Mobis Alabama, LLC is a Tier-1 supplier specializing in the manufacturing of automotive components for Hyundai Motors and Kia Motors. Mobis maintains their inbound and outbound inventory in vertical warehouse in the outskirts of Alabama under their supervision. This vertical warehouse is used to store raw material, components that are procured from various vendors to be used later at their plant, and to store the finished goods produced at the plant for shipment to its customers.

The different modules of *VertX* software used to process data related to a vertical warehouse include:

1. Warehouse Layout Management Module - To manage the Layout and Workflow within the warehouse to optimize space utilization.
2. Receiving & Shipping Module - To track incoming and outgoing shipments. It also covers raw material inspection of inbound inventory.
3. Inventory Management Module - To manage storage space allocation along with inventory movement and maintenance.
4. Real-time Location System (RTLS) Module - To track the location of items and optimize the flow of goods.
5. Order Management Module - To manage data related to storage-orders from various departments like procurement, production, and sales & distribution department. It involves tracking the status of orders and ensuring that they are fulfilled on time.
6. Quality Control Module: To manage quality control activities related to inbound raw materials & components procured by purchases department as well as semi-finished goods and outbound finished goods manufactured by the production department.
7. Barcode/QR Code Scanning Module - To improve accuracy and speed when scanning inventory items. Data of equipment related to scanning are handled in this module.
8. Analytics and Reporting Module - To provide insights into inventory levels, order fulfillment, and warehouse operations.
9. Equipment Management Module - To manage the data related to various equipment (Inventory movement equipment, support equipment, safety equipment and others) and logs related to equipment maintenance, damages and repairs are catered to in this module.
10. User Management Module – To create and maintain application users with different roles and manage appropriate authorizations.
11. Safety Drills, Safety Training Programs & Accident Reporting Module – To manage data related to various safety programs on which employees have to be regularly educated. Training programs are to be held, and employees have to attend and obtain necessary training & certification. Data related to Accident reporting and corrective actions taken are also processed.

|  |
| --- |
| Project Description in 4 different ways for VertX – Equipment Management Module:   1. VertX – Equipment Module deals with processing data related to various warehouse equipment viz., Inventory movement equipment, support equipment, safety equipment and others) that are used in a Vertical Warehouse, along with equipment maintenance, damages, and repair related data. 2. Equipment Module of VertX application process data related to diverse equipment, including Inventory movement equipment, support equipment, safety equipment, and more which are used in a Vertical Warehouse. This module also deals with logging of data related to equipment maintenance, damage, and repairs. 3. VertX is an application that manages Vertical Warehouse data. The Equipment Module provides seamless data processing of various equipment categories such as Inventory movement, support, safety, and other types of equipment. It also facilitates efficient management of equipment maintenance, damage, and repairs through logging capabilities. 4. VertX application's Equipment Module processes data of Inventory movement, support, and safety gear equipment for Vertical Warehouses. It also provides features for logging of maintenance, damage, and repair information. |

*VertX-Equipment Module* tables are broadly classified into below categories:

1. Equipment related tables
2. Maintenance task master data
3. Spare Parts master data (including suppliers of spare parts)
4. Logs which include maintenance logs, damage/malfunction logs and repair/replacement logs

**EQPT\_CATG** (Equipment Category)

|  |  |
| --- | --- |
| Column | Notes |
| ***catg\_id*** | Integer, PK, Use a sequence with a seed value of 101 |
| catg\_name | Varchar(50), Not NULL, Unique |
| notes | Varchar(150) |

**EQPT\_TYP** (Equipment Type is essentially sub-categories of equipment categories stored in EQPT\_CATG table)

|  |  |
| --- | --- |
| Column | Notes |
| typ\_id | Integer, PK, Use a sequence with a seed value of 1001 |
| eqpt\_catg\_id | Integer, Not NULL, FK references EQPT\_CATG.catg\_id |
| typ\_name | Varchar(50), Not NULL, Unique |
| mntc\_freq | Integer, Maintenance frequency in days (120 days or 180 days etc), Nullable [Some equipment like safety boots, glasses, hard hats, gloves do not have any maintenance activities] |
| var\_exist | Boolean. True if variants exist for the equipment type. In most cases this will be true. False if variants do not exist. |
| notes | Varchar(200) |

**NOTE**: Maintenance frequency is maintained at equipment type level.

**EQPT\_VAR** (Equipment Variant is further categorization of an equipment sub-category)

|  |  |
| --- | --- |
| Column | Notes |
| var\_id | Integer, PK, Use a sequence with a seed value of 10001 |
| eqpt\_typ\_id | Integer, Not NULL, FK references EQPT\_TYP.typ\_id |
| var\_name | Varchar(50), Not NULL, Unique |
| notes | Varchar(300) |

**EQPT\_VEND** (Vendor master of equipment sellers / distributors)

|  |  |
| --- | --- |
| Column | Notes |
| vend\_id | Integer, PK, Serial |
| vend\_reg\_name | Varchar(50), Not NULL, Unique, Registered name of vendor |
|  | Add address related fields |

**NOTE:** Equipment Vendor table design is kept simple intentionally for our requirement.

**EQPT** (Equipment instance)

|  |  |
| --- | --- |
| Column | Notes |
| ***eqpt\_id*** | Varchar(12), Serial |
| eqpt\_name | Varchar(50), Not NULL Unique (Eg: “Forklift #183A/22”) |
| eqpt\_typ\_id | Integer, Not NULL, FK referring EQPT\_TYP.typ\_id |
| eqpt\_var\_id | Integer, Nullable, FK referring EQPT\_VAR.var\_id |
| oem | Varchar(50), Not NULL, name of Original Equipment Manufacturer |
| eqpt\_vend\_id | Integer, Not NULL, FK referring EQPT\_VEND.vend\_id |
| mfr\_model | Varchar(50), Not NULL, Manufacturer Model number |
| load\_cpcty | Integer, Nullable, Load bearing capacity of the equipment, applicable in most case but not all. |
| load\_cpcty\_uom | Varchar(5), Nullable, Applicable when load capacity is specified, Only two possible values (lbs or ton – Pound or Ton) |
| mfg\_dt | Manufacturing date |
| pur\_dt | Date, Purchase date |
| wrty\_exp\_dt | Date, Warranty expiry date |
| wrty\_doc\_link | Varchar(255), Link to warranty document stored as a PDF or TIFF on the application server |
| Status | Varchar(50), possible values (“Operational”, “Under Planned Maintenance”, “Under Unplanned Maintenance” and “Out of Service”)  “Under Unplanned Maintenance” implies the equipment is unusable due to damage/malfunction.  “Out of Service” implies - The equipment is permanently taken out of service and is marked for resale or scrapping. |
| eqpt\_notes | Varchar(500) |
| loc\_id | Varchar(20), Location ID will have FK reference to location information in the warehouse which is based on the warehouse layout data. Though the provision for this field is made, processing this data is not in the scope of current requirement. This can be processed only when warehouse layout master data is available. |

**Search Functionality:**

Provide search functionality of equipment with below search parameters:

1. Equipment name
2. Equipment category / type / variant
3. Load capacity & UoM
4. OEM
5. Vendor
6. Status
7. Date ranges:
   1. Manufacturing date range
   2. Purchase date range
   3. Warranty expiry date range

*Equipment Vendor vs Original Equipment Manufacturer (OEM):* For example, DELL servers are sold in India by Redington Group (<https://redingtongroup.com/india/delltechnologies/>). In this case DELL is the OEM (Original Equipment Vendor) and Redington is the Equipment Vendor. Same is the case with Mac Books, Redington is an equipment vendor and Apple Inc. is the OEM.

**EQPT\_MNTC\_TASK** (Master data of maintenance tasks related to Equipment Type or Equipment Variant)

|  |  |
| --- | --- |
| Column | Notes |
| ***task\_id*** | Integer, PK, Use sequence with a seed value of 1001 |
| task\_name | Varchar(50), Not NULL, Unique |
| eqp\_typ\_id | Integer, Not NULL, FK referring EQPT\_TYP.typ\_id |
| eqp\_var\_id | Integer, Nullable, FK referring EQPT\_VAR.var\_id |
| task\_desc | Text, Not NULL, descriptive information about what kind of maintenance activities are to be performed for the task. |

NOTE: In the table EQPT\_MNTC\_TASK if the column eqp\_var\_id is NULL then such task is applicable to all variants of the equipment type. On the other hand, if eqp\_var\_id has a value; then such task is applicable to that variant only.

Notes about Maintenance and Repairs & Part Replacements:

The spare part replacements for an equipment can arise for 2 possible reasons:

1. During periodic maintenance activity if it is noticed that certain parts have undergone too much wear and tear, then part replacements may become necessary.
2. If equipment is malfunctioning or has undergone damage due to an accident, then also part replacement may become necessary.
3. Logs are to be maintained separately for equipment maintenance and for equipment damage/malfunctioning. Logs will have to be maintained for part replacement also.
4. To ascertain the cost of part that has been replaced in an equipment different costing methods are used for different parts. The different costing methods used will include:
   1. **FIFO (First In, First Out) Method:** The FIFO (First-In, First-Out) costing method assumes that the oldest spare parts you bought are the ones you sell first. This means the cost of the parts initially purchased is what gets assigned to the parts issued from inventory.
   2. **SIM (Specific Identification Method):** The Specific Identification Method is the most precise way to determine the unit cost of a spare part issued from inventory, but it requires more detailed record-keeping. When issuing a spare part, you identify it by its unique identifier and reference the recorded cost associated with that specific part. This becomes the unit cost for the issued spare part.
   3. **Standard Cost Method:** The standard cost method is a costing system that uses predetermined costs for parts based on historical data & expected material prices.

**SPARE\_PART**

|  |  |
| --- | --- |
| Column | Notes |
| ***part\_id*** | Integer, PK, Serial |
| part\_name | Varchar(50), Not NULL, Unique – name of the spare part |
| qoh | Integer, Quantity on Hand (computed column based on current qoh, add purchases, deduct replacement usage) |
| uom | Unit of measurement – the uom mentioned here should be used to account purchases and consumption of a part. |
| costing\_method | Varchar(50), possible values include “FIFO Method”, “Specific Identification Method” and “Standard Cost Method” |
| std\_cost | Decimal(15,2), Nullable,  This field though nullable will have to necessarily be captured when the spare parts costing method is “Standard Cost Method” |

NOTE: Costing method can be FIFO or SIM only when the uom if ‘PC’ (pieces). For uom value other than PC it is always “Standard Cost Method”. For uom ‘PC’ “Standard Cost Method” can also be used. Update of costing method should be available only for restricted users.

**SPARES\_SUPPLIER**

|  |  |
| --- | --- |
| Column | Notes |
| ***supplier\_id*** | Integer, PK, Serial |
| supplier\_name | Varchar(50), Not NULL |
|  | NOTE: Add address related fields |

**SPARES\_SUPPLIER\_PARTS** (Junction table between Spare parts and its suppliers)

|  |  |
| --- | --- |
| Column | Notes |
| ***supplier\_id*** | Integer, Composite PK, FK referring SPARES\_SUPPLIER.supplier\_id |
| ***part\_id*** | Integer, Composite PK, FK referring SPARE\_PART.part\_id |
| lead\_time | Integer, Lead time in days by the supplier to supply the spare part |

**SPARES\_PUR\_HIST** (Spare parts purchase history)

|  |  |
| --- | --- |
| Column | Notes |
| ***pur\_id*** | PK, Serial |
| part\_id | Integer, Not NULL, FK referring SPARE\_PART.part\_id |
| oem | Varchar(50), Name of original equipment manufacturer |
| supplier\_id | Integer, Not NULL, FK referring SPARES\_SUPPLIER.supplier\_id |
| Invoice\_id | Varchar(30), Not NULL  Supplier\_id & invoice\_id combination cannot get duplicated |
| part\_no | Varchar(50), Nullable, ID maintained by the supplier for this spare part |
| uid | Varchar(50), Nullable, unique number (could be alpha numeric) that is given by each manufacturer to identify each instance of the spare part. For a given manufacturer, this number is guaranteed to be unique.  There should be a composite unique key on the columns mfr & uid.  This field though nullable will have to necessarily be captured when the spare parts costing method is “Specific Identification Method”. Such purchase quantity of such parts will always have to be 1 for each purchase. |
| qty | Integer, Not NULL, Quantity purchased |
| unit\_price | Decimal(15,2), Not NULL |

NOTE: uom of purchase is applicable as captured in the SPARE\_PART table.

Part Number (part\_no) vs Unique ID (uid) example:

*AP-BTXSL-1223* is a number that is given by Amara Raja Batteries Ltd. for car batteries of a particular model. In a way it is the model number.­­­­

But each piece of battery under this mode will carry an unique number based on which the warranty card will be issued. The uid could be something like *EAB0252X208004-ABR-PR-12APBTX50*

**MNTC\_LOG\_H** (Maintenance Log Header)

|  |  |
| --- | --- |
| Column | Notes |
| ***log\_id*** | Integer, PK, Serial |
| eqpt\_id | Integer, Not NULL, FK referring EQPT.eqpt\_id |
| Created\_on |  |
| Created\_by | User id of the user who created this maintenance log |
| Start\_dt |  |
| Closure\_dt | Should be NULL until the maintenance log is closed. When the closure date is captured, it implies closure of the log. |
| Closed\_by | User id of the user who closes the maintenance log |
| Next\_due\_dt | Date. Editable only after closure of the maintenance log. System should suggest a value on closure by adding maintenance frequency no.of days to closure date, which the end user can edit. |
| Log\_notes | Text (remarks about the maintenance log, which also includes info related to part replacements suggested) |
| Mntc\_method | Varchar(50), Possible values (Inhouse, Manufacturer, Third Party) |
| Mntc\_inv\_doc | Link to uploaded PDF or TIFF document which is the invoice received when the maintenance is done by manufacturer or third party. Not applicable in case of Inhouse maintenance |
| Repl\_log\_id | Integer, nullable, FK with REPL\_LOG\_H.log\_id. Applicable when there are part replacements of the equipment in case of inhouse maintenance |
| Status | Varchar(50), Not NULL, Statuses include (New, WIP, Closed, Aborted)  If a replacement log has been raised (i.e, repl\_log\_id column has a value) then only after all the related part replacements are done, can the maintenance task be closed, until then it will remain WIP. Closure\_dt of the maintenance log can also be set only when all replacement request raised have been closed |

NOTE:

When a maintenance log is created for an equipment, the equipment status must be set to “Under Planned Maintenance”, and when maintenance log is being closed, the status can change to either “Operational” or “Out of Service” and appropriate log notes will have to be captured.

Log notes are mandatory when a log is being closed.

**MNTC\_LOG\_I** (Maintenance Log Item: data is populated in this table only for inhouse maintenance)

|  |  |
| --- | --- |
| Column | Notes |
| ***Log\_id*** | Integer, Composite PK, FK with MNTC\_LOG\_H.log\_id |
| ***Task\_id*** | Integer, Composite PK, FK with EQPT\_MNTC\_TASK.task\_id |
| Notes | Text, task notes |
| User\_id | Integer, user\_id of the employee who is heading the inhouse maintenance task. |
| Repl\_notes | Text. Based on the maintenance observations a note will be captured here regarding any part replacements required. Based on this info part replacement requirement has to be created. These are the notes by the user who is raising the replacement request. |

**DMG\_MF\_LOG** (Damage/Malfunction Log - This table will have only header data)

|  |  |
| --- | --- |
| Column | Notes |
| ***log\_id*** | Integer, PK, Serial |
| eqpt\_id | Integer, Not NULL, FK referring EQPT.eqpt\_id |
| Created\_on |  |
| Created\_by | User id of the user who created this damage/malfunction log |
| Start\_dt |  |
| Closure\_dt | Should be NULL until the Damage/Malfunction log is closed. When the closure date is captured, it implies closure of the log. |
| Closed\_by | User id of the user who closes the damage/malfunction log |
| Dmg\_mf\_notes | Text – Notes about the damage and/or malfunctioning of the equipment |
| Repl\_notes | Text – Notes about the suggested part replacements. These are the notes by the user who is raising the replacement request. |
| handling\_method | Varchar(50), Possible values (Inhouse, Manufacturer, Third Party) |
| inv\_doc | Link to uploaded PDF or TIFF document which is the invoice received when the repair is done by manufacturer or third party. Not applicable in case of Inhouse repairs |
| repl\_log\_id | Integer, nullable, FK with REPL\_LOG\_H.log\_id. Applicable when there are part replacements of the equipment in case of inhouse repairs |
| Status | Varchar(50), Not NULL, Statuses include (New, WIP, Closed, Aborted)  If a replacement log has been raised (i.e, repl\_log\_id column has a value) then only after all the related part replacements are done, can the maintenance task be closed, until then it will remain WIP. Closure\_dt of the log can also be set only when all replacement request raised have been closed |

NOTE:

When a damage/malfunction log is created for an equipment, the equipment status must be set to “Under Unplanned Maintenance”, and when maintenance log is being closed, the status can change to either “Operational” or “Out of Service” and appropriate log notes will have to be captured.

Log notes are mandatory when a log is being closed.

**REPL\_LOG\_H** (Replacement Log Header – this could be as result of either Maintenance or Damage/Malfunction)

|  |  |
| --- | --- |
| Column | Notes |
| ***log\_id*** | PK, Serial, Replacement Log ID |
| eqpt\_id | Integer, Not NULL, FK referring EQPT.eqpt\_id |
| repl\_trigger | Varchar(50) , 2 possible values (“Maintenance”, “Damage/Malfunction”) |
| repl\_log\_cl\_dt | Date, Replacement log closure date: The date on which the last of the 1 or more parts required for replacement has been completed and tested for. |

**REPL\_LOG\_I** (Replacement Log Item)

|  |  |
| --- | --- |
| Column | Notes |
| ***log\_id*** | Composite PK, FK referring REPL\_LOG\_H.log\_id |
| ***part\_id*** | Composite PK, FK referring SPARE\_PART.part\_id |
| repl\_dt | Date on which the last unit of the part’s replacement was completed and tested. |
| qty | Integer |
| unit\_cost | Decimal (17,2) - computed at the time of closure of this log item |
| repl\_notes | Text. Notes by the user who is responsible for the actual replacement of the part |
| tech\_id | Technician ID of the person who is responsible for part replacement. A technician can be an in-house employee or a contractor on hire. Maintaining technician data and having a FK relation can be taken up at a later stage |

NOTE:

uom of purchase is applicable as captured in the SPARE\_PART table.

When the replacement date of the last part of the same log is being captured, the corresponding logs replacement closure date will have to be captured with the same date value.

The unit\_cost of part **issued** for a replacement item can be computed in different ways which is captured in costing\_method column of SPARE\_PART table. The unit\_cost calculation will have to be handled in the application logic.