**ADO.NET si Entity Framework**

Diferenta principala in metodele si codul folosite de Entity Framework fata de ADO.NET este modelul orientat obiect al EF – instructiunile scrise in C# care modificate baza de date nu necesita cunostinte de SQL pentru manipularea datelor din tabele ci doar lucru cu clasele(entitatile) generate automat.

|  |  |
| --- | --- |
| ADO.NET | EF |
| Conexiune | |
| **Trebuie creata si deschisa conexiunea, prin connection string, se executa comenzile (prin obiecte ca SqlCommand) pe conexiunea respectiva apoi trebuie inchisa conexiunea si obiectele folosite.**  String connString=”User ID=..;Data Source=…;Password=…;…”  SqlConnection conn=new SqlConnection(connString);  conn.Open();  string strcomm= ”select/insert/update…”;  SqlCommand comm=new SqlCommand(strcomm,conn);  …//executa comanda  Comm.Dispose();  Conn.close(); | **Conexiunea si intructiunile pentru modificarea tabelelor se executa automat, se lucreaza cu un context( sau se poate apela explicit EntityConnection).**  EntityType context=new EntityType();  …Context.TableName.command  //ToList,Create, Select,Delete..  … Context.command  //Addtotable,SaveChanges..  Context.Dispose(); |
| Select | |
| **Se executa un query sql Select apoi datele sunt memorate intr-un obiect DataReader/DataSet.**  ….  String strcomm=”SELECT camp…  FROM t1…  WHERE cond…”;  SqlCommand comm=new SqlCommand(strcomm, conn);  DataReader dr=comm.ExecuteReader();  While(dr.Read())  … | **Se preiau datele prin lista context.TableName .**  …  Context.t1.Select(…);  //Sau  IQueryble<T1> date=from x in context.t1 where… select x…. |
| Insert – Update - Delete | |
| **Se executa comanda sql de tipul respectiv ce afecteaza baza de date direct.**  String strcomm=”UPDATE …  SET…  WHERE…”;  “INSERT INTO…  VALUES(….)”;  “DELETE FROM…”;  SqlCommand comm=new SqlCommand(strcomm, conn);  Comm.ExecuteNonQuery();  …. | **\*Update:**  **Se schimba datele membre ale obiectului de tipul tabelului dorit.**  TableName x=….  x.camp1=…  x.camp2=…  **\*Insert:**  **Se creeaza un obiect de tipul clasei tabelului respectiv si adauga la context.**  TableName x=new TableName();  x.camp1=…  x.camp2=..  context.AddToTableName(x); |
| **Commit;** | **\*Delete:**  **Se sterge din context obiectul tabelului.**  TableName x=….  Context.TableName.DeleteObject(x);  **Context.SaveChanges();** |