

ORACLE使用

by 2251551 程鸿磊

Oracle主要用到的3个工具：sqlplus(命令行工具)，SQL developer(图形界面工具)和 Database Configuration Assistant(配置工具)

SQL*Plus

官方文档链接：https://docs.oracle.com/cd/B19306_01/server.102/b14357/qstart.htm#i1049125

什么是SQL*Plus？

SQLPlus 是Oracle提供的命令行工具，使用SQLPlus可以在命令行交互执行建表、查询等语句，其是Oracle的即时客户端（instant client）的一个功能。

什么情况需要SQL*Plus？

对于开发人员和数据维护人员而言，最常使用的Oracle的数据操作工具是SQL Developer，SQL Developer号称是Oracle的集成开发环境，类似Java中的Eclipse，微软系列的Visual Studio。SQL Developer的功能很强大，常用的功能有：

- 运行SQL语句和脚本
- 查询各种类型的数据库对象
- 编辑和调试PL/SQL语句（提供格式化SQL语句的功能）
- 执行和产生报表

SQL Developer需要启动后使用，在持续集成（CI）的使用中，比如在应用部署过程中自动化执行SQL脚本的状况上就不适合了，这个时候需要命令行的方式执行SQL语句，这就可以选择使用SQLPlus了，此外Oracle也提供了一个SQL Developer的命令行的版本：SQLcl，使用SQLcl，也可以达成SQLPlus类似功能。

SQL*Plus的安装

SQL*Plus 不能单独安装运行，需要在instant client基础上运行。

1. 官网下载两个安装包
-

```
1 instantclient-basic-windows.x64.zip
2 instantclient-sqlplus-windows.x64.zip
```

2. 解压到同一个目录下

使用解压工具（这里使用WinRAR），对两个压缩包点击“解压到当前文件夹”，如果这两个压缩包中同一个目录的话，解压后的会是同一个目录，这里是instantclient_19_6；如果压缩包不在同一个目录，也可以手动复制。

3. 测试运行sqlplus

找到sqlplus.exe点击即可运行 或者 命令行中切换到exe所在文件夹，窗口中输入sqlplus /NOLOG 即可启用

4. 配置环境变量

使用sqlplus.exe的完整路径或是切换到sqlplus.exe路径下运行sqlplus会比较繁琐，**将这个路径配置到系统的环境变量中，则在任何路径下都可以直接执行sqlplus。**

可以直接将这个路径添加到PATH中，但是较好的方式是**定义一个ORACLE_HOME的环境变量，再添加进PATH。**

控制面板中搜索环境变量，点击环境变量和新建，环境变量的取值为exe所在文件夹的路径，**创建了环境变量后，一定要在Path中再添加一项（如下）！！**

```
1 %ORACLE_HOME%    中间为路径的变量名
```

5. 添加字符集变量

依次点击 “计算机-> 右键 -> 属性 -> 高级系统设置 -> 环境变量 -> 新建用户变量 -> 变量名：NLS_LANG 变量值：SIMPLIFIED CHINESE_CHINA.ZHS16GBK 或者 AMERICAN_AMERICA.UTF8（根据你的数据库服务器字符集赋值）-> 依次点击确定” 即可；

NLS_LANG 用于指定 Oracle 客户端和服务端之间的字符集、语言和地区设置。它的格式为 **<LANGUAGE>_<TERRITORY>.<CHARACTERSET>**，在这里：

- **LANGUAGE** 是语言，**AMERICAN** 代表英语（美国）。
- **TERRITORY** 是地区，**AMERICA** 代表美国。
- **CHARACTERSET** 是字符集，**UTF8** 代表 UTF-8 编码。

通过设置这个变量，可以确保 Oracle 客户端正确处理字符集和语言区域信息。

注：客户端字符集必须和数据库端一致，否则可能出现乱码。

6. 在cmd输入sqlplus命令即可使用

- 连接：sqlplus 用户名/密码@数据库主机IP；如果成功则CMD会显示 `SQL>_`

本项目：sqlplus sys/tongjiSSEDBMS2024@120.26.138.61:1521/XE AS SYSDBA

```
命令提示符 - sqlplus sys/tongjiSSEDBMS2024@120.26.138.61:1521/XE AS SYSDBA
Microsoft Windows [版本 10.0.22631.3737]
(c) Microsoft Corporation。保留所有权利。

C:\Users\lenovo> sqlplus sys/tongjiSSEDBMS2024@120.26.138.61:1521/XE AS SYSDBA

SQL*Plus: Release 23.0.0.0.0 - Production on Tue Jul 9 14:31:56 2024
Version 23.4.0.24.05

Copyright (c) 1982, 2024, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL>
```

ps：建表的话就先以最高权限的角色连接的

sqlplus具体用法

与课本SQL语法基本一致

以下命令大小写均可，[]部分内容输入命令时可以省略

1. 数据库连接和会话管理

- 启动

配置好环境变量后，在命令行输入命令即可

A. sqlplus /nolog (只是启动sqlplus而不连接数据库，使用nolog参数)

数据库实例启动成功(没有关闭且一直在运行)后，可以使用其他普通用户登陆连接到数据库了

B. sqlplus username/password@hostname:port/SID (一般使用，在启动sqlplus的同时连接到数据库)，具体含义如下

sys/tongjiSSEDBMS2024@120.26.138.61:1521/XE AS SYSDBA :这部分是连接字符串，包含了用户名、密码、主机名、端口号和服务名称。

- sys :用户名，即 SYS 用户。
- tongjiSSEDBMS2024 :密码，即 SYS 用户的密码。
- 120.26.138.61 :数据库服务器的主机名或 IP 地址。
- 1521 : Oracle 数据库的监听端口号。
- XE : Oracle 数据库的服务名称。
- AS SYSDBA :指定连接时以 SYSDBA 权限进行连接，通常用于高级管理操作。

- 退出

输入EXIT命令或QUIT命令，退出sqlplus工具

- 重建连接与断开连接

CONN[ECT] 用户名/密码

CONN[ECT]命令先断开当前连接，然后建立新的连接

DISC[ONNECT]

该命令的作用是断开与数据库的连接，但不退出SQL*Plus环境

2. SQL 命令

- 查询数据

```
1 SELECT * FROM table_name;
```

- 插入数据

```
1 INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

- 更新数据

```
1 UPDATE table_name SET column1 = value1 WHERE condition;
```

- 删除数据

```
1 DELETE FROM table_name WHERE condition;
```

3. 数据库对象管理：表的创建，删除，修改，查看

- 创建

```
1 CREATE TABLE table_name (  
2     column1 datatype,  
3     column2 datatype,  
4     ...  
5 );
```

- 删除

```
1 DROP TABLE table_name;
```

- 修改

```
1 ALTER TABLE table_name ADD column_name datatype;
```

- 显示数据库对象结构

```
1 DESCRIBE table_name
```

4. PL/SQL 块

- 匿名块

```
1 BEGIN  
2     -- PL/SQL 代码  
3 END;
```

- 创建存储过程

```
1 CREATE OR REPLACE PROCEDURE procedure_name AS  
2 BEGIN  
3     -- PL/SQL 代码  
4 END procedure_name;
```

- 创建函数

```
1 CREATE OR REPLACE FUNCTION function_name RETURN return_datatype AS  
2 BEGIN  
3     -- PL/SQL 代码  
4 END function_name;
```

5. 用户管理

- 创建用户

```
1 CREATE USER username IDENTIFIED BY password;
```

- 删除用户

```
1 DROP USER username CASCADE;
```

- 授予权限

```
1 GRANT privilege TO username;
```

- 撤销权限

```
1 REVOKE privilege FROM username;
```

6. 其他常用命令

- 显示当前用户

```
1 SHOW USER;
```

- 列出数据库中的对象

```
1 SELECT * FROM user_tables;
```

- 显示当前会话信息

```
1 SHOW SGA;
```

- ```
1 SPPOOL output.txt;
2 -- SQL 命令
3 SPPOOL OFF;
```

### 1. 在VS中安装ASP.NET框架并创建一个新的项目





## ASP. NET Core Web 应用 与 ASP. NET Core Web API 区别

**ASP. NET Core Web 应用：**用于创建具有丰富UI的Web应用，关注视图和页面的渲染。

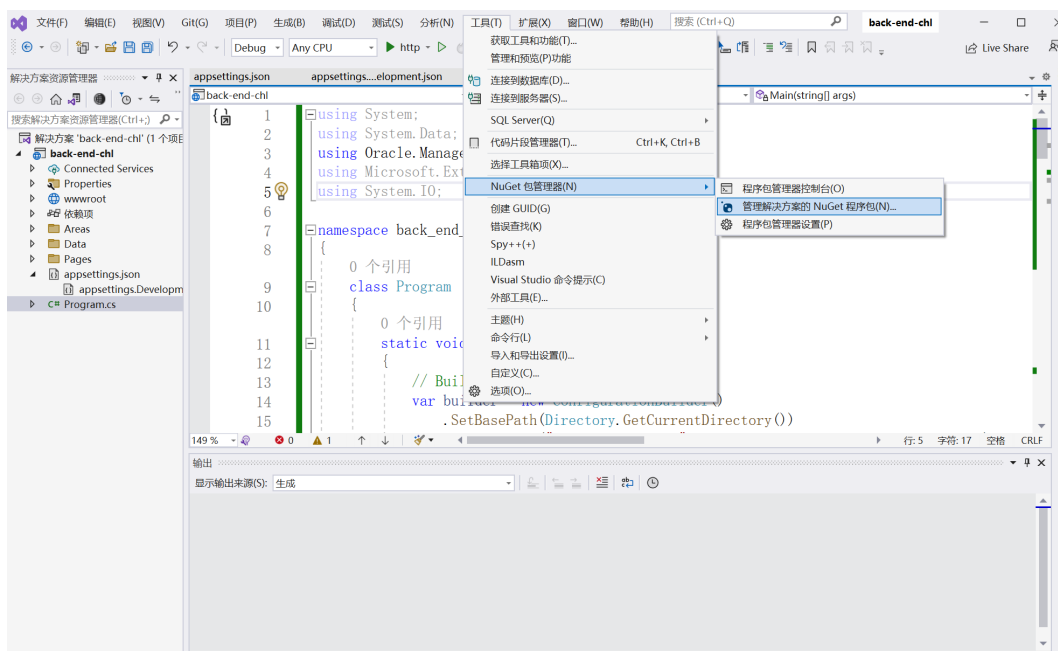
**ASP. NET Core Web API：**用于创建数据服务接口，关注数据的传递和处理。

以下摘自官方文档对选择模板时的建议：[官方文档链接](#)

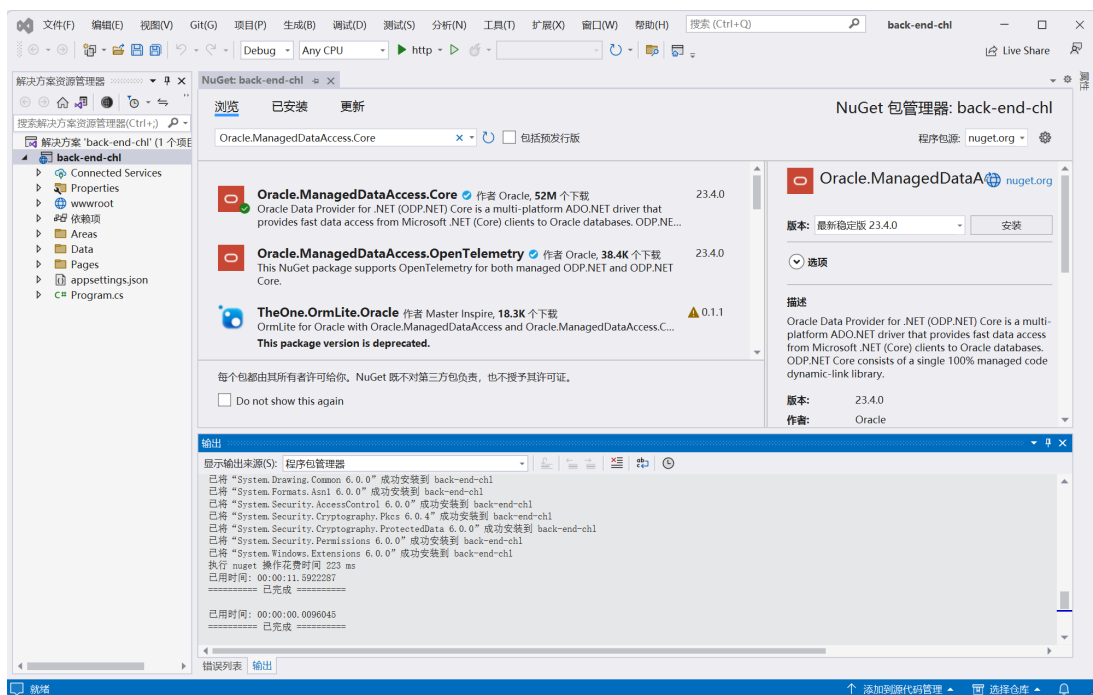
在创建新的 ASP. NET Core 应用时，应考虑好要构建的应用类型。在 IDE 中或使用 `dotnet new` CLI 命令创建新项目时，可以从多个模板中进行选择。最常见的项目模板包括空、Web API、Web 应用和 Web 应用（模型-视图-控制器）。虽然只能在首次创建项目时做出此决定，但此决定可以撤销。Web API 项目使用标准的“模型 - 视图 - 控制器”控制器（默认情况下，它只缺少视图）。同样，默认的 Web 应用模板使用 Razor Pages，因此也缺少 Views 文件夹。可以稍后向这些项目添加 Views 文件夹以支持基于视图的行为。默认情况下，Web API 和模型 - 视图 - 控制器项目不包含 Pages 文件夹，但可以稍后添加一个以支持基于 Razor Pages 的行为。可以将这三个模板视为支持三种不同类型的默认用户交互：数据 (Web API)、基于页面和基于视图。但是，如果愿意，可以在单个项目中混合和匹配任何或所有这些模板。

## 2. 安装ODP.NetCore 和 Oracle.ManagedDataAccess.Core

PS：对于每一个新创建的项目都要重新安装属于这个项目的安装包（各项目间不通用）







## ODP.NetCore 和 Oracle.ManagedDataAccess.Core

**Oracle Data Provider for .NET Core (ODP.NetCore)** 是 Oracle 提供的用于 .NET Core 平台的数据库访问库。它允许 .NET Core 应用程序与 Oracle 数据库进行高效通信。

**Oracle.ManagedDataAccess.Core** 是一个完全托管的 .NET Core 数据访问库。它不需要 Oracle 客户端软件，所有操作都在托管代码中完成。

### 主要特性包括：

- a. **简化部署**：不需要安装 Oracle 客户端软件，简化了部署过程。
- b. **跨平台支持**：支持在 Windows、Linux 和 macOS 上运行。
- c. **高级功能**：支持大多数 Oracle 特有的功能，如批量操作、事务、存储过程、LOB 处理等。
- d. **一致性**：提供与 Oracle.ManagedDataAccess（用于 .NET Framework）的 API 一致性，方便从 .NET Framework 向 .NET Core 迁移。

## 3. 更新 `appsettings.json` 文件

主要需要更改连接字符串（第2行）

```
1 {
2 "ConnectionStrings": {
3 "OracleDbContext": "User Id=sys;Password=tongjiSSEDBMS2024;Data
4 Source=120.26.138.61:1521/XE;DBA Privilege=SYSDBA"
5 },
6 "Logging": {
7 "LogLevel": {
```

```

7 "Default": "Information",
8 "Microsoft.AspNetCore": "Warning"
9 }
10 },
11 "AllowedHosts": "*"
12 }

```

#### 4. 测试代码

修改program.cs代码

```

1 using System;
2 using System.Data;
3 using Oracle.ManagedDataAccess.Client;
4 using Microsoft.Extensions.Configuration;
5 using System.IO;
6
7 namespace back_end_zwc
8 {
9 class Program
10 {
11 static void Main(string[] args)
12 {
13 // Build configuration
14 var builder = new ConfigurationBuilder()
15 .SetBasePath(Directory.GetCurrentDirectory())
16 .AddJsonFile("appsettings.json", optional: true,
17 reloadOnChange: true);
18
19 IConfiguration configuration = builder.Build();
20
21 // Get connection string
22 string connectionString =
23 configuration.GetConnectionString("OracleDbContext");
24
25 // Connect to Oracle database and run query
26 using (OracleConnection conn = new
27 OracleConnection(connectionString))
28 {
29 try
30 {
31 conn.Open();
32 Console.WriteLine("Connection established.");
33
34 string sql = "SELECT * FROM YOUR_TABLE_NAME"; // 替换为实际的
35 表名

```

```
32 OracleCommand cmd = new OracleCommand(sql, conn);
33
34 OracleDataReader reader = cmd.ExecuteReader();
35
36 while (reader.Read())
37 {
38 Console.WriteLine(reader.GetString(0)); // 根据实际的列数
据类型和索引进行调整
39 }
40
41 reader.Close();
42 }
43 catch (Exception ex)
44 {
45 Console.WriteLine("Error: " + ex.Message);
46 }
47 finally
48 {
49 conn.Close();
50 }
51 }
52 }
53 }
54 }
```