Programmer’s Manual  
JUNGLE MARKET™

short line

Team 4

Prepared by Justin Pilecki  
December 5, 2020

Version 1.0

# Table of Contents

# Introduction

# 1.1 Purpose

# 1.2 Scope

# 1.3 Overview

# Getting Started

# 2.1 File setup

# 2.2 File descriptions

# 2.3 Modifying database

# Coding Standards

# 3.1 PL/SQL

# 3.2 HTML

# 3.3 CSS

# 3.4 PHP

# Modules

# 4.1 User Interface and Site Map

# 4.2 Relational Structure of Database

# Introduction

# 1.1 Purpose

# This manual will provide programmers with the overall framework of the JUNGLE MARKET application from the front-end webpage to the back-end database.

# 1.2 Scope

# The scope of the JUNGLE MARET application is to provide a fully connected e-commerce platform. The actions of a customer browsing the website will have an impact on the underlying database pertaining to the information that must be tracked. The specific database information that is impacted through the website includes:

# Product: inventory, price, sales Finances: total site-wide profit, company bonus Customer: ID, name, phone, email Employee: salary

# 1.3 Overview

# This document will provide summary and initial setup of files, order of execution, coding standards, and module descriptions.

# Getting Started

# 2.1 File Setup

# The creation of JUNGLE MARKET’s database tables is located in the file seDesign.sql and must be the first file run. It is the file that all other files act upon. An optional database population file titled sePopulate.sql is able to initially populate the database with any data it is programmed to insert, although this can be done at the command line instead.

# Several PHP files make up the front-end interface. These file are: main\_hub.php, login\_page.php, inventory\_update.php, inventory.php, insert-function.php, hsu\_conn\_sess.php, customer\_form.php, and after\_login\_temp.php. All of these PHP files must be located within the public\_html working directory for access through a web browser.

# The remainder of the SQL files, project\_insert.sql, testing.sql, and total\_sales\_view.sql, should be executed after the setup of the PHP files.

# Several image files are used in the front-end as JPG and PNG. These are: logo.png, apple.jpg, banana.jpg, baobab.jpg, cacao.jpg, cedar.jpg, coconut.jpg, date.jpg, eucalyptus.jpg, fern.jpg, foxglove.jpg, gumbolimbo.jpg, maple.jpg, oak.jpg, palm.jpg, and redwood.jpg. These images should be in the public\_html directory along with the PHP files.

# The last file is 458Project.css, which should be set up in the public\_html directory along with the PHP files.

# 2.2 File Descriptions

# seDesign.sql - The initial [drop and] creation of database tables and their attribute fields.

# sePopulate.sql - An optional population of the seDesign tables. [NOTE: if skipped, tables must be populated manually through command line]

# project\_insert.sql - An SQL procedure that inserts a new customer into the Customer database table. Most attributes inserted come from the new customer form in the customer\_form.php file, although the ID attribute is selected from the previous row’s ID with the integer 2 added.

# testing.sql - An SQL procedure which compares the actual sales attribute from the Quarterly\_Finances table against the projected sales attribute. If the actual sales exceed projected sales, the Employee table is updated to add $3,000.00 USD to every employee’s salary, reflecting a company-wide bonus. [When actual sales does not exceed projected sales, there is no bonus given]

# total\_sales\_view.sql - An SQL view which computes the sum of sales from every product in the Product table into a single attribute containing total profit.

# main\_hub.php - The hub for controlling web sessions and displaying the JUNGLE MARKET™ logo.

# customer\_form.php - Displays the HTML form for a new customer to fill out and submit.

# after\_login\_temp.php - A PHP function that announces successful login to the user.

# login\_page.php - Displays an HTML form for Oracle login credentials with a submit button.

# inventory\_update.php - A PHP function that updates the stock of - and profit from - a product purchased through the web interface. The impacted database table is Product.

# inventory.php - A PHP function that queries the database Product table for product ID, name, price, stock quantity, and a picture, displaying these in the webpage for the customer.

# insert\_function.php - A PHP function that inserts a new customer into the database Customer table using the inputs from the form is customer\_form.php.

# hsu\_conn\_sess.php - A PHP function that attempts to connect an Oracle username and password to the database.

# destroy\_and\_exit.php - A PHP function that displays an error message and destroys the current session.

# 458Project.css - A CSS file which styles the HTML content of the website.

# logo.png - The current logo of the store name in PNG format.

# apple.jpg - A thumbnail for one of the current products for sale in JPG format.

# banana.jpg - A thumbnail for one of the current products for sale in JPG format.

# baobab.jpg - A thumbnail for one of the current products for sale in JPG format.

# cacao.jpg - A thumbnail for one of the current products for sale in JPG format.

# cedar.jpg - A thumbnail for one of the current products for sale in JPG format.

# coconut.jpg - A thumbnail for one of the current products for sale in JPG format.

# date.jpg - A thumbnail for one of the current products for sale in JPG format.

# eucalyptus.jpg - A thumbnail for one of the current products for sale in JPG format.

# fern.jpg - A thumbnail for one of the current products for sale in JPG format.

# foxglove.jpg - A thumbnail for one of the current products for sale in JPG format.

# gumbolimbo.jpg - A thumbnail for one of the current products for sale in JPG format.

# maple.jpg - A thumbnail for one of the current products for sale in JPG format.

# oak.jpg - A thumbnail for one of the current products for sale in JPG format.

# palm.jpg - A thumbnail for one of the current products for sale in JPG format.

# redwood.jpg - A thumbnail for one of the current products for sale in JPG format.

# 2.3 Modifying database

# Inserting new data into a database table can be achieved by entering into the command line in SQLplus or editing sePopulate.sql with the desired edits. Insertion into tables follows the syntax:

# insert into [table] values ( [val1], [val2], [val3], [val4], [val5], [val6] );

# Deletion of data from a database will usually be done at the command line in SQLplus. Deletion follows the syntax:

# delete from [table] where [condition];

# Updating data in a database table will usually be done at the command line in SQLplus. Update follows the syntax:

# update [table] set [column] = [value] where [condition];

# Coding Standards

# 3.1 PL/SQL - Long statements meeting or exceeding 80 characters will be intended and continued on the next line. Every statement [such as WHERE, FROM, and INTO] is to have its own line.

# 3.2 HTML - All elements are written in lower-case and are both explicitly opened and closed. Attribute values are written within quotation marks.

# 3.3 CSS - External CSS stylesheets are used instead of in-line CSS, and all CSS property names are written in lower-case. Curly braces have their own lines, and properties are indented.

# 3.4 PHP - Expression tags are used inline without semicolons, regular tags all end with a semicolon. Print and echo statements are entirely avoided in favor of expression tags.

# 

# 

# Modules

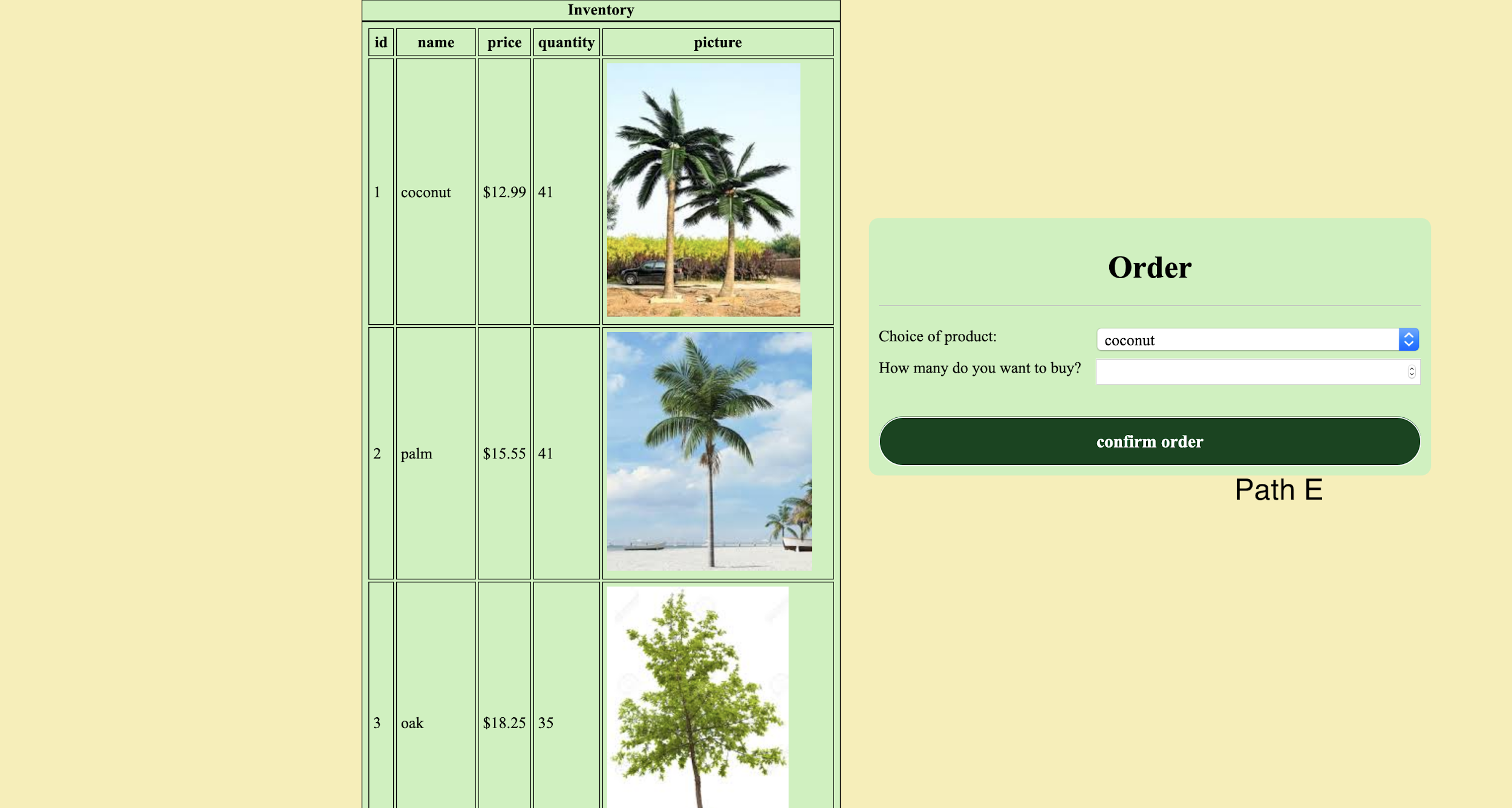
# 4.1 User Interface and Site Map

# **Path A**: Submits Oracle credentials and directs to the inventory page for shopping. **Path B**: Submits Oracle credentials and directs to a new customer sign up page. **Path C**: Inserts new customer into database. **Path D**: Directs to inventory page. **Path E**: Submits order for an amount of product selected. **Path F**: Return to main page.



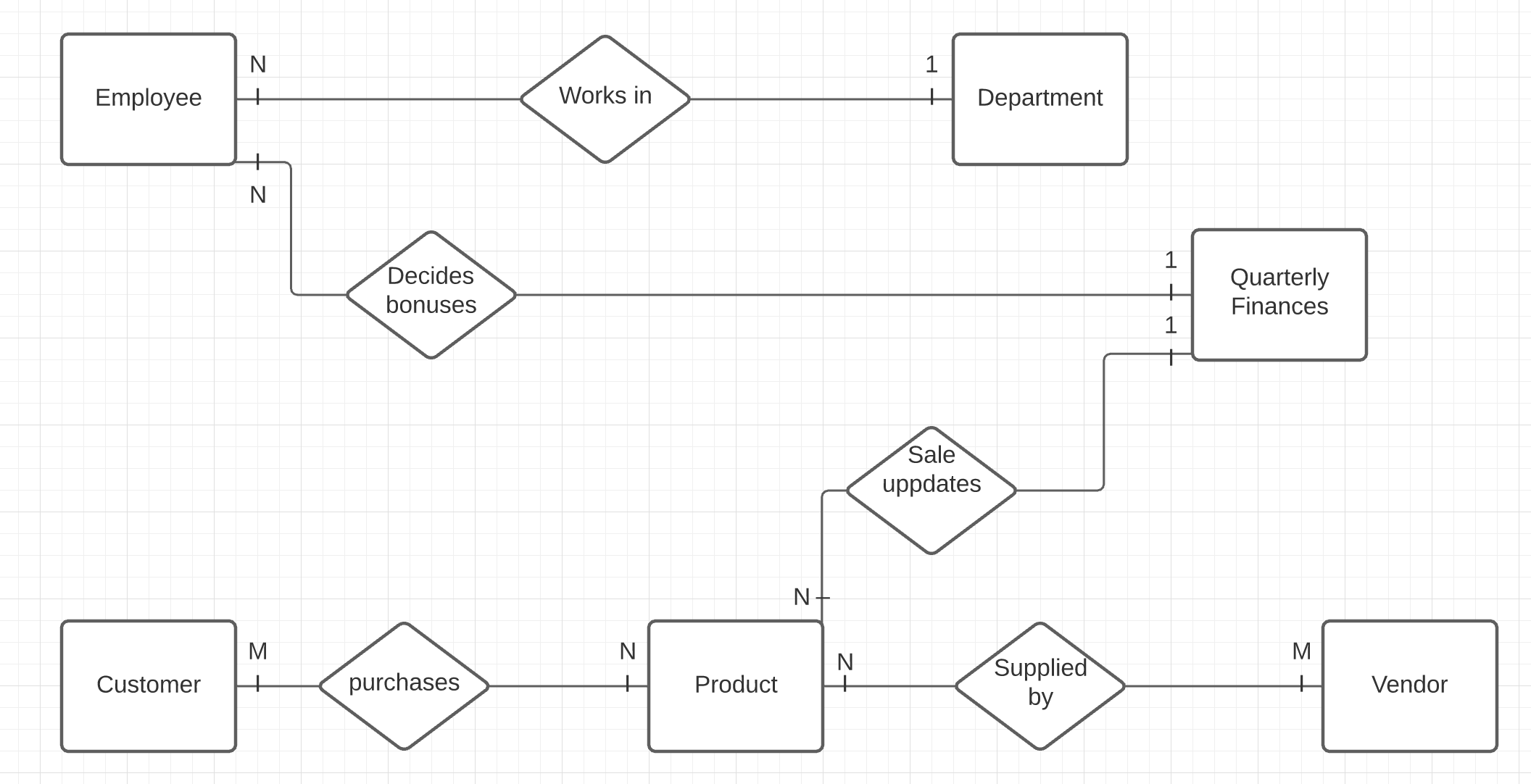


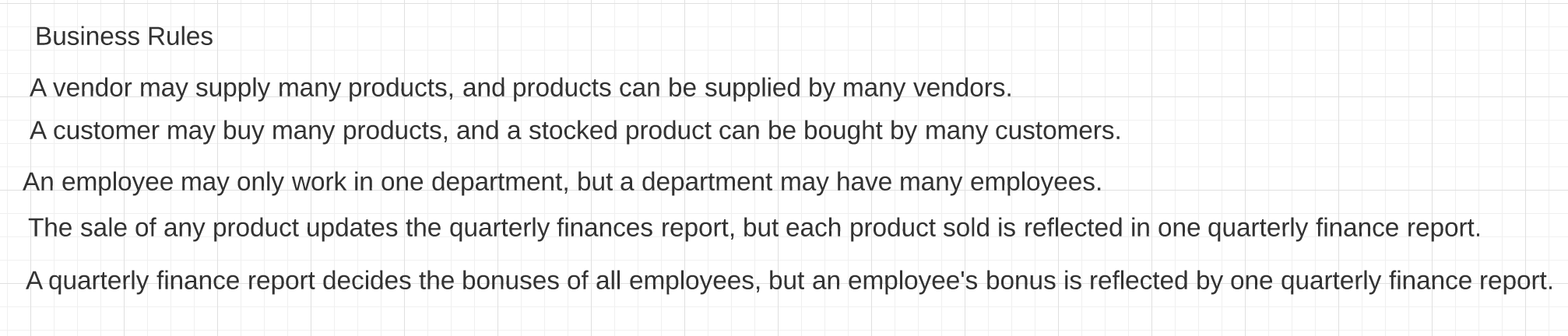






# 4.2 Relational Structure of Database





# Cardinalities: 1 = one N, M = many

# *Fin.*

short dash