

Laurea in Ingegneria dei sistemi logistici

Insegnamento di Sistemi di Elaborazione I

Esercitazione: Il DBMS MySQL

Indice della lezione

1. MySql

1. Download e installazione
2. Il client mysql
3. MySQL Workbench
4. Importare Script

2. Gestione di una base di dati

1. Creazione e gestione di una base di dati
2. Creazione e gestione delle tabelle
3. Inserimento dati
4. Interrogazione

3. Esercitazione su interrogazioni

Il DBMS MySQL

- MySQL è un DBMS gratuito ed open-source.
(<http://www.mysql.com>)
- MySQL può essere utilizzato in Windows, Linux, MacOS X.

Download e Installazione di MySQL

Per l'installazione procedere come segue:

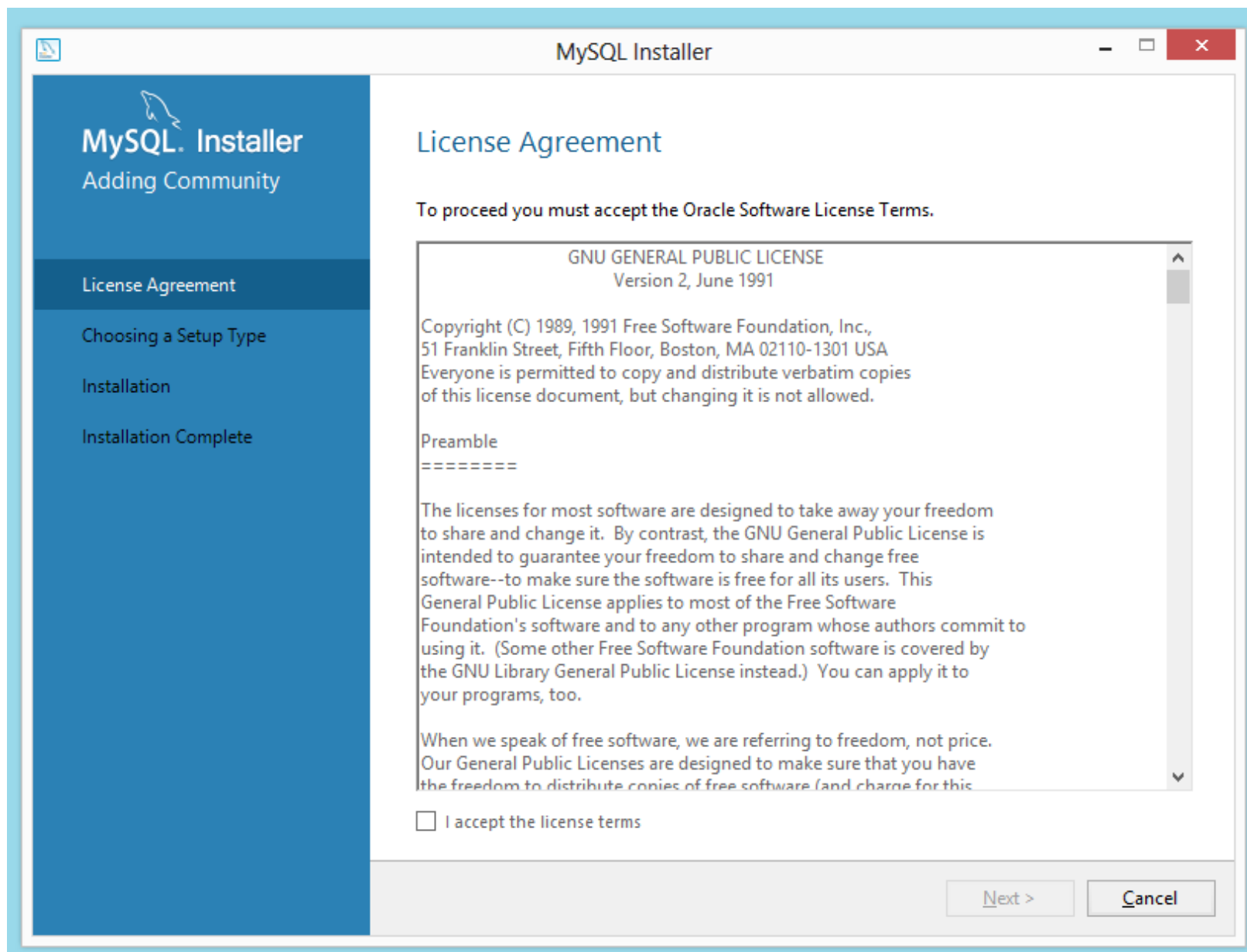
1. Scaricare l'ultima versione di MySQL Community Server (In questa esercitazione si farà riferimento alla versione MySQL 5) all'indirizzo:

<http://dev.mysql.com/downloads/mysql/>

Solo per utenti Windows direttamente l'installer:

<http://dev.mysql.com/downloads/installer/>

2. La procedura di installazione cambia da sistema a sistema.
3. Quello di cui si ha bisogno è:
 1. MySQL Server
 2. MySQL Client
4. L'installazione **developer** è quella che ha i prodotti pensati per i sviluppatori
5. Dopo l'installazione, MySQL ha bisogno di essere configurato
6. Anche questo cambia da sistema a sistema.
7. Tenete a mente che:
 1. In questa fase è possibile assegnare una password all'utente root
 2. È possibile specificare se MySQL server deve essere lanciato all'avvio



MySQL. Installer

Adding Community

License Agreement

Choosing a Setup Type

Installation

Installation Complete

MySQL Installer

License Agreement

To proceed you must accept the Oracle Software License Terms.

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble
=====

The licenses for most software are designed to take away your freedom
to share and change it. By contrast, the GNU General Public License is
intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have
the freedom to distribute copies of free software (and charge for this

I accept the license terms

Next >

Cancel



MySQL Installer



 **MySQL. Installer**
Adding Community

License Agreement

Choosing a Setup Type

Installation

Installation Complete

License Agreement

To proceed you must accept the Oracle Software License Terms.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble
=====

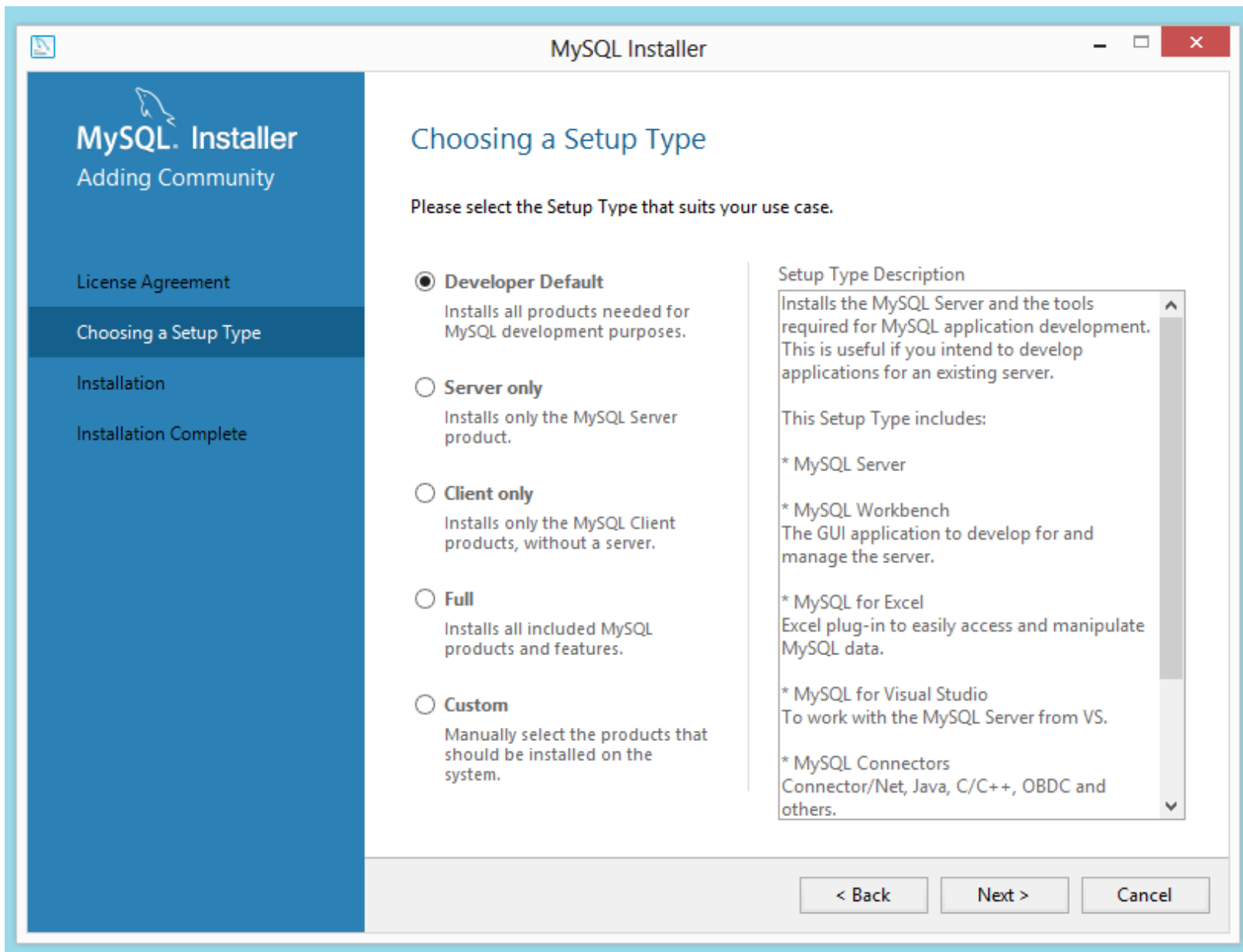
The licenses for most software are designed to take away your freedom
to share and change it. By contrast, the GNU General Public License is
intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have
the freedom to distribute copies of free software (and charge for this

I accept the license terms

Next >

Cancel



MySQL. Installer

Adding Community

License Agreement

Choosing a Setup Type

Installation

Installation Complete

Choosing a Setup Type

Please select the Setup Type that suits your use case.

- Developer Default**
Installs all products needed for MySQL development purposes.
- Server only**
Installs only the MySQL Server product.
- Client only**
Installs only the MySQL Client products, without a server.
- Full**
Installs all included MySQL products and features.
- Custom**
Manually select the products that should be installed on the system.

Setup Type Description

Installs the MySQL Server and the tools required for MySQL application development. This is useful if you intend to develop applications for an existing server.

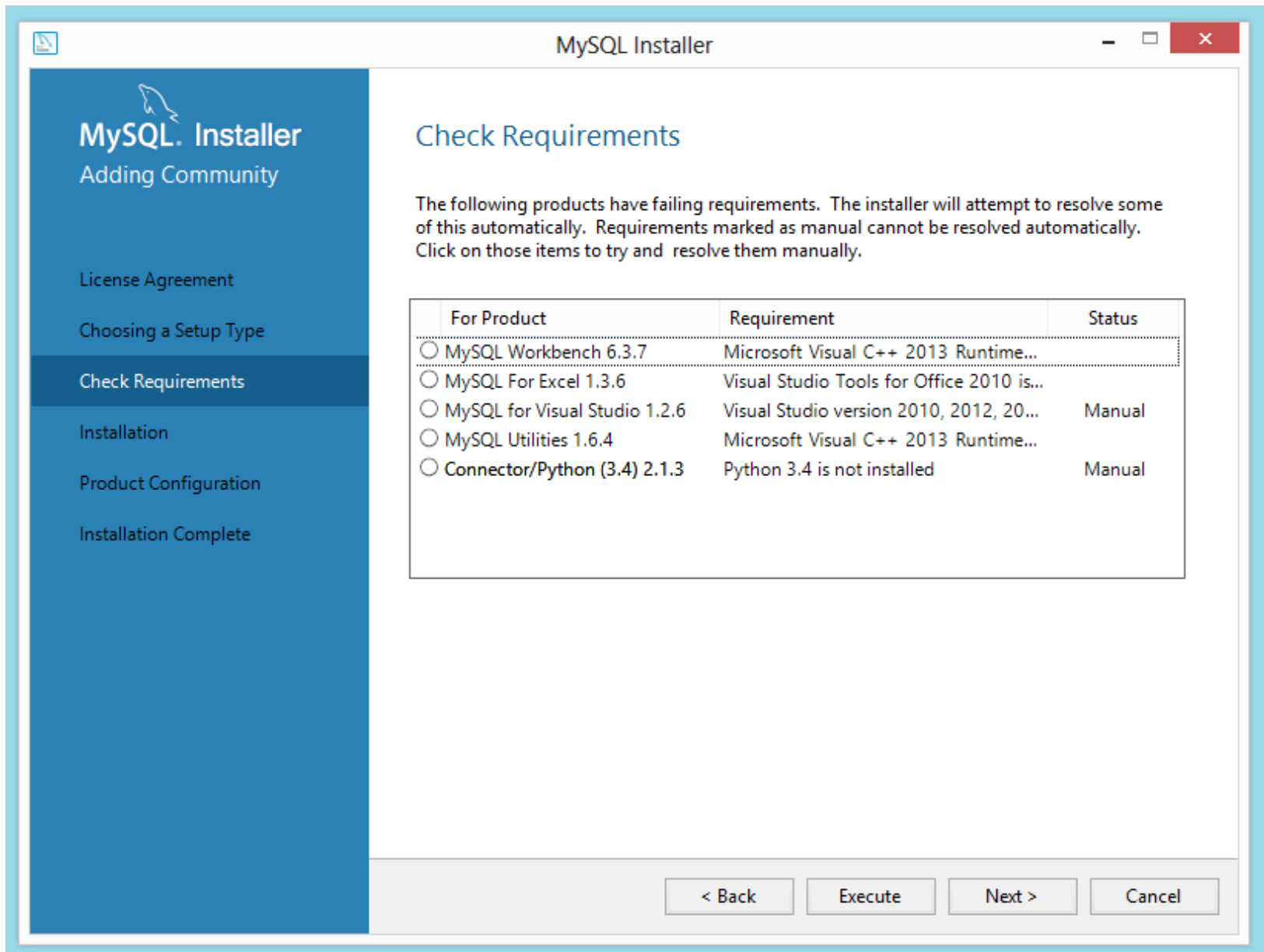
This Setup Type includes:

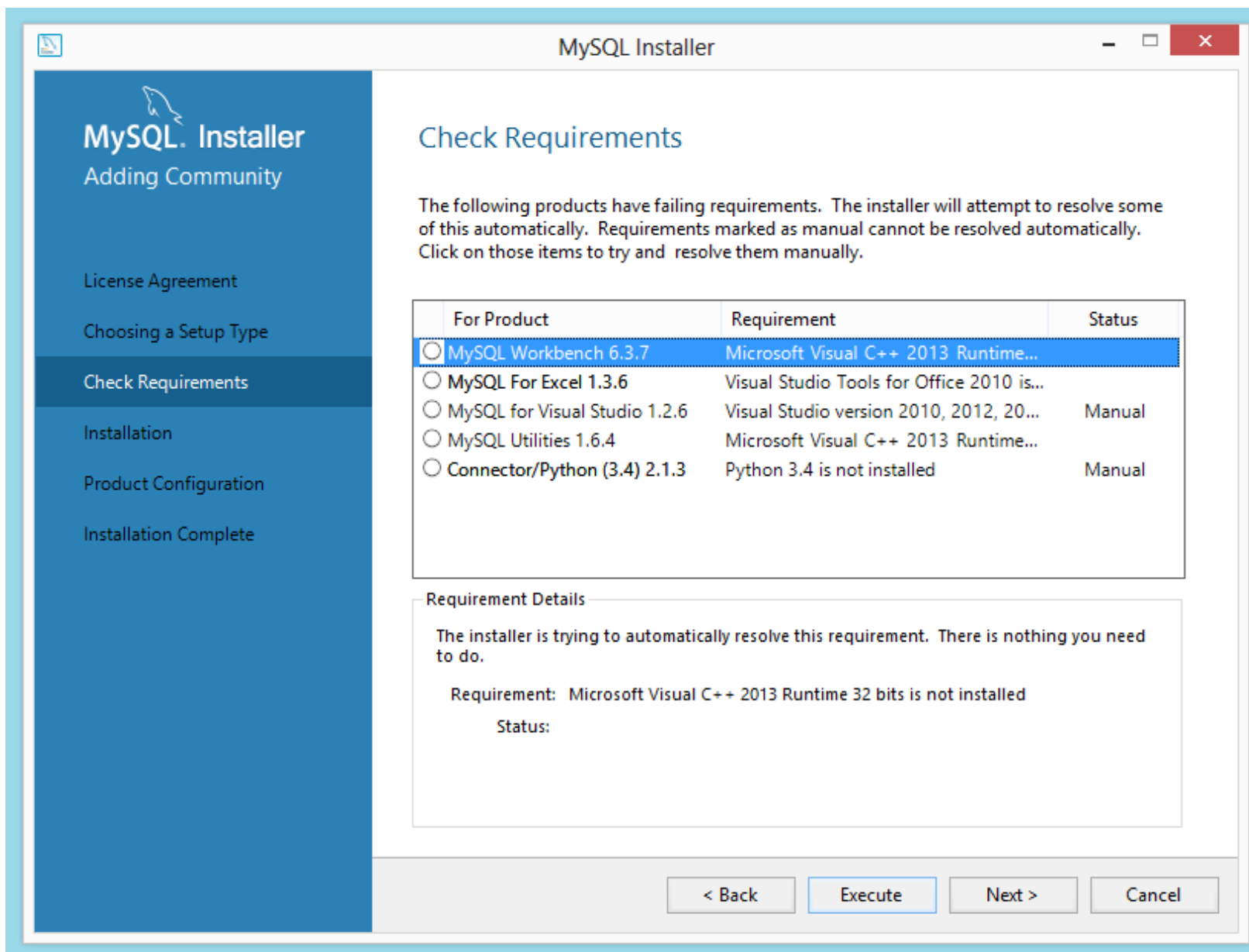
- * MySQL Server
- * MySQL Workbench
The GUI application to develop for and manage the server.
- * MySQL for Excel
Excel plug-in to easily access and manipulate MySQL data.
- * MySQL for Visual Studio
To work with the MySQL Server from VS.
- * MySQL Connectors
Connector/Net, Java, C/C++, ODBC and others.

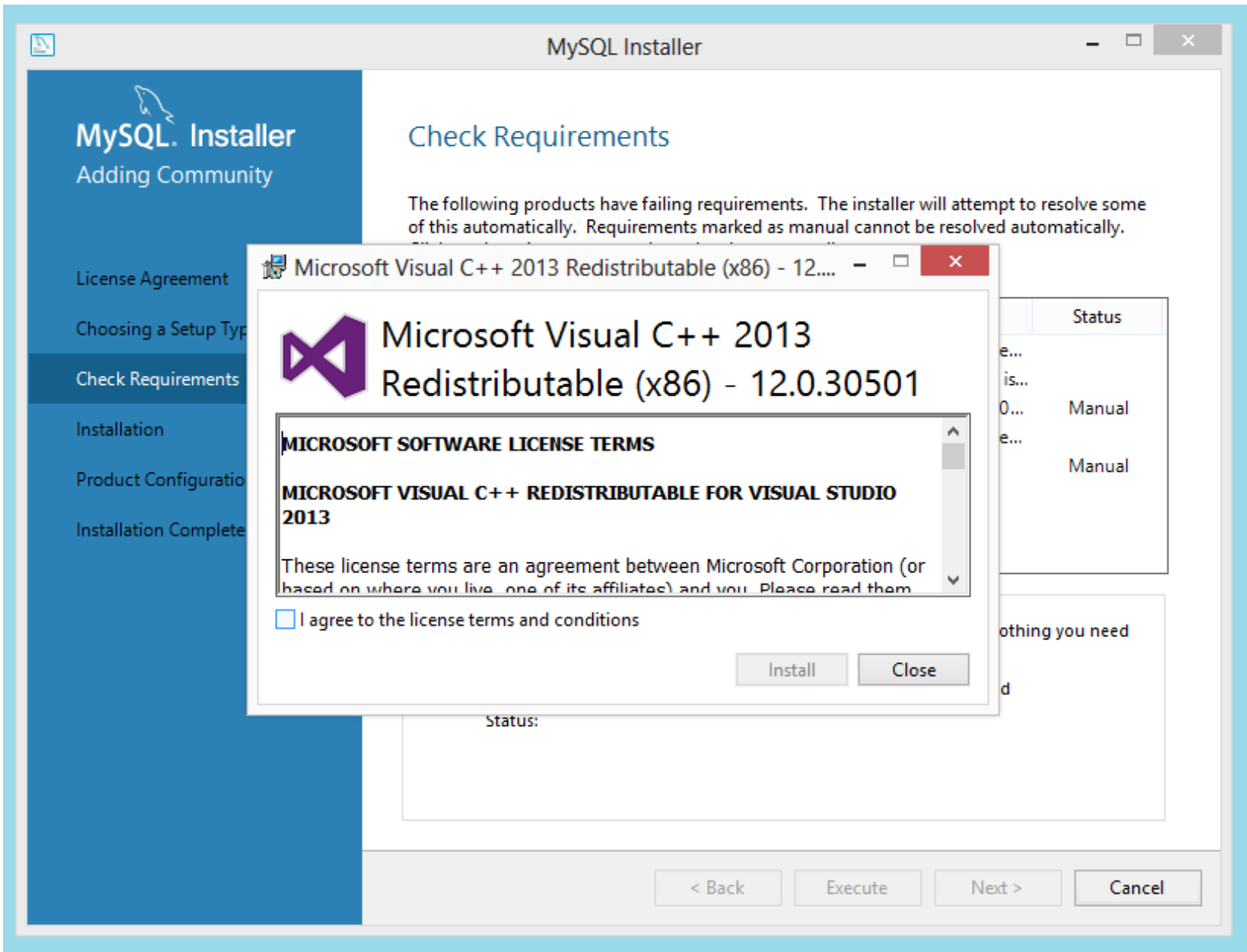
< Back

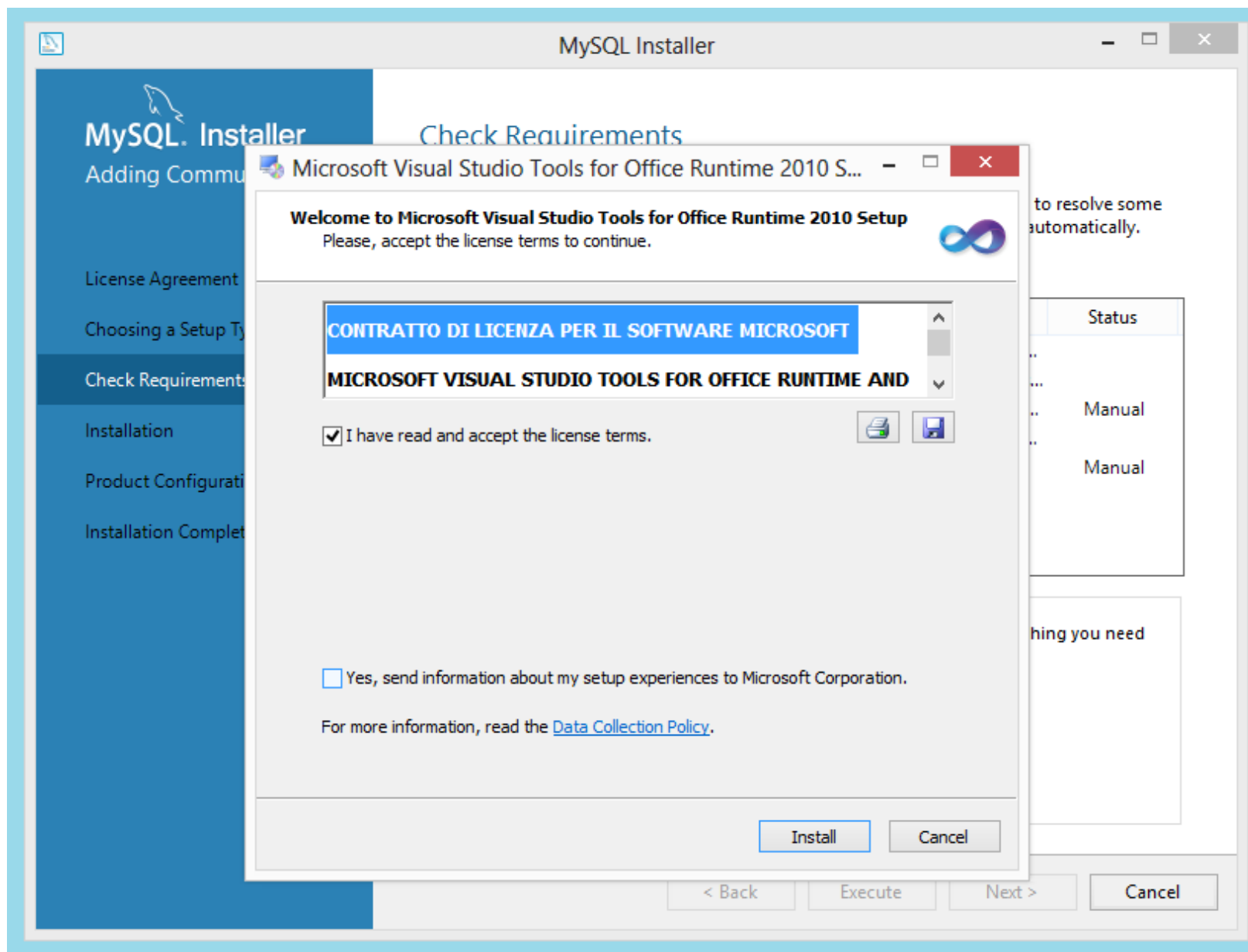
Next >

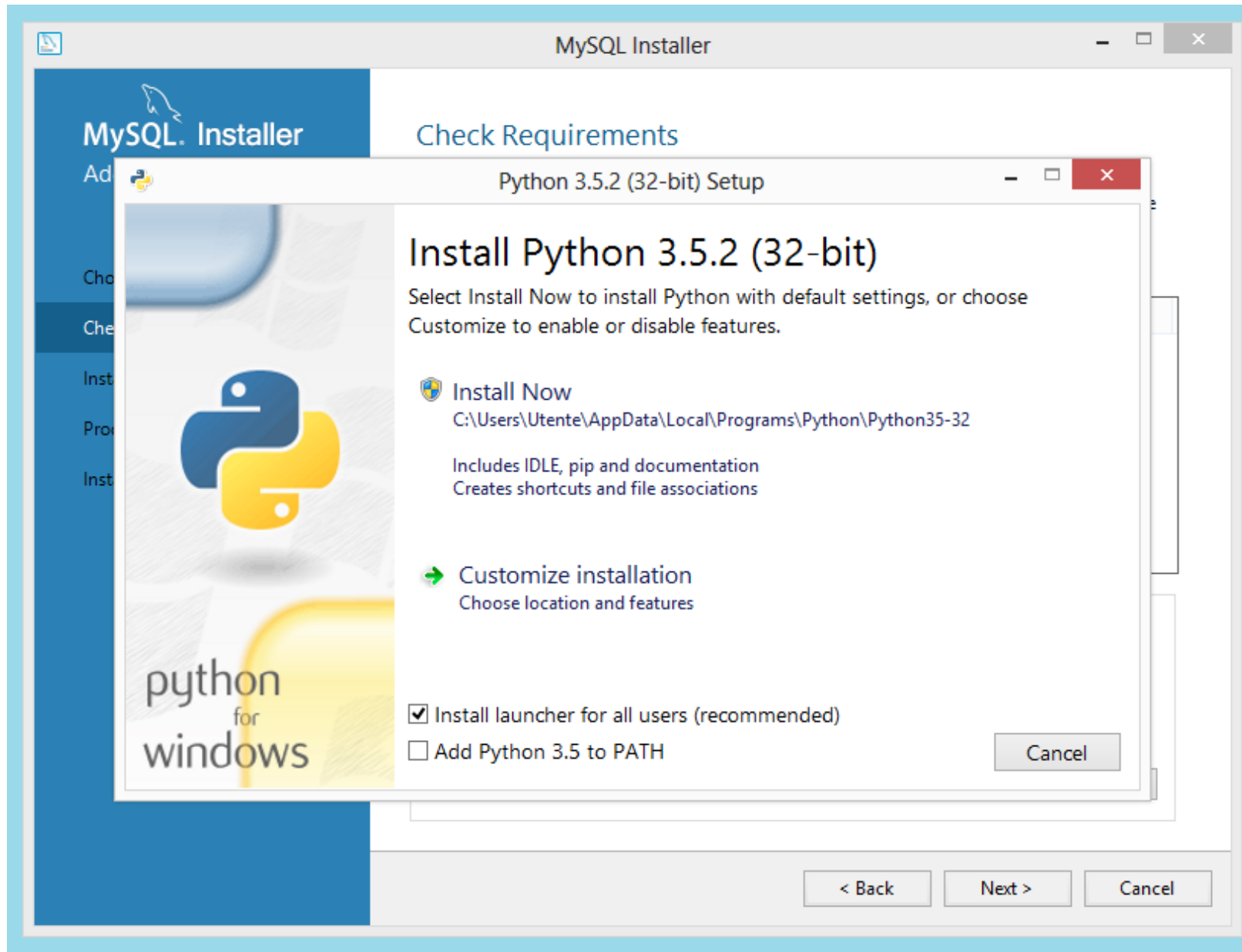
Cancel













MySQL Installer

MySQL. Installer
Adding Community

Choosing a Setup Type














Installation

Product Configuration

Installation Complete

Installation

Press Execute to upgrade the following products.

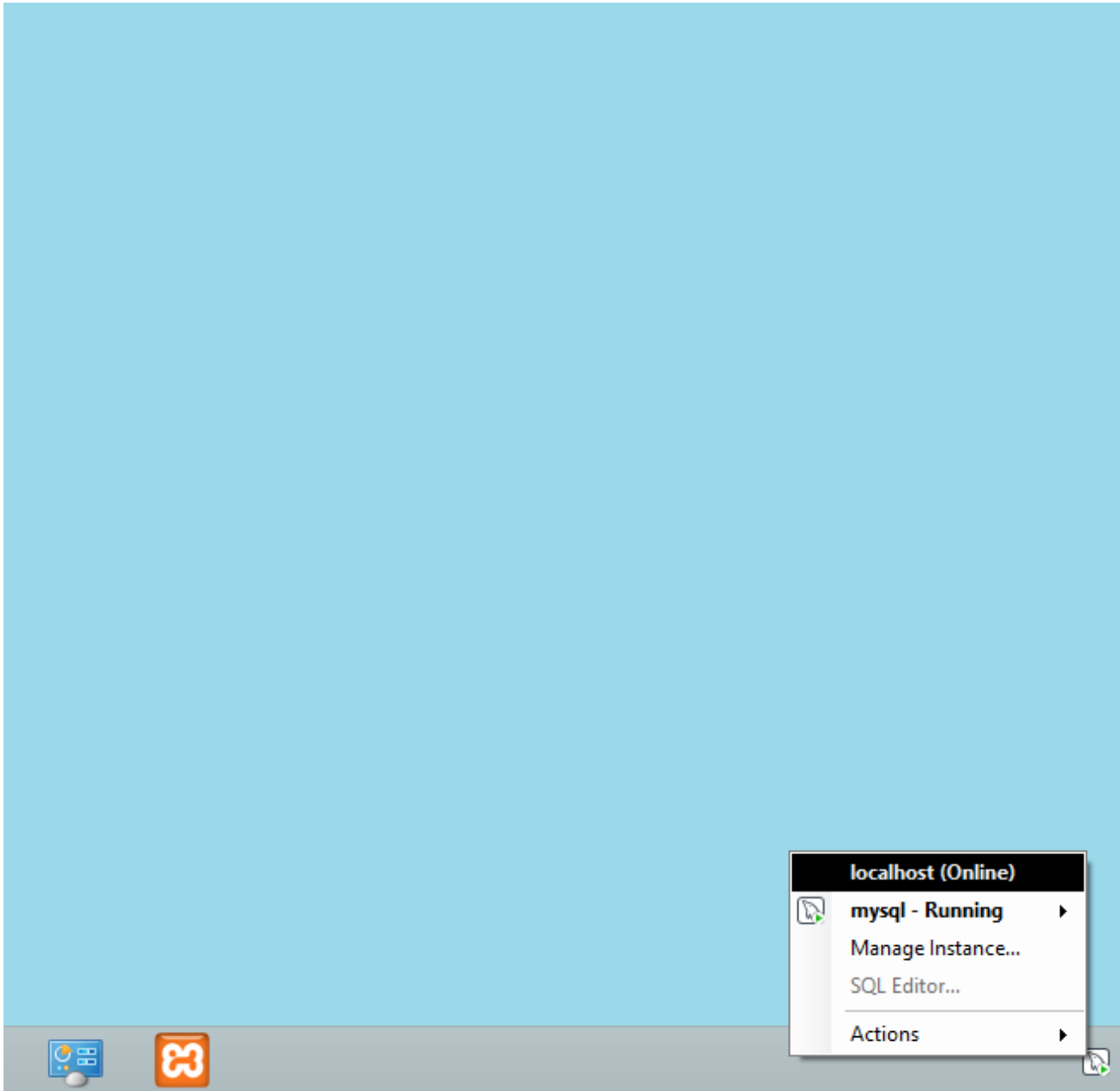
Product	Status	Progress	Notes
  MySQL Server 5.7.16	Installing		
 MySQL Workbench 6.3.7	Ready to Install		
 MySQL Notifier 1.1.7	Ready to Install		
 MySQL For Excel 1.3.6	Ready to Install		
 MySQL Utilities 1.6.4	Ready to Install		
 Connector/ODBC 5.3.6	Ready to Install		
 Connector/C++ 1.1.7	Ready to Install		
 Connector/J 5.1.39	Ready to Install		
 Connector/NET 6.9.9	Ready to Install		
 MySQL Connector/C 6.1.6	Ready to Install		
 MySQL Documentation 5.7.16	Ready to Install		
 Samples and Examples 5.7.16	Ready to Install		

Show Details >


< Back

Execute

Cancel



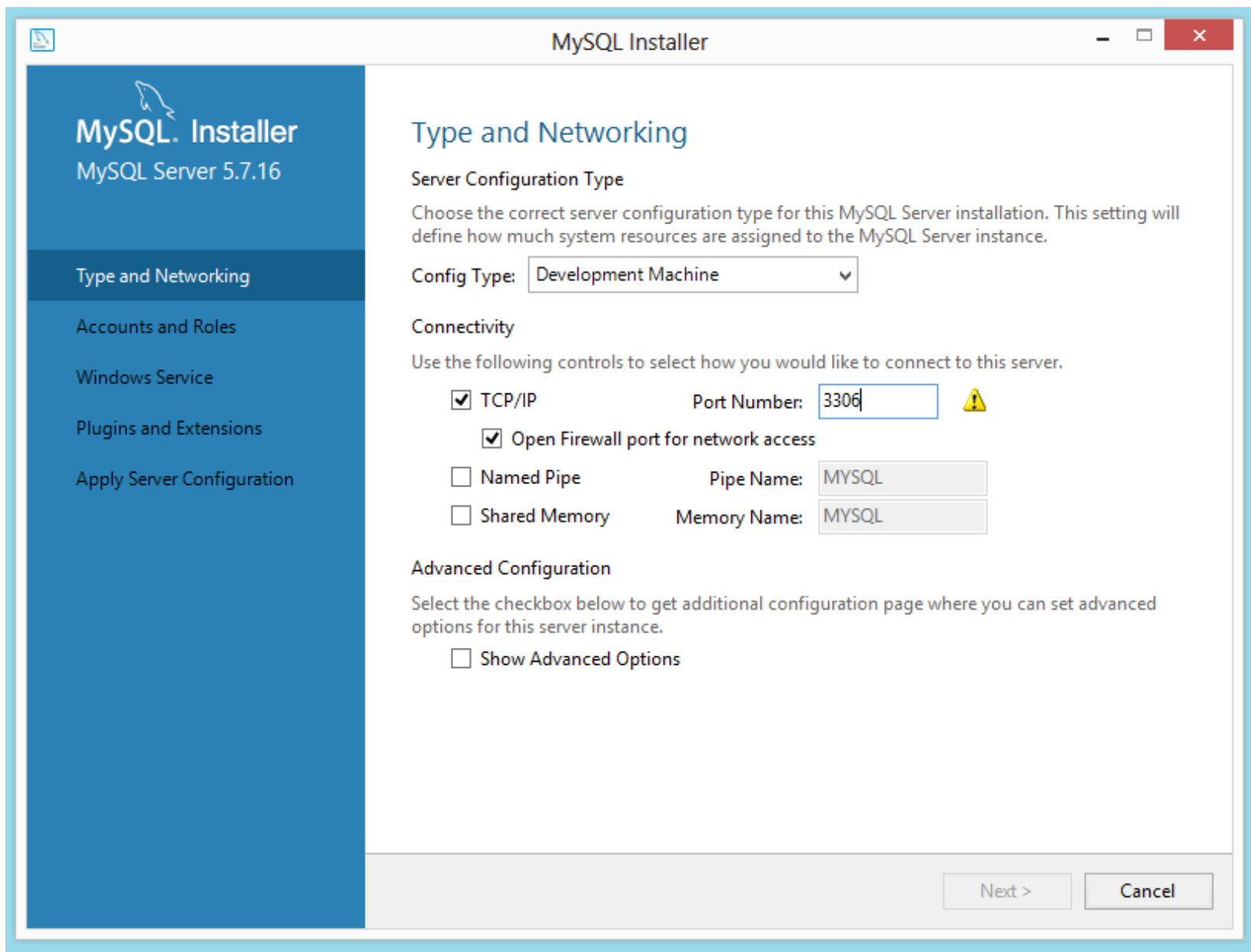
localhost (Online)

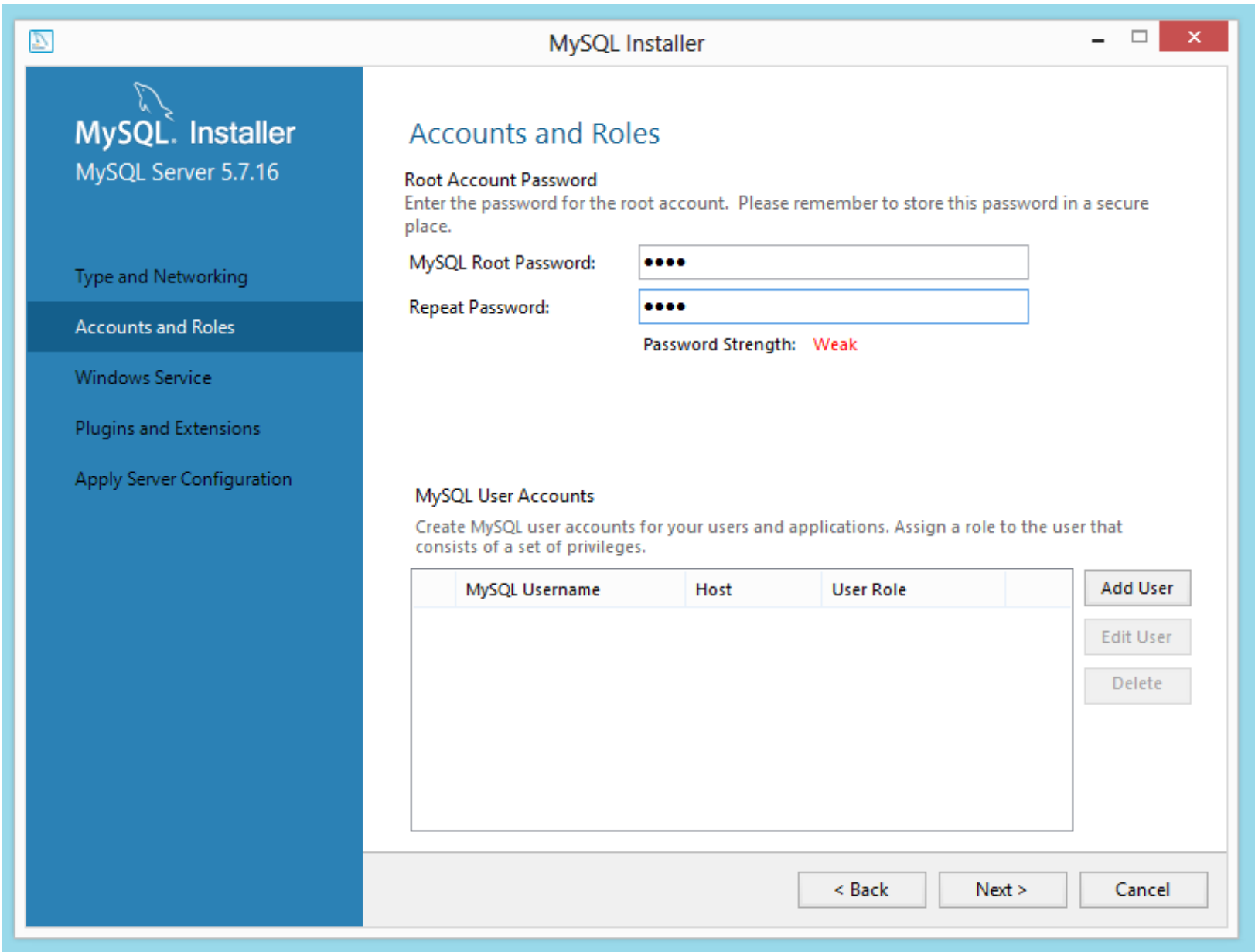
 **mysql - Running** ▶

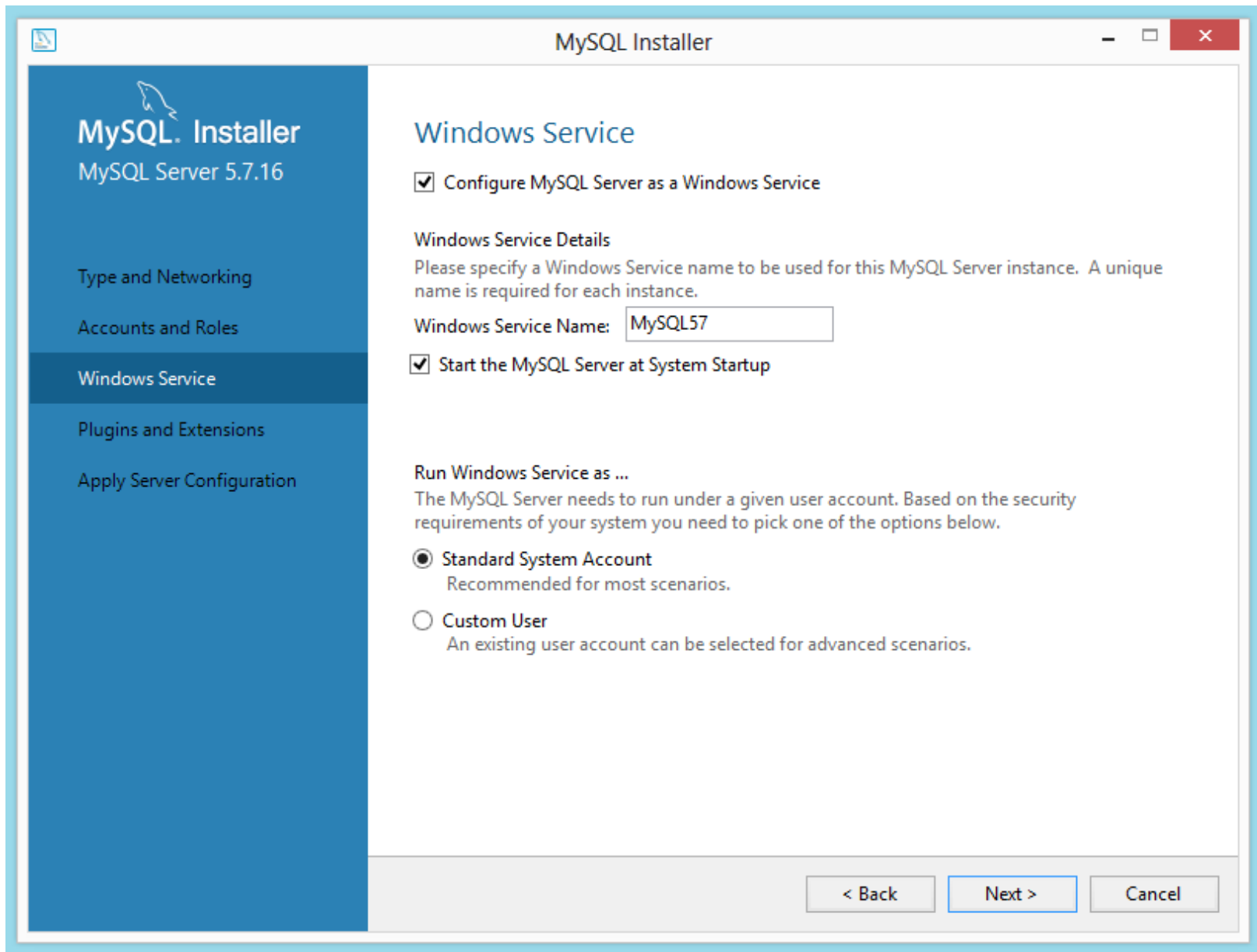
Manage Instance...

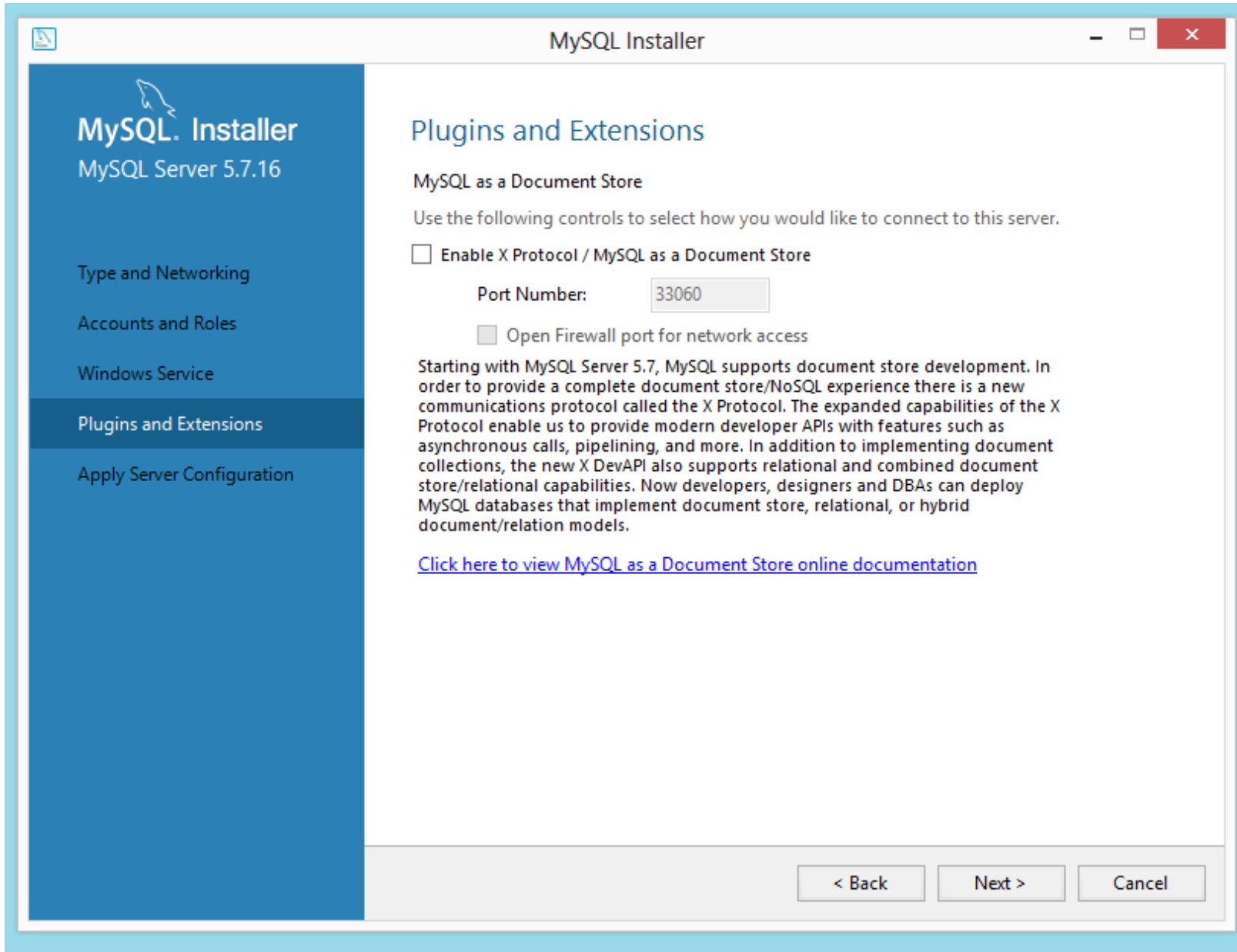
SQL Editor...

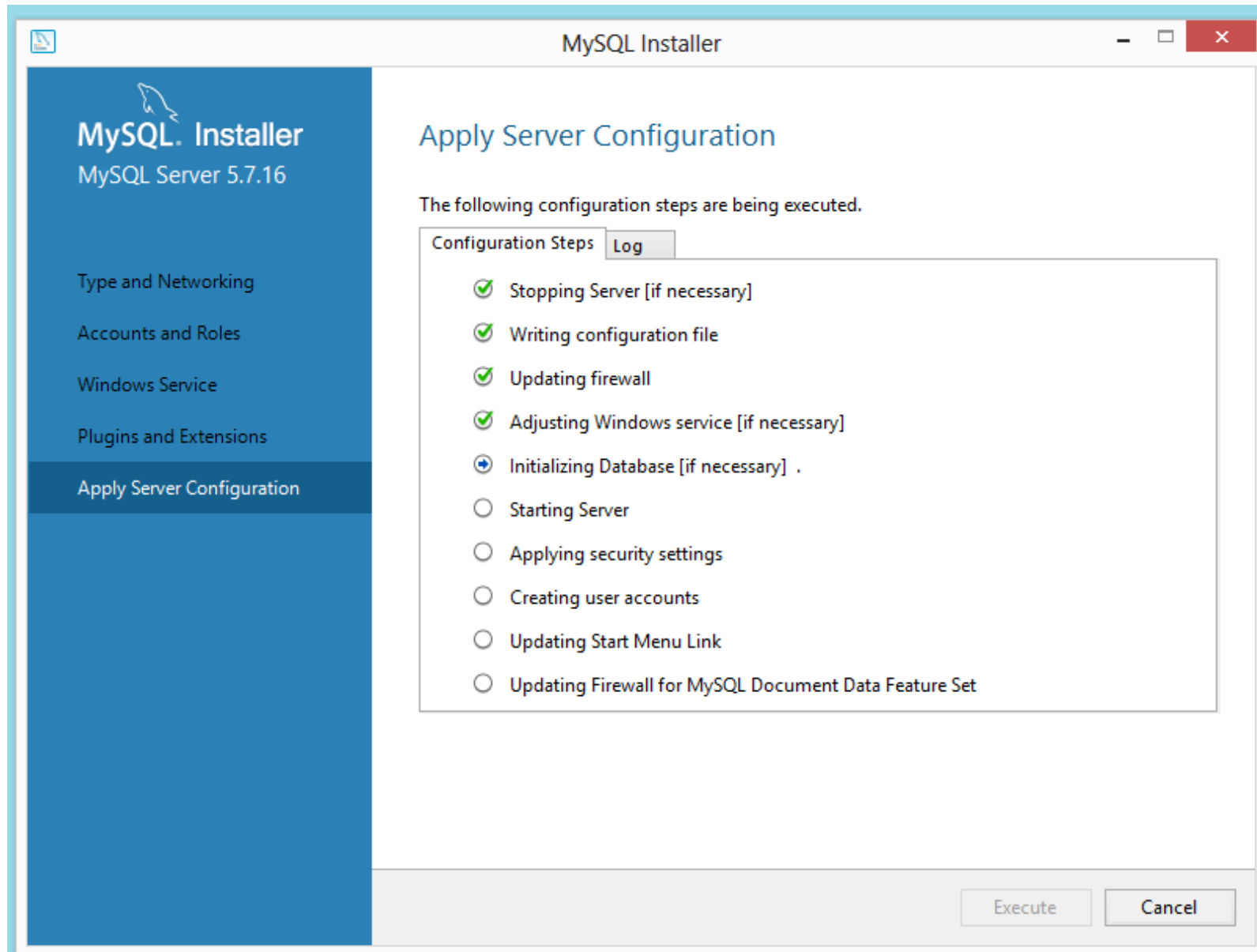
Actions ▶













Services List Changed  
New service added to monitored list: MySQL57

Il “client mysql”

- Il client mysql è il programma client testuale che consente di collegarsi al server MySQL per sfruttarne le funzionalità. (viene installato insieme al server MySQL)

Il “client mysql”

Per lanciare il client mysql è sufficiente richiamarlo dal prompt indicandogli utenza e password (nel nostro caso, useremo sempre l'utente root):

```
shell>mysql --user=root --password=xxx
```

oppure:

```
shell> mysql -uroot -p
```

In questo caso sarà il programma a chiedervi di introdurre la password senza visualizzarla.

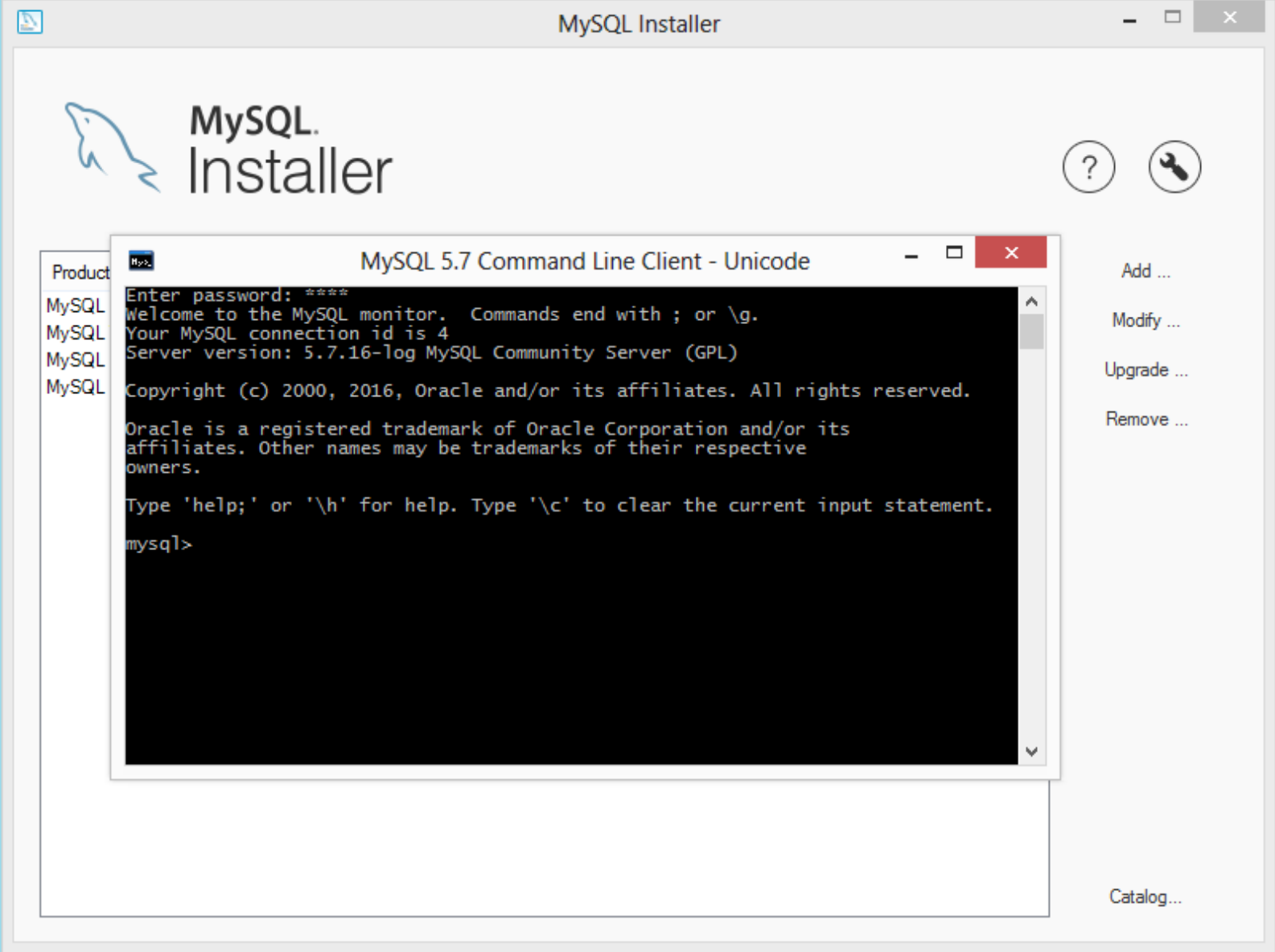
Una volta connessi al DBMS, appare il prompt di mysql:

```
mysql>
```

A questo punto si posso digitare i comandi SQL per interagire con il DBMS.

Per chiudere il client digitiamo:

```
mysql> quit
```



MySQL workbench


- Il client mysql non è il massimo dell'usabilità...
- MySQL workbench è una applicazione grafica che fornisce le stesse funzionalità. Può essere scaricata gratuitamente su:

<http://www.mysql.it/products/workbench/>

Ma nella installazione **developer** è installato di default








- È molto più semplice utilizzare workbench per manipolare le query.

MySQL Connections

 Filter connections


<p>Local instance mysql</p> <p>root localhost:3306</p>	<p>Local instance MySQL57</p> <p>root localhost:3310</p>
--	--

Shortcuts

-  MySQL Utilities
-  Database Migration
-  MySQL Bug Reporter
-  Workbench Blogs
-  Planet MySQL
-  Workbench Forum
-  Scripting Shell

Models

sakila_full

 ..\MySQL Workbench 6.3 CE/extras
sakila
08 Jun 16, 15:51

Importare i comandi

Anziché eseguire comandi SQL digitandoli su terminale è spesso più conveniente scriverli in un file di testo e poi richiamarli dall'interprete dei comandi MySQL.

Supponiamo di aver scritto alcuni comandi SQL in un file `query.sql` nella directory corrente. Possiamo eseguire il file da MySQL con il comando:

```
mysql> source query.sql
```

Ovviamente è possibile anche specificare il path completo del file.

I permessi in MySQL

Una volta connessi al server, l'utente deve possedere i permessi necessari per lavorare sulle varie basi di dati.

Per chiedere quali basi di dati gestite dal DBMS sono accessibili dall'utente (nel nostro caso, root) utilizziamo il comando:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.05 sec)
```

Per selezionare il database che si vuole utilizzare il comando è **use**

```
mysql> use test;
Database changed
```

Creazione di una base di dati

- Per poter creare un nuovo database su cui lavorare utilizziamo il comando:

```
CREATE DATABASE [IF NOT EXISTS] nome_db
```

- Con l'opzione `IF NOT EXISTS` possiamo evitare la segnalazione di errore nel caso esista già un database con lo stesso nome.
- Per eliminare un database si utilizza l'istruzione:

```
DROP DATABASE [IF EXISTS] nome_db
```
- Con l'opzione `IF EXISTS` possiamo evitare la segnalazione di errore nel caso non esista un database chiamato `nome_db`.

ES: Creazione di un database

Creiamo il DB “esempio” utilizzando il “client mysql”.

Le istruzioni da utilizzare sono:

```
mysql> CREATE DATABASE esempio;  
Query OK, 1 row effected (0.06 sec)
```

Ora i database gestiti dall'utente root sono:

```
mysql> show databases;  
  
+-----+  
| Database  
+-----+  
| information_schema  
| mysql  
| test  
| esempio  
+-----+  
4 rows in set (0.05 sec)
```

Creazione delle tabelle (1 / 4)

- Per selezionare la base di dati su cui lavorare usare il comando `USE nome_database`
- Una volta selezionata la base di dati, l'istruzione per definire uno schema di relazione (specificando attributi e vincoli) in MySQL è `CREATE TABLE [IF NOT EXISTS] nome_tabella`
`[(`
`[definizione attributi]`
`[opzioni di tabella]`
`)]`
- La tabella viene creata nel database in uso. In generale, è possibile indicare espressamente in quale database creare la tabella usando `nome_db.nome_tabella`.
- `IF NOT EXISTS` si usa per evitare messaggi di errore nel caso la tabella esista già.

Creazione delle tabelle (2/4)

```
CREATE TABLE [IF NOT EXISTS] nome_tabella
[(
    [definizione attributi]
    [opzioni di tabella]
)]
```

- Le **definizioni attributi** si riferiscono agli attributi della tabella, la loro sintassi è:

nome_colonna TIPO

[NOT NULL | NULL] (di default può contenere valori NULL)

[DEFAULT valore] (usato per impostare un valore di default)

[AUTO_INCREMENT] (per attributi di tipo intero per avere un valore sequenziale generato automaticamente)

[UNIQUE | [PRIMARY] KEY] (UNIQUE rappresenta una superchiave, PRIMARY KEY indica la chiave primaria, oltre a non ammettere duplicati non può contenere valori NULL)

[reference_definition] ()

Creazione delle tabelle (3/4)

- Reference_definition

Tramite le reference_definition è possibile definire vincoli di integrità referenziale, ovvero l'attributo su cui è definito può assumere solo valori specificati nell'attributo di un'altra tabella.

```
REFERENCES nome_tabella [(colonna_indice,...)]
```

Creazione delle tabelle (4/4)

```
CREATE TABLE [IF NOT EXISTS] nome_tabella
[(
  [definizione attributi]
  [opzione, ...]
  [opzioni di tabella]
)]
```

- Le **opzioni tabella** si riferiscono all'intera tabella e permettono di definire diverse proprietà di questa.

- Le più importanti sono:

PRIMARY KEY (nome_attributo1, nome_attributo2,...)

Permette di definire come chiave primaria della tabella un insieme di attributi di questa.

INDEX (nome_attributo1, nome attributo2,...)

Permette di definire degli indici su uno o più attributi della tabella

FOREING KEY (nome_att1,nome,att2,...)

REFERENCE nome_tab(nome_att1,nome,att2,...)

Permette di definire vincoli di integrità referenziale su più attributi

ES: Creazione di una tabella (1 / 2)

Vogliamo creare le seguenti tabelle:

- individui(nome, reddito, eta, sesso)
 - nome è una stringa di 20 caratteri (chiave primaria)
 - reddito è un intero di 10 cifre
 - eta è un intero di 3 cifre
 - sesso è un carattere
- genitori(figlio,genitore)
 - figlio (stringa di 20 caratteri, chiave esterna su INDIVIDUI)
 - genitore (stringa di 20 caratteri, chiave esterna su INDIVIDUI)
 - chiave primaria formata da “figlio” e “genitore”

Creazione di una tabella (2/2)

Creazione tabella Individui:

```
mysql> CREATE TABLE Individui(  
    Nome      CHARACTER(20)  PRIMARY KEY,  
    Reddito   NUMERIC(10),  
    Eta       NUMERIC(3),  
    Sesso     CHARACTER,  
);
```

Creazione tabella Genitori:

```
mysql> CREATE TABLE Genitori(  
    Figlio CHARACTER(20) REFERENCES Individui(Nome),  
    Genitore CHARACTER(20) REFERENCES Individui(Nome),  
    PRIMARY KEY (Figlio,Genitore)  
);
```

Visualizzare le tabelle di un database

Per visualizzare le tabelle di un database usare il comando:

```
mysql> show tables;
```

Dopo la creazione delle tabelle “Individui” e “Genitori” il risultato sarà

```
mysql> show tables;
+-----+
| Tables_in_esempio |
+-----+
| genitori          |
| individui         |
+-----+
2 rows in set (0.01 sec)
```

Visualizzare lo schema di una tabella

Per visualizzare lo schema della tabella `nome_tabella` si utilizza l'istruzione

`SHOW COLUMNS FROM nome_tabella`

oppure

`DESCRIBE nome_tabella`

```
mysql> Describe genitori;
```

Field	Type	Null	Key	Default	Extra
Figlio	char(20)	NO	PRI		
Genitore	char(20)	NO	PRI		

2 rows in set (0.07 sec)

Modificare una tabella

È possibile modificare una tabella attraverso il comando `ALTER TABLE`:

```
ALTER TABLE nome_tabella
  ADD nome_attributo TIPO |
  ADD INDEX (nome_attributo,...) |
  ADD [CONSTRAINT [nome_vincolo]] PRIMARY KEY (nome_attributo,...) |
  ADD [CONSTRAINT [nome_vincolo]] UNIQUE (nome_attributo,...)
  ADD [CONSTRAINT [nome_vincolo]]
  FOREIGN KEY (colonna_indice,...) [reference_definizione] |
  CHANGE vecchio_attributo nuovo_attributo TIPO |
  DROP nome_attributo
  DROP PRIMARY KEY
  DROP INDEX nome_attributo
  .....
  .....
```

Ridenominazione di una tabella

Per ridenominare una tabella usare il comando

```
ALTER TABLE Nome_Tabella RENAME Nuovo_Nome;
```

ES: Vogliamo cambiare il nome della tabella “Individui” con “Persone”

```
mysql> ALTER TABLE Individui RENAME Persone;
```

Ridenominazione di una colonna

Per ridenominare una colonna di una tabella utilizzare il comando:

```
ALTER TABLE Nome_Tabella CHANGE  
Nome_Colonna_da_cambiare Nuovo_Nome_Colonna  
Proprietà_della_Nuova_Colonna;
```

ES: Vogliamo cambiare il nome del campo “Eta” con “Anni”

```
mysql> ALTER TABLE Persone  
        CHANGE Eta Anni NUMERIC(3);
```

NOTA: Per modificare il tipo di colonna utilizzare il medesimo comando cambiando solo il tipo della colonna.

Aggiungere una nuova colonna

Per aggiungere una nuova colonna ad una tabella utilizzare il comando:

```
ALTER TABLE Nome_Tabella ADD  
Nome_della_Nuova_Colonna Proprietà_Colonna;
```

ES: Aggiungiamo la colonna “n_telefono” alla tabella “Persone”

```
mysql> ALTER TABLE Persone  
        ADD n_telefono NUMERIC(20);
```

Per eliminare una colonna utilizzare il comando:

```
ALTER TABLE Nome_Tabella DROP Nome_Colonna_da_canc
```


Aggiungere un vincolo di chiave esterna (1)

Per aggiungere un vincolo di chiave esterna utilizzare il comando:

```
ALTER TABLE Nome_Tabella  
ADD CONSTRAINT [nome_vincolo]  
FOREIGN KEY (nome_col_che_referenzia)  
REFERENCE Nome_Tabella_Referenziata(nome_colonna_refe);
```

Aggiungere un vincolo di chiave esterna (2)

ES: Date le tabelle:

- Aziende(Nome,Sede,Capitale)
- GruppoAziendale(Nome,Capogruppo)

Vogliamo aggiungere alla tabella GruppoAziendale un vincolo di chiave esterna sull'attributo Capogruppo su Aziende

```
mysql> ALTER TABLE GruppoAziendale
        ADD CONSTRAINT fk_capogruppo
        FOREIGN KEY (Capogruppo)
        REFERENCES Aziende(Nome);
```

Aggiungere un vincolo di chiave esterna (3)

La possibilità di aggiungere un vincolo di integrità referenziale permette di realizzare vincoli **ciclici**:

ES: Persona(Cod_Fiscale, Luogo_Nascita)

Citta(Nome, Sindaco)

Dov'è il problema?

Aggiungere un vincolo di chiave esterna (4)

```
mysql> CREATE TABLE Persona(  
    Cod_Fiscals CHARACTER(20) primary key,  
    Luogo_Nascita CHARACTER(1),  
    FOREIGN KEY (Luogo_Nascita) REFERENCES Citta(Nome));
```

Facciamo riferimento alla tabella “Citta” che ancora NON ESISTE!!

Lo stesso accade se proviamo a creare prima la tabella “Citta”

```
mysql> CREATE TABLE Citta(  
    Nome CHARACTER(20) primary key,  
    Sindaco CHARACTER(20),  
    FOREIGN KEY (Sindaco) REFERENCES Persona(Cod_Fiscale));
```

La tabella “Persona” ancora non esiste

Aggiungere un vincolo di chiave esterna (5)

Soluzione

Eseguo le seguenti istruzione in quest'ordine:

1. Creo la tabella "Persona" SENZA vincoli di foreign key;
2. Creo la tabella "Citta" CON i vincoli di foreing key verso la tabella "Persona" (che ora esiste);
3. Aggiungo il vincolo di foreign key alla tabella "Persona" verso la tabella "Citta" (che ora esiste).

Eliminare una tabella

È possibile eliminare una o più tabelle utilizzando il comando:

```
DROP TABLE [IF EXISTS] nome_tabella [, nome_tabella]
```

Con l'opzione `IF EXISTS` possiamo evitare la segnalazione di errore nel caso non esista una tabella chiamata `nome_tabella`.

ES:

```
mysql> DROP TABLE Genitori;
```

Inserimento dei dati

Per inserire dei dati in una tabella si utilizza l'istruzione:

```
INSERT INTO nome_tabella [(nome_attributo1,nome_attributo2,...)]  
VALUES (valore1,valore2,...);
```

Attenzione:

- L'ordinamento degli attributi (se presente) e dei valori è significativo.
- Le due liste di attributi e di valori devono avere lo stesso numero di elementi.
- Se la lista di attributi è omessa, si fa riferimento a tutti gli attributi della relazione secondo l'ordine con cui sono stati definiti.
- Se la lista di attributi non contiene tutti gli attributi della relazione, per gli altri viene inserito un valore nullo (che deve essere permesso) o un valore di default.

ES: Inserimento dati

Inseriamo alcune tuple nella tabella
Persone(Nome, Reddito, Eta, Sesso)

```
mysql> INSERT INTO PERSONE (Nome,Reddito,Eta,Sesso)
      VALUES ('Aldo',25,15,'M');
mysql> INSERT INTO PERSONE (Nome,Reddito,Eta,Sesso)
      VALUES ('Andrea',27,21,'M');
mysql> INSERT INTO PERSONE (Nome,Reddito,Eta,Sesso)
      VALUES ('Luisa',75,87,'F');
mysql> INSERT INTO PERSONE (Nome,Reddito,Eta,Sesso)
      VALUES ('Maria',55,42,'F');
```


Eliminazione di dati dalle tabelle

Per eliminare una ennupla utilizzare il comando:

```
DELETE FROM nome_tabella [WHERE condizione]
```

ES: Eliminiamo tutte le persone con meno di 18 anni dalla tabella "Persone"

```
mysql> DELETE FROM persone WHERE eta<18;
```

Interrogare un database

Per effettuare un'interrogazione in SQL si utilizza l'istruzione **SELECT**

```
SELECT nome_attributo,...,nome_attributo  
FROM nome_tabella, ...,nome_tabella  
[WHERE condizione]
```

Le tre parti sono solitamente chiamate:

- target list
- clausola from
- clausola where

Le ridenominazioni

SQL permette di specificare un “alias” degli attributi (nella target list usando il comando **AS**) e delle tabelle (nella clausola FROM).

La ridenominazione è usata per:

1. Ottenere segnature più esplicative nei risultati;
2. Creare abbreviazione ed evitare ambiguità;

```
mysql> SELECT p.nome as donne  
        FROM persone p  
        WHERE p.sesso = "F";
```

Preparare la base di dati.

Tutte gli esercizi che devono essere eseguiti su una specifica base di dati.

Si può comunque utilizzare uno script per creare e popolare la base di dati Esercitazione.

Soluzione (0)

Installare la base di dati per l'esercitazione.

```
mysql> source install1.sql;
....
Query OK, 1 row affected (0.02 sec)
Query OK, 1 row affected (0.03 sec)
Query OK, 1 row affected (0.03 sec)
Query OK, 1 row affected (0.05 sec)
....
6 rows in set (0.00 sec)
```

Esercizio 1

Data la tabella:

persone(nome, reddito, eta, sesso)

Effettuare le seguenti interrogazioni:

1. Trovare nome e sesso delle persone con più di 30 anni;
2. Trovare nome e sesso delle persone con più di 30 e reddito superiore a 60;
3. Trovare nome e reddito delle persone che hanno il nome che comincia per “M”;
4. Trovare i generi di sesso che compaiono nella tabella “persone”;
5. Trovare nome, età e reddito delle persone che hanno 30 o 40 anni ed un reddito maggiore o uguale a 50;
6. Trovare il nome delle persone la cui età è sconosciuta.

Soluzione (1)

Trovare il nome e sesso delle persone con più di 30 anni

```
mysql> SELECT nome, sesso FROM persone  
        WHERE eta > 30;
```

```
+-----+-----+  
| nome          | sesso |  
+-----+-----+  
| AntonGiulio  | M     |  
| Luigi         | M     |  
| Luisa        | F     |  
| Maria        | F     |  
| Olga         | F     |  
| Sergio       | M     |  
+-----+-----+  
6 rows in set (0.00 sec)
```

Soluzione (2)

Trovare nome e sesso delle persone con più di 30 anni e reddito superiore a 60

```
mysql> SELECT nome, sesso
        FROM persone
        WHERE reddito > 60 AND eta > 30 ;
+-----+-----+
| Nome   | Sesso |
+-----+-----+
| Luisa  | F     |
| Sergio | M     |
+-----+-----+
2 rows in set (0.00 sec)
```


Soluzione (3)

Trovare nome e reddito delle persone che hanno il nome che comincia per "M"

```
mysql> SELECT nome, reddito
        FROM persone
        WHERE nome LIKE 'M%';
```

```
+-----+-----+
| Nome   | Reddito |
+-----+-----+
| Maria  | 55      |
| Michelangelo | 79      |
+-----+-----+
2 rows in set (0.01 sec)
```

Soluzione (4)

Trovare tutti i generi di sesso che compaiono nella tabella “persone”

```
mysql> SELECT DISTINCT sesso AS genere  
FROM persone;
```

```
+-----+  
| genere |  
+-----+  
|      F |  
|      M |  
+-----+
```

```
2 rows in set (0.01 sec)
```

Soluzione (5)

Trovare nome, eta e reddito delle persone che hanno 30 o 40 anni ed un reddito maggiore o uguale a 50

```
mysql> SELECT nome, reddito, eta
        FROM persone
        WHERE (eta=30 OR eta=40) AND reddito>=50;
```

```
+-----+-----+-----+
| nome          | reddito | eta  |
+-----+-----+-----+
| Beatrice      | 79      | 30   |
| Leonardo      | 79      | 30   |
| Luigi         | 50      | 40   |
| Michelangelo  | 79      | 30   |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Soluzione (6)

Trovare il nome delle persone la cui età è sconosciuta

```
mysql> SELECT nome FROM persone
        WHERE eta IS NULL;
+-----+
| nome  |
+-----+
| Diana |
+-----+
1 row in set (0.00 sec)
```

Esercizio 2

Date le tabelle:

persone(nome, reddito, eta, sesso)

genitori(figlio,genitore)

Effettuare le seguenti interrogazioni:

7. Trovare nome e reddito dei genitori di 'Aldo';
8. Trovare la relazione che mostra per ogni persona i nonni;
9. Trovare il reddito medio dei padri per età;
10. Trovare la relazione che mostra le coppie di fratelli (due persone sono fratelli se hanno un genitore in comune).

Soluzione (7)

Trovare nome e reddito dei genitori di 'Aldo';

```
mysql> SELECT genitore, reddito
        FROM persone, genitori
        WHERE figlio= "Aldo" AND genitore=nome;
```

```
+-----+-----+
| genitore | reddito |
+-----+-----+
| Franco   | 60      |
| Maria    | 55      |
+-----+-----+
```

```
2 rows in set (0.00 sec)
```

Soluzione (8)

Trovare la relazione che mostra per ogni persona i nonni;

```
mysql> SELECT P.Figlio AS Nipote, N.Genitore AS Nonno  
        FROM GENITORI P, GENITORI N  
        WHERE P.Genitore=N.Figlio;
```

Soluzione (9)

Trovare il reddito medio dei padri per età;

```
mysql> SELECT eta, AVG(reddito) AS AVG_REDDITO
        FROM Persone, Genitori
        WHERE nome = genitore AND sesso = 'M'
        GROUP BY eta;
```


Soluzione (10)

Trovare la relazione che mostra le coppie di fratelli
(due persone sono fratelli se hanno un genitore in comune).

```
mysql> SELECT distinct G1.Figlio ASNome1,  
                        G2.Figlio AS Nome2  
FROM GENITORI G1, GENITORI G2  
WHERE G1.Genitore = G2.Genitore and  
      G1.Figlio <> G2.Figlio;
```

Esercizio 3

Date le tabelle:

persone(nome, reddito, eta, sesso)

genitori(figlio,genitore)

Effettuare le seguenti interrogazioni:

11. Trovare le persone che sono genitori di almeno 2 figli;
12. Trovare la relazione che mostra per ciascun figlio i rispettivi genitori (Padre, Madre, Figlio);
13. Trovare l'elenco ordinato dei genitori i cui figli guadagnano tutti più di 20 milioni.

Soluzione (1 1)

Trovare le persone che sono genitori di almeno 2 figli;
CONTARE!!!

```
mysql> SELECT genitore
        FROM Genitori
        GROUP BY genitore
        HAVING COUNT(figlio) >= 2;
```

Soluzione (12)

Trovare la relazione che mostra per ciascun figlio i rispettivi genitori (Padre, Madre, Figlio);

```
mysql> SELECT GP.Genitore AS Padre,  
           GM.Genitore AS Madre, GP.Figlio  
FROM genitori GP, genitori GM,  
     persone PP, persone PM  
WHERE GP.Figlio=GM.Figlio AND  
       GP.Genitore=PP.Nome AND PP.Sesso='M' AND  
       GM.Genitore=PM.Nome AND PM.Sesso='F';
```

Soluzione (13)

Trovare l'elenco ordinato dei genitori i cui figli guadagnano tutti più di 20 milioni.

```
mysql> SELECT DISTINCT genitore
FROM Genitori
WHERE genitore NOT IN
  (SELECT genitore
   FROM Genitori, Persone
   WHERE nome =figlio AND
         reddito <= 20
  )
ORDER BY genitore;
```