**Task 1**

CREATE TABLE ZIPCODES (

zip INT,

city VARCHAR(255)

);

CREATE TABLE EMPLOYEES (

eno INT PRIMARY KEY,

ename VARCHAR(255),

zip INT,

hdate VARCHAR(255)

);

CREATE TABLE PARTS (

pno INT PRIMARY KEY,

pname CHAR(255),

qoh INT,

price DECIMAL

);

CREATE TABLE CUSTOMERS (

cno INT PRIMARY KEY,

cname VARCHAR(255),

street VARCHAR(255),

zip INT,

phone VARCHAR(255)

);

CREATE TABLE ORDERS (

ono INT PRIMARY KEY,

cno INT,

eno INT,

received DATE,

shipped DATE,

FOREIGN KEY(eno)

REFERENCES EMPLOYEES(eno),

FOREIGN KEY(cno)

REFERENCES CUSTOMERS(cno)

);

CREATE TABLE ODETAILS (

ono INT,

pno INT,

qty INT,

PRIMARY KEY (ono,pno),

FOREIGN KEY(pno)

REFERENCES PARTS(pno)

);

CREATE TABLE ORDERS\_ERROR (

Error\_Date DATE,

ono INT,

Error\_Msg VARCHAR(255),

FOREIGN KEY(ono)

REFERENCES ORDERS(ono)

);

**Task 2**

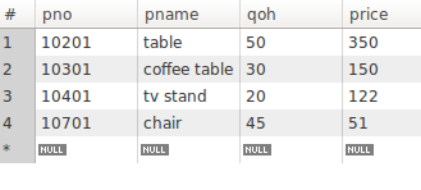
delimiter ||

CREATE TRIGGER insertOrdersD AFTER INSERT ON ODETAILS

FOR EACH ROW

BEGIN

UPDATE PARTS SET qoh = qoh - new.qty WHERE pno = new.pno;

 END;

||

**Task 3**

delimiter ||

CREATE TRIGGER updateQ AFTER INSERT ON ODETAILS

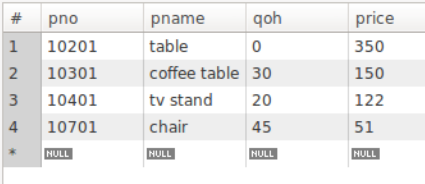
FOR EACH ROW

BEGIN

IF (SELECT qoh FROM PARTS WHERE pno = new.pno)>= new.qty THEN

UPDATE PARTS SET qoh = qoh - new.qty WHERE PARTS.pno = new.pno;

ELSE

INSERT INTO ORDERS\_ERROR values(CURRENT\_DATE, ono, "Quantity on hold is not enough!");

END IF;

END;

||

**Task 4**

**Serializable:** the Session 2 query couldn’t load after step 2, and it ended up going into a *lockwait* since it took too long to gather data. Committing the Session 1 transaction allowed me to query the Customers table and see the new data.

**Read Uncommitted:** Could see the data after running the first 2 steps without any issues. After going back to session 1 and committing the transactions, could still view the Session 1 from within Session 2.