



Informatyka

8.Bazy danych / DDL / DML

Opracował: Maciej Penar

Spis treści

1. Zagadnienia teoretyczne	3
2. Przygotowanie do zajęć.....	3
3. DDL	4

1. 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 oznacza ACID w kontekście transakcji.
10. Co to oznacza CRUD?
11. Co oznacza stwierdzenie „Schemat Bazy Danych w 1 postaci normalnej”. Ile postaci znasz. Co oznaczają.
- 12. Jak wyrazić związek wiele-do-wielu w Relacyjnych Bazach Danych?**
13. Wyróżyć konstruktywną opinię na temat wpływu postaci normalnych na wydajność zapytań w Bazach Danych.
14. Wyróżyć konstruktywną opinię na temat przetwarzania transakcyjnego (OLTP) oraz analitycznego (OLAP). Czy intuicja podpowiada Ci jakiego typu zapytania pojawiają się w obu rodzajach.
15. Wyróżyć konstruktywną opinię na temat doboru klucza głównego. GUID czy surogat całkowitoliczbowy?

2. 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

3. SQL: Data Definition Language

Zaprojektować bazę danych dla lokalnego sklepu spożywczego. Pan szef sklepikarz opowiada o swoim biznesie mniej więcej tak: *potrzebuję skomputeryzować mój biznes. Chciałbym móc wyświetlać paragon klientowi z pewną liczbą pozycji na paragonie. Każda pozycja to konkretny towar, ze swoją nazwą, rodzajem, dostawcą. Oczywiście że klient może kupić kilka razy ten sam towar, ale na paragonie wyświetlam wtedy tylko jedną pozycję ale opisuje ją liczbą ile danego towaru klient kupił. Czasem zdarzają się niezgodności na paragonach, więc chciałbym odróżniać jakoś wycofane pozycje, czy nawet wycofane paragony. Szczerze powiedziawszy, ja klientów nie odróżniam i nie potrzebuję – są dla mnie anonimowi. W przeciwieństwie do dostawców towarów – potrzebuje ich nazwy, telefonu kontaktowego i adresu. Do dyspozycji w sklepie mamy kilka kas oraz pracuje u mnie kilku sprzedawców wśród których panuje hierarchia. Jestem oczywiście na szczycie tej hierarchii, ale nie wiem ile pracowników jest pod mną. Sprzedawcy przed skorzystaniem z kasy muszą wpisać się na listę informującą o której godzinie rozpoczęli pracę i na zakończenie muszą uzupełnić o której godzinie skończyli. Kasy identyfikowane są fabrycznie przez GUID.*

Wybrać jedną z opcji:

1. Utworzyć bazę danych sqlite za pomocą wyrażeń CREATE TABLE, zapisać w pliku shop.db
2. Napisać tylko wyrażenia CREATE TABLE
3. Narysować diagram (https://pl.wikipedia.org/wiki/Diagram_zwi%C4%85zk%C3%B3w_encji)

4. SQL: 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 'address': (2,'Rzeszów','Podkarpackie','35-101','Ulica słoneczna', '12b/2')
5. Dodać do tabeli 'person': (1,'Maciej','Abacki','999999999','+48101101101', '1')
6. Wykonać aktualizację rekordu z tabeli 'address'. Rekord o identyfikatorze 2 powinien mieć ustawiony numer mieszkania na '13b/2').
7. Trudne zadanie:
 - a. Znaleźć sposób na wyświetlenie „opisu zapytania” (planu logicznego)

- b. **Sprawdzić plan zapytania** „SELECT Imię, Nazwisko FROM person WHERE telefon = ?” dla niepoindeksowanej tabeli. Zamiast ? podać jakiś numer telefonu. Zamiast „Imię” i „Nazwisko” podać nazwy atrybutów wynikające z wyrażenia „CREATE TABLE”
- c. Utworzyć indeks na atrybucie „telefon”
- d. **Ponownie sprawdzić plan zapytania** - skomentować