# Assignment 2 5%

## Due Date: November 2, 2017 midnight

This assignment covers the material from Chapters 8 DS book and SQL DDL and DML statements.

**Part 1** **PL/SQL Procedure 3 %**

The following relations (based on chapter 4, DS textbook) form part of a relational database schema:

Hotel (hotelNo, hotelName, hotel\_address, city, province, country)

Room (roomID, roomNo, hotelNo, type, and price)

Booking (BookingID, guestNo, dateFrom, dateTo, roomID)

Guest (guestNo, guestName, guestAddress, comments)

Where Hotel contains hotel details and hotelNo is the primary key; Room contains room details for each hotel and roomID is the primary key; Room type specifies “single”, “double” or “family.” Booking contains details of the bookings and BookingID is the primary key; and the Guest table contains guest details and guestNo is the primary key. Comments are a string of characters and digits stored a variable character type with maximum 128 characters.

**Preliminary task:**

Create required tables (use **prefix A2\_ for their names**) and add data. You can use CASE tool or SQLDeveloper.

1. **Problem:** Our reservation center gets a lot of “last minute” requests for a one night reservation starting the same day. The staff would like to have a quick method of finding an available room (any type any hotel) in a specific city. For example:a guest arrives to Kamloops in the late afternoon and needs to stay for one night in Kamloops. The system should find all hotels in Kamloops with at least one available room for tonight.

**Hints for solution**:

Create a stored procedure LIST\_AVAILABLE with one input parameter: p\_city. This procedure should list all hotels in the specified city with at least one room available for today’s (SYSDATE) night. The room is available if there are no reservations for today’s night. List the “title”: Hotels available in city ….for 2016-10-26 and a line for each hotel name and address.

CREATE OR REPLACE PROCEDURE CHECK\_AVAILABLE\_ROOMS (P\_CITY VARCHAR)

IS

hotel\_number NUMBER;

name\_hotel VARCHAR(35);

hotel\_address VARCHAR(60);

hotel\_city VARCHAR(20);

CURSOR rooms\_cursor IS

SELECT HOTELNO, HOTELNAME, HOTEL\_ADDRESS, CITY FROM

(SELECT H.HOTELNO, H.HOTELNAME, H.HOTEL\_ADDRESS, H.CITY

FROM A2\_ROOM R LEFT OUTER JOIN A2\_BOOKING B ON R.ROOMID = B.ROOMID

JOIN A2\_HOTEL H ON R.HOTELNO = H.HOTELNO

WHERE

R.ROOMID NOT IN (SELECT ROOMID FROM A2\_BOOKING)

OR B.DATETO < SYSDATE

GROUP BY H.HOTELNO, H.HOTELNAME, H.HOTEL\_ADDRESS, H.CITY

ORDER BY H.HOTELNO)

WHERE UPPER (CITY) =UPPER (P\_CITY);

BEGIN

OPEN rooms\_cursor;

LOOP

FETCH rooms\_cursor INTO hotel\_number,name\_hotel,hotel\_address,hotel\_city;

EXIT WHEN rooms\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE (hotel\_number||' '||name\_hotel||' '||hotel\_address||' '||UPPER (hotel\_city));

END LOOP;

CLOSE rooms\_cursor;

END;

EXECUTE CHECK\_AVAILABLE\_ROOMS ('KAMLOOPS');



**HETEL 5 is not included since all its rooms are booked.**

1. **Problem:** The database designer has forgotten to create an attribute for the phone numbers. The hotel staff was using the comments (attribute) to enter textual comments and also phone numbers. Ooops.. The hotel manager is very upset and your task is to provide a temporarily solution. The manager wants to have a list of the guests (guestName, and possible phone number extracted from the comments. This has to be done within one hour (before the manager gets back from lunch).

**Hints for solution:**

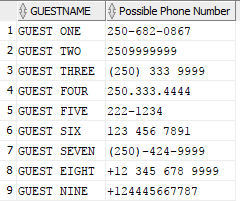
Create an SQL statement to list the guests (guest names) and the parts of the comments matching phone patterns. The examples of phone numbers: (250) 333 9999, 250-371-5592, 371-5592 250.333.4444. Good news: Oracle supports Regular Expressions (!). Idea: Use the regular expressions to find potential phone numbers in comments.

Oracle starting from 10g uses regular expressions. Read Chapter 10 (Oracle 11g SQL book) pp. 357-359 (on reserve).Chapter 10 (Oracle 12c SQL book) pp. 374-377. There are five regexp functions in Oracle SQL and PL/SQL: REGEXP\_LIKE, REGEXP\_INSTR, REGEXP\_REPLACE, REGEXP\_SUBSTR, and REGEXP\_COUNT.

**Submit**

* Listing of the Stored Procedure
* Test of the procedure showing the available rooms
* SQL query to find the phone numbers in comments.
* Test running the SQL query

SELECT GUESTNAME,REGEXP\_SUBSTR(COMMENTS,'[0-9]{10}|[0-9]{3}.{1}[0-9]{3}.{1}[0-9]{4}|.{1}[0-9]{3}.{1}.{1}[0-9]{3}.{1}[0-9]{4}|[+]{1}[0-9]{2}[0-9]{3}[0-9]{3}[0-9]{4}|[+]{1}[0-9]{2}.{1}[0-9]{3}.{1}[0-9]{3}.{1}[0-9]{4}|[0-9]{7}|[0-9]{3}.{1}[0-9]{4}')"Possible Phone Number" FROM A2\_GUEST;



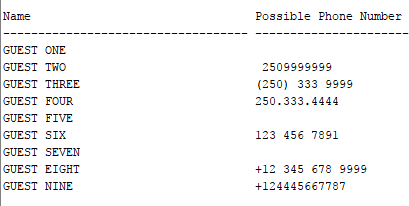
SELECT GUESTNAME "Name",

REGEXP\_SUBSTR (COMMENTS,

'([+]?[0-9]{2})?[(,[:space:]]{0,1}[0-9]{3}[),.,-,[:space:]]{0,2}[0-9]{3,}[),.,-,[:space:]]{0,2}[0-9]{4,}')

"Possible Phone Number"

FROM A2\_GUEST;



* RETURNS MOST OF THE NUMBERS FROM THE COMMENTS

**Part 2** **Archiving** 2 **%**

**Bookings should be archived after 2 years. Create the** ARCHIVED\_BOOKING table which will have the same columns as the BOOKING table and additionally a column called ARCHIVED\_DATE. Create a stored procedure to archive bookings older than 2 years (based on DATE\_TO and SYSDATE). Add the old bookings to the ARCHIVED\_BOOKING table and remove the old bookings from the BOOKING table.

**Submit**

1. Create statement for the ARCHIVED\_BOOKING table
2. Listing of the Stored Procedure
3. Test demonstrating the execution of the archived procedure

CREATE TABLE A2\_ARCHIVED\_BOOKING

AS (SELECT \* FROM A2\_BOOKING WHERE 0=1);

ALTER TABLE A2\_ARCHIVED\_BOOKING

ADD (ARCHIVED\_DATE DATE DEFAULT SYSDATE NOT NULL);

CREATE OR REPLACE PROCEDURE ARCHIVE\_OLD\_BOOKINGS

AS

BEGIN

INSERT INTO A2\_ARCHIVED\_BOOKING (BOOKINGID, GUESTNO, DATEFROM, DATETO, ROOMID, ARCHIVED\_DATE)

SELECT A.BOOKINGID, A.GUESTNO, A.DATEFROM, A.DATETO, A.ROOMID, SYSDATE

FROM A2\_BOOKING A

WHERE A.DATETO < add\_months (SYSDATE,-24);

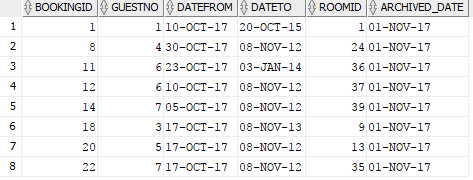
COMMIT;

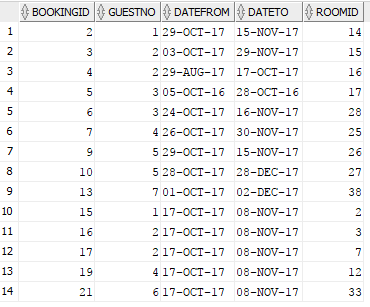
DELETE FROM A2\_BOOKING WHERE DATETO < add\_months (SYSDATE,-24);

COMMIT;

END;

EXECUTE ARCHIVE\_OLD\_BOOKINGS;





**Hand-ins:**