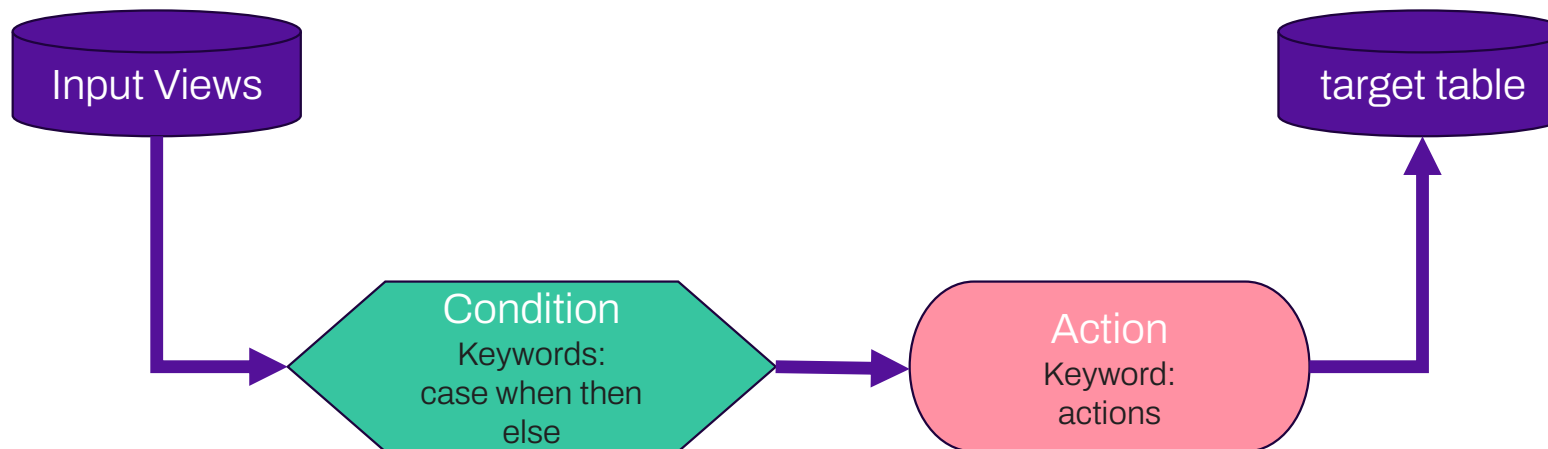


Business Rules (15)

Business rules consist of 4 components

- » Input: Which input values are required for the calculation?
- » Condition: When should the calculation be executed?
- » Action: How should the calculation be performed?
- » Output: What are the result values and where are they stored?



Business Rules (15)

Example of defining a simple input view for a business rule

- » de facto, the modeler only specifies which tables / fields are required as input for the business rule
- » the necessary join conditions for the tables, taking into account the bitemporal history, are automatically added by the DVG generator
- » If required, the modeler can also use aggregations, database functions, where clauses, business join conditions, etc. in the input view (not included in the example)
- » the creation of input views is optimally supported by templates, code completion and validation messages
- » the structure of the metadata is based on SQL, as data modelers are generally experienced to working with SQL

```
view PositionsKpi simple
from hub VaultWebshop.HubPosition alias POS {
  satellites {
    SatPosition alias POSsat;
  }
  joins {
    link VaultWebshop.LinkPosition_Bestellung alias LBest {
      joins {
        hub VaultWebshop.HubBestellung alias Best {
          satellites {
            SatBestellung alias Bestsat;
          }
        }
      }
    }
    link VaultWebshop.LinkLieferung alias LLiefer {
      satellites {
        SatLieferung alias Liefersat;
      }
    }
  }
}
attributes {
  POS.HashKey alias HashKey primaryKey;
  POSsat.Menge alias Menge any_value;
  POSsat.Preis alias Preis any_value;
  Bestsat.Wunschdatum alias Wunschdatum any_value;
  Bestsat.Rabatt alias Rabatt any_value;
  Liefersat.Lieferdatum alias Lieferdatum any_value;
}
;
```



Business Rules (15)

Input view
(see previous page)

Output table

```
rule PositionsKpi source Berechnung.PositionsKpi target Businessvault.PositionKpi {  
  block "Erlös"  
    actions {  
      HashKey -> HashKey;  
      Preis * Menge * (1 - coalesce(RabattBestellung,0) : 100) * (1 - coalesce(RabattPosition,0) : 100) -> Erlös;  
    }  
  block "Lieferung erfolgt?"  
    case  
      when isNull(Lieferdatum) then  
        actions {  
          1 -> offenePosition;  
          0 -> geliefert;  
        }  
      else  
        actions {  
          0 -> offenePosition;  
          1 -> geliefert;  
          daydiff(Wunschdatum,Lieferdatum) -> DifferenzLieferdatum;  
        }  
    case  
      when SourceArt = "Road" then  
        actions {  
          target.Erlös * 0,98 -> Erlös;  
        }  
      else  
        actions {  
        }  
    }  
}
```

First calculation block, which is
always executed

Formula
editor

Second calculation block with
conditions

Delivery not yet made

Delivery has been made

Database function
used

Example of a business rule definition

- » the modeler can use a formula editor and all database functions as part of the conditions/calculations
- » the creation of input views is optimally supported by templates, code completion and validation messages

