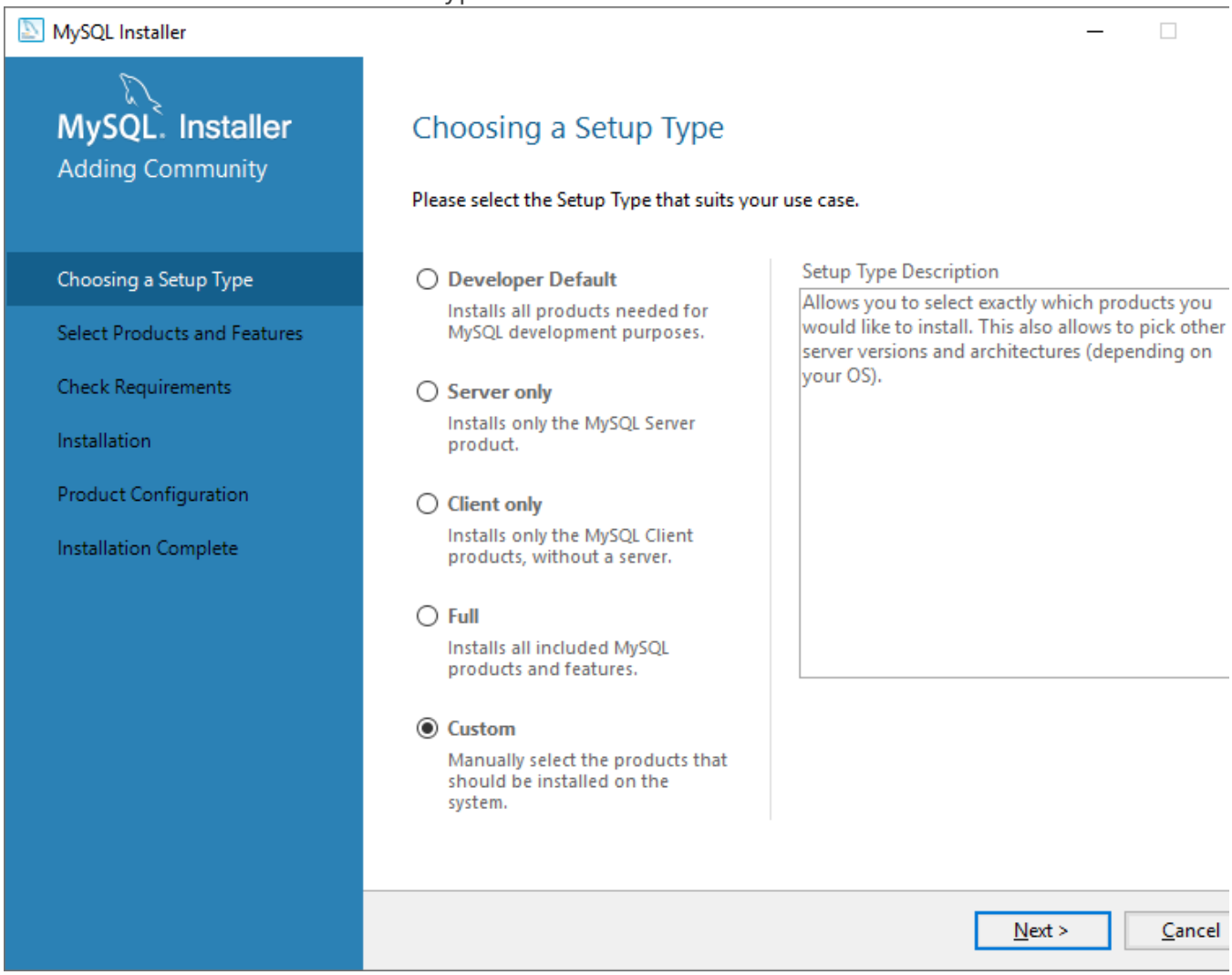
Installation of the Database to Save Enterprise Architect Model

If a database engine is installed on the computer or available on the network, it is possible to skip this chapter.

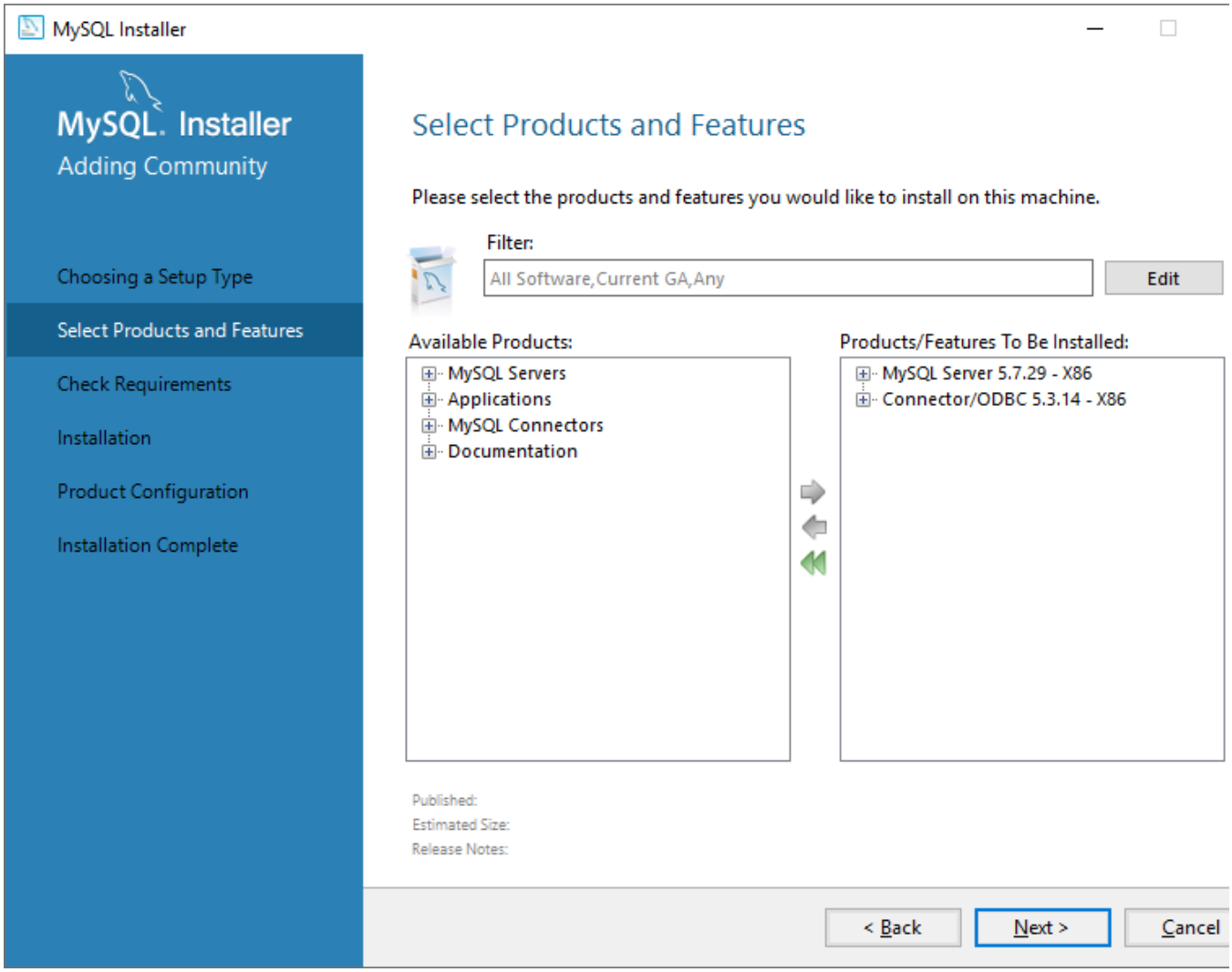
Here you can learn how to install one of the supported database engines.
The one we're using is **MySql**, which is recommended as ideal for the needs of both the Enterprise Architect program and EaInfoport

1. Before installing MySQL, it is necessary to ensure that the **vcredist_x86.exe** helper package in version 2013 is installed on the computer. Search the internet for "Visual C++ Redistributable Packages for Visual Studio 2013", save the vcredist_x86.exe variant to the disk and run it.
   After approval of the licence, the package will be installed. There is no need to restart the computer. Delete the vcredist_x86.exe file
2. Download the 32-bit web installer for the 5.7.x series from
   https://downloads.mysql.com/archives/installer/. So currently *mysql-installer-web-community-5.7.28.0.msi*
   Save the file to the local disk (for example, to the C:\EAInfoport folder)
3. Run the file

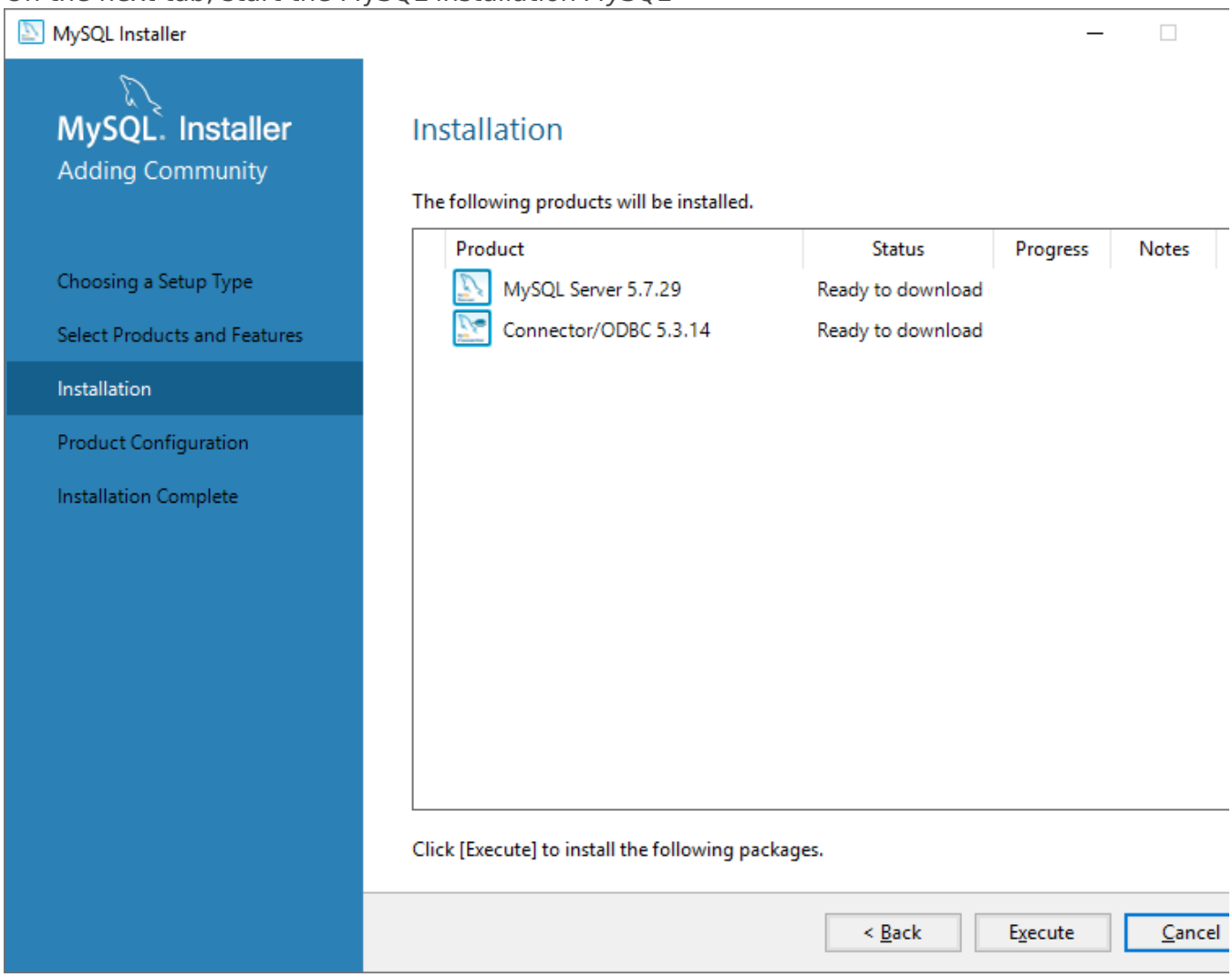
4. Select *Custom* as the installation type



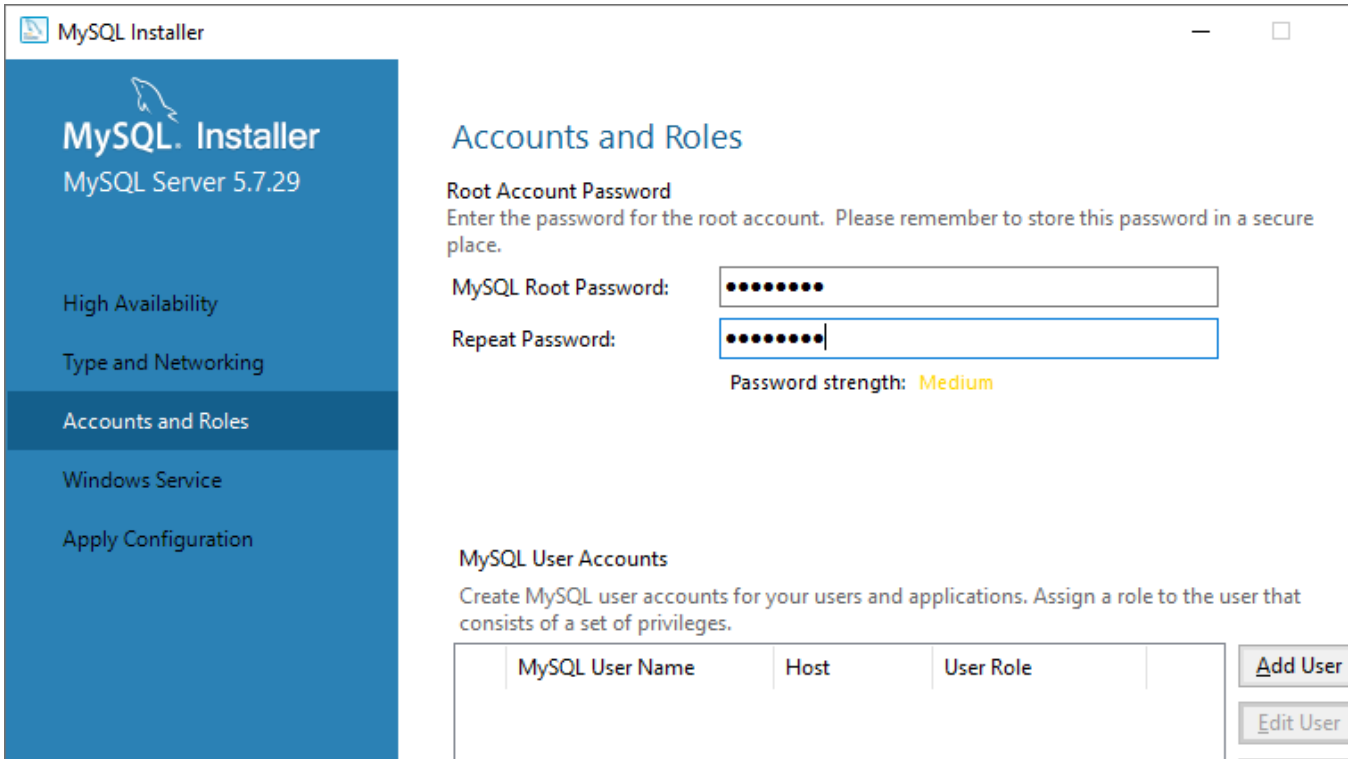
5. Select two items to install:
   MySQL Server 5.7.x – X86
   Connector/ODBC 5.3.x – X86

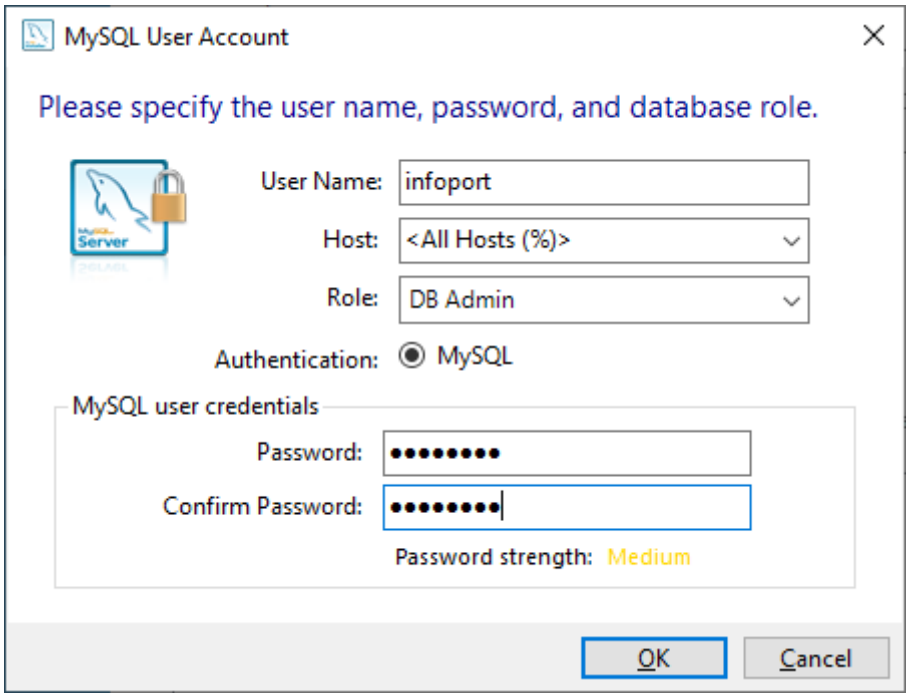


6. On the next tab, start the MySQL installation *MySQL*



7. Leave the settings on the other tabs as default.
   Only on the tab with the settings of accounts and roles, first fill in the root password (*for the database administrator*) and then create a user infoport, under which you will log into the database later.
   All passwords (*mainly the administrator ones*) should be secure. For demonstration purposes, we use the phrase P@ssw0rd as a password in the manual. (*It will be seen later in the connection definition*)





8. Delete the *mysql-installer-web-community-5.7.28.0.msi* installation file from the disk. This completes the installation of the database and the ODBC driver (*but it will still need to be configured – see below*)

Specific Setting for PostgreSQL

Problem

The application uses Entity Framework Core, which, when connecting to PostgreSQL, expects the system migrations table `__EFMigrationsHistory` to be created in the schema that is set as the first item in `search_path`.

If the user has `search_path` in PostgreSQL set to a different schema (e.g. `public`), but the application uses its own schema (e.g. `infoport`), EF Core creates or looks for the `__EFMigrationsHistory` table in the wrong schema. As a result, the application:

- does not find existing migrations,
- tries to create the tables again,
- or fails with an error on startup.

Solution – setting search_path for the user in PostgreSQL

Please set the correct search_path for the user under which the application logs into the database.

1. Set search_path for the user

```
ALTER USER <user> SET search_path = infoport, public;
```

After this setting, PostgreSQL will automatically use the **infoport** schema as the default on every login, which is necessary for the application to run correctly.

2. Verification

After logging in as the given user:

```
SHOW search_path;
```

Expected output:

Code

```
infoport, public
```

Alternative – setting in the connection string

If it is not possible to change the user's setting, search_path can also be set in the connection string:

```
Search Path=infoport
```

or

```
Search_path=infoport
```

Important notice

If search_path is specified in the connection string, it **overrides** the user setting in PostgreSQL. Therefore, only **one** of the above options should be used.

Summary

The Infoport application requires that the user used by the application has set:

Code

```
search_path = infoport, public
```

Without this setting, PostgreSQL stores the system migrations table in a different schema, which causes the application to malfunction.

Revision #5
Created 7 April 2022 10:25:52
Updated 14 August 2026 08:44:27 by Jitka Kalíšová