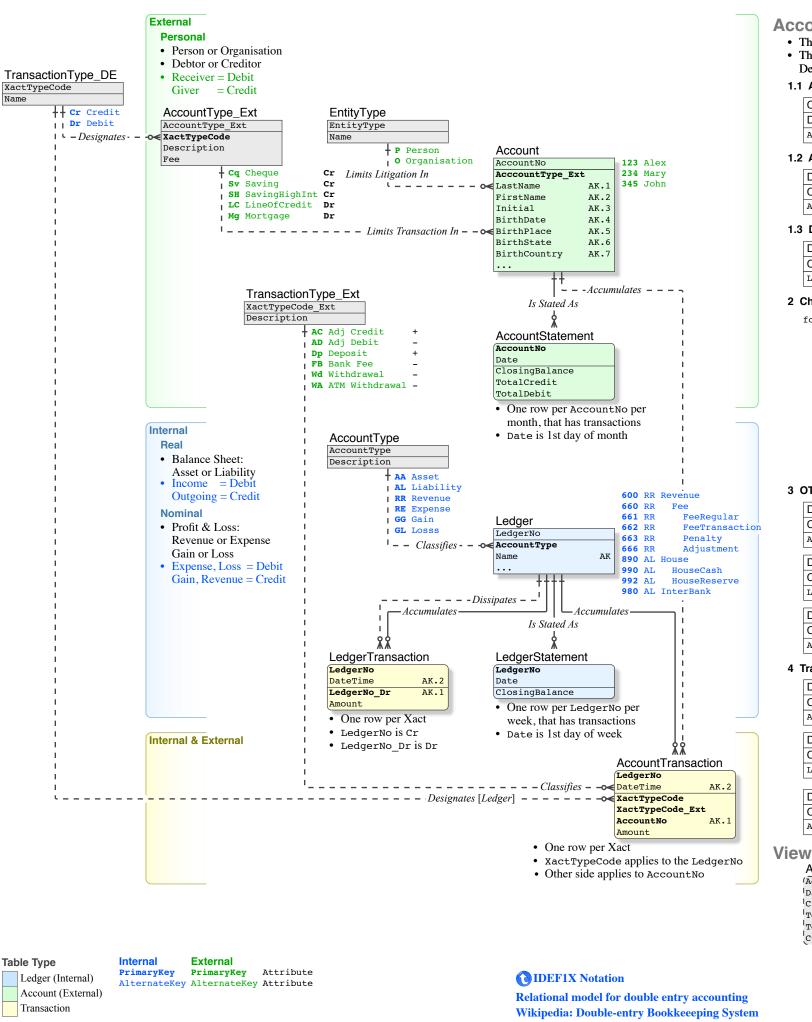
# **Double Entry Account**





## **Accounting Transaction**

- These are the steps required for the requested operations.
- They are given as business Transactions; the DEA Credit/ Debit pair, followed by the row.

#### 1.1 Alex Deposits \$50

Credit	50	123	Alex					
Debit	50	990	Liability	HouseCash				
Account	Kact+	990	2019.12.23	Dr Dp 123		50		

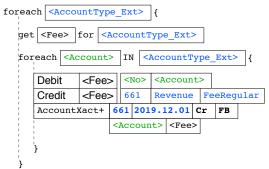
#### 1.2 Alex Withdraws \$75

Deb	oit	75	123	Alex					
Cre	dit	75	990	Liability	HouseCash				
Acc	ount	Xact+	990	2019.12.23	Cr Wd 123		75	]	

#### 1.3 Day End Procedure (Batch Job, One Task)

Debit	456000	990	Liability	Hous	ouseCash ouseReserve		
Credit	456000	992	Liability	Hous			
LedgerXa	act+	992	2019.12.23	990	456000	Γ	

### 2 Charge Monthly Fee (Batch Job, One Task)

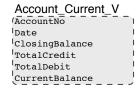


#### 3 OTC Transaction · International Transfer \$500

Debit	500	234	Mary						
Credit	500	992	Liability	HouseReserve			9		
Account	Kact+	992	2019.12.23	Cr Wd 234		234	50	00	
Debit	500	992	Liability	HouseReserve			9		
Credit	500	980	Liability	Interbank					
LedgerXact+		980	2019.12.23	992	500	)			
						_			
Debit 30		234	Mary						
Credit	30	662	Revenue	Fee	rans	sacti	on		
AccountXact+		662	2019.12.23	Cr	Wd	234	3	30	

### 4 Transfer \$100

Debit	100	234	Mary							
Credit	100	992	Liability	HouseReserve			:			
Account	(act+	992	2019.12.23	Cr Wd 234		100				
Debit	100	992	Liability	HouseReserve			!			
Credit	100	990	Liability	HouseCash						
LedgerXact+		990	2019.12.23	992	100	0				
Debit	100	990	Liability	HouseCash						
Credit	100	345	John							
Account	(act+	990	2019.12.23	Dr	Dp	345	100			



Account\_Month\_V OpeningDate OpeningBalance TotalCredit TotalDebit

```
SQL
CREATE VIEW Account Current V
 SELECT AccountNo,
    Date = DATEADD( DD, -1, GETDATE() ),
                                             -- show /as of/ previous day
    ASS.ClosingBalance,
                                              -- 1st of this month
    TotalCredit = (
         SELECT SUM( Amount )
            FROM AccountTransaction ATT
            WHERE ATT.AccountNo = ASS.AccountNo
                 AND XactTypeCode_Ext IN ( "AC", "Dp" )
                 -- >= 1st day of this month yy.mm.01 /AND <= current date/
                 AND DateTime >= CONVERT( CHAR(6), GETDATE(), 2 ) + "01"
    TotalDebit =
         SELECT SUM( Amount )
            FROM AccountTransaction ATT
            WHERE ATT.AccountNo = ASS.AccountNo
                 AND XactTypeCode_Ext NOT IN ( "AC", "Dp"
                AND DateTime >= CONVERT( CHAR(6), GETDATE(), 2 ) + "01"
    CurrentBalance = ClosingBalance +
         <TotalCredit> - -- subquery above
                         -- subquery above
         <TotalDebit>
     FROM AccountStatement ASS
                                             -- 1st day of this month
     WHERE ASS.Date = CONVERT( CHAR(6), GETDATE(), 2 ) + "01"
 CREATE PROC Account_Withdraw_tr (
     @AccountNo.
     @Amount
     ) AS
     IF EXISTS ( SELECT 1
                                                -- validate before verb
            FROM AccountCurrent V
             WHERE AccountNo = @AccountNo
                 AND CurrentBalance >= @Amount -- withdrawal is possible
         BEGIN
         SELECT @LedgerNo = LedgerNo
            FROM Ledger
            WHERE Name = "HouseCash'
         BEGIN TRAN
         INSERT AccountTransaction
            VALUES ( @LedgerNo, GETDATE(), "Cr", "Wd", @AccountNo, @Amount )
         COMMIT TRAN
         END
 CREATE PROC Account_Deposit_tr (
     @AccountNo.
     @Amount.
     ) AS
    ... IF EXISTS, etc ...
                                              -- validate before verb
         SELECT @LedgerNo ...
         BEGIN TRAN
         INSERT AccountTransaction
            VALUES ( @LedgerNo, GETDATE(), "Dr", "Dp", @AccountNo, @Amount )
         COMMIT TRAN
 CREATE PROC Ledger_Xact_tr (
     @LedgerNo, -- Credit Ledger Account
     @LedgerNo_Dr, -- Debit Ledger Account
     @Amount
    ) AS
    ... IF EXISTS, etc ...
         BEGIN
         SELECT @LedgerNo ...
         BEGIN TRAN
         INSERT LedgerTransaction
            VALUES ( @LedgerNo, GETDATE(), @LedgerNo_Dr, @Amount )
         COMMIT TRAN
        END
CREATE PROC Account_MonthEnd_btr ( ... )
 ... begin loop
 ... batch transaction control (eg. 500 rows per xact), etc ...
 INSERT AccountStatement
    SELECT ACT.AccountNo,
```

CONVERT( CHAR(6), GETDATE(), 2 ) + "01", -- 1st day THIS month

AMV.ClosingBalance.

JOIN Account Month V AMV

ON ACT.AccountNo = AMV.AccountNo

WHERE AMV.OpeningDate = DATEADD( MM, -1, ACT.Date )

AMV.TotalCredit,

AMV.TotalDebit

... batch transaction control, etc ...

FROM Account ACT

... end loop

-- follow link for code

-- 1st day PREVIOUS month

-- for PREVIOUS month