Uniwersytet w Białymstoku

Wydział Matematyczno-Fizyczny

Instytut Matematyki

Elżbieta Izabela Misiewicz

Projekt i implementacja systemu do układania harmonogramu zajęć

Praca dyplomowa napisana
pod kierunkiem
dr hab. K. Prażmowskiego

Białystok 2003

Składam serdeczne podziękowania

Elżbieta Izabela Misiewicz

Spis treści

Wstęp

Tekst wstępu do pracy. Tekst wstępu do pracy.

Rozdział 1

MySQL relacyjna baza danych

Obecnie coraz częściej zachodzi potrzeba przechowywania i pobierania, oraz dokonywania operacji na dużej ilości danych. Do przechowywania danych najlepiej używać baz danych, które wykorzystywane są w wielu dziedzinach życia. Używane są np.: w bankach, w dużych przedsiębiorstwach do przechowywania informacji o kontach albo danych personalnych. Przechowywanie danych w bazach odbywa się w określony sposób. Na bazę składają się tabele, a w nich rekordy podzielone na pola. Bazą danych, która została wykorzystana przeze mnie do utworzenia systemu wspomagającego układanie planów zajęć jest relacyjna baza danych - MySQL, obsługującą język zapytań SQL.

1.1 SQL

SQL to Strukturalny Język Zapytań (Stuctured Query Language), najbardziej powszechny i ustandaryzowany język dostępu do baz danych, język czwartej generacji, który został w ciągu wielu lat opracowany przez grupę badawczą IBM. Stał się międzynarodowym standardem dla języków baz danych i występuje obecnie w produktach większości liczących się firm, zajmujących się sprzedażą oprogramowania dla baz danych.
Ponieważ SQL jest językiem standaryzowanym, dostęp do danych jest możliwy przez wiele różnych platform sprzętowych, w tym komputery osobiste, mikrokomputery i duże komputery. Inna korzyść związana z SQL polega na tym, że takie oprogramowanie jak arkusze kalkulacyjne czy procesory tekstów mogą importować dane z baz danych opartych na SQL.
Polecenia SQL są stosowane w celu uzyskania dostępu do danych i sterowania operacjami w bazie danych. Użytkownik mający małe doświadczenie w przetwarzaniu danych lub nie mający go wcale, może szybko nauczyć się podstawowych konstrukcji SQL, a ekspert od przetwarzania danych, może znaleźć w tym języku pełny zestaw potrzebnych mu narzędzi. Zatem jest to język, którego używają zarówno zwykli użytkownicy, jak i fachowcy.
SQL jest językiem strukturalnym, zdefiniowanym za pomocą reguł składnio-

wych. Występują w nim trzy rodzaje poleceń. Pierwszy to polecenia języka definiowania danych, które umożliwiają tworzenie obiektów bazy danych, takich jak tabela. Drugi rodzaj to polecenia języka operowania danymi, które są używane do wydobywania informacji z bazy danych, usuwania informacji czy też dodawania ich do niej. Trzeci rodzaj to polecenia języka administrowania danymi, które służą do przyznawania i odwoływania uprawnienia dostępu do bazy danych.
Język SQL jest o wiele bardziej wydajny niż tradycyjne języki programowania, takie jak np. Cobol. Program może być zapisany w kilku wierszach kodu, a nie jak w przypadku tradycyjnego języka programowania - na kilku stronach. Z tego powodu koszty tworzenia aplikacji są jedynie ułamkiem kosztów występujących przy użyciu tradycyjnych języków. SQL może być użyty jako standardowe narzędzie umożliwiające dostęp do danych w różnorodnych środowiskach z różnym sprzętem komputerowym i różnymi systemami operacyjnymi.

1.2 MySQL

MySQL (relacyjna baza danych) to serwer bazodanowy, wyprodukowany przez firmę T.c.X DattaKonsultAB. MySQL jest wielowątkowym systemem zarządzania bazą danych zdolnym pomieścić nawet kilkadziesiąt milionów rekordów (wielkość ta zależy jedynie od fizycznych możliwości komputera), obsługujący język zapytań SQL . Opracowany w 1996 roku, miał stanowić serwer SQL umożliwiający bardzo szybką obsługę transakcji w bardzo dużych bazach danych. Jest jednym z najpopularniejszych interface-ów języka SQL, charakteryzującym się ogromną elastycznością, jeszcze większą prędkością, pracą z wieloma użytkownikami oraz co najważniejsze jest prosty w obsłudze.
MySQL doskonale współpracuje z językami programowania: C, Perl, Java (poprzez interfejs JDBC), Pyton oraz Meta-HTML i PHP. MySQL może zostać uruchomiony na maszynach pracujących pod kontrolą takich systemów jak Unix, Mac OS, OS/2, Windows. W MySQL-u można między innymi tworzyć nowe bazy danych, a w nich tabele, dodawać nowe rekordy, edytować je lub usuwać. MySQL jest relacyjnym systemem zarządzania bazą danych (RDBMS). Relacyjna baza danych przechowuje dane w oddzielnych tabelach, zamiast umieszczać wszystko w jednym zbiorze danych. Takie podejście owocuje szybkością i elastycznością. Tabele są połączone ze sobą poprzez zdefiniowane związki między nimi co umożliwia łączenie na żądanie danych z kilku tabel. Relacyjne bazy danych są dużo wydajniejsze niż pliki tekstowe, a ich obsługa za pomocą języka zapytań SQL jest dosyć prosta. Bazy te składają się z kolumn i rekordów. Każda kolumna ma swoją nazwę, a każdy rekord przechowuje inną pozycję. Pola są określonego typu, dlatego można w nich łatwo przechowywać teksty, liczby, daty czy obrazki. Klucze podstawowe umożliwia-

ją jednoznaczną identyfikację danego rekordu. W relacyjnych bazach danych kolejność rekordów nie ma znaczenia, ponieważ za pomocą języka SQL można nie tylko wybrać interesujące nas rekordy, ale również je posortować.
w MySQL-u istnieją jednak też pewne ograniczenia. MySQL nie zapewnia pełnej implementacji SQL, często więc obsłużenie pewnych funkcji jest nieco bardziej złożone niż w komercyjnych systemach, np. brak transakcji, zagnieżdżonych zapytań. Mimo tego ten system relacyjnych baz danych jest darmowy i ciągle rozwijany (na licencjach OpenSource), a co za tym idzie, oferuje wciąż coraz więcej możliwości, które sprawiają, iż stanowi poważną konkurencję dla podobnych, lecz komercyjnych produktów. Spójność, szybkość i bezpieczeństwo sprawiają, że MySQL jest doskonałym narzędziem stosowanym przy dostępie do baz danych przez Internet.

1.3 Składnia języka SQL

Baza danych złożona jest z tabel w których umieszczane są informacje. Aby utworzyć tabelę w MySQL-u używa się polecenia `CREATE`.Składnia polecenia (w uproszczeniu) jest następująca:

```
CREATE TABLE nazwa_tabeli
(nazwa_kolumny typ_danych[(długość) opcje],
...nazwa_kolumny typ_danych[(długość) opcje]) [opcje_tabeli]
```

Każda kolumna w zależności od rodzaju informacji w niej się znajdujących posiada własny typ danych. W MySQL-u występują m.in. następujące typy danych:

- `CHAR`(długość) Ten typ danych (char = znak) jest używany do przechowywanie danych tekstowych o z góry ustalonej długości z ograniczeniem do 255 znaków.
- `VARCHAR`(długość) Pole tego typu (VAR = variable = zmienny) jest modyfikacją pola typu `CHAR`. Przechowuje ono tylko tyle danych, ile ich naprawdę jest w przeciwieństwie do pola `CHAR`, które bez względu na długość danych przechowuje ich tyle, ile jest zadeklarowane. `VARCHAR` jest znacznie bardziej elastycznym rodzajem pola niż `CHAR`, należy jednak pamiętać iż jest on również znacznie wolniejszy - czasami, jak mówi dokumentacja, nawet do 50 procent.
- `INTEGER`(długość) [Unsigned] Ten typ danych przechowuje liczby z zakresu od -2147483648 do 2147483647. Użycie opcjonalnego pola Unsigned oznacza, iż nie chcemy przechowywać informacji o ewentualnym znaku liczby (np. dane są zawsze dodatnie) - wtedy zakres pola `INT` zmienia się na : 0 do 4294967295.

- `DATE` Pole tego typu przechowuje dane związane z datą. Domyślnym formatem przechowywania danych jest "RRRR-MM-DD". Zakres tego pola sięga od "0000-00-00" do "9999-12-31".
- `TEXT/BLOB` Powyższe typy danych używane są, jeśli zaistnieje potrzeba przechowania dłuższego ciągu znaków (z zakresu od 255 do 65535 znaków). W odróżnieniu od pól typu `CHAR` lub `VARCHAR` nie następuje tutaj obcięcie końcówki danych (chyba że ilość znaków, które chcemy umieścić w tym polu, przekracza jego górną granicę). Jedyna różnica zachodząca między `BLOB`-em a `TEXT`-em jest taka, iż pola typu `TEXT` są porównywane bez rozróżnienia na wielkość liter a pola typu `BLOB` oczywiście z uwzględnieniem wielkości liter.

Po typie danych w nawiasach kwadratowych umieszczane są elementy opcjonalne. Opcje, które mogą wystąpić po określeniu typu i długości danych to np. `PRIMARY KEY, NULL, NOT NULL, UNIQUE, DEFAULT` oraz kilka innych.

- `PRIMARY KEY` to klucz służący do rozróżnienia różnych danych. Żadne dwa pola w jednej tablicy nie mogą mieć tego samego klucza (primary key).
- `AUTO_INCREMENT` - kolumna z tą funkcją wzrasta swą własną wartość o jeden wyżej od poprzedniej. Przykładem pola, w którym niemal wymagane jest użycie tej funkcji jest np. pole typu liczba porządkowa.
- `NOT NULL` - oznacza, iż danej kolumnie nie można przypisać wartości NULL (pusta).

Aby obejrzeć strukturę tabeli, czyli informacje jakie rekordy (o jakiej nazwie) i jakiego typu są w niej przechowywane, należy użyć polecenia `DESCRIBE`. Do pobierania danych z tabeli lub tabel na podstawie zadanych warunków służy instrukcja `SELECT`, podstawowa instrukcją języka SQL. Wynikiem jej wywołania (o ile nie wystąpi błąd) jest zawsze pewna tabela.
Składnia tej instrukcji jest dość złożona, należy przy tym pamiętać, że kolejność klauzul (tj. odpowiednich słów kluczowych) jest istotna.

```
SELECT wyrażenie1, wyrażenie2, ... FROM tabela WHERE warunek
```

Warunek podany po słowie kluczowym `WHERE` ogranicza działanie instrukcji `SELECT` do wierszy spełniających ten warunek. Powinien on być wyrażeniem logicznym, zbudowanym z wykorzystaniem nazw kolumn tabeli.

Rozdział 2

Tabele

Tworząc projekt, który ma wspomagać układanie harmonogramu zajęć dla inststytu szkoły wyższej, należy posiadać odpowiednie informacje. W tym rozdziale zostaną omówione wszelkie niezbędne do zaplanowania, a także do późniejszej realizacji czyli układania harmonogramu zajęć, ustalania obciążeń pracowników, a także do sprawnego systemu rezerwacji sal, dane.
Zostały one posegregowane i zgromadzone w kilku tabelach, między którymi zachodzą pewnego rodzaju relacje. Każda tabela zawiera pięć identycznych rekordów, czyli: `cID, cTS, cCT,cOWNER` i `cGROUP`. Są to standardowe, niezbędne, wykorzystywane przez RDM, czyli Remote Data Managment, rekordy. Pierwszy z nich `cID` jest typu *integer* i stanowi on klucz tabeli. Rekord każdej tabeli dzięki numerowi `cID` ma unikalną numerację.`cTS`(Time Stamp), podobnie jak `cCT`(Create Time) (jest typu *timestamp*. `cTS` informuje o czasie ostatniej modyfikacji,a `cCT` o czasie utworzenia rekordu. `cOWNER` i `cGROUP`, to ciągi znaków *varchar*, które tak jak w UNIX mówią o prawach do plików. I tak, `cOWNER` określa właściciela rekordu,a `cGROUP` grupię właścicieli danych rekordów.
W bazie danych MYSQL znajdują się tabele ze szczegółowymi informacjami na temat pracowników Wydziału Matematyczno-Fizycznego, przedmiotów na określonych kierunkach i specjalnościach,rodzajów prowadzonych zajęć. Ponieważ niektóre zajęcia prowadzone są przez osoby nie będące pracownikami wydziału, sytuacja ta spowodowała powstanie tabeli tNauczyciele.

2.1 Nauczyciele

tNauczyciele

Field	Type	Null	Key	Default	Extra
cID	int(11)		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	

```
| cCT        | timestamp(14) | YES  |     | NULL    |                |
| cOWNER     | varchar(16)   | YES  |     | NULL    |                |
| cGROUP     | varchar(16)   | YES  |     | NULL    |                |
| pracownik  | int(11)       | YES  |     | NULL    |                |
| nauczyciel | varchar(64)   | YES  |     | NULL    |                |
+------------+---------------+------+-----+---------+----------------+
```

W tej tabeli do pięciu standardowych rekordów `cID, cTS, cCT,cOWNER, cGROUP` zostały dodane jeszcze dwa `pracownik` i `nauczyciel`. Każdej osobie prowadzącej zajęcia został przypisany unikalny numer `cID`. Rekord `pracownik` jest zmienną typu *integer*.Jest to numer pobierany z tabeli tPracownicy (tam występuje jako cID)aby odpowiednio i jednoznacznie zidentyfikować osobę, która jest pracownikiem Wydziału Matematyczno-Fizycznego prowadzącą dane zajęcia. `Nauczyciel` jest zmienną typu *varchar*. Umieszczone w tym rekordzie informacje to nazwiska i imiona wszystkich osób, które prowadzą zajęcia, niezależne od tego czy są, czy też nie, pracownikami wydziału. Tabele tNauczyciele i tPracownicy doskonale obrazują relacyjność bazy danych, a w naszym przypadku MySQL-a. Trzymane w tabelach informacje są w pewien logiczny sposób powiązane ze sobą. I tak np.:w tPracownicy znajdują się imiona i nazwiska pracowników Wydziału Matematyczno-Fizycznego,a w tabeli tNauczyciele umieszczone są już tylko odpowiednie numerki, w których zakodowane są wszelkie informacje na temat danego pracownika. Gdyby odwoływać się do `pracownika` poprzez imię i nazwisko, mogłoby to nie być jednoznaczne np.: w przypadku gyby dwie osoby pracujące na wydziale miałyby to samo imię i nazawisko, nie wiedzielibyśmy tak naprawdę o którego pracownika dokładnie chodzi.

2.2 Życzenia

Kolejna tabela to tabela tZyczenia. W niej znajdują się wszelkie informacje niezbędne do sprecyzowania "życzeń" nauczycieli, czyli preferowanych do prowadzenia zajęć dni i godzin.

tZyczenia

```
+--------------+------------------+------+-----+---------+----------------+
| Field        | Type             | Null | Key | Default | Extra          |
+--------------+------------------+------+-----+---------+----------------+
| cID          | int(11)          |      | PRI | NULL    | auto_increment |
| cTS          | timestamp(14)    | YES  |     | NULL    |                |
| cCT          | timestamp(14)    | YES  |     | NULL    |                |
| cOWNER       | varchar(16)      | YES  |     | NULL    |                |
```

cGROUP	varchar(16)	YES		NULL	
pracownik	int(11)	YES		NULL	
semestr	int(2)	YES		NULL	
dzien	varchar(16)	YES		NULL	
godz	varchar(16)	YES		NULL	
zyczenia	enum('T','O','N')	YES		O	
uzasadnienie	varchar(128)	YES		NULL	

Oprócz standardowych pięciu rekordów w tabeli znajdują się rekordy: `pracownik`, właściwie numer określający pracownika pobierany z tNauczyciele. Dalej `dzien` (dzień) i `godz` (godzina), zapisane w tabeli jako *varchar*. Kolejny rekord nazwany jest `zyczenia` (życzenia). I tu każdy pracownik składając swoje życzenia ma do wyboru jedną z trzech opcji:T - czyli "tak", N-"nie" i O-"jest mi to obojętne". Jeżeli pracownik oznaczy jakiś termin jako "N" w kolejnej rubryce zwanej `uzasadnienie`, będzie musiał podać powód dlaczego jego zajęcia nie mogą odbywać się w tym właśnie terminie. `Uzasadnienie` ma typ *varchar*.

2.3 Fakultety

Od trzeciego roku w Instytucie Matematyki pojawiają się przedmioty fakultatywne. Są to przedmioty wybierane przez studentów z listy zaproponowanych przedmiotów, związane z rodzajem seminariów, w których studenci uczestniczą. Ponieważ są to przedmioty wybierane, wystąpiła potrzeba utworzenia tabeli z takimi właśnie informacjami. W rekordach `przedmiot, nazwaprzed,fakultet,kierunek, rok, spec, osob` są takie dane gromadzone.

tFakultety

Field	Type	Null	Key	Default	Extra
cID	int(11)		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	
cCT	timestamp(14)	YES		NULL	
cOWNER	varchar(16)	YES		NULL	
cGROUP	varchar(16)	YES		NULL	
przedmiot	int(11)	YES		NULL	
nazwaprzed	varchar(128)	YES		NULL	
fakultet	varchar(64)	YES		NULL	
kierunek	varchar(16)	YES		NULL	

rok	int(11)	YES		NULL	
spec	varchar(16)	YES		NULL	
osob	int(11)	YES		NULL	

Dla przykłądu na trzecim roku studiów dziennych matematyki ogólnej występują cztery wykłady fakultatywne. Należy więc każdemu z wykładów przypisać odpowiedni przedmiot z listy przedmiotów fakultatywnych. Dlatego też w tabeli tFakultety pojawiły się rekordy `przedmiot, nazwaprzed, fakultet`. `Przedmiot` jest typu *int*, jest to liczba całkowita pobierena z tabeli tECTSmat podobnie jak nazwa przedmiotu `nazwaprzed`, która jest typu *varchar*. `Fakultet` typu *varchar* to odpowiednia nazwa wykładu fakultatywnego np.: wykład fakultatywny II. Oczywiście dopisując odpowiednią nazwę przedmiotu fakultatywnego do wykładu fakultatywnego musimy wziąć pod uwagę `kierunek, rok ,spec`(specjalność) dla której dany przedmiot ma być przypisany. Ważną informacją jest również liczba osób uczestnicząca w konkretnym wykładzie fakultatywnym. Dlatego też ostatni rekord zawiera właśnie taką informację `osob`.

Jeżeli posiadamy już pełne informacje na temat wszelkich zajęć prowadzonych na wydziele, kolejnym krokiem jest przypisanie osoby prowadzącej do konkretnego przedmiotu. W tym celu utworzona została tabela tObsada.

2.4 Obsada

tObsada

Field	Type	Null	Key	Default	Extra
cID	int(11)		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	
cCT	timestamp(14)	YES		NULL	
cOWNER	varchar(16)	YES		NULL	
cGROUP	varchar(16)	YES		NULL	
pracownik	int(11)	YES		NULL	
przedmiot	int(11)	YES		NULL	
kierunek	varchar(16)	YES		NULL	
rodz_zajec	enum('W','C', 'L','K','S')	YES		NULL	
grup	int(11)	YES		NULL	
godz_grup	int(11)	YES		NULL	

W tej tabeli znajduje się jedenaście rekordów. Poza standarowymi pięcioma jest sześć rekordów niezbędnych do planowania i przyporządkowania pracwnikom odpowiednich przedmiotów.
Poza `pracownikiem` oraz `przedmiotem` w tabeli znajdują się rekordy `kierunek`, rodz_zajec, `grup` oraz godz_grup. Rekord `pracownik` podobnie jak przyporządkowany mu `przedmiot` jest typu *integer*, czyli jest to liczba całkowita, która jest przekazywana z tabeli tNauczyciele (tam jako cID)aby odpowiednio i jednoznacznie zidentyfikować osobę prowadzącą dane zajęcia. Aby uniknąć niejednoznaczności `pracownik` jest zmienną typu *integer* . `Przedmiot` jest także typu *integer*, a nazwy wszystkich przedmiotów przekazywane są z tabeli tECTSmat. Kolejny rekord to `kierunek` typu *varchar*, w który zawiera informacje o kierunku studiów.Dzięki niemu można po odpowiednich modyfikacjach układać plany zajęć nie tylko dla Instytutu Matematyki. Rekord rodz_zajec informuje szczegółowo o typie zajęć, i tak do wyboru mamy: W-"wykład", C-"ćwiczenia,L-"laboratorium", K-"konwersatorium" i S-"seminarium". Chodzi tu o odpowiednie dobranie wielkości sali do rodzaju zajęć oraz ustalenia ilości studentów w nich uczestniczących. Pozostałe rekordy `grup` dotyczy ilości grup uczestniczących w zajęciach, a godz_grup ilości godzin zajęć przeznaczonych na jedną grupę . Obie te wielkości są w tabeli zapisane jako *integer*.

2.5 Liczba studentów

Ważną rzeczą by prawidłowo utworzyć harmonogram zajęć jest wiedza dotycząca nie tylko przedmiotu oraz osób go prowadzących, ale także ilości studentów na poszczególnych kierunkach i specjalnościach. Dlatego właśnie została utworzona w bazie kolejna tabela nazwana tLiczStud.

tLiczStud

Field	Type	Null	Key	Default	Extra
cID	int(11)		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	
cCT	timestamp(14)	YES		NULL	
cOWNER	varchar(16)	YES		NULL	
cGROUP	varchar(16)	YES		NULL	
kierunek	varchar(16)	YES		NULL	
spec	varchar(16)	YES		NULL	
rok	int(11)	YES		NULL	

osob	int(11)	YES		NULL	

Oprócz pięciu podstawowych rekordów występują w niej też takie rekordy jak : `kierunek` i `spec` (specjalność) oba typu *varchar*. Poza danymi na temat kierunku i specjalności są też rekordy:`rok` i `osob` (ilość osób) w poszczególnych grupach. Oba zostały określone jako *integer*.

2.6 Rezerwacje

Tabela tRezerwacje została utworzona z myślą o rezerwacjach, tych cyklicznych, czyli rezerwacjach sal na cotygodniowe zajęcia, a także do rezerwacji sal na kolokwia oraz na sesje egzaminacyjne.

tRezerwacje

Field	Type	Null	Key	Default	Extra
cID	int(11)		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	
cCT	timestamp(14)	YES		NULL	
cOWNER	varchar(16)	YES		NULL	
cGROUP	varchar(16)	YES		NULL	
poczatek	datetime	YES		NULL	
koniec	datetime	YES		NULL	
nr_sali	varchar(8)	YES		NULL	
budynek	varchar(64)	YES		NULL	
kto	varchar(64)	YES		NULL	
po_co	varchar(128)	YES		NULL	

Tak jak każda tabela posiada 5 pięć standardowych rekordów: `cID, cTS`, `cOWNER,cGROUP`. Kolejne są niezbędne aby określić termin rezerwacji, miejsce, numer sali oraz osobę rezerwującą, a także cel rezerwacji.
Termin rezerwacji określony jest przez `początek` i `koniec`. Te rekordy są typu *timestamp*, a więc zawierają informacje nie tylko na temat dnia, miesiąca i roku ale również godziny rezerwacji. Pozostałe rekordy są typu tekstowego *varchar*. W nich zawarte są wszelkie szczegółowe informacje o dokonanej rezerwacji.

2.7 Sale

W tabeli tRezerwacje znajduje sie m.in. rekord nr_sali. Składający rezerwacje musi posiadać szczegółowe informacje na temat budynku, w którym znajduje się sala czy też pojemności sali. Takie właśnie informacje zgromadzone są w tabeli tSale.

tSale

Field	Type	Null	Key	Default	Extra
cID	int(11)		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	
cCT	timestamp(14)	YES		NULL	
cOWNER	varchar(16)	YES		NULL	
cGROUP	varchar(16)	YES		NULL	
nr_sali	varchar(8)	YES		NULL	
budynek	varchar(64)	YES		NULL	
pojemnosc	int(11)	YES		NULL	

Dane z tej tabeli wykorzystywane są nie tylko przy rezerwacjach sal na okres sesji ale przede wszystkim przy układaniu harmonogramów zajęć, w szczególności do dobierania odpowiednich sal do rodzaju zajęć oraz ilości studentów biorących w nich udział. Oprócz `cID ,cTS,cCT,cOWNER,cGROUP` w tej tabeli znajdują się trzy inne rekordy. Ponieważ zajęcia prowadzone są w kilku obiektach w tabeli znalazł się rekord nazwany `budynek` typu *varchar*, w którym zapisywane są dane dotyczące położenia budynku. W chwili obecnej zajęcia odbywają sie w trzech budynkach: na ul. Akademickiej, Sosnowej i Lipowej. Jeżeli znamy położenie budynku, następnym krokiem jest ustalenie numeru sali (nr_sali), w której zajęcia mają się odbywać. Ponieważ numery sal mogą być różnie określane np.: 23A, typ tego rekordu został więc określony jako *varchar*. Aby właściwie dobrać salę do rodzaju zajęć musimy wiedzieć ilu studentów może ona pomieścić. Do określenia tej wielkości, czyli pojemności sal utworzony został rekord `pojemnosc` typu *integer*.

Rozdział 3

Układanie planu

tRozkladDzi

Field	Type	Null	Key	Default	Extra
cID	int(10) unsigned		PRI	NULL	auto_increment
cTS	timestamp(14)	YES		NULL	
cCT	timestamp(14)	YES		NULL	
cOWNER	varchar(16)	YES		NULL	
cGROUP	varchar(16)	YES		NULL	
cKierunek	varchar(32)				
cSpec	varchar(255)				
cRok	char(3)				
cGrupa	varchar(8)	YES		NULL	
cSemestr	enum('zimowy','letni')			zimowy	
cPrzedmiot	varchar(128)	YES		NULL	
cRodzaj	varchar(32)	YES		NULL	
cProwadzacy	varchar(32)				
cDzien	varchar(12)	YES		NULL	
cPocz	time	YES		NULL	
cKoniec	time	YES		NULL	
cMiejsce	varchar(64)	YES		NULL	
cNote	varchar(128)	YES		NULL	

Tekst pracy. Tekst

pracy. Tekst pracy. Tekst

Spis literatury

[1] www.mysql.com/documentation/

[2] www.mhtml.com

[3] www.pckurier.pl/webmaster/1999/grudzien/felsztukier/mysql.html

[4] www.webton.pl/index.php?s=tmysql