**Individual Assessment**

**Member 2: Ho Kah Mun (2207230)**

|  |  |
| --- | --- |
| **Two Queries** | **1.Find the total amount of payment of each customer**  SELECT CustomerName, 'RM'||to\_char(max(amount),'9999.99') AS "TOTAL"  FROM customer c, payment p  WHERE c.customer\_id=p.customer\_id  GROUP by CustomerName  ORDER by 2;    This query displays each customer's name along with the total amount they've paid, formatting it in the 'RM' currency format and rename it as “TOTAL”, and sorts the total amount in ascending order based on the total payment amount.  Customer and payment tables are connected based on the customer ID. It ensures that the data retrieved corresponds to the same customer in both tables.  **2.Display the employee\_ID and employee name who live in Penang**  SELECT employee\_ID AS "ID" , (FirstName||' '||LastName) AS "Employee Name"  FROM employee  WHERE address LIKE '%Penang';    This query displays the employee ID and rename as “ID”, the last name and first name combine as full name of employees and rename as “Employee Name” whose address includes the word 'Penang'. |
| **Two Stored Procedure** | **1.**  **Modify the price of menu item by using the ID of menu item if the current price of food has dropped or rised.**  CREATE OR REPLACE PROCEDURE updateMenuItemPrice  (  id in INT,  current\_price IN NUMBER  )  IS  BEGIN  UPDATE menuitem  SET price=current\_price  WHERE MenuItem\_id=id;  COMMIT;  END;  /  EXECUTE updateMenuItemPrice('007', 25.00);    Before executing, the original price is RM23.50 for the menu item id which is 007.      After executing the command, the price for the id of menu item 007 has been changed to RM25.  This procedure updates the 'price' column in the 'menuitem' table to the new price specified by 'current\_price', where the 'MenuItem\_id' matches the 'id' provided.  **2. This procedure is used to update the position of the Employee**  CREATE OR REPLACE PROCEDURE update\_emp\_position  (  Emp\_id IN VARCHAR2,  positionId IN VARCHAR2  )  IS  BEGIN  Update Employee  SET Position\_ID = positionId  WHERE Employee\_ID = Emp\_id;  COMMIT;  END;  /  EXECUTE update\_emp\_position('E7810','P002');    Before executing,the employee with ID ‘E7810’ is a waiter .    After executing the command, the position of the employee with ID ‘E7810’ has been updated to the cook.  This procedure updates the position of an employee based on their employee ID. The 'Position\_ID' column in the 'Employee' table is updated to the new position ID specified by 'positionId', where the 'Employee\_ID' matches the 'Emp\_id' provided after running the procedure. |
| **Two Functions** | **1.** **Function helps to calculate the average payment.**  SET SERVEROUTPUT ON  CREATE OR REPLACE FUNCTION CalPaymentAvg  RETURN VARCHAR2  IS  averagePayment VARCHAR2(10);  BEGIN  SELECT to\_char(AVG(amount),'999.99')  INTO averagePayment  FROM Payment;  RETURN averagePayment;  END;  /    DECLARE  average\_payment VARCHAR2(10);  BEGIN  average\_payment := CalPaymentAvg();  DBMS\_OUTPUT.PUT\_LINE('The average of all the payment amount is RM'||average\_payment);  END;  /    This function calculates the average payment amount from the 'Payment' table using the AVG() function and converts it to a string format with two decimal places using the to\_char() function. It assigns the result to the 'averagePayment' variable.  The DBMS\_OUTPUT.PUT\_LINE() procedure prints out the result and displaying the average payment amount preceded by 'RM'. After running the function, it display the message “The average of all the payment amount is RM 45.56 ” .  **2. This function is used to find the position of the employees.**  CREATE OR REPLACE FUNCTION findposition  (  P\_EmpID VARCHAR2  )  RETURN VARCHAR2  IS  E\_Position VARCHAR2(100);  BEGIN  SELECT PositionName  INTO E\_Position  FROM Employee e, Position p  WHERE e.Position\_ID = p.Position\_ID  AND Employee\_ID = P\_EmpID;  RETURN E\_Position;  END;  /    SET SERVEROUTPUT ON  DECLARE  E\_ID CHAR(5);  E\_PositionName VARCHAR2(100);  BEGIN  E\_ID := '&E\_ID';  E\_PositionName := findposition(E\_ID);  DBMS\_OUTPUT.PUT\_LINE(chr(10));  DBMS\_OUTPUT.PUT\_LINE('The position of the employee ID '||E\_ID||' is '||E\_PositionName||'.');  END;  /      This function retrieves the position name of an employee based on their employee ID from the 'Position' table where the 'Position\_ID' matches between the 'Employee' and 'Position' tables, and the 'Employee\_ID' matches the provided employee ID ('P\_EmpID'). It assigns the result to the 'E\_Position' variable.  DBMS\_OUTPUT.PUT\_LINE() procedure prints out the result , displaying the position of the employee corresponding to the entered employee ID. From the situation above, it will display the message “The position of the employee ID E4720 is manager.” after entering the ID which is E4720. |