



易鲸捷 Designer 用户指南 3.0.0

2020 年 12 月

版权

© Copyright 2015-2020 贵州易鲸捷信息技术有限公司

公告

本文档包含的信息如有更改，恕不另行通知。

保留所有权利。除非版权法允许，否则在未经 Esgyn 预先书面许可的情况下，严禁改编或翻译本手册的内容。Esgyn 对于本文中所包含的技术或编辑错误、遗漏概不负责。

Esgyn 产品和服务附带的正式担保声明中规定的担保是该产品和服务享有的唯一担保。本文中的任何信息均不构成额外的保修条款。

声明

Microsoft® 和 Windows® 是美国微软公司的注册商标。Java® 和 MySQL® 是 Oracle 及其子公司的注册商标。Bosun 是 Stack Exchange 的商标。Apache®、Hadoop®、HBase®、Hive®、openTSDB®、Sqoop® 和 Trafodion® 是 Apache 软件基金会的商标。Esgyn 和 EsgynDB 是 Esgyn 的商标。

目录

目录.....	i
前言.....	vii
本文简介	vii
目标读者	vii
修订历史	vii
符号约定	ix
相关文档	xii
1. 简介.....	1
2. 安装与卸载	2
2.1 安装准备.....	3
2.1.1 软件环境	3
2.1.2 硬件环境	5
2.1.3 安装文件	6
2.2 安装易鲸捷 Designer	7
2.3 卸载易鲸捷 Designer	8
3. 界面概览.....	9
3.1 菜单栏.....	10
3.2 工具栏.....	11
3.3 快捷键栏.....	12
3.4 状态栏.....	12
3.5 工作区（编辑器和视图）	13
3.5.1 更改工作区布局	13
4. 数据库连接	15
4.1 如何创建数据库连接.....	16
4.2 如何编辑数据库连接.....	23

4.3	如何断开数据库连接.....	25
4.4	如何删除数据库连接.....	27
4.5	如何设置默认 CQD	28
5.	SQL 编辑器.....	31
5.1	界面概览.....	32
5.1.1	更改界面布局	33
5.1.1.1	设置水平/垂直布局	33
5.1.1.2	更改面板大小.....	34
5.1.1.3	最大化结果面板.....	35
5.1.1.4	隐藏面板.....	36
5.1.2	语法高亮	37
5.1.3	超链接	37
5.1.4	切换数据源/Catalog/Schema	38
5.1.4.1	切换数据源.....	38
5.1.4.2	切换 Catalog/Schema	39
5.2	SQL 脚本模板.....	40
5.2.1	查看 SQL 脚本模板.....	40
5.2.2	使用 SQL 脚本模板.....	41
5.2.3	新增/编辑/删除 SQL 脚本模板.....	42
5.2.4	导入/导出 SQL 脚本模板.....	43
5.3	SQL 脚本内容助理	44
5.4	SQL 脚本格式化	45
5.4.1	格式化 SQL 脚本.....	45
5.4.2	更改 SQL 脚本的大小写	46
5.4.3	注释 SQL 脚本.....	47
5.5	SQL 脚本管理.....	48

5.5.1	保存脚本	48
5.5.2	重命名脚本	49
5.5.3	导入脚本	49
5.6	SQL 脚本执行	50
5.6.1	如何创建 SQL 脚本	50
5.6.2	如何编辑 SQL 脚本	52
5.6.2.1	使用 SQL 编辑器的编辑选项	52
5.6.2.2	使用菜单栏的编辑选项	62
5.6.3	如何执行 SQL 脚本	63
5.6.3.1	如何执行（串行）SQL 脚本	64
5.6.3.2	如何执行（并行）SQL 脚本	69
5.6.3.3	如何取消执行 SQL	72
5.6.4	如何处理 SQL 查询结果	73
5.6.4.1	使用查询结果页面的编辑选项	73
5.6.4.2	使用日志页面的编辑选项	80
5.6.5	如何解析 SQL 查询计划	81
5.6.6	如何导出查询结果	82
5.6.6.1	如何导出查询结果至文件	82
5.6.6.2	如何导出查询结果至数据库表	89
5.6.6.3	如何导出部分结果集	95
5.6.7	如何导出表数据至 SQL 文件	95
5.6.7.1	如何导出表数据至 SQL 文件	96
6.	表管理	98
6.1	如何创建表	99
6.2	如何查看表	111
6.2.1	如何查看表的属性	111

6.2.2	如何查看表的数据	113
6.2.3	如何查看表的 ER 图	115
6.3	如何编辑表.....	116
6.3.1	如何编辑表的属性	116
6.3.2	如何编辑表的数据	121
6.3.3	如何编辑表的 ER 图	125
6.4	如何删除数据库对象.....	127
6.4.1	如何删除表	127
6.4.2	如何删除索引	128
6.4.3	如何删除序列	129
6.4.4	如何删除视图	130
6.4.5	如何删除约束	131
6.4.5.1	如何删除外键.....	132
6.4.5.2	如何删除 CHECK 约束.....	134
6.4.5.3	如何删除 UNIQUE 约束	136
6.4.5.4	如何删除 NOT NULL 约束.....	138
6.5	如何导出表的数据.....	139
6.5.1	如何导出表的数据至文件	140
6.5.2	如何导出表的数据至数据库表	142
6.6	如何导入表的数据.....	143
7.	存储过程管理	162
7.1	如何创建存储过程.....	163
7.1.1	方法 1: 通过 SQL 语句创建.....	163
7.1.2	方法 2: 通过数据库导航窗口创建	164
7.2	如何查看存储过程.....	166
7.3	如何编辑存储过程.....	167

7.3.1	方法 1: 通过 SQL 语句编辑.....	167
7.3.2	方法 2: 通过数据库导航窗口编辑	168
7.4	如何调用存储过程.....	170
7.5	如何删除存储过程.....	171
7.5.1	方法 1: 通过 SQL 语句进行删除。	171
7.5.2	方法 2: 通过数据库导航窗口删除。	172
8.	函数管理.....	173
8.1	如何创建函数.....	174
8.1.1	方法 1: 通过 SQL 语句创建.....	174
8.1.2	方法 2: 通过数据库导航窗口创建	175
8.2	如何查看函数.....	177
8.3	如何编辑函数.....	178
8.3.1	方法 1: 通过 SQL 语句进行编辑.....	178
8.3.2	方法 2: 通过数据库导航窗口编辑	179
8.4	如何调用函数.....	181
8.5	如何删除函数.....	182
8.5.1	方法 1: 通过 SQL 语句进行删除.....	182
8.5.2	方法 2: 通过数据库导航窗口删除	182
9.	视图管理.....	184
9.1	如何创建视图.....	185
9.1.1	方法 1: 通过 SQL 语句创建.....	185
9.1.2	方法 2: 通过数据库导航窗口创建	186
9.2	如何编辑视图.....	188
9.3	如何删除视图.....	189
9.3.1	方法 1: 通过 SQL 语句进行删除.....	189
9.3.2	方法 2: 通过数据库导航窗口删除	189

10. 触发器管理	191
10.1 如何创建触发器.....	192
10.1.1 方法 1: 通过 SQL 语句创建.....	192
10.1.2 方法 2: 通过数据库导航窗口创建	193
10.2 如何编辑触发器.....	196
10.3 如何删除触发器.....	198
10.3.1 方法 1: 通过 SQL 语句进行删除.....	198
10.3.2 方法 2: 通过数据库导航窗口删除	198
11. 包管理.....	200
11.1 如何创建包.....	201
11.2 如何编辑包.....	202
11.3 如何删除包.....	203
12. 事务.....	204
12.1 提交模式.....	205
12.1.1 自动提交模式	205
12.1.2 手动提交模式	205
12.2 事务隔离等级.....	206
12.3 事务日志.....	206
13. 注意事项.....	207
附录.....	209
附录 1. 数据类型映射规则	209
附录 2. HBase 选项及其有效值和默认值.....	210

前言

本文简介

本指南介绍如何使用易鲸捷 Designer。

目标读者

本指南的目标读者为易鲸捷 Designer 管理员和用户。

修订历史

版本	日期	说明
3.0.0	2020 年 12 月	<ol style="list-style-type: none">1.增加第 13 章注意事项2.增加 5.6.6.3 如何导出部分结果集3.增加 5.6.3.3 如何取消执行 SQL4.在 2.1 安装准备增加日志和缓存文件位置提示5.在 2.3 卸载易鲸捷 Designer 中增加日志和缓存文件位置提示6.在 4.1 如何创建数据库连接中增加提示, URL 模板不可编辑。
2.2.0	2020 年 7 月	<ol style="list-style-type: none">1.增加 9 视图管理2.增加 10 触发器管理3.增加 11 包管理4.增加表管理中 Range 的用法5.增加表管理中列字符集和字符单元的设置6.存储过程和函数的创建修改删除增加界面操作

		7.增加 4.5 如何设置默认 CQD
2.0.8	2019 年 6 月	<ol style="list-style-type: none"> 1. 修改 4.1 中的添加 JDBC 驱动部分 2. 增加 5.6.7 如何导出表数据至 SQL 文件 3. 增加 7.存储过程管理 4. 增加 8.函数管理
2.0.7	2019 年 3 月	
2.0.6	2018 年 10 月	

符号约定

描述语法时，本指南采用以下符号约定。

- 大写字母

表示关键字和保留字。未在方括号中的内容是必选的语法项。



示例

```
SELECT
```

- 小写字母

表示变量。未在方括号中的内容是必选的语法项。



示例

```
file-name
```

- [] 方括号

表示可选的语法项。



示例

```
DATETIME [start-field TO] end-field
```

如果方括号中包括多个语法项，您可以选择某一语法项或不选。

多个语法项可以垂直排列，每个语法选项用方括号括起；也可以水平排列，用竖线隔开。



示例

```
DROP SCHEMA schema [ CASCADE ][ RESTRICT ]  
DROP SCHEMA schema [ CASCADE | RESTRICT ]
```

- {} 大括号

表示必选的语法项。



示例

```
FROM { grantee [, grantee ] ... }
```

如果大括号中包括多个语法选项，您必须选择某一语法项。

多个语法项可以垂直排列，每个语法项用大括号括起；也可以水平排列，用竖线隔开。



示例

```
INTERVAL { start-field TO end-field } { single-field }  
INTERVAL { start-field TO end-field | single-field }
```

- | 竖线

隔开方括号或大括号中的多个语法项。



示例

```
{expression | NULL}
```

- ... 省略号

。紧跟在方括号或大括号之后，表示括号内的语法项可以重复任意次。



示例

```
ATTRIBUTE[S] attribute [, attribute] ...  
{, sql-expression } ...
```

。紧跟在单个语法项之后，表示该语法项可以重复任意次。



示例

```
expression-n ...
```

- 标点符号

- 圆括号、逗号、分号和其它符号，请参照以下示例输入。



示例

```
DAY (datetime-expression)
@script-file
```

- 符号（例如，方括号或大括号）周围的引号表示该符号是必选项。



示例

```
"{" module-name [, module-name] ... }"
```

- 语法项的间距

- 语法项之间必须有空格。如果某一语法项是圆括号或逗号，则不需要空格。



示例

```
DAY (datetime-expression) DAY(datetime-expression)
```

- 如果语法项之间没有空格，则不允许空格。在以下示例中，句点与任何其他语法项之间不允许有空格。



示例

```
myfile.sh
```

- 行距

如果一条命令过长（超过一行），则每个连续行需缩进三个空格，并通过空行与前一行分隔。

行距用于区分连续行中的语法项与垂直列表（多个语法项）中的项。



示例

```
match-value [NOT] LIKE _pattern
    [ESCAPE esc-char-expression]
```

相关文档

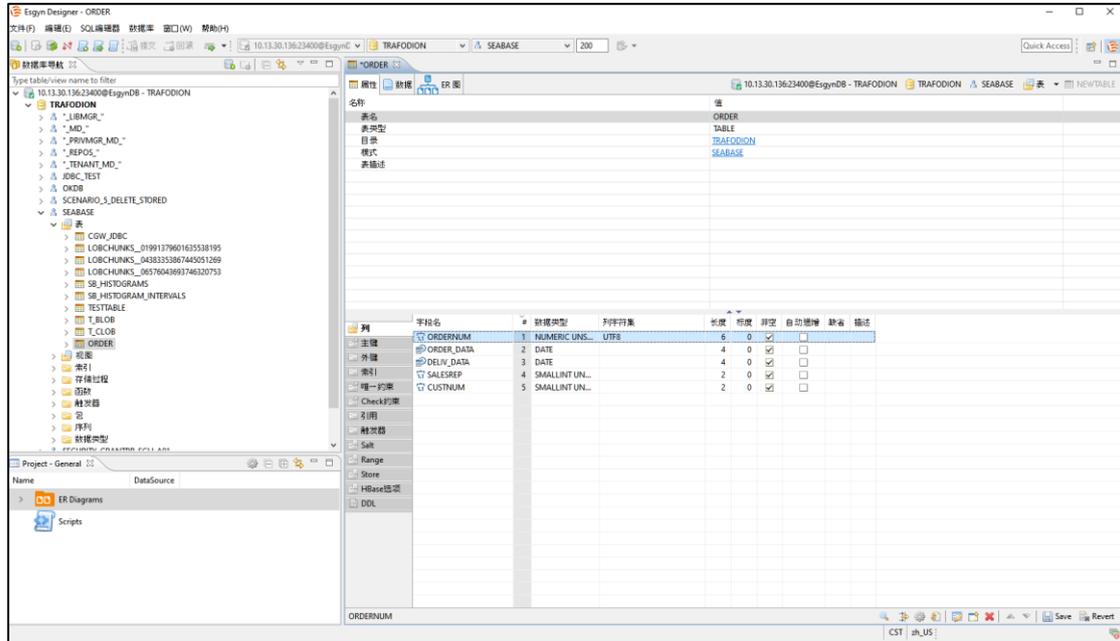
本指南为 EsgynDB 文档库的一部分，EsgynDB 文档库包括但不限于以下文档：

文档名称	说明
《EsgynDB ADO.NET Provider 手册》	本文介绍如何使用 EsgynDB ADO.NET Provider 访问 EsgynDB。
《EsgynDB 备份与恢复指南》	本文介绍 EsgynDB 时间点恢复功能。
《EsgynDB CQD 参考手册》	本文介绍控制查询默认 (Control Query Default) 设置，它能更改当前编译器会话环境，影响查询执行计划。
《EsgynDB 安装指南》	本文介绍安装 EsgynDB，包括安装前准备、安装 Hadoop 发行版、故障排除、配置、启用安全功能、提高安全性和卸载 EsgynDB 等。
《Esgyn DBManager 用户指南》	本文介绍网页管理工具 Esgyn DBManager，包括仪表盘、工作负载、日志、数据库、连接服务、查询工作台、安全、租户、警报和工具等。
《EsgynDB Windows ODBC 驱动安装指南》	本文介绍如何在 Window 系统安装 EsgynDB ODBC 驱动。
《EsgynDB 客户端安装指南》	介绍如何安装和配置各类客户端应用程序，例如，Trafodion 命令行界面 (TrafCI)、Trafodion odb、DbVisualizer、Squirrel SQL 客户端和 JDBC/ODBC 驱动程序，方便用户连接并使用 EsgynDB。
《EsgynDB SQL 手参考册》	本文介绍 EsgynDB SQL 语句、实用程序、函数和其它 SQL 语言元素的语法。

《Trafodion 命令行界面指南》	本文介绍 Trafodion 命令行界面 (TrafCI), 该界面允许用户连接至 EsgynDB, 交互式地或通过脚本访问 EsgynDB。
《EsgynDB 连接服务参考指南》	本文介绍如何对 EsgynDB 连接服务 (Database Connectivity Service, DCS) 进行配置、调优、排故和调试, 这为基于 ODBC/JDBC API 开发的应用程序提供了访问 EsgynDB SQL 数据库服务器的框架; 另外, 本文还介绍了 DCS API。
《Trafodion odb 用户指南》	本文介绍如何配置并使用 Trafodion odb 工具, 该工具是一个多线程的 ODBC 命令行工具, 能对数据执行 Trickle Load。
《Trafodion REST 服务器参考指南》	本文介绍如何启动、关闭、配置、排故和调试 Trafodion REST (Representational State Transfer) 服务器; 另外, 本文简要介绍了 REST API。
《EsgynDB 多数据中心复制指南》	本文介绍如何使用 EsgynDB 同步复制功能和 xdc 工具等。
《EsgynDB 迁移工具用户指南》	本文介绍如何安装和使用 EsgynDB 迁移工具。
《EsgynDB 傻瓜手册系列— —如何使用 EsgynDB? 》	本文介绍如何快速了解并使用 EsgynDB。
《EsgynDB 傻瓜手册系列— —如何集成 EsgynDB 与 Zeppelin? 》	本文介绍如何安装、配置和使用 Zeppelin。

1. 简介

易鲸捷 Designer 是基于 C/S 模式¹的通用数据库管理工具，它基于 Eclipse RCP 框架，使用 Java 编写，通过 JDBC 驱动程序与数据库进行交互。



易鲸捷 Designer 提供多种功能，例如：

- 创建/编辑/执行/管理 SQL 脚本
- 创建/编辑/管理数据库连接
- 创建/编辑/管理数据库对象（例如，Schema、表、视图、索引、存储过程、函数、包和序列等）
- 导入/导出数据

¹ 即客户端/服务器模式。

2. 安装与卸载

本章讲述以下内容：

2.1 安装准备

2.2 安装易鲸捷 Designer

2.3 卸载易鲸捷 Designer

2.1 安装准备

在安装 易鲸捷 Designer 之前，应完成以下准备工作：

- 正确地安装操作系统、合理地分配磁盘空间以及检查机器配置是否满足要求
- 关闭正在运行的杀毒和安全防护等软件
- 保证网络环境能正常工作



注意

如果非首次安装，为了防止对安装造成干扰，务必删除安装文件和缓存文件。

如果不删除缓存文件，可能会造成新版本功能不生效的问题。缓存文件存放在操作系统为用户分配的用户目录。以 Windows10 为例，缓存文件路径为：

- C:\Users\user-name\.esgyndb_designer4
- C:\Users\user-name\.esgyndb-driver

如需保留之前的 SQL 文件，请自行拷贝备份。



注意

从 Designer 3.0.0 开始，缓存文件和日志文件位于安装目录的 workspace 目录下，即：

- ...\esgyndb_designer\workspace

2.1.1 软件环境

安装 易鲸捷 Designer 需要以下软件环境：

操作系统	<ul style="list-style-type: none">• Windows 7 (64bit)• Windows 10 (64bit)• Linux (64bit)
网络协议	TCP/IP



注意

- 易鲸捷 Designer 要求 Linux 系统必须有 2.24 及以上版本的 GTK+，因此，建议您使用 RedHat 7 和 CentOS 7 等较新版本的 Linux 系统，这类系统默认的 GTK+ 符合易鲸捷 Designer 要求。
- 系统盘最好有 1G 以上的剩余空间。客户端需安装 JDK 1.8 及以上 64bit 版本。另外，请正确配置 JAVA_HOME 环境变量，并保证 JAVA 虚拟机能正常工作。更多关于 **JDK 的安装与配置**，请参阅 https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/。

2.1.2 硬件环境

您可以根据易鲸捷 Designer 和应用系统的需求，选择合适的硬件配置，例如，CPU 指标、内存和磁盘容量等。建议采用尽可能高的配置，尤其是作为数据库服务器的机器，基于 Java 的程序运行时最好有较大的内存。

安装 易鲸捷 Designer 需要以下硬件环境：

CPU	Intel Pentium 4（建议 Pentium 4 1.6 G 以上）处理器
内存	1G（建议 2G 以上）
硬盘	5G 以上可用空间
网卡	10M 以上支持 TCP/IP 协议的网卡
光驱	32 倍速以上光驱
显卡	支持 1024*768*256 以上彩色显示
显示器	SVGA 显示器

由于易鲸捷 Designer 是基于 C/S 方式的开发工具，一般应在网络环境下使用，客户端与服务器分别在不同的机器，所以硬件环境通常包括网络环境（一个局域网）。

2.1.3 安装文件

易鲸捷 Designer 由 Java 编写，全面支持 64 位 Windows/Linux 系统，能很好地利用 64 位系统的资源（例如，充分利用更大内存），表现出良好的性能。

以下是易鲸捷 Designer 的安装文件和对应系统：

系统	安装文件 ²
Windows 64 位	esgyndb_designer-2.0.8-win32.win32.x86_64.zip
Linux 64 位	esgyndb_designer-2.0.8-linux.gtk.x86_64.tar.gz

示例

以 Windows10 为例，解压易鲸捷 Designer 安装文件后，包括以下文件：

-  configuration
-  features
-  jdbc
-  p2
-  plugins
-  .eclipseproduct
-  artifacts.xml
-  cqd-export.properties
-  eclipsesec.exe
-  esgyndb_designer.exe
-  esgyndb_designer.ini
-  readme.txt

² 如需获取易鲸捷 Designer 安装文件，请联系 Esgyn 技术支持。

2.2 安装易鲸捷 Designer

1、解压安装文件。

安装文件可以解压至任意路径，例如，D:\esgyndb_designer。

2、双击启动文件 esgyndb_designer.exe。

2.3 卸载易鲸捷 Designer

如需卸载易鲸捷 Designer，删除安装文件和缓存文件即可。

提示

缓存文件存放在操作系统为用户分配的用户目录。

以 Windows10 为例，缓存文件路径为：

- C:\Users*user-name*\.esgyndb_designer4
- C:\Users*user-name*\.esgyndb-driver

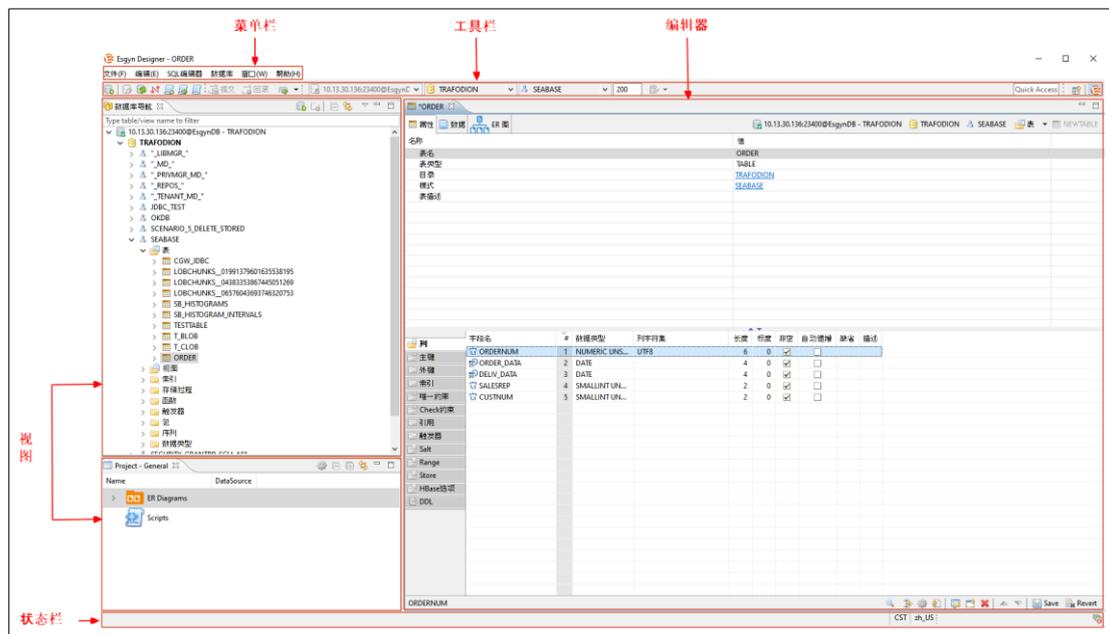
注意

从 Designer 3.0.0 开始，缓存文件和日志文件位于安装目录的 workspace 目录下，即：

- ...\esgyndb_designer\workspace

3. 界面概览

易鲸捷 Designer 界面包括菜单栏、工具栏、编辑器、快捷键栏、状态栏和视图。



本章讲述以下内容：

3.1 菜单栏

3.2 工具栏

3.3 快捷键栏

3.4 状态栏

3.5 工作区（编辑器和视图）

3.1 菜单栏

以下是菜单栏的主要功能和说明：

菜单栏功能	说明
文件	创建 SQL 编辑器、保存文件和设置属性等。
编辑	包括复制、剪切、粘贴、删除等通用命令。
SQL 编辑器	执行和管理 SQL 脚本等。
数据库	管理数据库驱动、连接和事务等。
窗口	显示数据库导航、项目浏览器和查询管理等。
帮助	显示工具版本和快捷键信息等。

3.2 工具栏

工具栏包括最基本和最常用功能的图标。

不同编辑器拥有不同的功能图标。彩色的图标是可用功能，灰色的图标是不可用功能。



以下是工具栏的主要功能和说明：

工具栏功能	说明
新建连接	更多信息，请参阅 4.1 如何创建数据库连接。
连接	
断开/重新连接	更多信息，请参阅 4.3 如何断开数据库连接。
断开连接	
新建 SQL 编辑器	更多信息，请参阅 5.6.1 如何创建 SQL 脚本。
近期 SQL 编辑器	
历史 SQL 编辑器	
提交	更多信息，请参阅 12.1 提交模式。
回滚	
提交模式	
事务隔离等级	更多信息，请参阅 12.2 事务隔离等级。
事务日志/待处理事务/查询管理器	更多信息，请参阅 12.3 事务日志。
数据源	更多信息，请参阅 5.1.4 切换数据源 /Catalog/Schema。
Catalog	
Schema	
最大化结果集	/
从导航设置连接	/

3. 界面概览

(续前表)

工具栏功能	说明
执行 (串行)	更多信息, 请 参阅 5.6.3.1 如何执行 (串行) SQL 脚本。
执行 (并行)	更多信息, 请 参阅 5.6.3.2 如何执行 (并行) SQL 脚本。
解析执行计划	更多信息, 请 参阅 5.6.5 如何解析 SQL 查询计划。
快速访问	/
打开其它界面	/
团队同步	/
资源	/

3.3 快捷键栏

快捷键栏包括编辑器和视图的快捷键。

如果存在编辑器或视图最小化, 则它 (或它们) 隐藏在快捷键栏中。

3.4 状态栏

状态栏显示任务的完成进度。

3.5 工作区（编辑器和视图）

工作区包括**编辑器**和**视图**。编辑器是您与数据库交互的界面。视图是显示信息的界面。

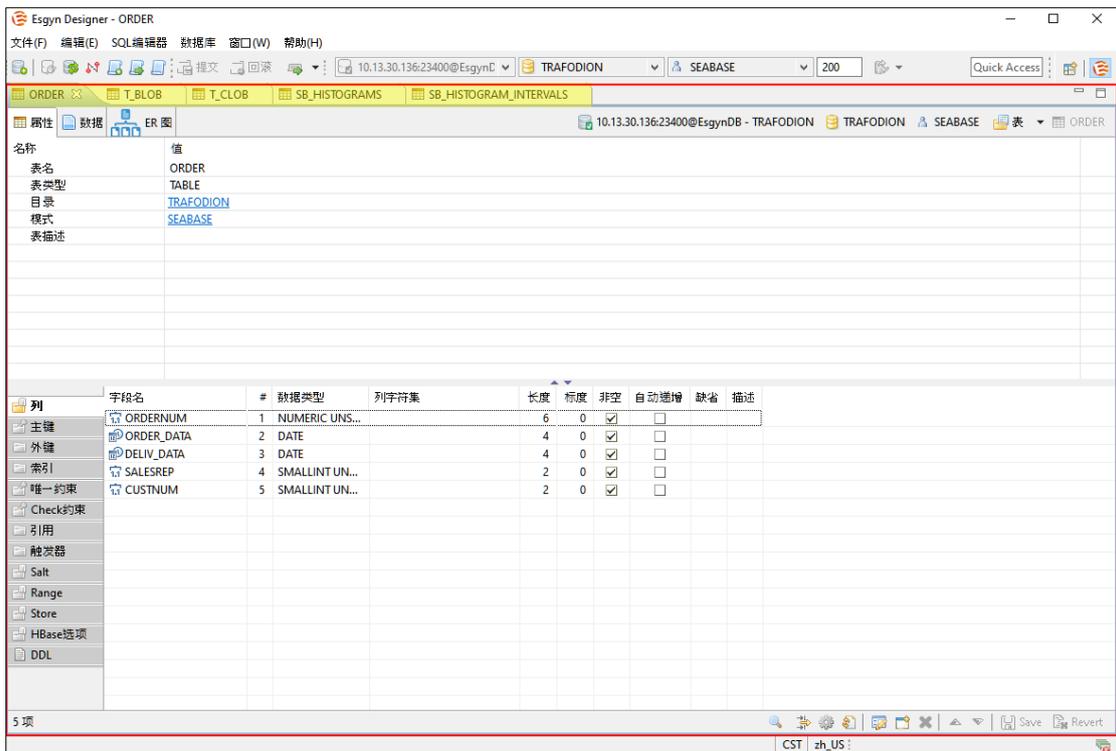
一次能打开单个或多个编辑器或视图,如果多个编辑器或视图的标签页长度大于窗口长度,则某些标签页会被隐藏。如需查看被隐藏的标签页,点击数字。



3.5.1 更改工作区布局

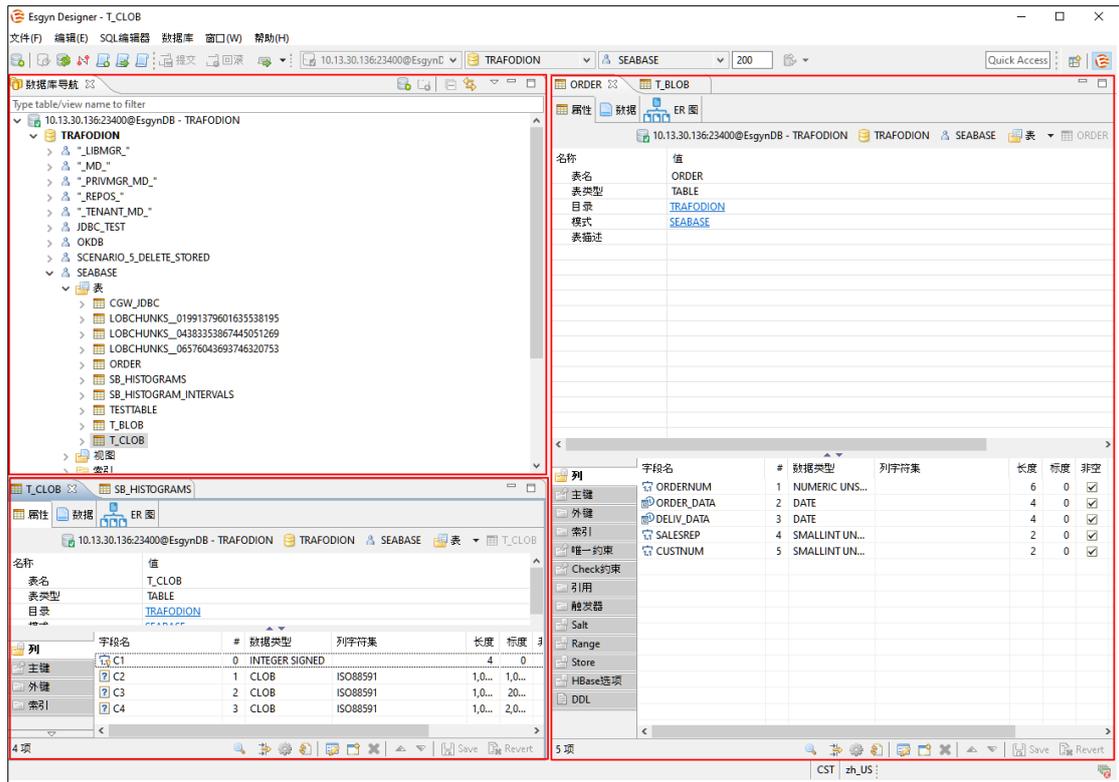
在工作区中,您可以移动编辑器或视图的位置,有以下2种布局方式:

- 多个标签页位于单个窗口中



3. 界面概览

- 多个标签页位于多个窗口中（平行/垂直）



如需**移动**编辑器或视图，点击并拖拽编辑器或视图的标签页。

如需**更改**编辑器或视图的**窗口大小**，将鼠标移至编辑器或视图的窗口边缘，出现双箭头后，点击并拖拽窗口至所需大小。

如需**关闭**编辑器或视图，点击关闭图标，或右键编辑器或视图的标签页，选择：

- 关闭
- 关闭所有标签页
- 关闭当前标签页左侧的标签页
- 关闭当前标签页右侧的标签页
- 关闭所有

4. 数据库连接

易鲸捷 Designer 通过 JDBC 驱动程序连接至数据库（支持 EsgynDB、Hive 和 Phoenix）。创建数据库连接后，您能与数据库进行交互。

本章讲述以下内容：

[4.1 如何创建数据库连接](#)

[4.2 如何编辑数据库连接](#)

[4.3 如何断开数据库连接](#)

[4.4 如何删除数据库连接](#)

[4.5 如何设置默认 CQD](#)

4.1 如何创建数据库连接

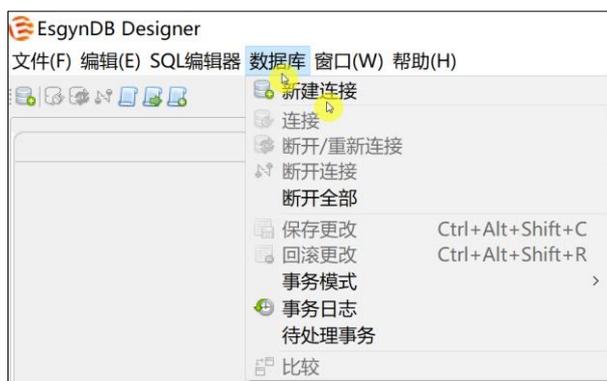
示例

本示例创建 EsgynDB 连接。

1. 打开新建连接窗口，有以下 2 种方法：

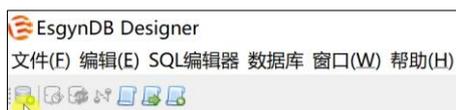
- 方法 1

在菜单栏中，点击数据库 > 新建连接。



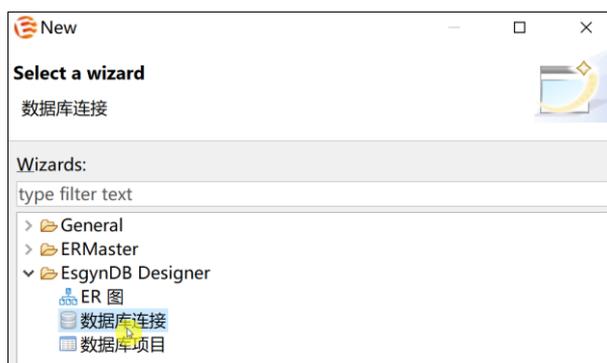
- 方法 2

在工具栏中，点击新建连接 () 图标。



- 方法 3

在易鲸捷 Designer 界面，同时按下 **Ctrl+N**。在弹出的界面中，点击数据库连接。



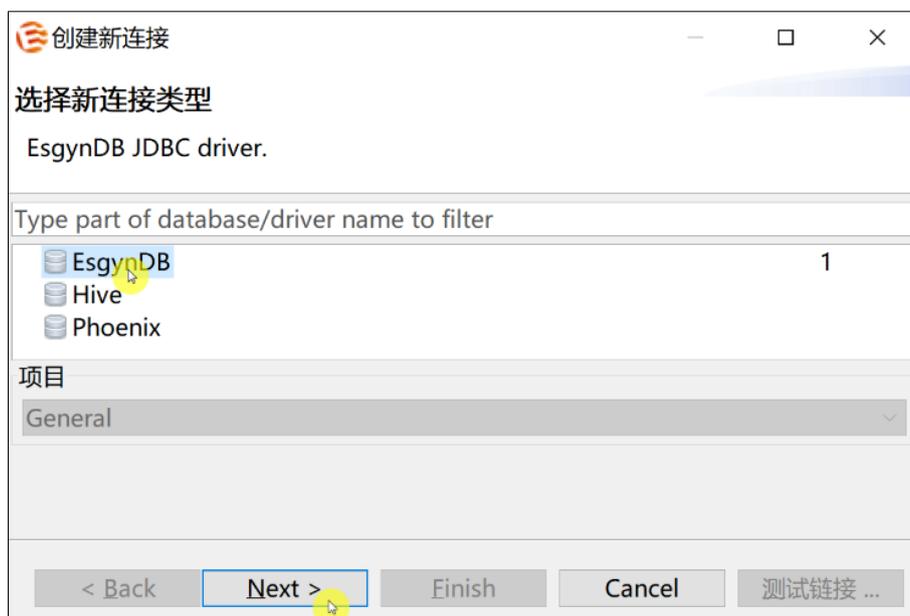
4. 数据库连接

2. 点击 **EsgynDB > 下一步**。



提示

易鲸捷 Designer 支持创建 **EsgynDB**、**Hive** 和 **Phoenix** 连接。



3. 输入连接信息，点击下一步。

连接选项	说明
JDBC URL	易鲸捷 Designer 自动填充。
主机/端口 (必选项)	<ul style="list-style-type: none">主机：输入主机地址。端口：EsgynDB 默认端口号为 23400，易鲸捷 Designer 自动填充。
数据库	易鲸捷 Designer 自动填充 TRAFODION。
租户 (可选项)	如需以租户身份登录，输入租户名。
用户名 (必选项)	数据库用户名。
密码 (必选项)	数据库密码。

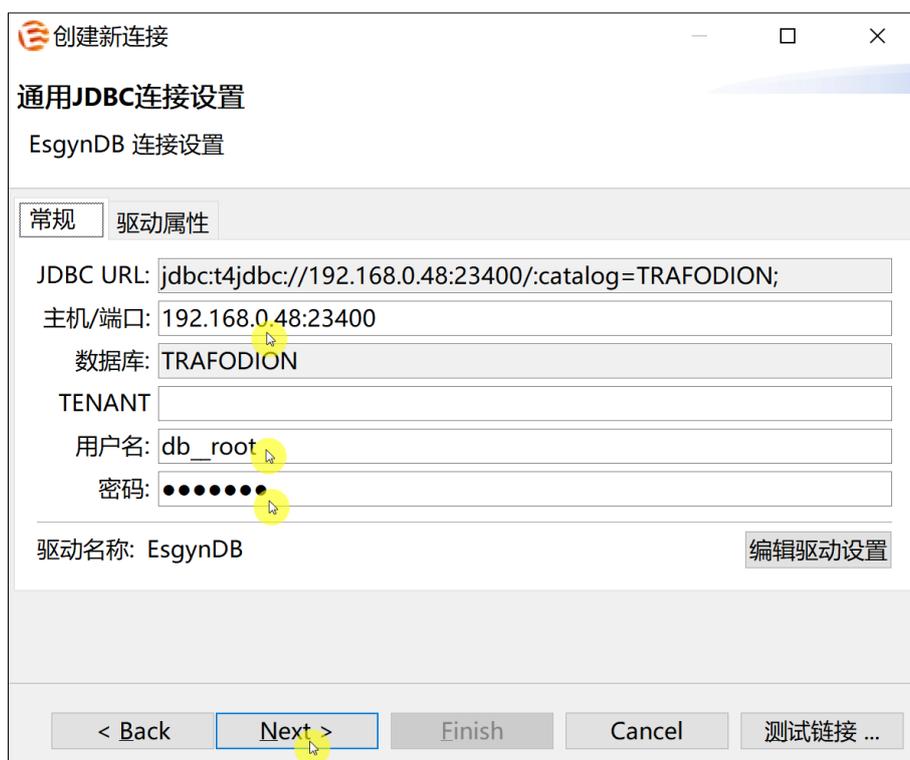
 提示

- 以上连接信息保存在

C:\Users\user-name\.esgyndb_designer4\General, 下次登录时, 易鲸捷 Designer 会自动使用这些连接信息。

- 注册后的服务器连接配置信息保存在

C:\Users\user-name\.esgyndb_designer4\General\.esgyndb-data-sources.xml 文件中。



创建新连接

通用JDBC连接设置

EsgynDB 连接设置

常规 驱动属性

JDBC URL: jdbc:t4jdbc://192.168.0.48:23400/:catalog=TRAFODION;

主机/端口: 192.168.0.48:23400

数据库: TRAFODION

TENANT

用户名: db_root

密码: ●●●●●●

驱动名称: EsgynDB [编辑驱动设置](#)

< Back **Next >** Finish Cancel 测试链接 ...

 **注意**

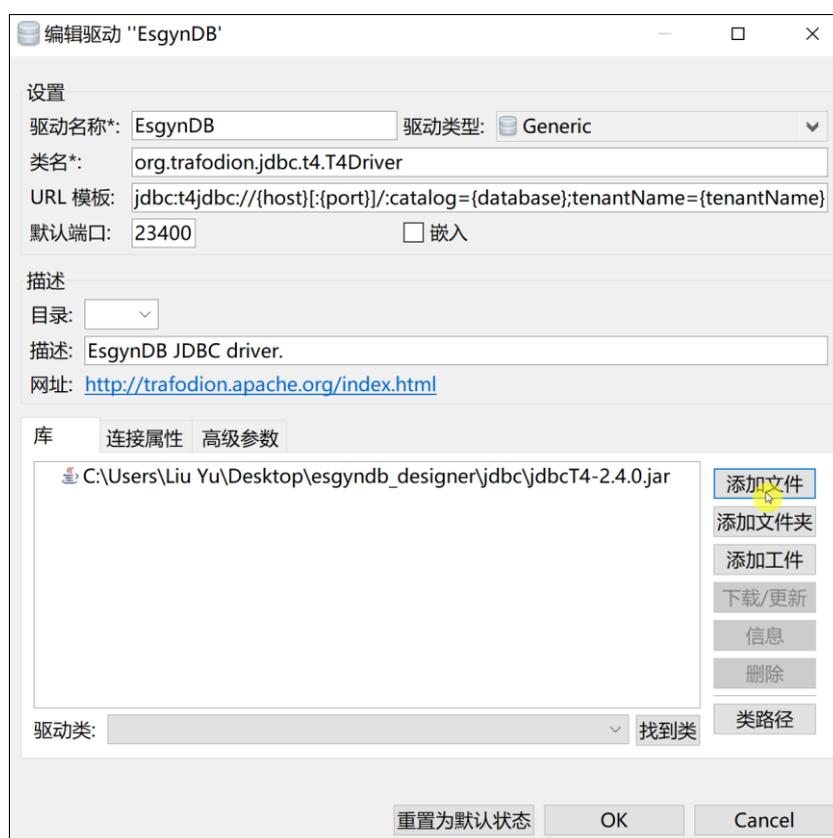
您需手动添加 JDBC 驱动，通过以下两种方法添加 JDBC 驱动：

- **方法 1**

在启动易鲸捷 Designer 前，手动把 jdbcT4-2.*.0.jar 文件放入目录 esgyndb_designer\jdbc 中。

- **方法 2**

在上图的窗口中，点击**编辑驱动设置**，在弹出的窗口中，点击**添加文件**。

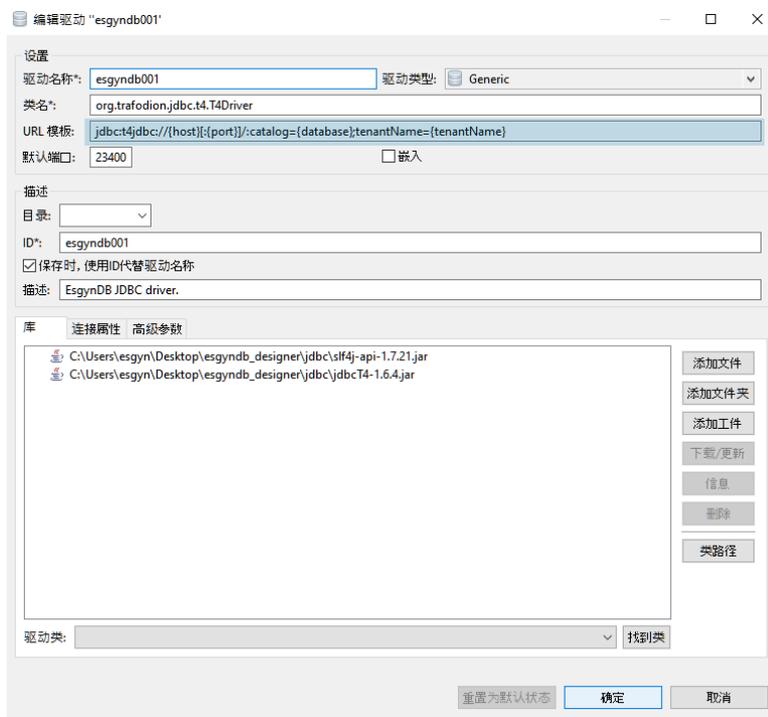


4. 数据库连接

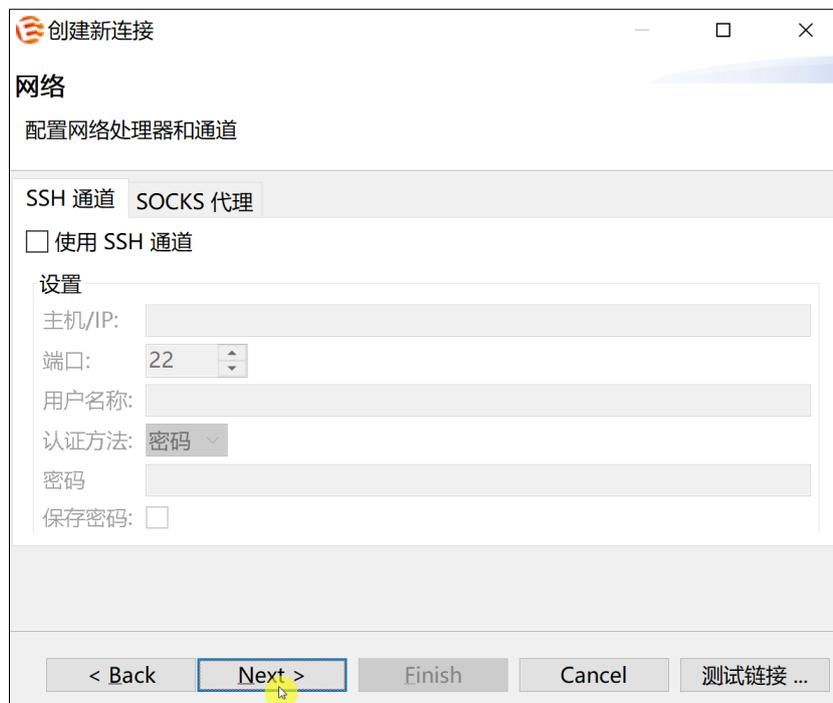


注意

从 Desginer 3.0.0 开始，用户不可以编辑 URL 模板项。

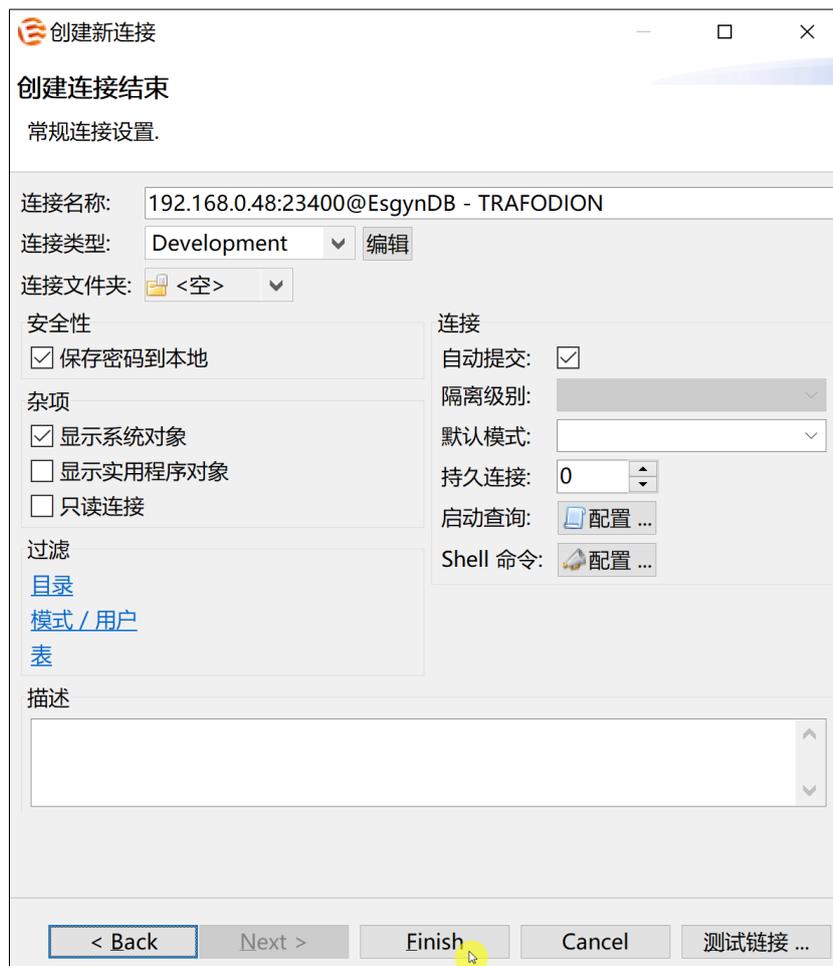


4. 接受默认设置，点击下一步。

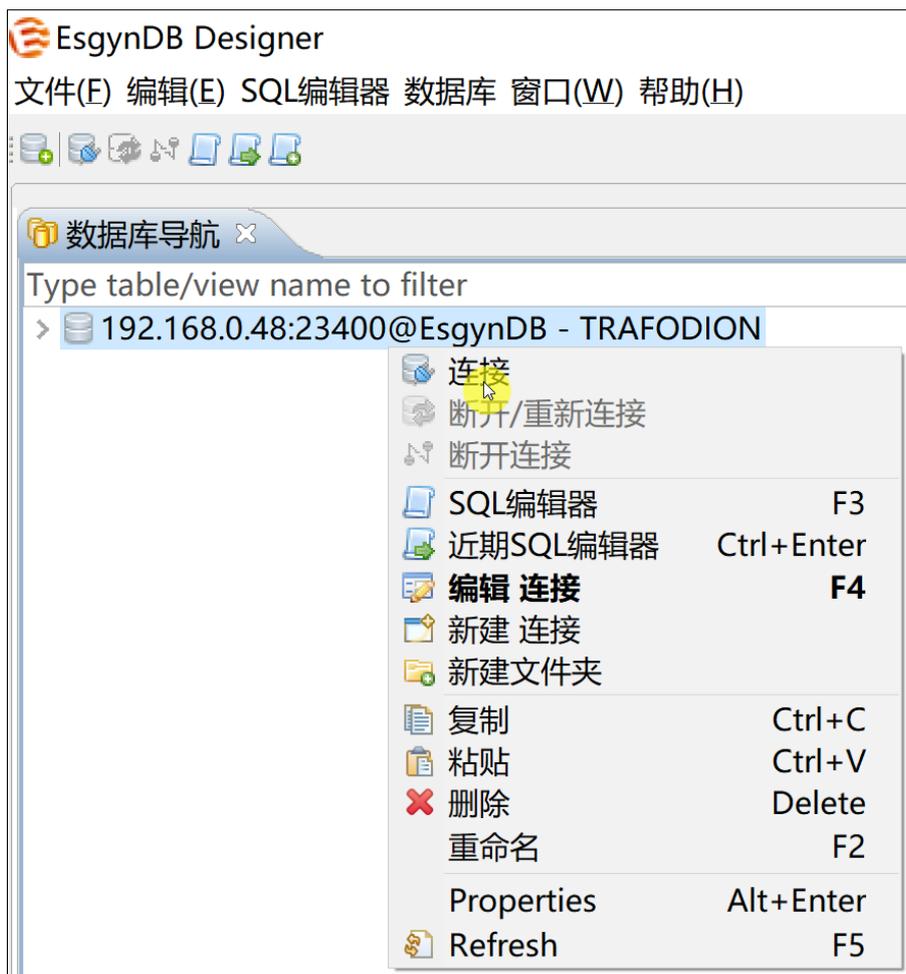


4. 数据库连接

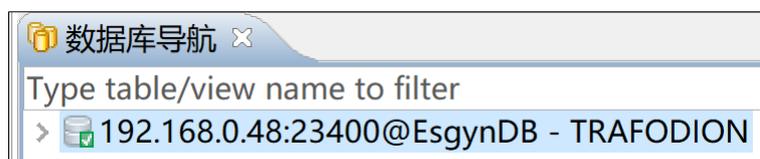
5. 接受默认设置，点击完成。



6. 在数据库导航窗口中，右键此前设置的连接，点击**连接**。



状态栏将显示连接进度。连接成功后，连接左边将显示绿色打钩 () 图标。



4.2 如何编辑数据库连接

1. 右键待编辑的连接，弹出的对话框将显示编辑选项。

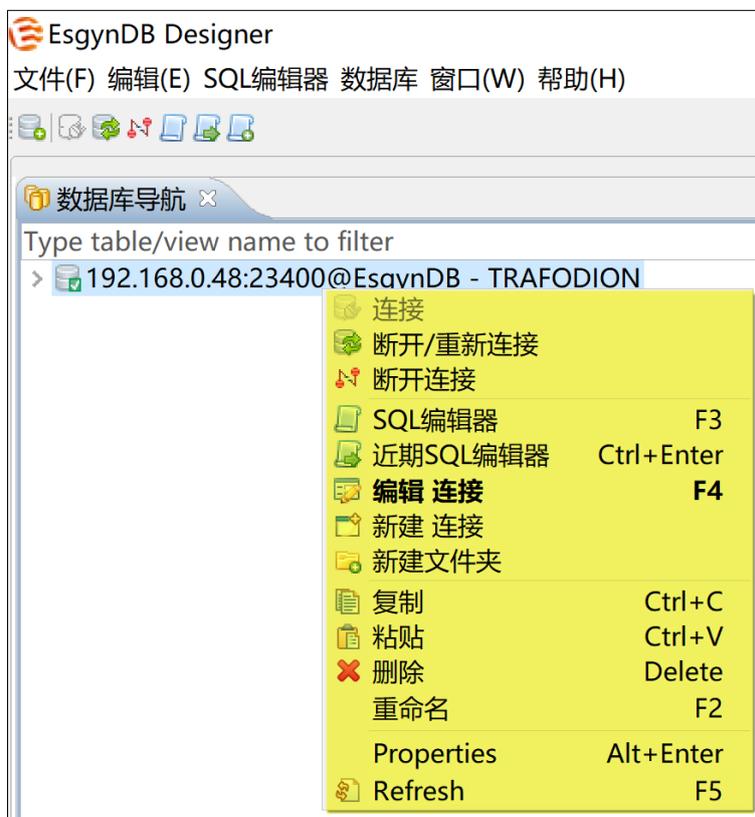


表 4-1 数据库连接 - 编辑选项 1

选项	说明	快捷键
一级选项		
连接	/	/
断开/重新连接	/	/
断开连接	/	/
SQL 编辑器	新建 SQL 编辑器	F3
近期 SQL 编辑器	打开最近的 SQL 编辑器	Ctrl + Enter
编辑连接	/	F4
新建连接	/	/
新建文件夹	为连接新建文件夹	/
复制	/	Ctrl + C
粘贴	/	Ctrl + V
删除	/	Delete
重命名	/	F2

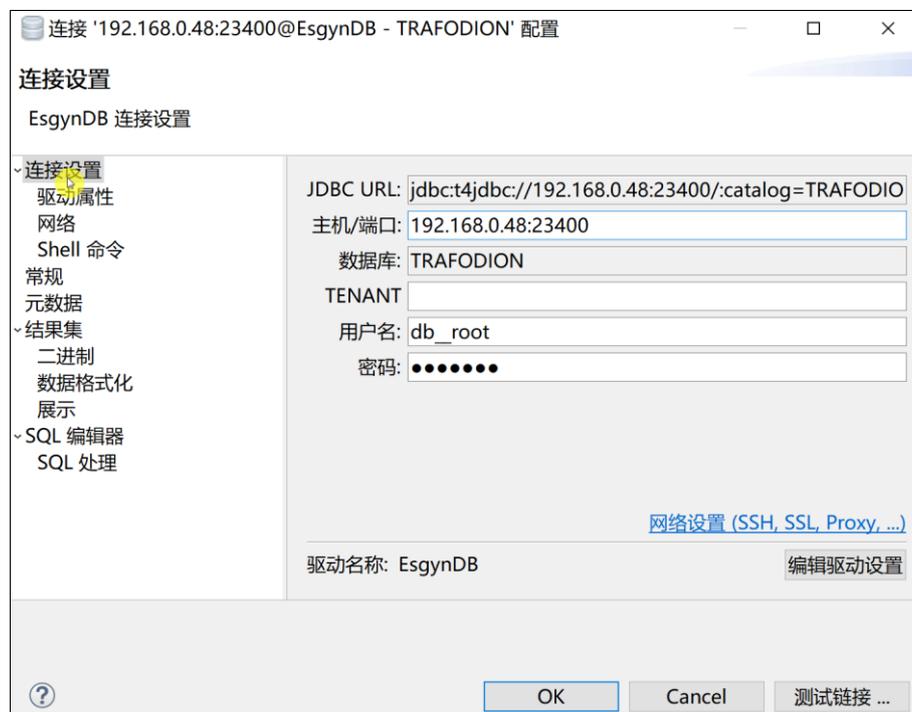
4. 数据库连接

表 4-2 (续前表) 数据库连接 - 编辑选项 2

选项			说明	快捷 键	
一级 选项	二级 选项	三级选项			
属性	SQL 编辑器	SQL 处理	请参阅 SQL 处理	Alt + Enter	
		SQL 完成/折叠	请参阅 SQL 补全		
		SQL 格式	请参阅 SQL 格式化		
		模板	请参阅 SQL 模板		
	元数据	/	请参阅 元数据		
	数据编 辑器	重启后重新打开数据库编 辑器	/		
		在编辑器打开时强制刷新 对象			
		在编辑器标题中显示对象 的完整名称			
		在编辑器保存时显示 SQL 预览对话框			
	结果集	二进制	请参阅 二进制		
		数据格式	请参阅 数据格式		
		表示	请参阅 表示		
	连接	禁用客户端身份识别	/		
		覆盖客户端名称			
		连接参数使用环境变量			
刷新				F5	

4. 数据库连接

3. 点击**编辑连接**。
4. 在弹出的对话框中，您能编辑**连接设置**（主机/端口、租户名、用户名、密码）、**驱动属性**和**网络**等设置。



5. 点击**OK**。

4.3 如何断开数据库连接

您能通过以下 3 种方法断开数据库连接：

- **方法 1**

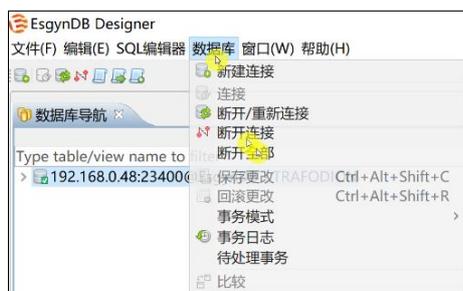
在数据库导航窗口，右键待断开的连接，点击**断开连接**。

4. 数据库连接



• 方法 2

在菜单栏中，点击**数据库 > 断开连接**。



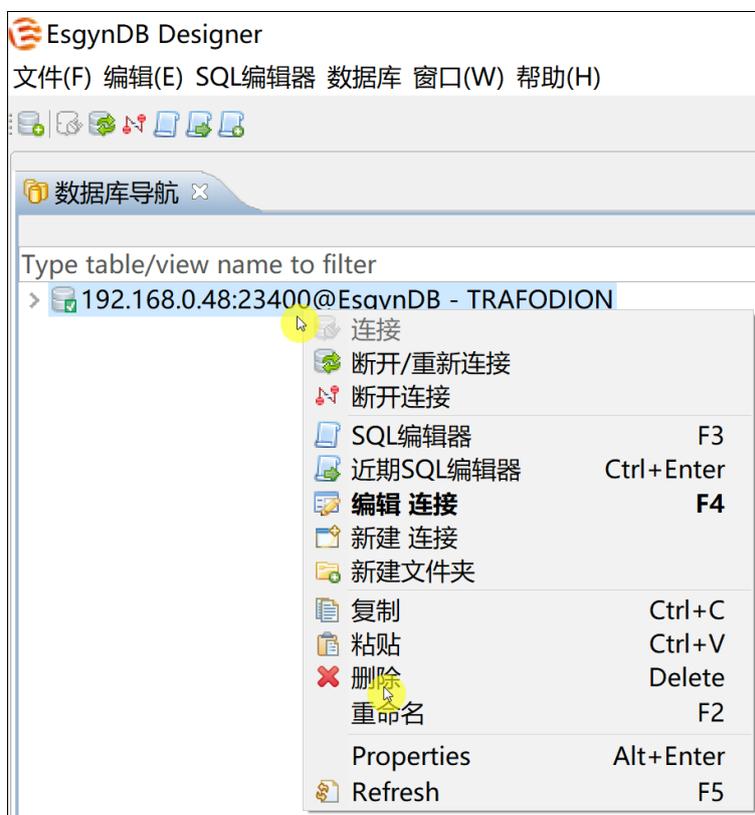
• 方法 3

在工具栏中，点击**断开连接** () 图标。



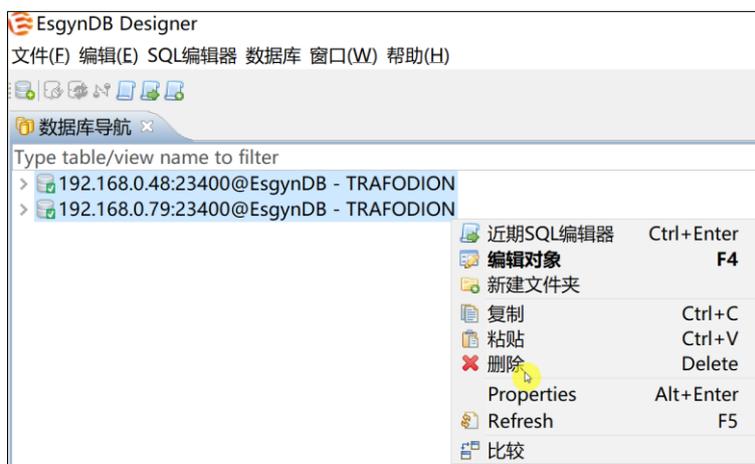
4.4 如何删除数据库连接

在数据库导航窗口，右键待删除的连接，点击删除连接。



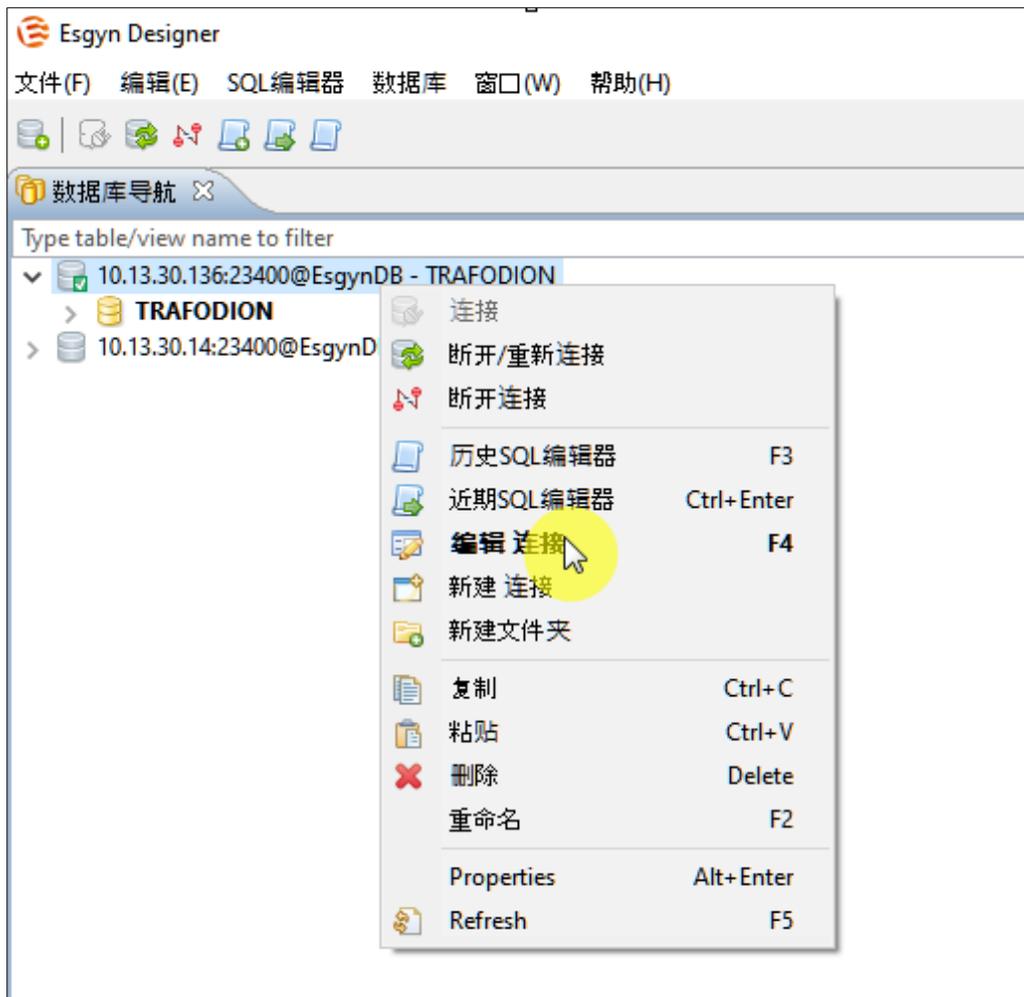
💡 提示

如需删除多个连接，选中多个连接的同时按住 shift，再点击删除。

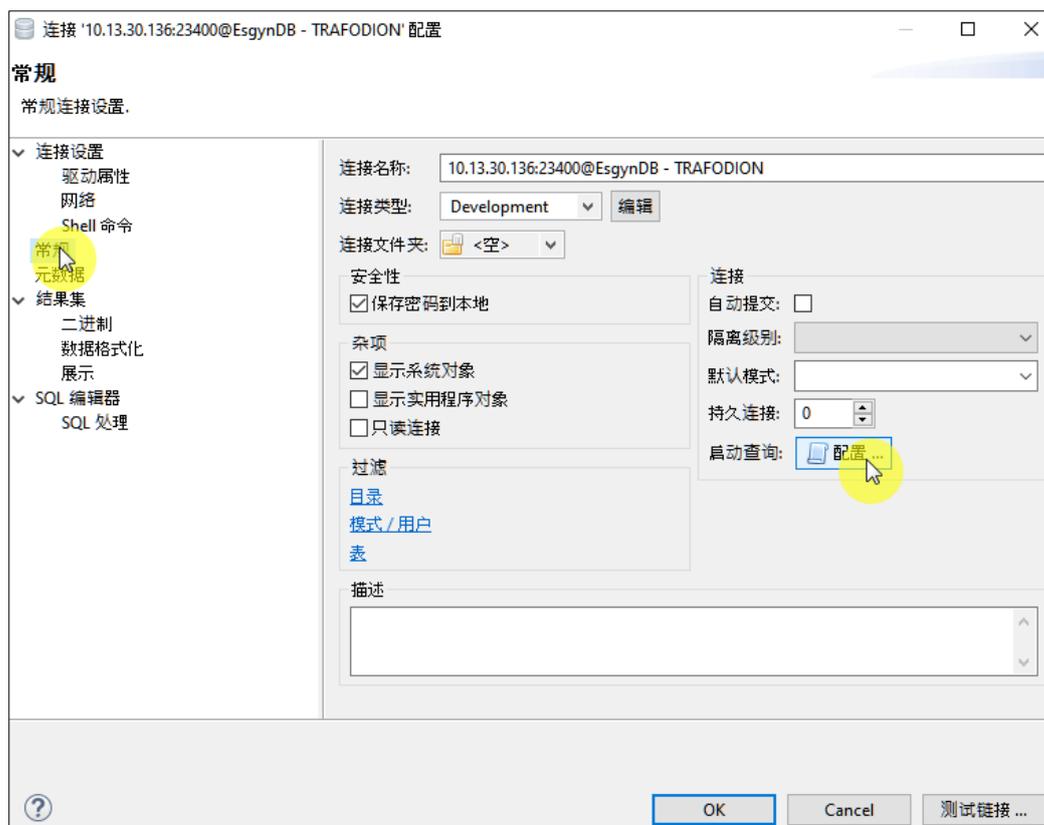


4.5 如何设置默认 CQD

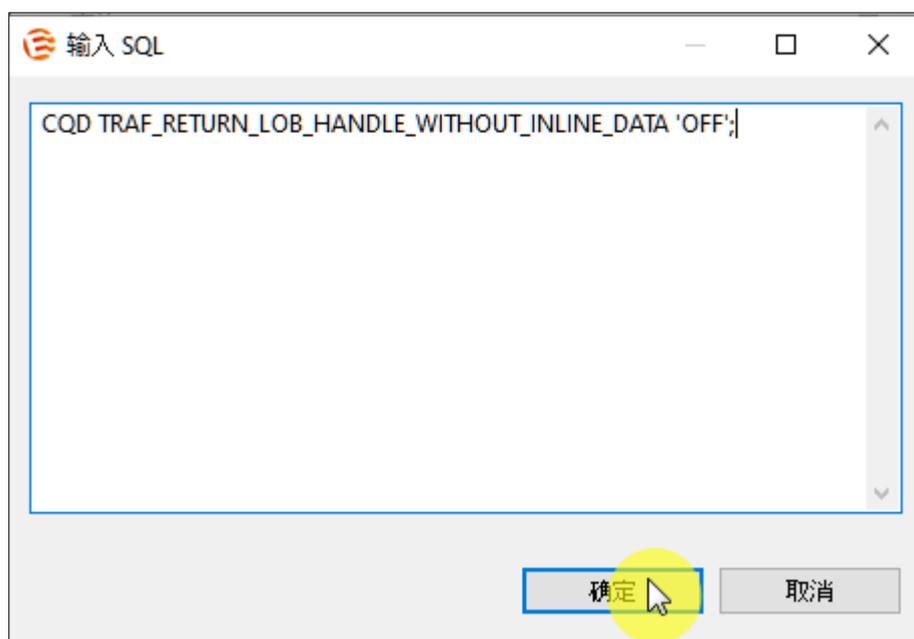
右击数据库连接，点击编辑连接



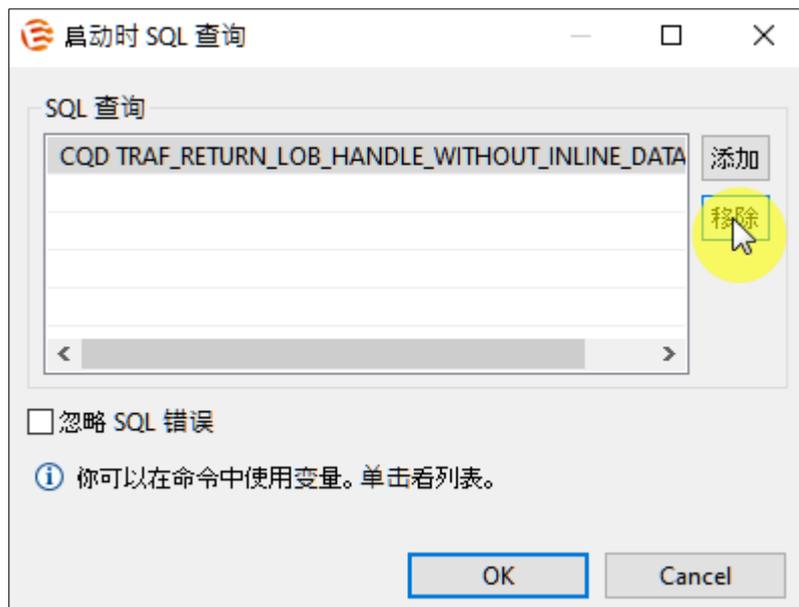
在连接配置窗口中，点击常规 -> 启动查询



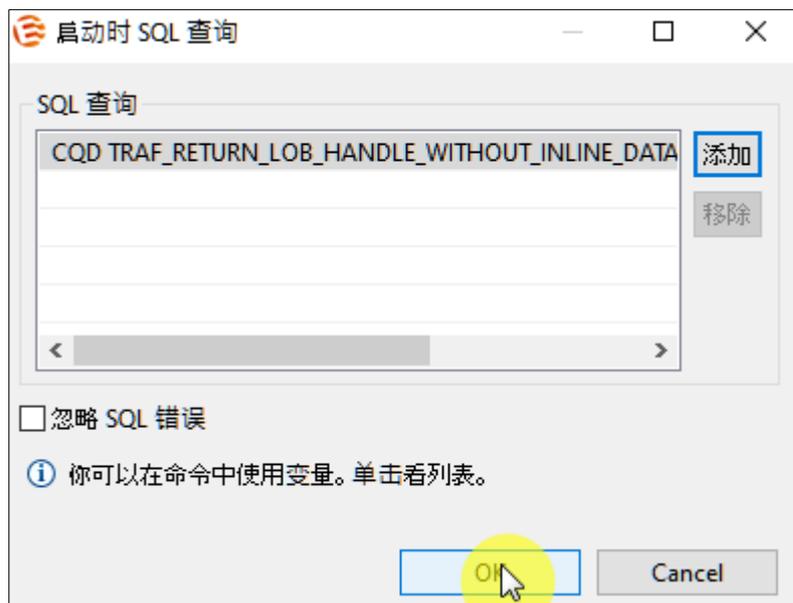
在启动时 SQL 查询窗口中点击添加按钮，输入 CQD 设置语句，点击确定按钮。



如需删除 CQD，可在启动时 SQL 查询窗口选中一条 CQD 设置，点击**移除**。



添加或移除 CQD 后，点击 OK 按钮完成 CQD 的配置。



 提示

CQD 设置更改后，需重新连接数据库才能生效。

5. SQL 编辑器

本章讲述以下内容：

[5.1 界面概览](#)

[5.2 SQL 脚本模板](#)

[5.3 SQL 脚本内容助理](#)

[5.4 SQL 脚本格式化](#)

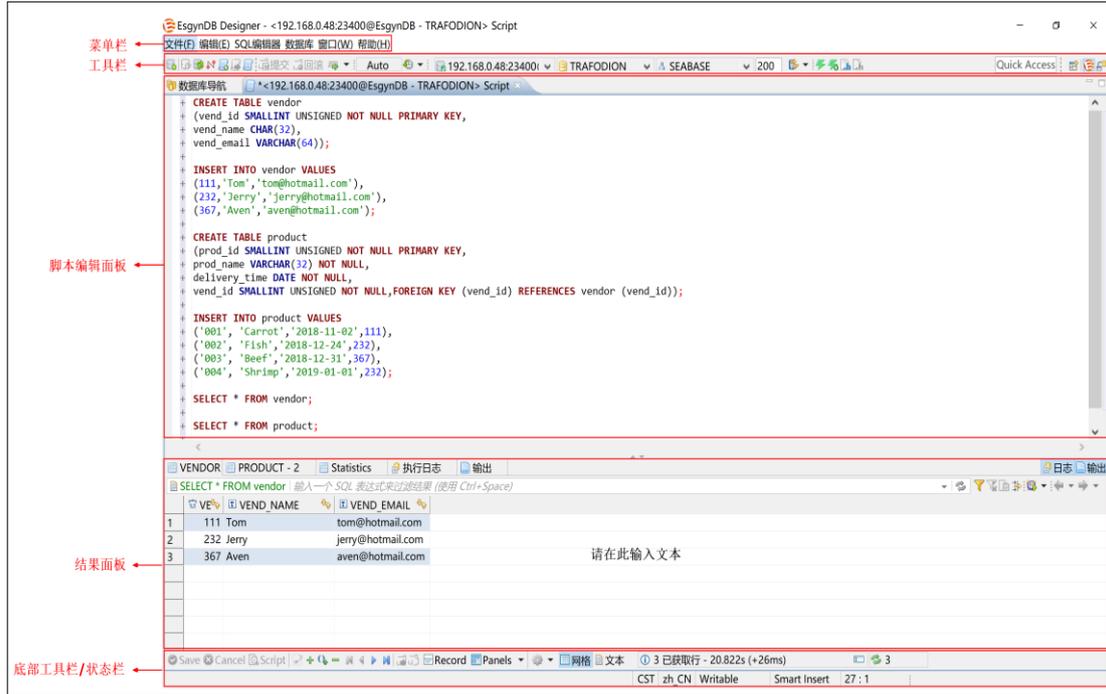
[5.5 SQL 脚本管理](#)

[5.6 SQL 脚本执行](#)

5.1 界面概览

SQL 编辑器界面主要包括菜单栏、工具栏、脚本编辑面板、结果面板和底部工具栏/状态栏。

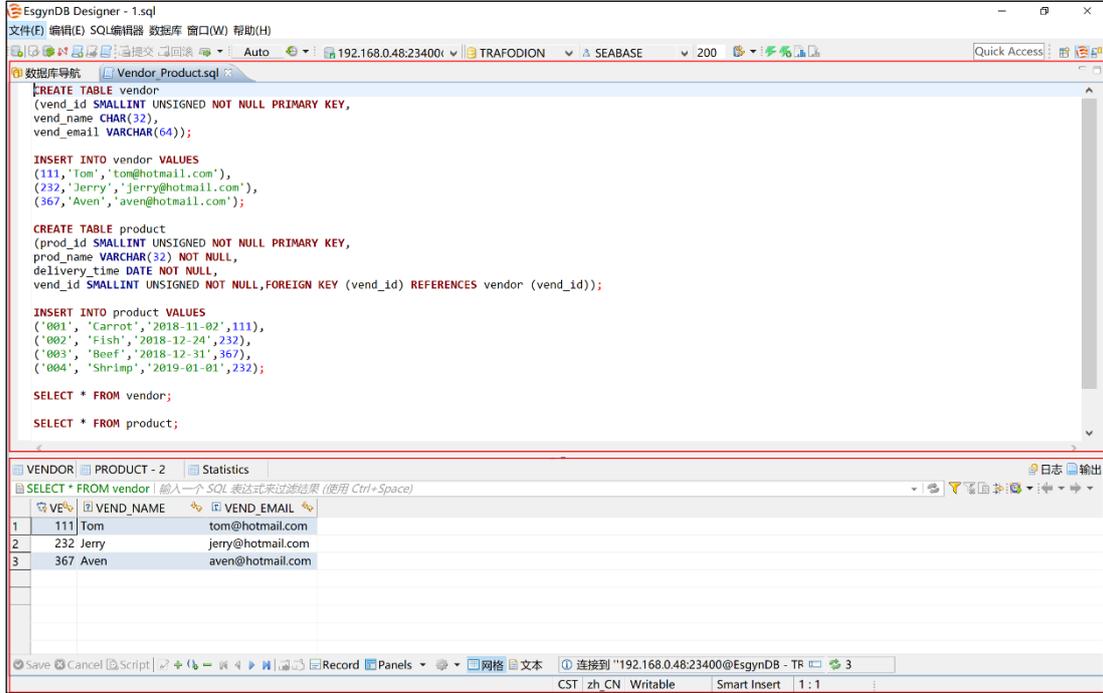
您可以在一个数据库连接中创建多个 SQL 脚本，每个 SQL 脚本在各自的脚本编辑器中运行。



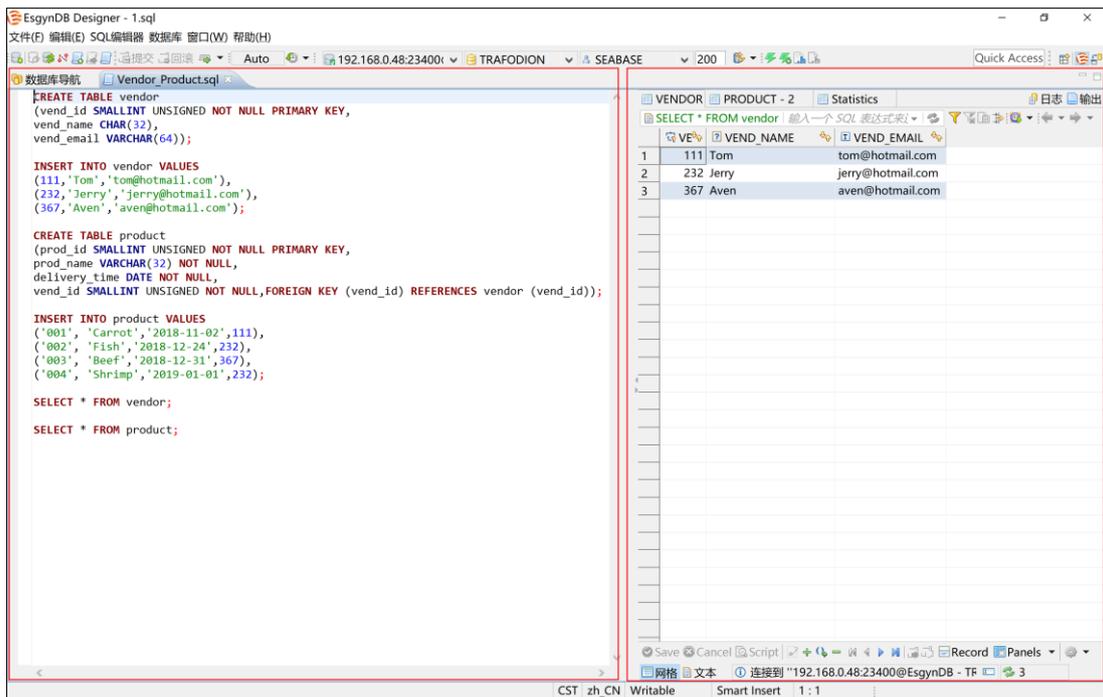
5.1.1 更改界面布局

5.1.1.1 设置水平/垂直布局

易鲸捷 Designer 的界面默认为水平布局。



如需更改为垂直布局，在脚本编辑面板中的任意区域，右键鼠标 > 点击布局 > 垂直。

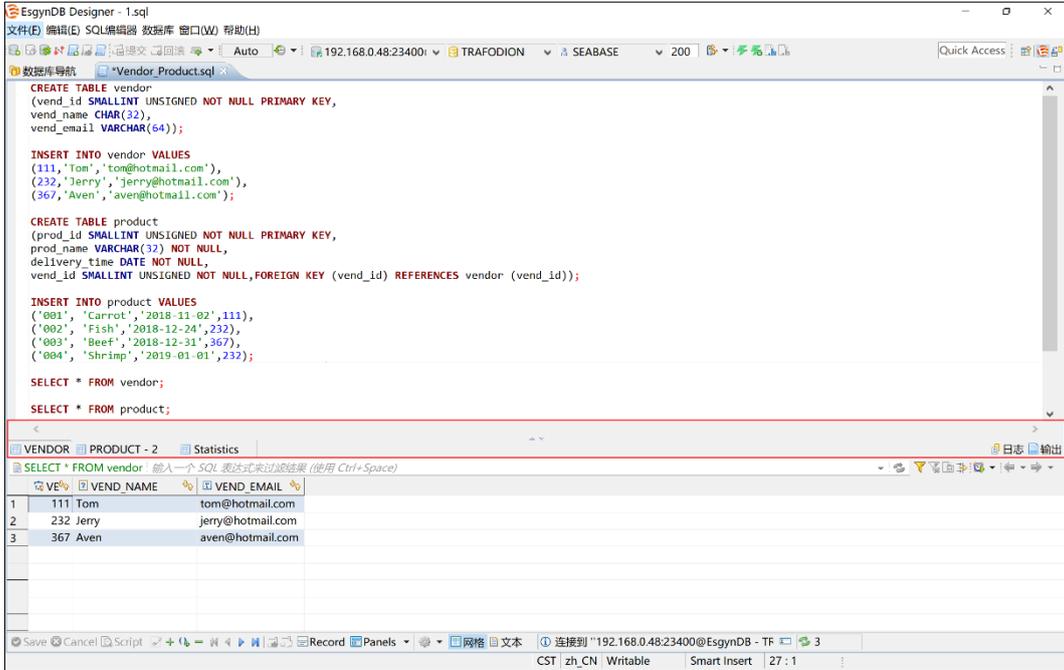


5.1.1.2 更改面板大小

易鲸捷 Designer 支持更改面板大小，执行以下步骤：

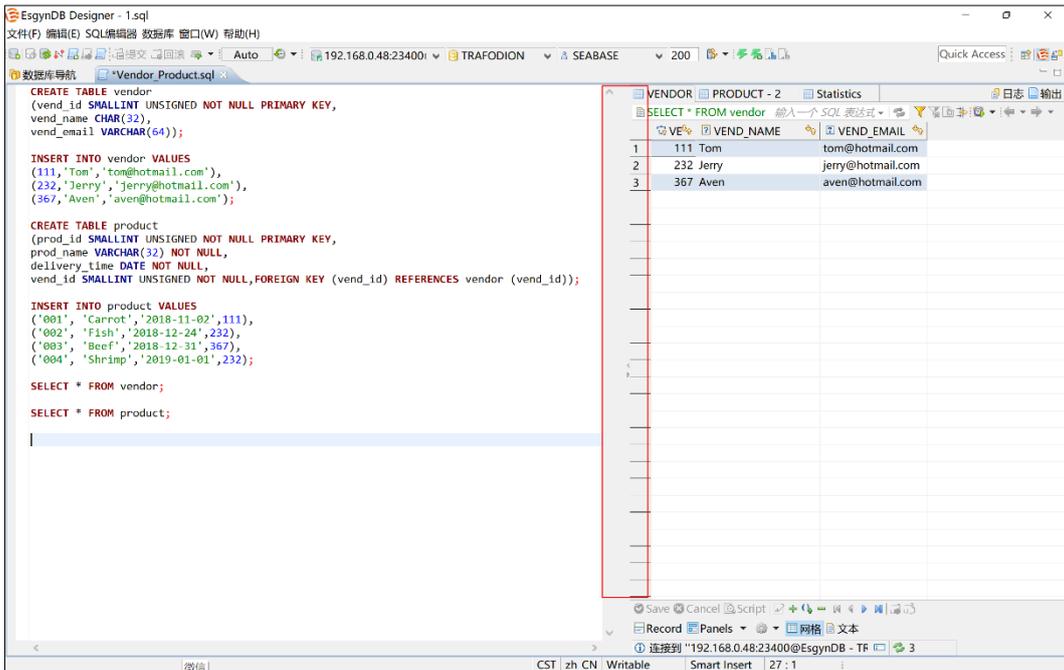
- 水平布局

将鼠标移至上下两个面板的交界处，待鼠标更改为双箭头时，上下拖动面板。



- 垂直布局

将鼠标移至左右两个面板的交界处，待鼠标更改为双箭头时，左右拖动面板。



5.1.1.3 最大化结果面板

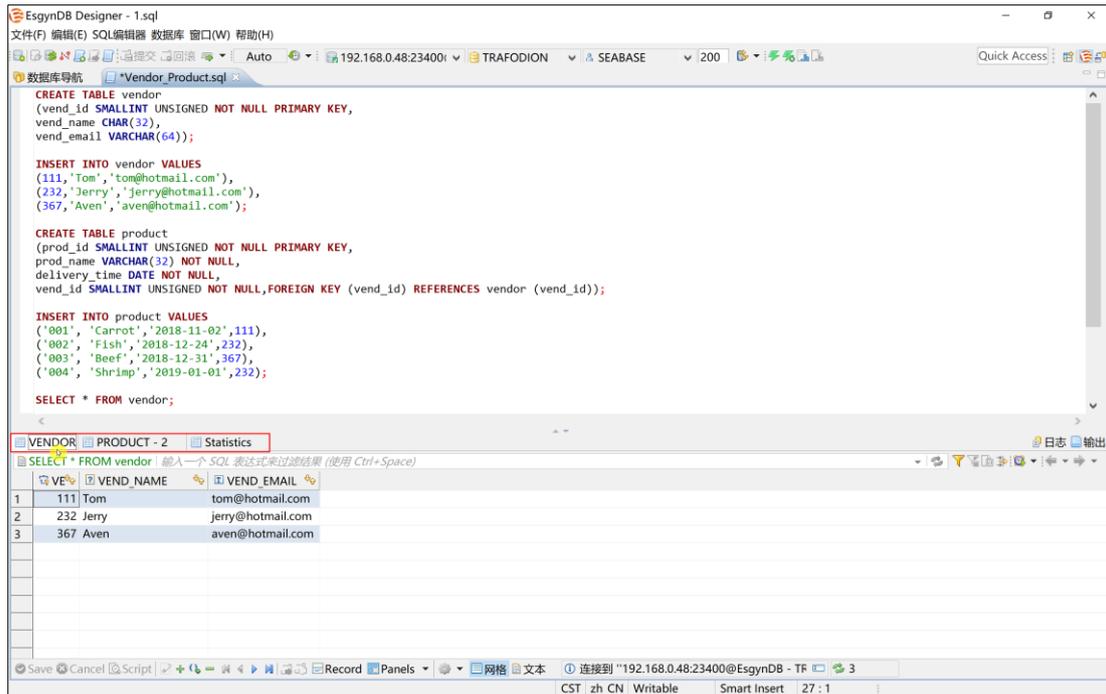
如需最大化结果面板，选择以下任一方法：

- **方法 1**

双击结果面板中的任一标签页名称。

示例

点击 VENDOR、PRODUCT 或 Statistics 都能最大化结果面板。



- **方法 2**

在脚本编辑面板中的任意区域，右键鼠标 > 选择布局 > 最大化结果面板。



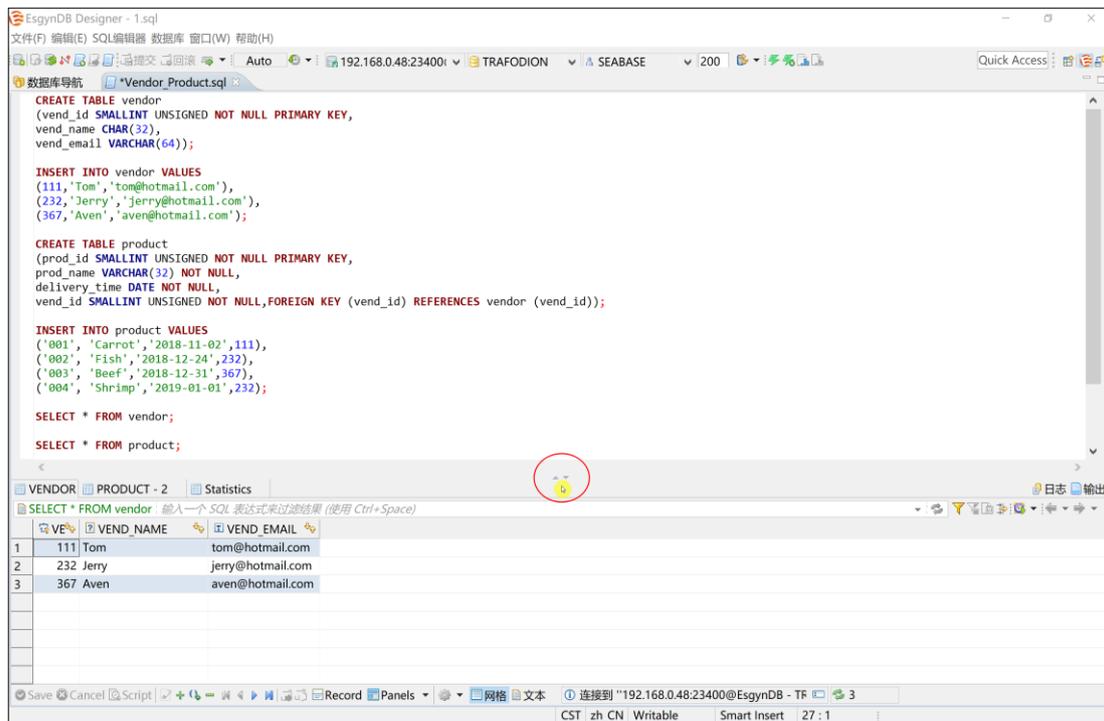
提示

如需还原面板，在最大化的结果面板中，双击任一标签页名称。

5.1.1.4 隐藏面板

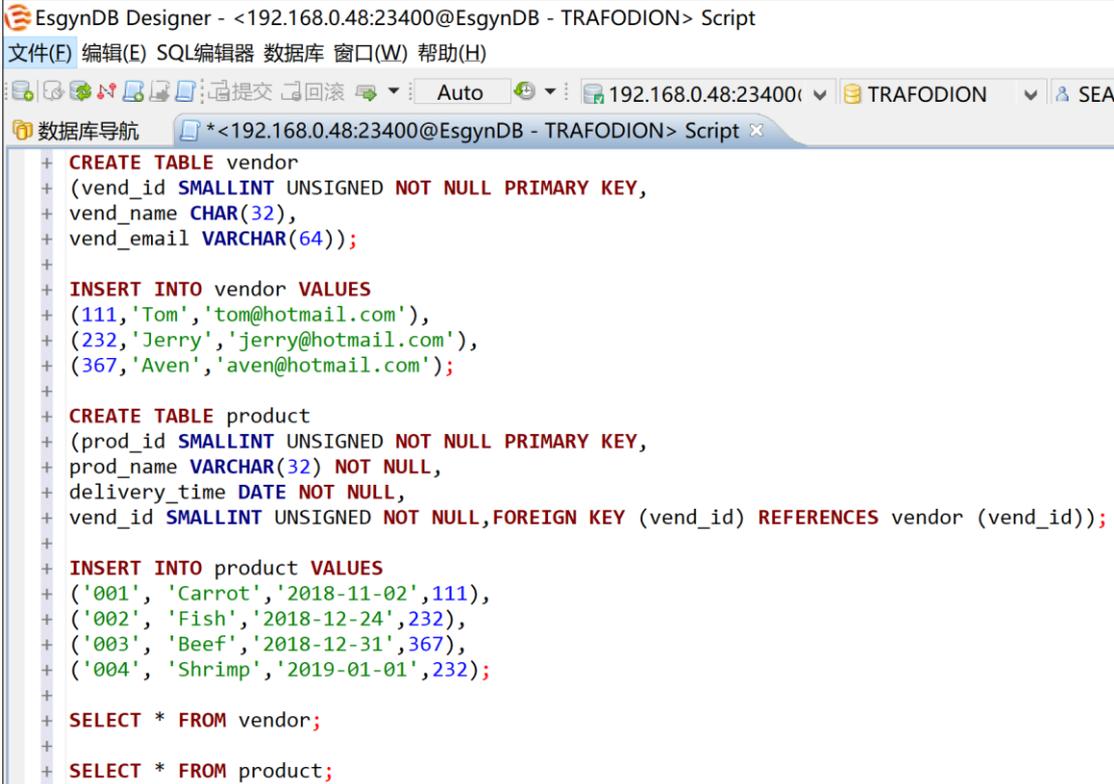
如需隐藏脚本编辑面板，点击三角形图标 ()。

如需隐藏结果面板，点击倒三角形图标 ()。



5.1.2 语法高亮

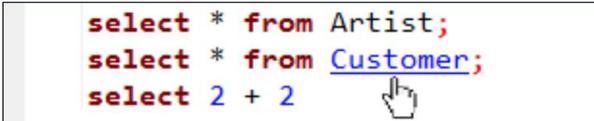
易鲸捷 Designer 支持 SQL 语法高亮显示，方便阅读和输入。



```
EsgynDB Designer - <192.168.0.48:23400@EsgynDB - TRAFODION> Script
文件(E) 编辑(E) SQL编辑器 数据库 窗口(W) 帮助(H)
Auto 192.168.0.48:23400( TRAFODION SEA
数据库导航 * <192.168.0.48:23400@EsgynDB - TRAFODION> Script x
+ CREATE TABLE vendor
+ (vend_id SMALLINT UNSIGNED NOT NULL PRIMARY KEY,
+ vend_name CHAR(32),
+ vend_email VARCHAR(64));
+
+ INSERT INTO vendor VALUES
+ (111, 'Tom', 'tom@hotmail.com'),
+ (232, 'Jerry', 'jerry@hotmail.com'),
+ (367, 'Aven', 'aven@hotmail.com');
+
+ CREATE TABLE product
+ (prod_id SMALLINT UNSIGNED NOT NULL PRIMARY KEY,
+ prod_name VARCHAR(32) NOT NULL,
+ delivery_time DATE NOT NULL,
+ vend_id SMALLINT UNSIGNED NOT NULL, FOREIGN KEY (vend_id) REFERENCES vendor (vend_id));
+
+ INSERT INTO product VALUES
+ ('001', 'Carrot', '2018-11-02', 111),
+ ('002', 'Fish', '2018-12-24', 232),
+ ('003', 'Beef', '2018-12-31', 367),
+ ('004', 'Shrimp', '2019-01-01', 232);
+
+ SELECT * FROM vendor;
+
+ SELECT * FROM product;
```

5.1.3 超链接

在脚本编辑面板中，按住 Ctrl 的同时将鼠标移至 SQL 脚本上，如果 SQL 脚本是表名称或视图名称，它将会显示成一个超链接，点击该超链接能直接跳转至该对象的编辑界面。



```
select * from Artist;
select * from Customer;
select 2 + 2
```

5.1.4 切换数据源/Catalog/Schema

在保持当前 SQL 脚本不变的情况下，您能切换数据源（连接）/Catalog/Schema。

5.1.4.1 切换数据源

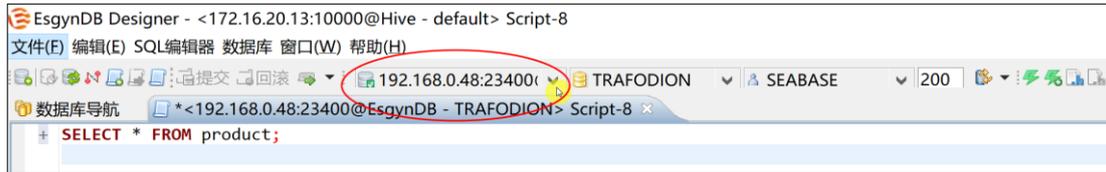
如需切换数据源，选择以下任一方法：

- 方法 1

在脚本编辑界面同时按下 **Ctrl+9**，在弹出的界面中，选择所需数据源。

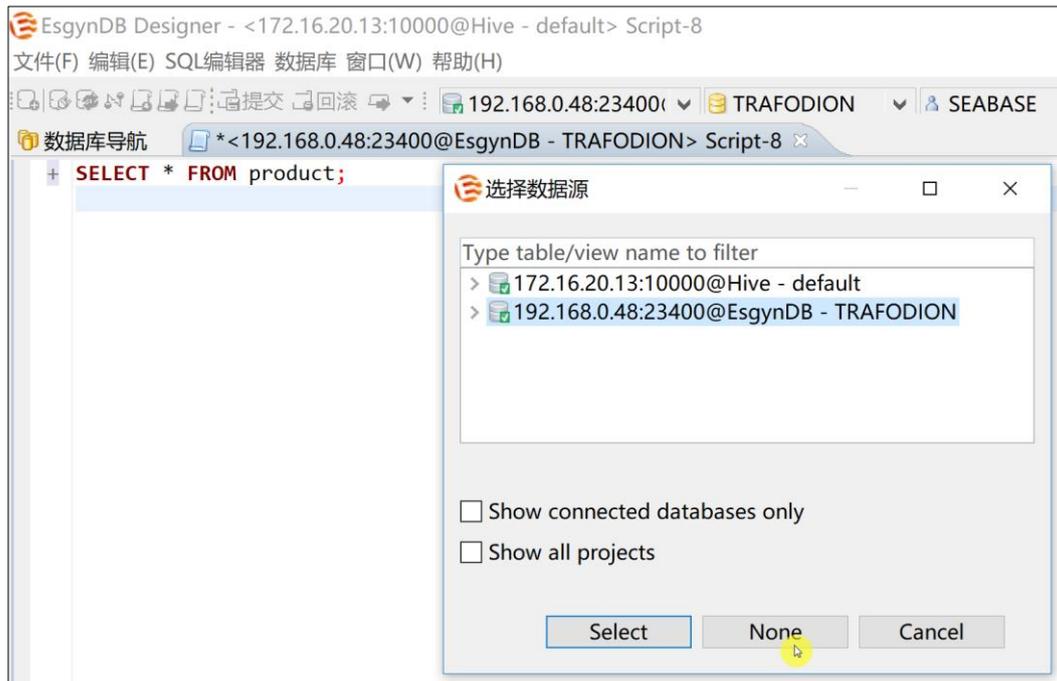
- 方法 2

在工具栏中，在**选择数据源**的下拉列表中选择所需数据源。



提示

如需断开连接，点击 **None**。



5.1.4.2 切换 Catalog/Schema

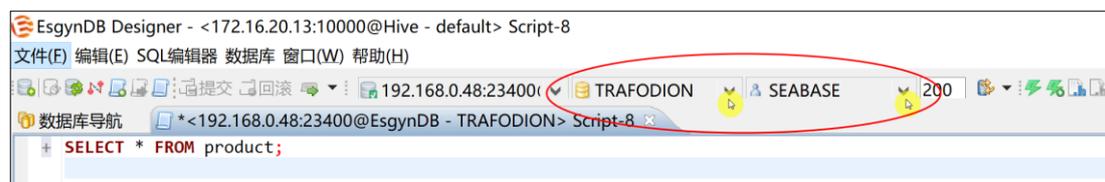
如需切换连接，选择以下任一方法：

- **方法 1**

在脚本编辑界面同时按下 Ctrl+0，在弹出的界面中，选择所需 Catalog 或 Schema。

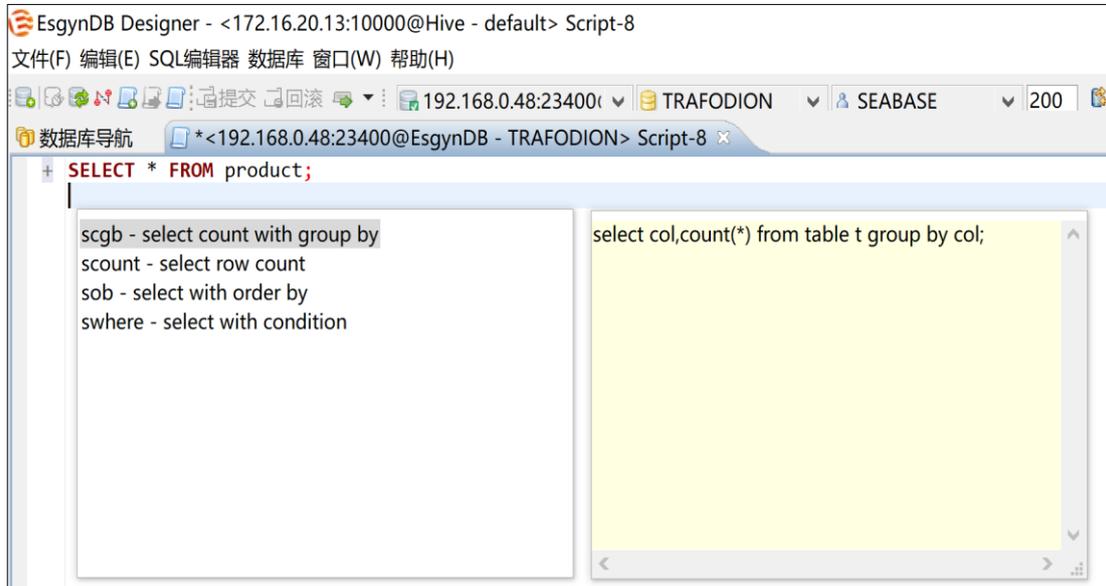
- **方法 2**

在工具栏中，在 **Catalog** 或 **Schema** 的下拉列表中选择所需 Catalog 或 Schema。



5.2 SQL 脚本模板

SQL 脚本模板帮助快速输入最常用的 SQL 语句。



5.2.1 查看 SQL 脚本模板

如需查看 SQL 脚本模板，选择以下任一方法：

- **方法 1**

在脚本编辑面板中的任意区域，同时按下 **Ctrl+Alt+Space**，弹出的对话框将显示可用的 SQL 脚本模板。

- **方法 2**

在脚本编辑面板中的任意区域，右键鼠标 > 点击 **SQL 模板**，弹出的对话框将显示可用的 SQL 脚本模板。

5.2.2 使用 SQL 脚本模板

如需使用 SQL 脚本模板，选择以下任一方法：

- **方法 1**

在脚本编辑面板中，输入 SQL 脚本模板名称，点击 **Tab** 键。

- **方法 2**

在脚本编辑面板中的任意区域，右键鼠标 > 点击 **SQL 模板**，弹出的对话框将显示可用的 SQL 脚本模板，双击选择所需模板。

5.2.3 新增/编辑/删除 SQL 脚本模板

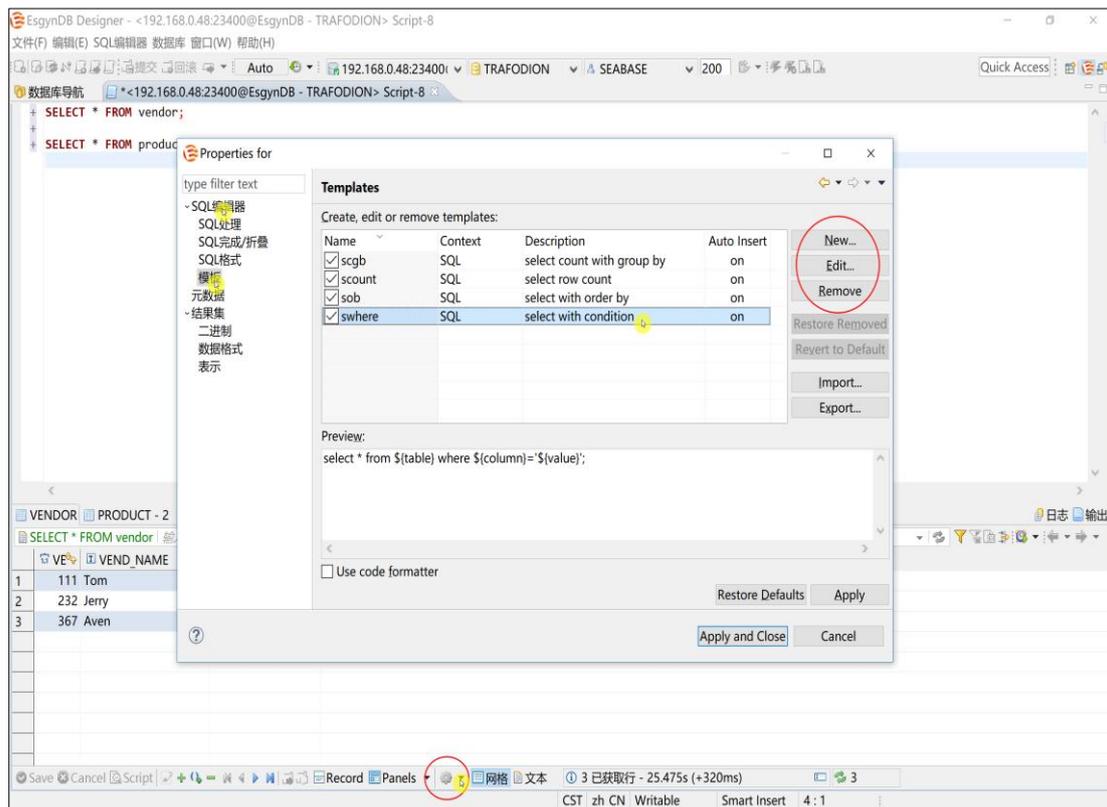
以下是 SQL 脚本模板的部分变量及其描述：

变量	描述
<code>#{table}</code>	输入表名称
<code>#{column}</code>	输入列名称
<code>#{value}</code>	输入值

如需新增/编辑/删除 SQL 脚本模板，选择以下任一方法：

- 方法 1

在底部工具栏，点击配置图标 () > 点击首选项 > 点击 SQL 编辑器 > 点击模板，点击新增/编辑/删除。



- 方法 2

在脚本编辑面板中的任意区域，右键鼠标 > 点击首选项 > 点击通用 > 点击编辑器 > 点击 SQL 编辑器 > 点击模板 > 点击新增/编辑/删除。

5.2.4 导入/导出 SQL 脚本模板



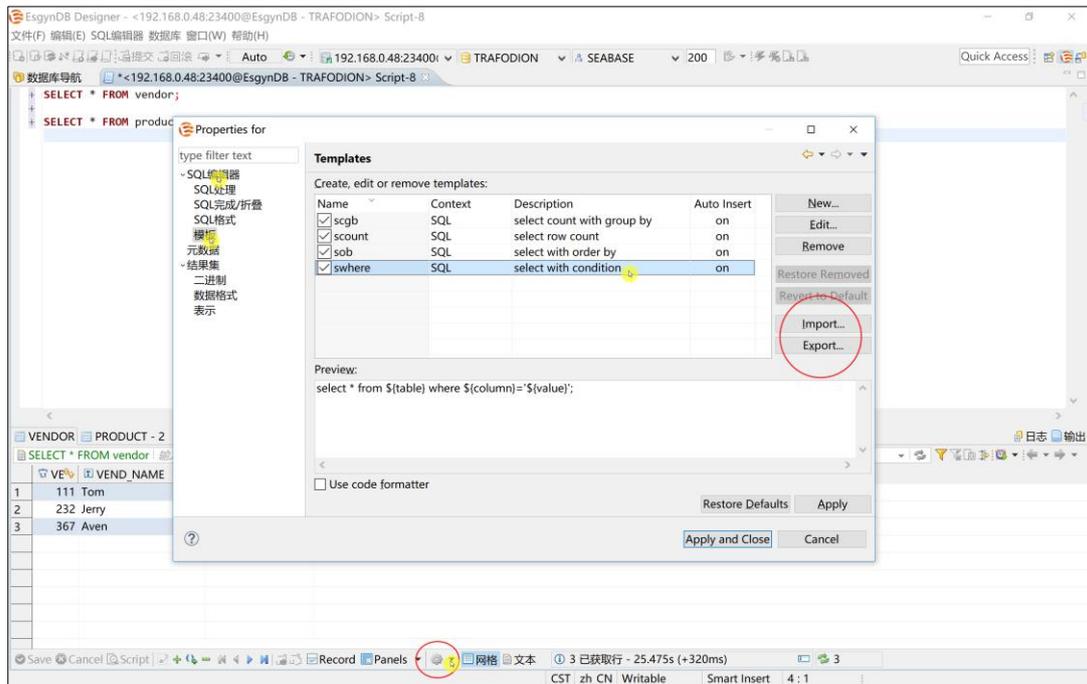
提示

易鲸捷 Designer 支持导入/导出.xml 文件。

如需导入/导出 SQL 脚本模板，选择以下任一方法：

- 方法 1

在底部工具栏，点击配置图标 () > 点击首选项 > 点击 SQL 编辑器 > 点击模板，点击导入/导出。



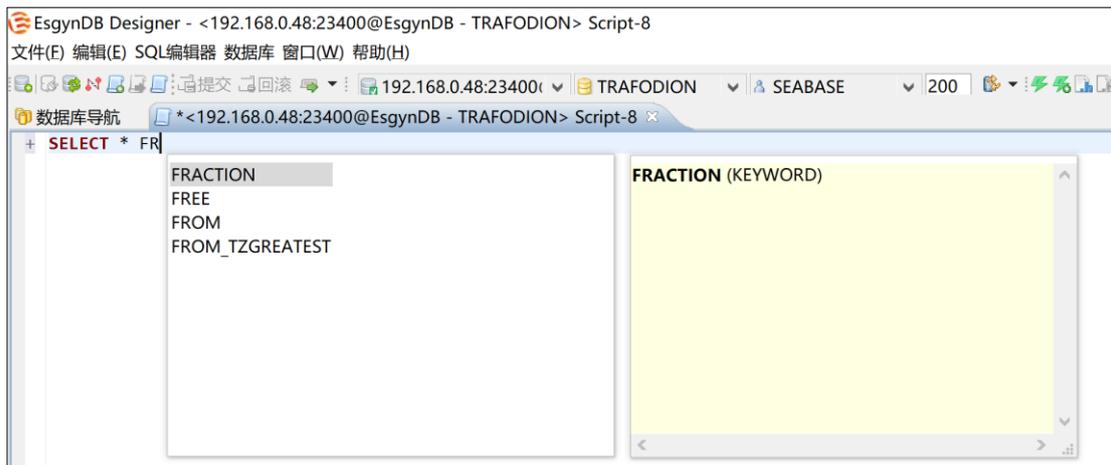
- 方法 2

在脚本编辑面板中的任意区域，右键鼠标 > 点击首选项 > 点击通用 > 点击编辑器 > 点击 SQL 编辑器 > 点击模板 > 点击导入/导出。

5.3 SQL 脚本内容助理

SQL 脚本内容助理搜索数据库中的所有对象，能自动填充数据库的对象名称、SQL 命令和关键字。

如果您正在输入 SQL 关键字，易鲸捷 Designer 会自动开启 SQL 脚本内容助理功能，提示即将输入的内容。



如需使用 SQL 脚本内容助理，选择以下任一方法：

- **方法 1**

在待提示的脚本位置，同时按下 **Ctrl+Space**，弹出的对话框将显示 SQL 脚本内容助理的提示结果。

- **方法 2**

在待提示的脚本位置，右键鼠标 > 点击 **SQL 内容助理**，弹出的对话框将显示 SQL 脚本内容助理的提示结果。

5.4 SQL 脚本格式化

5.4.1 格式化 SQL 脚本

易鲸捷 Designer 支持更改脚本的格式，提高脚本的可读性。

“格式化 SQL 脚本”前

```
CREATE TABLE product
(prod_id
SMALLINT UNSIGNED
NOT NULL PRIMARY KEY,
prod_name
VARCHAR(32)
NOT NULL,
delivery_time
DATE
NOT NULL,
vend_id
SMALLINT
UNSIGNED
NOT NULL,
FOREIGN KEY (vend_id)
REFERENCES vendor (vend_id));

INSERT INTO product VALUES
('001', 'Carrot', '2018-11-02', 111),
('002', 'Fish', '2018-12-24', 232),
('003', 'Beef', '2018-12-31', 367),
('004', 'Shrimp', '2019-01-01', 232);
```

“格式化 SQL 脚本”后

```
CREATE
TABLE
product(
prod_id SMALLINT UNSIGNED NOT NULL PRIMARY KEY,
prod_name VARCHAR(32) NOT NULL,
delivery_time DATE NOT NULL,
vend_id SMALLINT UNSIGNED NOT NULL,
FOREIGN KEY(vend_id) REFERENCES vendor(vend_id)
);

INSERT
INTO
product
VALUES(
'001',
'Carrot',
'2018-11-02',
111
),
(
'002',
'Fish',
'2018-12-24',
232
),
(
'003',
'Beef',
'2018-12-31',
367
),
(
'004',
'Shrimp',
'2019-01-01',
232
);
```

如需脚本格式化 SQL 脚本，选择以下任一方法：

- 方法 1

在脚本编辑面板中的任意区域，同时按下 **Ctrl+Shift+F**。

- 方法 2

在脚本编辑面板中的任意区域，右键鼠标 > 点击**格式化** > 点击**格式化 SQL**。

5.4.2 更改 SQL 脚本的大小写

易鲸捷 Designer 支持更改 SQL 脚本的大小写。

全部大写

```
CREATE TABLE PRODUCT
(PROD_ID SMALLINT UNSIGNED NOT NULL PRIMARY KEY,
PROD_NAME VARCHAR(32) NOT NULL,
DELIVERY_TIME DATE NOT NULL,
VEND_ID SMALLINT UNSIGNED NOT NULL,
FOREIGN KEY (VEND_ID) REFERENCES VENDOR (VEND_ID));

INSERT INTO PRODUCT VALUES
('001', 'CARROT', '2018-11-02', 111),
('002', 'FISH', '2018-12-24', 232),
('003', 'BEEF', '2018-12-31', 367),
('004', 'SHRIMP', '2019-01-01', 232);
```

全部小写

```
create table product
(prod_id smallint unsigned not null primary key,
prod_name varchar(32) not null,
delivery_time date not null,
vend_id smallint unsigned not null,
foreign key (vend_id) references vendor (vend_id));

insert into product values
('001', 'carrot', '2018-11-02', 111),
('002', 'fish', '2018-12-24', 232),
('003', 'beef', '2018-12-31', 367),
('004', 'shrimp', '2019-01-01', 232);
```

如需将 SQL 脚本更改成大写/全部小写，选择以下任一方法：

- 方法 1

选中待修改的脚本，同时按下 **Ctrl+Shift+U** (大写) 或 **Ctrl+Shift+L** (小写)。

- 方法 2

选中待修改的脚本，右键鼠标 > 点击格式化 > 点击更改成大写或更改成小写。

5.4.3 注释 SQL 脚本

易鲸捷 Designer 支持对 SQL 脚本进行行注释或块注释

行注释

```
--create table product
--(prod_id smallint unsigned not null primary key,
--prod_name varchar(32) not null,
--delivery_time date not null,
--vend_id smallint unsigned not null,
--foreign key (vend_id) references vendor (vend_id));
--
--insert into product values
--('001', 'carrot','2018-11-02',111),
--('002', 'fish','2018-12-24',232),
--('003', 'beef','2018-12-31',367),
--('004', 'shrimp','2019-01-01',232);
```

块注释

```
/*
--create table product
--(prod_id smallint unsigned not null primary key,
--prod_name varchar(32) not null,
--delivery_time date not null,
--vend_id smallint unsigned not null,
--foreign key (vend_id) references vendor (vend_id));
--
--insert into product values
--('001', 'carrot','2018-11-02',111),
--('002', 'fish','2018-12-24',232),
--('003', 'beef','2018-12-31',367),
--('004', 'shrimp','2019-01-01',232);
*/
```

如需对 SQL 脚本进行行注释或块注释，选择以下任一方法：

- **方法 1**

选中待修改的脚本，同时按下 **Ctrl+/**（行注释）或 **Ctrl+Shift+/**（块注释）。

- **方法 2**

选中待修改的脚本，右键鼠标 > 点击**格式化** > 点击**切换行注释**或**切换块注释**。

5.5 SQL 脚本管理

5.5.1 保存脚本



提示

在脚本编辑面板中输入的脚本将自动保存为 **Script-n.sql** 文件，保存路径为 **C:\Users\username\.esgyndb_designer4\General\Scripts**。

如需保存脚本，选择以下任一方法：

- 方法 1

在菜单栏中，点击文件 > 保存。



- 方法 2

在脚本编辑面板中的任意区域，右键鼠标 > 点击文件 > 保存 SQL 脚本。

5.5.2 重命名脚本

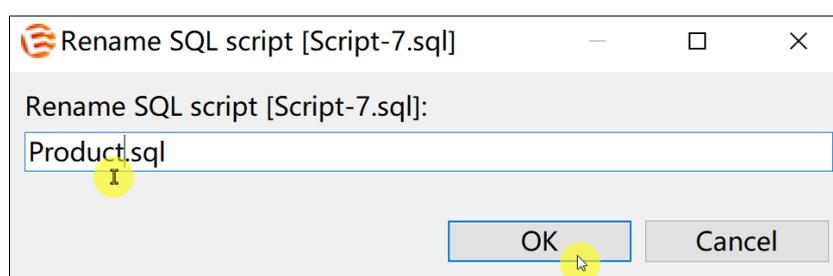
如需导入脚本，选择以下任一方法：

- **方法 1**

同时按下 **Ctrl+F2**，在弹出的对话框中输入新的脚本文件名称。

- **方法 2**

在脚本编辑面板中的任意区域，右键鼠标 > 点击**文件** > **重命名 SQL 脚本**，在弹出的对话框中输入新的脚本文件名称。



5.5.3 导入脚本

如需导入脚本，选择以下任一方法：

- **方法 1**

同时按下 **Ctrl+Shift+O**，在弹出的对话框中选择.sql 或.txt 文件。

- **方法 2**

在脚本编辑面板中的任意区域，右键鼠标 > 点击**文件** > **导入 SQL 脚本**，在弹出的对话框中选择.sql 或.txt 文件。

5.6 SQL 脚本执行

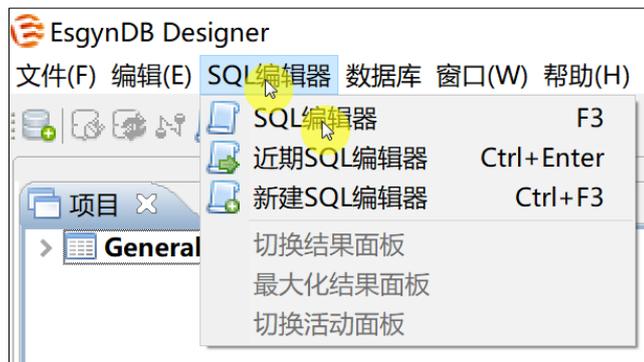
5.6.1 如何创建 SQL 脚本



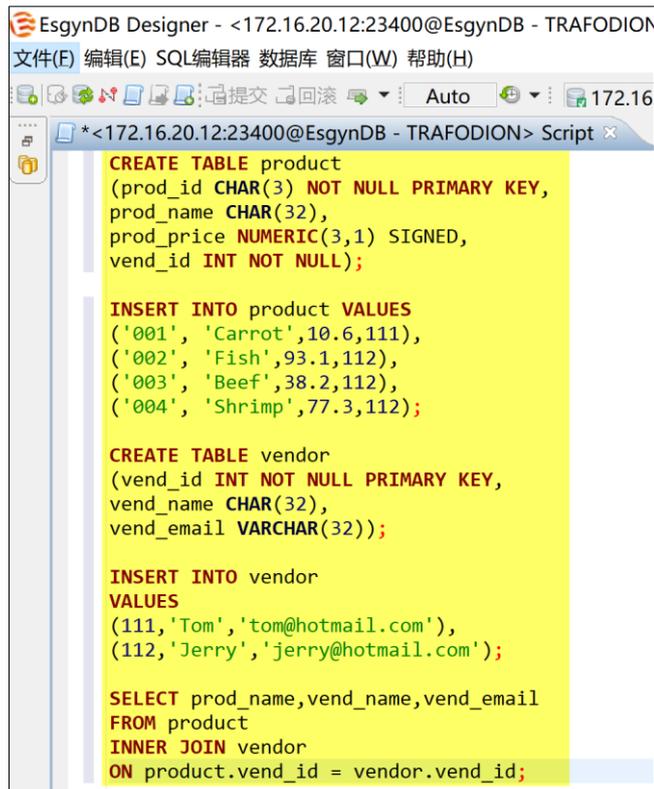
注意

创建 SQL 脚本前，必须先建立数据库连接。更多信息，请参阅 [4.1 如何创建数据库连接](#)。

1. 在菜单栏中，点击 **SQL 编辑器 > SQL 编辑器**。



2. 在弹出的对话框中，输入一条或多条 SQL 脚本。



3. 在菜单栏中:

- 如需使用默认文件名, 点击文件 > 保存。



提示

SQL 脚本的默认文件名是 **Script-n.sql**。

SQL 脚本的默认保存路径是

C:\Users\username\.esgyndb_designer4\General\Scripts。

- 如需重命名文件名, 点击文件 > 另存为, 输入文件名。

5.6.2 如何编辑 SQL 脚本

您能通过以下 2 种方法编辑 SQL 脚本：

- [5.6.2.1](#) 使用 SQL 编辑器的编辑选项
- [5.6.2.2.2](#) 使用菜单栏的编辑选项

5.6.2.1 使用 SQL 编辑器的编辑选项

您能使用 **SQL 编辑器的编辑选项** 编辑 SQL 脚本。

如何创建 SQL 脚本第 2 步的窗口中，右键鼠标，弹出的对话框将显示 **编辑选项**。

表 5-1 SQL 编辑器 - 编辑选项 1

选项		说明	快捷键
一级选项	二级选项		
撤销输入			Ctrl + Z
保存			Ctrl + S
剪切	/	/	Ctrl + X
复制			Ctrl + C
粘贴			Ctrl + V
执行	执行 (串行)	串行执行 SQL 语句。 更多信息, 请参阅 5.6.3.1 如何执行 (串行) SQL 脚本 。	Alt + X
	执行 (并行)	并行执行 SQL 语句。 更多信息, 请参阅 5.6.3.2 如何执行 (并行) SQL 脚本 。	Ctrl + Alt + X
	选择所有行	/	Ctrl + Alt + A
	解析执行计划	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">  注意 该功能每次只能对一条 SQL 脚本执行。 </div> 更多信息, 请参阅 5.6.5 如何解析 SQL 查询计划 。	Ctrl + Shift + E
	从查询中导出	查询结果能导出至文件或数据库表。	/

		<p>更多信息，请参阅：</p> <p>5.6.6.1 如何导出查询结果至文件</p> <p>5.6.6.2 如何导出查询结果至数据库表。</p>	
--	--	---	--

表 5-2 SQL 编辑器 - 编辑选项 2

选项		说明	快捷键
一级选项	二级选项		
文件	重命名 SQL 脚本	/	Ctrl + F2
	撤销输入		/
	导入 SQL 脚本	支持多种文件格式，包括.sql 和.txt。	/
	保存 SQL 脚本		/
格式化	格式化 SQL	调整 SQL 语句格式，例如，遇到关键字换行。	Ctrl + Alt + F
	转成大写	/	Ctrl + Shift + U
	转成小写	/	Ctrl + Shift + L
	切换行注释	切换成单行注释 (//)。	Ctrl + /
	切换块注释	切换成多行注释 (/* */)。	Ctrl + Shift + /
布局	水平布局	多个窗口水平排列。	/
	垂直布局	多个窗口垂直排列。	/
	切换结果面板	/	F6
	最大化结果面板		Shift + F6
	切换活动面板		Alt + F6
SQL 内容助理	/	提示输入信息。	Ctrl + Space
SQL 模板	/	SQL 语句模板。	Ctrl + Alt + Space

表 5-3 SQL 编辑器 - 编辑选项 3

选项		
一级选项	二级选项	三级选项
首选项	通用	始终在后台运行
		自动保存时间间隔
	外观	启用主题 (暂不可用)
		启用动画 (暂不可用)
		label 使用多种字体和颜色
		显示最近使用的 tab
		颜色和字体

表 5-4 SQL 编辑器 - 编辑选项 4

选项			
一级选项	二级选项	三级选项	四级选项
首选项	SQL 编辑器	启用数据源	
		连接	每个编辑器单独进行连接
			打开编辑器后连接
			查询语句执行时连接
		自动保存	编辑器关闭时自动保存脚本文件
			查询语句执行时自动保存脚本文件
		资源	在关闭时删除空脚本
			为每个连接创建脚本文件夹
			创建脚本文件夹
			脚本标题模式: <\${connectionName}> \${fileName}
		结果视图	出错时关闭结果标签
			结果排列方向

表 5-5 SQL 编辑器 - 编辑选项 5

选项						
一级选项	二级选项	三级选项	四级选项	五级选项	六级选项	
首选项	SQL 编辑器	SQL 处理	常见	执行查询前校验连接	/	
				查询完成后发出声音提示	/	
				SQL 执行后刷新活动模式	/	
			参数	启用 SQL 参数	/	
				匿名 SQL 参数	/	
				匿名参数标记	/	
				命名参数前缀	/	
				在 DDL 语句中启用参数	/	
			脚本	提交类型	在脚本末端	
					每行之后 (自动提交)	
					每行之后 不提交	
				单行提交	/	
				处理错误	停止 + 回滚 (暂不可用)	
			停止 + 提			

					交 (暂不可 用)
				获取结果集	/
				执行完后重置光标	/
				在脚本执行时最大化编辑器	/
				分隔符	语句分隔 符
					空白行是 语句分隔 符

表 5-6 SQL 编辑器 - 编辑选项 6

选项

一级 选项	二级 选项	三级 选项	四级 选项	五级选项	六级选项
首选 项	SQL 编辑器	SQL 格式 化	自动关 闭符号	单引号	/
				双引号	/
				括号	/
			格式化	格式	默认
					外部
				关键字格式	数据库
					大写
			小写		
			自动套 用格式	关键字大小写转换	/
				从源代码中提取 SQL 语句	/

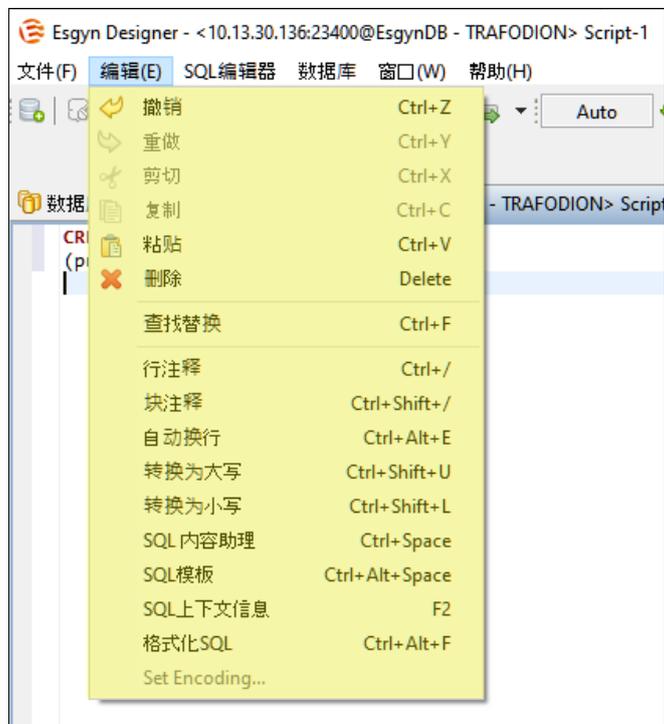
			启用自动激活	/	
			输入时激活	/	
			自动插入建议	/	
			插入字母	/	
		SQL 帮助/ 补全	隐藏非当前 schema 的重复名称	/	
			使用对象名称缩写	/	
			在表/列名后插入空格	/	
			使用全局搜索（在所有 schema 中）	/	
			折叠	启用折叠	/
			模板	 提示 您能创建 SQL 语句模板。	/

表 5-7 SQL 编辑器 - 编辑选项 7

选项			
一级选项	二级选项	三级选项	四级选项
首选项	文本编辑器	/	tab 显示宽度
			为 tab 插入空格
			高亮当前行
			显示打印页边距
			显示行序号
			显示距离指示器
			显示空格字符
			允许拖放和删除文本
			在编辑导入的文件前, 发出警告提示
			在行起始处和结束处显示插入符
		外观颜色选项	
		访问	使用自定插入符
			使用加粗插入符
		注释类型	/
		连接模式	/
		快速对比	/
		拼写	/

5.6.2.2 使用菜单栏的编辑选项

您能使用菜单栏的编辑选项编辑 SQL 脚本, 这些编辑选项也在 5.6.2.1 使用 SQL 编辑器的编辑选项中。



5.6.3 如何执行 SQL 脚本



执行脚本前，必须先建立数据库连接。更多信息，请参阅 [4.1 如何创建数据库连接](#)。



执行 SQL 脚本有以下 2 种方法：

- 执行（串行）。更多信息，请参阅 [5.6.3.1 如何执行（串行）SQL 脚本](#)。
- 执行（并行）。更多信息，请参阅 [5.6.3.2 如何执行（并行）SQL 脚本](#)。

如果多条 SQL 语句存在前后依赖关系，则应使用执行（串行）方式，否则会报错。

5.6.3.1 如何执行（串行）SQL 脚本



本示例在创建 EsgynDB 连接后，执行以下多条 SQL 脚本，由于它们存在前后依赖关系，所以选择执行（串行）方式。

```
CREATE TABLE product
(prod_id CHAR(3) NOT NULL PRIMARY KEY,
prod_name CHAR(32),
prod_price NUMERIC(3,1) SIGNED,
vend_id INT NOT NULL);

INSERT INTO product VALUES
('001', 'Carrot',10.6,111),
('002', 'Fish',93.1,112),
('003', 'Beef',38.2,112),
('004', 'Shrimp',77.3,112);

CREATE TABLE vendor
(vend_id INT NOT NULL PRIMARY KEY,
vend_name CHAR(32),
vend_email VARCHAR(32));

INSERT INTO vendor
VALUES
(111,'Tom','tom@hotmail.com'),
(112,'Jerry','jerry@hotmail.com');

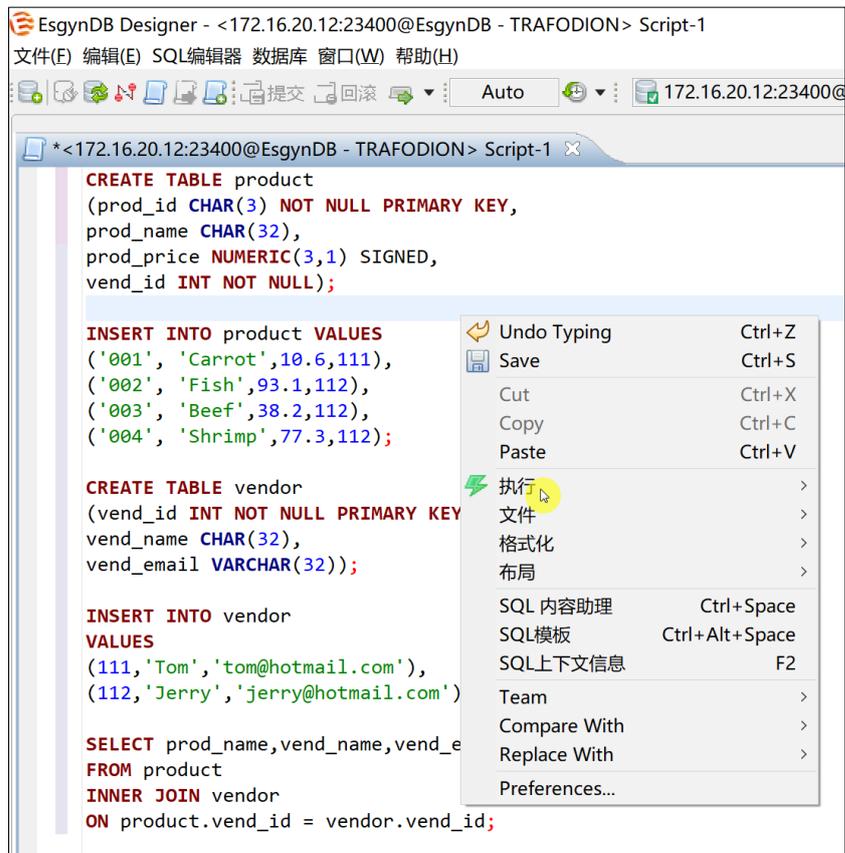
SELECT prod_name,vend_name,vend_email
FROM product
INNER JOIN vendor
ON product.vend_id = vendor.vend_id;
```

1. 在菜单栏中，点击 **SQL 编辑器 > 新建 SQL 编辑器**。
2. 在 SQL 编辑器窗口中，输入以上 SQL 脚本。
3. 点击**执行（串行）**。

有以下 3 种方法点击**执行（串行）**：

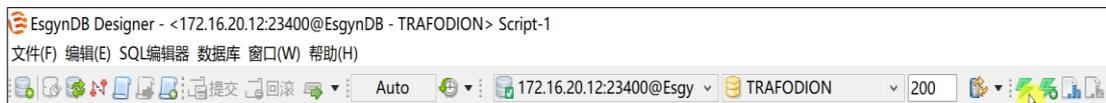
- **方法 1**

在 SQL 编辑器窗口，右键鼠标，在弹出的界面中，点击**执行 > 执行（串行）**。



- **方法 2**

在工具栏中，点击**执行（串行）** 图标 ()。



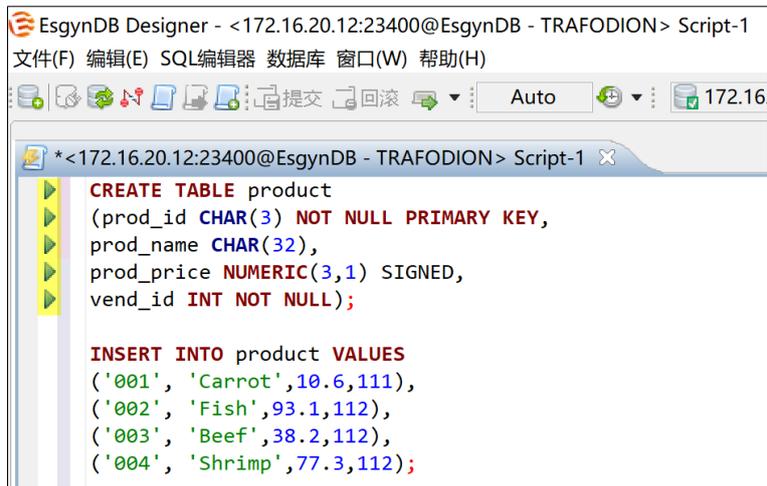
- 方法 3

在菜单栏中，点击 **SQL 编辑器 > 执行 (串行)**。

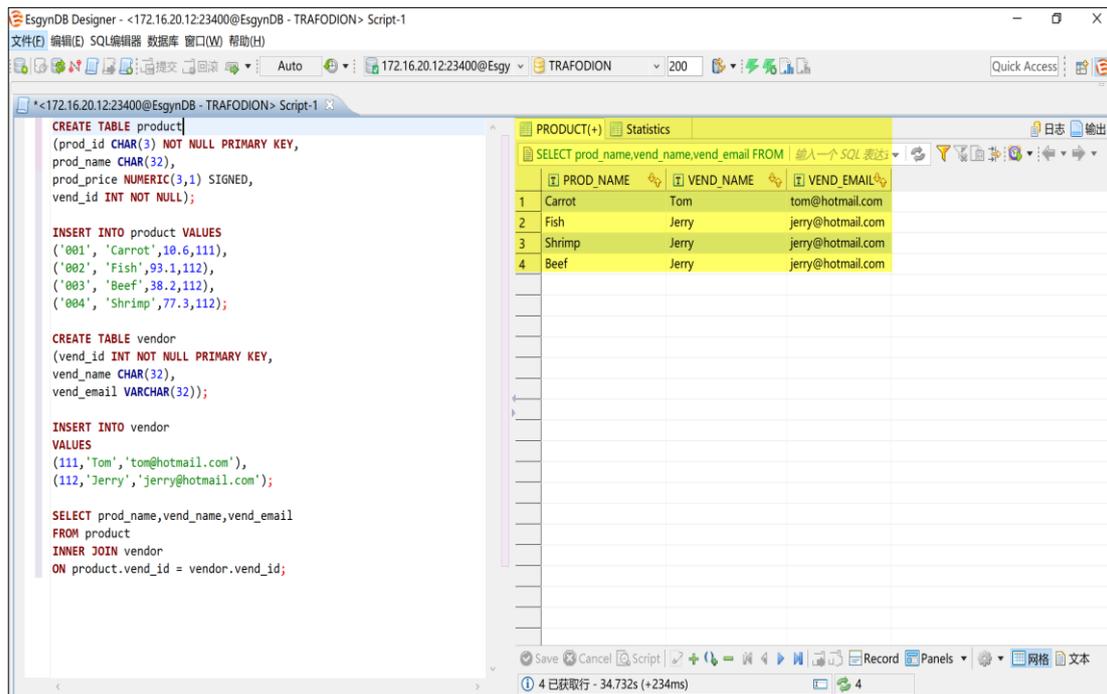


 提示

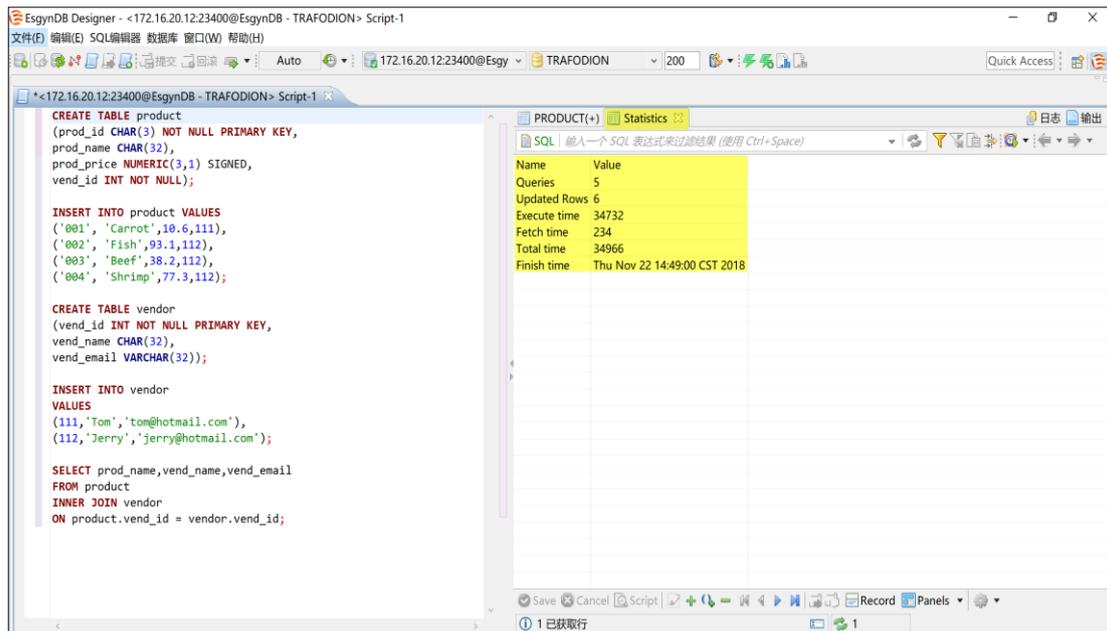
执行 SQL 脚本时，SQL 脚本的左列会显示上下移动的三角形图标 ()，表示该脚本正在执行。



4. 查看结果。



除了 SQL 脚本的执行结果, 执行(串行)还将生成统计信息, 即表 Statistics。



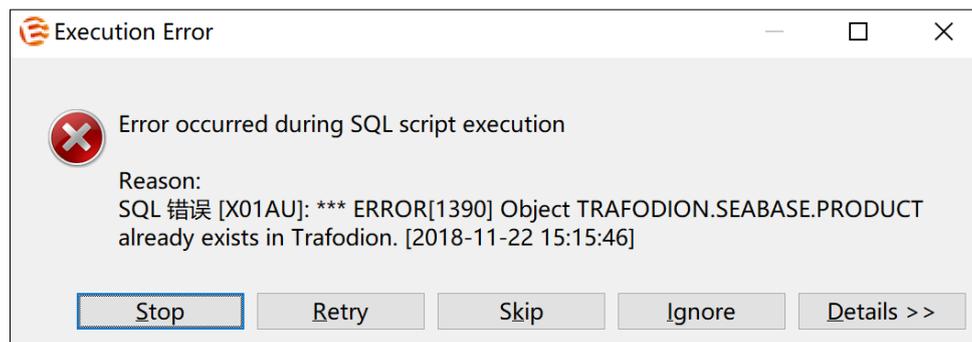
 提示

在生成结果前：

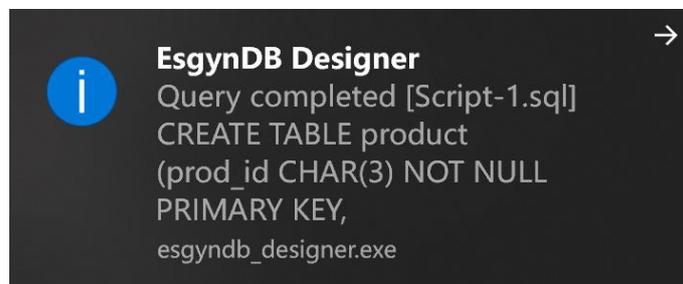
- 您离开了当前页面，并浏览该工具的其他页面，则：

执行成功的消息不会显示在该工具的其他页面上。

执行失败的消息将显示在该工具的其他页面上。



- 您离开了该工具的所有页面，则系统将发出**异步通知**，显示执行成功或失败的消息。



5.6.3.2 如何执行（并行）SQL 脚本

示例

本示例在创建 EsgynDB 连接后，执行以下多条 SQL 脚本，由于它们不存在前后依赖关系，所以选择执行（并行）方式。

```
SELECT prod_id,prod_price,prod_name
FROM product
ORDER BY prod_price DESC,prod_name ASC
LIMIT 3;

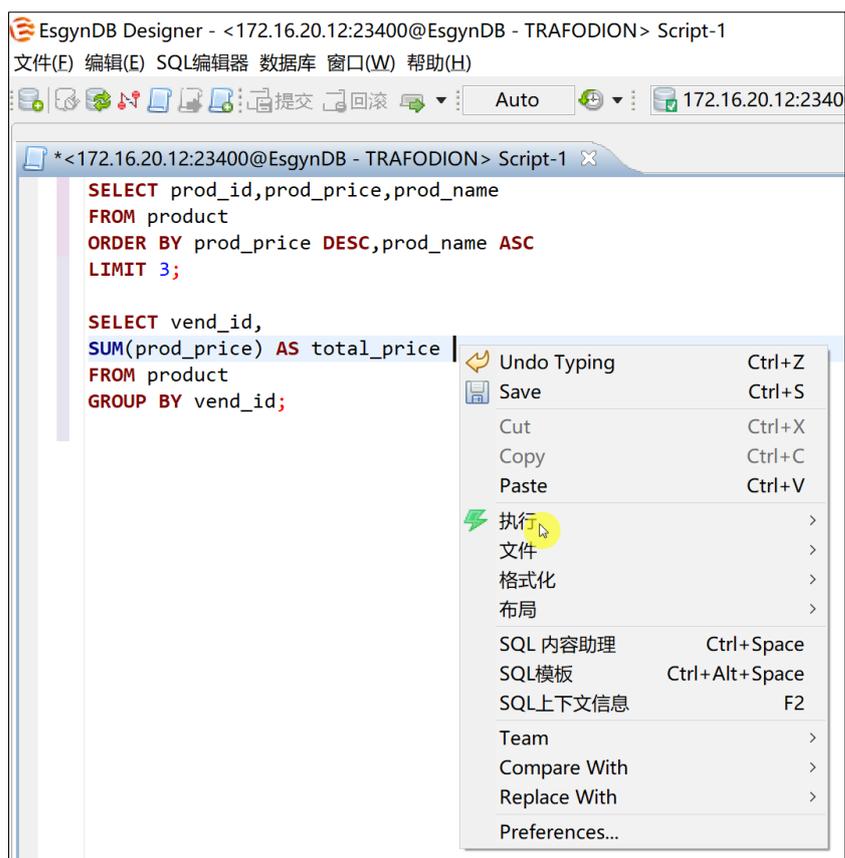
SELECT vend_id,
SUM(prod_price) AS total_price
FROM product
GROUP BY vend_id;
```

1. 在菜单栏中，点击 **SQL 编辑器 > 新建 SQL 编辑器**。
2. 在 SQL 编辑器窗口中，输入以上 SQL 脚本。
3. 点击 **执行（并行）**。

有以下 3 种方法点击 **执行（并行）**：

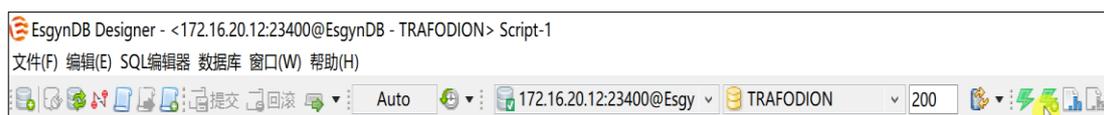
- **方法 1**

在 SQL 编辑器窗口，右键鼠标，在弹出的界面中，点击 **执行 > 执行（并行）**。



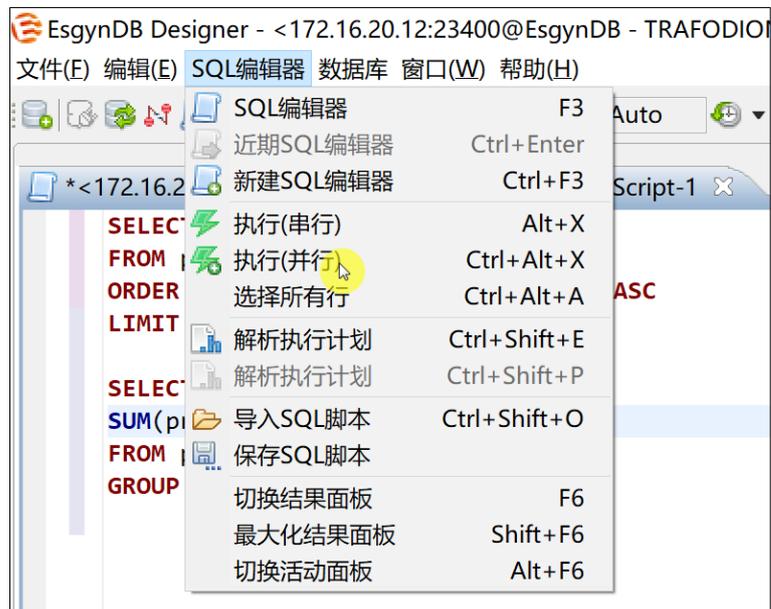
- 方法 2

在工具栏中，点击执行（并行）图标 ()。



- 方法 3

在菜单栏中，点击 SQL 编辑器 > 执行（并行）。

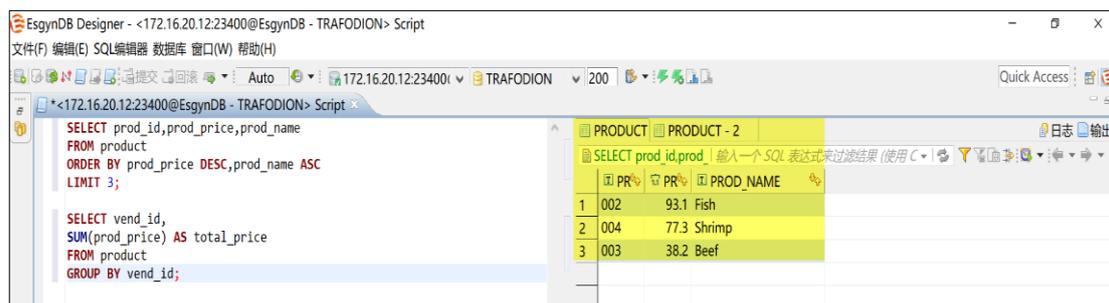


 提示

执行 SQL 脚本时，SQL 脚本的左列会显示上下移动的三角形图标 ()，表示该脚本正在执行。



4. 查看结果。



5.6.3.3 如何取消执行 SQL

在 SQL 执行过程中，Designer 会提示 SQL 执行的时间以及 Cancel 按钮，如需取消执行该 SQL，请点击

Cancel 。



5.6.4 如何处理 SQL 查询结果

5.6.4.1 使用查询结果页面的编辑选项



注意

如需使用所有编辑选项,则在新建 SQL 编辑器前,点击窗口 > 设置 > 结果集 > 勾选结果集是否可更新。该操作会降低性能。



在结果页面中,选中待处理的结果,右键鼠标,弹出的界面将显示编辑选项。

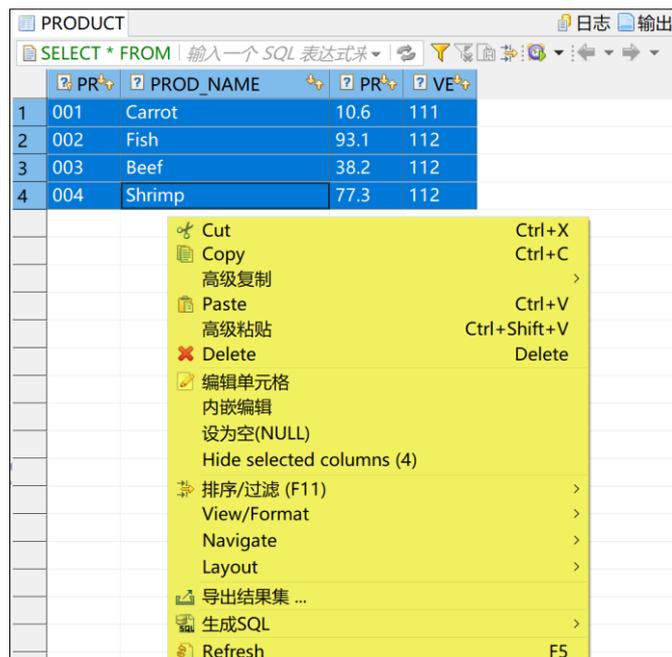


表 5-8 查询结果页面 - 编辑选项 1

查询结果页面 - 编辑选项		说明	快捷键
一级选项	二级选项		
剪切	/	/	Ctrl + X
复制	/	/	Ctrl + C
高级复制	复制字段名称	/	/
	复制行号	/	/
粘贴	/	/	/
高级粘贴	/	粘贴“高级复制”的内容	/
删除	/	/	Delete
编辑单元格	/	/	
内嵌编辑	/	直接在结果页面中编辑。	/
		 注意 编辑完成后，需点击左下方的保存图标。	
隐藏所选列	/	/	
排序/过滤	/	/	F11

表 5-9 结果页面 - 编辑选项 2

查询结果页面 - 编辑选项				
一级选项	二级选项	三级选项	四级选项	五级选项
查看/格式	查看	默认	/	/
		URL		
		二进制		
	结构复杂类型	/	/	/
	根据列分配颜色	/	/	/
	行颜色	/	/	/
	数据格式	SQL 编辑器	/	/
		元数据	元数据	为元数据打开单独的连接
				在 DDL 语句中启用大小写敏感
				显示表格行数
服务器端对象过滤	使用列名称而不是列标签			

表 5-10 查询结果页面 - 编辑选项 3

查询结果页面 - 编辑选项				
一级选项	二级选项	三级选项	四级选项	五级选项
查看/格式	数据格式	结果集	查询	查询集是否可更新
				最大结果集数
				自动获取下一段
				使用 SQL 限制获取大小
				服务器端结果排序
				读取表元数据 (唯一键)
				读取表引用 (外键)
				查询取消超时
			高级	使用获取大小
			其它	在 SQL 编辑器中保持光标
				发生错误时回滚
				使用将所有表列作为键
				在当前行之后插入新的行
				刷新后更新

表 5-11 查询结果页面 - 编辑选项 4

查询结果页面 - 编辑选项					
一级选项	二级选项	三级选项	四级选项	五级选项	六级选项
查看/格式	数据格式	结果集	二进制	二进制数据	二进制数据格式化程序
					二进制编辑器
					二进制串的最大长度
					在内存中保存的 LOB 的最大长度
					字符串编码
					缓存 CLOB 值
					缓存 BLOB 值
					LOB 缓存最大尺寸
				LOB	将 LONG 列作为 LOB 编辑
					文本编辑器的最大文本长度
					当应用编辑时提交会话
					当应用内容时提交会话

			数据格式	区域	语言
					国家
					变体 (暂不可用)
					区域
				格式化	类型
					设置
					示例

表 5-12 查询结果页面 - 编辑选项 4

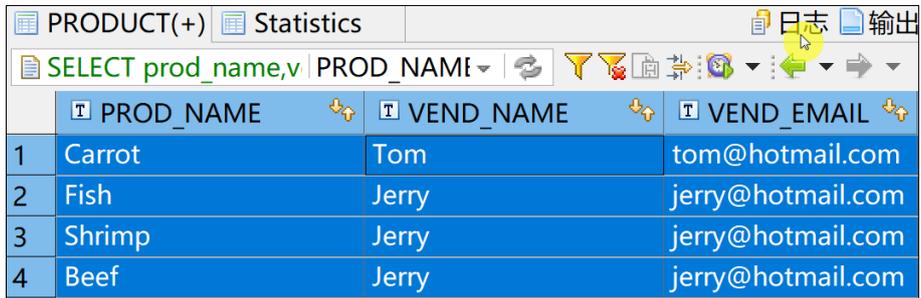
查询结果页面 - 编辑选项					
一级选项	二级选项	三级选项	四级选项	五级选项	六级选项
查看/格式	数据格式	结果集	表示	常见	在单/多行上切换到记录/表格模式
					在标题栏中显示列描述
					通过单元格的值计算列宽
					在状态中显示连接名
					复杂类型结构化
				表格	标记奇数或偶数行
					数字和时间左右对齐
					行批量大小
					显示单元格图标

5. SQL 编辑器

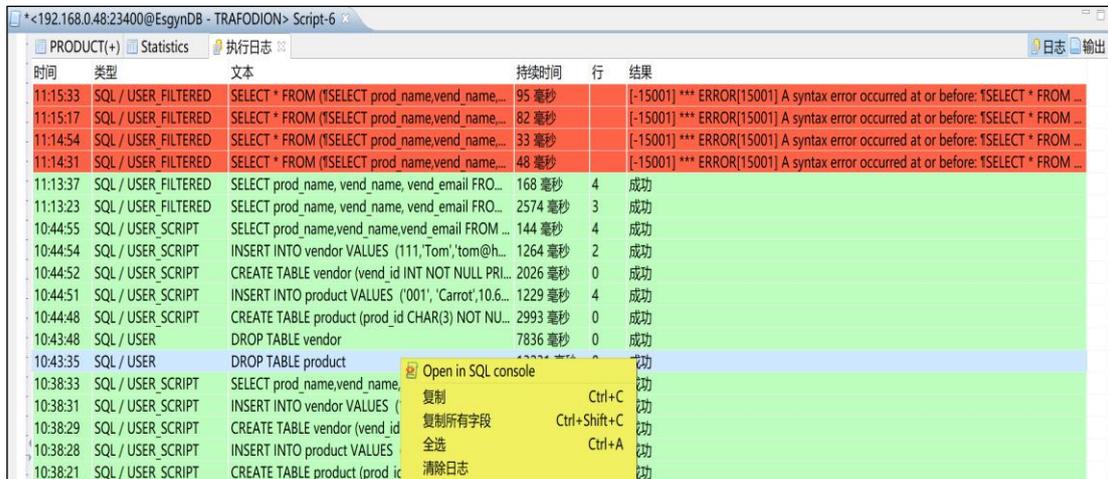
					列的最大长度
				纯文本	值格式
					行起始分隔符 (Leading Delimiter)
					行结尾分隔符 (Trailing Delimiter)

5.6.4.2 使用日志页面的编辑选项

1. 在结果页面中，点击页面右上方的日志，页面将显示所有日志。



2. 选中待编辑的条目，右键鼠标，页面将显示日志编辑选项。



日志编辑选项	快捷键
在新窗口查看 SQL 语句	/
复制	Ctrl + C
复制所有字段	Ctrl + Shift + C
全选	Ctrl + A
清除日志	/

5.6.5 如何解析 SQL 查询计划

您可以通过以下 3 种方法解析 SQL 查询计划：

- **方法 1**

在工具栏中，点击**解析执行计划** () 图标。

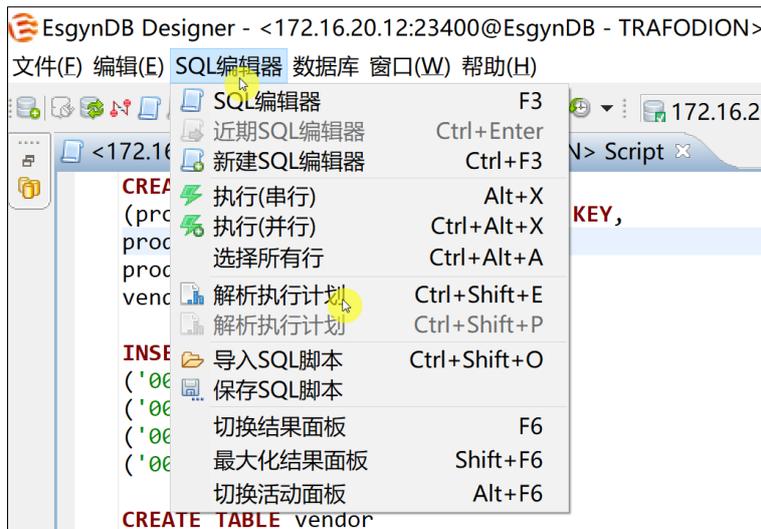


- **方法 2**

在 SQL 编辑器中，右键鼠标，选择**执行 > 解析执行计划**。

- **方法 3**

在菜单栏中，选择**SQL 编辑器 > 解析执行计划**。



结果如下所示：

ID	LID	RID	Opertion	Object	TotalCost	OperatorCost	DetailCost
-4	3		ROOT		0.003445197	0.0	CPU_TIME: 2.10434e-05 IO_TIME: 0.00341333 MSG_TIME: 1.08202e...
-3	2	1	HYBRID_HASH JOIN		0.003445197	6.080289E-6	CPU_TIME: 2.10434e-05 IO_TIME: 0.00341333 MSG_TIME: 1.08202e...
2			TRAFODION_SCAN	"TRAFODION"."SEABASE"."VENDOR"	0.0017210547	0.0017210547	CPU_TIME: 8.39299e-06 IO_TIME: 0.00170667 MSG_TIME: 5.99499e...
1			TRAFODION_SCAN	"TRAFODION"."SEABASE"."PRODUCT"	0.001718062	0.001718062	CPU_TIME: 6.57012e-06 IO_TIME: 0.00170667 MSG_TIME: 4.82524e...

5.6.6 如何导出查询结果

查询结果能导出至：

- 文件。

更多信息，请参阅 [5.6.6.1 如何导出查询结果至文件](#)。

- 数据库表。

更多信息，请参阅 [5.6.6.2 如何导出查询结果至数据库表](#)。

5.6.6.1 如何导出查询结果至文件

查询结果能导出至 **xml**、**json**、**html** 和 **csv** 文件。



注意

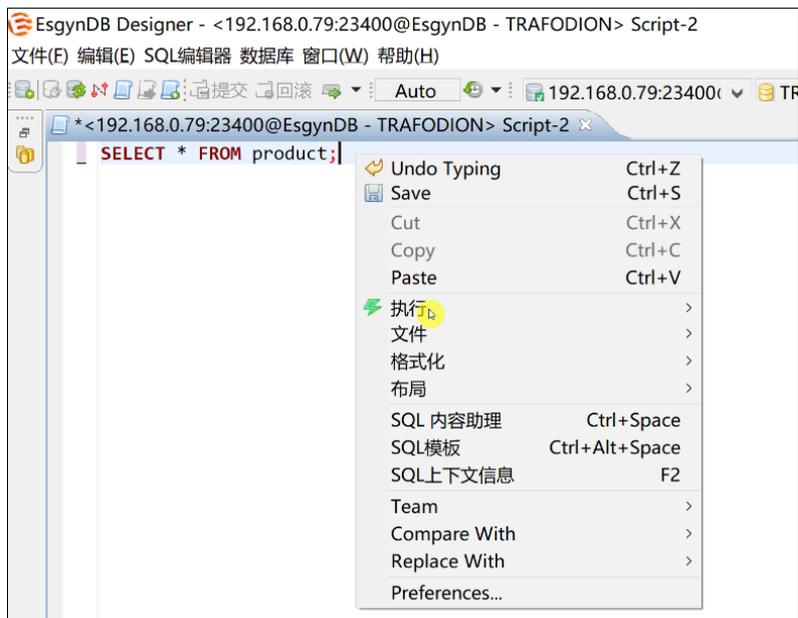
该功能仅适用于 **SELECT** 语句的查询结果。



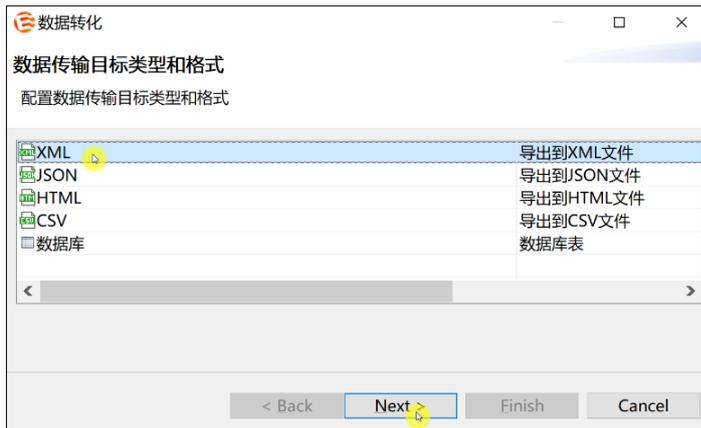
示例

本示例将 **SELECT * FROM product** 的查询结果导出至 **xml** 文件。

1. 在 SQL 编辑器中，右键鼠标，选择 **执行 > 从查询中导出**。



2. 选择 XML，点击下一步 (Next)。



3. 设置以下选项，点击下一步 (Next)。

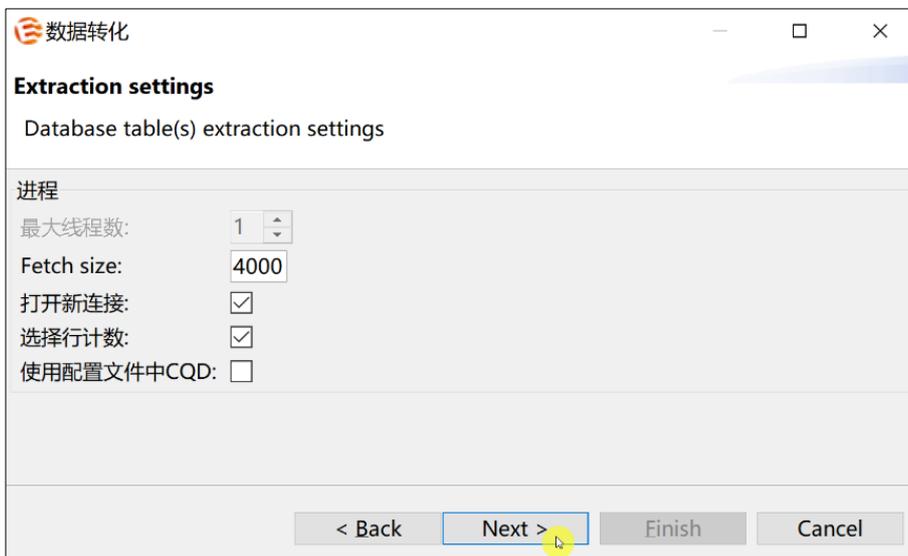


表 5-13 导出查询结果 - 设置 1

选项	说明
最大线程数	/
Fetch size	每次获取的行数。
打开新连接	导出查询结果使用新连接，在导出查询结果的同时，您能在数据库中执行其它操作。
选择行计数	状态栏显示任务的执行情况。

使用配置文
件中的
CQD

您能在 `cqd-export.properties` 文件中指定 CQD，该文件位于易鲸捷 Designer 安装程序的根目录。

```
cqd-export.properties ●  
1 # This file is a file that stores CQD parameters when the data is exported.  
2 # eg:  
3 # cq1=cqd parallel_num_esps '32'  
4 # cq2=cqd MDAM_SCAN_METHOD 'off'  
5 # cq3=.....  
6
```

4. 设置以下选项，点击下一步 (Next)。

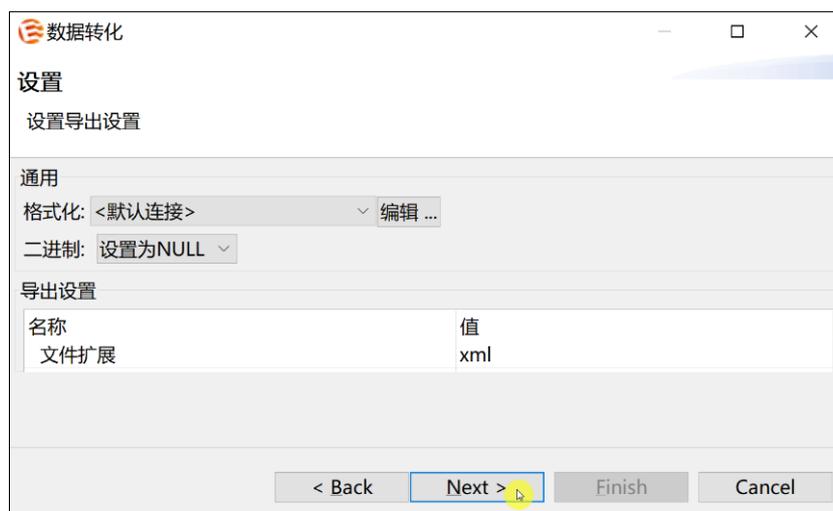


表 5-14 导出查询结果 - 设置 2

设置选项				
一级选项	二级选项	三级选项	四级选项	五级选项
通用	格式化	编辑	配置	管理配置
			区域	语言
				国家
				区域
			格式化	日期
				时间
				时间戳
	二进制	设置为 NULL	/	/
		保存到文件	/	/
		行内	/	/
导出设置	文件扩展 ³	/	/	

³ 如需更改导出文件类型，双击值，输入文件扩展名，支持 xml、json、html 和 csv。

5. 设置以下选项，点击下一步 (Next)。

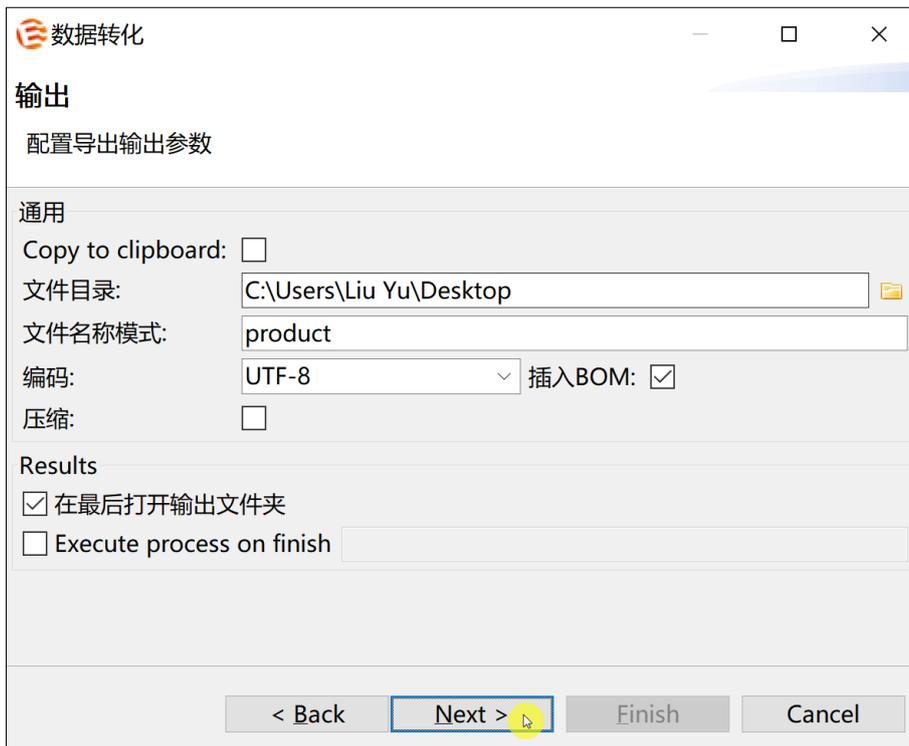
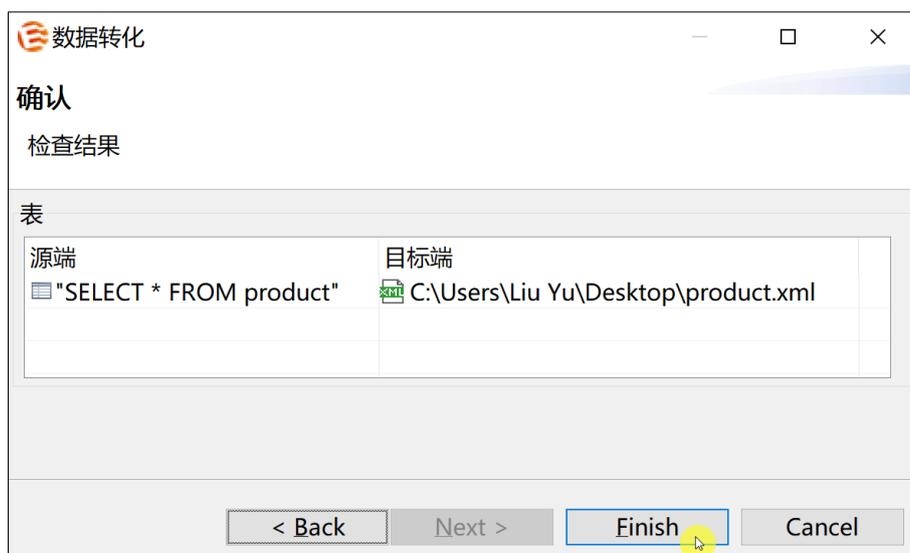


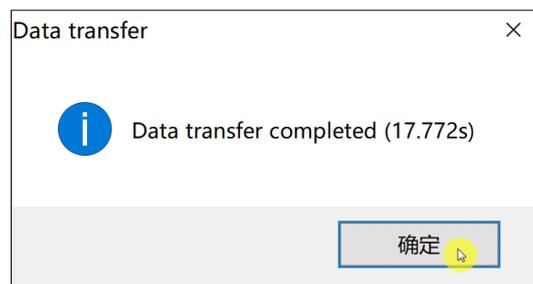
表 5-15 导出查询结果 - 设置 3

设置选项		说明
一级选项	二级选项	
通用	复制	复制查询结果。
	文件目录	/
	文件名称模式	默认文件名称模式为 \${table}_\${timestamp}，您也可以直接 输入文件名。该示例的文件名为 <i>product</i> 。
	编码	
	插入 BOM	插入字节顺序标记。
	压缩	将查询结果打包成 zip 文件。
结果	在最后打开输出文件夹	/

6. 确认源端和目标端信息，点击**完成 (Finish)**。



状态栏将显示数据导出进度。导出成功后，易鲸捷 Designer 将显示提示消息和导出用时。



以下是导出结果:

```
product.xml ●
1  <?xml version="1.0" ?>
2  <!DOCTYPE SELECT_FROM_product [
3  | <!ELEMENT SELECT_FROM_product (DATA_RECORD*)>
4  | <!ELEMENT DATA_RECORD (PROD_ID?,PROD_RATE?,PROD_DELIVERY?)+>
5  | <!ELEMENT PROD_ID (#PCDATA)>
6  | <!ELEMENT PROD_RATE (#PCDATA)>
7  | <!ELEMENT PROD_DELIVERY (#PCDATA)>
8  | ]>
9  <SELECT_FROM_product>
10 | <DATA_RECORD>
11 |   <PROD_ID>001</PROD_ID>
12 |   <PROD_RATE>67.21</PROD_RATE>
13 |   <PROD_DELIVERY>2018-11-02</PROD_DELIVERY>
14 | </DATA_RECORD>
15 | <DATA_RECORD>
16 |   <PROD_ID>002</PROD_ID>
17 |   <PROD_RATE>30.48</PROD_RATE>
18 |   <PROD_DELIVERY>2018-12-24</PROD_DELIVERY>
19 | </DATA_RECORD>
20 | <DATA_RECORD>
21 |   <PROD_ID>003</PROD_ID>
22 |   <PROD_RATE>12.45</PROD_RATE>
23 |   <PROD_DELIVERY>2018-12-31</PROD_DELIVERY>
24 | </DATA_RECORD>
25 | <DATA_RECORD>
26 |   <PROD_ID>004</PROD_ID>
27 |   <PROD_RATE>90.34</PROD_RATE>
28 |   <PROD_DELIVERY>2019-01-01</PROD_DELIVERY>
29 | </DATA_RECORD>
30 </SELECT_FROM_product>
```

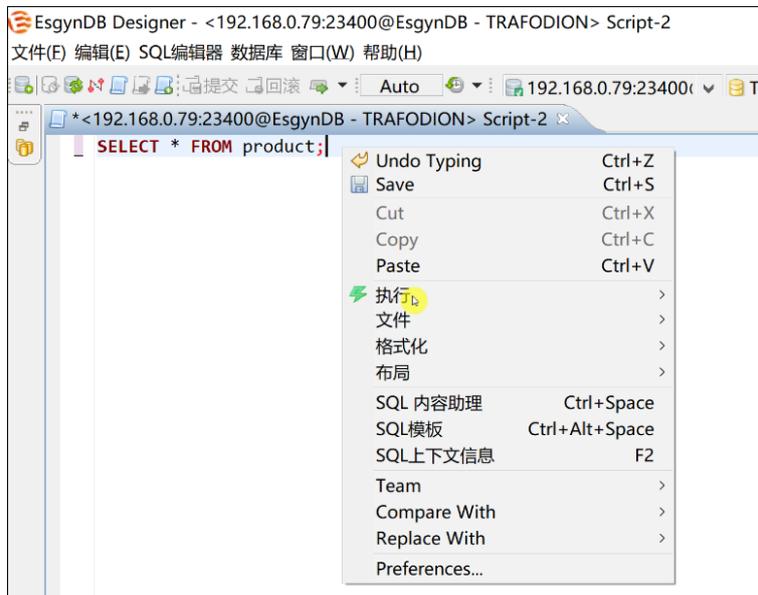
5.6.6.2 如何导出查询结果至数据库表

源表的查询结果能导出至新建的表和已存在的表。

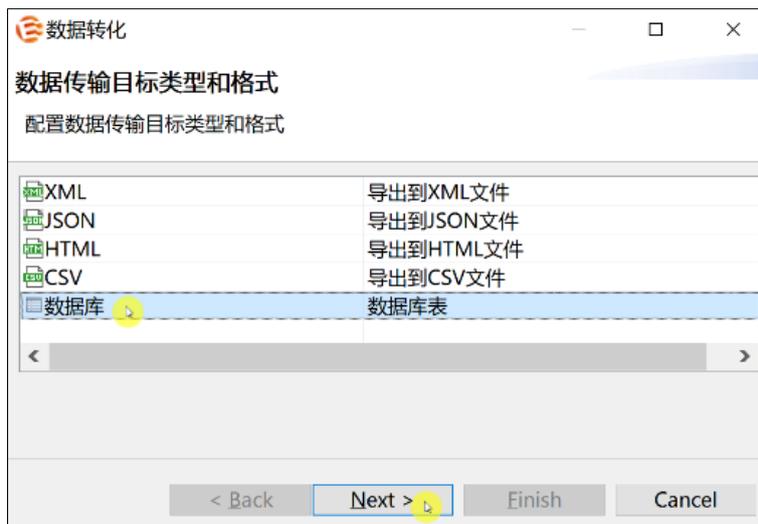
示例

本示例将源表 *product* 的查询结果导出至新表 *computer*。

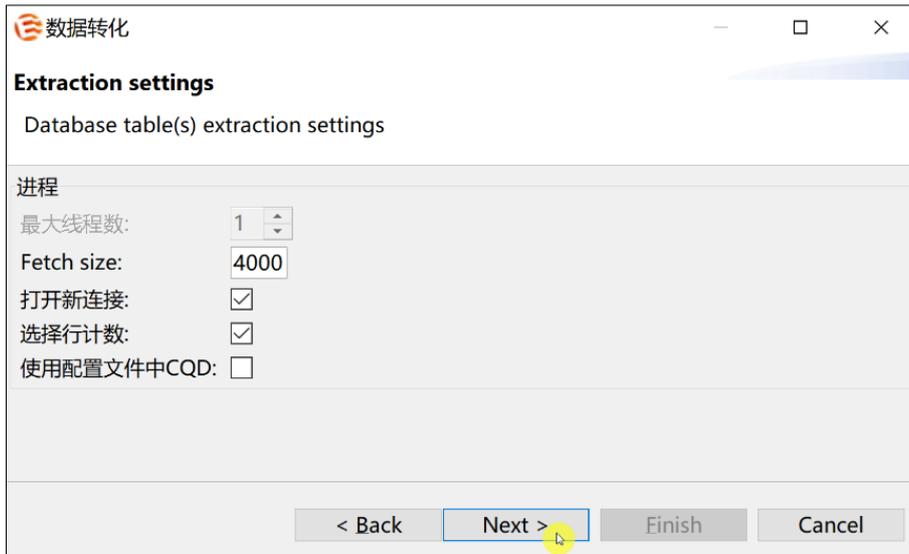
1. 在 SQL 编辑器中，输入 SQL 查询语句后，右击鼠标，选择执行 > 从查询中导出。



2. 选择数据库，点击下一步 (Next)。

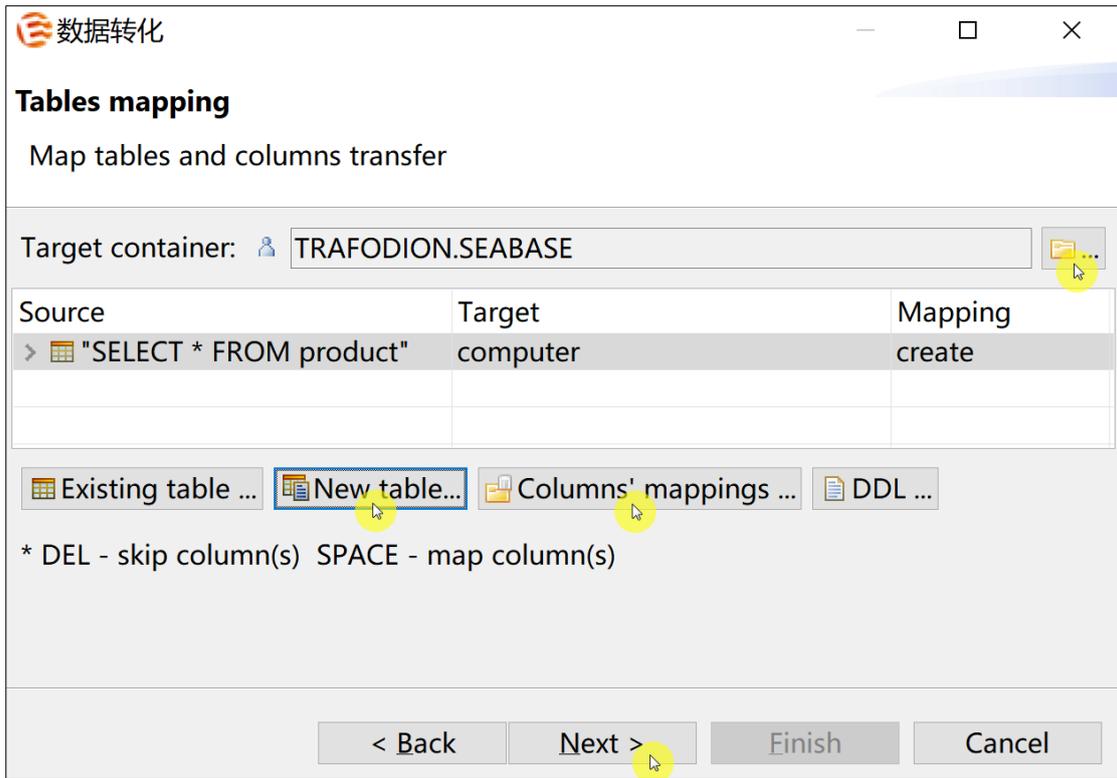


3. 设置导出选项，点击下一步 (Next)。

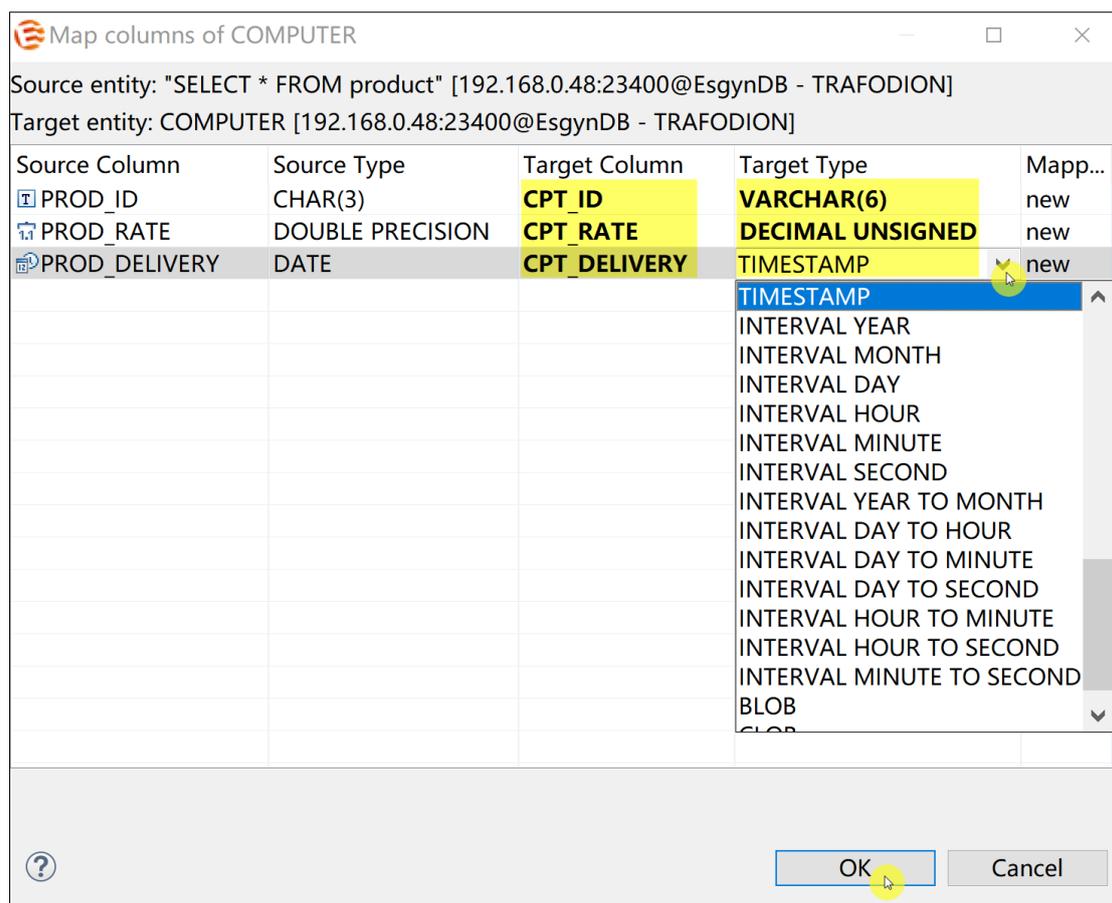


选项	说明
最大线程数	/
Fetch size	每次获取的行数。
打开新连接	导出查询结果使用新连接，在导出查询结果的同时，您能在数据库中执行其它操作。
选择行计数	易鲸捷 Designer 右下方显示任务进度。 如果数据较小，可能不会显示任务进度。
使用配置文件中的 CQD	您能在 <code>cqd-export.properties</code> 文件中指定 CQD，该文件与易鲸捷 Designer 启动文件保存在相同位置。 <pre> cqd-export.properties ● 1 # This file is a file that stores CQD 2 # parameters when the data is exported. 3 # eg: 4 # cq1=cqd parallel_num_esps '32' 5 # cq2=cqd MDAM_SCAN_METHOD 'off' 6 # cq3=..... </pre>

4. 设置目标表。

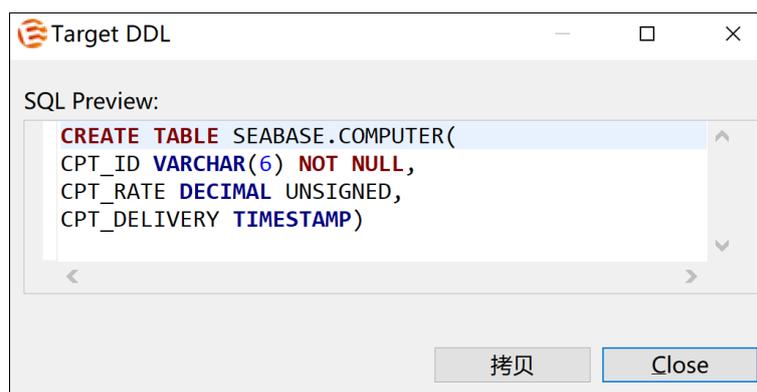


- (1) 点击打开 () 图标，选择目标表保存位置，本示例为 *SEABASE*。
- (2) 点击新建表 (New table)，输入目标表表名，本示例为 *computer*。
- (3) 点击列映射 (Columns' mappings)。
- (a) 如需重命名目标表列名，在目标列 (Target Column) 中输入列名。
- (b) 如需自定义映射规则，在目标类型 (Target Type) 中输入列类型，或在下拉列表中选择列类型。
- (4) 点击 **OK**。



提示

如需查看目标表的表结构，点击 **DDL**。



6. 设置数据加载，点击下一步 (Next)。

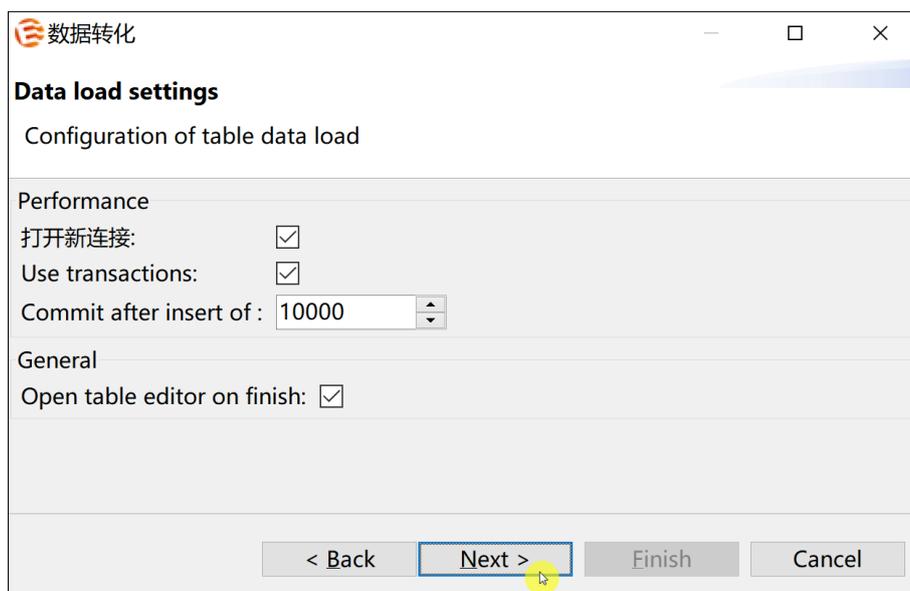
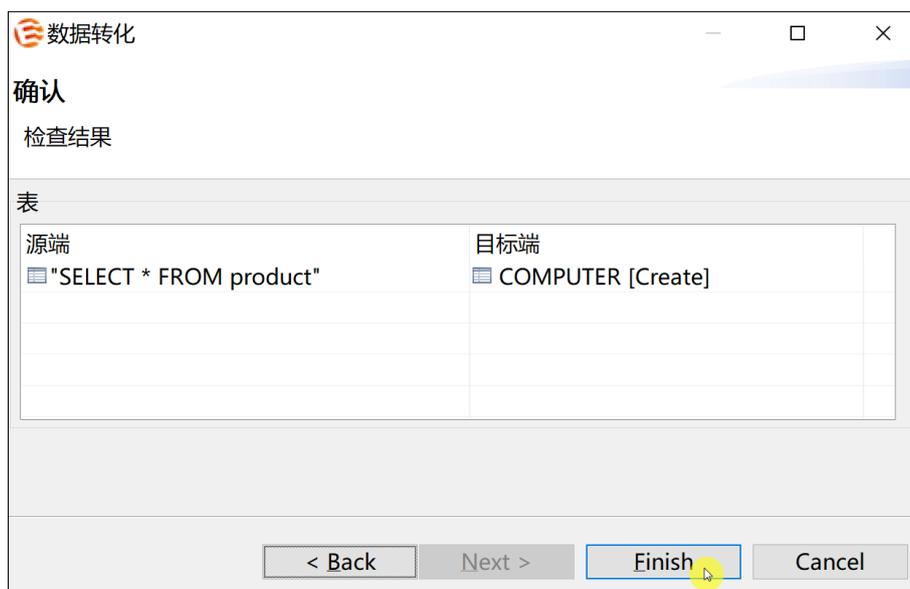


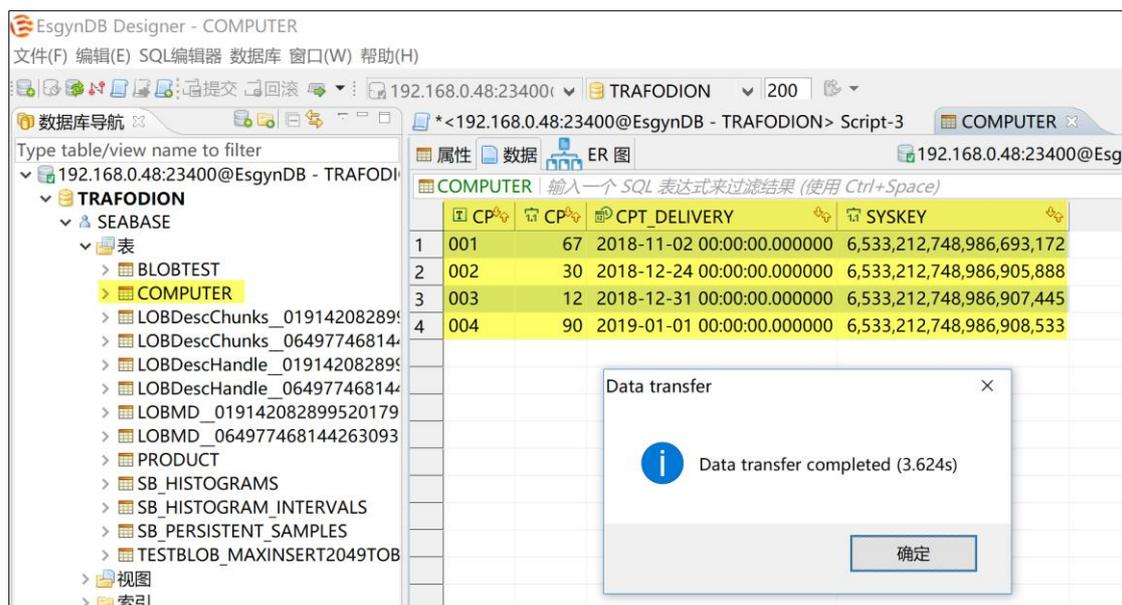
表 5-16 数据加载 - 设置

选项		说明
一级选项	二级选项	
性能	打开新连接	导出数据使用新连接。在导出数据的同时，您能在易鲸捷 Designer 中执行其它操作。
	使用事务	/
	插入 n 行后提交事务	事务具备 ACID 特性，使用多个事务能提高数据安全性。
通用	完成后打开数据库表	/

7. 点击完成 (Finish)。



状态栏将显示数据导出进度。导出成功后，易鲸捷 Designer 将显示提示消息、导出用时和新建表 *computer* 及其数据。

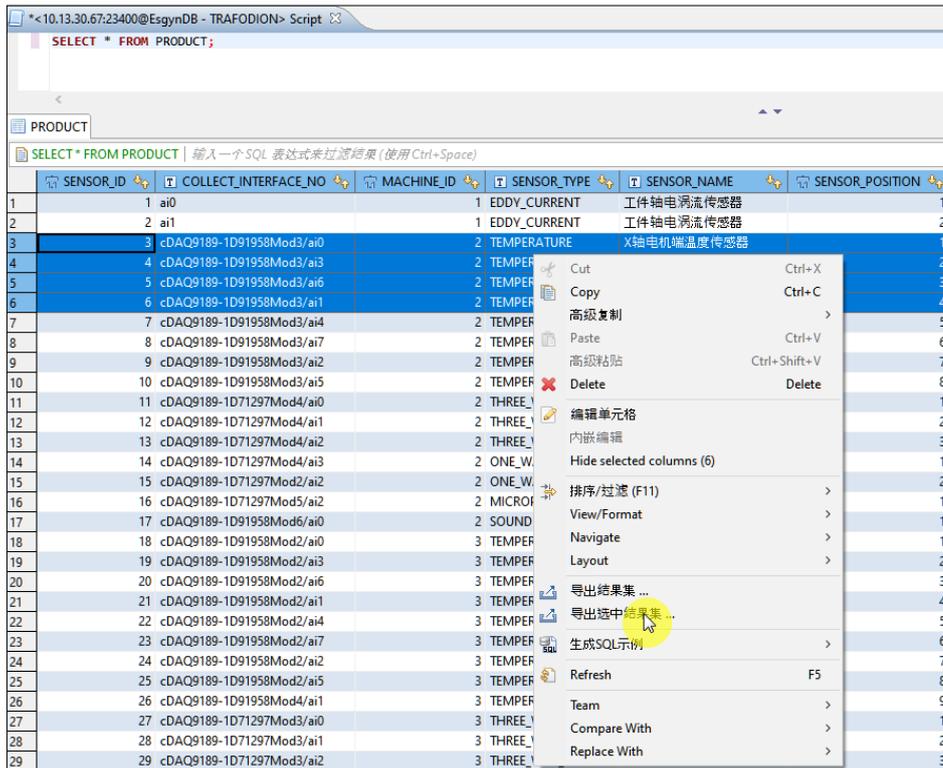


5.6.6.3 如何导出部分结果集

示例

本示例导出 SELECT * FROM product 查询结果的部分结果集。

1. 在 SQL 编辑器中，执行 SQL，在结果集窗口中选中一些行，点击导出选中结果集。



2. 后续步骤与 5.6.6.1 章节相同，依次点击下一步 (Next) 即可。

5.6.7 如何导出表数据至 SQL 文件

查询结果能导出至：

- SQL 文件。

更多信息，请参阅 [5.6.7.1 如何导出表数据至 SQL 文件](#)

5.6.7.1 如何导出表数据至 SQL 文件

表数据能导出至 xml、json、html、csv 和 SQL 文件，导出至 xml、json、html 和 csv 已经在 5.7.5.1 章节介绍，这里不再赘述，本节主要介绍导出表数据至 SQL 文件。



注意

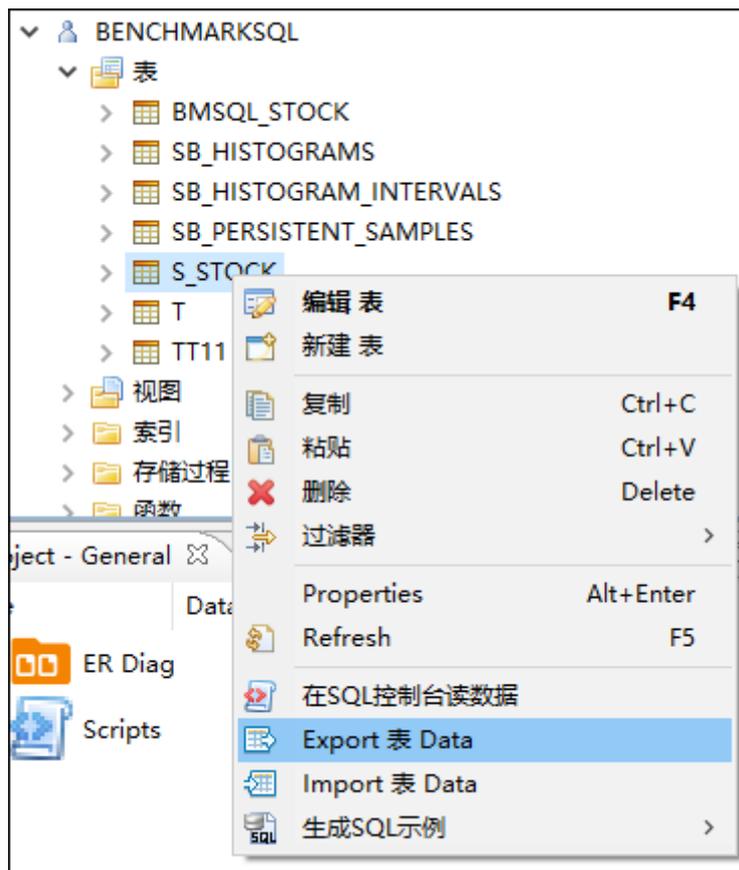
该功能仅适用于表数据的导出。



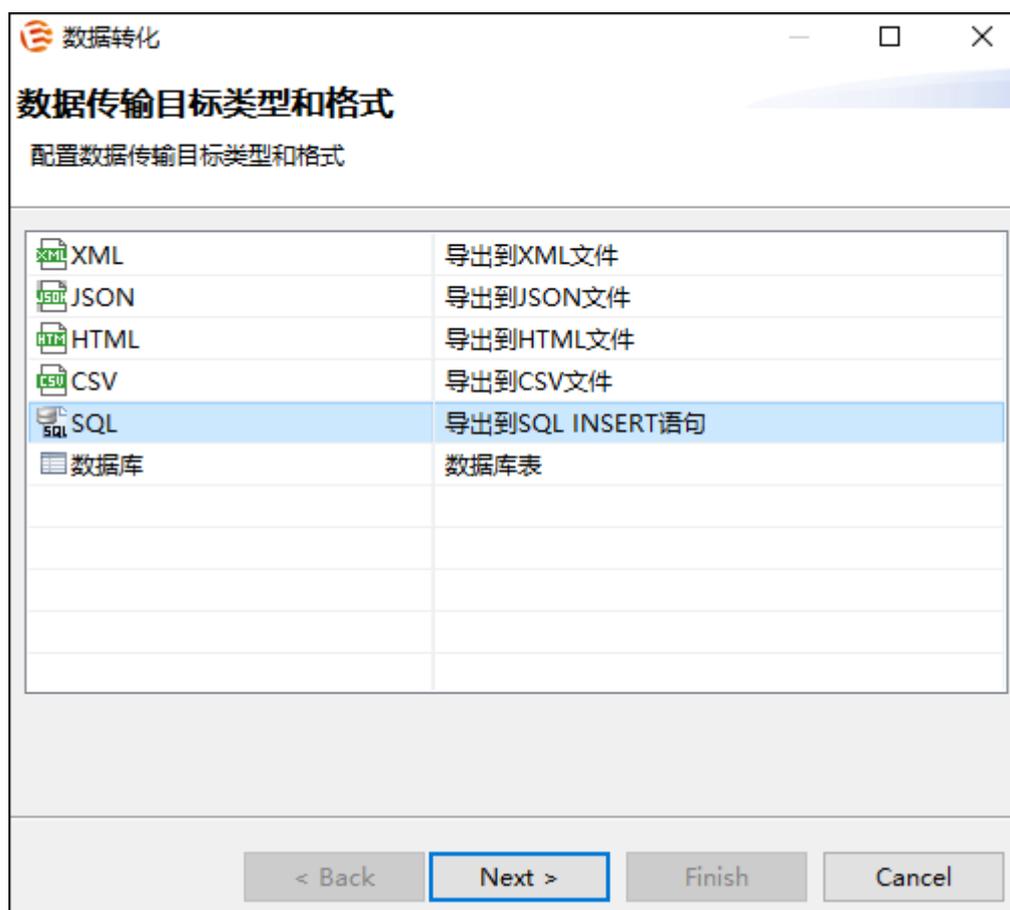
示例

本示例将表 S_STOCK 的数据导出至 SQL 文件。

1. 在导航中，右键表 S_STOCK，选择 > Export 表 Data。



2. 选择 SQL，点击下一步 (Next)。



3. 后续步骤与 5.6.6.1 章节相同，依次点击下一步 (Next) 即可。

6. 表管理

本章讲述以下内容:

6.1 如何创建表

6.2 如何查看表

6.3 如何编辑表

6.4 如何删除数据库对象

6.5 如何导出表的数据

6.6 如何导入表的数据

6.1 如何创建表

您可以通过以下 2 种方法创建表：

- **方法 1**

在 **SQL 编辑器** 中执行 SQL 语句创建表。更多信息，请参阅 [5.6SQL 脚本执行](#)

如何创建 SQL 脚本。

- **方法 2**

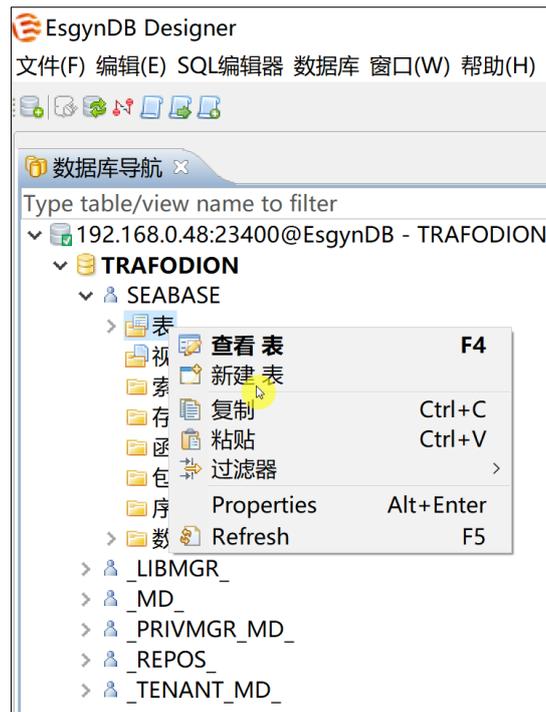
在 **数据库导航窗口** 中创建表。更多信息，请参阅以下内容。



示例

本示例在创建表 *prodcut* 时指定了主键、外键、Salt, Range 和 HBase 选项；在表 *product* 存在后，创建了索引、唯一约束和 Check 约束。

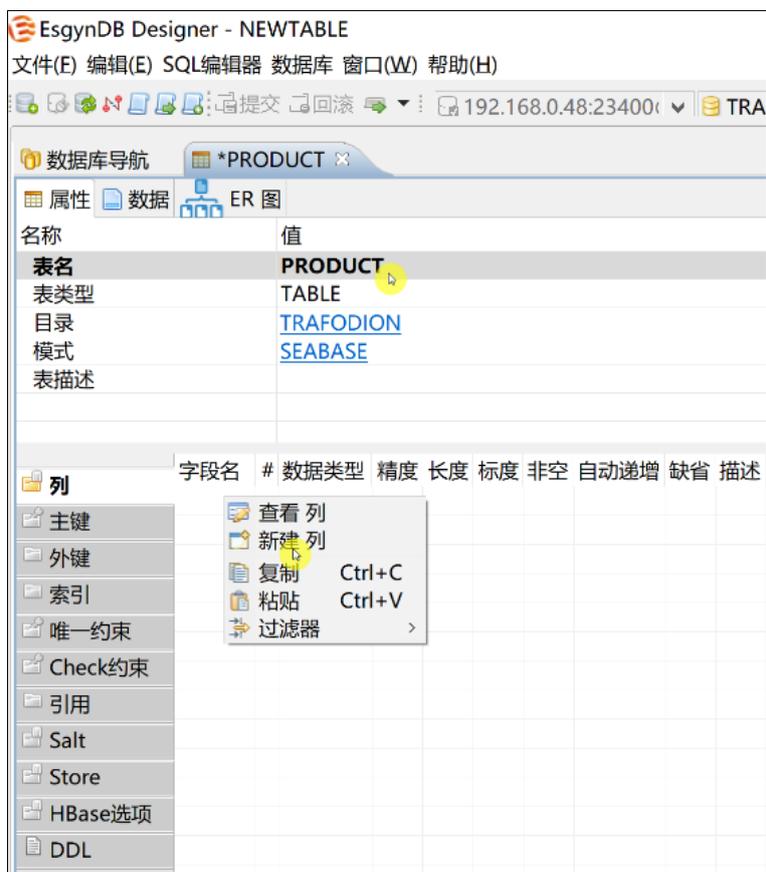
1. 在 **数据库导航窗口** 中，右键表，点击 **新建表**。



2. 输入表名 *product*。

3. 创建列。

(1) 点击列，右键空白区域，点击新建列。



(2) 输入字段名和数据类型等信息。

列	字段名	#	数据类型	列字符集	长度	标度	非空	自动递增	缺省	描述
主键	PRODUCT_ID	-1	SMALLINT UN...		0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
外键	PROD_NAME	-1	VARCHAR	UTF8	32		<input type="checkbox"/>	<input type="checkbox"/>		
索引	DELIVERY_TIME	-1	DATE		0		<input type="checkbox"/>	<input type="checkbox"/>		
	VEND_ID	-1	SMALLINT UN...		0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		



注意

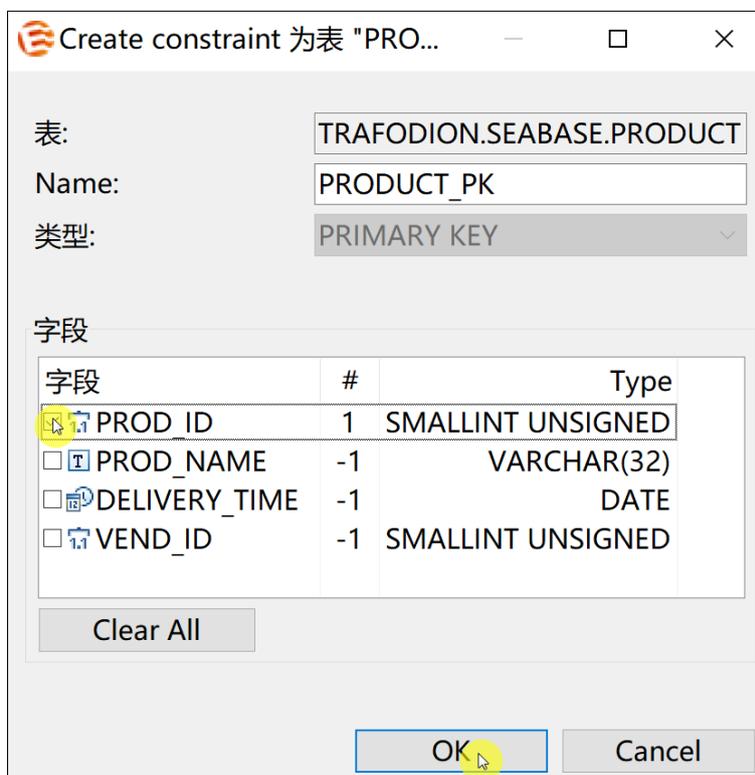
- 勿手动输入数据类型，应在下拉列表中选择数据类型。
- #列为字段的序号，创建表时默认为-1，保存表后会自动更新。
- 只能在创建表时指定 Salt、Store、Range 和 HBase 选项。
- 只能为已存在的表创建索引。EsgynDB 创建索引的 SQL 语法是 **CREATE INDEX Statement**⁴。
- 您可以通过命令行工具（例如，TrafiCI 和 SQLCI）在创建表时指定唯一约束和 Check 约束，但 EsgynDB 实际上是通过 **ALTER TABLE ... ADD CONSTRAINT UNIQUE** 和 **ALTER TABLE ... ADD CONSTRAINT CHECK (Condition)** 指定唯一约束和 Check 约束⁵。因此，在易鲸捷 Designer 中，只能为已存在的表指定唯一约束和 Check 约束。
- 当数据类型为 CHAR/VARCHAR 时，需要指定列字符集属性，默认为 UTF8，可以从下拉框中选择其他选项，目前提供的选择有 ISO88591、UCS2 和 UTF8。

⁴ 更多关于 CREATE INDEX Statement 的信息，请参阅《EsgynDB SQL 参考手册》。

⁵ 更多关于 ALTER TABLE Statement 的信息，请参阅《EsgynDB SQL 参考手册》。

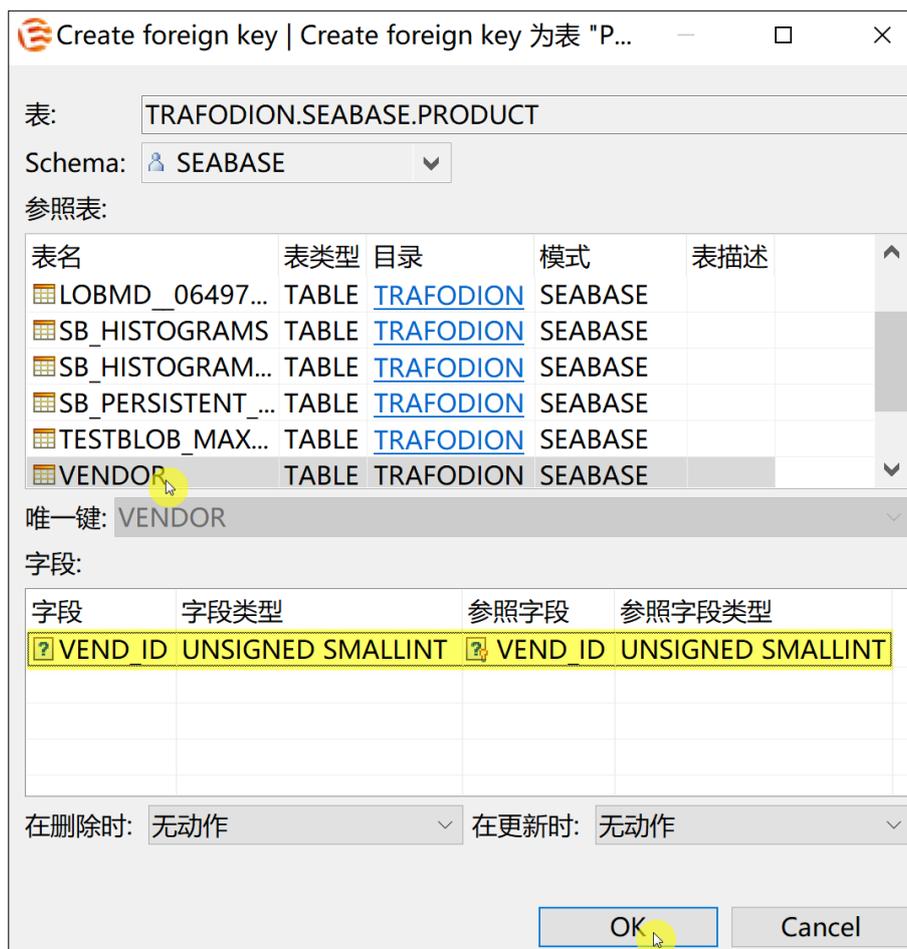
4. 指定主键。

- (1) 点击主键，右键空白区域，点击新建主键。
- (2) 在弹出的对话框中，勾选字段 **PROD_ID**。
- (3) 点击 **OK**。



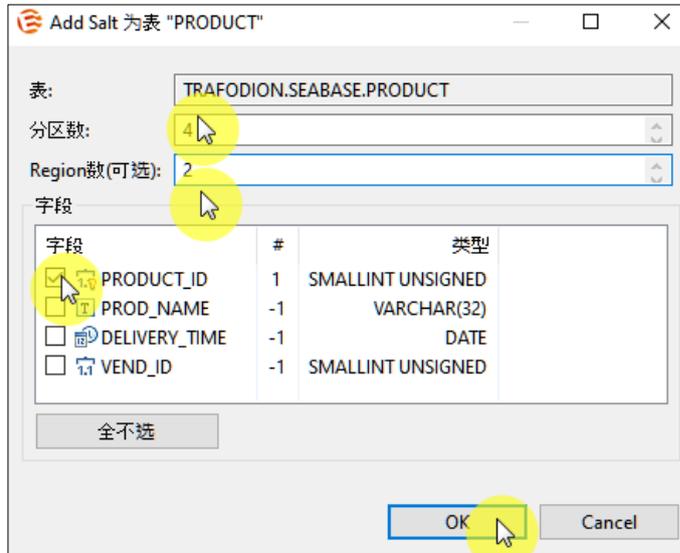
5. 指定外键。

- (1) 点击外键，右键空白区域，点击新建外键。
- (2) 在弹出的对话框中，选择表 **VENDOR**，易鲸捷 Designer 将显示自动匹配的字段。
- (3) 点击 **OK**。



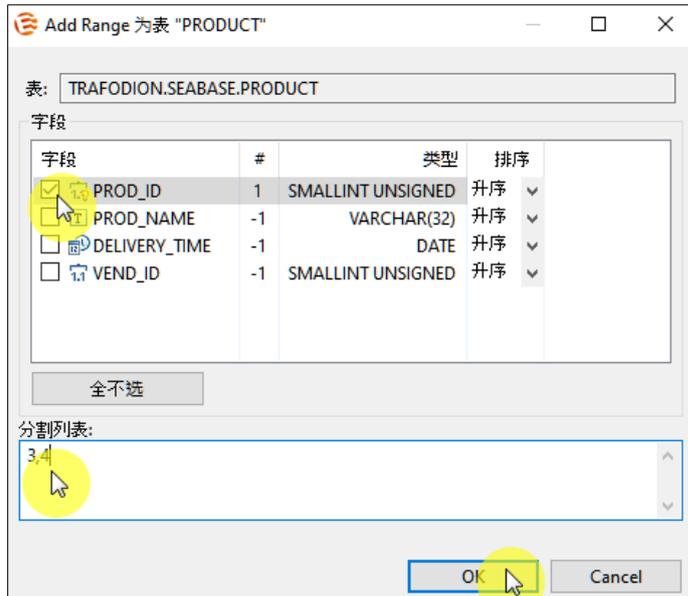
6. 指定 Salt。

- (1) 点击 **Salt**，右键空白区域，点击**新建 Salt**。
- (2) 在弹出的对话框中，在**分区数**中输入 **4**，**Region 数(可选)**输入 **2**，选择**字段 PROD_ID**。
- (3) 点击 **OK**。



7. 指定 **Range** （不支持同时指定 Salt 和 Range）

- (1) 点击 **Range**，右键空白区域，点击**新建 Range**。
- (2) 在弹出的对话框中，选择**字段 PROD_ID**，分割列表中输入 **3,4**。
- (3) 点击 **OK**。



8. 指定 HBase 选项。

- (1) 点击 **HBase 选项**。
- (2) 点击 **COMPRESSION**，在下拉列表中，选择 **SNAPPY**。
- (3) 点击 **DATA_BLOCK_ENCODING**，在下拉列表中，选择 **FAST_DIFF**。
- (4) 点击 **MEMSTORE_FLUSH_SIZE**，输入 **1073741824**。

列	Hbase选项名	Hbase选项值	描述
主键	BLOCKCACHE		启用hbase块缓存,默认值是true
	BLOCKSIZE		hbase块大小,默认值是64*1024,范围是(1024, 64 *1024*1024)
外键	BLOOMFILTER		布隆过滤器,用来提高随机读(get)的性能,默认值是ROW
索引	CACHE_BLOOMS_ON_WRITE		在写入布隆过滤器时将其加到缓存中,默认值是false
唯一约束	CACHE_DATA_ON_WRITE		在写数据时将其加到缓存中,默认值是false
Check约束	CACHE_INDEXES_ON_WRITE		在写索引时将其加到缓存中,默认值是false
引用	COMPACT		启用hbase压缩,默认值是true
	COMPACT_COMPRESSION		hbase压缩格式,默认值是NONE
Salt	COMPRESSION	SNAPPY	hbase压缩格式,默认值是NONE
Store	DATA_BLOCK_ENCODING	FAST_DIFF	hbase数据块编码格式,默认值是NONE
	DURABILITY		hbase持久性策略,默认值是SYNC_WAL
HBase选项	EVICT_BLOCKS_ON_CLOSE		关闭时将块从缓存中移除,默认值是false
DDL	IN_MEMORY		启用IN MEMORY缓存,默认值是false
	KEEP_DELETED_CELLS		保留删除的数据,默认值是false
	MAX_FILESIZE		HStoreFile的最大值,默认值是10*1024*1024*1024,范围是(2*1024*1024, 9223372036854775807)
	MAX_VERSIONS		指定保留数据的最大版本号,默认值是1
	MEMSTORE_FLUSH_SIZE	1073741824	MEMSTORE FLUSH阈值,默认值是1024*1024*128,范围是(1024*1024, 9223372036854775807)
	MIN_VERSIONS		查询时返回的最少版本号(超时的数据也会返回),默认值是0
	PREFIX_LENGTH_KEY		表key的最大长度,只在SPLIT_POLICY为KeyPrefixRegionSplitPolicy时才起作用,默认值是2
	REPLICATION_SCOPE		值为1时启用副本,值为0时禁用副本,默认值是1
	SPLIT_POLICY		hbase regin分裂策略
	TTL		列簇的存活阈值(Time To Live),默认值是-1



提示

- 如需查看 HBase 选项及其有效值和默认值，请参阅 [附录 2. HBase 选项及其有效值和默认值](#)。
- 更多关于如何在建表时指定 HBase 选项的信息，请参阅《EsgynDB SQL 参考手册》。

6. 表管理

9. 点击窗口右下方的**保存** () 图标。

10. 在弹出的窗口中，确认 SQL 语句无误，点击**保留**。



此时，表 *product* 创建成功。



注意

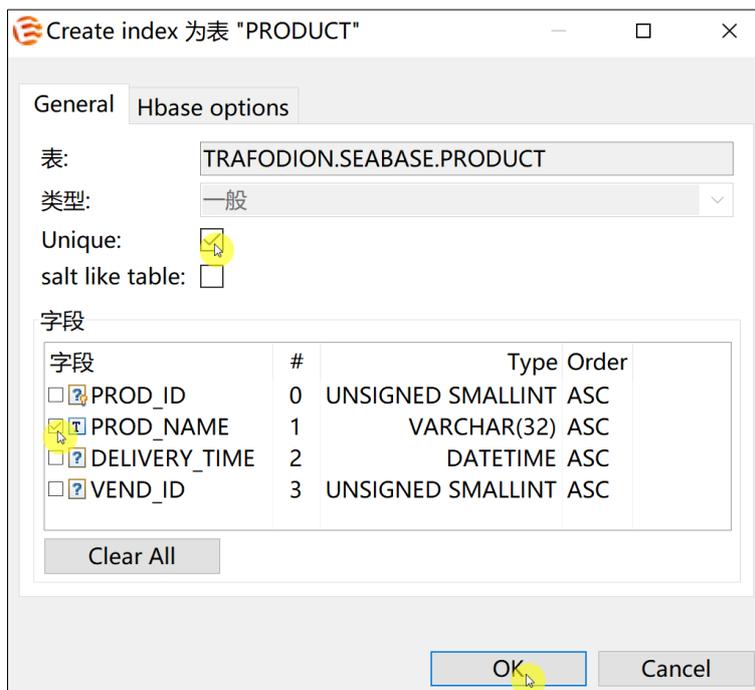
只能为已有的表创建索引、唯一约束和 Check 约束。

11. 创建索引。

(1) 点击索引，右键空白区域，点击**新建索引**。

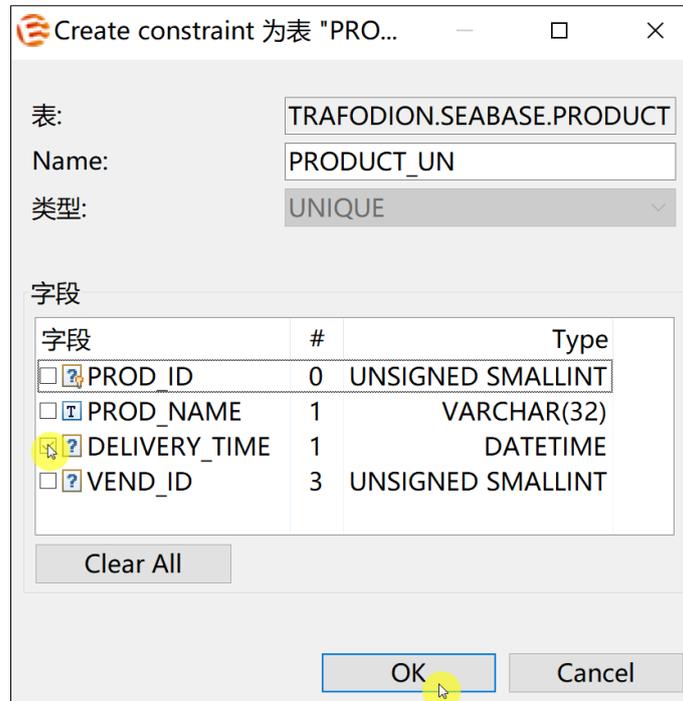
(2) 在弹出的对话框中，选择**唯一**和字段 **PROD_NAME**。

(3) 点击 **OK**。

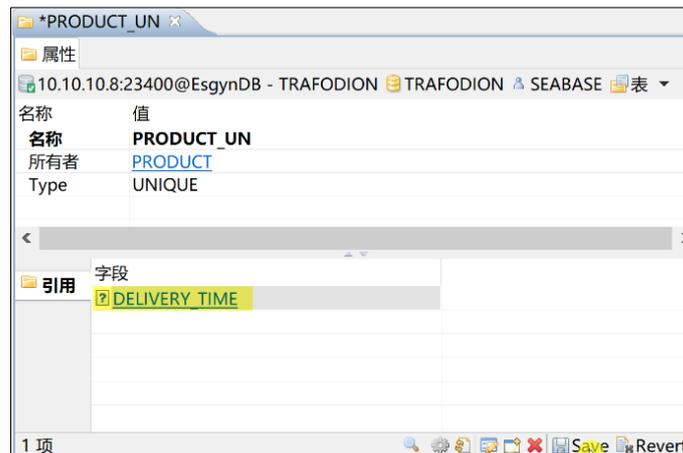


12. 创建唯一约束。

- (1) 点击**唯一约束**，右键空白区域，点击**新建唯一约束**。
- (2) 在弹出的对话框中，选择字段 **DELIVERY_TIME**，点击 **OK**。



- (3) 在弹出的对话框中，确认信息无误，点击**保存** () 图标。

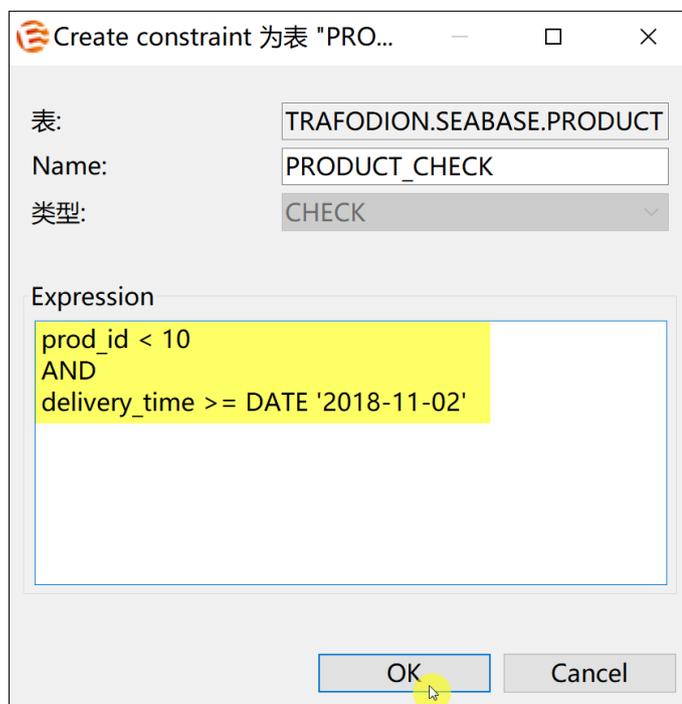


- (4) 在弹出的对话框中，确认 SQL 语句无误，点击**保留**。

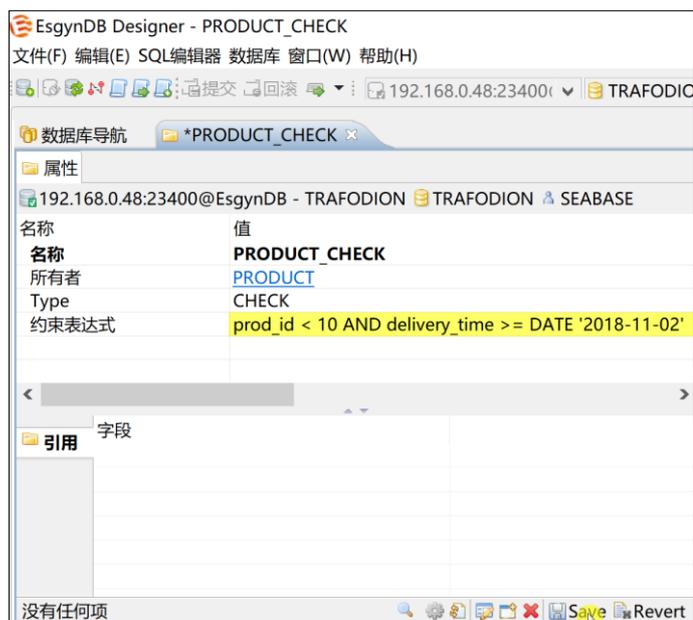


13. 创建 Check 约束。

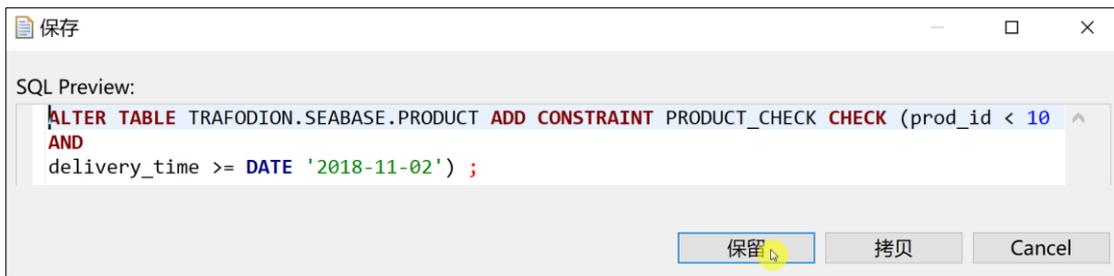
- (1) 点击 **Check 约束**，右键空白区域，点击**新建 Check 约束**。
- (2) 在弹出的对话框中，输入 **prod_id < 10 AND delivery_time >= DATE '2018-11-02'**。



- (3) 确认信息无误，点击**保存** () 图标。

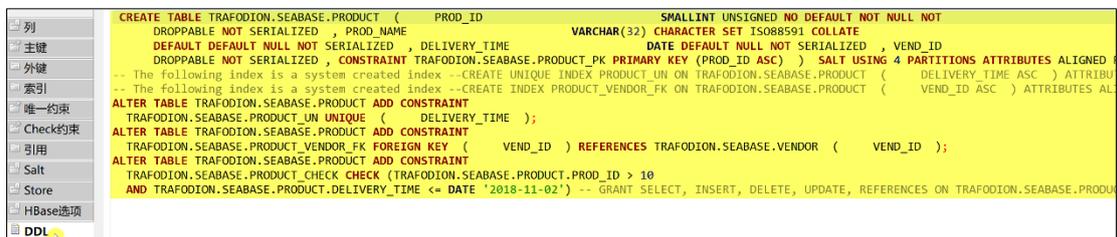


(4) 在弹出的对话框中，确认 SQL 语句无误，点击保留。



提示

- 如需查看表结构，点击 DDL。



- 如需将 DDL 导出至文件（支持多种格式，包括.sql 和.txt），点击保存  图标。
- 如需编辑 DDL 语句，点击打开  图标，DDL 语句将在 SQL 编辑器中打开，您能编辑并执行 SQL 语句。

总结

至此，在创建表 *product* 时指定了主键、外键、Salt、Range 和 HBase 选项；在表 *product* 成功创建后，指定了索引、唯一约束和 Check 约束。

6.2 如何查看表

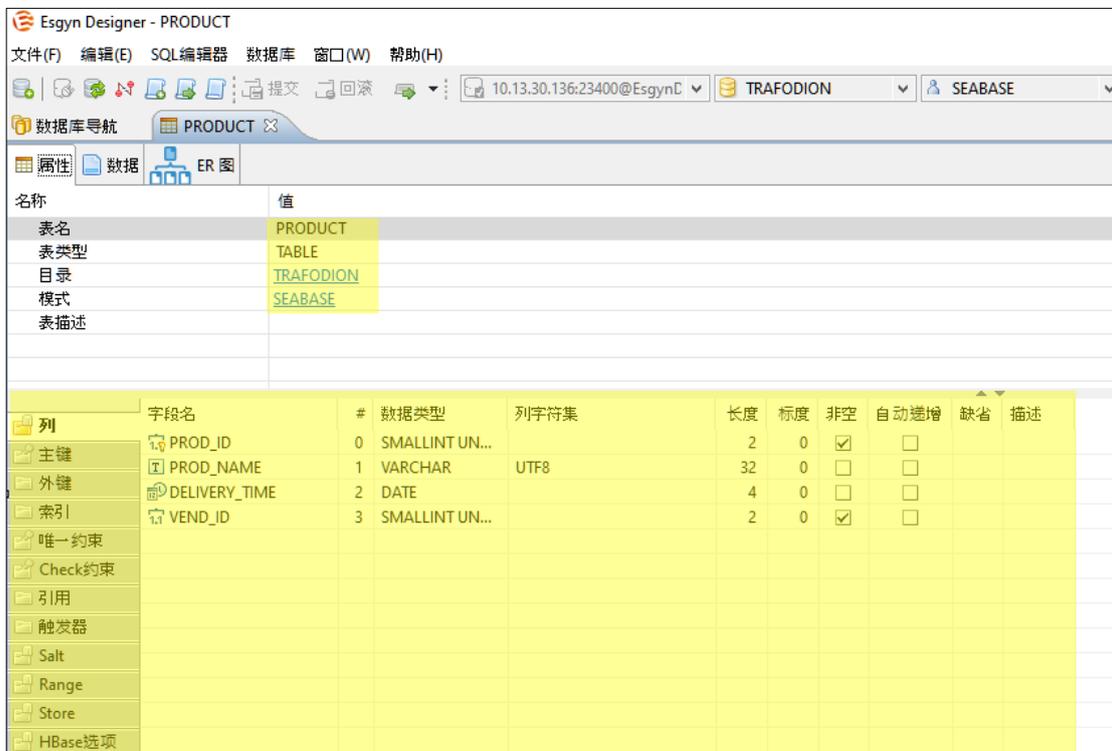
您能查看表的属性、数据和 ER 图。

6.2.1 如何查看表的属性

1. 在数据库导航窗口中，双击表 *product*。



2. 点击属性。





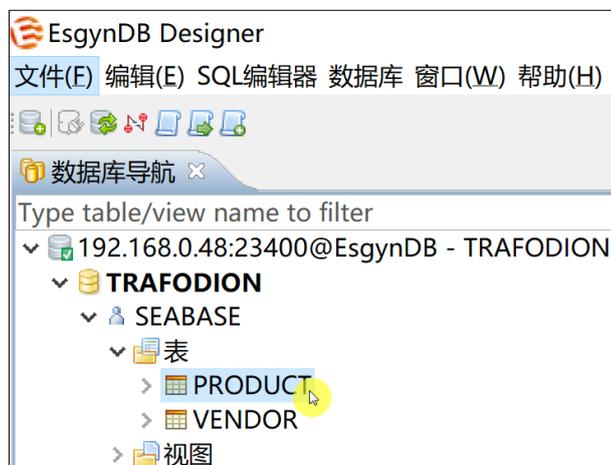
提示

如需查看/编辑数据库连接、查看 Catalog/Schema/表，点击下图中高亮的快捷方式。

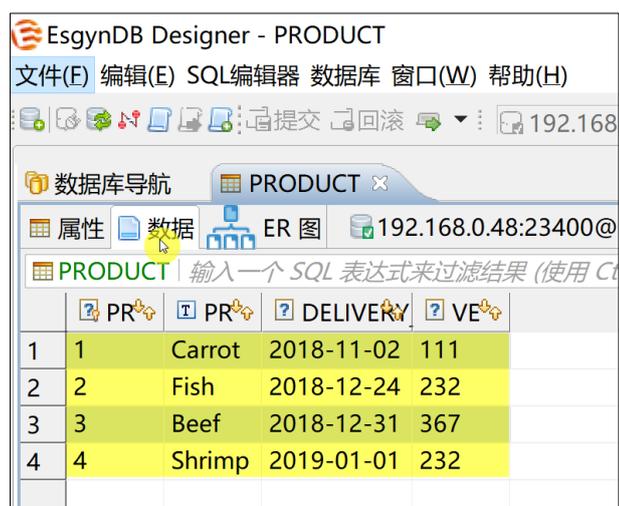
列	字段名	#	数据类型	列字符集	长度	精度	非空	自动递增	缺省	描述
主键	PROD_ID	0	SMALLINT UN...		2	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
外键	PROD_NAME	1	VARCHAR	UTF8	32	0	<input type="checkbox"/>	<input type="checkbox"/>		
索引	DELIVERY_TIME	2	DATE		4	0	<input type="checkbox"/>	<input type="checkbox"/>		
唯一约束	VEND_ID	3	SMALLINT UN...		2	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Check约束										
引用										
触发器										
Salt										
Range										
Store										
HBase选项										

6.2.2 如何查看表的数据

1. 在数据库导航窗口中，双击表 *product*。



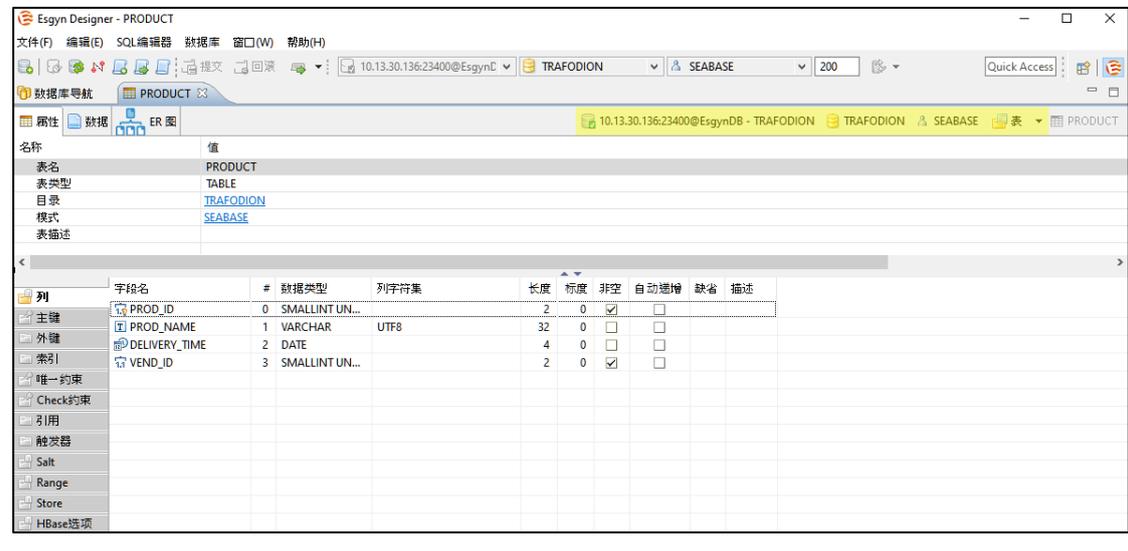
2. 点击数据。





提示

如需查看/编辑数据库连接、查看 Catalog/Schema/表，点击下图中高亮的快捷方式。



6.2.3 如何查看表的 ER 图

ER 图⁶描述表与表之间的关系，它显示表的属性和主外键关系等信息。

📄 示例

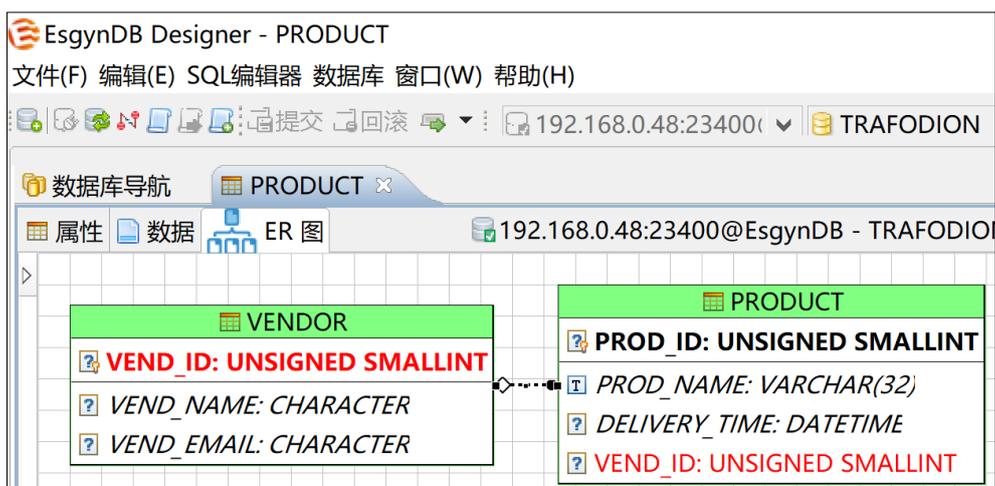
本示例查看表 *product* 的 ER 图。

1. 在数据库导航窗口中，双击表 *product*。



2. 点击 **ER 图**。该图显示了：

- 表 *product* 与其关联表（表 *vendor*）的属性（字段名称和字段类型）
- 表 *product* 与表 *vendor* 的关系（表 *product* 的 *vend_id* 是表 *vendor* 的外键）



⁶ ER 图（Entity Relationship Diagram），即**实体关系图**，提供了表示实体类型、属性和联系的方法，用于描述现实世界的概念模型。

6.3 如何编辑表

您能编辑表的属性、数据和 ER 图。

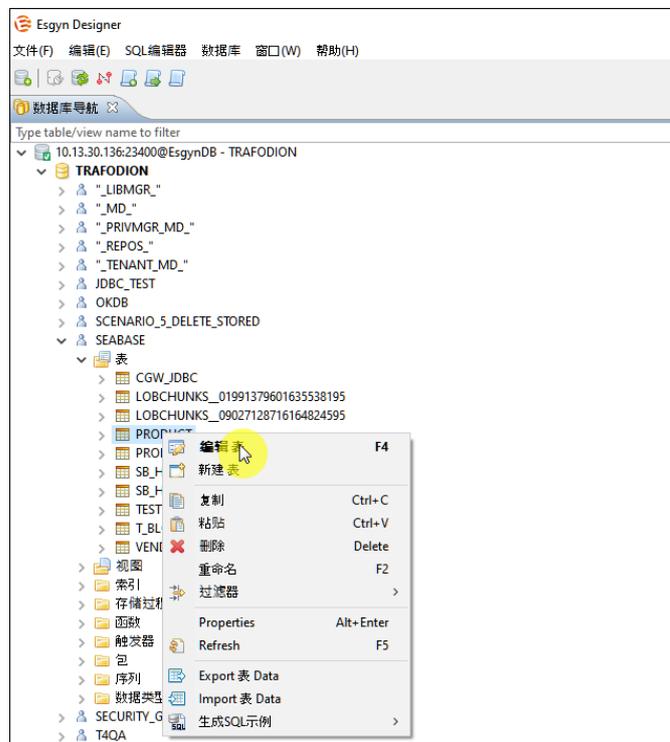
6.3.1 如何编辑表的属性

您可以通过以下 2 种方法编辑表的属性：

- 方法 1：直接编辑表的属性
- 方法 2：通过 ER 图编辑表的属性

方法 1：直接编辑表的属性

1. 在数据库导航窗口中，右键表 *product*。
2. 点击编辑表 > 属性。





注意

- 以下是对表进行增删改操作的注意事项。

能增加的属性	<ul style="list-style-type: none"> • 外键 • 索引 • 唯一约束 • Check 约束
不能增加的属性	<ul style="list-style-type: none"> • 主键 • Salt • Store • HBase 选项
能删除的属性	<ul style="list-style-type: none"> • 外键 • 索引 • 唯一约束 • Check 约束
不能删除的属性	<ul style="list-style-type: none"> • 主键 <hr/> <p> 注意</p> <p>在 EsgynDB 中：</p> <p>表存在后，不能修改或删除主键，所以，如需修改主键，只能删除表再重新建表⁷。</p> <p>一张表只能具备一个主键⁸。</p> <hr/>

⁷更多关于如何创建和修改表的信息，请参阅《EsgynDB SQL 参考手册》的 CREATE TABLE Statement 和 CREATE TABLE Statement 章节。

⁸更多关于主键的信息，请参阅《EsgynDB SQL 参考手册》的主键章节。

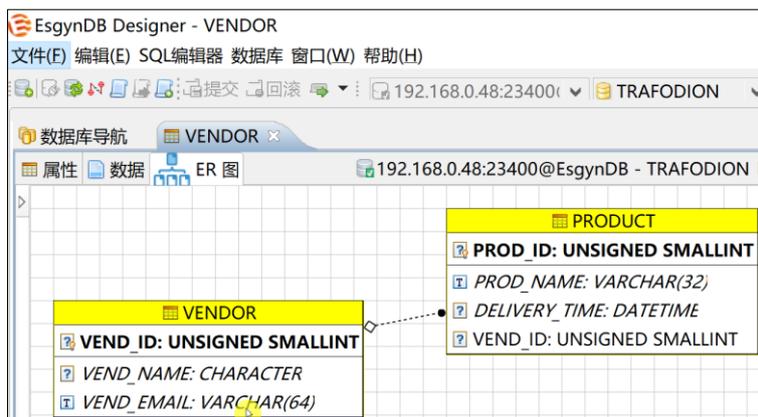
能修改(值)的属性	<ul style="list-style-type: none">• 字段名• 数据类型
不能修改(值)的属性	<ul style="list-style-type: none">• 表的 Catalog• 表的 Schema• 表名• 主键• 数据类型的长度• Salt• Store• HBase 选项

- 如果对表执行了错误的增删改操作，例如，删除了主键，保存操作后，易鲸捷 Designer 会报错，但主键并未显示在界面上，此时，请务必点击还原 ( Revert) 图标，主键才会恢复在界面上。其它操作同理。

方法 2: 通过 ER 图编辑表的属性

本示例将表 *vendor* 的字段 *vend_email* 的数据类型从 VARCHAR(64) 更改成 CHAR(64)。

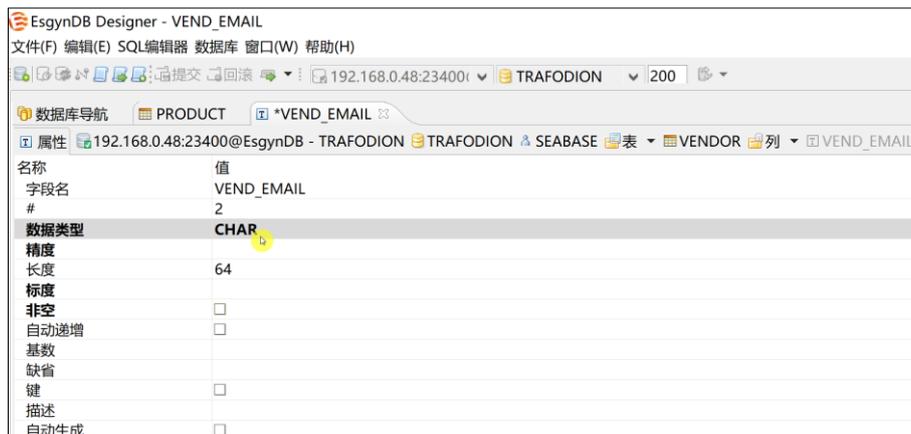
1. 在数据库导航窗口中，右键表 *vendor*，选择**编辑表**。
2. 点击 **ER 图**。
3. 双击字段 *vend_email*。



4. 弹出的对话框将显示字段 *vend_email* 的信息。在**数据类型**的下拉列表中，选择 **CHAR**。

**提示**

更多关于**数据类型映射规则**的信息，请参阅附录 1. 数据类型映射规则。



6. 表管理

5. 使用快捷键 **Ctrl + S** 保存信息。
6. 在弹出的对话框中，确认 SQL 语句无误，点击**保留**。

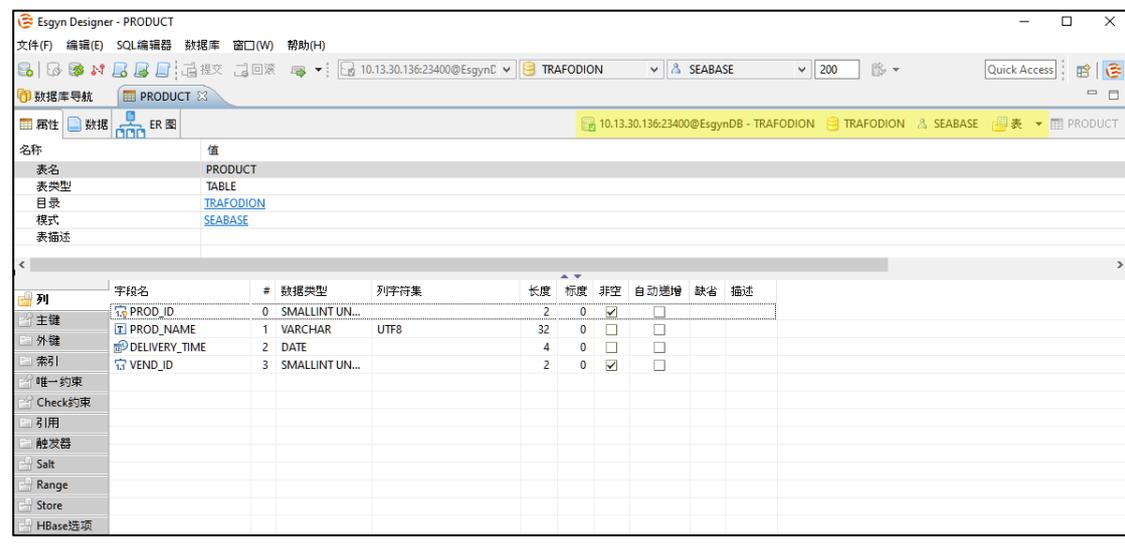


表 *vendor* 的字段 *vend_email* 的数据类型已从 `VARCHAR(64)` 更改成 `CHAR(64)`。



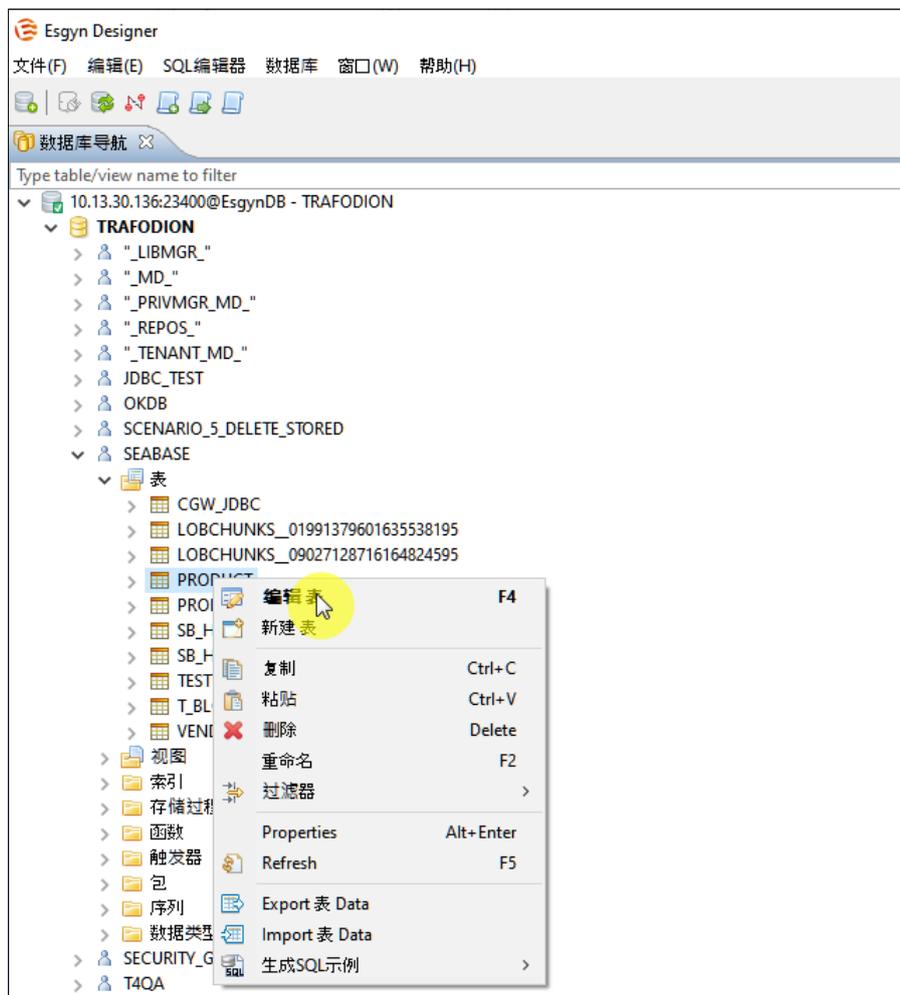
提示

如需查看或编辑数据库连接、查看 `catalog/schema/表/列`，点击下图中高亮的快捷方式。



6.3.2 如何编辑表的数据

1. 在数据库导航窗口中，右键表 *product*。
2. 点击编辑表。



3. 点击数据。

您能通过以下 2 种方法编辑数据：

- 方法 1：编辑数据工具栏
- 方法 2：右键结果集

• 方法 1: 编辑数据工具栏

编辑数据工具栏位于表数据的下方。

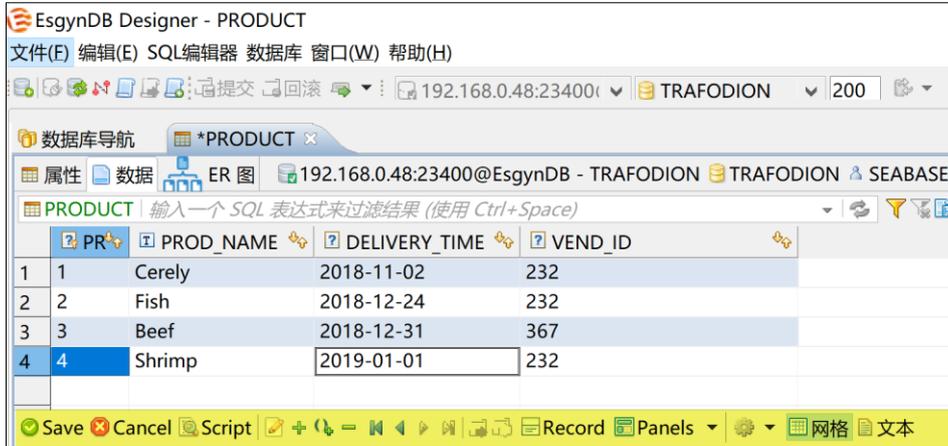
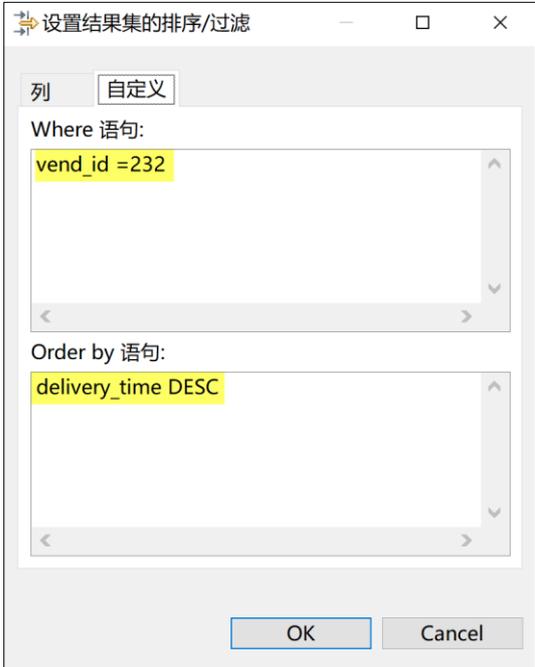
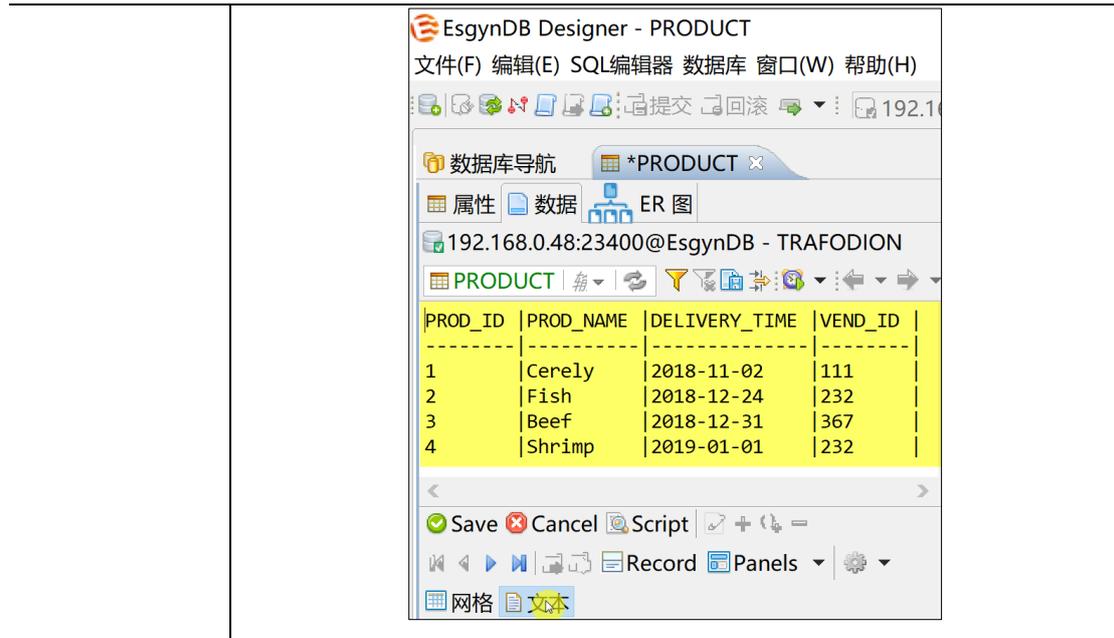


表 6-1 编辑数据工具栏

选项	说明
	如需手动输入值, 双击待修改的数据, 点击保存 () 图标。
	撤回操作。
	如需查看操作的 SQL 语句, 在保存操作前, 点击脚本 () 图标。
	添加行。
	复制当前行。
	删除行。

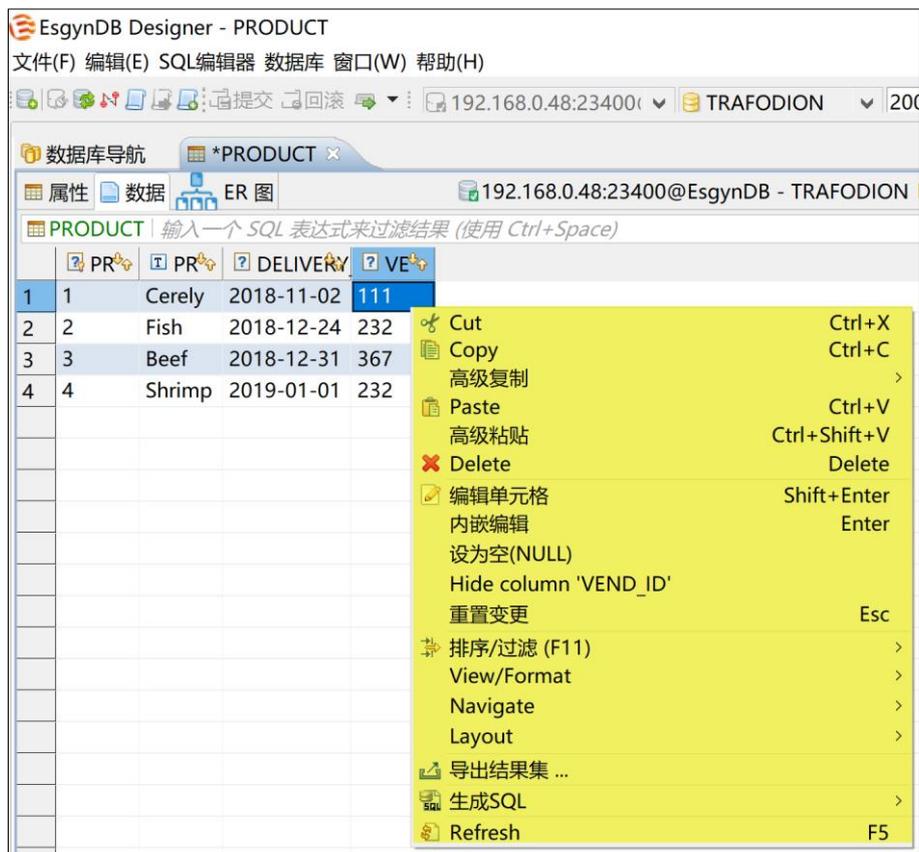
6. 表管理

 Record	查看单行数据。
 Panels	查看元数据、合计列和数值查看器。
	<p>如需设置结果集的排序/过滤，点击设置 () 图标 ></p> <p>Order/Filter，输入 WHERE 或 ORDER BY 条件。</p> <hr/> <p> 注意</p> <p>无需输入 WHERE 或 ORDER BY，直接输入条件语句。</p> <hr/> <div data-bbox="639 739 1174 1406"></div>
 文本	切换结果集视图。



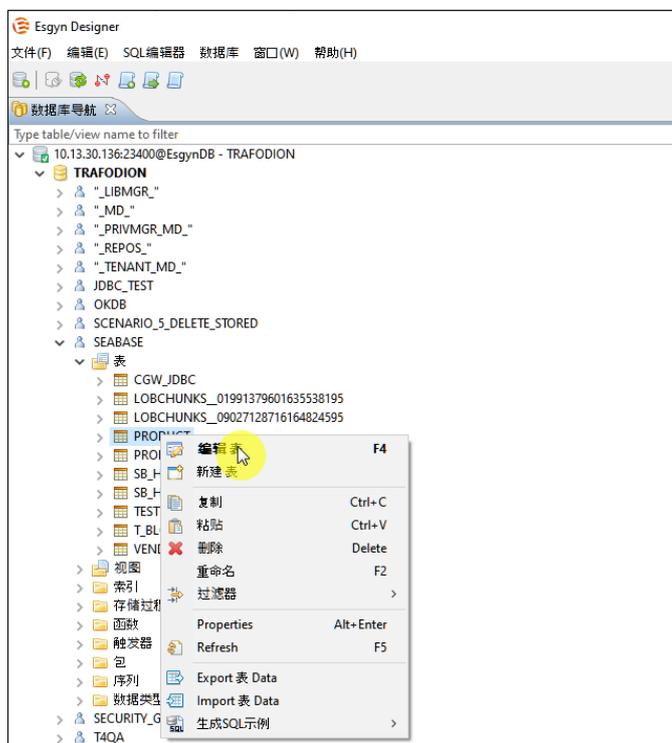
- 方法 2: 右键结果集

右键结果集，弹出的窗口将显示编辑选项，这些编辑选项也在 5.6.4.1 使用查询结果页面的编辑选项中。



6.3.3 如何编辑表的 ER 图

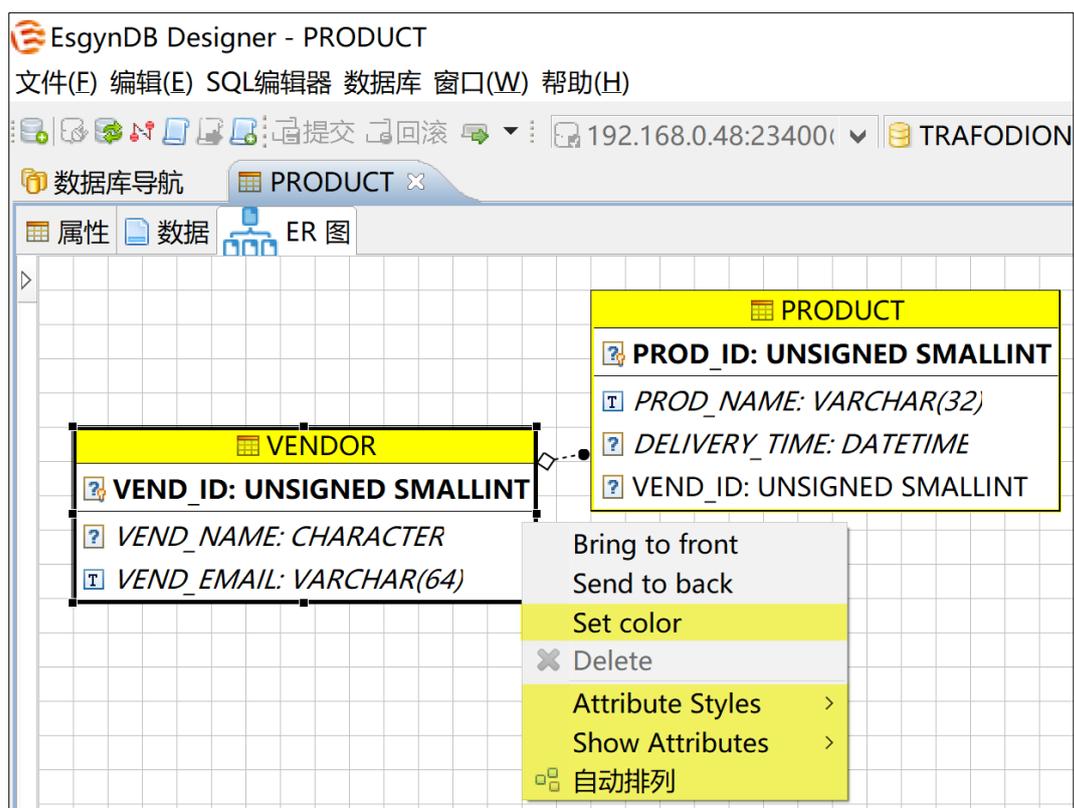
1. 在数据库导航窗口中，右键表 *product*，选择编辑表。



2. 点击 ER 图。

3. 右键待修改的 ER 图，在弹出的窗口中，您能编辑以下信息：

一级选项	二级选项
ER 图的颜色	/
属性风格	显示图标
	显示数据类型
	显示 NULL 属性
	显示备注
显示属性	所有
	显示键
	显示主键
	不显示



6.4 如何删除数据库对象

数据库包括以下对象⁹:

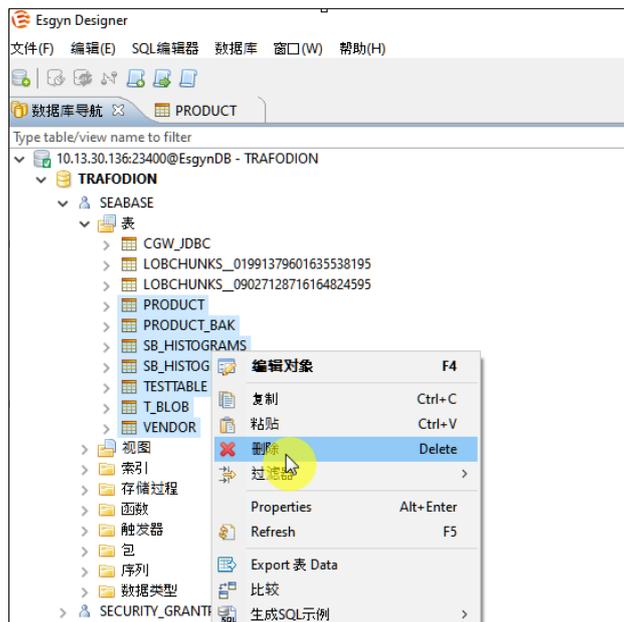
- 表
- 索引
- 序列
- 视图
- 约束（主键、外键、CHECK、UNIQUE 和 NOT NULL）

6.4.1 如何删除表

在数据库导航窗口中，右键待删除的表，点击删除。

提示

如需删除多张表，选中多张表的同时按下 Ctrl，再点击右键，点击删除。



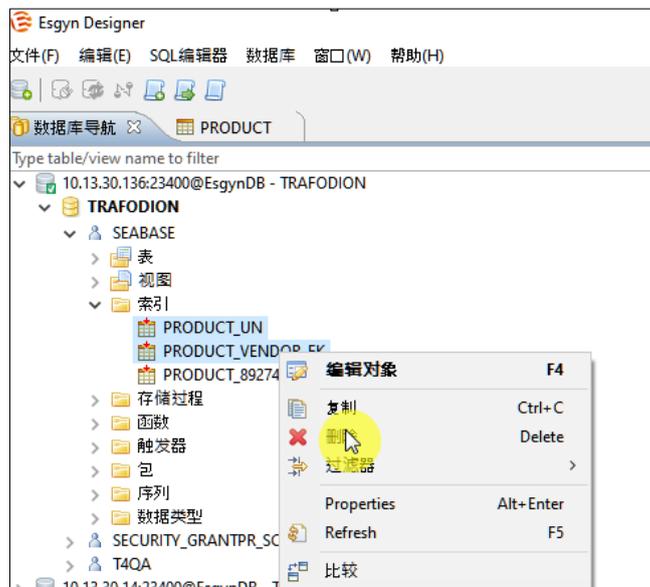
⁹ 更多关于数据库对象的信息，请参阅《EsgynDB SQL 参考手册》。

6.4.2 如何删除索引

在数据库导航窗口中，右键待删除的索引，点击删除。

提示

如需删除多个索引，选中多个索引的同时按下 Ctrl，再点击右键，点击删除。

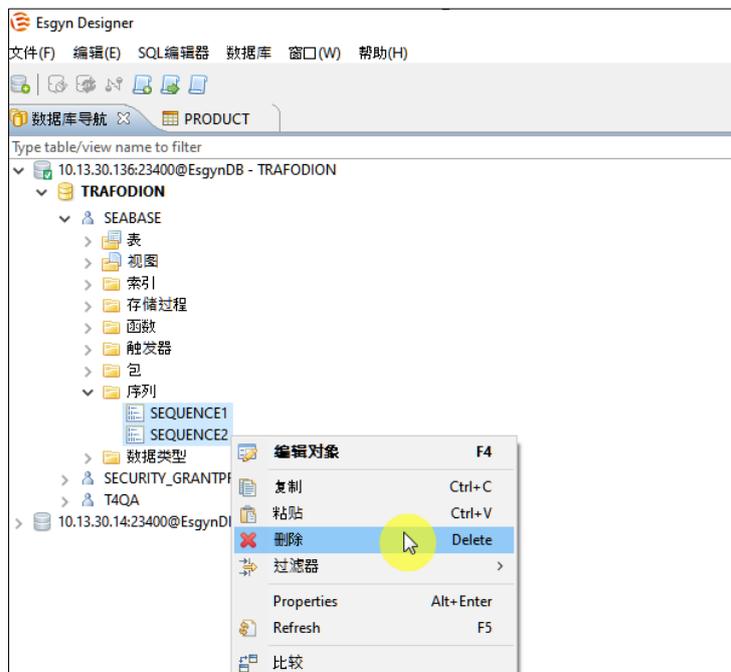


6.4.3 如何删除序列

在数据库导航窗口中，右键待删除的序列，点击删除。

💡 提示

如需删除多个序列，选中多个序列的同时按下 Ctrl，再点击右键，点击删除。



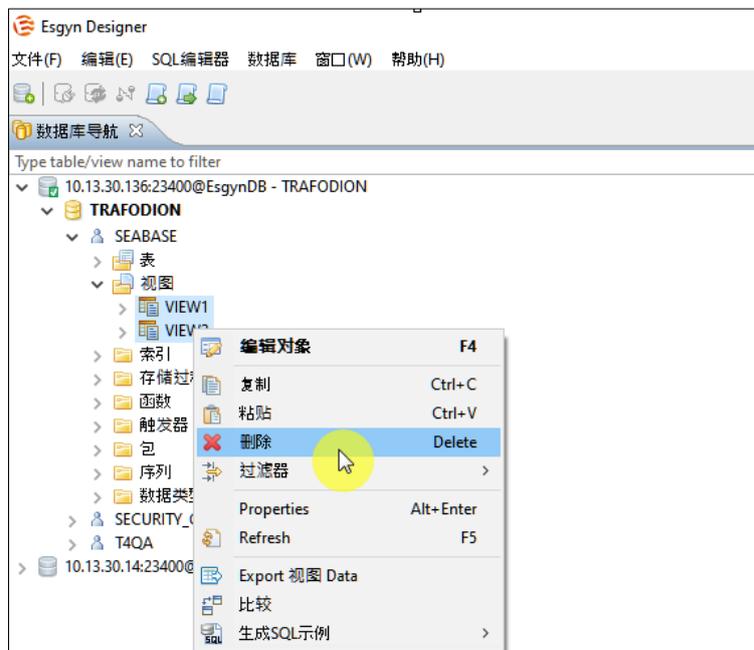
6.4.4 如何删除视图

在数据库导航窗口中，右键待删除的视图，点击删除。



提示

如需删除多个视图，选中多个视图的同时按下 Ctrl，再点击右键，点击删除。



6.4.5 如何删除约束

约束包括¹⁰:

- 主键
- 外键
- CHECK
- UNIQUE (唯一约束)
- NOT NULL



注意

在 EsgynDB 中，表存在后，不能修改或删除主键，所以，如需修改主键，只能删除表再重新建表¹¹。一张表只能有一个主键¹²。

¹⁰ 更多关于约束的信息，请参阅《EsgynDB SQL 参考手册》的约束章节。

¹¹ 更多关于如何创建和修改表的信息，请参阅《EsgynDB SQL 参考手册》的 **CREATE TABLE Statement** 章节。

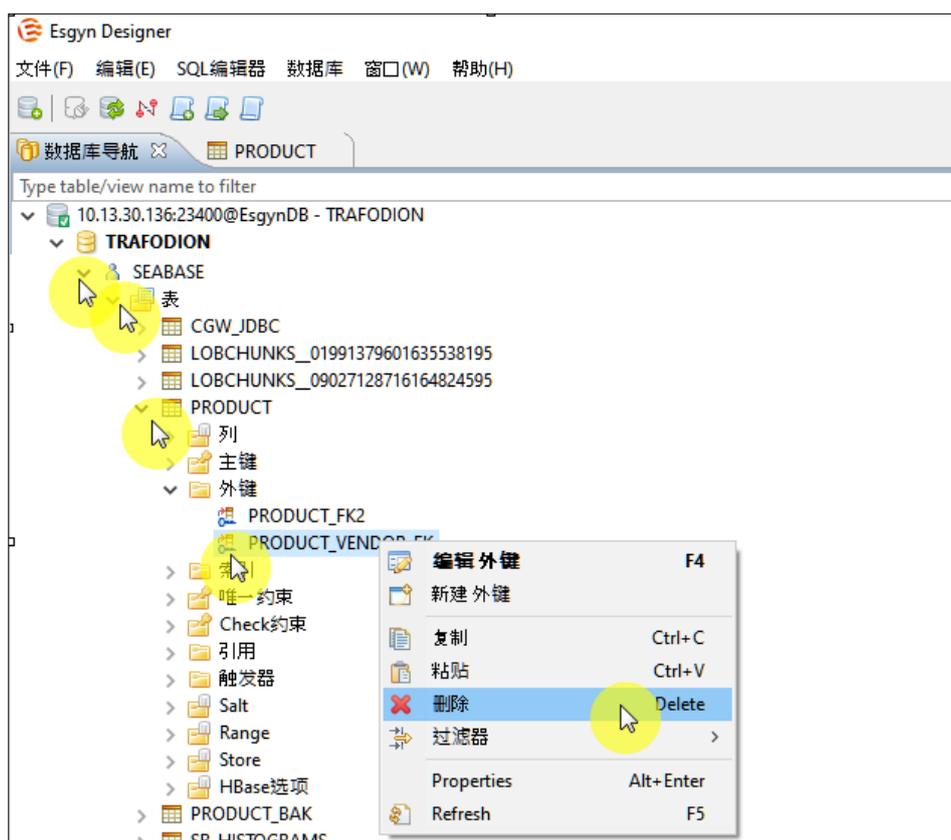
¹² 更多关于主键的信息，请参阅《EsgynDB SQL 参考手册》的主键章节。

6.4.5.1 如何删除外键

有以下 2 种方法删除外键：

- 方法 1

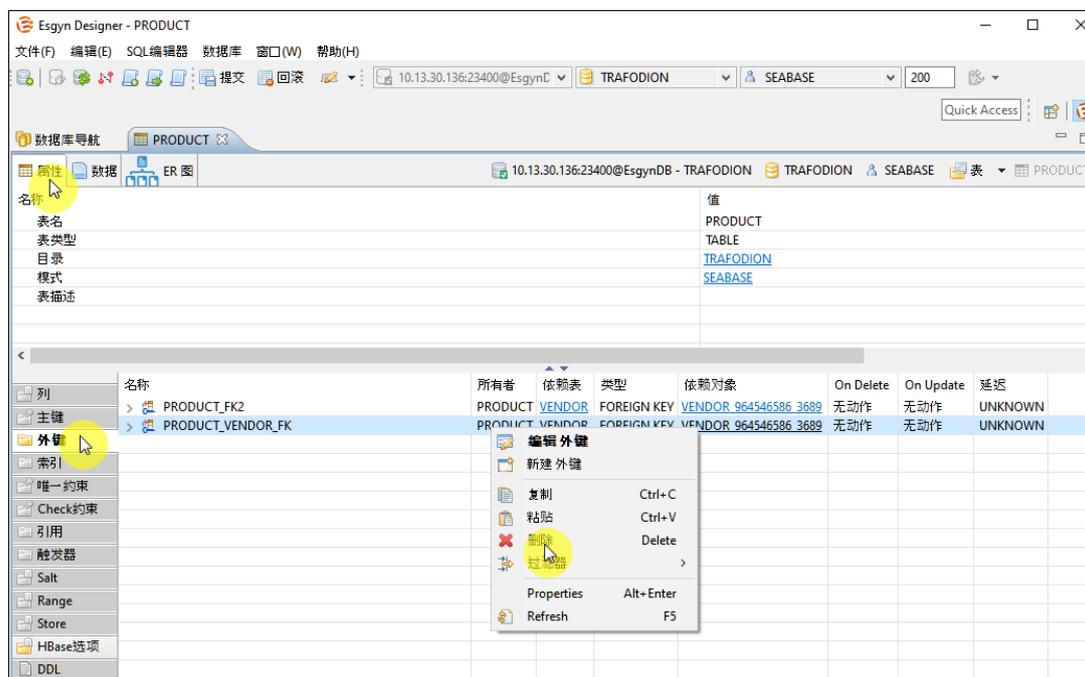
展开待选择的 schema > 展开表 > 展开待选择的表 > 展开外键 > 右键待选择的外键 > 点击删除。



6. 表管理

• 方法 2

1. 在表的属性界面中，点击外键，右键待删除的外键，点击删除。



2. 点击界面右下方的保存 () 图标。

3. 在弹出的对话框中，确认 SQL 语法无误，点击保留。

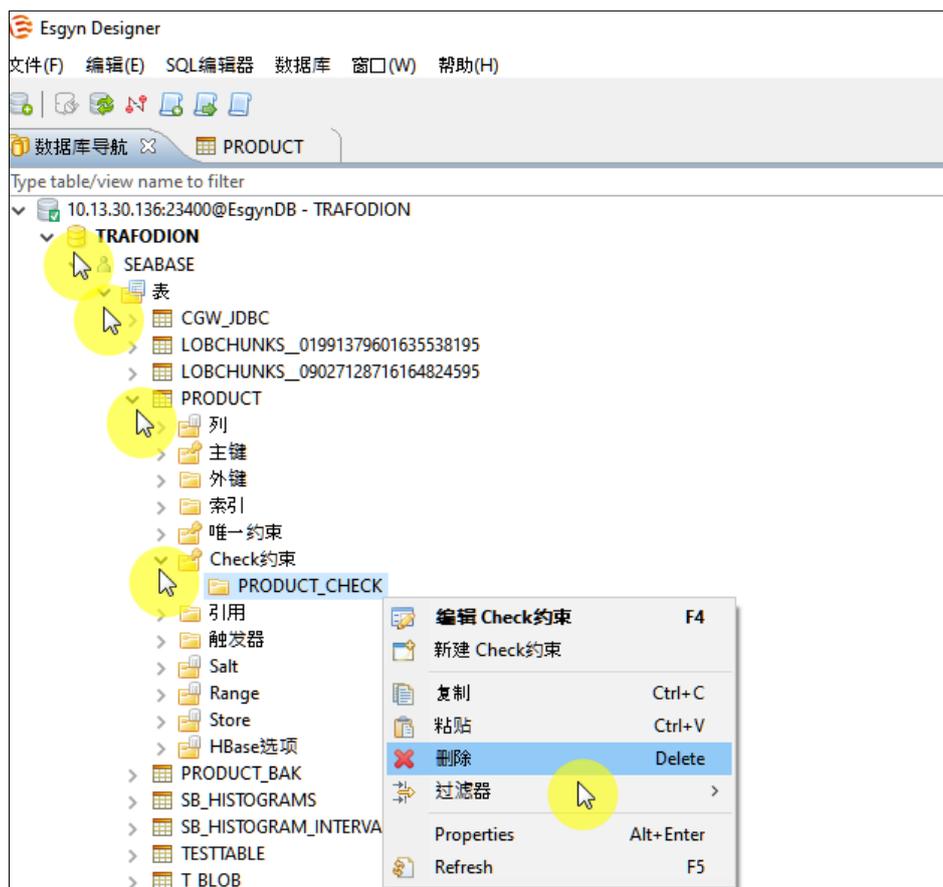


6.4.5.2 如何删除 CHECK 约束

有以下 2 种方法删除 CHECK 约束：

- 方法 1

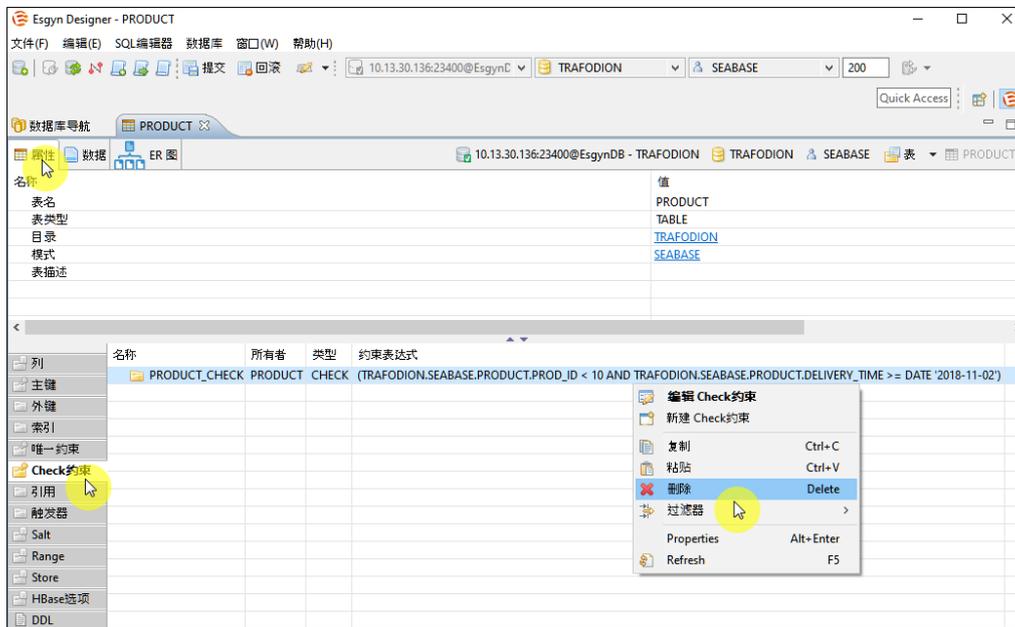
展开待选择的 schema > 展开表 > 展开待选择的表 > 展开 Check 约束 > 右键待选择的 Check 约束 > 点击删除。



6. 表管理

• 方法 2

在表的属性界面中，点击 Check 约束，右键待删除的 CHECK 约束，点击删除。

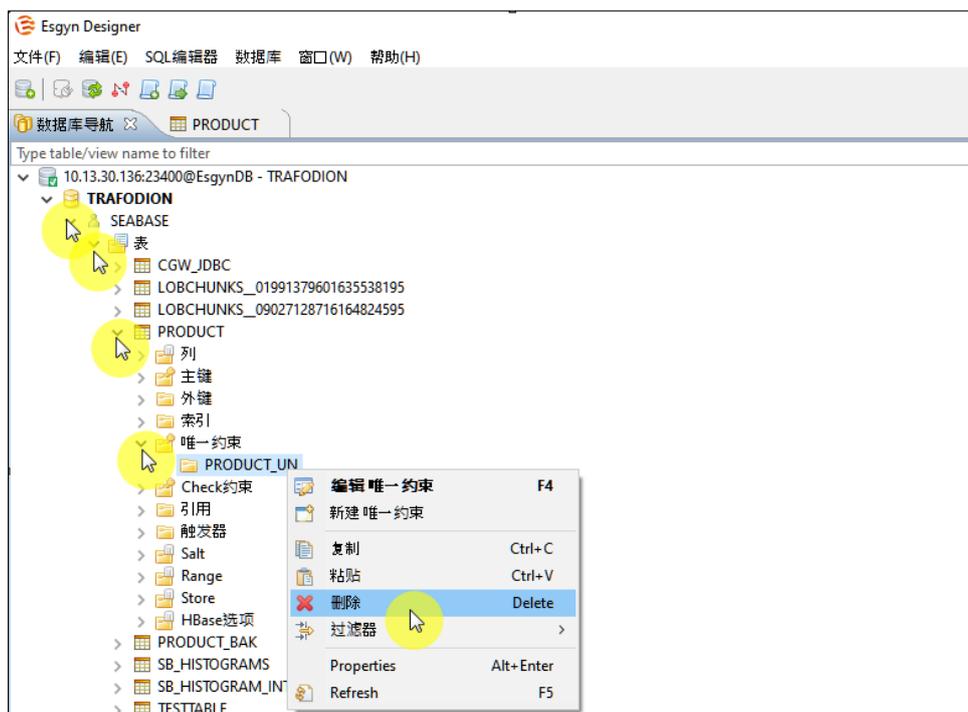


6.4.5.3 如何删除 UNIQUE 约束

有以下 2 种方法删除 UNIQUE 约束（唯一约束）：

- 方法 1

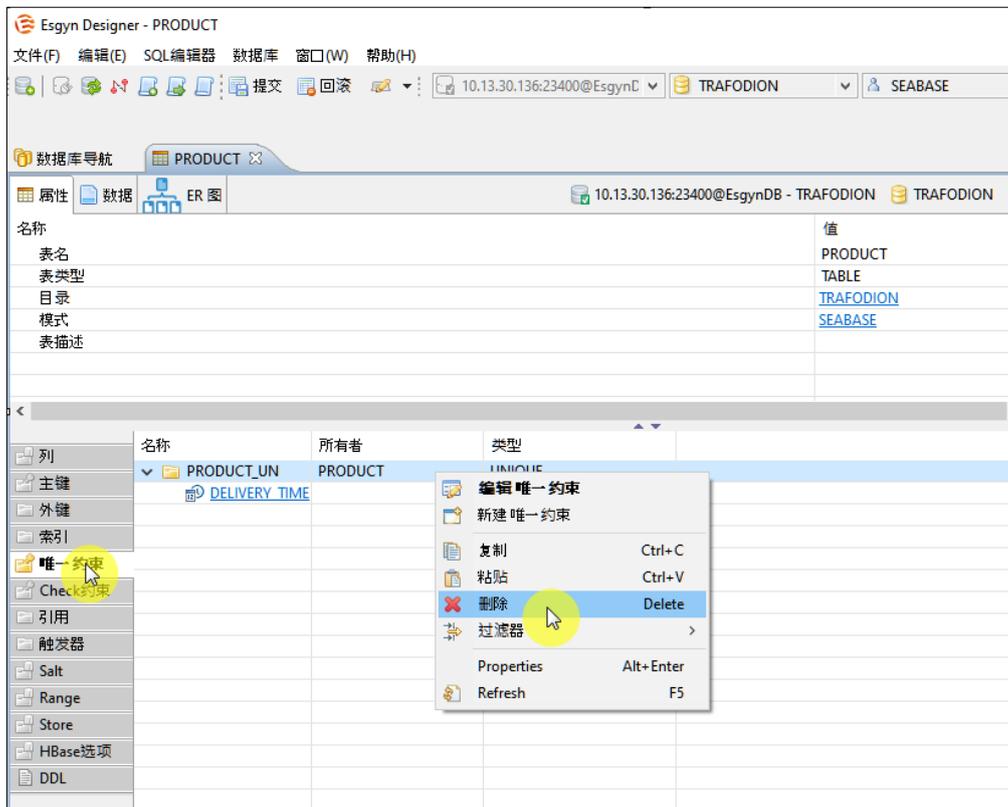
展开待选择的 schema > 展开表 > 展开待选择的表 > 展开唯一约束 > 右键待选择的唯一约束 > 点击删除。



6. 表管理

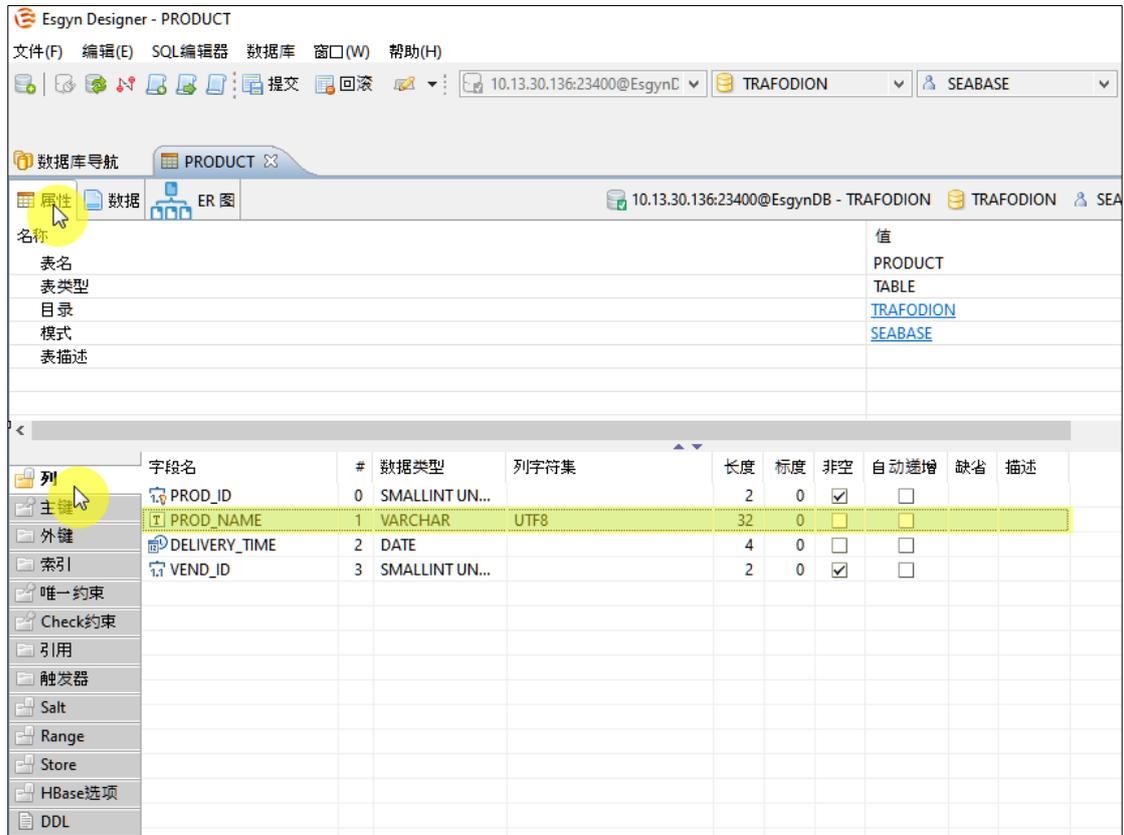
• 方法 2

在表的属性界面中，点击**唯一约束**，右键待删除的唯一约束，点击**删除**。



6.4.5.4 如何删除 NOT NULL 约束

1. 在表的属性界面中，点击列，取消勾选所需列的非空选项。



注意

一次仅能删除一列的 NOT NULL 约束。

2. 点击界面右下方的保存 () 图标。

3. 在弹出的对话框中，确认 SQL 语句无误，点击保留。



6.5 如何导出表的数据

表的数据能导出至：

- 文件。

更多信息，请参阅 [6.5.1 如何导出表的数据至文件](#)。

- 数据库表。

更多信息，请参阅 [6.5.2 如何导出表的数据至数据库表](#)。

6.5.1 如何导出表的数据至文件



提示

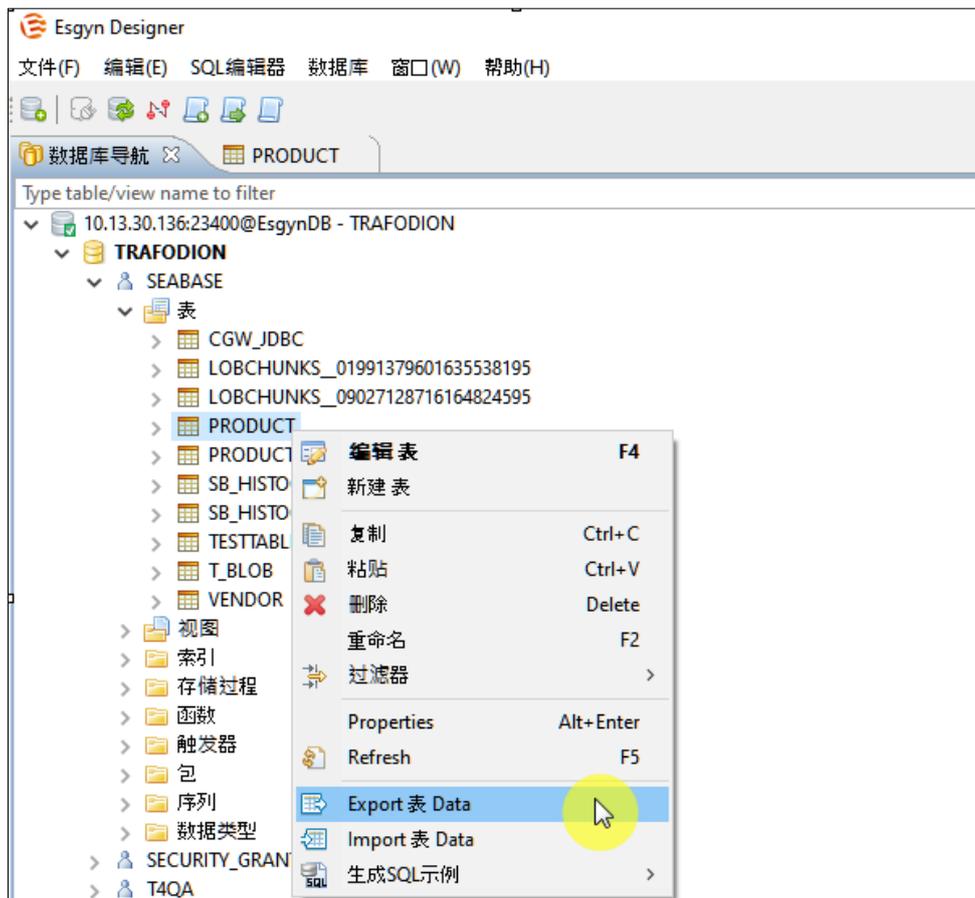
- 表的数据能导出至 **xml**、**json**、**html** 和 **csv** 文件。
- 表的 INSERT 语句能导出至 **.sql** 文件。



示例

本示例将表 *product* 的数据导出至 **xml** 文件。

1. 在数据库导航窗口中，右键表 *product*，点击导出表数据。
2. 后续步骤，请参阅 5.6.6.1 如何导出查询结果至文件的第 2 步至第 6 步。



**提示**

如需导出表的 INSERT 语句，在第 2 步点击 **SQL（导出到 SQL INSERT 语句）**，将导出.sql 文件。



以下是导出的 .sql 文件：

```
Insert Statement.sql
1  INSERT INTO TRAFODION.SEABASE.PRODUCT (PROD_ID,PROD_NAME,DELIVERY_TIME,VEND_ID) VALUES (
2  1,'Cerely','2018-11-02',232);
3  INSERT INTO TRAFODION.SEABASE.PRODUCT (PROD_ID,PROD_NAME,DELIVERY_TIME,VEND_ID) VALUES (
4  2,'Coriander','2018-12-24',639);
5  INSERT INTO TRAFODION.SEABASE.PRODUCT (PROD_ID,PROD_NAME,DELIVERY_TIME,VEND_ID) VALUES (
6  3,'Basil','2018-12-31',367);
7  INSERT INTO TRAFODION.SEABASE.PRODUCT (PROD_ID,PROD_NAME,DELIVERY_TIME,VEND_ID) VALUES (
8  4,'Durian','2019-01-01',825);
```

6.5.2 如何导出表的数据至数据库表

表的数据能导出至新建的表或已存在的表。



提示

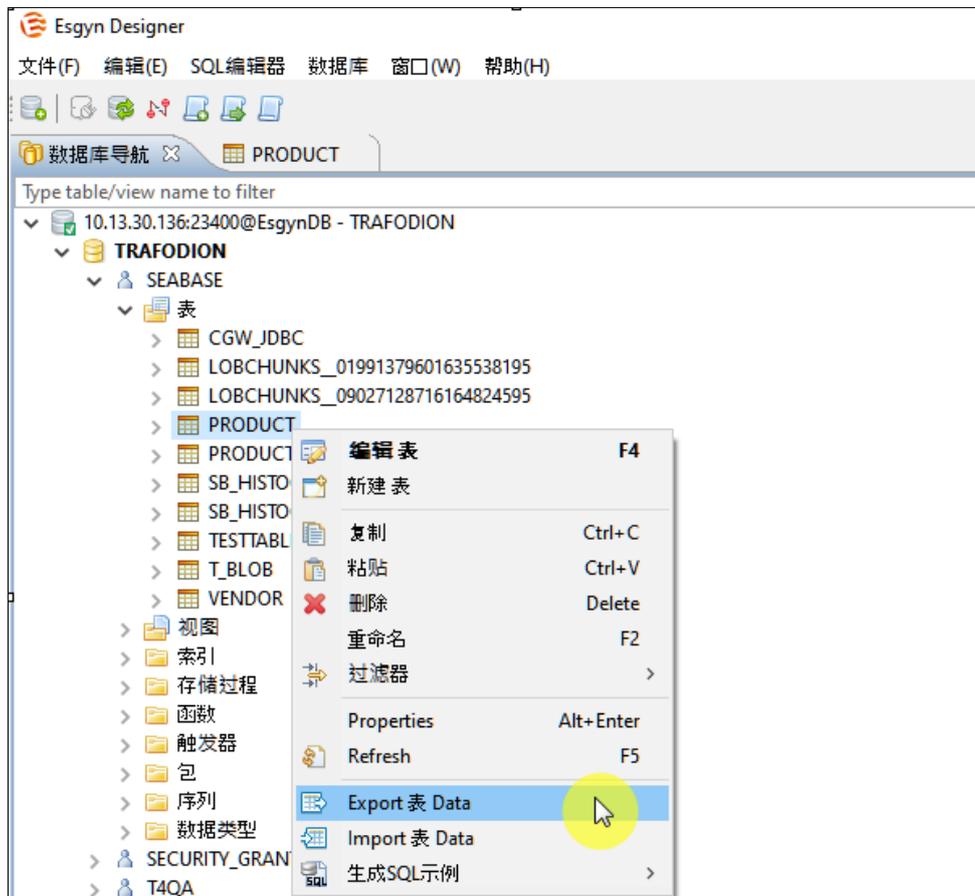
更多关于数据类型映射规则的信息，请参阅附录 1. 数据类型映射规则。



示例

本示例将表 *product* 的数据导出至新建的表 *computer*。

1. 在数据库导航窗口中，右键表 *product*，点击导出表数据。
2. 后续步骤，请参阅 5.6.6.2 如何导出查询结果至数据库表的第 2 步至第 6 步。



6.6 如何导入表的数据

源表的数据能导入至新建的表或已存在的表。

表 6-2 如需将源表数据导入至已存在的表，请参阅以下信息 1

序号	源表和目标表 满足以下条件	注意事项
①	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型相同	在当前版本中，无论将目标表列的数据类型指定为何种数据类型（即便是不能兼容的数据类型），数据都能导入成功，但实际上目标表列的数据类型并未更改，仍为创建目标表时指定的数据类型，即此时“指定目标列的数据类型”不生效 ¹³ 。
②	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容 <hr/>  提示 建议源表列的数据类型长度小于或等于目标表列的数据类型长度。	
③	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	导入失败。

示例请参阅表 6-3 将源表 t1 的数据导入至目标表 t2、t3 和 t4。

¹³ 该问题将在下一版本修复，预期效果为您无需指定目标列的数据类型，易鲸捷 Designer 将能正确地显示目标列的数据类型，且该值不能更改。

 示例

表 6-3 将源表 t1 的数据导入至目标表 t2、t3 和 t4

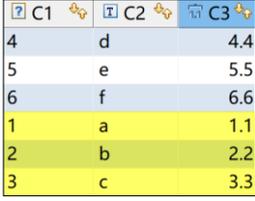
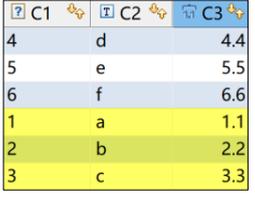
源表的结构和数据	源表与目标表同时满足以下条件	目标表的结构和数据	导入结果
CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);	① <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型相同	<pre>CREATE TABLE t2 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL);</pre> <pre>INSERT INTO t2 VALUES (4, 'd', 4.4), (5, 'e', 5.5), (6, 'f', 6.6);</pre>	导入成功，源表 t1 的数据导入至目标表 t2。 <pre>SELECT * FROM t2;</pre> 
CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);	② <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同,但能相互兼容。	<pre>CREATE TABLE t3 (c1 INT, c2 VARCHAR(64), c3 DOUBLE PRECISION);</pre> <pre>INSERT INTO t3 VALUES (4, 'd', 4.4), (5, 'e', 5.5), (6, 'f', 6.6);</pre>	导入成功，源表 t1 的数据导入至目标表 t3。 <pre>SELECT * FROM t3;</pre> 
CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);	③ <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同,且不能相互兼容。	<pre>CREATE TABLE t4 (c1 DATE, c2 TIME, c3 TIMESTAMP);</pre>	导入失败。

表 6-4 如需将源表的数据导入至已存在的表，请参阅以下信息 2

序号	源表和目标表 满足以下条件	注意事项
④	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量多于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型相同	在当前版本中： <ul style="list-style-type: none"> 对于列名相同的列而言，无论将目标列的数据类型指定为何种数据类型（即便是不能兼容的数据类型），数据都能导入成功，但实际上目标表列的数据类型并未更改，仍为创建目标表时指定的数据类型，即此时“指定目标列的数据类型”不生效。
⑤	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量多于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容 <hr/>  提示 建议源表列的数据类型长度小于或等于目标表列的数据类型长度。	<ul style="list-style-type: none"> 对于源表中“多余”的列而言，此时必须正确指定它的数据类型，即指定成相同或相互兼容（可能发生截断）的数据类型，即此时“指定目标列的数据类型”生效。
⑥	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量多于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	导入失败。

示例请参阅表 6-5 将源表 t1 的数据导入至目标表 t5、t6 和 t7。

 示例

表 6-5 将源表 t1 的数据导入至目标表 t5、t6 和 t7

源表的结构和数据	源表与目标表满足以下条件	目标表的结构和数据	导入结果																					
	④ <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量多于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型相同	<pre>CREATE TABLE t5 (c1 SMALLINT); INSERT INTO t5 VALUES (4), (5), (6);</pre>	导入成功，源表 t1 的数据导入至目标表 t5。 <pre>SELECT * FROM t5;</pre> <table border="1"> <thead> <tr> <th>C1</th> <th>C2</th> <th>C3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>a</td> <td>1.1</td> </tr> <tr> <td>2</td> <td>b</td> <td>2.2</td> </tr> <tr> <td>3</td> <td>c</td> <td>3.3</td> </tr> <tr> <td>4</td> <td>[NULL]</td> <td>[NULL]</td> </tr> <tr> <td>5</td> <td>[NULL]</td> <td>[NULL]</td> </tr> <tr> <td>6</td> <td>[NULL]</td> <td>[NULL]</td> </tr> </tbody> </table>	C1	C2	C3	1	a	1.1	2	b	2.2	3	c	3.3	4	[NULL]	[NULL]	5	[NULL]	[NULL]	6	[NULL]	[NULL]
C1	C2	C3																						
1	a	1.1																						
2	b	2.2																						
3	c	3.3																						
4	[NULL]	[NULL]																						
5	[NULL]	[NULL]																						
6	[NULL]	[NULL]																						
<pre>CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);</pre>	⑤ <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量多于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容。	<pre>CREATE TABLE t6 (c1 INT, c2 VARCHAR(64)); INSERT INTO t6 VALUES (4, 'd'), (5, 'e'), (6, 'f');</pre>	导入成功，源表 t1 的数据导入至目标表 t6。 <pre>SELECT * FROM t6;</pre> <table border="1"> <thead> <tr> <th>C1</th> <th>C2</th> <th>C3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>a</td> <td>1.1</td> </tr> <tr> <td>2</td> <td>b</td> <td>2.2</td> </tr> <tr> <td>3</td> <td>c</td> <td>3.3</td> </tr> <tr> <td>4</td> <td>d</td> <td>[NULL]</td> </tr> <tr> <td>5</td> <td>e</td> <td>[NULL]</td> </tr> <tr> <td>6</td> <td>f</td> <td>[NULL]</td> </tr> </tbody> </table>	C1	C2	C3	1	a	1.1	2	b	2.2	3	c	3.3	4	d	[NULL]	5	e	[NULL]	6	f	[NULL]
C1	C2	C3																						
1	a	1.1																						
2	b	2.2																						
3	c	3.3																						
4	d	[NULL]																						
5	e	[NULL]																						
6	f	[NULL]																						
	⑥ <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量多于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型不	<pre>CREATE TABLE t7 (c1 DATE, c2 TIME);</pre>	导入失败。																					

6. 表管理

	同，且不能相互兼容。		
--	------------	--	--

表 6-6 如需将源表数据导入至已存在的表，请参阅以下信息 3

序号	源表和目标表 满足以下条件	注意事项
⑦	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量少于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型相同	在当前版本中，对于列名相同的列而言，无论将目标列的数据类型指定为何种数据类型（即便是不能兼容的数据类型），数据都能导入成功，但实际上目标表列的数据类型并未更改，仍为创建目标表时指定的数据类型，即此时“指定目标列的数据类型”不生效。
⑧	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量少于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容 <hr/>  提示 建议源表列的数据类型长度小于或等于目标表列的数据类型长度。	
⑨	<input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量少于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	导入失败。

示例请参阅表 6-7 将源表 t1 的数据导入至目标表 t8、t9 和 t10。

 示例

表 6-7 将源表 t1 的数据导入至目标表 t8、t9 和 t10

源表的 结构和数据	源表与目标表 满足以下条件	目标表的结构和数据	导入结果																																			
<pre>CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);</pre>	<p>⑦</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量少于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型相同 	<pre>CREATE TABLE t8 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL, c4 INT); INSERT INTO t8 VALUES (4, 'd', 4.4, 70), (5, 'e', 5.5, 80), (6, 'f', 6.6, 90);</pre>	<p>导入成功，源表 t1 的数据导入至目标表 t8。</p> <pre>SELECT * FROM t8;</pre> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>C1</th> <th>C2</th> <th>C3</th> <th>C4</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>d</td> <td></td> <td>4.4</td> <td>70</td> </tr> <tr> <td>5</td> <td>e</td> <td></td> <td>5.5</td> <td>80</td> </tr> <tr> <td>6</td> <td>f</td> <td></td> <td>6.6</td> <td>90</td> </tr> <tr> <td>1</td> <td>a</td> <td></td> <td>1.1</td> <td>[NULL]</td> </tr> <tr> <td>2</td> <td>b</td> <td></td> <td>2.2</td> <td>[NULL]</td> </tr> <tr> <td>3</td> <td>c</td> <td></td> <td>3.3</td> <td>[NULL]</td> </tr> </tbody> </table>		C1	C2	C3	C4	4	d		4.4	70	5	e		5.5	80	6	f		6.6	90	1	a		1.1	[NULL]	2	b		2.2	[NULL]	3	c		3.3	[NULL]
		C1	C2	C3	C4																																	
	4	d		4.4	70																																	
5	e		5.5	80																																		
6	f		6.6	90																																		
1	a		1.1	[NULL]																																		
2	b		2.2	[NULL]																																		
3	c		3.3	[NULL]																																		
	<p>⑧</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量少于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容。 	<pre>CREATE TABLE t9 (c1 INT, c2 VARCHAR(64), c3 DOUBLE PRECISION, c4 INT); INSERT INTO t9 VALUES (40, 'de', 4.44, 70), (50, 'ef', 5.55, 80), (60, 'fg', 6.66, 90);</pre>	<p>导入成功，源表 t1 的数据导入至目标表 t9。</p> <pre>SELECT * FROM t9;</pre> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>C1</th> <th>C2</th> <th>C3</th> <th>C4</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>a</td> <td></td> <td>1.1</td> <td>[NULL]</td> </tr> <tr> <td>2</td> <td>b</td> <td></td> <td>2.2</td> <td>[NULL]</td> </tr> <tr> <td>3</td> <td>c</td> <td></td> <td>3.3</td> <td>[NULL]</td> </tr> <tr> <td>40</td> <td>de</td> <td></td> <td>4.44</td> <td>70</td> </tr> <tr> <td>50</td> <td>ef</td> <td></td> <td>5.55</td> <td>80</td> </tr> <tr> <td>60</td> <td>fg</td> <td></td> <td>6.66</td> <td>90</td> </tr> </tbody> </table>		C1	C2	C3	C4	1	a		1.1	[NULL]	2	b		2.2	[NULL]	3	c		3.3	[NULL]	40	de		4.44	70	50	ef		5.55	80	60	fg		6.66	90
	C1	C2	C3	C4																																		
1	a		1.1	[NULL]																																		
2	b		2.2	[NULL]																																		
3	c		3.3	[NULL]																																		
40	de		4.44	70																																		
50	ef		5.55	80																																		
60	fg		6.66	90																																		
	<p>⑨</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> 列名相同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量少于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型 	<pre>CREATE TABLE t10 (c1 DATE, c2 TIME, c3 TIMESTAMP, c4 DATE);</pre>	<p>导入失败。</p>																																			

6. 表管理

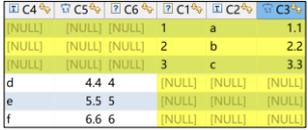
	不同，且不能相互兼容。		
--	-------------	--	--

表 6-8 如需将源表数据导入至已存在的表，请参阅以下信息 4

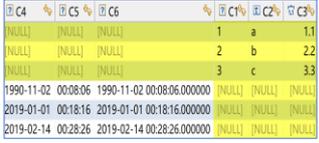
序号	源表和目标表 满足以下条件	注意事项
⑩	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型相同	必须正确指定目标表列的数据类型，即指定成相同或相互兼容(可能发生截断)的数据类型(即此时“指定目标列的数据类型”生效)，否则报错。
⑪	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容	
⑫	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	导入失败。

示例请参阅 [表 6-9](#) 将源表 t1 的数据导入至目标表 t11 和 t12 和 t13。

表 6-9 将源表 t1 的数据导入至目标表 t11 和 t12 和 t13

源表的结构和数据	源表与目标表满足以下条件	目标表的结构和数据	导入结果
	⑩ <input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型相同	<pre>CREATE TABLE t11 (c4 SMALLINT, c5 VARCHAR(32), c6 REAL); INSERT INTO t11 VALUES (4, 'd', 4.4), (5, 'e', 5.5), (6, 'f', 6.6);</pre>	导入成功, 源表 t1 的数据导入至目标表 t11。 <pre>SELECT * FROM t11;</pre> 
<pre>CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);</pre>	⑪ <input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型不同, 但能相互兼容	<pre>CREATE TABLE t12 (c4 INT, c5 VARCHAR(64), c6 DOUBLE PRECISION); INSERT INTO t12 VALUES (40, 'de', 4.44), (50, 'ef', 5.55), (60, 'fg', 6.66);</pre>	导入成功, 源表 t1 的数据导入至目标表 t12。 <pre>SELECT * FROM t12;</pre> 
	⑫ <input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量相同 <input checked="" type="checkbox"/> 列的数据类型	<pre>CREATE TABLE t13 (c4 DATE, c5 TIME, c6 TIMESTAMP); INSERT INTO t13 VALUES ('1990-11-02', '00:08:06', '1990-</pre>	导入成功, 源表 t1 的数据导入至目标表 t13。 <pre>SELECT * FROM t13;</pre>

6. 表管理

	<p>不同， 且不能 相互兼 容</p>	<p>11-02 00:08:06.000000'), ('2019-01- 01', '00:18:16', '2019- 01-01 00:18:16.000000'), ('2019-02- 14', '00:28:26', '2019- 02-14 00:28:26.000000');</p>	 <table border="1"> <thead> <tr> <th>C4</th> <th>C5</th> <th>C6</th> <th>C1%</th> <th>C2%</th> <th>C3%</th> </tr> </thead> <tbody> <tr> <td>NULL</td> <td>NULL</td> <td>NULL</td> <td>1</td> <td>a</td> <td>11</td> </tr> <tr> <td>NULL</td> <td>NULL</td> <td>NULL</td> <td>2</td> <td>b</td> <td>22</td> </tr> <tr> <td>NULL</td> <td>NULL</td> <td>NULL</td> <td>3</td> <td>c</td> <td>33</td> </tr> <tr> <td>1990-11-02</td> <td>00:08:06</td> <td>1990-11-02 00:08:06.000000</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>2019-01-01</td> <td>00:18:16</td> <td>2019-01-01 00:18:16.000000</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>2019-02-14</td> <td>00:28:26</td> <td>2019-02-14 00:28:26.000000</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table>	C4	C5	C6	C1%	C2%	C3%	NULL	NULL	NULL	1	a	11	NULL	NULL	NULL	2	b	22	NULL	NULL	NULL	3	c	33	1990-11-02	00:08:06	1990-11-02 00:08:06.000000	NULL	NULL	NULL	2019-01-01	00:18:16	2019-01-01 00:18:16.000000	NULL	NULL	NULL	2019-02-14	00:28:26	2019-02-14 00:28:26.000000	NULL	NULL	NULL
C4	C5	C6	C1%	C2%	C3%																																								
NULL	NULL	NULL	1	a	11																																								
NULL	NULL	NULL	2	b	22																																								
NULL	NULL	NULL	3	c	33																																								
1990-11-02	00:08:06	1990-11-02 00:08:06.000000	NULL	NULL	NULL																																								
2019-01-01	00:18:16	2019-01-01 00:18:16.000000	NULL	NULL	NULL																																								
2019-02-14	00:28:26	2019-02-14 00:28:26.000000	NULL	NULL	NULL																																								

6. 表管理

表 6-10 如需将源表数据导入至已存在的表，请参阅以下信息 4

序号	源表和目标表 满足以下条件	注意事项
⑬	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量多于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型相同	必须正确指定目标表列的数据类型，即指定成相同或相互兼容（可能发生截断）的数据类型（即此时“指定目标列的数据类型”生效），否则报错。
⑭	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量多于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容	
⑮	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量多于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	

示例请参阅表 6-11 将源表 t1 的数据导入至目标表 t14 和 t15 和 t16。

表 6-11 将源表 t1 的数据导入至目标表 t14 和 t15 和 t16

源表的结构和数据	源表与目标表满足以下条件	目标表的结构和数据	导入结果																																			
CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL); INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);	⑬ <input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量多于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型相同	<pre>CREATE TABLE t14 (c4 SMALLINT);</pre> <pre>INSERT INTO t14 VALUES (4), (5), (6);</pre>	导入成功，源表 t1 的数据导入至目标表 t14。 <pre>SELECT * FROM t14;</pre> <table border="1"> <thead> <tr> <th>C4</th> <th>C1</th> <th>C2</th> <th>C3</th> </tr> </thead> <tbody> <tr><td>[NULL]</td><td>1</td><td>a</td><td>1.1</td></tr> <tr><td>[NULL]</td><td>2</td><td>b</td><td>2.2</td></tr> <tr><td>[NULL]</td><td>3</td><td>c</td><td>3.3</td></tr> <tr><td>4</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> <tr><td>5</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> <tr><td>6</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> </tbody> </table>	C4	C1	C2	C3	[NULL]	1	a	1.1	[NULL]	2	b	2.2	[NULL]	3	c	3.3	4	[NULL]	[NULL]	[NULL]	5	[NULL]	[NULL]	[NULL]	6	[NULL]	[NULL]	[NULL]							
C4	C1	C2	C3																																			
[NULL]	1	a	1.1																																			
[NULL]	2	b	2.2																																			
[NULL]	3	c	3.3																																			
4	[NULL]	[NULL]	[NULL]																																			
5	[NULL]	[NULL]	[NULL]																																			
6	[NULL]	[NULL]	[NULL]																																			
INSERT INTO t1 VALUES (40, 'de'), (50, 'ef'), (60, 'fg');	⑭ <input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量多于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容	<pre>CREATE TABLE t15 (c4 INT, c5 VARCHAR(64));</pre> <pre>INSERT INTO t15 VALUES (40, 'de'), (50, 'ef'), (60, 'fg');</pre>	导入成功，源表 t1 的数据导入至目标表 t15。 <pre>SELECT * FROM t15;</pre> <table border="1"> <thead> <tr> <th>C4</th> <th>C5</th> <th>C1</th> <th>C2</th> <th>C3</th> </tr> </thead> <tbody> <tr><td>[NULL]</td><td>[NULL]</td><td>1</td><td>a</td><td>1.1</td></tr> <tr><td>[NULL]</td><td>[NULL]</td><td>2</td><td>b</td><td>2.2</td></tr> <tr><td>[NULL]</td><td>[NULL]</td><td>3</td><td>c</td><td>3.3</td></tr> <tr><td>40</td><td>de</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> <tr><td>50</td><td>ef</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> <tr><td>60</td><td>fg</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> </tbody> </table>	C4	C5	C1	C2	C3	[NULL]	[NULL]	1	a	1.1	[NULL]	[NULL]	2	b	2.2	[NULL]	[NULL]	3	c	3.3	40	de	[NULL]	[NULL]	[NULL]	50	ef	[NULL]	[NULL]	[NULL]	60	fg	[NULL]	[NULL]	[NULL]
C4	C5	C1	C2	C3																																		
[NULL]	[NULL]	1	a	1.1																																		
[NULL]	[NULL]	2	b	2.2																																		
[NULL]	[NULL]	3	c	3.3																																		
40	de	[NULL]	[NULL]	[NULL]																																		
50	ef	[NULL]	[NULL]	[NULL]																																		
60	fg	[NULL]	[NULL]	[NULL]																																		
INSERT INTO t1 VALUES (1990-11-02, '00:08:06'), (2019-01-01, '00:18:16');	⑮ <input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同 (源表列的数量多于目标表列的数量) <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	<pre>CREATE TABLE t16 (c4 DATE, c5 TIME);</pre> <pre>INSERT INTO t16 VALUES ('1990-11-02', '00:08:06'), ('2019-01-01', '00:18:16');</pre>	导入成功，源表 t1 的数据导入至目标表 t16。 <pre>SELECT * FROM t16;</pre> <table border="1"> <thead> <tr> <th>C4</th> <th>C5</th> <th>C1</th> <th>C2</th> <th>C3</th> </tr> </thead> <tbody> <tr><td>[NULL]</td><td>[NULL]</td><td>1</td><td>a</td><td>1.1</td></tr> <tr><td>[NULL]</td><td>[NULL]</td><td>2</td><td>b</td><td>2.2</td></tr> <tr><td>[NULL]</td><td>[NULL]</td><td>3</td><td>c</td><td>3.3</td></tr> <tr><td>1990-11-02</td><td>00:08:06</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> <tr><td>2019-01-01</td><td>00:18:16</td><td>[NULL]</td><td>[NULL]</td><td>[NULL]</td></tr> </tbody> </table>	C4	C5	C1	C2	C3	[NULL]	[NULL]	1	a	1.1	[NULL]	[NULL]	2	b	2.2	[NULL]	[NULL]	3	c	3.3	1990-11-02	00:08:06	[NULL]	[NULL]	[NULL]	2019-01-01	00:18:16	[NULL]	[NULL]	[NULL]					
C4	C5	C1	C2	C3																																		
[NULL]	[NULL]	1	a	1.1																																		
[NULL]	[NULL]	2	b	2.2																																		
[NULL]	[NULL]	3	c	3.3																																		
1990-11-02	00:08:06	[NULL]	[NULL]	[NULL]																																		
2019-01-01	00:18:16	[NULL]	[NULL]	[NULL]																																		

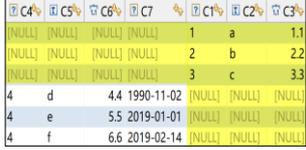
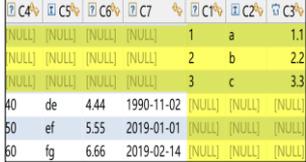
6. 表管理

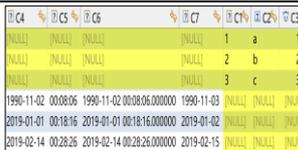
表 6-12 如需将源表数据导入至已存在的表，请参阅以下信息 4

序号	源表和目标表 满足以下条件	注意事项
⑩⑥	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量少于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型相同	必须正确指定目标表列的数据类型，即指定成相同或相互兼容（可能发生截断）的数据类型（即此时“指定目标列的数据类型”生效），否则报错。
⑩⑦	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量少于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，但能相互兼容	
⑩⑧	<input checked="" type="checkbox"/> 列名不同 <input checked="" type="checkbox"/> 列的数量不同（源表列的数量少于目标表列的数量） <input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容	

示例请参阅表 6-13 将源表 t1 的数据导入至目标表 t17 和 t18 和 t19。

表 6-13 将源表 t1 的数据导入至目标表 t17 和 t18 和 t19

源表的结构和数据	源表与目标表满足以下条件	目标表的结构和数据	导入结果
<p>CREATE TABLE t1 (c1 SMALLINT, c2 VARCHAR(32), c3 REAL);</p> <p>INSERT INTO t1 VALUES (1, 'a', 1.1), (2, 'b', 2.2), (3, 'c', 3.3);</p>	<p>①6</p> <p><input checked="" type="checkbox"/> 列名不同</p> <p><input checked="" type="checkbox"/> 列的数量不同 (源表列的数量少于目标表列的数量)</p> <p><input checked="" type="checkbox"/> 列的数据类型相同</p>	<pre>CREATE TABLE t17 (c4 SMALLINT, c5 VARCHAR(32), c6 REAL, c7 DATE); INSERT INTO t17 VALUES (4, 'd', 4.4, '1990-11-02'), (4, 'e', 5.5, '2019-01-01'), (4, 'f', 6.6, '2019-02-14');</pre>	<p>导入成功, 源表 t1 的数据导入至目标表 t17。</p> <p>SELECT * FROM t17;</p> 
<p>INSERT INTO t1 VALUES (40, 'de', 4.44, '1990-11-02'), (50, 'ef', 5.55, '2019-01-01'), (60, 'fg', 6.66, '2019-02-14');</p>	<p>①7</p> <p><input checked="" type="checkbox"/> 列名不同</p> <p><input checked="" type="checkbox"/> 列的数量不同 (源表列的数量少于目标表列的数量)</p> <p><input checked="" type="checkbox"/> 列的数据类型不同, 但能相互兼容</p>	<pre>CREATE TABLE t18 (c4 INT, c5 VARCHAR(64), c6 DOUBLE PRECISION, c7 DATE); INSERT INTO t18 VALUES (40, 'de', 4.44, '1990-11-02'), (50, 'ef', 5.55, '2019-01-01'), (60, 'fg', 6.66, '2019-02-14');</pre>	<p>导入成功, 源表 t1 的数据导入至目标表 t18。</p> <p>SELECT * FROM t18;</p> 
	<p>①8</p> <p><input checked="" type="checkbox"/> 列名不同</p> <p><input checked="" type="checkbox"/> 列的数量不同 (源表列的数量</p>	<pre>CREATE TABLE t19 (c4 DATE, c5 TIME, c6 TIMESTAMP, c7 DATE);</pre>	<p>导入成功, 源表 t1 的数据导入至目标表 t19。</p> <p>SELECT * FROM t19;</p>

<p>少于目标表列的数量)</p> <p><input checked="" type="checkbox"/> 列的数据类型不同，且不能相互兼容</p>	<pre>INSERT INTO t19 VALUES ('1990-11-02','00:08:06','1990-11-02 00:08:06.000000', '1990-11-03'), ('2019-01-01','00:18:16','2019-01-01 00:18:16.000000', '2019-01-02'), ('2019-02-14','00:28:26','2019-02-14 00:28:26.000000', '2019-02-15');</pre>	
---	---	---



示例

本示例将源表 *spice* 的数据导入至已存在的目标表 *vegetable*，它们的 DDL 部分相同。

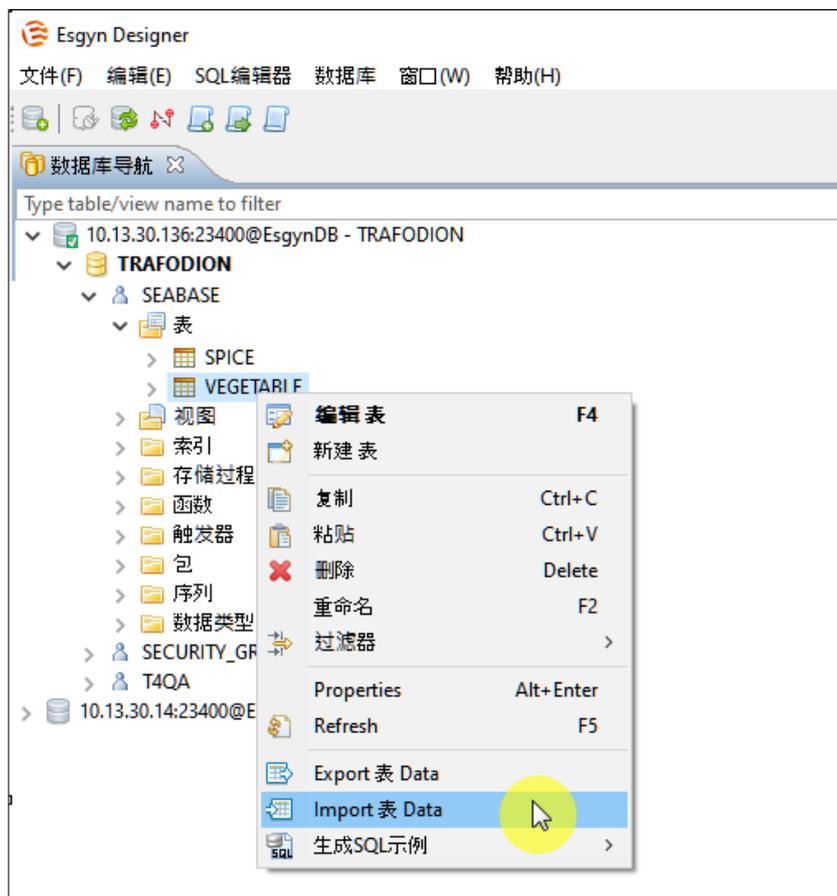
表 *spice* 的 DDL:

```
CREATE TABLE TRAFODION.SEABASE.SPICE
(
  PROD_ID SMALLINT UNSIGNED DEFAULT
  NULL NOT
  SERIALIZED
  , PROD_NAME VARCHAR(64) CHARACTER SET
  ISO88591 COLLATE
  DEFAULT DEFAULT NULL NOT SERIALIZED
  , BUY_TIMES SMALLINT UNSIGNED DEFAULT
  NULL NOT
  SERIALIZED
  , SALE_TIME DATE DEFAULT NULL NOT
  SERIALIZED
)
```

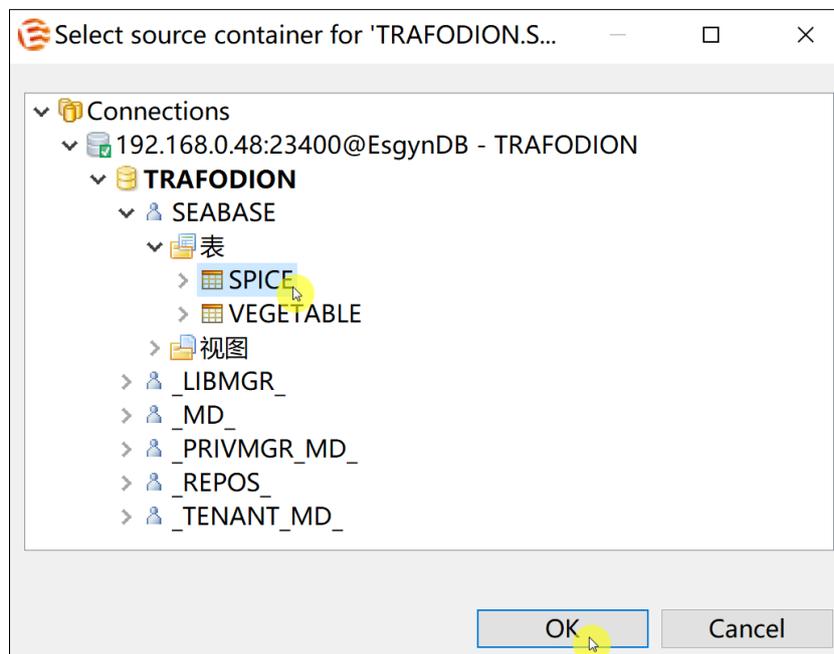
表 *vegetable* 的 DDL:

```
CREATE TABLE TRAFODION.SEABASE.VEGETABLE
(
  PROD_ID                SMALLINT UNSIGNED DEFAULT
NULL NOT
  SERIALIZED
,  PROD_NAME              VARCHAR(64) CHARACTER
SET ISO88591 COLLATE
  DEFAULT DEFAULT NULL NOT SERIALIZED
,  DELIVERY_TIME          DATE DEFAULT NULL NOT
SERIALIZED
,  VEND_ID                SMALLINT UNSIGNED DEFAULT
NULL NOT
  SERIALIZED
)
```

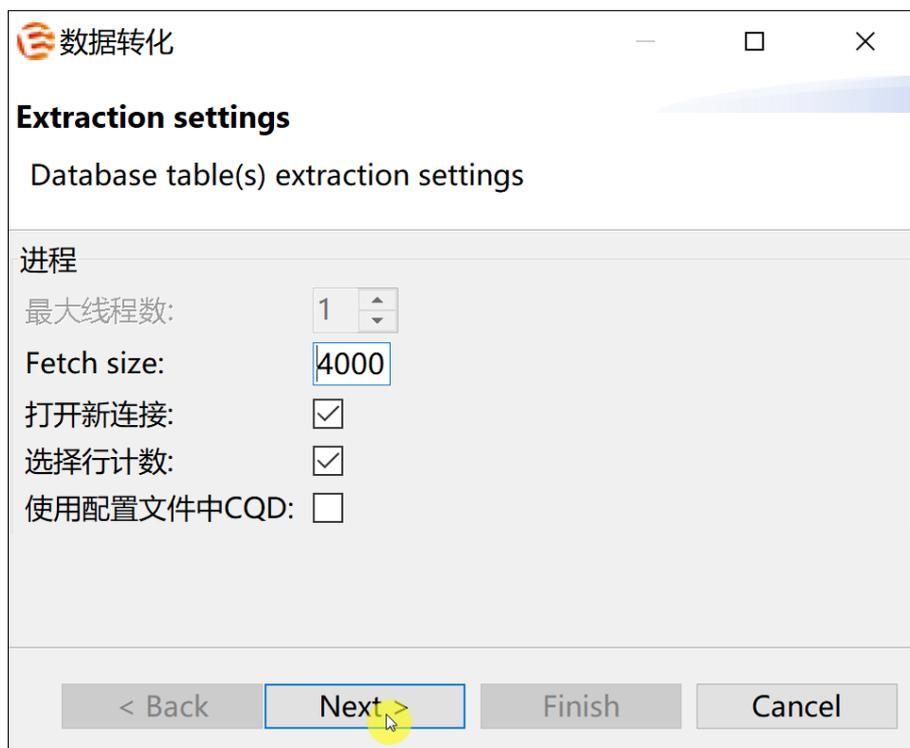
1. 在数据库导航窗口中，右键表 *vegetable*，点击导入表数据。



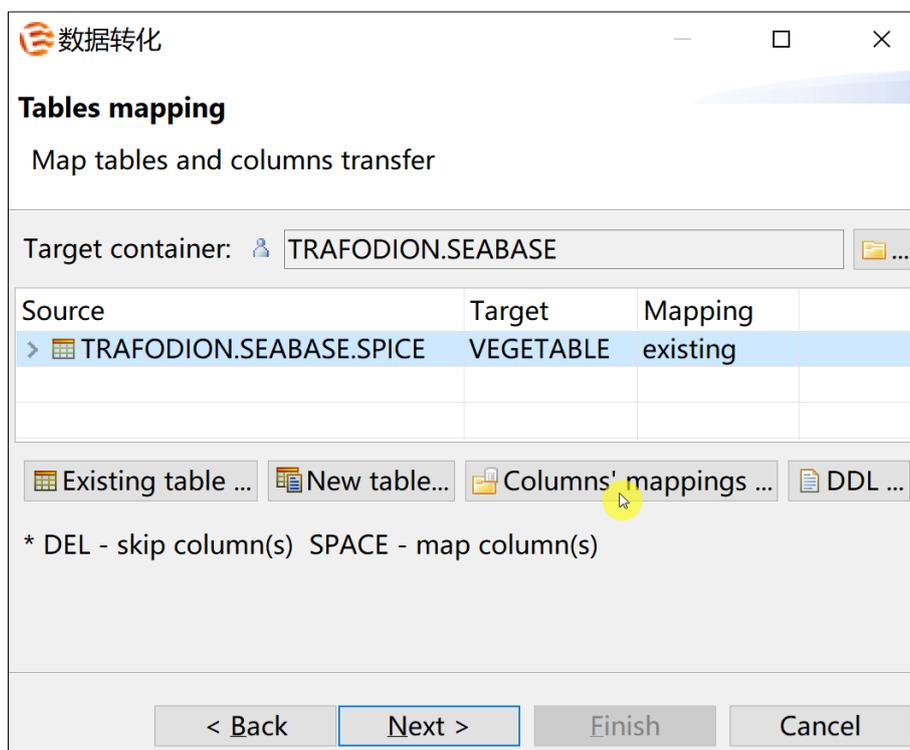
2. 选择源表 *spice*，点击 **OK**。



- 接受默认设置，点击下一步。

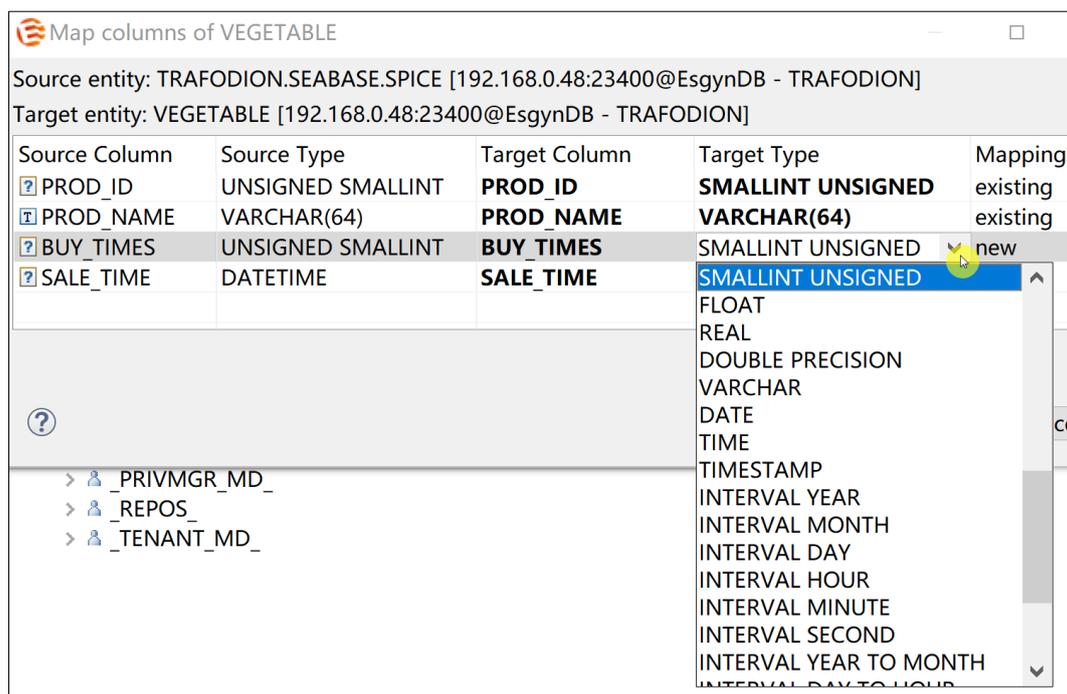


- 确认源表和目标表，点击列映射。

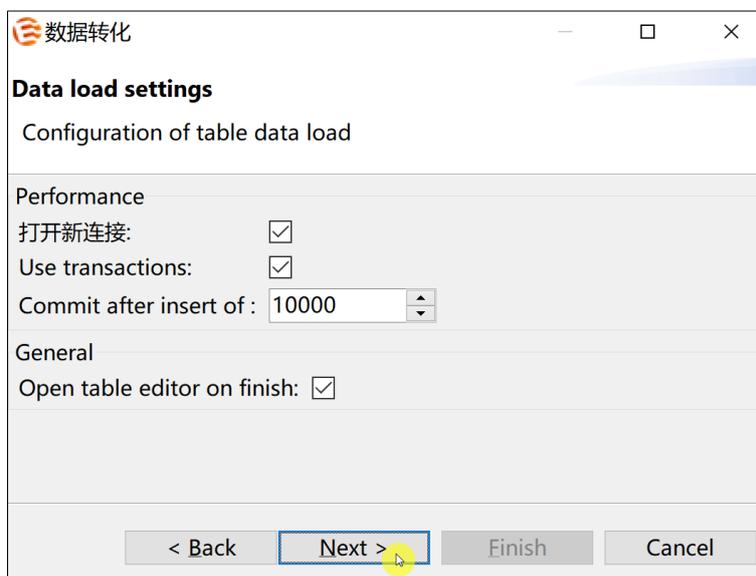


6. 表管理

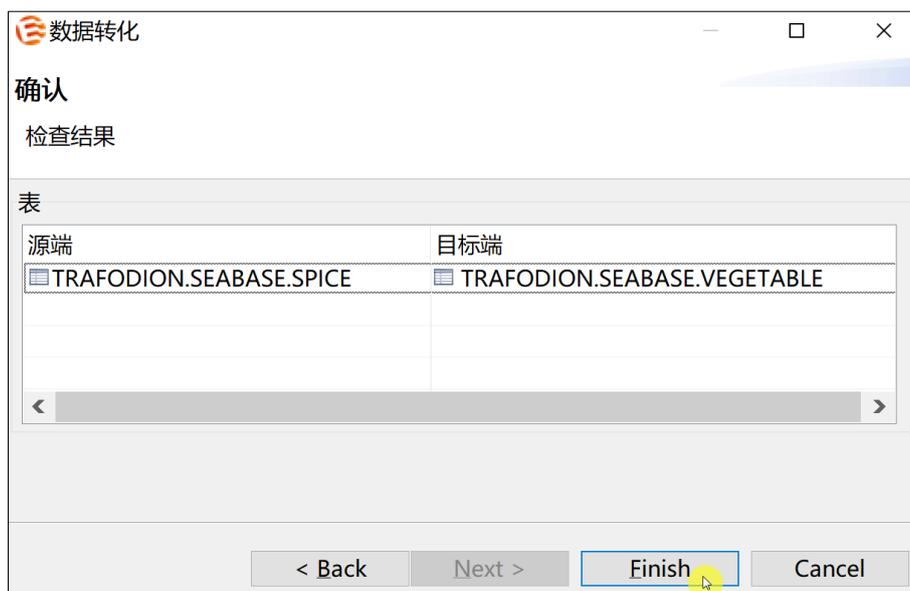
5. 在下拉列表中选择目标表列的数据类型，点击下一步。



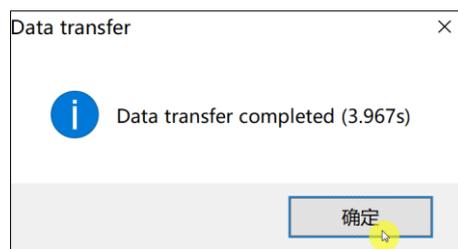
6. 接受默认设置，点击下一步。



7. 点击完成。



窗口右下方显示数据导入进度，完成后发出操作成功和用时消息。



源表 *spice* 的数据已成功导入至目标表 *vegetable* (如果没有数据, 则显示为 NULL)。

	PROD_ID	PROD_NAME	DELIVERY_TIME	VEND_ID	BUY_TIMES	SALE_TIME
1	1	Cerely	2018-11-02	111	[NULL]	[NULL]
2	2	Coriander	2018-12-24	232	[NULL]	[NULL]
3	3	Lettuce	2018-12-31	367	[NULL]	[NULL]
4	4	Luffa	2019-01-01	845	[NULL]	[NULL]
5	5	Basil	[NULL]	[NULL]	51	2018-12-01
6	6	Fennel	[NULL]	[NULL]	29	2019-01-03
7	7	Thyme	[NULL]	[NULL]	10	2019-02-14
8	8	Rosemary	[NULL]	[NULL]	17	2019-07-09

7. 存储过程管理

本章讲述以下内容:

7.1 如何创建存储过程

7.2 如何查看存储过程

7.3 如何编辑存储过程

7.4 如何调用存储过程

7.5 如何删除存储过程

7.1 如何创建存储过程

您可以通过以下 2 种方法创建存储过程：

- 方法 1

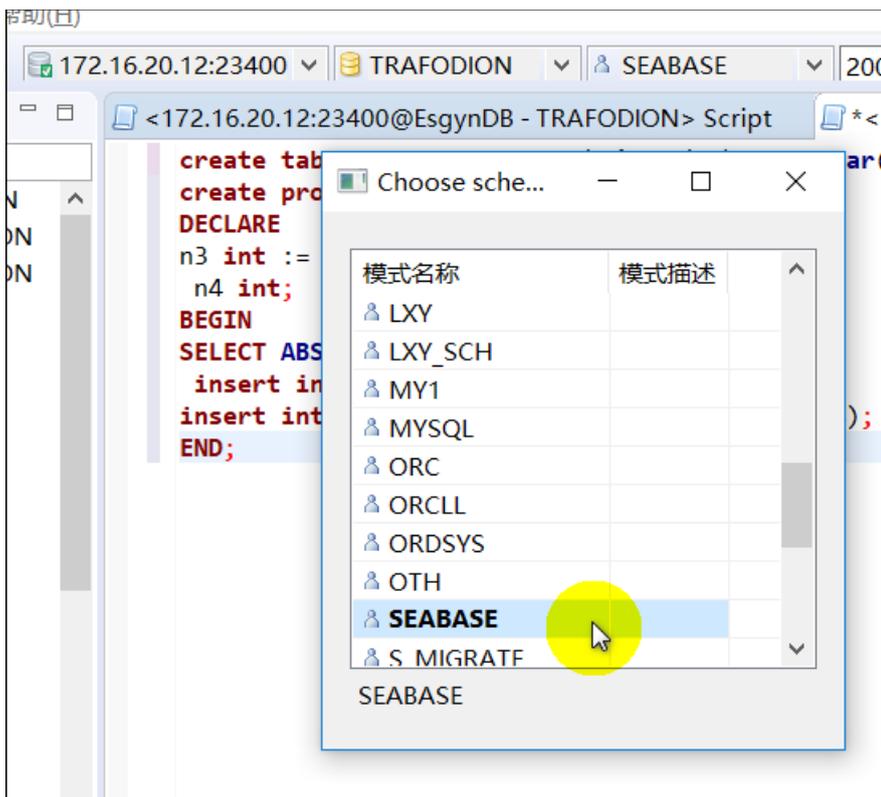
在 SQL 编辑器中执行 SQL 语句创建存储过程。更多信息，请参阅以下内容。

- 方法 2

在数据库导航窗口中创建存储过程。更多信息，请参阅以下内容。

7.1.1 方法 1：通过 SQL 语句创建

1. 打开编辑框窗口，选择对应的 schema。



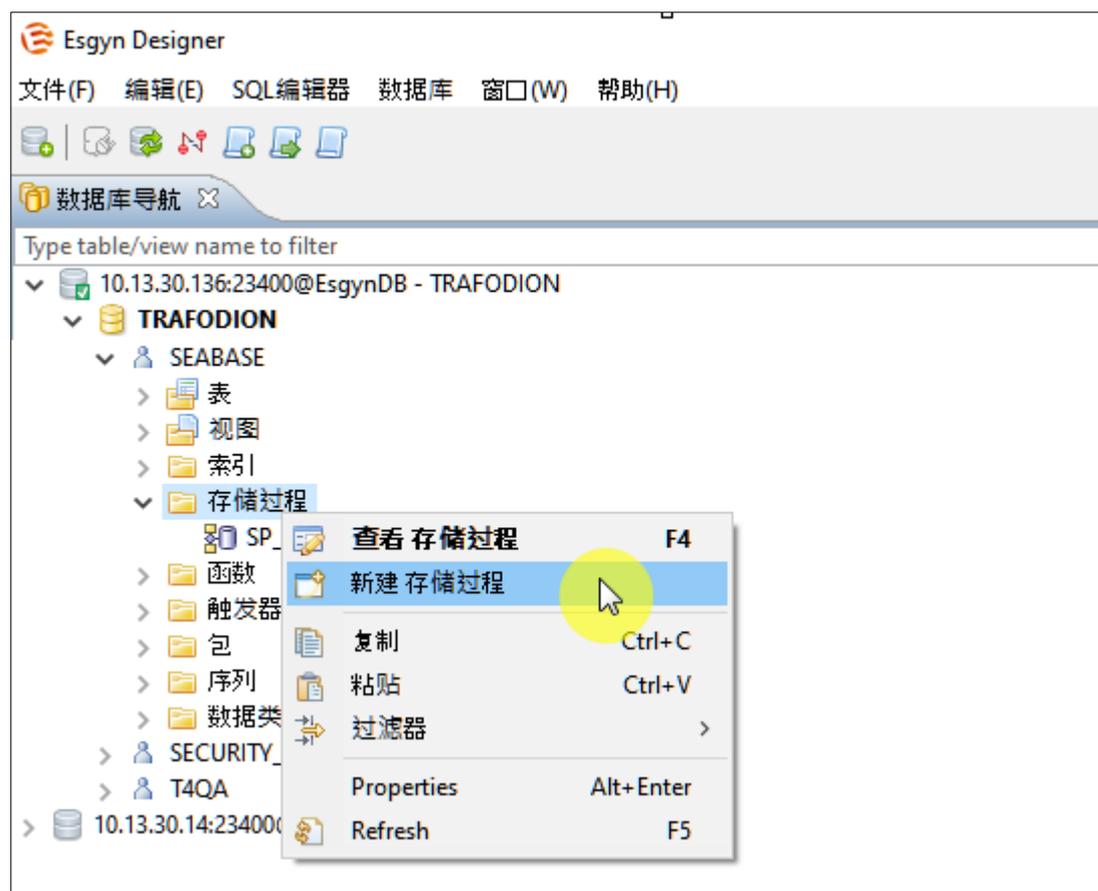
2. 输入创建存储过程语句，点击执行。

7. 存储过程管理

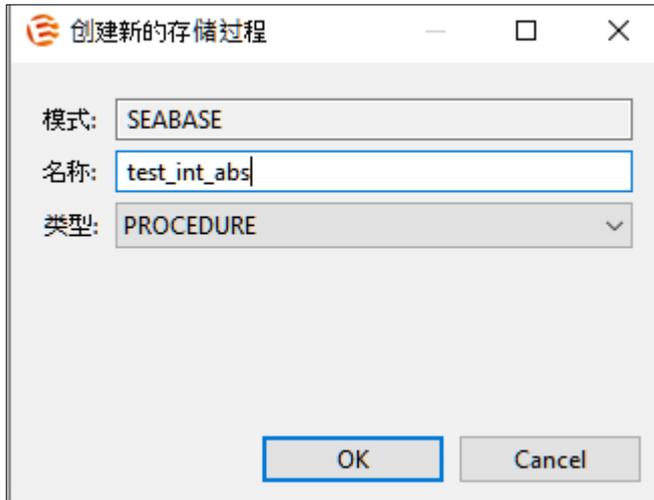
```
2.16.20.12:23400 | TRAFODION | SEABASE | 200 | 
<172.16.20.12:23400@EsgynDB - TRAFODION> Script | *<172.16.20.12:23400@EsgynDB - TRAFODION> Script-1
create table if not exists tb_int_abs(c1 varchar(60),c2 int);
create procedure test_int_abs() as
DECLARE
n3 int := -5;
n4 int;
BEGIN
SELECT ABS(n3) INTO n4 FROM DUAL;
insert into tb_int_abs values ('n3 = ',n3);
insert into tb_int_abs values ('abs(n3) = ',n4);
END;
```

7.1.2 方法 2：通过数据库导航窗口创建

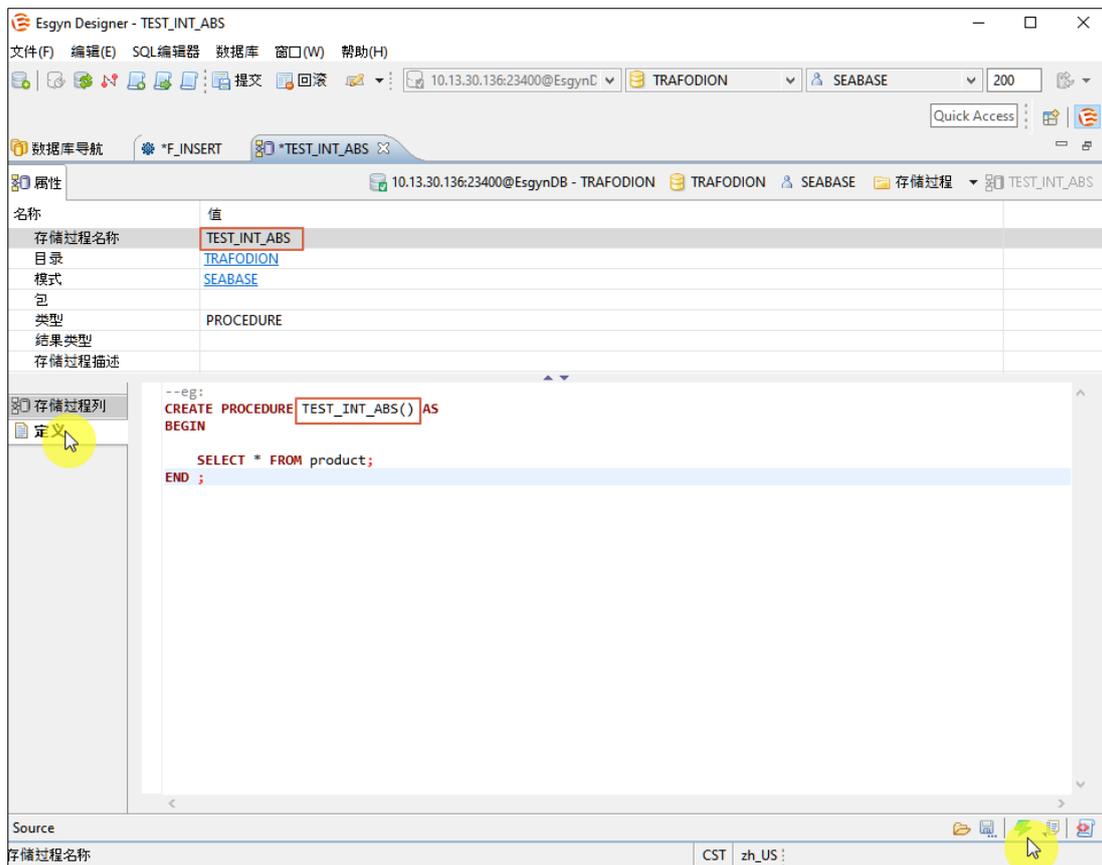
1. 在导航窗口右击存储过程，点击新建存储过程



2. 输入存储过程名称，点击 OK。



3. 在新建窗口中，点击定义，输入创建存储过程的 SQL，点击右下角编译存储过程

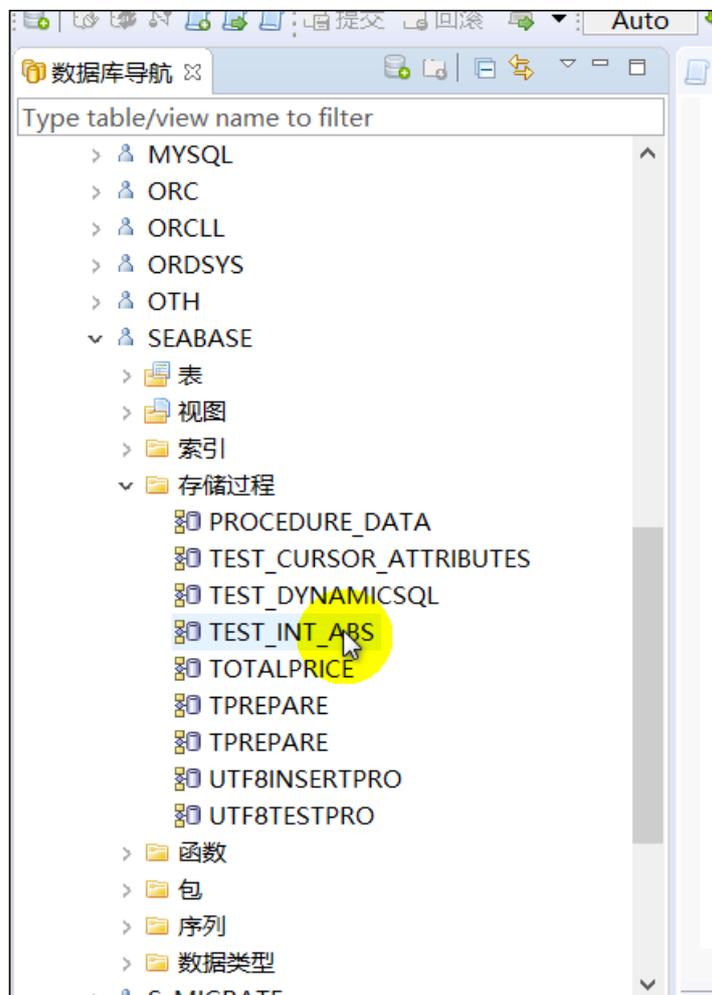


注意

- 存储过程属性窗口中的存储过程名称必须与定义窗口中的 SQL 中定义的存储过程名称一致。

7.2 如何查看存储过程

打开左侧数据库导航窗口，打开对应 schema 下的存储过程，双击选中的存储过程。



7.3 如何编辑存储过程

您能通过以下 2 种方法编辑存储过程：

- **方法 1**

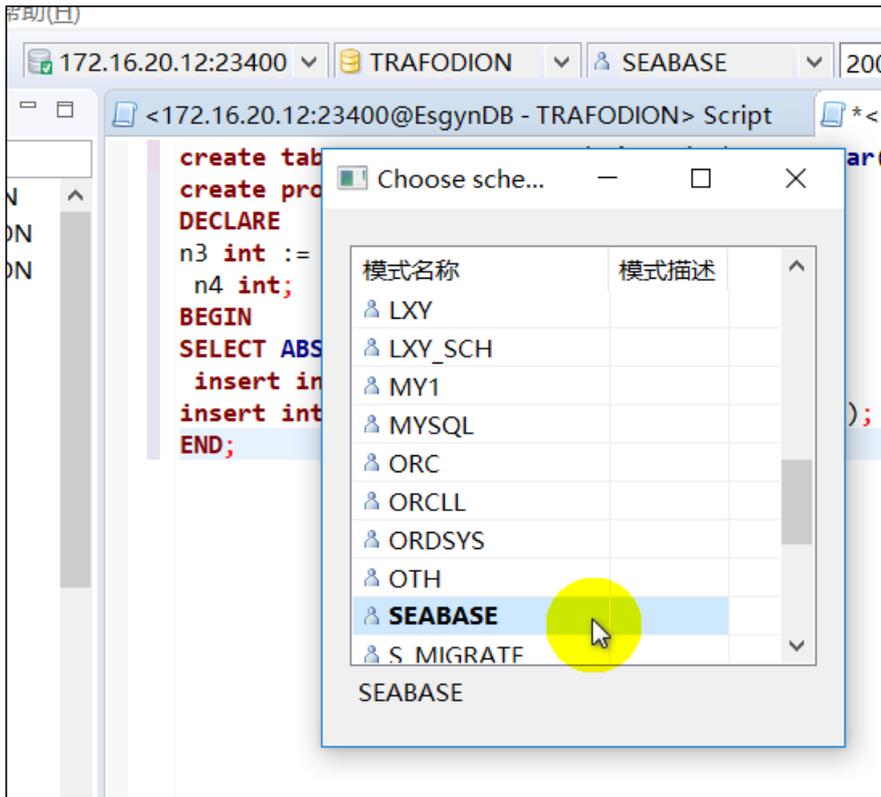
在 **SQL 编辑器** 中执行 SQL 语句编辑存储过程。更多信息，请参阅以下内容。

- **方法 2**

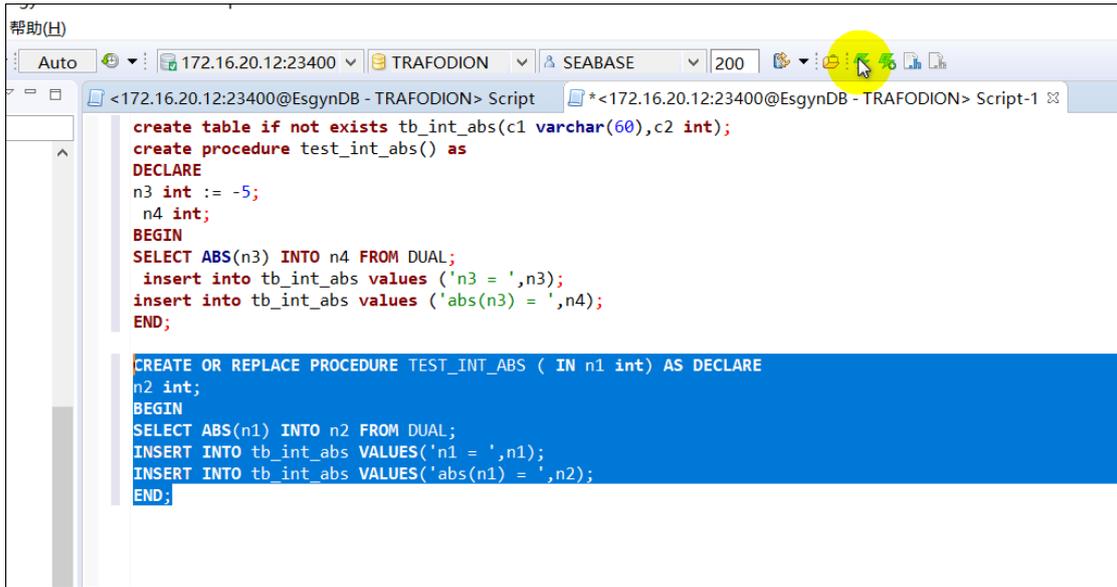
在 **数据库导航窗口** 中编辑存储过程。更多信息，请参阅以下内容。

7.3.1 方法 1：通过 SQL 语句编辑

1. 打开 **编辑框** 窗口，选择对应的 schema。



2. 输入编辑存储过程语句，选中修改语句，点击执行。

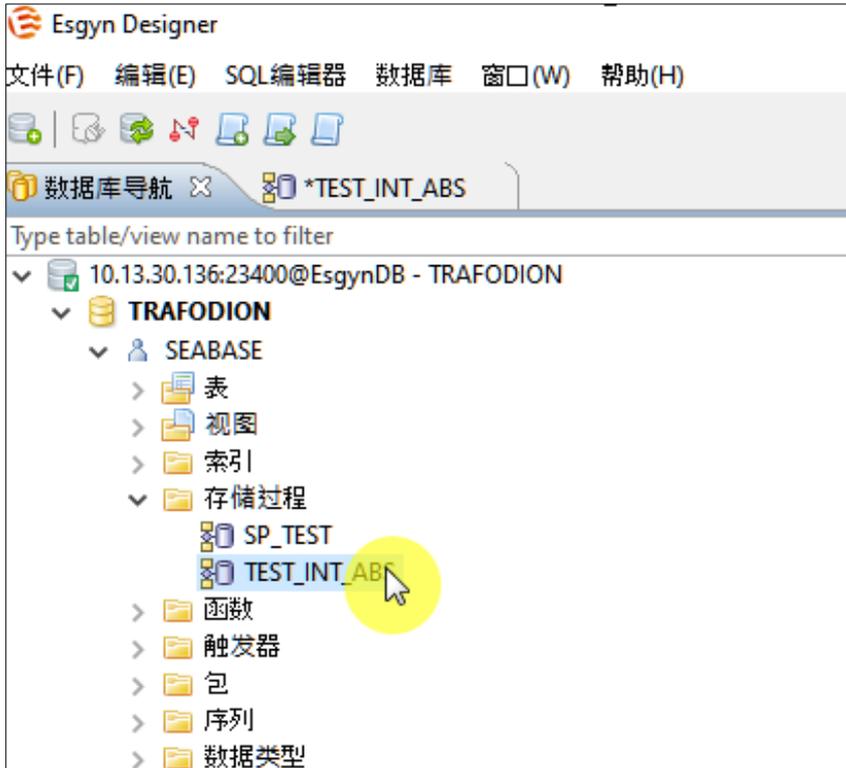


```
create table if not exists tb_int_abs(c1 varchar(60),c2 int);
create procedure test_int_abs() as
DECLARE
n3 int := -5;
n4 int;
BEGIN
SELECT ABS(n3) INTO n4 FROM DUAL;
insert into tb_int_abs values ('n3 = ',n3);
insert into tb_int_abs values ('abs(n3) = ',n4);
END;

CREATE OR REPLACE PROCEDURE TEST_INT_ABS ( IN n1 int) AS DECLARE
n2 int;
BEGIN
SELECT ABS(n1) INTO n2 FROM DUAL;
INSERT INTO tb_int_abs VALUES('n1 = ',n1);
INSERT INTO tb_int_abs VALUES('abs(n1) = ',n2);
END;
```

7.3.2 方法 2：通过数据库导航窗口编辑

1. 打开左侧数据库导航窗口，打开对应 schema 下的存储过程，双击选中的存储过程。



2. 点击定义，输入编辑存储过程的 SQL，点击右下角编译存储过程

7. 存储过程管理

The screenshot shows the Esgyn Designer interface for managing a stored procedure. The window title is "Esgyn Designer - TEST_INT_ABS". The menu bar includes "文件(F)", "编辑(E)", "SQL编辑器", "数据库", "窗口(W)", and "帮助(H)". The toolbar contains icons for file operations, database navigation, and execution. The status bar shows the connection "10.13.30.136:23400@EsgynDB", the database "TRAFODION", the user "SEABASE", and the port "200".

The "数据库导航" (Database Navigator) pane shows the path: "10.13.30.136:23400@EsgynDB - TRAFODION" > "TRAFODION" > "SEABASE" > "存储过程" (Stored Procedures). The "属性" (Properties) pane displays the following details for the procedure:

名称	值
存储过程名称	TEST_INT_ABS
目录	TRAFODION
模式	SEABASE
包	
类型	PROCEDURE
结果类型	
存储过程描述	

The "存储过程列表" (Stored Procedure List) pane is empty. The "定义" (Definition) pane shows the following SQL code:

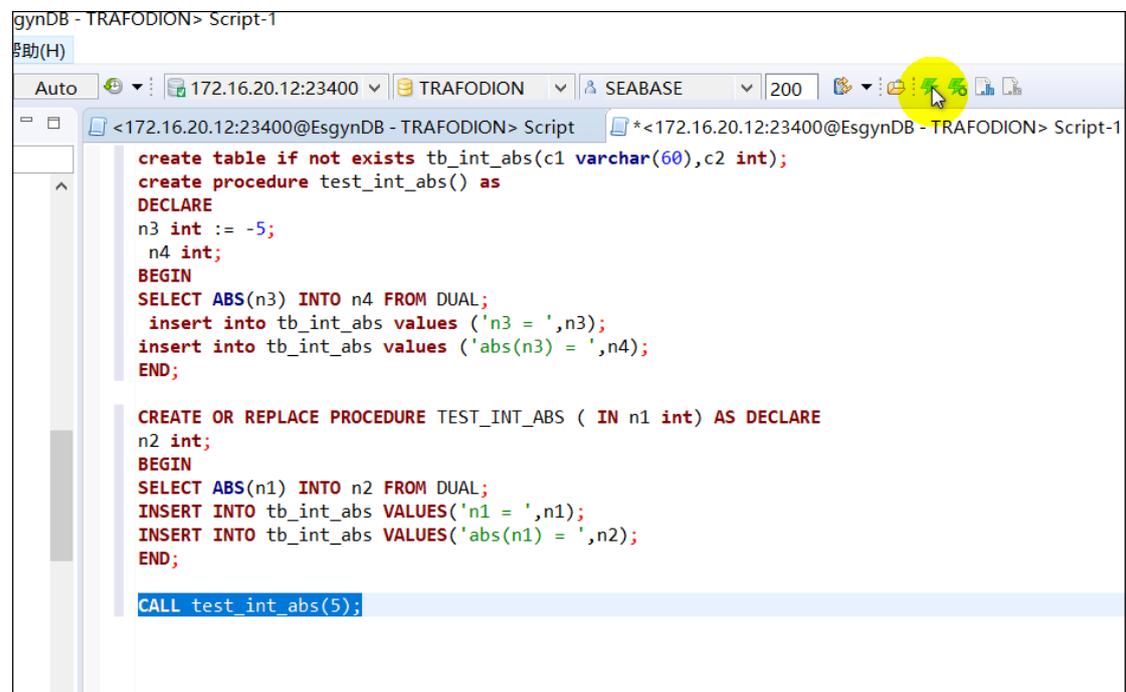
```
--eg:
CREATE OR REPLACE PROCEDURE TEST_INT_ABS() AS

BEGIN
    SELECT * FROM vendor;
END;
```

The "Source" pane at the bottom shows the system tray with a clock displaying "11:57" and the time zone "EST zh_US".

7.4 如何调用存储过程

在编辑器中输入调用语句，执行调用。



The screenshot shows a SQL script editor window titled "gynDB - TRAFODION> Script-1". The editor contains the following SQL code:

```
create table if not exists tb_int_abs(c1 varchar(60),c2 int);
create procedure test_int_abs() as
DECLARE
n3 int := -5;
n4 int;
BEGIN
SELECT ABS(n3) INTO n4 FROM DUAL;
insert into tb_int_abs values ('n3 = ',n3);
insert into tb_int_abs values ('abs(n3) = ',n4);
END;

CREATE OR REPLACE PROCEDURE TEST_INT_ABS ( IN n1 int) AS DECLARE
n2 int;
BEGIN
SELECT ABS(n1) INTO n2 FROM DUAL;
INSERT INTO tb_int_abs VALUES('n1 = ',n1);
INSERT INTO tb_int_abs VALUES('abs(n1) = ',n2);
END;

CALL test_int_abs(5);
```

The code defines a table `tb_int_abs` with columns `c1` (varchar(60)) and `c2` (int). It then defines two stored procedures: `test_int_abs()` and `TEST_INT_ABS (IN n1 int)`. The first procedure inserts values into the table based on the absolute value of `n3`. The second procedure inserts values into the table based on the absolute value of `n1`. Finally, the `CALL test_int_abs(5);` statement is highlighted in blue, indicating it is the current line of execution.

7.5 如何删除存储过程

您能通过以下 2 种方法删除存储过程：

- **方法 1**

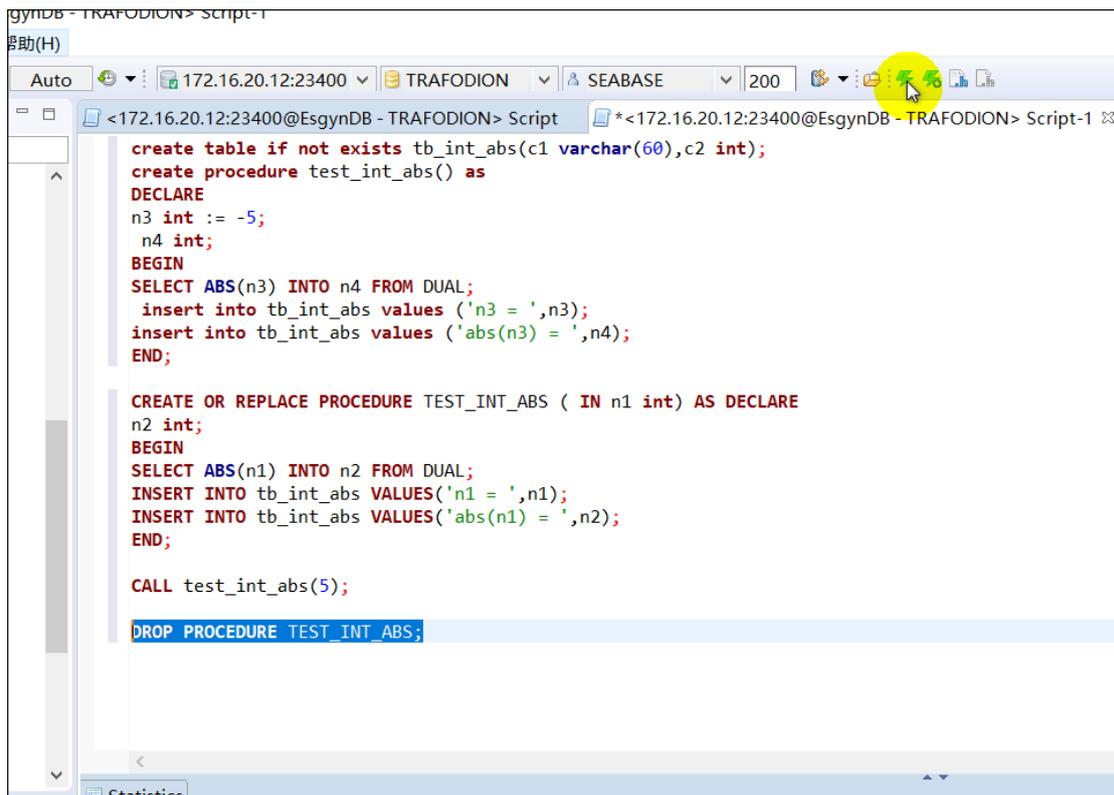
在 **SQL 编辑器** 中执行 SQL 语句删除存储过程。更多信息，请参阅以下内容。

- **方法 2**

在 **数据库导航窗口** 中删除存储过程。更多信息，请参阅以下内容。

7.5.1 方法 1：通过 SQL 语句进行删除。

在编辑器中输入删除语句，执行调用。



The screenshot shows a SQL editor window titled "EsgynDB - TRAFODION> Script-1". The editor contains the following SQL script:

```
create table if not exists tb_int_abs(c1 varchar(60),c2 int);
create procedure test_int_abs() as
DECLARE
n3 int := -5;
n4 int;
BEGIN
SELECT ABS(n3) INTO n4 FROM DUAL;
insert into tb_int_abs values ('n3 = ',n3);
insert into tb_int_abs values ('abs(n3) = ',n4);
END;

CREATE OR REPLACE PROCEDURE TEST_INT_ABS ( IN n1 int) AS DECLARE
n2 int;
BEGIN
SELECT ABS(n1) INTO n2 FROM DUAL;
INSERT INTO tb_int_abs VALUES('n1 = ',n1);
INSERT INTO tb_int_abs VALUES('abs(n1) = ',n2);
END;

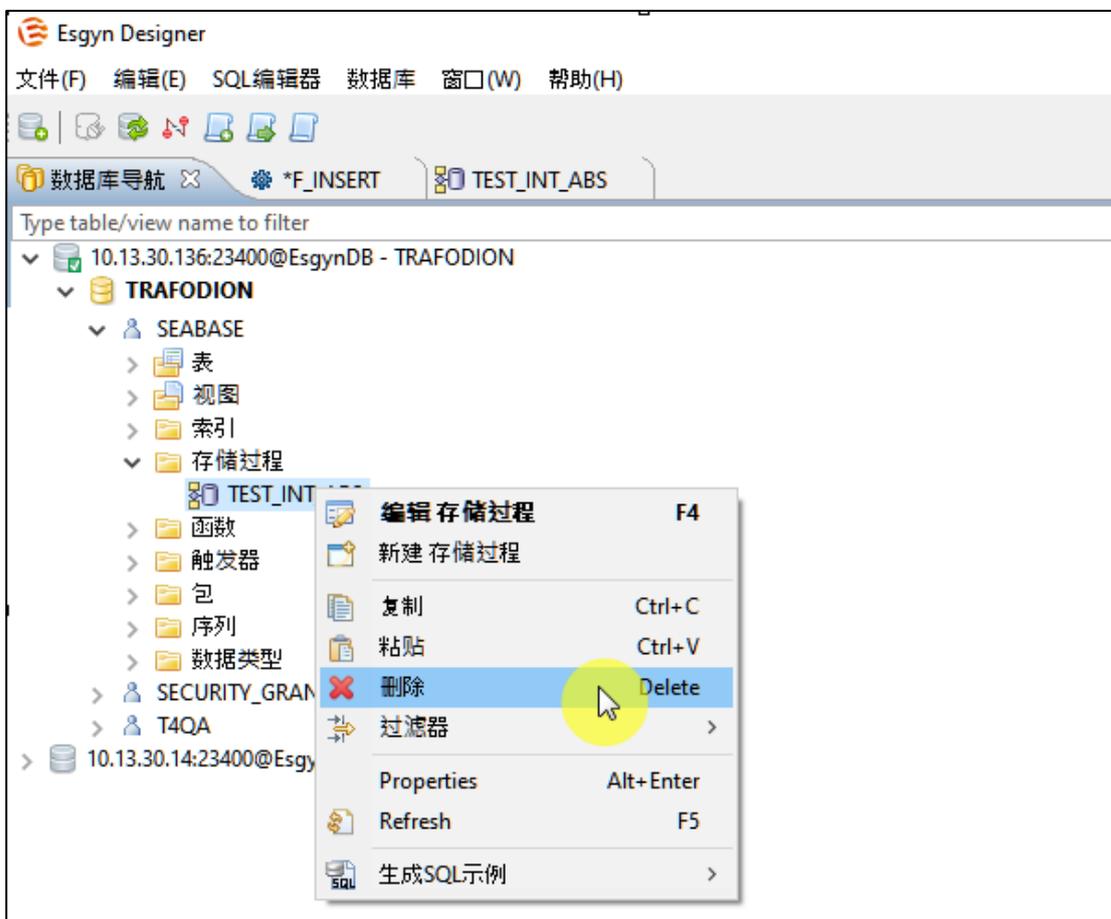
CALL test_int_abs(5);

DROP PROCEDURE TEST_INT_ABS;
```

The last line, `DROP PROCEDURE TEST_INT_ABS;`, is highlighted in blue. The window title bar shows the connection details: "172.16.20.12:23400", "TRAFODION", "SEABASE", and "200".

7.5.2 方法 2：通过数据库导航窗口删除。

1. 在数据库导航窗口中，右击存储过程 TEST_INT_ABS。
2. 点击删除。



提示

如需删除多个存储过程，选中多个存储过程的同时按下 Ctrl，再点击右键，点击删除。

8. 函数管理

本章讲述以下内容:

8.1 如何创建函数

8.2 如何查看函数

8.3 如何编辑函数

8.4 如何调用函数

8.5 如何删除函数

8.1 如何创建函数

您能通过以下 2 种方法创建函数：

- 方法 1

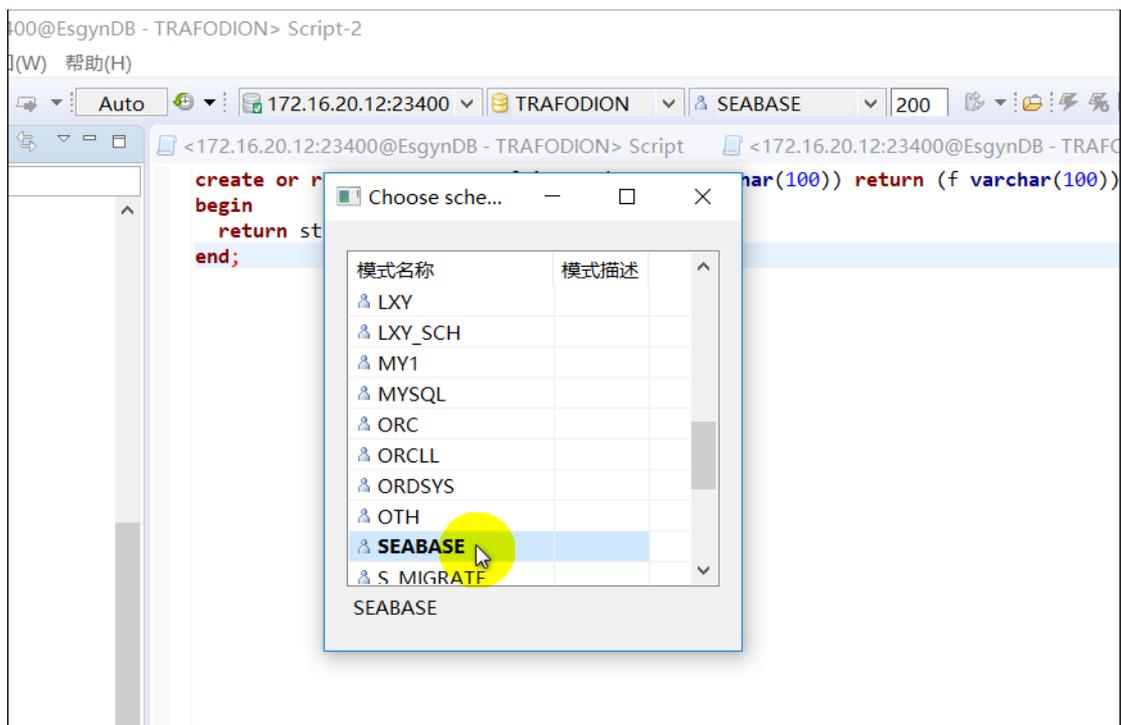
在 SQL 编辑器中执行 SQL 语句创建函数。更多信息，请参阅以下内容。

- 方法 2

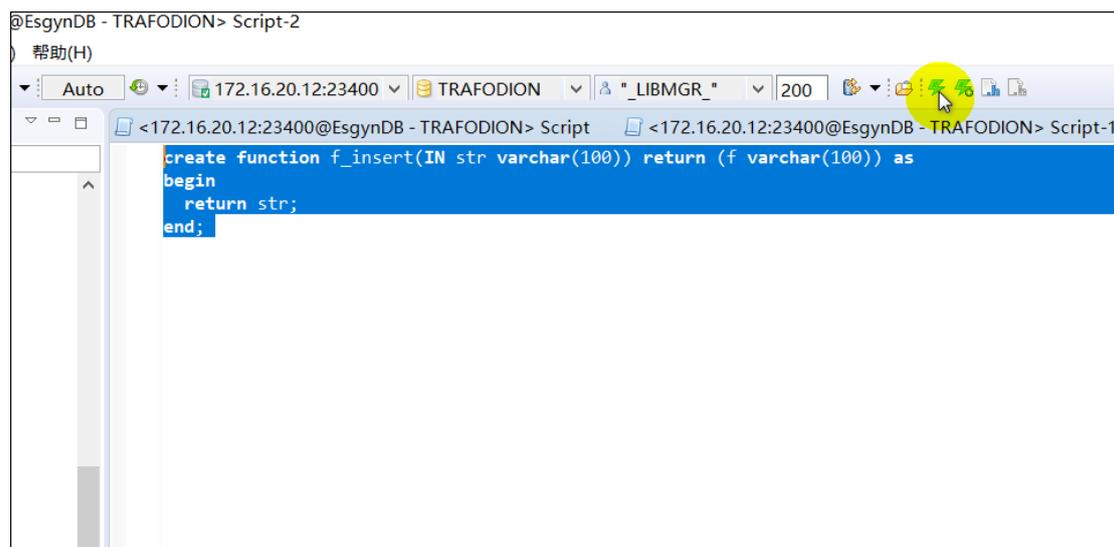
在数据库导航窗口中创建函数。更多信息，请参阅以下内容。

8.1.1 方法 1：通过 SQL 语句创建

1. 打开编辑框窗口，选择对应的 schema。

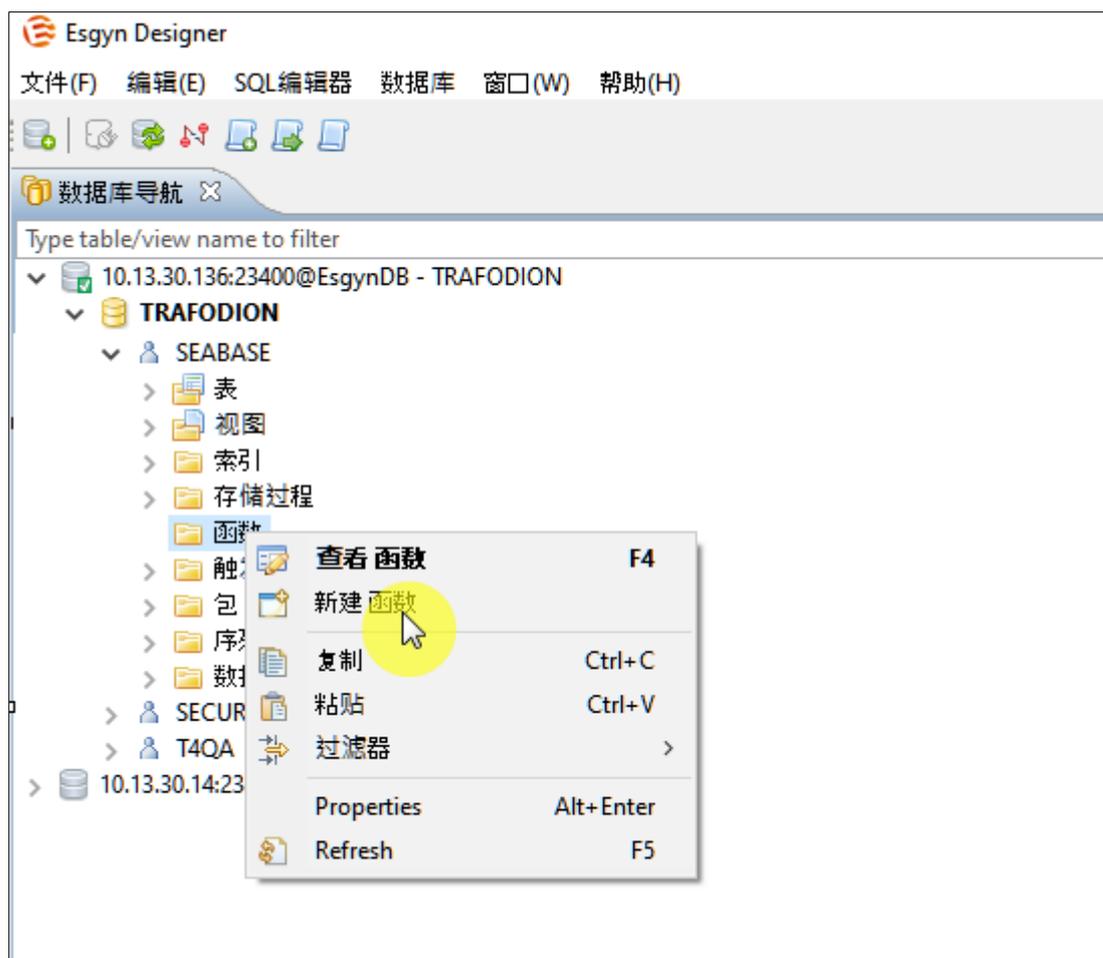


2. 输入创建函数语句，点击执行。



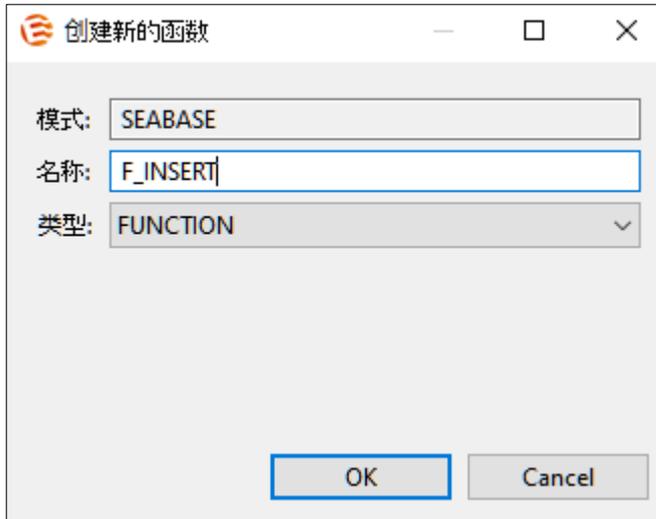
8.1.2 方法 2: 通过数据库导航窗口创建

1. 在导航窗口右击函数，点击新建函数

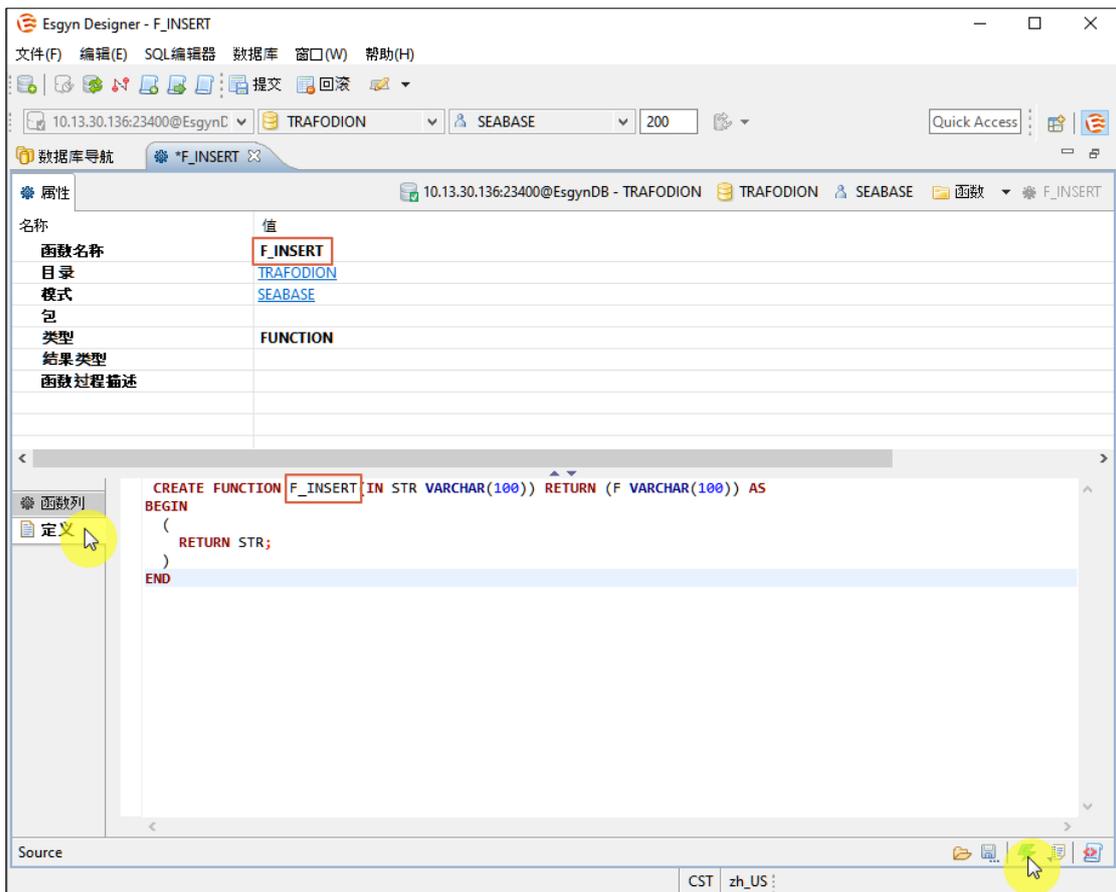


8. 函数管理

2. 输入函数名称，点击 **OK**。



3. 在新建窗口中，点击**定义**，输入创建函数的 SQL，点击右下角**编译函数**

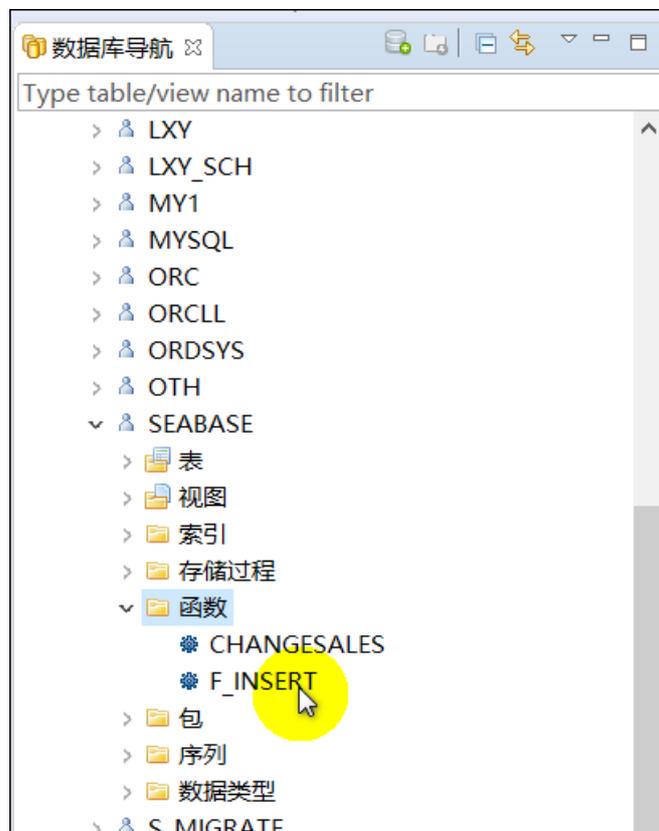


注意

- 函数属性窗口中的函数名称必须与定义窗口中的 SQL 中定义的函数名称一致。

8.2 如何查看函数

打开左侧数据库导航窗口，打开对应 schema 下的函数，双击选中的函数。



8.3 如何编辑函数

您能通过以下 2 种方法编辑函数：

- 方法 1

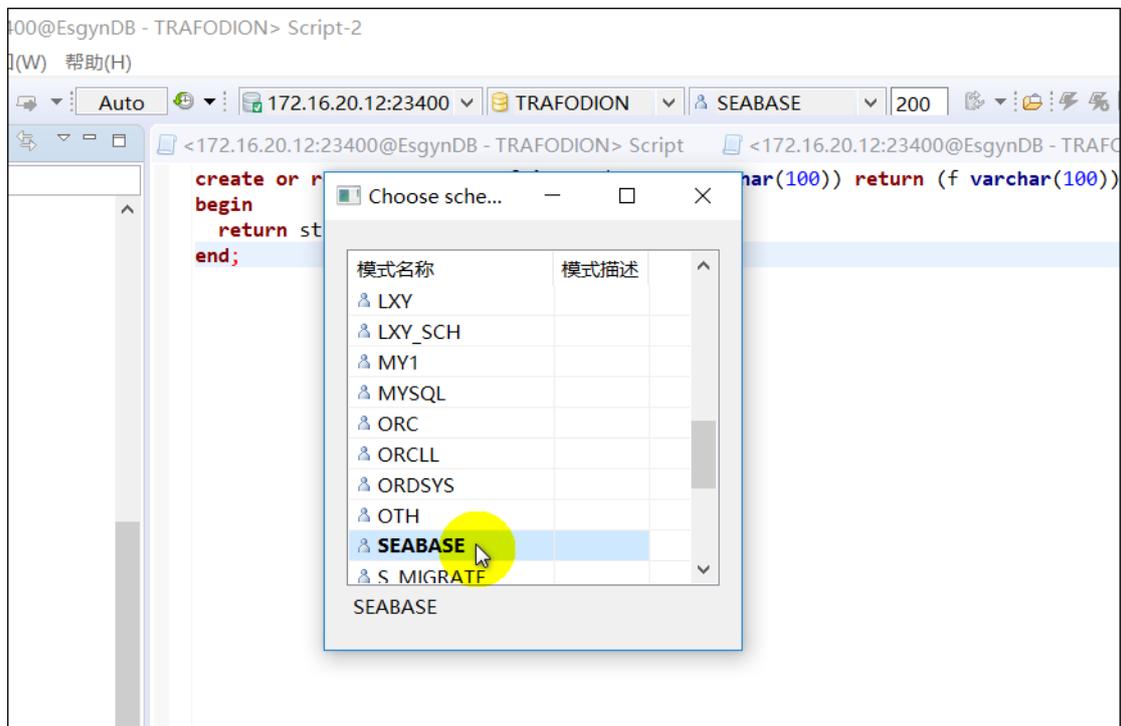
在 SQL 编辑器中执行 SQL 语句编辑函数。更多信息，请参阅以下内容。

- 方法 2

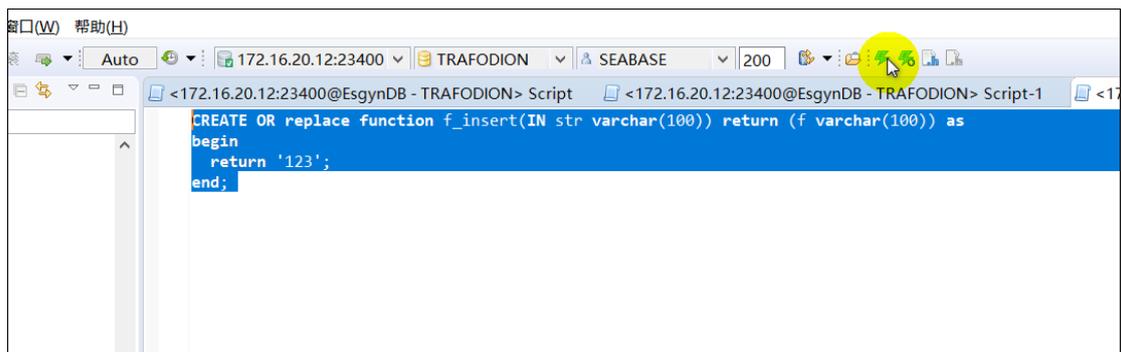
在数据库导航窗口中编辑函数。更多信息，请参阅以下内容。

8.3.1 方法 1：通过 SQL 语句进行编辑

1. 打开编辑框窗口，选择对应的 schema。

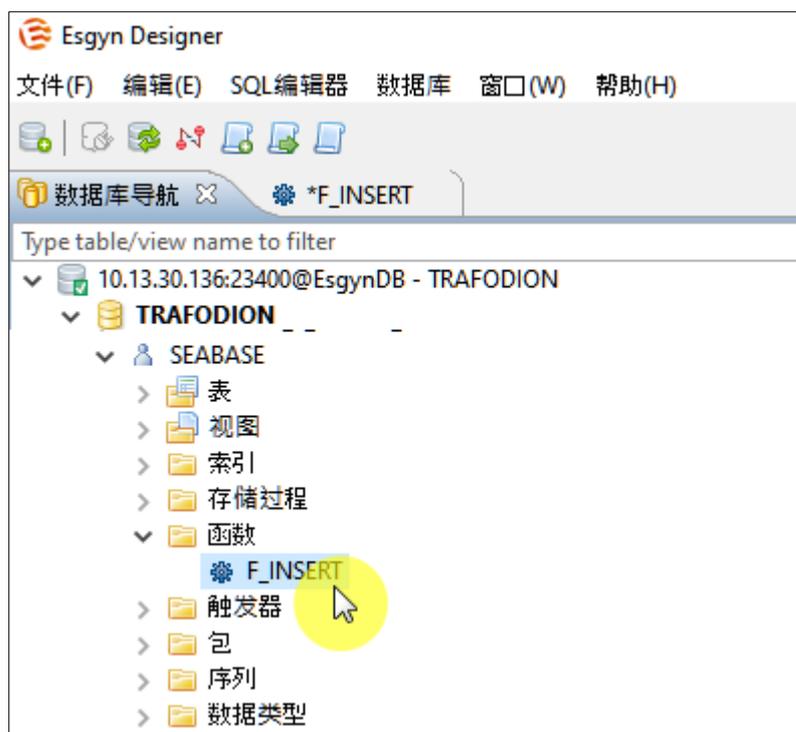


2. 编辑修改函数语句，选中修改语句，点击执行。



8.3.2 方法 2：通过数据库导航窗口编辑

1. 打开左侧数据库导航窗口，打开对应 schema 下的函数，双击选中的函数。



2. 点击定义，输入编辑函数的 SQL，点击右下角编译函数

8. 函数管理

The screenshot shows the Esgyn Designer interface for managing a function named F_INSERT. The top menu bar includes '文件(F)', '编辑(E)', 'SQL编辑器', '数据库', '窗口(W)', and '帮助(H)'. The toolbar contains icons for file operations and a '提交' (Submit) button. The status bar shows the connection '10.13.30.136:23400@EsgynDB' and the database 'TRAFODION'. The '数据库导航' (Database Navigator) pane shows the function 'F_INSERT' under the '函数' (Functions) folder. The '属性' (Properties) pane displays the following details:

名称	值
函数名称	F_INSERT
目录	TRAFODION
模式	SEABASE
包	
类型	FUNCTION
结果类型	
函数过程描述	

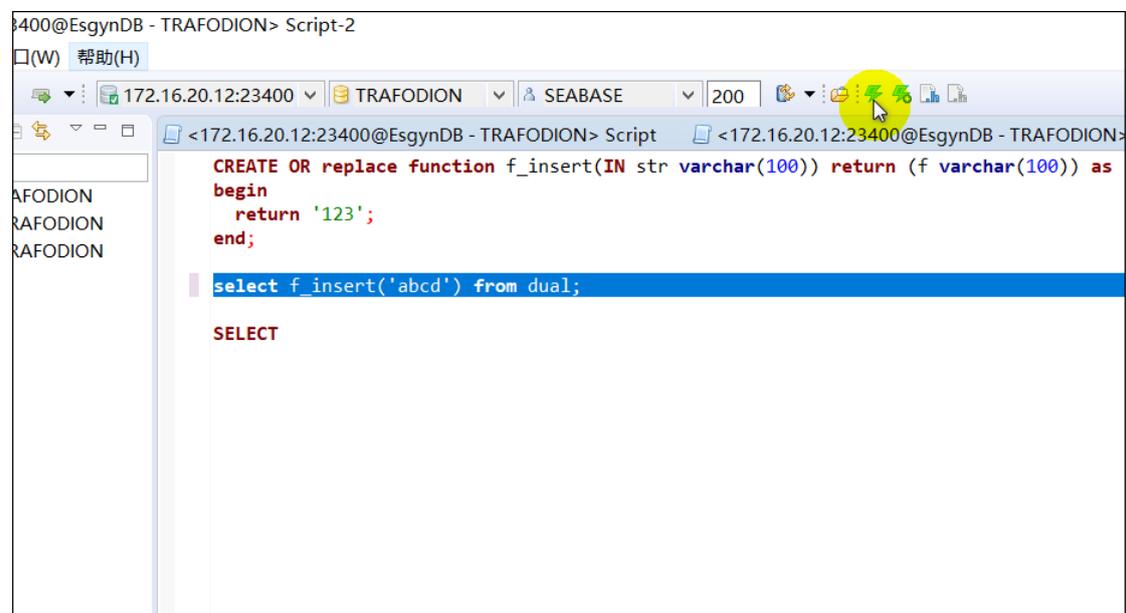
The '函数列表' (Function List) pane shows '定义' (Definition). The 'Source' editor contains the following SQL code:

```
CREATE OR REPLACE FUNCTION F_INSERT(IN STR VARCHAR(100)) RETURN (F VARCHAR(100)) AS
BEGIN
    (
        RETURN STR;
    )
END;
```

The bottom status bar shows 'Source' and 'CST zh_US'.

8.4 如何调用函数

在编辑器中输入调用语句，执行调用



The screenshot shows a SQL script editor window titled "Script-2" with a toolbar and a script area. The script area contains the following SQL code:

```
CREATE OR replace function f_insert(IN str varchar(100)) return (f varchar(100)) as
begin
    return '123';
end;

select f_insert('abcd') from dual;

SELECT
```

The script is displayed in a window with a toolbar at the top containing icons for file operations and a search icon. The window title bar shows the connection details: "172.16.20.12:23400 TRAFODION SEABASE 200". The script area has a blue highlight on the line "select f_insert('abcd') from dual;".

8.5 如何删除函数

您能通过以下 2 种方法删除函数：

- 方法 1

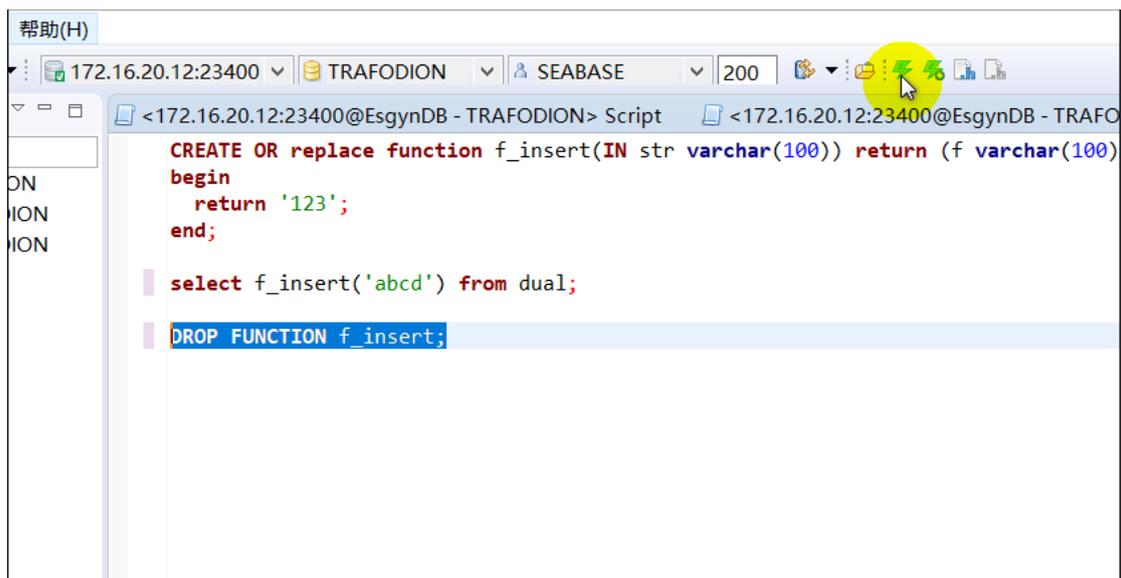
在 SQL 编辑器中执行 SQL 语句删除函数。更多信息，请参阅以下内容。

- 方法 2

在数据库导航窗口中删除函数。更多信息，请参阅以下内容。

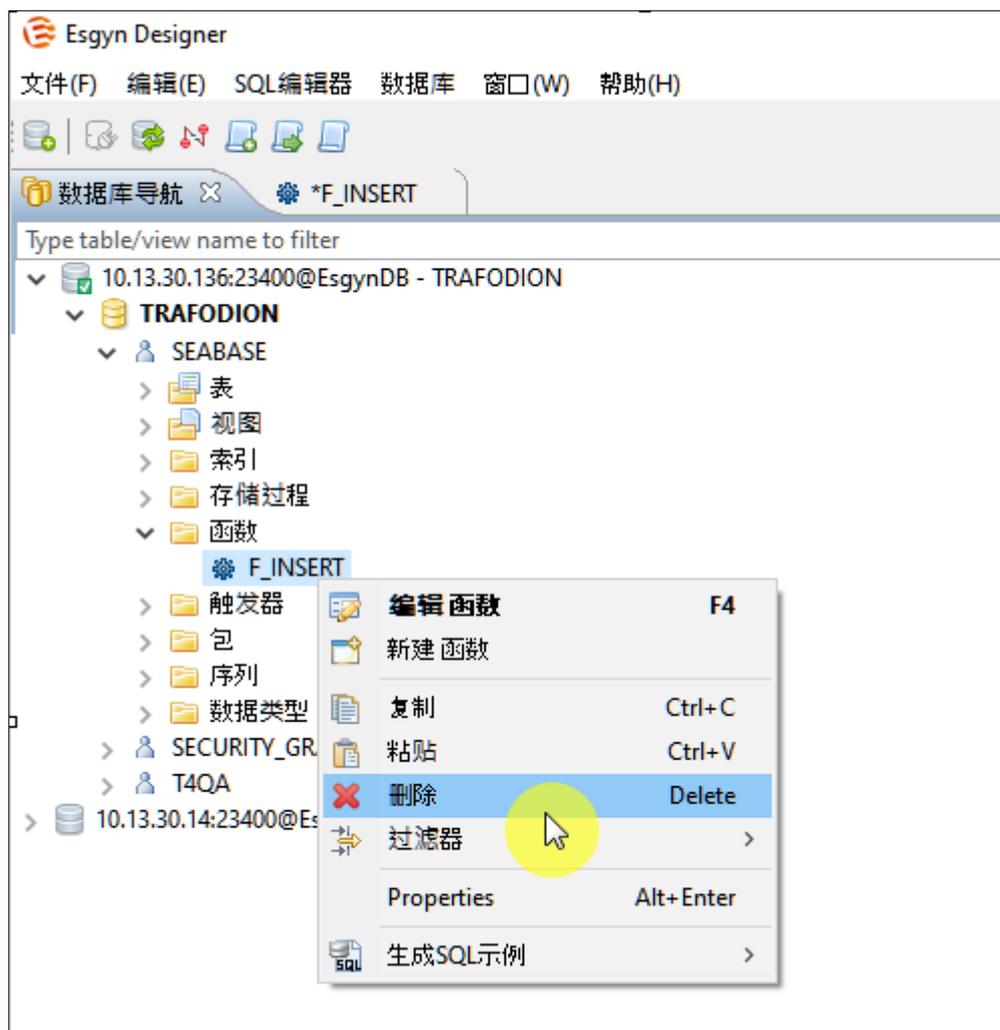
8.5.1 方法 1：通过 SQL 语句进行删除

在编辑器中输入删除语句，执行调用。



8.5.2 方法 2：通过数据库导航窗口删除

1. 在数据库导航窗口中，右击函数 F_INSERT。
2. 点击删除。



 提示

如需删除多个函数，选中多个函数的同时按下 Ctrl，再点击右键，点击删除。

9. 视图管理

本章讲述以下内容:

9.1 如何创建视图

9.2 如何编辑视图

9.3 如何删除视图

9.1 如何创建视图

您能通过以下 2 种方法创建视图：

- 方法 1

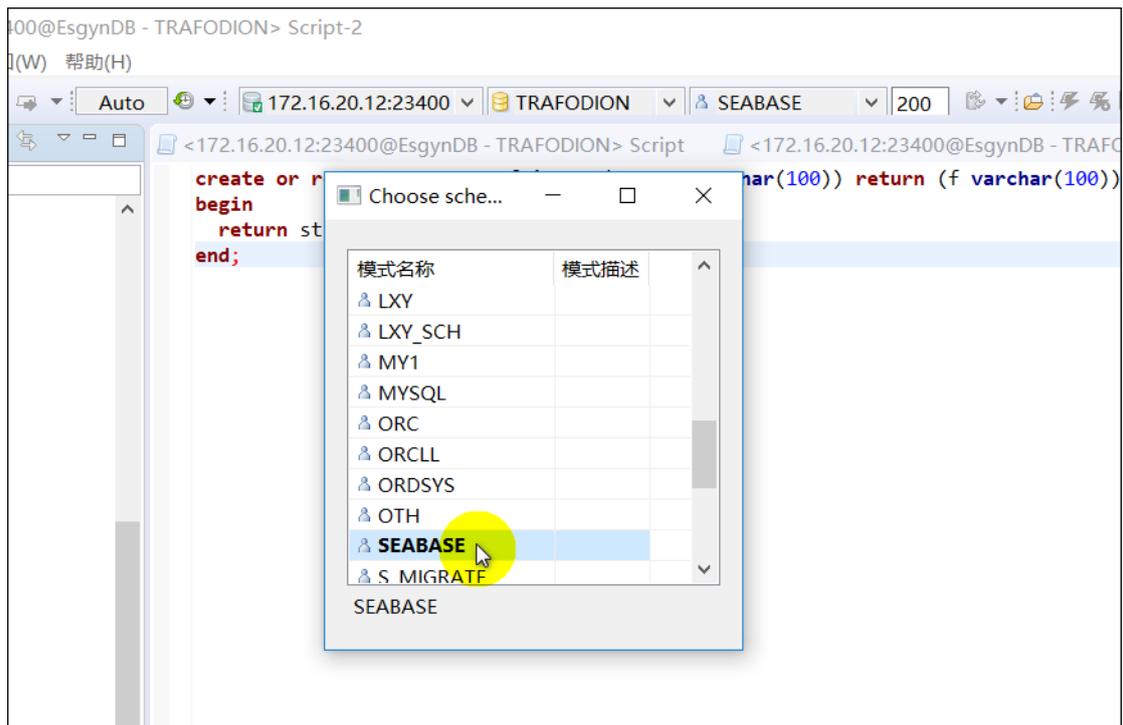
在 SQL 编辑器中执行 SQL 语句创建视图。更多信息，请参阅以下内容。

- 方法 2

在数据库导航窗口中创建视图。更多信息，请参阅以下内容。

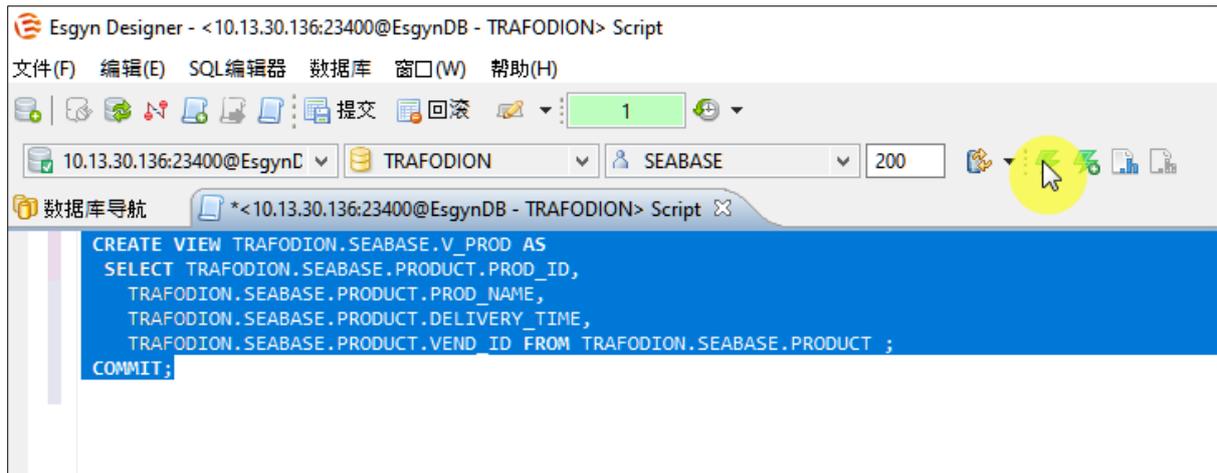
9.1.1 方法 1: 通过 SQL 语句创建

1. 打开编辑框窗口，选择对应的 schema。



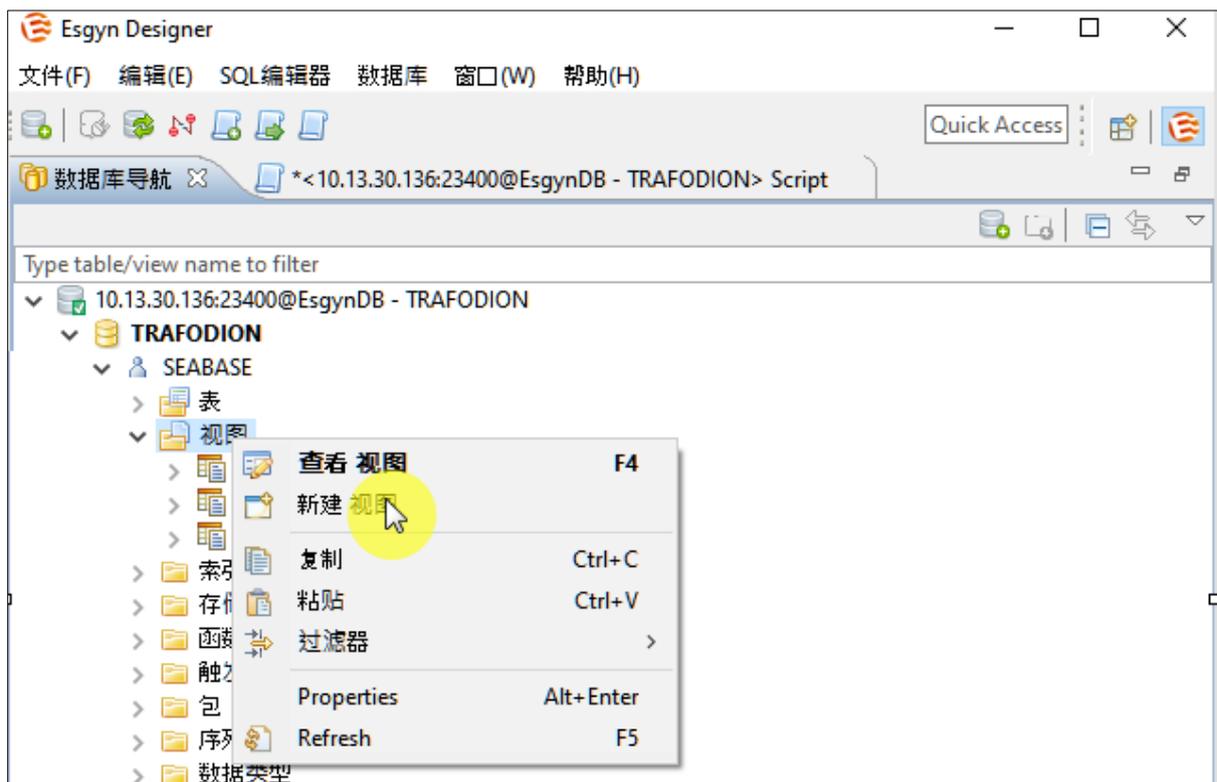
2. 输入创建视图语句，点击执行。

9. 视图管理



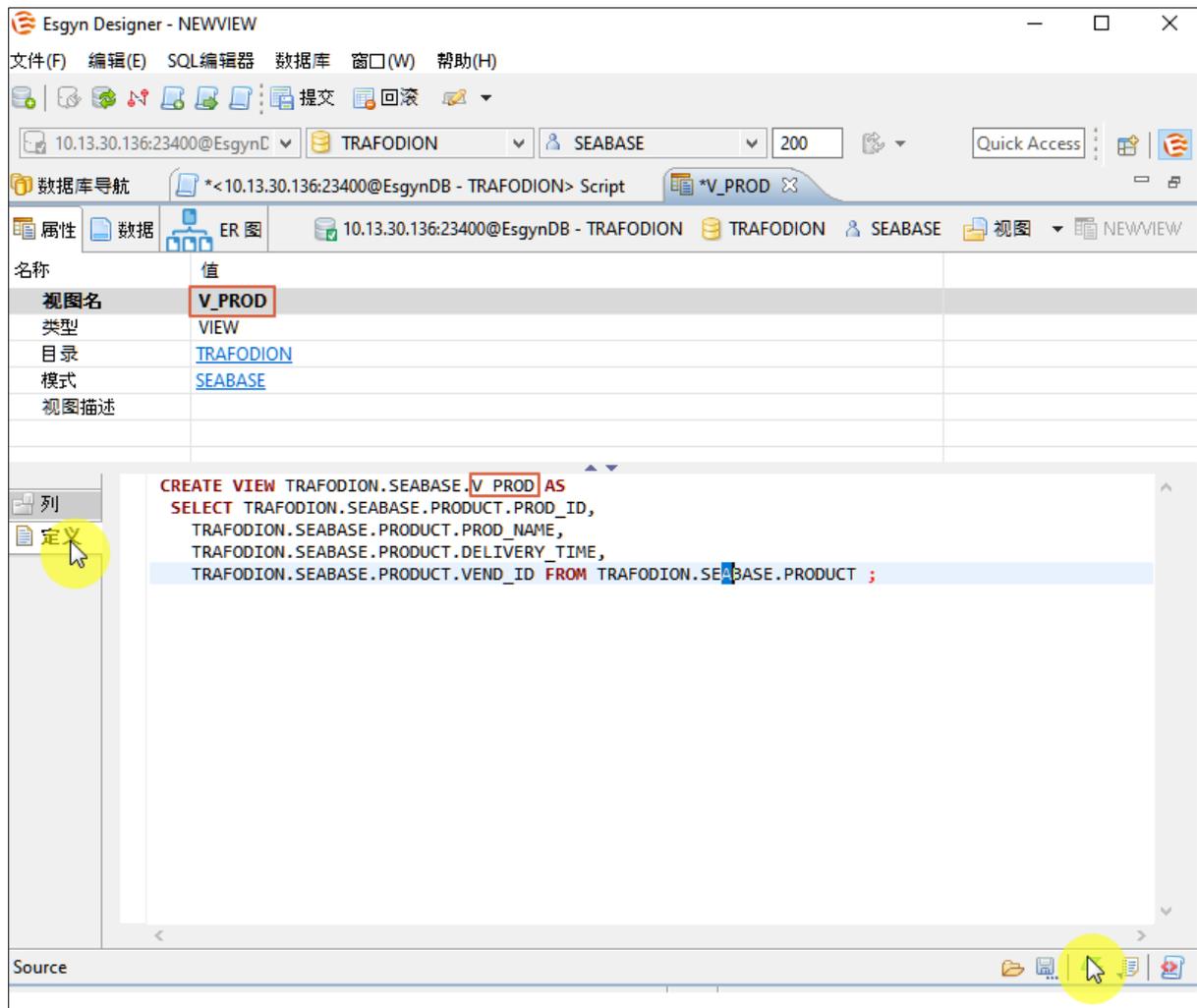
9.1.2 方法 2：通过数据库导航窗口创建

1. 在导航窗口右击视图，点击新建视图



2. 在新建窗口中，点击定义，输入创建视图的 SQL，点击右下角编译视图

9. 视图管理

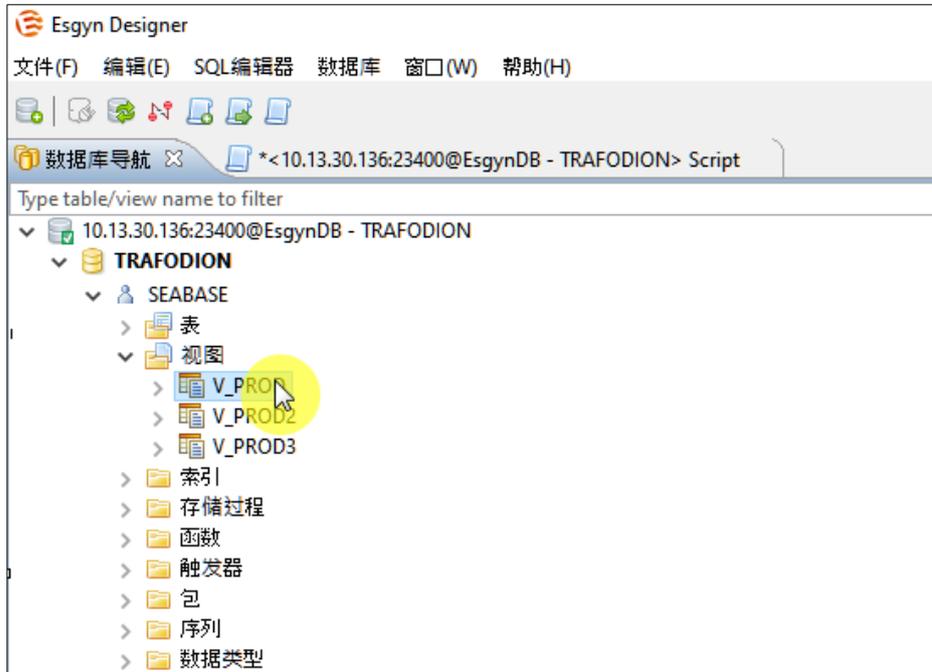


注意

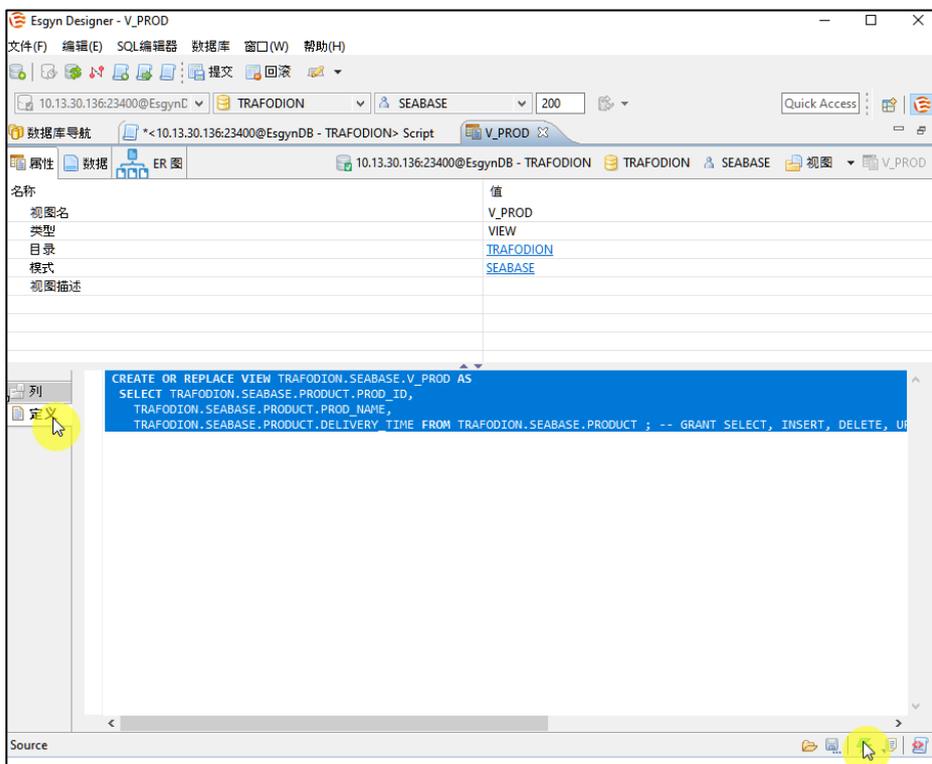
- 视图属性窗口中的视图名称必须与定义窗口中的 SQL 中定义的视图名称一致。

9.2 如何编辑视图

1. 打开左侧数据库导航窗口，打开对应 schema 下的视图，双击选中的视图。



2. 在属性窗口中点击定义，输入修改视图的 SQL，点击右下角编译视图



9.3 如何删除视图

您能通过以下 2 种方法删除视图：

- **方法 1**

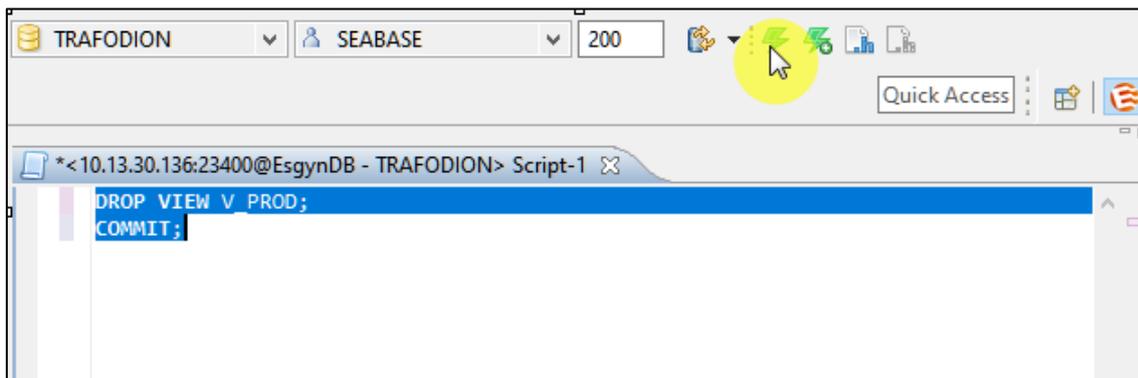
在 **SQL 编辑器** 中执行 SQL 语句删除视图。更多信息，请参阅以下内容。

- **方法 2**

在 **数据库导航窗口** 中删除视图。更多信息，请参阅以下内容。

9.3.1 方法 1: 通过 SQL 语句进行删除

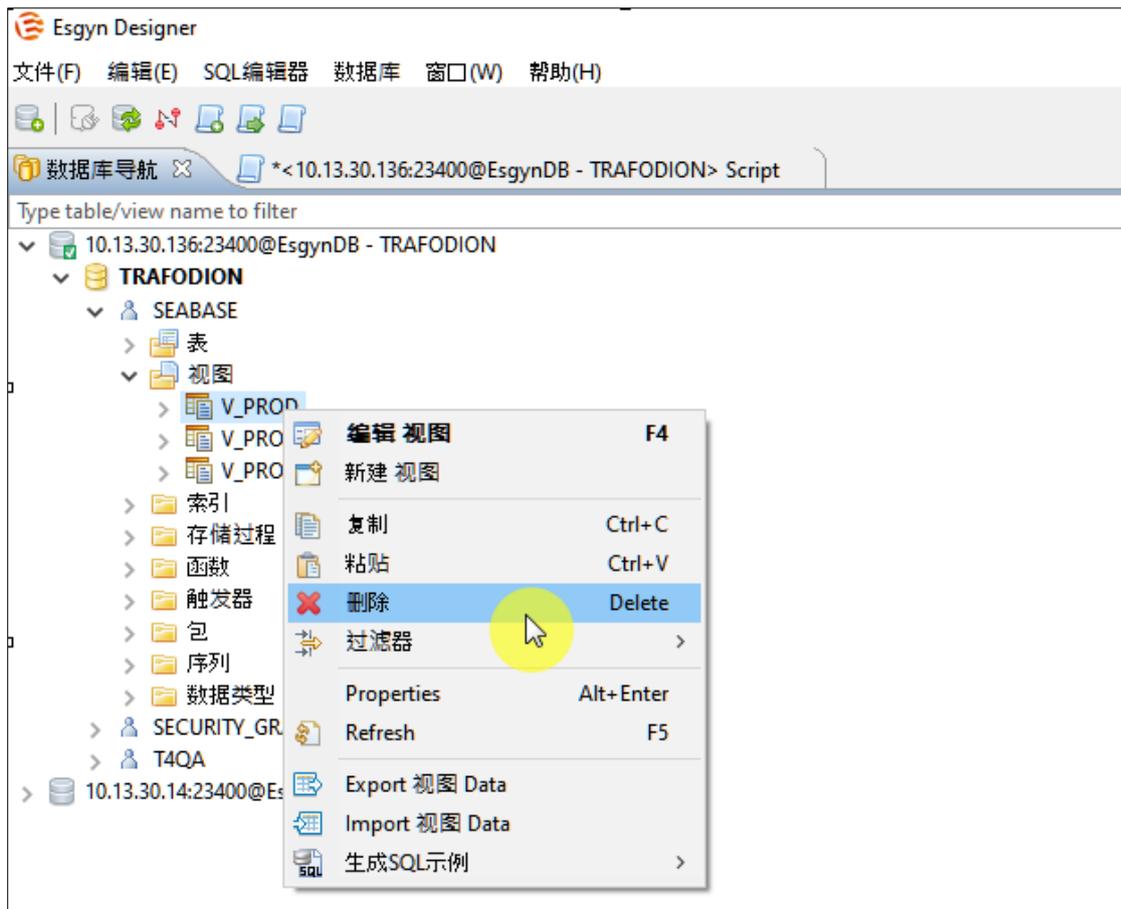
在编辑器中输入删除语句，执行调用。



9.3.2 方法 2: 通过数据库导航窗口删除

1. 在 **数据库导航窗口** 中，右击视图 V_PROD。
2. 点击 **删除**。

9. 视图管理



提示

如需删除多个视图，选中多个视图的同时按下 Ctrl，再点击右键，点击删除。

10. 触发器管理

本章讲述以下内容：

10.1 如何创建触发器

10.2 如何编辑触发器

10.3 如何删除触发器

10.1 如何创建触发器

您能通过以下 2 种方法创建触发器：

- 方法 1

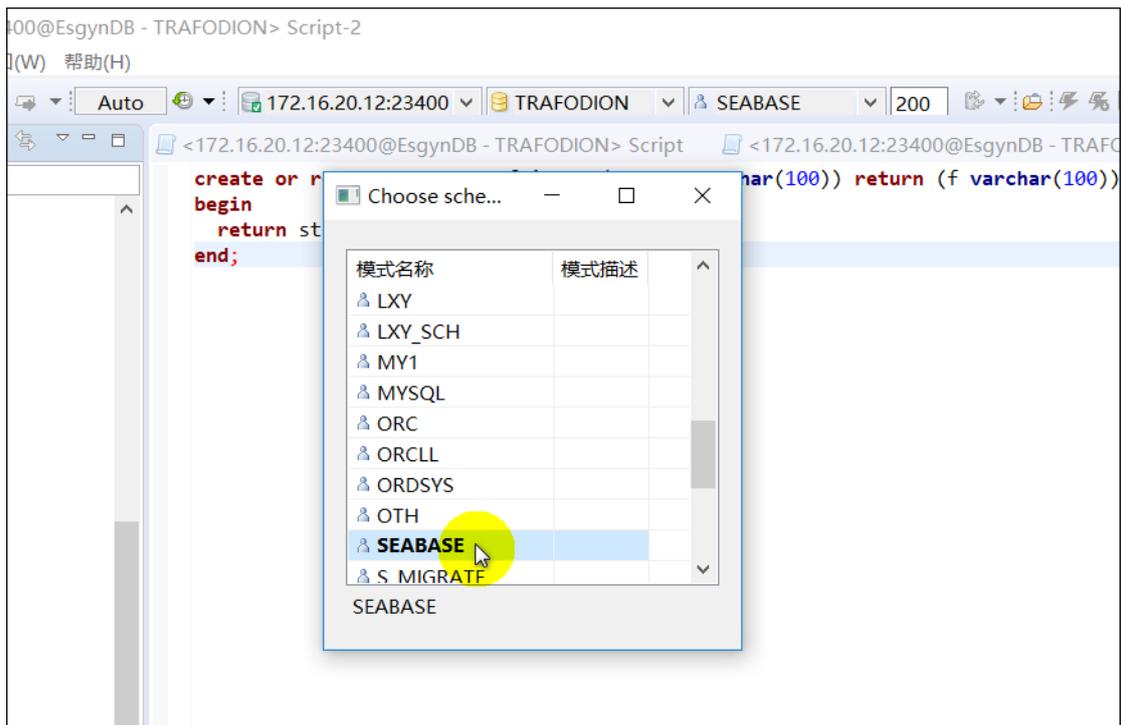
在 SQL 编辑器中执行 SQL 语句创建触发器。更多信息，请参阅以下内容。

- 方法 2

在数据库导航窗口中创建触发器。更多信息，请参阅以下内容。

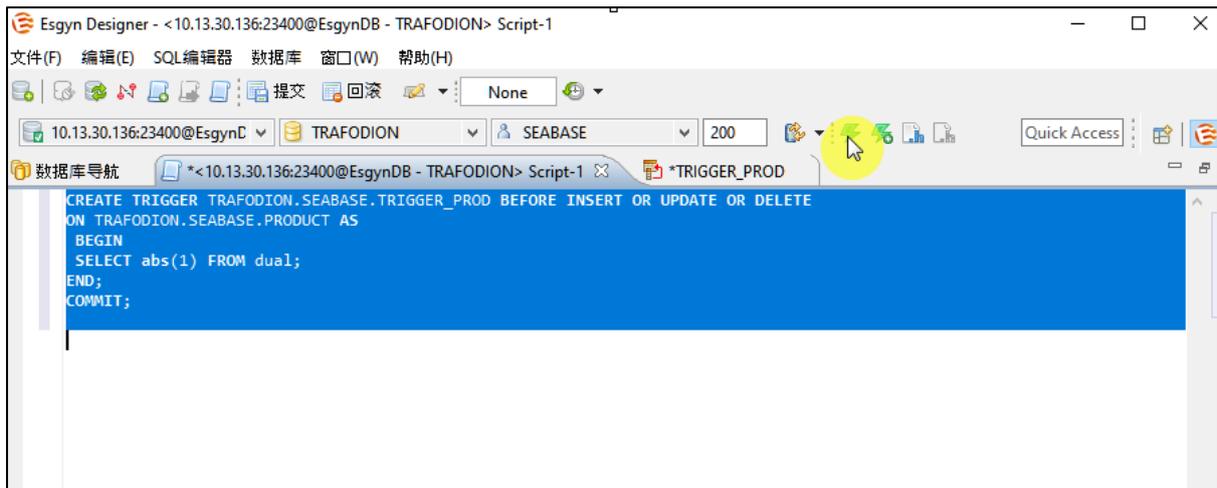
10.1.1 方法 1：通过 SQL 语句创建

1. 打开编辑框窗口，选择对应的 schema。



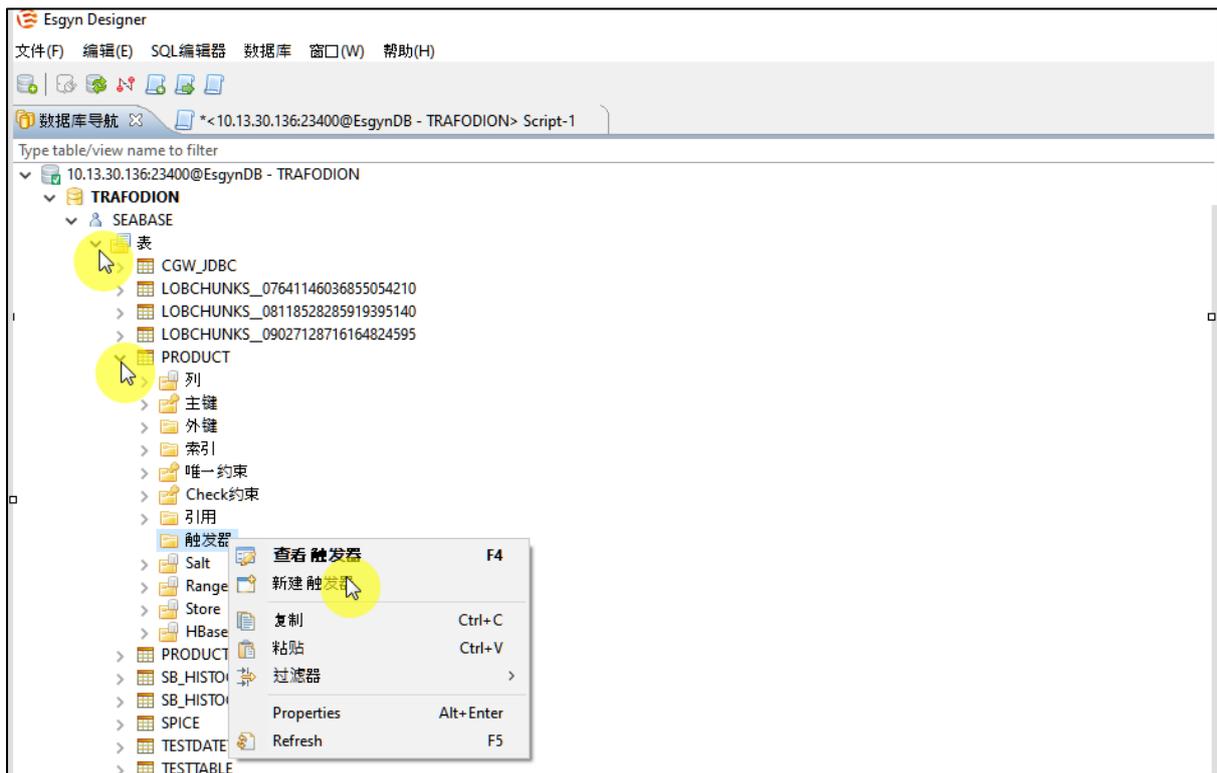
2. 输入创建触发器语句，点击执行。

10. 触发器管理



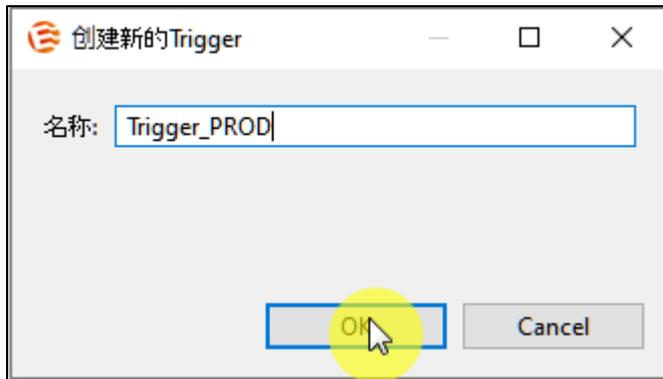
10.1.2 方法 2：通过数据库导航窗口创建

1. 在导航窗口表的下面右击触发器，点击新建触发器

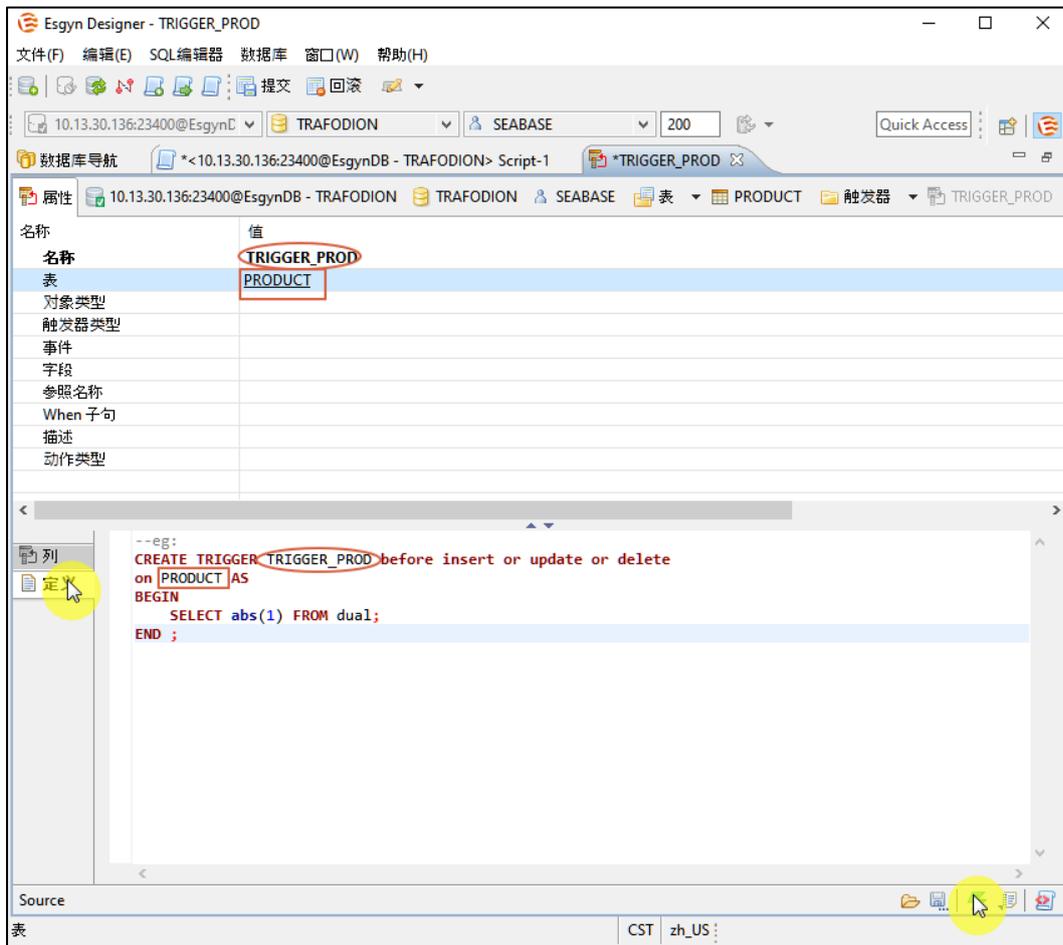


10. 触发器管理

2. 输入触发器名称，点击 OK。



3. 在新建窗口中，点击定义，输入创建触发器的 SQL，点击右下角编译触发器



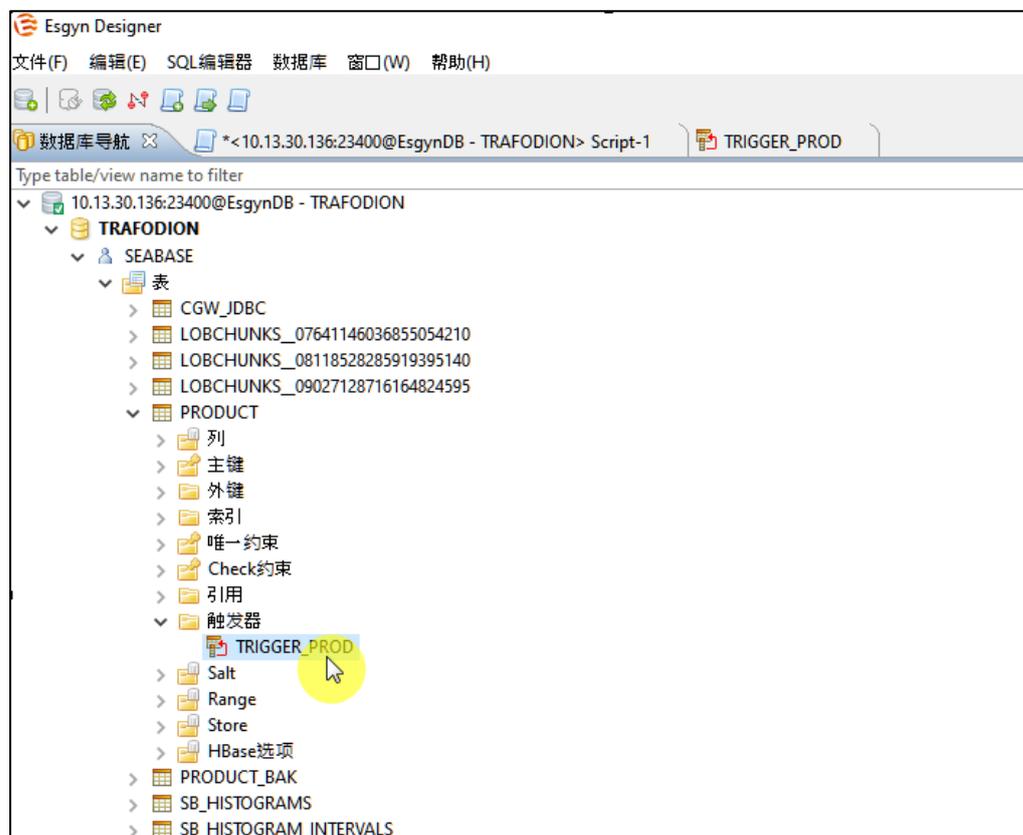


注意

- 触发器属性窗口中的名称必须与定义窗口中的 SQL 中定义的触发器名称一致, 表名称必须与语句中 **ON** 后面的表名一致。

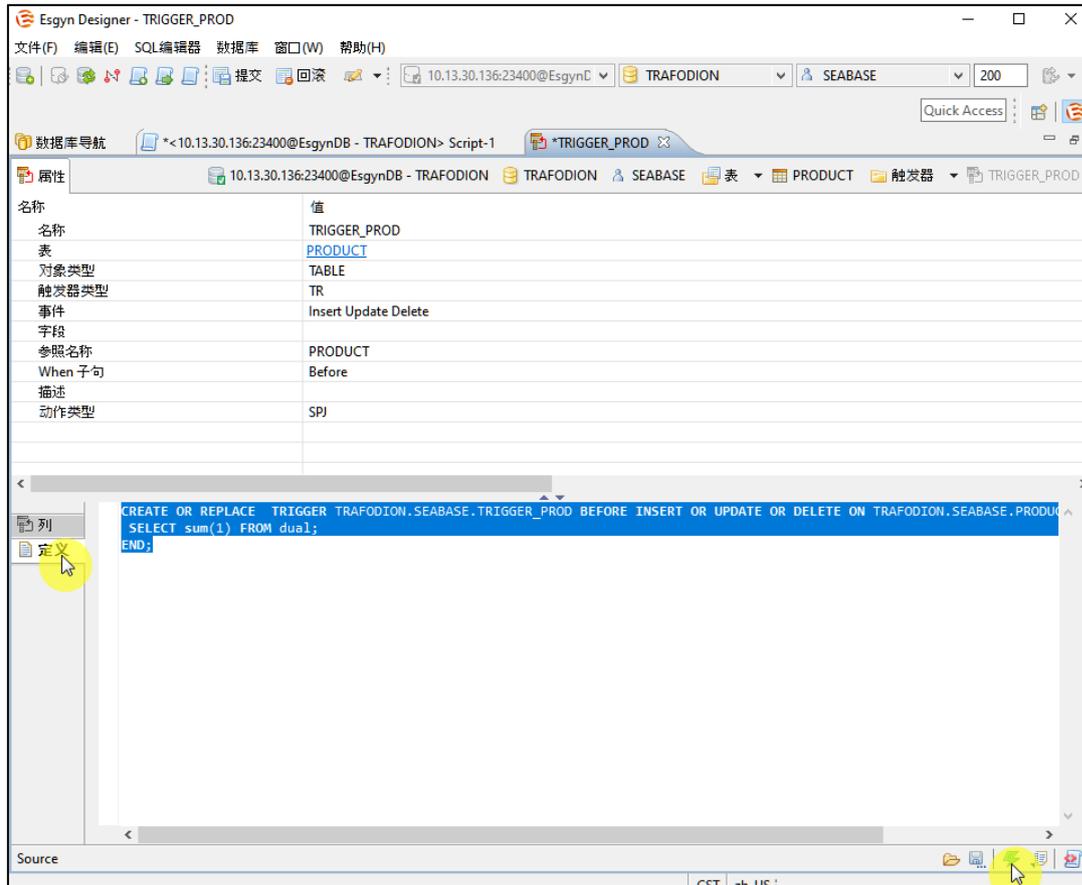
10.2 如何编辑触发器

1. 打开左侧数据库导航窗口,打开对应 schema 下对应表的触发器,双击选中的触发器。



2. 在属性窗口中点击定义,输入修改触发器的 SQL,点击右下角编译触发器

10. 触发器管理



10.3 如何删除触发器

您可以通过以下 2 种方法删除触发器：

- **方法 1**

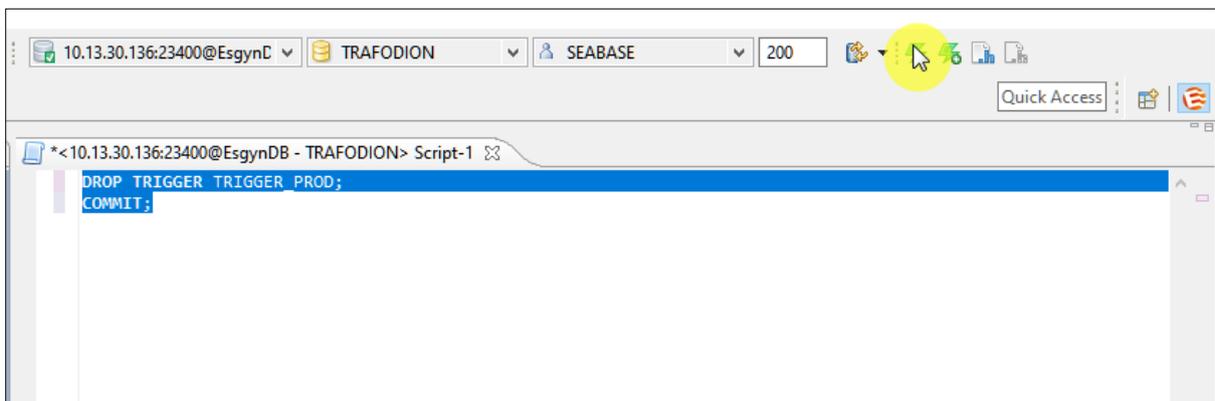
在 **SQL 编辑器** 中执行 SQL 语句删除触发器。更多信息，请参阅以下内容。

- **方法 2**

在 **数据库导航窗口** 中删除触发器。更多信息，请参阅以下内容。

10.3.1 方法 1：通过 SQL 语句进行删除

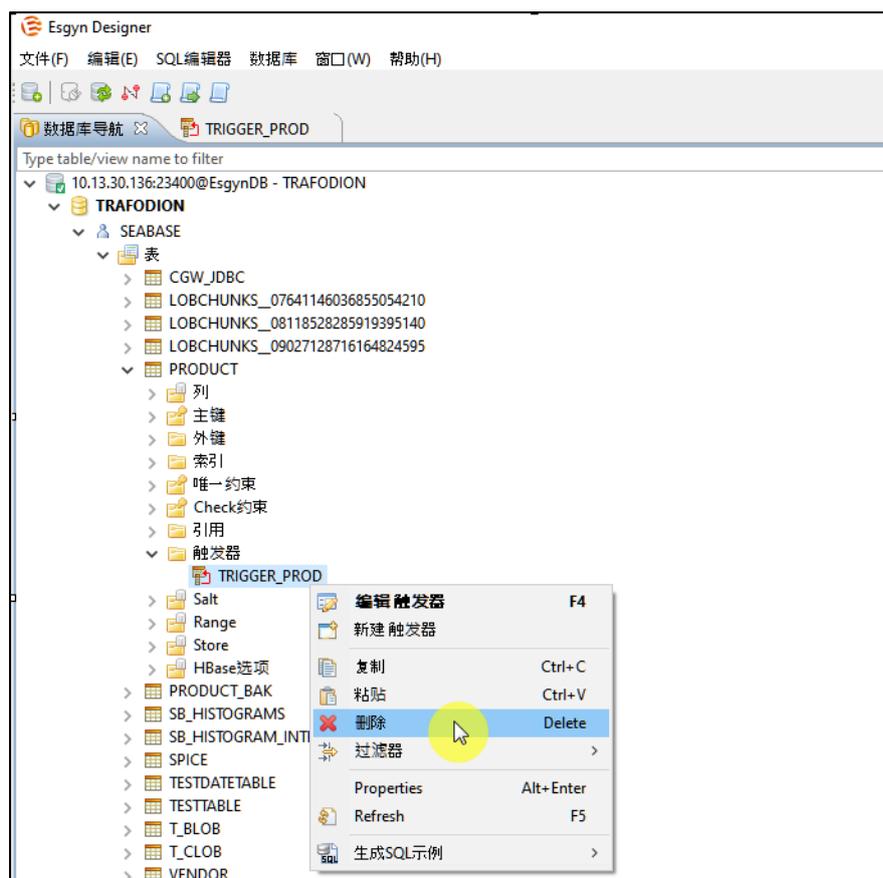
在编辑器中输入删除语句，执行调用。



10.3.2 方法 2：通过数据库导航窗口删除

1. 在 **数据库导航窗口** 中，右击触发器 TRIGGER_PROD。
2. 点击删除。

10. 触发器管理



提示

如需删除多个触发器，选中多个触发器的同时按下 Ctrl，再点击右键，点击删除。

11. 包管理

本章讲述以下内容:

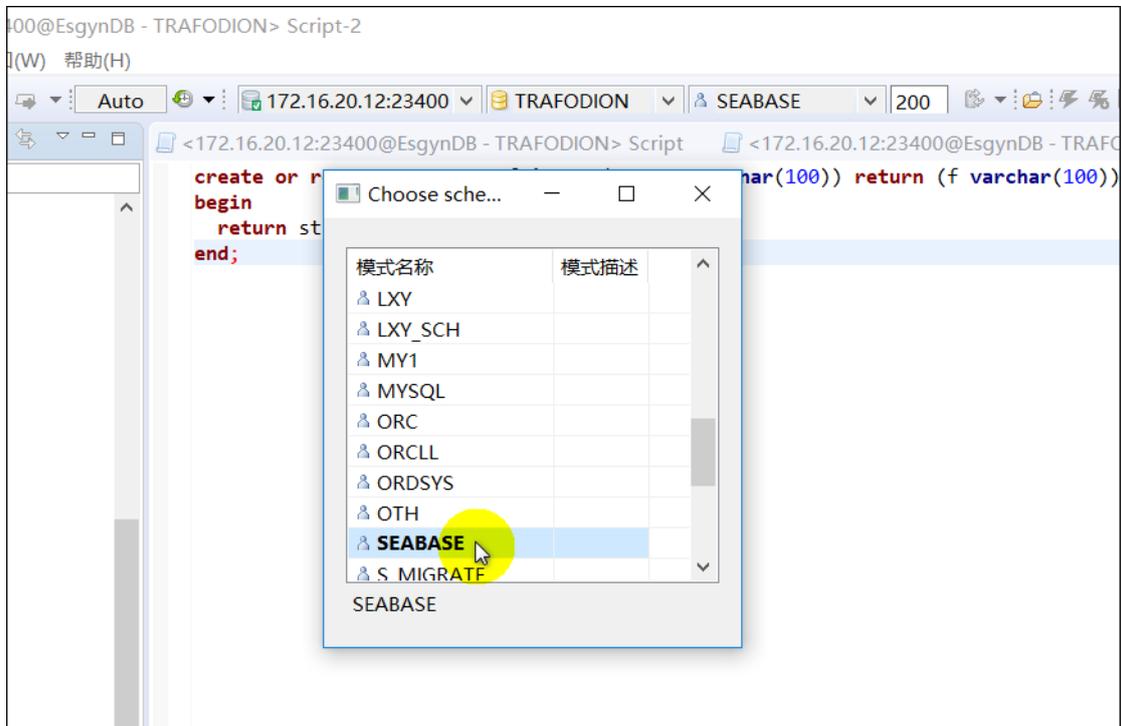
11.1 如何创建包

11.2 如何编辑包

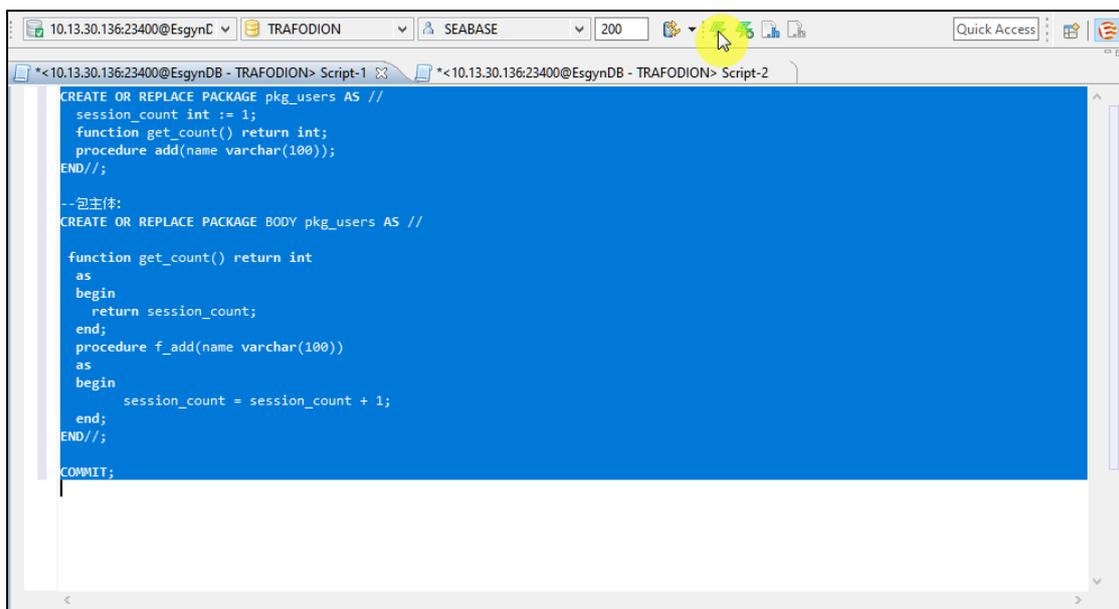
11.3 如何删除包

11.1 如何创建包

1. 打开编辑框窗口，选择对应的 schema。

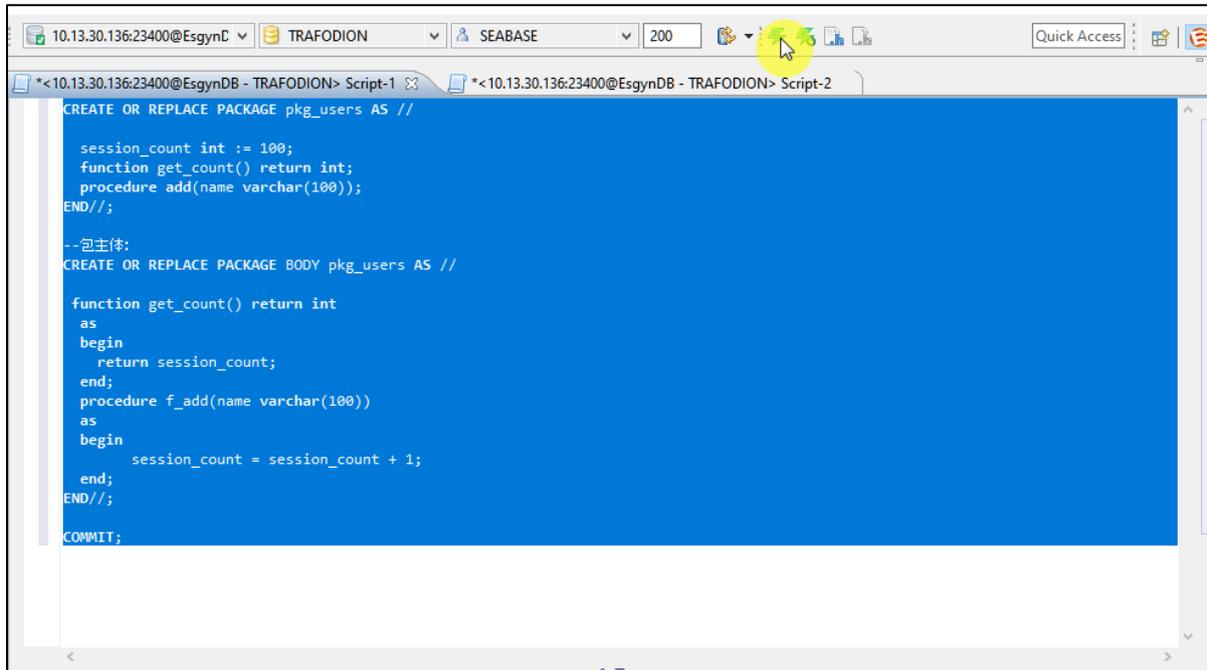


2. 输入创建包语句，点击执行。



11.2 如何编辑包

在编辑器中输入编辑语句，执行调用。



```
CREATE OR REPLACE PACKAGE pkg_users AS //  
  
    session_count int := 100;  
    function get_count() return int;  
    procedure add(name varchar(100));  
END//;  
  
--包主体:  
CREATE OR REPLACE PACKAGE BODY pkg_users AS //  
  
    function get_count() return int  
    as  
    begin  
        return session_count;  
    end;  
    procedure f_add(name varchar(100))  
    as  
    begin  
        session_count = session_count + 1;  
    end;  
END//;  
  
COMMIT;
```

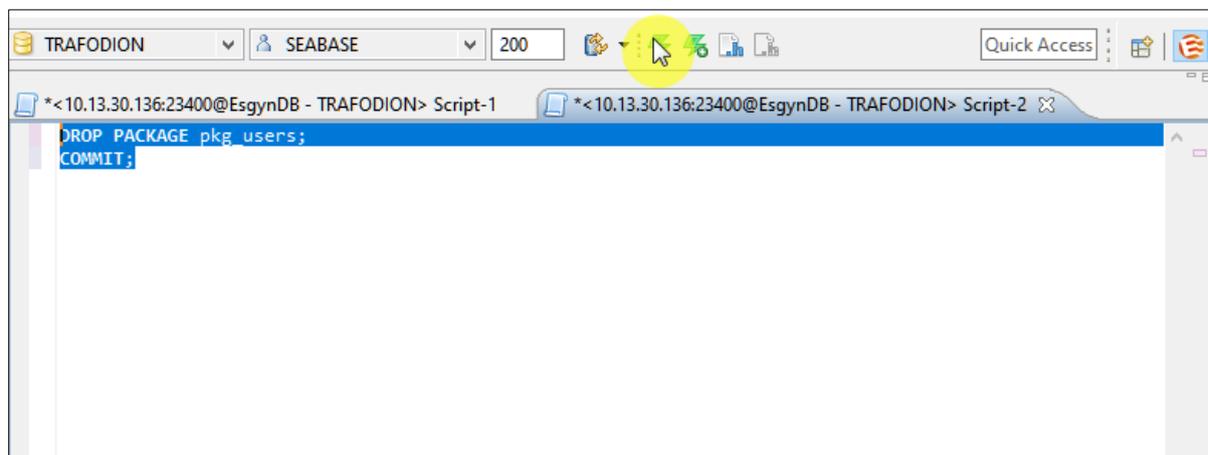


提示

创建和编辑包的语句与 SP SQL 手册略有不同，AS 后面需要跟"/"，END 后面需要跟"/;"，请注意查看以上用例！

11.3 如何删除包

在编辑器中输入删除语句，执行调用。



12. 事务

本章讲述以下内容:

12.1 提交模式

12.2 事务隔离等级

12.3 事务日志

12.1 提交模式

易鲸捷 Designer 支持自动提交模式和手动提交模式，这两种模式仅在 SQL 编辑器中有效。

12.1.1 自动提交模式

自动提交模式无需用户确认，修改立刻提交至数据库。

如需启用自动提交模式，在工具栏中，点击切换到自动提交 () 图标。

12.1.2 手动提交模式

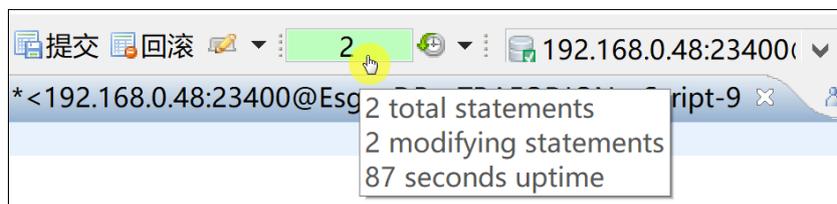
手动提交模式需要用户确认，再提交修改至数据库或回滚修改。

如需启用手动提交模式，在工具栏中，点击切换到手动提交 () 图标。

如需提交或回滚修改，在工具栏中，点击提交 () 或回滚 () 图标。

在该模式下执行 SQL 语句后：

- 如需查看统计信息，将鼠标移至提交图标的右边的方框，将显示已执行的 SQL 语句的数量和时间。

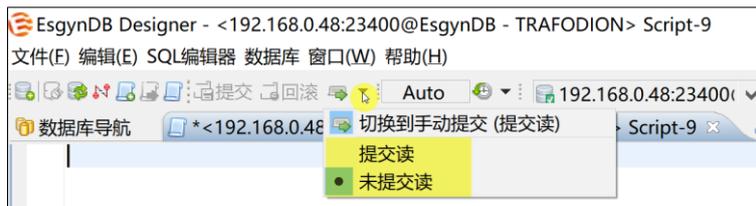


- 如需查看事务日志，点击方框里的数字。

12.2 事务隔离等级

自动提交模式和手动提交模式都能设置以下事务隔离等级：

- 提交读
- 未提交读



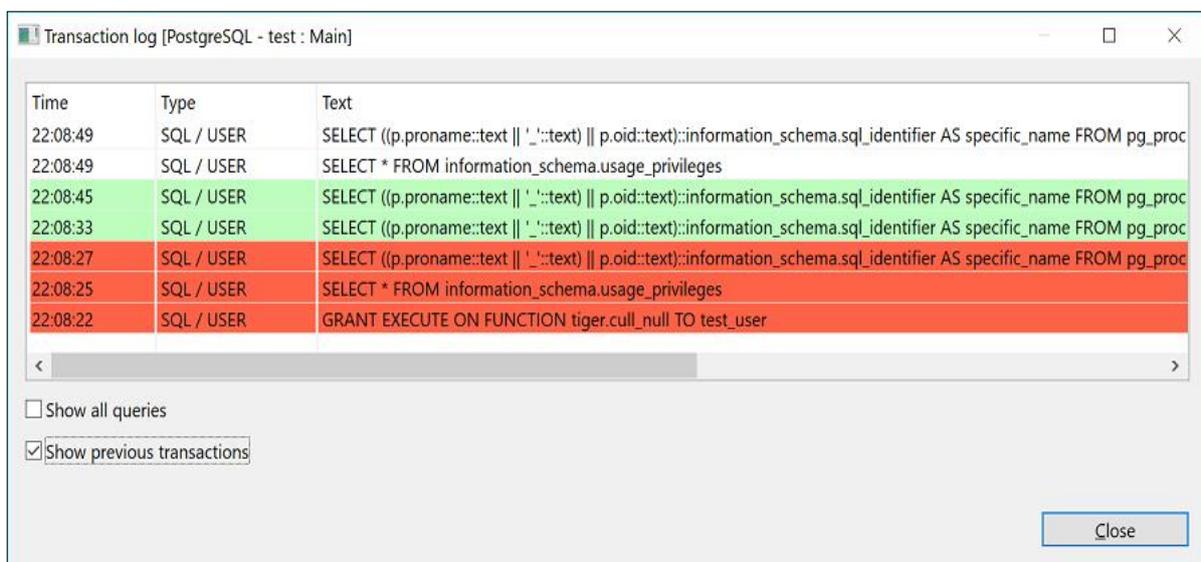
12.3 事务日志

事务日志显示当前会话的所有事务（例如，增删改查）。

如需查看事务日志，在工具栏中，点击事务日志（）图标。

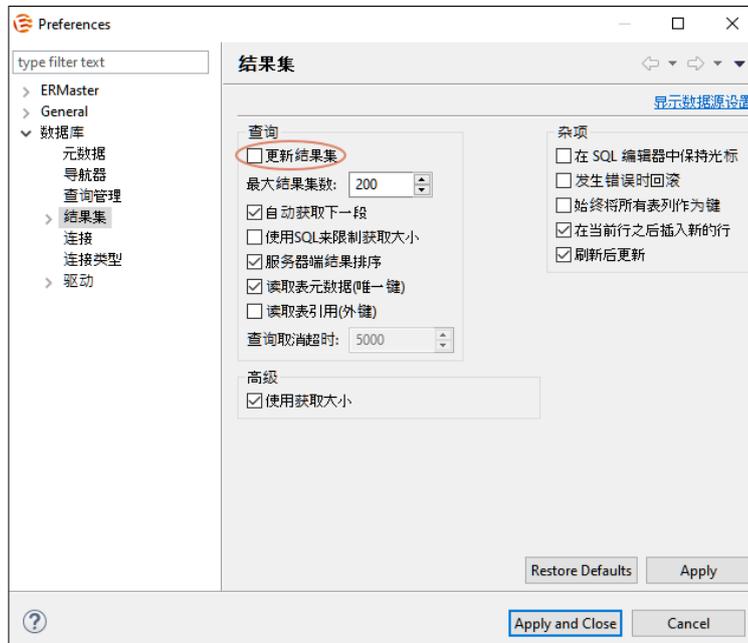
- 白色：正在执行或待处理（pending）的事务。
- 绿色：成功提交的事务。
- 红色：回滚的事务。

如需查看所有查询和所有事务，在对话框中勾选相应选项。



13. 注意事项

1. 查询系统 schema "_MD_" 下面的元数据表时，请不要勾选全局设置中的更新结果集的控件，否则会报如下错误：



```
org.esgyn.editor.model.exec.DBCEException: SQL 错误: ***
ERROR[4003] Column X.SYSKEY is not a column in table X, or,
after a NATURAL JOIN or JOIN USING, is no longer allowed to be
specified with a table correlation name. [2019-11-05 10:08:56]
*** ERROR[8822] The statement was not prepared. [2019-11-05
10:08:56]
    at
org.esgyn.editor.model.impl.jdbc.exec.JDBCStatementImpl.execute
Statement(JDBCStatementImpl.java:145)
    at
org.esgyn.editor.model.impl.jdbc.struct.JDBCTable.readData(JDB
CTable.java:227)
    at
org.esgyn.editor.ui.controls.resultset.ResultSetJobDataRead.ru
n(ResultSetJobDataRead.java:92)
    at
```

13. 注意事项

```
org.esgyn.editor.model.runtime.AbstractJob.run(AbstractJob.java:95)
```

```
at
```

```
org.eclipse.core.internal.jobs.Worker.run(Worker.java:56)
```

附录

附录 1. 数据类型映射规则

T 代表 TRUE, F 代表 FALSE。

	CHAR	VARCHAR	NUMERIC	SMALLINT	INT	LARGEINT	FLOAT	REAL	DOUBLE PRECISION	DECIMAL	DATE	TIME	TIME(P)	TIMESTAMP	TIMESTAMP(P)	BLOB	CLOB
CHAR		T	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
VARCHAR	T		F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
NUMERIC	F	F		T	T	T	T	T	T	T	F	F	F	F	F	F	F
SMALLINT	F	F	T		T	T	T	T	T	T	F	F	F	F	F	F	F
INT	F	F	T	T		T	T	T	T	T	F	F	F	F	F	F	F
LARGEINT	F	F	T	T	T		T	T	T	T	F	F	F	F	F	F	F
FLOAT	F	F	T	T	T	T		T	T	T	F	F	F	F	F	F	F
REAL	F	F	T	T	T	T	T		T	T	F	F	F	F	F	F	F
DOUBLE PRECISION	F	F	T	T	T	T	T	T		T	F	F	F	F	F	F	F
DECIMAL	F	F	T	T	T	T	T	T	T		F	F	F	F	F	F	F
DATE	F	F	F	F	F	F	F	F	F	F		F	F	F	F	F	F
TIME	F	F	F	F	F	F	F	F	F	F	F		T	F	F	F	F
TIME(P) ¹⁴	F	F	F	F	F	F	F	F	F	F	F	T		F	F	F	F
TIMESTAMP	F	F	F	F	F	F	F	F	F	F	F	F	F		T	F	F
TIMESTAMP(P)	F	F	F	F	F	F	F	F	F	F	F	F	F	T		F	F
BLOB	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F		F
CLOB	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	

¹⁴ TIME(P)是 TIME(PRECISION)的缩写, 其它类似情况同理。

附录 2. HBase 选项及其有效值和默认值



提示

更多关于如何在建表时指定 HBase 选项的信息，请参阅《EsgynDB SQL 参考手册》。

以下是 HBase 选项及其有效值和默认值。

HBase 选项	有效值	默认值
BLOCKCACHE	<ul style="list-style-type: none"> 'true' 'false' 	'true'
BLOCKSIZE	'1024' - '64*1024*1024'	'64*1024'
BLOOMFILTER	<ul style="list-style-type: none"> 'NONE' 'ROW' 'ROWCOL' 	'ROW'
CACHE_BLOOMS_ON_WRITE	<ul style="list-style-type: none"> 'true' 'false' 	'false'
CACHE_DATA_ON_WRITE	<ul style="list-style-type: none"> 'true' 'false' 	'false'
CACHE_INDEXES_ON_WRITE	<ul style="list-style-type: none"> 'true' 'false' 	'false'
COMPACT	<ul style="list-style-type: none"> 'true' 'false' 	'true'
COMPACT_COMPRESSION	<ul style="list-style-type: none"> 'GZ' 'LZ4' 'LZO' 'NONE' 'SNAPPY' 	'NONE'

(续前表)

HBase 选项	有效值	默认值
COMPRESSION	<ul style="list-style-type: none"> • 'GZ' • 'LZ4' • 'LZO' • 'NONE' • 'SNAPPY' 	'NONE'
DATA_BLOCK_ENCODING	<ul style="list-style-type: none"> • 'DIFF' • 'FAST_DIFF' • 'NONE' • 'PREFIX' 	'NONE'
DURABILITY	<ul style="list-style-type: none"> • 'USE_DEFAULT' • 'SKIP_WAL' • 'ASYNC_WAL' • 'SYNC_WAL' • 'FSYNC_WAL' 	'SYNC_WAL'
EVICT_BLOCKS_ON_CLOSE	<ul style="list-style-type: none"> • 'true' • 'false' 	'false'
IN_MEMORY	<ul style="list-style-type: none"> • 'true' • 'false' 	'false'
KEEP_DELETED_CELLS	<ul style="list-style-type: none"> • 'true' • 'false' 	'false'
MAX_FILESIZE	'2*1024*1024' - '263-1'	'10*1024*1024*1024'
MAX_VERSIONS	<ul style="list-style-type: none"> • '1' • '正整数' 	'1'
MEMSTORE_FLUSH_SIZE	'1024*1024' - '263-1'	'128*1024*1024'

(续前表)

HBase 选项	有效值	默认值
MIN_VERSIONS	<ul style="list-style-type: none"> '0' '正整数' 	'0'
PREFIX_LENGTH_KEY	<p>'正整数'</p> <p>如果 SPLIT_POLICY 为 KeyPrefixRegionSplitPolicy, 则该值生效。该值应小于表的 key 的最大长度。</p>	'2'
REPLICATION_SCOPE	<ul style="list-style-type: none"> '0' '1' 	'1'
SPLIT_POLICY	<ul style="list-style-type: none"> 'org.apache.hadoop.hbase.regionserver.ConstantSizeRegionSplitPolicy' 'org.apache.hadoop.hbase.regionserver.IncreasingToUpperBoundRegionSplitPolicy' 'org.apache.hadoop.hbase.regionserver.KeyPrefixRegionSplitPolicy' 	'org.apache.hadoop.hbase.regionserver.IncreasingToUpperBoundRegionSplitPolicy'
TTL	'正整数', 该值应小于或等于 2147483447。	'2147483647'