Department Store Database

Rosemary Bulgarelli

December 1st 2014

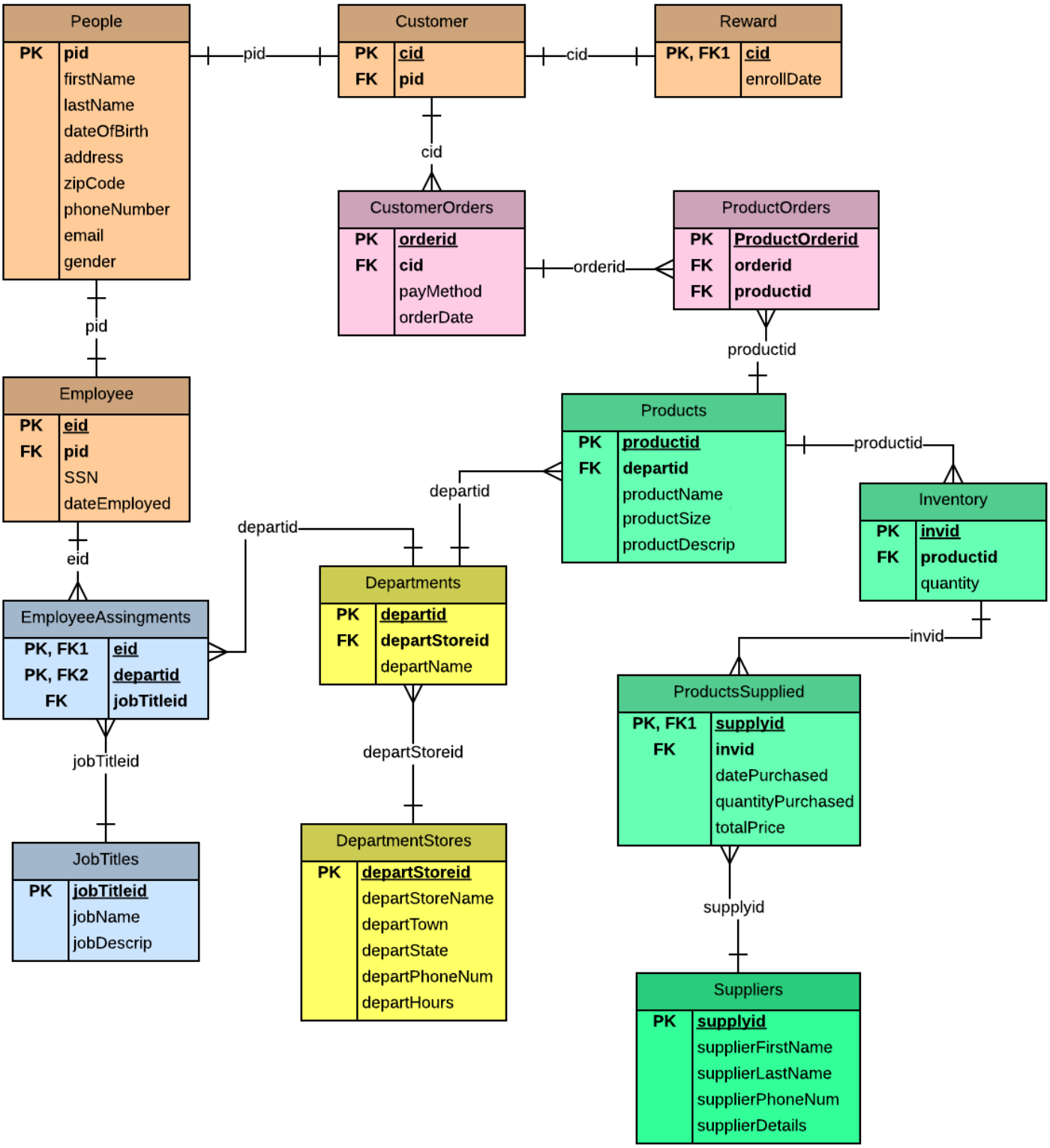
Table of Contents

Executive Summary

This document represents the design of database for a department store that sells clothing and accessories for men, women and children. The store has different departments for men and women and then smaller departments for teenagers, children and babies. The database is a way to organize inventory of the store, manage employees and store costumer information for a rewards program. Potential users could be management for these types of stores or anyone interested in running their own store.

The overall goal of this database is to organize and efficiently run a department store to allow for maximum profit. In order to accomplish these goals, the data should be stored in a structured centralized database that is easily accessible and that minimizes inconsistent and redundant data. All technical work was done in PostgreSQL 9.3.

E-R Diagram


People
PK pid
firstName
lastName
dateOfBirth
address
zipCode
phoneNumber
email
gender
pid
Customer
PK cid
FK pid
cid
Reward
PK, FK1 cid
enrollDate
cid
CustomerOrders
PK orderid
FK cid
payMethod
orderDate
orderid
ProductOrders
PK ProductOrderid
FK orderid
FK productid
productid
pid
Employee
PK eid
FK pid
SSN
dateEmployed
Products
PK productid
FK departid
productName
productSize
productDescrip
productid
Inventory
PK invid
FK productid
quantity
departid
departid
eid
EmployeeAssingments
PK, FK1 eid
PK, FK2 departid
FK jobTitleid
Departments
PK departid
FK departStoreid
departName
invid
ProductsSupplied
PK, FK1 supplyid
FK invid
datePurchased
quantityPurchased
totalPrice
jobTitleid
departStoreid
JobTitles
PK jobTitleid
jobName
jobDescrip
DepartmentStores
PK departStoreid
departStoreName
departTown
departState
departPhoneNum
departHours
supplyid
Suppliers
PK supplyid
supplierFirstName
supplierLastName
supplierPhoneNum
supplierDetails

Tables

People Table

Purpose

The people table holds all information for anyone who is a real person. It is generic for both staff and customers.

Create Statement

```
CREATE TABLE people(
  pid          char(200)      NOT NULL,
  firstName    VARCHAR(200) NOT NULL,
  lastName     VARCHAR(200) NOT NULL,
  dateOfBirth  DATE           NOT NULL,
  address      VARCHAR(200) NOT NULL,
  zipCode      VARCHAR(5)     NOT NULL,
  phoneNumber VARCHAR(12)   NOT NULL,
  email        VARCHAR(200) NOT NULL,
  gender       VARCHAR(6)     NOT NULL CHECK (gender IN ('Male', 'Female')),
  PRIMARY KEY (pid)
);
```

Functional Dependencies

pid → first, Name, lastName, dateOfBirth, address, zipCode, phoneNumber, email, gender

Sample Data

	pid text	firstname character varying(200)	lastname character varying(200)	dateofbirth date	address character varying(200)	zipcode character varying(5)	phonenumber character varying(12)	email character varying(200)	gender character varying(6)
1	p01	Rosemary	Bulgarelli	1992-01-18	P.O. Box 232	24425	541-353-7202	rosemary.bulgarelli!@hotmail.com	Female
2	p02	Olivia	Roberts	1952-12-02	123 Merry Lane	21235	541-123-7202	olivia.roberts@gmail.com	Female
3	p03	Julianna	Pokerson	1963-09-25	456 Christmas Lane	24255	422-353-7202	julianna.pokerson@gmail.com	Female
4	p04	Katheryn	Roberts	2010-10-10	789 Elf Train	14523	541-354-5602	katheryn.roberts@hotmail.com	Female
5	p05	Emily	Cook	1892-02-17	145 Person Lane	24578	551-353-7102	emily.cook@hotmail.com	Female
6	p06	Katie	Kroll	1625-06-25	1 ABC Road	05486	552-543-7202	katie.kroll!@hotmail.com	Female
7	p07	Carter	Johnson	1995-01-01	54 Carlidge road	48205	541-124-7205	carter.johnson@gmail.com	Male
8	p08	Josh	Giller	1999-02-19	63 Talk Street	15425	444-353-7770	josh.giller@yahoo.com	Male
9	p09	Brittany	Dodger	1985-11-20	72 Reading Rainbow	24565	123-456-7891	brittany.dodger@aol.com	Female
10	p10	Taylor	Nixon	1992-06-12	981 Greamoor Terrace	45231	541-015-7555	taylor.nixon@bing.com	Female
11	p11	Phil	Harriet	1990-09-25	3 Potluck Dinner Road	74235	541-012-7202	phil.harriet@hotmail.com	Male
12	p12	Tom	Ashe	1997-12-17	123 Abc Lane	47962	221-353-2222	tom.ashe@gmail.com	Male
13	p13	Jerry	Snitches	1985-05-04	24 Cross Circle	14752	152-475-7202	jerry.snitches@yahoo.com	Male
14	p14	Ryan	McMuffin	1985-06-25	928 Forest Lane	78945	999-985-7202	ryan.mcmuffin@gmail.com	Male
15	p15	Michael	Hanes	2000-04-12	345 Talk Lane	12345	991-952-7202	michael.hanes@gmail.com	Male

Customer Table

Purpose

This table separates customers from people .

Create Statement

```
CREATE TABLE customer(
  cid    char(200) NOT NULL,
  pid    char(200) NOT NULL REFERENCES people(pid),
  PRIMARY KEY (cid)
);
```

Functional Dependencies

cid →pid

Sample Data

	cid text	pid text
1	c01	p01
2	c02	p01
3	c03	p03
4	c04	p05
5	c05	p06
6	c06	p09
7	c07	p11

Employee Table

Purpose

This table separates employees from people.

Create Statement

```
CREATE TABLE employee(
  eid              char(200)       NOT NULL,
  pid              char(200)       NOT NULL REFERENCES people(pid),
  SSN              VARCHAR(9)  NOT NULL,
  DateEmployed  DATE            NOT NULL,
  PRIMARY KEY (eid)
);
```

Functional Dependencies

eid →pid, SSN, dateEmployed

Sample Data

	eid text	pid text	ssn character varying(9)	dateemployed date
1	e01	p02	123456789	2010-01-01
2	e02	p08	542586245	2012-02-04
3	e03	p10	783461255	2014-09-12
4	e04	p07	564312089	2004-10-31
5	e05	p04	731642895	2011-05-25
6	e06	p15	234679158	2002-07-14
7	e07	p07	613497258	2002-07-14
8	e08	p11	467913649	2012-09-11
9	e09	p12	134679025	2014-01-18

Reward Table

Purpose

This table shows which customers are involved in the rewards club.

Create Statement

```
CREATE TABLE rewards(
  cid          char(200) NOT NULL REFERENCES customer(cid),
  enrollDate  DATE       NOT NULL,
  PRIMARY KEY(cid)
);
```

Functional Dependencies

cid →enrollDate

Sample Data

	cid text	enrolldate date
1	c01	2012-01-18
2	c04	2013-07-16
3	c07	2014-12-21

Employee Assignments Table

Purpose

This table places employees in appropriate departments based on their job title.

Create Statement

```
CREATE TABLE employeeAssingments(
  eid              char(200) NOT NULL REFERENCES employee(eid),
  departid         char(200) NOT NULL REFERENCES departments(departid),
  jobTitleid       char(200) NOT NULL REFERENCES jobTitles(jobtitleid),
  PRIMARY KEY (eid, departid)
);
```

Functional Dependencies

eid, departid →jobTitleid

Sample Data

	eid text	departid text	jobtitleid text
1	e01	dp1	j01
2	e02	dp3	j03
3	e03	dp2	j02
4	e04	dp5	j05
5	e05	dp4	j04
6	e06	dp2	j03
7	e07	dp3	j05
8	e08	dp2	j01
9	e09	dp4	j02

Job Titles Table

Purpose

This table shows employees what their job title is and what that job requires.

Create Statement

```
CREATE TABLE jobTitles(
  jobTitleid    char(200)          NOT NULL,
  jobName       VARCHAR(200)  NOT NULL,
  jobDescrip    VARCHAR(200)  NOT NULL,
  PRIMARY KEY (jobTitleid)
);
```

Functional Dependencies

jobTitleid →jobName, jobDescrip

Sample Data

	jobtitleid text	jobname character varying(200)	jobdescrip character varying(200)
1	j01	Sales	Responsible for selling products on the floor of the store
2	j02	Manager	Responsible for all workers to be doing there job
3	j03	Phone Worker	Responsible for answering the phone and any questions asked by customers
4	j04	Tech	Responsible for keeping our website current
5	j05	Custodial	Responsible for cleaning the store after it has closed

Customer Orders Table

Purpose

This table keeps track of every order done at the store.

Create Statement

```
CREATE TABLE customerOrders(
  orderid      char(200) NOT NULL,
  cid          char(200) NOT NULL REFERENCES customers(cid),
  payMethod text         NOT NULL CHECK (payMethod IN ('Cash', 'Credit', 'Check', 'Debit')),
  orderDate  DATE        NOT NULL,
  PRIMARY KEY(orderid)
);
```

Functional Dependencies

orderid →cid, payMethod, orderDate

Sample Data

	orderid text	cid text	paymethod text	orderdate date
1	001	c01	Credit	2012-01-18
2	002	c02	Cash	2004-05-30
3	003	c03	Debit	2001-04-12
4	004	c04	Credit	2014-07-20
5	005	c05	Cash	2013-11-24
6	006	c06	Check	2007-08-28
7	007	c07	Debit	2014-12-03

Product Orders Table

Purpose

This table shows which products are bought in each customer order.

Create Statement

```
CREATE TABLE productOrders(
  productOrderid    char(200) NOT NULL,
  orderid           char(200) NOT NULL REFERENCES customerOrders(orderid),
  productid         char(200) NOT NULL REFERENCES products(productid),
  PRIMARY KEY(productOrderid)
);
```

Functional Dependencies

productOrderid→orderid, productid

Sample Data

	productorderid text	orderid text	productid text
1	pro1	001	pr2
2	pro2	003	pr3
3	pro3	002	pr9
4	pro4	004	pr17
5	pro5	005	pr24
6	pro6	007	pr19
7	pro7	006	pr13
8	pro8	004	pr12
9	pro9	002	pr25
10	pro10	001	pr21
11	pro11	007	pr32

Products Table

Purpose

This table shows every product with a description plus which department they belong in.

Create Statement

```
CREATE TABLE products(
  productid          char(200)          NOT NULL,
  departid           char(200)          NOT NULL REFERENCES departments(departid),
  productName        VARCHAR(200)   NOT NULL,
  productSize        text               NOT NULL,
  productDescrip     VARCHAR(200)   NOT NULL,
  PRIMARY KEY(productid)
);
```

Functional Dependencies

productid →departid, productName, productSize, productDescrip

Sample Data

See next page for sample…

	productid text	departid text	productname character varying(200)	productsize text	productdescrip character varying(200)
1	pr1	dp1	Jeans	30	Light-wash
2	pr2	dp1	Button Down	Medium	Plaid
3	pr3	dp1	Socks	Large	Colorful pattern
4	pr4	dp1	T-shirt	Small	California text on
5	pr5	dp1	Shoes	8	Winter Boots
6	pr6	dp1	Jacket	Extra-Larg	Suit Jacket
7	pr7	dp1	Pants	Large	Khaki dress pants
8	pr8	dp2	Jeans	Large	Acid-wash
9	pr9	dp2	Dress	Medium	Little Black Dress
10	pr10	dp2	Blouse	Small	Casual Pink with fl
11	pr11	dp2	Tights	One-size	Cat print
12	pr12	dp2	Shoes	6 1/2	Studded High Heel
13	pr13	dp2	Long-Sleeve Shirt	Large	Panda pattern
14	pr14	dp2	Shoes	8	Hot Pink Hunter Boo
15	pr15	dp3	Tank-top	Medium	Sequence covered
16	pr16	dp3	Skirt	Large	Hot Pink skater sty
17	pr17	dp3	Shirt	Small	Open back, lace top
18	pr18	dp3	Shirt	Extra-smal	Lace
19	pr19	dp3	Shoes	8 1/2	Nude 6-inch heel
20	pr20	dp3	Shirt	Medium	Polka Dotted, Black
21	pr21	dp3	Pants	Large	Cotten pajamas
22	pr22	dp4	Jeans	Small	Regular Wash
23	pr23	dp4	Shirt	Large	Ice-cream design
24	pr24	dp4	Shirt	Extra-Larg	Disney Car design
25	pr25	dp4	Shoes	4	Light-up, colorful
26	pr26	dp4	Skirt	Small	Pink-tutu
27	pr27	dp4	Backpack	One-Size	Red with superheros
28	pr28	dp4	Shirt	Large	Black
29	pr29	dp5	Jeans	2-4 years	Dark-wash
30	pr30	dp5	Pants	0-2 years	Pink
31	pr31	dp5	Onezie	0-1 years	Its a girl
32	pr32	dp5	Shirt	2-3 years	Daddys little girl
33	pr33	dp5	Dress	3-4 years	Party Dress
34	pr34	dp5	Shoes	5 infant	White Dress Shoes
35	pr35	dp5	Onezie	1-2 years	Its a boy!

Inventory Table

Purpose

This stable shows every product that the store has.

Create Statement

```
CREATE TABLE inventory(
  invid          char(200)  NOT NULL,
  productid      char(200)  NOT NULL REFERENCES products(productid),
  quantity       INTEGER   NOT NULL,
  PRIMARY KEY(invid)
);
```

Functional Dependencies

invid →productid, quantity

Sample Data

See next page for sample…

	invid text	productid text	quantity integer
1	i01	pr1	50
2	i02	pr2	100
3	i03	pr3	45
4	i04	pr4	25
5	i05	pr5	100
6	i06	pr6	200
7	i07	pr7	10
8	i08	pr9	120
9	i09	pr10	60
10	i10	pr11	75
11	i11	pr12	90
12	i12	pr13	100
13	i13	pr14	50
14	i14	pr15	40
15	i15	pr16	20
16	i16	pr17	5
17	i17	pr18	70
18	i18	pr19	110
19	i19	pr20	210
20	i20	pr21	500
21	i21	pr22	80
22	i22	pr23	90
23	i23	pr24	60
24	i24	pr25	45
25	i25	pr26	65
26	i26	pr27	95
27	i27	pr28	20
28	i28	pr29	30
29	i29	pr30	40
30	i30	pr31	60
31	i31	pr32	250
32	i32	pr33	300
33	i33	pr34	45
34	i34	pr34	85
35	i35	pr35	95

Products Supplied Table

Purpose

This table shows every product that is being brought by suppliers to the store.

Create Statement

```
CREATE TABLE productsSupplied(
  supplyid              char(200) NOT NULL REFERENCES suppliers(supplyid),
  invid                 char(200) NOT NULL REFERENCES inventory(invid),
  datePurchased        DATE       NOT NULL,
  quantityPurchased INTEGER   NOT NULL,
  totalPriceUSD        INTEGER   NOT NULL,
  PRIMARY KEY(supplyid, invid)
);
```

Functional Dependencies

supplyid → invid, datePurchased, quantityPurchased, totalPriceUSD

Sample Data

	supplyid text	invid text	datepurchased date	quantitypurchased integer	totalpriceusd integer
1	s01	i22	2014-12-03	100	1000
2	s01	i01	2014-12-03	50	2500
3	s01	i30	2014-12-03	25	750
4	s01	i07	2014-12-25	45	2025
5	s02	i10	2014-12-23	60	2400
6	s02	i02	2014-11-13	15	750
7	s02	i13	2014-01-01	35	2275
8	s02	i32	2014-11-02	90	1350
9	s03	i03	2014-11-03	75	1500
10	s03	i27	2014-12-01	55	2475
11	s03	i11	2014-12-02	80	3600

Suppliers Table

Purpose

This table shows information from every supplier that brings products to each store.

Create Statement

```
CREATE TABLE suppliers(
  supplyid              char(200)          NOT NULL,
  supplierFirstName  VARCHAR(200)  NOT NULL,
  supplierLastName   VARCHAR(200)  NOT NULL,
  supplierPhoneNum VARCHAR(12)    NOT NULL,
  supplierDetails        VARCHAR(200) NOT NULL,
  PRIMARY KEY(supplyid)
);
```

Functional Dependencies

supplyid → supplierFirstName, supplierLastName, supplierPhoneNum, supplierDetails

Sample Data

	supplyid text	supplierfirstname character varying(200)	supplierlastname character varying(200)	supplierphonenum character varying(12)	supplierdetails character varying(200)
1	s01	James	Parker	795-582-1235	Supplies Bottoms
2	s02	Zach	Goat	134-875-2015	Supplies Tops
3	s03	Polar	Express	62-654-8520	Supplies Accessories

Departments Table

Purpose

This table shows each department within each store.

Create Statement

```
CREATE TABLE departments(
  departid       char(200)          NOT NULL,
  departName  VARCHAR(200) NOT NULL,
  PRIMARY KEY(departid)
);
```

Functional Dependencies

departid → departName

Sample Data

	departid text	departname character varying(200)
1	dp1	Mens
2	dp2	Womens
3	dp3	Juniors
4	dp4	Childrens
5	dp5	Infants

Department Stores Table

Purpose

This table shows each department store in each location.

Create Statement

```
CREATE TABLE departmentStores(
  departStoreid       char(200)          NOT NULL,
  departStoreName  VARCHAR(200)  NOT NULL,
  departTown          VARCHAR(200)  NOT NULL,
  departState          VARCHAR(2)     NOT NULL,
  departPhoneNum  VARCHAR(12)    NOT NULL,
  departHours         VARCHAR(200) NOT NULL,
  PRIMARY KEY(departStoreid)
);
```

Functional Dependencies

departStoreid →departStoreName, departTown, departState, departPhone, departHours

Sample Data

	departstoreid text	departstorename character varying(200)	departtown character varying(200)	departstate character varying(2)	departphone character varying(12)	departhours character varying(200)
1	d01	Nordstrom	Palm Beach Gardens	FL	135-495-8542	8:00 AM - 9:00 PM
2	d02	Nordstrom	King of Prussia	PA	540-123-4564	8:00 AM - 9:00 PM
3	d03	Nordstrom	Boston	MA	123-456-7894	8:00 AM - 9:00 PM
4	d04	Nordstrom	New York City	NY	456-452-1254	8:00 AM - 9:00 PM

Views

Employee Location

Purpose

This is a list of the total staff, what their job is and where they are working within the store

Create Statement

```
CREATE VIEW EmployeeLocation as
SELECT distinct empA.eid AS "Employee ID" , depart.departName AS "Department Name",
job.jobName AS "Job Title", job.JobDescrip AS "Job Description",
departStore.departStoreName AS "Store Name", departStore.departState AS "Store
Location"
FROM employeeAssingments empA, departments depart, jobTitles job, departmentStores
departStore
WHERE empA.departid=depart.departid
    AND empA.jobTitleid=job.jobTitleid
    AND depart.departStoreid = departStore.departStoreid
```

Sample Data

	Employee ID character(200)	Department Name character varying(200)	Job Title character varying(200)	Job Description character varying(200)	Store Name character varying(200)	Store Location character varying(2)
1	e02	Juniors	Phone Worker	Responsible for answering the phone and any questions asked	Nordstrom	FL
2	e07	Childrens	Custodial	Responsible for cleaning the store after it has closed	Nordstrom	NY
3	e03	Childrens	Manager	Responsible for all workers to be doing there job	Nordstrom	NY
4	e09	Childrens	Manager	Responsible for all workers to be doing there job	Nordstrom	FL
5	e05	Infants	Tech	Responsible for keeping our website current	Nordstrom	PA
6	e08	Womens	Sales	Responsible for selling products on the floor of the store	Nordstrom	MA
7	e06	Womens	Phone Worker	Responsible for answering the phone and any questions asked	Nordstrom	FL
8	e01	Mens	Sales	Responsible for selling products on the floor of the store	Nordstrom	FL
9	e04	Womens	Custodial	Responsible for cleaning the store after it has closed	Nordstrom	PA

Total inventory of items

Purpose

This is to show the total inventory of the store after the suppliers are at the store.

Create Statement

CREATE VIEW InventoryView as

SELECT prod.productName AS "Product Name", prod.productid AS "Product ID", inventory.invid AS "Inventory ID", inventory.quantity AS "Inventory Quantity"

FROM products prod, inventory, productsSupplied prodSup

WHERE prod.productid=inventory.productid

AND inventory.invid=prodSup.invid

Sample Data

	Product Name character varying(200)	Product ID character(200)	Inventory ID character(200)	Inventory Quantity integer
1	Button Down	pr02	i02	100
2	Socks	pr03	i03	45
3	Pants	pr07	i07	10
4	Tights	pr11	i10	75
5	Shoes	pr12	i11	90
6	Shoes	pr14	i13	50
7	Shirt	pr28	i27	20
8	Dress	pr33	i32	300

Reports

Product Location

Purpose

Displays all products and the departments they are in.

Create Statement

SELECT prod.productName AS "Department", prod.productDescrip AS "Product Description", depart.departName AS "Department Name"

FROM products prod, departments depart

WHERE prod.departid=depart.departid

ORDER BY departName ASC;

Sample Data

	Department character varying(200)	Product Description character varying(200)	Department Name character varying(200)
1	Shirt	Black	Childrens
2	Jeans	Regular Wash	Childrens
3	Shirt	Ice-cream design	Childrens
4	Shirt	Disney Car design	Childrens
5	Shoes	Light-up, colorful	Childrens
6	Skirt	Pink-tutu	Childrens
7	Backpack	Red with superheros	Childrens
8	Onezie	Its a boy!	Infants
9	Pants	Pink	Infants
10	Onezie	Its a girl	Infants
11	Shirt	Daddys little girl	Infants
12	Dress	Party Dress	Infants
13	Shoes	White Dress Shoes	Infants
14	Jeans	Dark-wash	Infants
15	Shirt	Lace	Juniors
16	Pants	Cotten pajamas	Juniors
17	Shirt	Polka Dotted, Black	Juniors
18	Shoes	Nude 6-inch heel	Juniors
19	Shirt	Open back, lace top	Juniors
20	Skirt	Hot Pink skater sty	Juniors
21	Tank-top	Sequence covered	Juniors
22	Pants	Khaki dress pants	Mens
23	Jeans	Light-wash	Mens
24	Button Down	Plaid	Mens
25	Socks	Colorful pattern	Mens
26	T-shirt	California text on	Mens
27	Shoes	Winter Boots	Mens
28	Jacket	Suit Jacket	Mens
29	Shoes	Hot Pink Hunter Boo	Womens
30	Jeans	Acid-wash	Womens
31	Dress	Little Black Dress	Womens
32	Blouse	Casual Pink with fl	Womens
33	Tights	Cat print	Womens
34	Shoes	Studded High Heel	Womens
35	Long-Sleeve Shirt	Panda pattern	Womens

Best Department

Purpose

Determine which department has the most orders.

Create Statement

```
SELECT depart.departid, depart.DepartName AS "Department Name"
FROM productOrders prOr, products prod, departments depart
WHERE prOr.productid=prod.productid
 AND prod.departid=depart.departid
        GROUP BY depart.departid
        ORDER BY count(departName) DESC
 LIMIT 2
```

	departid character(200)	departname character varying(200)
1	dp07	Womens
2	dp03	Juniors

Stored Procedures

Purpose

Determine the price of each product supplied by the quantity and total price of entire purchase of goods.

Create Statement

```
CREATE OR REPLACE FUNCTION updateInventories(integer) returns integer as
$$
DECLARE
        supplyid        char(4) :=$1;
        quantityPurchased integer :=$2;
        totalPrice        integer :=$3;
BEGIN
   IF supplyid is NULL or supplyid<0
   THEN
        return 0;
   END IF;

   SELECT
        totalPrice = prodSup.quantityPurchased/prodSup.totalPrice;
end;
$$
language plpgsql;
```

Triggers

Update Inventory

Purpose

This trigger will allow an for the database to be updated when there are new products supplied. It will place the new items in the inventory. Once the updateInvetories stored procedure is being used that means the quantity of the item is being changed meaning the table needs to be updated as well.

Create Statement

CREATE OR REPLACE FUNCTION updateInventorys

AFTER UPDATE ON inventory

FOR EACH ROW EXECUTE updateInventories

Security

This database will have two types of users. One will be an admin and another will be a manager.

1. The admin can change anything within the database and maintain any needs of the database.

```
CREATE ROLE administration
GRANT SELECT,INSERT,UPDATE,ALTER
ON ALL TABLES IN SCHEMA PUBLIC
TO administration
```

2. The manager can update and perform queries for products and employee assignments within the database.

```
CREATE ROLE manager
GRANT SELECT,UPDATE
ON employeeAssingments, inventory
TO manager
```

Implementation Notes

- Implementation was basically straight forward and no issues really came up. Besides typical spelling errors that I wasn't able to find for 30+ minutes and during those 30+ minutes it made me seriously question my programming skills but then I came to realize that I should go back to second grade to learn how to spell simple words.

Known Problems

- It is currently not possible to add smaller products meaning that there are no names for products only whatever you want to add. This means that querying could be messed up.
- The rest of the database seems to be working fine with no problems
- My stored procedure and triggers are most likely incorrect. I have to get some help because I really had a hard time figuring out what to do.

Future Enhancements

- I would like to add an online version to monitor only orders and shipping. This would allow our business to grow.
- I would like to add a store hour feature so that the hours can change per day and holiday season. It would make it more personal to the customers.
- I would like to add a user login so there could be online shopping and to keep track of one customers past orders.
- I would like to add a way to also track shipments online for ease of customer worry.
- I would like to make an employment system that allows people to clock-in and clock-out of work.