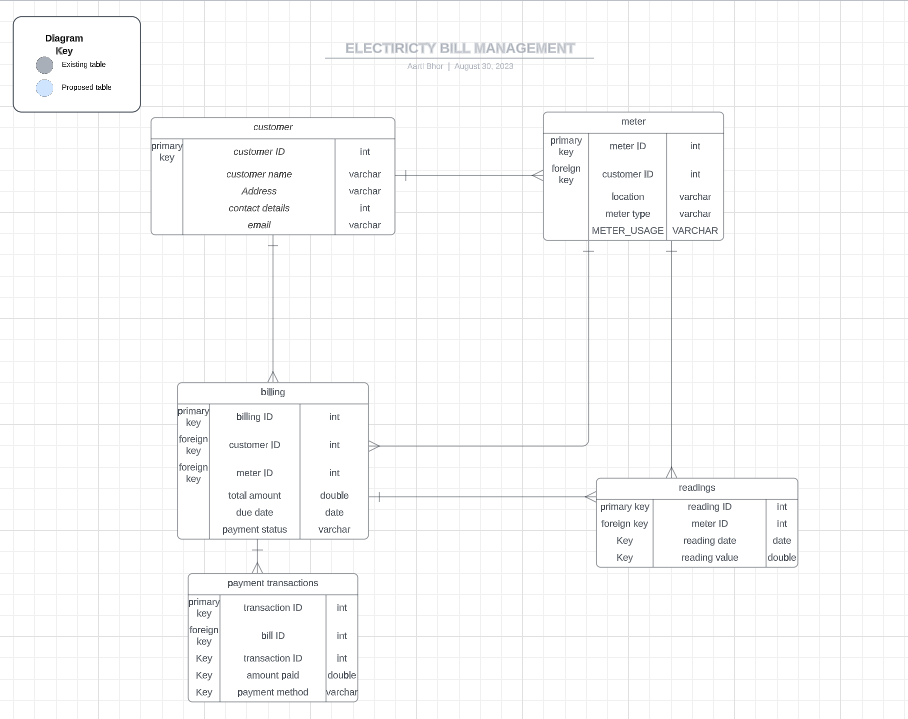
**DBMS PROJECT SEM3**

* TABLES REQUIRED:

1. Customers: cols-> customer id(primary key), name, address, contact details, email
2. Meters: cols-> meter id(primary key),customer id(foreign key),update last reading date, meter type
3. Readings: cols->reading id(primary key), meter id(foreign key), reading date, reading value
4. Billing: cols-> billing id(primary key), customer id(foreign key), meter id(foreign key), total amount, due date, payment status
5. Payment Transactions: cols-> transaction id(primary key), bill id(foreign key), transaction date, amount paid, payment method

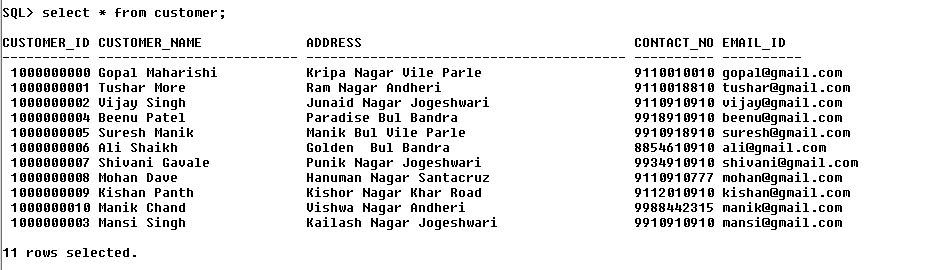
* Trigger: *(usage->to automatically calculate and update relevant data upon certain events)* to calculate the total amount in bills in billing table when a new bill is generated based on meter readings.
* Cursor: *(usage-> to retrieve and manipulate data in database)* to be used in procedure and trigger to process data by row to row
* Procedure: *(usage-> to perform specific task)* to record meter readings
* Transaction: (usage->to ensure data consistency and integrity and implementing error handling)
* DIAGRAM REPRESENTATION:



PL/SQL CODE:

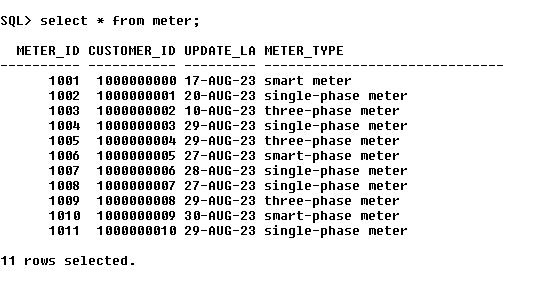
|  |
| --- |
| -- Customer:  CREATE TABLE customer(customer\_id number(10) primary key, customer\_name varchar(50), address varchar2(70),contact\_no number(10) unique, email\_id varchar2(20) );  INSERT INTO customer values(1000000000,’Gopal Maharishi’, ‘Kripa Nagar Vile Parle’,9110010010,’gopal@gmail.com’ );  INSERT INTO customer values(1000000001,’Tushar More’, ‘Ram Nagar Andheri’,9110018810,’tushar@gmail.com’ );  INSERT INTO customer values(1000000002,’Vijay Singh’, ‘Junaid Nagar Jogeshwari’,9110910910,’vijay@gmail.com’ );  INSERT INTO customer values(1000000003,’Mansi Singh’, ‘Kailash Nagar Jogeshwari’,9910910910,’mansi@gmail.com’ );  INSERT INTO customer values(1000000004,’Beenu Patel’, ‘Paradise Bul Bandra’,9918910910,’beenu@gmail.com’ );  INSERT INTO customer values(1000000005,’Suresh Manik’, ‘Manik Bul Vile Parle’,9910918910,’suresh@gmail.com’ );  INSERT INTO customer values(1000000006,’Ali Shaikh’, ‘Golden Bul Bandra’,8854610910,’ali@gmail.com’ );  INSERT INTO customer values(1000000007,’Shivani Gavale’, ‘Punik Nagar Jogeshwari’,9934910910,’shivani@gmail.com’ );  INSERT INTO customer values(1000000008,’Mohan Dave’, ‘Hanuman Nagar Santacruz’,9110910777,’mohan@gmail.com’ );  INSERT INTO customer values(1000000009,’Kishan Panth’, ‘Kishor Nagar Khar Road’,9122210910,’kishan@gmail.com’ );  INSERT INTO customer values(1000000010,’Manik Chand’, ‘Vishwa Nagar Andheri’,9112010910,’manik@gmail.com’ );  SELECT \* FROM customer; |

OUTPUT:



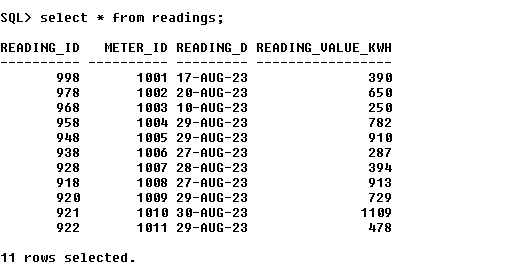
|  |
| --- |
| --Meter:  CREATE TABLE meter(meter\_id number primary key, customer\_id number(10) ,update\_last\_reading\_date date, meter\_type varchar(50),foreign key (customer\_id) references customer(customer\_id));  INSERT INTO meter values(1001, 1000000000, to\_date(’17-Aug-23’,’DD-MON-YY’),’smart meter’);  INSERT INTO meter values(1002, 1000000001, to\_date(’20-Aug-23’,’DD-MON-YY’),’single-phase meter’);  INSERT INTO meter values(1003, 1000000002, to\_date(’10-Aug-23’,’DD-MON-YY’),’three-phase meter’);  INSERT INTO meter values(1004, 1000000003, to\_date(’29-Aug-23’,’DD-MON-YY’),’single-phase meter’);  INSERT INTO meter values(1005, 1000000004, to\_date(’29-Aug-23’,’DD-MON-YY’),’three-phase meter’);  INSERT INTO meter values(1006, 1000000005, to\_date(’27-Aug-23’,’DD-MON-YY’),’smart-phase meter’);  INSERT INTO meter values(1007, 1000000006, to\_date(’28-Aug-23’,’DD-MON-YY’),’single-phase meter’);  INSERT INTO meter values(1008, 1000000007, to\_date(’27-Aug-23’,’DD-MON-YY’),’single-phase meter’);  INSERT INTO meter values(1009, 1000000008, to\_date(’29-Aug-23’,’DD-MON-YY’),’three-phase meter’);  INSERT INTO meter values(1010, 1000000009, to\_date(’30 -Aug-23’,’DD-MON-YY’),’smart-phase meter’);  INSERT INTO meter values(1011, 1000000010, to\_date(’29-Aug-23’,’DD-MON-YY’),’single-phase meter’);  SELECT \* FROM meter; |

OUTPUT:



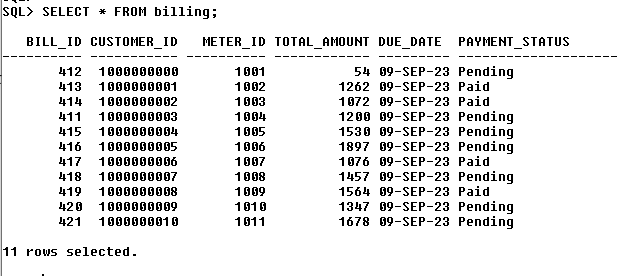
|  |
| --- |
| --Readings:  CREATE TABLE readings (reading\_id number primary key, meter\_id number, reading\_date date, reading\_value\_Kwh decimal(10, 2),foreign key (meter\_id) references meter(meter\_id));  INSERT INTO readings values( 998, 1001,to\_date(’17-Aug-23’,’DD-MON-YY’),390);  INSERT INTO readings values( 978, 1002,to\_date(’20-Aug-23’,’DD-MON-YY’),650);  INSERT INTO readings values( 968, 1003,to\_date(’10-Aug-23’,’DD-MON-YY’),250);  INSERT INTO readings values( 958, 1004,to\_date(’29-Aug-23’,’DD-MON-YY’),782);  INSERT INTO readings values( 948, 1005,to\_date(’29-Aug-23’,’DD-MON-YY’),910);  INSERT INTO readings values( 938, 1006,to\_date(’27-Aug-23’,’DD-MON-YY’),287);  INSERT INTO readings values( 928, 1007,to\_date(’28-Aug-23’,’DD-MON-YY’),394);  INSERT INTO readings values( 918, 1008,to\_date(’27-Aug-23’,’DD-MON-YY’),913);  INSERT INTO readings values( 920, 1009,to\_date(’29-Aug-23’,’DD-MON-YY’),729);  INSERT INTO readings values( 921, 1010,to\_date(’30 -Aug-23’,’DD-MON-YY’),1109);  INSERT INTO readings values( 922, 1011,to\_date(’29-Aug-23’,’DD-MON-YY’),478);  SELECT \* FROM readings; |

OUTPUT:



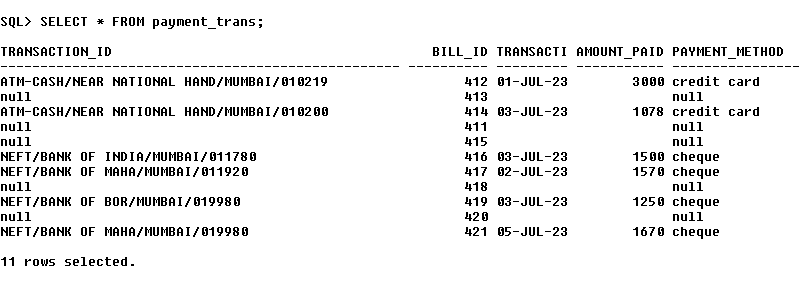
|  |
| --- |
| --Billling:  CREATE TABLE billing(bill\_id number primary key,  customer\_id number(10),  meter\_id number,  total\_amount decimal(10,2),  due\_date date,  payment\_status varchar (20) ,  foreign key (meter\_id)references meter(meter\_id));  INSERT INTO billing values(412, 1000000000, 1001,54,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  INSERT INTO billing values(413, 1000000001, 1002 ,1262,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Paid’);  INSERT INTO billing values(414, 1000000002, 1003,1072,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Paid’);  INSERT INTO billing values(411, 1000000003, 1004,1200,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  INSERT INTO billing values(415, 1000000004, 1005,1530,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  INSERT INTO billing values(416, 1000000005, 1006, 1897,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  INSERT INTO billing values(417, 1000000006, 1007,1076,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Paid’);  INSERT INTO billing values(418, 1000000007, 1008,1457,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  INSERT INTO billing values(419, 1000000008, 1009,1564,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Paid’);  INSERT INTO billing values(420, 1000000009, 1010 ,1347,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  INSERT INTO billing values(421, 1000000010, 1011,1678,to\_date(‘9-Sep-23’,’DD-MON-YY’),’Pending’);  SELECT \* FROM billing; |

OUTPUT:



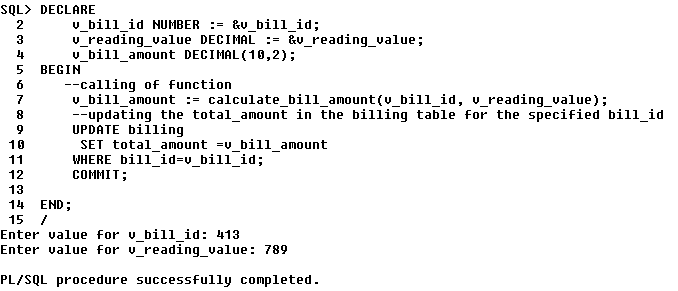
|  |
| --- |
| --Payment transactions:  CREATE TABLE payment\_trans(transaction\_id varchar2(50) ,  bill\_id number ,  transaction\_date date,  amount\_paid decimal(10,2),  payment\_method varchar(20),  foreign key (bill\_id) references billing(bill\_id));  INSERT INTO payment\_trans values(‘ATM-CASH/NEAR NATIONAL HAND/MUMBAI/010219’, 412,to\_date(‘01-Jul-23’,’DD-MON-YY’),3000,’credit card’ );  INSERT INTO payment\_trans values(‘null’, 413,null,null,’null’ );  INSERT INTO payment\_trans values(‘ATM-CASH/NEAR NATIONAL HAND/MUMBAI/010200’, 414, to\_date(‘03-Jul-23’,’DD-MON-YY’),1078,’credit card’ );  INSERT INTO payment\_trans values(‘null’, 411,null,null,’null’ );  INSERT INTO payment\_trans values(‘null’, 415,null,null,’null’ );  INSERT INTO payment\_trans values(‘NEFT/BANK OF INDIA/MUMBAI/011780’, 416, to\_date(‘03-Jul-23’,’DD-MON-YY’),1500,’cheque’ );  INSERT INTO payment\_trans values(‘NEFT/BANK OF MAHA/MUMBAI/011920’, 417, to\_date(‘02-Jul-23’,’DD-MON-YY’),1570,’cheque’ );  INSERT INTO payment\_trans values(‘null’, 418,null,null,’null’ );  INSERT INTO payment\_trans values(‘NEFT/BANK OF BOR/MUMBAI/019980’, 419, to\_date(‘03-Jul-23’,’DD-MON-YY’),1250,’cheque’ );  INSERT INTO payment\_trans values(‘null’, 420,null,null,’null’ );  INSERT INTO payment\_trans values(‘NEFT/BANK OF MAHA/MUMBAI/019980’, 421, to\_date(‘05-Jul-23’,’DD-MON-YY’),1670,’cheque’ );  SELECT \* FROM payment\_trans; |

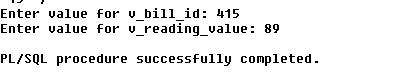
OUTPUT:

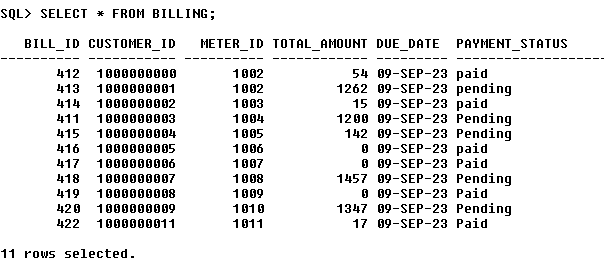


|  |
| --- |
| --FUNCTION🡪on entering meter\_id and reading\_value the total\_amount of billing table will be updated. |
| --FUNCTION: TO CALCULATE BILL AMOUNT  CREATE OR REPLACE FUNCTION calculate\_bill\_amount(p\_bill\_id IN NUMBER, p\_reading\_value IN DECIMAL)  RETURN DECIMAL  AS  v\_total\_amount DECIMAL;  BEGIN  v\_total\_amount :=p\_reading\_value \* 1.60;  RETURN v\_total\_amount;  END;  / |
| --CALLING OF FUNCTION:  DECLARE  v\_bill\_id NUMBER := &v\_bill\_id;  v\_reading\_value DECIMAL := &v\_reading\_value;  v\_bill\_amount DECIMAL(10,2);  BEGIN  --calling of function  v\_bill\_amount := calculate\_bill\_amount(v\_bill\_id, v\_reading\_value);  --updating the total\_amount in the billing table for the specified bill\_id  UPDATE billing  SET total\_amount =v\_bill\_amount  WHERE bill\_id=v\_bill\_id;  COMMIT;    END;  / |

OUTPUT:

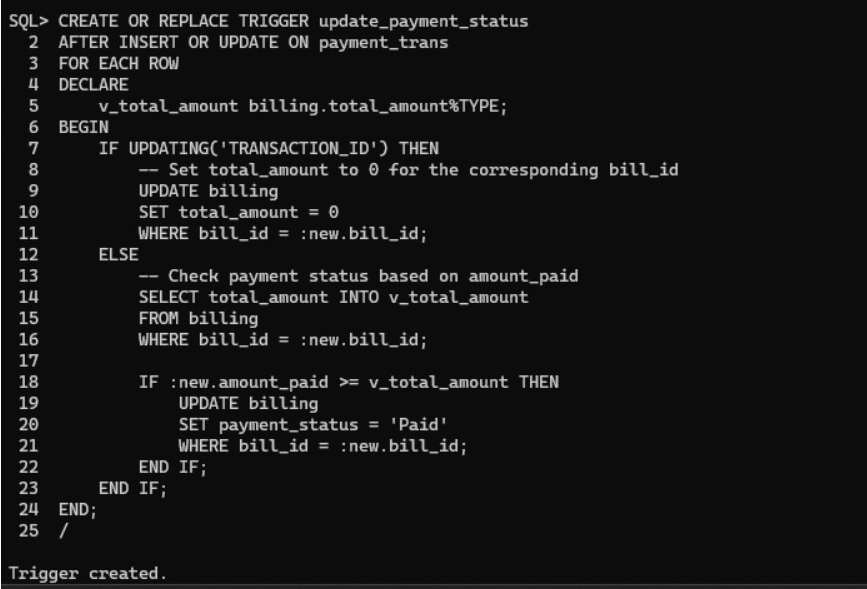




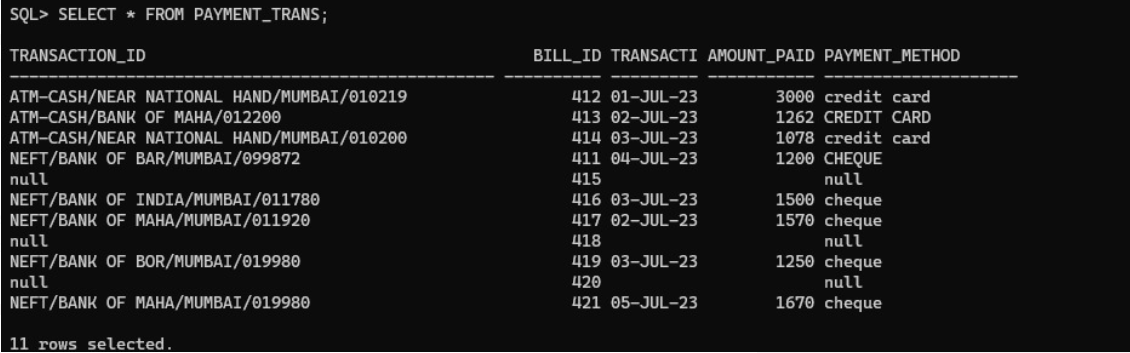


|  |
| --- |
| --TRIGGER🡪 SUCH THAT WHENEVER UPDATING TRANSACTION\_ID OF PAYMENT\_TRANS TABLE THE PAYMENT\_STATUS= PAID AND THE TOTAL AMOUNT=0 OF BILLING TABLE |
| --TRIGGER:  CREATE OR REPLACE TRIGGER update\_payment\_status  AFTER INSERT OR UPDATE ON payment\_trans  FOR EACH ROW  DECLARE  v\_total\_amount billing.total\_amount%TYPE;  BEGIN  SELECT total\_amount INTO v\_total\_amount  FROM billing  WHERE bill\_id = :new.bill\_id;  IF :new.amount\_paid >= v\_total\_amount THEN  UPDATE billing  SET payment\_status = 'Paid'  WHERE bill\_id = :new.bill\_id;  END IF;  END;  / |

OUTPUT:



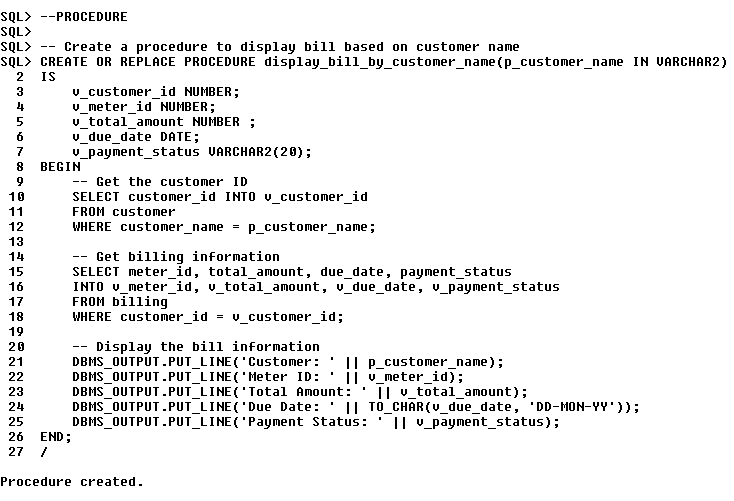


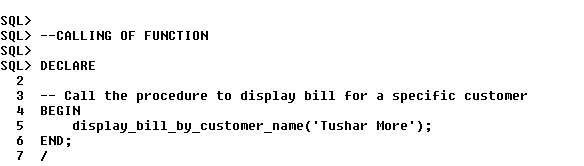


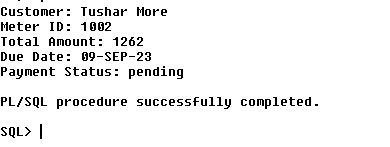


|  |
| --- |
| --PROCEDURE🡪 display bill based on customer\_name |
| --PROCEDURE  -- Create a procedure to display bill based on customer name  CREATE OR REPLACE PROCEDURE display\_bill\_by\_customer\_name(p\_customer\_name IN VARCHAR2)  IS  v\_customer\_id NUMBER;  v\_meter\_id NUMBER;  v\_total\_amount NUMBER ;  v\_due\_date DATE;  v\_payment\_status VARCHAR2(20);  BEGIN  -- Get the customer ID  SELECT customer\_id INTO v\_customer\_id  FROM customer  WHERE customer\_name = p\_customer\_name;  -- Get billing information  SELECT meter\_id, total\_amount, due\_date, payment\_status  INTO v\_meter\_id, v\_total\_amount, v\_due\_date, v\_payment\_status  FROM billing  WHERE customer\_id = v\_customer\_id;  -- Display the bill information  DBMS\_OUTPUT.PUT\_LINE('Customer: ' || p\_customer\_name);  DBMS\_OUTPUT.PUT\_LINE('Meter ID: ' || v\_meter\_id);  DBMS\_OUTPUT.PUT\_LINE('Total Amount: ' || v\_total\_amount);  DBMS\_OUTPUT.PUT\_LINE('Due Date: ' || TO\_CHAR(v\_due\_date, 'DD-MON-YY'));  DBMS\_OUTPUT.PUT\_LINE('Payment Status: ' || v\_payment\_status);  END;  / |
| --CALLING OF FUNCTION  DECLARE  -- Call the procedure to display bill for a specific customer  BEGIN  display\_bill\_by\_customer\_name(’Tushar More’);  END;  / |

OUTPUT:

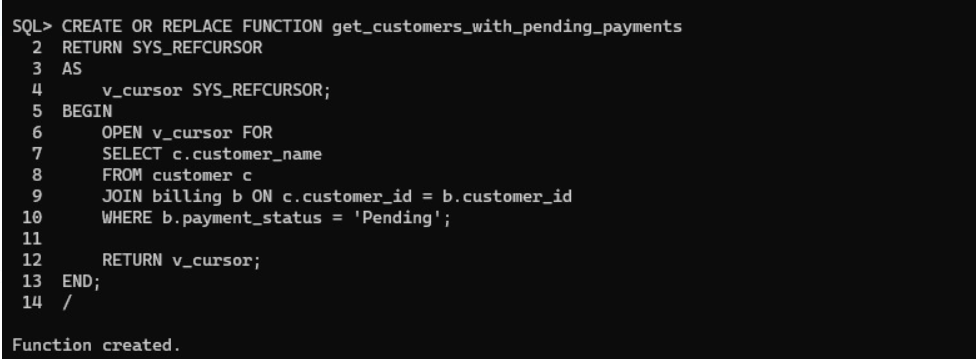


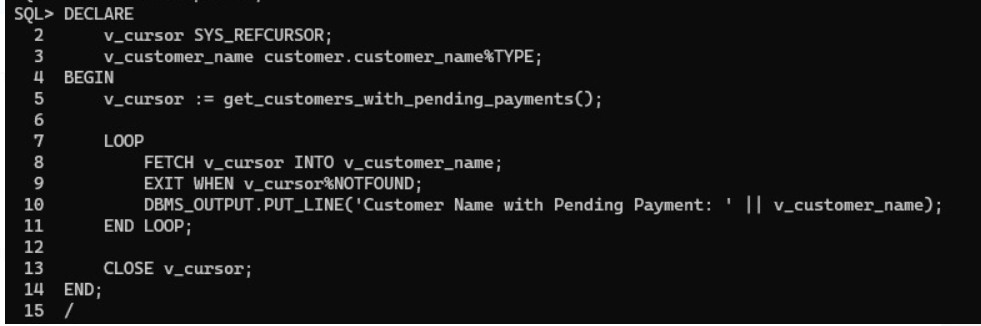


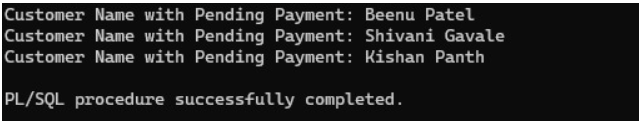


|  |
| --- |
| FUNCTION🡪 TO FIND CUSTOMER WITH PENDING BILLS |
| --FUNCTION  DECLARE  v\_cursor SYS\_REFCURSOR;  v\_customer\_name customer.customer\_name%TYPE;  BEGIN  v\_cursor := get\_customers\_with\_pending\_payments();    LOOP  FETCH v\_cursor INTO v\_customer\_name;  EXIT WHEN v\_cursor%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('Customer Name with Pending Payment: ' || v\_customer\_name);  END LOOP;    CLOSE v\_cursor;  END;  / |
| --CALLING OF FUNCTION  DECLARE  v\_cursor SYS\_REFCURSOR;  v\_customer\_name customer.customer\_name%TYPE;  BEGIN  v\_cursor := get\_customers\_with\_pending\_payments();    LOOP  FETCH v\_cursor INTO v\_customer\_name;  EXIT WHEN v\_cursor%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('Customer Name with Pending Payment: ' || v\_customer\_name);  END LOOP;    CLOSE v\_cursor;  END;  / |

OUTPUT:

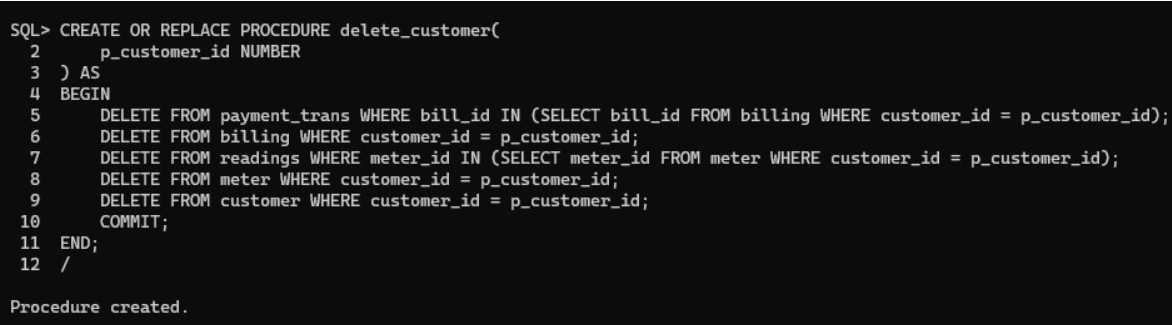


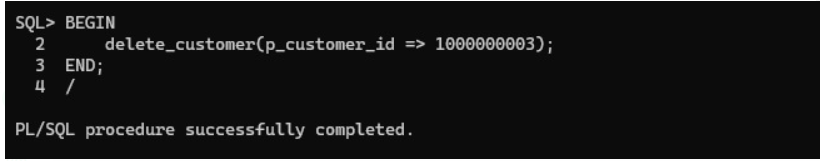


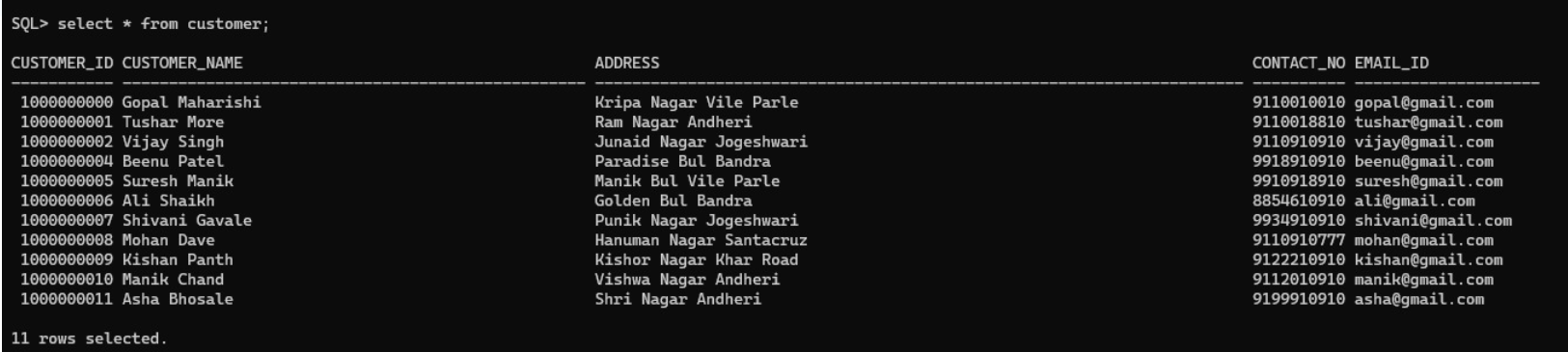


|  |
| --- |
| --PROCEDURE🡪 TO DELETE CUSTOMER ALONG WITH ITS DATA |
| --PROCEDURE  CREATE OR REPLACE PROCEDURE delete\_customer(  p\_customer\_id NUMBER  ) AS  BEGIN  DELETE FROM payment\_trans WHERE bill\_id IN (SELECT bill\_id FROM billing WHERE customer\_id = p\_customer\_id);  DELETE FROM billing WHERE customer\_id = p\_customer\_id;  DELETE FROM readings WHERE meter\_id IN (SELECT meter\_id FROM meter WHERE customer\_id = p\_customer\_id);  DELETE FROM meter WHERE customer\_id = p\_customer\_id;  DELETE FROM customer WHERE customer\_id = p\_customer\_id;  COMMIT;  END;  / |
| --CALLLING OF PROCEDURE  BEGIN  delete\_customer(p\_customer\_id => 1000000003);  END;  / |

OUTPUT:

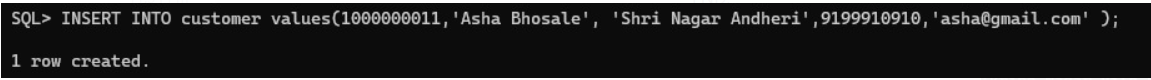






|  |
| --- |
| TRANSACTION🡪 TO INSERT DATA IN CUSTOMER TABLE |
| --TRANSACTION  INSERT INTO customer values(1000000011,'Asha Bhosale', 'Shri Nagar Andheri',9199910910,'asha@gmail.com' ); |

OUTPUT:



|  |
| --- |
| TRANSACTION🡪 TO UPDATE DATA IN CUSTOMER TABLE |
| --TRANSACTION update customer set customer\_name='Meena Patel', address='Kripe Nagar Vile Parle', contact\_no=8877787779, email\_id='meena@gmail.com' where customer\_id=1000000005; |

OUTPUT:

