

Integrated Supply Chain and Financial Management System

Project Report – Team 6

December 3rd, 2023,

Course details: MGS 613 – DBMS – Fall 2023

Abstract

The **Integrated Supply Chain and Financial Management System** project aims to develop a comprehensive Enterprise Resource Planning (ERP) Database Schema tailored for a manufacturing and supply chain company. This project endeavors to streamline operations, optimize workflow, and ensure meticulous financial oversight within the company's complex framework. By implementing this system, the goal is to regulate departmental structures, manage material procurement from diverse vendors, and facilitate seamless processing of invoices within the accounting department. This report presents the initial stages of database design and normalization tasks essential for achieving an efficient and robust database schema.

Table of Contents

Abstract	1
Team Information.....	2
Database Design and Normalization Tasks	2
Task 0: Metadata and Business Rules Repository	2
Task 1: Primary and Foreign Keys (EER-Notations)	4
Task 2: EER- Diagrams (Pre-Normalization)	5
Task 3: Relational Schema and Relationships	7
Task 4: Functional Dependencies and Normalization.....	10
Task 5: SQL queries of the database	15

Team Information

-Team Name: - Dynamo

Moderate Team – 4

-Team Members:

1. Keerthana Allam – SQL Developer
2. Satwik CR Jinna – Business Rules Analyst
3. Jhansi Sai Vigesna – Database Designer
4. Sri Pavani Madani – Quality Assurance and Testing Specialist

Database Design and Normalization Tasks

Task 0: Metadata and Business Rules Repository

Existing Business Rules: -

B1: The company has departments, warehouses, and production lines.

B2: The company designs and produces products.

B3: Each department, warehouse, and production line has multiple employees.

B4: Each employee works in only one department, warehouse, or production line.

B5: Some departments design products. Such departments design at least one product.

B6: Different departments must not work on the same product.

B7: Each product must be produced by one specific production line.

B8: Each production line can produce only one product. For repair purposes, production lines may produce no products.

B9: Raw materials are supplied by any number of warehouses, which are supplied by any number of vendors.

B10: Raw materials may also be directly supplied by vendors.

B11: Vendors can supply any number of materials.

B12: Vendors submit invoices if they receive any orders.

B13: Invoices must be processed by the accounting department.

Newly Added Business Rules: -

B1. Each employee must be allocated to a department.

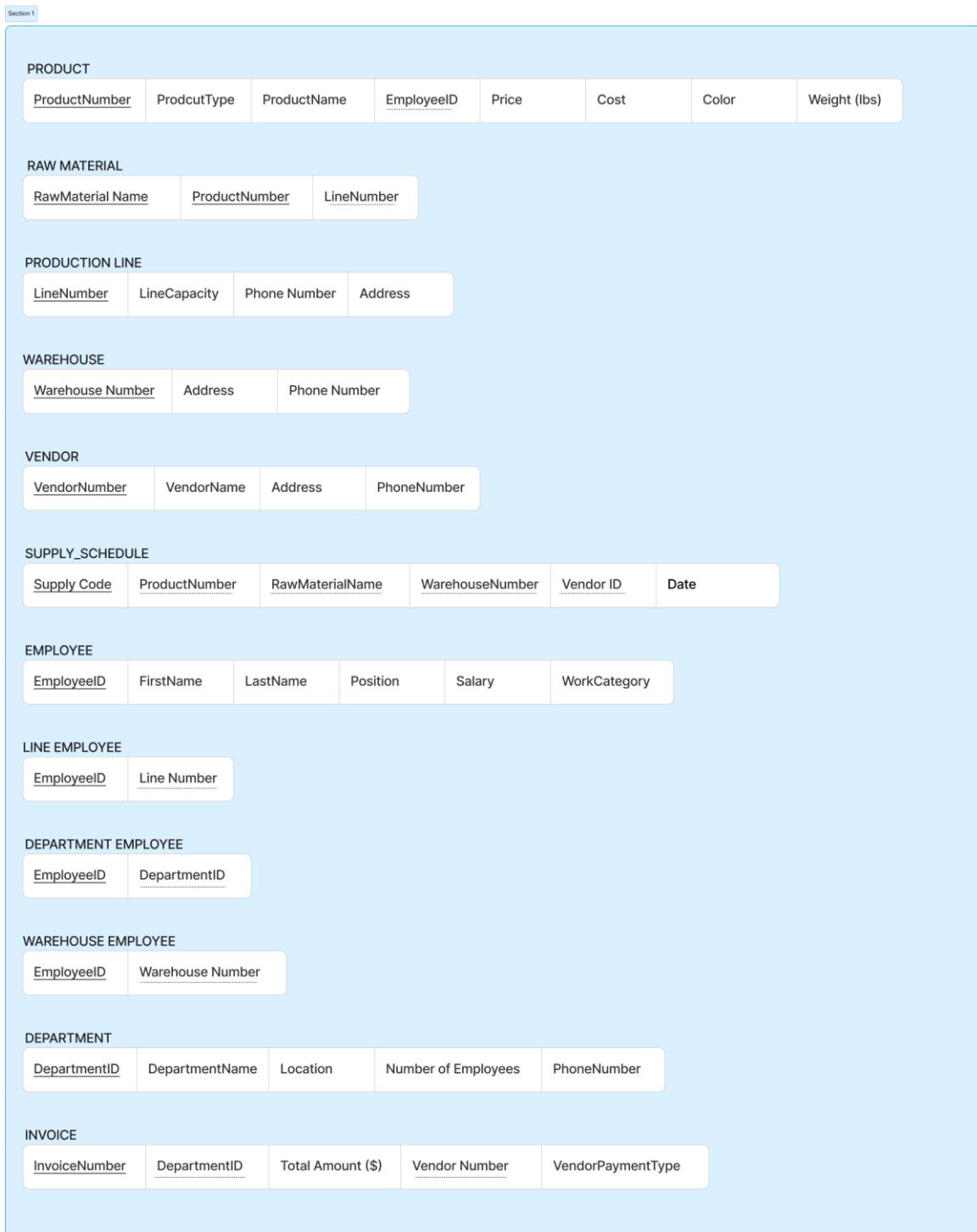
B2. Vendors may be listed in the system even if they do not currently have scheduled supplies.

METADATA:

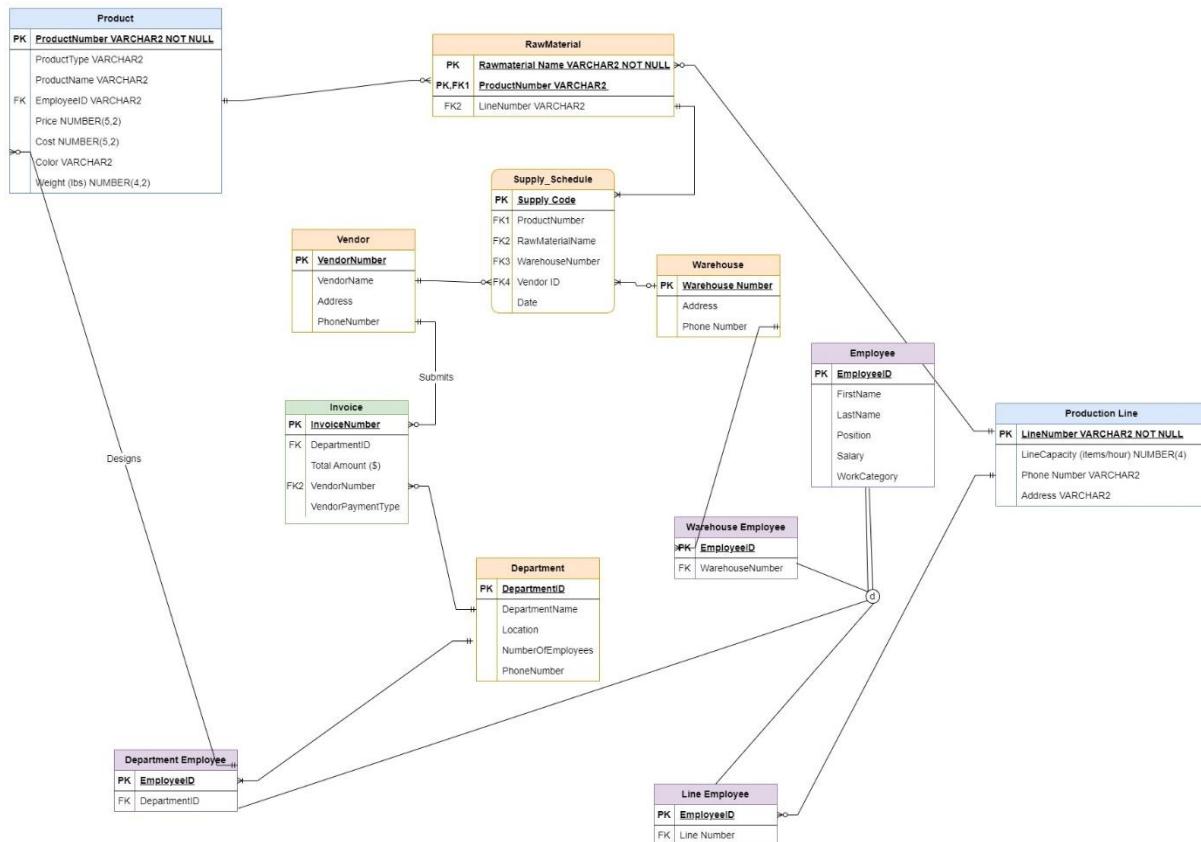
PRODUCT_METADATA					
Name	Type	Length	Description	Source	
ProductNumber	VARCHAR2	10	Product ID	ProductDB	
ProductType	VARCHAR2	20	Type of the product	ProductDB	
ProductName	VARCHAR2	50	Name of the product	ProductDB	
EmployeeID	VARCHAR2	10	ID of the employee	DepartmentEmployeeDB	
Price	NUMBER	10	Price of the product	ProductDB	
Cost	NUMBER	10	Cost of the product	ProductDB	
Color	VARCHAR2	10	Color of the product	ProductDB	
Weight(lbs)	NUMBER	10	Weight of the product	ProductDB	
PRODUCTION_LINE					
Name	Type	Length	Description	Source	
LineNumber	VARCHAR2	10	ID of Production Line	ProductionLineDB	
LineCapacity (items/hour)	NUMBER	6	Capacity of Production	ProductionLineDB	
PhoneNumber	VARCHAR2	12	Reception Number	ProductionLineDB	
Flat	VARCHAR2	25	Loaction of Production	ProductionLineDB	
Street	VARCHAR2	20	Loaction of Production	ProductionLineDB	
State	VARCHAR2	2	Loaction of Production	ProductionLineDB	
Postal Code	NUMBER	6	Loaction of Production	ProductionLineDB	
PRODUCT_PL					
Name	Type	Length	Description	Source	
ProductNumber	VARCHAR2	10	Product ID	ProductDB	
LineNumber	VARCHAR2	10	ID of Production Line	ProductionLineDB	
RAW_MATERIAL_PRODUCT					
Name	Type	Length	Description	Source	
ProductNumber	VARCHAR2	10	Product ID	ProductDB	
RawMaterial Name	VARCHAR2	50	Name of Raw material	RawMaterialDB	
WAREHOUSE					
Name	Type	Length	Description	Source	
WarehouseNumber	VARCHAR2	10	Warehouse ID	WarehouseDB	
Street	VARCHAR2	20	Location of Warehouse	WarehouseDB	
City	VARCHAR2	15	Location of Warehouse	WarehouseDB	
Phone Number	VARCHAR2	14	Contact of Warehouse	WarehouseDB	
VENDOR					
Name	Type	Length	Description	Source	
VendorNumber	VARCHAR2	20	Vendor ID	VendorDB	
VendorName	VARCHAR2	50	Name of Vendor	VendorDB	
Street	VARCHAR2	30	Vendor Address	VendorDB	
City	VARCHAR2	20	Vendor Address	VendorDB	
PhoneNumber	VARCHAR2	12	Vendor Contact	VendorDB	
SUPPLY_SCHEDULE					
Name	Type	Length	Description	Source	
SupplyCode	VARCHAR2	10	Surrogate Key	Supply_ScheduleDB	
ProductNumber	VARCHAR2	10	ID for each Product	ProductDB	
RawmaterialName	VARCHAR2	20	Name of the Raw Mate	RawMaterialDB	
WarehouseNumber	VARCHAR2	10	ID of each Warehouse	WarehouseDB	
VendorNumber	VARCHAR2	15	Vendor ID	VendorDB	
Date1	DATE		Date of Supply	Supply_ScheduleDB	
DEPARTMENT					
Name	Type	Length	Description	Source	
DepartmentID	VARCHAR2	10	Division code	DepartmentDB	
DepartmentName	VARCHAR2	30	Name of the Department	DepartmentDB	
NumberOfEmployees	NUMBER	5	Employees Count	DepartmentDB	
PhoneNumber	VARCHAR2	20	DepartmentContact	DepartmentDB	
Street	VARCHAR2	30	Loaction of Dept	DepartmentDB	
City	VARCHAR2	10	Loaction of Dept	DepartmentDB	
Employee					
Name	Type	Length	Description	Source	
EmployeeID	VARCHAR2	5	ID of the employee	EmployeeDB	
FirstName	VARCHAR2	15	First name of the Employee	EmployeeDB	
LastName	VARCHAR2	15	Last name of the Employee	EmployeeDB	
Position	VARCHAR2	50	Position of the Employee	EmployeeDB	
Salary	VARCHAR2	10	Salary of the Employee	EmployeeDB	
WorkCategory	VARCHAR2	30	Category of the work	EmployeeDB	
INVOICE					
Name	Type	Length	Description	Source	
InvoiceNumber	VARCHAR2	20	Billing Code	InvoiceDB	
Total Amount (\$)	NUMBER	10	Total amount of the invoice	InvoiceDB	
VendorNumber	VARCHAR2	10	Vendor ID	VendorDB	
DepartmentID	VARCHAR2	10	Division code	DepartmentDB	
DEPARTMENT_EMPLOYEE					
Name	Type	Length	Description	Source	
EmployeeID	VARCHAR2	10	ID of the employee	EmployeeDB	
DepartmentID	VARCHAR2	10	Division code	DepartmentDB	
LINE_EMPLOYEE					
Name	Type	Length	Description	Source	
EmployeeID	VARCHAR2	10	ID of the employee	EmployeeDB	
LINE_NUMBER	VARCHAR2	10	ID of Production Line	ProductionLineDB	
WAREHOUSE_EMPLOYEE					
Name	Type	Length	Description	Source	
EmployeeID	VARCHAR2	10	ID of the employee	EmployeeDB	
WAREHOUSE NUMBER	VARCHAR2	10	Warehouse ID	WarehouseDB	
VENDOR_PAYMENTMODE					
Name	Type	Length	Description	Source	
VendorNumber	VARCHAR2	20	Vendor ID	VendorDB	
Vendor_PaymentType	VARCHAR2	20	Primary Payment Mode	InvoiceDB	

Task 1: Primary and Foreign Keys (EER-Notations)

Primary & Foreign Keys are shown in following figure:



Task 2: EER- Diagrams (Pre-Normalization)



1. Product – Raw Material

- Each product can be associated with multiple raw materials (1:M)
- While most products are inherently composed of raw materials, a few exceptions (PR062, PR063, PR065, PR066) exist where products lack raw material associations (optional)
- Raw materials are acquired with the purpose of being used in the production of a product (Mandatory).

2. Raw material – Production Line

- Each production line can be delivered with multiple raw materials, Each instance of the raw material table is associated with a single production line. (M:1)
- For repair purposes, production lines may produce no products, so no raw material will be delivered (Optional)
- Raw materials are acquired with the purpose of being used in the production of a product (Mandatory).

3. Vendor – Supply Schedule

- Vendor can supply many raw materials (1:M)
- Every scheduled supply has a specified vendor (Mandatory).
- Vendors may be listed in the system even if they do not currently have scheduled supplies (Optional).

4. Warehouse - Supply Schedule

- Each warehouse can be associated with multiple supply schedules, but each supply schedule is related to only one warehouse (1:M)
- Since Vendors can supply directly to Production Line without involvement to Warehouse, Warehouse may exist without currently being associated with any scheduled supplies.(Optional)
- Since, each warehouse must be supplied with at least one raw material connection is Mandatory.

5. Supply Schedule – Raw Material

- Many supply schedules can be associated with one raw material, but each supply schedule is related to only one raw material (M:1)
- Each supply schedule is tied to a specific raw material, representing the product being supplied.(Mandatory)
- Each raw material is part of the supply chain, and there should be a record of when and where it is supplied. (Mandatory)

6. Employee

Super Type:

Employee:

Attributes: EmployeeID, FirstName, LastName, Position, Salary, WorkCategory

Subtypes:

Dept Employee:

Attributes: EmployeeID (inherits from the super type), DepartmentID (Foreign Key Attribute)

Warehouse Employees:

Attributes: EmployeeID (inherits from the super type), WarehouseNumber (Foreign Key Attribute)

Line Employees:

Attributes: EmployeeID (inherits from the super type), LineNumber (Foreign Key Attribute)

Relationship Characteristics:

Total: Every employee must be categorized as either a "Dept Employee," "Warehouse Employee," or "Line Employee." There are no employees that do not fall into one of these subtypes.

Disjoint: An employee cannot simultaneously belong to more than one subtype. In other words, an employee can only be a "Dept Employee," a "Warehouse Employee," or a "Line Employee," but not a combination of these.

7. Warehouse – Warehouse Employee (1:M)

8. Department – Department Employee (1:M)

9. Production Line – Line Employee (1:M)

10. Department Employee – Product

- Each department employee can be associated with multiple products, but each product is related to only one department employee. (1:M)
- Every product is designed by a Designer (Mandatory)
- Since department employees are not just designers, every instance in the Department Employee table is not associated with Product design. (Optional)

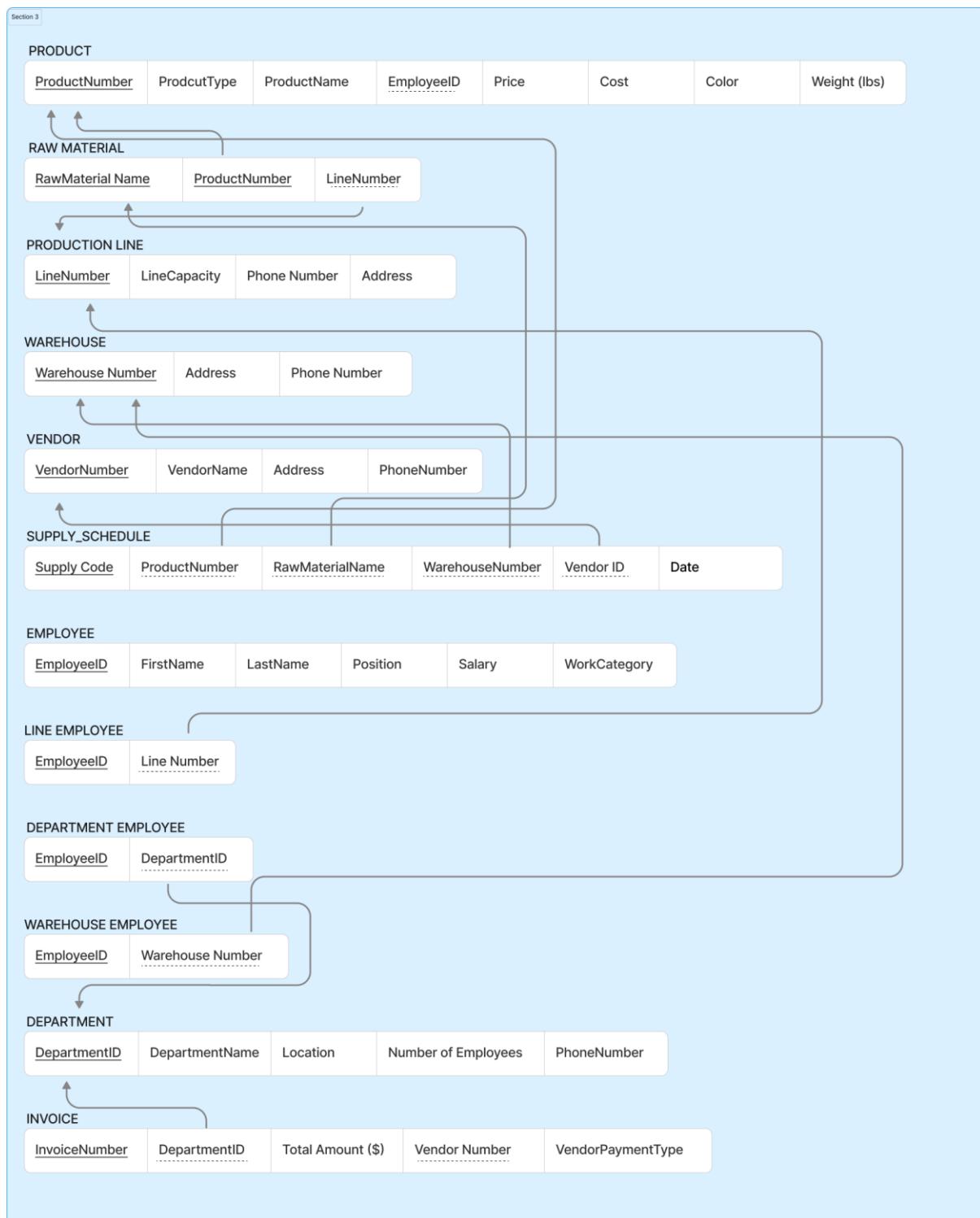
11. Vendor – Invoice

- Each vendor can be associated with multiple invoices, but each invoice is related to only one vendor (1:M).
- Every invoice must be associated with a vendor. (Mandatory on Invoice)
- There might be vendors in the system who have not raised any invoices yet. This flexibility accommodates situations where vendors are added to the system but have not conducted any transaction resulting in an invoice. (Optional)

12. Invoice – Department

- multiple invoices can be associated with one department, but each invoice is related to only one department (M:1)
- Every invoice must be associated with a department (Mandatory)
- Since not all departments necessarily deal with invoices. Only Accounting dept is associated with Invoices (Optional)

Task 3: Relational Schema and Relationships



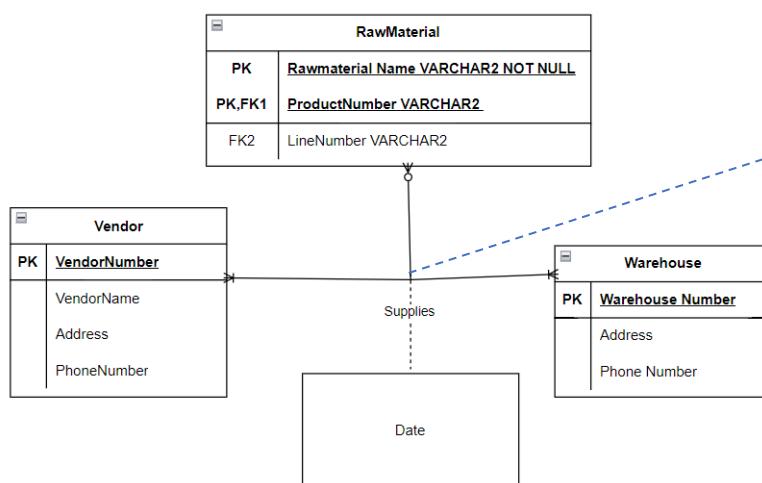
Unary Relationship:

No such relationships are found.

Binary Relationship:

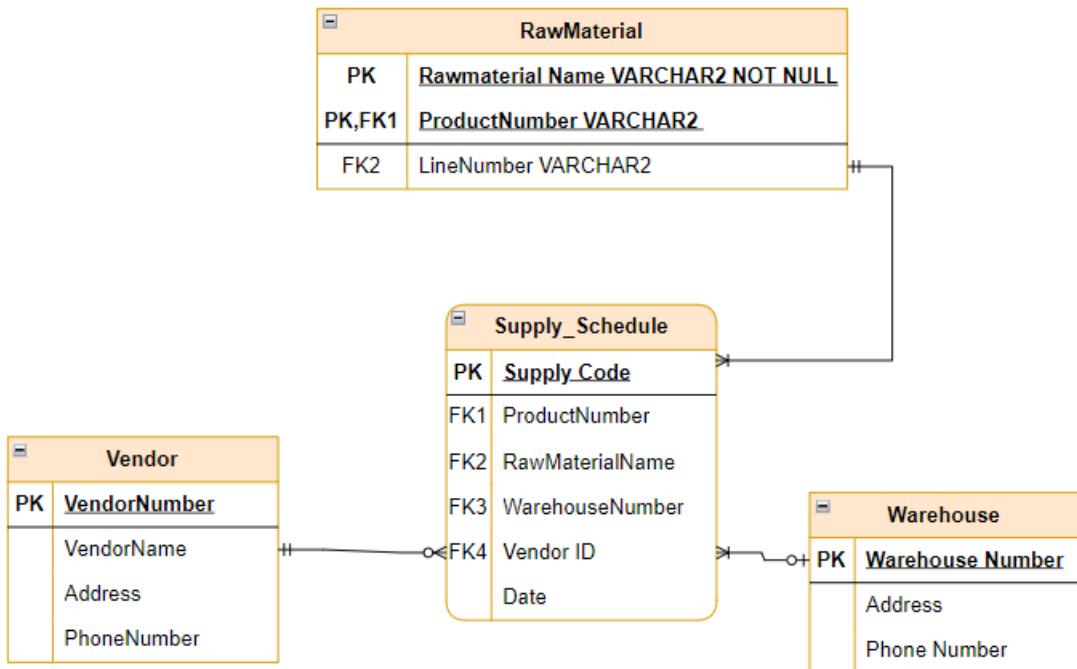
1. Product – Raw Material (1:M)
2. Raw material – Production Line (M:1)
3. Vendor – Supply Schedule (1:M)
4. Warehouse - Supply Schedule(1:M)
5. Supply Schedule – Raw Material(M:1)
6. Warehouse – Warehouse Employee (1:M)
7. Department – Department Employee (1:M)
8. Production Line – Line Employee (1:M)
9. Department Employee – Product (1:M)
10. Vendor – Invoice (1:M)
11. Invoice – Department(M:1)

Ternary Relationships:



For example, an instance is Vendor X Supplies Raw Material C of Product B to Warehouse Y on "12/02/2023".

Vendors can supply various parts to warehouses. The relationship Supplies is used to record the raw materials that are supplied by a given vendor to a particular warehouse. Thus, there are three entity types involved: VENDOR, RAW MATERIAL, and WAREHOUSE. There is one attribute on the relationship Supplies: Date (Date of Supply). For example, one instance of Supplies might record the fact that vendor X can supply raw material C of Product B to warehouse Y, on Date "12/02/2023".

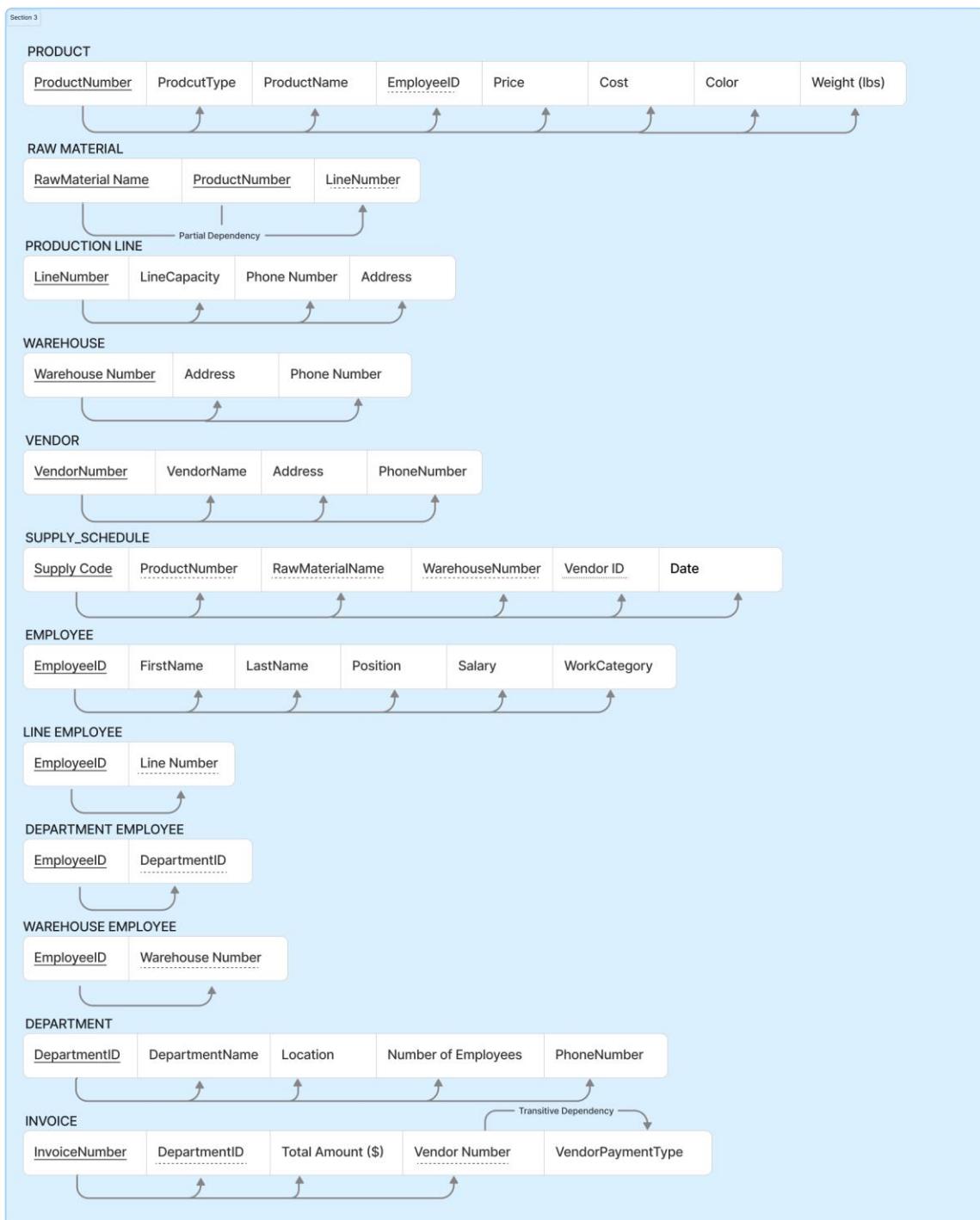


(Associative) Entity type “Supply_Schedule” is used to replace the Supplies relationship.

Task 4: Functional Dependencies and Normalization

Dependencies:

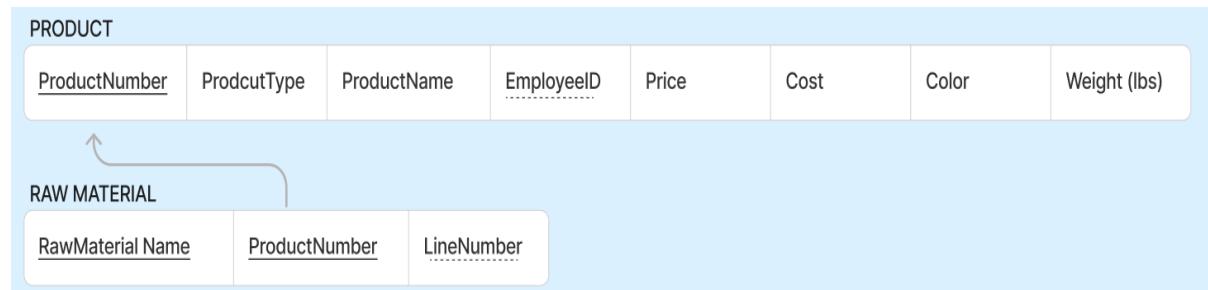
Section 4



Converting to 1NF:

In the mapping of regular entities to relations, simple attributes are directly mapped onto the relation. Composite attributes are represented using only their simple component attributes. Multivalued attributes are transformed into a separate relation with a foreign key derived from the superior entity.

As per this, the multivalued attribute "Raw Material" in the Product was removed, and a new table named "RawMaterial" was created. This new table includes attributes "Product Number", which is a foreign key from Product Table, "Raw Material Name," and "Line Number." This ensures a more structured and normalized representation of the data.

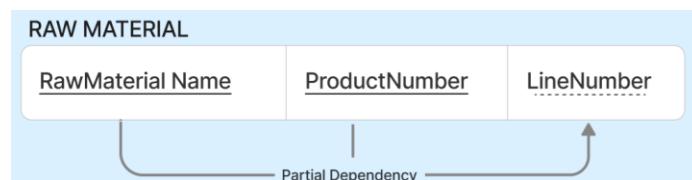


Insertion Anomaly:

In the structure of the Raw Material table, when incorporating an additional raw material for the production of, let's say, Product B. This necessitates the insertion of a Line Number, even if it proves to be redundant for this specific Product.

Deletion Anomaly:

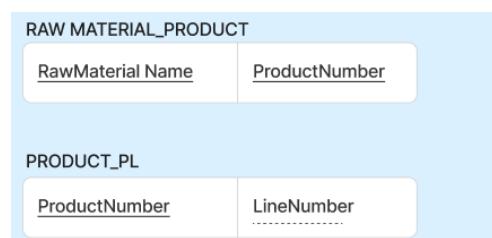
When attempting to terminate the production of a particular product. The inability to delete product details without removing the associated Line Number poses a challenge. Similarly, in the context of a production line being closed or undergoing repairs, attempting to delete its data becomes problematic without simultaneously deleting the associated product details.



Converting to 2NF:

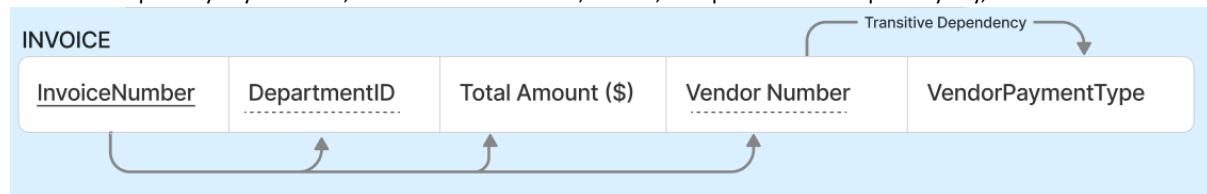
Line Number is partially dependent on Raw Material Name, Product Number.

Removal of the partial dependencies results in the formation of two new relations: RAWMATERIAL_PRODUCT with primary keys Raw Material Name and Product Number and PRODUCT_PL with primary key Product Number.

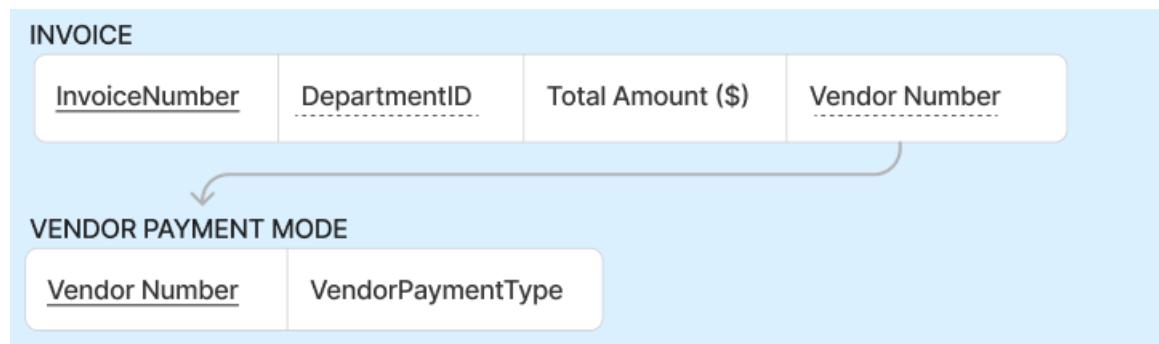


Transitive Dependency:

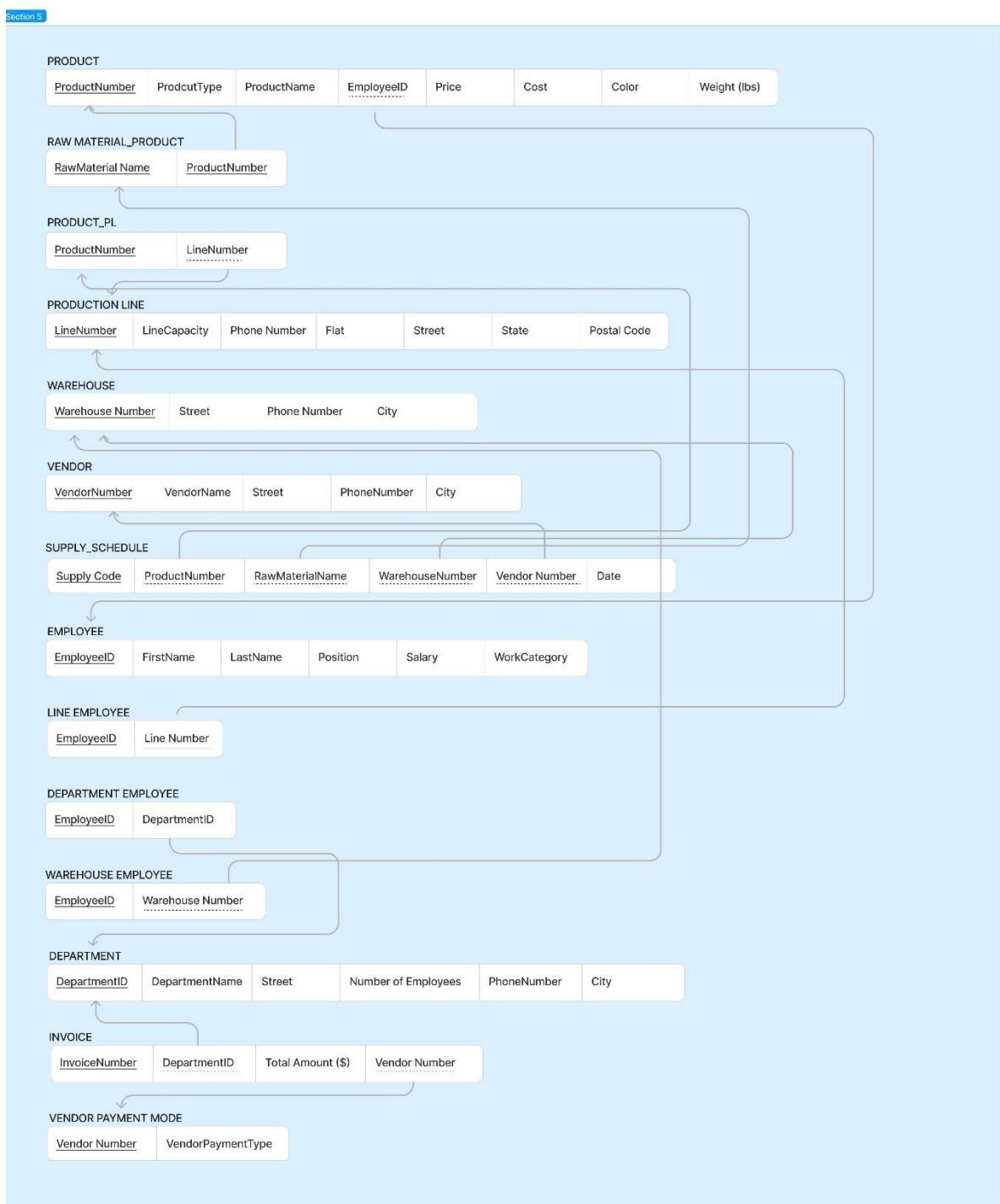
In the Invoice Table, a transitive dependency exists, wherein a non-primary key attribute, "Vendor Payment Type," relies on another non-primary key attribute, "Vendor Number." This, in turn, is dependent on the primary key, "Invoice Number."

**Converting to 3NF:**

A new relation named VENDOR PAYMENT MODE has been created to receive the components of the transitive dependency. The determinant Vendor Name becomes the primary key of this relation.



Relational Schema:

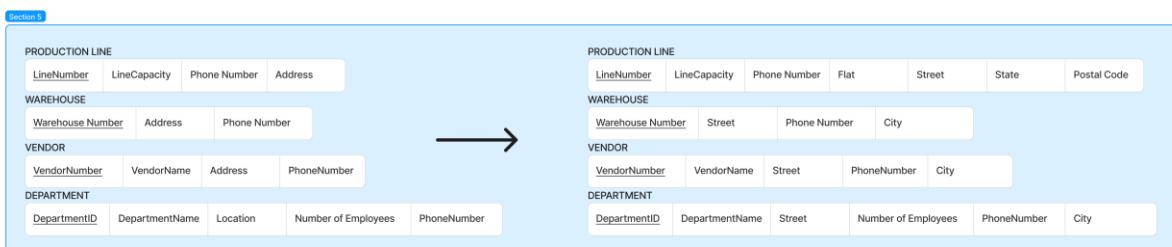


Data Cleaning:

In the raw material table, there is a product labeled as "PER030" for which there is no corresponding information in the Product table. Assuming that "PER030" is a typo, or a data entry error and it should be "PE030," this discrepancy has been addressed through data cleaning in the Raw Material table.

- Removing composite attributes

1. "Address" from Production Line table has been splitted to "Flat", "Street", "State", "Postal Code"
2. "Address" attribute from Warehouse Table has been splitted to "Street" & "City"
3. "Address" attribute from Vendor Table has been splitted to "Street" & "City".
4. "Location" attribute from Department Table has been splitted to "Street" & "City"



Task 5: SQL queries of the database

Below are the tables and corresponding SQL queries of the database.

Department:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for DBMS Project - Task 5 (the final), Object Browser, and Relaunch to update. The URL in the address bar is iacademy2.oracle.com/ords/f?p=4500:1001:4079362595990:FOCUS::RP:OBJECT_NAME:RAW_MATERIAL_PRODUCT. The schema selected is US_A926_SQL_S18. The main area displays the DEPARTMENT table structure with columns: DEPARTMENTID, DEPARTMENTNAME, NUMBEROFEmployees, PHONENUMBER, STREET, and CITY. Buttons for Add Column, Modify Column, Rename Column, Drop Column, Rename, Copy, Drop, Truncate, Create Lookup Table, and Create App are available. The left sidebar lists other tables: DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The bottom status bar shows the Oracle APEX version 22.2.1 and the date/time 12/3/2023 5:27 PM.

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the SQL tab selected. The top navigation bar and URL are identical to the previous screenshot. The main area displays the SQL code for creating the DEPARTMENT table:

```
CREATE TABLE "DEPARTMENT"
(
    "DEPARTMENTID" VARCHAR2(18) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "DEPARTMENTNAME" VARCHAR2(30) COLLATE "USING_NLS_COMP",
    "NUMBEROFEmployees" NUMBER(5,0) NOT NULL ENABLE,
    "PHONENUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP",
    "STREET" VARCHAR2(30) COLLATE "USING_NLS_COMP",
    "CITY" VARCHAR2(10) COLLATE "USING_NLS_COMP",
    CONSTRAINT "DEPARTMENT_PK" PRIMARY KEY ("DEPARTMENTID")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COMP"
```

The left sidebar lists the same set of tables as the previous screenshot. The bottom status bar shows the Oracle APEX version 22.2.1 and the date/time 12/3/2023 5:40 PM.

Department_Employee:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE (selected), EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The main panel displays the structure of the DEPARTMENT_EMPLOYEE table. The table has two columns: EMPLOYEEID (VARCHAR2(10)) and DEPARTMENTID (VARCHAR2(10)). The EMPLOYEEID column is marked as the primary key. The schema is set to US_A926_SQL_S18.

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	VARCHAR2(10)	No	-	1
DEPARTMENTID	VARCHAR2(10)	No	-	-

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the SQL tab selected. It displays the DDL code for creating the DEPARTMENT_EMPLOYEE table. The table has two columns: EMPLOYEEID and DEPARTMENTID. The EMPLOYEEID column is defined as VARCHAR2(10) with a constraint named DEPARTMENT_EMPLOYEE_PK, which makes it the primary key. The DEPARTMENTID column is also defined as VARCHAR2(10). The table is created using the USING INDEX ENABLE option and a default collation of "USING_NLS_COLLATION". An alter statement adds a foreign key constraint named DEPARTMENT_EMPLOYEE_CON, linking the DEPARTMENTID column to the DEPARTMENT table's DEPARTMENTID column.

```

CREATE TABLE "DEPARTMENT_EMPLOYEE"
(
    "EMPLOYEEID" VARCHAR2(10) COLLATE "USING_NLS_COLLATION" NOT NULL ENABLE,
    "DEPARTMENTID" VARCHAR2(10) COLLATE "USING_NLS_COLLATION" NOT NULL ENABLE,
    CONSTRAINT "DEPARTMENT_EMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COLLATION"

ALTER TABLE "DEPARTMENT_EMPLOYEE" ADD CONSTRAINT "DEPARTMENT_EMPLOYEE_CON" FOREIGN KEY ("DEPARTMENTID")
    REFERENCES "DEPARTMENT" ("DEPARTMENTID") ENABLE
/

```

Employee:

The screenshot shows the Oracle APEX Object Browser interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The EMPLOYEE object is selected and highlighted in green. The main panel displays the structure of the EMPLOYEE table. The table has six columns: EMPLOYEEID, FIRSTNAME, LASTNAME, POSITION, SALARY, and WORKCATEGORY. The EMPLOYEEID column is defined as VARCHAR2(5) and is the primary key. The FIRSTNAME, LASTNAME, and POSITION columns are VARCHAR2(15). The SALARY column is NUMBER(10,0). The WORKCATEGORY column is VARCHAR2(30). Buttons at the top of the table view include Add Column, Modify Column, Rename Column, Drop Column, Rename, Copy, Drop, Truncate, Create Lookup Table, and Create App.

SQL Query:

The screenshot shows the Oracle APEX Object Browser interface with the SQL tab selected. The left sidebar lists the same set of database objects as the previous screenshot. The EMPLOYEE object is selected. The main panel displays the SQL code for creating the EMPLOYEE table. The code is as follows:

```
CREATE TABLE "EMPLOYEE"
(
    "EMPLOYEEID" VARCHAR2(5) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "FIRSTNAME" VARCHAR2(15) COLLATE "USING_NLS_COMP",
    "LASTNAME" VARCHAR2(15) COLLATE "USING_NLS_COMP",
    "POSITION" VARCHAR2(50) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "SALARY" NUMBER(10,0) NOT NULL ENABLE,
    "WORKCATEGORY" VARCHAR2(30) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    CONSTRAINT "EMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
    USING INDEX ENABLE
)
DEFAULT COLLATION "USING_NLS_COMP"
```

The SQL tab also includes buttons for REST and Sample Queries.

Invoice:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE (which is selected and highlighted in green), LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The main workspace displays the schema definition for the INVOICE table. The table has four columns: INVOICENUMBER (VARCHAR2(20)), TOTAL_AMOUNT (NUMBER(10,2)), VENDORNUMBER (VARCHAR2(10)), and DEPARTMENTID (VARCHAR2(10)). The primary key constraint is defined as INVOICE_PK, which references the INVOICENUMBER column.

Column Name	Data Type	Nullable	Default	Primary Key
INVOICENUMBER	VARCHAR2(20)	No	-	1
TOTAL_AMOUNT	NUMBER(10,2)	No	-	-
VENDORNUMBER	VARCHAR2(10)	No	-	-
DEPARTMENTID	VARCHAR2(10)	No	-	-

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the SQL tab selected. It displays the generated SQL code for creating the INVOICE table. The code includes the table structure with columns INVOICENUMBER, TOTAL_AMOUNT, VENDORNUMBER, and DEPARTMENTID, along with their respective data types and constraints. It also includes the primary key constraint INVOICE_PK and the foreign key constraint INVOICE_CON.

```
CREATE TABLE "INVOICE"
(
    "INVOICENUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "TOTAL_AMOUNT" NUMBER(10,2) NOT NULL ENABLE,
    "VENDORNUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "DEPARTMENTID" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    CONSTRAINT "INVOICE_PK" PRIMARY KEY ("INVOICENUMBER")
    USING INDEX ENABLE
)
DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE "INVOICE" ADD CONSTRAINT "INVOICE_CON" FOREIGN KEY ("VENDORNUMBER")
    REFERENCES "VENDOR" ("VENDORNUMBER") ENABLE
/
```

Line_Employee:

The screenshot shows the Oracle APEX Object Browser interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMldb_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The 'LINE_EMPLOYEE' object is selected and highlighted in green. The main panel displays the 'LINE_EMPLOYEE' table structure with two columns: 'EMPLOYEEID' (VARCHAR2(10)) and 'LINENUMBER' (VARCHAR2(10)). The table has a primary key constraint named 'LINE_EMPLOYEE_PK' on the 'EMPLOYEEID' column.

SQL Query:

The screenshot shows the Oracle APEX Object Browser interface with the 'SQL' tab selected for the 'LINE_EMPLOYEE' table. The SQL code displayed is the CREATE TABLE statement for the 'LINE_EMPLOYEE' table:

```
CREATE TABLE "LINE_EMPLOYEE"
(
    "LINENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
    "EMPLOYEEID" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    CONSTRAINT "LINE_EMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
)
DEFAULT COLLATION "USING_NLS_COMP"
```

Product:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar has a tree view of tables under 'Tables'. The 'PRODUCT' table is selected and highlighted in green. The main panel displays the 'PRODUCT' table structure with columns: PRODUCTNUMBER, PRODUCUTTYPE, PRODUCTNAME, EMPLOYEEID, PRICE, COST, COLOR, and WEIGHT_LBS. The 'SQL' tab is active at the top of the main panel.

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTNUMBER	VARCHAR2(10)	No	-	1
PRODUCUTTYPE	VARCHAR2(20)	No	-	-
PRODUCTNAME	VARCHAR2(50)	Yes	-	-
EMPLOYEEID	VARCHAR2(10)	No	-	-
PRICE	NUMBER(10,2)	Yes	-	-
COST	NUMBER(10,2)	Yes	-	-
COLOR	VARCHAR2(10)	Yes	-	-
WEIGHT_LBS	NUMBER(10,2)	Yes	-	-

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar has a tree view of tables under 'Tables'. The 'PRODUCT' table is selected and highlighted in green. The main panel displays the SQL code for creating the 'PRODUCT' table. The 'SQL' tab is active at the top of the main panel.

```

CREATE TABLE "PRODUCT"
(
    "PRODUCTNUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "PRODUCUTTYPE" VARCHAR2(20) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "PRODUCTNAME" VARCHAR2(50) COLLATE "USING_NLS_COMP",
    "EMPLOYEEID" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "PRICE" NUMBER(10,2),
    "COST" NUMBER(10,2),
    "COLOR" VARCHAR2(10) COLLATE "USING_NLS_COMP",
    "WEIGHT_LBS" NUMBER(10,2),
    CONSTRAINT "PRODUCT_PK" PRIMARY KEY ("PRODUCTNUMBER")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE "PRODUCT" ADD CONSTRAINT "PRODUCT_CON" FOREIGN KEY ("EMPLOYEEID")
    REFERENCES "DEPARTMENT_EMPLOYEE" ("EMPLOYEEID") ENABLE
/

```

Production_Line:

The screenshot shows the Oracle APEX Object Browser interface. The left sidebar lists various database objects like DEPARTMENT_EMPLOYEE, EMPLOYEE, etc. The main panel displays the structure of the 'PRODUCTION_LINE' table. The table has seven columns: LINENUMBER, LINECAPACITY_ITEMS_PER_HR, PHONENUMBER, FLAT, STREET, STATE, and POSTALCODE. The 'Table' tab is selected, and the 'Data' tab is also visible. The schema is set to 'US_A926_SQL_S18'. The bottom status bar shows copyright information and the date '12/3/2023'.

Column Name	Data Type	Nullable	Default	Primary Key
LINENUMBER	VARCHAR2(10)	No	-	1
LINECAPACITY_ITEMS_PER_HR	NUMBER(6,2)	Yes	-	-
PHONENUMBER	VARCHAR2(12)	Yes	-	-
FLAT	VARCHAR2(25)	Yes	-	-
STREET	VARCHAR2(20)	Yes	-	-
STATE	VARCHAR2(2)	Yes	-	-
POSTALCODE	NUMBER(6,0)	Yes	-	-

SQL Query:

The screenshot shows the Oracle APEX Object Browser interface with the 'SQL' tab selected for the 'PRODUCTION_LINE' table. The SQL code generated for creating the table is displayed. The code includes the table name, column definitions (LINENUMBER, LINECAPACITY_ITEMS_PER_HR, PHONENUMBER, FLAT, STREET, STATE, POSTALCODE), constraints (PRIMARY KEY on LINENUMBER), and indexes (INDEX ENABLE). The schema is set to 'US_A926_SQL_S18'. The bottom status bar shows copyright information and the date '12/3/2023'.

```
CREATE TABLE "PRODUCTION_LINE"
(
    "LINENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "LINECAPACITY_ITEMS_PER_HR" NUMBER(6,2),
    "PHONENUMBER" VARCHAR2(12) COLLATE "USING_NLS_COMP",
    "FLAT" VARCHAR2(25) COLLATE "USING_NLS_COMP",
    "STREET" VARCHAR2(20) COLLATE "USING_NLS_COMP",
    "STATE" VARCHAR2(2) COLLATE "USING_NLS_COMP",
    "POSTALCODE" NUMBER(6,0),
    CONSTRAINT "PRODUCTION_LINE_PK" PRIMARY KEY ("LINENUMBER")
    USING INDEX ENABLE
)
DEFAULT COLLATION "USING_NLS_COMP"
```

Product_PL:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, and PRODUCT_PL. The PRODUCT_PL object is selected and highlighted in green. The main workspace displays the structure of the PRODUCT_PL table. The table has two columns: PRODUCTNUMBER (VARCHAR2(10)) and LINENUMBER (VARCHAR2(10)). The table is defined as the primary key. Below the table structure, there are links for 'Download' and 'Print'.

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTNUMBER	VARCHAR2(10)	No	-	1
LINENUMBER	VARCHAR2(10)	No	-	-

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the SQL tab selected. The left sidebar lists various database objects, and the PRODUCT_PL object is selected. The main workspace displays the SQL code for creating the PRODUCT_PL table. The code is as follows:

```

CREATE TABLE "PRODUCT_PL"
(
    "PRODUCTNUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "LINENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    CONSTRAINT "PRODUCT_PL_PK" PRIMARY KEY ("PRODUCTNUMBER")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COMP"
/

```

Supply_Schedule:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The 'SUPPLY_SCHEDULE' object is selected and highlighted in green. The main workspace displays the table structure for 'SUPPLY_SCHEDULE' with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
SUPPLYCODE	VARCHAR2(10)	No	-	1
PRODUCTNUMBER	VARCHAR2(10)	Yes	-	-
RAWMATERIALNAME	VARCHAR2(20)	Yes	-	-
WAREHOUSENUMBER	VARCHAR2(10)	Yes	-	-
VENDORNUMBER	VARCHAR2(15)	Yes	-	-
DATE1	DATE	Yes	-	-

Below the table structure, there are links for 'Download' and 'Print'. The bottom status bar indicates the schema is 'US_A926_SQL_S18' and the version is 'Oracle APEX 22.2.1'.

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the 'SQL' tab selected. The left sidebar lists the same set of database objects as before. The 'SUPPLY_SCHEDULE' object is selected. The main workspace displays the SQL code for creating the 'SUPPLY_SCHEDULE' table and establishing foreign key constraints:

```

CREATE TABLE "SUPPLY_SCHEDULE"
(
    "SUPPLYCODE" VARCHAR2(10) COLLATE "USING_NLS_COMP",
    "PRODUCTNUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
    "RAWMATERIALNAME" VARCHAR2(20) COLLATE "USING_NLS_COMP",
    "WAREHOUSENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
    "VENDORNUMBER" VARCHAR2(15) COLLATE "USING_NLS_COMP",
    "DATE1" DATE,
    CONSTRAINT "PK_PRIMARY_KEY" PRIMARY KEY ("SUPPLYCODE")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE "SUPPLY_SCHEDULE" ADD CONSTRAINT "FK_CONSTRAINT1" FOREIGN KEY ("PRODUCTNUMBER")
    REFERENCES "RAW_MATERIAL_PRODUCT" ("RAWMATERIALNAME", "PRODUCTNUMBER") ENABLE
/
ALTER TABLE "SUPPLY_SCHEDULE" ADD CONSTRAINT "FK_CONSTRAINT2" FOREIGN KEY ("WAREHOUSENUMBER")
    REFERENCES "WAREHOUSE" ("WAREHOUSENUMBER") ENABLE
/
ALTER TABLE "SUPPLY_SCHEDULE" ADD CONSTRAINT "FK_CONSTRAINT4" FOREIGN KEY ("VENDORNUMBER")
    REFERENCES "VENDOR" ("VENDORNUMBER") ENABLE
/

```

The bottom status bar indicates the schema is 'US_A926_SQL_S18' and the version is 'Oracle APEX 22.2.1'.

RawMaterial_Product:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT, DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT (which is selected and highlighted in green), SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, and WAREHOUSE. The main panel displays the RAW_MATERIAL_PRODUCT table definition. The table has two columns: RAWMATERIALNAME (VARCHAR2(50)) and PRODUCTNUMBER (VARCHAR2(10)). The primary key is defined as a composite key consisting of both columns. The interface includes tabs for Table, Data, Indexes, Model, Constraints, Grants, Statistics, UI Defaults, Triggers, Dependencies, SQL, REST, and Sample Queries. The schema is set to US_A926_SQL_S18.

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the SQL tab selected. The left sidebar lists the same database objects as before. The main panel displays the SQL code for creating the RAW_MATERIAL_PRODUCT table. The code includes the table structure with columns RAWMATERIALNAME and PRODUCTNUMBER, a composite primary key constraint named PK_RAWMATERIAL, and a foreign key constraint FK_PRODUCTNUMBER referencing the PRODUCT table. The code is wrapped in CREATE TABLE and ALTER TABLE statements. The schema is set to US_A926_SQL_S18.

Vendor:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, WAREHOUSE, and WAREHOUSE_EMPLOYEE. The 'VENDOR' object is selected and highlighted in green. The main panel displays the 'VENDOR' table structure with the following columns and their properties:

Column Name	Data Type	Nullable	Default	Primary Key
VENDORNUMBER	VARCHAR2(20)	No	-	1
VENDORNAME	VARCHAR2(50)	Yes	-	-
PHONENUMBER	VARCHAR2(12)	Yes	-	-
STREET	VARCHAR2(30)	Yes	-	-
CITY	VARCHAR2(20)	Yes	-	-

Below the table structure, there are buttons for 'Download' and 'Print'. The bottom status bar indicates the copyright year (1999-2022) and the Oracle APEX version (22.2.1).

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the 'SQL' tab selected. The left sidebar lists the same set of database objects as the previous screenshot. The 'VENDOR' object is selected. The main panel displays the SQL code for creating the 'VENDOR' table:

```

CREATE TABLE "VENDOR"
(
    "VENDORNUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "VENDORNAME" VARCHAR2(50) COLLATE "USING_NLS_COMP",
    "PHONENUMBER" VARCHAR2(12) COLLATE "USING_NLS_COMP",
    "STREET" VARCHAR2(30) COLLATE "USING_NLS_COMP",
    "CITY" VARCHAR2(20) COLLATE "USING_NLS_COMP",
    CONSTRAINT "VENDOR_PK" PRIMARY KEY ("VENDORNUMBER")
    USING INDEX ENABLE
)
DEFAULT COLLATION "USING_NLS_COMP"
/

```

The bottom status bar indicates the copyright year (1999-2022) and the Oracle APEX version (22.2.1).

Vendor_Paymentmode:

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. In the Object Browser on the left, 'VENDOR_PAYMENTMODE' is highlighted. The main workspace displays the table structure for 'VENDOR_PAYMENTMODE' with two columns: 'VENDORNUMBER' (VARCHAR2(20)) and 'VENDORPAYMENTTYPE' (VARCHAR2(20)). The table is set as the primary key. Below the table definition are 'Download' and 'Print' links.

Column Name	Data Type	Nullable	Default	Primary Key
VENDORNUMBER	VARCHAR2(20)	No	-	1
VENDORPAYMENTTYPE	VARCHAR2(20)	No	-	-

SQL Query:

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. In the Object Browser on the left, 'VENDOR_PAYMENTMODE' is highlighted. The main workspace displays the SQL code for creating the 'VENDOR_PAYMENTMODE' table:

```

CREATE TABLE "VENDOR_PAYMENTMODE"
(
    "VENDORNUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "VENDORPAYMENTTYPE" VARCHAR2(20) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    CONSTRAINT "VENDOR_PAYMENTMODE_PK" PRIMARY KEY ("VENDORNUMBER")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COMP"
/

```

Warehouse:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, WAREHOUSE, and WAREHOUSE_EMPLOYEE. The 'WAREHOUSE' object is selected and highlighted in green. The main workspace displays the table structure for 'WAREHOUSE' with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
WAREHOUSENUMBER	VARCHAR2(10)	No	-	1
PHONE_NUMBER	VARCHAR2(14)	Yes	-	-
STREET	VARCHAR2(20)	Yes	-	-
CITY	VARCHAR2(15)	Yes	-	-

Below the table structure, there are links for 'Download' and 'Print'. The status bar at the bottom indicates the schema is 'US_A926_SQL_S18' and the copyright notice is 'Copyright © 1999, 2022, Oracle and/or its affiliates.' The system tray shows the date and time as '12/3/2023 6:16 PM'.

SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface with the 'SQL' tab selected. The left sidebar lists the same set of database objects as the previous screenshot. The main workspace displays the SQL code for creating the 'WAREHOUSE' table:

```

CREATE TABLE "WAREHOUSE"
(
    "WAREHOUSENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "PHONE_NUMBER" VARCHAR2(14) COLLATE "USING_NLS_COMP",
    "STREET" VARCHAR2(20) COLLATE "USING_NLS_COMP",
    "CITY" VARCHAR2(15) COLLATE "USING_NLS_COMP",
    CONSTRAINT "WAREHOUSE_PK" PRIMARY KEY ("WAREHOUSENUMBER")
    USING INDEX ENABLE
) DEFAULT COLLATION "USING_NLS_COMP"
/

```

The status bar at the bottom indicates the schema is 'US_A926_SQL_S18' and the copyright notice is 'Copyright © 1999, 2022, Oracle and/or its affiliates.' The system tray shows the date and time as '12/3/2023 6:17 PM'.

Warehouse_Employee:

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: DEPARTMENT_EMPLOYEE, EMPLOYEE, HTMLDB_PLAN_TABLE, INVOICE, LINE_EMPLOYEE, PRODUCT, PRODUCTION_LINE, PRODUCT_PL, RAW_MATERIAL_PRODUCT, SUPPLY_SCHEDULE, VENDOR, VENDOR_PAYMENTMODE, WAREHOUSE, and WAREHOUSE_EMPLOYEE. The WAREHOUSE_EMPLOYEE object is selected and highlighted in green. The main workspace displays the table structure for WAREHOUSE_EMPLOYEE. The table has two columns: EMPLOYEEID (VARCHAR2(10)) and WAREHOUSENUMBER (VARCHAR2(10)). The EMPLOYEEID column is defined as the primary key. The table creation SQL is also visible in the workspace.

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	VARCHAR2(10)	No	-	1
WAREHOUSENUMBER	VARCHAR2(10)	No	-	-

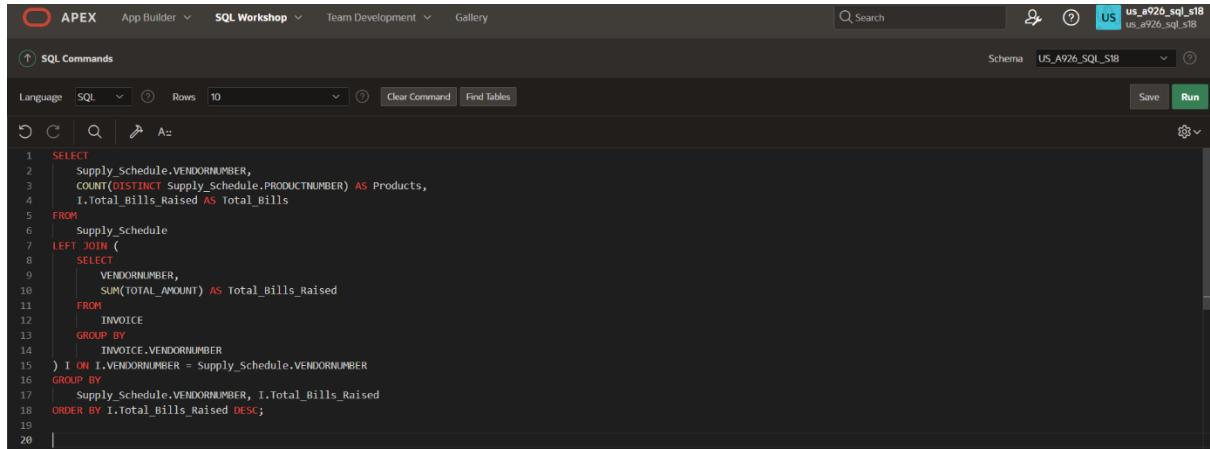
SQL Query:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one but with a different view. The left sidebar lists the same database objects. The WAREHOUSE_EMPLOYEE object is selected. The main workspace displays the generated SQL code for creating the WAREHOUSE_EMPLOYEE table. The code includes the table name, column definitions (WAREHOUSENUMBER and EMPLOYEEID), constraints (PRIMARY KEY and FOREIGN KEY), and indexes (USING INDEX). The table creation SQL is as follows:

```
CREATE TABLE "WAREHOUSE_EMPLOYEE"
(
    "WAREHOUSENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    "EMPLOYEEID" VARCHAR2(10) COLLATE "USING_NLS_COMP" NOT NULL ENABLE,
    CONSTRAINT "WAREHOUSE_EMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
    USING INDEX ENABLE
)
DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE "WAREHOUSE_EMPLOYEE" ADD CONSTRAINT "WAREHOUSE_EMPLOYEE_CON" FOREIGN KEY ("WAREHOUSENUMBER")
    REFERENCES "WAREHOUSE" ("WAREHOUSENUMBER") ENABLE
/
```

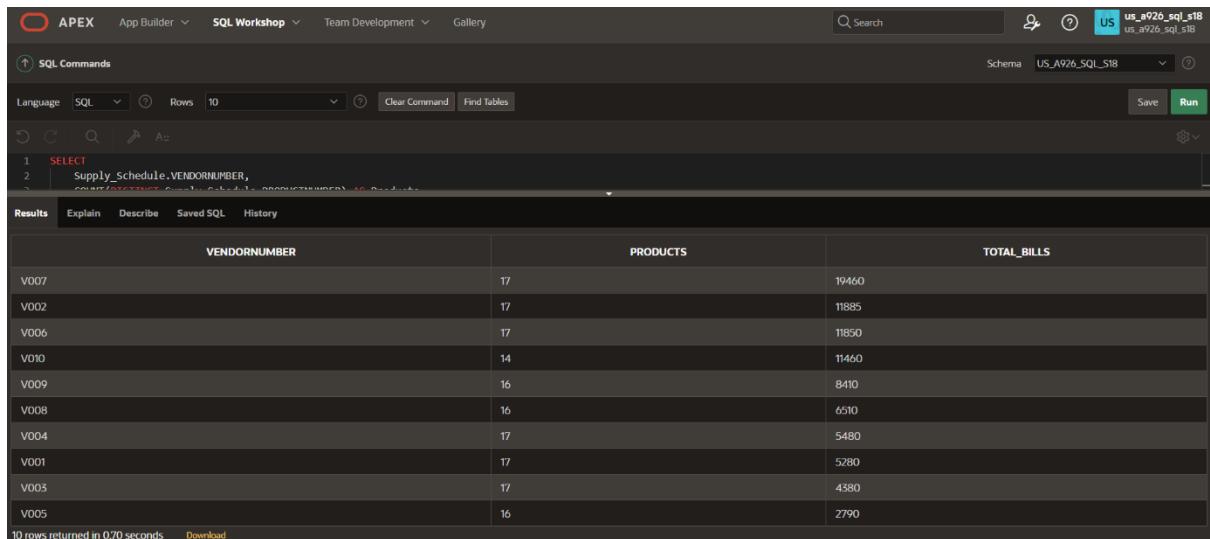
1. Focus on Vendor Management

SQL Code:



```
1 SELECT
2     Supply_Schedule.VENDORNUMBER,
3     COUNT(DISTINCT Supply_Schedule.PRODUCTNUMBER) AS Products,
4     I.Total_Bills_Raised AS Total_Bills
5 FROM
6     Supply_Schedule
7     LEFT JOIN (
8         SELECT
9             VENDORNUMBER,
10            SUM(TOTAL_AMOUNT) AS Total_Bills_Raised
11        FROM
12        INVOICE
13        GROUP BY
14        INVOICE.VENDORNUMBER
15    ) I ON I.VENDORNUMBER = Supply_Schedule.VENDORNUMBER
16 GROUP BY
17     Supply_Schedule.VENDORNUMBER, I.Total_Bills_Raised
18 ORDER BY I.Total_Bills_Raised DESC;
19
20
```

Output:



The screenshot shows the results of the executed SQL query. The results are displayed in a table with three columns: VENDORNUMBER, PRODUCTS, and TOTAL_BILLS.

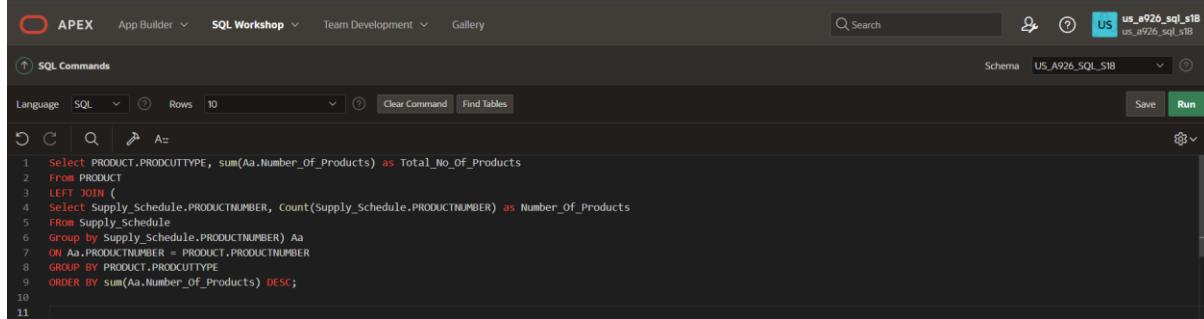
VENDORNUMBER	PRODUCTS	TOTAL_BILLS
V007	17	19460
V002	17	11885
V006	17	11850
V010	14	11460
V009	16	8410
V008	16	6510
V004	17	5480
V001	17	5280
V005	16	4580
V003		2790

It gives a list of vendors, count of products associated with each vendor, and the total amount of bills raised by each vendor.

These metrics collectively help in managing vendor relationships, assessing product diversity, and understanding the financial impact of transactions in an SCM database.

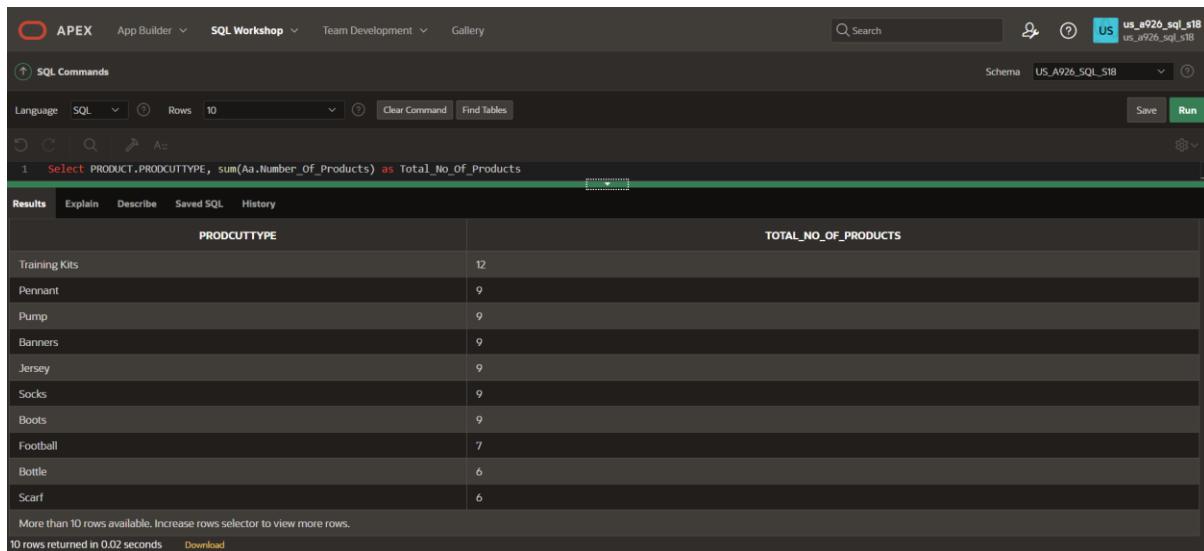
2. Demand Planning

Code:



```
1 Select PRODUCT.PRODUCTTYPE, sum(Aa.Number_of_Products) as Total_No_of_Products
2 From PRODUCT
3 LEFT JOIN (
4 Select Supply_Schedule.PRODUCTNUMBER, Count(Supply_Schedule.PRODUCTNUMBER) as Number_of_Products
5 From Supply_Schedule
6 Group by Supply_Schedule.PRODUCTNUMBER) Aa
7 On Aa.PRODUCTNUMBER = PRODUCT.PRODUCTNUMBER
8 Group By PRODUCT.PRODUCTTYPE
9 ORDER BY sum(Aa.Number_of_Products) DESC;
10
11
```

Output:



PRODUCTTYPE	TOTAL_NO_OF_PRODUCTS
Training Kits	12
Pennant	9
Pump	9
Banners	9
Jersey	9
Socks	9
Boots	9
Football	7
Bottle	6
Scarf	6

The output provides a summary of the production volume for each product type, highlighting which product types have the highest total number of products produced. It enables companies to understand which products are in higher demand, allowing for better inventory management and production planning.

3. Vendors list

Code & Output

The screenshot shows the Oracle APEX SQL Workshop interface. The query window contains the following SQL code:

```
1 Select DISTINCT(VendorName) from Vendor
2 JOIN Supply_Schedule ON Supply_Schedule.vendorNumber = Vendor.VendorNumber
3 JOIN RAW_MATERIAL_PRODUCT ON Supply_Schedule.RawMaterialName = RAW_MATERIAL_PRODUCT.RawMaterialName
4 JOIN Product ON RAW_MATERIAL_PRODUCT.ProductNumber = Product.ProductNumber
5 WHERE Product.ProductType = 'Training Kits';
```

The results pane displays a list of vendor names:

VENDORNAME
Starlight Synthetics
Galaxy Gear
AstroBottles Enterprises
BlueFab Textiles Ltd.
Celestial Ceramics Inc.
GreenTrend Accessories Co.
Rovers Rubber Co.
Universal Prints
YellowSports Equipments Inc.
Cosmic Craftsmen

Below the results, it says "10 rows returned in 0.10 seconds". The bottom status bar shows the schema as "US_A926_SQL_S18" and the session ID as "US_A926_SQL_S18".

It gives the list of Vendors who supply raw materials for Training Kits

Thank You...!