Informatyka

9. DQL

Opracował: Maciej Penar

Spis treści

[1. Przygotowanie do praktyki 3](#_Toc501220747)

[2. Źródło danych 3](#_Toc501220748)

[3. Structured Query Language (SQL) 4](#_Toc501220749)

[4. Ściąga SQL 5](#_Toc501220750)

# Przygotowanie do praktyki

Pobrać silnik bazy danych: sqlite

Dostępny tutaj: <https://www.sqlite.org/download.html>

Rozpakować. Uruchomić w linii komend za pomocą:

|  |
| --- |
| sqlite3.exe |

Jeśli chcielibyśmy otworzyć bazę danych to wykona to następująca komenda

|  |
| --- |
| sqlite3.exe nazwa\_bazy\_danych.db |

Po uruchomieniu w konsoli powinniśmy zobaczyć następujący komunikat:

|  |
| --- |
| SQLite version 3.21.0 2017-10-24 18:55:49  Enter ".help" for usage hints.  Connected to a transient in-memory database.  Use ".open FILENAME" to reopen on a persistent database.  sqlite> |

I jesteśmy w powłoce sqlite’a.

Przydatne komendy:

|  |  |
| --- | --- |
| Komenda | Co robi |
| .help | Wypisuje listę dostępnych komend |
| .tables | Wypisuje listę tabel |
| .save FILE | Zapisuje bazę danych do pliku |

# Źródło danych

Pobrać próbną bazę SQLite’a stąd:

* <http://www.sqlitetutorial.net/sqlite-sample-database/>
* https://github.com/mpenarprz/InformatykaA1/tree/master/dane/bazy%20danych

Rozpakować otworzyć (pod spodem komenda w Powershell):

|  |
| --- |
| .\sqlite3.exe .\chinook.db |

# Structured Query Language (SQL)

Napisać zapytania SQL. Zwrócić uwagę na **FORMATOWANIE ZAPYTAŃ**.

1. Wyświetlić całą zawartość tabeli *genres* (tzw. dump tabeli).
2. Wyświetlić pierwsze alfabetycznie tytuły pierwszych 5 rekordów z tabeli *albums*
3. Znaleźć kompozytora utworu (‘*tracks*’) o nazwie ‘No Futuro’
4. Znaleźć nazwy utworów oraz czasy trwania (w minutach) utworów które zajmują więcej niż 900000000 bajtów
5. Wyświetlić albumy artysty ‘Van Halen’.
6. Wyświetlić pierwsze alfabetycznie tytuły pierwszych 5 rekordów z tabeli *albums* zaczynających się od ‘*Ar*’
7. Wyświetlić alfabetycznie nazwy albumów które posiadają utwory z gatunku ‘Rock’ oraz ‘Pop’
8. Wyświetlić alfabetycznie utwory które należą do albumu ‘English Renaissance’ lub trwają 5 minut, a ich tytuł zaczyna się od liter
9. Ile jest albumów?
10. Ile jest utworów?
11. Ile jest kompozytorów (nie artystów).
12. Ile jest utworów bez kompozytora?
13. Jaka jest sumaryczna długość utworów. Wynik podać w godzinach.
14. Policzyć zestawienie ile utworów ma album. Na zestawieniu są wszystkie albumy.
15. Policzyć ile jest utworów których autorem jest autor albumu do którego należą te utwory.
16. Wyświetlić nazwy albumów oraz długości ich trwania w minutach które mają więcej niż 1 gatunek.
17. \* Wyświetlić nazwę albumu oraz tytuł najdłuższego utworu tego albumu
18. \* Wyświetlić 10 rekordów. Po 5 najdłuższych płyt w gatunkach Pop oraz Rock.
19. \* Wyświetlić wszystkie pary utworów z albumu ‘Chemical Wedding’ dla których pierwszy utwór z pary jest krótszy od drugiego utworu z pary.

# Ściąga SQL

Ściąga DQL w SQL. Wytłuszczoną czcionką zaznaczono słowa kluczowe.

|  |  |
| --- | --- |
| Przykład | Co oznacza |
| **SELECT**  \*  **FROM**  MY\_TABLE | Pobiera wszystko z tabeli MY\_TABLE |
| **SELECT**  \*  **FROM**  MY\_TABLE  **ORDER BY**  ATT | Pobiera wszystko z tabeli MY\_TABLE, sortuje po atrybucie ATT rosnąco |
| **ORDER BY**  ATT **ASC**,  ATT2 **DESC** | Sortowanie po kilku atrybutach. Specyfikacja sortowania rosnąco ASC, malejąco DESC. |
| **SELECT TOP** 10  \*  **FROM**  MY\_TABLE | Wybranie pierwszych 10 rekordów. Wynik niedeterministyczny. To chyba że użyte z ORDER BY. |
| **SELECT**  \*  **FROM**  MY\_TABLE  **LIMIT** 10 | Wybranie pierwszych 10 rekordów. Wynik niedeterministyczny. To chyba że użyte z ORDER BY. |
| **SELECT DISTINCT**  \*  **FROM**  MY\_TABLE | Pobiera wszystkie unikatowe rekordy z tabeli MY\_TABLE |
| **SELECT**  MY\_ATTRIBUTE AS A,  MY\_ATTRIBUTE2  **FROM**  MY\_TABLE | Pobiera atrybuty MY\_ATTRIBUTE, który zostaje przemianowany na A, oraz atrybut MY\_ATTRIBUTE2 z tabeli MY\_TABLE |
| **SELECT**  **\***  **FROM**  MY\_TABLE AS TTT  **INNER JOIN** YOUR\_TABLE AS KKK **ON** TTT.ATT = KKK.ATT | Pobiera wszystko ze złączenia pomiędzy tabelą MY\_TABLE oraz YOUR\_TABLE. Obu tabelom nadano aliasy (odpowiednio TTT/KKK). Złączenie jest po warunku równościowym na atrybucie ATT |
| **INNER JOIN**  **LEFT OUTER JOIN**  **RIGHT OUTER JOIN**  **FULL OUTER JOIN**  **CROSS JOIN** | Rodzaje złączeń w SQL |
| **SELECT**  **\***  **FROM**  MY\_TABLE,  MY\_TABLE2,  MY\_TABLE3 | Iloczyn kartezjański (CROSS JOIN) table MY\_TABLE, MY\_TABLE2, MY\_TABLE3 |
| **SELECT**  **\***  **FROM**  **([SQL]) ALIAS** | Opakowanie zapytania. W klauzuli FROM można użyć zapytania. |
| **SELECT**  **\***  **FROM**  MY\_TABLE  **WHERE**  A > 0 | Pobiera wszystkie atrybuty z odfiltrowanej tabeli MY\_TABLE. Filtrowanie zachodzi na warunku A > 0. |
| **WHERE**  **[warunek]**  **AND [warunek]** | Łączenie warunków w klauzuli where – logiczne AND |
| **WHERE**  **[warunek]**  **OR [warunek]** | Łączenie warunków w klauzuli where – logiczne OR |
| **NOT [warunek]** | Negacja warunku |
| **WHERE**  **ATT IN (1,2,3,10)** | Sprawdzenie czy atrybut ATT posiada wartość ze zbioru {1,2,3,10} |
| **WHERE**  **ATT IN ([SQL])** | Sprawdzenie czy atrybut ATT posiada wartość ze zbioru – dynamicznie wyliczony zbiór |
| **WHERE**  **EXISTS ([SQL])** | Sprawdzenie niepustości dynamicznie wyliczonego zbioru |
| **WHERE**  MY\_TEXT\_ATTRIBUTE **LIKE [wzorzec]** | Sprawdzenie czy wartość atrybutu MY\_TEXT\_ATTRIBUTE pasuje do wzorca |
| ? (czasem \_) – dowolny znak (regexp: ‘.’)  % - dowolny ciąg znaków (regexp: ‘.’\*) | Specjalny znaki we wzorcach |
| **SELECT**  ATT**,**  **COUNT(\*)**  **FROM**  MY\_TABLE  **GROUP BY**  ATT | Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT. |
| **SELECT**  ATT**,**  **COUNT(\*)**  **FROM**  MY\_TABLE  **WHERE**  A > 0  **GROUP BY**  ATT | Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT. Do agregacji wliczane są **tylko** rekordy spełniające warunek A>0 |
| **COUNT**  **SUM**  **MIN**  **MAX**  **AVG** | Rodzaje funkcji agregujących w SQL – podstawowe |
| **COUNT(\*)** | Wyjątkowa agregacja – ile jest wartości |
| **AVG(WIEK)** | Średnia wartość atrybutu WIEK |
| **COUNT(DISTINCT WIEK)** | Wyjątkowa agregacja – ile różnych wartości znajduje się w grupie |
| **SELECT**  ATT**,**  **COUNT(\*)**  **FROM**  MY\_TABLE  **GROUP BY**  ATT  **HAVING**  **AVG(**TTT**)** > 10 | Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT. Odfiltrowanie tych **grup** dla których agregacja AVG(TTT) osiąga wartość większą niż 10. |
| **[SQL]**  **UNION**  **[SQL]** | Suma wyników dwóch zapytań SQL.  Jako zbiór. |
| **[SQL]**  **UNION ALL**  **[SQL]** | Suma wyników dwóch zapytań SQL.  Jako multizbiór. |
| **UNION**  **UNION ALL**  **MINUS (EXCEPT)**  **MINUS (EXCEPT) ALL**  **INTERSECT** | Możliwe operacje na zbiorach w SQL. |