Lab 4 COMP 3610 2%

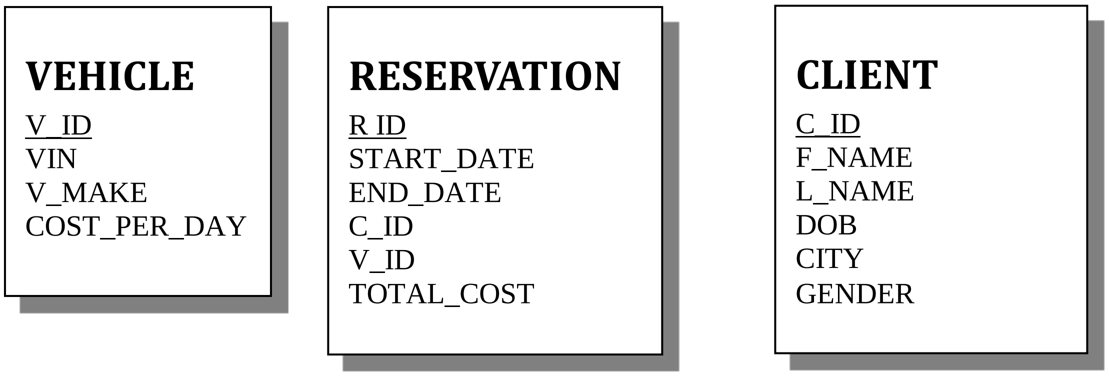
Due date October 27, 23:59:59

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.



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.

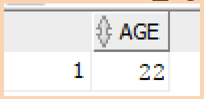
Problem 1 Age (function)
The manager is furious: some cars have been rented to the clients, who are not 21 years old... and to some who are more than 80 years old. The vehicle reservation system should check the age of the person renting the car. Your task is to write a function to calculate age (in 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;
```



```
     AGE
1     22
```

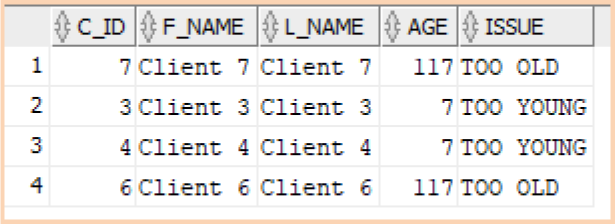
List all clients (C_ID, F_NAME, L_NAME, AGE, issue) who are younger than 21 and who are older than 80 years. Use one select statement and list the clients who have age “issues.” If they are too young, specify “too young.” If they are too old specify “too old.” Sort by the client id and use your new function.

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;
```



	C_ID	F_NAME	L_NAME	AGE	ISSUE
1	7	Client 7	Client 7	117	TOO OLD
2	3	Client 3	Client 3	7	TOO YOUNG
3	4	Client 4	Client 4	7	TOO YOUNG
4	6	Client 6	Client 6	117	TOO OLD

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).

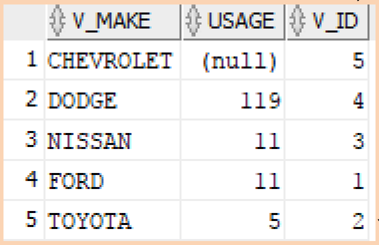
List car makes (V_MAKE) and their utilization (using your function). Sort the results from the most popular makes to the least popular makes. Include only the makes with some rentals.

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;
```



```
     V_MAKE      USAGE   V_ID
  1  CHEVROLET   (null)     5
  2  DODGE          119     4
  3  NISSAN          11     3
  4  FORD            11     1
  5  TOYOTA           5     2
```