



Katedra  
Informatyki i Automatyki  
Politechniki Rzeszowskiej

# Informatyka

Bazy danych / DDL / DML / DQL

Opracował: Maciej Penar

## Spis treści

Zagadnienia teoretyczne .....	3
Przygotowanie do praktyki.....	3
SQL/DML: Data Manipulation Language .....	4
SQL/DQL: Structured Query Language .....	4
Ściąga SQL .....	5

## Zagadnienia teoretyczne

Wykonać polecenia:

1. Wyjaśnij co to jest Baza Danych, z czego się składa?
2. Co oznacza **pasywność** Bazy Danych?
3. Wyjaśnij co to jest schemat bazy danych.
4. Co to jest relacyjna baza danych?
5. Co to jest relacja?
6. Co to jest związek? W tym związek jeden-do-jednego, jeden-do-wielu oraz wiele-do-wielu. Podaj przykłady.
7. Co to jest klucz, klucz obcy?
8. Co to jest indeks?
9. **Co to jest transakcja? Rozwinąć skrót ACID w kontekście transakcji.**
10. Co to oznacza CRUD?
11. **Jak wyrazić związek wiele-do-wielu w Relacyjnych Bazach Danych?**

## 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

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
```

## SQL/DML: Data Manipulation Language

Napisać SQL wykonujące:

1. Wyrażenie DDL tworzące tabelę 'address'
  - a. Identyfikator
  - b. Miasto, długość 32,
  - c. Województwo, długość 32
  - d. Kod pocztowy, 6 znaków
  - e. Ulica, 50 znaków
  - f. Numer mieszkania/domu, 10 znaków
2. Wyrażenie DDL tworzące tabelę 'person' zawierającą:
  - a. Identyfikator
  - b. Imię, długość 20
  - c. Nazwisko, długość 20
  - d. PESEL, stała długość 10, opcjonalny, unikatowy
  - e. Data urodzenia, format yyyy-mm-dd, opcjonalna
  - f. Numer telefonu, długość 13, opcjonalna (\* jak ktoś potrafi to niech wymusi obligatoryjny + na początku ciągu)
  - g. Wskazanie na rekord adres
3. Dodać do tabeli 'address': (1,'Rzeszów','Podkarpacie','35-100','Aleja X', '50')
4. Dodać do tabeli 'person': (1,'Maciej','Abacki','999999999','+48101101101', '1')
5. Wykonać aktualizację rekordu z tabeli 'address'. Rekord o identyfikatorze 2 powinien mieć ustawiony numer mieszkania na '13b/2').

## SQL/DQL: Structured Query Language

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

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. Policzyc zestawienie ile utworów ma album. Na zestawieniu są wszystkie albumy.
15. Policzyc 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.

## Ściągą SQL

Ściągą DQL w SQL. Wytyłuszczoną czcionką zaznaczono słowa kluczowe.

Przykład	Co oznacza
<b>SELECT</b> * <b>FROM</b> MY_TABLE	Pobiera wszystko z tabeli MY_TABLE
<b>SELECT</b> * <b>FROM</b> MY_TABLE <b>ORDER BY</b> ATT	Pobiera wszystko z tabeli MY_TABLE, sortuje po atrybucie ATT rosnąco
<b>ORDER BY</b> ATT <b>ASC</b> , ATT2 <b>DESC</b>	Sortowanie po kilku atrybutach. Specyfikacja sortowania rosnąco ASC, malejąco DESC.
<b>SELECT TOP 10</b> * <b>FROM</b> MY_TABLE	Wybranie pierwszych 10 rekordów. Wynik niedeterministyczny. To chyba że użyte z <b>ORDER BY</b> .
<b>SELECT</b> * <b>FROM</b> MY_TABLE <b>LIMIT 10</b>	Wybranie pierwszych 10 rekordów. Wynik niedeterministyczny. To chyba że użyte z <b>ORDER BY</b> .
<b>SELECT DISTINCT</b> * <b>FROM</b> MY_TABLE	Pobiera wszystkie unikatowe rekordy z tabeli MY_TABLE
<b>SELECT</b> MY_ATTRIBUTE AS A, MY_ATTRIBUTE2 <b>FROM</b> MY_TABLE	Pobiera atrybuty MY_ATTRIBUTE, który zostaje przemianowany na A, oraz atrybut MY_ATTRIBUTE2 z tabeli MY_TABLE
<b>SELECT</b> * <b>FROM</b> MY_TABLE AS TTT <b>INNER JOIN</b> YOUR_TABLE AS KKK <b>ON</b> TTT.ATT = KKK.ATT	Pobiera wszystko ze złączenia pomiędzy tabelą MY_TABLE oraz YOUR_TABLE. Oba tabelom nadano aliasy (odpowiednio TTT/KKK). Złączenie jest po warunku równościowym na atrybucie ATT
<b>INNER JOIN</b> <b>LEFT OUTER JOIN</b> <b>RIGHT OUTER JOIN</b> <b>FULL OUTER JOIN</b> <b>CROSS JOIN</b>	Rodzaje złączeń w SQL
<b>SELECT</b> * <b>FROM</b> MY_TABLE, MY_TABLE2, MY_TABLE3	Iloczyn kartezjański ( <b>CROSS JOIN</b> ) table MY_TABLE, MY_TABLE2, MY_TABLE3
<b>SELECT</b> * <b>FROM</b>	Opakowanie zapytania. W klauzuli <b>FROM</b> można użyć zapytania.

<b>([SQL]) ALIAS</b>	
<b>SELECT</b> * <b>FROM</b> MY_TABLE <b>WHERE</b> A > 0	Pobiera wszystkie atrybuty z odfiltrowanej tabeli MY_TABLE. Filtrowanie zachodzi na warunku A > 0.
<b>WHERE</b> [warunek] <b>AND</b> [warunek]	łączenie warunków w klauzuli where – logiczne AND
<b>WHERE</b> [warunek] <b>OR</b> [warunek]	łączenie warunków w klauzuli where – logiczne OR
<b>NOT</b> [warunek]	Negacja warunku
<b>WHERE</b> ATT IN (1,2,3,10)	Sprawdzenie czy atrybut ATT posiada wartość ze zbioru {1,2,3,10}
<b>WHERE</b> ATT IN ([SQL])	Sprawdzenie czy atrybut ATT posiada wartość ze zbioru – dynamicznie wyliczony zbiór
<b>WHERE</b> <b>EXISTS</b> ([SQL])	Sprawdzenie niepustości dynamicznie wyliczonego zbioru
<b>WHERE</b> MY_TEXT_ATTRIBUTE <b>LIKE</b> [wzorzec]	Sprawdzenie czy wartość atrybutu MY_TEXT_ATTRIBUTE pasuje do wzorca
? (czasem _) – dowolny znak (regexp: '.') % - dowolny ciąg znaków (regexp: '.*')	Specjalny znaki we wzorcach
<b>SELECT</b> ATT, <b>COUNT</b> (*) <b>FROM</b> MY_TABLE <b>GROUP BY</b> ATT	Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT.
<b>SELECT</b> ATT, <b>COUNT</b> (*) <b>FROM</b> MY_TABLE <b>WHERE</b> A > 0 <b>GROUP BY</b> ATT	Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT. Do agregacji wliczane są <b>tylko</b> rekordy spełniające warunek A>0
<b>COUNT</b> <b>SUM</b> <b>MIN</b> <b>MAX</b> <b>AVG</b>	Rodzaje funkcji agregujących w SQL – podstawowe
<b>COUNT</b> (*)	Wyjątkowa agregacja – ile jest wartości
<b>AVG</b> (WIEK)	Średnia wartość atrybutu WIEK
<b>COUNT</b> (DISTINCT WIEK)	Wyjątkowa agregacja – ile różnych wartości znajduje się w grupie
<b>SELECT</b> ATT, <b>COUNT</b> (*)	Utworzenie grup po wartościach atrybutu ATT oraz wyliczenie agregacji typu COUNT.

<b>FROM</b> MY_TABLE <b>GROUP BY</b> ATT <b>HAVING</b> AVG(TTT) > 10	Odfiltrowanie tych <b>grup</b> dla których agregacja AVG(TTT) osiąga wartość większą niż 10.
[SQL] <b>UNION</b> [SQL]	Suma wyników dwóch zapytań SQL. Jako zbiór.
[SQL] <b>UNION ALL</b> [SQL]	Suma wyników dwóch zapytań SQL. Jako multizbiór.
<b>UNION</b> <b>UNION ALL</b> <b>MINUS (EXCEPT)</b> <b>MINUS (EXCEPT) ALL</b> <b>INTERSECT</b>	Możliwe operacje na zbiorach w SQL.