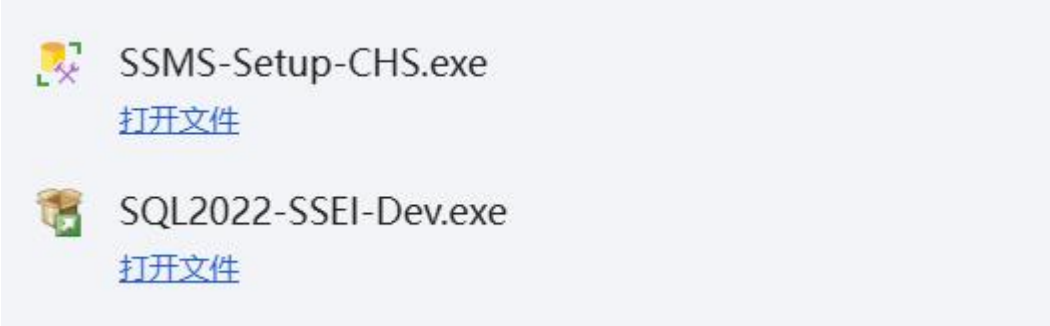
数据库实验-交互式 SQL

一、实验环境

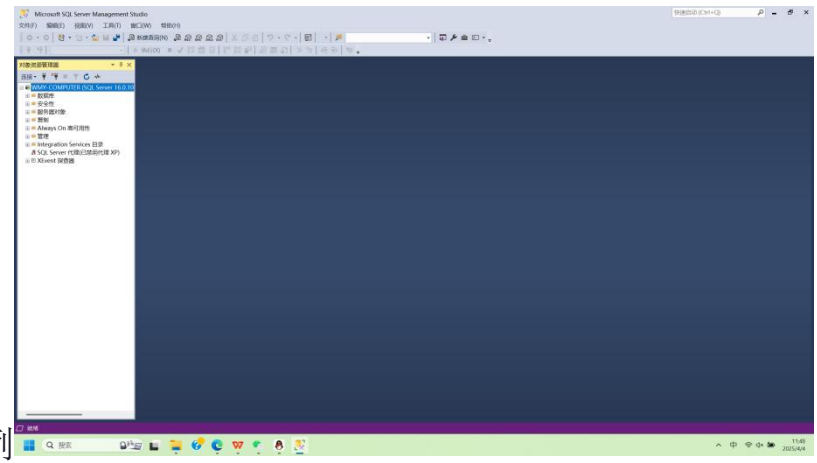
下载 sql server 和 sql server management studio 的安装包



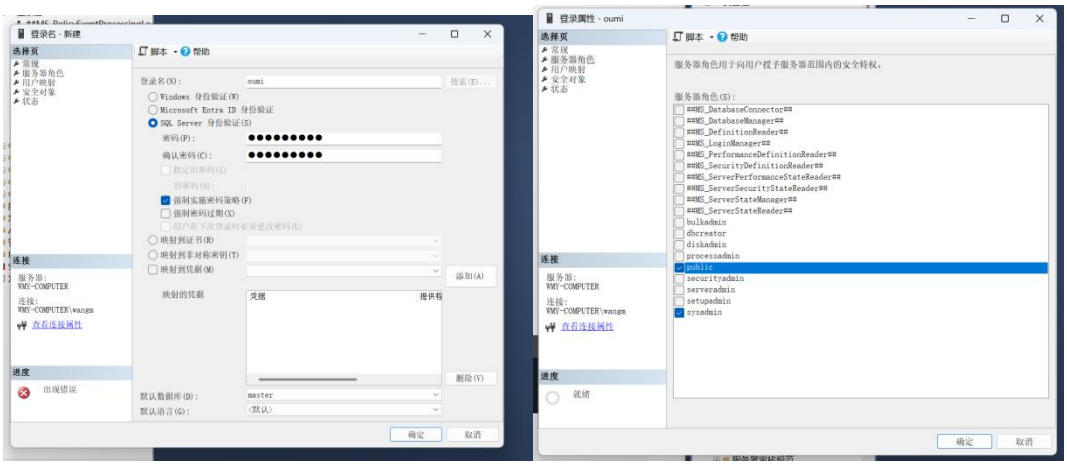
下载好软件

利用 Management Studio 创建一个用户，赋予 DBA 权限（sysadmin 角色）

1、打开 SQL Server Management Studio (SSMS)，并连接到相应的 SQL Server 实例



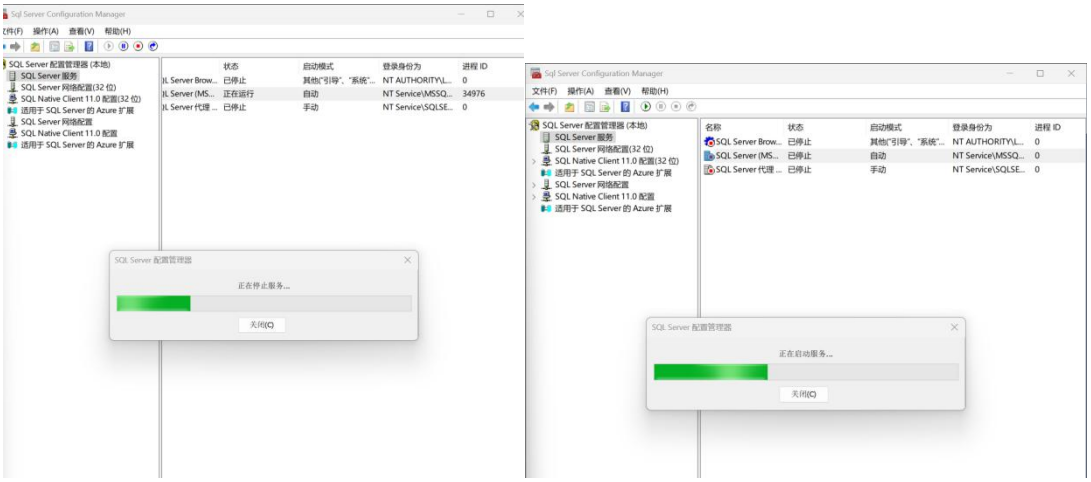
2、新建登陆名 为 oumi，并设置 sysadmin 角色



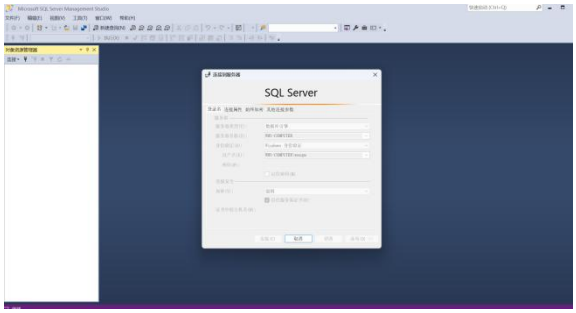
服务的启动和停止

利用 Configuration Manager 启动和停止数据库服务，并验证效果

停止服务；启动服务；



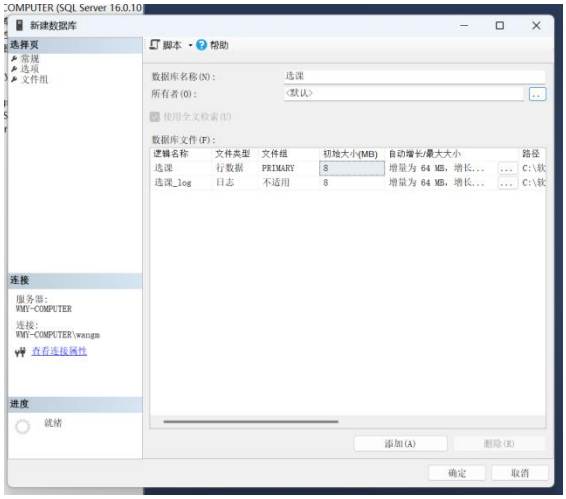
停止服务后，ssms 连接不上，，，验证成功



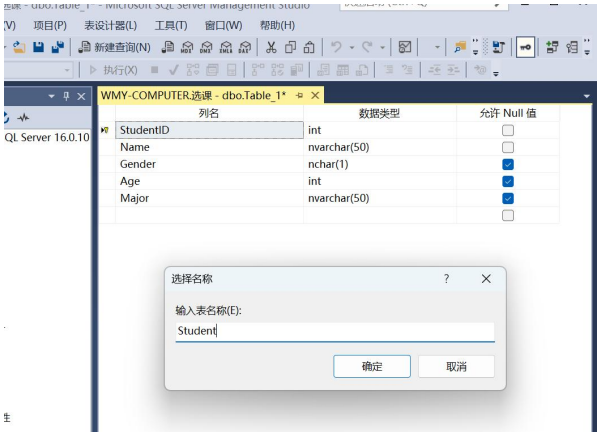
上面是开启服务的时候查询，下面是关闭了之后查询

```
C:\Users\wangm>sqlcmd -S localhost -Q "SELECT @@VERSION"


------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
Microsoft SQL Server 2022 (RTM) - 16.0.1000.6 (X64)
        Oct  8 2022 05:58:25
        Copyright (C) 2022 Microsoft Corporation
        Developer Edition (64-bit) on Windows 10 Home China 10.0 <X64> (Build 26100: ) (Hypervisor)


(1 行受影响)

C:\Users\wangm>sqlcmd -S localhost -Q "SELECT @@VERSION"
Sqlcmd: 错误: Microsoft ODBC Driver 17 for SQL Server : Named Pipes Provider: Could not open a connection to SQL Server
[2]. 。
Sqlcmd: 错误: Microsoft ODBC Driver 17 for SQL Server : Login timeout expired。
Sqlcmd: 错误: Microsoft ODBC Driver 17 for SQL Server : A network-related or instance-specific error has occurred while
establishing a connection to SQL Server. Server is not found or not accessible. Check if instance name is correct and if
 SQL Server is configured to allow remote connections. For more information see SQL Server Books Online.。
```

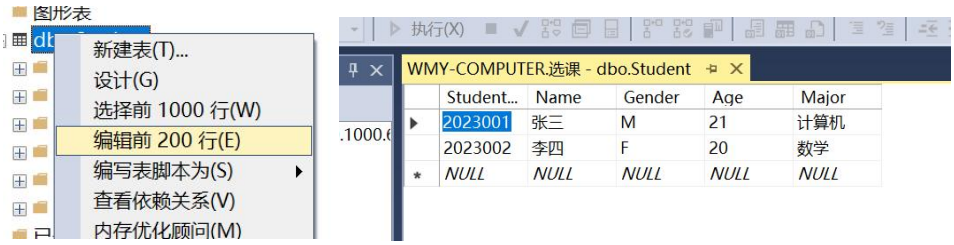
创建 选课 数据库：



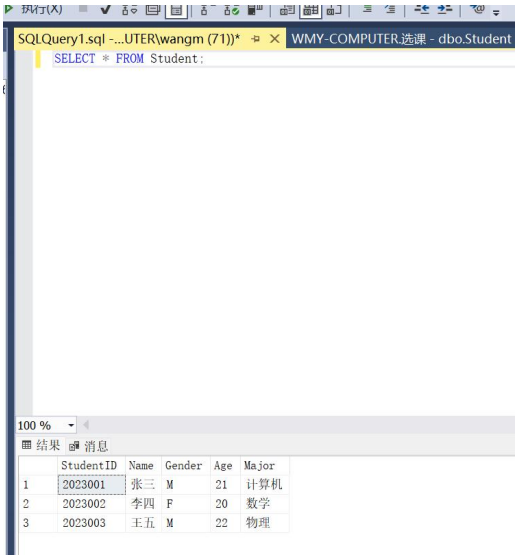
创建 Student 表：



向表插入数据：



查询验证：

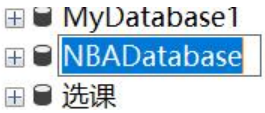


至此我们的实验环境已搭建完成。

二、实验内容与完成情况

1）数据定义：

首先新建数据库：NBADatabase



1、基本表的创建：

运行：

```
-- 1. 创建球队表(Teams)

CREATE TABLE Teams (
```

```
    TeamID INT PRIMARY KEY,
    TeamName VARCHAR(50) NOT NULL,
    City VARCHAR(50) NOT NULL,
    Conference VARCHAR(10) CHECK (Conference IN ('East', 'West')),
    Division VARCHAR(20),
    FoundedYear INT,
    HomeArena VARCHAR(100));

-- 2. 创建球员表(Players)
CREATE TABLE Players (
    PlayerID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    BirthDate DATE,
    Height DECIMAL(3,2), -- 单位：米
    Weight INT, -- 单位：公斤
    Position VARCHAR(20),
    JerseyNumber INT,
    TeamID INT,
    DraftYear INT,
    Country VARCHAR(50));
```

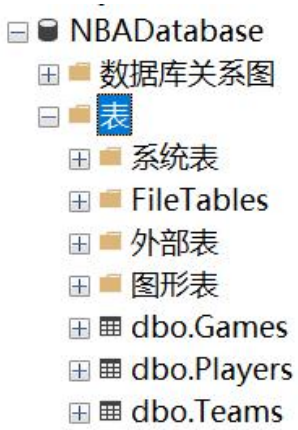
```
-- 3. 创建比赛表(Games)
CREATE TABLE Games (
    GameID INT PRIMARY KEY,
    GameDate DATE NOT NULL,
    HomeTeamID INT NOT NULL FOREIGN KEY REFERENCES Teams(TeamID),
    AwayTeamID INT NOT NULL FOREIGN KEY REFERENCES Teams(TeamID),
    HomeScore INT,
    AwayScore INT,
    Venue VARCHAR(100),
    Season VARCHAR(10));
```

创建完成后，NBADatabase 数据库具有

Teams（球队表），

Players（球员表），

Games（比赛表），



2、基本表的修改：

运行：

```
-- 修改 Players 表，添加 IsActive 字段（是否现役）
ALTER TABLE Players
ADD IsActive BIT DEFAULT 1;
```

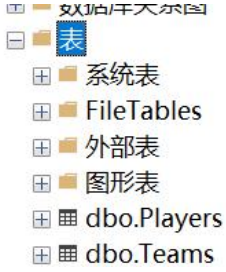
查看 Players 表：

已经包括 *IsActive 字段*

3、基本表的删除：

运行：

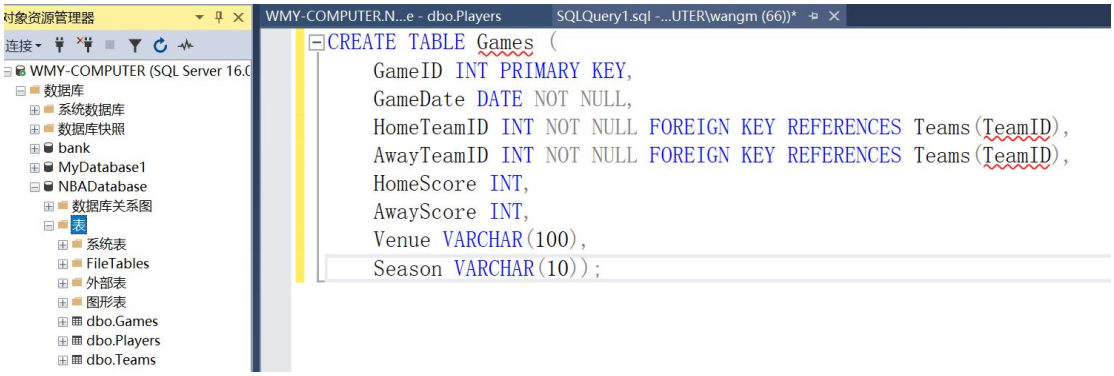
```
-- 删除 Games 表
DROP TABLE Games;
```



删除后 dbo.Games 果然不见了

```
-- 重新创建 Games 表(因为后续操作需要)
CREATE TABLE Games (
    GameID INT PRIMARY KEY,
    GameDate DATE NOT NULL,
```

```
    HomeTeamID INT NOT NULL FOREIGN KEY REFERENCES Teams(TeamID),
    AwayTeamID INT NOT NULL FOREIGN KEY REFERENCES Teams(TeamID),
    HomeScore INT,
    AwayScore INT,
    Venue VARCHAR(100),
    Season VARCHAR(10));
```



4、索引创建：

```
--在 Players 表的 LastName 上创建索引
CREATE INDEX IX_Players_LastName ON Players(LastName);
```

5、索引删除：

```
--删除 Players 表的 LastName 索引
DROP INDEX IX_Players_LastName ON Players;
```

2）数据操作：

1、插入数据：

运行：

```
-- 插入 5 支球队
INSERT INTO Teams VALUES
(1, 'Lakers', 'Los Angeles', 'West', 'Pacific', 1947, 'Cryp
to.com Arena'),
(2, 'Warriors', 'San Francisco', 'West', 'Pacific', 1946, '
Chase Center'),
(3, 'Celtics', 'Boston', 'East', 'Atlantic', 1946, 'TD Gard
en'),
(4, 'Bulls', 'Chicago', 'East', 'Central', 1966, 'United Ce
nter'),
(5, 'Nets', 'Brooklyn', 'East', 'Atlantic', 1967, 'Barclays
 Center');
-- 插入 5 名球员
INSERT INTO Players VALUES
(1, 'LeBron', 'James', '1984-12-30', 2.06, 113, 'Forward',
23, 1, 2003, 'USA', 1),
(2, 'Stephen', 'Curry', '1988-03-14', 1.91, 86, 'Guard', 30,
 2, 2009, 'USA', 1),
(3, 'Kevin', 'Durant', '1988-09-29', 2.08, 109, 'Forward',
7, 5, 2007, 'USA', 1),
(4, 'Giannis', 'Antetokounmpo', '1994-12-06', 2.11, 110, 'F
orward', 34, NULL, 2013, 'Greece', 1),
```

```
(5, 'Nikola', 'Jokic', '1995-02-19', 2.11, 129, 'Center', 15, NULL, 2014, 'Serbia', 1);
-- 插入 5 场比赛
INSERT INTO Games VALUES
(1, '2023-10-24', 1, 2, 123, 120, 'Crypto.com Arena', '2023-24'),
(2, '2023-10-25', 3, 5, 110, 115, 'TD Garden', '2023-24'),
(3, '2023-10-26', 2, 4, 118, 112, 'Chase Center', '2023-24'),
(4, '2023-10-27', 1, 3, 105, 108, 'Crypto.com Arena', '2023-24'),
(5, '2023-10-28', 5, 4, 112, 110, 'Barclays Center', '2023-24');
```

2、修改数据：

```
-- 更新 Giannis 的球队为雄鹿队(假设雄鹿 TeamID=6，需要先插入)
-- 先添加雄鹿队
INSERT INTO Teams VALUES (6, 'Bucks', 'Milwaukee', 'East', 'Central', 1968, 'Fiserv Forum');
-- 更新球员
UPDATE Players SET TeamID = 6 WHERE PlayerID = 4;
```

	PlayerID	FirstNa...	LastNa...	BirthDate	Height	Weight	Position	JerseyN...	TeamID	DraftYear	Country	IsActive
▶	1	LeBron	James	1984-12...	2.06	113	Forward	23	1	2003	USA	True
	2	Stephen	Curry	1988-03...	1.91	86	Guard	30	2	2009	USA	True
	3	Kevin	Durant	1988-09...	2.08	109	Forward	7	5	2007	USA	True
	4	Giannis	Antetok...	1994-12...	2.11	110	Forward	34	6	2013	Greece	True
	5	Nikola	Jokic	1995-02...	2.11	129	Center	15	*NULL*	2014	Serbia	True
*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

发现 4 号球员的 TeamID 已经更新成了 6.

3、删除数据：

```
--删除一场比赛

DELETE FROM Games WHERE GameID = 5;
```

WMY-COMPUTER....se - dbo.Games | WMY-COMPUTER.N...e - dbo.Players | SQLQue

	GameID	GameD...	HomeTe...	AwayTe...	HomeSc...	AwaySc...	Venue	Season
▶	1	2023-10...	1	2	123	120	Crypto.c...	2023-24
	2	2023-10...	3	5	110	115	TD Gard...	2023-24
	3	2023-10...	2	4	118	112	Chase C...	2023-24
	4	2023-10...	1	3	105	108	Crypto.c...	2023-24
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

dbo.Games 表变成了 4 场；

4、单表查询：

```
--查询所有东部球队

SELECT TeamName, City, Division

FROM Teams

WHERE Conference = 'East';
```

WMY-COMPUTER....se - dbo.Games | WMY-COMPUTER.N...e - dbo.Playe

```
SELECT TeamName, City, Division
FROM Teams
WHERE Conference = 'East';
```

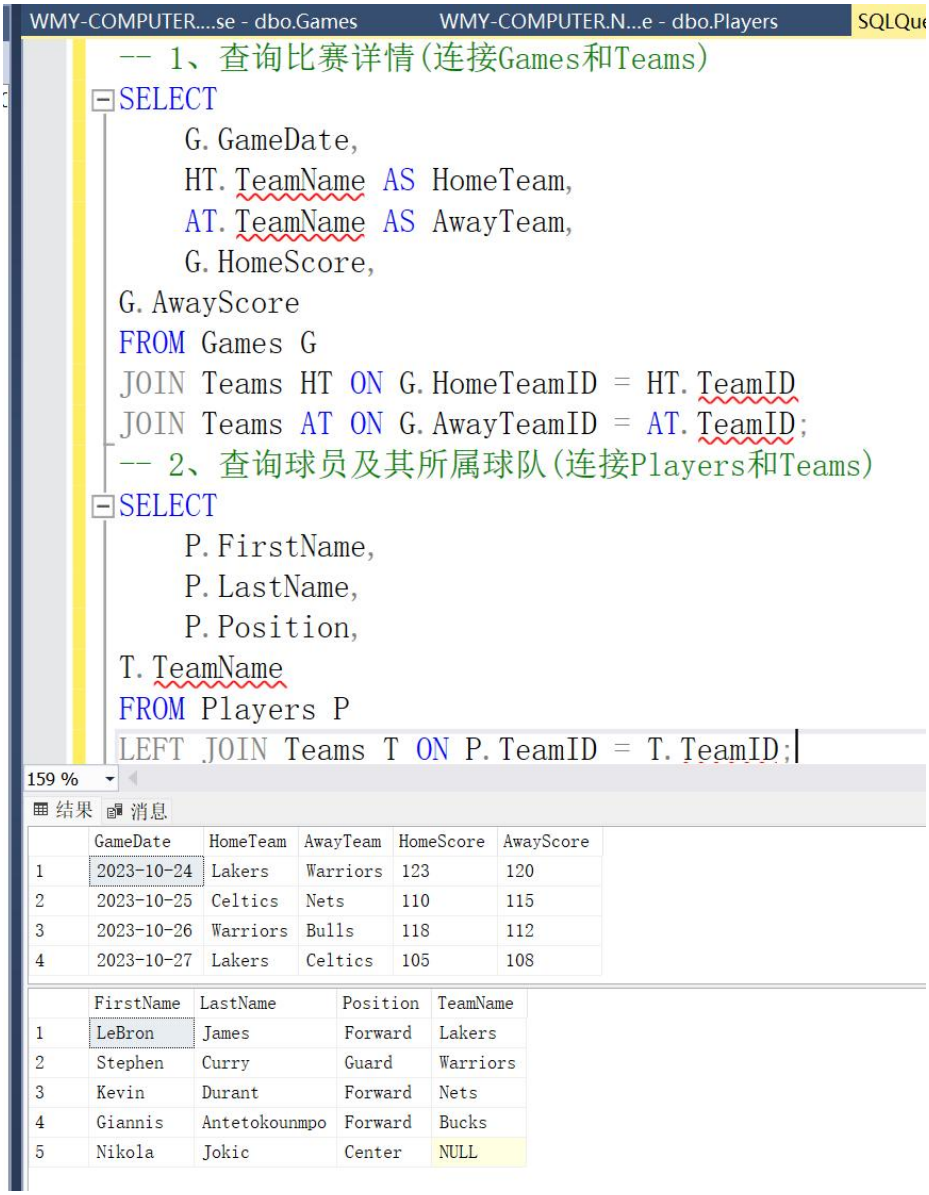
159 %

结果 | 消息

	TeamName	City	Division
1	Celtics	Boston	Atlantic
2	Bulls	Chicago	Central
3	Nets	Brooklyn	Atlantic
4	Bucks	Milwaukee	Central

5、连接查询：

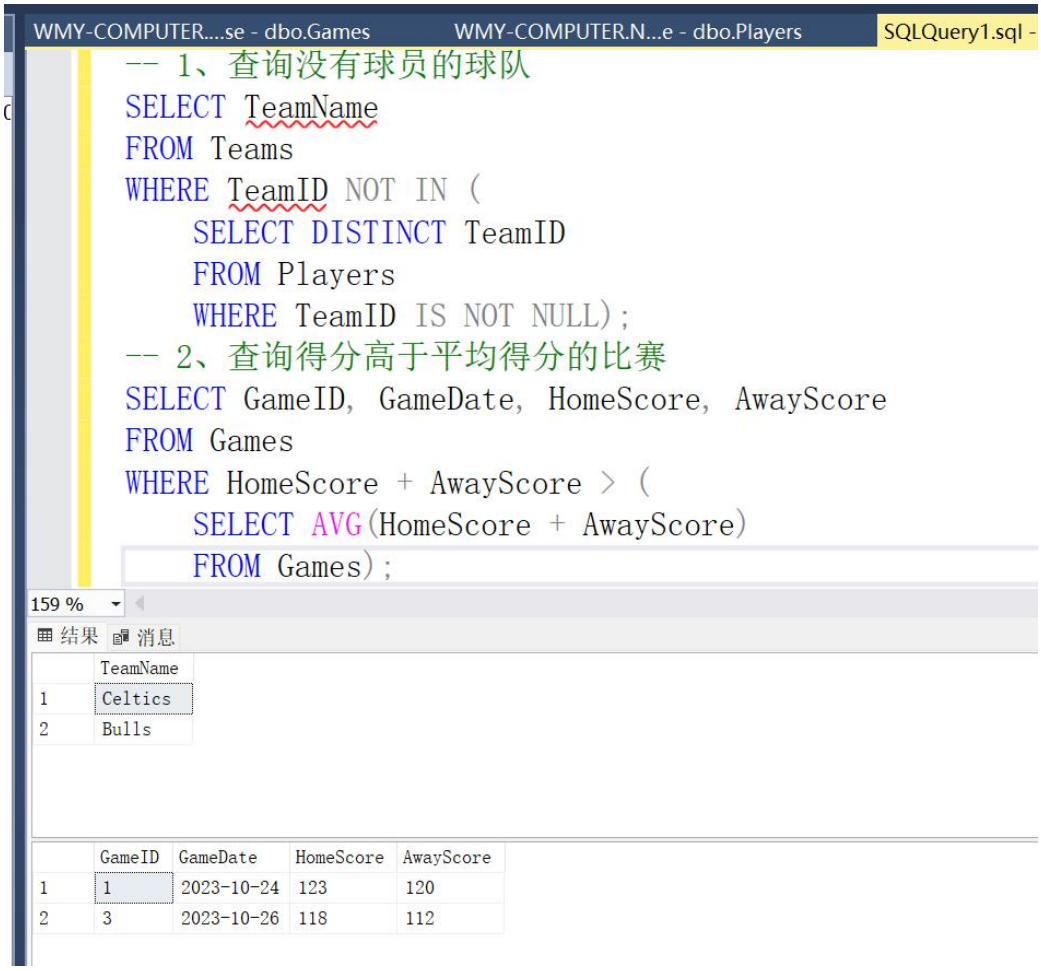
```
-- 1、查询比赛详情(连接 Games 和 Teams)
SELECT
    G.GameDate,
    HT.TeamName AS HomeTeam,
    AT.TeamName AS AwayTeam,
    G.HomeScore,
    G.AwayScore
    FROM Games G
    JOIN Teams HT ON G.HomeTeamID = HT.TeamID
    JOIN Teams AT ON G.AwayTeamID = AT.TeamID;
-- 2、查询球员及其所属球队(连接 Players 和 Teams)
SELECT
    P.FirstName,
    P.LastName,
    P.Position,
    T.TeamName
    FROM Players P
    LEFT JOIN Teams T ON P.TeamID = T.TeamID;
```



6、嵌套查询：

```
-- 1、查询没有球员的球队
SELECT TeamName
FROM Teams
WHERE TeamID NOT IN (
    SELECT DISTINCT TeamID
    FROM Players
    WHERE TeamID IS NOT NULL);
```

```
-- 2、查询得分高于平均得分的比赛
SELECT GameID, GameDate, HomeScore, AwayScore
FROM Games
WHERE HomeScore + AwayScore > (
    SELECT AVG(HomeScore + AwayScore)
    FROM Games);
```



7、集合查询：

```
-- 查询身高超过 2.05 米且体重超过 110 公斤的球员
SELECT FirstName, LastName
FROM Players
```

```
WHERE Height > 2.05
INTERSECT
SELECT FirstName, LastName
FROM Players
WHERE Weight > 110;
```

结果显示勒布朗詹姆斯和尼古拉约基奇都是身高超过 2.05 米且体重超过 110 公斤的球员。

三、出现的问题及解决方法

1、表删除失败

问题：最开始建的数据库里定义了 4 个表，多定义了一个，所以得缩减到 3 各表，计划先把这四个表都删掉，重新定义三个表，

在尝试删除 Teams 表时，出现错误："无法删除对象'Teams'，因为该对象正由 FOREIGN KEY 约束引用"。

错误原因：Teams 表被 Players 表通过 TeamID 间接关联

解决方法：因为 Teams 表被 Players 表通过 TeamID 间接关联,但我是要删除所有表，

因此，我先运行

```
DROP TABLE Players;
```

再运行

```
DROP TABLE Teams;
```

先删除 Players 表再删除 Teams 表就不会遇到报错了。

2、集合查询返回空结果

问题：最开始设计的集合查询返回空结果。

比如我如果运行：

```
-- 查询既有主场赢球记录又有客场赢球记录的球队
SELECT T.TeamName
FROM Teams T
WHERE T.TeamID IN (
    SELECT HomeTeamID
    FROM Games
    WHERE HomeScore > AwayScore)
    INTERSECT
    SELECT T.TeamName
    FROM Teams T
    WHERE T.TeamID IN (
    SELECT AwayTeamID
```

```
    FROM Games
    WHERE AwayScore > HomeScore);
```

它会返回：

实际上 NBA 赛制是每个球队都进行 82 场常规赛，基本上每支球队都会既有主场胜利又有客场胜利。

错误原因：Teams 和 Games 表中插入的数据有限，不存在满足条件的 TeamName。

解决方法：要在不改变每个表中的数据的情况下作集合查询又不产生空结果，最好是换一个可行的集合查询，最后我选择了查询身高超过 2.05 米且体重超过 110 公斤的球员，得到詹姆斯和约基奇两个结果。