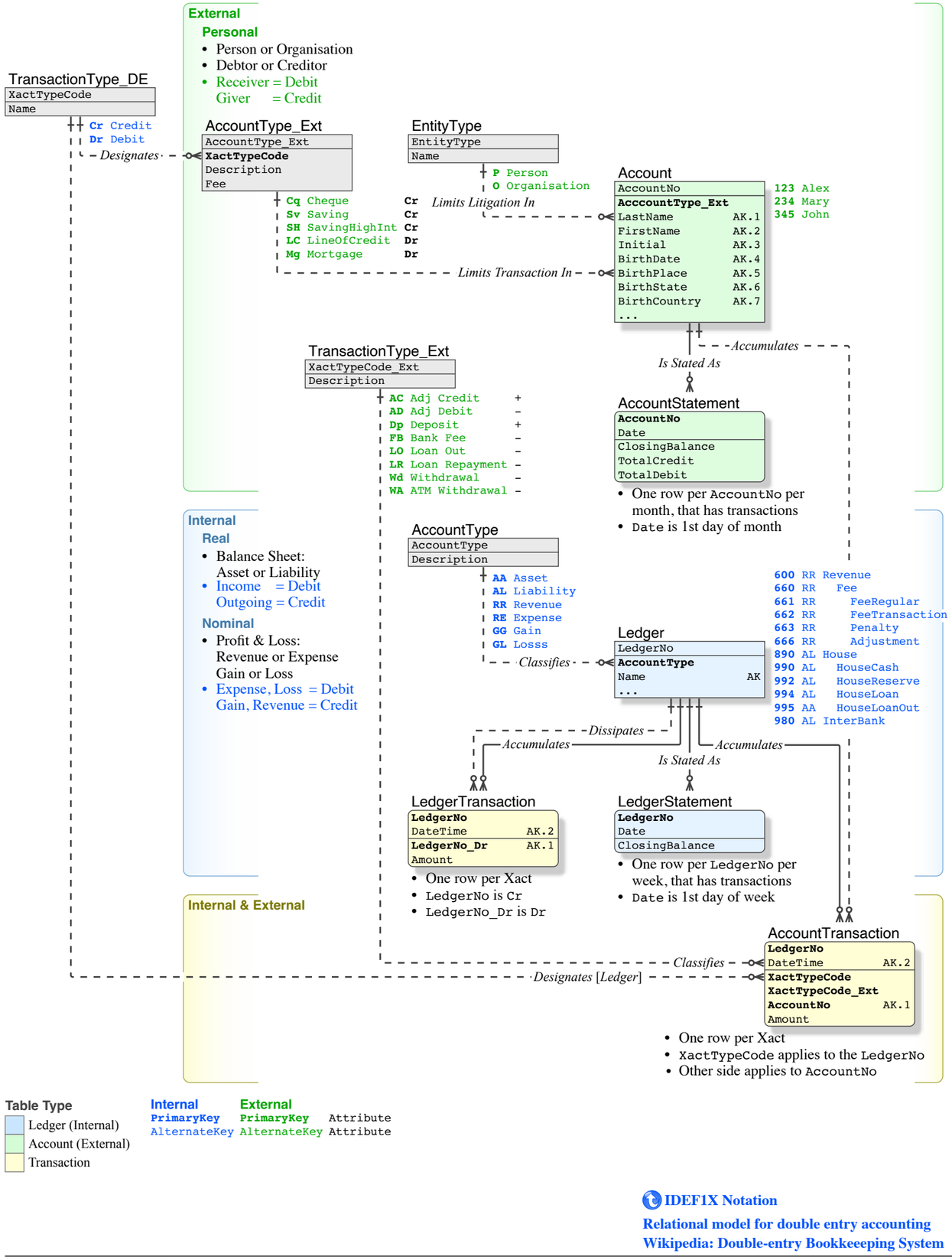
Double Entry Account



IDEF1X Notation

Relational model for double entry accounting

Wikipedia: Double-entry Bookkeeping System

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			
AccountXact+	990	2019.12.23	Dr	Dp	123	50	

1.2 Alex Withdraws $75

Debit	75	123	Alex				
Credit	75	990	Liability	HouseCash			
AccountXact+	990	2019.12.23	Cr	Wd	123	75	

1.3 Day End Procedure (Batch Job, One Task)

Debit	456000	990	Liability	HouseCash	
Credit	456000	992	Liability	HouseReserve	
LedgerXact+	992	2019.12.23	990	456000	

2 Charge Monthly Fee (Batch Job, One Task)

```
foreach <AccountType_Ext> {
    get <Fee> for <AccountType_Ext>
    foreach <Account> IN <AccountType_Ext> {
        Debit  <Fee>  <No>  <Account>
        Credit <Fee>  661   Revenue  FeeRegular
        AccountXact+  661  2019.12.01  Cr  FB
                      <Account>  <Fee>
    }
}
```

3 OTC Transaction · International Transfer $500

Debit	500	234	Mary				
Credit	500	992	Liability	HouseReserve			
AccountXact+	992	2019.12.23	Cr	Wd	234	500	

Debit	500	992	Liability	HouseReserve	
Credit	500	980	Liability	Interbank	
LedgerXact+	980	2019.12.23	992	500	

Debit	30	234	Mary				
Credit	30	662	Revenue	FeeTransaction			
AccountXact+	662	2019.12.23	Cr	Wd	234	30	

4 Transfer $100

Debit	100	234	Mary				
Credit	100	992	Liability	HouseReserve			
AccountXact+	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	

Debit	100	990	Liability	HouseCash			
Credit	100	345	John				
AccountXact+	990	2019.12.23	Dr	Dp	345	100	

View

Account_Current_V

AccountNo
Date
ClosingBalance
TotalCredit
TotalDebit
CurrentBalance

Account_Month_V

AccountNo
OpeningDate
OpeningBalance
TotalCredit
TotalDebit

SQL

```
CREATE VIEW Account_Current_V
    AS
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
        <TotalDebit>       -- subquery above
    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
        BEGIN
        SELECT @LedgerNo ...
        BEGIN TRAN
        INSERT AccountTransaction
            VALUES ( @LedgerNo, GETDATE(), "Dr", "Dp", @AccountNo, @Amount )
        COMMIT TRAN
        END
```

```
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 ( ... )
    AS
... 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,                       -- for PREVIOUS month
            AMV.TotalCredit,
            AMV.TotalDebit
        FROM Account ACT
            JOIN Account_Month_V AMV              -- follow link for code
                ON ACT.AccountNo = AMV.AccountNo
                                                  -- 1st day PREVIOUS month
        WHERE AMV.OpeningDate = DATEADD( MM, -1, ACT.Date )
... end loop
... batch transaction control, etc ...
```