Własciciel(Wid, Imie, Nazwisko, Adres)

Media(Mid, Rodzaj)

Oplata(Wid, Mid, DataOd, DataDo, DataWplaty, Kwota)

CREATE TABLE Wlasciciel (

Wid NUMBER PRIMARY KEY,

Imie VARCHAR2(50),

Nazwisko VARCHAR2(50),

Adres VARCHAR2(100)

);

CREATE TABLE Media (

Mid NUMBER PRIMARY KEY,

Rodzaj VARCHAR2(50)

);

CREATE TABLE Oplata (

Wid NUMBER,

Mid NUMBER,

DataOd DATE,

DataDo DATE,

DataWplaty DATE,

Kwota NUMBER(10, 2),

CONSTRAINT PK\_Oplata PRIMARY KEY (Wid, Mid, DataOd),

CONSTRAINT FK\_Oplata\_Wlasciciel FOREIGN KEY (Wid) REFERENCES Wlasciciel(Wid),

CONSTRAINT FK\_Oplata\_Media FOREIGN KEY (Mid) REFERENCES Media(Mid)

);

INSERT INTO Wlasciciel (Wid, Imie, Nazwisko, Adres) VALUES (1, 'Jan', 'Kowalski', 'Warszawa, ul. Zielona 10');

INSERT INTO Wlasciciel (Wid, Imie, Nazwisko, Adres) VALUES (2, 'Anna', 'Nowak', 'Kraków, ul. Słoneczna 15');

INSERT INTO Wlasciciel (Wid, Imie, Nazwisko, Adres) VALUES (3, 'Piotr', 'Wiśniewski', 'Gdańsk, ul. Morze 5');

INSERT INTO Media (Mid, Rodzaj) VALUES (1, 'Internet');

INSERT INTO Media (Mid, Rodzaj) VALUES (2, 'Telewizja');

INSERT INTO Media (Mid, Rodzaj) VALUES (3, 'Telefon');

INSERT INTO Oplata (Wid, Mid, DataOd, DataDo, DataWplaty, Kwota)

VALUES (1, 1, DATE '2025-01-01', DATE '2025-01-31', DATE '2025-01-05', 100.00);

INSERT INTO Oplata (Wid, Mid, DataOd, DataDo, DataWplaty, Kwota)

VALUES (2, 2, DATE '2025-01-01', DATE '2025-01-31', DATE '2025-01-10', 150.00);

INSERT INTO Oplata (Wid, Mid, DataOd, DataDo, DataWplaty, Kwota)

VALUES (3, 3, DATE '2025-01-01', DATE '2025-01-31', DATE '2025-01-15', 50.00);

SET SERVEROUTPUT ON

Zad. 1. Napisz procedurę, która wstawi właściciela o imieniu, nazwisku i  adresie zadanych na wejściu. Wid wylicz w procedurze, wstawiając właścicielowi identyfikator o 1 większy niż ostatnio wpisany. Przed wstawieniem danych upewnij się, że właściciela o zadanym imieniu, nazwisku i adresie nie ma już w tabeli. Jeśli jest, wypisz odpowiedni komunikat i nie wstawiaj danych ponownie.

CREATE OR REPLACE PROCEDURE WstawWlasciciela (v\_imie VARCHAR, v\_nazwisko VARCHAR, v\_adres VARCHAR) AS

v\_wid NUMBER;

v\_istnieje INTEGER := 0;

BEGIN

SELECT MAX(Wid) + 1 INTO v\_wid FROM Wlasciciel;

SELECT COUNT(\*) INTO v\_istnieje FROM Wlasciciel

WHERE Imie = v\_imie AND Nazwisko = v\_nazwisko AND Adres = v\_adres;

IF v\_istnieje = 0 THEN

INSERT INTO Wlasciciel (Wid, Imie, Nazwisko, Adres)

VALUES (v\_wid, v\_imie, v\_nazwisko, v\_adres);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Osoba o zadanych atrybutach juz istnieje.');

END IF;

END;

SELECT \* FROM Wlasciciel;

EXEC WstawWlasciciela('Tomasz', 'Kot', 'Warszawa, ul. Gdanska 15');

EXEC WstawWlasciciela('Jan', 'Kowalski', 'Warszawa, ul. Zielona 10');

EXEC WstawWlasciciela('Piotr', 'Wiśniewski', 'Gdańsk, ul. Morze 5');

EXEC WstawWlasciciela('Tomasz', 'Kot', 'Warszawa, ul. Gdanska 15');

Zad. 2. Napisz procedurę, która dla właściciela o imieniu, nazwisku i adresie podanych na wejściu wypisze rodzaje opłacanych mediów oraz liczbę rachunków opłaconych po terminie i średnią liczbę dni opóźnienia dla każdego medium osobno, pamiętając, że opłaty za media uiszcza się wielokrotnie na przestrzeni czasu. Jeśli właściciela o imieniu, nazwisku i adresie podanych na wejściu nie ma w bazie, wypisz odpowiedni komunikat. Jeśli właściciel opłaca wszystko w terminie, wypisz odpowiedni komunikat.

CREATE OR REPLACE PROCEDURE PokazOplacaneMedia (v\_imie VARCHAR, v\_nazwisko VARCHAR, v\_adres VARCHAR) AS

CURSOR v\_media IS SELECT M.Rodzaj FROM Media M

INNER JOIN Oplata O ON O.Mid = M.Mid

INNER JOIN Wlasciciel W ON O.Wid = W.Wid

WHERE W.Imie = v\_imie AND W.Nazwisko = v\_nazwisko AND W.Adres = v\_adres;

v\_ile\_po\_czasie INTEGER;

CURSOR v\_srednia\_opoznienia IS SELECT M.Rodzaj, AVG(O.DataWplaty - O.DataDo) Srednia FROM Media M

INNER JOIN Oplata O ON O.Mid = M.Mid

INNER JOIN Wlasciciel W ON O.Wid = W.Wid

WHERE W.Imie = v\_imie AND W.Nazwisko = v\_nazwisko AND W.Adres = v\_adres AND O.DataWplaty > O.DataDo

GROUP BY M.Rodzaj;

v\_istnieje INT := 0;

v\_po\_terminie INT := 0;

BEGIN

SELECT COUNT(\*) INTO v\_istnieje FROM Wlasciciel W

WHERE W.Imie = v\_imie AND W.Nazwisko = v\_nazwisko AND W.Adres = v\_adres;

IF v\_istnieje<> 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Rodzaje mediow: ');

FOR v\_medium IN v\_media LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_medium.Rodzaj);

END LOOP;

SELECT COUNT(\*) INTO v\_po\_terminie FROM Media M

INNER JOIN Oplata O ON O.Mid = M.Mid

INNER JOIN Wlasciciel W ON O.Wid = W.Wid

WHERE W.Imie = v\_imie AND W.Nazwisko = v\_nazwisko AND W.Adres = v\_adres AND O.DataWplaty > O.DataDo;

IF v\_po\_terminie = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Osoba o zadanych atrybutach placi wszystkie rachunki w terminie.');

ELSE

SELECT COUNT(\*) INTO v\_ile\_po\_czasie FROM Media M

INNER JOIN Oplata O ON O.Mid = M.Mid

INNER JOIN Wlasciciel W ON O.Wid = W.Wid

WHERE W.Imie = v\_imie AND W.Nazwisko = v\_nazwisko AND W.Adres = v\_adres AND O.DataWplaty > O.DataDo;

DBMS\_OUTPUT.PUT\_LINE('Liczba rachunkow oplaconych po terminie: ' || v\_ile\_po\_czasie);

FOR v\_opoznienie IN v\_srednia\_opoznienia LOOP

DBMS\_OUTPUT.PUT\_LINE('Medium: ' || v\_opoznienie.Rodzaj || ' Srednie opoznienie: ' || v\_opoznienie.Srednia);

END LOOP;

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Osoba o zadanych atrybutach nie istnieje.');

END IF;

END;

Zad. 3. Napisz wyzwalacz, który przy wstawianiu i aktualizacji danych w tabeli Oplata nie pozwoli wstawić daty do oraz daty wpłaty wcześniejszych niż data od.

CREATE OR REPLACE TRIGGER SprawdzDateOdDo

BEFORE INSERT OR UPDATE

ON Oplata

FOR EACH ROW

BEGIN

IF :NEW.DataDo < :NEW.DataOd THEN

RAISE\_APPLICATION\_ERROR(-20100, 'Data do musi być późniejsza niż data od.');

END IF;

IF :NEW.DataWplaty < :NEW.DataOd THEN

RAISE\_APPLICATION\_ERROR(-20101, 'Data wpłaty musi być późniejsza niż data od.');

END IF;

END;

Zad. 4. Napisz wyzwalacz, który przy aktualizacji pola Mid w tabeli Media zaktualizuje wszystkie wartości Mid w tabeli Oplata dla danego rodzaju mediów. Załóż, że na tabelach nie ma założonych więzów spójności referencyjnej.

CREATE OR REPLACE TRIGGER ZaktualizujMedia

AFTER UPDATE

ON Media

FOR EACH ROW

BEGIN

IF :OLD.Mid <> :NEW.Mid THEN

UPDATE Oplata

SET Mid = :NEW.Mid

WHERE Mid = :OLD.Mid;

END IF;

END;