1- Baixe a última versão do Dev-CPP (4.9.9.2 foi a que utilizei) e instale.  
2- Execute o Dev e vá no menu Ferramentas>Atualizações. Em seguida no combobox “Select devpak server” selecione “devpaks.org community Devpaks” (2ª opção do combo) e clique no botão “Check for updates”.  
3- Após o recebimento dos pacotes, no combobox “Groups”, selecione “Data Base” e procure pela “libmysql” na lista. Selecione-a e clique em “Download Selected”.  
4- Após o download uma guia de instalação será iniciada e o pacote será instalado no Dev.  
5- Crie um novo projeto “Empty” e salve-o como “.c”.  
6- Inclua as bibliotecas e no código:

Vá em no menu:

Ferramentas>Opções do Compilador>Diretórios>C Includes

e adicione a pasta include do MySQL a lista. Aqui estava no seguinte endereço: “C:\Arquivos de programas\MySQL\MySQL Server 5.0\include”

7- Abra as propriedades do projeto. (Clique com o botão direito do mouse sobre o nome do projeto ou Alt+P).  
8- Vá na aba “Parâmetros” e na área que está escrito Linker adicione a linha:

-lmysql

10- Clique em ok. Para que o sistema consiga acessar o mysql é preciso que a dll “libmysql.dll” esteja na pasta do projeto. Se não à tem, pode baixá-la [aqui.](http://www.dll-files.com/dllindex/dll-files.shtml?libmysql)

Outro detalhe, o servidor MySQL que utilizei foi o da ferramenta VertrigoServ que pode ser baixada [aqui](http://vertrigo.sourceforge.net/?lang=br). A vantagem é que o Vertrigo é muito mais fácil de instalar e configurar que o servidor MySQL original.

Obs: Crie a base de dados no PHPMyAdmin do Vertrigo antes de fazer o teste!

Abaixo segue o código de teste que utilizei:

#include  
#include  
#include  
#include  
#include

int main(){

MYSQL \*sock;  
sock = mysql\_init(0);  
if (!sock){  
printf(“\nErro na inicializacao!”);  
getche();  
exit(1);  
}

if (!mysql\_real\_connect(sock, ” [127.0.0.1](http://127.0.0.1/)“//ip do servidor

, “root”//usuario do MySQL  
, “vertrigo”//Senha  
, “Trabalho\_BD2″//nome da base de dados  
, 0, NULL, 0)){  
printf(“\nErro na conexao!”);  
getche();  
exit(1);  
}  
else printf(“\nConectado!”);

mysql\_close(sock);

system(“PAUSE>>null”);

return 0;  
}

Também consegui fazer a comunicação entre a API do MySQL e o Borland C++ Builder, mas como deu um pouco mais de trabalho, posto amanhã! rsrs