```
SELECT * FROM DEPARTMENTS;


DECLARE
    numer_max departments.department_id%TYPE;
    nowy_numer departments.department_id%TYPE;
    nazwa departments.department_name%TYPE := 'Development';
BEGIN
    SELECT MAX(department_id) INTO numer_max FROM DEPARTMENTS;
    nowy_numer := numer_max + 10;
    INSERT INTO DEPARTMENTS(department_id, department_name)
        VALUES (nowy_numer, nazwa);
END;
/


SELECT * FROM DEPARTMENTS;


DELETE FROM DEPARTMENTS WHERE DEPARTMENT_ID = 280;


DECLARE
    numer_max departments.department_id%TYPE;
    nowy_numer departments.department_id%TYPE;
    nazwa departments.department_name%TYPE := 'Development';
BEGIN
    SELECT MAX(department_id) INTO numer_max FROM DEPARTMENTS;
    nowy_numer := numer_max + 10;
    INSERT INTO DEPARTMENTS(department_id, department_name)
        VALUES (nowy_numer, nazwa);
    UPDATE departments SET location_id = 3000
        WHERE department_id = nowy_numer;
END;
/

CREATE TABLE nowa_tabela (tekst VARCHAR2(10));

BEGIN
    FOR i IN 1..10 LOOP
        IF i NOT IN (4, 6) THEN
            INSERT INTO nowa_tabela (tekst)
            VALUES (TO_CHAR(i));
        END IF;
    END LOOP;
END;
/

SELECT * FROM nowa_tabela;

SELECT * FROM COUNTRIES;

SET SERVEROUTPUT ON;

DECLARE
    kraj countries%ROWTYPE;
BEGIN
    SELECT * INTO kraj FROM COUNTRIES
```

```
        WHERE country_id = 'CA';
    DBMS_OUTPUT.PUT_LINE('Nazwa kraju: ' || kraj.country_name);
    DBMS_OUTPUT.PUT_LINE('Region ID: ' || kraj.region_id);
END;
/

SELECT * FROM EMPLOYEES;

DECLARE
    CURSOR pracownicy_cursor IS
        SELECT last_name, salary FROM EMPLOYEES
            WHERE department_id = 50;
    v_nazwisko employees.last_name%TYPE;
    v_wynagrodz employees.salary%TYPE;
BEGIN
    OPEN pracownicy_cursor;
    LOOP
        FETCH pracownicy_cursor INTO v_nazwisko, v_wynagrodz;
        EXIT WHEN pracownicy_cursor%NOTFOUND;
        IF v_wynagrodz > 3100 THEN
            DBMS_OUTPUT.PUT_LINE(v_nazwisko || ' nie dawać podwyżki');
        ELSE
            DBMS_OUTPUT.PUT_LINE(v_nazwisko || ' dać podwyżkę');
        END IF;
    END LOOP;
    CLOSE pracownicy_cursor;
END;
```

```
/


DECLARE
   CURSOR pracownicy_cursor(p_min NUMBER, p_max NUMBER, p_fragment VARCHAR2) IS
      SELECT first_name, last_name, salary FROM EMPLOYEES
         WHERE salary BETWEEN p_min AND p_max AND LOWER(first_name) LIKE '%' || LOWER(p_fragment) || '%';
   v_imie employees.first_name%TYPE;
   v_nazwisko employees.last_name%TYPE;
   v_zarobki employees.salary%TYPE;
BEGIN
   DBMS_OUTPUT.PUT_LINE('--- Pracownicy z widełkami 1000–5000 i imieniem zawierającym "a" ---');
   OPEN pracownicy_cursor(1000, 5000, 'a');
   LOOP
      FETCH pracownicy_cursor INTO v_imie, v_nazwisko, v_zarobki;
      EXIT WHEN pracownicy_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE(v_imie || ' ' || v_nazwisko || ' - ' || v_zarobki);
   END LOOP;
   CLOSE pracownicy_cursor;

   DBMS_OUTPUT.PUT_LINE(CHR(10) || '--- Pracownicy z widełkami 5000–20000 i imieniem zawierającym "u" ---');
   OPEN pracownicy_cursor(5000, 20000, 'u');
   LOOP
      FETCH pracownicy_cursor INTO v_imie, v_nazwisko, v_zarobki;
      EXIT WHEN pracownicy_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE(v_imie || ' ' || v_nazwisko || ' - ' || v_zarobki);
   END LOOP;
   CLOSE pracownicy_cursor;
```

```
END;
/

SELECT * FROM JOBS;
SELECT * FROM EMPLOYEES;

CREATE OR REPLACE PROCEDURE dodaj_job (
    p_job_id jobs.job_id%TYPE,
    p_job_title jobs.job_title%TYPE
) AS
BEGIN
    INSERT INTO JOBS (job_id, job_title)
        VALUES (p_job_id, p_job_title);
    DBMS_OUTPUT.PUT_LINE('Dodano nowy JOB: ' || p_job_id || ' - ' || p_job_title);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Błąd: ' || SQLERRM);
END;
/

BEGIN
    dodaj_job('TKI_JOB', 'Somethinger');
END;
/

SELECT * FROM JOBS;
```

```
CREATE OR REPLACE PROCEDURE zmien_job_title (
    p_job_id jobs.job_id%TYPE,
    p_nowy_tytul JOBS.job_title%TYPE
) AS v_rows_updated NUMBER;
BEGIN
    UPDATE JOBS SET job_title = p_nowy_tytul
        WHERE job_id = p_job_id;
    v_rows_updated := SQL%ROWCOUNT;
    IF v_rows_updated = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nie zaktualizowano żadnego wiersza – brak JOB_ID');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Zmieniono tytuł stanowiska na: ' || p_nowy_tytul);
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Błąd: ' || SQLERRM);
END;
/

BEGIN
    zmien_job_title('TKI_JOB', 'Everythinger');
END;
/

CREATE OR REPLACE PROCEDURE usun_job (
    p_job_id jobs.job_id%TYPE
) AS
```

```
BEGIN
   DELETE FROM JOBS
   WHERE job_id = p_job_id;
   IF SQL%ROWCOUNT = 0 THEN
      RAISE_APPLICATION_ERROR(-20002, 'Nie usunięto żadnego joba – brak dopasowania.');
   ELSE
      DBMS_OUTPUT.PUT_LINE('Usunięto JOB: ' || p_job_id);
   END IF;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Błąd: ' || SQLERRM);
END;
/

BEGIN
   usun_job('TKI_JOB');
END;
/

SELECT * FROM EMPLOYEES;

CREATE OR REPLACE PROCEDURE pobierz_pracownika (
   p_emp_id employees.employee_id%TYPE,
   p_nazwisko OUT employees.last_name%TYPE,
   p_zarobki OUT employees.salary%TYPE
) AS
BEGIN
```

```
    SELECT last_name, salary INTO p_nazwisko, p_zarobki FROM EMPLOYEES
        WHERE employee_id = p_emp_id;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nie znaleziono pracownika o ID: ' || p_emp_id);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Błąd: ' || SQLERRM);
END;
/

DECLARE
    v_nazwisko employees.last_name%TYPE;
    v_zarobki  employees.salary%TYPE;
BEGIN
    pobierz_pracownika(100, v_nazwisko, v_zarobki);
    DBMS_OUTPUT.PUT_LINE('Nazwisko: ' || v_nazwisko || ', Zarobki: ' || v_zarobki);
END;
/

CREATE OR REPLACE PROCEDURE dodaj_pracownika (
    p_first_name employees.first_name%TYPE DEFAULT 'Nowy',
    p_last_name employees.last_name%TYPE DEFAULT 'Pracownik',
    p_salary employees.salary%TYPE DEFAULT 1000,
    p_job_id employees.job_id%TYPE DEFAULT 'IT_PROG',
    p_email employees.email%TYPE DEFAULT 'nowy@firma.com',
    p_hire_date employees.hire_date%TYPE DEFAULT SYSDATE,
    p_department_id employees.department_id%TYPE DEFAULT 50
) AS
BEGIN
    IF p_salary > 20000 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Nie można dodać pracownika z wynagrodzeniem powyżej 20000.');
    END IF;
    INSERT INTO EMPLOYEES (
        employee_id, first_name, last_name, salary,
        job_id, email, hire_date, department_id
    ) VALUES (
        EMPLOYEES_SEQ.NEXTVAL, p_first_name, p_last_name, p_salary,
        p_job_id, p_email, p_hire_date, p_department_id
    );
    DBMS_OUTPUT.PUT_LINE('Dodano pracownika: ' || p_first_name || ' ' || p_last_name);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Błąd: ' || SQLERRM);
END;
/

BEGIN
    dodaj_pracownika(p_salary => 15000);
END;
/

SELECT * FROM EMPLOYEES;
```