SHANGHAI KAI XING INDUSTRY AND TRADE CO., LTD

Data Architecture Report

"Designing Prosperity One Thread at a Time"

Jane Park, Ping Au Lin, James Tang, Alex Ung

Table of Contents

EXECUTIVE SUMMARY 3

BUSINESS FUNCTION-TO-DATA ENTITY MATRIX 5

USER VIEWS 6

SALES AND ORDER ERD 9

SHIPPING ERD 10

SHANGHAI KAI XING CDM 11

DOMAIN DEFINITIONS AND BUSINESS RULES/CONSTRAINTS 12

SHANGHAI KAI XING CDM IN 3NF 13

PROCESS VS. ENTITY MATRIX (CUD/CRUD) 14

TRANSACTION ANALYSIS FORMS 15

COMPOSITE USAGE MAP 19

DDL STATEMENTS 20

SAMPLE DML QUERIES 47

DISCUSSION OF THE PHYSICAL DESIGN 49

SUMMARY AND CONCLUSION 53

Executive Summary

In this report, we will conceptualize the back-end work for our data architecture project for the Chinese based firm Shanghai Kai Xing Industry and Trade Co., LTD. There were several flaws in certain database functions that we found in our initial consultation with the company that we saw as opportunities to help.

- There was no immediate way to handle changing quantities
- No efficient way to keep track of sales orders and update quantity on hand
- No efficient way to keep track of shipments
- The client did not have a database and used paper records

Our report will visualize numerous components of the project from the beginning where we designed the database to organizing the structure of the database and then implementing the design work. In particular, we will show:

- Entity Relationship Diagrams
- Conceptual Data Model in the third normal form
- Process versus Entity Matrix and data volume analysis
- Database optimization with denormalization
- Data Definition Language Code (DDL)
- Data Manipulation Language Code (DML)

By consulting with the client to diagnose their unique problem, brainstorming solutions and handwriting the database out, and then to physically create the database, our team feels confident we have successfully implemented key and necessary changes to the client's database. These

components will illustrate the work that went on from brainstorming the database on paper to implementing it fully in Oracle SQL Developer to deploy in the customer's database.

Business Function – to – Entity Matrix

PROCESS VS ENTITY MATRIX

Function \ Entity	INVOICE	INVOICE_LOC	INVOICE_TT	INVOICE_PMT_TYPE	INVOICE_DETAIL	SHIPPER	PMT_TYPE	BUYER_PMT_TYPE	LOC	TT	BUYER	ORDER	BRAND	ORDER_DETAIL	PRODUCT	COMPOSITION	DESIGN	PRODUCT_DETAIL	COLOR	SIZE	CONTENT	CARTON	PACKAGING	SHIPPING
Sales																								
Shipping																								

Legend:

Yellow: Entities belonging to the Sales Function

Green: Entities belonging to the Shipping Function

User View – Purchase Order

Shanghai Mingbo Apparel Co., Ltd.
329 Xingdacun-changxin, Yexie Zhe, SonJiang Qu, Shanghai, China
TEL: 86-159-0062-5617 FAX: 86-(0)21-57627020

Purchase Order

Buyer	Purchase Order Number	Contact	Name/Tel
MOMOKorea	9.534E+09	82-70-4334-8685	Ji-woo Shin

		Procurement Date	Estimated Delivery
Raw Material	70% Wool/ 30%Cashemre	2018.8.27	2018.9.10

Poduct Detail Number	Waist (cm)	Neck (cm)	Chest (cm)	Shoulder (cm)	Wrist (cm)	Hips (cm)	Knee (cm)	Crotch (cm)	Thigh (cm)	Comments	Quantity (Kg)	Price (Kg/RMB)	Price after Tax RMB
26846	44	35	30	17	12	42	x	x	x		21	1,000	1,050
26910	46	39	32	15	14	45	x	x	x		21	1,500	1,575
26847	45	37	33	17	13	43	x	x	x		86	1,200	1260
26813	44	35	30	15	12	42	x	x	x		53	1,700	1,785
26912	45	37	34	17	13	43	x	x	x		21	1,200	1,260
26848	48	50	36	18	16	47	x	x	x		48	1,400	1470
TOTAL	272	233	195	99	80	262					250	8,000	8,400

EMPLOYEE NAME EMPLOYEE TEL: 1.59E+10

This is a translated purchase order provided to us from the client in which we used to model the Sales function for the company.

User View – Sales Order

Room 1045, No 35 Rijing Road, Wai Gao Qiao Free Zone Shanghai China
TEL: 86-13601620582 (Mr. Yin, Manager)
TEL: 86-15900625617 (Mr. James Park, Mingbo, President)

INVOICE

Company Pearltex Korea
Address 23, Samsung-ro 119-gi:, Gangnam-gu, Seoul, Korea.
Brand evercube
Tel 82-2-515-1519
Fax. 82-2-515-2324

Invoice No.: 18081901

(Unit: PCS, USD)

Style No.	Yarn Composition	Quantity	Unit Price	Amount
8EF-S01	100% Cashmere	200	47.57	9,514.00
8EF-S02	100% Cashmere	160	52.65	8,424.00
8EF-S03	100% Cashmere	160	50.94	8,150.40
8EF-S04	100% Cashmere	160	46.02	7,363.20
8EF-S05	100% Cashmere	160	46.21	7,393.60
8EF-S06	100% Cashmere	160	47.19	7,550.40
8EF-S07	100% Cashmere	160	50.80	8,128.00
8EF-S08	100% Cashmere	160	37.42	5,987.20
8EF-S09	100% Cashmere	160	50.90	8,144.00
8EF-S10	100% Cashmere	160	47.72	7,635.20
8EF-S11	70%Wool/30%Cashmere	160	34.79	5,566.40
8EF-S12	70%Wool/30%Cashmere	160	37.47	5,995.20
8EF-S13	70%Wool/30%Cashmere	160	43.02	6,883.20
8EF-S14	70%Wool/30%Cashmere	160	33.16	5,305.60
8EF-S15	70%Wool/30%Cashmere	160	58.05	9,288.00
8EF-S16	70%Wool/30%Cashmere	160	50.66	8,105.60
TOTAL		2,600		119,434.00

Country of origin: Made in China.
Delivery: By Sep.20th, 2018. on shipment about 50% of total
Payment: 50% by T/T after contract.
50% by L/C at shight.
Quantity: It is acceptable +/- 5%.
Remarks: The unit price can be changed on the base of revised sample weight.

Shipper: Shanghai Kaixing Industry and Trade Co., Ltd.
Address: Room 1045, No 35 Rijing Road, Wai Gao Qiao Free Zone Shanghai China

Signed by: Shanghai Kaixing Industry and Trade Co., Ltd. Pearltex Korea

1. COMPANY NAME: SHANGHAI KAIXING INDUSTRY AND TRADE CO.,LTD.
2. COMPANY ADDRESS:ROOM 1045 NO.35 RIJING ROAD WAI GAO QIAO FREE ZONE SHANGHAI CHINA
   POST CODE: 200085, PHONE NO.021-63577141
3. BANK NAME:BANK OF COMMUNICATIONS SHANGHAI HONGKOU SUB-BRANCH
4. BANK ADDRESS:NO.263 SIPING ROAD HONGKOU DISTRICT SHANGHAI CHINA
5. BRANCH NAME: BANK OF COMMUNICATIONS SHANGHAI HONGKOU SUB-BRANCH
6. ACCOUNT NO.310066030146300008779
7. SWIFT CODE: COMMCNSHSHI
8. BENEFICIARY: SHANGHAI KAIXING INDUSTRY AND TRADE CO.,LTD.

This is a translated invoice provided to us from the client in which we used to model the Sales function for the company. There are several unique features in this invoice that are not commonly found that we used to model our Sales function such as the unique payment terms at the bottom and the yarn composition.

User View - Shipping

上 海 名 博 服 饰 有 限 公 司

SHANGHAI MINGBO APPAREL CO., LTD.

No. 329, Xingdacun changxin, Yexie Zhen, Songjiang Qu, Shanghai, China

TEL: 86-1590-625-617 FAX: 86-(0)21-5762-7020

Brand	a. t. corner
Product Detail Number	6352714
Packages Included	2842, 7931
Quantity	1074
Total Carton No.	14
Shipping No.	100734523748
Buyer	YOONHA KNIT
Address	15 Yeongjung-ro, Yeongdeungpo-dong, Yeongdeungpo-gu, Seoul, South Korea

Date: 2018. 08. 09.

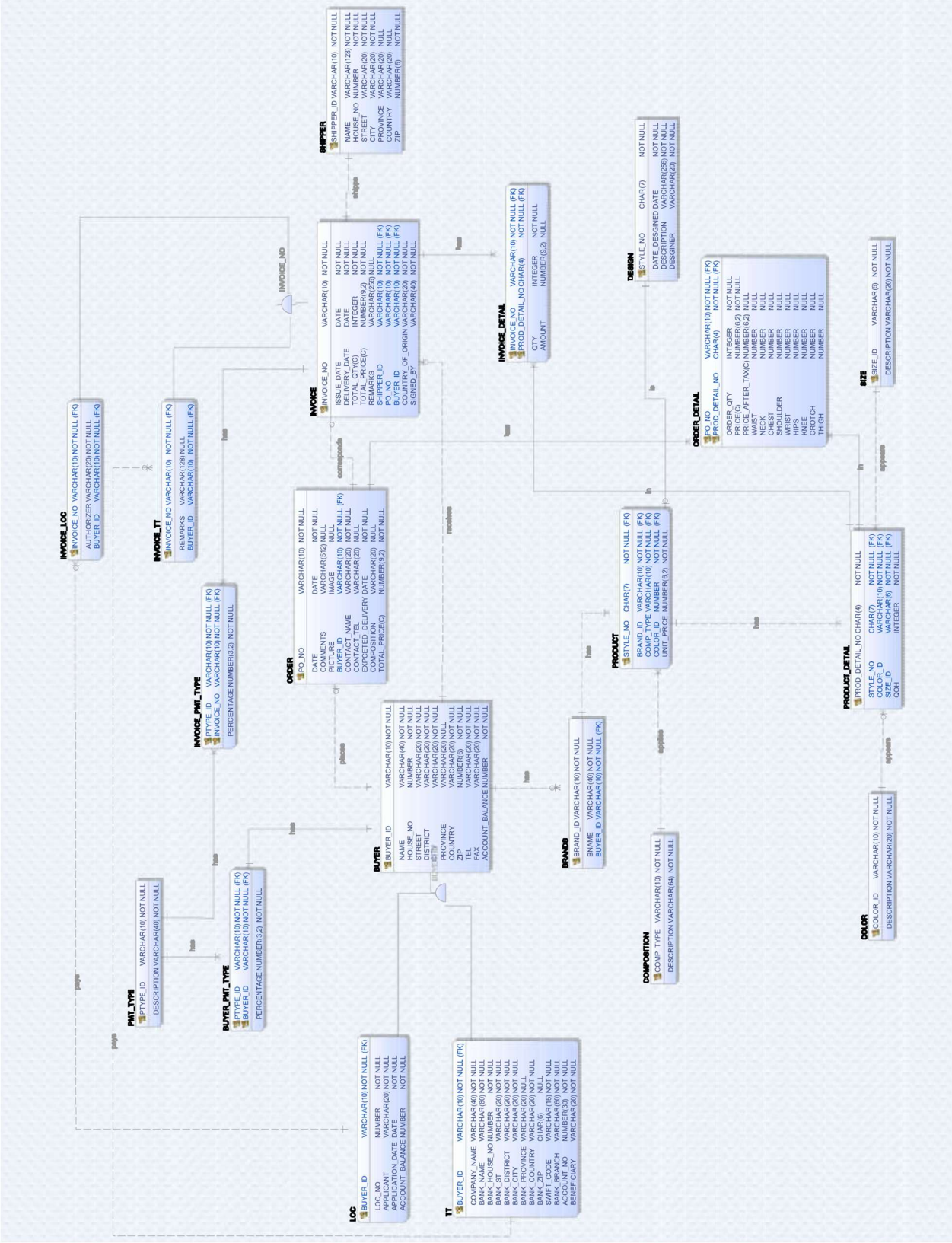
(Unit: PC, KG, CM)

Brand	Product Detail Number	Carton No.	Coulor	S	M	Total	Gross weight (kg)	Net weight (kg)	Measurement (cbm)
a. t. corner	6352714	1	BK	80		80	15.2	14.2	0.084
		2	BK	80		80	15.2	14.2	0.084
		3	BK	80		80	15.1	14.1	0.084
		4	BK		80	80	16	15	0.084
		5	BK		64	64	13	12	0.084
		6	G2	80		80	15.5	14.5	0.084
		7	G2	80		80	15.5	14.5	0.084
		8	G2		80	80	16.4	15.4	0.084
		9	BK	21		21	16.7	15.7	0.084
			G2	36	28	64			
		10	P2	80		80	15.7	14.7	0.084
		11	P2	57		57	11.5	10.5	0.084
		12	P2		73	73	15	14	0.084
		13	B1	80		80	15	14	0.084
		14	B1	20	55	75	15	14	0.084
Total				694	380	1074	211	197	1.176

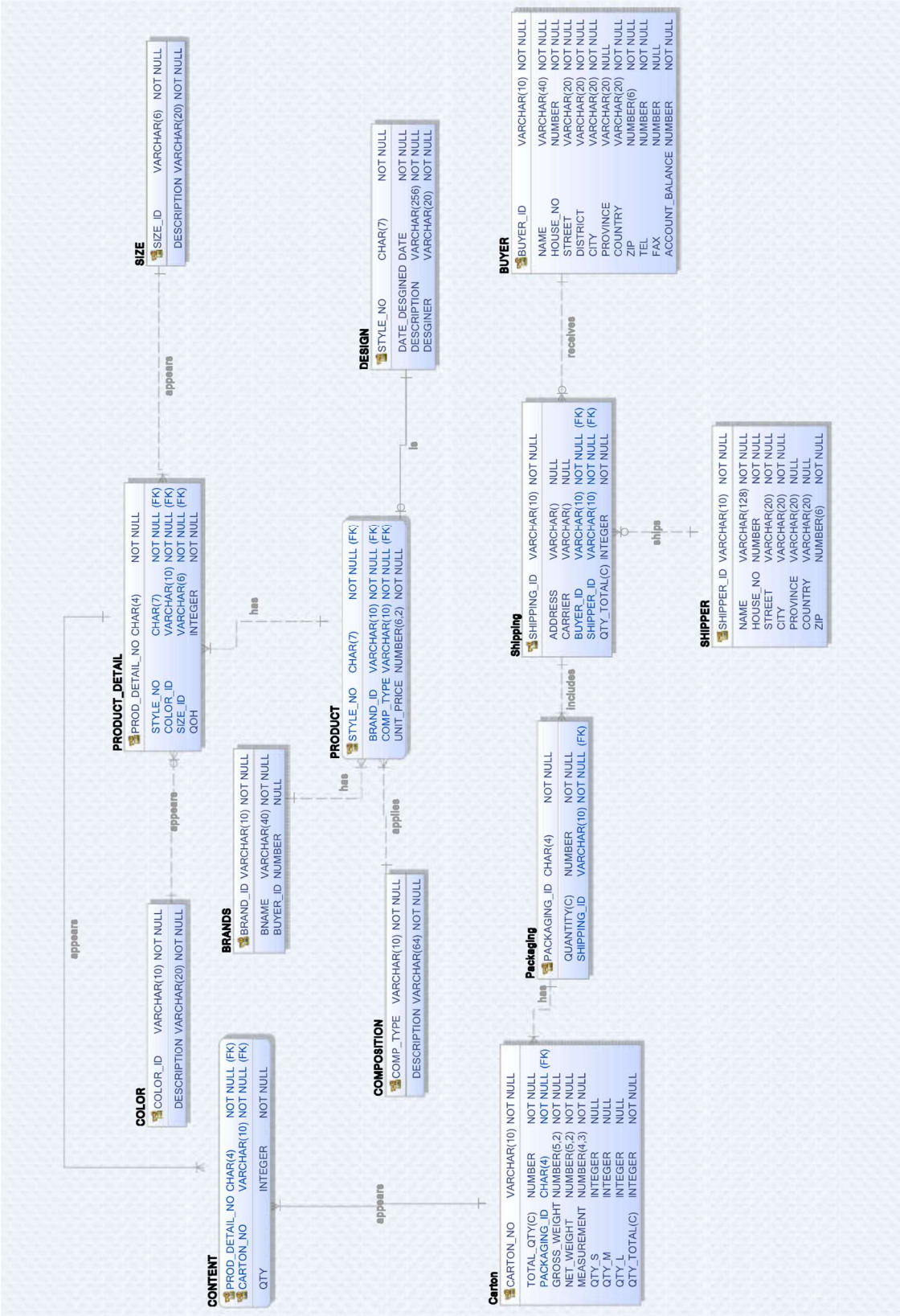
Carrier China Post

This is a shipping document provided to us from the client that we modeled our shipping user view after. There are several noteable key items here that helped us make our DDLs and ERD such as the Product Detail Number, shipping number, and the various attributes like carton, buyer and carton numbers.

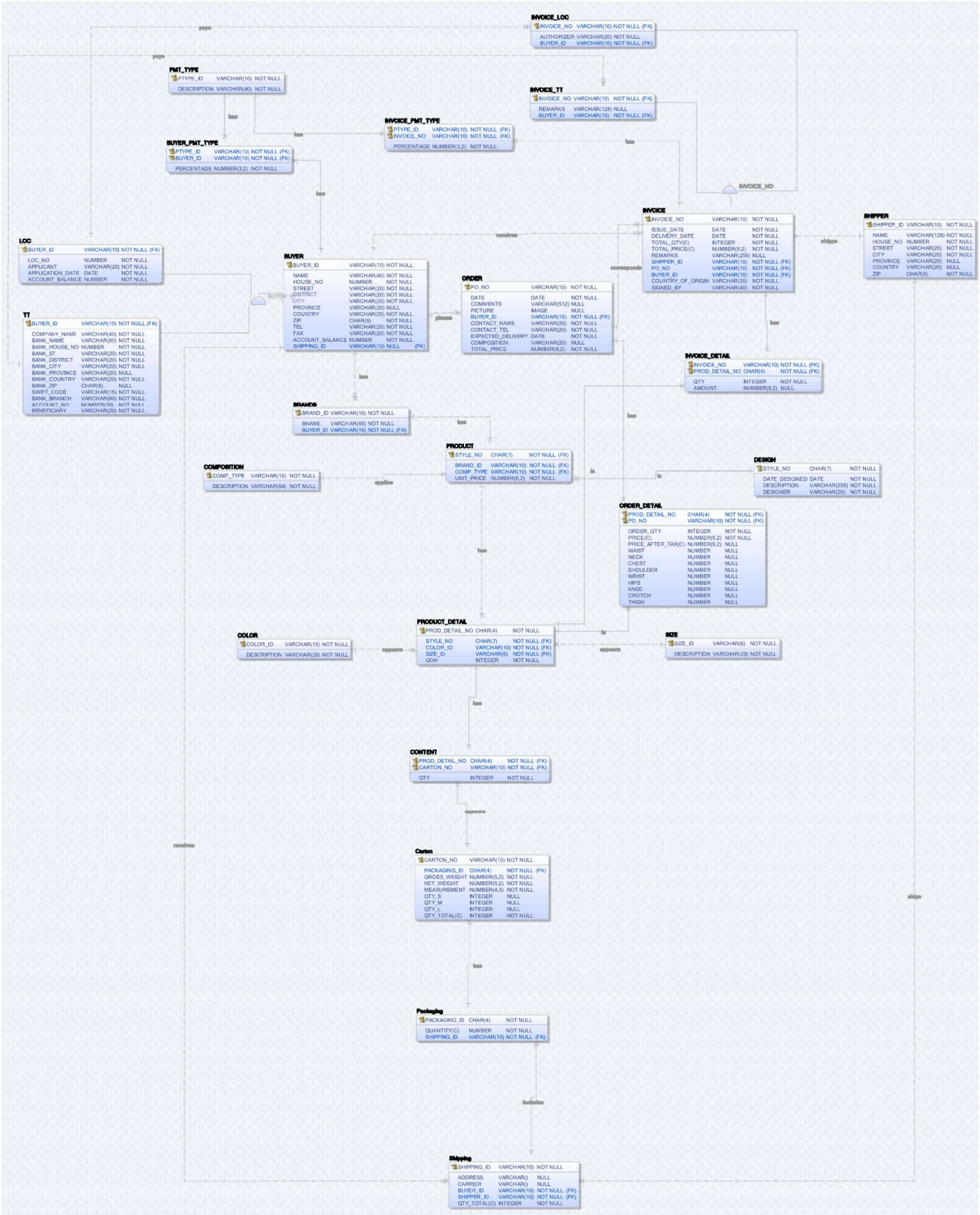
Sales Entity Relationship Diagram



Shipping Entity Relationship Diagram



Conceptual Data Model

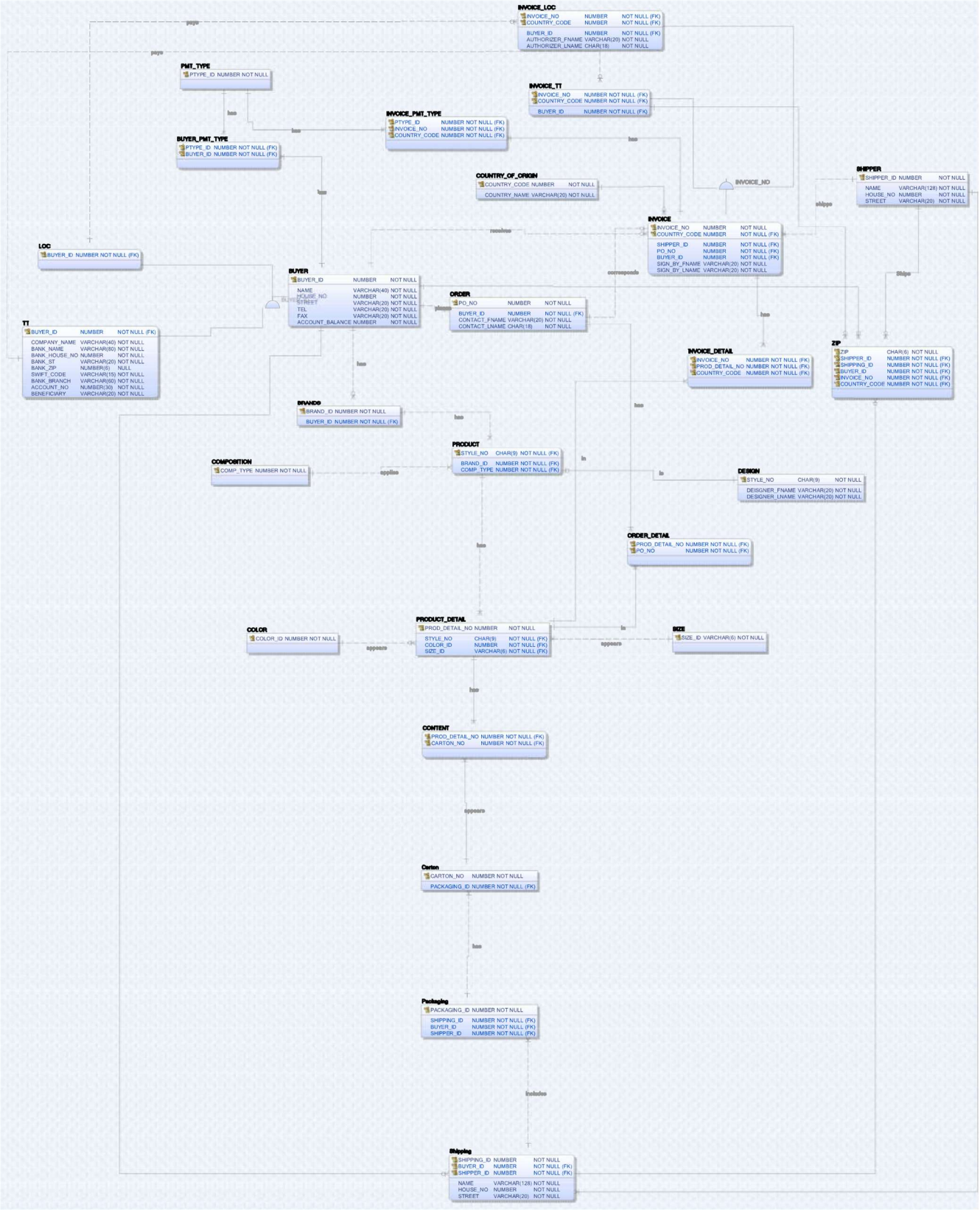


Domain Definitions, Business Rules and Constraints

ATTRIBUTE	DOMAIN NAME	DESCRIPTION	DOMAIN
LOC_NO	Payment Type	Payment type number	INTEGER: 6
ACCOUNT_BALANCE	Account Information	Balance in an Account	INTEGER: 40
ACCOUNT_NO	Seller/Buyer Information	Account Number	INTEGER: 30
ADDRESS	Seller/Buyer Information	Address for buyer	INTEGER
AMOUNT	Account Information	Amount due to seller	INTEGER: 40
APPLICANT	Seller/Buyer Information	Name of applicant for a specific payment type	CHARACTER: 20
APPLICATION_DATE	Account Information	Date applied	DATE: MM/DD/YY
AUTHORIZER	Account Information	Who authorized applicant	CHARACTER: 20
BANK_BRANCH	Bank Information	Where the bank is located	CHARACTER: 60
BANK_CITY	Bank Information	Where the bank is located	CHARACTER: 20
BANK_COUNTRY	Bank Information	Where the bank is located	CHARACTER: 20
BANK_DISTRICT	Bank Information	Where the bank is located	CHARACTER: 20
BANK_HOUSE_NO	Bank Information	Where the bank is located	INTEGER: 20
BANK_NAME	Bank Information	Bank's name	CHARACTER: 80
BANK_PROVINCE	Bank Information	Where the bank is located	CHARACTER: 20
BANK_ST	Bank Information	Where the bank is located	CHARACTER: 20
BANK_ZIP	Bank Information	Where the bank is located	INTEGER: 10
BENEFICIARY	Seller/Buyer Information	Who is receiving the money	CHARACTER: 20
BNAME	Buyer Information	Name of person receiving the money	CHARACTER: 40
BRAND_ID	Product Details	Brand of product being made	INTEGER: 10
BUYER_ID	Buyer Information	Who the buyer is	INTEGER: 10
CARRIER	Shipper Information	How the goods will be delivered	CHARACTER: 30
CARTON_NO	Shipper Information	Unique identifier to track contents of an order	INTEGER: 10
CHEST	Product Details	Item dimmensions	INTEGER: 10
CITY	Buyer Information	City the seller is located in	CHARACTER: 20
COLOR_ID	Product Details	Unqiue color identifier	INTEGER: 10
COMP_TYPE	Product Details	The material the goods are made of	CHARACTER: 10
COMPANY_NAME	Seller Information	The name of the company buying the goods	CHARACTER: 40
COUNTRY	Seller/Buyer Information	Where a company is located	CHARACTER: 20
COUNTRY_OF_ORIGIN	Seller/Buyer Information	Where a company is located	CHARACTER: 20
CROTCH	Product Details	Item dimmensions	INTEGER: 10
DATE_DESIGNED	Product Details	When the order was created by manufacturer	DATE: MM/DD/YY
DELIVERY_DATE	Product Details	When the order was delivered	DATE: MM/DD/YY
DESIGNER	Product Details	Who designed the goods that were made	CHARACTER: 20
DISTRICT	Seller/Buyer Information	Where a company is located	CHARACTER: 20
FAX	Seller/Buyer Information	How to reach seller or buyer	INTEGER: 20
GROSS_WEIGHT	Product Details	Weight of the product being shipped out	INTEGER: 5
HIPS	Product Details	Item dimmensions	INTEGER: 10
HOUSE_NO	Buyer Information	Where a company is located	INTEGER: 20
INVOICE_NO	Product Details	Identifying how much a buyer owes a seller	INTEGER: 10
ISSUE_DATE	Product Details	When the order was created by manufacturer	DATE: MM/DD/YY
KNEE	Product Details	Item dimmensions	INTEGER: 10
MEASUREMENT	Product Details	Item dimmensions	INTEGER: 40
NAME	Seller/Buyer Information	Name of buyer or shipper	CHARACTER: 50
NECK	Product Details	Item dimmensions	INTEGER: 10
NET_WEIGHT	Product Details	Weight of all items being shipped out	INTEGER: 5
ORDER_QTY	Product Details	How much of an item was made	INTEGER: 10
PACKAGING_ID	Product Details	Package tracker to track order	INTEGER: 4
PO_NO	Product Details	Identifier for a purchase order	INTEGER: 10

For Shanghai Kai Xing, orders are placed sparsely throughout the month. The business rules are that an order can have many styles and each style can have many colors and sizes. We modeled accordingly to these unique business rules. For all primary keys and foreign keys, constraints are made so that it is easier to enable/disable the keys. This will enhance the integrity of the database. For all primary keys and foreign keys, ‘primary key’ constraints and ‘references’ constraints are made so that it is easier to enable/disable the keys. This will enhance the integrity of the database. ‘Not null’ constraints are made for attributes that require a not null value for the same reason. In addition, ‘check’ constrains are being places on values like quantities and prices to ensure that a value will not fall below zero. Another check is being placed on the attributes Qty_S, Qty_M and Qty_L in the Carton table as the attributes separately do not have a ‘not null’ constraints. However, at least one of these values should return as ‘not null’ in every record as it would mean that a carton would appear empty before shipping it. Lastly, a ‘check’ constraint in the Invoice table is created for the attributes Delivery_date and Issue_date to ensure that the delivery will take place after it got issued.

Consolidated Data Mode in 3NF



Process versus Entity Matrix and CUD/CRUD ratio analysis

PROCESS VS ENTITY MATRIX

| Function \ Entity | INVOICE | | | | INVOICE LOC | | | | INVOICE TT | | | | INVOICE_PMT_TYPE | | | | INVOICE_DETAIL | | | | SHIPPER | | | | PMT_TYPE | | | | BUYER_PMT_TYPE | | | | LOC | | | | TT | | | | BUYER | | | | ORDER | | | | BRAND | | | | ORDER_DETAIL | | | | PRODUCT | | | | COMPOSITION | | | | DESIGN | | | | PRODUCT_DETAIL | | | | COLOR | | | | SIZE | | | | CONTENT | | | | CARTON | | | | PACKAGING | | | | SHIPPING | | | | Peak/hr | Total | % of grand total | CUD | CRUD | CUD Ratio |
|---|---|---

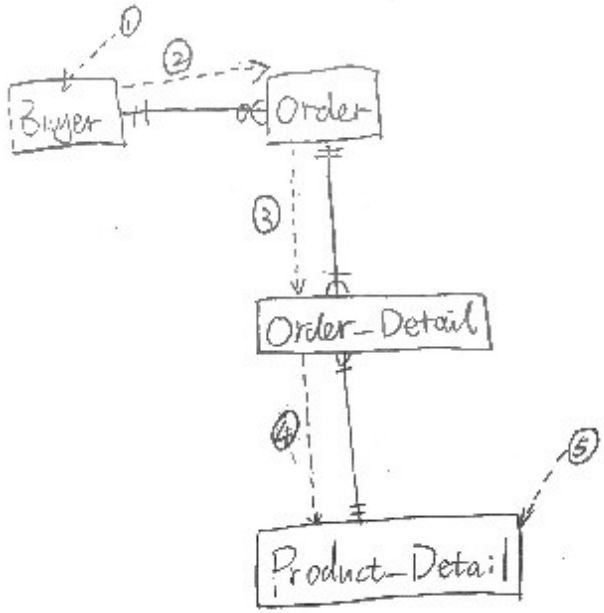
Transaction Analysis Forms (1) – CREATE ORDER (PO001)

TRANSACTION ANALYSIS FORM

TRANSACTION NUMBER: PO001
TRANSACTION NAME: Create Order

TRANSACTION VOLUME:
AVERAGE: 0.5 /HR PEAK: 2 /HR

TRANSACTION MAP:



NO.	TABLE NAME	ACCESS	NO. OF REFS PER TRANS	NO. OF REFS PER PERIOD
1	Buyer	R	1	6
2	order	C	1	6
3	order Detail	C	16	96
4	Product Detail	R	16	96
5				
6				
7				
8				
9				
10				
	TOTAL REFERENCES:		34	204

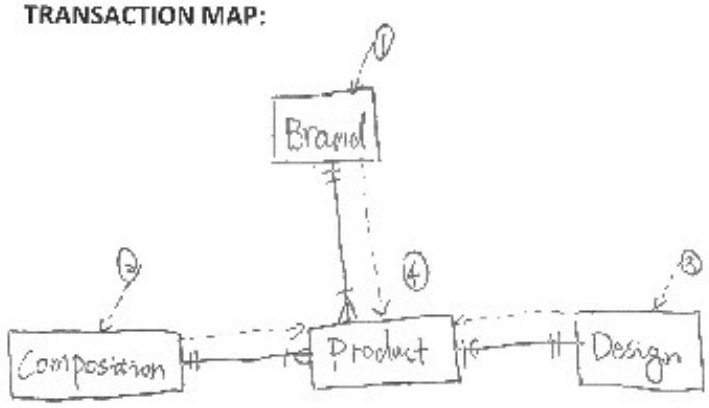
Transaction Analysis Form (2) – CREATE PRODUCT (PR001)

TRANSACTION ANALYSIS FORM

TRANSACTION NUMBER: PR001
TRANSACTION NAME: Create Product

TRANSACTION VOLUME:
AVERAGE: 0.2 **/HR** **PEAK:** 2 **/HR**

TRANSACTION MAP:



NO.	TABLE NAME	ACCESS	NO. OF REFS PER TRANS	NO. OF REFS PER PERIOD
1	Brand	R	1	5
2	Composition	R	1	10
3	Design	R	1	10
4	Product	C	1	10
5				
6				
7				
8				
9				
10				
	TOTAL REFERENCES:		5	35

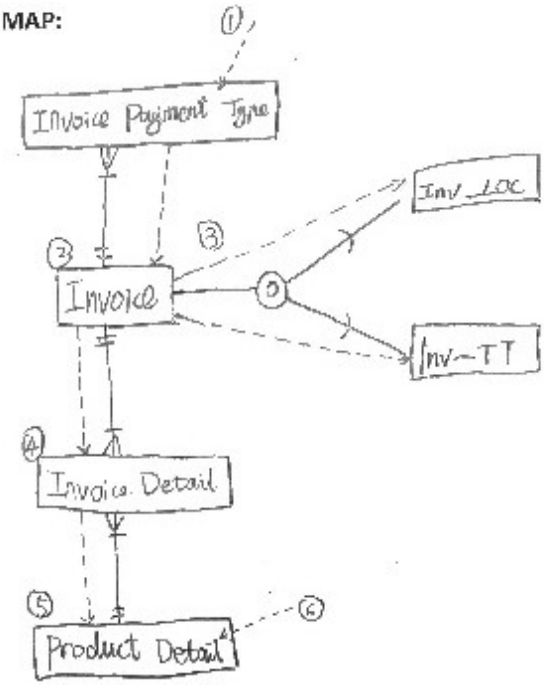
Transaction Analysis Form (3) – CREATE INVOICE (IN098)

TRANSACTION ANALYSIS FORM

TRANSACTION NUMBER: IN098
TRANSACTION NAME: Create Invoice

TRANSACTION VOLUME:
AVERAGE: 0.5 /HR **PEAK:** 2 /HR

TRANSACTION MAP:



NO.	TABLE NAME	ACCESS	NO. OF REFS PER TRANS	NO. OF REFS PER PERIOD
1	Invoice Payment Type	R	1	10
2	Invoice	C	1	10
3	Inv_LOC	C	0.74	7.4
4	Inv-TT	C	0.74	7.4
5	Invoice Detail	C	12	120
6	Product Detail	R	16	160
7				
8				
9				
10				
	TOTAL REFERENCES:		35.48	354.8

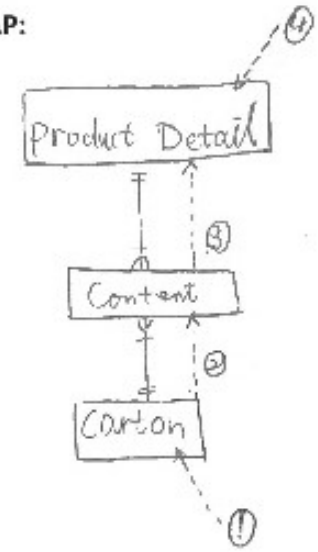
Transaction Analysis Form (4) – CREATE CONTENT (CT003)

TRANSACTION ANALYSIS FORM

TRANSACTION NUMBER: CT003
TRANSACTION NAME: Create Content

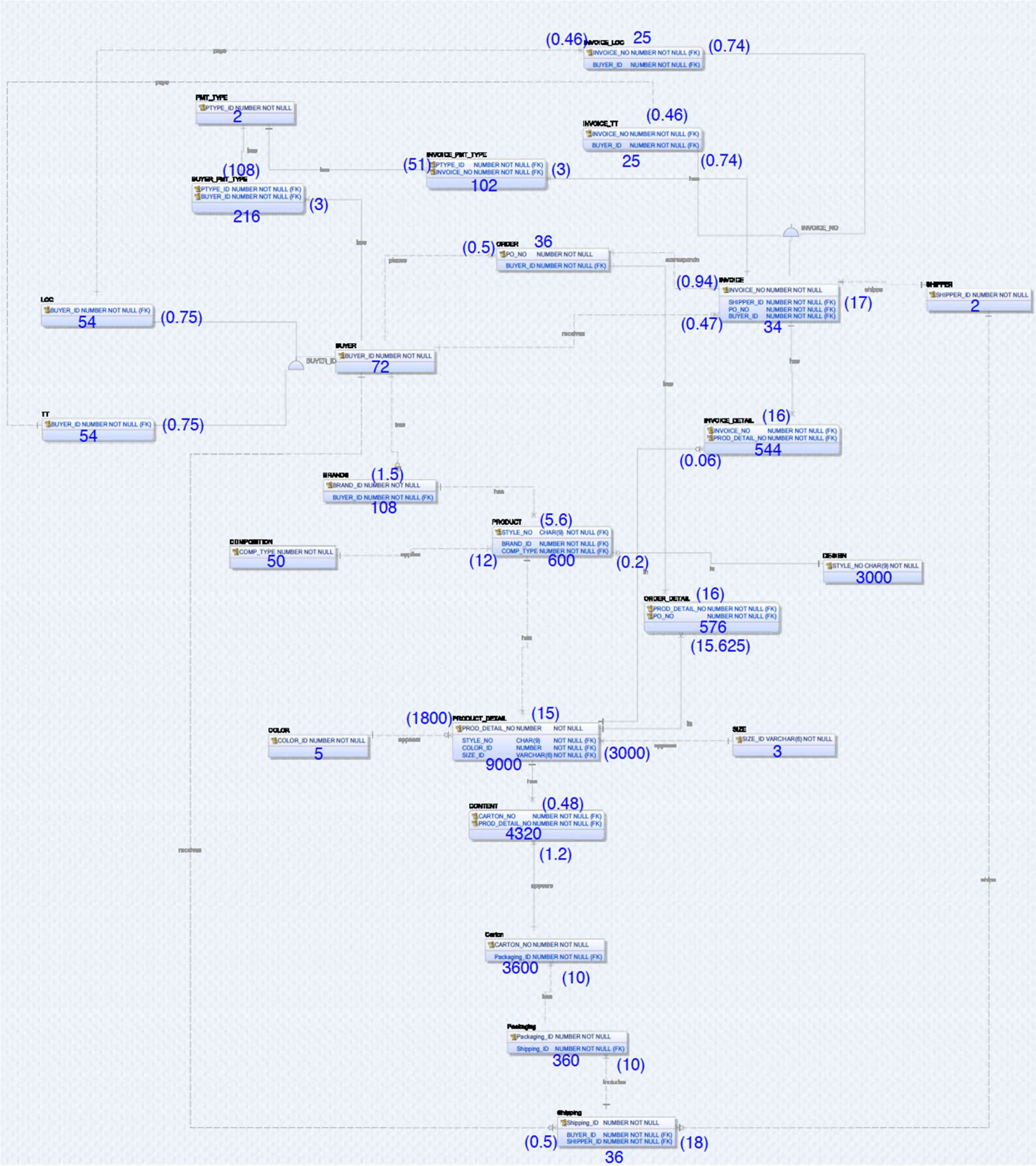
TRANSACTION VOLUME:
AVERAGE: 1 /HR **PEAK:** 24 /HR

TRANSACTION MAP:



NO.	TABLE NAME	ACCESS	NO. OF REFS PER TRANS	NO. OF REFS PER PERIOD
1	Carton	R	1	20
2	Content	C	1.2	24
3	product Detail	R	1.2	24
4				
5				
6				
7				
8				
9				
10				
	TOTAL REFERENCES:		3.4	68

Composite Usage Map



Data Definition Language

```
spool /Users/PingAu/Desktop/Final_DSO435.lst;

SET ECHO ON;
SET PAGESIZE 60;
SET LINESIZE 100;
SET AUTOCOMMIT ON;

ALTER SESSION SET RECYCLEBIN=OFF;
COLUMN PRICE FORMAT $9,999.99;
COLUMN TOTAL_PRICE FORMAT $99,999,999.99;
COLUMN ACCOUNT_BALANCE FORMAT $99,999,999.99;
COLUMN PRICE_AFTER_TAX FORMAT $9,999.99;
COLUMN UNIT_PRICE FORMAT $9,999.99;

ALTER SESSION SET NLS_DATE_FORMAT = 'mm-dd-yy';

DROP TRIGGER ID_INSERT;

DROP TABLE INVOICE_PMT_TYPE CASCADE CONSTRAINTS;
DROP TABLE INVOICE_DETAIL CASCADE CONSTRAINTS;
DROP TABLE INVOICE_TT CASCADE CONSTRAINTS;
DROP TABLE INVOICE_LOC CASCADE CONSTRAINTS;
DROP TABLE INVOICE CASCADE CONSTRAINTS;
DROP CLUSTER INVOICE_SUB;
```

```
DROP TABLE PRODUCT CASCADE CONSTRAINTS;
DROP TABLE ORDER_DETAIL CASCADE CONSTRAINTS;
DROP TABLE "CONTENT" CASCADE CONSTRAINTS;
DROP TABLE PRODUCT_DETAIL CASCADE CONSTRAINTS;
DROP TABLE CARTON CASCADE CONSTRAINTS;
DROP TABLE PACKAGING CASCADE CONSTRAINTS;
DROP TABLE SHIPPING CASCADE CONSTRAINTS;
DROP TABLE BRAND CASCADE CONSTRAINTS;
DROP TABLE "ORDER" CASCADE CONSTRAINTS;
DROP TABLE BUYER_PMT_TYPE CASCADE CONSTRAINTS;
DROP TABLE SHIPPER CASCADE CONSTRAINTS;
DROP TABLE LOC CASCADE CONSTRAINTS;
DROP TABLE TT CASCADE CONSTRAINTS;
DROP TABLE BUYER CASCADE CONSTRAINTS;
DROP CLUSTER BUYER_SUB;
DROP TABLE PMT_TYPE CASCADE CONSTRAINTS;
DROP TABLE DESIGN CASCADE CONSTRAINTS;
DROP TABLE "SIZE" CASCADE CONSTRAINTS;
DROP TABLE COLOR CASCADE CONSTRAINTS;
DROP TABLE COMPOSITION CASCADE CONSTRAINTS;
DROP VIEW INVOICE_DETAIL_AMOUNT;

CREATE CLUSTER BUYER_SUB (BUYER_SUB_CLUSTER VARCHAR2(10));
CREATE CLUSTER INVOICE_SUB (INVOICE_SUB_CLUSTER VARCHAR2(10));
CREATE INDEX BUYER_SUB_CLUSTER_INDEX ON CLUSTER BUYER_SUB;
CREATE INDEX INVOICE_SUB_CLUSTER_INDEX ON CLUSTER INVOICE_SUB;
```

```
CREATE TABLE COMPOSITION (
   COMP_TYPE          VARCHAR2(10)    CONSTRAINT COMPOSITION_PK PRIMARY KEY,
   DESCRIPTION        VARCHAR2(64)    CONSTRAINT COMPOSITION_DESC_NULL NOT
NULL
   );

CREATE TABLE COLOR (
   COLOR_ID           VARCHAR2(10)    CONSTRAINT COLOR_PK PRIMARY KEY,
   DESCRIPTION        VARCHAR2(20)    CONSTRAINT COLOR_DESC_NULL NOT NULL
   );

CREATE TABLE "SIZE" (
   SIZE_ID            VARCHAR2(6)     CONSTRAINT SIZE_PK PRIMARY KEY,
   DESCRIPTION        VARCHAR2(20)    CONSTRAINT SIZE_DESC_NULL NOT NULL
   );

CREATE TABLE DESIGN (
   STYLE_NO           CHAR(7)         CONSTRAINT DESIGN_PK PRIMARY KEY,
   DATE_DESIGNED      DATE            CONSTRAINT DESIGN_DATE_DESIGNED_NULL NOT
NULL,
   DESCRIPTION        VARCHAR2(256)   CONSTRAINT DESIGN_DESC_NULL NOT NULL,
   DESIGNER           VARCHAR2(20)    CONSTRAINT DESIGN_DESIGNER_NULL NOT NULL
   );

CREATE TABLE PMT_TYPE (
```

```
    PTYPE_ID          VARCHAR2(10)    CONSTRAINT PMT_TYPE_PK PRIMARY KEY,
    DESCRIPTION       VARCHAR2(40)    CONSTRAINT PMT_TYPE_DESC_NULL NOT NULL
    );

CREATE TABLE BUYER (
    BUYER_ID          VARCHAR2(10)    CONSTRAINT BUYER_PK PRIMARY KEY,
    NAME              VARCHAR2(40)    CONSTRAINT BUYER_NAME_NULL NOT NULL,
    HOUSE_NO          NUMBER(10)      CONSTRAINT BUYER_HOUSE_NO_NULL NOT NULL,
    STREET            VARCHAR2(20)    CONSTRAINT BUYER_STREET_NULL NOT NULL,
    DISTRICT          VARCHAR2(20)    CONSTRAINT BUYER_DISTRICT_NULL NOT NULL,
    CITY              VARCHAR2(20)    CONSTRAINT BUYER_CITY_NULL NOT NULL,
    PROVINCE          VARCHAR2(20),
    COUNTRY           VARCHAR2(20)    CONSTRAINT BUYER_COUNTRY_NULL NOT NULL,
    ZIP               NUMBER(6),
    TEL               NUMBER(12)      CONSTRAINT BUYER_TEL_NULL NOT NULL,
    FAX               NUMBER(12),
    ACCOUNT_BALANCE   NUMBER          CONSTRAINT
BUYER_ACCOUNT_BALANCE_NULL NOT NULL)

    CLUSTER BUYER_SUB (BUYER_ID);

CREATE TABLE SHIPPER (
    SHIPPER_ID        VARCHAR2(10)    CONSTRAINT SHIPPER_PK PRIMARY KEY,
    NAME              VARCHAR2(128)   CONSTRAINT SHIPPER_NAME_NULL NOT NULL,
    HOUSE_NO          NUMBER(10)      CONSTRAINT SHIPPER_HOUSE_NO_NULL NOT NULL,
    STREET            VARCHAR2(40)    CONSTRAINT SHIPPER_STREET_NULL NOT NULL,
```

```
   CITY            VARCHAR2(20)    CONSTRAINT SHIPPER_CITY_NULL NOT NULL,
   PROVINCE        VARCHAR2(20),
   COUNTRY         VARCHAR2(20),
   ZIP             NUMBER(6)       CONSTRAINT SHIPPER_ZIP_NULL NOT NULL
   );

CREATE TABLE TT (
   BUYER_ID        VARCHAR2(10)    CONSTRAINT TT_PK PRIMARY KEY,
   COMPANY_NAME    VARCHAR2(40)    CONSTRAINT TT_COMPANY_NAME_NULL NOT
NULL,
   BANK_NAME       VARCHAR2(80)    CONSTRAINT TT_BANK_NAME_NULL NOT NULL,
   BANK_HOUSE_NO   NUMBER(10)      CONSTRAINT TT_BANK_HOUSE_NO_NULL NOT
NULL,
   BANK_ST         VARCHAR2(20)    CONSTRAINT TT_BANK_ST_NULL NOT NULL,
   BANK_DISTRICT   VARCHAR2(20)    CONSTRAINT TT_BANK_DISTRICT_NULL NOT
NULL,
   BANK_CITY       VARCHAR2(20)    CONSTRAINT TT_BANK_CITY_NULL NOT NULL,
   BANK_PROVINCE   VARCHAR2(20),
   BANK_COUNTRY    VARCHAR2(20)    CONSTRAINT TT_BANK_COUNTRY_NULL NOT
NULL,
   BANK_ZIP        NUMBER(6),
   SWIFT_CODE      VARCHAR2(15)    CONSTRAINT TT_SWIFT_CODE_NULL NOT NULL,
   BANK_BRANCH     VARCHAR2(60)    CONSTRAINT TT_BANK_BRANCH_NULL NOT
NULL,
   ACCOUNT_NO      NUMBER(30)      CONSTRAINT TT_ACCOUNT_NO_NULL NOT NULL,
   BENEFICIARY     VARCHAR2(20)    CONSTRAINT TT_BENEFICIARY_NULL NOT NULL
```

```
)
CLUSTER BUYER_SUB (BUYER_ID);

CREATE TABLE LOC (
BUYER_ID          VARCHAR2(10)    CONSTRAINT LOC_PK PRIMARY KEY,
LOC_NO            VARCHAR2(10)    CONSTRAINT LOC_LOC_NO_NULL NOT NULL,
APPLICANT         VARCHAR2(20)    CONSTRAINT LOC_APPLICANT_NULL NOT NULL,
APPLICATION_DATE    DATE          CONSTRAINT LOC_APPLICATION_DATE_NULL NOT
NULL
)
CLUSTER BUYER_SUB (BUYER_ID);

CREATE TABLE BUYER_PMT_TYPE (
PTYPE_ID          VARCHAR2(10)    CONSTRAINT BUYER_PMT_TYPE_PTYPE_ID_FK
REFERENCES PMT_TYPE,
BUYER_ID          VARCHAR2(10)    CONSTRAINT BUYER_PMT_TYPE_BUYER_ID_FK
REFERENCES BUYER,
PERCENTAGE        NUMBER(3,2)     CONSTRAINT
BUYER_PMT_TYPE_PERCENTAGE_NULL NOT NULL,
PRIMARY KEY(PTYPE_ID, BUYER_ID)
);

CREATE TABLE "ORDER" (
PO_NO             VARCHAR2(10)    CONSTRAINT ORDER_PK PRIMARY KEY,
"DATE"            DATE            CONSTRAINT ORDER_DATE_NULL NOT NULL,
COMMENTS          VARCHAR2(512),
```

```
   PICTURE            BFILE,
   BUYER_ID           VARCHAR2(10)    CONSTRAINT ORDER_BUYER_ID_NULL NOT NULL
                                      CONSTRAINT ORDER_BUYER_ID_FK REFERENCES BUYER,
   CONTACT_NAME       VARCHAR2(20)    CONSTRAINT ORDER_CONTACT_NAME_NULL
NOT NULL,
   CONTACT_TEL        VARCHAR2(20)    CONSTRAINT ORDER_CONTACT_TELL_NULL NOT
NULL,
   EXPECTED_DELIVERY  DATE            CONSTRAINT ORDER_EXPECTED_DELIVERY_NULL
NOT NULL,
   COMPOSITION        VARCHAR2(20),
   TOTAL_PRICE        NUMBER(10,2)    CONSTRAINT ORDER_TOTAL_PRICE_NULL NOT
NULL
                                      CONSTRAINT ORDER_TOTAL_PRICE_CK CHECK(TOTAL_PRICE >= 0)
   );

CREATE TABLE BRAND (
   BRAND_ID           VARCHAR2(10)    CONSTRAINT BRAND_PK PRIMARY KEY,
   BNAME              VARCHAR2(40)    CONSTRAINT BRAND_BNAME_NULL NOT NULL,
   BUYER_ID           VARCHAR2(10)    CONSTRAINT BRAND_BUYER_ID_NULL NOT NULL
                                      CONSTRAINT BRAND_BUYER_ID_FK REFERENCES BUYER
   );

CREATE TABLE SHIPPING (
   SHIPPING_ID        VARCHAR2(10)    CONSTRAINT SHIPPING_PK PRIMARY KEY,
   ADDRESS            VARCHAR2(60),
   CARRIER            VARCHAR2(40),
```

```
   BUYER_ID          VARCHAR2(10)    CONSTRAINT SHIPPING_BUYER_ID_NULL NOT NULL
                                     CONSTRAINT SHIPPING_BUYER_ID_FK REFERENCES BUYER,
   SHIPPER_ID        VARCHAR2(10)    CONSTRAINT SHIPPING_SHIPPER_ID_NULL NOT
NULL
                                     CONSTRAINT SHIPPING_SHIPPER_ID_FK REFERENCES SHIPPER,
   QTY_TOTAL         INTEGER         CONSTRAINT SHIPPING_QTY_TOTAL_NULL NOT NULL
                                     CONSTRAINT SHIPPING_QTY_TOTAL_CK CHECK(QTY_TOTAL >= 0)
   );

CREATE TABLE PACKAGING (
   PACKAGING_ID      CHAR(4)    CONSTRAINT PACKAGING_PK PRIMARY KEY,
   QUANTITY          NUMBER(10)      CONSTRAINT PACKAGING_QUANTITY_NULL NOT
NULL,
   SHIPPER_ID        VARCHAR2(10)    CONSTRAINT PACKAGING_SHIPPER_ID_NULL NOT
NULL
                                     CONSTRAINT PACKAGING_SHIPPER_ID_FK REFERENCES SHIPPER
   );

CREATE TABLE CARTON (
   CARTON_NO         VARCHAR2(10)    CONSTRAINT CARTON_PK PRIMARY KEY,
   TOTAL_QTY         NUMBER(10)      CONSTRAINT CARTON_TOTAL_QTY_NULL NOT
NULL,
   PACKAGING_ID      CHAR(4)         CONSTRAINT CARTON_PACKAGING_ID_NULL NOT
NULL
                                     CONSTRAINT CARTON_PACKAGING_ID_FK REFERENCES
PACKAGING,
```

```
   GROSS_WEIGHT       NUMBER(5,2)     CONSTRAINT CARTON_GROSS_WEIGHT_NULL NOT
NULL,
   NET_WEIGHT         NUMBER(5,2)     CONSTRAINT CARTON_NET_WEIGHT_NULL NOT
NULL,
   MEASUREMENT        NUMBER(4,3)     CONSTRAINT CARTON_MEASUREMENT_NULL
NOT NULL,
   QTY_S              INTEGER         CONSTRAINT CARTON_QTY_S_CK CHECK(QTY_S >= 0),
   QTY_M              INTEGER         CONSTRAINT CARTON_QTY_M_CK CHECK(QTY_M >= 0),
   QTY_L              INTEGER         CONSTRAINT CARTON_QTY_L_CK CHECK(QTY_L >= 0)
   );

CREATE TABLE PRODUCT_DETAIL (
   PROD_DETAIL_NO     CHAR(4)         CONSTRAINT PRODUCT_DETAIL_PK PRIMARY KEY,
   STYLE_NO           CHAR(7)         CONSTRAINT PRODUCT_DETAIL_STYLE_NO_NULL NOT
NULL
                                      CONSTRAINT PRODUCT_DETAIL_STYLE_NO_FK REFERENCES
DESIGN,
   COLOR_ID           VARCHAR2(10)    CONSTRAINT PRODUCT_DETAIL_COLOR_ID_NULL
NOT NULL
                                      CONSTRAINT PRODUCT_DETAIL_COLOR_ID_FK REFERENCES
COLOR,
   SIZE_ID            VARCHAR2(6)     CONSTRAINT PRODUCT_DETAIL_SIZE_ID_NULL NOT
NULL
                                      CONSTRAINT PRODUCT_DETAIL_SIZE_ID_FK REFERENCES "SIZE",
   QOH                NUMBER(6)       CONSTRAINT PRODUCT_DETAIL_QOH_NULL NOT NULL
   );
```

```
CREATE TABLE "CONTENT" (
   PROD_DETAIL_NO      CHAR(4)        CONSTRAINT CONTENT_PROD_DETAIL_NO_FK
REFERENCES PRODUCT_DETAIL,
   CARTON_NO           VARCHAR2(10)    CONSTRAINT CONTENT_CARTON_NO_FK
REFERENCES CARTON,
   QTY                 INTEGER        CONSTRAINT CONTENT_QTY_NULL NOT NULL,
   PRIMARY KEY(PROD_DETAIL_NO, CARTON_NO)
   );

CREATE TABLE ORDER_DETAIL (
   PROD_DETAIL_NO      CHAR(4)        CONSTRAINT ORDER_DETAIL_PROD_DETAIL_NO_FK
REFERENCES PRODUCT_DETAIL,
   PO_NO               VARCHAR2(10)    CONSTRAINT ORDER_DETAIL_PO_NO_FK REFERENCES
"ORDER",
   ORDER_QTY           INTEGER        CONSTRAINT ORDER_DETAIL_ORDER_QTY_NULL NOT
NULL
                                CONSTRAINT ORDER_DETAIL_ORDER_QTY_CK CHECK(ORDER_QTY
>= 0),
   PRICE               NUMBER(6,2)     CONSTRAINT ORDER_DETAIL_PRICE_NULL NOT NULL
                                CONSTRAINT ORDER_DETAIL_PRICE_CK CHECK(PRICE >= 0),
   PRICE_AFTER_TAX     NUMBER(6,2)     CONSTRAINT ORDER_DETAIL_PRICE_TAX_CK
CHECK(PRICE_AFTER_TAX >= 0),
   WAIST               NUMBER(5),
   NECK                NUMBER(5),
   CHEST               NUMBER(5),
```

```
   SHOULDER          NUMBER(5),
   WRIST             NUMBER(5),
   HIPS              NUMBER(5),
   KNEE              NUMBER(5),
   CROTCH            NUMBER(5),
   THIGH             NUMBER(5),
   PRIMARY KEY(PROD_DETAIL_NO, PO_NO)
   );

CREATE TABLE PRODUCT (
   STYLE_NO          CHAR(7)        CONSTRAINT PRODUCT_STYLE_NO_PK PRIMARY KEY
                                    CONSTRAINT PRODUCT_STYLE_NO_FK REFERENCES DESIGN,
   BRAND_ID          VARCHAR2(10)   CONSTRAINT PRODUCT_BRAND_ID_FK REFERENCES
BRAND
                                    CONSTRAINT PRODUCT_BRAND_ID_NULL NOT NULL,
   COMP_TYPE         VARCHAR2(10)   CONSTRAINT PRODUCT_COMP_TYPE_FK
REFERENCES COMPOSITION
                                    CONSTRAINT PRODUCT_COMP_TYPE_NULL NOT NULL,
   UNIT_PRICE        NUMBER(6,2)    CONSTRAINT PRODUCT_UNIT_PRICE_NULL NOT NULL
                                    CONSTRAINT PRODUCT_UNIT_PRICE_CK CHECK(UNIT_PRICE >= 0)
   );

CREATE TABLE INVOICE (
   INVOICE_NO        VARCHAR2(10)   CONSTRAINT INVOICE_PK PRIMARY KEY,
   ISSUE_DATE        DATE           CONSTRAINT INVOICE_ISSUE_DATE_NULL NOT NULL,
```

```
   DELIVERY_DATE      DATE          CONSTRAINT INVOICE_DELIVERY_DATE_NULL NOT
NULL,
   TOTAL_QTY          INTEGER       CONSTRAINT INVOICE_TOTAL_QTY_NULL NOT NULL,
   TOTAL_PRICE        NUMBER(9,2)    CONSTRAINT INVOICE_TOTAL_PRICE_NULL NOT
NULL
                          CONSTRAINT INVOICE_TOTAL_PRICE_CK CHECK(TOTAL_PRICE >=
0),
   REMARKS            VARCHAR2(256),
   SHIPPER_ID         VARCHAR2(10)    CONSTRAINT INVOICE_SHIPPER_ID_NULL NOT NULL
                          CONSTRAINT INVOICE_SHIPPER_ID_FK REFERENCES SHIPPER,
   PO_NO              VARCHAR2(10)    CONSTRAINT INVOICE_PO_NO_NULL NOT NULL
                          CONSTRAINT INVOICE_PO_NO_FK REFERENCES "ORDER",
   BUYER_ID           VARCHAR2(10)    CONSTRAINT INVOICE_BUYER_ID_NULL NOT NULL
                          CONSTRAINT INVOICE_BUYER_ID_FK REFERENCES BUYER,
   COUNTRY_OF_ORIGIN   VARCHAR2(20)    CONSTRAINT
INVOICE_COUNTRY_OF_ORIGIN_NULL NOT NULL,
   SIGNED_BY          VARCHAR2(40)    CONSTRAINT INVOICE_SIGNED_BY_NULL NOT
NULL,
   CONSTRAINT INVOICE_DELIVERY_CK CHECK(DELIVERY_DATE>ISSUE_DATE))
   CLUSTER INVOICE_SUB (INVOICE_NO);

CREATE TABLE INVOICE_DETAIL (
   INVOICE_NO         VARCHAR2(10)    CONSTRAINT INVOICE_DETAIL_INVOICE_NO_FK
REFERENCES INVOICE,
   PROD_DETAIL_NO     CHAR(4)         CONSTRAINT
INVOICE_DETAIL_PROD_DETAIL_NO_FK REFERENCES PRODUCT_DETAIL,
```

```
   QTY              INTEGER        CONSTRAINT INVOICE_DETAIL_QTY_NULL NOT NULL,
   PRIMARY KEY(INVOICE_NO, PROD_DETAIL_NO)
   );

CREATE VIEW INVOICE_DETAIL_AMOUNT (PROD_DETAIL_NO,AMOUNT) AS
   SELECT INVOICE_DETAIL.PROD_DETAIL_NO, UNIT_PRICE * QTY
   FROM PRODUCT, INVOICE_DETAIL, PRODUCT_DETAIL
   WHERE PRODUCT.STYLE_NO = PRODUCT_DETAIL.STYLE_NO AND
         PRODUCT_DETAIL.PROD_DETAIL_NO = INVOICE_DETAIL.PROD_DETAIL_NO
   ;

CREATE TABLE INVOICE_PMT_TYPE (
   PTYPE_ID           VARCHAR2(10)    CONSTRAINT INVOICE_PMT_TYPE_PTYPE_ID_FK
REFERENCES PMT_TYPE,
   INVOICE_NO         VARCHAR2(10)    CONSTRAINT INVOICE_PMT_TYPE_INVOICE_NO_FK
REFERENCES INVOICE,
   PERCENTAGE         NUMBER(3,2)     CONSTRAINT
INVOICE_PMT_TYPE_PERCENTAGE_NULL NOT NULL,
   PRIMARY KEY(PTYPE_ID, INVOICE_NO)
   );

CREATE TABLE INVOICE_LOC (
   INVOICE_NO         VARCHAR2(10)    CONSTRAINT INVOICE_LOC_PK PRIMARY KEY
                                      CONSTRAINT INVOICE_LOC_INVOICE_NO_FK REFERENCES
INVOICE,
```

```
   AUTHORIZER        VARCHAR2(20)    CONSTRAINT INVOICE_LOC_AUTHORIZER_NULL
NOT NULL,
   BUYER_ID          VARCHAR2(10)    CONSTRAINT INVOICE_LOC_BUYER_ID_NULL NOT
NULL
                                     CONSTRAINT INVOICE_LOC_ID_FK REFERENCES BUYER
   )
   CLUSTER INVOICE_SUB (INVOICE_NO);

CREATE TABLE INVOICE_TT (
   INVOICE_NO        VARCHAR2(10)    CONSTRAINT INVOICE_TT_PK PRIMARY KEY
                                     CONSTRAINT INVOICE_TT_INVOICE_NO_FK REFERENCES INVOICE,
   REMARKS           VARCHAR2(128),
   BUYER_ID          VARCHAR2(10)    CONSTRAINT INVOICE_TT_BUYER_ID_NULL NOT
NULL
                                     CONSTRAINT INVOICE_TT_ID_FK REFERENCES BUYER
   )
   CLUSTER INVOICE_SUB (INVOICE_NO);

--Indexes
CREATE INDEX ORDER_DETAIL_ORDER_QTY_EX ON ORDER_DETAIL (ORDER_QTY);
CREATE INDEX ORDER_DETAIL_PRICE_EX ON ORDER_DETAIL (PRICE);
CREATE INDEX ORDER_DETAIL_PRICE_AFTER_TAX ON ORDER_DETAIL
(PRICE_AFTER_TAX);

CREATE INDEX CARTON_TOTAL_QTY_EX ON CARTON (TOTAL_QTY);
CREATE INDEX CARTON_PACKAGING_ID_EX ON CARTON (PACKAGING_ID);
```

```
CREATE INDEX CONTENT_CARTON_NO ON CONTENT (CARTON_NO);
CREATE INDEX CONTENT_QTY ON CONTENT (QTY);

CREATE INDEX INVOICE_DETAIL_PROD_DETAIL_NOEX ON INVOICE_DETAIL
(PROD_DETAIL_NO);
CREATE INDEX INVOICE_DETAIL_QTY_EX ON INVOICE_DETAIL (QTY);

CREATE INDEX PRODUCT_DETAIL_STYLE_NO_EX ON PRODUCT_DETAIL (STYLE_NO);
CREATE INDEX PRODUCT_DETAIL_COLOR_ID_EX ON PRODUCT_DETAIL (COLOR_ID);
CREATE INDEX PRODUCT_DETAIL_SIZE_ID_EX ON PRODUCT_DETAIL (SIZE_ID);
CREATE INDEX PRODUCT_DETAIL_QOH_EX ON PRODUCT_DETAIL (QOH);

CREATE INDEX BUYER_NAME_EX ON BUYER (NAME);
CREATE INDEX BUYER_TEL_EX ON BUYER (TEL);
CREATE INDEX BUYER_ACCOUNT_BALANCE_EX ON BUYER (ACCOUNT_BALANCE);

--TRIGGERS
CREATE TRIGGER ID_INSERT
AFTER INSERT ON INVOICE_DETAIL
FOR EACH ROW
BEGIN
UPDATE PRODUCT_DETAIL
SET QOH = QOH - :NEW.QTY
WHERE PRODUCT_DETAIL.prod_detail_no = :NEW.prod_detail_no;
END;
```

```
/

--INSERTS

INSERT INTO COMPOSITION VALUES ('CP01', '100% Cashmere');

INSERT INTO COMPOSITION VALUES ('CP02', '70%Wool/30%Cashmere');

INSERT INTO COMPOSITION VALUES ('CP03', '37%Viscose/22%NY/15%WOOL/26%PBT');


INSERT INTO COLOR VALUES ('CL01', 'BK');

INSERT INTO COLOR VALUES ('CL02', 'G2');

INSERT INTO COLOR VALUES ('CL03', 'P2');

INSERT INTO COLOR VALUES ('CL04', 'B1');

INSERT INTO COLOR VALUES ('CL05', 'P1');


INSERT INTO "SIZE" VALUES ('SZ01', 'Small');

INSERT INTO "SIZE" VALUES ('SZ02', 'Medium');

INSERT INTO "SIZE" VALUES ('SZ03', 'Large');


INSERT INTO DESIGN VALUES ('8EF-S01', '08-19-18', 'a.t.corner 2018 EF 1', 'P.Y.G');

INSERT INTO DESIGN VALUES ('8EF-S02', '08-19-18', 'a.t.corner 2018 EF 2', 'K.F.H');

INSERT INTO DESIGN VALUES ('8EF-S03', '08-19-18', 'a.t.corner 2018 EF 3', 'K.F.H');

INSERT INTO DESIGN VALUES ('8EF-S04', '08-19-18', 'a.t.corner 2018 EF 4', 'L.E.Y');

INSERT INTO DESIGN VALUES ('8EF-S05', '08-19-18', 'a.t.corner 2018 EF 5', 'P.Y.G');

INSERT INTO DESIGN VALUES ('8EF-S06', '08-19-18', 'a.t.corner 2018 EF 6', 'P.Y.G');

INSERT INTO DESIGN VALUES ('8EF-S07', '08-19-18', 'a.t.corner 2018 EF 7', 'K.F.H');

INSERT INTO DESIGN VALUES ('8EF-S08', '08-19-18', 'a.t.corner 2018 EF 8', 'P.Y.G');

INSERT INTO DESIGN VALUES ('8EF-S09', '08-19-18', 'a.t.corner 2018 EF 9', 'L.E.Y');

INSERT INTO DESIGN VALUES ('8EF-S10', '08-19-18', 'a.t.corner 2018 EF 10', 'P.S.H');
```

```
INSERT INTO DESIGN VALUES ('8EF-S11', '08-19-18', 'a.t.corner 2018 EF 11', 'P.Y.G');
INSERT INTO DESIGN VALUES ('8EF-S12', '08-19-18', 'a.t.corner 2018 EF 12', 'K.F.H');
INSERT INTO DESIGN VALUES ('8EF-S13', '08-19-18', 'a.t.corner 2018 EF 13', 'P.Y.G');
INSERT INTO DESIGN VALUES ('8EF-S14', '08-19-18', 'a.t.corner 2018 EF 14', 'L.E.Y');
INSERT INTO DESIGN VALUES ('8EF-S15', '08-19-18', 'a.t.corner 2018 EF 15', 'P.Y.G');
INSERT INTO DESIGN VALUES ('8EF-S16', '08-19-18', 'a.t.corner 2018 EF 16', 'K.F.H');

INSERT INTO PMT_TYPE VALUES ('P01', 'LOC');
INSERT INTO PMT_TYPE VALUES ('P02', 'TT');

INSERT INTO BUYER VALUES ('B01', 'Pearltex Korea', 23, 'Samsung-ro 119-gi',
   'Gangnam-gu', 'Seoul','',  'Korea', '',  8225151519, 8225152324, 0);
INSERT INTO BUYER VALUES ('B02', 'YOONHA KNIT', 70, 'DUSAN-RO',
   'GEUMCHEON-GU', 'Seoul', '',  'Korea', 08584, 82221813618, 82221813619, 870000);
INSERT INTO BUYER VALUES ('B03', 'DAEYUN SO., LTD', 1, 'Gangnamdaero 51 Gil',
   'Seocho-gu', 'Seoul', '', 'Korea', '', 827043348685, 827043348685, 765400);

INSERT INTO SHIPPER VALUES ('S01', 'SHANGHAI KAIXING INDUSTRY AND TRADE CO.,
LTD', 1045,
   'RIJING ROAD', 'SHANGHAI', '', 'CHINA', 200085);
INSERT INTO SHIPPER VALUES ('S02', 'SHANGHAI HUIJING CO., LTD', 3394,
   'HUAIHAIJING ROAD', 'SHANGHAI', '', 'CHINA', 210392);
-- order is from buyer 03

INSERT INTO TT VALUES ('B01', 'Pearltex', 'BANK OF COMMUNICATIONS', 263, 'SIPING
ROAD',
```

```
    'HONG KOU DISTRICT', 'SHANGHAI', '', 'CHINA', 201203, 'COMMCNSHSHI', 'HONGKOU SUB-
BRANCH',
    310046030146340005779, 'KAIXING');

INSERT INTO LOC VALUES ('B01', 'L01', 'LEE HOI HYUN', '10-08-18');
INSERT INTO LOC VALUES ('B02', 'L02', 'PARK JUN SUNG', '11-24-18');
INSERT INTO LOC VALUES ('B03', 'L03', 'YUN HA YEON', '11-27-18');

INSERT INTO BUYER_PMT_TYPE VALUES ('P01', 'B01', 0.5);
INSERT INTO BUYER_PMT_TYPE VALUES ('P01', 'B02', 0.7);
INSERT INTO BUYER_PMT_TYPE VALUES ('P01', 'B03', 0.5);
INSERT INTO BUYER_PMT_TYPE VALUES ('P02', 'B01', 0.5);
INSERT INTO BUYER_PMT_TYPE VALUES ('P02', 'B02', 0.3);
INSERT INTO BUYER_PMT_TYPE VALUES ('P02', 'B03', 0.5);
-- we need to explain that the percentage is TT

INSERT INTO "ORDER" VALUES ('O01', '08-27-18', '',
    '', 'B03', 'JI-WOO SHIN', '82-70-4334-8685', '09-10-18', '70%Wool/30%Cashmere',
    1000);
INSERT INTO "ORDER" VALUES ('O02', '08-27-18', '',
    '', 'B03', 'JI-WOO SHIN', '82-70-4334-8685', '09-10-18', '70%Wool/30%Cashmere',
    1500);
INSERT INTO "ORDER" VALUES ('O03', '08-27-18', '',
    '', 'B03', 'JI-WOO SHIN', '82-70-4334-8685', '09-10-18', '70%Wool/30%Cashmere',
    1200);
INSERT INTO "ORDER" VALUES ('O04', '08-27-18', '',
```

```
    ", 'B03', 'JI-WOO SHIN', '82-70-4334-8685', '09-10-18', '70%Wool/30%Cashmere',
    1700);
INSERT INTO "ORDER" VALUES ('O05', '08-27-18', ",
    ", 'B03', 'JI-WOO SHIN', '82-70-4334-8685', '09-10-18', '70%Wool/30%Cashmere',
    1200);
INSERT INTO "ORDER" VALUES ('O06', '08-27-18', ",
    ", 'B03', 'JI-WOO SHIN', '82-70-4334-8685', '09-10-18', '70%Wool/30%Cashmere',
    1400);

INSERT INTO BRAND VALUES ('R01', 'evercube', 'B01');
INSERT INTO BRAND VALUES ('R02', 'a. t. corner', 'B02');
INSERT INTO BRAND VALUES ('R03', 'CC Collect', 'B03');

INSERT INTO SHIPPING VALUES ('SG01', '15 Yeongjung-ro, Yeongdeungpo-dong, Seoul, South
Korea',
    'China Post', 'B03', 'S01', 1074);
INSERT INTO SHIPPING VALUES ('SG02', '70 Dusan-ro, Geumcheon-gu, Seoul, South Korea',
    'China Post', 'B02', 'S01', 972);
INSERT INTO SHIPPING VALUES ('SG03', '83 Daegu-ro, Dogok-dong, Seoul, South Korea',
    'China Post', 'B03', 'S01', 1302);
INSERT INTO SHIPPING VALUES ('SG04', '19 Gibon-ro, Choguseo-dong, Seoul, South Korea',
    'China Post', 'B03', 'S01', 506);

INSERT INTO PACKAGING VALUES ('2842', 10, 'S01');
INSERT INTO PACKAGING VALUES ('7931', 14, 'S01');
INSERT INTO PACKAGING VALUES ('2018', 13, 'S01');
```

```
INSERT INTO PACKAGING VALUES ('1094', 8, 'S01');

INSERT INTO CARTON VALUES ('C01', 1074, '2842', 210.8, 196.8, 1.176, 694, 380, 0);
INSERT INTO CARTON VALUES ('C02', 972, '7931', 173.9, 140.7, 1.039, 392, 580, 0);
INSERT INTO CARTON VALUES ('C03', 1308, '2018', 249.2, 194.2, 1.013, 803, 499, 0);
INSERT INTO CARTON VALUES ('C04', 792, '1094', 120.6, 118.7, 1.219, 208, 298, 0);
INSERT INTO CARTON VALUES ('C05', 390, '2842', 210.8, 196.8, 1.176, 0, 50, 650);
INSERT INTO CARTON VALUES ('C06', 408, '7931', 173.9, 140.7, 1.039, 90, 0, 590);
INSERT INTO CARTON VALUES ('C07', 205, '2018', 249.2, 194.2, 1.013, 130, 450, 0);
INSERT INTO CARTON VALUES ('C08', 902, '1094', 120.6, 118.7, 1.219, 0, 650, 70);
INSERT INTO CARTON VALUES ('C09', 308, '2842', 210.8, 196.8, 1.176, 490, 160, 0);
INSERT INTO CARTON VALUES ('C10', 914, '7931', 173.9, 140.7, 1.039, 340, 430, 0);
INSERT INTO CARTON VALUES ('C11', 1308, '2018', 249.2, 194.2, 1.013, 0, 257, 695);
INSERT INTO CARTON VALUES ('C12', 792, '1094', 120.6, 118.7, 1.219, 220, 0, 540);
INSERT INTO CARTON VALUES ('C13', 1074, '2842', 210.8, 196.8, 1.176, 467, 90, 0);
INSERT INTO CARTON VALUES ('C14', 972, '7931', 173.9, 140.7, 1.039, 75, 240, 300);

--Change number on carton

INSERT INTO PRODUCT_DETAIL VALUES ('PD01', '8EF-S01', 'CL01', 'SZ01',500);
INSERT INTO PRODUCT_DETAIL VALUES ('PD02', '8EF-S01', 'CL01', 'SZ02',0);
INSERT INTO PRODUCT_DETAIL VALUES ('PD03', '8EF-S01', 'CL02', 'SZ01',45);
INSERT INTO PRODUCT_DETAIL VALUES ('PD04', '8EF-S01', 'CL02', 'SZ02',135);
INSERT INTO PRODUCT_DETAIL VALUES ('PD05', '8EF-S01', 'CL03', 'SZ01',190);
INSERT INTO PRODUCT_DETAIL VALUES ('PD06', '8EF-S01', 'CL03', 'SZ02',0);
INSERT INTO PRODUCT_DETAIL VALUES ('PD07', '8EF-S01', 'CL04', 'SZ01',545);
```

```
INSERT INTO PRODUCT_DETAIL VALUES ('PD08', '8EF-S01', 'CL04', 'SZ02',230);
INSERT INTO PRODUCT_DETAIL VALUES ('PD09', '8EF-S01', 'CL05', 'SZ01',40);
INSERT INTO PRODUCT_DETAIL VALUES ('PD10', '8EF-S01', 'CL05', 'SZ02',10);
-- style from purchase order and change in color and size for each product detail

INSERT INTO "CONTENT" VALUES ( 'PD01', 'C01',80);
INSERT INTO "CONTENT" VALUES ( 'PD01','C02', 80);
INSERT INTO "CONTENT" VALUES ( 'PD01','C03', 80);
INSERT INTO "CONTENT" VALUES ( 'PD02','C04', 80);
INSERT INTO "CONTENT" VALUES ( 'PD02','C05', 64);
INSERT INTO "CONTENT" VALUES ( 'PD03', 'C06',80);
INSERT INTO "CONTENT" VALUES ( 'PD03', 'C07',80);
INSERT INTO "CONTENT" VALUES ( 'PD04','C08', 80);
INSERT INTO "CONTENT" VALUES ( 'PD04', 'C09',28);
INSERT INTO "CONTENT" VALUES ( 'PD05','C10', 80);
INSERT INTO "CONTENT" VALUES ('PD05','C11', 57);
INSERT INTO "CONTENT" VALUES ('PD06', 'C12', 73);
INSERT INTO "CONTENT" VALUES ( 'PD07','C13', 80);
INSERT INTO "CONTENT" VALUES ( 'PD08','C14', 55);

INSERT INTO ORDER_DETAIL VALUES ('PD01', 'O01', 21, 1000, 1050, 44, 35, 30, 17, 12, 42, '', '',
');
INSERT INTO ORDER_DETAIL VALUES ('PD02', 'O02', 21, 1500, 1575, 46, 39, 32, 15, 14, 45, '', '',
');
INSERT INTO ORDER_DETAIL VALUES ('PD03', 'O03', 86, 1200, 1260, 45, 37, 33, 17, 13, 43, '', '',
');
```

```
INSERT INTO ORDER_DETAIL VALUES ('PD04', 'O04', 53, 1700, 1785, 44, 35, 30, 15, 12, 42, '', '',
');
INSERT INTO ORDER_DETAIL VALUES ('PD05', 'O05', 21, 1200, 1260, 45, 37, 34, 17, 13, 43, '', '',
');
INSERT INTO ORDER_DETAIL VALUES ('PD05', 'O06', 48, 1400, 1470, 48, 50, 36, 18, 16, 47, '', '',
');

INSERT INTO PRODUCT VALUES ('8EF-S01', 'R03', 'CP01', 47.57);
INSERT INTO PRODUCT VALUES ('8EF-S02', 'R03', 'CP01', 52.65);
INSERT INTO PRODUCT VALUES ('8EF-S03', 'R03', 'CP01', 50.94);
INSERT INTO PRODUCT VALUES ('8EF-S04', 'R03', 'CP01', 46.02);
INSERT INTO PRODUCT VALUES ('8EF-S05', 'R03', 'CP01', 46.21);
INSERT INTO PRODUCT VALUES ('8EF-S06', 'R03', 'CP01', 47.19);
INSERT INTO PRODUCT VALUES ('8EF-S07', 'R03', 'CP01', 50.80);
INSERT INTO PRODUCT VALUES ('8EF-S08', 'R03', 'CP01', 37.42);
INSERT INTO PRODUCT VALUES ('8EF-S09', 'R03', 'CP01', 50.90);
INSERT INTO PRODUCT VALUES ('8EF-S10', 'R03', 'CP01', 47.72);
INSERT INTO PRODUCT VALUES ('8EF-S11', 'R03', 'CP02', 34.79);
INSERT INTO PRODUCT VALUES ('8EF-S12', 'R03', 'CP02', 37.47);
INSERT INTO PRODUCT VALUES ('8EF-S13', 'R03', 'CP02', 43.02);
INSERT INTO PRODUCT VALUES ('8EF-S14', 'R03', 'CP02', 33.16);
INSERT INTO PRODUCT VALUES ('8EF-S15', 'R03', 'CP02', 58.05);
INSERT INTO PRODUCT VALUES ('8EF-S16', 'R03', 'CP02', 50.66);

INSERT INTO INVOICE VALUES ('I01', '10-18-18', '12-11-18', 2600, 119434, '', 'S01', 'O01', 'B03',
'KOREA', 'PARK HOI GUEN');
```

```
INSERT INTO INVOICE VALUES ('I02', '09-09-18', '10-09-18', 3082, 104852, '', 'S01', 'O02', 'B01',
'KOREA', 'PARK HOI GUEN');
INSERT INTO INVOICE VALUES ('I03', '04-04-18', '06-06-18', 2084, 104927, '', 'S01', 'O03', 'B03',
'KOREA', 'PARK HOI GUEN');
INSERT INTO INVOICE VALUES ('I04', '04-04-18', '05-05-18', 1049, 195038, '', 'S01', 'O04', 'B03',
'KOREA', 'PARK HOI GUEN');

INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD01', 200);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD02', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD03', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD04', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD05', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD06', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD07', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD08', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD09', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I01', 'PD10', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I02', 'PD01', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I02', 'PD02', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I02', 'PD03', 200);
INSERT INTO INVOICE_DETAIL VALUES ('I02', 'PD04', 160);
INSERT INTO INVOICE_DETAIL VALUES ('I02', 'PD05', 200);
INSERT INTO INVOICE_DETAIL VALUES ('I02', 'PD06', 160);

INSERT INTO INVOICE_PMT_TYPE VALUES ('P01', 'I01', 0.5);
INSERT INTO INVOICE_PMT_TYPE VALUES ('P02', 'I01', 0.5);
```

```
INSERT INTO INVOICE_PMT_TYPE VALUES ('P01', 'I02', 0.3);
INSERT INTO INVOICE_PMT_TYPE VALUES ('P02', 'I02', 0.7);
INSERT INTO INVOICE_PMT_TYPE VALUES ('P01', 'I03', 0.5);
INSERT INTO INVOICE_PMT_TYPE VALUES ('P02', 'I03', 0.5);

INSERT INTO INVOICE_LOC VALUES ('I01', 'P.H,G.', 'B03');
INSERT INTO INVOICE_LOC VALUES ('I02', 'Y.S.S.', 'B01');
INSERT INTO INVOICE_LOC VALUES ('I03', 'K.S.W.', 'B03');
INSERT INTO INVOICE_LOC VALUES ('I04', 'K.G.S.', 'B03');

INSERT INTO INVOICE_TT VALUES ('I01', '', 'B03');
INSERT INTO INVOICE_TT VALUES ('I02', '', 'B01');
INSERT INTO INVOICE_TT VALUES ('I03', '', 'B03');
INSERT INTO INVOICE_TT VALUES ('I04', '', 'B03');

-- Select Statements
SELECT * FROM COMPOSITION;
SELECT * FROM COLOR;
SELECT * FROM "SIZE";
SELECT * FROM DESIGN;
SELECT * FROM PMT_TYPE;
SELECT * FROM BUYER;
SELECT * FROM SHIPPER;
SELECT * FROM TT;
SELECT * FROM LOC;
SELECT * FROM BUYER_PMT_TYPE;
```

```
SELECT * FROM "ORDER";

SELECT * FROM BRAND;

SELECT * FROM SHIPPING;

SELECT * FROM PACKAGING;

SELECT * FROM CARTON;

SELECT * FROM PRODUCT_DETAIL;

SELECT * FROM CONTENT;

SELECT * FROM ORDER_DETAIL;

SELECT * FROM PRODUCT;

SELECT * FROM INVOICE;

SELECT * FROM INVOICE_DETAIL;

SELECT * FROM INVOICE_PMT_TYPE;

SELECT * FROM INVOICE_LOC;

SELECT * FROM INVOICE_TT;


-- Describe Statements

DESCRIBE COMPOSITION;

DESCRIBE COLOR;

DESCRIBE "SIZE";

DESCRIBE DESIGN;

DESCRIBE PMT_TYPE;

DESCRIBE BUYER;

DESCRIBE SHIPPER;

DESCRIBE TT;

DESCRIBE LOC;

DESCRIBE BUYER_PMT_TYPE;
```

```
DESCRIBE "ORDER";

DESCRIBE BRAND;

DESCRIBE SHIPPING;

DESCRIBE PACKAGING;

DESCRIBE CARTON;

DESCRIBE PRODUCT_DETAIL;

DESCRIBE CONTENT;

DESCRIBE ORDER_DETAIL;

DESCRIBE PRODUCT;

DESCRIBE INVOICE;

DESCRIBE INVOICE_DETAIL;

DESCRIBE INVOICE_PMT_TYPE;

DESCRIBE INVOICE_LOC;

DESCRIBE INVOICE_TT;


SELECT AVG(delivery_date - issue_date) AS AvgDaysDifference

FROM invoice

WHERE (delivery_date - issue_date) > 0;


SELECT DISTINCT (c.description)

FROM invoice_pmt_type ipt, invoice i, buyer b, brand r, product p, composition c, pmt_type t

WHERE i.invoice_no =  ipt.invoice_no AND b.buyer_ID = i.buyer_ID AND

   b.buyer_Id = r.buyer_ID AND p.brand_Id = r.brand_id AND

   c.comp_type = p.comp_type AND t.ptype_ID = 'P01' AND ipt.ptype_id IN (

      SELECT ptype_id

      FROM invoice_pmt_type
```

```
    WHERE percentage < 0.5);

SELECT count(distinct(d.designer))
from design d, product p
where d.Date_Designed > '01-01-18' AND p.brand_id = 'R03' AND p.style_no = d.style_no;

SELECT AVG(unit_price) AS CL01Price
FROM product
where color_id = 'CL01'
MINUS
SELECT AVG(unit_price) AS CL02Price
FROM product
where color_id = 'CL02';

spool off;
```

Sample Queries

1. Average number of days it takes to deliver after the invoice has been issued

```
SELECT AVG(delivery_date - issue_date) AS AvgDaysDifference
FROM invoice
WHERE (delivery_date - issue_date) > 0;
```

	AVGDAYSDIFFERENCE
1	44.5

2. The number of designers for brand number three in 2018

```
SELECT count(distinct(d.designer))
FROM design d, product p
WHERE d.Date_Designed > '01-01-18' AND p.brand_id = 'R03' AND p.style_no = d.style_no;
```

	COUNT(DISTINCT(D.DESIGNER))
1	4

3. The color of the product with the highest unit price

```
SELECT p.color_id
FROM product p
WHERE p.unit_price = (
   SELECT max(unit_price)
   FROM product);
```

	COLOR_ID
1	CL01

4. The style numbers of designs by PE SH and style numbers of clothes that come in color number one

```
SELECT style_no
FROM design
WHERE designer = 'P.S.H.'
UNION
SELECT Style_no
FROM product
WHERE color_ID = 'CL01';
```

	STYLE_NO
1	8EF-S01
2	8EF-S03
3	8EF-S05
4	8EF-S09
5	8EF-S10
6	8EF-S11
7	8EF-S13
8	8EF-S15
9	8EF-S16

5. The composition for clothing that has payment type of TT under 50%

```
SELECT DISTINCT (c.description)
FROM invoice_pmt_type ipt, invoice i, buyer b, brand r, product p, composition c, pmt_type t
WHERE i.invoice_no =  ipt.invoice_no AND b.buyer_ID = i.buyer_ID AND
   b.buyer_Id = r.buyer_ID AND p.brand_Id = r.brand_id AND
   c.comp_type = p.comp_type AND t.ptype_ID = 'P01' AND ipt.ptype_id IN (
      SELECT ptype_id
      FROM invoice_pmt_type
      WHERE percentage < 0.5);
```

	DESCRIPTION
1	70%Wool/30%Cashmere
2	100% Cashmere

6. The style numbers of clothes sold at a higher number than the average

```
SELECT distinct pd.style_no
FROM invoice_detail id, product_detail pd
WHERE id.prod_detail_no = pd.prod_detail_no AND id.amount > (
   SELECT avg(id.amount)
   FROM invoice_detail id, product_detail pd
   WHERE id.prod_detail_no = pd.prod_detail_no);
```

	STYLE_NO
1	8EF-S01

Summary Discussion of the Physical Design

Table "Hotness" Analysis Form

	CUD/CRUD Ratio	Number of Records	Percentage of all hits	Hotness
Invoice	0.67	34	0.60%	MEDIUM
Design	0.67	3000	1.20%	MEDIUM
Product	0.67	600	0.60%	MEDIUM
Order	0.52	36	0.80%	MEDIUM
Order_Detail	0.5	576	12.50%	HOT
Composition	0.5	50	0.80%	MEDIUM
Shipping	0.5	36	0.80%	MEDIUM
Content	0.45	4320	10.30%	HOT
Carton	0.45	3600	8.60%	HOT
Invoice_Detail	0.33	544	18.70%	HOT
Shipper	0.33	2	1.20%	COLD
Brand	0.33	72	0.60%	COLD
Buyer_PMT_Type	0.33	144	2.30%	COLD
LOC	0.3	54	1%	COLD
TT	0.3	54	1%	COLD
Buyer	0.3	72	1.90%	COLD
Size	0.25	3	1.60%	COLD

Product_Detail	0.25	9000	23%	HOT
Invoice_LOC	0.2	25	0.70%	COLD
Invoice_TT	0.2	25	0.70%	COLD
Invoice_PMT_Type	0.2	51	1.90%	COLD
Color	0.17	5	2.30%	COLD
Packaging	0.09	360	4.30%	MEDIUM
PMT_Type	0.07	2	2.90%	COLD

To determine what tables and attributes to index; the amount of records, hits and the CUD/CRUD ratio are considered. The information in the table above are derived from the the 'process versus entity matrix' and the 'composite usage map'. If a cell appears as red, it means that the criteria contributes to a potential 'hot' table. Yellow cells fall in the middle, while green cells are considered to be less 'hot'.

The tables Order_Detail, Content, Carton, Invoice_Detail and Product_Detail are marked at the hottest tables. This corresponds with the low volume, but big orders. The attribute 'Design' also has two red cells, but is marked as medium hot, because only the attributes Date_Designed, Description and Designer and Style_No resides in the table. The latter attribute can also be found in the entity Product, and will therefore have less reads.

As the hot tables will have a lot of hits and reads, it is wished that the tables get indexed. As the first primary key will get automatically indexed by Oracle, these attributes are not being

considered. The most common attributes, secondary primary key and foreign keys will get indexed.

For Order_Detail, the attributes Order_Qty, Price and Price_After_Tax is indexed, because these attributes will be read in high volumes.

The indexes for the foreign key Packaging_ID and the other attribute Total_Qty are created for the table Carton. The latter is expecting a high amount of reads.

For the same reasons, indexes or the foreign key Carton_No and attribute Qty are created for the table Content.

Even though Prod_Detail_No is stored as a composite primary key in the table Invoice_Detail, Oracle only stores the first given primary key. Therefore an index is created for Prod_Detail_No. The second attribute that will get indexed for this table is Qty.

The table with the highest amount of indexes is Product_Detail, as there are several attributes that will get read in high volumes. Firstly, three indexes are created for the residing three foreign keys. In addition, the attribute QOH is indexed, as it will get read once the quantity on hand is running low and potentially needs to get restocked.

In addition, several indexes are made for the medium hot table Buyer, as there will be a high amount of reads to retrieve information about the customer. The most important attributes like Name, Tel and Account_Balance are indexed.

Lastly indexes are created for the two clusters that contains the super- and subclasses, because it will allow to insert values into the clusters after getting indexed.

We would use a trigger for Quantity on Hand because every time a new order comes in, the company has to deduct the quantity amounts from what they have in their inventories to reflect the quantity needs of a specific order. We figured that since every order would have a known quantity associated with it, we would see the Quantity on Hand attribute be used a lot, so we decided to cluster it based off those expectations.

We used a cluster for the payment types LOC and TT. We wanted to cluster the subclasses and superclasses together for a more efficient data model, as this makes subclasses and superclasses one table.

Summary and Conclusion

Some of the difficulties we encountered were interpreting the business rules and understanding how the business was run from the perspective of the manufacturer. It was especially hard to gather information to solve this problem due to the time zone differences between Los Angeles and Hong Kong where our nights after class was in the early morning hours in Hong Kong and so forth. Our point of connection to the source for our project, Jane, did an exceptional job in establishing that line of communication to help us gather essential answers we needed to go about designing the database to better suit her dad's company. One of the best features of our database is that we can now track sales functions and shipping functions for Shanghai Kai Xing Industry. Another hurdle our team had to clear was subclassing the different payment types. For this particular company, a payment can be made using one type of payment or a combination of two different types of payments: Letter of Credit or Telegraphic Transfer. Because a buyer can pay with both types for one invoice, we decided that the subclasses overlapped with each other. It took us some time to figure out that we should also subclass invoice because payment types show differently on the invoice. As a result, we subclassed invoice into invoice_loc and invoice_tt. Then, we made the invoice subclasses reference buyer subclasses because we wanted to show which buyer pays for the invoice in what types of payments. When making the CUD/CRUD tables, we ran into trouble with the payment types again. We can create these payment types, but it didn't make sense at first to also read a payment type. Because we didn't read the payment types, our CUD/CRUD ratio was 1 which is bad. To overcome this, we consulted Professor Shook and realized that although the cardinalities did not show the reads explicitly, these tables actually get read whenever a payment is made. Another difficulty that we

overcame was to insert pictures of our styles into the database. We never did this in class. We consulted Stanley, our TA, and got the statements for creating BFILEs.

Apart from the challenges mentioned above, the hardest part was being patient. Not only did we encounter technical problems with the software, but we also had to consistently update our models to make them as perfect as possible. This requires great attention to details and patience.

Going into this class in August, our group thought it wouldn't be possible to create something as meaningful as this project by semester's end. We all agreed that the best part of the project was seeing how our database architecture skills were able to help a company become more efficient in monitoring its business functions. There was nothing to dislike about our database. We saw our first database as our child and we have learned to love it and all its perfections and imperfections unconditionally. We gained a high-level understanding of what a data architect does for companies. From building the ERDs on paper to writing the DDL to create the tables and using DML to query results, it was a gratifying experience to able to partake in this miniature data consulting job for our final project.