CNIT 372: Database Programming
Group 4: Sam Cyr, Ryan Flanagan, Sumair Ahmed

Group Project - Final Report

Background

We chose to create a database for a car dealership, using Oracle's sqldeveloper, and we plan to use what we've learned about PL/SQL so far in class in order to make our own datatype: COLOR. The goal of the database will be an efficient way for employees to search for cars, customers, and car sales, and sort them by whatever parameter they would like. The database will need to be quick and easily accessible, which is why we chose oracle sqldeveloper to host our database, since we have prior experience with the app, and we know it will get the job done.

Database Description

The database will consist of 4 related tables: CAR, EMPLOYEE, CUSTOMER, and CAR_SALE. The columns of the car table are vin and the datatype is varchar2(17), make and the datatype is varchar2(4), year and the datatype is varchar2(30), and color which we make our own datatype for, but for now the datatype will be varchar2(15). The second table is the Employee table and the columns (columns:datatype format) are EmployeeID:varchar2(10), EmpFirstName:varchar2(30), EmpLastName:varchar2(30), Phone:varchar2(10), and Email:varchar2(100). The next table is the Customer table, and the columns are CUstomerID;varchar2(10), CustFirstName:varchar2(30), CustLastName:varchar2(30), Phone:varchar2(10), Email:varchar2(100). And our final table is Car_Sale and the columns are SaleID:varchar2(10), vin(from Car table), CustomerID(from Customer table), EmployeeID (from Employee table), and price:numeric(6,2). The cardinality is as follows: One employee to many Car_sales, One Car to One Car_sale, and One Customer to one or many Car_Sale(s).

Solutions

Which employee has had the most number of car sales in this quarter?

This question can help the dealership know who their best sales people are, and can help them know who deserves a raise, promotion, or bonus for the quarter.

```
PROCEDURE get_employee_most_car_sales(emp_id OUT VARCHAR2, num_sales OUT
NUMBER)
  IS
    v_max_sales NUMBER := 0;
  BEGIN
    FOR c IN (
        SELECT EmployeeID, COUNT(*) AS sales_count
        FROM CAR_SALE
        WHERE TRUNC(SaleDate, 'Q') = TRUNC(SYSDATE, 'Q')
        GROUP BY EmployeeID
    ) LOOP
        IF c.sales_count > v_max_sales THEN
            v_max_sales := c.sales_count;
            emp_id := c.EmployeeID;
            num_sales := c.sales_count;
        END IF;
    END LOOP;
  END;
```

What is the total value of cars sold by each employee for the quarter?

This question can help the dealership know which employees are meeting their sales quota and which are lacking, and who needs to have a meeting with management.

```
PROCEDURE employee_sales_quarter
  AS
    v_quarter_start DATE := TRUNC(SYSDATE, 'Q');
    v_quarter_end DATE := ADD_MONTHS(TRUNC(SYSDATE, 'Q'), 3) - 1;
    v_sales_count NUMBER;
  BEGIN
```

```
    FOR emp IN (SELECT EmployeeID, EmpFirstName, EmpLastName FROM EMPLOYEE)
    LOOP
      SELECT COUNT(*)
      INTO   v_sales_count
      FROM   CAR_SALE
      WHERE  EmployeeID = emp.EmployeeID
      AND    SaleDate BETWEEN v_quarter_start AND v_quarter_end;

      IF v_sales_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Employee: ' || emp.EmpFirstName || ' ' || emp.EmpLastName);
        DBMS_OUTPUT.PUT_LINE('Total sales for the quarter: ');
        FOR sales IN (SELECT SUM(Price) AS total_sales
                      FROM CAR_SALE
                      WHERE EmployeeID = emp.EmployeeID
                      AND SaleDate BETWEEN v_quarter_start AND v_quarter_end)
        LOOP
          DBMS_OUTPUT.PUT_LINE('$' || TO_CHAR(sales.total_sales));
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('-------------------------------------');
      END IF;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Employees not shown have 0 Sales');
  END;
```

What is the average value of car sold each day during this year?

This question can help the dealership know the statistics of which season, and day of the week is their peak sales time, so they can play to their strengths on those days.

```
FUNCTION get_avg_car_sale_price(sale_date IN DATE) RETURN NUMBER
  IS
    total_price NUMBER := 0;
    num_sales NUMBER := 0;
  BEGIN
    FOR sale IN (SELECT * FROM CAR_SALE WHERE SaleDate = sale_date)
    LOOP
      total_price := total_price + sale.Price;
      num_sales := num_sales + 1;
    END LOOP;
    IF num_sales > 0 THEN
      RETURN total_price / num_sales;
    ELSE
      RETURN 0;
    END IF;
  END;
```

Which employee has the highest value of cars sold each day during this month?

The information provided by this question will be used by management to assess their sales team. It will identify employees that are doing well and also identify employees that may need additional coaching. More specific than quarter values, it will identify day to day trends for the employees.

```
FUNCTION find_top_selling_employee RETURN VARCHAR2
  AS
    top_employee VARCHAR2(30);
  BEGIN
    SELECT e.EmpFirstName || ' ' || e.EmpLastName INTO top_employee
    FROM EMPLOYEE e
    JOIN CAR_SALE s ON e.EmployeeID = s.EmployeeID
    WHERE s.SaleDate BETWEEN TRUNC(SYSDATE, 'MONTH') AND SYSDATE
    GROUP BY e.EmpFirstName, e.EmpLastName
    ORDER BY SUM(s.Price) DESC
    FETCH FIRST 1 ROWS ONLY;

    RETURN top_employee;
  END;
```

What is the most popular car this year?

This information may show up in many management reports. It can drive management decisions for ordering, marketing, and pricing their stable of vehicles.

```
FUNCTION GET_MOST_POPULAR_CAR_THIS_YEAR RETURN VARCHAR2
  IS
    v_year CAR.Year%TYPE;
    v_max_sales NUMBER := 0;
    v_max_make VARCHAR2(30);
    v_max_year VARCHAR2(4);
    v_result VARCHAR2(50);
  BEGIN
    SELECT TO_CHAR(SYSDATE, 'YYYY') INTO v_year FROM DUAL;

    FOR c IN (SELECT Make, Year, COUNT(*) AS num_sales FROM CAR_SALE
              JOIN CAR ON CAR_SALE.VIN = CAR.VIN
              WHERE EXTRACT(YEAR FROM SaleDate) = v_year
              GROUP BY Make, Year)
    LOOP
      IF c.num_sales > v_max_sales THEN
        v_max_sales := c.num_sales;
        v_max_make := c.Make;
        v_max_year := c.Year;
      END IF;
    END LOOP;

    v_result := v_max_make || ' ' || v_max_year;

    RETURN v_result;
  END;
```

What is the most popular car for each month?

Similar to the most popular car for the year, this information provides a more detailed dive into consumer preference. The further breakdown by month can take into account other factors that may change through the months.

```
DECLARE
v_max_sales NUMBER := 0;
    v_max_make VARCHAR2(30);
    v_max_year VARCHAR2(4);
    month NUMBER := 1;
    v_month varchar2(16);
BEGIN
    LOOP
    SELECT TO_CHAR(TO_DATE(month, 'MM'), 'Month') AS "Month Name" into
v_month FROM DUAL;
    SELECT Make, Year, COUNT(*) INTO v_max_make, v_max_year, v_max_sales
```

```
FROM CAR_SALE
              JOIN CAR ON CAR_SALE.VIN = CAR.VIN
              WHERE EXTRACT(MONTH FROM SaleDate) = month
              GROUP BY Make, Year
              ORDER BY COUNT(*) DESC
              FETCH FIRST 1 ROW ONLY;
DBMS_OUTPUT.PUT_LINE('The most popular car in ' || v_month || ' is: ' ||
v_max_make || ' ' || v_max_year);
    month := month+1;
    EXIT WHEN month = 13;
    END LOOP;

  END;
```

What is the most popular color of car for each month?

This question reveals if there is a pattern of customer’s preference of car color as it is affected by month. This information will aid in advertising efforts and ordering methodology.

```
DECLARE
v_max_sales NUMBER := 0;
    month NUMBER := 1;
    v_month varchar2(16);
    v_max_color varchar2(100);
BEGIN
LOOP
SELECT TO_CHAR(TO_DATE(month, 'MM'), 'Month') AS "Month Name" into v_month
FROM DUAL;
SELECT COLOR, COUNT(*)into v_max_color, v_max_sales FROM CAR_SALE
JOIN CAR ON CAR.VIN = CAR_SALE.VIN
WHERE EXTRACT(MONTH FROM SaleDate) = month
GROUP BY COLOR
ORDER BY COUNT(*) DESC
FETCH FIRST 1 ROW ONLY;
DBMS_OUTPUT.PUT_LINE('In ' || v_month ||', the most popular car color was
'|| v_max_color);
month := month+1;
EXIT WHEN month = 13;
END LOOP;
END;
```

Which customer paid the most for a car in each month?

This question reveals the top customer by month, which could be beneficial to the dealership because the employee that sold to the top customer could be rewarded with a bonus or other incentive for that month. This bonus for top sale of the month could also help the car dealership motivate other employees to work harder in order to earn the title of top monthly sales person, and receive the bonus.

```
DECLARE
v_max_price NUMBER := 0;
    v_custfname VARCHAR2(50);
    v_lastname VARCHAR2(50);
    month NUMBER := 1;
    v_month varchar2(16);
BEGIN
LOOP
SELECT TO_CHAR(TO_DATE(month, 'MM'), 'Month') AS "Month Name" into v_month
FROM DUAL;
SELECT MAX(PRICE), CUSTFIRSTNAME, CUSTLASTNAME INTO v_max_price,
v_custfname, v_lastname FROM CAR_SALE
JOIN CUSTOMER ON CAR_SALE.CUSTOMERID = customer.customerid
WHERE EXTRACT(MONTH FROM SaleDate) = month
GROUP BY customer.custfirstname, custlastname
ORDER BY MAX(PRICE) DESC
FETCH FIRST 1 ROW ONLY;
DBMS_OUTPUT.PUT_LINE('In ' || v_month || ', ' || v_custfname || ' ' ||
v_lastname || ' paid the most for a car');
month := month+1;
EXIT WHEN month = 13;
END LOOP;
END;
```

Which customers are buying the most cars every year?

This question tells you who are likely going to be your top customers. You can cater toward these customers by making sure they know that you value them. You can also offer them deals since they’ve shown interest in the cars.

```
CREATE OR REPLACE PACKAGE car_sales_pkg IS
  -- Function to get the top customers for a given year
  FUNCTION get_top_customers(year_in IN NUMBER)
    RETURN SYS_REFCURSOR;
  -- Procedure to print the top customers for a given year
  PROCEDURE print_top_customers(year_in IN NUMBER);

END car_sales_pkg;
/

CREATE OR REPLACE PACKAGE BODY car_sales_pkg IS
  -- Function to get the top customers for a given year
  FUNCTION get_top_customers(year_in IN NUMBER)
    RETURN SYS_REFCURSOR
  IS
    top_customers SYS_REFCURSOR;
  BEGIN
    OPEN top_customers FOR
      SELECT c.custfirstname || ' ' || c.custlastname as custname, COUNT(*)
as num_cars_bought
      FROM customer c
      JOIN car_sale cs ON c.customerid = cs.customerid
      WHERE EXTRACT(YEAR FROM cs.saledate) = year_in
      GROUP BY c.custfirstname || ' ' || c.custlastname
      ORDER BY num_cars_bought DESC;
    RETURN top_customers;
  END;
  -- Procedure to print the top customers for a given year
  PROCEDURE print_top_customers(year_in IN NUMBER)
  IS
    top_customers SYS_REFCURSOR;
    custfirstname customer.custfirstname%TYPE;
    num_cars_bought NUMBER;
  BEGIN
    top_customers := get_top_customers(year_in);
    DBMS_OUTPUT.PUT_LINE('Top customers for year ' || year_in || ':');
    LOOP
      FETCH top_customers INTO custfirstname, num_cars_bought;
      EXIT WHEN top_customers%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE(custfirstname || ' bought ' || num_cars_bought
|| ' cars.');
    END LOOP;
    CLOSE top_customers;
  END;
END car_sales_pkg;
```

What year of car is getting sold the most in each year?

This question tells you the cars of which year that are getting sold the most. This helps so that we can market cars of that year more, since they are the most popular. It will also help since we can buy more cars of that year, since they are in demand.

```
CREATE OR REPLACE PACKAGE carss_sales_pkg IS
  -- Function to get the top year of car sold for each year
  FUNCTION get_top_year_of_car(year_in IN NUMBER)
    RETURN CAR.year%TYPE;

  -- Procedure to print the top year of car sold for each year
  PROCEDURE print_top_year_of_car;

END carss_sales_pkg;

CREATE OR REPLACE PACKAGE BODY carss_sales_pkg IS
  -- Function to get the top year of car sold for each year
  FUNCTION get_top_year_of_car(year_in IN NUMBER)
    RETURN CAR.year%TYPE
  IS
    top_year CAR.year%TYPE;
  BEGIN
    SELECT year
    INTO top_year
    FROM (
      SELECT c.year, COUNT(*) as num_cars_sold
      FROM car c
      JOIN car_sale s ON c.vin = s.vin
      WHERE EXTRACT(YEAR FROM s.saledate) = year_in
      GROUP BY c.year
      ORDER BY num_cars_sold DESC
    ) WHERE ROWNUM = 1;
    RETURN top_year;
  END;

  -- Procedure to print the top year of car sold for each year
  PROCEDURE print_top_year_of_car
  IS
    year_list SYS_REFCURSOR;
    year_in NUMBER;
    top_year CAR.year%TYPE;
  BEGIN
```

```
    OPEN year_list FOR
      SELECT DISTINCT
    EXTRACT(YEAR FROM "A1"."SALEDATE") "EXTRACT(YEARFROMSALEDATE)"
FROM
    "CAR_SALE" "A1"
ORDER BY
    EXTRACT(YEAR FROM "A1"."SALEDATE");
    LOOP
      FETCH year_list INTO year_in;
      EXIT WHEN year_list%NOTFOUND;
      top_year := get_top_year_of_car(year_in);
      DBMS_OUTPUT.PUT_LINE('Year ' || year_in || ': ' || top_year);
    END LOOP;
    CLOSE year_list;
  END;
END carss_sales_pkg;
```

Team Contributions

Sam Cyr

- Presentation
  - Slides for questions 1-5
- Code
  - Created Car table, data, and care_sale data.
  - Wrote functions and procedures for questions 1-5
- Report
  - Solutions for questions 1-5

Sumair Ahmed

- Presentation
  - Tables, and Questions 6-8
- Code
  - Created Employee table and example data
  - Wrote solutions for questions 6-8

Ryan Flanagan

- Presentation
  - Intro slides, background info, motivations, solutions for questions 9 and 10
- Code
  - Created Inserts for Customer table
  - Wrote packages, functions and executions for questions 9 and 10
- Report
  - Solutioins for questions 9 and 10