There are some shops which contain some products. These products are available for sale on few days but are out of stock on other days.

Create tables and indexes according to below schema:

**StockAvailability**

stkId INT

prodId INT

stkAvail INT

price DECIMAL(10,2)

date DATE

**Indexes**

PRIMARY

UNIQUE

**Product**

prodId INT

shopIdINT

price DECIMAL(10,2)

maxAvail INT

date DATE

**Indexes**

PRIMARY

Fk\_SKU\_Outlet\_idx

**Shop**

shopId INT

shopNameVARCHAR(45)

**Indexes**

PRIMARY

1. Insert some sample data into Shop and Product table
2. Create a stored procedure with date and prodId as parameters
3. Select all Stores with product availability on that date
4. The stored procedure should check if products are available in that store. If product is available, insert into the ‘StockAvailability’ table with max availability number as the stkAvailability number and date.
5. If no product is available in a store, it should show an error code: ‘Error code: 5000 – Item Out of Stock’
6. The procedure should make sure that the stkAvail is always less than or equal to maxAvail. If stkAvail is greater than maxAvail, it should show an error code: ‘Error code: 5001 – Stock limit exceeded’
7. Use cursors to insert the stockAvailablity.

1)

INSERT INTO *Shop* (*shopId*,*shopName*) VALUES (*1*,*‘ShopA’*);

INSERT INTO *Product* (prodId ,shopId,price , maxAvail , date) VALUES (1,*1*,*10000*, 10, 2021-05-26);

2)

CREATE PROCEDURE productDataGivenDate @Date date, @prodId int  
AS  
SELECT \* FROM Product WHERE date = @Date AND prodId = @prodId  
GO;

3)

SELECT \* FROM Shop where shopId=select shopId from Product where date=date AND maxAvail >0;

4)

DECLARE @i int = 0

CREATE PROCEDURE insertStock  @Date date  
AS

BEGIN

FOR queryResult in SELECT \* FROM Product WHERE date = @date  AND maxAvail >0; LOOP

SET @i = @i + 1

INSERT INTO StockAvailability (stkId, prodId, stkAvail, price, date) VALUES (i,*‘*queryResult.prodId’, *‘*queryResult.maxAvail’,*‘*queryResult.price’, *‘*queryResult.price’ );

END LOOP;

END;  
GO;

5)

CREATE PROCEDURE CheckStock  @StoreId int  
AS

SELECT IF(SELECT count(prodId) FROM Product WHERE storeId = @StoreId;, "Contain Product", " Error code: 5000 – Item Out of Stock ");

GO;

6)

CREATE PROCEDURE validateStock  @ProductId int  
AS

SELECT

IF(SELECT COUNT(stkAvail) FROM StockAvailability WHERE prodId = @ ProductId; > SELECT COUNT(maxAvail) FROM Product WHERE prodId = @ ProductId;, " Error code: 5001 – Stock limit exceeded ", "No Stock Error");

GO;