Přidejte do tabulky **Purchase** atribut **code** (text, max. 50 znaků). Vytvořte proceduru **spTestAddPurchase** s parametry **serialNr**, **customerName**, **price** a **pieces**. Procedura přidá záznam do tabulky **Purchase** o nákupu produktu s daným sériovým číslem (dle **serialNr**), daným zákazníkem (dle **customerName**) s danou cenou a počtem kusů (**price** a **pieces**). Jestliže zákazník nebo produkt neexistují, vypište chybové hlášení a proceduru ukončete. Jako zaměstnanec se použije toho, kdo nejdéle nepotvrdil žádný nákup. Jako **nID** použijte nejvyšší ID nákupu zvýšené o 1. **purchaseDay** bude nastaven dle aktuálního data a času. Kód nákupu (**code**) bude sestaven následujícím způsobem: SSSS-YYYY-CCCC-XXXX, kde SSSS jsou první 4 znaky ze sériového čísla produktu, YYYY je rok nákupu, CCCC jsou poslední 4 znaky ze jména zákazníka a XXXX je pořadové číslo nákupu pro daného zákazníka v daném roce (vypočtěte jako počet nákupů pro daného zákazníka v daném roce + 1). Jestliže jsou sériové číslo produktu nebo jméno zákazníka kratší než 4 znaky, doplňte příslušnou část kódu znakem  \*.

Přidejte do tabulky **Purchase** atribut **priceDiff**, který bude představovat rozdíl v jednotkové ceně nákupu od posledního nákupu stejného produktu. Pozor, atribut **price** v tab. **Purchase** představuje celkovou cenu nákupu (ne cenu za jeden produkt). Vytvořte trigger **tgTestPriceDiff**, který při vložení nákupu vypočte a nastaví hodnotu nového atributu **priceDiff**.

--ALTER TABLE Purchase ADD Code VARCHAR(50);

CREATE OR REPLACE PROCEDURE spTestAddPurchase(p\_serialNr Product.serialNr%TYPE,

 p\_customerName Customer.name%TYPE, p\_price Purchase.price%TYPE, p\_pieces Purchase.pieces%TYPE) AS

 v\_pID INT;

 v\_cID INT;

 v\_eID INT;

 v\_nID INT;

 v\_purchaseDay TIMESTAMP;

 v\_code Purchase.code%TYPE;

 v\_seqNr INT;

BEGIN

 -- Zjisteni ID produktu + osetreni.

 BEGIN

 SELECT pID INTO v\_pID FROM Product WHERE serialNr = p\_serialNr;

 EXCEPTION

 WHEN NO\_DATA\_FOUND THEN

 dbms\_output.put\_line('Produkt s cislem ' || p\_serialNr || ' nebyl nalezen.');

 RETURN;

 WHEN TOO\_MANY\_ROWS THEN

 dbms\_output.put\_line('Vice produktu odpovida seriovemu cislu ' || p\_serialNr);

 RETURN;

 END;

 -- Zjisteni ID zakaznika + osetreni.

 BEGIN

 SELECT cID INTO v\_cID FROM Customer WHERE name = p\_customerName;

 EXCEPTION

 WHEN NO\_DATA\_FOUND THEN

 dbms\_output.put\_line('Zakaznik se jmenem ' || p\_customerName || ' nebyl nalezen.');

 RETURN;

 WHEN TOO\_MANY\_ROWS THEN

 dbms\_output.put\_line('V databazi existuje vice zakazniku se jmenem ' || p\_customerName);

 RETURN;

 END;

 -- Zjisteni zamestnance, ktery nejdele nic nepotvrdil.

 FOR v\_emp IN (

 SELECT

 Employee.eID, MAX(purchaseDay) AS lastPurchaseDay

 FROM

 Employee

 LEFT JOIN Purchase ON Employee.eID = Purchase.eID

 GROUP BY Employee.eID

 ORDER BY lastPurchaseDay

 ) LOOP

 v\_eID := v\_emp.eID;

 EXIT;

 END LOOP;

 -- Zjisteni ID nakupu.

 SELECT MAX(nID) + 1 INTO v\_nID

 FROM Purchase;

 v\_purchaseDay := CURRENT\_TIMESTAMP;

 -- Zjisteni sekvencniho cisla.

 SELECT COUNT(\*) + 1 INTO v\_seqNr

 FROM Purchase

 WHERE cID = v\_cID AND EXTRACT(YEAR FROM purchaseDay) = EXTRACT(YEAR FROM v\_purchaseDay);

 -- Sestaveni kodu.

 v\_code :=

 UPPER(

 RPAD(p\_serialNr, 4, '\*') || '-' ||

 TO\_CHAR(EXTRACT(year FROM v\_purchaseDay)) || '-' ||

 LPAD(p\_customerName, 4, '\*') || '-' ||

 LPAD(v\_seqNr, 4, 0)

 );

 -- Vlozeni zaznamu.

 INSERT INTO Purchase(nID, cID, pID, eID, purchaseDay, price, pieces, code)

 VALUES (v\_nID, v\_cID, v\_pID, v\_eID, v\_purchaseDay, p\_price, p\_pieces, v\_code);

END;

CREATE OR REPLACE TRIGGER tgTestPurchaseDiff BEFORE INSERT ON Purchase FOR EACH ROW

DECLARE

 v\_lastPrice INT;

BEGIN

 FOR v\_purch IN (

 SELECT \*

 FROM Purchase

 WHERE Purchase.pID = :new.pID

 ORDER BY purchaseDay DESC) LOOP

 v\_lastPrice := v\_purch.price / v\_purch.pieces;

 EXIT;

 END LOOP;

 :new.priceDiff := (:new.price / :new.pieces) - v\_lastPrice;

END;