2017-12-5

**CS6360.003 Database Design**

Team members:

Tianrou Chang – txc172430

Xiaoyu Zhang – xxz173130

Yidan Sheng – yxs173130

Used Furniture Store

Table of Contents

[Requirements 2](#_Toc500250616)

[Replenish stock process 2](#_Toc500250617)

[Furniture 3](#_Toc500250618)

[Deal process 3](#_Toc500250619)

[Stock 4](#_Toc500250620)

[Employee system 4](#_Toc500250621)

[Profit 4](#_Toc500250622)

[Modeling of Requirements as ER-Diagram 5](#_Toc500250623)

[Mapping of ERD in Relational Schema 6](#_Toc500250624)

[SQL Statements to create Relations in DB and Add Constraints 8](#_Toc500250625)

[Normalization of Relational Schema 11](#_Toc500250626)

[PL/SQL – Triggers 12](#_Toc500250627)

[Trigger-I Stock Capacity 12](#_Toc500250628)

[Trigger-II Deal Amount 14](#_Toc500250629)

[PL/SQL- Procedures 16](#_Toc500250630)

[Procedure-I Calculating Bonus 16](#_Toc500250631)

[Procedure-II Calculating Benefit for Each Deal 17](#_Toc500250632)

# Requirements

The used furniture company includes three branch stores and four warehouses.

Each branch store can be restocked from different warehouses, and each warehouse can deliver the furniture to different branch store.

Every branch store can check all the furniture information in the warehouses including which warehouse they stock in and which branch store they could be deliver and sold. So customers can buy the furniture they want at different branch store or get the information about which branch store they can get this furniture.

Each branch can get furniture from 1 ~ 4 warehouses (this number depends on the location).

## Replenish stock process

1. Supplier

The used furniture company will buy used furniture from suppliers, and the company should keep track of information:

* Supplier type: personal, company;
* Supplier name;
* Supplier contact information;

Each supplier should have a unique supplier number to distinguish them.

1. Supply-deal Record

When the used furniture company get furniture from the supplier, every deal should generate a supply record to save these information:

* Record number;
* Transaction date;
* Buying price;
* Furniture number;
* Stock number;
* Supplier number;

According to this record, the staff can check the initial information of every furniture. And if the deal has some problems, the company can contact supplier to solve them in time.

## Furniture

Every furniture stored in the warehouse needs to record the following information:

* Furniture number;
* Furniture type: chair, table, bed, cabinet, light etc.
* Furniture name;
* Stock number;
* Whether it is available;

According to this record, the staff can check the latest information of every furniture.

The customer can look for the furniture they want by use the furniture type.

## Deal process

1. Customer

The used furniture company will sell used furniture to customers, and the company should keep track of information:

* Customer name;
* Customer contact information;

1. Deal record

When the used furniture company sell furniture to the customer, every deal should generate a deal record to save these information:

* Deal number;
* Transaction date;
* Selling price;
* Furniture number;
* Cashier Ssn;
* Customer name;
* Customer contact information;

According to this record, the staff can check the selling information of each furniture. And if there are some problems with the deal, the staff can find the cashier who sell the furniture and the customer who buy this furniture at once.

When one deal completes successfully, the system should update the information of the furniture so that its status is marked as sold.

## Stock

For these 4 warehouses, the company should keep track of information:

* Stock number;
* Address;
* Quantity;
* Capacity;
* Manager Ssn;

When the company buy the furniture from the suppliers, the system should check whether there is enough space in the warehouse to store the furniture, and if not, change to another warehouse for storage.

When the storage is successful, the number of the furniture quantity in the warehouse in which it store should be increased by one.

When the company sell the furniture to the customers, the number of the furniture quantity in the warehouse in which the furniture store should be decreased by one.

## Employee system

1. Employee

Employees are divided into three types according to their work content:

(1)Cashier;

(2)Stock manager;

(3)Deliverer;

The cashiers work in the branch stores.

The Stock managers work in the warehouses.

The deliverers are responsible for deliver furniture from warehouses to branch stores and shipping furniture to customers.

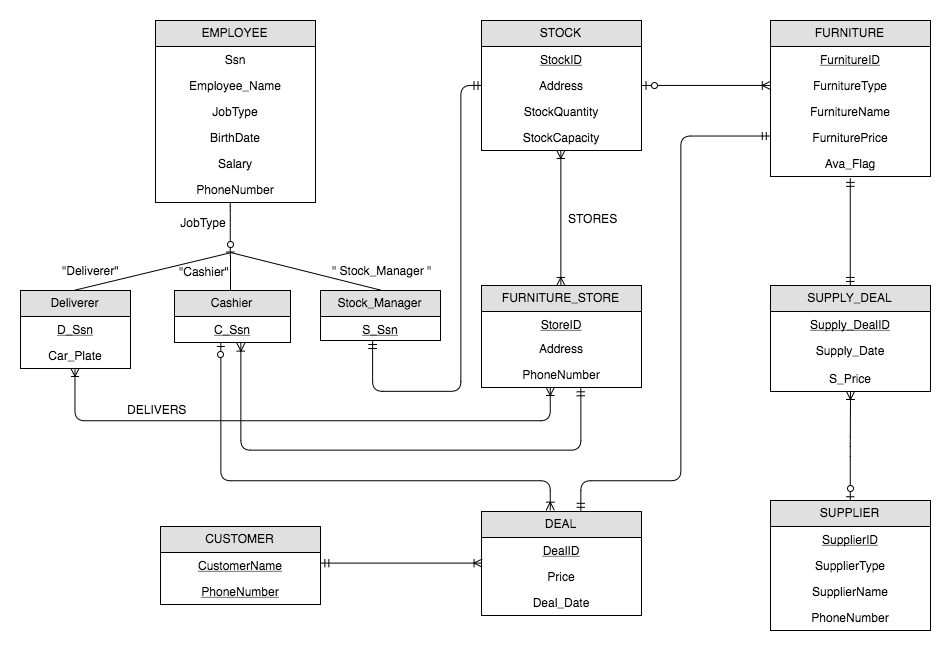
1. Bonus

When a cashier sells two or more pieces of furniture in one season, the company gives the cashier a bonus. The amount of the bonus was 20 percent of the cashier's original salary.

## Profit

According to the supply deal record and the deal record, the company can calculate the profit of each piece of furniture.

Modeling of Requirements as ER-Diagram:



1. A stock stores 0 (if empty) or more used furniture, a furniture is stored from 1 stock (N:1).
2. A stock offers furniture for 1 or more used furniture stores, a furniture store can be provided from 1 or more stock (M:N).
3. A supplier can generate 0 or more deal record, a deal record can be generated by 1 supplier (N:1).
4. A furniture information can be recorded at 1 supplying deal, a supply deal describes 1 used furniture supplying detail (1:1).
5. A customer may carry out 1 or more deals from furniture store, a deal can be carried out by 1 customer (1:N).
6. A furniture information can be recorded at 1 transaction, a deal is associated with and describes 1 furniture (1:1).
7. Employees are classified under deliverer, cashier and stock manager according to the job type.
8. A stock manager manages 1 stock, a stock must be managed by 1 stock manager (1:1).
9. A cashier is in charge of 1 or more transactions, a deal is under 1 cashier’s responsibility (N:1).
10. A cashier works for 1 furniture store, a furniture store employs 1 or more cashiers (1:N).
11. A deliverer delivers needed furniture to 1 or more furniture store, a furniture store employs 1 or more deliverers (M:N).

# **Mapping of ERD in Relational Schema**

1. FURNITURE­\_STORE

|  |  |  |
| --- | --- | --- |
| StoreID | Address | PhoneNumber |

* Primary Key: StoreID
* Foreign Keys: None

1. STOCK

|  |  |  |  |
| --- | --- | --- | --- |
| StockID | Address | StockQuantity | StockCapacity |

* Primary Key: StockID
* Foreign Keys: None

1. FURNITURE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FurnitureID | FurnitureType | FurnitureName | StockID | Ava\_Flag |

* Primary Key: FurnitureID
* Foreign Keys: StockID

1. DEAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DealID | Price | Deal\_Date | FurnitureID | Cashier\_Ssn | CustomerName | C\_PhoneNumber |

* Primary Key: DealID
* Foreign Keys : FurnitureID, CustomerName, C\_PhoneNumber ,Cashier\_Ssn

1. SUPPLIER

|  |  |  |  |
| --- | --- | --- | --- |
| SupplierID | SupplierType | SupplierName | PhoneNumber |

* Primary Key: SupplierID
* Foreign Keys: None

1. CUSTOMER

|  |  |
| --- | --- |
| CustomerName | PhoneNumber |

* Primary Key: CustomerName, PhoneNumber
* Foreign Keys: None

1. SUPPLY\_DEAL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Supply\_DealID | Supply\_Deal\_Date | SupplierID | StockID | FurnitureID | S\_Price |

* Primary Key: Supply\_DealID
* Foreign Keys: SupplierID, StockID, FurnitureID

1. EMPLOYEE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ssn | JobType | Employee\_ Name | BirthDate | Salary | PhoneNumber |

* Primary Key: Ssn
* Foreign Keys: None

1. CASHIER

|  |  |
| --- | --- |
| C\_Ssn | StoreID |

* Primary Key: C\_Ssn
* Foreign Keys: C\_Ssn , StoreID

1. DELIVERER

|  |  |
| --- | --- |
| D\_Ssn | Car\_plate |

* Primary Key: D\_Ssn
* Foreign Keys: D\_Ssn

1. STOCK\_MANAGER

|  |  |
| --- | --- |
| S\_Ssn | StockID |

* Primary Key: S\_Ssn
* Foreign Keys: S\_Ssn ,StockID

1. DELIVERS

|  |  |
| --- | --- |
| D\_Ssn | StoreID |

* Primary Key: D\_Ssn, StoreID
* Foreign Keys: D\_Ssn, StoreID

1. STORES

|  |  |
| --- | --- |
| StoreID | StockID |

* Primary Key: StoreID ,StockID
* Foreign Keys: StoreID ,StockID

# SQL Statements to create Relations in DB and Add Constraints

CREATE TABLE FURNITURE\_STORE(

STOREID INTEGER,

ADDRESS VARCHAR(50),

PHONENUMBER CHAR(10),

CONSTRAINT FURNITURE\_STORE\_PK PRIMARY KEY (STOREID)

);

CREATE TABLE STOCK(

STOCKID INTEGER,

ADDRESS VARCHAR(50),

STOCKQUANTITY INTEGER,

STOCKCAPACITY INTEGER NOT NULL,

MANAGERSSN CHAR(9),

CONSTRAINT STOCK\_PK PRIMARY KEY (STOCKID)

);

CREATE TABLE FURNITURE(

FURNITUREID INTEGER,

FURNITURETYPE VARCHAR(20) NOT NULL,

FURNITURENAME VARCHAR(30),

STOCKID INTEGER,

AVA\_FLAG CHAR(1) NOT NULL,

CONSTRAINT FURNITURE\_PK PRIMARY KEY (FURNITUREID)

);

CREATE TABLE DEAL(

DEALID INTEGER,

PRICE INTEGER,

DEAL\_DATE DATE DEFAULT TRUNC(SYSDATE),

FURNITUREID INTEGER,

CASHIER\_SSN CHAR(9),

CUSTOMERNAME VARCHAR(30),

C\_PHONENUMBER CHAR(10),

CONSTRAINT DEAL\_PK PRIMARY KEY (DEALID)

);

CREATE TABLE SUPPLIER(

SUPPLIERID INTEGER,

SUPPLIERTYPE VARCHAR(20),

SUPPLIERNAME VARCHAR(30) DEFAULT 'UNKNOWN',

PHONENUMBER CHAR(10),

CONSTRAINT SUPPLIER\_PK PRIMARY KEY (SUPPLIERID)

);

CREATE TABLE CUSTOMER(

CUSTOMERNAME VARCHAR(30),

PHONENUMBER CHAR(10),

CONSTRAINT CUSTOMER\_PK PRIMARY KEY (CUSTOMERNAME,PHONENUMBER)

);

CREATE TABLE SUPPLY\_DEAL(

SUPPLY\_DEALID INTEGER,

SUPPLY\_DEAL\_DATE DATE DEFAULT TRUNC(SYSDATE),

SUPPLIERID INTEGER,

STOCKID INTEGER,

FURNITUREID INTEGER,

S\_PRICE INTEGER NOT NULL,

CONSTRAINT SUPPLY\_DEAL\_PK PRIMARY KEY (SUPPLIERID)

);

CREATE TABLE EMPLOYEE(

SSN CHAR(9),

JOBTYPE VARCHAR(20) DEFAULT 'CASHIER',

EMPLOYEE\_NAME VARCHAR(30),

BIRTHDATE DATE,

SALARY INTEGER DEFAULT 10000,

PHONENUMBER CHAR(10) NOT NULL,

CONSTRAINT EMPLOYEE\_PK PRIMARY KEY (SSN));

CREATE TABLE CASHIER(

C\_SSN CHAR(9),

STOREID INTEGER,

CONSTRAINT CASHIER\_PK PRIMARY KEY (C\_SSN));

CREATE TABLE DELIVERER(

D\_SSN CHAR(9),

CAR\_PLATE VARCHAR(10) NOT NULL,

CONSTRAINT DELIVERER\_PK PRIMARY KEY (D\_SSN));

CREATE TABLE STOCK\_MANAGER

(

S\_SSN CHAR(9),

STOCKID INTEGER,

CONSTRAINT STOCK\_MANAGER\_PK PRIMARY KEY (S\_SSN));

CREATE TABLE DELIVERS

(

D\_SSN CHAR(9),

STOREID INTEGER,

CONSTRAINT DELIVERS\_PK PRIMARY KEY (D\_SSN,STOREID));

CREATE TABLE STORES

(

STOREID INTEGER,

STOCKID INTEGER,

CONSTRAINT STORES\_PK PRIMARY KEY (STOREID,STOCKID));

ALTER TABLE STOCK ADD CONSTRAINT STOCK\_FK FOREIGN KEY(MANAGERSSN) REFERENCES STOCK\_MANAGER(S\_SSN) ON DELETE SET NULL;

ALTER TABLE FURNITURE ADD CONSTRAINT FURNITURE\_FK FOREIGN KEY(STOCKID) REFERENCES STOCK(STOCKID) ON DELETE SET NULL;

ALTER TABLE DEAL ADD CONSTRAINT DEAL\_FKA FOREIGN KEY(FURNITUREID) REFERENCES FURNITURE(FURNITUREID)ON DELETE SET NULL;

ALTER TABLE DEAL ADD CONSTRAINT DEAL\_FKB FOREIGN KEY(CUSTOMERNAME,C\_PHONENUMBER) REFERENCES CUSTOMER(CUSTOMERNAME, PHONENUMBER)ON DELETE SET NULL;

ALTER TABLE DEAL ADD CONSTRAINT DEAL\_FKD FOREIGN KEY(CASHIER\_SSN) REFERENCES CASHIER(C\_SSN)ON DELETE SET NULL;

ALTER TABLE SUPPLY\_DEAL ADD CONSTRAINT SUPPLY\_DEAL\_FKA FOREIGN KEY(SUPPLIERID) REFERENCES SUPPLIER(SUPPLIERID)ON DELETE SET NULL;

ALTER TABLE SUPPLY\_DEAL ADD CONSTRAINT SUPPLY\_DEAL\_FKB FOREIGN KEY(STOCKID) REFERENCES STOCK(STOCKID)ON DELETE SET NULL;

ALTER TABLE SUPPLY\_DEAL ADD CONSTRAINT SUPPLY\_DEAL\_FKC FOREIGN KEY(FURNITUREID) REFERENCES FURNITURE(FURNITUREID)ON DELETE SET NULL;

ALTER TABLE CASHIER ADD CONSTRAINT CASHIER\_FK FOREIGN KEY(STOREID) REFERENCES FURNITURE\_STORE(STOREID)ON DELETE SET NULL;

ALTER TABLE STOCK\_MANAGER ADD CONSTRAINT STOCK\_MANAGER\_FK FOREIGN KEY(STOCKID) REFERENCES STOCK(STOCKID)ON DELETE SET NULL;

# **Normalization of Relational Schema**

The following Functional Dependencies exists in the relational schema –

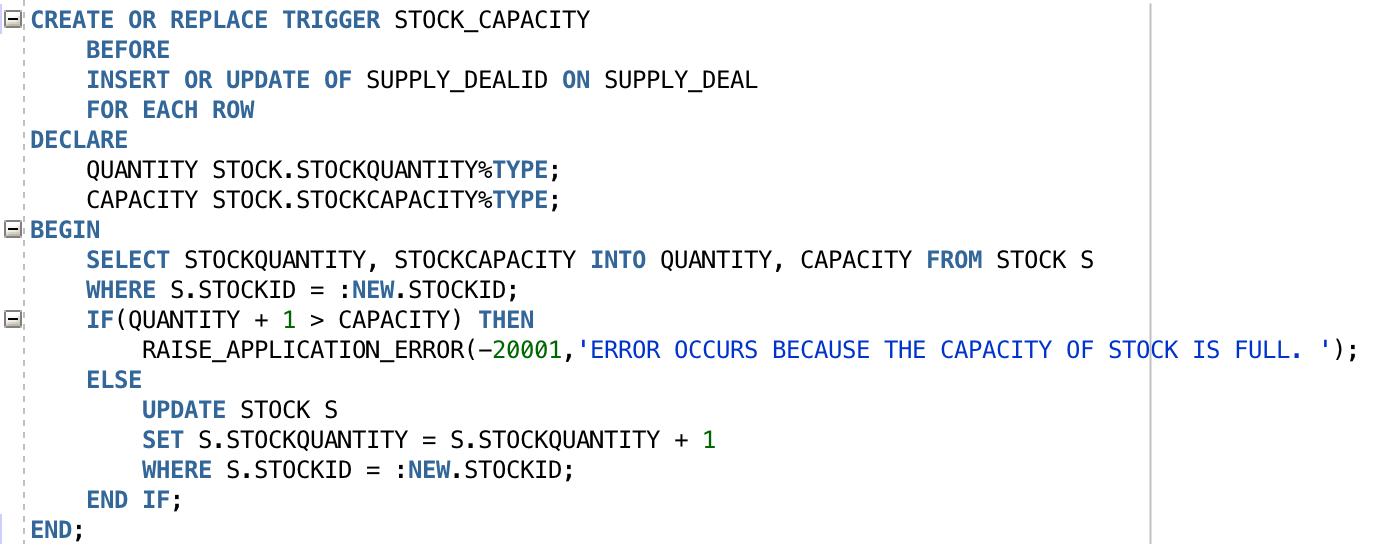
1. FURNITURE­\_STORE { StoreID -> Address, PhoneNumber}
2. STOCK { StockID -> Address, StockQuantity, StockCapacity }
3. FURNITURE { FurnitureID -> FurnitureType, FurnitureName, StockID ,Ava\_Flag}
4. DEAL {DealID -> Price, Deal\_Date, FurnitureID, Cashier\_Ssn, CustomerName, C\_PhoneNumber }
5. SUPPLIER { SupplierID -> SupplierType, SupplierName, PhoneNumber}
6. CUSTOMER { CustomerName , PhoneNumber }
7. SUPPLY\_DEAL { Supply\_DealID -> Supply\_Deal\_Date, SupplierID, StockID , FurnitureID , S\_Price }
8. EMPLOYEE { Ssn -> JobType , Employee\_ Name , BirthDate , Salary , PhoneNumber }
9. CASHIER { C\_Ssn -> StoreID }
10. DELIVERER { D\_Ssn -> Car\_plate }
11. STOCK\_MANAGER{ S\_Ssn -> StockID }
12. DELIVERS { D\_Ssn , StoreID }
13. DELIVERER { StoreID ,StockID }

The above functional dependencies cause the schema to be in third normal form.

# PL/SQL – Triggers

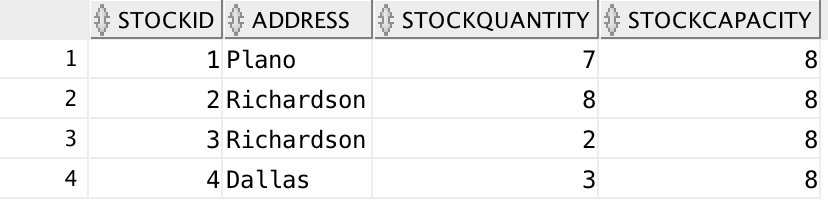
## Trigger-I Stock Capacity

When the company buys the furniture from the supplier, this furniture will store at the certain stock. If that stock capacity is not full in warehouse, the number of the furniture quantity in the warehouse in which it store should be increased by one. Otherwise, the system raises an error.



1. Positive Test Case:

Before insert, stock Table has shown,



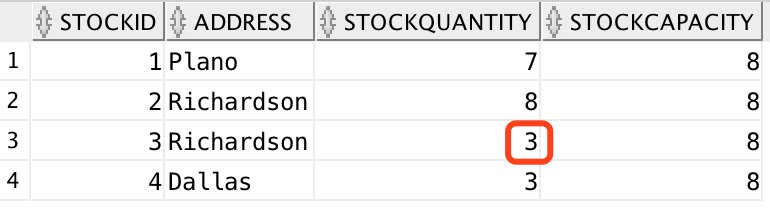
Test by SQL:

INSERT INTO SUPPLY\_DEAL(Supply\_DealID,SUPPLY\_DEAL\_DATE,SUPPLIERID,STOCKID,FURNITUREID,S\_Price)

VALUES (23,to\_date ( '03/20/1995' , 'MM/DD/YYYY' ),2,3,23,60);

Positive Test Case Output:

The No.3 Stock is not full, then the quantity of this stock increases by 1.



1. Negative Test Case:

Test by SQL:

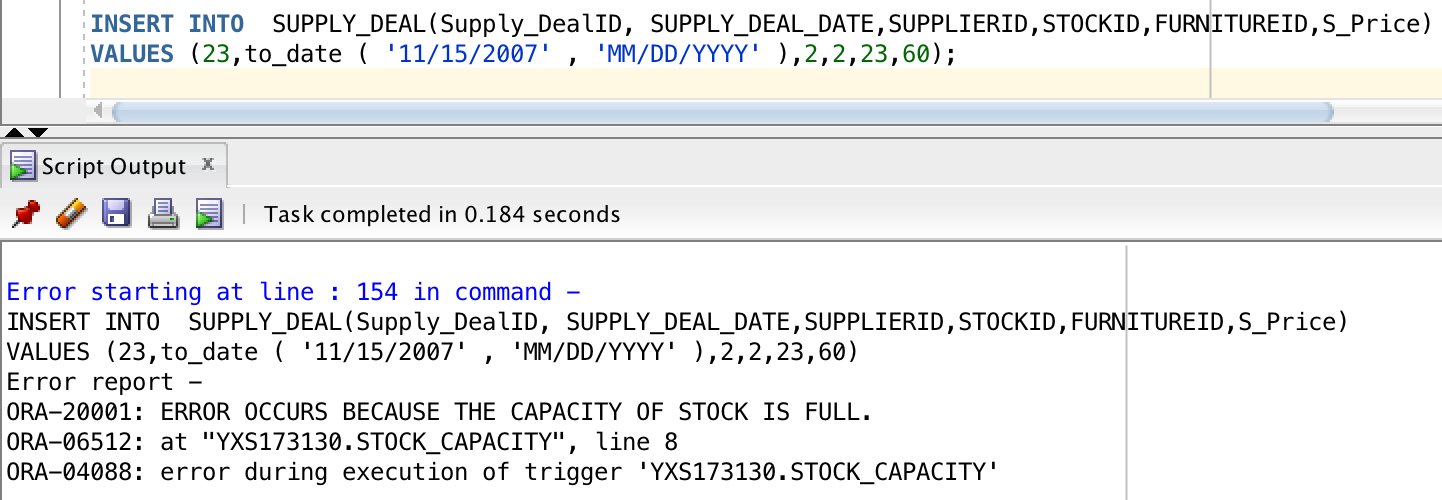
INSERT INTO

SUPPLY\_DEAL(Supply\_DealID,SUPPLY\_DEAL\_DATE,SUPPLIERID,STOCKID,FURNITUREID,S\_Price)

VALUES (23,to\_date ( '11/15/2007' , 'MM/DD/YYYY' ),2,2,23,60);

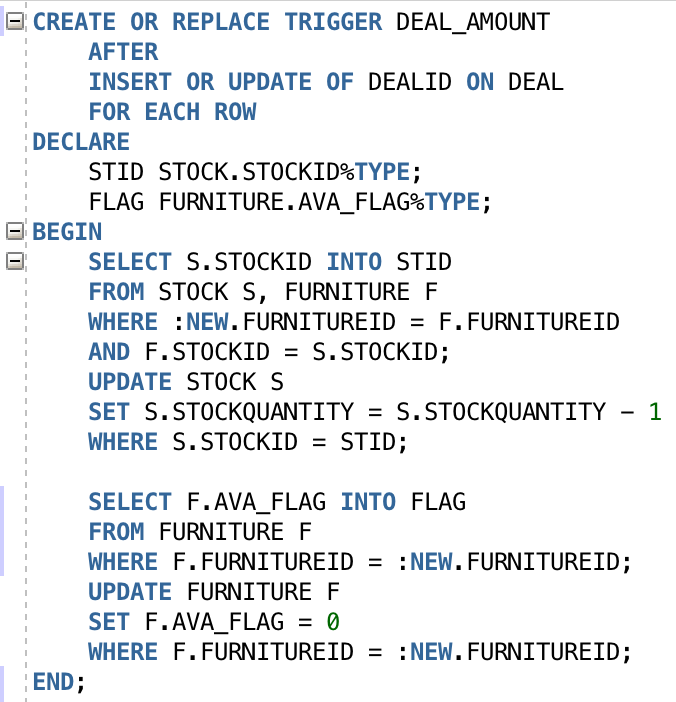
Negative Test Output:

The No.2 Stock is not full, the information can’t be inserted and raises a self-defined error.



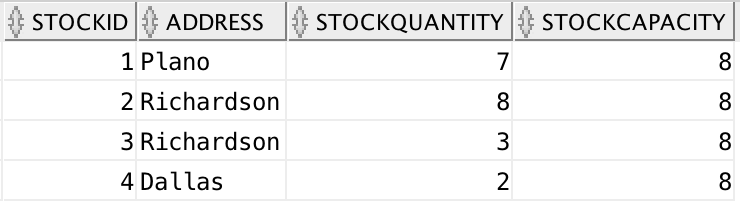
## Trigger-II Deal Amount

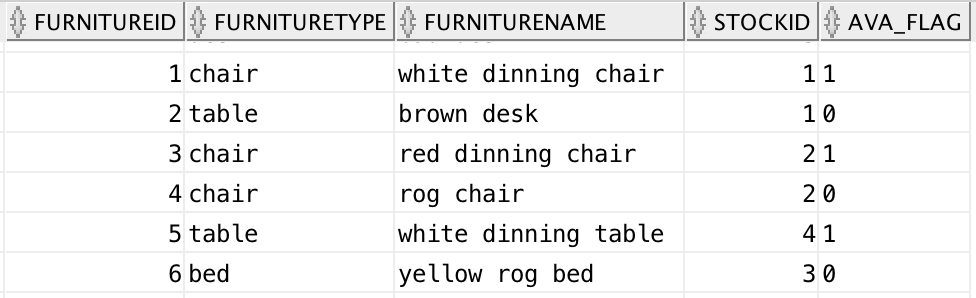
When one deal completes successfully, the number of the furniture quantity in the warehouse in which the furniture store should be decreased by one, the system should update the information of the furniture so that its status is marked as sold.



Test Case:

Furniture Table and Stock Table has shown,





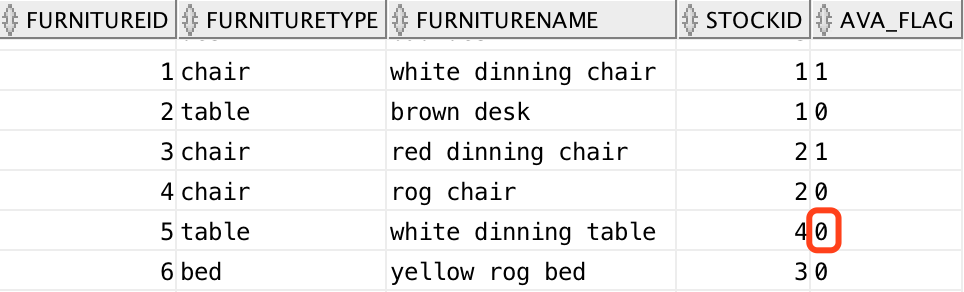
Test by SQL:

INSERT INTO DEAL(DEALID, PRICE, DEAL\_DATE, FURNITUREID, CASHIER\_SSN, CUSTOMERNAME, C\_PHONENUMBER)

VALUES(10,110,to\_date ( '03/27/2017' , 'MM/DD/YYYY' ),5,333333333,'Tom',4693343687);

Test Output:

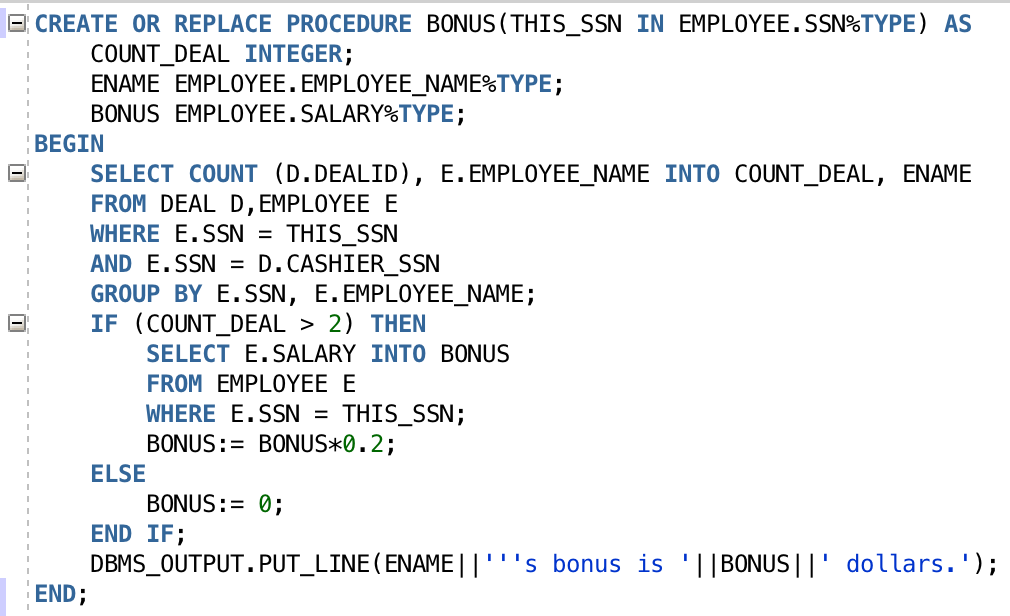
When the deal is done, the furniture which was sold should be marked as not available. Then the associated stock quantity will decrease by 1.



# PL/SQL- Procedures

## Procedure-I Calculating Bonus

When a sell man sells two or more furniture in one season, the company gives this employee the extra bonus. The amount of the bonus was 20 percent of this employee’s original salary.



Arg: (Employee SSN IN)

Test by SQL:

DECLARE

THIS\_SSN CHAR(9);

BEGIN

THIS\_SSN := '222222222';

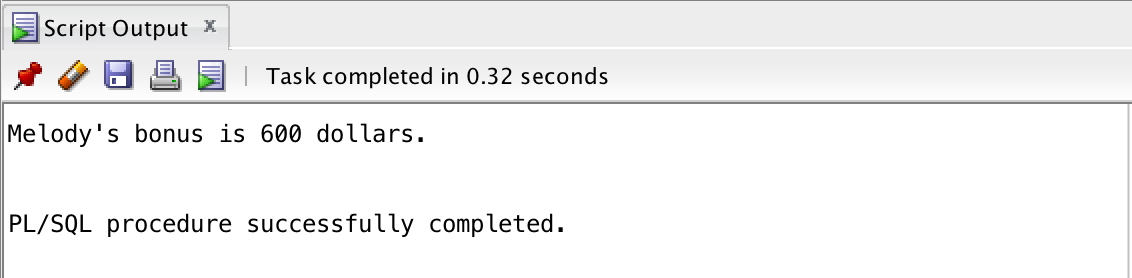
BONUS(

THIS\_SSN => THIS\_SSN

);

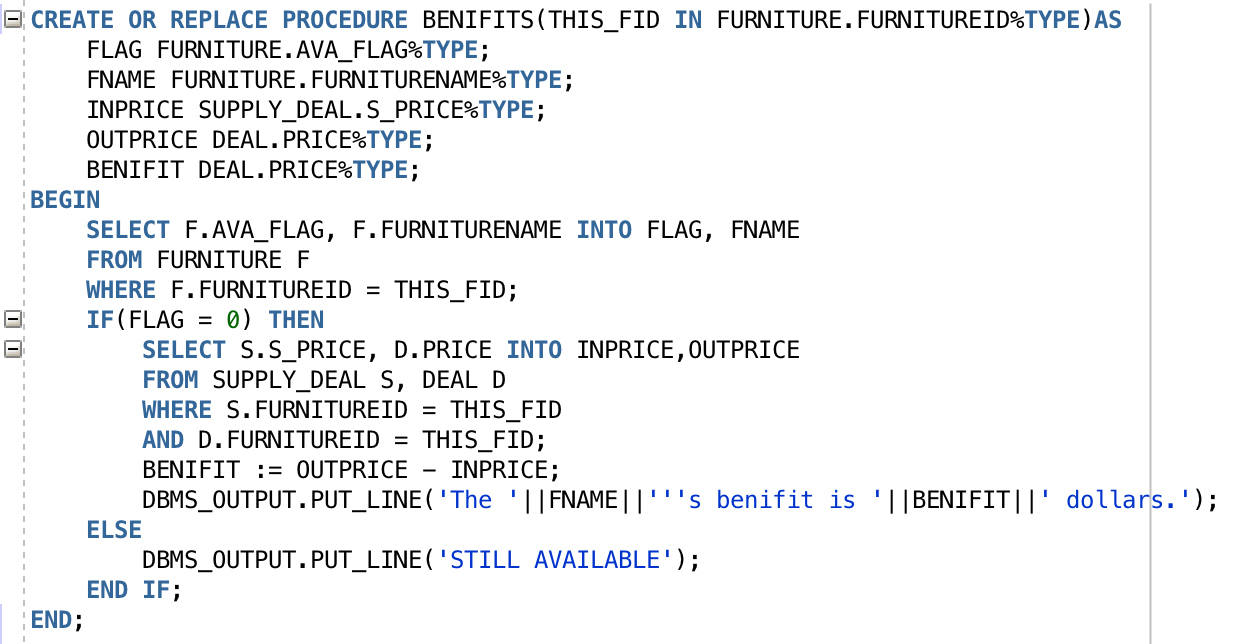
END;

Test Case Output:



## Procedure-II Calculating Benefit for Each Deal

According to the supply deal record and the deal record, the company can calculate the profit(buying price – selling price) of each furniture.



Arg: (Furniture FurnitureID IN)

Test Case SQL:

DECLARE

THIS\_FID INTEGER;

BEGIN

THIS\_FID := 2;

BENIFITS(

THIS\_FID => THIS\_FID

);

END;

Test Case Output:

