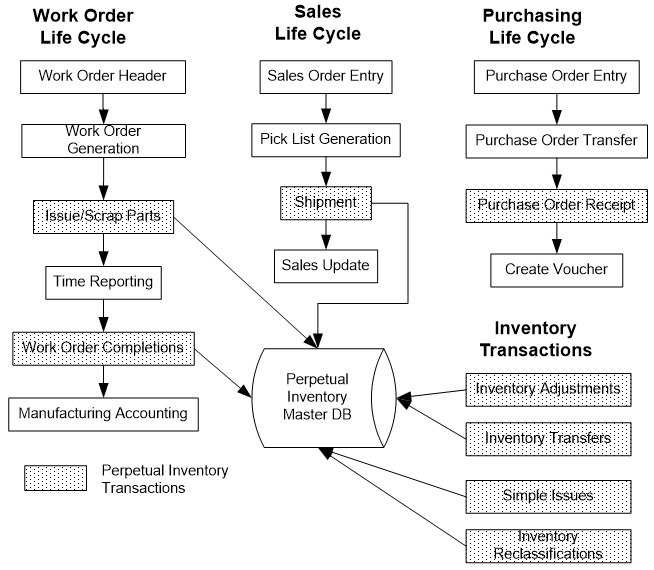
# **Inventory Data Warehouse**

The Inventory Data Warehouse provides data to answer business intelligence questions about inventory transaction cycles. This document provides background on inventory transaction concepts, details about the snowflake schema to support inventory transaction cycles, and a data dictionary about the table design for the data warehouse.

## ***1. Inventory Transaction Cycles***

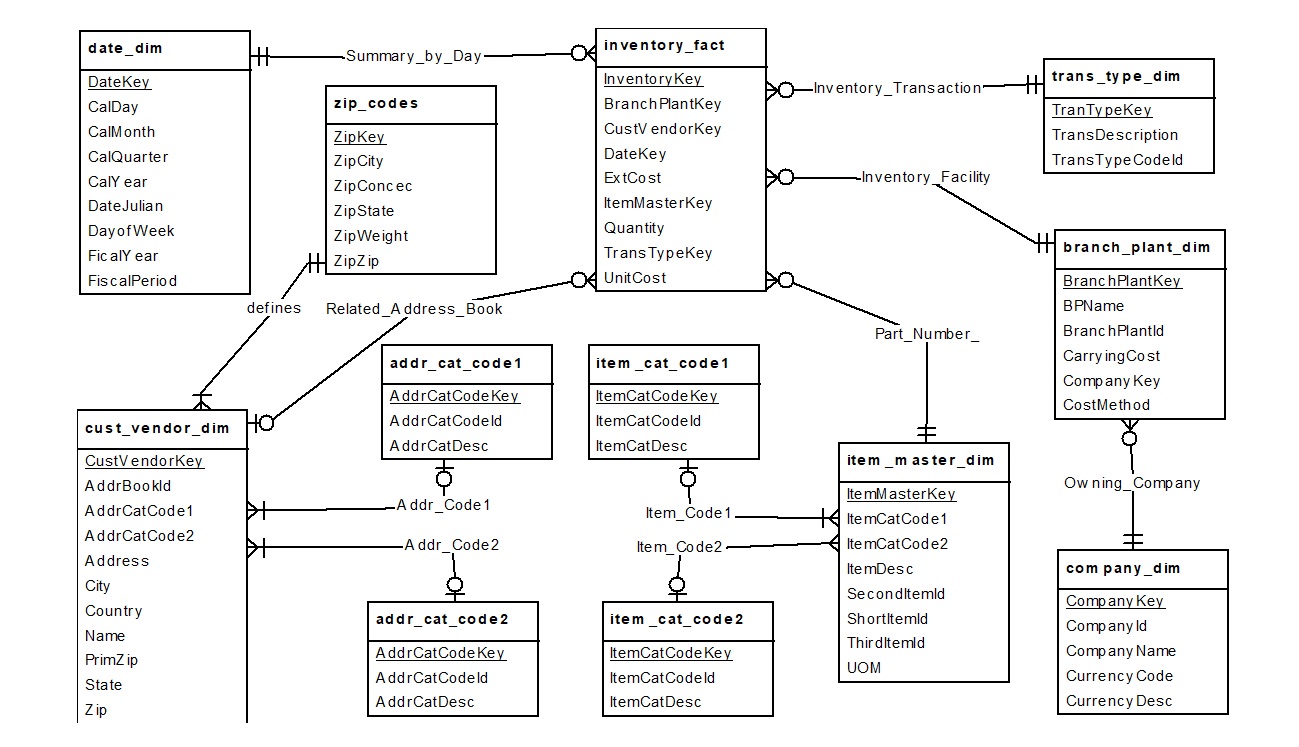
Inventory that is bought, sold, consumed, and produced is the heart of any manufacturing and/or distribution company. Inventory transactions are frequent and commonplace. The volume and significance of inventory transactions make them important in a data warehouse design.Typically, Enterprise Resource Planning (ERP) software provides modules related to Manufacturing, Distribution/Logistics, Financials, and HR/Payroll. The work order, sales, and purchase life cycles affect the perpetual inventory balance as shown in Figure 1. In addition, inventory transactions including adjustments, transfers, issues, and reclassifications affect the perpetual inventory balance.



**Figure 1:** Life Cycles Affecting the Perpetual Inventory Balance

## ***2. Snowflake Schema Description***

To support reporting about inventory management, Figure 2 shows a snowflake schema for the perpetual inventory balance. The snowflake schema provides a template that can be customized to individual organizations. Dimension entity types such as *Addr\_Cat\_Code1* allow an organization to customize the design to specific requirements. The fact entity type, *Inventory\_Fact*, contains several measures along with relationships to associated dimension entity types. Several dimension entity types are related directly to the *Inventory\_Fact* entity type. Other dimension entity types such as *Item\_Cat\_Code1* are indirectly related to the *Inventory\_Fact* entity type.



**Figure 2:** Generic Snowflake Schema for the Perpetual Inventory Balance

The ERD representation of the snowflake schema is converted to a table design using the normal conversion rules. In the conversion process, 1-M relationships convert to foreign keys in the child tables. Appendix A contains a data dictionary for the table design.

## ***Appendix A: Data Dictionary for the Snowflake Schema Table Design***

Appendix A contains a brief description of each column in the tables of the Inventory Data Warehouse Schema.

## ***Address Category 1 Table (addr\_cat\_code1)***

This table defines address category codes related to customers/vendors. These codes allow customers/vendors to be grouped. Example grouping might be customer type, customer area, etc.

AddrCatCodeKey Unique primary key value

AddrCatCodeId Four character category code

AddrCatDesc Thirty character category code description

## ***Address Category 2 Table (addr\_cat\_code2)***

This table defines address category codes related to customers/vendors. These codes allow customers/vendors to be grouped. Example grouping might be customer type, customer area, etc.

AddrCatCodeKey Unique primary key value

AddrCatCodeId Four character category code

AddrCatDesc Thirty character category code description

## ***Item Category 1 Table (item\_cat\_code1)***

This table defines item master category codes related to item masters (parts) These codes allow part numbers to be grouped. Example grouping might be product class, spare part, finished good, etc.

ItemCatCodeKey Unique primary key value

ItemCatCodeId Four character category code

ItemCatDesc Thirty character category code description

## ***Item Category 2 Table (item\_cat\_code2)***

This table defines item master category codes related to item masters (parts) These codes allow part numbers to be grouped. Example grouping might be product class, spare part, finished good, etc.

ItemCatCodeKey Unique primary key value

ItemCatCodeId Four character category code

ItemCatDesc Thirty character category code description

## ***Zip Codes Table (zip\_codes)***

This table provides the basis to create many unique customer records for a variety with a variety of zip codes.

ZipKey Primary Key, user defined.

ZipCity City related to zip code

ZipState State related to zip code

ZipZip Zip Code

ZipConsec The zip code plus this number define the range of zip codes for this city

ZipWeight The weight (percentage \* 100) that will be applied to creating customers. All ZipWeight columns totaled should equal 100.

## ***Date Sequence Table (date\_dim)***

This table provides the date pattern. Date patterns can be daily, five days per week, weekly or monthly.

DateKey Unique primary key value

CalDay Calendar day from 1 to 31.

CalMonth Calendar month from 1 to 12

CalQuarter Calendar quarter from 1 to 4

CalYear Calendar year valid for ranges from 1900 to 2100

DateJulian Date in the form of YYYYDDD. Where YYYY is the year and DDD is the sequential date.

DayOfWeek Day of the week, 1 to 7, 1 is Sunday, 2 is Monday, etc

FiscalYear Corresponding Fiscal Year

FiscalPeriod Corresponding Fiscal Period

## ***Transaction Type Table (trans\_type\_dim)***

This table defines the various types of inventory transactions. Examples include transfers, adjustments, shipments, receipts, etc. Some of the codes may not be used in the sample data for the trans\_type\_dim table.

TransTypeKey Primary Key, coded to the following values.

TransTypeId =1 then inventory adjustment (IA)

TransTypeId =2 then inventory transfer (IT)

TransTypeId =3 then inventory simple issue (IS)

TransTypeId =4 then purchase order receipt (OV)

TransTypeId =5 then sales order shipment (AR)

TransTypeId =6 then mfg issue (IM)

TransTypeId =7 then mfg completion (IC)

TransTypeId =8 then mfg parent scrap (IS)

TransTypeId =9 then mfg component scrap (IZ)

TransTypeCodeId Corresponding One World cardex code, an example is IA for an adjustment.

TransDescription Transaction Type Description

## ***Customer Vendor Table (cust\_vendor\_dim)***

This table defines possible customers and vendors involved with related sales and purchasing related transactions.

CustVendorKey Unique primary key value

AddrBookId One World related address book number

Name Customer Name

Address Address

City

State

PrimZip Integer form of the zip code

Zip Zip code that could be in various forms (nnnnn, nnnnn-nnnn, etc)

Country Country

AddrCatCode1 OneWorld related Category code, foreign key to the address category code 1

AddrCatCode2 OneWorld related Category code, foreign key to the address category code 2

## ***Item Master Table (item\_master\_dim)***

This table defines item masters (ie part numbers).

ItemMasterKey Unique primary key value

ShortItemId OneWorld related short item id

SecondItemId OneWorld related 2nd item number

ThirdItemId OneWorld related 3rd item number

ItemCatCode1 OneWorld related category code, foreign key to the item category code1 table

ItemCatCode2 OneWorld related category code, foreign key to the item category code2 table

ItemDesc OneWorld related item master description

UOM OneWorld related primary unit of measure

## ***Company Table (company\_dim)***

This table contains company records including the base currency.

CompanyKey Unique primary key value

CompanyId OneWorld related 5 character company id

CompanyName OneWorld related company name

CurrencyCode OneWorld related currency code

CurrencyDesc OneWorld related currency description

## ***Branch Plant Table (branch\_plant\_dim)***

This table contains the Branch Plant information.

BranchPlantKey Unique primary key value

BranchPlantId JDE related Branch Plant Id (12 character MCU)

CompanyKey Owning company for this branch, foreign key to Company table.

CarryingCost Carrying Cost percentage defined as a decimal

CostMethod OneWorld related Cost Method.

BPName OneWorld related Branch Plant Name

## ***Inventory Transaction Fact Table (inventory\_fact)***

This table contains the inventory transactions facts. Integer keys are used to help limit the size of the rows. The measures are unit cost, quantity, and extended cost.

InventoryKey Unique primary key value

BranchPlantKey Transaction Branch, Foreign key to the branch plant table

DateKey Transaction Date, foreign key to the date table

ItemMasterKey Transaction Part Number, foreign key to the item master table

TransTypeKey Transaction Type, foreign key to the transaction type table

CustVendorKey Optional address book key that is a foreign key to the customer vendor table. This column allows null values. The column is not null only on sales and purchasing transactions.

UnitCost Unit cost with up to 4 decimals of precision

Quantity Quantity with up to 4 decimals of precision

ExtCost Extended Cost with up to 2 decimals of precision