

Oracle Name: _____
Practice Class: _____
Backup Date: _____

Set the database to the **ARCHIVED LOG MODE** and restart it.

Create a new TABLESPACE:

Name: **SHOP_LASTNAME SHOP_GRIESMAYER**
Datafile: **shop_lastname1.dbf shop_griesmayer1.dbf**
Size: 200 MByte; autoextend up to 300 MByte
Datafile: **shop_lastname2.dbf shop_griesmayer2.dbf**
Size: 100 MByte

Keep the SQL-Statements for this task!

Create a new USER:

Name: **LASTNAME GRIESMAYER**

Use the **LASTNAME** user and create the following table:

LASTNAME_PRODUCT		
PRODUCT_ID	INTEGER	PRIMARY KEY
PRODUCT_NAME	VARCHAR(15)	
LAST_CHANGE	DATE	
CURRENT_STOCK	INTEGER	
MAXIMUM_STOCK	INTEGER	

Add 1000 rows.

LASTNAME_ORDER	
PRODUCT_ID	INTEGER
ORDER_DATE	DATE
ORDER_AMOUNT	INTEGER

NO PRIMARY KEY!!!

Add a TRIGGER:

Always check that the **CURRENT_STOCK** is less or equal to the **MAXIMUM_STOCK**. Throw an **raise_application_error**.

Add a PROCEDURE (pseudocode):

```
CREATE OR REPLACE PROCEDURE PROC_BUY_SELL (PROD_ID, AMOUNT)
BEGIN
  IF PROD_ID does not exist
    ROLLBACK;
    raise_application_error(....)
  END IF;

  IF AMOUNT = 0
    ROLLBACK;
    raise_application_error(....)
  END IF;

  START TRANSACTION SERIALIZABLE;

  IF AMOUNT < 0
    UPDATE LAST_NAME_PRODUCT
    SET CURRENT_STOCK = CURRENT_STOCK+AMOUNT
    WHERE PRODUCT_ID = PROD_ID;

    COMMIT;
  END IF;

  IF AMOUNT > 0
    UPDATE LASTNAME_PRODUCT
    SET CURRENT_STOCK = CURRENT_STOCK+AMOUNT
    WHERE PRODUCT_ID = PROD_ID;

    INSERT INTO LASTNAME_ODER VALUES (PROD_ID, SYSDATE, AMOUNT);

    COMMIT;
  END IF;

EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    raise_application_error(SQLCODE, SQLERRM);

END;
```

```
FOR I=1 TO 100000
  RANDOM NUMBER FOR PROD_ID (-20 .. 1000)
  RANDOM NUMBER FOR AMOUNT (-5 .. 5)
  CALL PROC_BUY_SELL (PROD_ID, AMOUNT)
LOOP
```

Copy the datafiles to a backup directory.

SHUTDOWN and delete the original datafiles.
Recover the last consistent state of the datafiles.
Check the result.

