

Bazy Danych

Ćwiczenie 5: Implementacja przykładowej struktury bazy relacyjnej

opracował: dr hab. inż. Artur Gramacki (a.gramacki@issi.uz.zgora.pl)

1. Cel ćwiczenia

Celem ćwiczenia jest utworzenie w języku SQL przykładowej struktury relacyjnej. Ćwiczenie rozpoczyna się od zapoznania się z przygotowanymi przez nauczyciela szczegółowymi założeniami. Należy określić wymagane relacje i związki między nimi. Następnie, z wykorzystaniem programu *Toad Data Modeler* (TDM) omawianego na poprzednim laboratorium, należy zaimplementować strukturę relacyjną. Na koniec należy utworzyć skrypt ładujący zestaw przykładowych danych (po kilka-kilkanaście rekordów do każdej tabeli).

2. Założenia

Należy zaprojektować relacyjny model danych wg. założeń podanych poniżej. Każde z tych założeń musi być uwzględnione w projektowanym modelu. Efektem końcowym powinien być rysunek pokazujący poszczególne tabele (wraz z wyszczególnieniem ich kolumn) oraz powiązania integralnościowe między nimi (klucze obce).

Tworzymy bazę danych pt „Elektroniczny indeks”.

- O studentach rejestrujemy następujące dane: imię, nazwisko, miejscowość z której pochodzi, kod pocztowy, pesel oraz numer indeksu. Miejscowości powinny być *skatalogowane*. Każdy student studiuje na dokładnie jednym wydziale (nie rozważamy sytuacji, gdy np. student ma ustalony indywidualny tok studiów i może go realizować na kilku różnych wydziałach Uczelni jednocześnie).
- Każdy student w czasie trwania studiów uczy się wielu różnych przedmiotów, które są wykładane przez różnych wykładowców. Wykładowcy nie są przywiązani na trwałe do przedmiotów i może się zdarzyć (i tak często bywa w praktyce), że różni studenci mają ten sam przedmiot z różnymi wykładowcami.
- Rejestrujemy oceny, jakie studenci otrzymują (wpisy do indeksu) z poszczególnych przedmiotów. Z danego przedmiotu dany student może otrzymać tylko jedną ocenę. Musimy jednak przewidzieć sytuacje, gdy student nie zalicza przedmiotu (dostaje ocenę niedostateczną i jest ona wpisana do indeksu) i ma możliwość poprawki. Wówczas otrzymana ocena poprawkowa jest drugą oceną wpisaną do indeksu (pierwszą jest otrzymana wcześniej ocena niedostateczna).

- Rejestrujemy też daty wpisów ocen do indeksu. Poszczególne przedmioty dla poszczególnych studentów wykładane są w danym semestrze (letni lub zimowy) oraz w danym roku akademickim (np: 2015/2016). Informacje te muszą znaleźć się w naszym elektronicznym indeksie (będą np. pozwalały wyliczać średnie ocen studentów w poszczególnych semestrach).
- Jeżeli uznasz, że jakieś dodatkowe informacje powinny również być przechowywane w naszym elektronicznym indeksie, zaimplementuj je.

Zwrócić uwagę na sensowne zaprojektowanie:

- nazw dla tabel i kolumn,
 - typów danych dla poszczególnych kolumn,
 - ograniczeń nakładanych na kolumny,
 - kluczy głównych oraz obcych.
3. Wygenerować w programie TDM kompletny skrypt implementujący zaprojektowaną bazę danych. Uruchomić go w konsoli MySQL-ai sprawdzić czy działa poprawnie.
- Pamiętać o wpisaniu na początku skryptu poleceń kasujących (`DROP`) wszystkie tabele w modelu, gdyż dobrym zwyczajem jest poprzedzenie tworzenia tabel wykasowaniem istniejących. Próba utworzenia tabeli o takiej samej nazwie jak już istniejąca nie powiedzie się - chyba, że użyjemy konstrukcji `CREATE TABLE IF NOT EXISTS`. Szczegóły patrz dokumentacja. Polecenia kasujące tabele można wpisać ręcznie ale wygodniej, aby zrobił to za nas program TDM.
4. Utworzyć skrypt ładujący do poszczególnych tabel zestaw przykładowych danych (poleceniem `INSERT`). Dane powinny być „sensowne” (przykładowo dla kolumn: *imie*, *nazwisko*, *data_urodzenia* sensowne dane to „Marcin Kowalski 12-10-1954” a zupełnie bezsensowne „jghfgsas jhfjfdjf 01-01-1400”). W każdej tabeli powinno znaleźć się przynajmniej po kilka rekordów. Tam, gdzie wyda się to celowe, rekordów tych powinno być więcej.