
WYNIKI I TESTY KWEREND *SQL*

31 maja 2020

W bieżącym dokumencie, dla każdego zapytania umieszczono dwie tabele, odpowiednio wynikową i porównawczą zawierającą właściwy fragment pełnej bazy danych, agregowany w danej kwerendzie. Ręczne sprawdzenie zgodności danych w każdej parze tabel pozwoliło stwierdzić poprawność działania kolejnych instrukcji.

Zapytanie 1.

Kod zapytania i tabela wynikowa

```
SELECT ID, FirstName AS Imie, LastName AS Nazwisko
FROM Readers
WHERE ID IN
(
    SELECT DISTINCT ReaderID
    FROM Borrows b
    JOIN BookCopies bc ON b.BookCopyID = bc.ID
    GROUP BY b.ReaderID, bc.BookID, b.BorrowDate
    HAVING COUNT(*) > 1
)
```

Results		Messages	
	ID	Imie	Nazwisko
1	1	Harry	Adams
2	5	Leo	Cooper
3	19	Demi	Moore

Rysunek 1: Treść zapytania 1. oraz tabela wynikowa.

Tabela porównawcza

```
SELECT r.ID, FirstName, LastName,
       BookID, BookCopyID, BorrowDate
FROM Readers r
JOIN Borrows br ON br.ReaderID = r.ID
JOIN BookCopies bc ON bc.ID = br.BookCopyID
ORDER BY r.ID, BookID, BorrowDate
```

	ID	FirstName	LastName	BookID	BookCopyID	BorrowDate
1	1	Harry	Adams	1	1	2020-01-01
2	1	Harry	Adams	1	2	2020-01-01
3	2	Olivier	Allen	1	2	2020-02-28
4	3	Noah	Baker	1	5	2020-03-17
5	3	Noah	Baker	3	8	2020-03-17
6	3	Noah	Baker	4	11	2020-03-17
7	4	Cole	Baker	48	71	2020-04-11
8	5	Leo	Cooper	4	12	2020-04-04
9	5	Leo	Cooper	4	13	2020-04-04
10	5	Leo	Cooper	5	14	2020-04-04
11	5	Leo	Cooper	6	15	2020-04-04
12	7	Charlie	Carter	15	28	2020-04-09
13	9	Cole	Collins	14	27	2020-04-09
14	13	Ava	Fisher	16	29	2020-05-11
15	14	Lily	Ford	20	33	2020-05-13
16	15	Emily	Hamilton	20	34	2020-04-13
17	15	Emily	Hamilton	29	43	2020-04-16
18	18	Brooke	Lawrence	30	49	2019-01-01
19	18	Brooke	Lawrence	31	50	2018-03-01
20	18	Brooke	Lawrence	31	50	2018-04-01
21	18	Brooke	Lawrence	31	50	2019-01-01
22	19	Demi	Moore	1	4	2018-04-01
23	19	Demi	Moore	1	1	2019-02-17
24	19	Demi	Moore	1	2	2019-02-17
25	19	Demi	Moore	8	17	2018-03-01
26	19	Demi	Moore	30	49	2019-01-17
27	20	Ana	Holmes	5	14	2018-03-01
28	20	Ana	Holmes	5	14	2018-04-12
29	20	Ana	Holmes	9	20	2018-03-01
30	20	Ana	Holmes	10	21	2018-03-01
31	20	Ana	Holmes	11	22	2018-03-01
32	20	Ana	Holmes	37	57	2020-05-13
33	20	Ana	Holmes	37	58	2020-05-27

Rysunek 2: Wiersze zaznaczone kolorem przechowują jedyne obserwacje spełniające warunki narzucone w zapytaniu i są spójne z danymi znajdującymi się w tabeli wynikowej, zatem polecenie 1. działa poprawnie

Zapytanie 2.

Kod zapytania i tabela wynikowa

Wersja 1.

```
SELECT bc.BookID AS ID, b.Title AS Tytul, COUNT(*) AS IleRazy,  
       MIN(BorrowDate) AS PierwszeWypozyczenie  
FROM BookCopies bc  
JOIN Borrows br ON bc.ID = br.BookCopyID  
JOIN Books b ON bc.BookID = b.ID  
GROUP BY bc.BookID, b.Title
```

Wersja 2.

```
SELECT b.ID, b.Title AS Tytul, IleRazy, PierwszeWypozyczenie  
FROM Books b  
RIGHT JOIN  
(  
    SELECT bc.BookID, COUNT(*) AS IleRazy,  
           MIN(BorrowDate) AS PierwszeWypozyczenie  
    FROM BookCopies bc  
    JOIN Borrows br ON bc.ID = br.BookCopyID  
    GROUP BY bc.BookID) Tmp  
ON b.ID = Tmp.BookID
```

	ID	Tytul	IleRazy	PierwszeWypozyczenie
1	1	The Adventures of Sherlock Holmes	7	2018-04-01
2	3	The Devil in the White City	1	2020-03-17
3	4	Gone Girl	3	2020-03-17
4	5	In the Woods	3	2018-03-01
5	6	Still Life	1	2020-04-04
6	8	And Then There Were None	1	2018-03-01
7	9	A Simple Favor	1	2018-03-01
8	10	Before I Go to Sleep	1	2018-03-01
9	11	Bird Box	1	2018-03-01
10	14	It	1	2020-04-09
11	15	The Outsider	1	2020-04-09
12	16	Misery	1	2020-05-11
13	20	Elevation	2	2020-04-13
14	29	Five Feet Apart	1	2020-04-16
15	30	Harry Potter and the Philosophers Stone	2	2019-01-01
16	31	Harry Potter and the Chamber of Secrets	3	2018-03-01
17	37	A Game of Thrones	2	2020-05-13
18	48	Catching Fire	1	2020-04-11

Rysunek 3: Alternatywne treści zapytania 2. oraz odpowiadająca im tabela wynikowa

Tabela porównawcza

```
SELECT bc.BookID, b.Title, br.BorrowDate
FROM BookCopies bc
JOIN Borrows br ON bc.ID = br.BookCopyID
JOIN Books b ON bc.BookID = b.ID
ORDER BY Title, br.BorrowDate
```

Results		Messages	
	BookID	Title	BorrowDate
1	37	A Game of Thrones	2020-05-13
2	37	A Game of Thrones	2020-05-27
3	9	A Simple Favor	2018-03-01
4	8	And Then There Were None	2018-03-01
5	10	Before I Go to Sleep	2018-03-01
6	11	Bird Box	2018-03-01
7	48	Catching Fire	2020-04-11
8	20	Elevation	2020-04-13
9	20	Elevation	2020-05-13
10	29	Five Feet Apart	2020-04-16
11	4	Gone Girl	2020-03-17
12	4	Gone Girl	2020-04-04
13	4	Gone Girl	2020-04-04
14	31	Harry Potter and the Chamber of Secrets	2018-03-01
15	31	Harry Potter and the Chamber of Secrets	2018-04-01
16	31	Harry Potter and the Chamber of Secrets	2019-01-01
17	30	Harry Potter and the Philosophers Stone	2019-01-01
18	30	Harry Potter and the Philosophers Stone	2019-01-17
19	5	In the Woods	2018-03-01
20	5	In the Woods	2018-04-12
21	5	In the Woods	2020-04-04
22	14	It	2020-04-09
23	16	Misery	2020-05-11
24	6	Still Life	2020-04-04
25	1	The Adventures of Sherlock Holmes	2018-04-01
26	1	The Adventures of Sherlock Holmes	2019-02-17
27	1	The Adventures of Sherlock Holmes	2019-02-17
28	1	The Adventures of Sherlock Holmes	2020-01-01
29	1	The Adventures of Sherlock Holmes	2020-01-01
30	1	The Adventures of Sherlock Holmes	2020-02-28
31	1	The Adventures of Sherlock Holmes	2020-03-17
32	3	The Devil in the White City	2020-03-17
33	15	The Outsider	2020-04-09

Rysunek 4: Poziome linie służą zgrupowaniu książek względem ID. Ręczne porównanie tabeli testowej i wynikowej, poprzez zliczenie ilości wystąpień poszczególnych książek w tabeli testowej i odczytaniu najwcześniejszej daty wypożyczenia, pozwala na stwierdzenie, że dane w obydwu tabelach są zgodne. I tym samym polecenie 2. działa poprawnie.

Zapytanie 3.

Kod zapytania i tabela wynikowa

```
SELECT b.ID, b.Title as Tytul
FROM Books b
WHERE EXISTS (
    SELECT bc.BookID
    FROM BookCopies bc
    GROUP BY bc.BookID
    HAVING SUM(CAST(Available AS INT)) = 0
    AND bc.BookID = b.ID
)
```

Results			Messages	
	ID	Tytul		
1	31	Harry Potter and the Chamber of Secrets		
2	36	Harry Potter and the Deathly Hallows		
3	38	A Clash of Kings		
4	44	Alices Adventures in Wonderland		
5	49	Mockingjay		

Rysunek 5: Treść zapytania 3. oraz tabela wynikowa

Tabela porównawcza

```
SELECT ID, BookID, Available
FROM BookCopies
ORDER BY ID, Available
```

Results				Results			
	ID	BookID	Available		ID	BookID	Available
1	1	1	1	37	37	23	1
2	2	1	1	38	38	24	1
3	3	1	1	39	39	25	1
4	4	1	1	40	40	26	1
5	5	1	1	41	41	27	1
6	6	2	1	42	42	28	1
7	7	3	1	43	43	29	1
8	8	3	1	44	44	30	1
9	9	3	1	45	45	30	1
10	10	3	1	46	46	30	1
11	11	4	1	47	47	30	1
12	12	4	1	48	48	30	1
13	13	4	1	49	49	30	1
14	14	5	1	50	50	31	0
15	15	6	1	51	51	32	1
16	16	7	1	52	52	33	1
17	17	8	1	53	53	34	1
18	18	8	0	54	54	35	1
19	19	8	1	55	55	36	0
20	20	9	1	56	56	36	0
21	21	10	1	57	57	37	1
22	22	11	1	58	58	37	1
23	23	12	1	59	59	38	0
24	24	13	1	60	60	38	0
25	25	13	1	61	61	39	1
26	26	14	1	62	62	40	1
27	27	14	1	63	63	41	1
28	28	15	1	64	64	42	1
29	29	16	1	65	65	43	1
30	30	17	1	66	66	44	0
31	31	18	1	67	67	45	1
32	32	19	1	68	68	46	1
33	33	20	1	69	69	47	1
34	34	20	1	70	70	48	1
35	35	21	1	71	71	48	1
36	36	22	1	72	72	49	0

Rysunek 6: W celu sprawdzenia poprawności należy porównać pola *ID* tabeli wynikowej z *BookID* testowej. Okazało się, że identyfikatory *ID* wszystkich obserwacji z tabeli porównawczej, dla których w księgozbiorze nie ma dostępnych egzemplarzy, pokrywają się z danymi zwróconymi w ramce wynikowej. Stąd kwerenda 3. działa poprawnie

Zapytanie 4.

Kod zapytania i tabela wynikowa

```
SELECT b.ID, b.Title AS Tytul, MAX(br.Rating) AS NajwyzszaOcena,  
       MIN(br.Rating) AS NajnizszaOcena,  
       ROUND(AVG(CAST(br.Rating AS FLOAT)), 2) AS SredniaOcena  
FROM Books b  
JOIN BookRatings br ON br.BookID = b.ID  
JOIN Readers r ON br.ReaderID = r.ID  
WHERE r.SignOutDate IS NULL  
GROUP BY b.ID, b.Title
```

	ID	Tytul	NajwyzszaOcena	NajnizszaOcena	SredniaOcena
1	1	The Adventures of Sherlock Holmes	4	4	4
2	2	The Hound of the Baskervilles	5	5	5
3	3	The Devil in the White City	0	0	0
4	4	Gone Girl	1	1	1
5	5	In the Woods	5	5	5
6	6	Still Life	4	4	4
7	8	And Then There Were None	4	4	4
8	9	A Simple Favor	4	4	4
9	13	The Da Vinci Code	1	1	1
10	15	The Outsider	5	5	5
11	18	If It Bleeds	4	4	4
12	21	Insomnia	5	3	4,25
13	22	Guns, Germs, and Steel	2	2	2
14	23	War and Peace	4	3	3,5
15	33	Harry Potter and the Goblet of Fire	2	2	2
16	34	Harry Potter and the Order of Phoenix	2	2	2
17	35	Harry Potter and the Half-Blood Prince	3	3	3
18	36	Harry Potter and the Deathly Hallows	2	2	2
19	41	A Dance with Dragons	2	2	2
20	42	The Lord of the Rings	3	3	3
21	43	The Hobbit	5	4	4,33
22	45	Sylvie and Bruno	5	5	5
23	46	The Hunting of the Snark	3	3	3
24	47	The Hunger Games	0	0	0
25	48	Catching Fire	1	0	0,67
26	49	Mockingjay	1	1	1
27	50	The Ballad of Songbirds and Snakes	4	1	1,43

Rysunek 7: Tabela wynikowa zapytania 4.

Fragment tabeli porównawczej

```
SELECT b.ID, b.Title, Rating, SignOutDate
FROM Books b
JOIN BookRatings br ON br.BookID = b.ID
JOIN Readers r ON br.ReaderID = r.ID
ORDER BY ID
```

	ID	Title	Rating	SignOutDate
1	1	The Adventures of Sherlock Holmes	4	NULL
2	1	The Adventures of Sherlock Holmes	4	NULL
3	1	The Adventures of Sherlock Holmes	5	2020-05-02
4	2	The Hound of the Baskervilles	5	NULL
5	3	The Devil in the White City	1	2020-05-02
6	3	The Devil in the White City	0	NULL
7	4	Gone Girl	1	NULL
8	4	Gone Girl	1	NULL
9	4	Gone Girl	1	NULL
10	5	In the Woods	5	NULL
11	6	Still Life	4	NULL
12	6	Still Life	4	NULL
13	6	Still Life	4	NULL
14	8	And Then There Were None	4	NULL
15	8	And Then There Were None	4	NULL
16	9	A Simple Favor	4	NULL
17	9	A Simple Favor	4	NULL
18	11	Bird Box	2	2020-03-01
19	11	Bird Box	5	2020-03-01
20	13	The Da Vinci Code	0	2020-03-01
21	13	The Da Vinci Code	4	2020-03-01
22	13	The Da Vinci Code	0	2020-05-02
23	13	The Da Vinci Code	1	NULL
24	15	The Outsider	5	NULL
25	17	Carrie	2	2020-05-02
26	17	Carrie	4	2020-03-01
27	17	Carrie	2	2020-03-01
28	18	If It Bleeds	4	NULL
29	21	Insomnia	5	NULL
30	21	Insomnia	5	NULL
31	21	Insomnia	3	NULL
32	21	Insomnia	4	NULL
33	22	Guns, Germs, and Steel	2	NULL

Rysunek 8: Poziome linie służą zgrupowaniu książek względem ID. Sprawdzenie zgodności danych w tabelach wymaga ręcznego zliczenia poszczególnych obserwacji i wyznaczenia statystyk, przy czym uwzględniamy tylko te wiersze z tabeli porównawczej, dla których pole *SignOutDate* ma wartość *NULL*. Dane w obydwu tabelach są tożsame, zatem zapytanie działa poprawnie.

Zapytanie 5.

Kod zapytania i tabela wynikowa

```
SELECT e.ID, e.FirstName AS Imie, e.LastName AS Nazwisko, IleWypozycczen
FROM Employees e
RIGHT JOIN
(
SELECT b.EmployeeID, COUNT(*) AS IleWypozycczen
FROM Borrows b
GROUP BY b.EmployeeID
) Tmp
ON e.ID = Tmp.EmployeeID
```

	ID	Imie	Nazwisko	IleWypozycczen
1	1	Gabriella	Green	14
2	2	Tom	Bolton	5
3	3	Benedict	Montez	11
4	4	Anastasia	Montez	3

Rysunek 9: Treść zapytania 1. oraz tabela wynikowa

Tabela porównawcza

```
SELECT b.ID, b.EmployeeID, e.FirstName, e.LastName
FROM Borrows b
JOIN Employees e on b.EmployeeID = e.ID
ORDER BY EmployeeID
```

	ID	EmployeeID	FirstName	LastName
1	7	1	Gabriella	Green
2	8	1	Gabriella	Green
3	9	1	Gabriella	Green
4	10	1	Gabriella	Green
5	11	1	Gabriella	Green
6	12	1	Gabriella	Green
7	13	1	Gabriella	Green
8	14	1	Gabriella	Green
9	15	1	Gabriella	Green
10	16	1	Gabriella	Green
11	17	1	Gabriella	Green
12	18	1	Gabriella	Green
13	28	1	Gabriella	Green
14	29	1	Gabriella	Green
15	1	2	Tom	Bolton
16	2	2	Tom	Bolton
17	23	2	Tom	Bolton
18	24	2	Tom	Bolton
19	19	2	Tom	Bolton
20	20	3	Benedict	Montez
21	21	3	Benedict	Montez
22	22	3	Benedict	Montez
23	25	3	Benedict	Montez
24	26	3	Benedict	Montez
25	27	3	Benedict	Montez
26	3	3	Benedict	Montez
27	30	3	Benedict	Montez
28	31	3	Benedict	Montez
29	32	3	Benedict	Montez
30	33	3	Benedict	Montez
31	4	4	Anastasia	Montez
32	5	4	Anastasia	Montez
33	6	4	Anastasia	Montez

Rysunek 10: Poziome linie oddzielają wypożyczenia przypadające na poszczególnych pracowników. Po ich zliczeniu, okazuje się, że dane przechowywane w porównywanych tabelach są zgodne. Stąd: instrukcja działa poprawnie