**tables**

CREATE TABLE Flats (

FlatID NUMBER(4) CONSTRAINT FLAT\_ID\_PK PRIMARY KEY

);

CREATE TABLE Shops (

ShopID NUMBER(3) CONSTRAINT SHOP\_ID\_PK PRIMARY KEY

);

CREATE TABLE Property (

PropertyID INT PRIMARY KEY,

PropertyType VARCHAR(20),

CONSTRAINT CheckPropertyType1 CHECK (PropertyType IN ('Flat', 'Shop'))

);

INSERT INTO Property (PropertyID, PropertyType)

SELECT FlatID, 'Flat' AS PropertyType

FROM Flats;

INSERT INTO Property (PropertyID, PropertyType)

SELECT ShopID + (SELECT MAX(PropertyID) FROM Property), 'Shop' AS PropertyType

FROM Shops;

ALTER TABLE Shops

ADD PropertyID INT,

ADD CONSTRAINT FK\_PropertyID

FOREIGN KEY (PropertyID)

REFERENCES Property(PropertyID);

ALTER TABLE Flats

ADD PropertyID INT,

ADD CONSTRAINT FK1\_PropertyID

FOREIGN KEY (PropertyID)

REFERENCES Property(PropertyID);

CREATE TABLE Halls (

HallID NUMBER(2),

BookingID INT,

DateOfBooking DATE,

Duration VARCHAR2(10),

FlatID NUMBER(4),

PRIMARY KEY (HallID, BookingID),

FOREIGN KEY (FlatID) REFERENCES Flats(FlatID) ON DELETE CASCADE

);

CREATE TABLE Maintenance (

Receiptno INT PRIMARY KEY,

PaymentDate DATE,

PropertyType VARCHAR(10),

PropertyID INT,

GeneratorFacility BOOLEAN,

Amount DECIMAL(10, 2),

Balance DECIMAL(10, 2),

FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID) ON DELETE CASCADE,

CONSTRAINT CheckPropertyType CHECK (PropertyType IN ('Flat', 'Shop'))

);

**trigger**

DELIMITER //

CREATE TRIGGER update\_balance\_insert\_trigger BEFORE INSERT ON Maintenance

FOR EACH ROW

BEGIN

IF NEW.Amount < 4500 THEN

IF NEW.GeneratorFacility = TRUE THEN

SET NEW.Balance = 5500 - NEW.Amount; -- If generator facility, calculate from 5500

ELSE

SET NEW.Balance = 4500 - NEW.Amount; -- If no generator facility, calculate from 4500

END IF;

ELSE

SET NEW.Balance = 0; -- If amount is greater than or equal to 4500, set Balance to 0

END IF;

END;

//

CREATE TRIGGER update\_balance\_update\_trigger BEFORE UPDATE ON Maintenance

FOR EACH ROW

BEGIN

IF NEW.Amount <> OLD.Amount OR NEW.GeneratorFacility <> OLD.GeneratorFacility THEN

IF NEW.Amount < 4500 THEN

IF NEW.GeneratorFacility = TRUE THEN

SET NEW.Balance = 5500 - NEW.Amount; -- If generator facility, calculate from 5500

ELSE

SET NEW.Balance = 4500 - NEW.Amount; -- If no generator facility, calculate from 4500

END IF;

ELSE

SET NEW.Balance = 0; -- If amount is greater than or equal to 4500, set Balance to 0

END IF;

END IF;

END;

//

DELIMITER ;

**view**

CREATE VIEW Maintenance\_View AS

SELECT

PropertyID,

PropertyType,

YEAR(PaymentDate) AS `Year`,

SUM(CASE WHEN MONTH(PaymentDate) = 1 THEN Amount ELSE 0 END) AS `Jan`,

SUM(CASE WHEN MONTH(PaymentDate) = 2 THEN Amount ELSE 0 END) AS `Feb`,

SUM(CASE WHEN MONTH(PaymentDate) = 3 THEN Amount ELSE 0 END) AS `Mar`,

SUM(CASE WHEN MONTH(PaymentDate) = 4 THEN Amount ELSE 0 END) AS `Apr`,

SUM(CASE WHEN MONTH(PaymentDate) = 5 THEN Amount ELSE 0 END) AS `May`,

SUM(CASE WHEN MONTH(PaymentDate) = 6 THEN Amount ELSE 0 END) AS `Jun`,

SUM(CASE WHEN MONTH(PaymentDate) = 7 THEN Amount ELSE 0 END) AS `Jul`,

SUM(CASE WHEN MONTH(PaymentDate) = 8 THEN Amount ELSE 0 END) AS `Aug`,

SUM(CASE WHEN MONTH(PaymentDate) = 9 THEN Amount ELSE 0 END) AS `Sep`,

SUM(CASE WHEN MONTH(PaymentDate) = 10 THEN Amount ELSE 0 END) AS `Oct`,

SUM(CASE WHEN MONTH(PaymentDate) = 11 THEN Amount ELSE 0 END) AS `Nov`,

SUM(CASE WHEN MONTH(PaymentDate) = 12 THEN Amount ELSE 0 END) AS `Dec`,

SUM(Amount) AS TotalAmount,

SUM(CASE WHEN (PropertyType IN ('Flat', 'Shop')) THEN Balance ELSE 0 END) AS Balance

FROM Maintenance

GROUP BY PropertyID, PropertyType, YEAR(PaymentDate);