

Sprawozdanie

Weronika Sieniawska

1. Temat

Sieć powiązań muzycznych – artyści, albumy, współprace i gatunki muzyczne.

2. Wstęp

Celem projektu było zaprojektowanie i implementacja grafowej bazy danych odwzorowującej powiązania funkcjonujące w świecie muzyki. Model obejmuje artystów, albumy, utwory, gatunki muzyczne, współprace oraz relacje z wytwórniami. Dane zostały załadowane z plików CSV i przetworzone w środowisku Neo4j Aura przy użyciu języka zapytań Cypher.

Projekt demonstruje możliwości baz NoSQL w zakresie modelowania danych o silnie relacyjnym charakterze oraz prezentuje praktyczne zastosowanie grafów w analizie sieci współprac, rekomendacji gatunkowych i struktury twórczości artystów.

3. Wybór technologii

Dane muzyczne mają charakter sieciowy, są ściśle połączone. Artyści ze sobą współpracują, tworzą albumy składające się z wielu utworów i dodatkowo są związani z wytwórniami, taki splątany układ Neo4j potrafi odwzorować najprościej i najwydajniej. Inne bazy NoSQL świetnie radzą sobie z dokumentami czy dużymi zbiorami danych, ale nie są stworzone do analizowania relacji, przez co zapytania o współprace czy zależności między artystami stają się w nich sztucznie skomplikowane. W Neo4j relacje są bardzo ważne, więc można je przeszukiwać bezpośrednio, a zapytania typu „którogo współpracował z kim i jak daleko sięga ta sieć” działają szybko i naturalnie. Dzięki temu ta technologia najlepiej oddaje to, jak faktycznie wygląda świat muzyki, czyli jako sieć połączeń, a nie zbiór odizolowanych rekordów.

4. Model danych

Model został zbudowany na podstawie danych załadowanych z plików CSV. Obejmuje pięć głównych typów węzłów oraz zestaw relacji odzwierciedlających rzeczywiste zależności w świecie muzyki.



Węzły:

Artysta

Właściwości: artysta_id, nazwa, kraj, aktywny_od, aktywny_do

Album

Właściwości: album_id, tytul, rok

Utwór

Właściwości: utwor_id, tytul, dlugosc, numer_w_albumie

Gatunek

Właściwości: gatunek_id, nazwa

Wytwarzająca

Właściwości: wytworzyciel_id, nazwa, kraj

Relacje

(:Artysta)-[:CREATED]->(:Album) – artysta stworzył album
(:Album)-[:CONTAINS]->(:Utwór) – album zawiera utwór
(:Artysta)-[:PERFORMED]->(:Utwór) – artysta wykonał utwór
(:Artysta)-[:FEATURED_ON]->(:Utwór) – gościnny występ
(:Artysta)-[:COLLABORATED_WITH]->(:Artysta) – współpraca między artystami
(:Utwór)-[:IN_GENRE]->(:Gatunek) – utwór należy do gatunku
(:Artysta)-[:SIGNED_TO]->(:Wytwarzająca) – artysta związany z wytwórną

```
1 LOAD CSV WITH HEADERS FROM
2   "https://raw.githubusercontent.com/WeronikaSieniawska/Projekt_ko-
3   cowy_neo4j/main/dane/wspolprace.csv" AS row
4
5 WITH row,
6      TRIM(row.artysta_id_a) AS artystaA,
7      TRIM(row.artysta_id_b) AS artystaB
8 WHERE artystaA <> "" AND artystaB <> ""
9
10 MATCH (a:Artysta {artysta_id: artystaA})
11 MATCH (b:Artysta {artysta_id: artystaB})
12
13 MERGE (a)-[r:COLLABORATED_WITH]-(b)
14 SET r.rok_pierwszej_wspolpracy =
15   toInteger(row.rok_pierwszej_wspolpracy),
16   r.notatki = row.notatki;
```

Completed after 241 ms

```
neo4j$ LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/WeronikaSieniawska/Projekt_ko-cowy_neo4j/main/dane/wspolprace.csv"
```

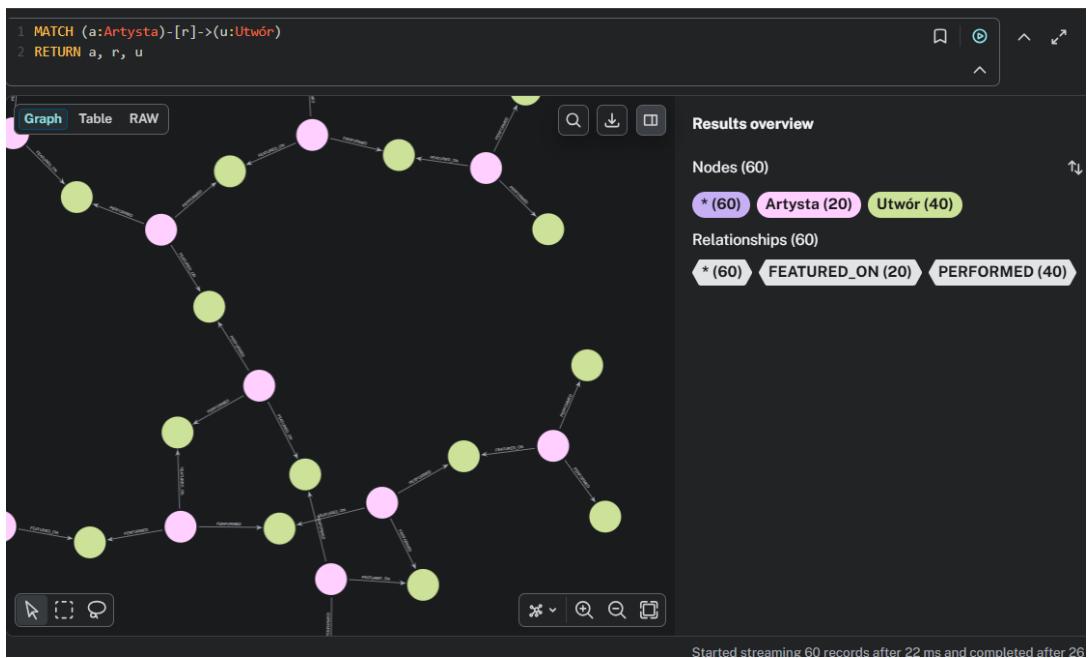
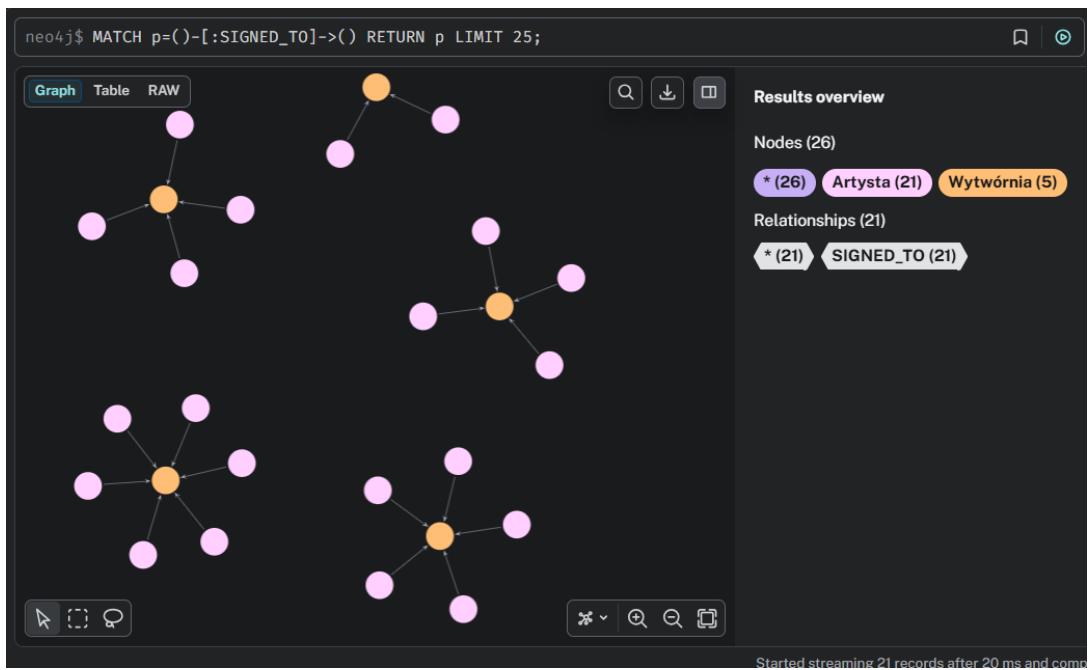
Completed after 196 ms

```
neo4j$ LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/WeronikaSieniawska/Projekt_ko-cowy_neo4j/main/dane/wspolprace.csv"
```

Completed after 271 ms

```
neo4j$ LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/WeronikaSieniawska/Projekt_ko-cowy_neo4j/main/dane/wspolprace.csv"
```

Completed after 255 ms



Relacje te zostały utworzone na podstawie danych wejściowych i są wykorzystywane w zapytaniach analitycznych.

5. Funkcjonalność systemu

Baza umożliwia analizę danych muzycznych na wielu poziomach, w tym:

- przegląd twórczości artystów (albumy, utwory, występy gościnne),
- analizę współpracy bezpośrednich i wielopoziomowych,
- wyszukiwanie powiązań artysta–album–utwór,
- analizę gatunków i rekomendacji,
- analizę struktury wytwórni,
- identyfikację najpopularniejszych elementów (gatunków, utworów, artystów),
- wyszukiwanie najkrótszych ścieżek współpracy.

```
1 MATCH (u:Utwór)-[:IN_GENRE]->(g:Gatunek)
2 RETURN g.nazwa AS gatunek, count(u) AS liczba_utworow
3 ORDER BY liczba_utworow DESC;
```

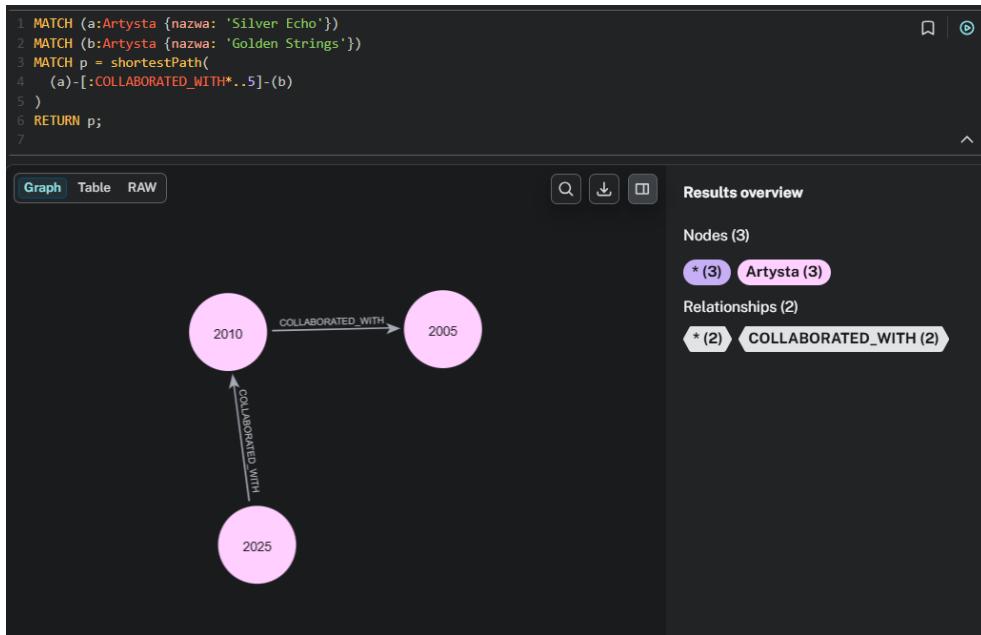
Table RAW

gatunek	liczba_utworow
1 "Elektronika"	10
2 "Pop"	8
3 "Ambient"	8
4 "Indie"	6
5 "Rock"	4
6 "Synthwave"	4


```
1 MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:SIGNED_TO]->(w:Wytwórnia)
2 RETURN w;
```

Graph Table RAW

w
1 Polska



W projekcie zastosowano również zestaw operacji CRUD, umożliwiających rozwój i modyfikację bazy.

Przykładowe operacje CREATE:

- dodanie nowego artysty wraz z przypisaniem do wytwórni,
- utworzenie albumu i automatyczne dodanie utworów,
- dodanie współpracy między artystami.

```

1 MATCH (a:Artysta {nazwa: "Crystal Wave"})
2 CREATE (al:Album {
3   album_id: "AL1200",
4   tytul: "Frozen Echoes",
5   rok: 2025
6 })
7 MERGE (a)-[:CREATED]->(al)
8 WITH al
9 UNWIND [
10   {id:"U1201", t:"Glacier Pulse", d:210, nr:1},
11   {id:"U1202", t:"Northern Lights", d:245, nr:2}
12 ] AS utw
13 CREATE (u:Utwór {
14   utwór_id: utw.id,
15   tytul: utw.t,
16   dlugosc: utw.d,
17   numer_w_albumie: utw.nr
18 })
19 MERGE (al)-[:CONTAINS]->(u);

```

Created 3 nodes, created 3 relationships, set 11 properties, added 3 labels

```

1 MATCH (w:Wytwórnia {nazwa: "Starline Records"})
2 CREATE (a:Artysta {
3   artysta_id: "A1200",
4   nazwa: "Crystal Wave",
5   kraj: "Islandia",
6   aktywny_od: 2018
7 })
8 MERGE (a)-[:SIGNED_TO]->(w);

```

Created 1 node, created 1 relationship, set 4 properties, added 1 label

Przykładowe operacje READ:

- wyszukiwanie artystów bez współpracy,
- identyfikacja utworów bez gatunku,
- sprawdzanie albumów pozbawionych utworów.

```
1 MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:PERFORMED]->(u:Utwór)-[:IN_GENRE]->(g:Gatunek)
2 RETURN u.tytul AS utwor, g.nazwa AS gatunek;
```

Table RAW

utwor	gatunek
1 "Fragments"	"Elektronika"
2 "Neon Skies"	"Elektronika"

Przykładowe operacje UPDATE:

- aktualizacja właściwości artysty,
- zmiana gatunku utworu,

```
1 MATCH (u:Utwór {tytul: "Emberfall"})-[:IN_GENRE]->(:Gatunek)
2 DELETE r
3 WITH u
4 MATCH (g:Gatunek {nazwa: "Elektronika"})
5 MERGE (u)-[:IN_GENRE]->(g)
6 RETURN u.tytul
```

Table RAW

u.tytul

```
1 "Emberfall"
```

✓ Created 1 relationship, deleted 1 relationship

```
1 MATCH (a:Artysta {nazwa: "Crystal Wave"})
2 WHERE a.kraj IS NULL OR a.kraj = ""
3 SET a.kraj = "Islandia";
```

No changes, no records

Przykładowe operacje DELETE:

- usuwanie współpracy artysty,
- usuwanie albumu wraz z jego utworami,
- usuwanie relacji PERFORMED według warunków.

The screenshot shows the Neo4j browser interface. On the left, a code editor displays a Cypher query:

```
1 MATCH (a:Artysta)
2 WHERE NOT (a)-[]->()
3 DELETE a
```

Below the code, a message says "No changes, no records".

The main area shows a graph with two nodes: "G3" (orange circle) and "220" (light green circle). The "Results overview" panel on the right indicates "Nodes (2)" with categories: "*" (2), Gatunek (1), and Utwór (1).

At the bottom, a message says "Deleted 1 relationship" and "Started streaming 1 record after 26 ms".

6. Podsumowanie

Projekt przedstawia kompletny model grafowej bazy danych odwzorowującej sieć powiązań muzycznych. Zastosowanie Neo4j pozwoliło na naturalne i wydajne modelowanie relacji między artystami, albumami, utworami, gatunkami i wytwórniami. Baza umożliwia zarówno podstawowe operacje CRUD, jak i zaawansowane analizy sieciowe, co czyni ją elastycznym i praktycznym narzędziem do pracy z danymi muzycznymi.

Model jest spójny, skalowalny i gotowy do dalszej rozbudowy, a jego struktura stanowi przykład efektywnego wykorzystania baz NoSQL w projektach analitycznych.

7. Przykłady zapytań

-- PODSTAWOWE ZAPYTANIA

-- 1. Wszystkie albumy danego artysty

```
MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:CREATED]->(al:Album)  
RETURN al;
```

-- 3. Wszystkie utwory, na których artysta wystąpił gościnnie (featured)

```
MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:FEATURED_ON]->(u:Utwór)  
RETURN u;
```

-- 4. Wszystkie utwory artysty

```
MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:PERFORMED|FEATURED_ON]->(u:Utwór)  
RETURN u;
```

-- 5. Albumy i utwory artysty

```
MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:CREATED]->(al:Album)-[:CONTAINS]->(u:Utwór)  
RETURN al, u;
```

-- WSPOLPRACE MIEDZY ARTYSTAMI

-- 6. Bezpośrednie współprace artysty

```
MATCH (a:Artysta {nazwa: 'Silver Echo'})-[:COLLABORATED_WITH]-(b:Artysta)  
RETURN DISTINCT b;
```

-- 8. Liczba relacji współpracy

```
MATCH (a:Artysta)-[:COLLABORATED_WITH]-()  
RETURN a.nazwa AS artysta, count(*) AS liczba_wspolprac  
ORDER BY liczba_wspolprac DESC;
```

-- 9. Najkrótsza ścieżka współpracy między dwoma artystami

```
MATCH p = shortestPath(  
    (a:Artysta {nazwa: 'Silver Echo'})-[:COLLABORATED_WITH*]-(b:Artysta {nazwa: 'Golden  
Strings'})  
)  
RETURN p;
```

-- GATUNKI I REKOMENDACJE

-- 10. Wszystkie utwory danego gatunku

```
MATCH (u:Utwór)-[:IN_GENRE]->(g:Gatunek {nazwa: 'Elektronika'})  
RETURN u;
```

```
-- 12. Rekomendacje artystów na podstawie wspólnych gatunków
MATCH (a1:Artysta)-[:PERFORMED]->(u:Utwór)-[:IN_GENRE]->(g:Gatunek)<-[IN_GENRE]-
(u2:Utwór)-[:PERFORMED]->(a2:Artysta)
WHERE a1 <> a2
RETURN a1.nazwa AS Artysta1, a2.nazwa AS Artysta2, count(g) AS WspolneGatunki
ORDER BY WspolneGatunki DESC
LIMIT 10

-- 13. Gatunki najczęściej wykonywane przez artystów współpracujących ze sobą
MATCH (a:Artysta)-[:COLLABORATED_WITH]-(b:Artysta)
MATCH (b)-[:PERFORMED]->(u:Utwór)-[:IN_GENRE]->(g:Gatunek)
RETURN a.nazwa AS artysta, COLLECT(DISTINCT g.nazwa) AS gatunki
ORDER BY SIZE(gatunki) DESC;

-- WYTÓRZNIE

-- 14. Artyści podpisani z daną wytwórnią (Starline Records)
MATCH (a:Artysta)-[:SIGNED_TO]->(w:Wytwórnia {nazwa: 'Starline Records'})
RETURN a;

-- 16. Wytwórnies o największej różnorodności gatunkowej
MATCH (w:Wytwórnia)-[:SIGNED_TO]-(a:Artysta)
MATCH (a)-[:CREATED]->(:Album)-[:CONTAINS]->(:Utwór)-[:IN_GENRE]->(g:Gatunek)
RETURN w.nazwa AS wytwornia, COUNT(DISTINCT g) AS liczba_gatunkow
ORDER BY liczba_gatunkow DESC;

-- ZAPYTANIA ANALITYCZNE

-- 17. Najpopularniejsze gatunki (liczba utworów)
MATCH (u:Utwór)-[:IN_GENRE]->(g:Gatunek)
RETURN g.nazwa AS gatunek, count(u) AS liczba_utworow
ORDER BY liczba_utworow DESC;

-- 20. Utwory z gościnnymi występami
MATCH (gosc:Artysta)-[:FEATURED_ON]->(u:Utwór)
MATCH (gosc)-[:PERFORMED]->(u2:Utwór)
WITH u.tytul AS utwor, collect(DISTINCT u2.tytul) AS goscinnie
RETURN utwor, goscinnie
ORDER BY utwor;

-- 21. Artyści, którzy współpracowali z największa liczba innych artystów
MATCH (a:Artysta)-[:COLLABORATED_WITH]-(b:Artysta)
RETURN a.nazwa AS artysta, count(DISTINCT b) AS liczba_wspolprac
ORDER BY liczba_wspolprac DESC;

-- 24. Najbardziej wpływowe utwory (najwięcej powiązań)
MATCH (u:Utwór)
OPTIONAL MATCH (u)-[:PERFORMED]->(a1:Artysta)
```

```

OPTIONAL MATCH (u)<-[FEATURED_ON]-(a2:Artysta)
OPTIONAL MATCH (u)-[IN_GENRE]->(g:Gatunek)
RETURN u.tytul AS utwor,
       COUNT(DISTINCT a1) + COUNT(DISTINCT a2) + COUNT(DISTINCT g) AS powiazania
ORDER BY powiazania DESC;

-- 25. Wytwornie z największa liczbą artystów
MATCH (a:Artysta)-[:SIGNED_TO]->(w:Wytwórnia)
WITH w, count(a) AS liczba_artystow
ORDER BY liczba_artystow DESC
LIMIT 1
RETURN w.nazwa AS wytwornia, liczba_artystow

-- CRUD

-- 26. Dodanie nowego artysty wraz z automatycznym przypisaniem do wytworni
MATCH (w:Wytwórnia {nazwa: "Starline Records"})
CREATE (a:Artysta {
    artysta_id: "A1200",
    nazwa: "Crystal Wave",
    kraj: "Islandia",
    aktywny_od: 2018
})
MERGE (a)-[:SIGNED_TO]->(w);

-- 27. Dodanie albumu i automatyczne powiazanie go z artystą oraz utworami
MATCH (a:Artysta {nazwa: "Crystal Wave"})
CREATE (al:Album {
    album_id: "AL1200",
    tytul: "Frozen Echoes",
    rok: 2025
})
MERGE (a)-[:CREATED]->(al)
WITH al
UNWIND [
    {id:"U1201", t:"Glacier Pulse", d:210, nr:1},
    {id:"U1202", t:"Northern Lights", d:245, nr:2}
] AS utw
CREATE (u:Utwór {
    utwor_id: utw.id,
    tytul: utw.t,
    dlugosc: utw.d,
    numer_w_albumie: utw.nr
})
MERGE (al)-[:CONTAINS]->(u);

```

```

-- 28. Dodanie współpracy między artystami tylko wtedy, gdy jeszcze nie istnieje
MATCH (a1:Artysta {nazwa: "Silver Echo"})
MATCH (a2:Artysta {nazwa: "Crystal Wave"})
MERGE (a1)-[r:COLLABORATED_WITH]-(a2)
ON CREATE SET r.rok_pierwszej_wspolpracy = 2025,
    r.notatki = "Pierwsza współpraca na albumie Frozen Echoes";

-- 29. Aktualizacja kraju artysty tylko jeśli jest nieznany
MATCH (a:Artysta {nazwa: "Crystal Wave"})
WHERE a.kraj IS NULL OR a.kraj = ""
SET a.kraj = "Islandia";

-- 30. Zmiana gatunku utworu
MATCH (u:Utwór {tytuł: "Northern Lights"})-[r:IN_GENRE]->(:Gatunek)
DELETE r
WITH u
MATCH (g:Gatunek {nazwa: "Elektronika"})
MERGE (u)-[:IN_GENRE]->(g);

-- 31. Usunięcie wszystkich współprac artysty, ale bez usuwania artysty
MATCH (:Artysta {nazwa: "Silver Echo"})-[r:COLLABORATED_WITH]-()
DELETE r;

-- 32. Usunięcie relacji PERFORMED tylko dla utworów krótszych niż 2 minuty
MATCH (a:Artysta)-[r:PERFORMED]->(u:Utwór)
WHERE u.dlugosc < 120
DELETE r;

-- 33. Przeniesienie artysty do innej wytworni
MATCH (a:Artysta {nazwa: "Silver Echo"})-[old:SIGNED_TO]->(:Wytwarznia)
DELETE old
WITH a
MATCH (w:Wytwarznia {nazwa: "Starline Records"})
MERGE (a)-[:SIGNED_TO]->(w);

-- 34. Zastąpienie gościnnego występu pełnym wykonaniem
MATCH (a:Artysta {nazwa: "Silver Echo"})-[f:FEATURED_ON]->(u:Utwór)
DELETE f
MERGE (a)-[:PERFORMED]->(u);

-- 35. Usunięcie artystów, którzy nie mają żadnych relacji
MATCH (a:Artysta)
WHERE NOT (a)-[]->()
DELETE a

```