Uwagi do kodu

1. Nazwa pliku, który ma być przetwarzany, jest błędna (dataa.csv zamiast data.csv)
2. Przy tworzeniu listy ImportedObjects niepotrzebnie dodawany jest do niej od razu pierwszy obiekt (linijka 16)
3. StreamReader powinien być używany w bloku using (linijka 18)
4. Brakuje sprawdzenia, czy nie iterujemy po pustej tablicy (linijki 32-38) - values może mieć 1, 2 lub 7 elementów, w zależności od tego, czy odczytana linia pliku jest pusta, opisuje element database, czy element table/column
5. Iteracja: (linijka 27)
   for (int i = 0; i <= importedLines.Count; i++)
   powinna odbywać się inaczej:
   for (int i = 0; i < importedLines.Count; i++),
   ponieważ używając obecnego sposobu wykraczamy poza długość listy)
6. Klasa ImportedObject posiada właściwość Name oraz Type, a nie powinna, ponieważ klasa, po której dziedziczy, zawiera te właściwości, mamy powielanie kodu
7. Klasy ImportedObjectBaseClass, ImportedObject, DataReader są umieszczone w jednym pliku, dla porządku każda klasa powinna być opisana w osobnym pliku
8. Pole klasy DataReader - ImportedObjects powinno być interfejsem typu <ImportedObjectBaseClass>, aby móc objąć zarówno obiekty klasy ImportedObjectBaseClass, jak i ImportedObject
9. Niepotrzebnie tworzony jest każdorazowo obiekt klasy ImportedObject (linijka 31), ponieważ nie wszystkie obiekty zawierają wszystkie wymagane przez tę klasę pola. Powinny być tworzone obiekty klasy ImportedObjectBaseClass przy 2 wartościach wyciągniętych z linijki pliku CSV, a przy 7 wartościach obiekty klasy ImportedObject (jeśli pozostajemy przy tworzeniu obiektu w ten sposób i posiadamy tylko 2 klasy opisujące obiekty - czyli te obecnie implementowane ImportedObjectBaseClass oraz ImportedObject).
10. Pętla przypisująca liczbę dzieci:
    var importedObject = ImportedObjects.ToArray()[i]; (linijka 55)
    - powyższa linijka wykonywana jest w każdej iteracji, co jest nieoptymalne ze względu na wykorzystanie pamięci
    Lepiej byłoby przekształcić ImportedObjects do typu Array przed iteracją, a w jej trakcie tylko odwoływać się do indeksu (jeśli koniecznie chcemy korzystać z metody ToArray()).
    Jeszcze lepszym rozwiązaniem jest iterowanie tak, jak w linijce 43:
    foreach (var importedObject in ImportedObjects)
11. Linijki 45-49
    Sądzę, że zastępowanie nowej linii (Environment.NewLine) nie ma sensu, bo skoro streamReader czyta linia po linii, to nie wyodrębniłby jednego wyrazu między średnikami razem ze znakiem nowej linii.
    Ponadto, importedObject.ParentType powinien również zostać przekonwertowany do wielkich liter, ponieważej przy przypisywaniu dzieci może zostać zaniżona ich liczba.
12. Przypisywanie liczby dzieci (linijki 53-66) może być napisane lepiej, ponieważ obecnie dla każdego elementu następuje tyle iteracji, ile wynosi długość listy ImportedObjects, a można by podzielić obiekty na podtypy i iterować tylko po podtypach niższego rzędu od obiektów, którym przypisujemy liczbę dzieci. Poza tym przy braku początkowego zdefiniowania liczby dzieci dla danego obiektu, dodajemy 1 do nulla, a to może powodować problemy.

13. Przy wypisywaniu wartości na wyjście również iterację można by przeprowadzić lepiej, znowu następują niepotrzebne iteracje.
14. Klasę DataReader powinno się podzielić na wiele metod, ponieważ metoda ImportAndPrintData jest za długa i nie wykonuje jednego zadania, tylko wiele.
15. Obiekty pobierane z pliku CSV powinno się podzielić na 3 kategorie, a nie tylko 2 (powinna być utworzona jeszcze jedna klasa)
16. Pole public double NumberOfChildren powinno być innego typu (całkowitego)