**Customer data analysis:**

compCustomer (accountNo in number, monthOfService in date yearOfService in date)

*Algorithm:* Compute the report for analysis of usage of resources by customer in a duration of time.

Gas and electricity usage for the month specified will be calculated from ‘monthly\_bill’ table by comparing bill\_date.

Usage of resource for ‘monthOfService’ last year can be computed by comparing the year from ‘bill\_date’ with ‘yearOfService’ decrementing by 1.

Average usage and bottom 20% usage of customer in the same zip-code will be computed by calling function usage(). Parameters for the function will be computed in the calling block.

*Input:*1) accountNo

2) monthOfService

3) yearOfService

*Output:* Usage for the month.

Usage for the month in previous year

Average usage of customers in the same zip-code

Average usage of bottom 20% users

CREATE OR REPLACE PROCEDURE compCustomer (accountNo IN int, monthOfService IN int, yearOfService IN int) IS

customer\_name varchar (50); --Indicates who owns the account.

zip\_code varchar (30); --Holds the zip code which will be used as an input for calling feature 13.

previous\_year int; --Holds the previous year where yearOfService is the current year.

current\_eu number; --Holds the value of electricity usage for the current month and year (as specified in the parameters).

current\_gu number; --Holds the value of gas usage for the current month and year (as specified in the parameters).

lastyr\_eu number; --Holds the value of electricity usage for the same month in the previous year (a.k.a previous\_year).

lastyr\_gu number; --Holds the value of gas usage for the same month in the previous year (a.k.a previous\_year).

month\_of\_service varchar(30); --Translates the monthOfService input parameter monthOfService into a string.

BEGIN

previous\_year := yearOfService - 1;

SELECT electricity\_usage, gas\_usage, zip, cname INTO current\_eu, current\_gu, zip\_code, customer\_name

FROM house\_address h

INNER JOIN account a ON (h.hid = a.hid)

INNER JOIN monthly\_bill m ON (a.aid = m.aid)

INNER JOIN customer c ON (a.cid = c.cid)

WHERE a.aid = accountNo

AND extract (year FROM bill\_date) = yearOfService

AND extract (month FROM bill\_date) = monthOfService;

SELECT electricity\_usage, gas\_usage INTO lastyr\_eu, lastyr\_gu

FROM house\_address h

INNER JOIN account a ON (h.hid = a.hid)

INNER JOIN monthly\_bill m ON (a.aid = m.aid)

WHERE a.aid = accountNo

AND extract (year FROM bill\_date) = previous\_year

AND extract (month FROM bill\_date) = monthOfService;

IF monthOfService = 1 THEN

month\_of\_service := 'January';

ELSIF monthOfService = 2 THEN

month\_of\_service := 'February';

ELSIF monthOfService = 3 THEN

month\_of\_service := 'March';

ELSIF monthOfService = 4 THEN

month\_of\_service := 'April';

ELSIF monthOfService = 5 THEN

month\_of\_service := 'May';

ELSIF monthOfService = 6 THEN

month\_of\_service := 'June';

ELSIF monthOfService = 7 THEN

month\_of\_service := 'July';

ELSIF monthOfService = 8 THEN

month\_of\_service := 'August';

ELSIF monthOfService = 9 THEN

month\_of\_service := 'September';

ELSIF monthOfService = 10 THEN

month\_of\_service := 'October';

ELSIF monthOfService = 11 THEN

month\_of\_service := 'November';

ELSIF monthOfService = 12 THEN

month\_of\_service := 'December';

ELSE

dbms\_output.put\_line('Invalid date.');

END IF;

dbms\_output.put\_line('Comparative Study for account #: ' || accountNo);

dbms\_output.put\_line('Account owner: ' || customer\_name);

dbms\_output.put\_line('Electricity usage in ' || month\_of\_service || ', ' || yearOfService || ' is ' || current\_eu || '.');

dbms\_output.put\_line('Gas usage in ' || month\_of\_service || ', ' || yearOfService || ' is ' || current\_gu || '.');

dbms\_output.put\_line('Electricity usage in ' || month\_of\_service || ', ' || previous\_year || ' is ' || lastyr\_eu || '.');

dbms\_output.put\_line('Gas usage in ' || month\_of\_service || ', ' || previous\_year || ' is ' || lastyr\_gu || '.');

usage (zip\_code, monthOfService, yearOfService); --Calls feature 13 to complete the rest of the problem.

EXCEPTION

WHEN no\_data\_found THEN

DBMS\_OUTPUT.PUT\_LINE('No usage data found.');

END;

EXEC compCustomer (10, 10, 2016);