



KRYSTIAN BROŻEK
O PROGRAMOWANIU LUDZKIM GŁOSEM

Rozmowa kwalifikacyjna bazy danych



O PROGRAMOWANIU
LUDZKIM GŁOSEM

Więcej znajdziesz na moim blogu krystianbrozek.pl



Zobacz moje szkolenia na geek-on.pl



Usłysz mnie na IT Kwadrans.pl



Spis treści

Rozmowa kwalifikacyjna	1
bazy danych.....	1
Na co trzeba zwrócić uwagę przygotowując się do rozmowy	3
GROUP BY i JOINy.....	3
Myśl.....	3
Zadawaj pytania	3
Najczęściej używane funkcje w zadaniach na rozmowie kwalifikacyjnej	3
Pytania teoretyczne	4
Pytania teoretyczne – odpowiedzi.....	5
Zadania na rozmowę kwalifikacyjną – zestaw nr 1.....	6
Zestaw nr 1 – odpowiedzi	7
Ile paragonów zostało wystawionych w tym roku?	7
Który pracownik wystawił najwięcej paragonów?.....	7
Policz wartość każdego paragonu zawierającego przynajmniej dwie pozycje.	7
Zadania na rozmowę kwalifikacyjną – zestaw nr 2	8
Zestaw nr 2 – odpowiedzi	9
Znajdź pracowników urodzonych po roku 1987	9
Podaj imię i nazwisko pracownika oraz imię i nazwisko szefa projektu w którym brał udział – tylko unikalne wystąpienia, pomiń samego siebie	9
Znajdź pracowników, którzy nigdy nie byli przewodniczącym grupy, ani nie zarządzali projektem	9
Zadania na rozmowę kwalifikacyjną – zestaw 3	10
Zestaw nr 3 – odpowiedzi	11
Znajdź powiązane filmy do każdego filmu	11
Ile filmów miało premierę w Polsce w innym roku niż rok produkcji?	11
Pokaż filmy i kraj produkcji, które miały premierę przynajmniej w dwóch państwach	11



Na co trzeba zwrócić uwagę przygotowując się do rozmowy

GROUP BY i JOINy

Najczęściej pojawiającymi się pytaniami praktycznymi są te, w których wykorzystuje się GROUP BY i JOINy. Są one najtrudniejsze w zrozumieniu na początku nauki, więc łatwo sprawdzić, czy ktoś zna SQLa poprzez pisanie zapytań z ich wykorzystaniem.

Myśl

To druga prawie zawsze sprawdzana umiejętność. Nie zawsze trzeba dokończyć pisać zapytanie – rekruter może przerwać w trakcie, bo dopisanie go do końca to kwestia chwili, bo wiadomo, że kandydat wie jak je rozwiązać. Zatem rekruterzy sprawdzają czy potrafisz myśleć, po części jest to związane z faktem, że podejście do przetwarzania danych w bazach jest raczej całościowe (czyli jedną instrukcją uzyskujemy ostateczny wynik), a w programowaniu piszemy kod krok po kroku co ma wykonywać.

Zadawaj pytania

Nigdy nie zgaduj. Nie jesteś czegoś pewien, to zapytaj. Każdy ze stresu może zapomnieć nazwy funkcji, więc podpytaj jak jej nie będziesz pamiętać, bo sprawdza się czy umiesz używać SQLa, a nie czy pamiętasz całą dokumentację i wszystkie funkcje w niej zawarte.

Nigdy też nie zgaduj. Mogą pojawić się zadania podchwytliwe, gdzie nie będzie wiadomo od razu jakie dane wyciągnąć. Zgadując pokazujesz, że w pracy też będziesz zgadywał, czyli trzeba będzie po Tobie często poprawiać – ja bym nie chciał z taką osobą pracować 😊

Najczęściej używane funkcje w zadaniach na rozmowie kwalifikacyjnej

SUM, AVG, COUNT – skoro grupujemy, to trzeba będzie wykorzystać też funkcje agregujące

SUBSTRING, SUBSTR, SUB, MID – w zależności od silnika funkcja ma inną nazwę. Jest to funkcja wyciągająca fragment tekstu ze środka przydatna np. przy znalezieniu daty urodzenia z PESELu.

Funkcje daty i czasu – a w szczególności taka, którą wyciągniesz rok, miesiąc i dzień.



Pytania teoretyczne

1. Różnica między WHERE i HAVING
2. Jak wybrać 3 wiersze
3. Różnica między UNION i UNION ALL
4. Co jest szybsze, LIKE czy =?
5. Jakie są rodzaje złączeń?
6. * Co to jest transakcja?
7. * Co to jest trigger?
8. * Czym się różni DELETE od TRUNCATE?



Pytania teoretyczne – odpowiedzi

1. Różnica między WHERE i HAVING

WHERE służy do filtrowania wierszy i można w nim używać funkcji zwykłych (skalarnych). HAVING służy do filtrowania grup, czyli nie użyjemy go bez GROUP BY, można w nim używać funkcji agregujących i zwykłych (skalarnych), ale te drugie warto przetrzucić do WHERE, ze względu na wydajność (prawie zawsze można to zrobić).

2. Jak wybrać 3 wiersze

W przypadku MSSQL na początku piszemy SELECT TOP 3..., w innych silnikach (np. MySQL, SQLite) na samym końcu zapytania dopisujemy LIMIT 3. W przypadku bazy Oracle trzeba zrobić podzapytanie numerujące wiersze, a głównym zapytaniem pofiltrować w WHERE NumerWiersza <= 3

3. Różnica między UNION i UNION ALL

UNION ALL zwraca nam wszystko. UNION wyciąga tylko unikalne wartości w ramach całego zapytania (nawet jeśli w jednym z zapytań są powtórzone wartości, to też duplikat zostanie pominięty).

4. Co jest szybsze, LIKE czy =?

Porównanie jest znacznie szybsze. W przypadku LIKE przyrównujemy do schematu, więc algorytm jest bardziej złożony.

5. Jakie są rodzaje złączeń?

Najbardziej podstawowe rodzaje to:

Złączenie wewnętrzne, INNER JOIN – łączymy tylko te wiersze, które mają parę.

Złączenia zewnętrzne lewo i prawo stronne, LEFT OUTER JOIN, RIGHT OUTER JOIN (słowo outer jest zwykle pomijane, nie trzeba go podawać) – lewa lub prawa tabela jest cała, a z drugiej tabeli dołączamy odpowiedniki. Jeśli nie znajdziemy pasującego wiersza, wtedy pojawią się NULLe.

6. * Co to jest transakcja? (pytanie najczęściej pojawiające się dla programistów i pro testerów, choć juniorzy też powinni wiedzieć czym jest transakcja)

Transakcja to tak jakby paczka zapytań wysyłanych w całości. Domyślnie wykonuje się wszystko albo nic, czyli wysyłając w transakcji dodanie 3 wierszy i pojawi się błąd na dodawaniu trzeciego (np. tekst jest dłuższy niż kolumna ma wielkość), wtedy wycofywane są dwa dodane wiersze.

7. * Co to jest trigger? (pytanie dla programistów i testerów)

Trigger to procedura składowana uruchamiana przed lub zamiast INSERT, UPDATE lub DELETE. Możemy wykorzystać np. do odkładania logów, czyli przed uruchomieniem UPDATE zapisujemy informację kto, o której godzinie i jaką zmianę wprowadził.

8. * Czym się różni DELETE od TRUNCATE? (pytanie dla programistów i testerów)

TRUNCATE jest wykonywany poza transakcją, czyści całą tabelę i jest szybszy. DELETE wykonywany jest w transakcji i usuwa wybrane rekordy w WHERE.



Zadania na rozmowę kwalifikacyjną – zestaw nr 1

Produkty
EAN
Nazwa
Cena
Opis

ZawartoscParagonu
EAN
Paragon
Ilosc

Paragon
NrParagonu
DataZakupu
NumerSprzedawcy

Sprzedawcy
Id
Imie
Nazwisko

1. Ile paragonów zostało wystawionych w tym roku?
2. Który pracownik wystawił najwięcej paragonów?
3. Policz wartość każdego paragonu zawierającego przynajmniej dwie pozycje.



Zestaw nr 1 – odpowiedzi

Ile paragonów zostało wystawionych w tym roku?

Musimy policzyć ilość wierszy w tabeli z paragonami (każdy wiersz odpowiada jednemu wystawionemu paragonowi).

```
SELECT COUNT(*)
FROM Paragon
WHERE DataZakupu >= '2019-01-01'
```

Możemy skorzystać też z funkcji daty i czasu. MSSQL udostępnia funkcję YEAR, która wyciąga nam sam rok:

```
SELECT COUNT(*)
FROM Paragon
WHERE YEAR(DataZakupu) = 2019
```

Istnieje jeszcze rozwiązanie bez wskazywania roku. Skoro pytanie jest o aktualny rok, możemy wyciągnąć go z daty, która jest w tym momencie:

```
SELECT COUNT(*)
FROM Paragon
WHERE YEAR(DataZakupu) = YEAR(GETDATE())
```

Który pracownik wystawił najwięcej paragonów?

Tym razem trzeba połączyć tabele Paragon i Sprzedawcy ze sobą, aby znaleźć imię i nazwisko pracownika. Następnie grupujemy po imieniu i nazwisku, sortujemy malejąco po ilości elementów w danej grupie (aby pracownik z największą ilością był w pierwszym wierszu), a na koniec ograniczamy się tylko do jednego wiersza. W zależności od silnika bazy danych ostatnia operacja będzie się różniła. Poniżej przykład z TOP 1 działający w MSSQL:

```
SELECT TOP 1 s.Imie, s.Nazwisko
FROM Paragon p
     INNER JOIN Sprzedawcy s
           ON p.NumerSprzedawcy = s.Id
GROUP BY s.Imie, s.Nazwisko
ORDER BY COUNT(*) DESC
```

Policz wartość każdego paragonu zawierającego przynajmniej dwie pozycje.

Tutaj wystarczy połączyć ze sobą dwie tabele – Produkty i ZawartoscParagonu. Można dodać tabelę Paragon, ale nie wypłynie ona w żaden sposób na nasze rozwiązanie, więc jest zbędna - numer paragonu mamy w tabeli ZawartoscParagonu i tyle informacji będzie odpowiednie. Na koniec bierzemy tylko te grupy, które mają przynajmniej 2 wiersze.

```
SELECT zp.Paragon AS NumerParagonu, SUM(zp.Ilosc * p.Cena) AS Kwota
FROM Produkty p
     INNER JOIN ZawartoscParagonu zp
           ON p.EAN = zp.EAN
GROUP BY zp.Paragon
HAVING COUNT(*) >= 2
```



Zadania na rozmowę kwalifikacyjną – zestaw nr 2

Pracownicy
PESEL
Imie
Nazwisko
DataUrodzenia

Projekty
NumerProjektu
DataRozpoczenia
SzefProjektu

PracownicyProjekty
NrProj
Pracownik
Grupa

Grupy
Id
Nazwa
Przewodniczacy

1. Znajdź pracowników urodzonych po roku 1987
2. Podaj imię i nazwisko pracownika oraz imię i nazwisko szefa projektu w którym brał udział – tylko unikalne wystąpienia, pomiń samego siebie
3. Znajdź pracowników, którzy nigdy nie byli przewodniczącym grupy, ani nie zarządzali projektem



Zestaw nr 2 – odpowiedzi

Znajdź pracowników urodzonych po roku 1987

Zadanie dość proste sprawdzające, czy w ogóle potrafimy pisać zapytania.

```
SELECT *
FROM Pracownicy
WHERE DataUrodzenia >= '1987-01-01'
```

Podaj imię i nazwisko pracownika oraz imię i nazwisko szefa projektu w którym brał udział – tylko unikalne wystąpienia, pominię samego siebie

Zadanie niezbyt skomplikowane – dużo pisania i DISTINCT, o którym zdarza się zapomnieć.

W pierwszej kolejności łączymy tabele: Pracownicy z PracownicyProjekty, później dołączamy Projekty i już wiemy kto w jakim projekcie pracował. Na koniec dołączamy drugi raz tabelę Pracownicy. Konieczne jest nadawanie aliasów, ponieważ korzystamy dwukrotnie z tej samej tabeli (pracownicy).

Dodajmy warunek, który pominie nam pracowników, którzy brali udział w projekcie, którego byli szefem – najprościej sprawdzić po polu jednoznacznie określającym nam pracownika, czyli po numerze PESEL.

Na koniec DISTINCT, który pominie nam duplikaty. Zamiast niego możemy użyć też grupowania danych po wszystkich czterech kolumnach wyciąganych w SELECT – zapytanie wygląda mniej ładnie, ale działa bardzo dobrze 😊

```
SELECT DISTINCT p1.Imie AS ImiePracownika,
               p1.Nazwisko AS NazwiskoPracownika,
               p2.Imie AS ImieSzefa,
               p2.Nazwisko AS NazwiskoSzefa
FROM Pracownicy p1
   INNER JOIN PracownicyProjekty pp
           ON p1.PESEL = pp.Pracownik
   INNER JOIN Projekty p
           ON pp.NrProj = p.NumerProjektu
   INNER JOIN Pracownicy p2
           ON p2.SzefProjektu = PESEL
WHERE p1.PESEL <> p2.PESEL
```

Znajdź pracowników, którzy nigdy nie byli przewodniczącym grupy, ani nie zarządzali projektem

Trudność zadania polega na tym, że musimy użyć LEFT JOIN lub RIGHT JOIN w taki sposób, aby wziąć całą tabelę Pracownicy. Wtedy wybieramy tych pracowników, którzy nie połączyli się do żadnej z tabel. Zwróć uwagę na warunek: to jest często popełniany błąd u osób początkujących – o brak wartości pytamy się **IS NULL**, a nie **= NULL**.

```
SELECT p.Imie, p.Nazwisko
FROM Pracownicy p
   LEFT JOIN Projekty proj
           ON proj.SzefProjektu = p.PESEL
   LEFT JOIN Grupy g
           ON g.Przewodniczacy = p.PESEL
WHERE proj.SzefProjektu IS NULL
   AND g.Przewodniczacy IS NULL
```



Zadania na rozmowę kwalifikacyjną – zestaw 3

Filmy

NrFilmu	Tytuł	RokProdukcji	KrajProd
1	Shrek	2001	1
2	Shrek 2	2004	1
3	Pianista	2002	2
4	Złap mnie jeśli potrafisz	2002	3

Filmy powiązane

Film	Powiazany
1	2
2	1

Kraje

KodKraju	Nazwa
0	świat
1	USA
2	Polska
3	Kanada

PremieraFilmu

Film	Kraj	DataPremiery
1	2	2001-07-13
1	0	2001-04-22
2	2	2004-07-02
3	0	2002-05-24
4	2	2003-02-28
4	0	2002-12-16

1. Znajdź powiązane filmy do każdego filmu
2. Ile filmów miało premierę w Polsce w innym roku niż rok produkcji?
3. Pokaż filmy i kraj produkcji, które miały premierę przynajmniej w dwóch państwach



Zestaw nr 3 – odpowiedzi

Znajdź powiązane filmy do każdego filmu

Cała trudność zadania polega na tym, że należy nadać różne aliasy na tabelę Filmy, ponieważ używamy jej dwukrotnie.

```
SELECT f1.Tytuł AS TytułFilmu,  
       f2.Tytuł AS FilmPowiazany  
FROM Filmy f1  
     INNER JOIN FilmyPowiazane fp  
           ON f1.NrFilmu = fp.Film  
     INNER JOIN Filmy f2  
           ON f2.NrFilmu = fp.Powiazany
```

Ile filmów miało premierę w Polsce w innym roku niż rok produkcji?

Trzeba połączyć odpowiednie tabele ze sobą, czyli Filmy, PremieraFilmu i Kraj. Następnie wybieramy Polskę. W MSSQL do dat mamy funkcje Year, Month i Day, ale jeśli korzystasz z innego silnika to koniecznie przypomnij sobie funkcję wyciągającą jakąś część z daty, takie pytania dość często pojawiają się na rozmowach.

```
SELECT *  
FROM Filmy f  
     INNER JOIN PremieraFilmu pf  
           ON f.NrFilmu = pf.Film  
     INNER JOIN Kraje k  
           ON k.KodKraju = pf.Kraj  
WHERE k.Nazwa = 'Polska'  
     AND YEAR(pf.DataPremiery) <> f.RokProdukcji
```

Pokaż filmy i kraj produkcji, które miały premierę przynajmniej w dwóch państwach

Zadanie sprawdza 3 rzeczy. Po pierwsze nie ma takiego przykładu w danych, co może wydać się mylące. Świat nie jest krajem, więc też powinniśmy się go pozbyć. Trzecia rzecz to użycie HAVING, którego często się zapomina, z racji jego niezbyt częstego występowania.

```
SELECT *  
FROM Filmy f  
     INNER JOIN PremieraFilmu pf  
           ON f.NrFilmu = pf.Film  
     INNER JOIN Kraje k  
           ON k.KodKraju = pf.Kraj  
WHERE k.Nazwa <> 'Świat'  
GROUP BY f.Tytuł  
HAVING COUNT(*) >= 2
```