

目录

入门	2
连接设置	4
MySQL 常规设置	7
Oracle 常规设置	8
PostgreSQL 常规设置	9
SQLite 常规设置	10
SQL Server 常规设置	11
SSH 设置 (只限于 MySQL、Oracle 及 PostgreSQL 并只支持 SSH2 通讯协定)	12
SSH 通道的好处	13
密码验证	14
公钥验证	15
HTTP 设置 (只限于 MySQL、PostgreSQL 及 SQLite)	17
SSL 设置 (只限于 MySQL 及 PostgreSQL)	18
安装 OpenSSL 及 MySQL 或 PostgreSQL	19
设置 MySQL 或 PostgreSQL 的 SSL 证书	20
设置 Navicat 的客户端证书	23
高级设置	25
设置高级数据库属性 (只限于 MySQL 及 PostgreSQL)	28
与数据库或模式工作	29
MySQL 数据库管理	30
Oracle 模式管理	32
PostgreSQL 数据库管理	33
PostgreSQL 数据库设计器	35
编辑 PostgreSQL 数据库常规设置	36
SQLite 数据库管理	37
与数据库或模式对象工作	39
查看对象信息	40

入门

要在 Navicat 开始 MySQL、Oracle、PostgreSQL 或 SQLite 数据库的工作，你要首先用连接窗口创建一个或多个连接。如果你是一个服务器新手和不肯定如何工作，你可能想看看：

- [MySQL 用户手册](#)
- [Oracle 数据库文件](#)
- [PostgreSQL 用户手册](#)
- [SQLite 用户手册](#)

点击  或选择文件 ->  **新建连接** 来设置连接属性。

- [连接设置](#)

在创建连接后，你可以连接到数据库，管理它的对象、表中的数据等。请看下面的帮助，以了解如何用最简单的方法运行这些操作。

- [与数据库或模式工作](#)
- [与数据库或模式的对象工作](#)

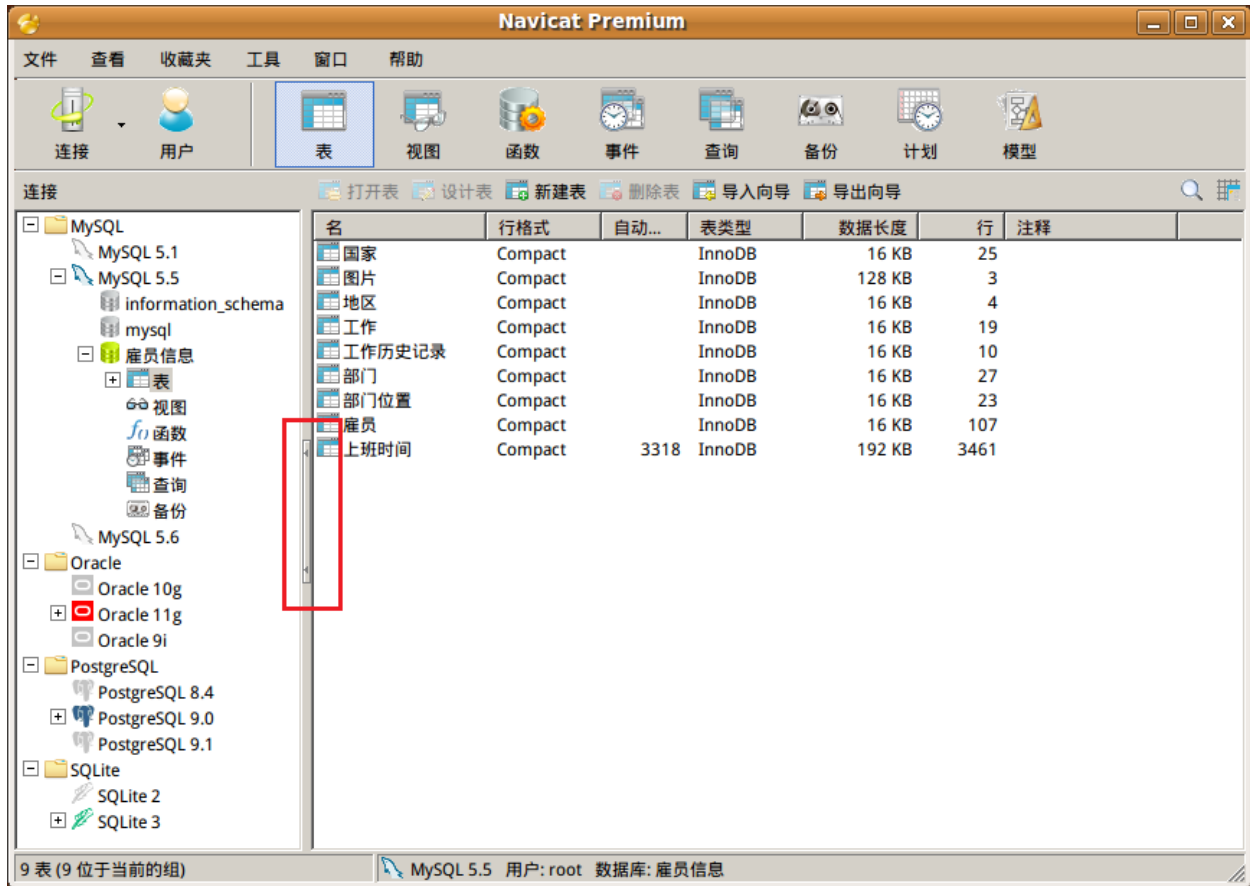
Navicat 浏览器！

Navicat 窗口包括一个导览窗格（左边的窗格）及一个对象窗格（右边的窗格）。

导览窗格一个是导览连接、数据库及数据库对象的基本途径。它采用树状结构，让你透过弹出菜单快捷及方便地使用数据库和它们的对象。

对象窗格显示开启表、查询等。在窗口顶部的工具栏提供其他控制项，你可以用它来操作你的数据。

要查看或隐藏此窗口，点击下面的红色的标示或在主功能表選擇檢視 -> **连接树**。



连接设置

Navicat 集合实用工具来管理你的数据库。要开始在 Navicat 管理你的数据库，首先你需要做的是创建你的服务器连接。

创建连接

Navicat 提供了三种典型的方法来创建你的连接， 点击  或选择文件 ->  **新建连接** 来开始设置。

- [MySQL 常规设置](#)
- [Oracle 常规设置](#)
- [PostgreSQL 常规设置](#)
- [SQLite 常规设置](#)
- [SSH 设置](#)（只限于 MySQL、Oracle 及 PostgreSQL）
- [HTTP 设置](#)（只限于 MySQL、PostgreSQL 及 SQLite）

注意：对于 MySQL 或 PostgreSQL 服务器，一个常用的通讯协定 - **Secure Sockets Layer (SSL)** 是用于管理在网际网路上传输信息的安全（详细信息请看 [SSL 设置](#)）。

Navicat 提供测试用的评估帐号。

远程 MySQL 服务器连接设置为：


- 主机名或 IP 地址：server1.navicat.com
- 端口：4406
- 用户名：navicat
- 密码：testnavicat

远程 PostgreSQL 服务器连接设置为：

- 主机名或 IP 地址：server1.navicat.com
- 端口：5432
- 初始数据库：HR
- 用户名： navicat
- 密码：testnavicat


注意： Navicat 能让你连接到不同平台上的远程服务器，即 Windows、Mac、Linux 及 UNIX。

创建一个与现有连接相同属性的新连接

- 在导览窗格中的连接上右击并选择  **复制连接...**。
- 新创建的连接将被命名为「连接名_copy」。

删除连接

删除一个连接

- 在导览窗格中的连接上右击并选择  **删除连接**。
- 在对话框确认删除。

打开连接

打开一个连接

- 在导览窗格中双击以打开连接。


关闭连接

关闭一个连接

- 在导览窗格中的连接上右击并选择  **关闭连接**。

编辑连接

编辑一个连接的信息

- 如果连接打开着，关闭它。
- 在连接上右击并选择  **连接属性...**。

导出连接设置

导出连接设置

- 选择文件 -> **导出连接...**。
- 选择连接和导出文件路径。

导入连接设置

导入连接设置

- 选择文件 -> **导入连接...**。
- 指定连接设置文件路径。
- 如果连接已存在，在对话框确认替换或跳过。

取得连接信息

取得一个连接的信息

- 在导览窗格中打开连接。
- 在已打开的连接上右击并选择 **连接信息...**。

MySQL 常规设置

下列的指示引导你通过创建一个新连接的进程。要成功地创建一个新的连接到本机或远程 MySQL 服务器 - 不管通过 SSL、SSH 或 HTTP，都要在相应的框中设置连接属性：连接名、主机名、端口号码、用户名及密码。

默认情况下，MySQL 提供用户名为「root」和密码为空白。

连接名

一个最恰当地描述你的连接名。

主机名或 IP 地址

数据库所在的主机名或服务器的 IP 地址。

端口

连接到数据库服务器的 TCP/IP 端口。

用户名

连接到数据库服务器的用户名。

密码

连接到服务器的密码。

你可以连接到你的 MySQL 远程服务器，但基于安全原因，本机远程直接连接到 MySQL 服务器已禁用。因此，你可能不能使用 Navicat Premium 或其他类似的 MySQL 管理应用程序直接连接到远程服务器，除非已配置[用户权限](#)。

如果你的互联网服务提供商（ISP）不提供直接访问其服务器，安全通道协定（SSH）或 HTTP 是另一种解决办法。

Oracle 常规设置

下列的指示引导你通过创建一个新连接的进程。要成功地创建一个新的连接到本机或远程 Oracle 服务器 - 不管通过 SSH，都要在相应的框中设置连接属性：连接名、主机名、端口号码、用户名及密码。

默认情况下，Oracle 在安装时已创建一些用户帐号。管理帐号：SYS、SYSTEM、SYSMAN 及 DBSNMP。样本模式帐号：SCOTT、HR、OE、OC、PM、IX 及 SH。

连接名

一个最恰当地描述你的连接名。

连接类型

连接到服务器的连接类型：**Basic**。

Basic

在 Basic 模式中，Navicat 通过 Oracle Call Interface (OCI) 连接到 Oracle。OCI 是一个应用程序介面，让程序开发人员使用第三代语言原有进程或函数呼叫来访问 Oracle 数据库及控制全部 SQL 语句运行的阶段。OCI 是一个标准数据库访问的库和动态链接库形式检索函数。

主机名或 IP 地址

数据所在的主机名或服务器的 IP 地址。

端口

连接到数据库服务器的 TCP/IP 端口。

Service Name/SID

设置用户连接时用的 Service Name/SID。选择相应的选项按钮。

用户名

连接到数据库服务器的用户名。

密码

连接到服务器的密码。

PostgreSQL 常规设置

下列的指示引导你通过创建一个新连接的进程。要成功地创建一个新的连接到本机或远程 PostgreSQL 服务器 - 不管通过 SSL、SSH 或 HTTP，都要在相应的框中设置连接属性：连接名、主机名、端口号码、初始数据库、用户名及密码。

默认情况下，PostgreSQL 提供用户名为「postgres」和密码为空白。

连接名

一个最恰当地描述你的连接名。

主机名或 IP 地址

数据库所在的主机名或服务器的 IP 地址。

端口

连接到数据库服务器的 TCP/IP 端口。

初始数据库

连接时，用户连接到的初始数据库。

用户名

连接到数据库服务器的用户名。

密码

连接到服务器的密码。

如果你的互联网服务提供商 (ISP) 不提供直接访问其服务器，安全通道协定 (SSH) 或 HTTP 是另一种解决办法。

注意：基于安全原因，本机远程直接连接到 PostgreSQL 服务器已禁用。因此，你可能不能使用 Navicat Premium 或其他类似的 PostgreSQL 管理应用程序直接连接到远程服务器。如需详细信息，请参考下一段的服务器管理。

服务器管理：

默认情况下，PostgreSQL 只允许由使用 TCP/IP 连接的本机计算机连接。其他的机器将无法连接，除非你修改在 *postgresql.conf* 文件中的 *listen_addresses*，通过修改 *\$PGDATA/pg_hba.conf* 文件启用基于主机的认证及重开服务器。欲了解更多信息：[客户端身份验证](#)

SQLite 常规设置

下列的指示引导你通过创建一个新连接的进程。要成功地创建一个新的连接到本机或远程 SQLite 服务器 - 不管通过 HTTP，都要在相应的框中设置连接属性：连接名、类型及数据库名。

连接名

一个最恰当地描述你的连接名。

类型

指定数据库的类型。

现有的数据库文件

连接到 **数据库文件** 的现有数据库。

新建 SQLite 3

在 **数据库文件** 创建新的 SQLite 3 数据库。

新建 SQLite 2

在 **数据库文件** 创建新的 SQLite 2 数据库。

数据库文件

指定初始数据库文件。如果 HTTP 通道打开，你需要输入数据库在你网络服务器的绝对文件路径。

SQL Server 常规设置

下列的指示引导你通过创建一个新连接的进程。要成功地创建一个新的连接到本机或远程 SQL Server - 不管通过 SSH，都要在相应的框中设置连接属性：连接名、主机名及验证类型。

连接名

一个最恰当地描述你的连接名。

主机名或 IP 地址

数据库所在的主机名或服务器的 IP 地址。

验证

SQL Server 使用两种方式来验证连接到 SQL Server 数据库：SQL Server Authentication 及 Windows Authentication。

SQL Server Authentication

SQL Server Authentication 使用登录记录来验证连接。每次用户连接，都必须提供他们的用户名及密码。

用户名

连接到数据库服务器的用户名。

密码

连接到服务器的密码。

Windows Authentication

当一个用户通过一个视窗用户帐号连接，SQL Server 会使用操作系统中 Windows 主要代符来验证帐号名及密码。这意味着用户身份是由 Windows 确认。SQL Server 不会要求输入密码，并且不执行身份验证。

如果你的互联网服务提供商 (ISP) 不提供直接访问其服务器，安全通道协定 (SSH) 是另一种解决办法。

SSH 设置（只限于 MySQL、Oracle 及 PostgreSQL 并只支持 SSH2 通讯协定）

Secure SHell (SSH) 是一个通过网络登录其他计算机的程序，在远程服务器运行命令，和从一台机器移动文件到另一台。在不安全的网络中，它提供两台主机之间强大认证和安全加密的的通讯，被称为 **SSH Port Forwarding (Tunneling)**。通常情况下，它是使用为一个 Telnet 的加密版本。

在一个 Telnet 阶段作业，全部的通讯，包括用户名和密码，会用纯文本传输，让任何人都能监听你的阶段作业及窃取密码或其他信息。这种阶段作业也容易受到阶段作业劫持，一旦你验证，恶意用户就能接管这种阶段作业。SSH 的目的是防止这种漏洞，并允许你在不影响安全性的情况下访问远程服务器的 shell。

- [SSH 通道的好处](#)。

为确保进来的连接请求是由你发出，SSH 能够使用密码，或公开及私钥对（也称为公钥）验证机制。

- [密码验证](#)。
- [公钥验证](#)。

注意：请确保在 Linux 服务器的参数 - 「AllowTcpForwarding」设置值为「yes」，否则，会禁用 SSH port forwarding。要查找路径：[/etc/ssh/sshd_config](#)。在默认情况下，SSH port forwarding 应该已启用。请仔细检查该值的设置。

****** 即使服务器支持 SSH 通道，然而，如果 port forwarding 被禁用，Navicat 就无法通过 SSH 端口 22 连接。

SSH 通道的好处

SSH 有一个极好功能叫做 **SSH Port Forwarding**，有时也被称为 **SSH 通道**，它允许你创建一个安全的阶段作业，然后通过它打开随心所欲的 **TCP** 连接。通道可以随时创建，几乎不需要任何努力及编程，这令到它们非常有吸引力。在无数不同的方式，**SSH Port Forwarding** 可以用于安全通信。

许多提供服务器托管的托管公司会封锁托管公司网络以外访问服务器，以及只授予访问给本机 (**localhost**) 用户连接。



使用 **SSH** 的多个好处：

- 当服务器端口被封锁时，连接到使用了防火墙的服务器。
- 自动验证用户，没有发送纯文本的密码，以防止窃取密码。
- 多个强大的认证方法，防止安全威胁如欺骗的身份。
- 安全和快速的加密和压缩数据。
- 安全文件传输。

密码验证

使用这种模式下，SSH 是几乎和 telnet 程序相同。当你进行连接时，它会询问你的密码。输入它后，你会登录或被拒绝。你的密码会首先加密并由网络传送，然后在远程主机解密。这是鼓励大多数用户使用的模式，因为它不需要额外的设置或配置。

下列指示将引导你完成设置使用密码验证的 SSH 连接。要成功地创建一个 SSH 连接，在相应的框中设置 SSH 连接属性：主机名或 IP 地址、端口号码、用户名、验证方法及密码。

1. 点击  或选择文件 ->  **新建连接** 来设置连接属性。
2. 选择 **SSH** 选项卡并启用 **使用 SSH 通道**。
3. 填写所需的信息：

主机名或 IP 地址

SSH 服务器的主机。

端口

SSH 服务器的端口，默认情况下它是 22。

用户名

在 Linux 机器的用户。（这是一个 Linux 用户。它不是一个数据库服务器用户。）

验证方法

在**密码验证**和公钥验证间选择。

密码

这是一个 Linux 用户的密码。



常规	高级	SSL	SSH	HTTP
<input checked="" type="checkbox"/> 使用 SSH 通道				
主机名或 IP 地址:		SSH_服务器_IP_地址		
端口:		22		
用户名:		SSH_登录名		
验证方法:		密码		
密码:		*****		
<input checked="" type="checkbox"/> 保存密码				

4. 在常规设置页面的 Navicat 主机名是由你的数据库托管公司提供，应设置和 SSH 服务器相对。



公钥验证

公钥验证是基于电子签名的使用并提供最佳的安全验证。

要使用公钥验证，有四项需要：

- 在你连接到的远程服务器必须有你的公钥。
- 在你的本机客户端必须有你的私钥。
- 远程服务器必须配置为允许你使用你的公钥登入。
- 本机客户端必须配置为当你登入远程服务器时使用你的私钥。

下列指示将引导你完成配置使用公钥验证的 **SSH** 连接。要成功地创建一个 **SSH** 连接，在相应的框中设置 **SSH** 连接属性：主机名或 **IP** 地址、端口号码、用户名、验证方法、私钥及密码短语。

1. 点击  或选择文件 ->  **新建连接** 来设置连接属性。
2. 选择 **SSH** 选项卡并启用 **使用 SSH 通道**。
3. 填写所需的信息：

主机名或 **IP** 地址

SSH 服务器的主机。

端口

SSH 服务器的端口，默认情况下它是 **22**。

用户名

在 **Linux** 机器的用户。（这是一个 **Linux** 用户。它不是一个数据库服务器用户。）

验证方法

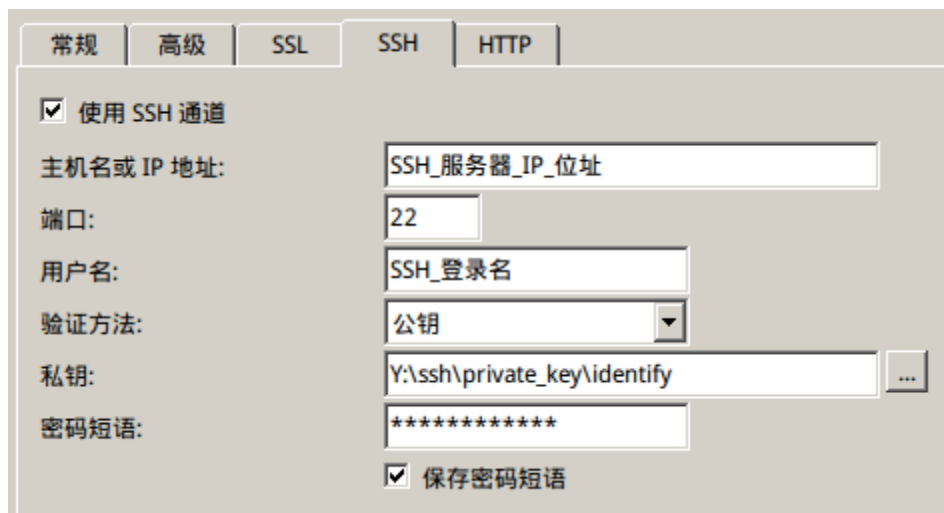
在密码验证和**公钥验证**间选择。

私钥

这是和你的公钥一起使用。私钥应该只有你可读。

密码短语

密码短语和密码完全一样，除了它应用于你生成的钥而不是帐号。密码短语的长度是 **1024** 字符以下。



4. 在常规设置页面的 Navicat 主机名是由你的数据库托管公司提供，应设置和 SSH 服务器相对。

HTTP 设置（只限于 MySQL、PostgreSQL 及 SQLite）

HTTP 通道是一种连接到服务器的方法，使用与网络服务器相同的通讯协定（`http://`）和相同的端口（端口 80）。这是当你的互联网服务供应商不允许直接连接到他们的服务器，但允许创建 HTTP 连接时使用。

创建 HTTP 连接的步骤：

1. 上传通道脚本

若要使用此连接方法，首先需要做的就是上传通道脚本到服务器位于的网络服务器。

注意： `ntunnel_mysql.php`、`ntunnel_pgsql.php` 或 `ntunnel_sqlite.php` 是在 Navicat 的安装文件夹中。

2. 设置 HTTP 通道

下列指示将引导你完成设置 HTTP 连接进程。

- i. 点击  或选择文件 ->  **新建连接** 来设置连接属性。
- ii. 选择 HTTP 选项卡并启用 **使用 HTTP 通道**。
- iii. 输入通道脚本的网址，例如：`http://www.navicat.com/ntunnel_mysql.php`。
- iv. 如果你的服务器安装了 ModSecurity，你可以勾选 **用 base64 编码传出查询** 选项。
- v. 如果通道脚本是在一个密码保护的服务器或你要通过代理服务器来访问网际网路，你可以在 **验证** 或 **代理服务器** 选项卡提供所需的详细验证数据。
- vi. 在常规设置页面的 Navicat 主机名是由你的数据库托管公司提供，应设置和 HTTP 服务器相对。

注意： HTTP 通道和 SSH 通道不能同时运作。当你选择了 HTTP 通道，SSH 通道禁用，反之亦然。

SSL 设置（只限于 MySQL 及 PostgreSQL）

Secure Sockets Layer(SSL) 是一个通过网际网路传输私人文件的协定。为了安全连接到 MySQL 或 PostgreSQL 服务器，首先你需要做的是安装 OpenSSL 库和下载 MySQL 或 PostgreSQL 数据库源。

创建 MySQL 或 PostgreSQL 及 Navicat 的 SSL 连接步骤：

1. [安装 OpenSSL 及 MySQL 或 PostgreSQL](#)。
2. [设置 MySQL 或 PostgreSQL 的 SSL 证书](#)。
3. [设置 Navicat 的客户端证书](#)。

注意：PostgreSQL 8.4 或以上版本支持。

安装 OpenSSL 及 MySQL 或 PostgreSQL

安装 OpenSSL

1. 下载 OpenSSL - <http://www.openssl.org>
2. Linux 命令 : [zcat 0.96l.tar.gz | tar xvf -]
3. Linux 命令 : [./config]
4. Linux 命令 : [make]
5. Linux 命令 : [make install]

安装 MySQL

1. 下载 MySQL - <http://www.mysql.com>
2. Linux 命令 : [./configure --with -vio --with -openssl]
3. Linux 命令 : [make]
4. Linux 命令 : [make install]

注意：请确保 MySQL 服务器支持 OpenSSL 使用查询语句 : [SHOW VARIABLES LIKE 'have_openssl';] - 返回值 = YES

安装 PostgreSQL

1. 下载 PostgreSQL - <http://www.postgresql.org>
2. Linux 命令 : [./configure --with-openssl]
3. Linux 命令 : [gmake]
4. Linux 命令 : [gmake install]

注意：请确保 PostgreSQL 服务器支持 OpenSSL 使用查询语句 : [SHOW ssl;] - 返回值 = ON

设置 MySQL 或 PostgreSQL 的 SSL 证书

要创建服务器或客户端证书，用 root 登录 Linux 服务器并使用 Shell 命令如下：

MySQL

1. `DIR=`pwd`/openssl`
2. `PRIV=$DIR/private`
3. `mkdir $DIR $PRIV $DIR/newcerts`
4. `cp /usr/share/ssl/openssl.cnf $DIR`
5. `replace ./demoCA $DIR -- $DIR/openssl.cnf`
6. 生成 Certificate Authority (CA)

```
/usr/local/ssl/bin/openssl req -new -x509 -keyout $PRIV/cakey.pem -out $DIR/cacert.pem -config $DIR/openssl.cnf
```

注意：如果「PEM」是需要的，请通过以下步骤输入另外的「PEM pass」。

7. 创建服务器的请求和密钥

```
/usr/local/ssl/bin/openssl req -new -keyout $DIR/server-key.pem -out $DIR/server-req.pem -days 3600 -config $DIR/openssl.cnf
```

8. 从密钥移除密码短语（非必须的）

```
/usr/local/ssl/bin/openssl rsa -in $DIR/server-key.pem -out $DIR/server-key.pem
```

9. 签署服务器证书

```
/usr/local/ssl/bin/openssl ca -policy policy_anything -out $DIR/server-cert.pem -config $DIR/openssl.cnf -infiles $DIR/server-req.pem
```

10. 创建客户端的请求和密钥

```
/usr/local/ssl/bin/openssl req -new -keyout $DIR/client-key.pem -out $DIR/client-req.pem -days 3600 -config $DIR/openssl.cnf
```

11.从密钥移除密码短语（非必须的）

```
/usr/local/ssl/bin/openssl rsa -in $DIR/client-key.pem -out $DIR/client-key.pem
```

12.签署客户端证书

```
/usr/local/ssl/bin/openssl ca -policy policy_anything -out $DIR/client-cert.pem  
-config $DIR/openssl.cnf -infiles $DIR/client-req.pem
```

13.创建一个 **my.cnf** 文件来测试证书。保存于 **/etc** 或 MySQL 数据目录（通常在源安装的 **/usr/local/var**）

my.cnf 范例属性：

```
[client]  
ssl-ca=$DIR/cacert.pem  
ssl-cert=$DIR/client-cert.pem  
ssl-key=$DIR/client-key.pem  
[mysqld]  
ssl-ca=$DIR/cacert.pem  
ssl-cert=$DIR/server-cert.pem  
ssl-key=$DIR/server-key.pem
```

14.要开始 MySQL 程序类型：

```
/usr/local/libexec/mysqld -u mysql &
```

or

```
/usr/local/sbin/mysqld -u &
```

PostgreSQL

1. 要创建一个服务器的快速自签署证书，使用以下的 **OpenSSL** 命令：

```
openssl req -new -text -out server.reqm
```

2. 填写 **openssl** 要求的信息。请确认你输入的本机主机名为「**Common Name**」；要求的密码可以留空。程序会生成一个密码保护的密钥；它不会接受一个少于 **4** 个字符长度的密码短语。要移除密码短语（你必须，如果你想要自动启动的服务器），运行命令：

```
openssl rsa -in privkey.pem -out server.key  
rm privkey.pem
```

3. 输入旧的密码短语来解开现有的密钥。现在做的：

```
openssl req -x509 -in server.req -text -key server.key -out server.crt
```



4. 将证书变为自签署证书和复制密钥及证书到服务器将在那裏查找他们。最后做：

```
chmod og-rwx server.key
```

设置 Navicat 的客户端证书

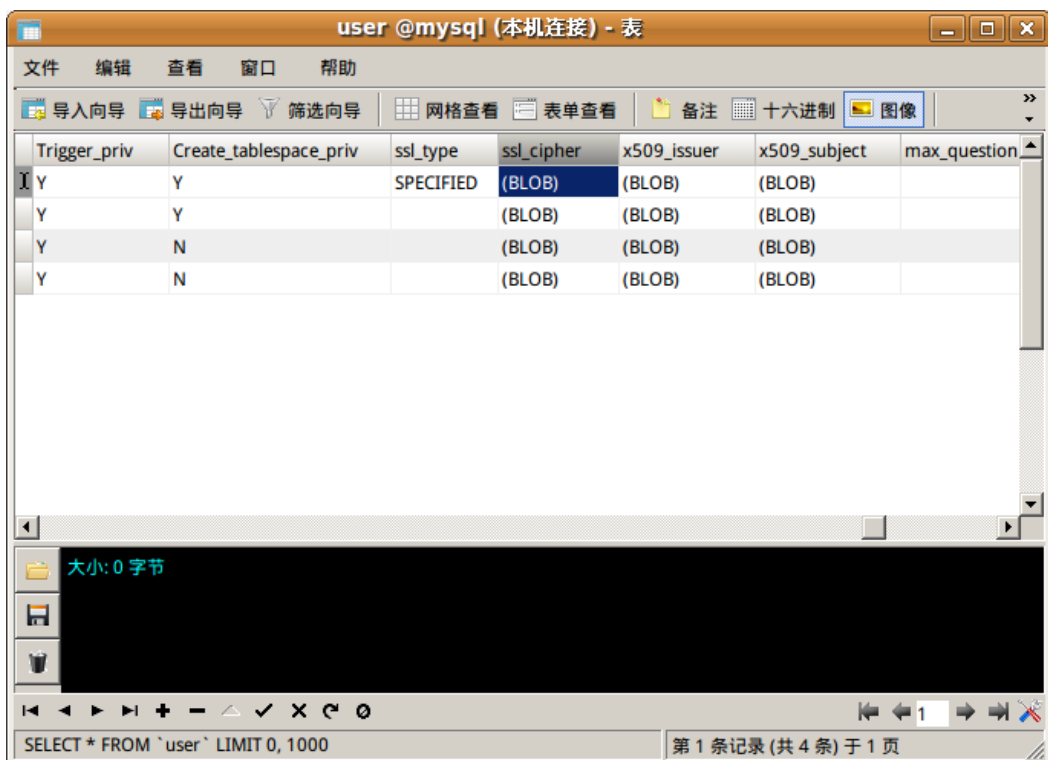
下列指示将引导你完成配置 Navicat 和 MySQL 或 PostgreSQL 服务器之间使用 SSL 的连接进程。要成功地创建一个 SSL 连接，请填写步骤 1 及步骤 2，并在相应的框中设置连接属性。

MySQL



1. 点击  或选择文件 ->  **新建连接** 来设置连接属性。
2. 选择 **SSL** 选项卡并启用 **使用 SSL**。
3. 提供验证数据，填写所需的信息：

客户端密钥、客户端证书及 **CA 证书** 通常保存在你的服务器 - `/usr/local/openssl`。请从远程服务器复制他们到本机计算机。指定密码检索表（非必须的）只需要在 **ssl_type** 栏位设置为「**SPECIFIED**」- [**ssl_type** 可以在名为「mysql」的系统数据库中找到 -> 「user」表]。指定密码检索表的范例是「EDH-RSA-DES-CBC3-SHA」，可以填写在上面显示的连接属性或下面的「mysql」数据库 -> 「user」表 -> 「ssl_cipher」blob 栏位。

注意：你可以保存你的指定密码检索表在一个文本文件，以便载入到到「ssl_cipher」blob 栏位。



PostgreSQL

1. 点击  或选择文件 ->  **新建连接** 来设置连接属性。
2. 选择 **SSL** 选项卡并启用 **使用 SSL**。
3. 选择 **SSL 模式**。

require - 只尝试一个 SSL 连接。

verify-ca - 只尝试一个 SSL 连接，并验证服务器证书是由受信任的 CA 发行。

verify-full - 只尝试一个 SSL 连接，验证服务器证书是由受信任的 CA 发行及服务器主机名符合证书内的。

4. 要提供验证数据，启用 **使用验证** 并填写所需的信息：

客户端密钥、客户端证书 及 **CA 证书** 通常保存在你的服务器 - [/usr/local/openssl](#)。请从远程服务器复制他们到本机计算机。

证书撤销列表 指定 SSL 证书撤销列表 (certificate revocation list - CRL) 的文件路径。

对于 PostgreSQL 服务器，OpenSSL 支持区域广泛不同强度的密文和验证算法。当密文列表可以在 OpenSSL 设置文件中指定，你可以透过修改 `postgresql.conf` 中的 `ssl_ciphers` 指定数据库要使用的密文。

高级设置

根据你的需要自定义连接选项。详细介绍如下：

设置保存路径

当创建一个新的连接，Navicat 将创建一个子文件夹在设置保存路径内。大多数文件都保存在该子文件夹：

Navicat 对象	服务器类型	扩展名
查询	全部	.sql
导出查询结果设置文件	MySQL	.npeq
	Oracle	.nopeq
	PostgreSQL	.nppeq
	SQLite	.nlpeq
导出视图结果设置文件	MySQL	.npev
	Oracle	.nopev
	PostgreSQL	.nppev
	SQLite	.nlpev
备份	MySQL、PostgreSQL 及 SQLite	压缩 (.psc)、未经压缩 (.psb)
备份设置文件	MySQL	.npb
	PostgreSQL	.nppb
	SQLite	.nlpb
导入向导设置文件	MySQL	.npi
	Oracle	.nopi
	PostgreSQL	.nppi
	SQLite	.nlpi
导出向导设置文件	MySQL	.npe
	Oracle	.nope
	PostgreSQL	.nppe
	SQLite	.nlpe
导出实体化视图设置文件	Oracle	.nopem
ER 图表文件	全部	.ned

其他文件均位于 **profiles** 目录。要查找路径，选择工具 -> 选项 -> 其他 -> 设置文件保存路径。

其他设置文件	服务器类型	扩展名
数据传输	MySQL	.npt
	Oracle	.nopt
	PostgreSQL	.nppt
	SQLite	.nlpt
	Premium（跨服务器）	.napt
数据同步	MySQL	.npd
	Oracle	.nopd
	PostgreSQL	.nppd
	SQLite	.nlpd
结构同步	MySQL	.nps
	Oracle	.nops
	PostgreSQL	.npps
	SQLite	.nlps
批处理作业	Premium（跨服务器）	.napj
模型文件	全部	.ndm
虚拟组	全部	vgroup.xml - 保存如何分类对象。

提示： 你的全部连接设置都保存在注册表。

参见：

日志文件

MySQL

编码

当使用 MySQL 字符集没有使用，选择一个代码页来和 MySQL 服务器沟通。

保持连接间隔（秒）

此选项允许你通过 ping 来保持连接。你可以在编辑栏位设置 ping 与 ping 之间的间隔。

使用 MySQL 字符集

如果使用 MySQL 4.1 或以上的版本，应该启用此选项。

使用压缩

此选项让你使用压缩通讯协定。使用它当客户端和服务端支持 zlib 压缩，以及客户端需要压缩。

自动连接

勾选了这个选项，当程序启动时，Navicat 会自动打开连接和显示登记的数据库。

使用名称管道、套接字

勾选了这个选项，Navicat 使用本机连接的套接字文件。

Oracle

角色

指示数据库用户用 **Default**、**SYSOPER** 或 **SYSDBA** 系统权限连接。

作业系统验证

勾选了这个选项，Oracle 数据库使用 Windows 用户登入证书来验证数据库用户。

自动连接

勾选了这个选项，当程序启动时，Navicat 会自动打开连接和显示登记的数据库。

PostgreSQL

保持连接间隔 (秒)

此选项允许你通过 ping 来保持连接。你可以在编辑栏位设置 ping 与 ping 之间的间隔。

自动连接

勾选了这个选项，当程序启动时，Navicat 会自动打开连接和显示登记的数据库。

SQLite

自动连接

勾选了这个选项，当程序启动时，Navicat 会自动打开连接和显示登记的数据库。

已加密

当连接到已加密的 SQLite 数据库，启用这个选项并提供 **密码**。

已附加的数据库

在连接中附加或分离数据库。

设置高级数据库属性（只限于 MySQL 及 PostgreSQL）

设置高级数据库属性，这不是强制性的。要开始使用高级数据库设置，勾选 **使用高级连接**。详细介绍如下：

显示已选择的数据库

在导览窗格中显示已选择的数据库处于 **关闭** 状态

- 在数据库列表中点击合意的数据库，核取钮将显示为

在导览窗格中显示已选择的数据库处于 **打开** 状态

- 在数据库列表中双击合意的数据库，核取钮将显示为

加入隐藏的数据库

加入一个隐藏的数据库

- 点击 **加入数据库到列表** 按钮。
- 输入数据库名。
- 在数据库列表中选择新加入的数据库。
- 输入你的互联网服务供应商提供的 **用户名** 和 **密码**。



移除数据库

移除一个数据库

- 在数据库列表中选择要移除的数据库。
- 点击 **从列表中移除数据库** 按钮。

注意：数据库只从数据库列表中移除，它仍然存在于服务器中。

与数据库或模式工作

在你创建连接后，你的数据库或模式会出现在左边的导览窗格。如果在选项窗口勾选 **在连接树中显示对象** 选项，全部数据库或模式对象也会显示在窗格。要连接到一个数据库或模式，只需在窗格中双击它。如果连接成功，该数据库或模式的节点会变成  或 ，并展开显示其对象的树。

在 **Oracle** 服务器，当你创建一个用户时，你也同时创建那个用户的模式。模式是一个用户创建数据库对象（例如表、视图、触发器、等）的逻辑容器。模式名是和用户名一样，并可以明确指由用户拥有的对象。



其他用户模式显示在  **模式** 下面。

MySQL 数据库管理

Navicat 提供全部管理及导览数据库所需的工具。请注意，要在 Navicat 中开始数据库的工作，你需要创建连接。

创建数据库

创建一个数据库

- 在导览窗格中双击以打开连接。
- 在已打开的连接上右击并选择  新建数据库...。
- 或
- 在现有的数据库上右击并选择  新建数据库...。
- 填写所需的信息：

输入数据库名

设置新的数据库名。

字符集

字符集指定默认的数据库字符集。

排序规则

排序规则指定默认的数据库排序规则。

删除数据库


删除一个数据库

- 在导览窗格中的数据库上右击并选择  删除数据库。
- 在对话框确认删除。


注意：此操作是不可逆转的。

打开数据库

打开一个隐藏数据库

- 在导览窗格中双击以打开连接。
- 在已打开的连接上右击并选择  打开数据库。
- 输入数据库名。

打开一个显示在导览窗格的数据库

- 在导览窗格中双击以打开数据库。
或
- 在数据库上右击并选择  打开数据库。

关闭数据库

关闭一个数据库

- 在导览窗格中的数据库上右击并选择  关闭数据库。

编辑数据库

编辑一个数据库

- 在导览窗格中的数据库上右击并选择  数据库属性...。

请注意，这个时候 MySQL 不支持通过它的介面重命名数据库。

访问保存数据库的目录。默认情况下，全部数据库保存于 MySQL 安装文件夹内一个名为 **data** 的目录。
例如：[/var/lib/mysql/](#)。


注意：你必须停止 MySQL，然后才能重命名数据库。

Oracle 模式管理

在 Navicat 中开始模式的工作，你需要创建连接。

打开模式

打开一个显示在导览窗格的模式

- 在导览窗格中双击以打开模式。
或
- 在模式上右击并选择  打开模式。

关闭模式

关闭一个模式



- 在导览窗格中的模式上右击并选择  关闭模式。

PostgreSQL 数据库管理

在 Navicat 中开始模式的工作，你需要创建连接。


创建数据库

创建一个数据库

- 在导览窗格中双击以打开连接。
- 在已打开的连接上右击并选择  新建数据库...。
- 或
- 在现有的数据库上右击并选择  新建数据库...。
- 在数据库设计器中适当的选项卡编辑数据库属性。


编辑数据库

编辑一个现有的数据库（管理它的常规等）

- 在导览窗格中的数据库上右击并选择  数据库属性...。
- 在数据库设计器中适当的选项卡编辑数据库属性。


删除数据库

删除一个数据库


- 在导览窗格中的数据库上右击并选择  删除数据库。
- 在对话框中确认删除。

打开数据库

打开一个隐藏数据库


- 在导览窗格中双击以打开连接。
- 在已打开的连接上右击并选择  打开数据库。
- 输入数据库名。

打开一个显示在导览窗格的数据库

- 在导览窗格中双击以打开数据库。
- 或
- 在数据库上右击并选择  打开数据库。

关闭数据库

关闭一个数据库

- 在导览窗格中的数据库上右击并选择  关闭数据库。

PostgreSQL 数据库设计器

数据库设计器 是与数据库工作的 Navicat 基本工具。它允许你创建新的数据库和编辑现有的数据库属性。

- [编辑数据库常规设置](#)
- 编辑数据库注释

编辑 PostgreSQL 数据库常规设置

要创建一个数据库，你一定要有 **可以创建数据库 (usecreatedb)** 权限。如何设置用户属性，请参照角色编辑器或用户编辑器。

数据库名

定义数据库的名。

编码

定义数据库的编码。如果省略，默认为范本数据库的编码。

拥有者

定义数据库的拥有者。如果省略，默认为运行命令的用户。只有超级用户可以创建其他用户拥有的数据库。

范本

由一个范本数据库创建数据库。

注意：在复制操作的期间，源数据库必须是空闲的（没有数据修改事务进行中）。**CREATE DATABASE** 将检查在开始操作时没有阶段作业（除本身）连接到源数据库。但这并不保证当复制进行时不会有改变，导致复制的数据库不一致。因此，建议视用作范本的数据库为只读。

表空间



定义数据库的表空间。如果省略，默认为 `pg_default`。

SQLite 数据库管理

Navicat 提供全部管理及导览数据库所需的工具。请注意，要在 Navicat 中开始数据库的工作，你需要创建连接。

附加数据库

附加一个数据库

- 在导览窗格中双击以打开连接。
- 在已打开的连接上右击并选择  附加数据库。
- 或
- 在现有的数据库上右击并选择  附加数据库。
- 填写所需的信息：

数据库文件

设置数据库的文件路径。

数据库名


输入 Navicat 显示的数据库名。

已加密

启用此选项，并提供连接到已加密的 SQLite 数据库的 密码。

打开数据库

打开一个显示在导览窗格的数据库

- 在导览窗格中双击以打开数据库。
- 或
- 在数据库上右击并选择  打开数据库。

分离数据库

分离一个数据库

- 在导览窗格中的数据库上右击并选择  分离数据库。

加密数据库

加密一个数据库

- 在导览窗格中的数据库上右击并选择 **加密数据库**。
- 输入密码。

解密数据库

解密一个数据库

- 在导览窗格中的数据库上右击并选择 **解密数据库**。
- 在对话框确认解密。

关闭数据库

关闭一个数据库

- 在导览窗格中的数据库上右击并选择  **关闭数据库**。

与数据库或模式对象工作

你可以透过导览窗格的节点访问你已选择数据库的数据库或模式对象。根据你连接到的服务器版本，支持的对象将会出现在对象树。要打开特定的对象，右击来打开适当的编辑器。在对象上右击来显示弹出菜单，让你在已选择的对象或数据库运行各种操作。

查看对象信息

要查看对象的信息，只需简单地在导览窗格或对象窗格选择一个对象并点击 **查看** -> **对象信息** 或在对象窗格中的对象上右击并选择 **对象信息** 来打开 **对象信息**。

- **常规**
以网格显示对象的信息。
- **DDL**
显示对象的 DDL 语句。
- **使用**
显示当前对象使用的对象。
注意：只限于 Oracle 及 PostgreSQL。
- **被使用**
显示哪个对象使用当前的对象。
注意：只限于 Oracle 及 PostgreSQL。
- **对象**
显示对象的表空间。
注意：只限于 Oracle 及 PostgreSQL。
- **预览**
显示查询的 sql 语句。
- **成员属于**
显示用户或角色分配到的角色。
注意：只限于 Oracle 及 PostgreSQL。
- **成员**
显示角色的成员。
注意：只限于 Oracle 及 PostgreSQL。
-