Katedra
Informatyki i Automatyki
Politechniki Rzeszowskiej

Informatyka

9. DQL

Opracował: Maciej Penar

Spis treści

1. 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

2. 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
```

3. 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.

4. C++

Napisać klasę **MovieDB** realizującą bazę danych dot. filmów **w pliku**.

Kilka wskazówek ogólnych:

- Powinna istnieć możliwość wczytania danych z pliku
- Powinna istnieć możliwość zapisu do pliku
- Format rekordu wewnątrz pliku – dowolny. Sugerowany: xml lub json
- Powinna istnieć gwarancja poprawnego zamknięcia pliku

Dane filmów:

- Identyfikator
- Tytuł
- Gatunek (może być kilka)
- Nazwisko i imię reżysera
- Rok produkcji
- Czas trwania w minutach

Funkcjonalność klasy **MovieDB**:

- Wczytać bazę danych
- Zapisać bazę danych
- Utworzyć nową bazę danych do pliku o wybranej nazwie
- Wyświetlić dane filmu
- Znaleźć film:
    - Na podstawie identfikatora
    - Na podstawie gatunku
    - Na podstawie tytułu

Przestroga końcowa w formie ponurego żartu:

NIE IMPLEMENTOWAĆ JĘZYKA

Zainteresowanych odsyłam: http://www.antlr.org/

5. 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**	Opakowanie zapytania. W klauzuli FROM można użyć zapytania.

([SQL]) ALIAS	
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(*)**	Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT.

FROM MY_TABLE **GROUP BY** ATT **HAVING** **AVG(**TTT**)** > 10	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.