|  |
| --- |
| **Lab 4 COMP 3610 2%**  **Due date October 27, 23:59:59** |

**Objectives:**

* Create tables
* Design and implement simple PL/SQL functions
* Use PL/SQL function in SQL query

**Small database:**

This is a relatively simplified database to store data on clients, cars, and rentals. The RESERVATION table includes the data on past rentals and future reservations. The cars can be rented only once a day. The START\_DATE and END\_DATE include time, but rentals are calculated in full days. This means that a car rented on 2017-10-10 23:30:00 and returned on 2017-10-11 01:00:00 has been rented for two day.

**This is not the best design** ☺ but it is gives us a chance to exercise some PL/SQL and SQL.

# CLIENT

C\_ID

F\_NAME

L\_NAME

DOB

CITY

GENDER

# VEHICLE

V\_ID

VIN

V\_MAKE

COST\_PER\_DAY

# RESERVATION

R ID

START\_DATE

END\_DATE

C\_ID

V\_ID

TOTAL\_COST

**Step 1:**

Create three tables and add data. V\_ID, R\_ID, C\_ID are numbers and primary keys. Make sure that you specify the entity integrity and referential integrity constraints.

**Problem 1 Age (function)**

The manager is furious: some cars have been rented to the clients, who are not 21 years old... and to some who are more than 80 years old. The vehicle reservation system should check the age of the person renting the car. Your task is to write a function to calculate age (in whole years) based on DOB. The age is calculated as of today (SYSDATE on the server). The function can be tested using SQL SELECT statement, for example,

SELECT CALC\_AGE (*actual parameter*) FROM DUAL;

CREATE OR REPLACE FUNCTION calc\_age(dob DATE)

RETURN NUMBER

AS

BEGIN

RETURN EXTRACT (YEAR FROM SYSDATE)-EXTRACT (YEAR FROM dob);

END calc\_age;

SELECT calc\_age(to\_date('05/10/1995','DD/MM/YYYY'))"AGE" FROM DUAL; 

**List all clients (C\_ID, F\_NAME, L\_NAME, AGE, issue) who are younger than 21 and who are older than 80 years. Use one select statement and list the clients who have age “issues.” If they are too young, specify “too young.” If they are too old specify “too old.” Sort by the client id and use your new function.**

**Submit:**

1. Create statement for the function.
2. Select statement to list clients < 21 years and older than 80 (using your function).
3. Results from the query (2).

CREATE OR REPLACE FUNCTION get\_issue(age NUMBER)

RETURN VARCHAR2

AS

BEGIN

IF age < 21 THEN

RETURN 'TOO YOUNG';

END IF;

IF age > 80 THEN

RETURN 'TOO OLD';

END IF;

RETURN 'NO ISSUE';

END get\_issue;

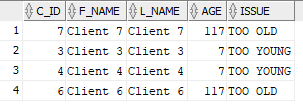
SELECT CLIENT.C\_ID, CLIENT.F\_NAME, CLIENT.L\_NAME, calc\_age (CLIENT.DOB)"AGE", get\_issue (calc\_age(CLIENT.DOB)) "ISSUE"

FROM RESERVATION JOIN CLIENT

ON RESERVATION.C\_ID = CLIENT.C\_ID

WHERE calc\_age (CLIENT.DOB) < 21 OR calc\_age (CLIENT.DOB)>80

ORDER BY CLIENT.C\_ID;



**METHOD 2:**

SELECT CLIENT.C\_ID, CLIENT.F\_NAME, CLIENT.L\_NAME, calc\_age (CLIENT.DOB)"AGE", 'TOO YOUNG' "ISSUE"

FROM RESERVATION JOIN CLIENT

ON RESERVATION.C\_ID = CLIENT.C\_ID

WHERE calc\_age (CLIENT.DOB) < 21

UNION

SELECT CLIENT.C\_ID, CLIENT.F\_NAME, CLIENT.L\_NAME, calc\_age (CLIENT.DOB)"AGE", 'TOO OLD' "ISSUE"

FROM RESERVATION JOIN CLIENT

ON RESERVATION.C\_ID = CLIENT.C\_ID

WHERE calc\_age (CLIENT.DOB) > 80;

**Problem 2 Car Usage (function)**

The manager wants to check the **past utilization** (number of days rented out) of cars in current year (SYSDATE on the server). Your task is to write a function to return total number of days rented for a specified car (the parameter is V\_ID). Include only the past rentals (do not include reservations or rentals not ended as of SYSDATE). The function has one formal parameter: V\_ID (number).

**List car makes (V\_MAKE) and their utilization (using your function). Sort the results from the most popular makes to the least popular makes. Include only the makes with some rentals.**

**Submit:**

1. Create statement for the function.
2. Select statement to list all cars and their utilization (using your function).
3. Results from the query (2).

CREATE OR REPLACE FUNCTION get\_usage(vehicle\_id NUMBER)

RETURN NUMBER AS

v\_usage NUMBER;

BEGIN

SELECT SUM (USAGE) INTO v\_usage FROM

(SELECT v\_id, END\_DATE-START\_DATE"USAGE"

FROM RESERVATION

WHERE RESERVATION.V\_ID = vehicle\_id

AND END\_DATE<SYSDATE

AND EXTRACT (YEAR FROM SYSDATE) = EXTRACT (YEAR FROM RESERVATION.END\_DATE))

WHERE v\_id = vehicle\_id

GROUP BY vehicle\_id;

RETURN v\_usage;

END get\_usage;

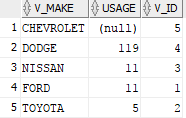
SELECT VEHICLES.V\_MAKE, get\_usage (RESERVATION.V\_ID)"USAGE", RESERVATION.V\_ID

FROM VEHICLES LEFT OUTER JOIN RESERVATION

ON RESERVATION.V\_ID = VEHICLES.V\_ID

GROUP BY RESERVATION.V\_ID, VEHICLES.V\_MAKE

ORDER BY "USAGE" DESC;

`