EX-07 **22-02-25**

CREATION OF SIMPLE PROCEDURES AND FUNCTIONS WITH EXCEPTIONS USING PL/SQL

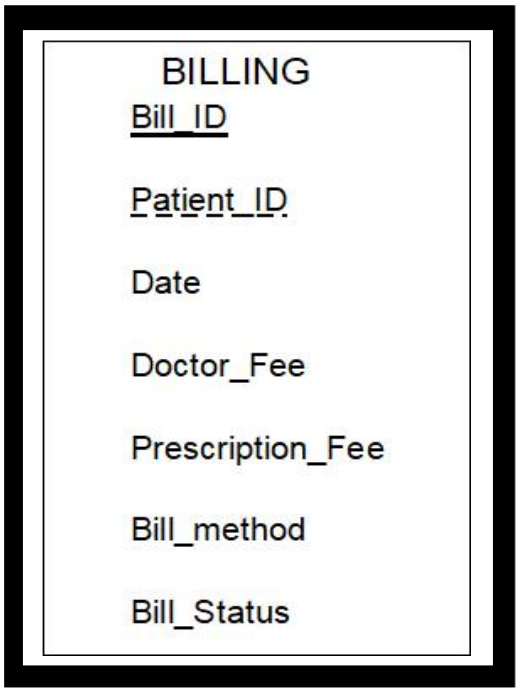
AIM:

To create

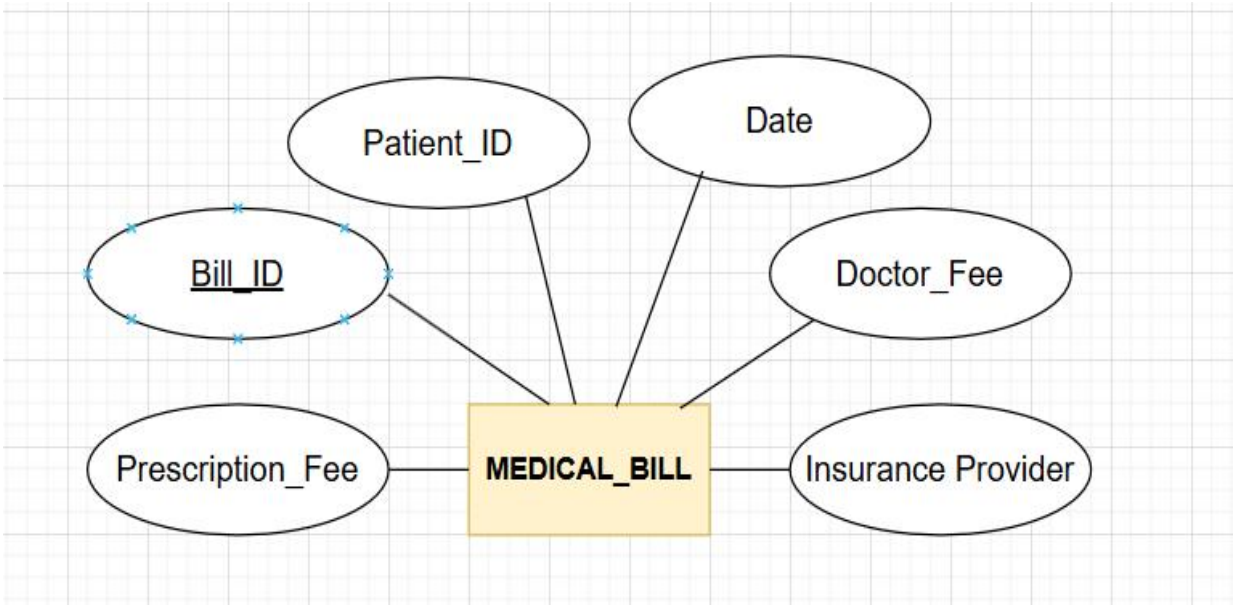
· A stored procedure for updating the total and result values of the given Medical_Bill along with exception of user defined and built-in types.

· A function to calculate the Medical_Bill using ORACLE - PL/SQL.

SCHEMA DIAGRAM:



ER DIAGRAM:



EX-07 **22-02-25**

DATA DICTIONARY:

1.Medical Bill Relation

Attribute	Datatype	Constraints	Remarks
Bill_ID	NUMBER	PRIMARY KEY	Unique identifier for each bill
Patient_Name	VARCHAR(100)	NOT NULL	Full name of the patient
Date	DATE	NOT NULL	Date when bill was generated
Doctor_Fee	NUMBER(10,2)	CHECK (Doctor_fee>=0)	Consultation charges
Prescription_Fee	NUMBER(10,2)	CHECK (Prescription_Fee >= 0)	Cost of prescribed medicines
Insurance_Provider	NUMBER(10,2)	DEFAULT 'No Insurance'	Name of the insurance company (if any)

CODING:

CREATION AND INSERTION

```
CREATE TABLE medical_bill (
    bill_id NUMBER PRIMARY KEY,
    patient_name VARCHAR2(100) NOT NULL,
    bill_date DATE DEFAULT SYSDATE,
    doctor_fee NUMBER CHECK (doctor_fee >= 0),
    prescription_fee NUMBER CHECK (prescription_fee >= 0),
    total_fee NUMBER GENERATED ALWAYS AS (doctor_fee + prescription_fee) VIRTUAL,
    insurance_provider VARCHAR2(100)
};
```

```
INSERT INTO medical_bill (bill_id, patient_name, bill_date, doctor_fee, prescription_fee,
insurance_provider)
VALUES (101, 'Rahul Sharma', TO_DATE('2024-03-09', 'YYYY-MM-DD'), 500, 200, 'Star
Health');

INSERT INTO medical_bill (bill_id, patient_name, bill_date, doctor_fee, prescription_fee,
insurance_provider)
VALUES (102, 'Neha Kapoor', TO_DATE('2024-03-08', 'YYYY-MM-DD'), 700, 300, 'Max
Bupa');

INSERT INTO medical_bill (bill_id, patient_name, bill_date, doctor_fee, prescription_fee,
insurance_provider)
VALUES (103, 'Arjun Reddy', TO_DATE('2024-03-07', 'YYYY-MM-DD'), 600, 250, NULL);
```

PROCEDURE

```
 CREATE OR REPLACE PROCEDURE insert_medical_bill (
    p_bill_id IN NUMBER,
    p_patient_name IN VARCHAR2,
    p_bill_date IN DATE,
    p_doctor_fee IN NUMBER,
    p_prescription_fee IN NUMBER,
    p_insurance_provider IN VARCHAR2 DEFAULT 'No Insurance'
)
IS
    v_total_fee NUMBER;
BEGIN

    v_total_fee := p_doctor_fee + p_prescription_fee;

    INSERT INTO medical_bill (Bill_ID, Patient_Name, Bill_Date, Doctor_Fee,
Prescription_Fee, Total_Fee, Insurance_Provider)
    VALUES (p_bill_id, p_patient_name, p_bill_date, p_doctor_fee, p_prescription_fee,
v_total_fee, p_insurance_provider);

    DBMS_OUTPUT.PUT_LINE('Medical bill for ' || p_patient_name || ' inserted successfully.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END insert_medical_bill;
/
```

EX-07 **22-02-25**

FUNCTION

```
CREATE OR REPLACE FUNCTION calculate_final_amount (p_bill_id IN NUMBER)
RETURN NUMBER
IS
   v_total_fee NUMBER;
   v_insurance_provider VARCHAR2(100);
   v_discount NUMBER := 0;
   v_final_amount NUMBER;
BEGIN
   -- Fetch total fee and insurance provider
   SELECT Total_Fee, COALESCE(Insurance_Provider, 'No Insurance')
   INTO v_total_fee, v_insurance_provider
   FROM medical_bill
   WHERE Bill_ID = p_bill_id;

   -- Apply 20% discount if insurance is available
   IF v_insurance_provider != 'No Insurance' THEN
      v_discount := v_total_fee * 0.20;
   END IF;

   -- Calculate final amount
   v_final_amount := v_total_fee - v_discount;

   RETURN v_final_amount;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RETURN -1;  -- Return -1 if no bill is found
   WHEN OTHERS THEN
      RETURN -2;  -- Return -2 for other errors
END calculate_final_amount;
/
```

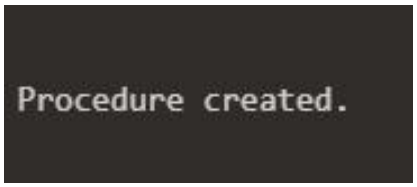
Anonymous Block

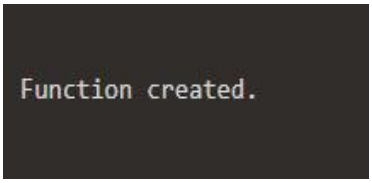
```
DECLARE
   v_bill_id NUMBER := TO_NUMBER(:P1_BILL_ID);  -- Replace with actual APEX input field
   v_patient_name VARCHAR2(100);
   v_bill_date DATE;
   v_total_fee NUMBER;
   v_insurance_provider VARCHAR2(100);
   v_final_amount NUMBER;
BEGIN
   SELECT Patient_Name, Bill_Date, Total_Fee, COALESCE(Insurance_Provider, 'No
Insurance')
   INTO v_patient_name, v_bill_date, v_total_fee, v_insurance_provider
   FROM medical_bill
   WHERE Bill_ID = v_bill_id;

   v_final_amount := calculate_final_amount(v_bill_id);
```

```
    DBMS_OUTPUT.PUT_LINE('Bill ID: ' || v_bill_id);
    DBMS_OUTPUT.PUT_LINE('Patient Name: ' || v_patient_name);
    DBMS_OUTPUT.PUT_LINE('Bill Date: ' || TO_CHAR(v_bill_date, 'YYYY-MM-DD'));
    DBMS_OUTPUT.PUT_LINE('Total Fee: ' || v_total_fee);
    DBMS_OUTPUT.PUT_LINE('Insurance Provider: ' || v_insurance_provider);
    DBMS_OUTPUT.PUT_LINE('Final Amount to be Paid: ' || v_final_amount);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No record found for Bill ID: ' || v_bill_id);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

OUTPUT

BILL_ID	PATIENT_NAME	BILL_DATE	DOCTOR_FEE	PRESCRIPTION_FEE	TOTAL_FEE	INSURANCE_PROVIDER
101	Rahul Sharma	03-09-2024	500	200	700	Star Health
102	Neha Kapoor	03-08-2024	700	300	1000	Max Bupa
103	Arjun Reddy	03-07-2024	600	250	850	






```
Bill ID: 102
Patient Name: Neha Kapoor
Bill Date: 2024-03-08
Total Fee: 1000
Insurance Provider: Max Bupa
Final Amount to be Paid: 800

Statement processed.
```

```
Bill ID: 103
Patient Name: Arjun Reddy
Bill Date: 2024-03-07
Total Fee: 850
Insurance Provider: No Insurance
Final Amount to be Paid: 850

Statement processed.
```

EX-07 **22-02-25**

RESULT:

Hence Simple Procedures and Functions are created with exceptions using PL/SQL for medical bill in Health clinicManagement system.

RECORD RUBRICS:

Parameters	**Marks**
Perfection in Design/Queries (10)	
Completion (05)	
Neatness (05)	
Prompt Timing (05)	
Total (25)	