

939 Progress Avenue, Scarborough, Ontario, Canada M1G 3T8

Final Exam (Hands on)   
COMP214

*Prof.Zoran Sarajlic*Win 2023

This is an open-book test, and you can use any resources (e.g., lecture notes, examples, your assignments, etc.) that you feel may help. However you are not allowed to communicate with anyone during the test. Please keep in mind that you are responsible for securing your own code. You will be penalized if another person has your code, or similar with your code.

Instructions: **Be sure to read the following general instructions carefully**:

* This test must be completed individually by all students
* Submit your solution **through dropbox for Final Test Hans-on within class time**. You must name your submission according to the following rule: **studentID(yourlastname)\_FinalExam.doc**

**Questions [12 marks]**

1. **[5 marks]** Assignment : Adding Descriptions for Order Status Codes When a shopper returns to the Web site to check an order’s status, information from the BB\_BASKETSTATUS table is displayed. However, only the status code is available in the BB\_BASKETSTATUS table,not the status description. Create a function named STATUS\_DESC\_SF that accepts a stage ID and returns the status description. The descriptions for stage IDs are listed in Table 6-3. Test the function in a SELECT statement that retrieves all rows in the BB\_BASKETSTATUS table for basket 4 and displays the stage ID and its description.

TABLE 6-3 Basket Stage Descriptions

Stage ID Description

1 Order submitted

2 Accepted, sent to shipping

3 Back-ordered

4 Cancelled

5 Shipped

1. **[5 marks] Assignment : Tracking Pledge Payment** Activity The DoGood Donor organization wants to track all pledge payment activity. Each time a pledge payment is added, changed, or removed, the following information should be captured in a separate table: username (logon), current date, action taken (INSERT, UPDATE, orDELETE), and the idpay value for the payment record. Create a table named DD\_PAYTRACK to hold this information. Include a primary key column to be populated by a sequence, and create a new sequence named DD\_PTRACK\_SEQ for the primary key column. Create a single trigger for recording the requested information to track pledge payment activity, and test the trigger.
2. **[2 marks]** Use ***Mongo shell*** or Mongo Compass commands to answer following questions in restarurants.json from MongoDB Lab Exercise (Week 12):
   1. List name only, and sort by zip restaurants in 7 Avenue and having Irish cousin.
   2. List address for Caffe Grazie in Manhattan cuisine Italian with grade B and score between 15 and 20.

***Only correctly executed and giving correct result code will be awarded with full mark(s).***

--1

CREATE OR REPLACE FUNCTION STATUS\_DESC\_SF

(stage\_id IN bb\_basketstatus.idstage%TYPE)

RETURN VARCHAR2

IS

lv\_result varchar2(20);

BEGIN

IF stage\_id = 1 THEN

lv\_result := 'ORDER SUBMITTED';

ELSIF stage\_id = 2 THEN

lv\_result := 'ACCPEDTED, sent to shipping';

ELSIF stage\_id = 3 THEN

lv\_result := 'BACK-ORDERED';

ELSIF stage\_id = 4 THEN

lv\_result := 'CANCELLED';

ELSE

lv\_result := 'SHIPPED';

END IF;

RETURN lv\_result;

END;

SELECT idstatus, idbasket, idstage, STATUS\_DESC\_SF(idstage) from bb\_basketstatus;

--2

DROP TABLE DD\_PAYTRACK;

CREATE TABLE DD\_PAYTRACK (

idpaytrack NUMBER(4,0) PRIMARY KEY,

username varchar2(30) DEFAULT USER,

current\_date date DEFAULT SYSDATE,

action\_taken varchar2(15),

idpay NUMBER(6,0)

);

--select USER, sysdate from dual;

DROP SEQUENCE DD\_PTRACK\_SEQ;

CREATE SEQUENCE DD\_PTRACK\_SEQ

START WITH 100

INCREMENT BY 10;

CREATE OR REPLACE TRIGGER DD\_PAYMENT\_TRG

BEFORE INSERT OR UPDATE OR DELETE

ON DD\_PAYMENT

FOR EACH ROW

DECLARE

lv\_action varchar2(20);

BEGIN

DBMS\_OUTPUT.PUT\_LINE('DD\_PAYMENT\_TRG Fired');

IF INSERTING THEN

lv\_action := 'INSERT';

INSERT INTO DD\_PAYTRACK (idpaytrack, username, current\_date,action\_taken,idpay)

VALUES (DD\_PTRACK\_SEQ.nextval, USER, SYSDATE, lv\_action,:NEW.idpay);

END IF;

IF DELETING THEN

lv\_action := 'DELETE';

INSERT INTO DD\_PAYTRACK (idpaytrack, username, current\_date,action\_taken,idpay)

VALUES (DD\_PTRACK\_SEQ.nextval, USER, SYSDATE, lv\_action,:OLD.idpay);

END IF;

IF UPDATING THEN

lv\_action := 'UPDATE';

INSERT INTO DD\_PAYTRACK (idpaytrack, username, current\_date,action\_taken,idpay)

VALUES (DD\_PTRACK\_SEQ.nextval, USER, SYSDATE, lv\_action,:NEW.idpay);

END IF;

END;

--INSERT

INSERT INTO DD\_PAYMENT

VALUES(1465,109,30,SYSDATE,'CC');

--UPDATE

UPDATE DD\_PAYMENT

SET payamt = 40

WHERE idpay = 1465;

--DELETE

DELETE FROM DD\_PAYMENT WHERE idpay =1465;

--CHECK DD\_PAYTRACK

SELECT \* from DD\_PAYTRACK;

--3

1. db.restaurants.find(

{ "address.street": "7 Avenue", "cuisine": "Irish" },

{ "name": 1, "\_id": 0 }

).sort({ "address.zipcode": 1 })

1. db.restaurants.find({

"address.city": "Manhattan",

"cuisine": "Italian",

"grades.grade": "B",

"grades.score": { "$gte": 15, "$lte": 20 },

"name": "Caffe Grazie"

});