LINQ Introduction Lab

For your assignment, do these steps:

1. Get an understanding of the requirement. It should be plain from the method signature below, but if it isn't, talk to your instructor for clarifications
2. Write a test for that requirement. Note that since we're testing against the actual database, these are technically integration tests and not unit tests.
3. Create or open the repository
4. Write the method stub, throwing a new NotImplementedException.
5. Compile to a red test
6. Make the test turn green by filling out the method stub and writing any backing code that is needed.

Customer things

Repository	Write this method
CustomerRepository	public bool Exists(string CustomerId)
CustomerRepository	public Customer ReadById(string CustomerId)
CustomerRepository	public List<Customer> ReadByCompanyName(string CompanyName)

Employee things

Repository	Write this method
EmployeeRepository	public bool Exists(int EmployeeId)
EmployeeRepository	public List<Employee> ReadAll()
EmployeeRepository	public List<Employee> ReadByLastName(string LastName)
EmployeeRepository	public Employee ReadById(int EmployeeId)

Product things

Repository	Write this method
ProductRepository	public List<Product> ReadByProductName(string ProductName)
ProductRepository	public Product ReadById(int ProductId)
ProductRepository	public bool Exists(int ProductId)
ProductRepository	public List<Product> ReadAll()

Order things

Repository	Write this method
OrderRepository	public List<Order> ReadAll()
OrderRepository	public List<Order> ReadByCustomerId(int CustomerId)

Bonus!

if you're finished early, do this:
You should have a method in your cart class that allows you to change the quantity of a product in the cart. It probably has a foreach statement to find the product. Refactor that foreach to use LINQ instead. (Hint: you'll want to use a where clause to find the OrderDetail line with the given productId).

Extra Bonus!

Find any other foreach statement in your library and refactor it in the same way.