

Zadania SQL z rozwiązaniami

Relacyjne Bazy Danych · składnia Oracle SQL

Zestaw ćwiczeń do samodzielnego pisania zapytań — od podstaw SELECT po DML. Najpierw spróbuj napisać zapytanie samodzielnie, potem sprawdź rozwiązanie. Zapytania możesz przetestować w Oracle (np. Oracle Live SQL / XE), tworząc tabele wg schematu poniżej.

Schemat przykładowej bazy (Oracle)

Wszystkie zadania odnoszą się do poniższego schematu. Klucze główne podkreślono pojęciowo jako PK, klucze obce jako FK.

Tabela	Kolumny (PK = klucz główny, FK = klucz obcy)
DZIALY	id_dzialu PK, nazwa, lokalizacja, id_kierownika FK→PRACOWNICY
PRACOWNICY	id_prac PK, imie, nazwisko, stanowisko, pensja, prowizja (NULL), data_zatr, id_dzialu FK→DZIALY, id_szefa FK→PRACOWNICY
PROJEKTY	id_proj PK, nazwa, data_start, data_koniec (NULL), id_dzialu FK→DZIALY
PRZYDZIALY	(id_prac, id_proj) PK; id_prac FK→PRACOWNICY, id_proj FK→PROJEKTY, godziny, rola

Powiązania: DZIAŁ 1—N PRACOWNICY · PRACOWNIK 1—N PRACOWNICY (szef, rekurencyjny) · DZIAŁ 1—N PROJEKTY · PRACOWNIK M—N PROJEKTY (przez PRZYDZIALY).

A · SELECT — podstawy (WHERE, ORDER BY)

Zadanie 1.

Wypisz nazwisko i pensję pracowników z pensją powyżej 8000, posortowane malejąco wg pensji.

Rozwiązanie

```
SELECT nazwisko, pensja
FROM pracownicy
WHERE pensja > 8000
ORDER BY pensja DESC;
```

Zadanie 2.

Wypisz różne (bez powtórzeń) stanowiska występujące w firmie.

Rozwiązanie

```
SELECT DISTINCT stanowisko
FROM pracownicy;
```

Zadanie 3.

Wypisz pracowników z działu 30, których nazwisko zaczyna się na literę „Z”.

Rozwiązanie

```
SELECT imie, nazwisko
FROM pracownicy
WHERE id_działu = 30 AND nazwisko LIKE 'Z%';
```

Zadanie 4.

Wypisz nazwisko i roczną pensję ($12 \times$ pensja) jako kolumnę o aliasie ROCZNIE, posortowane wg tej kolumny rosnąco.

Rozwiązanie

```
SELECT nazwisko, pensja * 12 AS rocznie
FROM pracownicy
ORDER BY rocznie;
```

Aliasu ROCZNIE można użyć w ORDER BY (wykonuje się po SELECT), ale nie w WHERE.

B · Operatory, funkcje i NULL

Zadanie 5.

Dla każdego pracownika wypisz nazwisko i całkowite wynagrodzenie = pensja + prowizja, traktując brak prowizji (NULL) jako 0.

Rozwiązanie

```
SELECT nazwisko, pensja + NVL(prowizja, 0) AS calosc
FROM pracownicy
ORDER BY calosc DESC;
```

Bez NVL wynik byłby NULL dla pracowników bez prowizji (NULL „zaraża” arytmetykę).

Zadanie 6.

Wypisz pracowników, którzy NIE mają przyznanej prowizji.

Rozwiązanie

```
SELECT nazwisko
FROM pracownicy
WHERE prowizja IS NULL;
```

Uwaga: WHERE prowizja = NULL nie zadziała — porównanie z NULL daje UNKNOWN.

Zadanie 7.

Wypisz etykietę w formacie „NAZWISKO I.” (nazwisko wielkimi literami + inicjał imienia) dla pracowników działu 30.

Rozwiązanie

```
SELECT UPPER(nazwisko) || ' ' || SUBSTR(imie,1,1) || '.' AS etykieta
FROM pracownicy
WHERE id_działu = 30;
```

Zadanie 8.

Wypisz nazwisko, pensję oraz kategorię: 'wysoka' (>10000), 'srednia' (>7000), w przeciwnym razie 'niska'.

Rozwiązanie

```
SELECT nazwisko, pensja,
CASE WHEN pensja > 10000 THEN 'wysoka'
WHEN pensja > 7000 THEN 'srednia'
ELSE 'niska' END AS kategoria
FROM pracownicy;
```

Zadanie 9.

Wypisz nazwisko i staż pracy w pełnych latach (na dzień dzisiejszy) dla każdego pracownika.

Rozwiązanie

```
SELECT nazwisko,
TRUNC(MONTHS_BETWEEN(SYSDATE, data_zatr) / 12) AS staz_lat
FROM pracownicy
ORDER BY staz_lat DESC;
```

C · Złączenia (JOIN)

Zadanie 10.

Wypisz nazwisko pracownika i nazwę jego działu — tylko dla działów w Warszawie.

Rozwiązanie

```
SELECT p.nazwisko, d.nazwa
FROM pracownicy p
JOIN dzialy d ON p.id_działu = d.id_działu
WHERE d.lokalizacja = 'Warszawa';
```

Zadanie 11.

Wypisz nazwę każdego działu oraz liczbę zatrudnionych w nim pracowników — także działu bez pracowników (liczba 0).

Rozwiązanie

```
SELECT d.nazwa, COUNT(p.id_prac) AS liczba
FROM dzialy d
LEFT JOIN pracownicy p ON d.id_dzialu = p.id_dzialu
GROUP BY d.nazwa
ORDER BY liczba DESC;
```

LEFT JOIN zachowuje działy bez pracowników; COUNT(p.id_prac) nie liczy NULL, więc da 0.

Zadanie 12.

Wypisz nazwisko każdego pracownika i nazwisko jego szefa (pracownicy bez szefa też mają się pojawić).

Rozwiązanie

```
SELECT p.nazwisko AS pracownik, s.nazwisko AS szef
FROM pracownicy p
LEFT JOIN pracownicy s ON p.id_szefa = s.id_prac
ORDER BY p.id_prac;
```

Samozłączenie tabeli PRACOWNICY (związek rekurencyjny szef-podwładny).

D · Grupowanie i operatory algebraiczne

Zadanie 13.

Dla każdego działu podaj średnią pensję; pokaż tylko działy, w których średnia przekracza 7000.

Rozwiązanie

```
SELECT id_dzialu, AVG(pensja) AS srednia
FROM pracownicy
GROUP BY id_dzialu
HAVING AVG(pensja) > 7000
ORDER BY srednia DESC;
```

Warunek na agregat → HAVING (nie WHERE).

Zadanie 14.

Dla każdego projektu podaj łączną liczbę przepracowanych godzin (z tabeli PRZYDZIALY).

Rozwiązanie

```
SELECT pr.nazwa, SUM(z.godziny) AS razem_godzin
FROM projekty pr
JOIN przydzialy z ON pr.id_proj = z.id_proj
GROUP BY pr.nazwa
ORDER BY razem_godzin DESC;
```

Zadanie 15.

Wypisz nazwiska pracowników, którzy zarabiają powyżej 10000 LUB mają przyznaną prowizję (bez duplikatów).

Rozwiązanie

```
SELECT nazwisko FROM pracownicy WHERE pensja > 10000
UNION
SELECT nazwisko FROM pracownicy WHERE prowizja IS NOT NULL
ORDER BY nazwisko;
```

UNION usuwa duplikaty; UNION ALL zachowałby powtórzenia.

E · Podzapytania

Zadanie 16.

Wypisz pracowników, których pensja jest wyższa niż średnia pensja w całej firmie.

Rozwiązanie

```
SELECT nazwisko, pensja
FROM pracownicy
WHERE pensja > (SELECT AVG(pensja) FROM pracownicy)
ORDER BY pensja DESC;
```

Podzapytanie zwykłe zwracające jedną wartość.

Zadanie 17.

Wypisz pracowników zatrudnionych w działach, które mieszczą się w Warszawie (użyj podzapytania z IN).

Rozwiązanie

```
SELECT nazwisko
FROM pracownicy
WHERE id_działu IN
(SELECT id_działu FROM działy WHERE lokalizacja = 'Warszawa');
```

Zadanie 18.

Wypisz pracowników zarabiających więcej niż KAŻDY pracownik działu 30 (operator ALL).

Rozwiązanie

```
SELECT nazwisko, pensja
FROM pracownicy
WHERE pensja > ALL (SELECT pensja FROM pracownicy WHERE id_działu = 30);
```

> ALL = więcej niż maksimum z listy.

Zadanie 19.

Wypisz pracowników, którzy NIE są przypisani do żadnego projektu (NOT EXISTS).

Rozwiązanie

```
SELECT nazwisko
FROM pracownicy p
WHERE NOT EXISTS
(SELECT 1 FROM przydziały z WHERE z.id_prac = p.id_prac)
ORDER BY nazwisko;
```

Podzapytanie skorelowane — odwołuje się do p.id_prac z zapytania zewnętrznego.

Zadanie 20.

Dla każdego pracownika wypisz jego pensję oraz średnią pensję w jego dziale (podzapytanie skorelowane).

Rozwiązanie

```
SELECT p.nazwisko, p.pensja,
(SELECT ROUND(AVG(p2.pensja))
FROM pracownicy p2
WHERE p2.id_dzialu = p.id_dzialu) AS srednia_dzialu
FROM pracownicy p;
```

F · DDL i DML

Zadanie 21.

Napisz polecenie tworzące tabelę DZIALY z kluczem głównym, nazwą (wymagana) i kontrolą budżetu (> 0).

Rozwiązanie

```
CREATE TABLE dzialy (
id_dzialu NUMBER PRIMARY KEY,
nazwa VARCHAR2(50) NOT NULL,
lokalizacja VARCHAR2(50),
budzet NUMBER CHECK (budzet > 0)
);
```

Zadanie 22.

Dodaj do tabeli PRACOWNICY klucz obcy do działu (kolumna id_dzialu odwołująca się do DZIALY).

Rozwiązanie

```
ALTER TABLE pracownicy
ADD CONSTRAINT fk_prac_dzial
FOREIGN KEY (id_dzialu) REFERENCES dzialy(id_dzialu);
```

Zadanie 23.

Podnieś o 10% pensje wszystkim pracownikom działu IT (id_dzialu = 20) i zatwierdź zmiany.

Rozwiązanie

```
UPDATE pracownicy
SET pensja = pensja * 1.1
WHERE id_dzialu = 20;
COMMIT;
```

Zadanie 24.

Wstaw nowy dział o id 50, nazwie 'R&D;' w Warszawie. Wyjaśnij, w jakiej kolejności wstawiać dział i pracownika tego działu.

Rozwiązanie

```
INSERT INTO dzialy (id_dzialu, nazwa, lokalizacja)
VALUES (50, 'R&D', 'Warszawa');
COMMIT;
```

Najpierw dział (tabela nadrzędna), dopiero potem pracownika z id_dzialu = 50 — inaczej więz referencyjny zostanie naruszony.