Alcune istruzioni SQL – notare che SQL è case insensitive:  
**SELECT \* FROM artisti;**  
Seleziona tutte le righe (\*) della tabella `artisti`;

**SELECT \* FROM artisti WHERE AR\_Nome = 'Leonardo da Vinci';**Seleziona le righe il cui AR\_Nome vale ‘Leonardo da Vinci’;

**SELECT \* FROM artisti WHERE AR\_Nome LIKE '%Leo%';**Seleziona le righe il cui AR\_ Nome contiene la sottostringa ‘Leo’. Il LIKE si usa per mettere questi caratteri speciali. Nell’esempio, viene indicato che prima ci può essere qualsiasi cosa, poi ‘Leo’, poi qualsiasi altra cosa;

**SELECT QQ\_Tecnica FROM quadri;**  
Ottieni una tabella con tutti di i valori assunti da quell’attributo (attenzione ai duplicati!!).

**SELECT DISTINCT QQ\_Tecnica FROM quadri;**  
Ottieni i valori di quell’attributo senza duplicati.

Ora vediamo l’SQLCommandDemo.

Installare il MySQL Connector (NET). Scaricarlo da MySQL e mettilo nel progetto.  
Sistemare il riferimento con quello corretto (sottocartella Framework). Nel progetto di esempio apriamo il file ClassSqlCommandDemoMySql.cs.

La stringa di connessione ha 4 elementi:

* Istanza del motore a cui mi connetto;
* Database a cui mi connetto;
* User ID (username);
* Password.

Vedi le prime righe e le righe che comprendono la 134, più il file GetTabellaClienti\_MySQL.cs.

Aggiungi riferimento: click destro su Dipendenze, Aggiungi riferimento a progetto, Sfoglia, Sfoglia, prendi il connettore NET e confermagli tutto. Si troverà sotto Assembly (è una .dll).

Ricordare che le classi è meglio metterle public invece di internal, così non ci sono problemi.

La classe ClassDB potrebbe essere così:

public class ClassDB

{

private string stringaConn = "Server=localhost;Database=arte;Uid=root;Pwd=''";

public DataTable GetTabellaArtisti()

{

MySqlConnection conn = new MySqlConnection(stringaConn);

MySqlCommand com = new MySqlCommand();

MySqlDataAdapter adapter = new MySqlDataAdapter();

DataTable tabella = new DataTable();

com.Connection = conn;

com.CommandText = "SELECT \* FROM artisti";

adapter.SelectCommand = com;

// Adattiamo il formato dei dati per la DataTable

adapter.Fill(tabella);

return tabella;

}

}

Notare che gli apici obliqui (``) non servono!

Poi istanziamo ClassDB e usiamo il nostro metodo per mostrare i dati nella DataGridView.

In phpmyadmin abbiamo anche il designer.

Nota: per provare le query SQL, usare phpmyadmin. Poi copiarle ed incollarle nel codice una volta verificate.

Vedere il progetto fatto in classe con .NET 8 LTS (Core) chiamato *DB\_Arte\_NET*.