
Fantasy Glassworks Database System

Karen Prieto

INFO 605: Intro to Database Management

Instructor: Stephen Frein

Term Project

08/25/11

I certify that:

- This paper/project/exam is entirely my own work.
- I have not quoted the words of any other person from a printed source or a website without indicating what has been quoted and providing an appropriate citation.
- I have not submitted this paper / project to satisfy the requirements of any other course.

Signature: Karen M. Staples Prieto, Student #11735761

Date: 08/25/11 _____

Table of Contents

1.0 Overview	5
2.0 Requirements	5
2.1 Products	5
2.2 Show Schedule	6
2.3 Orders	6
2.4 Business Questions.....	7
2.5 Assumptions	7
3.0 ER Model.....	8
4.0 Relational Schema	9
5.0 Data Dictionary.....	10
6.0 DDL.....	17
6.1 CraftShow	17
6.2 Payment	17
6.3 Customer	18
6.4 Orders	19
6.5 Shipment	20
6.6 Image.....	21
6.7 Box	21
6.8 Pattern	22
6.9 Glass	23
6.10 Product.....	24
6.11 CreatedWith.....	25
6.12 Sequences & Triggers.....	25
6.13 Views	26
7.0 DML	26
7.1 CraftShow	26
7.1.1 Inserts.....	26
7.1.2 Updates/Deletes	27
7.2 Payment	27
7.2.1 Inserts.....	27
7.2.2 Updates/Deletes	27
7.3 Customer	28
7.3.1 Inserts.....	28
7.3.2 Updates/Deletes	28
7.4 Orders	28
7.4.1 Inserts.....	28
7.4.2 Updates/Deletes	28

7.5 Shipment	28
7.5.1 Inserts.....	28
7.5.2 Updates/Deletes	29
7.6 Image.....	29
7.6.1 Inserts.....	29
7.6.2 Updates/Deletes	29
7.7 Box	29
7.7.1 Inserts.....	29
7.7.2 Updates/Deletes	30
7.8 Pattern	30
7.8.1 Inserts.....	30
7.8.2 Updates/Deletes	31
7.9 Glass	31
7.9.1 Inserts.....	31
7.9.2 Updates/Deletes	32
7.10 Product.....	32
7.10.1 Inserts.....	32
7.10.2 Updates/Deletes	32
7.11 CreatedWith.....	32
7.11.1 Inserts.....	32
7.11.2 Updates/Deletes	34
8.0 Queries	34
8.1 Products	34
8.1.1 Populate Product Page	34
8.1.2 Site Searches	36
8.1.3 Product Catalog	37
8.1.4 Pattern/Glass Usage 1	37
8.1.5 Pattern/Glass Usage 2	38
8.2 Craft Shows.....	39
8.3 Orders	39
8.3.1 Subtotal.....	39
8.3.2 Tax	40
8.3.3 Handling	40
8.3.4 Shipping	40
8.4 Fullfillment	41
8.4.1 Pending Shipments.....	41
8.4.2 Pending Pickups	46
8.5 Business Reporting.....	46
8.5.1 Quarterly	46
8.5.2 Annual.....	47
9.0 Lessons Learned	47
10.0 Bibliography	48

1.0 Overview

This document describes business requirements, design, and resulting implementation for the Fantasy Glassworks database system.

2.0 Requirements

A database system is needed to support e-commerce for *Fantasy Glassworks*, a website that showcases stained glass artwork created by an independent crafter in New Jersey¹. Currently this website is plain HTML. Sales inquiries are handled via email and phone.

A database system will allow sales transactions to be conducted online. In addition, a database will simplify site maintenance by allowing product information to be displayed dynamically. And it will retain a history of crafting supplies used for planning purposes.

Major items to be recorded in the database are products, a craft show schedule, and orders.

2.1 Products

Due to the nature of stained glass, each product displayed on the website is unique, although many will be based on the same pattern. When a customer orders a product, s/he is ordering the exact item depicted on the page.

Each product page lists several characteristics (which are searchable). These include:

- Description (text paragraph)
- Theme (floral, animal, other)
- Pattern Type (traditional, layered, 3D)
- Primary Color (red, pink, orange, yellow, blue, green, aqua, purple, multi, black, white, clear)
- Height (inches)
- Width (inches)
- Weight (ounces)
- Thumbnail Image (JPG, GIF, PNG)
- Full-size Image (JPG, GIF, PNG)
- Price
- Packing cost (This depends on the type of box associated with the assembled pattern.)
- Status (Available, Sold, Shipped).

Pages for certain products display one or more additional characteristics:

- Secondary Color (red, pink, orange, yellow, blue, green, aqua, purple, black, white, clear)
- Depth (inches). This should be required for 3D pattern types.
- Alternate Thumbnail (JPG, GIF, PNG)
- Alternate Full-size (JPG, GIF, PNG).

¹ The site is not currently active, but a zip'd copy is available.

Other characteristics are tracked, but not displayed on-screen:

- Pattern (vendor, part#)
- Glass type (vendor, part#, type (translucent/opal/wispy), texture (yes/no), coating (yes/no))
- Box (height, width, length, weight (with packing materials), cost (of box and packing materials)).

This information is used to determine packing & shipping costs, and for order shipping prep. Note that some items are small enough to occupy less than a full box – for example, the small box accommodates four butterflies.

Updates to product data will be entered via non-Internet-facing HTML forms by the proprietor of Fantasy Glassworks. Access to this administration feature will be controlled by Windows / DBMS security on the proprietor's private computer.

The proprietor would like the ability to retrieve an on-demand list of all products currently for sale on the site, including theme, pattern type, color, price, packing type. This list will help her decide what new products to create. She would also like to search for all products with certain pattern or glass characteristics, since she often refers to existing products for information while creating new ones.

2.2 Show Schedule

On the website, a Show Schedule page lists all upcoming craft shows at which Fantasy Glassworks will be present. This information is also displayed as a single-select list on the Orders page when a customer chooses "Pick Up" in lieu of shipping. Each show includes:

- Date
- Name
- Location
- Webpage (optional).

Updates to show data will be handled in the same way as product updates.

2.3 Orders

Orders involve least one product and one U.S. customer. (Non-U.S. customers are advised to call/email for information, and are beyond the scope of the website upgrade.)

Customers wishing to order must enter a name, email address, billing address (street, apt#, city, state, zip), phone number (domestic), and payment information (PayPal account # or credit card number, exp date, and PIN). The customer must also specify a shipping address, and may optionally specify a gift message, which will be printed on the packing slip.

The product subtotal is calculated and displayed based on the products ordered. If the shipping address is within New Jersey, state tax is added. The shipping & handling total is also calculated and displayed: shipping per box is determined through an interface with USPS.com, based on destination zip code, box weight, and box dimensions. Handling is a sum of the cost of each box used for the order.

Customers may optionally designate the purchase as "Pick Up," in which case, ~~the packing and~~ shipping costs are removed from the final total, New Jersey state tax is applied (if not already applied), and the customer is prompted to choose a location from the Show Schedule.

When payment is processed (through an online payment processing vendor), a unique order number is displayed on the page, and sent to the customer in a confirmation email along with order details.

All order data, except payment information, are retained for internal reporting purposes. Payment information is not retained beyond payment processing. Note that customer accounts (i.e. passwords) are beyond the scope of this upgrade.

At the end of each day, the proprietor will need a "to be shipped" report of all non-pickup sales, grouped by order, with shipping details and with products listed by box. And before each craft show, the proprietor would like a list of all "pickup" orders scheduled for the show, including all product and customer details.

At the end of each tax quarter, the proprietor will need a list of all in-state sales within the quarter: Order number, subtotal, shipping cost, tax, and total, with overall totals for the quarter. At the end of the year a similar report is needed for all sales within the year.

2.4 Business Questions

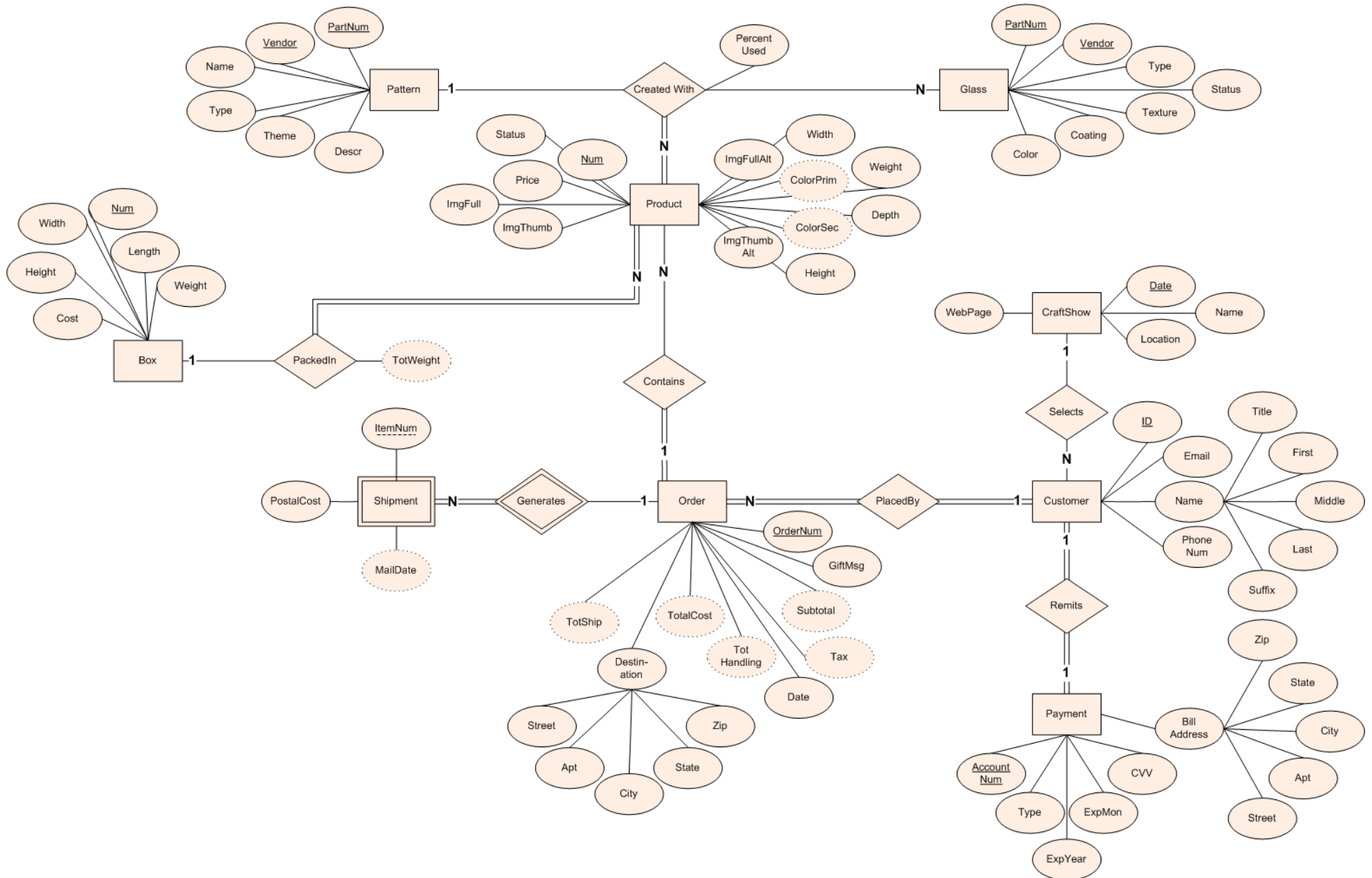
Minimally, the database needs to answer these questions:

1. What products are available for sale?
2. Which products currently for sale have certain characteristics?
3. Where can I (customer) examine or pick up products?
4. How much does an order cost?
5. What products must be shipped tomorrow?
6. Who is picking up which products at the next show?
7. What are sales details for last quarter or year?

2.5 Assumptions

- Came/leading, the other component of stained glass, is stipulated by each pattern, but neither the proprietor nor customers need to look up the came/leading used for each product.
- In cases where multiple products created with a particular pattern can fit into a single box, only one box type will be associated with the product. In other words, the system will not dynamically "upgrade" to a bigger box if many items of the same pattern are ordered. A moderate level of shipping efficiency is acceptable.
- There is no need to reconcile customers across orders. If the same customer with the same information submits more than one orders, data for the customer will be duplicated. Repeat sales do occur, but not frequently enough to merit the overhead of issuing and managing passwords needed to associate returning customers with existing data.

3.0 ER Model



4.0 Relational Schema

Mapped translation (Frein, Week3, Slide 38).



5.0 Data Dictionary

CraftShow: Scheduled craft show.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
Date	Date of craft show.	DATE	>= 09/01/2011	Yes	Yes	No
Name	Craft show name.	VARCHAR2(40)	All	Yes	No	No
Location	Craft show location.	VARCHAR2(40)	All	Yes	No	No
WebPage	Craft show URL.	VARCHAR2(60)	www.*.com/net/etc ²	No	No	No

Payment: Payment remitted by customer.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
AccountNum	Credit card account number.	CHAR(19)	####-####-####-#### ²	Yes	Yes	No
Type	Credit card type.	VARCHAR2(10)	'Visa', 'Mastercard', 'Discover'	Yes	No	No
ExpMon	Month of credit card expiration date.	CHAR(3)	'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'	Yes	No	No
ExpYear	Credit card expiration date.	NUMBER(4)	BETWEEN 2011 AND 2050	Yes	No	No
CVN	Credit card verification number.	NUMBER(3)	All	Yes	No	No
BillAddressStreet	Billing street address.	VARCHAR2(40)	All	Yes	No	No
BillAddressApt	Billing apartment number.	CHAR(3)	All	No	No	No
BillAddressCity	Billing city.	VARCHAR2(20)	All	Yes	No	No

² Due to technical difficulties with CHECK (REGEXP_LIKE, this domain constraint was not implemented (ORA-00920).

Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
BillAddressState	Billing state.	CHAR(2)	'AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'	Yes	No	No
BillAddressZip	Billing zip code.	CHAR(10)	#####-#### ²	Yes	No	No

Customer: Person who places an order.

Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
ID	Unique customer ID.	NUMBER	All	Yes	Yes	No
Email	Customer email address.	VARCHAR2(30)	*@.xxx.* ²	Yes	No	No
PhoneNum	Customer phone number.	CHAR(12)	###-###-#### ²	Yes	No	No
NameTitle	Customer title.	VARCHAR2(10)	All	No	No	No
NameFirst	Customer first name.	VARCHAR2(20)	All	Yes	No	No
NameMiddle	Customer middle name.	VARCHAR2(20)	All	No	No	No
NameLast	Customer last name.	VARCHAR2(30)	All	Yes	No	No
NameSuffix	Customer suffix.	VARCHAR2(5)	All	No	No	No
PmtAccountNum	Account number of remitted payment.	CHAR(19)	#####-#####-#####-##### ²	Yes	No	Yes
ShowDate	Date of selected craft show.	DATE	>= 09/01/2011	No	No	Yes

Order: Order placed by customer.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
Num	Unique order number.	NUMBER	All	Yes	Yes	No
Date	Date of order submission.	DATE	>= 09/01/2011	Yes	No	No
DestStreet	Destination street.	VARCHAR2(40)	All	No	No	No
DestApt	Destination apt.	CHAR(3)	All	No	No	No
DestCity	Destination city.	VARCHAR2(20)	All	No	No	No
DestState	Destination state.	CHAR(2)	'AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'	No	No	No
DestZip	Destination zip code.	CHAR(10)	#####-#### ²	No	No	No
GiftMsg	Optional text for packing slip.	VARCHAR2(60)	All	No	No	No
CustomerID	Unique ID of ordering customer.	NUMBER	All	Yes	No	Yes

Shipment: Shipment generated by an order.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
OrderNum	Order that generated the shipment.	NUMBER	All	Yes	Yes	Yes
ItemNum	Item number within containing shipment.	NUMBER(2)	All	Yes	Yes	No
PostalCost	Output from usps.com.	NUMBER(4,2)	All	No	No	No

Product: Stained glass artwork sold by Fantasy Glassworks.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
Num	Unique product number.	NUMBER	All	Yes	Yes	No
Price	Selling price.	NUMBER(5,2)	All	Yes	No	No
Status	Sales status.	VARCHAR2(9)	'Available', 'Sold', 'Shipped', 'Shrink'	Yes	No	No
Height	Height in inches.	NUMBER(3,1)	<= 72.0	Yes	No	No
Width	Width in inches.	NUMBER(3,1)	<= 72.0	Yes	No	No
Depth	Depth in inches.	NUMBER(3,1)	<= 72.0	No	No	No
Weight	Weight in ounces.	NUMBER(4,1)	<= 320.0	Yes	No	No
ImgFull	Filename of full-sized image.	VARCHAR2(24)	*.JPG, *.GIF, *.PNG	Yes	No	Yes
BoxNum	Number of associated box	NUMBER	All	Yes	No	Yes
OrderNum	Order that contains the product.	NUMBER	All	No	No	Yes

Image: Images of each product.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
ImgFull	Filename of full-sized image.	VARCHAR2(24)	*.JPG, *.GIF, *.PNG	Yes	Yes	No
ImgThumb	Filename of thumbnail image.	VARCHAR2(24)	*.JPG, *.GIF, *.PNG	Yes	No	No
ImgFullAlt	Filename of full-sized supplemental image.	VARCHAR2(24)	*.JPG, *.GIF, *.PNG	No	No	No
ImgThumbAlt	Filename of thumbnail supplemental image.	VARCHAR2(24)	*.JPG, *.GIF, *.PNG	No	No	No

Box: Box in which product is packed.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
Num	Unique box number.	NUMBER	All	Yes	Yes	No
Height	Box height in inches.	NUMBER(3,1)	All	Yes	No	No
Width	Box width in inches.	NUMBER(3,1)	All	Yes	No	No
Length	Box length in inches.	NUMBER(3,1)	All	Yes	No	No
Weight	Box weight in ounces.	NUMBER(4,1)	All	Yes	No	No
Cost	Cost of box and packing materials, plus approximate value of time involved in packing.	NUMBER(4,2)	All	Yes	No	No

CreatedWith: Relationship between pattern, glass, and product.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
PatternVendor	Vendor of associated pattern.	VARCHAR2(40)	All	Yes	No	Yes
PatternPartNum	Part # of associated pattern.	VARCHAR2(20)	All	Yes	No	Yes
GlassVendor	Vendor of associated glass.	VARCHAR2(40)	All	Yes	Yes	Yes
GlassPartNum	Part # of associated glass.	VARCHAR2(20)	All	Yes	Yes	Yes
PercentUsed	Amount of glass used in the associated product.	NUMBER(3)	BETWEEN 0 AND 100	Yes	No	No
ProdNum	Unique product number.	NUMBER	All	Yes	Yes	Yes

Pattern: Pattern used to create products.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
Vendor	Pattern vendor.	VARCHAR2(40)	All	Yes	Yes	No
PartNum	Pattern part number.	VARCHAR2(20)	All	Yes	Yes	No
Type	Pattern type.	VARCHAR2(11)	'Traditional', 'Layered', '3d'	Yes	No	No
Theme	Pattern theme.	VARCHAR2(6)	'Floral', 'Animal', 'Other'	Yes	No	No
Name	Short description of pattern.	VARCHAR2(20)	All	Yes	No	No
Descr	Descriptive paragraph about assembled pattern.	VARCHAR2(300)	All	Yes	No	No

Glass: Glass used to create products.						
Attribute Name	Attribute Description	Data Type	Domain	NotNull?	PK	FK
Vendor	Vendor of associated glass.	VARCHAR2(40)	All	Yes	Yes	No
PartNum	Part # of associated glass.	VARCHAR2(20)	All	Yes	Yes	No
Type	Glass type.	VARCHAR2(11)	'Translucent', 'Opal', 'Wispy'	Yes	No	No
Texture	Whether or not glass is textured.	CHAR(1)	'Y','N'	Yes	No	No
Coating	Whether or not glass is coated.	CHAR(1)	'Y','N'	Yes	No	No
Color	Glass color.	VARCHAR2(6)	'Red', 'Pink', 'Orange', 'Yellow', 'Blue', 'Green', 'Aqua', 'Purple', 'Black', 'White', 'Cream', 'Brown', 'Multi', 'Clear'	Yes	No	No
Status	Glass availability.	VARCHAR2(9)	'Available', 'Reorder', 'Legacy'	Yes	No	No

6.0 DDL

6.1 CraftShow

```
CREATE TABLE CraftShow (
    ShowDate DATE CONSTRAINT CraftShow_ck_ShowDate CHECK (ShowDate >=
to_date('09/01/2011','mm/dd/yyyy')) CONSTRAINT CraftShow_NN_Date NOT NULL
CONSTRAINT CraftShow_PK PRIMARY KEY,
    Name VARCHAR2(40) CONSTRAINT CraftShow_NN_Name NOT NULL,
    Location VARCHAR2(40) CONSTRAINT CraftShow_NN_Location NOT NULL,
    WebPage VARCHAR2(60) CONSTRAINT CraftShow_ck_WebPage CHECK (REGEXP_LIKE
(WebPage, '(https?:\|/|ww[[:alnum:]]\.)[\w\_-]+(\.[\w\_-]+)+)' )2
);
```

```
DESCRIBE CraftShow;
```

Name	Null?	Type
SHOWDATE	NOT NULL	DATE
NAME	NOT NULL	VARCHAR2(40)
LOCATION	NOT NULL	VARCHAR2(40)
WEBPAGE		VARCHAR2(60)

6.2 Payment

```
CREATE TABLE Payment (
    AccountNum CHAR(19) CONSTRAINT Payment_NN_AccountNum NOT NULL CONSTRAINT
Payment_PK PRIMARY KEY CONSTRAINT Payment_CK_AccountNum CHECK (REGEXP_LIKE
(AccountNum, '^\d{4}-\d{4}-\d{4}-\d{4}$'))2,
    Type VARCHAR2(10) CONSTRAINT Payment_CK_Type CHECK (Type IN ('Visa',
'Mastercard', 'Discover')) CONSTRAINT Payment_NN_Type NOT NULL,
    ExpMon CHAR(3) CONSTRAINT Payment_CK_ExpMon CHECK (ExpMon IN ('Jan', 'Feb',
'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'))
CONSTRAINT Payment_NN_ExpMon NOT NULL,
    ExpYear NUMBER(4) CONSTRAINT Payment_CK_ExpYear CHECK (ExpYear BETWEEN 2011 AND
2050) CONSTRAINT Payment_NN_ExpYear NOT NULL,
    CVN NUMBER(3) CONSTRAINT Payment_NN_CVN NOT NULL,
    BillAddressStreet VARCHAR2(40) CONSTRAINT Payment_NN_BillAddressStreet NOT NULL,
    BillAddressApt CHAR(3),
    BillAddressCity VARCHAR2(20) CONSTRAINT Payment_NN_BillAddressCity NOT NULL,
    BillAddressState CHAR(2) CONSTRAINT Payment_CK_BillAddressState CHECK
(BillAddressState IN ('AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL',
'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI',
'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH',
```

```
'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV',
'WI', 'WY')) CONSTRAINT Payment_NN_BillAddressState NOT NULL,
BillAddressZip CHAR(10) CONSTRAINT Payment_NN_BillAddressZip NOT NULL CONSTRAINT
Payment_CK_BillAddressZip CHECK (REGEXP_LIKE (BillAddressZip, ' ^\d{5}-\d{4}$'))
2
```

);

```
DESCRIBE Payment;
```

Name	Null?	Type
ACCOUNTNUM	NOT NULL	CHAR(19)
TYPE	NOT NULL	VARCHAR2(10)
EXPMON	NOT NULL	CHAR(3)
EXPYEAR	NOT NULL	NUMBER(4)
CVN	NOT NULL	NUMBER(3)
BILLADDRESSSTREET	NOT NULL	VARCHAR2(40)
BILLADDRESSAPT		CHAR(3)
BILLADDRESSCITY	NOT NULL	VARCHAR2(20)
BILLADDRESSSTATE	NOT NULL	CHAR(2)
BILLADDRESSZIP	NOT NULL	CHAR(10)

6.3 Customer

```
CREATE TABLE Customer (
    ID NUMBER CONSTRAINT Customer_NN_ID NOT NULL CONSTRAINT Customer_PK PRIMARY KEY,
    Email VARCHAR2(30) CONSTRAINT Customer_NN_Email NOT NULL CONSTRAINT
    Customer_CK_Email CHECK (REGEXP_LIKE (Email, '^w+@w+(\.w+)+$') )2,
    PhoneNum CHAR(12) CONSTRAINT Customer_NN_PhoneNum NOT NULL CONSTRAINT
    Customer_CK_PhoneNum CHECK (REGEXP_LIKE (PhoneNum, '^d{3}- d{3}-d{4}$') )2,
    NameTitle VARCHAR2(10),
    NameFirst VARCHAR2(20) CONSTRAINT Customer_NN_NameFirst NOT NULL,
    NameMiddle VARCHAR2(20),
    NameLast VARCHAR2(30) CONSTRAINT Customer_NN_NameLast NOT NULL,
    NameSuffix VARCHAR2(5),
    PmtAccountNum CHAR(19) CONSTRAINT Customer_NN_PmtAccountNum NOT NULL CONSTRAINT
    Customer_fk_PmtAccountNum REFERENCES Payment(AccountNum) CONSTRAINT
    Customer_CK_PmtAccountNum CHECK (REGEXP_LIKE (PmtAccountNum, '^d{4}-d{4}-
    \d{4}-\d{4}$') )2,
    ShowDate DATE CONSTRAINT Customer_fk_ShowDate REFERENCES CraftShow(ShowDate)
);
```

```
DESCRIBE Customer;
```

Name	Null?	Type
ID	NOT NULL	NUMBER
EMAIL	NOT NULL	VARCHAR2(30)
PHONENUM	NOT NULL	CHAR(12)
NAMETITLE		VARCHAR2(10)
NAMEFIRST	NOT NULL	VARCHAR2(20)
NAMEMIDDLE		VARCHAR2(20)
NAMELAST	NOT NULL	VARCHAR2(30)
NAMESUFFIX		VARCHAR2(5)
PMTACCOUNTNUM	NOT NULL	CHAR(19)
SHOWDATE		DATE

6.4 Orders

```
CREATE TABLE Orders (
    Num NUMBER CONSTRAINT Orders_NN_Num NOT NULL CONSTRAINT Orders_PK PRIMARY KEY,
    OrderDate DATE DEFAULT SYSDATE CONSTRAINT Orders_CK_OrderDate CHECK (OrderDate
    >= to_date('09/01/2011', 'mm/dd/yyyy')) CONSTRAINT Orders_NN_OrderDate NOT NULL,
    DestStreet VARCHAR2(40),
    DestApt CHAR(3),
    DestCity VARCHAR2(20),
    DestState CHAR(2) CONSTRAINT Orders_CK_DestState CHECK (DestState IN ('AL',
    'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN',
    'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE',
    'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC',
    'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY')),
    DestZip CHAR(10) CONSTRAINT Orders_CK_DestZip CHECK (REGEXP_LIKE (DestZip,
    '^d{5}- d{4}$'))2,
    GiftMsg VARCHAR2(60),
    CustomerID NUMBER CONSTRAINT Orders_NN_CustomerID NOT NULL CONSTRAINT
    Orders_FK_CustomerID REFERENCES Customer(ID)
);
```

DESCRIBE Orders;

Name	Null?	Type
NUM	NOT NULL	NUMBER
ORDERDATE	NOT NULL	DATE
DESTSTREET		VARCHAR2(40)
DESTAPT		CHAR(3)
DESTCITY		VARCHAR2(20)
DESTSTATE		CHAR(2)
DESTZIP		CHAR(10)
GIFTMSG		VARCHAR2(60)
CUSTOMERID	NOT NULL	NUMBER

6.5 Shipment

```
CREATE TABLE Shipment (  
    OrderNum NUMBER CONSTRAINT Shipment_NN_OrderNum NOT NULL CONSTRAINT  
    Shipment_FK_OrderNum REFERENCES Orders(Num) ON DELETE CASCADE,  
    ItemNum NUMBER(2) CONSTRAINT Shipment_NN_ItemNum NOT NULL,  
    PostalCost NUMBER(4,2),  
    CONSTRAINT Shipment_PK PRIMARY KEY(OrderNum,ItemNum)  
);
```

DESCRIBE Shipment;

Name	Null?	Type
ORDERNUM	NOT NULL	NUMBER
ITEMNUM	NOT NULL	NUMBER(2)
POSTALCOST		NUMBER(4,2)

6.6 Image

This entity is not represented in the ER Model, but was added in order to achieve 3NF, since ImgThumb, a non-key, (as well as ImgFullAlt and ImgThumbAlt, where they exist) is unique to each product.

```
CREATE TABLE Image (
    ImgFull VARCHAR2(24) DEFAULT 'TBS.gif' CONSTRAINT Image_CK_ImgFull CHECK
    (ImgFull LIKE '%.jpg' OR ImgFull LIKE '%.gif' OR ImgFull LIKE '%.png')
    CONSTRAINT Image_NN_ImgFull NOT NULL CONSTRAINT Image_PK PRIMARY KEY,
    ImgThumb VARCHAR2(24) DEFAULT 'TBSsm.gif' CONSTRAINT Image_CK_ImgThumb CHECK
    (ImgThumb LIKE '%.jpg' OR ImgThumb LIKE '%.gif' OR ImgThumb LIKE '%.png')
    CONSTRAINT Image_NN_ImgThumb NOT NULL CONSTRAINT Payment_uq_Name UNIQUE,
    ImgFullAlt VARCHAR2(24) CONSTRAINT Image_CK_ImgFullAlt CHECK (ImgFullAlt LIKE
    '%.jpg' OR ImgFullAlt LIKE '%.gif' OR ImgFullAlt LIKE '%.png'),
    ImgThumbAlt VARCHAR2(24) CONSTRAINT Image_CK_ImgThumbAlt CHECK (ImgThumbAlt
    LIKE '%.jpg' OR ImgThumbAlt LIKE '%.gif' OR ImgThumbAlt LIKE '%.png')
);
```

```
DESCRIBE Image;
```

Name	Null?	Type
IMGFULL	NOT NULL	VARCHAR2(24)
IMGTHUMB	NOT NULL	VARCHAR2(24)
IMGFULLALT		VARCHAR2(24)
IMGTHUMBALT		VARCHAR2(24)

6.7 Box

```
CREATE TABLE Box (
    Num NUMBER CONSTRAINT Box_NN_Num NOT NULL CONSTRAINT Box_PK PRIMARY KEY,
    Height NUMBER(3,1) CONSTRAINT Box_NN_Height NOT NULL,
    Width NUMBER(3,1) CONSTRAINT Box_NN_Width NOT NULL,
    Length NUMBER(3,1) CONSTRAINT Box_NN_Length NOT NULL,
    Weight NUMBER(4,1) CONSTRAINT Box_NN_Weight NOT NULL,
    Cost NUMBER(4,2) CONSTRAINT Box_NN_Cost NOT NULL
);
```

DESCRIBE Box;

Name	Null?	Type
NUM	NOT NULL	NUMBER
HEIGHT	NOT NULL	NUMBER(3,1)
WIDTH	NOT NULL	NUMBER(3,1)
LENGTH	NOT NULL	NUMBER(3,1)
WEIGHT	NOT NULL	NUMBER(4,1)
COST	NOT NULL	NUMBER(4,2)

6.8 Pattern

```
CREATE TABLE Pattern (
    Vendor VARCHAR2(40) CONSTRAINT Pattern_NN_Vendor NOT NULL,
    PartNum VARCHAR2(20) CONSTRAINT Pattern_NN_PartNum NOT NULL,
    Type VARCHAR2(11) CONSTRAINT Pattern_CK_Type CHECK (Type IN ('Traditional',
    'Layered', '3d')) CONSTRAINT Pattern_NN_Type NOT NULL,
    Theme VARCHAR2(6) CONSTRAINT Pattern_CK_Theme CHECK (Theme IN ('Floral',
    'Animal', 'Other')) CONSTRAINT Pattern_NN_Theme NOT NULL,
    Name VARCHAR2(20) CONSTRAINT Pattern_NN_Name NOT NULL,
    Descr VARCHAR2(300) CONSTRAINT Pattern_NN_Descr NOT NULL,
    CONSTRAINT Pattern_PK PRIMARY KEY (Vendor,PartNum)
);
```

DESCRIBE Pattern;

Name	Null?	Type
VENDOR	NOT NULL	VARCHAR2(40)
PARTNUM	NOT NULL	VARCHAR2(20)
TYPE	NOT NULL	VARCHAR2(11)
THEME	NOT NULL	VARCHAR2(6)
NAME	NOT NULL	VARCHAR2(20)
DESCR	NOT NULL	VARCHAR2(300)

6.9 Glass

```
CREATE TABLE Glass (  
    Vendor VARCHAR2(40) CONSTRAINT Glass_NN_Vendor NOT NULL,  
    PartNum VARCHAR2(20) CONSTRAINT Glass_NN_PartNum NOT NULL,  
    Type VARCHAR2(11) CONSTRAINT Glass_CK_Type CHECK (Type IN ('Translucent',  
    'Opal', 'Wispy')) CONSTRAINT Glass_NN_Type NOT NULL,  
    Texture CHAR(1) CONSTRAINT Glass_CK_Texture CHECK (Texture IN ('Y','N'))  
    CONSTRAINT Glass_NN_Texture NOT NULL,  
    Coating CHAR(1) CONSTRAINT Glass_CK_Coating CHECK (Coating IN ('Y','N'))  
    CONSTRAINT Glass_NN_Coating NOT NULL,  
    Color VARCHAR2(6) CONSTRAINT Glass_CK_Color CHECK (Color IN ('Red', 'Pink',  
    'Orange', 'Yellow', 'Blue', 'Green', 'Aqua', 'Purple', 'Black', 'White',  
    'Cream','Multi','Clear')) CONSTRAINT Glass_NN_Color NOT NULL,  
    Status VARCHAR2(9) CONSTRAINT Glass_CK_Status CHECK (Status IN ('Available',  
    'Reorder','Legacy')) CONSTRAINT Glass_NN_Status NOT NULL,  
    CONSTRAINT Glass_PK PRIMARY KEY(Vendor,PartNum)  
);
```

```
DESCRIBE Glass;
```

Name	Null?	Type
VENDOR	NOT NULL	VARCHAR2(40)
PARTNUM	NOT NULL	VARCHAR2(20)
TYPE	NOT NULL	VARCHAR2(11)
TEXTURE	NOT NULL	CHAR(1)
COATING	NOT NULL	CHAR(1)
COLOR	NOT NULL	VARCHAR2(6)
STATUS	NOT NULL	VARCHAR2(9)

6.10 Product

```
CREATE TABLE Product (
    Num NUMBER CONSTRAINT Product_NN_Num NOT NULL CONSTRAINT Product_PK PRIMARY KEY,
    Price NUMBER(5,2) CONSTRAINT Product_NN_Price NOT NULL,
    Status VARCHAR2(9) DEFAULT 'Available' CONSTRAINT Product_CK_Status CHECK
    (Status IN ('Available', 'Sold', 'Shipped','Shrink')) CONSTRAINT
    Product_NN_Status NOT NULL,
    Height NUMBER(3,1) CONSTRAINT Image_CK_Height CHECK (Height <= 72.0) CONSTRAINT
    Product_NN_Height NOT NULL,
    Width NUMBER(3,1) CONSTRAINT Image_CK_Width CHECK (Width <= 72.0) CONSTRAINT
    Product_NN_Width NOT NULL,
    Depth NUMBER(3,1) CONSTRAINT Image_CK_Depth CHECK (Depth <= 72.0),
    Weight NUMBER(4,1) CONSTRAINT Image_CK_Weight CHECK (Weight <= 320.0) CONSTRAINT
    Product_NN_Weight NOT NULL,
    ImgFull VARCHAR2(24) CONSTRAINT Product_CK_ImgFull CHECK (ImgFull LIKE '%.jpg'
    OR ImgFull LIKE '%.gif' OR ImgFull LIKE '%.png') CONSTRAINT Product_NN_ImgFull
    NOT NULL CONSTRAINT Product_FK_ImgFull REFERENCES Image(ImgFull),
    BoxNum NUMBER CONSTRAINT Product_NN_BoxNum NOT NULL CONSTRAINT Product_FK_BoxNum
    REFERENCES Box(Num),
    OrderNum NUMBER CONSTRAINT Product_FK_OrderNum REFERENCES Orders(Num)
);
```

```
DESCRIBE Product;
```

Name	Null?	Type
NUM	NOT NULL	NUMBER
PRICE	NOT NULL	NUMBER(5,2)
STATUS	NOT NULL	VARCHAR2(9)
HEIGHT	NOT NULL	NUMBER(3,1)
WIDTH	NOT NULL	NUMBER(3,1)
DEPTH		NUMBER(3,1)
WEIGHT	NOT NULL	NUMBER(4,1)
IMGFULL	NOT NULL	VARCHAR2(24)
BOXNUM	NOT NULL	NUMBER
ORDERNUM		NUMBER

6.11 CreatedWith

```
CREATE TABLE CreatedWith (
    PatternVendor VARCHAR2(40) CONSTRAINT CreatedWith_NN_PatternVendor NOT NULL,
    PatternPartNum VARCHAR2(20) CONSTRAINT CreatedWith_NN_PatternPartNum NOT NULL,
    GlassVendor VARCHAR2(40) CONSTRAINT CreatedWith_NN_GlassVendor NOT NULL,
    GlassPartNum VARCHAR2(20) CONSTRAINT CreatedWith_NN_GlassPartNum NOT NULL,
    PercentUsed NUMBER(3) CONSTRAINT CreatedWith_CK_PercentUsed CHECK (PercentUsed
    BETWEEN 0 AND 100) CONSTRAINT CreatedWith_NN_PercentUsed NOT NULL,
    ProdNum NUMBER CONSTRAINT CreatedWith_NN_ProdNum NOT NULL CONSTRAINT
    CreatedWith_FK_ProdNum REFERENCES Product(Num),
    CONSTRAINT CreatedWith_PK PRIMARY KEY(GlassVendor, GlassPartNum, ProdNum),
    CONSTRAINT CreatedWith_FK_Pattern FOREIGN KEY(PatternVendor,PatternPartNum)
    REFERENCES Pattern(Vendor,PartNum),
    CONSTRAINT CreatedWith_FK_Glass FOREIGN KEY (GlassVendor, GlassPartNum)
    REFERENCES Glass(Vendor,PartNum)
);
DESCRIBE CreatedWith;
```

Name	Null?	Type
PATTERNVENDOR	NOT NULL	VARCHAR2(40)
PATTERNPARTNUM	NOT NULL	VARCHAR2(20)
GLASSVENDOR	NOT NULL	VARCHAR2(40)
GLASSPARTNUM	NOT NULL	VARCHAR2(20)
PERCENTUSED	NOT NULL	NUMBER(3)
PRODNUM	NOT NULL	NUMBER

6.12 Sequences & Triggers

```
CREATE SEQUENCE CustomerID_SEQ START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE OrdersNum_SEQ START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE BoxNum_SEQ START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE ProductNum_SEQ START WITH 1 INCREMENT BY 1;
CREATE TRIGGER CustomerID_TRG BEFORE INSERT ON Customer FOR EACH ROW WHEN (NEW.ID
IS NULL) BEGIN SELECT CustomerID_SEQ.NEXTVAL INTO :NEW.ID FROM DUAL; END; /
CREATE TRIGGER OrdersNum_TRG BEFORE INSERT ON Orders FOR EACH ROW WHEN (NEW.Num IS
NULL) BEGIN SELECT OrdersNum_SEQ.NEXTVAL INTO :NEW.Num FROM DUAL; END; /
CREATE TRIGGER BoxNum_TRG BEFORE INSERT ON Box FOR EACH ROW WHEN (NEW.Num IS NULL)
BEGIN SELECT BoxNum_SEQ.NEXTVAL INTO :NEW.Num FROM DUAL; END; /
CREATE TRIGGER ProductNum_TRG BEFORE INSERT ON Product FOR EACH ROW WHEN (NEW.Num
IS NULL) BEGIN SELECT ProductNum_SEQ.NEXTVAL INTO :NEW.Num FROM DUAL; END; /
```

6.13 Views

List of all products currently for sale on the site, including theme, pattern type, color, price, packing type.

```
CREATE VIEW ProductCatalog AS SELECT DISTINCT
a.Name,a.Type,a.Theme,p.Price,p.BoxNum,p.ImgFull,p.Num,g.Color AS PrimaryColor FROM
Product p, CreatedWith c, Pattern a, Glass g,
(SELECT ProdNum, MAX(PercentUsed) AS HighPct FROM CreatedWith GROUP BY ProdNum)
primaryPerProd
WHERE p.Status='Available' AND p.Num=c.ProdNum AND g.Vendor=c.GlassVendor AND
g.PartNum=c.GlassPartNum AND c.PatternVendor=a.Vendor AND
c.PatternPartNum=a.PartNum AND c.Prodnum=primaryPerProd.ProdNum AND
c.PercentUsed=primaryPerProd.HighPCT
ORDER BY a.Theme, p.Price;
```

Daily "to be shipped" report of all non-pickup sales, grouped by order, with shipping details and with products listed by box.

Note: Some redesign would be needed in order to achieve this view. See comments in Section [8.4.1](#) on page 41.

List of all "pickup" orders scheduled for the show, including all product and customer details.

```
CREATE VIEW ShowPrep AS SELECT o.Num AS OrderNum, o.OrderDate, s.ShowDate,
c.NameTitle, c.NameFirst, c.NameLast, c.PhoneNum, b.Num AS BoxNum, p.Num AS
ProductNum FROM Orders o, CraftShow s, Box b, Product p, Customer c WHERE
o.OrderDate < s.ShowDate AND o.CustomerID=c.ID AND c.ShowDate=s.ShowDate AND
o.Num=p.OrderNum AND p.BoxNum=b.Num;
```

Quarterly and annual sales reports.

Note: Some redesign would be needed in order to achieve this view. See comments in Section [8.4.1](#) on page 41.

7.0 DML

7.1 CraftShow

7.1.1 Inserts

```
INSERT INTO CraftShow(ShowDate, Name, Location, WebPage) VALUES
(to_date('09/24/2011','mm/dd/yyyy'),'Maywood Street Fair','Main St. Maywood
NJ','http://www.theadvertisingalliance.com/maywood0506.htm');
INSERT INTO CraftShow(ShowDate, Name, Location, WebPage) VALUES
(to_date('09/17/2011','mm/dd/yyyy'),'Somerville Fall Festival','Main St. Somerville
NJ','');
INSERT INTO CraftShow(ShowDate, Name, Location) VALUES
(to_date('11/19/2011','mm/dd/yyyy'),'North Hunterdon Arts & Crafts Festival','
Hunterdon Community Center');
```

7.1.2 Updates/Deletes

Change date of a particular show. Add URL for a particular show:

```
UPDATE CraftShow SET ShowDate=to_date('09/25/2011','mm/dd/yyyy') WHERE
ShowDate=to_date('09/24/2011','mm/dd/yyyy');
UPDATE CraftShow SET WebPage='http://www.hunterdoncraft.com/xmas2011' WHERE
ShowDate=to_date('11/19/2011','mm/dd/yyyy');
```

Remove a craft show that has not been selected by any customer;

```
DELETE FROM CraftShow WHERE ShowDate=to_date('09/17/2011','mm/dd/yyyy');
```

Note: Unlike the other updates, this delete was not applied to the DB.
(Otherwise the statement represented here would not be executable.)

7.2 Payment

7.2.1 Inserts

```
INSERT INTO Payment(AccountNum, Type, ExpMon, ExpYear, CVN, BillAddressStreet,
BillAddressApt, BillAddressCity, BillAddressState, BillAddressZip) VALUES ('6011-
1234-1234-1234','Discover','May','2013','222','615 Dukes
Parkway','#1','Manville','NJ','08835-1234');
```

```
INSERT INTO Payment(AccountNum, Type, ExpMon, ExpYear, CVN, BillAddressStreet,
BillAddressCity, BillAddressState,BillAddressZip) VALUES ('4039-4321-4321-
4321','Visa','Jan','2013','913','53 Willow Grove Rd.','Brunswick','ME','04011-
1234');
```

```
INSERT INTO Payment(AccountNum, Type, ExpMon, ExpYear, CVN, BillAddressStreet,
BillAddressCity, BillAddressState,BillAddressZip) VALUES ('6011-3456-3456-
3456','Discover','Jan','2014','626','3141 Chestnut
Street','Philadelphia','PA','19104-1234');
```

7.2.2 Updates/Deletes

Change payment details for particular order:

```
UPDATE Payment SET AccountNum='5321-3456-3456-3456', Type='Mastercard',
ExpMon='Feb', ExpYear='2013', CVN='532' WHERE AccountNum IN (SELECT c.PmtAccountNum
FROM Customer c, Orders o WHERE c.ID=o.CustomerId AND o.Num=2);
```

With current constraints, updating Payment.AccountNum, as above, would only be permissible if "update on cascade" were set on Customer.PmtAccountNum (which doesn't exist in Oracle).

```
UPDATE Payment p, Customer c SET c.PmtAccountNum='5321-3456-3456-3456',
p.AccountNum='5321-3456-3456-3456', p.Type='Mastercard', p.ExpMon='Feb',
p.ExpYear='2013', p.CVN='532' WHERE AccountNum IN (SELECT c.PmtAccountNum FROM
Customer c, Orders o WHERE c.ID=o.CustomerId AND o.Num=2);
```

The above attempt to update both the PK and FK simultaneously was rejected by Oracle. This type of update must be done via trigger or stored procedure.

Given the assumption that the account number is only written to the DB after payment is verified and processed, AND given that customer would need to enter fresh payment for any subsequent orders (no customer account) this update would never be necessary. However in the event that the

website is upgraded to retain password-accessed profiles, with stored, reusable, payment information, this design will be problematic. I *believe* the fix would be to reference CustomerID as a FK in the Payment table, rather than the reverse.

```
UPDATE Payment SET BillAddressZip='08835-1220' WHERE AccountNum IN (SELECT
c.PmtAccountNum FROM Customer c, Orders o WHERE c.ID=o.CustomerId AND o.Num=2);
```

7.3 Customer

7.3.1 Inserts

```
INSERT INTO Customer(Email, PhoneNum, NameTitle, NameFirst, NameLast, NameSuffix,
PmtAccountNum) VALUES ('kmprieto@optonline.net', '908-575-
1127', 'Mr.', 'Michael', 'Prieto', 'Jr.', '6011-1234-1234-1234');
```

```
INSERT INTO Customer(Email, PhoneNum, NameFirst, NameMiddle, NameLast,
PmtAccountNum) VALUES ('chinette@gme.net', '207-729-
8978', 'Jeannette', 'H', 'Staples', '4039-4321-4321-4321');
```

```
INSERT INTO Customer(Email, PhoneNum, NameTitle, NameFirst, NameLast,
PmtAccountNum, ShowDate) VALUES ('teach@frein.net', '610-580-
3861', 'Professor', 'Stephen', 'Frein', '6011-3456-3456-
3456', to_date('11/19/2011', 'mm/dd/yyyy'));
```

7.3.2 Updates/Deletes

Add a selected show date for a particular order:

```
UPDATE Customer SET ShowDate=to_date('09/25/2011', 'mm/dd/yyyy') WHERE ID IN (SELECT
CustomerID FROM Orders where Num=4);
```

7.4 Orders

7.4.1 Inserts

```
INSERT INTO Orders(OrderDate, DestStreet, DestCity, DestState, DestZip, CustomerID)
VALUES (to_date('09/10/2011', 'mm/dd/yyyy'), '53 Willow Grove
Rd.', 'Brunswick', 'ME', '04011-1234', 3);
```

```
INSERT INTO Orders(OrderDate, DestStreet, DestApt, DestCity, DestState, DestZip,
GiftMsg, CustomerID) VALUES (to_date('09/02/2011', 'mm/dd/yyyy'), '615 Dukes
Parkway', '#1', 'Manville', 'NJ', '08835-1234', 'For my darling mooshee.', 2);
```

```
INSERT INTO Orders(OrderDate, GiftMsg, CustomerID) VALUES
(to_date('09/05/2011', 'mm/dd/yyyy'), 'Your project gets an A.', 5);
```

7.4.2 Updates/Deletes

Add a gift message for a particular order:

```
UPDATE Orders SET GiftMsg='Congratulations!' WHERE Num=1;
```

7.5 Shipment

7.5.1 Inserts

```
INSERT INTO Shipment(OrderNum, ItemNum, PostalCost) VALUES (4,1,4.80);
```

```
INSERT INTO Shipment(OrderNum, ItemNum, PostalCost) VALUES (1,1,4.90);
```

```
INSERT INTO Shipment(OrderNum, ItemNum, PostalCost) VALUES (1,2,4.90);
```

```
INSERT INTO Shipment(OrderNum, ItemNum, PostalCost) VALUES (1,3,5.08);
```

7.5.2 Updates/Deletes

Insert postal cost for particular order item:

```
UPDATE Shipment SET PostalCost=NULL WHERE OrderNum=1 AND ItemNum=3;
```

Note: Ran the above since I had already inserted all PostalCost values.

```
UPDATE Shipment SET PostalCost=5.08 WHERE OrderNum=1 AND ItemNum=3;
```

Delete the shipment corresponding to an order for which show date was added (ref Section [7.3.2](#) on page 28):

```
DELETE FROM Shipment WHERE OrderNum = 4;
```

7.6 Image

7.6.1 Inserts

```
INSERT INTO Image(ImgFull, ImgThumb, ImgFullAlt, ImgThumbAlt) VALUES  
( 'butterfly1_lg.jpg', 'butterfly1_sm.jpg', 'butterfly1_lgb.jpg', 'butterfly1_smb.jpg' )  
;
```

```
INSERT INTO Image(ImgFull, ImgThumb, ImgFullAlt, ImgThumbAlt) VALUES  
( 'butterfly2_lg.jpg', 'butterfly2_sm.jpg', 'butterfly2_lgb.jpg', 'butterfly2_smb.jpg' )  
;
```

```
INSERT INTO Image(ImgFull, ImgThumb, ImgFullAlt, ImgThumbAlt) VALUES  
( 'butterfly3_lg.jpg', 'butterfly3_sm.jpg', 'butterfly3_lgb.jpg', 'butterfly3_smb.jpg' )  
;
```

```
INSERT INTO Image(ImgFull, ImgThumb) VALUES ( 'rose1_lg.jpg', 'rose1_sm.jpg' );
```

```
INSERT INTO Image(ImgFull, ImgThumb) VALUES  
( 'critters1_lg.jpg', 'critters1_sm.jpg' );
```

```
INSERT INTO Image(ImgFull, ImgThumb, ImgFullAlt, ImgThumbAlt) VALUES  
( 'treelg1_lg.jpg', 'treelg1_sm.jpg', 'treelg1_lgb.jpg', 'treelg1_smb.jpg' );
```

```
INSERT INTO Image(ImgFull, ImgThumb, ImgFullAlt, ImgThumbAlt) VALUES  
( 'treesm1_lg.jpg', 'treesm1_sm.jpg', 'treesm1_lgb.jpg', 'treesm1_smb.jpg' );
```

```
INSERT INTO Image(ImgFull, ImgThumb, ImgFullAlt, ImgThumbAlt) VALUES  
( 'butterfly4_lg.jpg', 'butterfly4_sm.jpg', 'butterfly4_lgb.jpg', 'butterfly4_smb.jpg' )  
;
```

```
INSERT INTO Image(ImgFull, ImgThumb) VALUES ( 'wreath1_lg.jpg', 'wreath1_sm.jpg' );
```

7.6.2 Updates/Deletes

Add alternate images for a particular product:

```
UPDATE Image SET ImgFullAlt='wreath1_lgb.jpg', ImgThumbAlt='wreath1_smb.jpg' WHERE  
ImgFull IN (SELECT ImgFull FROM Product WHERE Num=9);
```

7.7 Box

7.7.1 Inserts

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES (8.6,5.4,1.6,2.0,1.50);
```

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES  
(13.0,13.0,1.6,6.0,2.50);
```

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES (8.6,5.4,5.4,4.0,2.00);
```

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES
(13.6,11.9,3.4,6.0,2.00);
```

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES (9.0,9.0,5.5,6.0,2.00);
```

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES
(11.0,11.0,7.0,7.5,3.50);
```

```
INSERT INTO Box(Height, Width, Length, Weight, Cost) VALUES
(23.7,11.8,3.0,8.0,4.00);
```

7.7.2 Updates/Deletes

Change cost of particular box:

```
UPDATE Box SET Cost=1.75 WHERE Num=1;
```

7.8 Pattern

7.8.1 Inserts

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Relei','6215-01','Traditional','Animal','Duck Couple','A pair of mallards take
flight over a secluded pond.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Relei','6215-02','Traditional','Animal','Critter Squares','Know someone who likes
"creepy crawlies?" These critters will keep him or her company: frog, butterfly,
ladybug, and goldfish framed in opaque glass and accented with glass beads.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Croyle','6965-01','Layered','Animal','Enclosed Butterfly','Interlaced arcs
enclose a butterfly in this colorful modern piece.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Croyle','6965-02','3d','Animal','Butterfly','These delicate creations come in all
colors and are wonderful for windows, mirrors, and any smooth surface. The attached
suction cup lets you place and re-place your butterfly with ease. And their bright
colors and fit-in-your-hand size make these butterflies perfect gifts!');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Croyle','6965-03','Layered','Floral','Rose','A single rose centered in a clear
oval. Another great gift!');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Croyle','6965-04','Layered','Floral','Holly Wreath','Layered light and dark
leaves and bright berries form an ever-fresh holiday wreath. Reversible: display
with either light leaves or dark leaves forward.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Aubin','6277-01','3d','Floral','Large Tree Whirl','Six arcs of pale glass
intersect to form a spherical enclosure. Within is a tree with emerald leaves,
entwined with vines that cascade along the top and sides. Includes a securely-
attached chain for hanging.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Aubin','6277-01b','3d','Floral','Small Tree Whirl','Six arcs of pale glass
intersect to form a spherical enclosure. Within is a tree with emerald leaves,
entwined with vines that cascade along the top and sides. Includes a securely-
attached chain for hanging.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Aubin', '6277-02', '3d', 'Floral', 'Large Floral Whirl', 'Flowers and butterflies
framed within a glass sphere form a beautiful centerpiece for your dining table,
sunroom, or nook.');
```

```
INSERT INTO Pattern(Vendor, PartNum, Type, Theme, Name, Descr) VALUES
('Aubin', '6277-02b', '3d', 'Floral', 'Small Floral Whirl', 'Flowers and butterflies
framed within a glass sphere form a beautiful centerpiece for your dining table,
sunroom, or nook.');
```

7.8.2 Updates/Deletes

Update description for particular pattern:

```
UPDATE Pattern SET Descr='A single rose centered in a clear oval.' WHERE
Vendor='Croyle' AND PartNum='6965-03';
```

7.9 Glass

7.9.1 Inserts

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Bullseye', '90COE-32-568', 'Translucent', 'N', 'Y', 'Blue', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Bullseye', '90COE-32-569', 'Translucent', 'N', 'Y', 'Aqua', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Bullseye', '90COE-32-572', 'Translucent', 'N', 'Y', 'Green', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Bullseye', '90COE-32-575', 'Translucent', 'N', 'Y', 'Green', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Kokomo', 'RD000', 'Wispy', 'N', 'N', 'Black', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Kokomo', 'RD003', 'Wispy', 'N', 'N', 'Red', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Kokomo', 'RD013', 'Wispy', 'N', 'N', 'Orange', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Kokomo', 'RD050', 'Wispy', 'N', 'N', 'Brown', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Kokomo', 'RD111', 'Wispy', 'N', 'N', 'Cream', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Kokomo', 'RGB103', 'Opal', 'Y', 'N', 'Multi', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Wissmach', 'WGC118', 'Translucent', 'Y', 'Y', 'Yellow', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Wissmach', 'WGC109', 'Translucent', 'N', 'Y', 'White', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Wissmach', 'WGC223', 'Wispy', 'Y', 'N', 'Green', 'Available');
```

```
INSERT INTO Glass(Vendor, PartNum, Type, Texture, Coating, Color, Status) VALUES
('Wissmach', 'WGC000', 'Translucent', 'N', 'N', 'Clear', 'Available');
```

7.9.2 Updates/Deletes

Change availability of a particular glass:

```
UPDATE Glass SET Status='Legacy' WHERE Vendor='Kokomo' AND PartNum='RGB103';
```

7.10 Product

7.10.1 Inserts

```
INSERT INTO Product(Price, Status, Height, Width, Depth, Weight, ImgFull, BoxNum, OrderNum) VALUES (12.00, 'Sold', 4.0, 3.5, 2.3, 4.0, 'butterfly3_lg.jpg', 3, 1);
```

```
INSERT INTO Product(Price, Status, Height, Width, Depth, Weight, ImgFull, BoxNum) VALUES (12.00, 'Available', 4.0, 3.5, 2.3, 4.0, 'butterfly1_lg.jpg', 3);
```

```
INSERT INTO Product(Price, Status, Height, Width, Depth, Weight, ImgFull, BoxNum) VALUES (12.00, 'Available', 4.0, 3.5, 2.3, 4.0, 'butterfly2_lg.jpg', 3);
```

```
INSERT INTO Product(Price, Status, Height, Width, Depth, Weight, ImgFull, BoxNum, OrderNum) VALUES (58.00, 'Sold', 7.3, 4.3, 7.3, 12.0, 'treesml_lg.jpg', 5, 1);
```

```
INSERT INTO Product(Price, Status, Height, Width, Depth, Weight, ImgFull, BoxNum) VALUES (12.00, 'Available', 4.0, 3.5, 2.3, 4.0, 'butterfly4_lg.jpg', 3);
```

```
INSERT INTO Product(Price, Status, Height, Width, Weight, ImgFull, BoxNum, OrderNum) VALUES (20.00, 'Sold', 8.5, 6.0, 7.0, 'rose1_lg.jpg', 1, 1);
```

```
INSERT INTO Product(Price, Status, Height, Width, Weight, ImgFull, BoxNum) VALUES (35.00, 'Available', 12.0, 12.0, 9.0, 'critters1_lg.jpg', 2);
```

```
INSERT INTO Product(Price, Status, Height, Width, Weight, ImgFull, BoxNum) VALUES (45.00, 'Available', 12.5, 12.5, 12.0, 'wreath1_lg.jpg', 2);
```

```
INSERT INTO Product(Price, Status, Height, Width, Depth, Weight, ImgFull, BoxNum, OrderNum) VALUES (88.00, 'Sold', 9.3, 6.3, 9.3, 16.5, 'treelg1_lg.jpg', 6, 2);
```

7.10.2 Updates/Deletes

Add order number & change status of a particular product:

```
UPDATE Product SET OrderNum=4, Status='Sold' WHERE Num=6;
```

Change price of all unsold products based on a particular pattern:

```
UPDATE Product SET Price=12.95 WHERE Num IN (SELECT ProdNum FROM CreatedWith WHERE PatternVendor='Croyle' AND PatternPartNum='6965-03') AND Status='Available';
```

7.11 CreatedWith

7.11.1 Inserts

```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum, PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Bullseye', '90COE-32-572', 70, 2);
```

```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum, PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Wissmach', 'WGC223', 30, 2);
```

```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum, PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Kokomo', 'RD003', 70, 3);
```

```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum, PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Wissmach', 'WGC118', 30, 3);
```

```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum, PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Wissmach', 'WGC118', 70, 4);
```



```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Kokomo', 'RD013', 40, 4);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Wissmach', 'WGC000', 50, 5);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Wissmach', 'WGC118', 30, 5);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Bullseye', '90COE-32-572', 20, 5);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Relei', '6215-02', 'Kokomo', 'RD111', 60, 6);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Relei', '6215-02', 'Bullseye', '90COE-32-568', 13, 6);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Relei', '6215-02', 'Kokomo', 'RD013', 13, 6);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Relei', '6215-02', 'Bullseye', '90COE-32-572', 12, 6);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Relei', '6215-02', 'Kokomo', 'RD003', 6, 6);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Relei', '6215-02', 'Kokomo', 'RD000', 6, 6);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01', 'Wissmach', 'WGC000', 30, 7);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01', 'Wissmach', 'WGC223', 30, 7);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01', 'Bullseye', '90COE-32-575', 20, 7);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01', 'Kokomo', 'RD050', 20, 7);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01b', 'Wissmach', 'WGC000', 30, 8);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01b', 'Wissmach', 'WGC223', 30, 8);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01b', 'Bullseye', '90COE-32-575', 20, 8);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Aubin', '6277-01b', 'Kokomo', 'RD050', 20, 8);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Bullseye', '90COE-32-569', 70, 9);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-03', 'Bullseye', '90COE-32-568', 30, 9);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-04', 'Bullseye', '90COE-32-572', 45, 10);
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum,
PercentUsed, ProdNum) VALUES ('Croyle', '6965-04', 'Bullseye', '90COE-32-575', 45, 10);
```

```
INSERT INTO CreatedWith(PatternVendor, PatternPartNum, GlassVendor, GlassPartNum, PercentUsed, ProdNum) VALUES ('Croyle', '6965-04', 'Kokomo', 'RD013', 10, 10);
```

7.11.2 Updates/Deletes

Change glass %s for particular product:

```
UPDATE CreatedWith SET PercentUsed=20 WHERE GlassVendor='Wissmach' AND GlassPartNum='WGC000' AND ProdNum=5;
```

```
UPDATE CreatedWith SET PercentUsed=60 WHERE GlassVendor='Wissmach' AND GlassPartNum='WGC118' AND ProdNum=5;
```

8.0 Queries

8.1 Products

8.1.1 Populate Product Page

"Each product page lists several characteristics about the unique product."

```
SELECT DISTINCT p.Num,
a.Name, a.Descr, a.Type, a.Theme, p.Price, p.ImgFull, i.ImgThumb, i.ImgFullAlt, i.ImgThumbAlt, g.Color AS PrimaryColor
FROM Product p, CreatedWith c, Pattern a, Glass g, Image i,
(SELECT ProdNum, MAX(PercentUsed) AS HighPct FROM CreatedWith GROUP BY ProdNum)
primaryPerProd
WHERE p.Status='Available' AND p.Num=c.ProdNum AND c.PatternVendor=a.Vendor AND
c.PatternPartNum=a.PartNum AND c.GlassVendor=g.Vendor AND c.GlassPartNum=g.PartNum
AND p.ImgFull=i.ImgFull AND c.Prodnum=primaryPerProd.ProdNum AND
c.PercentUsed=primaryPerProd.HighPCT
ORDER BY a.Theme, p.Price;
```

NUM NAME

DESCR

TYPE THEME PRICE IMGFULL IMGTHUMB

IMGFULLALT IMGTHUMBALT PRIMAR

3 Butterfly

These delicate creations come in all colors and are wonderful for windows, mirrors, and any smooth surface. The attached suction cup lets you place and re-place your butterfly with ease. And their bright colors and fit-in-your-hand size make these butterflies perfect gifts!

NUM NAME

DESCR

TYPE THEME PRICE IMGFULL IMGTHUMB

```

-----
IMGFULLALT          IMGTHUMBALT          PRIMAR
-----
3d          Animal          12.95 butterfly1_lg.jpg          butterfly1_sm.jpg
butterfly1_lgb.jpg          butterfly1_smb.jpg          Red
      4 Butterfly
These delicate creations come in all colors and are wonderful for windows, mirro
NUM NAME
-----

```

DESCR

```

-----
TYPE          THEME          PRICE IMGFULL          IMGTHUMB
-----
IMGFULLALT          IMGTHUMBALT          PRIMAR
-----
rs, and any smooth surface. The attached suction cup lets you place and re-place
your butterfly with ease. And their bright colors and fit-in-your-hand size mak
e these butterflies perfect gifts!
3d          Animal          12.95 butterfly2_lg.jpg          butterfly2_sm.jpg
butterfly2_lgb.jpg          butterfly2_smb.jpg          Yellow
NUM NAME
-----

```

DESCR

```

-----
TYPE          THEME          PRICE IMGFULL          IMGTHUMB
-----
IMGFULLALT          IMGTHUMBALT          PRIMAR
-----
      8 Critter Squares
Know someone who likes "creepy crawlies?" These critters will keep him or her co
mpany: frog, butterfly, ladybug, and goldfish framed in opaque glass and accente
d with glass beads.
NUM NAME
-----

```

DESCR

```

-----
TYPE          THEME          PRICE IMGFULL          IMGTHUMB
-----
IMGFULLALT          IMGTHUMBALT          PRIMAR
-----
Traditional Animal          35 critters1_lg.jpg          critters1_sm.jpg
                                          Cream
      9 Holly Wreath
Layered light and dark leaves and bright berries form an ever-fresh holiday wrea

```

NUM	NAME	DESCR
th.	Reversible:	display with either light leaves or dark leaves forward.
Layered	Floral	12.95 wreath1_lg.jpg wreath1_sm.jpg
wreath1_lgb.jpg		wreath1_smb.jpg Green

8.1.2 Site Searches

"Product characteristics are searchable." (For example, Type.)

```
SELECT DISTINCT p.Num, a.Theme, a.Name, p.Price, i.ImgThumb from Product p,
CreatedWith c, Pattern a, Image i WHERE a.Type='3d' AND p.Status='Available' AND
p.Num=c.ProdNum AND c.PatternVendor=a.Vendor AND c.PatternPartNum=a.PartNum AND
p.ImgFull = i.ImgFull ORDER by a.Theme, p.Price;
```

NUM	THEME	NAME	PRICE	IMGTHUMB
3	Animal	Butterfly	12.95	butterfly1_sm.jpg
4	Animal	Butterfly	12.95	butterfly2_sm.jpg

```
SELECT DISTINCT p.Num, a.Theme, a.Name, p.Price, i.ImgThumb from Product p,
CreatedWith c, Pattern a, Image i WHERE a.Type='Layered' AND p.Status='Available'
AND p.Num=c.ProdNum AND c.PatternVendor=a.Vendor AND c.PatternPartNum=a.PartNum AND
p.ImgFull = i.ImgFull ORDER by a.Theme, p.Price;
```

NUM	THEME	NAME	PRICE	IMGTHUMB
9	Floral	Holly Wreath	12.95	wreath1_sm.jpg

```
SELECT DISTINCT p.Num, a.Theme, a.Name, p.Price, i.ImgThumb from Product p,
CreatedWith c, Pattern a, Image i WHERE a.Type='Traditional' AND
p.Status='Available' AND p.Num=c.ProdNum AND c.PatternVendor=a.Vendor AND
c.PatternPartNum=a.PartNum AND p.ImgFull = i.ImgFull ORDER by a.Theme, p.Price;
```

NUM	THEME	NAME	PRICE	IMGTHUMB
8	Animal	Critter Squares	35	critters1_sm.jpg

8.1.3 Product Catalog

"List of all products currently for sale on the site, including theme, pattern type, color, price, packing type."

```
SELECT DISTINCT a.Name,a.Type,a.Theme,p.Price,p.BoxNum,p.ImgFull,p.Num,g.Color AS
PrimaryColor FROM Product p, CreatedWith c, Pattern a, Glass g,
(SELECT ProdNum, MAX(PercentUsed) AS HighPct FROM CreatedWith GROUP BY ProdNum)
primaryPerProd WHERE p.Status='Available' AND p.Num=c.ProdNum AND
g.Vendor=c.GlassVendor AND g.PartNum=c.GlassPartNum AND c.PatternVendor=a.Vendor
AND c.PatternPartNum=a.PartNum AND c.Prodnum=primaryPerProd.ProdNum AND
c.PercentUsed=primaryPerProd.HighPCT
ORDER BY a.Theme, p.Price;
```

NAME	TYPE	THEME	PRICE	BOXNUM
Butterfly butterfly1_lg.jpg	3d	Animal 3 Red	12.95	3
Butterfly butterfly2_lg.jpg	3d	Animal 4 Yellow	12.95	3
Critter Squares critters1_lg.jpg	Traditional	Animal 8 Cream	35	2

NAME	TYPE	THEME	PRICE	BOXNUM
Holly Wreath wreath1_lg.jpg	Layered	Floral 9 Green	12.95	2

8.1.4 Pattern/Glass Usage 1

"List of all products with certain pattern or glass characteristics." (For example, Wispy glass.)

```
SELECT DISTINCT p.Num, a.Name, a.Type, a.Theme, p.Price, p.BoxNum, p.ImgFull,
c.PercentUsed FROM Product p, CreatedWith c, Glass g, Pattern a WHERE
g.Type='Wispy' AND a.Vendor=c.PatternVendor AND a.PartNum=c.PatternPartNum AND
g.Vendor=c.GlassVendor AND g.PartNum=c.GlassPartNum AND c.PercentUsed >= 50 AND
c.ProdNum=p.Num ORDER BY a.Theme, p.Price;
```

NUM NAME	TYPE	THEME	PRICE	BOXNUM
3 Butterfly butterfly1_lg.jpg	3d	Animal	12.95	3
8 Critter Squares	Traditional	Animal	35	2

critters1_lg.jpg 60

(Or a Croyle pattern.)

```
SELECT DISTINCT p.Num, a.Name,a.Type, a.Theme, p.Price, p.BoxNum, p.ImgFull FROM
Product p, CreatedWith c, Pattern a WHERE a.Vendor='Croyle' AND
a.Vendor=c.PatternVendor AND a.PartNum=c.PatternPartNum AND c.ProdNum=p.Num ORDER
BY a.Theme,p.Price;
```

NUM	NAME	TYPE	THEME	PRICE	BOXNUM

IMGFULL					

2	Butterfly	3d	Animal	12	3
butterfly3_lg.jpg					
6	Butterfly	3d	Animal	12	3
butterfly4_lg.jpg					
3	Butterfly	3d	Animal	12.95	3
butterfly1_lg.jpg					
NUM	NAME	TYPE	THEME	PRICE	BOXNUM

IMGFULL					

4	Butterfly	3d	Animal	12.95	3
butterfly2_lg.jpg					
9	Holly Wreath	Layered	Floral	12.95	2
wreath1_lg.jpg					
7	Rose	Layered	Floral	20	1
rosel_lg.jpg					
6 rows selected.					

8.1.5 Pattern/Glass Usage 2

Check which products use a particular pattern or glass

```
SELECT DISTINCT ProdNum FROM CreatedWith WHERE PatternVendor='Croyle' AND
PatternPartNum='6965-03' ;
```

PRODNUM
7

```
SELECT DISTINCT ProdNum FROM CreatedWith WHERE GlassVendor='Wissmach' ;
```

PRODNUM
2

3
4
5
7
10

6 rows selected.

8.2 Craft Shows

List all upcoming craft shows.

```
SELECT * from CraftShow where ShowDate > SYSDATE;
```

```
SHOWDATE  NAME
```

```
-----  
LOCATION
```

```
-----  
WEBPAGE
```

```
-----  
17-SEP-11 Somerville Fall Festival  
Main St. Somerville NJ  
25-SEP-11 Maywood Street Fair  
Main St. Maywood NJ  
http://www.theadvertisingalliance.com/maywood0506.htm
```

```
SHOWDATE  NAME
```

```
-----  
LOCATION
```

```
-----  
WEBPAGE
```

```
-----  
19-NOV-11 North Hunterdon Arts & Crafts Festival  
Hunterdon Community Center  
http://www.hunterdoncraft.com/xmas2011
```

8.3 Orders

8.3.1 Subtotal

"The product subtotal is calculated and displayed based on the products ordered. "

```
SELECT o.Num, SUM(p.Price) AS SubTot FROM Orders o, Product p WHERE  
p.OrderNum=o.Num GROUP BY o.Num;
```

```
NUM      SUBTOT  
-----  
1         90  
2         88  
4         12
```

8.3.2 Tax

"If the shipping address is within New Jersey, state tax is added."

```
SELECT o.Num, round ((SUM(p.Price) * 0.07),2) AS Tax FROM Orders o, Product p where
DestState != 'NJ' GROUP BY o.Num;
```

NUM	TAX
1	18.47

8.3.3 Handling

"Handling is a sum of the cost of each box used for the order."

```
SELECT o.Num, SUM(b.Cost) AS Handling FROM Box b, Product p, Orders o WHERE
o.Num=p.OrderNum AND p.BoxNum=b.Num GROUP BY o.Num;
```

NUM	HANDLING
1	5.75
2	3.5
4	2

8.3.4 Shipping

"Shipping per box is determined through an interface with USPS.com, based on destination zip code, box weight, and box dimensions."

Note: The query to generate output values to USPS.com is not supported by the current design. For suspected root cause, see comments in Section [8.4.1](#) on page 41.

Attempted Query:

```
SELECT s.OrderNum, s.ItemNum,o.DestZip, (b.Weight + p.Weight) AS TotalWeight,
b.Height, b.Width, b.Length FROM Shipment s, Orders o, Box b, Product p WHERE
s.OrderNum=o.Num AND o.Num=p.OrderNum AND p.BoxNum=b.Num;
```

Desired Results:

ORDERNUM	ITEMNUM	DESTZIP	TOTALWEIGHT	HEIGHT	WIDTH	LENGTH
1	1	04011-1234	8	8.6	5.4	5.4
1	3	04011-1234	18	9	9	5.5
1	2	04011-1234	9	8.6	5.4	1.6
4	1	08835-1234	8	8.6	5.4	5.4

10 rows selected.

Actual Results:

ORDERNUM	ITEMNUM	DESTZIP	TOTALWEIGHT	HEIGHT	WIDTH	LENGTH
1	1	04011-1234	8	8.6	5.4	5.4
1	2	04011-1234	8	8.6	5.4	5.4
1	3	04011-1234	8	8.6	5.4	5.4
1	1	04011-1234	18	9	9	5.5
1	2	04011-1234	18	9	9	5.5
1	3	04011-1234	18	9	9	5.5
4	1	08835-1234	8	8.6	5.4	5.4
1	1	04011-1234	9	8.6	5.4	1.6
1	2	04011-1234	9	8.6	5.4	1.6
1	3	04011-1234	9	8.6	5.4	1.6

10 rows selected.

```
SELECT o.Num, SUM(PostalCost) AS ShippingTotal FROM Shipment s, Orders o WHERE
s.OrderNum=o.Num GROUP BY o.Num;
```

NUM	SHIPPINGTOTAL
1	14.88
4	4.8

8.4 Fulfillment

8.4.1 Pending Shipments

"At the end of each day, the proprietor will need a "to be shipped" report of all non-pickup sales, grouped by order, with shipping details and with products listed by box. "

Note: This report is not supported by the current design.

I believe this failing is caused by lack of a direct connection between Shipment Item and Product: the DB has no way to equate a particular shipment item with the product it contains. Therefore an ambiguous dataset is returned.

This reveals that the ER model *should* have included a binary relationship between Shipment and Product, or a ternary relationship between Shipment, Order, and Product.

Attempted Query:

```
SELECT o.Num, o.OrderDate, o.DestStreet, o.DestApt, o.DestCity, o.DestState,
o.DestZip, o.GiftMsg, s.ItemNum, s.PostalCost, b.Num AS Box, p.Num FROM Orders o,
Shipment s, Box b, Product p WHERE o.OrderDate>to_date('09/01/2011','mm/dd/yyyy')
AND o.Num=s.OrderNum AND o.Num=p.OrderNum AND p.BoxNum=b.Num;
```

Desired Results:

```

NUM ORDERDATE DESTSTREET DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX          NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick          ME 04011-1234
Congratulations!                                1
      4.9          3          2

```

```

NUM ORDERDATE DESTSTREET DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX          NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick          ME 04011-1234
Congratulations!                                2
      4.9          1          7

```

```

NUM ORDERDATE DESTSTREET DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX          NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick          ME 04011-1234
Congratulations!                                3
      5.08         5          5

```

```

NUM ORDERDATE DESTSTREET DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX          NUM
-----

```

```

          4 02-SEP-11 615 Dukes Parkway          #1
Manville          NJ 08835-1234
For my darling mooshee.                          1
          4.8          3          6
    
```

Actual Results:

```

          NUM ORDERDATE DESTSTREET          DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG          ITEMNUM
-----
POSTALCOST          BOX          NUM
-----
    
```

```

          1 10-SEP-11 53 Willow Grove Rd.
Brunswick          ME 04011-1234
Congratulations!                          1
          4.9          3          2
    
```

```

          NUM ORDERDATE DESTSTREET          DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG          ITEMNUM
-----
POSTALCOST          BOX          NUM
-----
    
```

```

          1 10-SEP-11 53 Willow Grove Rd.
Brunswick          ME 04011-1234
Congratulations!                          2
          4.9          3          2
    
```

```

          NUM ORDERDATE DESTSTREET          DES
-----
DESTCITY          DE DESTZIP
-----
GIFTMSG          ITEMNUM
-----
POSTALCOST          BOX          NUM
-----
    
```

```

          1 10-SEP-11 53 Willow Grove Rd.
Brunswick          ME 04011-1234
Congratulations!                          3
          5.08          3          2
    
```

```

          NUM ORDERDATE DESTSTREET          DES
-----
DESTCITY          DE DESTZIP
-----
    
```

```

-----
GIFTMSG                                     ITEMNUM
-----
POSTALCOST      BOX      NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick              ME 04011-1234
Congratulations!                                     1
      4.9          5          5
      NUM ORDERDATE DESTSTREET                                     DES
-----

```

```

-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                     ITEMNUM
-----
POSTALCOST      BOX      NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick              ME 04011-1234
Congratulations!                                     2
      4.9          5          5
      NUM ORDERDATE DESTSTREET                                     DES
-----

```

```

-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                     ITEMNUM
-----
POSTALCOST      BOX      NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick              ME 04011-1234
Congratulations!                                     3
      5.08         5          5
      NUM ORDERDATE DESTSTREET                                     DES
-----

```

```

-----
DESTCITY          DE DESTZIP
-----
GIFTMSG                                     ITEMNUM
-----
POSTALCOST      BOX      NUM
-----
      4 02-SEP-11 615 Dukes Parkway          #1
Manville              NJ 08835-1234
For my darling mooshee.                               1
      4.8          3          6
-----

```

Fantasy Glassworks Database System

```

      NUM ORDERDATE DESTSTREET                                DES
-----
DESTCITY                DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX        NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick                ME 04011-1234
Congratulations!                                1
      4.9          1          7

```

```

      NUM ORDERDATE DESTSTREET                                DES
-----
DESTCITY                DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX        NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick                ME 04011-1234
Congratulations!                                2
      4.9          1          7

```

```

      NUM ORDERDATE DESTSTREET                                DES
-----
DESTCITY                DE DESTZIP
-----
GIFTMSG                                ITEMNUM
-----
POSTALCOST        BOX        NUM
-----
      1 10-SEP-11 53 Willow Grove Rd.
Brunswick                ME 04011-1234
Congratulations!                                3
      5.08         1          7

```

10 rows selected.

8.4.2 Pending Pickups

"Before each craft show, the proprietor would like a list of all "pickup" orders scheduled for the show, including all product and customer details."

```
SELECT o.Num AS OrderNum, o.OrderDate, s.ShowDate, c.NameTitle, c.NameFirst,
c.NameLast, c.PhoneNum, b.Num AS BoxNum, p.Num AS ProductNum FROM Orders o,
CraftShow s, Box b, Product p, Customer c WHERE o.OrderDate < s.ShowDate AND
o.CustomerID=c.ID AND c.ShowDate=s.ShowDate AND o.Num=p.OrderNum AND
p.BoxNum=b.Num;
```

ORDERNUM	ORDERDATE	SHOWDATE	NAMETITLE	NAMEFIRST		
					NAMELAST	PHONENUM
					BOXNUM	PRODUCTNUM
4	02-SEP-11	25-SEP-11	Mr.	Michael		
					Prieto	908-575-1127
						3
						6
2	05-SEP-11	19-NOV-11	Professor	Stephen		
					Frein	610-580-3861
						6
						10

8.5 Business Reporting

Note: These reports are not supported by the current design. For suspected root cause, see comments in Section [8.4.1](#) on page 41.

In addition, both of these queries currently apply tax indiscriminately, whereas they should only apply tax for orders associated with a Show Date with a Destination Zip within New Jersey. The correction for this was not applied due to constraints. This raises a question of whether the initial calculated values should have been stored in the DB for "historical preservation."

8.5.1 Quarterly

"At the end of each tax quarter, the proprietor will need a list of all in-state sales within the quarter: Order number, subtotal, shipping cost, tax, and total, with overall totals for the quarter. "

Attempted Query:

```
SELECT o.Num, SUM(p.Price) AS SubTotal, (SUM(p.Price) * 0.07) AS Tax, SUM(b.Cost)
AS Handling, SUM(PostalCost) AS ShippingTotal, (SUM(p.Price) + (SUM(p.Price) *
0.07) + SUM(b.Cost) + SUM(PostalCost)) AS OrderTotal
FROM Orders o, Product p, Box b, Shipment s WHERE o.Num IN
(SELECT Num from Orders where OrderDate BETWEEN to_date('7/01/2011','mm/dd/yyyy')
AND to_date('9/30/2011','mm/dd/yyyy'))
AND o.Num = p.OrderNum AND p.BoxNum = b.Num AND o.Num = S.OrderNum
GROUP BY o.Num;
```

Desired Results:

NUM	SUBTOTAL	TAX	HANDLING	SHIPPINGTOTAL	ORDERTOTAL
1	90	0	5.5	14.88	110.38
2	88	6.16	4	0	98.16
4	12	.84	2	0	14.84

Actual Results:

NUM	SUBTOTAL	TAX	HANDLING	SHIPPINGTOTAL	ORDERTOTAL
1	270	18.9	17.25	44.64	350.79
4	12	.84	2	4.8	19.64

8.5.2 Annual

"At the end of the year a similar report is needed for all sales within the year."

Attempted Query:

```
SELECT o.Num, SUM(p.Price) AS SubTotal, (SUM(p.Price) * 0.07) AS Tax, SUM(b.Cost)
AS Handling, SUM(PostalCost) AS ShippingTotal, (SUM(p.Price) + (SUM(p.Price) *
0.07) + SUM(b.Cost) + SUM(PostalCost)) AS OrderTotal
FROM Orders o, Product p, Box b, Shipment s WHERE o.Num IN
(SELECT Num from Orders where OrderDate BETWEEN to_date('01/01/2011', 'mm/dd/yyyy')
AND to_date('12/31/2011', 'mm/dd/yyyy'))
AND o.Num = p.OrderNum AND p.BoxNum = b.Num AND o.Num = S.OrderNum
GROUP BY o.Num;
```

Results similar to Section 8.5.1.

9.0 Lessons Learned

- Requirements don't always capture what's needed.
For example, the initial design of Orders omitted a date field because the requirements didn't specifically mention one. The need for a date field was not discovered until queries for quarterly and annual sales reports were drafted.
- Relationships documented in the ER Model influence reporting capabilities.
In this project, some reporting requirements cannot be met because shipment items have no direct tie to products. Based on this example, it seems advisable to confirm the ER model against reporting requirements before designing the relational schema.
- Calculation results don't necessarily match the precision of the original values – the ROUND function can be used to fix that.
- Dates are an unavoidable pain.
- Ampersands in values are best avoided.
- Altering a constraint involves first dropping the constraint, the re-adding it with the modified syntax.

10.0 Bibliography

- AlanP. (2003, Oct 15). How to update multiple tables with sql. [Message 2]. Message posted to <http://www.dbforums.com/oracle/947094-how-update-multiple-tables-sql.html>
- Camire, B. (2002, Oct 21). Using & (ampersand) in INSERT statements. Message posted to <http://forums.devx.com/archive/index.php/t-17686.html>
- Elmasri, R. & Navathe, S. B. (2011). *Fundamentals of Database Systems, 6th ed.* Boston: Addison-Wesley.
- Forta, B. (2004). Sams Teach Yourself Regular Expressions in 10 Minutes. Available from <http://my.safaribooksonline.com/book/programming/regular-expressions/0672325667>
- Frein, S. (2011). Week 3: Relational Schemas. [Powerpoint slides].
- Frein, S. (2011). SQL 1: Basic DDL. [Powerpoint slides].
- Frein, S. (2011). SQL 2: Basic DML. [Powerpoint slides].
- Frein, S. (2011). SQL 3: Advanced concepts. [Powerpoint slides].
- Getting ORA-00905 error when trying to add a foreign key. (2010, Dec 29). Message posted to <http://stackoverflow.com/questions/4558075/getting-ora-00905-error-when-trying-to-add-a-foreign-key>
- Oracle/PLSQL: Insert a date/time value into an Oracle table. (2011). Retrieved August 20, 2011 from http://www.techonthenet.com/oracle/questions/insert_date.php
- Oracle/PLSQL: Round Function (with numbers). (2011). Retrieved August 21, 2011 from http://www.techonthenet.com/oracle/functions/round_nbr.php
- Oracle Corp. (2002). "CASCADE UPDATE Trigger Code." *Oracle Migration Workbench Reference Guide for Microsoft Access 2.0, 95, 97, 2000 Migrations Release 9.2.0 for Microsoft Windows 98/2000/NT, and Microsoft Windows XP.* Retrieve August 21, 2011 from http://download.oracle.com/docs/html/B10255_01/b.htm#639834
- Oracle Corp. (2002). *Oracle 9i Error Messages Release 2 (9.2).* Retrieved August 12, 2011 from http://download.oracle.com/docs/cd/B10500_01/server.920/a96525/e900.htm
- Oracle Corp. (2005). "Using Regular Expressions in Oracle Database." *Oracle® Database Application Developer's Guide – Fundamentals 10g Release 2 (10.2).* Retrieved August 8, 2011 from http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14251/adfns_regexp.htm#CHDJAGEF
- Oracle NULLs. (n.d.) Retrieved August 21, 2011 from <http://psoug.org/reference/null.html>
- SQL CHECK Constraint (2011). Retrieved August 21, 2011 from http://www.w3schools.com/sql/sql_check.asp
- Updating two tables at once. (2010, Aug 3). Messages posted to <https://forums.oracle.com/forums/thread.jspa?threadID=1108735>
- USPS. (2011). Postage Price Calculator. Retrieved August 14, 2011 from <http://postcalc.usps.gov/Default.aspx>
- Williams, B. (2001). Oracle Sequence for Auto-Increment. Retrieved August 15, 2011 from http://www.databaseanswers.org/sql_scripts/ora_sequence.htm