULL, IdStatus int NOT NULL, StatusName nvarchar(-1) NOT NULL, ClaimCode nvarchar(50) NOT NULL, ConfirmationCode nvarchar(50) NOT NULL, AmountInDollars money NOT NULL, Fee money NOT NULL, AgentCommission money NOT NULL, CorporateCommission money NOT NULL, DateOfTransfer datetime NOT NULL, ExRate money NOT NULL, ReferenceExRate money NOT NULL, AmountInMN money NOT NULL, Folio int NOT NULL, DepositAccountNumber nvarchar(-1) NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser int NOT NULL, TotalAmountToCorporate money NOT NULL, BeneficiaryName nvarchar(-1) NOT NULL, BeneficiaryFirstLastName nvarchar(-1) NOT NULL, BeneficiarySecondLastName nvarchar(-1) NOT NULL, BeneficiaryAddress nvarchar(-1) NOT NULL, BeneficiaryCity nvarchar(-1) NOT NULL, BeneficiaryState nvarchar(-1) NOT NULL, BeneficiaryCountry nvarchar(-1) NOT NULL, BeneficiaryZipcode nvarchar(-1) NOT NULL, BeneficiaryPhoneNumber nvarchar(-1) NOT NULL, BeneficiaryCelularNumber nvarchar(-1) NOT NULL, BeneficiarySSNumber nvarchar(-1) NULL, BeneficiaryBornDate datetime NULL, BeneficiaryOccupation nvarchar(-1) NULL, BeneficiaryNote nvarchar(-1) NOT NULL, CustomerName nvarchar(-1) NOT NULL, CustomerIdAgentCreatedBy int NOT NULL,

```
CustomerIdCustomerIdentificationType int NULL, CustomerFirstLastName nvarchar(-1) NOT NULL,
CustomerSecondLastName nvarchar(-1) NOT NULL, CustomerAddress nvarchar(-1) NOT NULL,
CustomerCity nvarchar(-1) NOT NULL, CustomerState nvarchar(-1) NOT NULL, CustomerCountry
nvarchar(-1) NOT NULL, CustomerZipcode nvarchar(-1) NOT NULL, CustomerPhoneNumber nvarchar(-1)
NOT NULL, CustomerCelullarNumber nvarchar(-1) NOT NULL, CustomerSSNumber nvarchar(-1) NOT
NULL, CustomerBornDate datetime NULL, CustomerOccupation nvarchar(-1) NOT NULL,
CustomerIdentificationNumber nvarchar(-1) NOT NULL, CustomerExpirationIdentification datetime
NULL, IdOnWhoseBehalf int NULL, Purpose nvarchar(200) NOT NULL, Relationship nvarchar(200) NOT
NULL, MoneySource nvarchar(200) NOT NULL, AgentCommissionExtra money NULL,
AgentCommissionOriginal money NULL, ModifierCommissionSlider money NULL,
ModifierExchangeRateSlider money NULL, CustomerIdCarrier int NULL, ReviewDenyList bit NULL,
ReviewOfac bit NULL, ReviewKYC bit NULL, IdSeller int NOT NULL, ReviewRejected bit NULL,
ReviewGateway bit NULL, ReviewReturned bit NULL, OriginExRate money NOT NULL, OriginAmountInMN
money NOT NULL, DateStatusChange datetime NULL, CustomerIdentificationIdCountry int NULL,
CustomerIdentificationIdState int NULL, IdReasonForCancel int NULL, IdBeneficiaryIdentificationType int
NULL, BeneficiaryIdentificationNumber nvarchar(-1) NULL, AgentNotificationSent bit NOT NULL,
EmailByJobSent bit NOT NULL, FromStandByToKYC bit NULL, CustomerIdCountryOfBirth int NULL,
BeneficiaryIdCountryOfBirth int NULL, AccountTypeId int NULL, ReviewId bit NULL,
CustomerOccupationDetail nvarchar(-1) NULL, TransferIdCity int NULL, BeneficiaryIdCarrier int NULL,
FeeSecondary money NULL, NumModify int NULL, CustomerIdOccupation int NULL,
CustomerIdSubOccupation int NULL, CustomerSubOccupationOther varchar(50) NULL,
IdPaymentMethod int NOT NULL, Discount money NOT NULL, DateOfTransferUTC datetime NULL,
OperationFee money NOT NULL, IsValidCustomerPhoneNumber bit NULL, IdDialingCodePhoneNumber
int NULL, IdDialingCodeBeneficiaryPhoneNumber int NULL, IsRequiredCustomerPhoneNumber bit NULL,
IsRefunded bit NULL, ReviewCancelled bit NULL) , FOREIGN KEY (AccountTypeId) REFERENCES
AccountType(AccountTypeId), FOREIGN KEY (IdPaymentMethod) REFERENCES
PaymentMethod(IdPaymentMethod), FOREIGN KEY (TransferIdCity) REFERENCES City(IdCity), FOREIGN
KEY (IdReasonForCancel) REFERENCES ReasonForCancel(IdReasonForCancel), FOREIGN KEY
(IdDialingCodeBeneficiaryPhoneNumber) REFERENCES
DialingCodePhoneNumber(IdDialingCodePhoneNumber), FOREIGN KEY (IdDialingCodePhoneNumber)
REFERENCES DialingCodePhoneNumber(IdDialingCodePhoneNumber), FOREIGN KEY
(CustomerIdCountryOfBirth) REFERENCES CountryBirth(IdCountryBirth), FOREIGN KEY
(BeneficiaryIdCountryOfBirth) REFERENCES CountryBirth(IdCountryBirth)
```

```
CREATE TABLE [dbo].[AgentBalance] ( IdAgentBalance int NOT NULL PRIMARY KEY, IdAgent int NOT NULL,
TypeOfMovement nvarchar(-1) NOT NULL, DateOfMovement datetime NOT NULL, Amount money NOT
NULL, Reference nvarchar(-1) NOT NULL, Description nvarchar(-1) NOT NULL, Country nvarchar(-1) NOT
NULL, Commission money NOT NULL, DebitOrCredit nvarchar(-1) NOT NULL, Balance money NOT NULL,
IdTransfer int NULL, FxFee money NOT NULL, IsMonthly bit NULL) , FOREIGN KEY (IdAgent) REFERENCES
Agent(IdAgent)
```

```
CREATE TABLE [dbo].[Agent] ( IdAgent int NOT NULL PRIMARY KEY, IdAgentCommunication int NOT NULL,
IdAgentType int NOT NULL, IdUserSeller int NOT NULL, IdUserOpeningSalesRep int NULL, IdAgentStatus
int NOT NULL, IdAgentPaymentSchema int NOT NULL, IdAgentReceiptType int NULL, IdAgentBankDeposit
int NULL, AgentName nvarchar(-1) NOT NULL, AgentCode nvarchar(-1) NOT NULL, AgentAddress
```

nvarchar(-1) NOT NULL, AgentCity nvarchar(-1) NOT NULL, AgentState varchar(5) NULL, AgentZipcode nvarchar(-1) NOT NULL, AgentPhone nvarchar(-1) NOT NULL, AgentFax nvarchar(-1) NOT NULL, AgentEmail nvarchar(-1) NOT NULL, AgentContact nvarchar(-1) NOT NULL, AgentTimeInBusiness nvarchar(-1) NOT NULL, AgentBusinessType nvarchar(-1) NOT NULL, GuarantorName nvarchar(-1) NULL, GuarantorLastName nvarchar(-1) NULL, GuarantorSecondLastName nvarchar(-1) NULL, GuarantorAddress nvarchar(-1) NULL, GuarantorCity nvarchar(-1) NULL, GuarantorState nvarchar(-1) NULL, GuarantorZipcode nvarchar(-1) NULL, GuarantorPhone nvarchar(-1) NULL, GuarantorCel nvarchar(-1) NULL, GuarantorEmail nvarchar(-1) NULL, GuarantorSSN nvarchar(-1) NULL, GuarantorIdType nvarchar(-1) NULL, GuarantorIdNumber nvarchar(-1) NULL, GuarantorIdExpirationDate date NULL, GuarantorBornDate date NULL, TaxID nvarchar(-1) NOT NULL, CreditAmount money NOT NULL, AmountRequiredToAskId money NOT NULL, CreationDate datetime NOT NULL, OpenDate datetime NOT NULL, Notes nvarchar(-1) NOT NULL, CloseDate datetime NOT NULL, BusinessPermissionNumber nvarchar(-1) NULL, BusinessPermissionExpiration date NULL, DoneOnSundayPayOn int NOT NULL, DoneOnMondayPayOn int NOT NULL, DoneOnTuesdayPayOn int NOT NULL, DoneOnWednesdayPayOn int NOT NULL, DoneOnThursdayPayOn int NOT NULL, DoneOnFridayPayOn int NOT NULL, DoneOnSaturdayPayOn int NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser int NOT NULL, SwitchCommission bit NOT NULL, SwitchExrate bit NOT NULL, CommissionTop money NOT NULL, CommissionBottom money NOT NULL, ExrateTop money NOT NULL, ExrateBottom money NOT NULL, ACHWellsFargo bit NULL, ShowAgentProfitWhenSendingTransfer bit NOT NULL, GuarantorBornCountry nvarchar(-1) NULL, ExcludeReportSignatureHold bit NOT NULL, ExcludeReportExRates bit NOT NULL, ShowLogo bit NULL, CancelReturnCommission bit NOT NULL, IdAgentClass int NOT NULL, IdOwner int NULL, IdAgentCollectType int NOT NULL, AccountNumber nvarchar(-1) NULL, RoutingNumber nvarchar(-1) NULL, RetainMoneyCommission bit NOT NULL, DoingBusinessAs nvarchar(-1) NULL, IdAgentCommissionPay int NULL, SubAccount nvarchar(-1) NULL, AccountNumberCommission nvarchar(-1) NULL, RoutingNumberCommission nvarchar(-1) NULL, UsePin bit NOT NULL, UsePayNow bit NOT NULL, IdCounty int NULL, IdCountyGuarantor int NULL, SuspendedDatePendingFile datetime NULL, BlockPhoneTransactions bit NULL, MoneyAlertInvitation bit NOT NULL, CheckEditMicr bit NOT NULL, NoteCreditAmountChange varchar(-1) NULL, IdTimeZone int NULL, OfacBusinessChecked bit NULL, OfacGuarantorChecked bit NULL, OfacOwnerChecked bit NULL, NeedsWFSubaccount bit NOT NULL, NeedsWFSubaccountDate datetime NOT NULL, NeedsWFSubaccountIduser int NOT NULL, RequestWFSubaccount bit NOT NULL, RequestWFSubaccountDate datetime NOT NULL, RequestWFSubaccountIdUser int NOT NULL, IdAgentsReportWellsFargo int NOT NULL, StateCode varchar(-1) NULL, IdLogAccVerif int NULL, IdAgentGroup int NULL, IsSwitchSpecExRateGroup bit NOT NULL, ExpirationDateExRateGroup datetime NULL, ApplyKYCRules bit NULL, IdAgentEntityType int NULL, IdAgentTaxIdType int NULL, AgentBusinessEmail varchar(100) NULL, AgentBusinessWebsite varchar(150) NULL, AgentFinCENReg bit NULL, AgentFinCENRegExpiration date NULL, AgentCheckCasher bit NULL, AgentCheckLicense bit NULL, AgentCheckLicenseNumber varchar(50) NULL, MailCheckTo varchar(20) NULL, ComplianceOfficerDateOfBirth date NULL, ComplianceOfficerPlaceOfBirth varchar(250) NULL, ComplianceOfficerName varchar(200) NULL, IdAgentBusinessType int NULL, IdTaxIDType int NULL, OfacComplianceOfficerChecked bit NULL, IdAgent bigint NOT NULL PRIMARY KEY, AgentName varchar(200) NOT NULL, Address varchar(200) NULL, City varchar(200) NOT NULL, State varchar(200) NULL, ReceiveCapability bit NULL, SendCapability bit NULL, AgentPhone varchar(200) NULL, DateOfLastChange datetime NULL, CreationDate datetime NOT NULL, Active bit NOT NULL) , FOREIGN KEY (IdAgentBankDeposit) REFERENCES AgentBankDeposit(IdAgentBankDeposit), FOREIGN KEY

(IdAgentCommunication) REFERENCES AgentCommunication(IdAgentCommunication), FOREIGN KEY (IdAgentPaymentSchema) REFERENCES AgentPaymentSchema(IdAgentPaymentSchema), FOREIGN KEY (IdAgentReceiptType) REFERENCES AgentReceiptType(IdAgentReceiptType), FOREIGN KEY (IdAgentStatus) REFERENCES AgentStatus(IdAgentStatus), FOREIGN KEY (IdAgentType) REFERENCES AgentType(IdAgentType), FOREIGN KEY (IdUserSeller) REFERENCES Users(IdUser), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser), FOREIGN KEY (IdUserOpeningSalesRep) REFERENCES Users(IdUser), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser), FOREIGN KEY (IdAgentBusinessType) REFERENCES AgentBusinessTypes(IdAgentBusinessType), FOREIGN KEY (IdAgentCommissionPay) REFERENCES AgentCommissionPay(IdAgentCommissionPay), FOREIGN KEY (IdAgentClass) REFERENCES AgentClass(IdAgentClass), FOREIGN KEY (IdAgentCollectType) REFERENCES AgentCollectType(IdAgentCollectType), FOREIGN KEY (IdTaxIDType) REFERENCES TaxIDType(IdTaxIDType), FOREIGN KEY (IdOwner) REFERENCES Owner(IdOwner), FOREIGN KEY (IdAgentGroup) REFERENCES AgentGroup(IdAgentGroup)

CREATE TABLE [MoneyGram].[Agent] (IdAgent int NOT NULL PRIMARY KEY, IdAgentCommunication int NOT NULL, IdAgentType int NOT NULL, IdUserSeller int NOT NULL, IdUserOpeningSalesRep int NULL, IdAgentStatus int NOT NULL, IdAgentPaymentSchema int NOT NULL, IdAgentReceiptType int NULL, IdAgentBankDeposit int NULL, AgentName nvarchar(-1) NOT NULL, AgentCode nvarchar(-1) NOT NULL, AgentAddress nvarchar(-1) NOT NULL, AgentCity nvarchar(-1) NOT NULL, AgentState varchar(5) NULL, AgentZipcode nvarchar(-1) NOT NULL, AgentPhone nvarchar(-1) NOT NULL, AgentFax nvarchar(-1) NOT NULL, AgentEmail nvarchar(-1) NOT NULL, AgentContact nvarchar(-1) NOT NULL, AgentTimeInBusiness nvarchar(-1) NOT NULL, AgentBusinessType nvarchar(-1) NOT NULL, GuarantorName nvarchar(-1) NULL, GuarantorLastName nvarchar(-1) NULL, GuarantorSecondLastName nvarchar(-1) NULL, GuarantorAddress nvarchar(-1) NULL, GuarantorCity nvarchar(-1) NULL, GuarantorState nvarchar(-1) NULL, GuarantorZipcode nvarchar(-1) NULL, GuarantorPhone nvarchar(-1) NULL, GuarantorCel nvarchar(-1) NULL, GuarantorEmail nvarchar(-1) NULL, GuarantorSSN nvarchar(-1) NULL, GuarantorIdType nvarchar(-1) NULL, GuarantorIdNumber nvarchar(-1) NULL, GuarantorIdExpirationDate date NULL, GuarantorBornDate date NULL, TaxID nvarchar(-1) NOT NULL, CreditAmount money NOT NULL, AmountRequiredToAskId money NOT NULL, CreationDate datetime NOT NULL, OpenDate datetime NOT NULL, Notes nvarchar(-1) NOT NULL, CloseDate datetime NOT NULL, BusinessPermissionNumber nvarchar(-1) NULL, BusinessPermissionExpiration date NULL, DoneOnSundayPayOn int NOT NULL, DoneOnMondayPayOn int NOT NULL, DoneOnTuesdayPayOn int NOT NULL, DoneOnWednesdayPayOn int NOT NULL, DoneOnThursdayPayOn int NOT NULL, DoneOnFridayPayOn int NOT NULL, DoneOnSaturdayPayOn int NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser int NOT NULL, SwitchCommission bit NOT NULL, SwitchExrate bit NOT NULL, CommissionTop money NOT NULL, CommissionBottom money NOT NULL, ExrateTop money NOT NULL, ExrateBottom money NOT NULL, ACHWellsFargo bit NULL, ShowAgentProfitWhenSendingTransfer bit NOT NULL, GuarantorBornCountry nvarchar(-1) NULL, ExcludeReportSignatureHold bit NOT NULL, ExcludeReportExRates bit NOT NULL, ShowLogo bit NULL, CancelReturnCommission bit NOT NULL, IdAgentClass int NOT NULL, IdOwner int NULL, IdAgentCollectType int NOT NULL, AccountNumber nvarchar(-1) NULL, RoutingNumber nvarchar(-1) NULL, RetainMoneyCommission bit NOT NULL, DoingBusinessAs nvarchar(-1) NULL, IdAgentCommissionPay int NULL, SubAccount nvarchar(-1) NULL, AccountNumberCommission nvarchar(-1) NULL, RoutingNumberCommission nvarchar(-1) NULL, UsePin bit NOT NULL, UsePayNow bit NOT NULL, IdCounty int NULL, IdCountyGuarantor int NULL, SuspendedDatePendingFile datetime NULL, BlockPhoneTransactions bit NULL, MoneyAlertInvitation bit NOT NULL, CheckEditMicr bit NOT NULL, NoteCreditAmountChange varchar(-1) NULL, IdTimeZone int NULL, OfacBusinessChecked bit NULL, OfacGuarantorChecked bit NULL, OfacOwnerChecked bit NULL, NeedsWFSubaccount bit NOT NULL, NeedsWFSubaccountDate datetime NOT NULL, NeedsWFSubaccountIduser int NOT NULL, RequestWFSubaccount bit NOT NULL, RequestWFSubaccountDate datetime NOT NULL, RequestWFSubaccountIdUser int NOT NULL, IdAgentsReportWellsFargo int NOT NULL, StateCode varchar(-1) NULL, IdLogAccVerif int NULL, IdAgentGroup int NULL, IsSwitchSpecExRateGroup bit NOT NULL, ExpirationDateExRateGroup datetime NULL, ApplyKYCRules bit NULL, IdAgentEntityType int NULL, IdAgentTaxIdType int NULL, AgentBusinessEmail varchar(100) NULL, AgentBusinessWebsite varchar(150) NULL, AgentFinCENReg bit NULL, AgentFinCENRegExpiration date NULL, AgentCheckCasher bit NULL, AgentCheckLicense bit NULL, AgentCheckLicenseNumber varchar(50) NULL, MailCheckTo varchar(20) NULL, ComplianceOfficerDateOfBirth date NULL, ComplianceOfficerPlaceOfBirth varchar(250) NULL, ComplianceOfficerName varchar(200) NULL, IdAgentBusinessType int NULL, IdTaxIDType int NULL, OfacComplianceOfficerChecked bit NULL, IdAgent bigint NOT NULL PRIMARY KEY, AgentName varchar(200) NOT NULL, Address varchar(200) NULL, City varchar(200) NOT NULL, State varchar(200) NULL, ReceiveCapability bit NULL, SendCapability bit NULL, AgentPhone varchar(200) NULL, DateOfLastChange datetime NULL, CreationDate datetime NOT NULL, Active bit NOT NULL) , FOREIGN KEY (IdAgentBankDeposit) REFERENCES AgentBankDeposit(IdAgentBankDeposit), FOREIGN KEY (IdAgentCommunication) REFERENCES AgentCommunication(IdAgentCommunication), FOREIGN KEY (IdAgentPaymentSchema) REFERENCES AgentPaymentSchema(IdAgentPaymentSchema), FOREIGN KEY (IdAgentReceiptType) REFERENCES AgentReceiptType(IdAgentReceiptType), FOREIGN KEY (IdAgentStatus) REFERENCES AgentStatus(IdAgentStatus), FOREIGN KEY (IdAgentType) REFERENCES AgentType(IdAgentType), FOREIGN KEY (IdUserSeller) REFERENCES Users(IdUser), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser), FOREIGN KEY (IdUserOpeningSalesRep) REFERENCES Users(IdUser), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser), FOREIGN KEY (IdAgentBusinessType) REFERENCES AgentBusinessTypes(IdAgentBusinessType), FOREIGN KEY (IdAgentCommissionPay) REFERENCES AgentCommissionPay(IdAgentCommissionPay), FOREIGN KEY (IdAgentClass) REFERENCES AgentClass(IdAgentClass), FOREIGN KEY (IdAgentCollectType) REFERENCES AgentCollectType(IdAgentCollectType), FOREIGN KEY (IdTaxIDType) REFERENCES TaxIDType(IdTaxIDType), FOREIGN KEY (IdOwner) REFERENCES Owner(IdOwner), FOREIGN KEY (IdAgentGroup) REFERENCES AgentGroup(IdAgentGroup)

CREATE TABLE [dbo].[AgentOtherCharge] (IdAgentOtherCharge int NOT NULL PRIMARY KEY, IdAgent int NOT NULL, IdAgentBalance int NOT NULL, Amount money NOT NULL, ChargeDate datetime NOT NULL, Notes nvarchar(-1) NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser int NOT NULL, IdOtherChargesMemo int NOT NULL, OtherChargesMemoNote nvarchar(-1) NULL, IsReverse bit NOT NULL) , FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser), FOREIGN KEY (IdAgent) REFERENCES Agent(IdAgent), FOREIGN KEY (IdAgentBalance) REFERENCES AgentBalance(IdAgentBalance), FOREIGN KEY (IdOtherChargesMemo) REFERENCES OtherChargesMemo(IdOtherChargesMemo)

CREATE TABLE [dbo].[AgentCollectionDetail] (IdAgentCollectionDetail int NOT NULL PRIMARY KEY, IdAgentCollection int NOT NULL, LastAmountToPay money NOT NULL, ActualAmountToPay money NOT NULL, AmountToPay money NOT NULL, AmountExpected money NULL, Note nvarchar(-1) NULL, IdAgentCollectionConcept int NOT NULL, CreationDate datetime NOT NULL, DateofLastChange datetime NOT NULL, EnterByIdUser int NOT NULL) , FOREIGN KEY (IdAgentCollection) REFERENCES

AgentCollection(IdAgentCollection), FOREIGN KEY (IdAgentCollectionConcept) REFERENCES AgentCollectionConcept(IdAgentCollectionConcept)

CREATE TABLE [dbo].[CountryCurrency] (IdCountryCurrency int NOT NULL PRIMARY KEY, IdCountry int NOT NULL, IdCurrency int NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser int NOT NULL, CountryCode varchar(10) NOT NULL, BaseCurrency varchar(10) NOT NULL, LocalCurrency varchar(10) NULL, ReceiveCurrency varchar(10) NOT NULL, IndicativeRateAvailable bit NOT NULL, DeliveryOption varchar(30) NULL, ReceiveAgentID varchar(10) NULL, ReceiveAgentAbbreviation varchar(10) NULL, MgManaged varchar(20) NULL, AgentManaged varchar(20) NULL, ValidationExprs varchar(20) NULL, CheckDigitAlg varchar(20) NULL, DateOfLastChange datetime NULL, CreationDate datetime NOT NULL, ActiveForMaxi bit NOT NULL) , FOREIGN KEY (IdCountry) REFERENCES Country(IdCountry), FOREIGN KEY (IdCurrency) REFERENCES Currency(IdCurrency), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser)

CREATE TABLE [MoneyGram].[CountryCurrency] (IdCountryCurrency int NOT NULL PRIMARY KEY, IdCountry int NOT NULL, IdCurrency int NOT NULL, DateOfLastChange datetime NOT NULL, EnterByIdUser int NOT NULL, CountryCode varchar(10) NOT NULL, BaseCurrency varchar(10) NOT NULL, LocalCurrency varchar(10) NULL, ReceiveCurrency varchar(10) NOT NULL, IndicativeRateAvailable bit NOT NULL, DeliveryOption varchar(30) NULL, ReceiveAgentID varchar(10) NULL, ReceiveAgentAbbreviation varchar(10) NULL, MgManaged varchar(20) NULL, AgentManaged varchar(20) NULL, ValidationExprs varchar(20) NULL, CheckDigitAlg varchar(20) NULL, DateOfLastChange datetime NULL, CreationDate datetime NOT NULL, ActiveForMaxi bit NOT NULL) , FOREIGN KEY (IdCountry) REFERENCES Country(IdCountry), FOREIGN KEY (IdCurrency) REFERENCES Currency(IdCurrency), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser)

CREATE TABLE [dbo].[AgentCollection] (IdAgentCollection int NOT NULL PRIMARY KEY, IdAgent int NOT NULL, AmountToPay money NOT NULL, Fee money NOT NULL, EnterByIdUser int NOT NULL, CreationDate datetime NOT NULL, DateofLastChange datetime NOT NULL) , FOREIGN KEY (IdAgent) REFERENCES Agent(IdAgent), FOREIGN KEY (EnterByIdUser) REFERENCES Users(IdUser)

CREATE TABLE [dbo].[Users] (IdUser int NOT NULL PRIMARY KEY, UserName nvarchar(-1) NOT NULL, UserLogin nvarchar(-1) NOT NULL, UserPassword nvarchar(-1) NOT NULL, DateOfCreation datetime NOT NULL, CreatedByIdUser int NOT NULL, IdUserType int NOT NULL, ChangePasswordAtNextLogin bit NOT NULL, AllowToRegisterPc bit NOT NULL, IdGenericStatus int NOT NULL, salt nvarchar(50) NOT NULL, DateOfLastChange datetime NULL, EnterByIdUser int NOT NULL, FirstName nvarchar(100) NULL, LastName nvarchar(50) NULL, SecondLastName nvarchar(50) NULL) , FOREIGN KEY (IdGenericStatus) REFERENCES GenericStatus(IdGenericStatus), FOREIGN KEY (IdUserType) REFERENCES UsersType(IdUserType)

CREATE TABLE [dbo].[SpecialCommissionBalance] (IdSpecialCommissionBalance int NOT NULL PRIMARY KEY, IdAgent int NOT NULL, DateOfMovement datetime NOT NULL, Commission money NOT NULL, IdSpecialCommissionRule int NOT NULL, DateOfApplication datetime NOT NULL) , FOREIGN KEY (IdAgent) REFERENCES Agent(IdAgent)

CREATE TABLE [dbo].[PosTransfer] (IdPosTransfer int NOT NULL PRIMARY KEY, IdTransfer int NULL, IdTransferClosed int NULL, AuthorizationNo nvarchar(10) NOT NULL, BatchNo nvarchar(100) NOT NULL, ReferenceNo nvarchar(100) NOT NULL, TerminalId nvarchar(100) NOT NULL, MerchantId nvarchar(100)

```
NOT NULL, AccountNo nvarchar(100) NOT NULL, IdPosStatus smallint NOT NULL, IdCardEntryMode
smallint NOT NULL, IdCardType smallint NOT NULL, IdPosActionType int NOT NULL, CreationDate
datetime NULL, IdUser int NULL) , FOREIGN KEY (IdUser) REFERENCES Users(IdUser), FOREIGN KEY
(IdTransfer) REFERENCES Transfer(IdTransfer), FOREIGN KEY (IdTransferClosed) REFERENCES
TransferClosed(IdTransferClosed)

*/


BEGIN TRY

  DECLARE @IdCountryCurrency INT,

          @StartDate DATE,

          @EndDate DATE,

          @Range1 DECIMAL(18, 2),

          @Range2 DECIMAL(18, 2),

          @State NVARCHAR(50),

          @ReportCorpType INT;


  SET @IdCountryCurrency = 1;

  SET @StartDate = '2023-01-01';

  SET @EndDate = '2023-06-01';

  SET @Range1 = NULL;

  SET @Range2 = NULL;

  SET @State = NULL;

  SET @ReportCorpType = 1;



  SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;


  DROP TABLE IF EXISTS #Transfer,

               #TransferRejected,

               #TransferExpired,
               #TransferClosedCancel,

               #TransferClosed,

               #TransferUnclaimed,

               #TransferClosedUnclaimed,

               #AgentBalance,

               #Agent,

               #AgentOtherCharge,

               #AgentCollectionDetail,

               #CountryCurrency,

               #AgentCollection,

               #Users,

               #SpecialCommissionBalance,

               #PosTransfer,

               #OtherCharges;




  SELECT *

  INTO #Transfer

  FROM dbo.Transfer

  WHERE IdCountryCurrency = CASE

                  WHEN @IdCountryCurrency = 0 THEN

                    IdCountryCurrency

                  ELSE

                    @IdCountryCurrency

                END

      AND DateOfTransfer > @StartDate

      AND DateOfTransfer < @EndDate
      OR (
          DateStatusChange > @StartDate
          AND DateStatusChange < @EndDate
          AND IdStatus IN ( 31, 22, 27 )
        )
        AND
        (
          (
            @Range1 IS NULL
            AND @Range2 IS NULL
          )
          OR AmountInDollars
      BETWEEN @Range1 AND @Range2
        );



  SELECT IdAgent,
      IdCountryCurrency,
      DateOfTransfer,
      DateStatusChange,
      IdStatus,
      AmountInDollars
  INTO #TransferClosed
  FROM dbo.TransferClosed
  WHERE IdCountryCurrency = CASE
                    WHEN @IdCountryCurrency = 0 THEN
                      IdCountryCurrency
                    ELSE
                      @IdCountryCurrency
                  END
      AND DateOfTransfer > @StartDate
      AND DateOfTransfer < @EndDate
      OR (
          DateStatusChange > @StartDate
          AND DateStatusChange < @EndDate
          AND IdStatus IN ( 31, 22 )
        )
        AND
        (
          (
            @Range1 IS NULL
            AND @Range2 IS NULL
          )
          OR AmountInDollars
      BETWEEN @Range1 AND @Range2
        );



  SELECT IdAgentBalance,
      DateOfMovement,
      TypeOfMovement
  INTO #AgentBalance
  FROM AgentBalance
  WHERE DateOfMovement >= @StartDate
```

```
    AND DateOfMovement < @EndDate
    AND
    (
      TypeOfMovement = 'CGO'
      OR TypeOfMovement = 'DEBT'
    );


SELECT IdAgentBalance,
     IdOtherChargesMemo
INTO #AgentOtherCharge
FROM AgentOtherCharge
WHERE IdOtherChargesMemo IN ( 6, 9, 13, 19, 4, 5, 11, 12, 16, 17, 18, 24, 25 );



SELECT DateofLastChange
INTO #AgentCollectionDetail
FROM AgentCollectionDetail
WHERE DateofLastChange >= @StartDate
    AND DateofLastChange < @EndDate;


SELECT IdAgent
INTO #AgentCollection
FROM AgentCollection;
```

```
SELECT IdAgent,
    DateOfApplication
INTO #SpecialCommissionBalance
FROM dbo.SpecialCommissionBalance
WHERE DateOfApplication >= @StartDate
   AND DateOfApplication < @EndDate;
```

```
SELECT IdCountryCurrency
INTO #CountryCurrency
FROM CountryCurrency;
```

```
SELECT IdAgent,
    AgentName,
    AgentCode,
    IdUserSeller,
    AgentState
INTO #Agent
FROM Agent
WHERE AgentState = ISNULL(@State, AgentState);
```

```
SELECT IdUser,
    UserName
INTO #Users
FROM Users;
```

```
SELECT IdTransfer,
    IdPosTransfer
INTO #PosTransfer
FROM PosTransfer;
```

```
SELECT t.IdAgent,
    IdCountry,
    t.IdCountryCurrency,
    IdGateway,
    IdPayer,
    IdPaymentType,
    AmountInDollars
INTO #TransferRejected
FROM Transfer t
  JOIN Agent a
     ON t.IdAgent = a.IdAgent
    JOIN CountryCurrency cc
      ON t.IdCountryCurrency = cc.IdCountryCurrency
  WHERE t.IdCountryCurrency = IIF(@IdCountryCurrency = 0, t.IdCountryCurrency,
@IdCountryCurrency)
     AND DateStatusChange > @StartDate
     AND DateStatusChange < @EndDate
     AND t.IdAgent IN
       (
         SELECT IdAgent FROM #Agent
       )
     AND IdStatus = 31
     AND
     (
       (
         @Range1 IS NULL
         AND @Range2 IS NULL
       )
       OR AmountInDollars
     BETWEEN @Range1 AND @Range2
     );



  SELECT IdAgent,
      IdCountryCurrency,
      DateOfTransfer,
      DateStatusChange,
      IdStatus,
      AmountInDollars
```

```
INTO #TransferExpired
FROM dbo.Transfer
WHERE IdCountryCurrency = CASE
                WHEN @IdCountryCurrency = 0 THEN
                  IdCountryCurrency
                ELSE
                  @IdCountryCurrency
              END
   AND DateOfTransfer > @StartDate
   AND DateOfTransfer < @EndDate
   OR (
       DateStatusChange > @StartDate
       AND DateStatusChange < @EndDate
       AND IdStatus IN ( 31, 22, 27 )
     )
     AND
     (
       (
         @Range1 IS NULL
         AND @Range2 IS NULL
       )
       OR AmountInDollars
   BETWEEN @Range1 AND @Range2
     );



SELECT IdTransferClosed,
    IdAgent,
    IdCountry,
    IdCountryCurrency,
    IdGateway,
    IdPayer,
    IdPaymentType,
    AmountInDollars
INTO #TransferClosedCancel
FROM TransferClosed t
WHERE IdCountryCurrency = IIF(@IdCountryCurrency = 0, IdCountryCurrency, @IdCountryCurrency)
   AND DateStatusChange > @StartDate
   AND DateStatusChange < @EndDate
   AND IdAgent IN
     (
       SELECT IdAgent FROM #Agent
     )
   AND IdStatus = 22
   AND
   (
     (
       @Range1 IS NULL
       AND @Range2 IS NULL
     )
     OR t.AmountInDollars
   BETWEEN @Range1 AND @Range2
   );



SELECT ab.IdAgent,
          SUM(IIF(ab.DebitOrCredit = 'Credit', ab.Amount, ab.Amount * (-1))) OVER (PARTITION BY
ab.IdAgent) AS OtherCharges1,

          SUM(IIF(oc.IdOtherChargesMemo IN ( 6, 9, 13, 19 ),

                  IIF(ab.DebitOrCredit = 'Credit', ab.Amount, ab.Amount * (-1)),

                  0)

             ) OVER (PARTITION BY ab.IdAgent) AS OtherChargesC1,

          SUM(IIF(oc.IdOtherChargesMemo IN ( 4, 5, 11, 12, 16, 17, 18, 24, 25 ),

                  IIF(ab.DebitOrCredit != 'Credit', ab.Amount, ab.Amount * (-1)),

                  0)

             ) OVER (PARTITION BY ab.IdAgent) AS OtherChargesD1

   INTO #OtherCharges

   FROM AgentBalance ab

       JOIN AgentOtherCharge oc

           ON ab.IdAgentBalance = oc.IdAgentBalance

              AND oc.IdOtherChargesMemo IN ( 6, 9, 13, 19, 4, 5, 11, 12, 16, 17, 18, 24, 25 )

   WHERE ab.DateOfMovement >= @StartDate

         AND ab.DateOfMovement < @EndDate

         AND

         (

             ab.TypeOfMovement = 'CGO'

             OR ab.TypeOfMovement = 'DEBT'

         );




   INSERT INTO #OtherCharges

   SELECT AC.IdAgent,

          SUM(o.AmountToPay) OVER (PARTITION BY AC.IdAgent) AS OtherCharges2,
         SUM(IIF(o.AmountToPay > 0, o.AmountToPay, 0)) OVER (PARTITION BY AC.IdAgent) AS
OtherChargesC2,
         SUM(IIF(o.AmountToPay < 0, o.AmountToPay * (-1), 0)) OVER (PARTITION BY AC.IdAgent) AS
OtherChargesD2
   FROM AgentCollectionDetail o
      INNER JOIN AgentCollection AC
         ON AC.IdAgentCollection = o.IdAgentCollection
   WHERE @ReportCorpType = 1
        AND o.DateofLastChange >= @StartDate
        AND o.DateofLastChange < @EndDate;



   SELECT IdAgent,
          IdCountry,
          t.IdCountryCurrency
   INTO #TransferUnclaimed
   FROM Transfer t
      JOIN CountryCurrency cc
         ON t.IdCountryCurrency = cc.IdCountryCurrency
   WHERE t.IdCountryCurrency = CASE
                                 WHEN @IdCountryCurrency = 0 THEN
                                    t.IdCountryCurrency
                                 ELSE
                                    @IdCountryCurrency
                              END
        AND DateStatusChange > @StartDate
        AND DateStatusChange < @EndDate
        AND IdStatus = 27
```

```
    AND
    (
       (
          @Range1 IS NULL
          AND @Range2 IS NULL
       )
       OR t.AmountInDollars
    BETWEEN @Range1 AND @Range2
    );



SELECT IdAgent,
     IdCountry,
     IdCountryCurrency
INTO #TransferClosedUnclaimed
FROM dbo.TransferClosed t
WHERE IdCountryCurrency = CASE
                    WHEN @IdCountryCurrency = 0 THEN
                       IdCountryCurrency
                    ELSE
                       @IdCountryCurrency
                 END
    AND DateStatusChange > @StartDate
    AND DateStatusChange < @EndDate
    AND IdStatus = 27
    AND
    (
       (
            @Range1 IS NULL
            AND @Range2 IS NULL
          )
          OR t.AmountInDollars
       BETWEEN @Range1 AND @Range2
       );

END TRY
BEGIN CATCH
   DECLARE @ErrorMessage NVARCHAR(4000);
   DECLARE @ErrorSeverity INT;
   DECLARE @ErrorState INT;
   SELECT @ErrorMessage = ERROR_MESSAGE(),
        @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE();
   RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
END CATCH;




@codigo2 =  " ------------------------------Calculate OutPut --------------------------------------------------
   SELECT A.*,
        B.NumTrans1,
        B.AmountTrans1,
        B.FxResult1,
        B.AgentcommissionMonthly1,
        B.AgentcommissionRetain1,
B.IncomeFee1,
B.FxFee1,
B.FxFeeM1,
B.FxFeeR1,
B.BankCommission1,
B.PayerCommission1,
B.CogsTrans1,
B.DCTr1,
B.MerchantFee1,
B.CashDiscount1,
B.NetFee1,
C.NumTrans2,
C.AmountTrans2,
C.FxResult2,
C.AgentcommissionMonthly2,
C.AgentcommissionRetain2,
C.IncomeFee2,
C.FxFee2,
C.FxFeeM2,
C.FxFeeR2,
C.BankCommission2,
C.PayerCommission2,
C.CogsTrans2,
C.DCTr2,
C.MerchantFee2,
C.CashDiscount2,
C.NetFee2,
D.NumTransRej1,
D.AmountTransRej1,
D.FxResultRej1,
D.AgentcommissionMonthlyRej1,
D.AgentcommissionRetainRej1,
D.IncomeFeeRej1,
D.FxFeeRej1,
D.FxFeeRejM1,
D.FxFeeRejR1,
D.BankCommission3,
D.PayerCommission3,
D.CogsRej1,
D.DCTr3,
D.MerchantFee3,
D.CashDiscount3,
D.NetFee3,
E.NumTrans2Rej,
E.AmountTrans2Rej,
E.FxResult2Rej,
E.AgentcommissionMonthly2Rej,
E.AgentcommissionRetain2Rej,
E.IncomeFee2Rej,
E.FxFee2Rej,
E.FxFeeM2Rej,
E.FxFeeR2Rej,
E.BankCommission4,
E.PayerCommission4,
E.CogsRej2,
E.DCTr4,
E.MerchantFee4,
E.CashDiscount4,
E.NetFee4,
F.CogsCancel1,
F.NumCancel1,
F.AmountCancel1,
F.FxResultCancel1,
F.AgentcommissionMonthlyCan1,
F.AgentcommissionRetainCan1,
F.FxFeeCan1,
F.FxFeeCanM1,
F.FxFeeCanR1,
F.IncomeFeeCan1,
F.IncomeFeeCancelLikeReject1,
F.AgentcommissionRetainCancelLikeReject1,
F.BankCommission5,
F.PayerCommission5,
F.DCTr5,
F.MerchantFee5,
F.CashDiscount5,
F.NetFee5,
G.CogsCancel2,
G.NumCancel2,
G.AmountCancel2,
G.FxResultCancel2,
G.AgentcommissionMonthlyCan2,
G.AgentcommissionRetainCan2,
G.FxFeeCan2,
G.FxFeeCanM2,
G.FxFeeCanR2,
G.IncomeFeeCan2,
    G.IncomeFeeCancelLikeReject2,
    G.AgentcommissionRetainCancelLikeReject2,
    G.BankCommission6,
    G.PayerCommission6,
    G.DCTr6,
    G.MerchantFee6,
    G.CashDiscount6,
    G.NetFee6,
    H.OtherCharges1,
    H.OtherChargesC1,
    H.OtherChargesD1,
    I.UnclaimedCOGS1,
    I.UnclaimedNumTrans1,
    I.UnclaimedAmount1,
    J.UnclaimedCOGSClosed,
    J.UnclaimedNumTransClosed,
    J.UnclaimedAmountClosed,
    K.OtherCharges2,
    K.OtherChargesC2,
    K.OtherChargesD2
INTO #Result
FROM #Temp A
  LEFT JOIN #temp1 B
     ON (A.IdAgent = B.IdAgent)
      AND (A.IdCountry = B.IdCountry)
      AND A.IdCountryCurrency = B.IdCountryCurrency
  LEFT JOIN #temp2 C
     ON (A.IdAgent = C.IdAgent)
      AND (A.IdCountry = C.IdCountry)
    AND A.IdCountryCurrency = C.IdCountryCurrency
LEFT JOIN #temp3 D
  ON (A.IdAgent = D.IdAgent)
    AND (A.IdCountry = D.IdCountry)
    AND A.IdCountryCurrency = D.IdCountryCurrency
LEFT JOIN #temp4 E
  ON (A.IdAgent = E.IdAgent)
    AND (A.IdCountry = E.IdCountry)
    AND A.IdCountryCurrency = E.IdCountryCurrency
LEFT JOIN #temp5 F
  ON (A.IdAgent = F.IdAgent)
    AND (A.IdCountry = F.IdCountry)
    AND A.IdCountryCurrency = F.IdCountryCurrency
LEFT JOIN #temp6 G
  ON (A.IdAgent = G.IdAgent)
    AND (A.IdCountry = G.IdCountry)
    AND A.IdCountryCurrency = G.IdCountryCurrency
LEFT JOIN #temp7 H
  ON (A.IdAgent = H.IdAgent) --other charges
LEFT JOIN #temp8 I
  ON (A.IdAgent = I.IdAgent)
    AND (A.IdCountry = I.IdCountry)
    AND A.IdCountryCurrency = I.IdCountryCurrency
LEFT JOIN #temp9 J
  ON (A.IdAgent = J.IdAgent)
    AND (A.IdCountry = J.IdCountry)
    AND A.IdCountryCurrency = J.IdCountryCurrency
LEFT JOIN #temp10 K
  ON (A.IdAgent = K.IdAgent); --other charges
```

```
    UPDATE #Result

    SET NumTrans = COALESCE(NumTrans1, 0) + COALESCE(NumTrans2, 0),

        NumCancel = COALESCE(NumCancel1, 0) + COALESCE(NumCancel2, 0) + COALESCE(NumTransRej1,
0)

                + COALESCE(NumTrans2Rej, 0),

        AmountTrans = COALESCE(AmountTrans1, 0) + COALESCE(AmountTrans2, 0),

        AmountCancel = COALESCE(AmountCancel1, 0) + COALESCE(AmountCancel2, 0) +
COALESCE(AmountTransRej1, 0)

                  + COALESCE(AmountTrans2Rej, 0),

        OtherCharges = COALESCE(OtherCharges1, 0) + COALESCE(OtherCharges2, 0),

        OtherChargesD = COALESCE(OtherChargesD1, 0) + COALESCE(OtherChargesD2, 0),

        OtherChargesC = COALESCE(OtherChargesC1, 0) + COALESCE(OtherChargesC2, 0),

        CogsCancel = COALESCE(CogsCancel1, 0) + COALESCE(CogsCancel2, 0) + COALESCE(CogsRej1, 0)

                + COALESCE(CogsRej2, 0),

        FxResult = COALESCE(FxResult1, 0) + COALESCE(FxResult2, 0) - COALESCE(FxResultRej1, 0)

               - COALESCE(FxResult2Rej, 0) - COALESCE(FxResultCancel1, 0) - COALESCE(FxResultCancel2, 0),

        AgentcommissionMonthly = COALESCE(AgentcommissionMonthly1, 0) +
COALESCE(AgentcommissionMonthly2, 0)

                            - COALESCE(AgentcommissionMonthlyRej1, 0) -
COALESCE(AgentcommissionMonthly2Rej, 0)

                            - COALESCE(AgentcommissionMonthlyCan1, 0) -
COALESCE(AgentcommissionMonthlyCan2, 0),

        AgentcommissionRetain = COALESCE(AgentcommissionRetain1, 0) +
COALESCE(AgentcommissionRetain2, 0)

                           - COALESCE(AgentcommissionRetainRej1, 0) - COALESCE(AgentcommissionRetain2Rej,
0)

                           - COALESCE(AgentcommissionRetainCan1, 0) -
COALESCE(AgentcommissionRetainCan2, 0)

                           - COALESCE(AgentcommissionRetainCancelLikeReject1, 0)

                           - COALESCE(AgentcommissionRetainCancelLikeReject2, 0),
        FxFee = COALESCE(FxFee1, 0) + COALESCE(FxFee2, 0) - COALESCE(FxFeeRej1, 0) -
COALESCE(FxFee2Rej, 0),

        FxFeeM = COALESCE(FxFeeM1, 0) + COALESCE(FxFeeM2, 0) - COALESCE(FxFeeRejM1, 0) -
COALESCE(FxFeeM2Rej, 0),

        FxFeeR = COALESCE(FxFeeR1, 0) + COALESCE(FxFeeR2, 0) - COALESCE(FxFeeRejR1, 0) -
COALESCE(FxFeeR2Rej, 0),

        UnclaimedNumTrans = COALESCE(UnclaimedNumTrans1, 0) +
COALESCE(UnclaimedNumTransClosed, 0),

        UnclaimedAmount = COALESCE(UnclaimedAmount1, 0) + COALESCE(UnclaimedAmountClosed, 0),

        UnclaimedCOGS = COALESCE(UnclaimedCOGS1, 0) + COALESCE(UnclaimedCOGSClosed, 0),

        BankCommission = COALESCE(BankCommission1, 0) + COALESCE(BankCommission2, 0) -
COALESCE(BankCommission3, 0)

                        - COALESCE(BankCommission4, 0) - COALESCE(BankCommission5, 0) -
COALESCE(BankCommission6, 0),

        PayerCommission = COALESCE(PayerCommission1, 0) + COALESCE(PayerCommission2, 0) -
COALESCE(PayerCommission3, 0)

                         - COALESCE(PayerCommission4, 0) - COALESCE(PayerCommission5, 0)

                         - COALESCE(PayerCommission6, 0),

        IncomeFee = COALESCE(IncomeFee1, 0) + COALESCE(IncomeFee2, 0),

        NetFee = (COALESCE(IncomeFee1, 0) + COALESCE(IncomeFee2, 0))

                 - (COALESCE(IncomeFeeRej1, 0) + COALESCE(IncomeFee2Rej, 0) + COALESCE(IncomeFeeCan1,
0)

                   + COALESCE(IncomeFeeCan2, 0)

                   )

                 - ((COALESCE(CashDiscount1, 0) + COALESCE(CashDiscount2, 0))

                   - (COALESCE(CashDiscount3, 0) + COALESCE(CashDiscount4, 0))

                   - (COALESCE(CashDiscount5, 0) + COALESCE(CashDiscount6, 0))

                   ),

        FeeCanR = COALESCE(IncomeFeeRej1, 0) + COALESCE(IncomeFee2Rej, 0) +
COALESCE(IncomeFeeCan1, 0)

                  + COALESCE(IncomeFeeCan2, 0),

        CashDiscount = (COALESCE(CashDiscount1, 0) + COALESCE(CashDiscount2, 0))
```

```
            - (COALESCE(CashDiscount3, 0) + COALESCE(CashDiscount4, 0))

            - (COALESCE(CashDiscount5, 0) + COALESCE(CashDiscount6, 0)),

    DCTran = COALESCE(DCTr1, 0) + COALESCE(DCTr2, 0),

    MerchantFee = COALESCE(MerchantFee1, 0) + COALESCE(MerchantFee2, 0);


  IF @Type = 1

  BEGIN

    UPDATE #Result

    SET AgentcommissionMonthly = AgentcommissionMonthly - FxFeeM

    WHERE AgentcommissionMonthly > 0;


    UPDATE #Result

    SET AgentcommissionRetain = AgentcommissionRetain - FxFeeR

    WHERE AgentcommissionRetain > 0;

  END;

  ELSE

  BEGIN

    UPDATE #Result

    SET AgentcommissionMonthly = AgentcommissionMonthly - FxFeeM,

      AgentcommissionRetain = AgentcommissionRetain - FxFeeR;

  END;


  UPDATE #Result

  SET NumNet = NumTrans - NumCancel,

    AmountNet = AmountTrans - AmountCancel,

    Result = FxResult + NetFee - AgentcommissionMonthly - AgentcommissionRetain - FxFee -
PayerCommission

        + UnclaimedAmount - UnclaimedCOGS - MerchantFee; ---BankCommission

  UPDATE #Result
```

```
SET CogsNet = AmountNet - FxResult;

UPDATE #Result

SET OtherCharges = 0

WHERE OtherCharges IS NULL;

UPDATE #Result

SET OtherChargesD = 0

WHERE OtherChargesD IS NULL;

UPDATE #Result

SET OtherChargesC = 0

WHERE OtherChargesC IS NULL;

UPDATE #Result

SET NetResult = Result + OtherCharges,

   CogsTrans = CogsCancel + CogsNet;


--------------------------------------------- Calculo de Other Products

CREATE TABLE #tOtherProd

(

   [idAgent] INT,

   [AgentName] VARCHAR(100),

   [AgentCode] VARCHAR(50),

   [Total] INT,

   [CancelsTotal] INT,

   [TotalNet] INT,

   [Amount] MONEY,

   [CGS] MONEY,

   [Fee] MONEY,

   [FeeM] MONEY,

   [FeeR] MONEY,

   [ProviderComm] MONEY,
   [CorpCommission] MONEY,
   [AgentCommMonthly] MONEY,
   [AgentCommRetain] MONEY,
   [FX] MONEY,
   [CheckFees] MONEY,      /*2015-Ago-15*/
   [ReturnedFee] MONEY,
   [TransactionFee] MONEY, /*2015-Sep-21*/
   [CustomerFee] MONEY,    /*2015-Sep-21*/
   [ProccessingFee] MONEY, /*2020-Jul*/
   [ScannerFee] MONEY      /*2015-Sep-21*/
);



--------------------------------------------- Calculo DepositAgent
SELECT IdAgent,
       SUM(DepositAgent) DepositAgent
INTO #tDepositAgent
FROM
(
   SELECT ab.IdAgent,
          ISNULL((SUM(   CASE
                            WHEN DebitOrCredit = 'Credit' THEN
                                Amount
                            ELSE
                                0
                        END
                     ) - SUM(   CASE
                                    WHEN DebitOrCredit = 'Debit' THEN
                                        Amount
```

```
                    ELSE
                      0
                  END
                )
          ) * FactorNew,
          0
          ) DepositAgent
    FROM #TempAgents
      INNER JOIN AgentBalance ab
        ON ab.IdAgent = #TempAgents.IdAgent
      INNER JOIN Agent ag
        ON ab.IdAgent = ag.IdAgent
          AND ag.IdAgentBankDeposit NOT IN ( 42, 43, 46 )
      LEFT JOIN #bankcommission bc
        ON bc.DateOfBankCommission = CAST(DATEADD(dd, - (DAY(DateOfMovement) - 1), DateOfMovement) AS DATE)
    WHERE ab.TypeOfMovement = 'DEP'
       AND DateOfMovement >= @StartDate
       AND DateOfMovement < @EndDate
    GROUP BY ab.IdAgent,
        bc.FactorNew
  ) cteD
  GROUP BY IdAgent;
  --------------------------------------------- Calculo DepositAgent


  ------------------------------Output -------------------------------------------------
  SELECT t.IdAgent,
      AgentCode,
      AgentName,
NumTrans,
NumCancel,                                  --No mostrar
NumNet,
AmountTrans,
AmountCancel,
AmountNet,
CogsTrans,
CogsCancel,
CogsNet,                                  --No mostrar
FxResult,
IncomeFee,
t.FeeCanR,
t.CashDiscount,
t.NetFee,
AgentcommissionMonthly,
AgentcommissionRetain,
FxFeeM,
FxFeeR,
ISNULL(SpecialCommission, 0) SpecialCommission,                 --No mostrar
                                      ----
PayerCommission,                              --Cambio en el calculo
                                      ----
UnclaimedAmount,
UnclaimedCOGS,
OtherCharges,
OtherChargesC,
OtherChargesD,                               --No mostrar
Result,
                                      ----
```

```
      NetResult - ISNULL(SpecialCommission, 0) NetResult,              --Cambio en el calculo --
Profit
      CASE
        WHEN NumNet != 0 THEN
      (NetResult - ISNULL(SpecialCommission, 0)) / NumNet
        ELSE
          0
      END Margin,                                       --Cambio en el calculo
                                                  ----
      ISNULL(UserName, '') Parent,
      SalesRep,                                       --No mostrar
      CASE
        WHEN @Type = 2 THEN
          CASE
            WHEN ISNULL(c.CountryCode, '') = 'HTI' THEN
              'HAI'
            WHEN ISNULL(c.CountryCode, '') = 'PRY' THEN
              'PAR'
            ELSE
              ISNULL(c.CountryCode, '')
          END
        WHEN @Type = 3 THEN
          CASE
            WHEN ISNULL(t.IdCountryCurrency, 0) != 0 THEN
              CASE
                WHEN ISNULL(c2.CountryCode, '') = 'HTI' THEN
                  'HAI' + '/' + ISNULL(cu.CurrencyCode, '')
                WHEN ISNULL(c2.CountryCode, '') = 'PRY' THEN
                  'PAR' + '/' + ISNULL(cu.CurrencyCode, '')
```

```
                            ELSE
                                ISNULL(c2.CountryCode, '') + '/' + ISNULL(cu.CurrencyCode, '')
                        END
                    ELSE
                        ''
                END
            ELSE
                ''
        END CountryCode,
                                                                                 --------------------
        c.CountryName,
        (AgentcommissionMonthly + AgentcommissionRetain + FxFeeM + FxFeeR +
ISNULL(SpecialCommission, 0))
        + OtherChargesC - OtherChargesD CommSeller,
        DepositAgent BkFeesSeller,
        [OtherProducts] OtherProductsSeller,
        (FxResult + NetFee)
        - ((AgentcommissionMonthly + AgentcommissionRetain + FxFeeM + FxFeeR +
ISNULL(SpecialCommission, 0))
           + OtherChargesC - OtherChargesD
          ) - PayerCommission - DepositAgent + [OtherProducts] - MerchantFee ProfitSeller, ---W2+X2
        CASE
            WHEN ISNULL(NumNet, 0) > 0 THEN
        ((FxResult + NetFee)
         - ((AgentcommissionMonthly + AgentcommissionRetain + FxFeeM + FxFeeR +
ISNULL(SpecialCommission, 0))
            + OtherChargesC - OtherChargesD
           ) - PayerCommission - DepositAgent + [OtherProducts] - MerchantFee
        )
        / NumNet
```

```
      ELSE
        0
    END MarginSeller,
    t.DCTran AS DCTran,
    t.MerchantFee

--------------------
FROM
(
  SELECT #Result.IdAgent,
      IdCountry,
      IdCountryCurrency,
      #Result.AgentCode,
      #Result.AgentName,
      NumTrans,
      NumCancel,
      NumNet,
      AmountTrans,
      AmountCancel,
      AmountNet,
      CogsTrans,
      CogsCancel,
      CogsNet,
      FxResult,
      IncomeFee,
      AgentcommissionMonthly,
      AgentcommissionRetain,
      FxFeeM,
      FxFeeR,
        CASE
          WHEN PayerCommission > 0 THEN
            PayerCommission
          ELSE
            0
        END AS PayerCommission,
        UnclaimedAmount,
        UnclaimedCOGS,
        OtherCharges,
        OtherChargesC,
        OtherChargesD,
        Result,
        FxResult + NetFee - AgentcommissionMonthly - AgentcommissionRetain - FxFeeM - FxFeeR
        - CASE
           WHEN PayerCommission > 0 THEN
             PayerCommission
           ELSE
             0
         END - UnclaimedAmount + UnclaimedCOGS - OtherChargesC + OtherChargesD - MerchantFee
AS NetResult,
        (
          SELECT IdUserSellerParent FROM Seller WHERE IdUserSeller = IdSalesRep
        ) AS IdUserSellerParent,
        SalesRep,
        BankCommission,
        ISNULL(#tOtherProd.[CorpCommission], 0.0) [OtherProducts],
        ISNULL(#tDepositAgent.[DepositAgent], 0.0) [DepositAgent],
        FeeCanR,
        CashDiscount,
    DCTran,
    NetFee,
    MerchantFee
FROM #Result
  -- Join with #tOtherProd to get CorpCommission for each agent
  LEFT JOIN #tOtherProd
    ON #Result.IdAgent = #tOtherProd.idAgent
  -- Join with #tDepositAgent to get DepositAgent for each agent
  LEFT JOIN
  (
    SELECT IdAgent,
        SUM(DepositAgent) AS DepositAgent
    FROM
    (
      SELECT ab.IdAgent,
          ISNULL((SUM(   CASE
                   WHEN DebitOrCredit = 'Credit' THEN
                     Amount
                   ELSE
                     0
                 END
               ) - SUM(   CASE
                       WHEN DebitOrCredit = 'Debit' THEN
                         Amount
                       ELSE
                         0
                     END
                   )
             ) * bc.FactorNew,
                 0
                 ) AS DepositAgent
          FROM #TempAgents
            INNER JOIN AgentBalance ab
              ON ab.IdAgent = #TempAgents.IdAgent
            INNER JOIN Agent ag
              ON ab.IdAgent = ag.IdAgent
                AND ag.IdAgentBankDeposit NOT IN ( 42, 43, 46 )
            LEFT JOIN #bankcommission bc
              ON bc.DateOfBankCommission = CAST(DATEADD(dd, - (DAY(DateOfMovement) - 1), DateOfMovement) AS DATE)
          WHERE ab.TypeOfMovement = 'DEP'
             AND DateOfMovement >= @StartDate
             AND DateOfMovement < @EndDate
          GROUP BY ab.IdAgent,
               bc.FactorNew
        ) cteD
        GROUP BY IdAgent
      ) AS #tDepositAgent
        ON #Result.IdAgent = #tDepositAgent.IdAgent

  ---------------------------------------------------
  ) t
    LEFT JOIN Users u
      ON u.IdUser = ISNULL(IdUserSellerParent, 0)
    LEFT JOIN #tempSC s
      ON s.IdAgent = t.IdAgent
    LEFT JOIN Country c
      ON t.IdCountry = c.IdCountry
    LEFT JOIN CountryCurrency cc
        ON t.IdCountryCurrency = cc.IdCountryCurrency
    LEFT JOIN Country c2
        ON c2.IdCountry = cc.IdCountry
    LEFT JOIN Currency cu
        ON cu.IdCurrency = cc.IdCurrency
ORDER BY AgentCode,
         CASE
             WHEN @Type = 2 THEN
                 CASE
                     WHEN ISNULL(c.CountryCode, '') = 'HTI' THEN
                         'HAI'
                     WHEN ISNULL(c.CountryCode, '') = 'PRY' THEN
                         'PAR'
                     ELSE
                         ISNULL(c.CountryCode, '')
                 END
             WHEN @Type = 3 THEN
                 CASE
                     WHEN ISNULL(t.IdCountryCurrency, 0) != 0 THEN
                         CASE
                             WHEN ISNULL(c2.CountryCode, '') = 'HTI' THEN
                                 'HAI' + '/' + ISNULL(cu.CurrencyCode, '')
                             WHEN ISNULL(c2.CountryCode, '') = 'PRY' THEN
                                 'PAR' + '/' + ISNULL(cu.CurrencyCode, '')
                             ELSE
                                 ISNULL(c2.CountryCode, '') + '/' + ISNULL(cu.CurrencyCode, '')
                         END
                     ELSE
```

```
            ''
      END
   ELSE
      ''
END;"
```