

# ***Codd Manufacturing Company***



Database Management Final Project

Zachary Miller

Due:

Wednesday April 20th, 2016

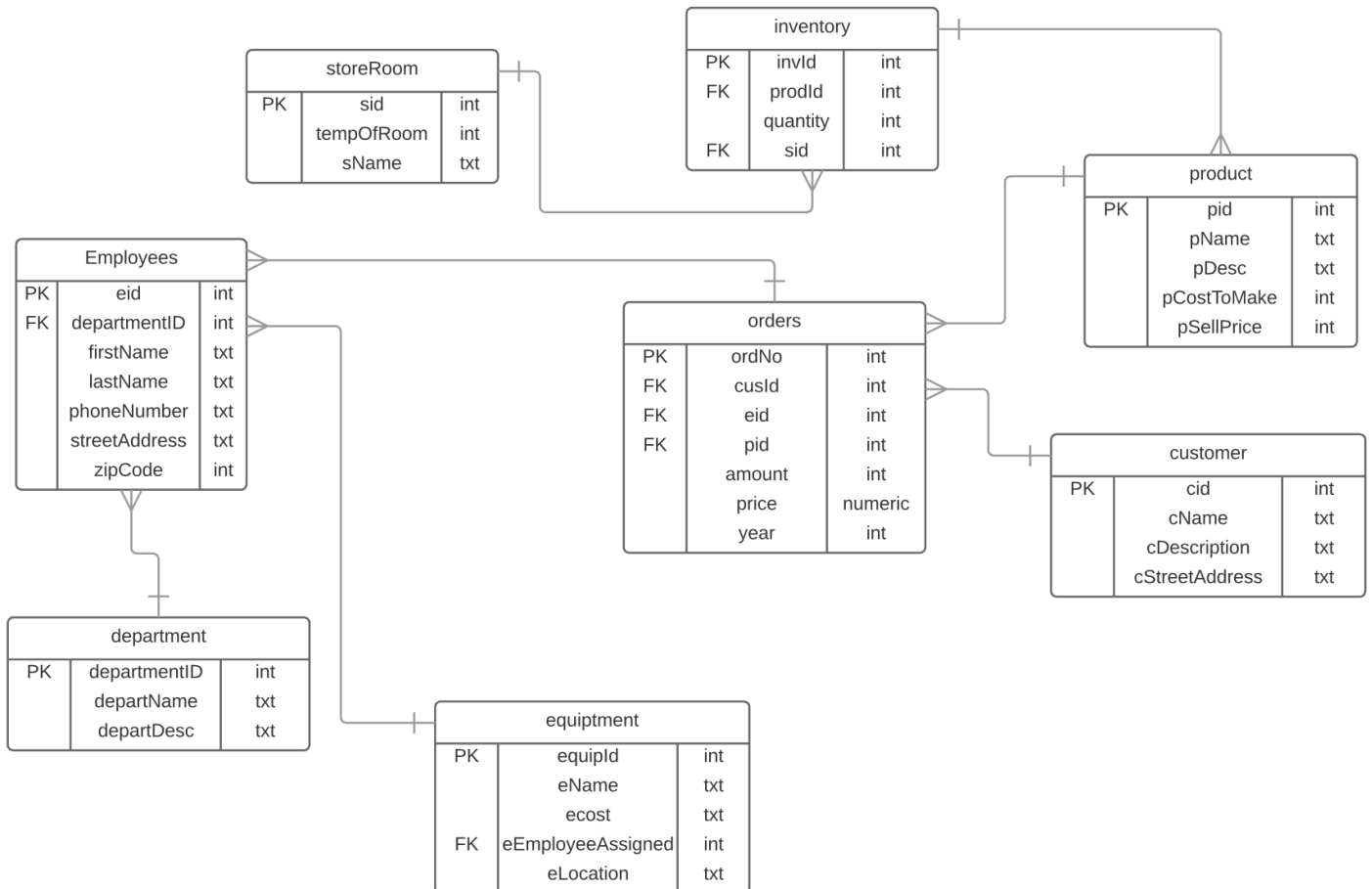
## Table of Contents

Cover page.....	1
Table of Contents.....	2
Executive Summary.....	3
Entity Relation Diagram.....	4
Tables.....	5 - 12
Views.....	13
Report.....	14
Objective.....	15
Implementation Notes.....	16
Known Problems.....	17
Future Enhancements.....	18

***Executive Summary:***

There are manufacturing companies throughout the world. Most are in need of more organization skills and arrangement. They run into problems such as inventory recording, client information tracking, employee tracking, asset tracking, amongst other issues. If these manufacturing companies were given a more comprehensive database and technical support, these companies could excel at their mission; provide the best possible products at a low price.

## Entity Relationship Diagram:



**Tables:** The following section shows and explains each table in the database.

EMPLOYEES: This table displays all of the employees in the company. It has their personal information along with their department id to determine where in the plant they work.

Functional Dependency:

eid -> firstName,lastName,phoneNumber,streetAddress,zipCode, departmentId

Table Data:

	eid integer	firstname text	lastname text	phonenum text	streetaddress text	zipcode integer	departmentid integer
1	1	Zach	Miller	5185775933	24 Burton Rd Loudonville NY	12211	1
2	2	Andrew	Smith	8455775933	87 Sunset Rd Lagrange NY	12603	3
3	3	Pari	Hake	8456738765	86 Sunset Ave Poughkeepsie NY	12601	4
4	4	Patrick	Chavez	2034589760	78 Tram Dr Albany NY	12211	5
5	5	Kimberly	Heyde	7083589760	24 West Cedar Poughkeepsie NY	12601	2
6	6	Peyton	Patota	9142879908	127 Cream Street Albany NY	12211	3
7	7	Paul	Orlando	5184589760	65 Sunset Ave Albany NY	12211	6
8	8	Kayla	Hello	2034589760	73 Helloitsme Dr Albany NY	12211	6
9	9	Sydney	Chardonayee	6794589760	198 Gone Dr Pougkeepsie NY	12601	3
10	10	Jessica	Leggio	5674589760	400 Dance Dr Albany NY	12211	3
11	11	Nicole	Piccone	2034502760	654 Hello St Albany NY	12211	1
12	12	Lauren	Saltz	2034789760	54 Mansion Terr Poughkeepsie NY	12211	1

```
CREATE TABLE public.employees
```

```
(
  eid integer NOT NULL,
  firstname text NOT NULL,
  lastname text NOT NULL,
  phonenum text NOT NULL,
  streetaddress text NOT NULL,
  zipcode integer NOT NULL,
  departmentid integer NOT NULL,
  primary Key (eid)
);
```

DEPARTMENT: This table displays the departmentId, which is the primary key, department name and the department description. This table helps to see what each employee is assigned to.

Functional Dependencies: departmentId -> departName, departDesc

Table Data:

	departmentid integer	departname text	departdesc text
1	1	Tech Department	Technology Department
2	2	Accounting Department	Accounting Department
3	3	Management Department	Managerial Positions
4	4	Janitorial Department	Cleaning positions
5	5	Production Department	Manufacturing Production positions
6	6	Grounds Department	Grounds Management positions

```
CREATE TABLE public.department
(
  departmentid integer NOT NULL,
  departname text NOT NULL,
  departdesc text,
  PRIMARY KEY (departmentid)
);
```

INVENTORY: This table shows us the quantity of each product that the company produces

Functional Dependencies: invId-> prodId, quantity, sid

Table Data:

	invId integer	prodId integer	quantity integer	sid integer
1	1	1	10	2
2	2	2	100	3
3	3	3	0	6
4	4	4	49	3
5	5	5	14	1
6	6	6	5	4
7	7	7	1000	5
8	8	8	15	6

```
CREATE TABLE public.inventory
(
  invId integer NOT NULL,
  prodId integer NOT NULL,
  quantity integer NOT NULL,
  sid integer NOT NULL,
  PRIMARY KEY (invId)
);
```

STORE ROOM: This table stores the information regarding each room that stores the products that are manufactured in the plant.

Functional Dependencies: sid -> tempOfRoom, sName

Table Data:

	sid integer	sname text	roomtemp text
1	1	Storage Room A	45
2	2	Storage Room B	30
3	3	Storage Room C	22
4	4	Storage Room D	60
5	5	Storage Room AA	70
6	6	Storage Room AB	20

```
CREATE TABLE storeroom(
  sid integer NOT NULL,
  sname text NOT NULL,
  roomtemp text NOT NULL,
  PRIMARY KEY (sid)
);
```



CUSTOMER: This table holds information regarding the name, description, and location of the customers of the manufacturing company.

Functional Dependencies:

cid -> cName, cDescription, cStreetAddress

Table Data:

	cid integer	cname text	cdescription text	cstreetaddress text
1	1	Ayco	Secondary Selling Company	333 South Main Street Saratoga Springs, NY
2	2	Marist College	Book Store Selling Company	3433 North Road Poughkeepsie, NY
3	3	Giacomos By Marist	Pizza Distribution Company	3434 North Road Poughkeepsie, NY
4	4	Toys R Us	Secondary Selling Company	12 North Main Street Albany, NY
5	5	Target	Secondary Selling Company	15 Northmountain Street Denver, CO

```
CREATE TABLE customer
(
  cid integer NOT NULL,
  cname text NOT NULL,
  cdescription text NOT NULL,
  cstreetaddress text NOT NULL,
  PRIMARY KEY (cid)
);
```

PRODUCT: This table has information regarding the description, cost to make, selling price, and description of the products that the manufacturing company has.

Functional Dependencies:

pid -> pName, pDesc, pCostToMake, pSellPrice

Table Data:

	pid integer	pname text	pdescription text	pcosttomake numeric(15,2)	psellprice numeric(15,2)
1	1	Baseball	Professional Baseball	7.89	15.99
2	2	Pen	Professional Felt tip Pen	10.49	25.99
3	3	Child Cart	Plastic Childrens car	30.89	80.99
4	4	Loopin Chewie	Childs game-loopin chewie	2.39	17.99
5	5	Rock em Sock em	Rock em Sock em Robots Anniversary Edition	1.89	19.99
6	6	Panasonic TV	Panasonic 55 Television	100.89	700.99
7	7	Toaster	Professional style toaster	9.89	25.99
8	8	Pizza Box	Giaccomos Pizza Box 16	2.89	4.99

```
CREATE TABLE product
(
  pid integer NOT NULL,
  pname text NOT NULL,
  pdescription text NOT NULL,
  pcosttomake numeric(15,2) NOT NULL,
  psellprice numeric(15,2) NOT NULL,
  PRIMARY KEY (pid)
);
```

EQUIPMENT: This table displays information regarding the assets (equipment) that the manufacturing company owns. This has information regarding their purchase cost, location in the plant, etc.

Functional Dependencies: equipId -> eName, eCost, eEmployeeAssigned, eLocation

Table Data:

equipid integer	ename text	ecost text	eemployeeassigned integer	edescription text	elocation text
1	2 Ton Press	20,000.00	12	2 Ton Press	Production Floor
2	Lawn Mower ZT	11,000.00	2	Ariens Lawn Mower	Garage One
3	2 Ton Press	20,000.00	12	2 Ton Press	Production Floor
4	Company Truck	9,000.00	1	2013 GMC 3500	Garage Two
5	Lenovo Laptop	1,010.00	4	Travel Laptop	Managerial Department

```
CREATE TABLE equipment
(
  equipid integer NOT NULL,
  ename text NOT NULL,
  ecost text NOT NULL,
  eemployeeassigned integer NOT NULL,
  edescription text NOT NULL,
  elocation text NOT NULL,
  PRIMARY KEY (equipid)
)
```

ORDERS: This table holds information regarding the order information for the manufacturing plant. It has such data as the quantity purchased, customer information for the order, etc.

Functional dependencies:

ordno -> cusld, eid, prodld, amount, price, yearSold

Table Data:

	ordno integer	cusid integer	eid integer	prodid integer	amount integer	price numeric(15,2)	yearpurchased integer
1	1	3	1	4	10	179.99	2016
2	2	5	4	8	15	500.98	2015
3	3	3	1	8	150	60.98	2016
4	4	1	10	6	190	760.98	2015

```
CREATE TABLE public.orders
(
  ordno integer NOT NULL,
  cusid integer NOT NULL,
  eid integer NOT NULL,
  prodid integer NOT NULL,
  amount integer NOT NULL,
  price numeric(15,2) NOT NULL,
  yearpurchased integer NOT NULL
  PRIMARY KEY (ordno)
);
```

## Views:

### ASSIGNEDEMPLOYEE:

The purpose of this view is to give management a sense of who is assigned to use each piece of equipment. This can be helpful in the case of an issue with an employee or the piece of equipment, then allowing the management to take proper actions to fix the issues.

```

CREAT VIEW assignedEmployee AS
  SELECT equ.equipid,
         equ.ename,
         equ.eemployeeassigned,
         emp.eid,
         emp.firstname,
         emp.lastname,
         emp.departmentid
  FROM employees emp,
       equipment equ
 WHERE emp.eid = equ.eemployeeassigned
 ORDER BY equ.equipid ASC

```

### Sample Output:

	equipid integer	ename text	eemployeeassigned integer	eid integer	firstname text	lastname text	departmentid integer
1	1	2 Ton Press	12	12	Lauren	Saltz	1
2	2	Lawn Mower ZT	2	2	Andrew	Smith	3
3	3	2 Ton Press	12	12	Lauren	Saltz	1
4	4	Company Truck	1	1	Zach	Miller	1
5	5	Lenovo Laptop	4	4	Patrick	Chavez	5

**Reports:**

Reports are necessary for managers to check in on the process of their company among other aspects of the company and its daily operations.

**CUSTOMER PURCHASES:**

This report describes the amount of orders each customer has put in along with the dollar amount associated with their purchases.

```
SELECT ordno, cusid, prodid, SUM(price)
FROM orders
GROUP BY cusid, ordno
```

	<b>ordno</b> <b>integer</b>	<b>cusid</b> <b>integer</b>	<b>prodid</b> <b>integer</b>	<b>sum</b> <b>numeric</b>
<b>1</b>	<b>1</b>	<b>3</b>	<b>4</b>	<b>179.99</b>
<b>2</b>	<b>3</b>	<b>3</b>	<b>8</b>	<b>60.98</b>
<b>3</b>	<b>4</b>	<b>1</b>	<b>6</b>	<b>760.98</b>
<b>4</b>	<b>2</b>	<b>5</b>	<b>8</b>	<b>500.98</b>

***Objective:***

The objective of this documentation and the accompanied database is to provide a model for manufacturing companies' database architecture, support, and purpose. Though this is a small scale database meant to be a model, it can be adapted to fit the needs of larger corporations. This was designed and implemented through the use of Postgres database and PgAdmin3. This model can be an insight on how to arrange and implement on a much larger scale.

***Implementation Notes:***

- For work stations that are attached to the storage rooms, they should have access to the inventory table as well. They will be able to check and update what products are in each storage room. Such changes about the store room (Temp, name, store room status) would be able to be updated there as well.

- Work Stations throughout the work/ manufacturing floor should be able to access what products they are producing as to get the specifications on the products. They would also be able to have access to the inventory table to update the numbers in inventory once products are completed.

- Work stations in the office areas of the manufacturing plant should have access to inventory tables and equipment tables as to see and update the status of each as the order table is updated and referenced.



***Known Problems:***

Known problems with the database consist of the orders table would need to check if the inventory quantity is sufficient to fill an order. If it is not sufficient, signals should be in place to tell the production floor to being working on those specific products.

Other known issues include no information in the database describing the finances of the company. This could cause an issue to implement based upon viewing and updating rights, but those could be implementations in the future.

***Future Enhancements:***

Future enhancements would include expanding the database to include financial information regarding the sales throughout the years for the company. Issues along with this would include security and restricting access to whom can change and update the figures. This can be fixed by restricting updating access to the accounting department and overall access can be had by the database administrator and the owner of the company.

Smaller future enhancements can be implemented such as: give a shift manager delegations and providing them with access to a new table describing the hours for different employees.

Some enhancements are larger than other, but with time you can work into the database and design the needs and wants of the company.