

# MySQL Manual

## Instalacija MySQL (Windows OS)

Ukoliko je potrebno da imamo MySQL Server na Windows mašini, sve potrebne programe za rad možemo da instaliramo preko MySQL Installer – a (link: <https://dev.mysql.com/downloads/installer/> ). Ovo je paket sa svim potrebnim programima. U toku instalacije nudi se više opcija za *Setup Type*, ali opcija *Developer Default* uključuje sve potrebne programe za razvoj aplikacije. Klikom na *Execute* instalacija se obavlja i nakon toga sledi konfigurisanje MySQL servera. Za Config Type, za ratvoj, potrebno je izabrati *Development Computer* opciju. Ovde možemo da promenimo i mrežne postavke. Podrazumevani port je 3306. Sledeće podešavanje je vezano za logovanje. Ukoliko aplikacija ne može da se ažurira da koristi MySQL8, onda možemo da izaberemo kompatibilnu autentikaciju za verziju 5, a inače biramo opciju sa jakom enkripcijom (preporučeno). Sledeći prozor nam otvara postavljanje šifre za logovanje (*MySQL Root Password*) i prozor nakon toga možemo da ostavimo sa podrazumevanim parametrima (eventualno možemo da promenimo ime servera ukoliko je to potrebno). Kada potvrdimo konfiguracione korake (potvrdom na *Execute*) proces je završen. Nakon ovoga sledi faza kreiranja i administracije baze podataka. Popularan alat za ove svrhe za MySQL bazu je MySQL Workbench.

## MySQL i Docker

Umesto instaliranja servera, možemo da koristimo i MySQL preko Docker-a. Ovo zahteva da već imamo instaliran Docker na našem sistemu. Povlačimo novu MySQL sliku:

```
docker pull mysql/mysql-server
```

Na ovaj način povlačimo najnoviju sliku. Ukoliko hoćemo neku drugu verziju onda moramo da naznačimo verziju tako što ćemo da dodamo na kraj komande:

```
docker pull mysql/mysql-server:8.0.28
```

Da bismo bili sigurni da je sve u redu, pokrenemo komandu `docker images` i izlaz treba da bude sličan kao na slici 1.

```
medium$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
mysql/mysql-server  latest      434c35b82b08  2 months ago  417MB
medium$
```

Slika 1. Lista slika

Nakon ovoga, pokrećemo instancu. Komanda se sastoji iz više delova:

- `docker run`: komanda za pokretanje docker kontejnera iz slike
- `--name='my_sql_container'`: nakon argumenta, definišemo ime kontejnera. Možemo i da izostavimo, i onda će se generisati automatski.
- `-d`: argument koji obezbeđuje da se nakon pokretanja komande terminal oslobađa, da neće biti zakačen za kontejner (detached mode)
- `-p 3306:3306`: mapiranje porta iz docker kontejnera na host port. MySQL je mapiran na 3306, ali ukoliko je nama potrebno da ga koristimo na drugom portu (3306 je zauzet ili nam iz nekog drugog razloga ne odgovara), ovde bismo to promenili.
- `mysql/mysql-server`: ime slike

Tako da je kompletna komanda:

```
docker run --name='my_sql_container' -d -p 3306:3306 mysql/mysql-server
```

```
medium$ docker run --name='my_sql_container' -d -p 3306:3306 mysql/mysql-server
ac7286f937b257d0eef422c72480233c403d964f0379be6ddc0c4ad50dbf70c2
```

Takođe, i komandom `docker ps` možemo da vidimo koji su kontejneri pokrenuti.

```
docker logs my_sql_container
```

```
[root@mysql ~]# docker image 8.0.28-1.2.7-server
[Info] No password option specified for new database.
[Info] A random online password will be generated.
[Info] Initializing database
[Info] [root@mysql ~]# docker logs -f mysql01 (Server) /usr/bin/mysqld (mysql 8.0.28) initializing server in progress as process 17
2022-08-16T07:22:46.408853Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld initialization has started.
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld initialization has ended.
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld is ready to use, it is created with an empty password ! Please consider switching off the --initialize-insecure option.
[Info] Database Database initialized
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld (mysql 8.0.28) starting as process 64
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld initialization has started.
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld initialization has ended.
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) CA certificate ca.pem is not signed.
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2022-08-16T07:22:46.432362Z I [System] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld (mysql 8.0.28) ready for connections. Version: '8.0.28' socket: '/var/lib/mysql/mysql.sock' port: 3306 MySQL Community Server - GPL.
Warning: Unable to load /usr/share/zoneinfo/iso3166.tab as time zone. Skipping it.
Warning: Unable to load /usr/share/zoneinfo/leapseconds as time zone. Skipping it.
Warning: Unable to load /usr/share/zoneinfo/extended.gmt as time zone. Skipping it.
Warning: Unable to load /usr/share/zoneinfo/zone.tab as time zone. Skipping it.
Warning: Unable to load /usr/share/zoneinfo/zone1978.tab as time zone. Skipping it.
[Info] Initializing mysqld
[Info] [root@mysql ~]# docker logs -f mysql01 (Server) mysqld (mysql 8.0.28) ready for connections. Version: '8.0.28' socket: '/var/lib/mysql/mysql.sock' port: 3306 MySQL Community Server - GPL.
```

Ovo nije dovoljno da bi sve radilo, zato što moramo da damo dozvole korisniku. To ćemo da uradimo tako što ćemo da izvršimo:

-it (interaktivni mod), *bash* (hoćemo da pokrenemo bash)

Sada možemo da vidimo listu korisnika:

```
use mysql;  
select user from user;
```

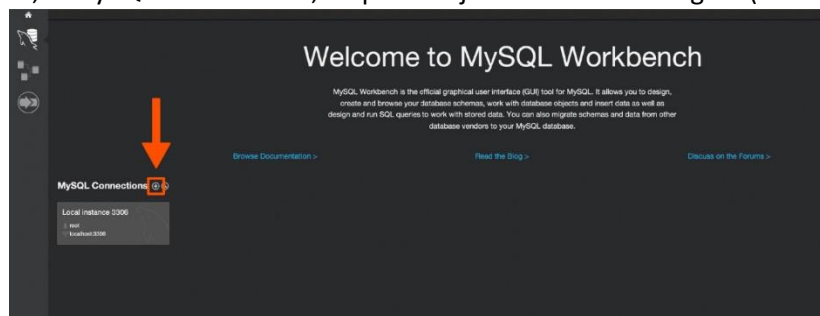
Konačno, pravimo korisnika za konekciju. Neka se zove dbeaver sa istom šifrom:

```
CREATE USER 'dbeaver'@'%' IDENTIFIED BY 'dbeaver';
GRANT ALL PRIVILEGES ON *.* TO 'dbeaver'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Nakon ovoga, možemo da se konektujemo na bazu preko klijenta (npr Workbench). Podešavanje za konekciju (slika 5) bi u ovom slučaju bila – Host: localhost, Username: dbeaver, Password: dbeaver i port ostaje 3306.

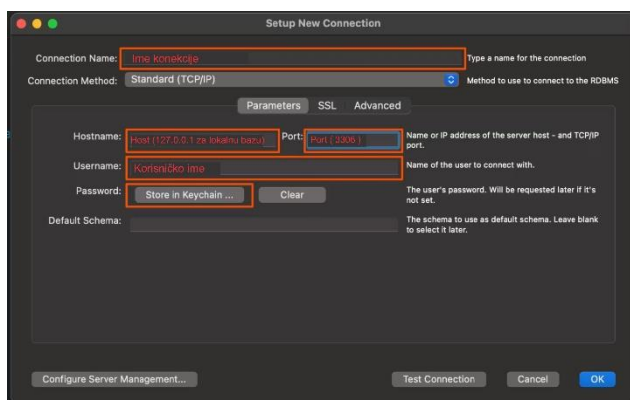
## Pristup MySQL bazi iz Workbench-a

MySQL Workbench je softver za rad sa sistemima za razvoj relacionih baza podataka. Da bismo se konektovali na bazu, u MySQL Workbenchu, na početnoj strani bitamo + dugme (slika 4).



Slika 4. Iniciranje konekcije

Otvora se pop-up prozor i tu upisujemo kako želimo da se zove na konekcija (*Connection Name*), zatim *Hostname*, *Port*, *Username* i *Password* (slika 5). Nakon potvrde na *Test Connection*, ukoliko je sve u redu, kreirali smo novu konekciju. Nakon ovoga možemo da koristimo bazu.



Slika 5. Parametri nove konekcije

## MySQL i Entity Framework – Povezivanje MySQL baze i .NET Core

Pretpostavimo da već imamo .NET Core Web App projekat kreiran. Da bismo povezali bazu, potrebno je da instaliramo nekoliko paketa preko NuGet:

- Microsoft.EntityFrameworkCore
- Microsoft.EntityFrameworkCore.Tools
- Pomelo.EntityFrameworkCore.MySql

Konkretno verzije trebalo bi da budu kompatibilne sa verzijom .NET-a. Na primer ako koristimo .NET 6, onda bi i Microsoft.EntityFrameworkCore trebalo da bude verzije 6.

Ako koristimo Visual Studio onda možemo da dodjemo do instalacije preko: projekat u VS -> Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution... i onda pretražimo pakete. Ako koristimo Visual Studio Code, onda je dobro da imamo NuGet Gallery ekstenziju instaliranu, pa preko komande otvaramo nuget galeriju i pretražujemo pakete.

Kreiramo klasu modela. Model može da ima razna polja, kao i anotacije da li je potrebna validacija za neko polje (često korišćena anotacija [Required]).

```
using System.ComponentModel.DataAnnotations;
namespace API.Entities;
public class AppUser
{
    [Key]
    public int Id { get; set; }
    public string UserName { get; set; }
}
```

U novom folderu Data u kreiramo i novu klasu *DataContext.cs* koja nasleđuje *DbContext* klasu. *DbContext* klasa je veza za interakciju entitetskih klasa i baze podataka. *DbContext* sadrži svojstva za pristup *DbSet*-ovima za sve entitetske klase koje su mapirane u tabele baze. *DbSet<TEntity>* predstavlja kolekciju entitetskih objekata koji odgovaraju redovima u bazi podataka.

```
using API.Entities;
using Microsoft.EntityFrameworkCore;
namespace API.Data;
public class DataContext : DbContext
{
    public DataContext(DbContextOptions options) : base(options)
    { }
    public DbSet<AppUser> Users { get; set; }
}
```

Dalje, pravimo string za konekciju koji predstavlja sve kredencijale za konektovanje sa bazom. Možemo da definišemo i unutar klase, međutim pošto je to osetljiv podatak, bolje je da držimo ovkve podatke u *appsettings.json* fajlu.

```
"ConnectionStrings": {
  "DefaultConnection": "Server=localhost;Database=name_of_db;User=username;Password=password;"
}
```

Zatim podešavamo aplikaciju da koristi MySQL bazu. U *Program.cs* fajlu dodajemo:

```
var connectionString =
builder.Configuration.GetConnectionString("DefaultConnection");
builder.Services.AddDbContext<DataContext>(opt =>
{
    opt.UseMySQL(connectionString, ServerVersion.AutoDetect(connectionString));
});
```

Nakon ovoga, radimo jos migraciju i ažuriranje baze. Inače ovo je potrebno da uradimo i nakon svake promene modela (*InitialCreate* je na primer za prvo kreiranje, kako menjamo model naziv treba da sugeriše šta je promenjeno):

```
dotnet ef migrations add InitialCreate
```

```
dotnet ef database update
```