* ADO.NET
  + Old, but: consistent, stable
  + Was designed to support large loads, security, scalability
  + Heart: System.Data namespace
  + Compatible with more than just SQL Server (System.Data.OracleClient)
  + **Disconnected model**: Open connection, do INSERT/UPDATE whatever, then close immediately
    - Connections are expensive (network, etc)
    - Locks, concurrency issues
    - **Do not forget to close connections (try-finally)**
  + **Connection pooling:** not good that we are closing/opening connections willy nilly. Create instead some connections (eg 50)
    - Number of active connections: specified in DB connection string, else default 100
* **Data providers:** components designed for data manipulation + forward/read only access to data
  + DbConnection
  + DbCommand
  + DbDataReader: when speed is needed (forward + read only), similar to Stream
  + DbDataAdapter: to populate DataSet, DataTable
  + DataSet: inmemory copy of RDBMS, collection of DataTables
  + DataTable

DataSet vs DataReader: what to use to query: datareader is always used, even with DataAdapter

* DataReader: **async** stuff DataAdapter: **sync**
* DataAdapter fetches tables, rows then hydrate business object collection (slow)
* Both support multiple result sets
* DataAdapter: Fill() only completes when everything’s ready DataReader: HasRows, but it can be incomplete yet (has to manually check)
* DataReader: iterate only once (DataTable: any times)
* DataSets: can be loaded from XML ~serializable



