

# **Pine Valley Furniture Data Ware House**



**IT 651  
Spring 2018**

# **Table of Contents**

<b>I Introduction</b>	<b>2</b>
<b>II Background</b>	<b>3</b>
<b>III Requirements</b>	<b>6</b>
<b>IV Data Modeling</b>	<b>13</b>
<b>V Extraction, Transformation, and Loading By Informatica and Basic SQL</b>	<b>15</b>
<b>VI Data Query</b>	<b>19</b>
<b>VII Conclusions</b>	<b>23</b>
<b>Appendix A</b>	<b>24</b>

## I. Introduction

### Why Data Warehouse?

Pine Valley Furniture Company is a small manufacturer of goods furniture but its sale has been increasing by around 25% per year which is a very good growing rate.

With using the Online Transaction Processing (OLTP) system which can keep the information of customers list, orders, products, invoices, bills of materials, routing, work centers, and progress report. They realized that data warehouse would be able to higher the chance to keep the growing percentage or even increase it.

A data warehouse, initially focused on sales, will allow the company executives to have the following advantages:

- The OLTP System has now a considerable large database, the data warehouse will provide faster access to the information about sales, customers and products. Additionally, this process to develop data warehouse will increase data quality.
- The organized and optimized information will increase the flexibility of the entire company's analysis to facilitate business decisions, especially it will do a better marketing campaign so eventually to keep customer satisfaction which is the key point for the company success.
- It can share information among different departments to help improve processes and quality. Allowing analysis of the company information will provide insights into the processes and it might yield to ideas for re-engineering those processes.
- Empower users to quickly and easily create professional reports, charts, graphs, summarizations, statistics, comparisons and forecasts without depending on information technology staff.
- Provides access to data for applications without impacting the OLTP System.
- Ability to use descriptive names rather than only cryptic names in OLTP database.
- It set up a foundation for future applications

After evaluating the OLTP System, it had an excellent results. So it is an important factor for the growth. Then they decided to start the data warehouse project considering the potential additional advantages they could have with it.

The data warehouse will start with one subject: Sales and eventually grow to contain some other important subjects for the company. Sales subject was prioritized in terms of desired positive impact and contribution to the company goals and vision.

## II. Background

### Profiles

#### User, data and process

The majority of authorized users will only be all the managers and marketing executives of Pine Valley Furniture Company. The person who has responsibility for creating sales reports via data warehouse, however, will be the marketing executives. For each product lines, they will access this sales data warehouse consistently and access current information on daily, weekly, monthly, quarterly, and yearly periods on an as needed basis.

#### Decision, components, and tools

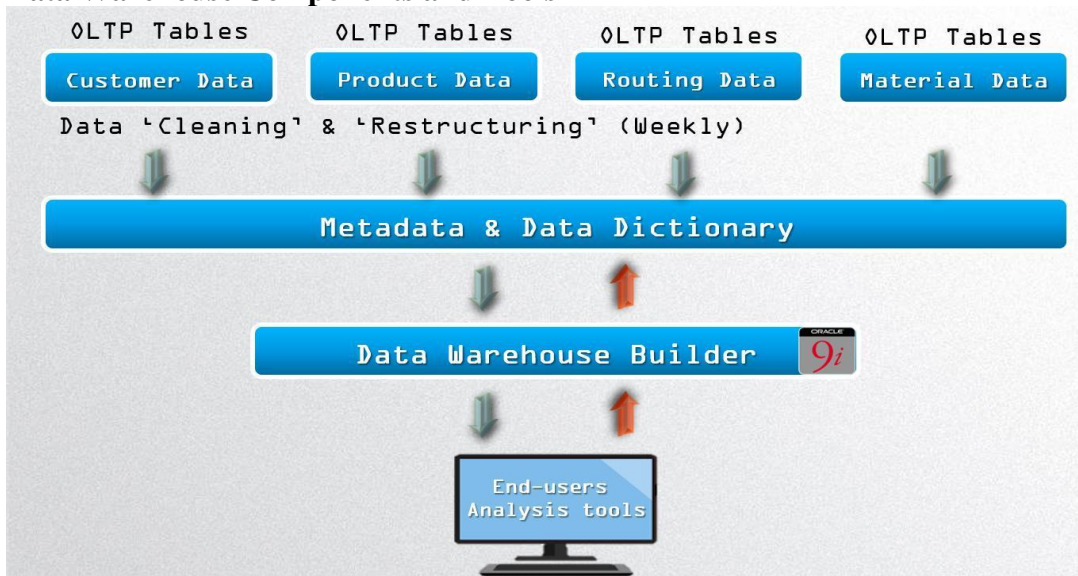
Previously, building a business intelligence system required the integration of multiple server products. In addition, to the database, completely, separate engines for extraction, transformation and loading, on-line analytical processing (OLAP). This meant that several different engines had to be learned and managed, and multiple copies of the metadata and data needed to be synchronized. The new way to meet all business intelligence applications is to use Oracle 9i platform. The core benefits of the relational database-scalability, reliability and security- are now extended to the entire business intelligence platform.

For this project the following components have been chosen:

Data scrubbing and cleansing utilities, using:

- Language: SQL
- Software: Oracle 9i Warehouse Builder
- Database: Oracle 8i

#### Data Warehouse Components and Tools



## Application

During the implementation of project, we chose Oracle 9i Warehouse Builder as the tool for developing data warehouse project. Oracle warehouse Builder (OWB) is a tool for designing, deploying and managing enterprise data warehouse, data marts and e-business applications.

OWB consists of the following primary functional components:

- Graphical User Interface

The OWB user interface runs on a windows workstation. It provides wizards and graphical editors for designing relational and multidimensional schemas, and for designing dataflow diagrams that define the ETL (extraction, transformation and loading) process.

- Repository

The OWB repository is a set of tables in an Oracle database that the OWB graphical user interface accesses via a Java-based access layer. Multiple users are supported at the same time, and all the users' work-source definitions, data warehouse design and source-to-target mappings-is stored in this repository. Public views make the OWB repository metadata accessible to third-party applications.

- Code Generator

Code generator code to implement the data warehouse; extract, transform and load data from various sources into the target data warehouse; and manage recurring tasks.

- Integrators

Integrators are components that are dedicated to extracting metadata and data from a particular type of source.

- Bridges

The OWB bridges facilitate the exchange of metadata with related oracle products, third-party CASE tools, or and OMG CWM compliant meta source.

- Brower

The OWB Browser enables Web-Based browsing of the metadata stored in the OWB repository. In addition to standardized report it also provides graphical impact analysis and data lineage.

- Runtime

The OWB Runtime is a set of tables, sequences, packages and triggers that are installed in the target database. These database objects are the foundation for OWB's auditing and error detection/correction capabilities. The audit data store in the Runtime tables can be viewed with the OWB Runtime Audit Viewer.

### III. Requirements

#### Business Modeling

The Pine Valley Furniture Company business model, the method of doing business and generate revenue is marketing its furniture to a wide variety of retail furniture stores, including some large chains.

A meaningful business model is critical to the success of this project. Even the most elegant technology will not satisfy business needs if it doesn't take user's needs and environments into account. We collaborated efficiently with the senior and line managers, clerks and IT staff. Together, all parties come to understand the short and long term business goals, end-user requirements, and available data resources. That way, we do not end up with a business model that is academically pleasing but has no buy-in from users or management.

The business model will become the bedrock on which all design and development of the data warehouse rests. Every move will be tested against the model: Does this feature meet user needs? Will the architecture support management's long-term goals? Did we include the right level of detail to support require business activities?

Moreover, the effort we spend on the business model and the resultant architecture is well worth it. It will be possible to add fields, data sources, a web application, and other features to this solution -- without significant delay or cost -- all thanks to the quality of the business model.

Company goals:

- Achieve Sales Growth of 25% per year for the next 5 years
- Create a marketing presence as a leader in case goods furniture.
- Reach 100 new customers per year for next 5 years

Company vision:

Pine Valley furniture company aspires to be the best manufacturer of goods Furniture company in the U.S. Northeast region, leader in style, value, and service.

Our strategy has been to focus on what our customers need and give the customer what they want, when they want it. Our primary efforts focus on value with immediate delivery of our products.

Innovative marketing concepts, quality products and state of the art manufacturing methods have always been driven by the demand for customer satisfaction. At Pine Valley we want the stream of satisfied customers continue to grow as we continue to be creative and diversifies to expand business and improve quality into the next millennium.

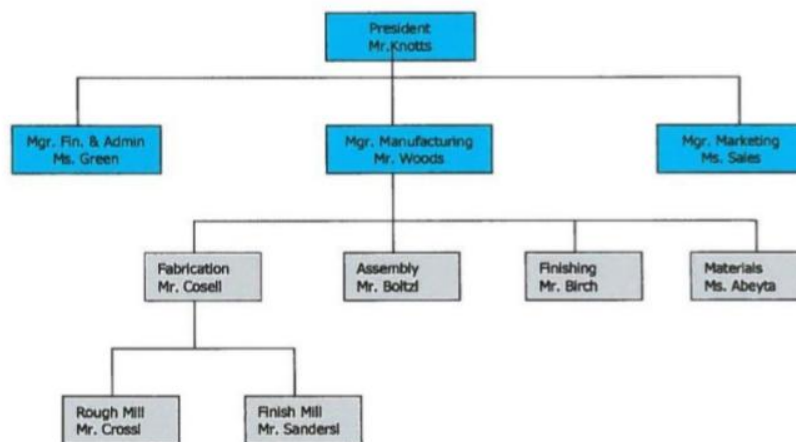
### Company Mission Statement:

Our objective is to ensure financial and moral success for the whole staff long term by manufacturing and providing the customers with a consistently excellent case goods furniture at a moderate cost.

### Critical success factors

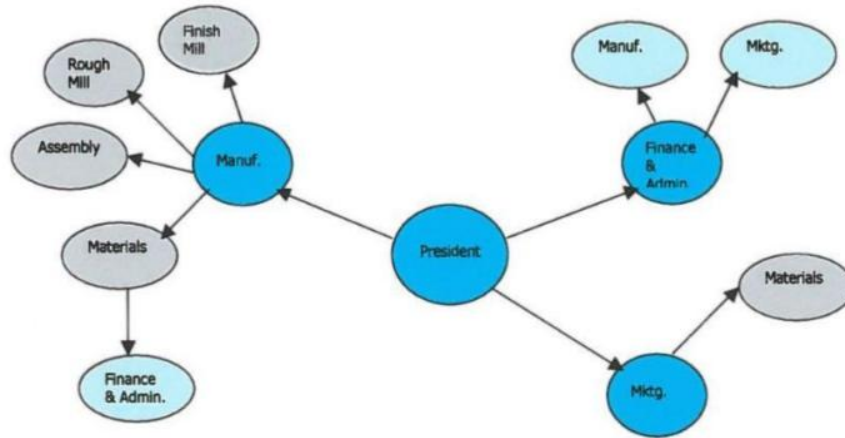
- First phase is small enough to be implemented quickly (3-4 months time frame)
- Focus in on delivering the majority of sales information during the first phase
- Data model is derived from existing models in Orders system (OLTP System) and changed to meet new requirement
- Business rules are predefined for data translation - documentation is available
- Reports are defined, specification created, and documentation kept as metadata
- End user resources are made easily available
- Customer buying repeats
- Be recognized as a leading case goods furniture company.

### Modeling the organization

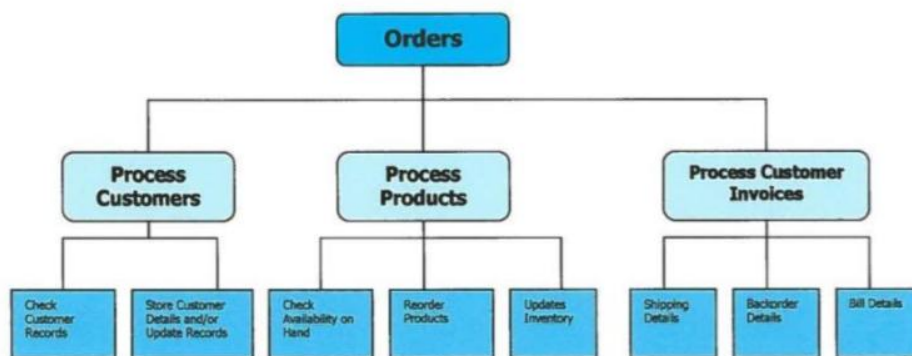




## Logical Decision Making Chart



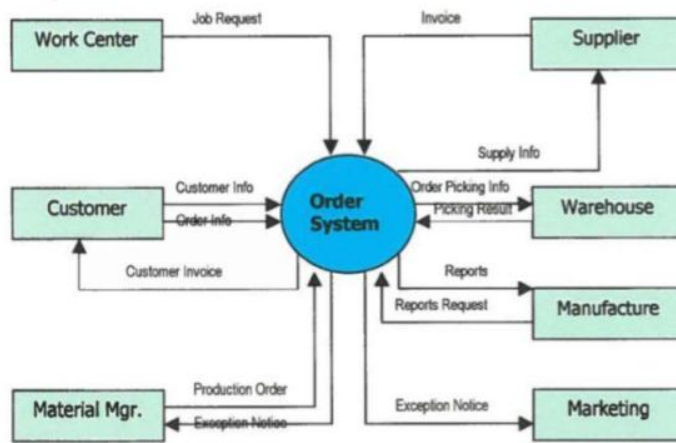
## Modeling the business Process



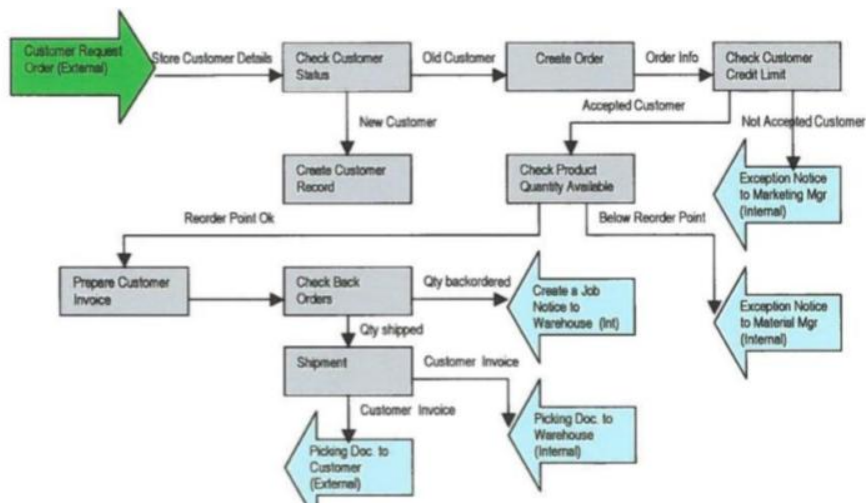




## Data Flow Chart - Current OLTP System



## Process Flow Diagram



## **Initial Requirements**

### **Project Scope**

Organize core sales information ensuring the focus will be on the business functions of Customer locations, product preferences, credit, and sales amount information.

Information provided in the first phase of the project will form a foundation for inconsistent, accurate and current sales-related information that will enable more efficient marketing analysis and decision making across the company.

Implementing an enterprise-wide sales data warehouse will allow:

- User-customizable database querying and reporting via secure Data Warehouse Analysis tools.
- Preserve and enhance standard query and reporting functions for users who require them.

The implementation will be an ongoing process that may span multiple years but delivers business value in 3-month cycles in parallel with the progressive, staged implementation of the overall data warehouse architecture. This value is in the form of report generation, cube analytics and ad hoc reporting. It will enable all levels of users (executive, managerial and operational) access to consistent and current information on daily, weekly, monthly, quarterly, and yearly periods on an as needed basis.

### **Deliverable descriptions phase 1 (success criteria)**

- Impromptu Reports(IR)- Standardized reports in pdf, excel, and text formats
  - Top (#) Reports about Customer Locations
  - Top (#) Reports about Customer Product preferences
  - Top (#) Reports about Customer Buying Periods
  - Top (#) Reports using Product data
  - Top (#) Reports using Sales data
- Simple Ad-hoc queries.

### **Deliverable descriptions phase 1**

- Ad-hoc queries for advanced users.
- Cubes for multi-dimensional analysis.
- Web access to data warehouse.

### **General descriptions phase 3**

- Incorporate new subjects to the data warehouse
- Incorporate new analysis tools for the users

### **Data Requirements**

The primary source of information will be from the order system. The initial amount of historical data will include calendar and fiscal year end snapshots from 1999-forward and month end snapshots from 1999-forward. Also, the year to date (2002-forward) information will be updated each week with any new changes to reflect current operations.

### **Trainings**

- Introduction
  - Confidentiality Agreement
  - Handout Materials (Available resources, business rules, handbook)
- Understanding the sales Data Warehouse
  - Requirements defined
  - Understanding defined
  - Sales Metadata
  - Tools Environment
- Security
  - Basics
  - How to access the data warehouse
- Impromptu Reports
  - Using its functionality
  - Customizing Reports
  - Creating new Reports
- Questions and Answers

### **Security**

Adhere to established security rules and definitions regarding access to protected sales data.

### **Ongoing Support Plan**

The process of growing and extending a data warehouse is never ending, just as organizations never stop growing and changing. Thus, the following aspects need to be considered:

- Information systems trainers

- Help desks
- Software upgrades
- Additional training - Advanced or new functionality

### **Analyzing Data Sources**

According to the available information from the OLTP system, it is possible to have in the data warehouse summarized information:

- To get total sales by customer, period of time (day, week, month, or year), type of product, or even particular products, etc.
- To know how many products sold by customer, period of time, and type of product.
- To discover patterns about customers: what kind of products they buy more, where are they located, what type of customers they are, etc.
- To discover patterns about product trends: what kind of products customers buy more, where are they sold, how long it takes to produce them and to deliver them to the customers, etc.
- To drill down from summary data sets into specific data underlying the summaries. Statistical analysis tools provide summary information too and help determine the degree of relationship between two factors, such as locations and sales.

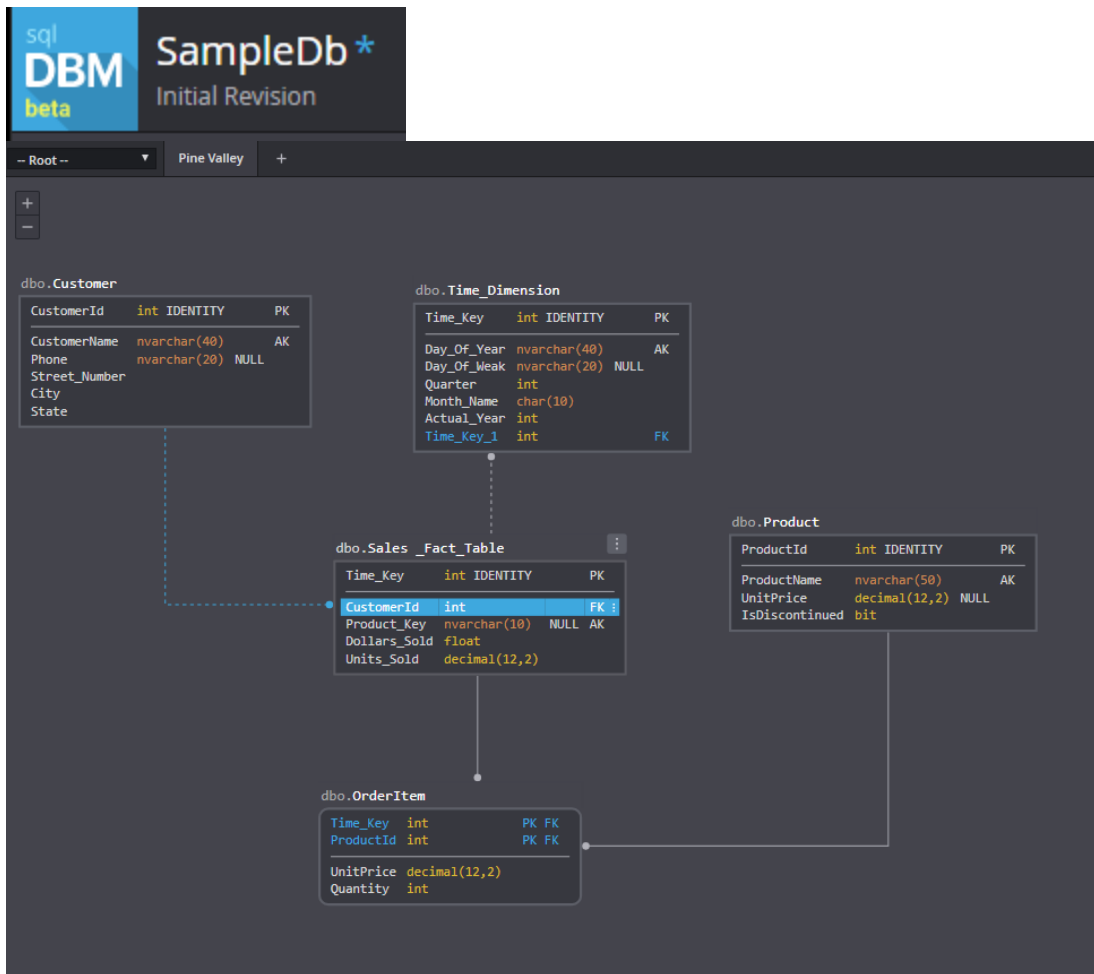
## **IV. Data Modeling**

The logical model defines the data entities and the relationships between those entities required to solve the business entities: We chose the following entities:

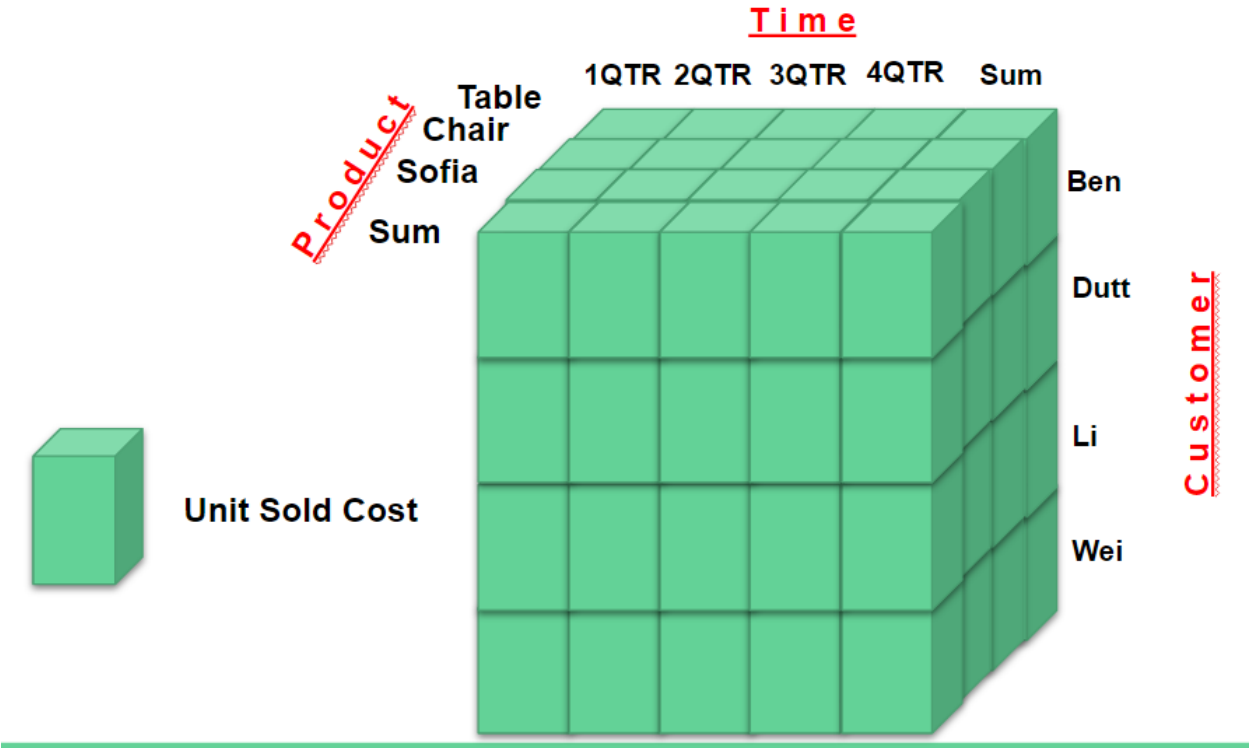
- Customers
- Products
- Orders
- Time

We will use a granularity that will cover each order for a specific customer.

The following is a representation of our Schema:



The multidimensional data model is an integral part of On-Line Analytical Processing, or OLAP. Because OLAP is on-line, it must provide answers quickly; analysts pose iterative queries during interactive sessions, not in batch jobs that run overnight. And because OLAP is also analytic, the queries are complex. We will represent our multidimensional data with the following model:



## V. Extraction, Transformation, and Loading Populating the Data Warehouse

There are several alternative means for moving data from one Oracle database into the warehouse, such as: the export/import of entire tables, using triggers, the unload/reload utility, using CREATE TABLE...AS SELECT, the SQL\*Plus COPY command, etc.

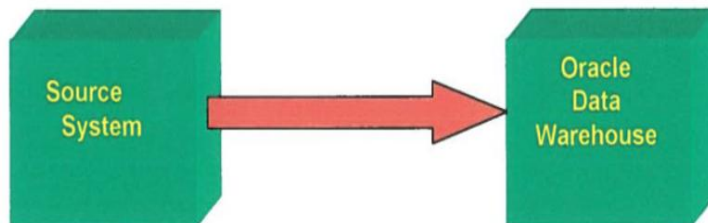
Each of these alternatives has merits and drawbacks that make it useful in different situations. In some cases the best solution may be a combination of these techniques.

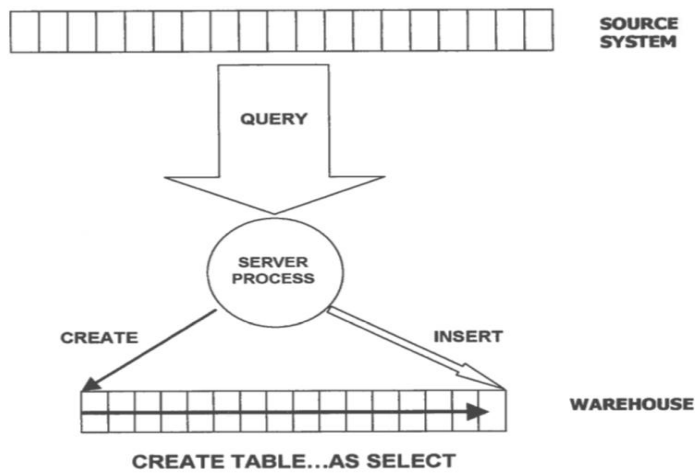
For populating our Oracle Data Warehouse we developed a series of scripts where we used the Power Center Informatica which is a high end operation at this level. Due to the licence constraints, we performed SQL CREATE TABLE command with a sub-query to retrieve data from an existing table (or join of multiple tables).

When sufficient CPU and I/O resources are available, the CREATE TABLE...AS SELECT can be performed much more quickly using the query capabilities of Oracle. The CREATE TABLE...AS SELECT operation has turned into one of the most valuable features introduced to benefit data warehousing.

The scripts create tables in our data warehouse that will be populated with the records from the source system.

The key here is that the sub-query executes entirely in the remote database and performs the necessary joins to find the data needed for the data warehouse.

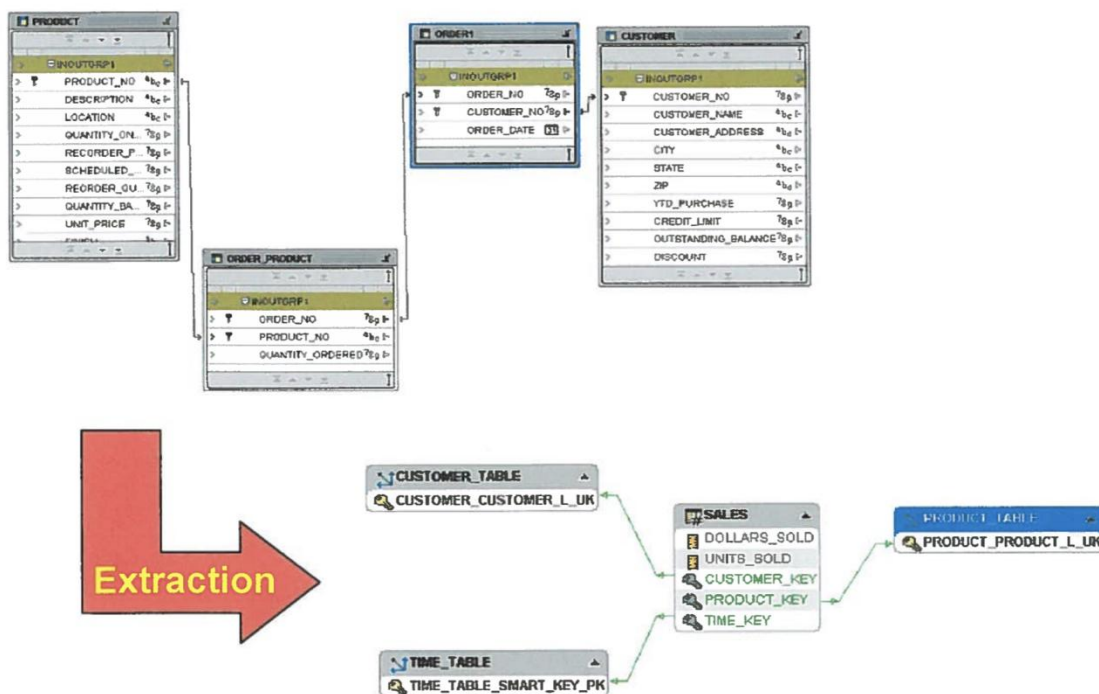




We do not use all the tables for populating the warehouse; we only extract data from the following tables:

- PRODUCT,
- ORDER— PRODUCT, ORDER
- CUSTOMER

Here is a layout of the extraction process:





## Scripts for Extraction

We did a generic extraction (or complete file copies) from the application systems. We then pick and choose which specific data fields to use and how to manipulate them during the load process. If we later determined that some additional source data is needed, we just change the load process to include the new data. The extract process is unaffected.

The advantage in this approach is that the warehouse team is the one who takes the responsibility of accommodating any changes in file structures introduced by application changes.

Using SQL combined with PL/SQL provides easy means of manipulating raw data once it has been loaded into the warehouse. When the raw data can be processed using SQL's set processing features, joins and Oracle's parallel capabilities, the work can usually be done very efficiently.

### *Script for Creating the product\_table dimension*

```
CREATE TABLE "PRODUCT_TABLE"
  ("PRODUCT_L_DESCRIPTION",
   "PRODUCT_L_PRODUCT_KEY" ,
   "PRODUCT_L_UNIT_PRICE")
  TABLESPACE "USERS"
AS
SELECT description, product_no, unit_price
FROM product;
```

In the subquery we load the data from the product table, using the attributes description, product—no, and unit—price.

In the subquery we load the data from the customer table, using the attributes customer—no, customer—name, customer address, city and state.

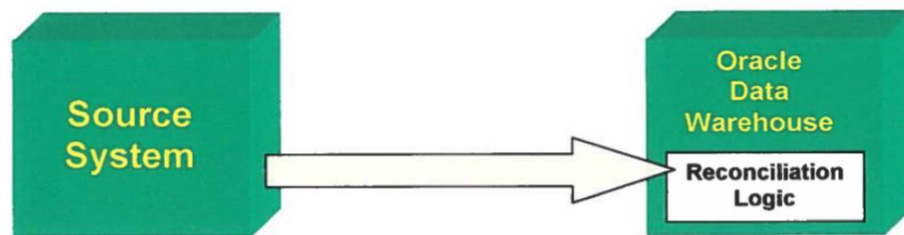
In the subquery we load the data from the order table, using the attribute order\_date.

Here we are processing the source data in order to use it in the time dimension; we are extracting from the order\_date attribute all the needed information to populate dimension (day, quarter, month, year attributes).

### Reconciling *Time\_table* dimension

```
UPDATE TIME_TABLE SET QUARTER = 1 WHERE QUARTER<4;
UPDATE TIME_TABLE SET QUARTER = 2 WHERE QUARTER>3 AND QUARTER<7;
UPDATE TIME_TABLE SET QUARTER = 3 WHERE QUARTER>6 AND QUARTER<10;
UPDATE TIME_TABLE SET QUARTER = 4 WHERE QUARTER>9;
```

After the data is loaded, we reconcile the quarter attribute.



According with the month, we update the quarter attribute with the corresponding quarter. If

### Script for Creating the *sales* fact table

```
CREATE TABLE "SALES"
  ("CUSTOMER_L_CUSTOMER_KEY" ,
   "TIME_KEY" ,
   "PRODUCT_L_PRODUCT_KEY" ,
   "DOLLARS_SOLD" ,
   "UNITS_SOLD" )
  TABLESPACE "USERS"
AS
SELECT order.customer_no,
       order.order_date,
       order_product.product_no,
       (order_product.quantity_ordered * product.unit_price),
       order_product.quantity_ordered
  FROM order, order_product, product
 WHERE order.order_no = order_product.order_no AND
       order_product.product_no = product.product_no;
```

the month is 4, then update to make it 1.

In the sub-query we load the data from the order, order^product, and product tables, joining the tables to get the desired result. With the following operations we obtain measures for ofo//ars—so/d and ii/i/fs—so/cf.

dollars\_sold = *order\_product.quantity\_ordered \* product.unit\_price*

units\_sold = *order\_product.quantity\_ordered*

## VI. Data Query

To this point, we have stated how we built the data warehouse. But once we have it built, what are some of the things we can do with all this data information.

The most common reporting tool is the SQL language itself. Traditional SQL statements can fulfill most of the reporting needs of the enterprise.

A SQL statement can retrieve the detailed results, and then a reporting tool would total the results separately.

### Index Summary:

Index Summary							
Index Name	Index Type	Uniqueness	Table Owner	Table Name	Columns	Size	Tablespace
PRODUCT_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">PRODUCT</a>	1	64K	USERS
PRODUCEDIN_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">PRODUCEDIN</a>	2	0	USERS
SALESPERSON_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">SALESPERSON</a>	1	64K	USERS
WORKCENTER_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">WORKCENTER</a>	1	64K	USERS
CUSTOMER_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">CUSTOMER</a>	1	64K	USERS
TERRITORY_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">TERRITORY</a>	1	64K	USERS
DOESBUSINESSIN_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">DOESBUSINESSIN</a>	2	64K	USERS
EMPLOYEE_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">EMPLOYEE</a>	1	64K	USERS
SKILL_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">SKILL</a>	1	64K	USERS
EMPLOYEESKILLS_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">EMPLOYEESKILLS</a>	2	64K	USERS
ORDER_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">Order</a>	1	64K	USERS
PRODUCTLINE_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">PRODUCTLINE</a>	1	64K	USERS
WORKSIN_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">WORKSIN</a>	2	0	USERS
VENDOR_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">VENDOR</a>	1	0	USERS
USES_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">USES</a>	2	0	USERS
SUPPLIES_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">SUPPLIES</a>	2	0	USERS
RAWMATERIAL_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">RAWMATERIAL</a>	1	0	USERS
ORDERLINE_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">ORDERLINE</a>	2	64K	USERS

**Example Product Table:**

Step1:

CREATE TABLE Product

(

ProductID NUMBER (11, 0) NOT NULL,

ProductLineID NUMBER (11, 0),

ProductDescription VARCHAR2 (50),

ProductFinish VARCHAR2 (20),

ProductStandardPrice DECIMAL (6, 2),

CONSTRAINT Product\_PK PRIMARY KEY (ProductID),

CONSTRAINT Product\_FK1 FOREIGN KEY (ProductLineID)

REFERENCES ProductLine (ProductLineID)

);

Step2:

Insert query and Drop query

INSERT INTO Product (ProductID, ProductDescription,  
ProductFinish, ProductStandardPrice, ProductLineID)

VALUES (1, 'End Table', 'Cherry', 175, 1);

DROP TABLE Product CASCADE CONSTRAINTS;

Step3:

Product table:

ProductID	ProductLineID	ProductDescription	ProductFinish	ProductStandardPrice
1	1	End Table	Cherry	175.00
2	2	Coffee Table	Natural Ash	200.00
3	2	Computer Desk	Natural Ash	375.00
4	3	Entertainment Center	Natural Maple	650.00
5	1	Writers Desk	Cherry	325.00
6	2	8-Drawer Desk	White Ash	750.00
7	2	Dining Table	Natural Ash	800.00
8	3	Computer Desk	Walnut	250.00

## Examples of different queries that can be performed with the actual data

### 1. Customers who have ordered multiple times.

```

SELECT * FROM (SELECT COUNT(0.CustomerID)Frequent_Customer, C.*
FROM Orders O LEFT JOIN Customer C ON C.CustomerID = 0.CustomerID GROUP BY 0.CustomerID) a
WHERE a.Frequent_Customer > 1;

```

Frequent_Customer	CustomerID	CustomerName	CustomerAddress	CustomerCity	CustomerState	CustomerPostalCode
2	1	Contemporary Casuals	1355 S Hines Blvd	Gainesville	FL	32601-2871

### 2. Biggest sales of the year.

```

SELECT sales.OrderID, SUM(sales.cost)Total, C.*
FROM (SELECT OrderID, OL.ProductID, OL.OrderedQuantity * P.ProductStandardPrice AS cost
FROM OrderLine OL LEFT JOIN Product P ON P.ProductID = OL.ProductID) sales
LEFT JOIN Orders O ON O.OrderID = sales.OrderID
LEFT JOIN Customer C ON C.CustomerID = sales.CustomerID
GROUP BY sales.OrderID
ORDER BY Total DESC;

```

OrderID	Total	CustomerID	CustomerName	CustomerAddress	CustomerCity	CustomerState	CustomerPostalCode
1009	3700.00	4	Eastern Furniture	1925 Beltline Rd.	Carteret	NJ	07008-3188
1006	2900.00	2	Value Furniture	15145 S.W. 17th St.	Plano	TX	75094-7743
1010	2500.00	1	Contemporary Casuals	1355 S Hines Blvd	Gainesville	FL	32601-2871
1004	2000.00	5	Impressions	5585 Westcott Ct.	Sacramento	CA	94206-4056
1005	1950.00	3	Home Furnishings	1900 Allard Ave.	Albany	NY	12209-1125
1008	1875.00	12	Battle Creek Furniture	345 Capitol Ave. SW	Battle Creek	MI	49015-3401
1002	1875.00	8	California Classics	816 Peach Rd.	Santa Clara	CA	96915-7754
1001	1400.00	1	Contemporary Casuals	1355 S Hines Blvd	Gainesville	FL	32601-2871
1003	1125.00	15	Mountain Scenes	4132 Main Street	Ogden	UT	84403-4432
1007	925.00	11	American Euro Lifestyles	2424 Missouri Ave N.	Prospect Park	NJ	07508-5621

### 3. Check orders for the month of October.

```
SELECT * FROM Orders WHERE OrderDate LIKE '2010-10%'
```

1 Result			2 Profiler	3 Messages	4 Table Data
(Read Only)					
OrderID	CustomerID	OrderDate			
1001	1	2010-10-21 02:15:36			
1002	8	2010-10-21 03:45:26			
1003	15	2010-10-22 01:12:00			
1004	5	2010-10-22 01:15:20			
1005	3	2010-10-24 05:30:22			
1006	2	2010-10-24 06:10:42			
1007	11	2010-10-27 12:09:35			
1008	12	2010-10-30 11:55:35			

### 4. Check states with multiple orders.

```
SELECT * FROM (SELECT COUNT(CustomerState) Popular_States, CustomerState
FROM Customer GROUP BY CustomerState) states
WHERE states.Popular_States > 1;
SELECT * FROM Orders O LEFT JOIN Customer C ON C.CustomerID = O.CustomerID
WHERE O.CustomerID IS NOT NULL
ORDER BY O.CustomerID ASC;
```

1 Result		2 Profiler	3 Messages	4 Table Data	5 Info
(Read Only)					
Popular_States	CustomerState				
2	CA				
3	FL				
2	NJ				

## VII. Conclusion

In this project we have implemented a business model for a furniture company name “Pine Valley Furniture Company”, where our initial task started with a proper analysis of the requirements. Later, our focus is on drafting the requirements and build a data warehouse with three main categories consisting of User data and process, application, Decision, components and Tools. Our further progress led into designing the flowcharts to have a better understanding of the workflow. Analysing the data sources is an important point to quote, which is implemented on OLTP system, where we retain the information about different customer details and other product information that we use in this project. Later in this portion, we discovered patterns up on which, queries are executed. Data modelling technique is used in this project design in order to create data cube, up on which queries are performed on multi dimension, which can be even termed as multi dimensional data cube. One of the important technique that we have undergone is, ETL, obrevating as Extraction, Transformation and Loading, where we use these to store data, clean data and Load data. These are created and executed on queries, in order to achieve the requirements mentioned.

## APPENDIX A

Following is the code for creating the database, table and the entries in them.

```
/* Formatted on 11/30/2017 3:09:43 PM (QP5 v5.185.11230.41888) */
```

```
DROP TABLE Uses      CASCADE CONSTRAINTS;
DROP TABLE WorksIn   CASCADE CONSTRAINTS;
DROP TABLE WorkCenter CASCADE CONSTRAINTS;
DROP TABLE DoesBusinessIn CASCADE CONSTRAINTS;
DROP TABLE EmployeeSkills CASCADE CONSTRAINTS;
DROP TABLE Supplies  CASCADE CONSTRAINTS;
DROP TABLE ProducedIn CASCADE CONSTRAINTS;
DROP TABLE OrderLine CASCADE CONSTRAINTS;
DROP TABLE Product   CASCADE CONSTRAINTS;
DROP TABLE ProductLine CASCADE CONSTRAINTS;
DROP TABLE Order      CASCADE CONSTRAINTS;
DROP TABLE Salesperson CASCADE CONSTRAINTS;
DROP TABLE Vendor     CASCADE CONSTRAINTS;
DROP TABLE Skill      CASCADE CONSTRAINTS;
DROP TABLE RawMaterial CASCADE CONSTRAINTS;
DROP TABLE Territory  CASCADE CONSTRAINTS;
DROP TABLE Employee   CASCADE CONSTRAINTS;
DROP TABLE Customer   CASCADE CONSTRAINTS;
```

```
CREATE USER pvalley IDENTIFIED BY "welcome123";
GRANT DBA TO pvalley;
```

```
CREATE TABLE Customer
(
  CustomerID      NUMBER (11, 0) NOT NULL,
  CustomerName    VARCHAR2 (25) NOT NULL,
  CustomerAddress VARCHAR2 (30),
  CustomerCity    VARCHAR2 (20),
  CustomerState   CHAR (2),
  CustomerPostalCode VARCHAR2 (10),
  CONSTRAINT Customer_PK PRIMARY KEY (CustomerID)
);
```



```
CREATE TABLE Territory
(
  TerritoryID   NUMBER (11, 0) NOT NULL,
  TerritoryName VARCHAR2 (50),
  CONSTRAINT Territory_PK PRIMARY KEY (TerritoryID)
);
```

```
CREATE TABLE DoesBusinessIn
(
  CustomerID   NUMBER (11, 0) NOT NULL,
  TerritoryID  NUMBER (11, 0) NOT NULL,
  CONSTRAINT DoesBusinessIn_PK PRIMARY KEY (CustomerID, TerritoryID),
  CONSTRAINT DoesBusinessIn_FK1 FOREIGN KEY
    (CustomerID)
    REFERENCES Customer (CustomerID),
  CONSTRAINT DoesBusinessIn_FK2 FOREIGN KEY
    (TerritoryID)
    REFERENCES Territory (TerritoryID)
);
```

```
CREATE TABLE Employee
(
  EmployeeID      VARCHAR2 (10) NOT NULL,
  EmployeeName    VARCHAR2 (25),
  EmployeeAddress VARCHAR2 (30),
  EmployeeBirthDate DATE,
  EmployeeCity    VARCHAR2 (20),
  EmployeeState   CHAR (2),
  EmployeeZipCode VARCHAR2 (10),
  EmployeeDateHired DATE,
  EmployeeSupervisor VARCHAR2 (10),
  CONSTRAINT Employee_PK PRIMARY KEY (EmployeeID)
);
```

```
CREATE TABLE Skill
(
    SkillID      VARCHAR2 (12) NOT NULL,
    SkillDescription  VARCHAR2 (30),
    CONSTRAINT Skill_PK PRIMARY KEY (SkillID)
);
```

```
CREATE TABLE EmployeeSkills
(
    EmployeeID  VARCHAR2 (10) NOT NULL,
    SkillID     VARCHAR2 (12) NOT NULL,
    CONSTRAINT EmployeeSkills_PK PRIMARY KEY (EmployeeID, SkillID),
    CONSTRAINT EmployeeSkills_FK1 FOREIGN KEY
        (EmployeeID)
        REFERENCES Employee (EmployeeID),
    CONSTRAINT EmployeeSkills_FK2 FOREIGN KEY
        (SkillID)
        REFERENCES Skill (SkillID)
);
```

```
CREATE TABLE Order
(
    OrderID     NUMBER (11, 0) NOT NULL,
    CustomerID  NUMBER (11, 0),
    OrderDate   DATE DEFAULT SYSDATE,
    CONSTRAINT Order_PK PRIMARY KEY (OrderID),
    CONSTRAINT Order_FK1 FOREIGN KEY
        (CustomerID)
        REFERENCES Customer (CustomerID)
);
```

```
CREATE TABLE WorkCenter
(
  WorkCenterID      VARCHAR2 (12) NOT NULL,
  WorkCenterLocation VARCHAR2 (30),
  CONSTRAINT WorkCenter_PK PRIMARY KEY (WorkCenterID)
);
```

```
CREATE TABLE ProductLine
(
  ProductLineID   NUMBER (11, 0) NOT NULL,
  ProductLineName VARCHAR2 (50),
  CONSTRAINT ProductLine_PK PRIMARY KEY (ProductLineID)
);
```

```
CREATE TABLE Product
(
  ProductID          NUMBER (11, 0) NOT NULL,
  ProductLineID      NUMBER (11, 0),
  ProductDescription  VARCHAR2 (50),
  ProductFinish       VARCHAR2 (20),
  ProductStandardPrice DECIMAL (6, 2),
  CONSTRAINT Product_PK PRIMARY KEY (ProductID),
  CONSTRAINT Product_FK1 FOREIGN KEY
    (ProductLineID)
    REFERENCES ProductLine (ProductLineID)
);
```

```

CREATE TABLE ProducedIn
(
  ProductID    NUMBER (11, 0) NOT NULL,
  WorkCenterID VARCHAR2 (12) NOT NULL,
  CONSTRAINT ProducedIn_PK PRIMARY KEY (ProductID, WorkCenterID),
  CONSTRAINT ProducedIn_FK1 FOREIGN KEY
    (ProductID)
    REFERENCES Product (ProductID),
  CONSTRAINT ProducedIn_FK2 FOREIGN KEY
    (WorkCenterID)
    REFERENCES WorkCenter (WorkCenterID)
);

```

```

CREATE TABLE OrderLine (
  OrderID      NUMBER (11, 0) NOT NULL,
  ProductID    NUMBER (11, 0) NOT NULL,
  OrderedQuantity  NUMBER (11, 0),
  CONSTRAINT OrderLine_PK PRIMARY KEY (OrderID, ProductID),
  CONSTRAINT OrderLine_FK1 FOREIGN KEY (OrderID) REFERENCES Order
(OrderID),
  CONSTRAINT OrderLine_FK2 FOREIGN KEY
    (ProductID)
    REFERENCES Product (ProductID)
);

```

```

CREATE UNIQUE INDEX PVALLEY.OrderLine_PK ON PVALLEY.ORDERLINE
(ORDERID, PRODUCTID)
LOGGING
STORAGE (
  BUFFER_POOL    DEFAULT
  FLASH_CACHE    DEFAULT
  CELL_FLASH_CACHE DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE PVALLEY.ORDERLINE
ADD CONSTRAINT OrderLine_PK
PRIMARY KEY (ORDERID, PRODUCTID);

```

```
CREATE TABLE RawMaterial
(
  MaterialID      VARCHAR2 (12) NOT NULL,
  MaterialName    VARCHAR2 (30),
  MaterialStandardCost  DECIMAL (6, 2),
  UnitOfMeasure   VARCHAR2 (10),
  CONSTRAINT RawMaterial_PK PRIMARY KEY (MaterialID)
);
```

```
CREATE TABLE Salesperson
(
  SalespersonID   NUMBER (11, 0) NOT NULL,
  SalespersonName VARCHAR2 (25),
  SalespersonPhone VARCHAR2 (50),
  SalespersonFax  VARCHAR2 (50),
  TerritoryID     NUMBER (11, 0),
  CONSTRAINT Salesperson_PK PRIMARY KEY (SalesPersonID),
  CONSTRAINT Salesperson_FK1 FOREIGN KEY
    (TerritoryID)
    REFERENCES Territory (TerritoryID)
);
```

```
CREATE TABLE Vendor
(
  VendorID      NUMBER (11, 0) NOT NULL,
  VendorName    VARCHAR2 (25),
  VendorAddress VARCHAR2 (30),
  VendorCity    VARCHAR2 (20),
  VendorState   CHAR (2),
  VendorZipcode VARCHAR2 (50),
  VendorFax     VARCHAR2 (10),
  VendorPhone   VARCHAR2 (10),
  VendorContact VARCHAR2 (50),
  VendorTaxID   VARCHAR2 (50),
  CONSTRAINT Vendor_PK PRIMARY KEY (VendorID)
);
```

```
CREATE TABLE Supplies
```

```
(
  VendorID      NUMBER (11, 0) NOT NULL,
  MaterialID     VARCHAR2 (12) NOT NULL,
  SuppliesUnitPrice  DECIMAL (6, 2),
  CONSTRAINT Supplies_PK PRIMARY KEY (VendorID, MaterialID),
  CONSTRAINT Supplies_FK1 FOREIGN KEY
    (MaterialID)
    REFERENCES RawMaterial (MaterialID),
  CONSTRAINT Supplies_FK2 FOREIGN KEY
    (VendorID)
    REFERENCES Vendor (VendorID)
);
```

```
CREATE TABLE Uses
```

```
(
  ProductID      NUMBER (11, 0) NOT NULL,
  MaterialID     VARCHAR2 (12) NOT NULL,
  GoesIntoQuantity  INTEGER,
  CONSTRAINT Uses_PK PRIMARY KEY (ProductID, MaterialID),
  CONSTRAINT Uses_FK1 FOREIGN KEY (ProductID) REFERENCES Product
(ProductID),
  CONSTRAINT Uses_FK2 FOREIGN KEY
    (MaterialID)
    REFERENCES RawMaterial (MaterialID)
);
```

```
CREATE TABLE WorksIn
```

```
(
  EmployeeID     VARCHAR2 (10) NOT NULL,
  WorkCenterID   VARCHAR2 (12) NOT NULL,
  CONSTRAINT WorksIn_PK PRIMARY KEY (EmployeeID, WorkCenterID),
  CONSTRAINT WorksIn_FK1 FOREIGN KEY
    (EmployeeID)
    REFERENCES Employee (EmployeeID),
  CONSTRAINT WorksIn_FK2 FOREIGN KEY
    (WorkCenterID)
    REFERENCES WorkCenter (WorkCenterID)
);
```

DELETE FROM TABLE Users;

DELETE FROM TABLE WorksIn;

DELETE FROM TABLE WorkCenter;

DELETE FROM TABLE DoesBusinessIn;

DELETE FROM TABLE EmployeeSkills;

DELETE FROM TABLE Supplies;

DELETE FROM TABLE ProducedIn;

DELETE FROM TABLE OrderLine;

DELETE FROM TABLE Product;

DELETE FROM TABLE ProductLine;

DELETE FROM TABLE Order;

DELETE FROM TABLE Salesperson;

DELETE FROM TABLE Vendor;

DELETE FROM TABLE Skill;

DELETE FROM TABLE RawMaterial;

DELETE FROM TABLE Territory;

DELETE FROM TABLE Employee;

DELETE FROM TABLE Customer;

INSERT INTO Customer (CustomerID,  
CustomerName,

```
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (1,  
        'Contemporary Casuals',  
        '1355 S Hines Blvd',  
        'Gainesville',  
        'FL',  
        '32601-2871');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (2,  
        'Value Furniture',  
        '15145 S.W. 17th St.',  
        'Plano',  
        'TX',  
        '75094-7743');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (3,  
        'Home Furnishings',  
        '1900 Allard Ave.',  
        'Albany',  
        'NY',  
        '12209-1125');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,
```



```
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (4,  
        'Eastern Furniture',  
        '1925 Beltline Rd.',  
        'Carteret',  
        'NJ',  
        '07008-3188');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (5,  
        'Impressions',  
        '5585 Westcott Ct.',  
        'Sacramento',  
        'CA',  
        '94206-4056');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (6,  
        'Furniture Gallery',  
        '325 Flatiron Dr.',  
        'Boulder',  
        'CO',  
        '80514-4432');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,
```

```
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (7,  
        'Period Furniture',  
        '394 Rainbow Dr.',  
        'Seattle',  
        'WA',  
        '97954-5589');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (8,  
        'California Classics',  
        '816 Peach Rd.',  
        'Santa Clara',  
        'CA',  
        '96915-7754');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (9,  
        'M and H Casual Furniture',  
        '3709 First Street',  
        'Clearwater',  
        'FL',  
        '34620-2314');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,
```

```
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (10,  
        'Seminole Interiors',  
        '2400 Rocky Point Dr.',  
        'Seminole',  
        'FL',  
        '34646-4423');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (11,  
        'American Euro Lifestyles',  
        '2424 Missouri Ave N.',  
        'Prospect Park',  
        'NJ',  
        '07508-5621');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (12,  
        'Battle Creek Furniture',  
        '345 Capitol Ave. SW',  
        'Battle Creek',  
        'MI',  
        '49015-3401');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,
```

```
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (13,  
        'Heritage Furnishings',  
        '66789 College Ave.',  
        'Carlisle',  
        'PA',  
        '17013-8834');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (14,  
        'Kaneohe Homes',  
        '112 Kiowai St.',  
        'Kaneohe',  
        'HI',  
        '96744-2537');  
  
INSERT INTO Customer (CustomerID,  
        CustomerName,  
        CustomerAddress,  
        CustomerCity,  
        CustomerState,  
        CustomerPostalCode)  
VALUES (15,  
        'Mountain Scenes',  
        '4132 Main Street',  
        'Ogden',  
        'UT',  
        '84403-4432');  
  
INSERT INTO Territory (TerritoryID, TerritoryName)
```

```
VALUES (1, 'SouthEast');
```

```
INSERT INTO Territory (TerritoryID, TerritoryName)  
VALUES (2, 'SouthWest');
```

```
INSERT INTO Territory (TerritoryID, TerritoryName)  
VALUES (3, 'NorthEast');
```

```
INSERT INTO Territory (TerritoryID, TerritoryName)  
VALUES (4, 'NorthWest');
```

```
INSERT INTO Territory (TerritoryID, TerritoryName)  
VALUES (5, 'Central');
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (1, 1);
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (1, 2);
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (2, 2);
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (3, 3);
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (4, 3);
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (5, 2);
```

```
INSERT INTO DoesBusinessIn (CustomerID, TerritoryID)  
VALUES (6, 5);
```

```
INSERT INTO Employee (EmployeeID,
```

```
        EmployeeName,  
        EmployeeAddress,  
        EmployeeCity,  
        EmployeeState,  
        EmployeeZip,  
        EmployeeDateHired,  
        EmployeeBirthDate,  
        EmployeeSupervisor)  
VALUES ('123-44-345',  
        'Jim Jason',  
        '2134 Hilltop Rd',  
        ",  
        'TN',  
        ",  
        '12/Jun/99',  
        ",  
        '454-56-768');
```

```
INSERT INTO Employee (EmployeeID,  
        EmployeeName,  
        EmployeeAddress,  
        EmployeeCity,  
        EmployeeState,  
        EmployeeZip,  
        EmployeeDateHired,  
        EmployeeBirthDate,  
        EmployeeSupervisor)  
VALUES ('454-56-768',  
        'Robert Lewis',  
        '17834 Deerfield Ln',  
        'Nashville',  
        'TN',  
        ",  
        '01/Jan/99',  
        ",  
        ");
```

```
INSERT INTO Skill (SkillID, SkillDescription)
```

```
VALUES ('BS12', '12in Band Saw');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('QC1', 'Quality Control');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('RT1', 'Router');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('SO1', 'Sander-Orbital');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('SB1', 'Sander-Belt');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('TS10', '10in Table Saw');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('TS12', '12in Table Saw');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('UC1', 'Upholstery Cutter');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('US1', 'Upholstery Sewer');

INSERT INTO Skill (SkillID, SkillDescription)
VALUES ('UT1', 'Upholstery Tacker');

INSERT INTO EmployeeSkills (EmployeeID, SkillID)
VALUES ('123-44-345', 'BS12');

INSERT INTO EmployeeSkills (EmployeeID, SkillID)
VALUES ('123-44-345', 'RT1');

INSERT INTO EmployeeSkills (EmployeeID, SkillID)
VALUES ('454-56-768', 'BS12');

INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1001, '21/Oct/10', 1);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1002, '21/Oct/10', 8);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1003, '22/Oct/10', 15);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1004, '22/Oct/10', 5);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1005, '24/Oct/10', 3);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1006, '24/Oct/10', 2);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1007, '27/Oct/10', 11);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1008, '30/Oct/10', 12);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1009, '05/Nov/10', 4);
```

```
INSERT INTO Order (OrderID, OrderDate, CustomerID)
VALUES (1010, '05/Nov/10', 1);
```

```
INSERT INTO ProductLine (ProductLineID, ProductLineName)
VALUES (1, 'Cherry Tree');
```

```
INSERT INTO ProductLine (ProductLineID, ProductLineName)
VALUES (2, 'Scandinavia');
```

```
INSERT INTO ProductLine (ProductLineID, ProductLineName)
VALUES (3, 'Country Look');
```

```
INSERT INTO Product (ProductID,
                    ProductDescription,
```



```
        ProductFinish,  
        ProductStandardPrice,  
        ProductLineID)  
VALUES (1,  
        'End Table',  
        'Cherry',  
        175,  
        1);  
  
INSERT INTO Product (ProductID,  
        ProductDescription,  
        ProductFinish,  
        ProductStandardPrice,  
        ProductLineID)  
VALUES (2,  
        'Coffee Table',  
        'Natural Ash',  
        200,  
        2);  
  
INSERT INTO Product (ProductID,  
        ProductDescription,  
        ProductFinish,  
        ProductStandardPrice,  
        ProductLineID)  
VALUES (3,  
        'Computer Desk',  
        'Natural Ash',  
        375,  
        2);  
  
INSERT INTO Product (ProductID,  
        ProductDescription,  
        ProductFinish,  
        ProductStandardPrice,  
        ProductLineID)  
VALUES (4,  
        'Entertainment Center',  
        'Natural Maple',  
        650,
```

3);

```
INSERT INTO Product (ProductID,  
                    ProductDescription,  
                    ProductFinish,  
                    ProductStandardPrice,  
                    ProductLineID)
```

```
VALUES (5,  
        'Writers Desk',  
        'Cherry',  
        325,  
        1);
```

```
INSERT INTO Product (ProductID,  
                    ProductDescription,  
                    ProductFinish,  
                    ProductStandardPrice,  
                    ProductLineID)
```

```
VALUES (6,  
        '8-Drawer Desk',  
        'White Ash',  
        750,  
        2);
```

```
INSERT INTO Product (ProductID,  
                    ProductDescription,  
                    ProductFinish,  
                    ProductStandardPrice,  
                    ProductLineID)  
VALUES (7,  
        'Dining Table',  
        'Natural Ash',  
        800,  
        2);
```

```
INSERT INTO Product (ProductID,  
                    ProductDescription,  
                    ProductFinish,  
                    ProductStandardPrice,  
                    ProductLineID)  
VALUES (8,  
        'Computer Desk',  
        'Walnut',  
        250,  
        3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)  
VALUES (1001, 1, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)  
VALUES (1001, 2, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)  
VALUES (1001, 4, 1);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)  
VALUES (1002, 3, 5);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)  
VALUES (1003, 3, 3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1004, 6, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1004, 8, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1005, 4, 3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1006, 4, 1);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1006, 5, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1006, 7, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1007, 1, 3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1007, 2, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1008, 3, 3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1008, 8, 3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1009, 4, 2);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1009, 7, 3);
```

```
INSERT INTO OrderLine (OrderID, ProductID, OrderedQuantity)
VALUES (1010, 8, 10);
```

```
INSERT INTO Salesperson (SalesPersonID,  
                          SalesPersonName,  
                          SalesPersonPhone,  
                          SalesPersonFax,  
                          TerritoryID)  
VALUES (1,  
        'Doug Henny',  
        '8134445555',  
        "",  
        1);
```

```
INSERT INTO Salesperson (SalesPersonID,  
                          SalesPersonName,  
                          SalesPersonPhone,  
                          SalesPersonFax,  
                          TerritoryID)  
VALUES (2,  
        'Robert Lewis',  
        '8139264006',  
        "",  
        2);
```

```
INSERT INTO Salesperson (SalesPersonID,  
                          SalesPersonName,  
                          SalesPersonPhone,  
                          SalesPersonFax,  
                          TerritoryID)  
VALUES (3,  
        'William Strong',  
        '5053821212',  
        "",  
        3);
```

```
INSERT INTO Salesperson (SalesPersonID,  
                          SalesPersonName,  
                          SalesPersonPhone,  
                          SalesPersonFax,  
                          TerritoryID)  
VALUES (4,  
        'Julie Dawson',  
        '4355346677',  
        "",  
        4);
```

```
INSERT INTO Salesperson (SalesPersonID,  
                          SalesPersonName,  
                          SalesPersonPhone,  
                          SalesPersonFax,  
                          TerritoryID)  
VALUES (5,  
        'Jacob Winslow',  
        '2238973498',  
        "",  
        5);
```

```
INSERT INTO WorkCenter (WorkCenterID, WorkCenterLocation)  
VALUES ('SM1', 'Main Saw Mill');
```

```
INSERT INTO WorkCenter (WorkCenterID, WorkCenterLocation)  
VALUES ('WR1', 'Warehouse and Receiving');
```

```
INSERT INTO WORKS_IN (EmployeeID, Work_CenterID)  
VALUES ('123-44-345', 'SM1');
```

Following is the table summary of all the tables used in the database.

### Table Summary

<b>Table Name</b>	<b># Cols</b>	<b># Indexes</b>	<b>Size</b>	<b>Tablespace</b>
<u>CUSTOMER</u>	6	1	64K	USERS
<u>DOESBUSINESSIN</u>	2	1	64K	USERS
<u>EMPLOYEE</u>	9	1	64K	USERS
<u>EMPLOYEESKILLS</u>	2	1	64K	USERS
<u>ORDERLINE</u>	3	1	64K	USERS
<u>Order</u>	3	1	64K	USERS
<u>PRODUCEDIN</u>	2	1	0	USERS
<u>PRODUCT</u>	5	1	64K	USERS
<u>PRODUCTLINE</u>	2	1	64K	USERS
<u>RAWMATERIAL</u>	4	1	0	USERS
<u>SALESPERSON</u>	5	1	64K	USERS
<u>SKILL</u>	2	1	64K	USERS
<u>SUPPLIES</u>	3	1	0	USERS
<u>TERRITORY</u>	2	1	64K	USERS
<u>USES</u>	3	1	0	USERS
<u>VENDOR</u>	10	1	0	USERS
<u>WORKCENTER</u>	2	1	64K	USERS
<u>WORKSIN</u>	2	1	0	USERS

Following is the table definition or the structure of the tables included in the database.

### Customer Table

Columns in Table CUSTOMER

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>CUSTOMERID</b>	NUMBER(11)	Y		
2	CUSTOMERNAME	VARCHAR2(25 BYTE)	Y		
3	CUSTOMERADDRESS	VARCHAR2(30 BYTE)			
4	CUSTOMERCITY	VARCHAR2(20 BYTE)			
5	CUSTOMERSTATE	CHAR(2 BYTE)			
6	CUSTOMERPOSTALCODE	VARCHAR2(10 BYTE)			

Indexes on Table CUSTOMER

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">CUSTOMER_PK</a>	NORMAL	CUSTOMERID	<none>				

PK, UK, & Check Constraints on Table CUSTOMER

Constraint Name	Type	Condition	Status	Columns
CUSTOMER_PK	Primary Key		ENABLED	CUSTOMERID

Table CUSTOMER does  
not have...

A comment

Partitions

Foreign Key Constraints

### DoesBusinessIn Table

Columns in Table DOESBUSINESSIN

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>CUSTOMERID<sub>1</sub></b>	NUMBER(11)	Y		
2	<b>TERRITORYID<sub>2</sub></b>	NUMBER(11)	Y		

Indexes on Table DOESBUSINESSIN

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">DOESBUSINESSIN_PK</a>	NORMAL	CUSTOMERID, TERRITORYID	<none>				

Foreign Key Constraints on Table DOESBUSINESSIN

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
DOESBUSINESSIN_FK1	CUSTOMERID	PVALLEY	<a href="#">CUSTOMER</a>	CUSTOMERID	NO ACTION	ENABLED	No	No
DOESBUSINESSIN_FK2	TERRITORYID	PVALLEY	<a href="#">TERRITORY</a>	TERRITORYID	NO ACTION	ENABLED	No	No

PK, UK, & Check Constraints on Table DOESBUSINESSIN

Constraint Name	Type	Condition	Status	Columns
DOESBUSINESSIN_PK	Primary Key		ENABLED	CUSTOMERID, TERRITORYID

Table  
DOESBUSINESSIN  
does not have...

A comment

Partitions



## Employee Table

Columns in Table EMPLOYEE

Col ID	Column Name	Data Type	Not Null Data Def. Comments
1	<b>EMPLOYEEID</b>	VARCHAR2(10 BYTE)	Y
2	EMPLOYEEENAME	VARCHAR2(25 BYTE)	
3	EMPLOYEEADDRESS	VARCHAR2(30 BYTE)	
4	EMPLOYEEBIRTHDATE	DATE	
5	EMPLOYEECITY	VARCHAR2(20 BYTE)	
6	EMPLOYEESTATE	CHAR(2 BYTE)	
7	EMPLOYEEZIPCODE	VARCHAR2(10 BYTE)	
8	EMPLOYEEDATEHIRED	DATE	
9	EMPLOYEESUPERVISOR	VARCHAR2(10 BYTE)	

Indexes on Table EMPLOYEE

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">EMPLOYEE_PK</a>	NORMAL	EMPLOYEEID	<none>				

PK, UK, &amp; Check Constraints on Table EMPLOYEE

Constraint Name	Type	Condition	Status	Columns
EMPLOYEE_PK	Primary Key		ENABLED	EMPLOYEEID

Table EMPLOYEE does  
not have...

A comment

Partitions

Foreign Key Constraints

## EmployeeSkills Table

Columns in Table EMPLOYEESKILLS

Col ID	Column Name	Data Type	Not Null Data Def. Comments
1	<b>EMPLOYEEID<sub>1</sub></b>	VARCHAR2(10 BYTE)	Y
2	<b>SKILLID<sub>2</sub></b>	VARCHAR2(12 BYTE)	Y

Indexes on Table EMPLOYEESKILLS

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">EMPLOYEESKILLS_PK</a>	NORMAL	EMPLOYEEID, SKILLID	<none>				

Foreign Key Constraints on Table EMPLOYEESKILLS

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
EMPLOYEESKILLS_FK1	EMPLOYEEID	PVALLEY	<a href="#">EMPLOYEE</a>	EMPLOYEEID	NO ACTION	ENABLED	No	No
EMPLOYEESKILLS_FK2	SKILLID	PVALLEY	<a href="#">SKILL</a>	SKILLID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table EMPLOYEESKILLS

Constraint Name	Type	Condition	Status	Columns
EMPLOYEESKILLS_PK	Primary Key		ENABLED	EMPLOYEEID, SKILLID

Table  
EMPLOYEESKILLS  
does not have...

A comment

Partitions

## Orders Table

### Columns in Table Order

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>ORDERID</b>	NUMBER(11)	Y		
2	CUSTOMERID	NUMBER(11)			
3	ORDERDATE	DATE		SYSDATE	

### Indexes on Table Order

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">ORDER_PK</a>	NORMAL	ORDERID	<none>				

### Foreign Key Constraints on Table Order

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
ORDER_FK1	CUSTOMERID	PVALLEY	<a href="#">CUSTOMER</a>	CUSTOMERID	NO ACTION	ENABLED	No	No

### PK, UK, & Check Constraints on Table Order

Constraint Name	Type	Condition	Status	Columns
ORDER_PK	Primary Key		ENABLED	ORDERID

Table Order  
does not  
have...  
A comment  
Partitions

## OrderLine Table

### Columns in Table ORDERLINE

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>ORDERID<sub>1</sub></b>	NUMBER(11)	Y		
2	<b>PRODUCTID<sub>2</sub></b>	NUMBER(11)	Y		
3	ORDEREDQUANTITY	NUMBER(11)			

### Indexes on Table ORDERLINE

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">ORDERLINE_PK</a>	NORMAL	ORDERID, PRODUCTID	<none>				

### Foreign Key Constraints on Table ORDERLINE

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
ORDERLINE_FK1	ORDERID	PVALLEY	<a href="#">Order</a>	ORDERID	NO ACTION	ENABLED	No	No
ORDERLINE_FK2	PRODUCTID	PVALLEY	<a href="#">PRODUCT</a>	PRODUCTID	NO ACTION	ENABLED	No	No

### PK, UK, & Check Constraints on Table ORDERLINE

Constraint Name	Type	Condition	Status	Columns
ORDERLINE_PK	Primary Key		ENABLED	ORDERID, PRODUCTID

Table  
ORDERLINE  
does not  
have...  
A comment  
Partitions

## ProducedIn Table

Columns in Table PRODUCEDIN

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>PRODUCTID<sub>1</sub></b>	NUMBER(11)	Y		
2	<b>WORKCENTERID<sub>2</sub></b>	VARCHAR2(12 BYTE)	Y		

Indexes on Table PRODUCEDIN

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">PRODUCEDIN_PK</a>	NORMAL	PRODUCTID, WORKCENTERID	<none>				

Foreign Key Constraints on Table PRODUCEDIN

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
PRODUCEDIN_FK1	PRODUCTID	PVALLEY	<a href="#">PRODUCT</a>	PRODUCTID	NO ACTION	ENABLED	No	No
PRODUCEDIN_FK2	WORKCENTERID	PVALLEY	<a href="#">WORKCENTER</a>	WORKCENTERID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table PRODUCEDIN

Constraint Name	Type	Condition	Status	Columns
PRODUCEDIN_PK	Primary Key		ENABLED	PRODUCTID, WORKCENTERID

Table  
PRODUCEDIN  
does not have...  
A comment  
Partitions

## Product Table

Columns in Table PRODUCT

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>PRODUCTID</b>	NUMBER(11)	Y		
2	PRODUCTLINEID	NUMBER(11)			
3	PRODUCTDESCRIPTION	VARCHAR2(50 BYTE)			
4	PRODUCTFINISH	VARCHAR2(20 BYTE)			
5	PRODUCTSTANDARDPRICE	NUMBER(6,2)			

Indexes on Table PRODUCT

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">PRODUCT_PK</a>	NORMAL	PRODUCTID	<none>				

Foreign Key Constraints on Table PRODUCT

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
PRODUCT_FK1	PRODUCTLINEID	PVALLEY	<a href="#">PRODUCTLINE</a>	PRODUCTLINEID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table PRODUCT

Constraint Name	Type	Condition	Status	Columns
PRODUCT_PK	Primary Key		ENABLED	PRODUCTID

Table  
PRODUCT  
does not  
have...  
A comment  
Partitions

**ProductLine Table**

Columns in Table PRODUCTLINE

Col ID	Column Name	Data Type	Not Null Data Def. Comments
1	<b>PRODUCTLINEID</b>	NUMBER(11)	Y
2	PRODUCTLINENAME	VARCHAR2(50 BYTE)	

Indexes on Table PRODUCTLINE

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">PRODUCTLINE_PK</a>	NORMAL	PRODUCTLINEID	<none>				

PK, UK, &amp; Check Constraints on Table PRODUCTLINE

Constraint Name	Type	Condition	Status	Columns
PRODUCTLINE_PK	Primary Key		ENABLED	PRODUCTLINEID

Table PRODUCTLINE

does not have...

A comment

Partitions

Foreign Key Constraints

**RawMaterial Table**

Columns in Table RAWMATERIAL

Col ID	Column Name	Data Type	Not Null Data Def. Comments
1	<b>MATERIALID</b>	VARCHAR2(12 BYTE)	Y
2	MATERIALNAME	VARCHAR2(30 BYTE)	
3	MATERIALSTANDARDCOST	NUMBER(6,2)	
4	UNITOFMEASURE	VARCHAR2(10 BYTE)	

Indexes on Table RAWMATERIAL

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">RAWMATERIAL_PK</a>	NORMAL	MATERIALID	<none>				

PK, UK, &amp; Check Constraints on Table RAWMATERIAL

Constraint Name	Type	Condition	Status	Columns
RAWMATERIAL_PK	Primary Key		ENABLED	MATERIALID

Table RAWMATERIAL

does not have...

A comment

Partitions

Foreign Key Constraints

**SalesPerson Table**

Columns in Table SALESPERSON

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>SALESPERSONID</b>	NUMBER(11)	Y		
2	SALESPERSONNAME	VARCHAR2(25 BYTE)			
3	SALESPERSONPHONE	VARCHAR2(50 BYTE)			
4	SALESPERSONFAX	VARCHAR2(50 BYTE)			
5	TERRITORYID	NUMBER(11)			

Indexes on Table SALESPERSON

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">SALESPERSON_PK</a>	NORMAL	SALESPERSONID	<none>				

Foreign Key Constraints on Table SALESPERSON

Constraint Name	Columns	Ref Table	Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
SALESPERSON_FK1	TERRITORYID	PVALLEY		<a href="#">TERRITORY</a>	TERRITORYID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table SALESPERSON

Constraint Name	Type	Condition	Status	Columns
SALESPERSON_PK	Primary Key		ENABLED	SALESPERSONID

Table  
SALESPERSON  
does not have...  
A comment  
Partitions

**Skills Table**

Columns in Table SKILL

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>SKILLID</b>	VARCHAR2(12 BYTE)	Y		
2	SKILLDESCRIPTION	VARCHAR2(30 BYTE)			

Indexes on Table SKILL

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">SKILL_PK</a>	NORMAL	SKILLID	<none>				

PK, UK, &amp; Check Constraints on Table SKILL

Constraint Name	Type	Condition	Status	Columns
SKILL_PK	Primary Key		ENABLED	SKILLID

Table SKILL does not  
have...  
A comment  
Partitions  
Foreign Key Constraints

## Supplies Table

Columns in Table SUPPLIES

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>VENDORID<sub>1</sub></b>	NUMBER(11)	Y		
2	<b>MATERIALID<sub>2</sub></b>	VARCHAR2(12 BYTE)	Y		
3	SUPPLIESUNITPRICE	NUMBER(6,2)			

Indexes on Table SUPPLIES

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">SUPPLIES_PK</a>	NORMAL	VENDORID, MATERIALID	<none>				

Foreign Key Constraints on Table SUPPLIES

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
SUPPLIES_FK1	MATERIALID	PVALLEY	<a href="#">RAWMATERIAL</a>	MATERIALID	NO ACTION	ENABLED	No	No
SUPPLIES_FK2	VENDORID	PVALLEY	<a href="#">VENDOR</a>	VENDORID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table SUPPLIES

Constraint Name	Type	Condition	Status	Columns
SUPPLIES_PK	Primary Key		ENABLED	VENDORID, MATERIALID

Table  
SUPPLIES  
does not  
have...  
A comment  
Partitions

## Territory Table

Columns in Table TERRITORY

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>TERRITORYID</b>	NUMBER(11)	Y		
2	TERRITORYNAME	VARCHAR2(50 BYTE)			

Indexes on Table TERRITORY

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">TERRITORY_PK</a>	NORMAL	TERRITORYID	<none>				

PK, UK, &amp; Check Constraints on Table TERRITORY

Constraint Name	Type	Condition	Status	Columns
TERRITORY_PK	Primary Key		ENABLED	TERRITORYID

Table TERRITORY does  
not have...  
A comment  
Partitions  
Foreign Key Constraints

## Uses Table

Columns in Table USES

Col ID	Column Name	Data Type	Not Null Data Def. Comments
1	<b>PRODUCTID<sub>1</sub></b>	NUMBER(11)	Y
2	<b>MATERIALID<sub>2</sub></b>	VARCHAR2(12 BYTE)	Y
3	GOESINTOQUANTITY	INTEGER	

Indexes on Table USES

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">USES_PK</a>	NORMAL	PRODUCTID, MATERIALID	<none>				

Foreign Key Constraints on Table USES

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
USES_FK1	PRODUCTID	PVALLEY	<a href="#">PRODUCT</a>	PRODUCTID	NO ACTION	ENABLED	No	No
USES_FK2	MATERIALID	PVALLEY	<a href="#">RAWMATERIAL</a>	MATERIALID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table USES

Constraint Name	Type	Condition	Status	Columns
USES_PK	Primary Key		ENABLED	PRODUCTID, MATERIALID

Table USES

does not

have...

A comment

Partitions

## Vendor Table

Columns in Table VENDOR

Col ID	Column Name	Data Type	Not Null Data Def. Comments
1	<b>VENDORID</b>	NUMBER(11)	Y
2	VENDORNAME	VARCHAR2(25 BYTE)	
3	VENDORADDRESS	VARCHAR2(30 BYTE)	
4	VENDORCITY	VARCHAR2(20 BYTE)	
5	VENDORSTATE	CHAR(2 BYTE)	
6	VENDORZIPCODE	VARCHAR2(50 BYTE)	
7	VENDORFAX	VARCHAR2(10 BYTE)	
8	VENDORPHONE	VARCHAR2(10 BYTE)	
9	VENDORCONTACT	VARCHAR2(50 BYTE)	
10	VENDORTAXID	VARCHAR2(50 BYTE)	

Indexes on Table VENDOR

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">VENDOR_PK</a>	NORMAL	VENDORID	<none>				

PK, UK, &amp; Check Constraints on Table VENDOR

Constraint Name	Type	Condition	Status	Columns
VENDOR_PK	Primary Key		ENABLED	VENDORID

Table VENDOR does

not have...

A comment

Partitions

Foreign Key Constraints

## WorkCenter Table

Columns in Table WORKCENTER

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>WORKCENTERID</b>	VARCHAR2(12 BYTE)	Y		
2	WORKCENTERLOCATION	VARCHAR2(30 BYTE)			

Indexes on Table WORKCENTER

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">WORKCENTER_PK</a>	NORMAL	WORKCENTERID	<none>				

PK, UK, &amp; Check Constraints on Table WORKCENTER

Constraint Name	Type	Condition	Status	Columns
WORKCENTER_PK	Primary Key		ENABLED	WORKCENTERID

Table WORKCENTER  
does not have...

A comment

Partitions

Foreign Key Constraints

## WorksIn Table

Columns in Table WORKSIN

Col ID	Column Name	Data Type	Not Null	Data Def.	Comments
1	<b>EMPLOYEEID<sub>1</sub></b>	VARCHAR2(10 BYTE)	Y		
2	<b>WORKCENTERID<sub>2</sub></b>	VARCHAR2(12 BYTE)	Y		

Indexes on Table WORKSIN

Uniqueness	Index Name	Type	Columns	Partitions	Part Type	Part Cols	Subpart Type	Subpart Cols
UNIQUE	<a href="#">WORKSIN_PK</a>	NORMAL	EMPLOYEEID, WORKCENTERID	<none>				

Foreign Key Constraints on Table WORKSIN

Constraint Name	Columns	Ref Table Owner	Ref Table Name	Ref Columns	Delete Rule	Status	Deferrable	Deferred
WORKSIN_FK1	EMPLOYEEID	PVALLEY	<a href="#">EMPLOYEE</a>	EMPLOYEEID	NO ACTION	ENABLED	No	No
WORKSIN_FK2	WORKCENTERID	PVALLEY	<a href="#">WORKCENTER</a>	WORKCENTERID	NO ACTION	ENABLED	No	No

PK, UK, &amp; Check Constraints on Table WORKSIN

Constraint Name	Type	Condition	Status	Columns
WORKSIN_PK	Primary Key		ENABLED	EMPLOYEEID, WORKCENTERID

Table  
WORKSIN  
does not  
have...

A comment

Partitions



Following are the screenshots of the table made with the entries.

**Customer Table**

CustomerID	CustomerName	CustomerAddress	CustomerCity	CustomerState	CustomerPostalCode
1	Contemporary Casuals	1355 S Hines Blvd	Gainesville	FL	32601-2871
2	Value Furniture	15145 S.W. 17th St.	Plano	TX	75094-7743
3	Home Furnishings	1900 Allard Ave.	Albany	NY	12209-1125
4	Eastern Furniture	1925 Beltline Rd.	Carteret	NJ	07008-3188
5	Impressions	5585 Westcott Ct.	Sacramento	CA	94206-4056
6	Furniture Gallery	325 Flatiron Dr.	Boulder	CO	80514-4432
7	Period Furniture	394 Rainbow Dr.	Seattle	WA	97954-5589
8	California Classics	816 Peach Rd.	Santa Clara	CA	96915-7754
9	M and H Casual Furniture	3709 First Street	Clearwater	FL	34620-2314
10	Seminole Interiors	2400 Rocky Point Dr.	Seminole	FL	34646-4423
11	American Euro Lifestyles	2424 Missouri Ave N.	Prospect Park	NJ	07508-5621
12	Battle Creek Furniture	345 Capitol Ave. SW	Battle Creek	MI	49015-3401
13	Heritage Furnishings	66789 College Ave.	Carlisle	PA	17013-8834
14	Kaneohe Homes	112 Kiowai St.	Kaneohe	HI	96744-2537
15	Mountain Scenes	4132 Main Street	Ogden	UT	84403-4432

**DoesBussinessIn Table**

CustomerID	TerritoryID
1	1
1	2
2	2
3	3
4	3
5	2
6	5

**Employee Table**

EmployeeID	EmployeeName	EmployeeAddress	EmployeeBirthDate	EmployeeCity	EmployeeState	EmployeeZip	EmployeeDateHired	EmployeeSupervisor
123-44-345	Jim Jason	2134 Hilltop Rd	1972-10-22		TN		1999-06-12	454-56-768
454-56-768	Robert Lewis	17834 Deerfield Ln	1969-10-11	Nashville	TN		1999-01-01	

**OrderLine Table**

OrderID	ProductID	OrderedQuantity
1001	1	2
1001	2	2
1001	4	1
1002	3	5
1003	3	3
1004	6	2
1004	8	2
1005	4	3
1006	4	1
1006	5	2
1006	7	2
1007	1	3
1007	2	2
1008	3	3
1008	8	3
1009	4	2
1009	7	3
1010	8	10

**EmployeeSkills Table**

EmployeeID	SkillID
123-44-345	BS12
123-44-345	RT1
454-56-768	BS12

**Orders Table**

OrderID	CustomerID	OrderDate
1001	1	2010-10-22
1002	8	2010-10-21
1003	15	2010-10-22
1004	5	2010-10-22
1005	3	2010-10-24
1006	2	2010-10-24
1007	11	2010-10-27
1008	12	2010-10-30
1009	4	2010-11-05
1010	1	2010-11-05

**Product Table**

ProductID	ProductLineID	ProductDescription	ProductFinish	ProductStandardPrice
1	1	End Table	Cherry	175.00
2	2	Coffee Table	Natural Ash	200.00
3	2	Computer Desk	Natural Ash	375.00
4	3	Entertainment Center	Natural Maple	650.00
5	1	Writers Desk	Cherry	325.00
6	2	8-Drawer Desk	White Ash	750.00
7	2	Dining Table	Natural Ash	800.00
8	3	Computer Desk	Walnut	250.00

**ProductLine Table**

ProductLineID	ProductLineName
1	Cherry Tree
2	Scandinavia
3	Country Look

**SalesPerson Table**

SalespersonID	SalespersonName	SalespersonPhone	SalespersonFax	TerritoryID
1	Doug Henny	8134445555		1
2	Robert Lewis	8139264006		2
3	William Strong	5053821212		3
4	Julie Dawson	4355346677		4
5	Jacob Winslow	2238973498		5

**Skills Table**

SkillID	SkillDescription
BS12	12in Band Saw
QC1	Quality Control
RT1	Router
SO1	Sander-Orbital
SB1	Sander-Belt
TS10	10in Table Saw
TS12	12in Table Saw
UC1	Upholstery Cutter
US1	Upholstery Sewer
UT1	Upholstery Tacker

Territory Table

TerritoryID	TerritoryName
1	SouthEast
2	SouthWest
3	NorthEast
4	NorthWest
5	Central

WorkCenter Table

WorkCenterID	WorkCenterLocation
SM1	Main Saw Mill
WR1	Warehouse and Receiving

WorksIn Table

EmployeeID	WorkCenterID
123-44-345	SM1

Following is the indexing done in the database between the tables.

Schema PVALLEY  
Oracle Version 11.2.0.4.0  
As Of 11/30/2017 3:40:25 PM

[Summary](#)

[Indexes](#)

[Tables](#)

Index Summary							
Index Name	Index Type	Uniqueness	Table Owner	Table Name	Columns	Size	Tablespace
PRODUCT_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">PRODUCT</a>	1	64K	USERS
PRODUCEDIN_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">PRODUCEDIN</a>	2	0	USERS
SALESPERSON_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">SALESPERSON</a>	1	64K	USERS
WORKCENTER_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">WORKCENTER</a>	1	64K	USERS
CUSTOMER_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">CUSTOMER</a>	1	64K	USERS
TERRITORY_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">TERRITORY</a>	1	64K	USERS
DOESBUSINESSIN_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">DOESBUSINESSIN</a>	2	64K	USERS
EMPLOYEE_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">EMPLOYEE</a>	1	64K	USERS
SKILL_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">SKILL</a>	1	64K	USERS
EMPLOYEE SKILLS_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">EMPLOYEE SKILLS</a>	2	64K	USERS
ORDER_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">Order</a>	1	64K	USERS
PRODUCTLINE_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">PRODUCTLINE</a>	1	64K	USERS
WORKSIN_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">WORKSIN</a>	2	0	USERS
VENDOR_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">VENDOR</a>	1	0	USERS
USES_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">USES</a>	2	0	USERS
SUPPLIES_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">SUPPLIES</a>	2	0	USERS
RAWMATERIAL_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">RAWMATERIAL</a>	1	0	USERS
ORDERLINE_PK	NORMAL	UNIQUE	PVALLEY	<a href="#">ORDERLINE</a>	2	64K	USERS