CPSC 304 Project Cover Page

Milestone #: __2_____

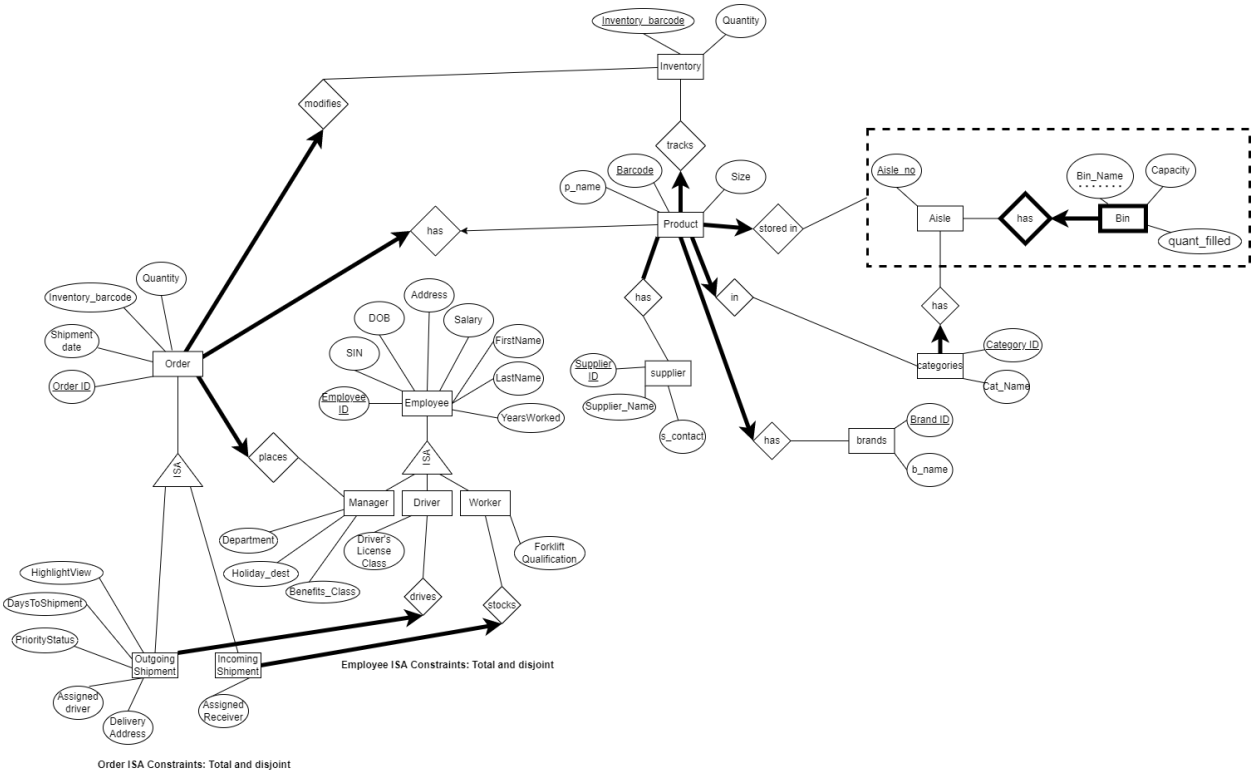
Date: __27/Jul/2022_____

Group Number: ___13_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Hai Xin Xi	10105161	z3j4e	has.x198@gmail.com
Jordan Zhao	32392152	q1f2b	zhao.jordan@hotmail.com
Syed Ahmed	34471268	a2o6b	syedw@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

1. ER Diagram:



Note: Changes made to the ER-diagram are as follows:

1. Some minor changes made to the ER diagram were made to improve readability. “Aisle #” was changed to “aisle_no”. “Row#/name” was changed to “bin_name”. ‘name’ in category was changed to “cat_name”. ‘Name’ in Supplier was changed to “supplier_name”. “Item-no” in Order changed to “Product_Barcode”.
2. Added “quant_filled” attribute to bin.
3. Inventory_ID was changed to Inventory_barcode and made the primary key of Inventory.
4. The arrows pointing from Driver and Worker to ‘drives’ and ‘stocks’, respectively, were changed to plain lines changing the key constraint from one-to-one to many-to-one. This was to ensure that a driver and worker could be assigned to more than one order.
5. In the Order relation, the attribute of “e_contact” was replaced with “YearsWorked”. Two attributes were added to Manageremployee: “Benefits_Class” and “Holiday_dest”. This was done so we could create two functional dependencies: (YearsWorked —> Benefits_Class) and (Benefits_Class —> Holiday_dest) within the ManagerEmployee relation then decompose the ManagerEmployee relation during the normalization process to ensure it is in BCNF.
6. Three attributes were added to OutgoingShipmentOrder: “PriorityStatus”, “DaysToShipment” and “HighlightView”. This was done so we could create two functional dependencies: (Quantity —> PriorityStatus) and (DaysToShipment —> HighlightView) within the OutgoingShipmentOrder relation and then decompose it during the normalization process to ensure it is in BCNF.

2. Schema:

Brands(Brand_ID: Integer,B_name: Char(30))

Aisle(Aisle_no: Integer)

Bin(Bin_name: CHAR(3), Capacity: Integer, **Aisle_no**: Integer, quant_filled: Integer)

Categories(Category_ID: Integer, Cat_name: Char(30), **Aisle_no**: Integer))

Inventory(**Inventory_barcode**: Integer, Quantity: Integer)
Constraints: Quantity NOT NULL

Supplier(S_contact: Char(50), Supplier_name: Char(40), Supplier_ID: Integer)

Product(Barcode: Integer, P_name: Char(30), Size: Char(6), **Bin_nam**e: Char(1), **Aisle_no**: Integer, **Category_ID**: Integer, **Brand_ID**: Integer, **Inventory_ID**: Integer, **Order_ID**: Integer)
Constraints: Bin_name NOT NULL, Aisle_no NOT NULL, Category_ID NOT NULL, Brand_ID NOT NULL, Inventory_ID NOT NULL)

ProductSupplier(**Barcode**: Integer, **SupplierID**: Integer)

ManagerEmployee(Employee_ID: Integer, SIN Integer, DOB Date, Address Char(50), Salary Integer, FirstName Char(15), LastName Char(15), YearsWorked Integer, Benefits_Class Char(1), Holiday_dest Char(15), Department Char(15))
Candidate Key: (SIN),(FirstName,LastName,DOB, Address)

DriverEmployee(Employee_ID Integer, SIN Integer, DOB Date, Address Char(50), Salary Integer, FirstName Char(15), LastName Char(15), YearsWorked Integer, DriverLicenseClass Char(10))
Candidate Key: (SIN),(FirstName,LastName,DOB, Address)

WorkerEmployee(Employee_ID Integer, SIN Integer, DOB Date, Address Char(50), Salary Integer,
FirstName Char(15), LastName Char(15), YearsWorked Integer, ForkLiftQual Char(10))
Candidate Key: (SIN),(FirstName,LastName,DOB, Address)

OutgoingShipmentOrder(Order_ID Integer, ShipmentDate Integer, Quantity Integer, **AssignedDriver** Integer, DeliveryAddress Char(50), PriorityStatus Char(6), DaysToShipment Integer, HighLightView Char(3), **Inventory_barcode** Integer, **Product_Barcode** Integer, **Manager** Integer)

Constraints: AssignedDriver NOT NULL, Inventory_ID NOT NULL, Product_Barcode NOT NULL, Manager NOT NULL

IncomingShipmentOrder(Order_ID Integer, ShipmentDate Integer, Quantity Integer, **AssignedReceiver** Integer, **Inventory_barcode** Integer, **Product_Barcode** Integer, **Manager** Integer)

Constraints: AssignedReceiver NOT NULL, Inventory_ID NOT NULL, Product_Barcode NOT NULL, Manager NOT NULL

3. Functional Dependencies:

Brands:
Brand_ID —> B_name

Aisle:
None (besides the trivial one: Aisle_no —> Aisle_no)

Bin:
Bin_name, Aisle_no —> Capacity, quant_filled

Categories:
Category_ID —> Cat_name, Aisle_no

Inventory:
Inventory_barcode —> Quantity

Supplier:
Supplier_ID —>S_contact, Supplier_name

Product:
Barcode —> P_name, Bin_name, Aisle_no, Category_ID, Brand_ID, Inventory_ID, Order_ID

ManagerEmployee:

1. Employee_ID —> SIN, DOB, Address, Salary, FirstName, LastName, YearsWorked, Benefits_Class, Holiday_dest, Department
2. SIN —> Employee_ID, DOB, Address, Salary, FirstName, LastName, YearsWorked, YearsWorked, Benefits_Class, Holiday_dest, Department
3. FirstName,LastName,DOB, Address —> SIN, Salary, YearsWorked, Benefits_Class, Holiday_dest, Department
4. YearsWorked —> Benefits_Class
5. Benefits_Class —> Holiday_dest

DriverEmployee:

1. Employee_ID —> SIN, DOB, Address, Salary, FirstName, LastName, YearsWorked, DriversLicenseClass
2. SIN —>Employee_ID, DOB, Address, Salary, FirstName, LastName, YearsWorked, DriversLicenseClass
3. FirstName,LastName,DOB, Address —> Employee_ID, Salary, YearsWorked, DriversLicenseClass

WorkerEmployee:

1. Employee_ID —> SIN, DOB, Address, Salary, FirstName, LastName, YearsWorked, ForkLiftQualification
2. SIN —> Employee_ID, DOB, Address, Salary, FirstName, LastName, YearsWorked, ForkLiftQualification
3. FirstName,LastName,DOB, Address —> Employee_ID, Salary, YearsWorked, ForkLiftQualification

OutgoingShipmentOrder:

1. Order_ID —> ShipmentDate, Quantity, AssignedDriver, DeliveryAddress, PriorityStatus, DaysToShipment, HighlightView, Inventory_barcode , Product_Barcode, Manager
2. Quantity —> PriorityStatus
3. DaysToShipment —> HighlightView

IncomingShipmentOrder:

Order_ID —> ShipmentDate, Quantity, AssignedReceiver, Inventory_barcode , Product_Barcode, Manager

4. Normalization:

Normalization of OutgoingShipmentOrder*: OutgoingShipmentOrder is not in BCNF. It's minimal key is Order_ID. The functional dependencies are:*

1. Order_ID —> ShipmentDate, Quantity, AssignedDriver, DeliveryAddress, PriorityStatus, DaysToShipment, HighlightView, Inventory_barcode , Product_Barcode, Manager
2. Quantity —> PriorityStatus
3. DaysToShipment —> HighlightView

FDs 2 and 3 violate BCNF because the LHS (Quantity) or (DaysToShipment) is not a superkey of the relation. Decomposing on FD (Quantity —> PriorityStatus) gives us the following two relations:

R1 (ShipmentDate, Quantity, AssignedDriver, DeliveryAddress, DaysToShipment, HighlightView, Inventory_barcode , Product_Barcode, Manager)

R2 (Quantity, PriorityStatus)

R2 is a two attribute relation and is therefore in BCNF. R1 is not in BCNF because the FD (DaysToShipment —> HighlightView) still applies to it and the LHS of it is not a superkey. Therefore, we decompose again to give:

R3 (ShipmentDate, Quantity, AssignedDriver, DeliveryAddress, DaysToShipment, Inventory_barcode , Product_Barcode, Manager)

R4 (DaysToShipment,HighlightView)

R4 is a two attribute relation and therefore in BCNF. R3 is in BCNF as only the first FD applies in it and the LHS of the FD is the minimum key. Our final decomposition product is R2, R3 and R4. All three relations are in BCNF.

Normalization of ManagerEmployee: ***ManagerEmployee is not in BCNF. It's minimal keys are Employee_ID and SIN. The functional dependencies are:***

1. Employee_ID —> SIN, DOB, Address, Salary, FirstName, LastName, YearsWorked, Benefits_Class, Holiday_dest, Department
2. SIN —> Employee_ID, DOB, Address, Salary, FirstName, LastName, YearsWorked, YearsWorked, Benefits_Class, Holiday_dest, Department
3. FirstName,LastName,DOB, Address —> SIN, Salary, YearsWorked, Benefits_Class, Holiday_dest, Department
4. YearsWorked —> Benefits_Class
5. Benefits_Class —> Holiday_dest
6. YearsWorked —> Holiday_dest (Implicit FD by transitive rule from FD4 and 5)

FDs 3 and 4 violate BCNF because the LHS (YearsWorked) and (Benefits_Class) is not a superkey of the relation WorkerEmployee. Decomposing on FD (YearsWorked —> Benefits_Class) gives us the following two relations:

R1 (SIN, DOB, Address, Salary, FirstName, LastName, E_contact, YearsWorked, Holiday_dest, ForkLiftQualification)

R2 (YearsWorked, Benefits_Class)

R2 is a two attribute relation and is therefore in BCNF. R1 is not in BCNF because the implicit FD (YearsWorked —> Holiday_dest) still applies. Therefore, we decompose again to give:

R3 (SIN, DOB, Address, Salary, FirstName, LastName, E_contact, YearsWorked, ForkLiftQualification)

R4 (YearsWorked, Holiday_dest)

R4 is a two attribute relation and therefore in BCNF. R3 is in BCNF as only the second FD apply to it and the LHS of the FD is the minimum key. Our final decomposition product is R2, R3 and R4. All three relations are in BCNF.

The new schema is:

Brands(Brand_ID: Integer,B_name: Char(30))

Aisle(Aisle_no: Integer)

Bin(Bin_name: CHAR(2), Capacity: Integer, **Aisle_no**: Integer, quant_filled: Integer)

Categories(Category_ID: Integer, Cat_name: Char(30), **Aisle_no**: Integer))

Inventory(**Inventory_barcode**: Integer, Quantity: Integer)
Constraints: Quantity NOT NULL

Supplier(S_contact: Char(50), Supplier_name: Char(40), Supplier_ID: Integer)

Product(Barcode: Integer, P_name: Char(30), Size: Char(6), **Bin_nam**e: Char(1), **Aisle_no**: Integer, **Category_ID**: Integer, **Brand_ID**: Integer, **Inventory_ID**: Integer, **Order_ID**: Integer)
Constraints: Bin_name NOT NULL, Aisle_no NOT NULL, Category_ID NOT NULL, Brand_ID NOT NULL, Inventory_ID NOT NULL)

ProductSupplier(**Barcode**: Integer, **SupplierID**: Integer)

ManagerEmployee(Employee_ID: Integer, SIN Integer, DOB Date, Address Char(50), Salary Integer, FirstName Char(15), LastName Char(15), YearsWorked Integer, Department Char(15))
Candidate Key: (SIN),(FirstName,LastName,DOB, Address)

DriverEmployee(Employee_ID Integer, SIN Integer, DOB Date, Address Char(50), Salary Integer, FirstName Char(15), LastName Char(15), YearsWorked Integer, DriverLicenseClass Char(10))
Candidate Key: (SIN),(FirstName,LastName,DOB, Address)

WorkerEmployee(Employee_ID Integer, SIN Integer, DOB Date, Address Char(50), Salary Integer,
FirstName Char(15), LastName Char(15), YearsWorked Integer, ForkLiftQual Char(10))
Candidate Key: (SIN),(FirstName,LastName,DOB, Address)

Benefits(**YearsWorked** Integer, Benefits_Class Char(1))

Holiday(**YearsWorked** Integer, Holiday_dest Char(15))

OutgoingShipmentOrder(Order_ID Integer, ShipmentDate Integer, Quantity Integer, **AssignedDriver** Integer, DeliveryAddress Char(50), DaysToShipment Integer, **Inventory_barcode** Integer, **Product_Barcode** Integer, **Manager** Integer)
Constraints: AssignedDriver NOT NULL, Inventory_ID NOT NULL, Product_Barcode NOT NULL, Manager NOT NULL

IncomingShipmentOrder(Order_ID Integer, ShipmentDate Integer, Quantity Integer, **AssignedReceiver** Integer, **Inventory_barcode** Integer, **Product_Barcode** Integer, **Manager** Integer)
Constraints: AssignedReceiver NOT NULL, Inventory_ID NOT NULL, Product_Barcode NOT NULL, Manager NOT NULL

Priority(**Quantity** Integer, PriorityStatus Char(6))

Highlight(**DaysToShipment** Integer, HighLightView Char(1))

5. SQL: Create Table DDL

```
CREATE TABLE Brands(
Brand_ID: Integer,
B_name: Char(30),
PRIMARY KEY (Brand_ID)
);

CREATE TABLE Aisle(
Aisle_no: Integer,
PRIMARY KEY (Aisle_no)
);

CREATE TABLE Bin(
Bin_name: CHAR(2),
Capacity: Integer,
Aisle_no: Integer,
quant_filled: Integer,
PRIMARY KEY (Bin_name,Aisle_no),
FOREIGN KEY (Aisle_no) REFERENCES Aisle ON DELETE NO ACTION
);

CREATE TABLE Categories(
Category_ID: Integer,
Cat_name: Char(30),
Aisle_no: Integer,
PRIMARY KEY (Category_ID),
FOREIGN KEY (Aisle_no) REFERENCES Aisle
);

CREATE TABLE Inventory(
Barcode: Integer NOT NULL DEFAULT ‘000’,
Quantity: Integer NOT NULL,
PRIMARY KEY (Barcode),
FOREIGN KEY (Barcode) REFERENCES Product(Barcode) ON DELETE SET DEFAULT,
UNIQUE (Barcode)
);


CREATE TABLE Supplier(
```

```
S_contact: Char(50),
Supplier_name: Char(40),
Supplier_ID: Integer,
PRIMARY KEY(Supplier_ID)
);


CREATE TABLE Product(
Barcode: Integer,
P_name: Char(30),
Size: Char(6),
Bin_name: Char(1) NOT NULL,
Aisle_no: Integer NOT NULL,
Category_ID: Integer NOT NULL,
Brand_ID: Integer NOT NULL,
Inventory_barcode: Integer NOT NULL,
Order_ID: Integer,
PRIMARY KEY (Barcode),
UNIQUE (Order_ID),
FOREIGN KEY (Bin_name,Aisle_no) REFERENCES Bin(Bin_name,Aisle_no),
FOREIGN KEY (Category_ID) REFERENCES Categories,
FOREIGN KEY (Brand_ID) REFERENCES Brands,
FOREIGN KEY (Inventory_barcode) REFERENCES Inventory,
FOREIGN KEY (Order_ID) REFERENCES Order
);

CREATE TABLE ProductSupplier(
Barcode: Integer,
SupplierID: Integer,
PRIMARY KEY (Barcode,SupplierID),
FOREIGN KEY Barcode REFERENCES Product,
FOREIGN KEY SupplierID REFERENCES Supplier,
);

CREATE TABLE ManagerEmployee(
Employee_ID Integer,
SIN Integer,
DOB Date,
Address Char(50),
Salary Integer,
```

```
FirstName Char(15),
LastName Char(15),
YearsWorked Integer,
Department Char(15),
PRIMARY KEY (Employee_ID),
UNIQUE (SIN)
);

CREATE TABLE DriverEmployee(
Employee_ID Integer,
SIN Integer,
DOB Date,
Address Char(50),
Salary Integer,
FirstName Char(15),
LastName Char(15),
YearsWorked Integer,
DriverLicenseClass Char(10),
PRIMARY KEY (Employee_ID),
UNIQUE (SIN)
);

CREATE TABLE WorkerEmployee(
Employee_ID Integer,
SIN Integer,
DOB Date,
Address Char(50),
Salary Integer,
FirstName Char(15),
LastName Char(15),
YearsWorked Integer,
ForkLiftQual Char(10),
PRIMARY KEY (Employee_ID),
UNIQUE (SIN)
);

CREATE TABLE Benefits(
YearsWorked Integer,
Benefits_Class Char(1),
Primary Key (YearsWorked)
```

```
Foreign Key (YearsWorked) REFERENCES ManagerEmployee(YearsWorked)
);

CREATE TABLE Holiday(
YearsWorked Integer,
Holiday_dest Char(15),
Primary Key (YearsWorked)
Foreign Key (YearsWorked) REFERENCES ManagerEmployee(YearsWorked)
);

CREATE TABLE OutgoingShipmentOrder(
Order_ID Integer,
ShipmentDate Integer,
Quantity Integer,
AssignedDriver Integer NOT NULL,
DeliveryAddress Char(50),
DaysToShipment Integer,
Inventory_barcode Integer NOT NULL,
Product_Barcode Integer NOT NULL,
Manager Integer NOT NULL,
PRIMARY KEY (Order_ID),
FOREIGN KEY (AssignedDriver) REFERENCES DriverEmployee,
FOREIGN KEY (Inventory_barcode) REFERENCES Inventory,
FOREIGN KEY (Product_Barcode) REFERENCES Product(Barcode),
FOREIGN KEY (Manager) REFERENCES ManagerEmployee,
);


CREATE TABLE IncomingShipmentOrder(
Order_ID Integer,
ShipmentDate Integer,
Quantity Integer,
AssignedReceiver Integer NOT NULL,
Inventory_barcode Integer NOT NULL,
Product_Barcode Integer NOT NULL,
Manager Integer NOT NULL,
PRIMARY KEY (Order_ID),
FOREIGN KEY (AssignedReceiver) REFERENCES WorkerEmployee,
FOREIGN KEY (Inventory_barcode) REFERENCES Inventory,
FOREIGN KEY (Product_Barcode) REFERENCES Product(Barcode),
```

```
FOREIGN KEY (Manager) REFERENCES ManagerEmployee,
);

CREATE TABLE Priority(
Quantity Integer,
PriorityStatus Char(6),
PRIMARY KEY (Quantity)
FOREIGN KEY (Quantity) REFERENCES OutgoingShipmentOrder(Quantity)
);

CREATE TABLE Highlight(
DaysToShipment Integer,
HighLightView Char(1),
PRIMARY KEY (DaysToShipment)
FOREIGN KEY (DaysToShipment) REFERENCES OutgoingShipmentOrder(DaysToShipment)
);

CREATE TABLE Inventory(
Inventory_barcode: Integer DEFAULT ‘000’,
Quantity: Integer NOT NULL,
PRIMARY KEY (Inventory_barcode),
FOREIGN KEY (Inventory_barcode) REFERENCES Inventory ON DELETE SET DEFAULT,
);
```

6. SQL: Populating Tables

```
INSERT INTO Brands VALUES (100, ‘Nike');
INSERT INTO Brands VALUES (200, ‘Adidas');
INSERT INTO Brands VALUES (300, ‘Fila');
INSERT INTO Brands VALUES (400, ‘Champion');
INSERT INTO Brands VALUES (500, ‘Columbia');

INSERT INTO Aisle VALUES (1);
INSERT INTO Aisle VALUES (2);
INSERT INTO Aisle VALUES (3);
INSERT INTO Aisle VALUES (4);
INSERT INTO Aisle VALUES (5);

INSERT INTO Bin VALUES (‘AE’, 1000, 1, 0);
INSERT INTO Bin VALUES (‘FG’, 1000, 1, 0);
INSERT INTO Bin VALUES (‘KO’, 1000, 1, 0);
INSERT INTO Bin VALUES (‘PT’, 1000, 1, 0);
INSERT INTO Bin VALUES (‘UZ’, 1000, 1, 0);
INSERT INTO Bin VALUES (‘AE’, 1000, 2, 0);
INSERT INTO Bin VALUES (‘FG’, 1000, 2, 0);
INSERT INTO Bin VALUES (‘KO’, 1000, 2, 0);
INSERT INTO Bin VALUES (‘PT’, 1000, 2, 0);
INSERT INTO Bin VALUES (‘UZ’, 1000, 2, 0);
INSERT INTO Bin VALUES (‘AE’, 1000, 3, 0);
INSERT INTO Bin VALUES (‘FG’, 1000, 3, 0);
INSERT INTO Bin VALUES (‘KO’, 1000, 3, 0);
INSERT INTO Bin VALUES (‘PT’, 1000, 3, 0);
INSERT INTO Bin VALUES (‘UZ’, 1000, 3, 0);
INSERT INTO Bin VALUES (‘AE’, 1000, 4, 0);
INSERT INTO Bin VALUES (‘FG’, 1000, 4, 0);
INSERT INTO Bin VALUES (‘KO’, 1000, 4, 0);
INSERT INTO Bin VALUES (‘PT’, 1000, 4, 0);
INSERT INTO Bin VALUES (‘UZ’, 1000, 4, 0);
INSERT INTO Bin VALUES (‘AE’, 1000, 5, 0);
INSERT INTO Bin VALUES (‘FG’, 1000, 5, 0);
INSERT INTO Bin VALUES (‘KO’, 1000, 5, 0);
INSERT INTO Bin VALUES (‘PT’, 1000, 5, 0);
INSERT INTO Bin VALUES (‘UZ’, 1000, 5, 0);
```

```
INSERT INTO Categories VALUES (1, 'T-shirt', 1);
INSERT INTO Categories VALUES (2, 'Shirt', 2);
INSERT INTO Categories VALUES (3, 'Pants', 3);
INSERT INTO Categories VALUES (4, 'Shorts', 4);
INSERT INTO Categories VALUES (5, 'Coats', 5);

INSERT INTO Inventory VALUES (364, 12);
INSERT INTO Inventory VALUES (374, 45);
INSERT INTO Inventory VALUES (384, 78);
INSERT INTO Inventory VALUES ( 394, 15);
INSERT INTO Inventory VALUES ( 354, 3);

INSERT INTO Supplier VALUES (9513124873, Acme, 302);
INSERT INTO Supplier VALUES (9513124192, Peak, 402);
INSERT INTO Supplier VALUES (9519021001, Admire, 502);
INSERT INTO Supplier VALUES (9517701000, Clothing Inc, 650);
INSERT INTO Supplier VALUES (9501000013, Divine Clothing, 71);

INSERT INTO Product VALUES (364, ‘Nike shirt 1’, ‘Small’, ‘KO’, 2, 2,100,364,NULL);
INSERT INTO Product VALUES (374, ‘Nike shirt 2’, ‘Small’, ‘KO’, 2, 2,100,374,NULL);
INSERT INTO Product VALUES (384, ‘Nike shirt 3’, ‘Small’, ‘KO’, 2, 2,100,384,NULL);
INSERT INTO Product VALUES (394, ‘Nike shirt 4’, ‘Small’, ‘KO’, 2, 2,100,394,NULL);
INSERT INTO Product VALUES (354, ‘Nike shirt 5’, ‘Small’, ‘KO’, 2, 2,100,354,NULL);

INSERT INTO ProductSupplier VALUES (364, 302);
INSERT INTO ProductSupplier VALUES (374, 402);
INSERT INTO ProductSupplier VALUES (384, 302);
INSERT INTO ProductSupplier VALUES (394, 302);
INSERT INTO ProductSupplier VALUES (354, 302);

INSERT INTO ManagerEmployee VALUES (39302, 593105933, ‘17/02/1995’, ‘45 Grove Drive’,
125000, ‘Adam’, ‘West’, 12, ‘Purchasing’);
INSERT INTO ManagerEmployee VALUES (33113, 593100141, ‘12/06/1985’, ‘75 Grove Drive’,
135000, ‘David’, ‘Ames’, 4, ‘Payroll’);
INSERT INTO ManagerEmployee VALUES (30014, 581105961, ‘19/01/1997’, ‘15 Grove Drive’,
145000, ‘John’, ‘West’, 9, ‘HR’);
INSERT INTO ManagerEmployee VALUES (35215, 500345930, ‘13/07/1975’, ‘95 Grove Drive’,
165100, ‘John’, ‘Xi’, 17, ‘Marketing’);
INSERT INTO ManagerEmployee VALUES (66546,871305965, ‘07/12/1998’, ‘25 Grove Drive’,
95000, ‘Smith’, West, 22, ‘Recruitment’);
```

INSERT INTO DriverEmployee VALUES (30002, 593111933, '17/02/1995', '45 Apple Drive', 85000, 'Michael', 'Johnson', 12, 'A');
INSERT INTO DriverEmployee VALUES (30523, 591110141, '12/06/1985', '75 Apple Drive', 95000, 'Thomas', 'Williams', 4, 'B');
INSERT INTO DriverEmployee VALUES (56114, 5811123961, '19/01/1997', '15 Apple Drive', 55000, 'Charles', 'Brown', 9, 'C');
INSERT INTO DriverEmployee VALUES (98115, 455545930, '13/07/1975', '95 Apple Drive', 65100, 'Christopher', 'Jones', 17, 'C');
INSERT INTO DriverEmployee VALUES (62146,932105965, '07/12/1998', '25 Apple Drive', 85000, 'Daniel', 'Garcia', 22, 'A');

INSERT INTO WorkerEmployee VALUES (39212, 591231933, '17/03/1995', '45 Chestnut Drive', 85000, 'Matthew', 'Miller', 15, 'Yes');
INSERT INTO WorkerEmployee VALUES (310233, 596540141, '12/07/1985', '75 Chestnut Drive', 95000, 'Anthony', 'Davis', 14, 'No');
INSERT INTO WorkerEmployee VALUES (516214, 5811682961, '19/03/1997', '15 Chestnut Drive', 55000, 'Mark', 'Rodriguez', 4, 'Yes');
INSERT INTO WorkerEmployee VALUES (900115, 455989830, '13/06/1975', '95 Chestnut Drive', 65100, 'Donald', 'Martinez', 27, 'No');
INSERT INTO WorkerEmployee VALUES (12146,91205455, '07/11/1998', '25 Chestnut Drive', 85000, 'Steven', 'Lopez', 21, 'No');

INSERT INTO Benefits VALUES (1, E);
INSERT INTO Benefits VALUES (2, E);
INSERT INTO Benefits VALUES (3, E);
INSERT INTO Benefits VALUES (4, E);
INSERT INTO Benefits VALUES (5, E);
INSERT INTO Benefits VALUES (6, D);
INSERT INTO Benefits VALUES (7, D);
INSERT INTO Benefits VALUES (8, D);
INSERT INTO Benefits VALUES (9, D);
INSERT INTO Benefits VALUES (10, D);
INSERT INTO Benefits VALUES (11, C);
INSERT INTO Benefits VALUES (12, C);
INSERT INTO Benefits VALUES (13, C);
INSERT INTO Benefits VALUES (14, C);
INSERT INTO Benefits VALUES (15, C);
INSERT INTO Benefits VALUES (16, B);
INSERT INTO Benefits VALUES (17, B);
INSERT INTO Benefits VALUES (18, B);
INSERT INTO Benefits VALUES (19, B);
INSERT INTO Benefits VALUES (20, B);
INSERT INTO Benefits VALUES (21, A);

```
INSERT INTO Benefits VALUES (22, A);
INSERT INTO Benefits VALUES (23, A);
INSERT INTO Benefits VALUES (24, A);
INSERT INTO Benefits VALUES (25, A);
INSERT INTO Benefits VALUES (26, A);
INSERT INTO Benefits VALUES (27, A);
INSERT INTO Benefits VALUES (28, A);
INSERT INTO Benefits VALUES (29, A);
INSERT INTO Benefits VALUES (30, A);

INSERT INTO Holiday VALUES (1, 'Vancouver');
INSERT INTO Holiday VALUES (2, 'Vancouver');
INSERT INTO Holiday VALUES (3, 'Vancouver');
INSERT INTO Holiday VALUES (4, 'Vancouver');
INSERT INTO Holiday VALUES (5, 'Vancouver');
INSERT INTO Holiday VALUES (6, 'Toronto');
INSERT INTO Holiday VALUES (7, 'Toronto');
INSERT INTO Holiday VALUES (8, 'Toronto');
INSERT INTO Holiday VALUES (9, 'Toronto');
INSERT INTO Holiday VALUES (10, 'Toronto');
INSERT INTO Holiday VALUES (11, 'New York City');
INSERT INTO Holiday VALUES (12, 'New York City');
INSERT INTO Holiday VALUES (13, 'New York City');
INSERT INTO Holiday VALUES (14, 'New York City');
INSERT INTO Holiday VALUES (15, 'New York City');
INSERT INTO Holiday VALUES (16, 'London');
INSERT INTO Holiday VALUES (17, 'London');
INSERT INTO Holiday VALUES (18, 'London');
INSERT INTO Holiday VALUES (19, 'London');
INSERT INTO Holiday VALUES (20, 'London');
INSERT INTO Holiday VALUES (21, 'Paris');
INSERT INTO Holiday VALUES (22, 'Paris');
INSERT INTO Holiday VALUES (23, 'Paris');
INSERT INTO Holiday VALUES (24, 'Paris');
INSERT INTO Holiday VALUES (25, 'Paris');
INSERT INTO Holiday VALUES (26, 'Paris');
INSERT INTO Holiday VALUES (27, 'Paris');
INSERT INTO Holiday VALUES (28, 'Paris');
INSERT INTO Holiday VALUES (29, 'Paris');
INSERT INTO Holiday VALUES (30, 'Paris');
```

INSERT INTO OutgoingShipmentOrder VALUES (48231, ‘12/06/1985’, 2, 30002, ‘123 Almond Drive’, 12, 354,354,35215);
INSERT INTO OutgoingShipmentOrder VALUES (48232, ‘13/06/1985’, 5, 30002, ‘123 Almond Drive’, 2, 364,364,35215);
INSERT INTO OutgoingShipmentOrder VALUES (48233, ‘14/06/1985’, 7, 30002, ‘123 Almond Drive’, 7, 374,374,35215);
INSERT INTO OutgoingShipmentOrder VALUES (48234, ‘15/06/1985’, 9, 30002, ‘123 Almond Drive’, 14, 384,384,35215);
INSERT INTO OutgoingShipmentOrder VALUES (48235, ‘16/06/1985’, 4, 30002, ‘123 Almond Drive’, 2, 394,394,35215);

INSERT INTO IncomingShipmentOrder VALUES (48641, ‘16/06/1984’, 14, 39212, 12, 354,354,39302);
INSERT INTO IncomingShipmentOrder VALUES (48671, ‘13/06/1984’, 12, 39212, 2, 364,364,39302);
INSERT INTO IncomingShipmentOrder VALUES (83133, ‘14/06/1984’, 17, 39212, 7, 374,374,39302);
INSERT INTO IncomingShipmentOrder VALUES (11134, ‘15/06/1984’, 15, 39212,16, 384,384,39302);
INSERT INTO IncomingShipmentOrder VALUES (73135, ‘16/06/1984’, 7, 310233,20, 394,394,39302);

INSERT INTO Priority VALUES (1, ‘low’)
INSERT INTO Priority VALUES (2, ‘low’)
INSERT INTO Priority VALUES (3, ‘low’)
INSERT INTO Priority VALUES (4, ‘low’)
INSERT INTO Priority VALUES (5, ‘low’)
INSERT INTO Priority VALUES (6, ‘medium’)
INSERT INTO Priority VALUES (7, ‘medium’)
INSERT INTO Priority VALUES (8, ‘medium’)
INSERT INTO Priority VALUES (9, ‘medium’)
INSERT INTO Priority VALUES (10, ‘medium’)
INSERT INTO Priority VALUES (11, ‘high’)
INSERT INTO Priority VALUES (12, ‘high’)
INSERT INTO Priority VALUES (13, ‘high’)
INSERT INTO Priority VALUES (14, ‘high’)
INSERT INTO Priority VALUES (15, ‘high’)

INSERT INTO Highlight VALUES (1, ‘Yes’)
INSERT INTO Highlight VALUES (2, ‘Yes’)
INSERT INTO Highlight VALUES (3, ‘Yes’)
INSERT INTO Highlight VALUES (4, ‘Yes’)

INSERT INTO Highlight VALUES (5, ‘Yes’)
INSERT INTO Highlight VALUES (6, ‘No’)
INSERT INTO Highlight VALUES (7, ‘No’)
INSERT INTO Highlight VALUES (8, ‘No’)
INSERT INTO Highlight VALUES (9, ‘No’)
INSERT INTO Highlight VALUES (10, ‘No’)
INSERT INTO Highlight VALUES (11, ‘No’)
INSERT INTO Highlight VALUES (12, ‘No’)
INSERT INTO Highlight VALUES (13, ‘No’)
INSERT INTO Highlight VALUES (14, ‘No’)
INSERT INTO Highlight VALUES (15, ‘No’)