# CDS 502
# DATABASE MANAGEMENT SYSTEMS

TEAM 14 - TRAILBLAZERS

## FINAL PROJECT

## Integrated Supply Chain and Financial Management System

The project aims to develop a comprehensive Enterprise Resource Planning (ERP) Database Schema for the company. The primary objective is to design and implement a database system that will streamline and optimize various aspects of the company's operations, as well as regulate and evaluate departmental structures and roles.

Proposed Timeline:

| TASK 0 | TASK 1 | TASK 2 | TASK 3 | TASK 4 | TASK 5 |
|---|---|---|---|---|---|
| 10/1 - 10/8 | 10/9 - 11/20 | 10/12 - 10/20 | 10/21 - 11/10 | 11/11 - 11/20 | 11/21 - 11/26 |

Real Timeline:

| TASK 0 | TASK 1 | TASK 2 | TASK 3 | TASK 4 | TASK 5 |
|---|---|---|---|---|---|
| 11/1 - 11/8 | 11/9 - 11/30 | 11/12 - 11/20 | 11/21 - 11/25 | 11/26 - 12/1 | 12/1 - 12/2 |

**TEAM MEMBERS:**

- Shloka Bhatt – Tech Lead
- Kumari Sweta – Coresearcher
- Alisha Ruqshan – Presentation Lead
- Pulkit Batra – Team Coordinator

# PROJECT OVERVIEW

The project is aimed at developing a comprehensive Enterprise Resource Planning (ERP) Database Schema for the company. The primary objective is to design and implement a database system that will streamline and optimize various aspects of the company's operations, from design to production, employee management, materials procurement, and financial oversight.

The company comprises of departments, warehouses, and production lines, each staffed with employees. Departments design unique products, and each product is manufactured by a specific production line. Raw materials are sourced from various warehouses, supplied by multiple vendors, including direct vendor supply. Vendors submit invoices for order processing by the accounting department.

## Summary:

- To Create an Enterprise Resource Planning (ERP) Database Schema.
- Streamline workflow from design to production.
- Regulate and evaluate departmental structures and roles.
- Optimize the management of materials from vendors.
- Handle invoices efficiently for financial operations.

## Core Components managed:

- Employee Management
- Product Lifecycle
- Raw Material Management
- Vendor Invoice Processing
- Business Rules and Attributes

## Set of Business Rules:

1. The company has departments, warehouses, and production lines.
2. The company designs and produces products.
3. Every department may possess no employees or multiple employees.
4. The warehouse can have employees or be without them.
5. Each warehouse and production line may have multiple employees.
6. Each employee works in only one department, warehouse, or production line.
7. Some departments design products. Such departments design at least one product.
8. Different departments must not work on the same product.
9. Each product must be produced by one specific production line.
10. A product may not be in production.
11. A product not in production will not have its raw material in the inventory.
12. Each production line can produce only one product. For repair purposes, production lines may produce no products.
13. During repair activities production lines may or may not have employees.
14. The company will purchase the raw material given it is being used for a product.
15. Raw materials are supplied by any number of warehouses, which are supplied by any number of vendors.
16. Raw materials may also be directly supplied by vendors.
17. All vendors are supposed to supply a minimum of one raw material.
18. Vendors submit invoices if they receive any orders.
19. Invoices must be processed by the accounting department.

We aim to provide a valuable solution that enhances operational efficiency, financial oversight, and employee management.

# REPOSITORY

The metadata repository is designed to comprehensively document the database schema for the Integrated Supply Chain and Financial Management System. It provides detailed information about each entity, their associated attributes, data types, and descriptions, allowing it easier to understand for all the stakeholders and comprehend the structure and purpose of the database.

We aim our metadata repository to:

- Serve as a reference document for developers, database administrators, and stakeholders involved in the project.
- Facilitate a clear understanding of the database schema's structure, enabling efficient database development, querying, and maintenance.

**Employee**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| EmployeeID | VARCHAR2 | ID for each employee | 5 | | | HR Department |
| FirstName | VARCHAR2 | First Name of the employee | 50 | | | HR Department |
| LastName | VARCHAR2 | Last Name of the employee | 50 | | | HR Department |
| Position | VARCHAR2 | Employee position in company | 50 | | | HR Department |
| Salary | NUMBER | Yearly Salary of the employee | 7 | 0 | 2000000 | Payroll System |
| WorkCategory | VARCHAR2 | Department of the Employee | 50 | | | HR Department |
| EmployeeType | VARCHAR2 | Type of Employee | 1 | | | HR Department |

**WarehouseEmployee**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| EmployeeID | VARCHAR2 | ID for each employee | 50 | | | HR Department |
| WarehouseNumber | VARCHAR2 | Unique number/ID for Warehouses | 50 | | | Warehouse Table |

**DepartmentEmployee**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| EmployeeID | VARCHAR2 | ID for each employee | 20 | | | HR Department |
| DepartmentID | VARCHAR2 | ID for each department | 20 | | | Department Table |

**ProductionEmployee**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| EmployeeID | VARCHAR2 | ID for each employee | 20 | | | HR Department |
| LineNumber | VARCHAR2 | Unique number/ID for Production lines | 20 | | | ProductionLine Table |

**Department**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| DepartmentID | VARCHAR2 | Unique ID for each Department | 10 | | | Board |
| DepartmentName | VARCHAR2 | Department Name | 50 | | | Board |
| StreetAddress | VARCHAR2 | Street address of the department | 500 | | | Board |
| City | VARCHAR2 | City of the department | 50 | | | Board |
| NumberOfEmployees | NUMBER | Number of employees in the department | 2 | 0 | 10 | HR Department |
| PhoneNumber | NUMBER | Phone number for the department | 10 | | | HR Department |

**Product**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| ProductNumber | VARCHAR2 | Product ID/product number | 10 | | | Sales/Marketing Department |
| ProductType | VARCHAR2 | Type of Product | 50 | | | Sales/Marketing Department |
| ProductName | VARCHAR2 | Product Name | 50 | | | Sales/Marketing Department |
| DesignerID | VARCHAR2 | Employee ID of who designed the product | 4 | | | Employee Table |
| Price | NUMBER | Product price | 4 | 1 | 150 | Sales/Marketing Department |
| Cost | NUMBER | Making cost of the product | 4 | 1 | 100 | Sales/Marketing Department |
| Color | VARCHAR2 | Color of product | 50 | | | Design Department |
| Weight_(lbs) | NUMBER | Weight of the product in pounds | 5 | 0.01 | 10.00 | Design Department |

**RawMaterial**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| ProductNumber | VARCHAR2 | Product ID/product number | 30 | | | Product Table |
| RawmaterialName | VARCHAR2 | Raw Material for Product | 30 | | | RawMaterial Table |

**ProductionLine**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| LineNumber | VARCHAR2 | Production line number/ID | 10 | | | Plant Manager |
| LineCapacity_(items/hour) | NUMBER | Highest number of items a line can produce in an hour | 3 | 100 | 250 | Plant Manager |
| PhoneNumber | NUMBER | Contact for that Production line | 30 | 10 | 10 | Plant Manager |
| StreetAddress | VARCHAR2 | Street address of the production line | 500 | | | Board |
| Area | VARCHAR2 | Area of the production line | 50 | | | Board |
| City | VARCHAR2 | Two-character city code for production line | 20 | | | Board |
| Zipcode | NUMBER | Zipcode of the production line | 5 | | | Plant Manager |

**Vendor**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| VendorNumber | VARCHAR2 | Vendor number/id | 10 | | | Sales/Marketing Department |
| VendorName | VARCHAR2 | Name of Vendor | 50 | | | Sales/Marketing Department |
| StreetAddress | VARCHAR2 | Street address of the vendor | 100 | | | Vendor/Company Registration Document |
| City | VARCHAR2 | City of the vendor | 500 | | | Vendor/Company Registration Document |
| Phone Number | NUMBER | Vendor contact | 20 | | | Vendor/Company Registration Document |

**Warehouse**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| WarehouseNumber | VARCHAR2 | Warehouse number/ID | 10 | | | Sales/Marketing Department |
| StreetAddress | VARCHAR2 | Street address of the warehouse | 500 | | | Warehouse Registration Document |
| City | VARCHAR2 | City of the warehouse | 50 | | | Warehouse Registration Document |
| Phone | NUMBER | Warehouse contact | 10 | | | Warehouse Registration Document |

**Invoice**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| InvoiceNumber | VARCHAR2 | Unique number/ID for each Invoice | 50 | | | System Generated |
| Total_Amount_($) | NUMBER | Total amount given on invoice | 7 | 0 | 10000 | System Calculated |
| VendorNumber | VARCHAR2 | Vendor Number/ID | 50 | | | Vendor Table |
| DepartmentID | VARCHAR2 | ID for each Department | 50 | | | Board |

**SupplySchedule**

| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| SupplyCode | VARCHAR2 | Code/ID for every supply | 50 | | | System Generated |
| ProductNumber | VARCHAR2 | Product ID/product number | 50 | | | Product Table |
| RawmaterialName | VARCHAR2 | Raw Material for Product | 50 | | | RawMaterial Table |
| WarehouseNumber | VARCHAR2 | Warehouse number/ID | 50 | | | Warehouse Table |
| VendorNumber | VARCHAR2 | Vendor number/ID | 50 | | | Vendor Table |
| Supply_Date | DATE | Date of supply | | | | System Generated |

**VendorPaymentType**

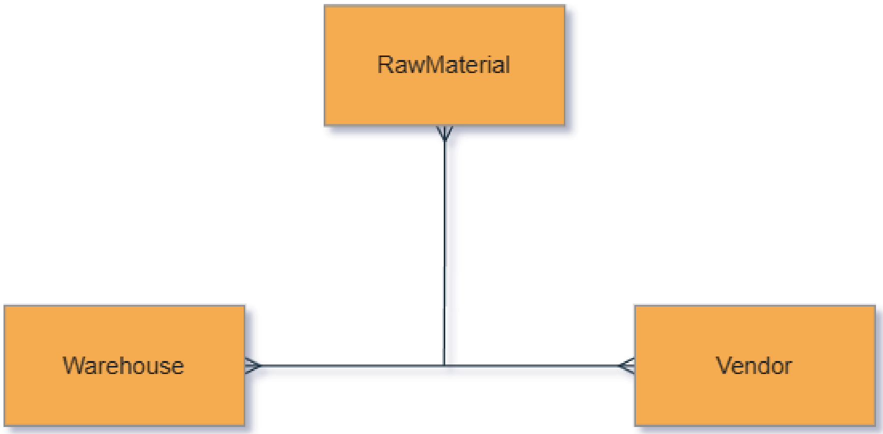
| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| VendorNumber | VARCHAR2 | Vendor Number/ID | 20 | | | Vendor Table |
| VendorPaymentType | VARCHAR2 | Payment mode | 50 | | | Selected Payment Mode |

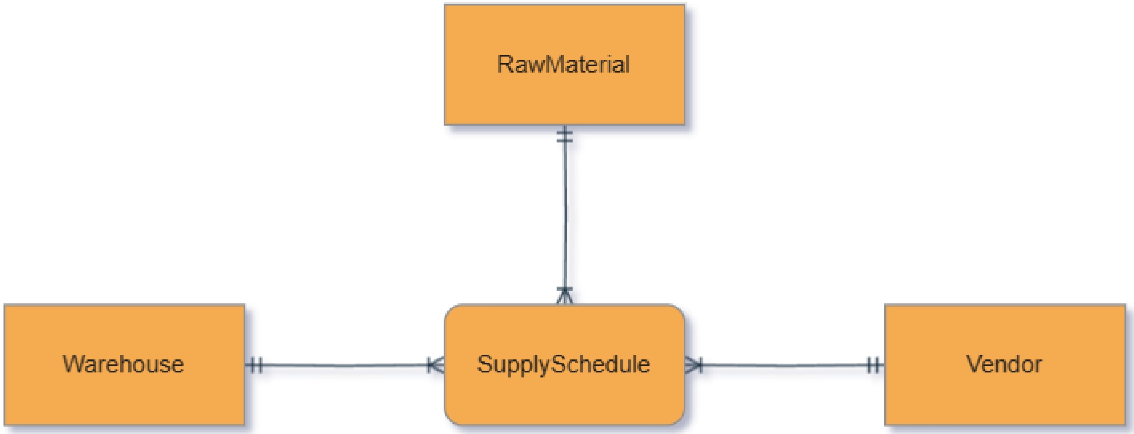
**ProductLineNumberRelation**

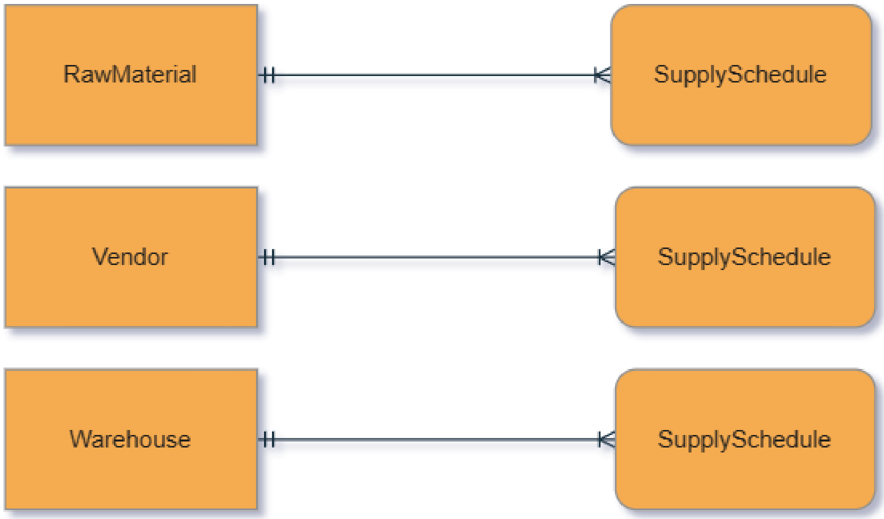
| Data Item Name | Data Item type | Data Item Description | Data Item length | Metadata Minimum | Metadata Maximum | Metadata Source |
|---|---|---|---|---|---|---|
| ProductNumber | VARCHAR2 | Product ID/ product number | 20 | | | Product Table |
| LineNumber | VARCHAR2 | Production line number/ID | 20 | | | ProductionLine Table |

# RELATIONSHIPS

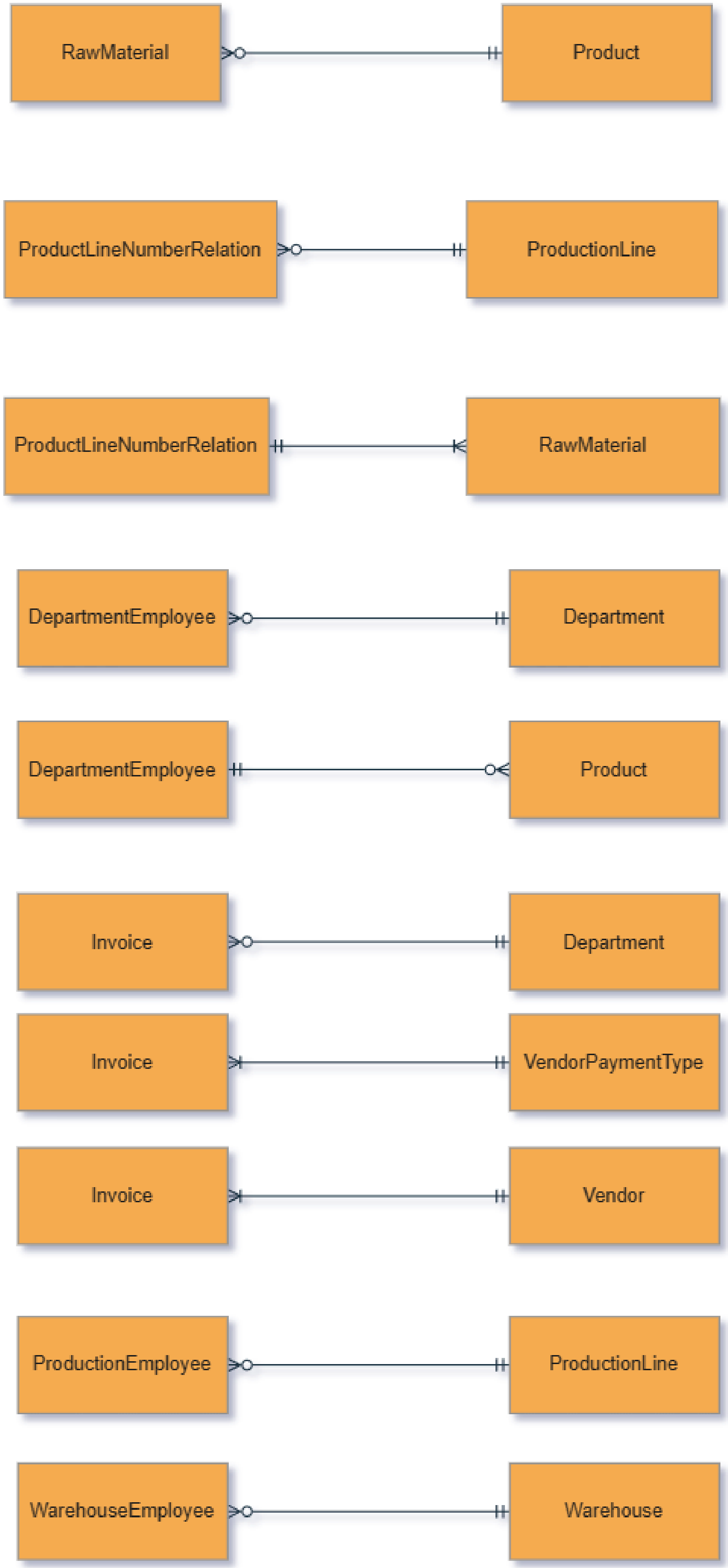
## Ternary Relationship:







## Binary Relationships:



- There are no entities with unary relationship.

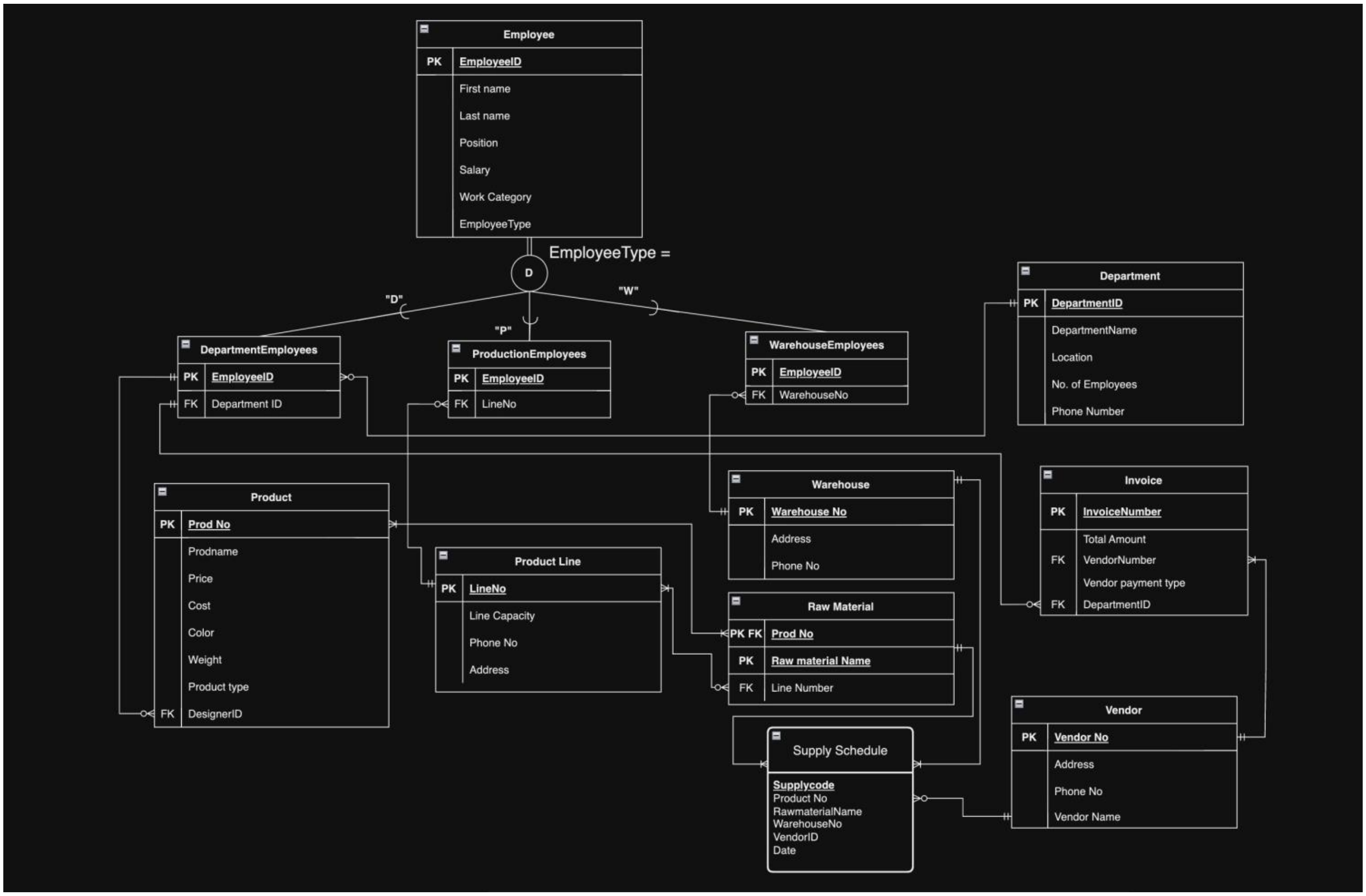
# PRE-NORMALIZATION EER DIAGRAM

- In the pre-normalization phase, we identified relationships between each entity and mapped it with appropriate cardinality as described in the previous step.

- We also performed a **specialization** process by creating super type and subtype relationships for the Employee table. Since the employee table contained information regarding all types of employees like department employees, warehouse employees and production line employees, it resulted in too many NULL values in the main employee table. Hence, we created a supertype Employee and sub types as DepartmentEmployee, ProductionEmployee and WarehouseEmployee. The relationship between supertype and subtype will be **complete and disjoint** as each employee must be part of exactly one of these types - department, warehouse or production line. The subtypes are divided based on the simple discriminator attribute: EmployeeType = “D” or “P” or “W”.

- We have also resolved many-to-many ternary relationship between Vendor, Warehouse and RawMaterial by introducing a new associative entity SupplySchedule and created three binary relationships with supply schedule.

After performing the above steps, we have obtained the following diagram:



# PRIMARY AND FOREIGN KEYS

| Employee | |
|---|---|
| Primary Key | EmployeeID |

| WarehouseEmployee | |
|---|---|
| Primary Key | EmployeeID |
| Foreign Key | WarehouseNumber |

| DepartmentEmployee | |
|---|---|
| Primary Key | EmployeeID |
| Foreign Key | DepartmentID |

| ProductionEmployee | |
|---|---|
| Primary Key | EmployeeID |
| Foreign Key | LineNumber |

| Department | |
|---|---|
| Primary Key | DepartmentID |

| Product | |
|---|---|
| Primary Key | ProductNumber |
| Foreign Key | DesignerID |

| RawMaterial | |
|---|---|
| Primary Key | ProductNumber |
| Primary Key | RawmaterialName |
| Foreign Key | ProductNumber |

| ProductionLine | |
|---|---|
| Primary Key | LineNumber |

| Vendor | |
|---|---|
| Primary Key | VendorNumber |

| Warehouse | |
|---|---|
| Primary Key | WarehouseNumber |

| Invoice | |
|---|---|
| Primary Key | InvoiceNumber |
| Foreign Key | VendorNumber |
| Foreign Key | DepartmentID |

| SupplySchedule | |
|---|---|
| Primary Key | SupplyCode |
| Foreign Key | ProductNumber |
| Foreign Key | WarehouseNumber |
| Foreign Key | VendorNumber |
| Foreign Key | RawmaterialName |

| VendorPaymentType | |
|---|---|
| Primary Key | VendorNumber |

| ProductLineNumberRelation | |
|---|---|
| Primary Key | ProductNumber |
| Foreign Key | LineNumber |

# DATA CLEANING AND MODIFICATION

## 1) Department:

- Split the multivalued attribute 'Address' into two separate attributes: 'StreetAddress' and 'City'.
- Renamed the attribute 'No of Employees' to 'NoOfEmployees'.
- Modified instances of 'PhoneNumber' to consist of 10 digits.

## 2) Employee:

- Introduced a new attribute 'EmployeeType', derived from 'EmployeeCategory'.
- Trimmed extra spaces in 4 instances of 'FirstName'.

## 3) Invoice:

- Renamed the attribute 'Total Amount ($)' to 'Total_Amount_($)'.
- Introduced a new attribute 'DepartmentID'.
- Trimmed extra spaces on the right side of instances of 'InvoiceNumber'.

## 4) Product:

- Renamed the attribute 'Prodcutype' to 'ProductType'.
- Trimmed extra spaces on the right side of 3 instances in 'ProductNumber'.
- Trimmed extra spaces on the right side of 39 instances in 'ProductName'.
- Modified 'Weight (lbs)' to 'Weight_(lbs)'.
- Rounded all instances of 'Weight_(lbs)' to two decimal points.

## 5) ProductionLine:

- Renamed the table from 'Production Line' to 'ProductionLine'.
- Trimmed blank spaces before and after 'LineNumber' and 'Line Capacity (items/hours)'.
- Modified the attribute 'Line Capacity (items/hours)' to 'LineCapacity_(items/hour)'.
- Trimmed blank spaces in all instances of 'LineNumber'.

## 6) ProductLineNumberRelation:

- Corrected the LineNumber for the product 'PRO55' from 'L010' to 'L014'.
- Rectified the LineNumber for the product 'PRO67' from 'L020' to 'L012'.

## 7) Rawmaterial:

- Altered the table name from 'Raw Material' to 'RawMaterial'.
- Adjusted instances of 'Card stock' within the 'RawmaterialName' attribute, converting occurrences to 'Card Stock'.

## 8) SupplySchedule:

- Renamed the table from 'Supply Schedule' to 'SupplySchedule'.
- Corrected value ‘PERO30’ to ‘PRO30’ in ‘ProductNumber’.
- Trimmed extra spaces on the right side of attributes 'SupplyCode', 'WarehouseNumber', 'VendorID', and 'Date'.
- Updated the attribute name 'VendorID' to 'VendorNumber'.
- Modified instances of 'Card stock' to 'Card Stock' within the 'RawmaterialName' attribute.

## 9) Vendor:

- Modified instances of 'PhoneNumber' to contain 10-digit numbers.
- Split the multivalued attribute 'Address' into two distinct attributes: 'StreetAddress' and 'City'.
- Trimmed extra spaces in the instances of 'VendorNumber' and 'VendorName'.

## 10) Warehouse:

- Modified 'PhoneNumber' instances to consist of 10-digit digits.
- Trimmed extra spaces on the right side of attributes 'WarehouseNumber' and 'Address'.
- Trimmed extra spaces on the left side of all instances of 'PhoneNumber'.
- Trimmed extra spaces on both sides of all instances of 'Address'.
- Trimmed extra spaces on the right side of all instances of 'Warehouse'.
- Split the multivalued attribute 'Address' into two separate attributes: 'StreetAddress' and 'City'.

# INTEGRITY CONSTRAINTS

## Referential Integrity:

- Within our relational database, we have implemented specific integrity constraints aimed at safeguarding the accuracy and consistency of our data:

  1) **Employee subtypes have cascade delete constraint**:
     To ensure that removing an **Employee** also deletes related subtype entries, such as ProductionEmployee, WarehouseEmployee, and DepartmentEmployee. This is achieved through foreign keys with **ON DELETE CASCADE** in the subtype tables.

  2) **Product-raw material has cascade delete constraint**:
     When a Product record is deleted, all corresponding RawMaterial records exclusively linked to that product are also deleted. This prevents unused raw materials from lingering in the database.

  3) **Department-DepartmentEmployee has cascade delete constraint**:
     It dictates that deleting a department record also removes all entries in the DepartmentEmployee table referencing that department. This ensures the removal of all employee associations with a nonexistent department when the department is closed.

  4) **Warehouse – WarehouseEmployee has cascade delete constraint**:
     The cascade delete constraint guarantees that deleting a Warehouse record results in the removal of all WarehouseEmployee records associated with that warehouse. This prevents records of employees being linked to non-existent warehouses.

  5) **ProductionLine – ProductionEmployee has cascade delete constraint**:
     It ensures that when a ProductionLine is deleted, all associated ProductionEmployee records referencing the deleted production line are also removed. This prevents employees from being associated with discontinued production lines.

- To summarize, a cascade delete constraint, a form of referential action tied to foreign keys, ensures the deletion of all linked records in the child table when the associated record in the parent table is deleted. This automatic removal of dependent records maintains referential integrity and prevents the existence of orphan records referencing non-existent data.

## Entity Integrity:

- It is specified that the primary key or composite primary key must not be NULL and must be unique within each table. We have verified this constraint for all tables, and they all adhere to the entity integrity requirement.

## Domain Constraints:

The enforcement of these constraints in a database is typically achieved through the creation of tables using SQL **CREATE TABLE** statements. These statements include specifications for data types, lengths, and potentially incorporate **CHECK** constraints to establish minimum and maximum values, ensuring the integrity and accuracy of the data within the database.

- **DEPARTMENT Table Constraints:**

  1) **DepartmentID**: VARCHAR(4)
     Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

  2) **DepartmentName**: VARCHAR(50)
     Must be a string with a maximum length of 50 characters. No minimum length specified.

  3) **Location**: VARCHAR(50)
     Must be a string with a maximum length of 50 characters. No minimum length specified.

  4) **No. of Employees**: Integer(2)
     Must be a 2-digit integer ranging from 0 to 10.

  5) **Phone Number**: Integer(10)
     Must be a 10-digit integer representing a phone number.

- **INVOICE Table Constraints:**

  1) **Invoice Number**: VARCHAR(7)
     Must be a 7-character string.No minimum or maximum value constraints indicated.

  2) **Total Amount**: Integer(5)
     Must be a 5-digit integer between 0 and 10000 inclusively.

  3) **Vendor Number**: VARCHAR(4)
     Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

- **LineNumber Table Constraints:**

  1) **Product Number**: VARCHAR(5)
     Must be a string with a maximum length of 5 characters. No minimum length specified.

  2) **Line Number**: VARCHAR(4)
     Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

- **PRODUCT Table Constraints:**

1) **Product Number**: VARCHAR(5)
   Must be a string with a maximum length of 5 characters. No minimum length specified.

2) **Product Type**: VARCHAR(20)
   Must be a string with a maximum length of 20 characters. No minimum length specified.

3) **Name**: VARCHAR(50)
   Must be a string with a maximum length of 50 characters. No minimum length specified.

4) **DesignerID**: VARCHAR(4)
   Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

5) **Price**: Integer(3)
   Must be a 3-digit integer between 1 and 150 inclusively.

6) **Cost**: Integer(3)
   Must be a 3-digit integer between 1 and 100 inclusively.

7) **Color**: VARCHAR(20)
   Must be a string with a maximum length of 20 characters. No minimum length specified.

8) **Weight**: Decimal(4,2)
   Must be a decimal number with up to 4 digits in total and 2 digits after the decimal point, ranging from 0.01 to 10.

- **PRODUCTION LINE Table Constraints:**

1) **Line Number**: VARCHAR(4)
   Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

2) **Line Capacity**: Integer(3)
   Must be a 3-digit integer between 100 and 250 inclusively.

3) **Phone Number**: Integer(10)
   Must be a 10-digit integer representing a phone number. It's specified to be exactly '10', which might be an error; typically, it should be a range or set to allow various valid phone numbers.

4) **Address**: VARCHAR(50)
   Must be a string with a maximum length of 50 characters. No minimum length specified.

- **VENDOR Table Constraints:**

  1) **Vendor Number**: VARCHAR(4)
     Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

  2) **Vendor Name**: VARCHAR(50)
     Must be a string with a maximum length of 50 characters. No minimum length specified.

  3) **Address**: VARCHAR(50)
     Must be a string with a maximum length of 50 characters. No minimum length specified.

  4) **Phone Number**: Integer(10)
     Must be a 10-digit integer representing a phone number.

- **VendorPayment Table Constraints:**

  1) **Vendor Number**: VARCHAR(4)
     Must be a string of exactly 4 characters. No minimum or maximum value constraints indicated.

  2) **Vendor Payment Type**: VARCHAR(20)
     Must be a string with a maximum length of 20 characters. No minimum length specified.

- **WAREHOUSE Table Constraints:**

  1) **Warehouse Number**: VARCHAR(5)
     Must be a string with a maximum length of 5 characters. No minimum length specified.

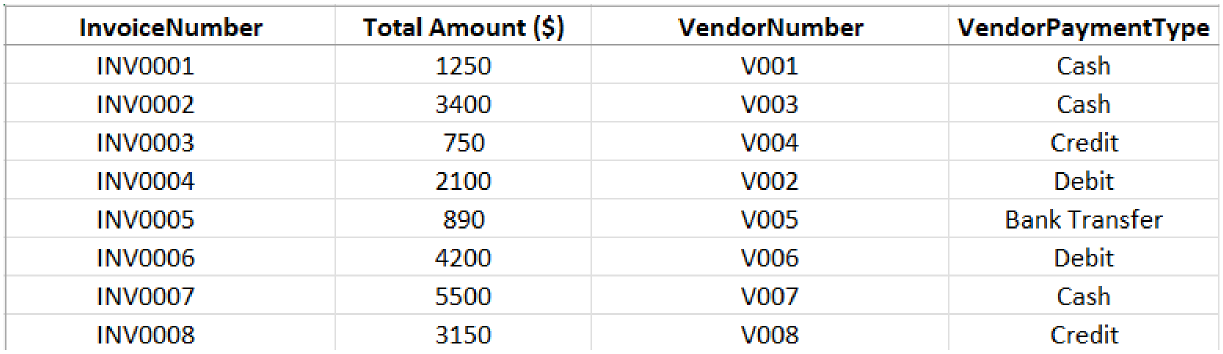
  2) **Address**: VARCHAR(50)
     Must be a string with a maximum length of 50 characters. No minimum length specified.

  3) **Phone Number**: Integer(10)
     Must be a 10-digit integer representing a phone number.

# ANOMALIES

Anomalies were identified in the following two tables:

## 1) Invoice Table:



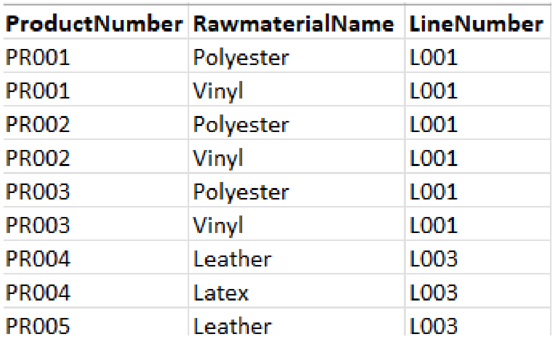
| InvoiceNumber | Total Amount ($) | VendorNumber | VendorPaymentType |
|---|---|---|---|
| INV0001 | 1250 | V001 | Cash |
| INV0002 | 3400 | V003 | Cash |
| INV0003 | 750 | V004 | Credit |
| INV0004 | 2100 | V002 | Debit |
| INV0005 | 890 | V005 | Bank Transfer |
| INV0006 | 4200 | V006 | Debit |
| INV0007 | 5500 | V007 | Cash |
| INV0008 | 3150 | V008 | Credit |

- Insertion Anomaly:
  When adding a new invoice for vendor V001, redundant information about the VendorPaymentType "Cash" must also be included.

- Deletion Anomaly:
  Deleting vendor V005 could result in the loss of the record associated with the "Bank Transfer" payment type since this payment type is unique to that vendor. This scenario is identified as a deletion anomaly.

- Updation Anomaly:
  To update the payment type for a specific vendor, such as V002, it is necessary to update all records containing invoices for the same vendor. This situation signifies an updating anomaly.

## 2) Rawmaterial Table:

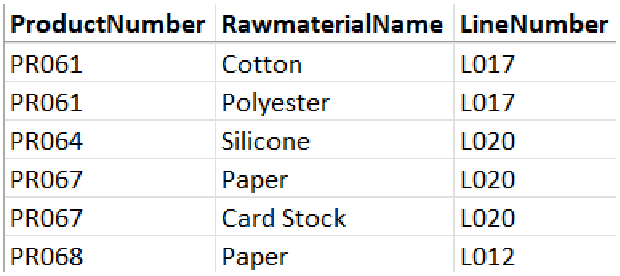


| ProductNumber | RawmaterialName | LineNumber |
|---|---|---|
| PR001 | Polyester | L001 |
| PR001 | Vinyl | L001 |
| PR002 | Polyester | L001 |
| PR002 | Vinyl | L001 |
| PR003 | Polyester | L001 |
| PR003 | Vinyl | L001 |
| PR004 | Leather | L003 |
| PR004 | Latex | L003 |
| PR005 | Leather | L003 |

- Insertion Anomaly:
  Adding a new raw material for product PR001 introduces redundant information of the associated line number L001 with PR001.

- Updation Anomaly:
  If the production line for any product changes, updating records for that specific product requires updating all related records. For instance, if one record of PR001 is updated with L002, line number information in 2 records of PR001 must also be updated.

- Deletion Anomaly:
  Since there is only one record for the "Silicone" raw material in the entire Rawmaterial table, deleting product number PR064 would result in the loss of information for this raw material. This situation is identified as a deletion anomaly.



| ProductNumber | RawmaterialName | LineNumber |
|---|---|---|
| PR061 | Cotton | L017 |
| PR061 | Polyester | L017 |
| PR064 | Silicone | L020 |
| PR067 | Paper | L020 |
| PR067 | Card Stock | L020 |
| PR068 | Paper | L012 |

**Post finding these anomalies, we have performed normalization which is covered in the next step.**

# NORMALIZATION

The process of normalization was executed in several stages:

1) The initial step towards achieving **First Normal Form (1NF)** involved the elimination of composite attributes from entity sets like address into separate attributes like StreetAddress, Area, City, Zipcode.

2) Advancing to **Second Normal Form (2NF)**, we addressed partial dependencies by segregating attributes that relied solely on a portion of a composite primary key into separate tables like in the initial Raw Material table, LineNumber depended solely on ProductNumber, which was a component of the composite primary key. The creation of the ProductLineNumberRelation table removed this dependency, clearly defining the relationship between LineNumber and ProductNumber and thereby eliminating the partial dependency.

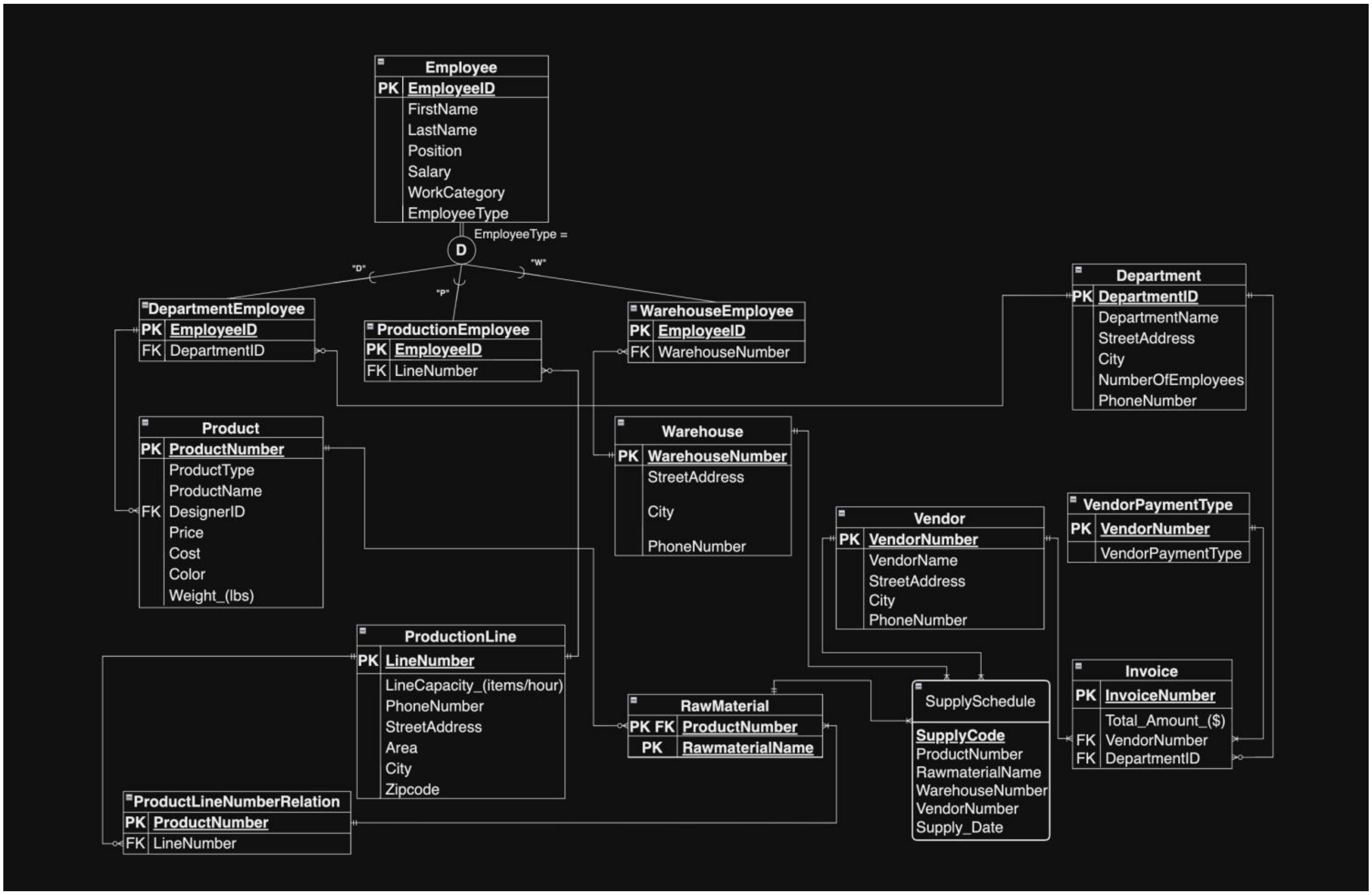
3) In the transition to **Third Normal Form (3NF)**, we eliminated transitive dependencies, like in the Invoice table, the VendorPaymentType was reliant on VendorNumber, a non-key attribute which created a transitive dependency since VendorNumber itself depended on the primary key, InvoiceNumber. To address this issue, a separate VendorPaymentType table was established, with VendorNumber as its primary key, effectively removing the transitive dependency. As a result, the Invoice table now incorporates VendorNumber as a foreign key, which is referenced to the VendorNumber primary key in the VendorPaymentType table.

Post normalization, every attribute in all tables have full dependency, i.e. fully relies on its corresponding primary key. For instance, in the Product table, attributes like ProductType, ProductName, DesignerID, Price, Cost, Color and Weight_(lbs) are entirely dependent on ProductNumber, which is the primary key.

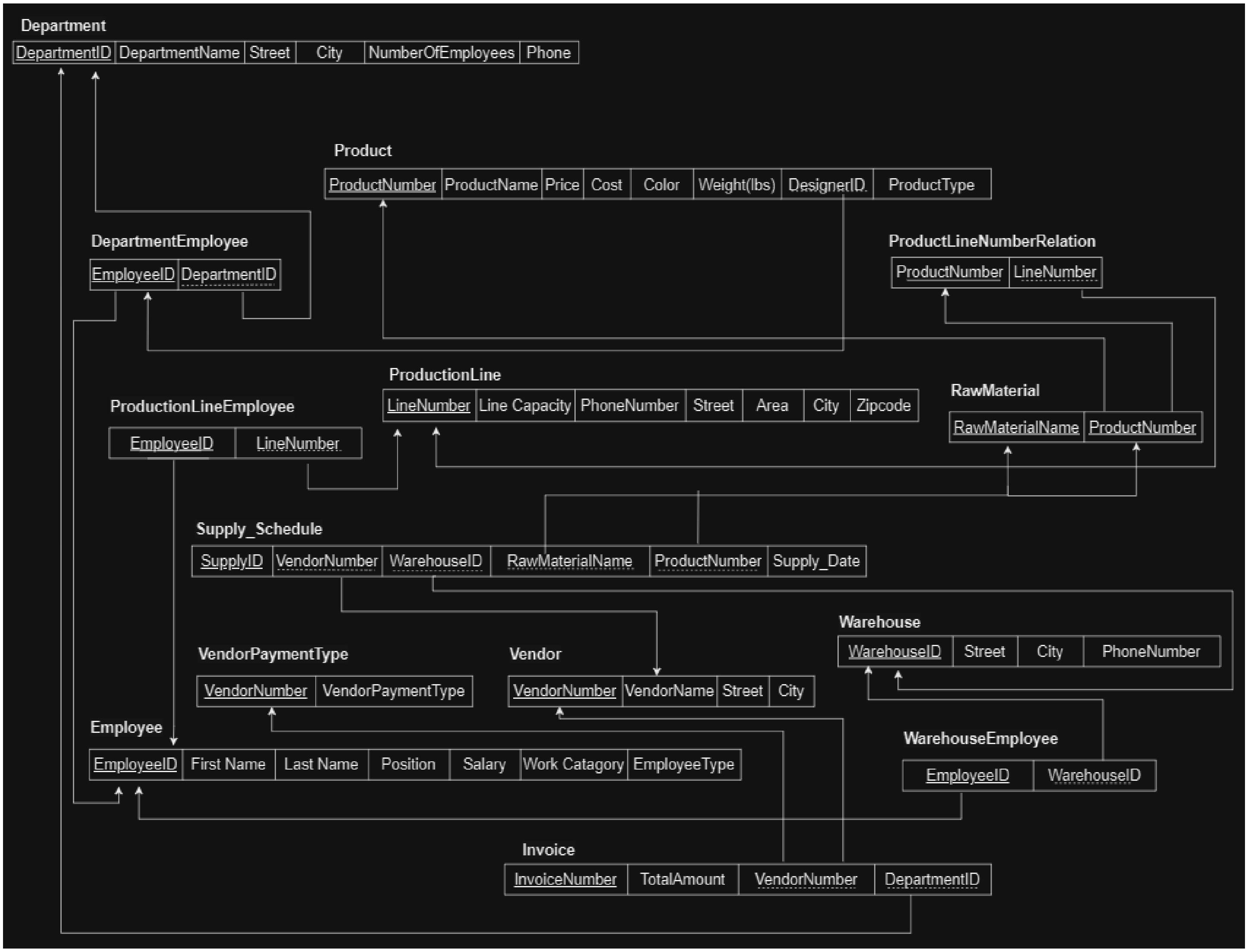
The normalization procedure guaranteed that the final Enhanced Entity-Relationship (EER) model aligns with the first three normal forms (1NF, 2NF, and 3NF). This alignment eradicates redundant data, confirms that data dependencies make logical sense, and enhances both the integrity and the efficiency of the database.

After performing normalization, we have achieved our finalized EER Diagram and Relational model as shown in the next step. Entities pertaining to the finalized EER Diagram are added to the Oracle Apex database and SQL operations will be performed on those tables.

# POST-NORMALIZATION EER DIAGRAM



# RELATIONAL SCHEMA



Department

| DepartmentID | DepartmentName | Street | City | NumberOfEmployees | Phone |
|---|---|---|---|---|---|

Product

| ProductNumber | ProductName | Price | Cost | Color | Weight(lbs) | DesignerID | ProductType |
|---|---|---|---|---|---|---|---|

DepartmentEmployee

| EmployeeID | DepartmentID |
|---|---|

ProductLineNumberRelation

| ProductNumber | LineNumber |
|---|---|

ProductionLine

| LineNumber | Line Capacity | PhoneNumber | Street | Area | City | Zipcode |
|---|---|---|---|---|---|---|

ProductionLineEmployee

| EmployeeID | LineNumber |
|---|---|

RawMaterial

| RawMaterialName | ProductNumber |
|---|---|

Supply_Schedule

| SupplyID | VendorNumber | WarehouseID | RawMaterialName | ProductNumber | Supply_Date |
|---|---|---|---|---|---|

Warehouse

| WarehouseID | Street | City | PhoneNumber |
|---|---|---|---|

VendorPaymentType

| VendorNumber | VendorPaymentType |
|---|---|

Vendor

| VendorNumber | VendorName | Street | City |
|---|---|---|---|

Employee

| EmployeeID | First Name | Last Name | Position | Salary | Work Catagory | EmployeeType |
|---|---|---|---|---|---|---|

WarehouseEmployee

| EmployeeID | WarehouseID |
|---|---|

Invoice

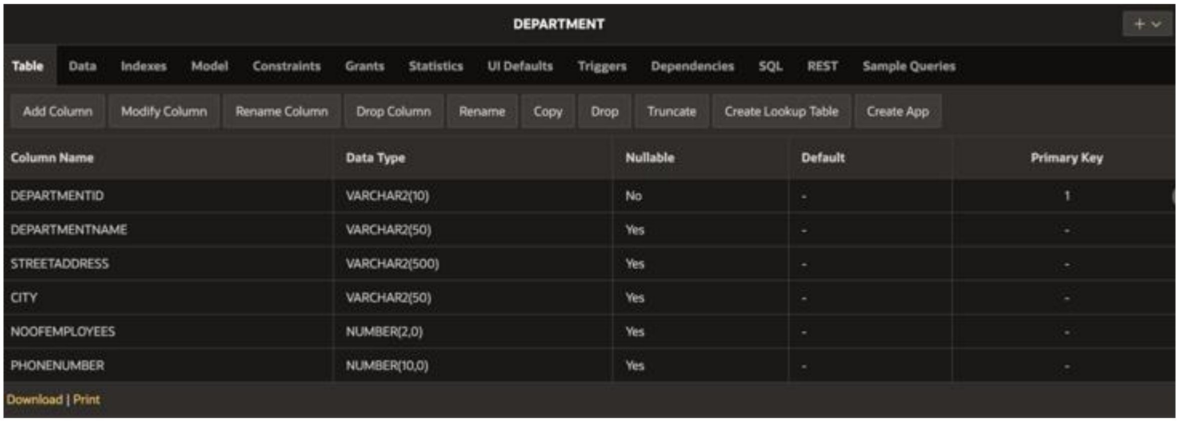
| InvoiceNumber | TotalAmount | VendorNumber | DepartmentID |
|---|---|---|---|

# TASK 5:

Table provided with a screenshot of the table itself & the screenshot of the associated SQL code:

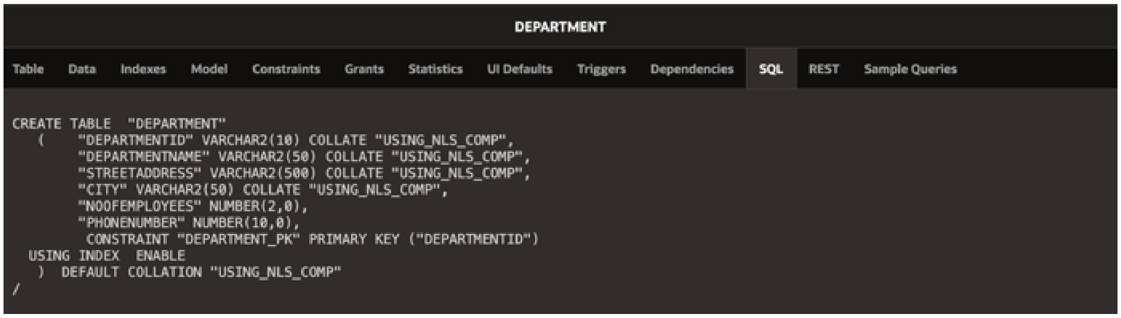
- ## DEPARTMENT:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| DEPARTMENTID | VARCHAR2(10) | No | - | 1 |
| DEPARTMENTNAME | VARCHAR2(50) | Yes | - | - |
| STREETADDRESS | VARCHAR2(500) | Yes | - | - |
| CITY | VARCHAR2(50) | Yes | - | - |
| NOOFEMPLOYEES | NUMBER(2,0) | Yes | - | - |
| PHONENUMBER | NUMBER(10,0) | Yes | - | - |

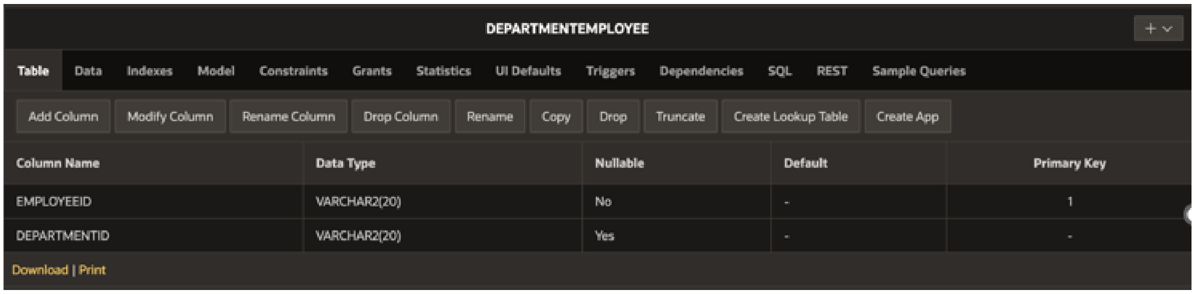
SQL:



```
CREATE TABLE  "DEPARTMENT"
   (    "DEPARTMENTID" VARCHAR2(10) COLLATE "USING_NLS_COMP",
        "DEPARTMENTNAME" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "STREETADDRESS" VARCHAR2(500) COLLATE "USING_NLS_COMP",
        "CITY" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "NOOFEMPLOYEES" NUMBER(2,0),
        "PHONENUMBER" NUMBER(10,0),
         CONSTRAINT "DEPARTMENT_PK" PRIMARY KEY ("DEPARTMENTID")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
```

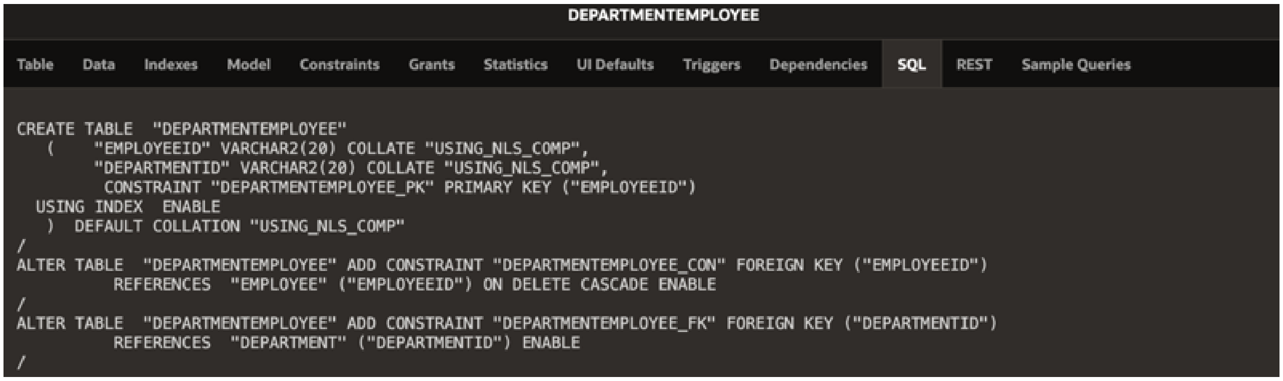
- ## DEPARTMENTEMPLOYEE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| EMPLOYEEID | VARCHAR2(20) | No | - | 1 |
| DEPARTMENTID | VARCHAR2(20) | Yes | - | - |

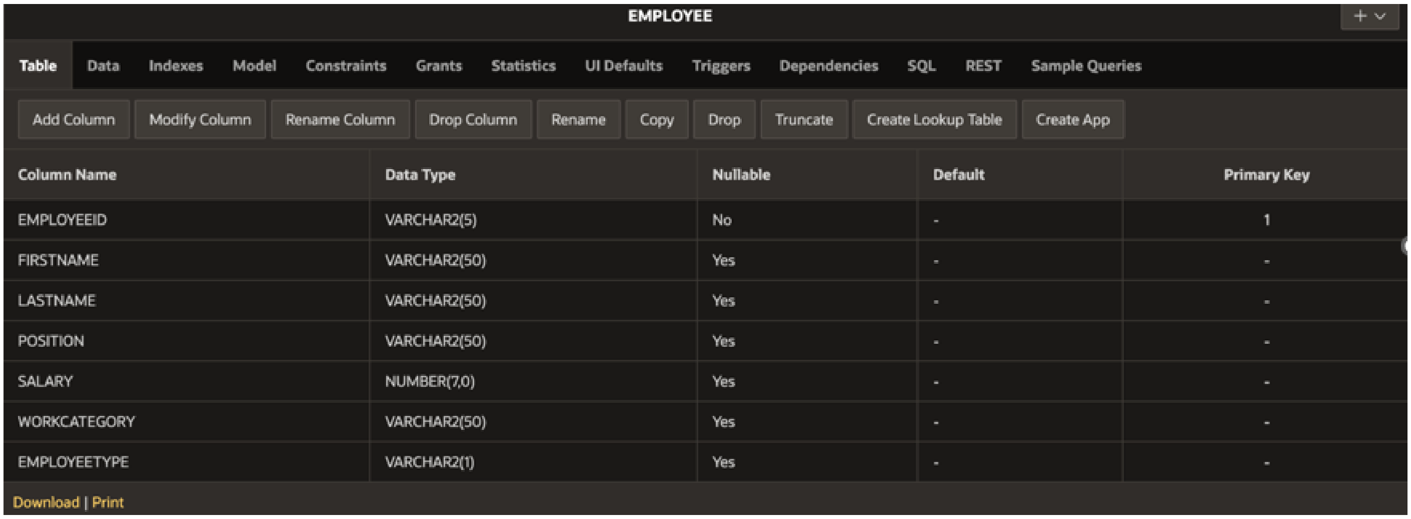
SQL:



```
CREATE TABLE  "DEPARTMENTEMPLOYEE"
   (    "EMPLOYEEID" VARCHAR2(20) COLLATE "USING_NLS_COMP",
        "DEPARTMENTID" VARCHAR2(20) COLLATE "USING_NLS_COMP",
         CONSTRAINT "DEPARTMENTEMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "DEPARTMENTEMPLOYEE" ADD CONSTRAINT "DEPARTMENTEMPLOYEE_CON" FOREIGN KEY ("EMPLOYEEID")
          REFERENCES  "EMPLOYEE" ("EMPLOYEEID") ON DELETE CASCADE ENABLE
/
ALTER TABLE  "DEPARTMENTEMPLOYEE" ADD CONSTRAINT "DEPARTMENTEMPLOYEE_FK" FOREIGN KEY ("DEPARTMENTID")
          REFERENCES  "DEPARTMENT" ("DEPARTMENTID") ENABLE
/
```

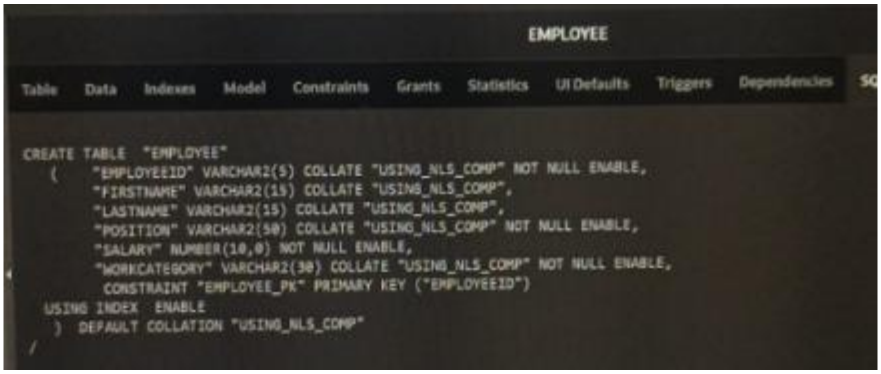
- EMPLOYEE:

TABLE



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| EMPLOYEEID | VARCHAR2(5) | No | - | 1 |
| FIRSTNAME | VARCHAR2(50) | Yes | - | - |
| LASTNAME | VARCHAR2(50) | Yes | - | - |
| POSITION | VARCHAR2(50) | Yes | - | - |
| SALARY | NUMBER(7,0) | Yes | - | - |
| WORKCATEGORY | VARCHAR2(50) | Yes | - | - |
| EMPLOYEETYPE | VARCHAR2(1) | Yes | - | - |

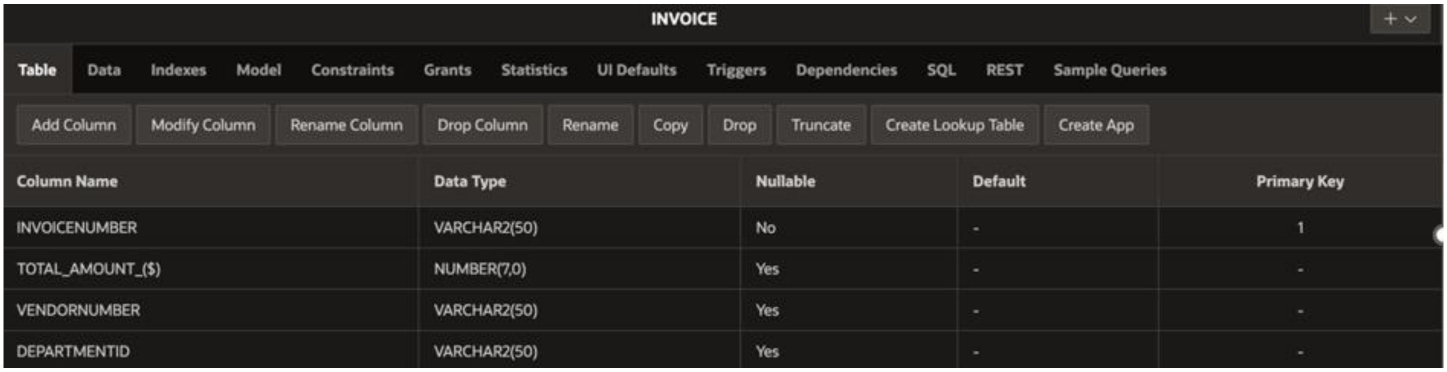
SQL:



```
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"
/
```

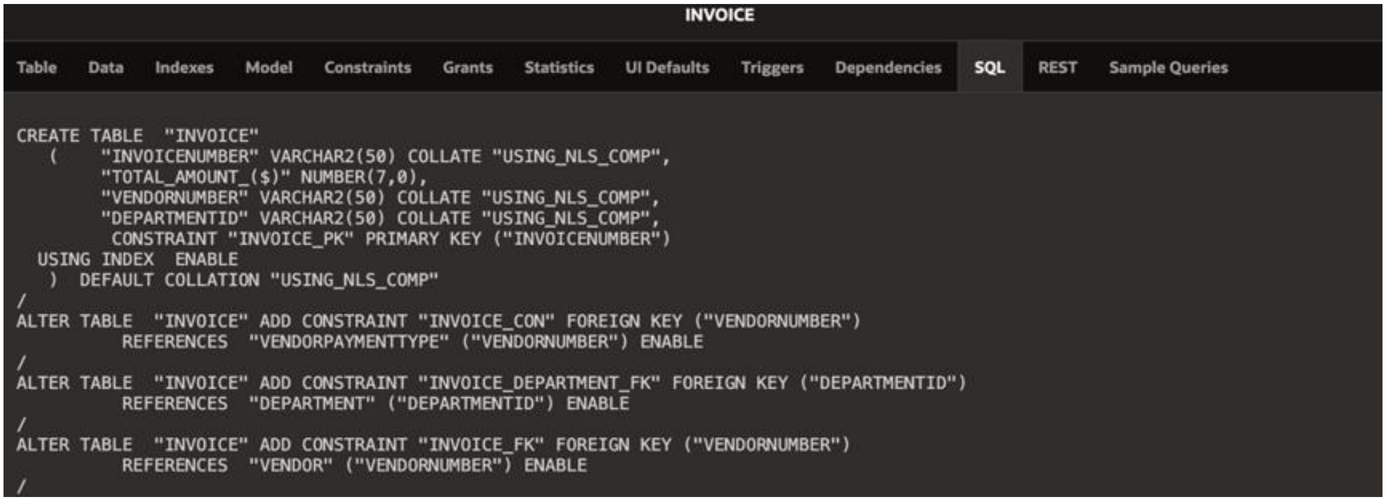
- INVOICE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| INVOICENUMBER | VARCHAR2(50) | No | - | 1 |
| TOTAL_AMOUNT_($) | NUMBER(7,0) | Yes | - | - |
| VENDORNUMBER | VARCHAR2(50) | Yes | - | - |
| DEPARTMENTID | VARCHAR2(50) | Yes | - | - |

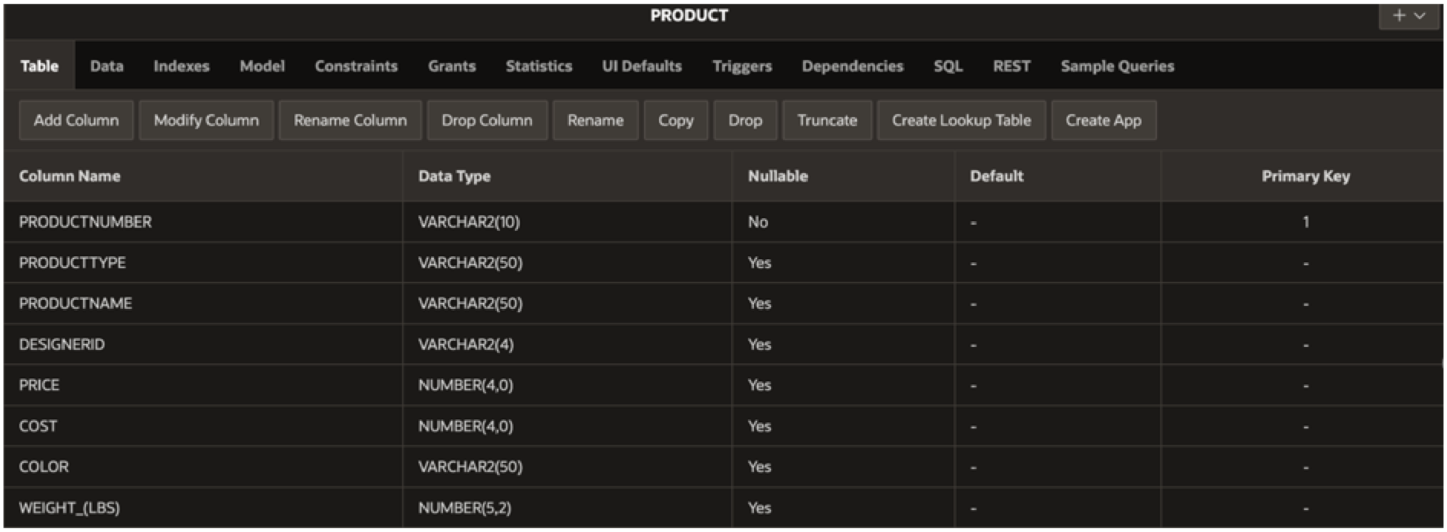
SQL:



```
CREATE TABLE  "INVOICE"
   (    "INVOICENUMBER" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "TOTAL_AMOUNT_($)" NUMBER(7,0),
        "VENDORNUMBER" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "DEPARTMENTID" VARCHAR2(50) COLLATE "USING_NLS_COMP",
         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  "VENDORPAYMENTTYPE" ("VENDORNUMBER") ENABLE
/
ALTER TABLE  "INVOICE" ADD CONSTRAINT "INVOICE_DEPARTMENT_FK" FOREIGN KEY ("DEPARTMENTID")
          REFERENCES  "DEPARTMENT" ("DEPARTMENTID") ENABLE
/
ALTER TABLE  "INVOICE" ADD CONSTRAINT "INVOICE_FK" FOREIGN KEY ("VENDORNUMBER")
          REFERENCES  "VENDOR" ("VENDORNUMBER") ENABLE
/
```

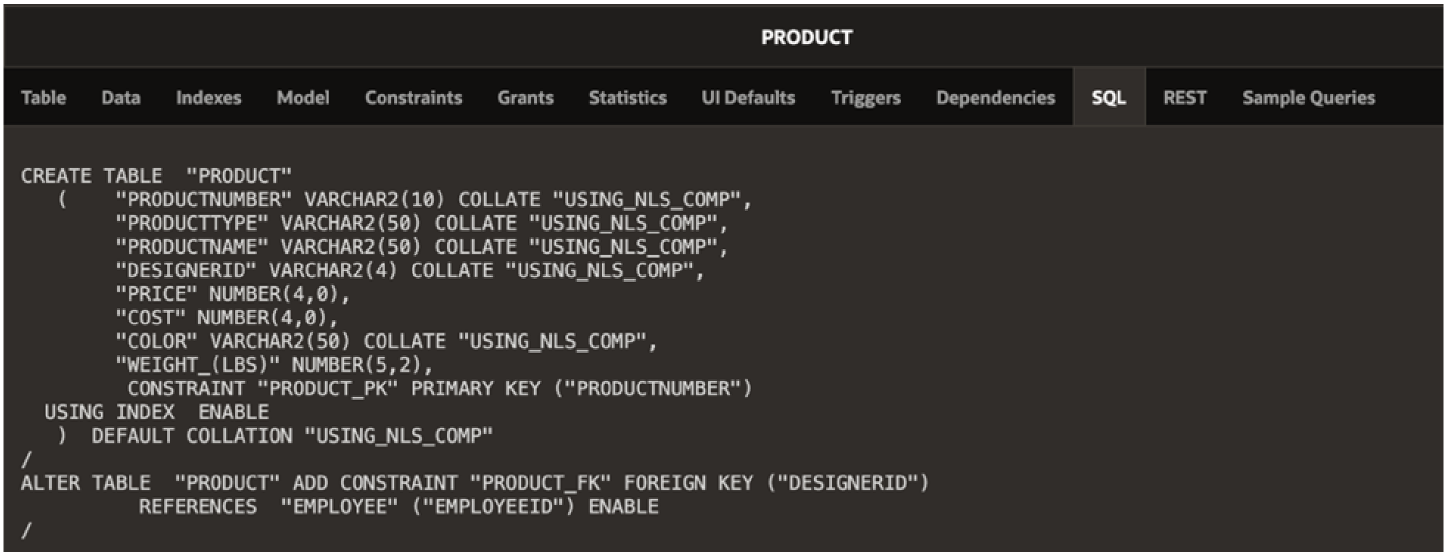
- PRODUCT:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| PRODUCTNUMBER | VARCHAR2(10) | No | - | 1 |
| PRODUCTTYPE | VARCHAR2(50) | Yes | - | - |
| PRODUCTNAME | VARCHAR2(50) | Yes | - | - |
| DESIGNERID | VARCHAR2(4) | Yes | - | - |
| PRICE | NUMBER(4,0) | Yes | - | - |
| COST | NUMBER(4,0) | Yes | - | - |
| COLOR | VARCHAR2(50) | Yes | - | - |
| WEIGHT_(LBS) | NUMBER(5,2) | Yes | - | - |

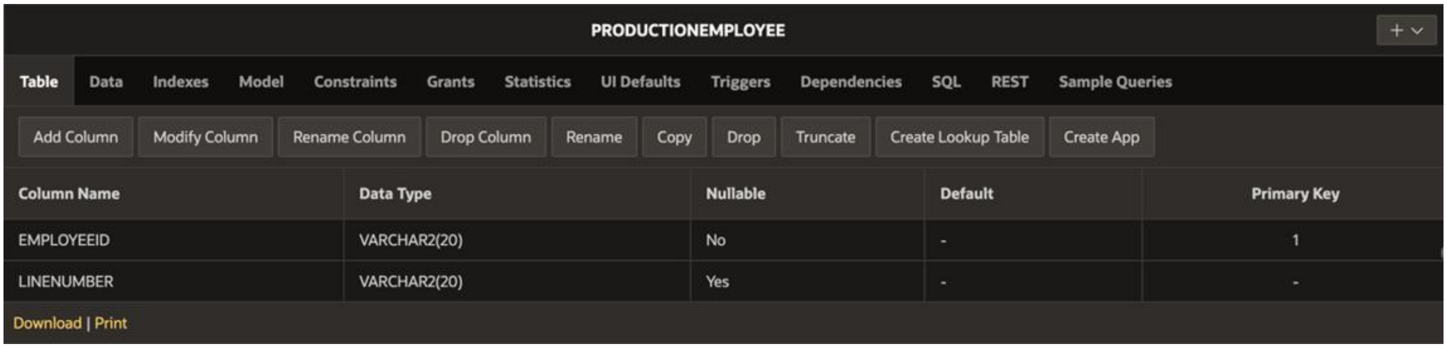
SQL:



```
CREATE TABLE  "PRODUCT"
   (    "PRODUCTNUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
        "PRODUCTTYPE" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "PRODUCTNAME" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "DESIGNERID" VARCHAR2(4) COLLATE "USING_NLS_COMP",
        "PRICE" NUMBER(4,0),
        "COST" NUMBER(4,0),
        "COLOR" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "WEIGHT_(LBS)" NUMBER(5,2),
         CONSTRAINT "PRODUCT_PK" PRIMARY KEY ("PRODUCTNUMBER")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "PRODUCT" ADD CONSTRAINT "PRODUCT_FK" FOREIGN KEY ("DESIGNERID")
          REFERENCES  "EMPLOYEE" ("EMPLOYEEID") ENABLE
/
```

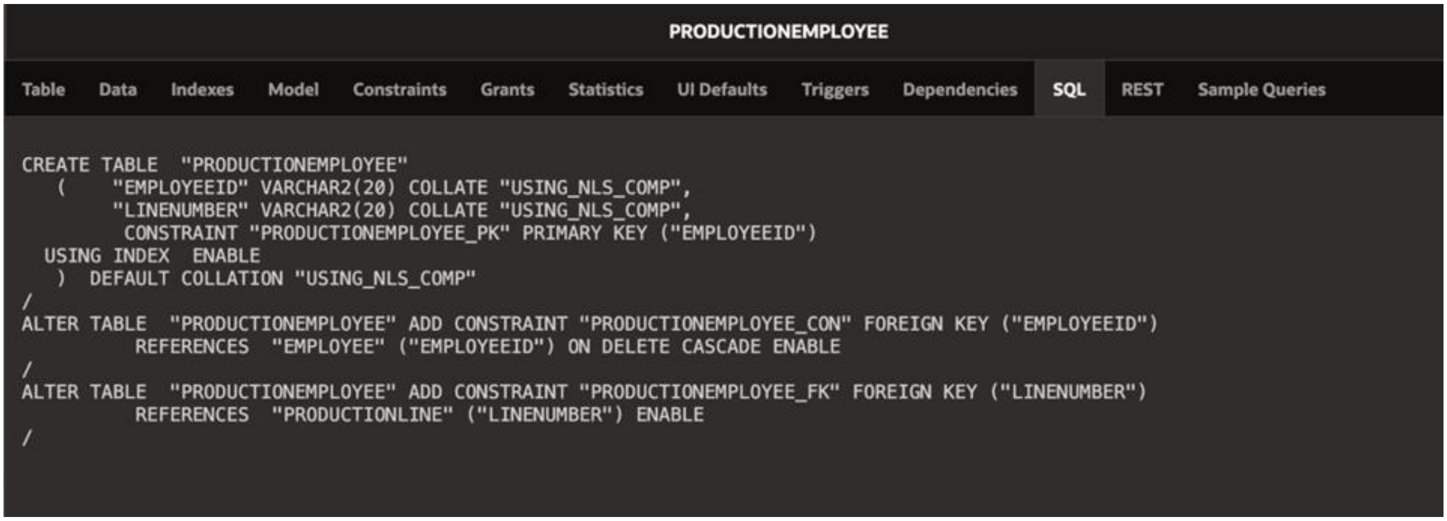
- PRODUCTIONEMPLOYEE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| EMPLOYEEID | VARCHAR2(20) | No | - | 1 |
| LINENUMBER | VARCHAR2(20) | Yes | - | - |

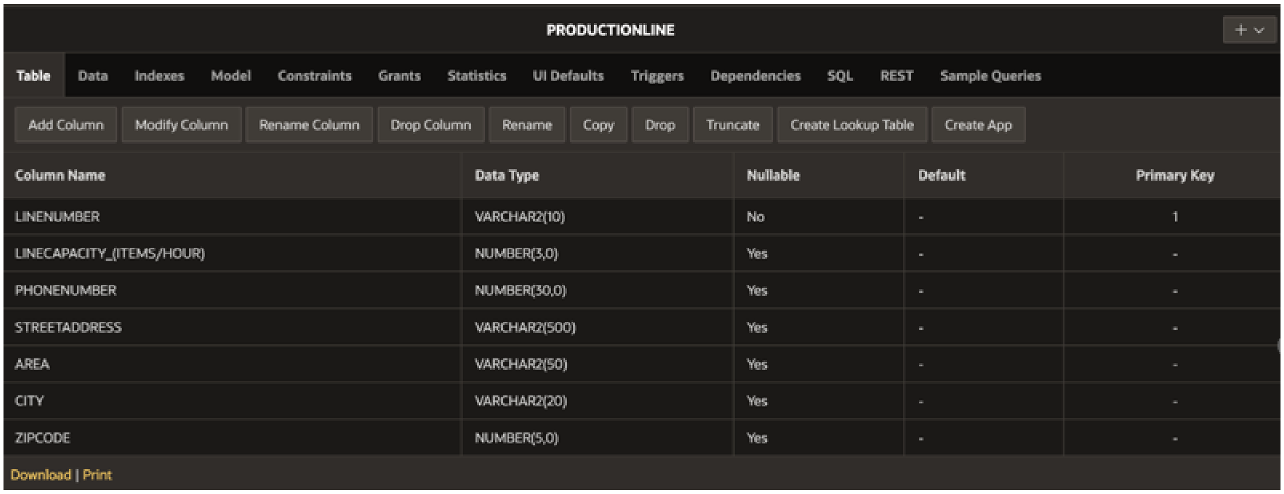
SQL:



```
CREATE TABLE  "PRODUCTIONEMPLOYEE"
   (    "EMPLOYEEID" VARCHAR2(20) COLLATE "USING_NLS_COMP",
        "LINENUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP",
         CONSTRAINT "PRODUCTIONEMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "PRODUCTIONEMPLOYEE" ADD CONSTRAINT "PRODUCTIONEMPLOYEE_CON" FOREIGN KEY ("EMPLOYEEID")
          REFERENCES  "EMPLOYEE" ("EMPLOYEEID") ON DELETE CASCADE ENABLE
/
ALTER TABLE  "PRODUCTIONEMPLOYEE" ADD CONSTRAINT "PRODUCTIONEMPLOYEE_FK" FOREIGN KEY ("LINENUMBER")
          REFERENCES  "PRODUCTIONLINE" ("LINENUMBER") ENABLE
/
```

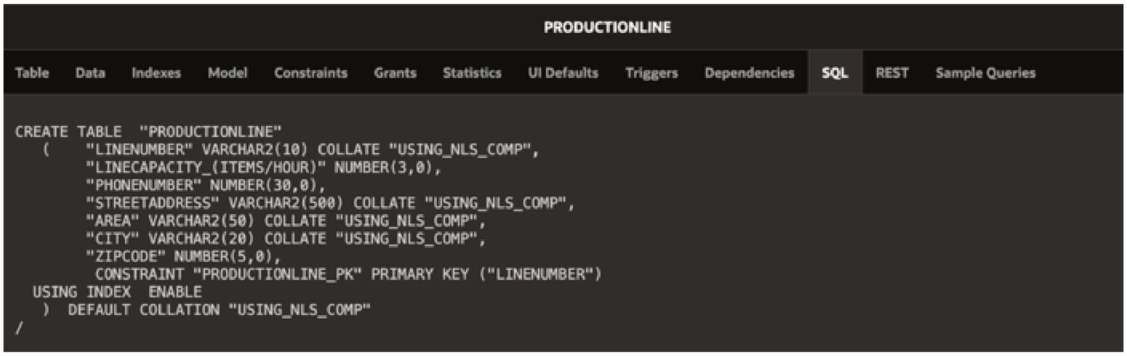
- PRODUCTIONLINE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| LINENUMBER | VARCHAR2(10) | No | - | 1 |
| LINECAPACITY_(ITEMS/HOUR) | NUMBER(3,0) | Yes | - | - |
| PHONENUMBER | NUMBER(30,0) | Yes | - | - |
| STREETADDRESS | VARCHAR2(500) | Yes | - | - |
| AREA | VARCHAR2(50) | Yes | - | - |
| CITY | VARCHAR2(20) | Yes | - | - |
| ZIPCODE | NUMBER(5,0) | Yes | - | - |

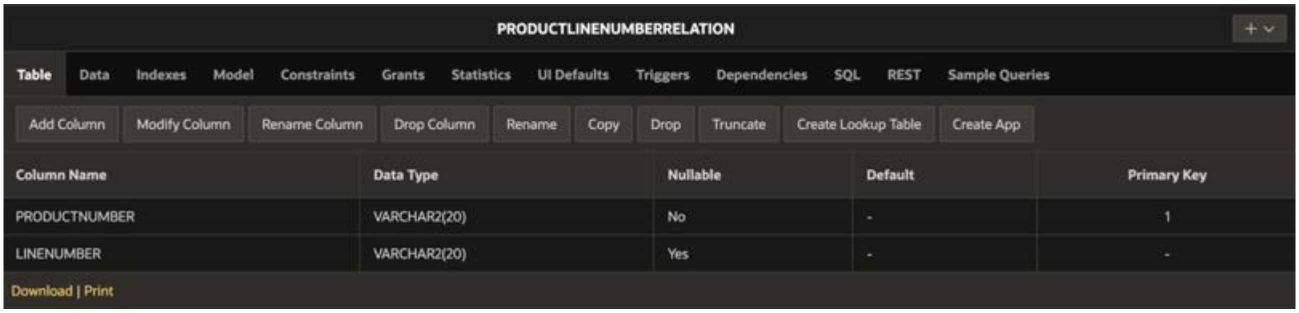
SQL:



```
CREATE TABLE  "PRODUCTIONLINE"
   (    "LINENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
        "LINECAPACITY_(ITEMS/HOUR)" NUMBER(3,0),
        "PHONENUMBER" NUMBER(30,0),
        "STREETADDRESS" VARCHAR2(500) COLLATE "USING_NLS_COMP",
        "AREA" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "CITY" VARCHAR2(20) COLLATE "USING_NLS_COMP",
        "ZIPCODE" NUMBER(5,0),
         CONSTRAINT "PRODUCTIONLINE_PK" PRIMARY KEY ("LINENUMBER")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
```

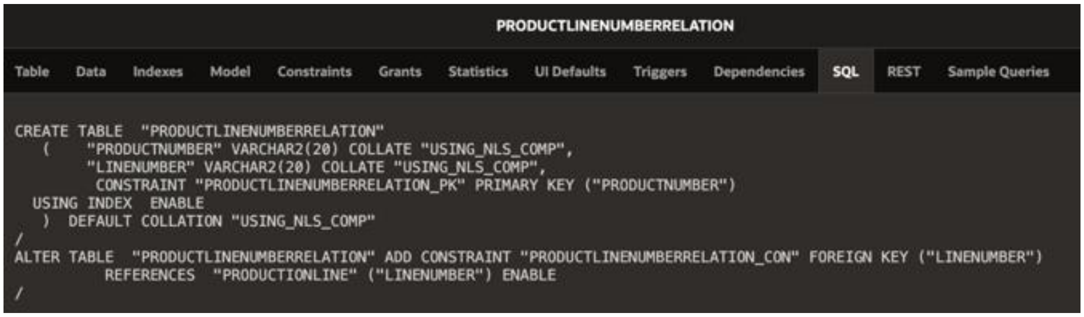
- PRODUCTLINENUMBERRELATION:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| PRODUCTNUMBER | VARCHAR2(20) | No | - | 1 |
| LINENUMBER | VARCHAR2(20) | Yes | - | - |

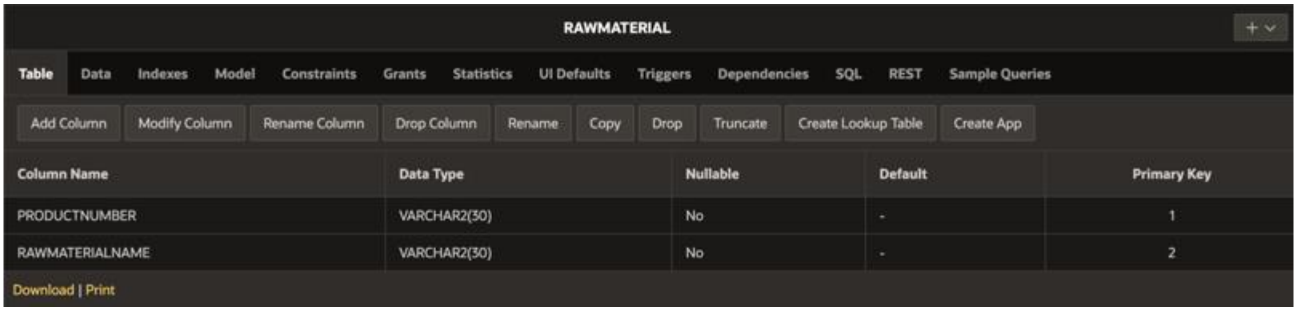
SQL:



```
CREATE TABLE  "PRODUCTLINENUMBERRELATION"
   (    "PRODUCTNUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP",
        "LINENUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP",
         CONSTRAINT "PRODUCTLINENUMBERRELATION_PK" PRIMARY KEY ("PRODUCTNUMBER")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "PRODUCTLINENUMBERRELATION" ADD CONSTRAINT "PRODUCTLINENUMBERRELATION_CON" FOREIGN KEY ("LINENUMBER")
          REFERENCES  "PRODUCTIONLINE" ("LINENUMBER") ENABLE
/
```

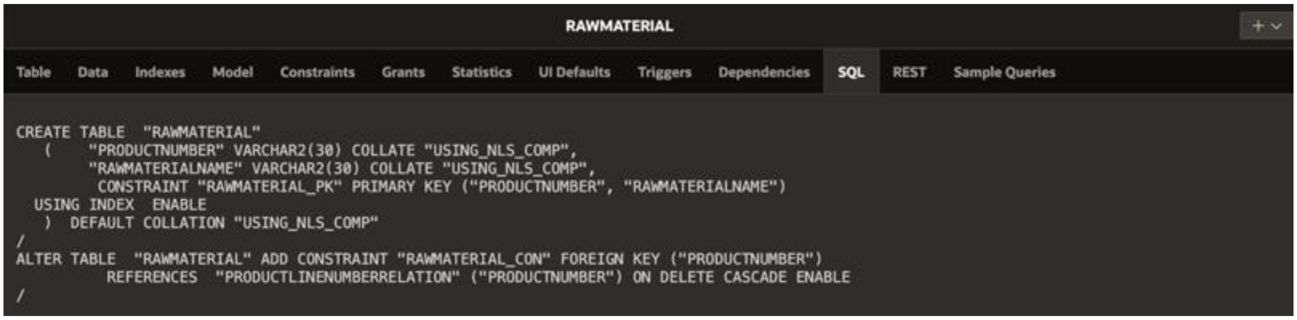
- RAWMATERIAL:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| PRODUCTNUMBER | VARCHAR2(30) | No | - | 1 |
| RAWMATERIALNAME | VARCHAR2(30) | No | - | 2 |

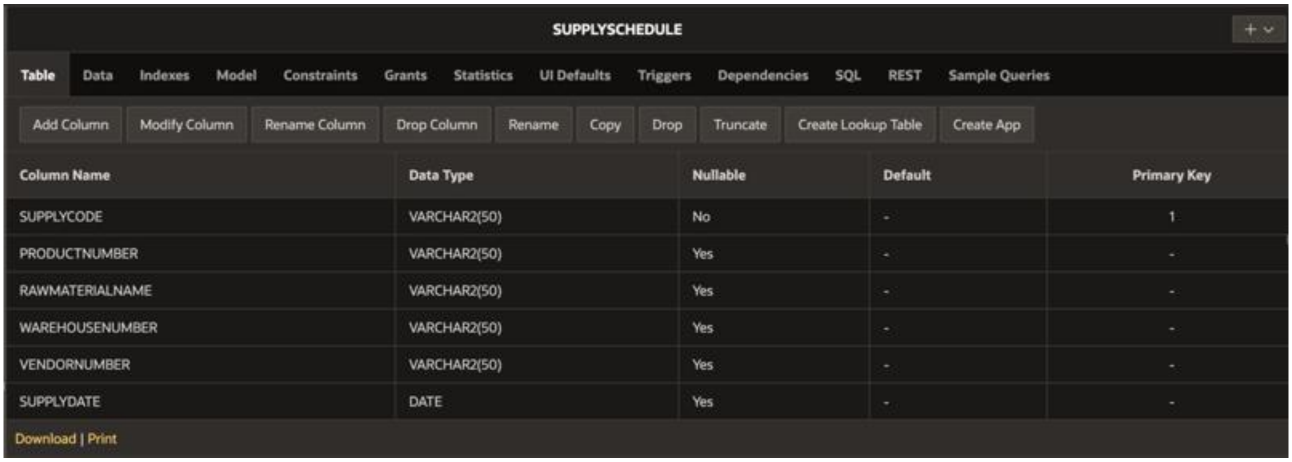
SQL:



```
CREATE TABLE  "RAWMATERIAL"
   (    "PRODUCTNUMBER" VARCHAR2(30) COLLATE "USING_NLS_COMP",
        "RAWMATERIALNAME" VARCHAR2(30) COLLATE "USING_NLS_COMP",
         CONSTRAINT "RAWMATERIAL_PK" PRIMARY KEY ("PRODUCTNUMBER", "RAWMATERIALNAME")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "RAWMATERIAL" ADD CONSTRAINT "RAWMATERIAL_CON" FOREIGN KEY ("PRODUCTNUMBER")
          REFERENCES  "PRODUCTLINENUMBERRELATION" ("PRODUCTNUMBER") ON DELETE CASCADE ENABLE
/
```

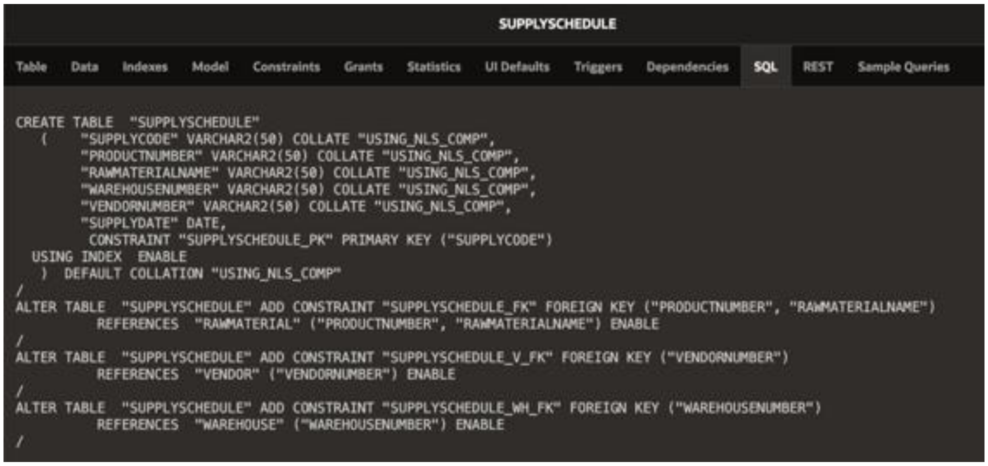
- SUPPLYSCHEDULE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| SUPPLYCODE | VARCHAR2(50) | No | - | 1 |
| PRODUCTNUMBER | VARCHAR2(50) | Yes | - | - |
| RAWMATERIALNAME | VARCHAR2(50) | Yes | - | - |
| WAREHOUSENUMBER | VARCHAR2(50) | Yes | - | - |
| VENDORNUMBER | VARCHAR2(50) | Yes | - | - |
| SUPPLYDATE | DATE | Yes | - | - |

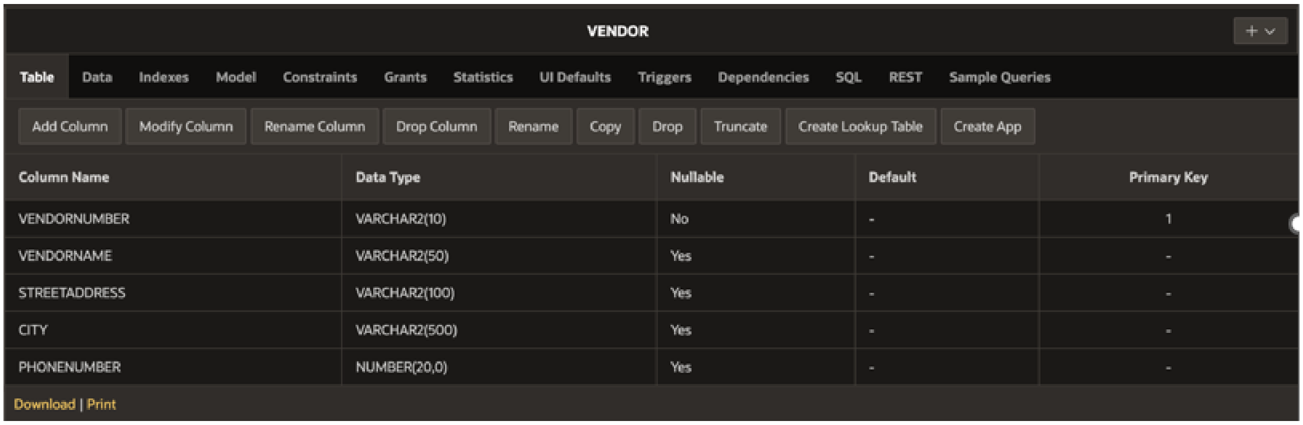
SQL:



```
CREATE TABLE  "SUPPLYSCHEDULE"
   (    "SUPPLYCODE" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "PRODUCTNUMBER" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "RAWMATERIALNAME" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "WAREHOUSENUMBER" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "VENDORNUMBER" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "SUPPLYDATE" DATE,
         CONSTRAINT "SUPPLYSCHEDULE_PK" PRIMARY KEY ("SUPPLYCODE")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "SUPPLYSCHEDULE" ADD CONSTRAINT "SUPPLYSCHEDULE_FK" FOREIGN KEY ("PRODUCTNUMBER", "RAWMATERIALNAME")
          REFERENCES  "RAWMATERIAL" ("PRODUCTNUMBER", "RAWMATERIALNAME") ENABLE
/
ALTER TABLE  "SUPPLYSCHEDULE" ADD CONSTRAINT "SUPPLYSCHEDULE_V_FK" FOREIGN KEY ("VENDORNUMBER")
          REFERENCES  "VENDOR" ("VENDORNUMBER") ENABLE
/
ALTER TABLE  "SUPPLYSCHEDULE" ADD CONSTRAINT "SUPPLYSCHEDULE_WH_FK" FOREIGN KEY ("WAREHOUSENUMBER")
          REFERENCES  "WAREHOUSE" ("WAREHOUSENUMBER") ENABLE
/
```

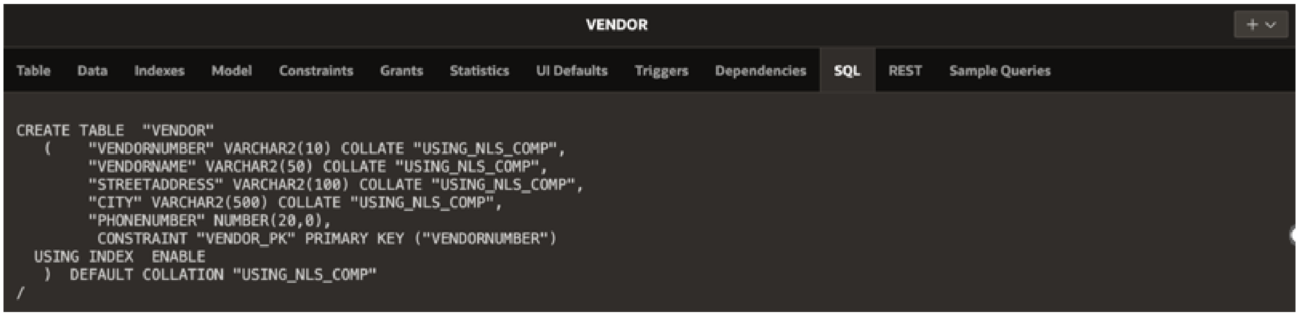
- VENDOR:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| VENDORNUMBER | VARCHAR2(10) | No | - | 1 |
| VENDORNAME | VARCHAR2(50) | Yes | - | - |
| STREETADDRESS | VARCHAR2(100) | Yes | - | - |
| CITY | VARCHAR2(500) | Yes | - | - |
| PHONENUMBER | NUMBER(20,0) | Yes | - | - |

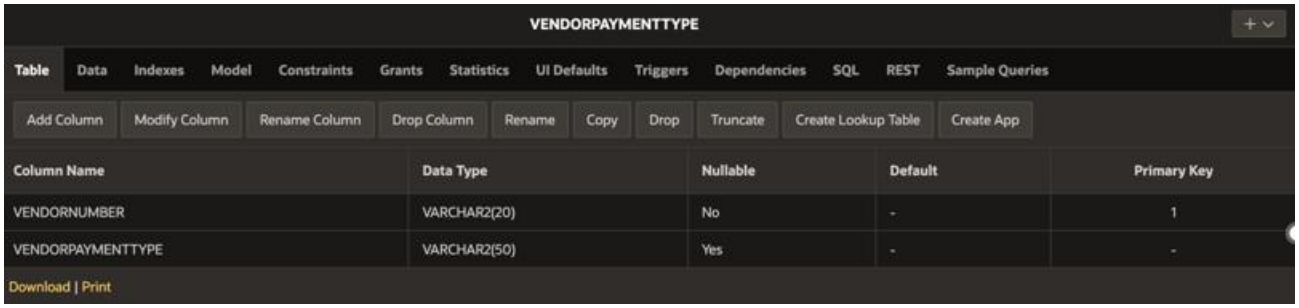
SQL:



```
CREATE TABLE  "VENDOR"
   (    "VENDORNUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
        "VENDORNAME" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "STREETADDRESS" VARCHAR2(100) COLLATE "USING_NLS_COMP",
        "CITY" VARCHAR2(500) COLLATE "USING_NLS_COMP",
        "PHONENUMBER" NUMBER(20,0),
         CONSTRAINT "VENDOR_PK" PRIMARY KEY ("VENDORNUMBER")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
```

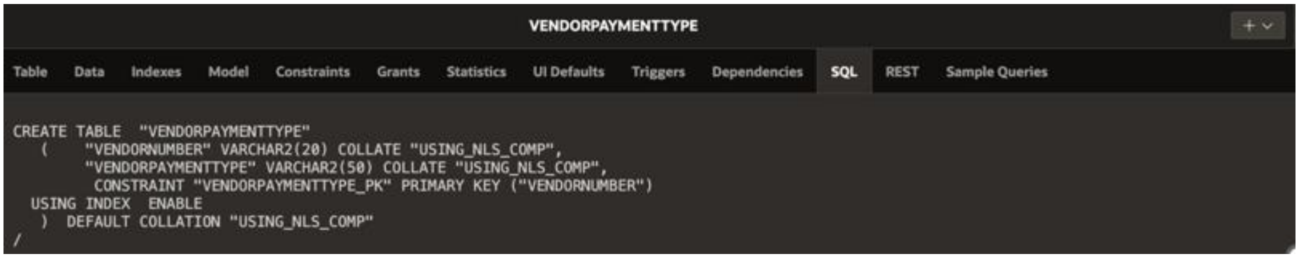
- VENDORPAYMENTTYPE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| VENDORNUMBER | VARCHAR2(20) | No | - | 1 |
| VENDORPAYMENTTYPE | VARCHAR2(50) | Yes | - | - |

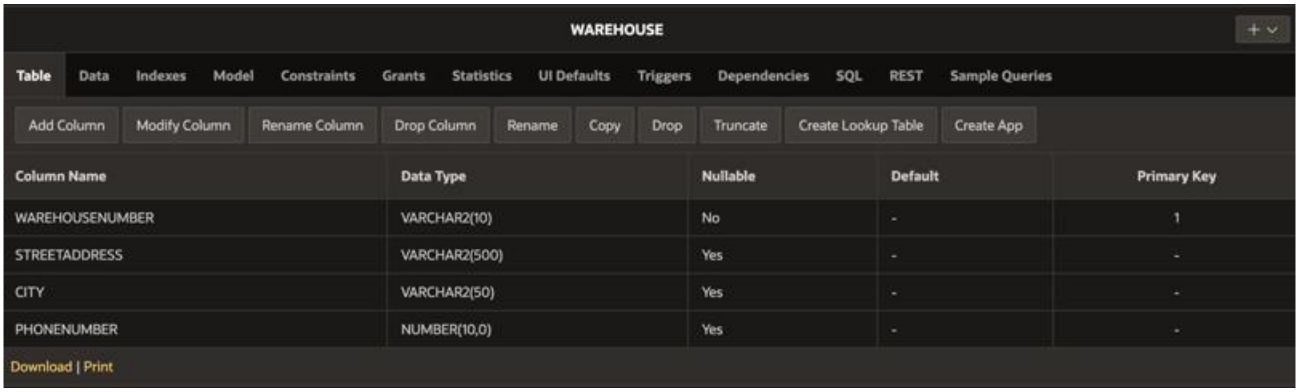
SQL:



```
CREATE TABLE  "VENDORPAYMENTTYPE"
   (    "VENDORNUMBER" VARCHAR2(20) COLLATE "USING_NLS_COMP",
        "VENDORPAYMENTTYPE" VARCHAR2(50) COLLATE "USING_NLS_COMP",
         CONSTRAINT "VENDORPAYMENTTYPE_PK" PRIMARY KEY ("VENDORNUMBER")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
```

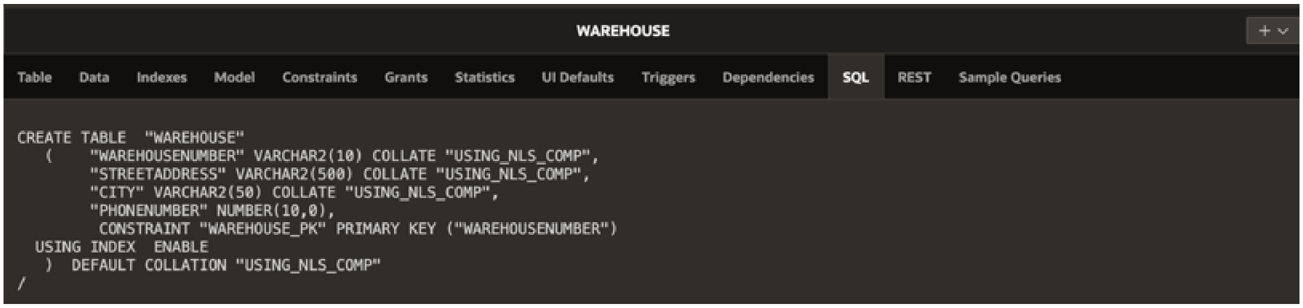
- WAREHOUSE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| WAREHOUSENUMBER | VARCHAR2(10) | No | - | 1 |
| STREETADDRESS | VARCHAR2(500) | Yes | - | - |
| CITY | VARCHAR2(50) | Yes | - | - |
| PHONENUMBER | NUMBER(10,0) | Yes | - | - |

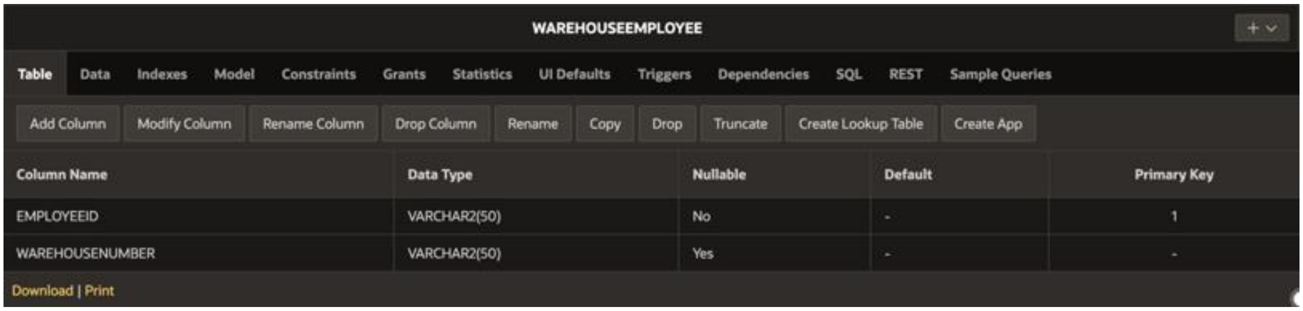
SQL:



```
CREATE TABLE  "WAREHOUSE"
   (    "WAREHOUSENUMBER" VARCHAR2(10) COLLATE "USING_NLS_COMP",
        "STREETADDRESS" VARCHAR2(500) COLLATE "USING_NLS_COMP",
        "CITY" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "PHONENUMBER" NUMBER(10,0),
         CONSTRAINT "WAREHOUSE_PK" PRIMARY KEY ("WAREHOUSENUMBER")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
```

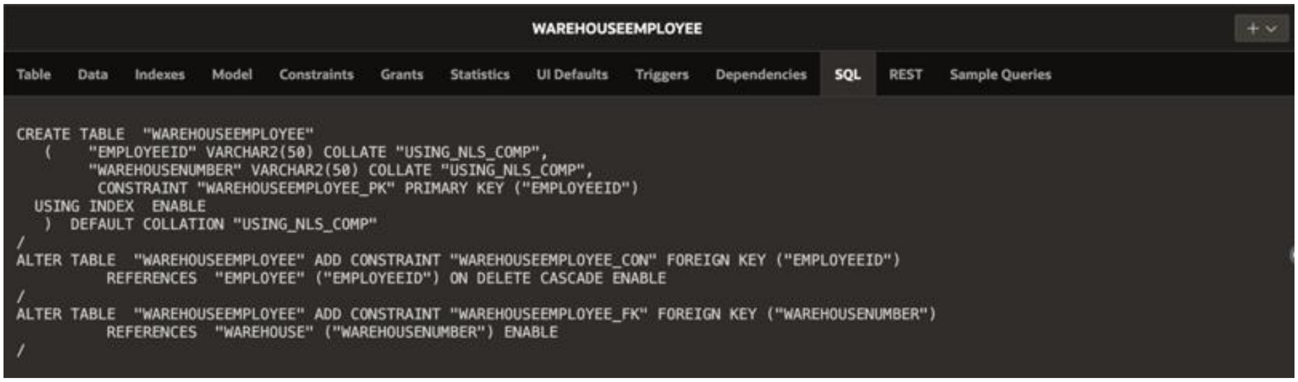
- ## WAREHOUSEEMPLOYEE:

TABLE:



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| EMPLOYEEID | VARCHAR2(50) | No | - | 1 |
| WAREHOUSENUMBER | VARCHAR2(50) | Yes | - | - |

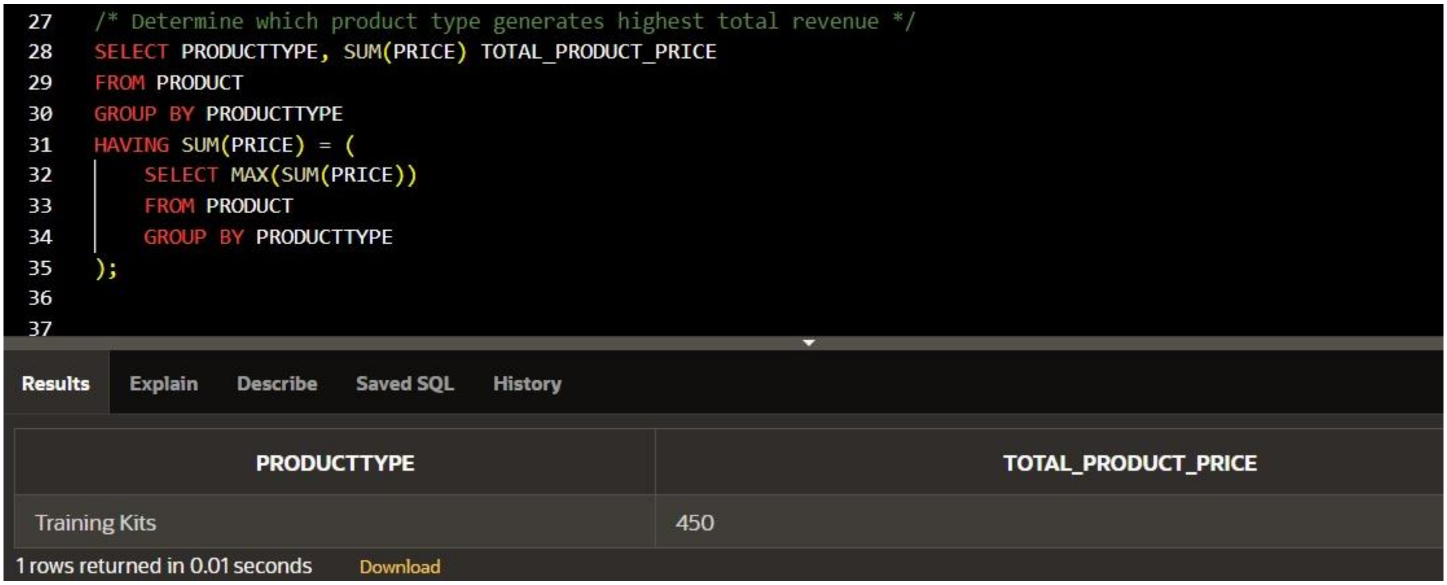
SQL:



```
CREATE TABLE  "WAREHOUSEEMPLOYEE"
   (    "EMPLOYEEID" VARCHAR2(50) COLLATE "USING_NLS_COMP",
        "WAREHOUSENUMBER" VARCHAR2(50) COLLATE "USING_NLS_COMP",
         CONSTRAINT "WAREHOUSEEMPLOYEE_PK" PRIMARY KEY ("EMPLOYEEID")
  USING INDEX  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP"
/
ALTER TABLE  "WAREHOUSEEMPLOYEE" ADD CONSTRAINT "WAREHOUSEEMPLOYEE_CON" FOREIGN KEY ("EMPLOYEEID")
          REFERENCES  "EMPLOYEE" ("EMPLOYEEID") ON DELETE CASCADE ENABLE
/
ALTER TABLE  "WAREHOUSEEMPLOYEE" ADD CONSTRAINT "WAREHOUSEEMPLOYEE_FK" FOREIGN KEY ("WAREHOUSENUMBER")
          REFERENCES  "WAREHOUSE" ("WAREHOUSENUMBER") ENABLE
/
```

# SQL QUERIES

1) Determine which product type generates highest total revenue:

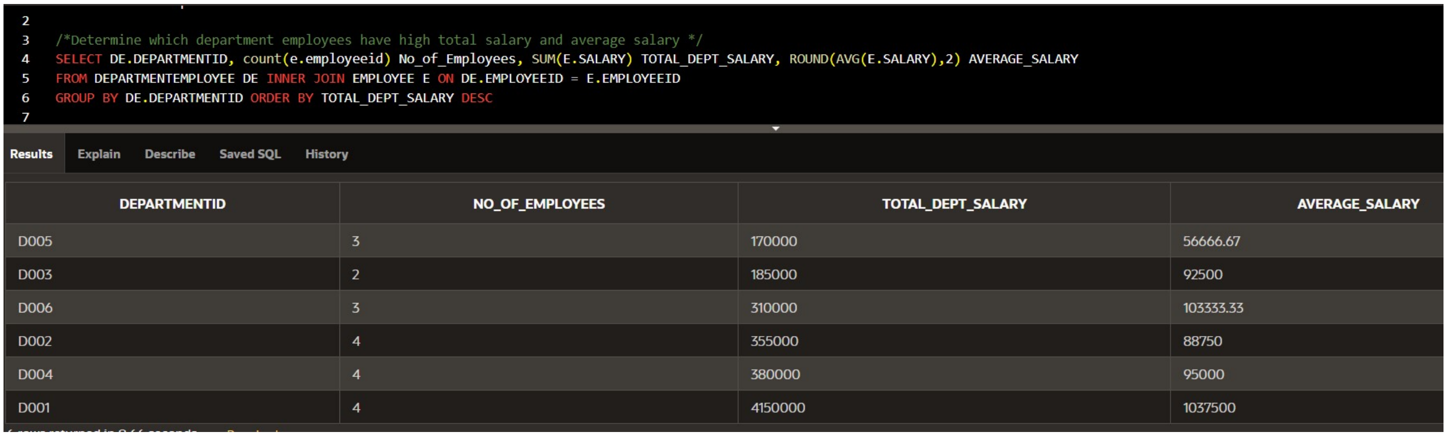


```
/* Determine which product type generates highest total revenue */
SELECT PRODUCTTYPE, SUM(PRICE) TOTAL_PRODUCT_PRICE
FROM PRODUCT
GROUP BY PRODUCTTYPE
HAVING SUM(PRICE) = (
    SELECT MAX(SUM(PRICE))
    FROM PRODUCT
    GROUP BY PRODUCTTYPE
);
```

| PRODUCTTYPE | TOTAL_PRODUCT_PRICE |
|---|---|
| Training Kits | 450 |

1 rows returned in 0.01 seconds

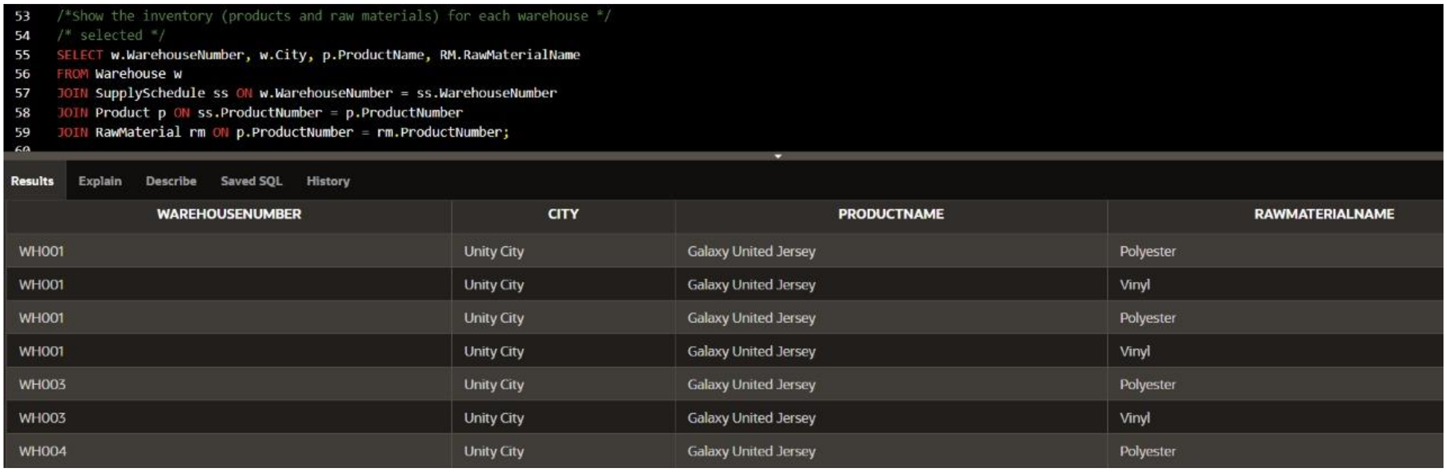
2) Determine which department employees have highest total salary and average salary:



```
/*Determine which department employees have high total salary and average salary */
SELECT DE.DEPARTMENTID, count(e.employeeid) No_of_Employees, SUM(E.SALARY) TOTAL_DEPT_SALARY, ROUND(AVG(E.SALARY),2) AVERAGE_SALARY
FROM DEPARTMENTEMPLOYEE DE INNER JOIN EMPLOYEE E ON DE.EMPLOYEEID = E.EMPLOYEEID
GROUP BY DE.DEPARTMENTID ORDER BY TOTAL_DEPT_SALARY DESC
```

| DEPARTMENTID | NO_OF_EMPLOYEES | TOTAL_DEPT_SALARY | AVERAGE_SALARY |
|---|---|---|---|
| D005 | 3 | 170000 | 56666.67 |
| D003 | 2 | 185000 | 92500 |
| D006 | 3 | 310000 | 103333.33 |
| D002 | 4 | 355000 | 88750 |
| D004 | 4 | 380000 | 95000 |
| D001 | 4 | 4150000 | 1037500 |

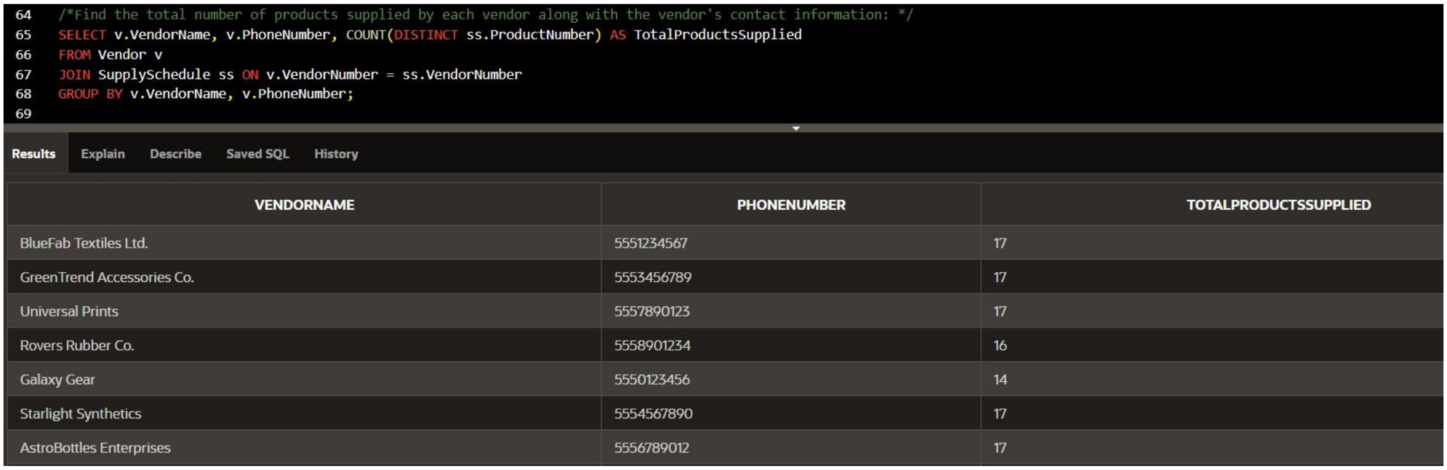
3) Display inventory of products and raw materials in each warehouse:



```sql
/*Show the inventory (products and raw materials) for each warehouse */
/* selected */
SELECT w.WarehouseNumber, w.City, p.ProductName, RM.RawMaterialName
FROM Warehouse w
JOIN SupplySchedule ss ON w.WarehouseNumber = ss.WarehouseNumber
JOIN Product p ON ss.ProductNumber = p.ProductNumber
JOIN RawMaterial rm ON p.ProductNumber = rm.ProductNumber;
```

| WAREHOUSENUMBER | CITY | PRODUCTNAME | RAWMATERIALNAME |
|---|---|---|---|
| WH001 | Unity City | Galaxy United Jersey | Polyester |
| WH001 | Unity City | Galaxy United Jersey | Vinyl |
| WH001 | Unity City | Galaxy United Jersey | Polyester |
| WH001 | Unity City | Galaxy United Jersey | Vinyl |
| WH003 | Unity City | Galaxy United Jersey | Polyester |
| WH003 | Unity City | Galaxy United Jersey | Vinyl |
| WH004 | Unity City | Galaxy United Jersey | Polyester |

4) Find the total number of products supplied by each vendor along with the vendor's contact information.



```sql
/*Find the total number of products supplied by each vendor along with the vendor's contact information: */
SELECT v.VendorName, v.PhoneNumber, COUNT(DISTINCT ss.ProductNumber) AS TotalProductsSupplied
FROM Vendor v
JOIN SupplySchedule ss ON v.VendorNumber = ss.VendorNumber
GROUP BY v.VendorName, v.PhoneNumber;
```

| VENDORNAME | PHONENUMBER | TOTALPRODUCTSSUPPLIED |
|---|---|---|
| BlueFab Textiles Ltd. | 5551234567 | 17 |
| GreenTrend Accessories Co. | 5553456789 | 17 |
| Universal Prints | 5557890123 | 17 |
| Rovers Rubber Co. | 5558901234 | 16 |
| Galaxy Gear | 5550123456 | 14 |
| Starlight Synthetics | 5554567890 | 17 |
| AstroBottles Enterprises | 5556789012 | 17 |