SUPPLIER		
P * SUPPLIER_ID	NUMBER (8)	
* SNAME	VARCHAR2 (40 BYTE)	
SPH	CHAR (13 BYTE)	
⌦ SUPPLIER_PK (SUPPLIER_ID)		

STAFF		
P * STAFF_NO	NUMBER (5)	
STAFF_NAME	VARCHAR2 (25 BYTE)	
⌦ STAFF_PK (STAFF_NO)		

STOCK		
P * STOCK_CODE	CHAR (8 BYTE)	
F * SUPPLIER_ID	NUMBER (8)	
STOCK_NAME	VARCHAR2 (20 BYTE)	
QUANTITYINSTOCK	NUMBER (*,0)	
PRICE	NUMBER (5,2)	
⌦ STOCK_PK (STOCK_CODE)		
⌦ SYS_C0014032 (SUPPLIER_ID)		

SALE		
PF * STOCK_CODE	CHAR (8 BYTE)	
PF * STAFF_NO	NUMBER (5)	
P * SALEDATE	DATE	
QUANTITY	NUMBER (*,0)	
⌦ SALE_PK (STOCK_CODE, STAFF_NO, SALEDATE)		
⌦ SYS_C0014035 (STOCK_CODE)		
⌦ SYS_C0014036 (STAFF_NO)		

Description: Each stock item sold from the shop has a stock code, a name and a price. Usually, but not always, it comes from a registered supplier. When someone new comes to work in the shop, they are given a staff_no and their name is recorded. Whenever a sale is made, the stock code and quantity of stock sold is recorded, along with the date of sale, and the staff_no of whoever made the sale.

A.   Write a program to record a sale:

1. In the declaration section,

    a. accept the staff_no (V_SNO), stock_code (V_SCODE) and quantity (V_QTY).

2. In the executable section,

    a. check that V_SNO is a staff_no is in the staff table. (Either count the rows in the staff table that match and use an IF statement, or select the row with primary key value V_SNO directly from the table and cause an exception if it is not there.

    b. Check that V_SCODE is a stock_code is in the stock table, and that quantityinstock is higher than the V_QTY quantity entered.

    c. Insert a row into the SALE table

    d. Update the STOCK table to reduce QTYINSTOCK by the amount sold (V_QTY)

    e. Make the data persistent.

3. In the exceptions section:

    a. Using the catch-all 'WHEN OTHERS THEN' exception, report the error

    b. Rollback.