Seems like the same issue as this one: jtds-driver-not-working-for-sql-sever-2008r2-and-denali-native-sspi-library-not

You should drop the appropriate ntlmauth.dll file from the JTDS download package into your JRE bin folder.

If you're running on a 64bit Windows machine:

This 32bit DLL:

Downloads >>> jtds-1.3.0-dist.zip >>> x86 >>> SSO >>> ntlmauth.dll

Goes here in this 32bit JRE location:

C:\Program Files (x86)\Java\jre7\bin

This 64bit DLL:

Downloads >>> jtds-1.3.0-dist.zip >>> x64 >>> SSO >>> ntlmauth.dll

Goes here in this 64bit JRE location:

C:\Program Files\Java\jre7\bin

If you're running on a 32bit Windows machine:

This 32bit DLL:

Downloads >>> jtds-1.3.0-dist.zip >>> x86 >>> SSO >>> ntlmauth.dll

Goes here in this 32bit JRE location:

C:\Program Files\Java\jre7\bin

If that doesn't work then try adding this line at the top of your main method:

```
System.out.println(java.lang.System.getProperty('java.library.path'));
```

It should output the actual JRE path being used by your program. Make sure the appropriate ntlmauth.dll is in that JRE's bin folder.

NOTE: Do NOT set the domain, user, or password properties of the connection when utilizing this method.

NOTE: If your Java client program is running on a non-windows machine, you're out of luck using the ntlmauth.dll method. This is a quote from the documentation included with the JTDS download package: Downloads >>> jtds-1.3.0-dist.zip >>> README.SSO

Starting with release 0.9.2 jTDS is capable of using the Windows credentials of the current user under whose account the client program is running to login to SQL Server (Windows Single Sign On).

Using Windows Single Sign On (SSO) clients will be able to connect to Microsoft SQL Server without providing credentials. It would dynamically read current user's credentials and connect to the database. provided that the Windows account has enough rights in the database. This is done using a native (Windows-only) library, ntlmauth.dll.