

Projekt bazy danych

Przychodnia lekarska

Podsumowanie etapu II

1. Wybór i konfiguracja środowiska

Serwer - *MySQL Community Server 5.7*

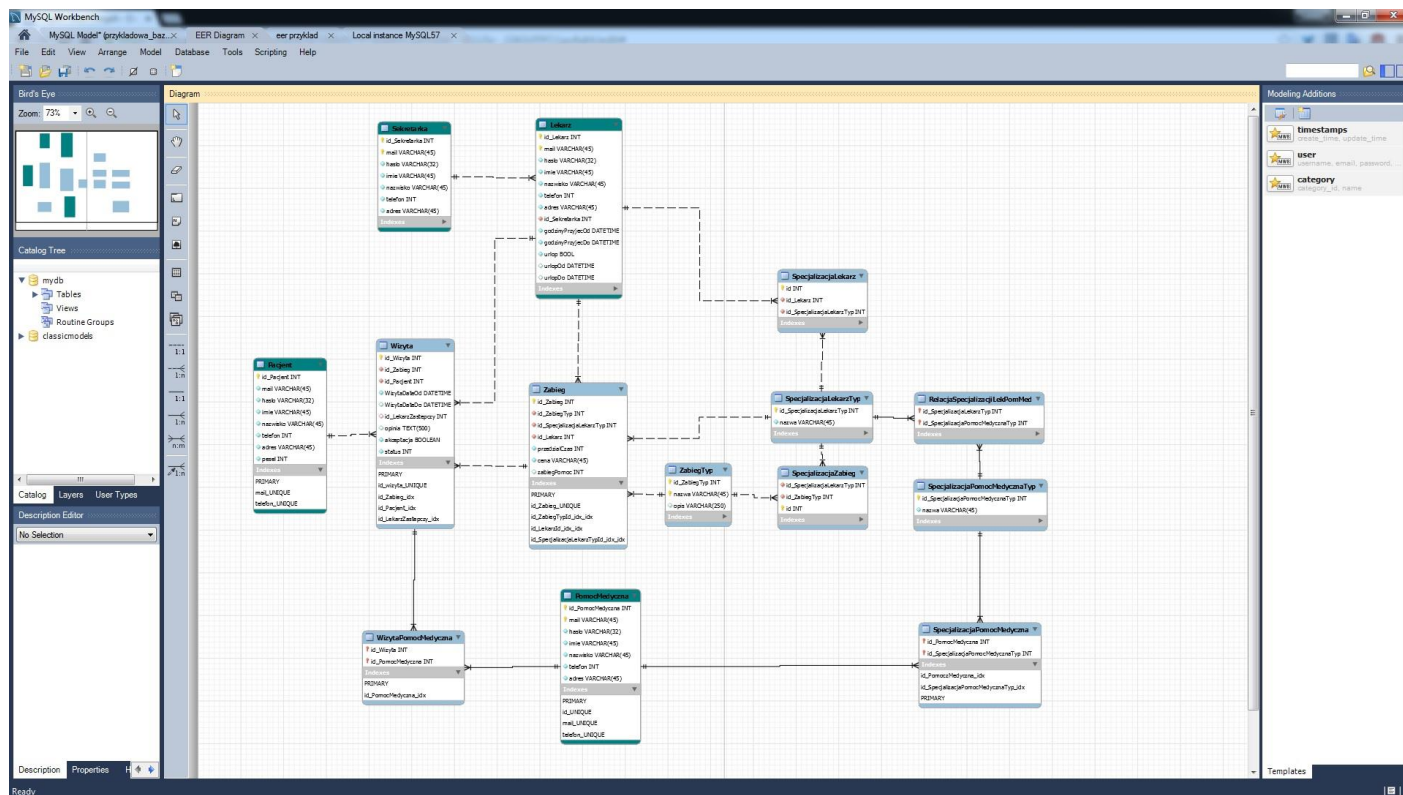
Jest to oprogramowanie “free software” dostępne na licencji GPL. Dostępne na większość popularnych platform systemowych oraz różnych architektur procesorów. M.in. Windows, Linux, MacOS X, Solaris. Głównymi zaletami MySQL w porównaniu z innymi serwerami bazodanowymi jest szybkość, niskie koszty użytkowania (open source), łatwa konfiguracja, wieloplatformowość i przenośność.

Narzędzie administracyjne - *MySQL Community Workbench 6.3*

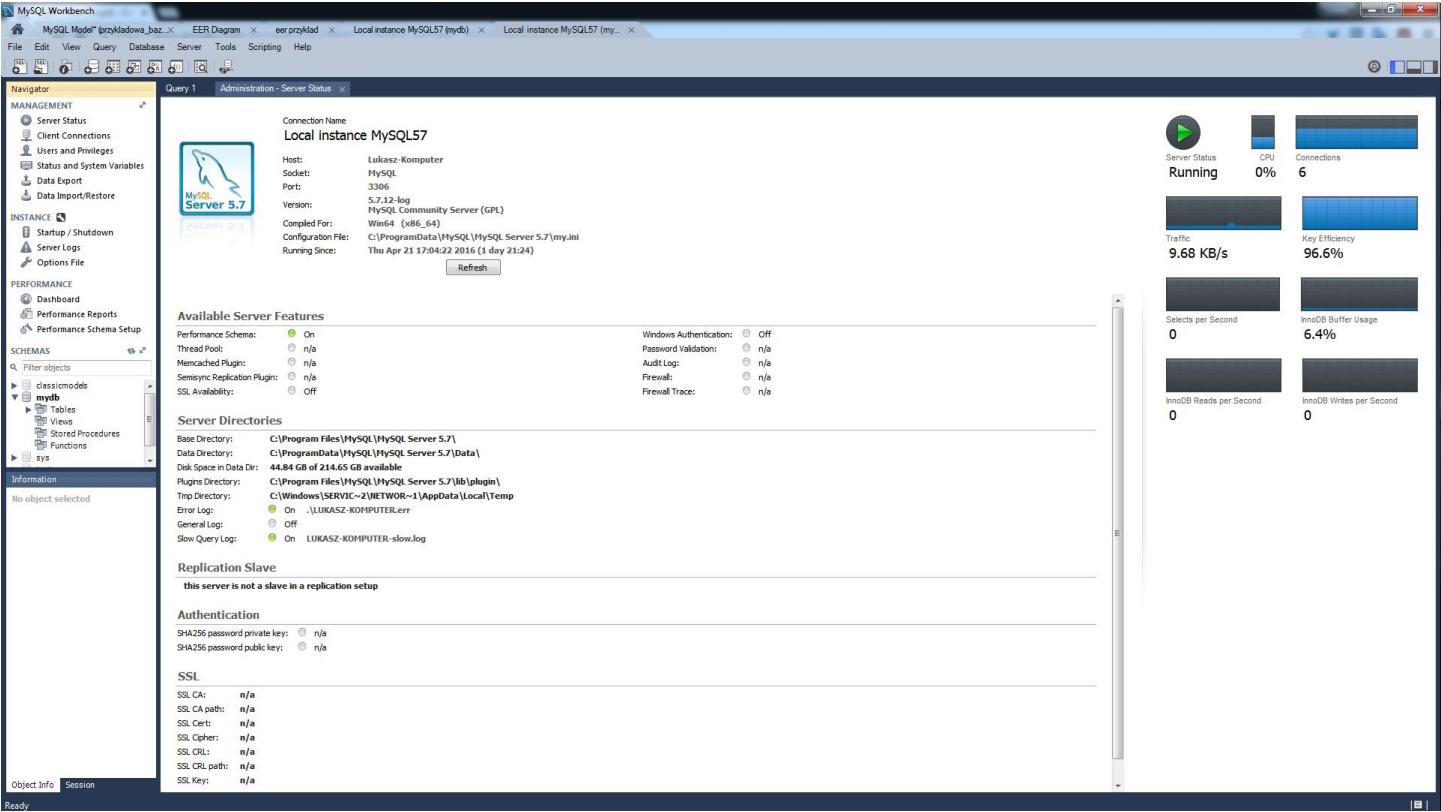
Narzędzie do zarządzania i modelowania baz danych MySQL. Umożliwia nam m.in. konfigurację serwera, projektowanie, tworzenie schematów baz danych. Umożliwia przeprowadzenie operacji reverse-engineering, generowanie skryptów SQL, przeglądanie danych. Dostarcza nam narzędzia wspomagające migracji i synchronizacji baz.

Przykładowe zrzuty ekranu z narzędzia administracyjnego:

1. Diagram EER przedstawiający schemat bazy danych



2. Narzędzie przedstawiające status serwera bazodanowego



3. Lista przedstawiająca wszystkie kolumny tabel wraz z ich właściwościami (typ danych, wartość domyślna, uprawnienia, nullable, typ kodowania znaków)

The screenshot shows the 'Columns' tab in MySQL Workbench for the 'lekarz' table. The table has 11 columns. The 'Count' is 70. The 'Output' panel shows the execution of a query that displays the table structure.

Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments
lekarz	id_lekarz	int(11)		NO			select,insert,update,references	auto_increment	
lekarz	mail	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
lekarz	haslo	varchar(32)		NO	utf8	utf8_general_ci	select,insert,update,references		
lekarz	imie	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
lekarz	nazwisko	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
lekarz	telefon	int(11)		NO			select,insert,update,references		
lekarz	adres	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
lekarz	id_Sekretarka	int(11)		NO			select,insert,update,references		
lekarz	godzinyPrzyjeciO	datetime		NO			select,insert,update,references		
lekarz	godzinyPrzyjeciD	datetime		NO			select,insert,update,references		
lekarz	urlop	tinyint(1)		NO			select,insert,update,references		
lekarz	urlopO	datetime		YES			select,insert,update,references		
lekarz	urlopD	datetime		YES			select,insert,update,references		
pacjent	id_pacjent	int(11)		NO			select,insert,update,references	auto_increment	
pacjent	mail	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pacjent	haslo	varchar(32)		NO	utf8	utf8_general_ci	select,insert,update,references		
pacjent	imie	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pacjent	nazwisko	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pacjent	telefon	int(11)		NO			select,insert,update,references		
pacjent	adres	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pacjent	pesel	int(11)		NO			select,insert,update,references		
pomoMedyczna	id_PomoMedyczna	int(11)		NO			select,insert,update,references	auto_increment	
pomoMedyczna	mail	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pomoMedyczna	haslo	varchar(32)		NO	utf8	utf8_general_ci	select,insert,update,references		
pomoMedyczna	imie	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pomoMedyczna	nazwisko	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
pomoMedyczna	telefon	int(11)		NO			select,insert,update,references		
pomoMedyczna	adres	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
relacjeSpecjalizacjaLekarz	id_SpecjalizacjaLekarz	int(11)		NO			select,insert,update,references		
relacjeSpecjalizacjaPomoc	id_SpecjalizacjaPomoc	int(11)		NO			select,insert,update,references		
sekretarka	id_Sekretarka	int(11)		NO			select,insert,update,references	auto_increment	
sekretarka	mail	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
sekretarka	haslo	varchar(32)		NO	utf8	utf8_general_ci	select,insert,update,references		
sekretarka	imie	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
sekretarka	nazwisko	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
sekretarka	telefon	int(11)		NO			select,insert,update,references		
sekretarka	adres	varchar(45)		NO	utf8	utf8_general_ci	select,insert,update,references		
specjalizacjaLekarz	id	int(11)		NO			select,insert,update,references	auto_increment	

2. Skrypt SQL tworzący bazę danych

Skrypt dostępny w pliku skrypt.sql

3. Przykładowe zapytania do bazy danych

Z założenia ma istnieć kilka sposobów tworzenia wizyty. Wyszukiwać można m.in. po nazwie specjalizacji lekarza.

rg = SpecjalizacjaLekarzTyp.nazwa

Pobranie listy wszystkich specjalizacji lekarzy

1	SELECT nazwa FROM mydb.SpecjalizacjaLekarzTyp;
---	--

Selectbox(**rg**.nazwa <nazwa specjalizacji>). nazwa → rg1 =

Lekarz.id_Lekarz, Lekarz.imie, Lekarz.nazwisko, SpecjalizacjaLekarz.id_SpecjalizacjaLekarzTyp

Po wybraniu konkretnej specjalizacji lekarza zwracamy wszystkich lekarzy z danej specjalizacji.

1	SELECT l.id_Lekarz, l.imie, l.nazwisko, sl.id_SpecjalizacjaLekarzTyp
2	FROM mydb.SpecjalizacjaLekarzTyp as st
3	INNER JOIN mydb.SpecjalizacjaLekarz as sl ON sl.id_SpecjalizacjaLekarzTyp=
4	st.id_SpecjalizacjaLekarzTyp
5	INNER JOIN mydb.Lekarz as l ON l.id_Lekarz = sl.id_Lekarz
6	WHERE st.nazwa='< <u>nazwa</u> >'

Selectbox(**rg1**.imie,**rg1**.nazwisko <imie,nazwisko lekarza>). id_Lekarz, id_SpecjalizacjaLekarzTyp →

rg2 = ZabiegTyp.*,id_Lekarz, id_SpecjalizacjaLekarzTyp, id_Zabieg wybranego lekarza z **rg1**

Następnie dla wybranego lekarza wyszukujemy zabiegi, które wykonuje.

1	SELECT zt.*, z.id_Lekarz, z.id_SpecjalizacjaLekarzTyp, z.id_Zabieg
2	FROM mydb.Zabieg as z
3	INNER JOIN mydb.ZabiegTyp as zt ON zt.id_ZabiegTyp=z.id_ZabiegTyp
4	WHERE z.id_Lekarz='< <u>id_Lekarz</u> >' and
5	z.id_SpecjalizacjaLekarzTyp='< <u>id_SpecjalizacjaLekarzTyp</u> >';

Selectbox(**rg2**.nazwa <nazwa zabiegu>). id_Zabieg → Tworzenie Wizyty

Posiadając już wszystkie potrzebne dane możemy przystąpić do tworzenia wizyty.

1	INSERT INTO mydb.Wizyta (id_Zabieg, id_Pacjent, WizytaDateOd, WizytaDateDo,
2	akceptacja,status)
3	VALUES ('< <u>id_Zabieg</u> >', '<id_Pacjent>', '<WizytaDateOd>', '<WizytaDateDo>', '0', '0');

Wartości WizytaDateOd, WizytaDateDo są obliczane w skrypcie na podstawie wolnego bloku czasowego dla danego zabiegu u wybranego lekarza.

Lekarze mają możliwość akceptacji wizyty. Proces ten w bazie danych polega na ustawieniu kolumny akceptacja w wierszu danej wizyty na wartość '1'.

1	UPDATE mydb.Wizyta SET akceptacja='1' WHERE id_Wizyta='<id_Wizyta>';
---	--

Każdy użytkownik posiada możliwość usunięcia konta. Operację tą wykonujemy poprzez wywołanie polecenia.

1	DELETE FROM mydb.Pacjent WHERE id_Pacjent='<twoje_id>';
---	---

4. Złożone zapytania SELECT

Zapytanie zwracające imię, nazwisko lekarza oraz ilość wizyt odbytych w tym miesiącu wraz z ilością zarobionych pieniędzy.

1	SELECT l.imie,l.nazwisko,w.status,COUNT(z.id_Lekarz) as IloscWizyt, SUM(z.cena) as Zarob
2	FROM mydb.Wizyta as w
3	INNER JOIN mydb.Zabieg as z
4	ON z.id_Zabieg = w.id_Zabieg
5	INNER JOIN mydb.Lekarz as l
6	ON l.id_Lekarz = z.id_Lekarz
7	WHERE w.akceptacja='1' and status IN('1','2') and w.WizytaDateDo
8	BETWEEN '<od>' and '<do>'
9	GROUP BY z.id_Lekarz, w.status
10	ORDER BY Zarob DESC

akceptacja = '1' - oznacza pozytywne zatwierdzenie wizyty,

status = '1' - wizyta się odbyła i nie została opłacona,

status = '2' - wizyta się odbyła i została opłacona.

