

数据库工程作业

要求:

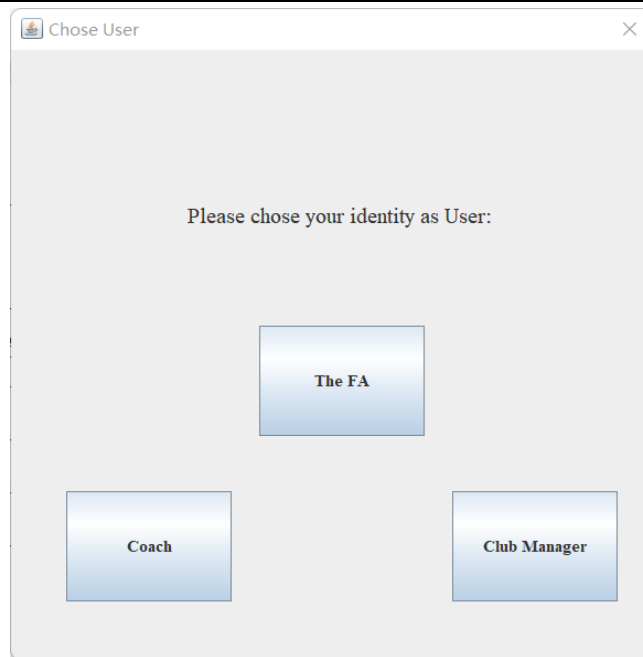
1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
2. 开发模式（B/S 或 C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle、MySQL 等），**不要使用桌面数据库。**
3. 报告中所列举的四种操作，**每种操作举一个例子即可。**
4. 作业成绩按照报告中的标准评分，**程序只实现报告中涉及的部分即可。**
5. 作业完成后，**请将工程作业报告和程序打包提交给助教老师，并联系助教老师进行系统说明和演示，回答相关问题。**

工程作业报告

1. 项目信息（10分）

学号	2111408	姓名	周钰宸	专业	信息安全
项目名称	英格兰超级联赛（Premier League）信息管理系统				
必备环境	MYSQL、JRE				
系统主要功能简介（4分）	<p>一. 领域简介:</p> <ol style="list-style-type: none">1. 英超是“欧洲足球五大联赛”之一，由 20 支球队组成，由超级联盟和英足总负责具体运作。是当今运作和经营最成功的联赛之一。2. 每个球队称为俱乐部(club)，每个俱乐部中都有各自所属的英国地区与城市，有一座坐落在本城市的主场球场(home stadium)。3. 英超联赛一般由九月到五月，每队踢 38 场比赛（19 个主场和 19 个客场）。在比赛中，每支球队获得 3 分，如果没有双方取得胜利，就各获 1 分。每场比赛都有属于本场的主裁(referee)、场地(pitch)、开球时间(kick off time)、主队(home team)、客队(visiting team)。4. 每个球员应该有自己的姓名、所属俱乐部、年龄、国籍、位置（其中只有四种位置：forward 前锋、midfielder 中场、defender 后卫、goalkeeper 门将）、转会身价(transfer value)。5. 不同位置的球员会有所具有的不同数据，比如前锋有进球、中场有助攻、后卫有拦截和解围、门将零封等。6. 裁判包括主裁和助理裁判等，应至少包含姓名和性别。 <p>二. 实现功能:</p> <ol style="list-style-type: none">1. 主界面：进入与退出，播放英超主题曲音乐与停止播放。2. 模式选择界面，根据身份，选择符合身份的模式进入对应界面。				

	<p>3. 俱乐部教练模式: (Club Coach) 显示 2022-2023 赛季英超的进球榜 (top goal scorers)</p> <p>4. 俱乐部总经理模式: (Club Manager)</p> <ul style="list-style-type: none"> (1) 向你拥有的俱乐部买入球员; (2) 卖出你拥有的俱乐部的对应球员; (3) 查询你拥有的俱乐部的全部球员 (4) 更新你所在的俱乐部的球员的最新位置和转会身价。 <p>5. 英足总模式: (The Football Association)</p> <ul style="list-style-type: none"> (1) 显示所有比赛、显示所有俱乐部、显示所有教练、显示所有裁判; (2) 显示前锋球员的所有信息、显示门将球员的所有信息; (3) 当赛季结束时, 积分榜末三位应该降入英格兰足球冠军联赛, 离开英格兰足球超级联赛。令对应的该退出英超的俱乐部降级 (Relegation)。
<p>系 统 主 要 页 面 截 图 (6 分)</p>	<p>(1) 主界面:</p>  <p>(2) 模式选择界面:</p>



(3) 教练模式: (Coach)

The screenshot shows a window titled "Premier League Database:Coach" with standard window controls. It contains a table with the following data:

name	club	goal	ranking
Erling Haaland	Manchester City	36	1
Harry Kane	Tottenham Hotspur	28	2
Mohamed Salah	Liverpool	19	3
Marcus Rashford	Manchester United	17	4
Gabriel Martinelli	Arsenal	15	5
Bukayo Saka	Arsenal	13	6
Alexander Isak	Newcastle United	10	7
Gabriel Jesus	Arsenal	10	7
Son Heung-Min	Tottenham Hotspur	10	7
Darwin Núñez	Liverpool	9	10
Julián Álvarez	Manchester City	9	10
Kai Havertz	Chelsea	7	12
Kaoru Mitoma	Brighton & Hove Albion	7	12
Cody Gakpo	Liverpool	6	14
Raheem Sterling	Chelsea	6	14
Jack Grealish	Manchester City	5	16
Antony Santos	Manchester United	4	17
Pierre-Emerick Aubameyang	Chelsea	1	18
Richarlison	Tottenham Hotspur	1	18
Wout Weghorst	Manchester United	0	20

Below the table, there is a label "Top Goal Scorers:" followed by a blue button with the text "Show".

(4) 俱乐部经理模式: (Club Manager)

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal
Marcus Rashford	25	England	forward	80000000	Manchester United
Mason Mount	24	England	midfielder	65000000	Chelsea

club: name: age: nationality:

club: name: club:

club: name: pos: value:

(5) 英足总模式: (The Football Association)

Premier League Database: The Football Association

kick_off_time	pitch	referee	home_team	visting_team
2023-04-26 20:00:00.0	Etihad Stadium	Michael Oliver	Manchester City	Arsenal
2023-04-23 14:00:00.0	St.Jame's Park	David Coote	Newcastle United	Tottenham Hotspur
2023-03-05 17:30:00.0	Anfield	Andy Madley	Liverpool	Manchester United
2023-01-04 13:30:30.0	Old Trafford	Stuart Attwell	Manchester United	Manchester City

for_info: mid_info: def_info: goa_info:

match: club: coach: referee:

club_name:

备注

由于时间原因，有许多预先设计好的功能尚未在前端实现，**这里列出所有预想但未在前端实现的功能：**

1. 模式选择界面：根据身份，应该输入对应的姓名和登录密码才能进入对应页面。我所实现的功能更像是一个足球不同身份的体验模式与体验游戏，而不是真实的管理系统。
2. 俱乐部教练模式：（Club Coach）
 - （1）根据你登录进来的身份只能对你所属于的俱乐部球员进行操作，而不是可以对并非你的俱乐部球员进行操作。
 - （2）根据你所在的俱乐部，显示对应的所有球员；若多输入球员位置，则改为显示所有该位置球员的具体进球等对应数据。
 - （3）不只显示球员的进球榜，还可以显示球员的助攻榜、零封榜等。
3. 俱乐部经理模式：（Club Manager）
 - （1）实现资金管理，为每家俱乐部添加资金属性，根据球员的转会身价实现俱乐部的收入与支出。
 - （2）实现球员转会：不只能够实现买与卖球员，同时实现向对应的英超或者其它联赛俱乐部售卖或者购买球员。
 - （3）实现雇佣与解雇主教练：为球队更新主教练。

<p>(4) 实现俱乐部信息的更改：更新俱乐部名、球场、地区等。</p> <p>(5) 根据你登录进来的身份只能对你所属于的俱乐部球员进行操作，而不是可以对并非属于你的俱乐部球员进行操作。</p> <p>4. 英足总模式：(The Football Association)</p> <p>(1) 实现雇佣与解雇裁判；</p> <p>(2) 实现每赛季末英冠的积分前两名升入英超，为英超添加新的俱乐部；</p> <p>(3) 实现赛程管理：添加赛程、取消赛程、更改赛程信息（裁判、场地、开球时间等）；</p> <p>(4) 实现对球员国籍、年龄等私密信息的更改。</p> <p>(5) 显示本赛季的球队积分榜，并为冠军球队颁发当赛季的英超冠军奖杯。</p>
--

2. 系统配置 (10 分)

说明		(2 分) 请说明系统配置情况 (后台数据库, 高级语言) ; (8 分) 请使用连接串连接高级语言和数据库, 并分析字符串的各个部分。		
配置 步骤 2 分	DBMS	1. MySQL 作为后端数据库管理系统, 数据引擎 innodb, 采用 utf-8 编码与 Unicode 字符集		
		2. Navicat 作为可视化操作的工具		
高级 语言		1. Java JDBC 接口实现与数据库的连接		
		2. Java Swing 与 AWT 包绘制 GUI 界面实现可视化前端		
连接串 分析 (6 分)	序号	名称	功能说明	取值
	1	url (统一资源定位符)	<p>1. <code>jdbc:mysql://</code>——JDBC 的连接协议和子协议部分, 指示要使用的数据库驱动程序和数据库类型。使用 MySQL 数据库。</p> <p>2. <code>localhost:3306</code>——MySQL 数据库服务器的主机名和端口号。localhost 表示使用本地主机, 而 3306 是 MySQL 数据库服务器的默认端口号。</p> <p>3. <code>premierleague</code>——所使用的后端 MySQL 中的数据库, 名为 premierleague。</p> <p>4. <code>?useUnicode=true&characterEncoding=utf8</code> ——连接 URL 的查询参数部分, 用于指定连接的附加选项。useUnicode=true 指示使用 Unicode 字符集, characterEncoding=utf8 指示使用 UTF-8 字符编码。这可以确保在与数据库进行数据交互时正确处理 Unicode 字符。</p>	整体取值: <code>jdbc:mysql://localhost:3306/premierleague?useUnicode=true&characterEncoding=utf8</code>
	2	user	连接数据库所需的用户名	root
	3	password	连接数据库所需的密码	保密
	4	Class.forName	加载驱动	<code>com.mysql.jdbc.Driver</code>
5	DriverManager	输入 url, user, password 后获取 MySQL 连接	url, user, password	

	r. getC onnect ion		
<p>连接串代码 (截屏) (2分)</p>	<p>JDBCUtils.java (JDBC 工具类)——实现了连接和关闭代码的简洁,避免了代码复用。</p> <p>1. 静态变量定义与加载驱动:</p> <pre>public class JDBCUtils { private static String url="jdbc:mysql://localhost:3306/premierleague?useUnicode=true&characterEncoding=utf8"; private static String user="root"; private static String password="53468896Aa@"; private JDBCUtils() {}; static { try { //加载驱动 Class.forName("com.mysql.jdbc.Driver"); } catch (Exception e) { e.printStackTrace(); } } }</pre> <p>2. 获取连接对象静态函数 (方便直接调用):</p> <pre>/** * 获取连接对象 * @return 连接对象 */ public static Connection getConnection() throws Exception { return DriverManager.getConnection(url, user, password); }</pre> <p>3. 释放资源的静态函数, 当在无需关闭 ResultSet 时候使用:</p> <pre>/** * 释放资源 * @param statement sql执行语句对象 * @param connection 连接对象 */ public static void close(Statement statement, Connection connection) { close(null, statement, connection); }</pre> <p>4. 释放资源的静态函数, 可以关闭 PreparedStatement、CallableStatement 等多种继承自 Statement 类的对象, 也可以用时关闭 ResultSet 类对象:</p>		

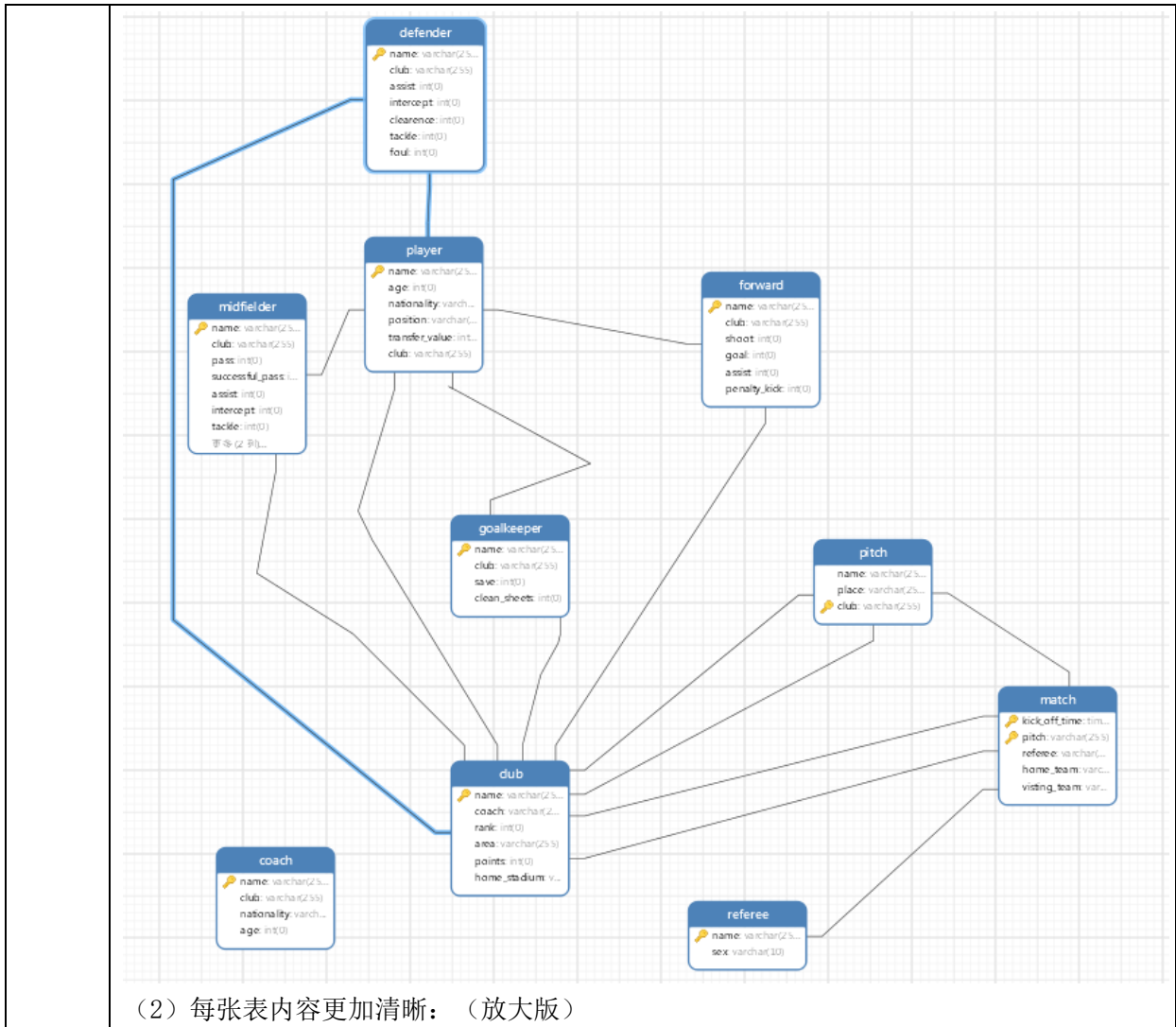
	<pre> /** * 释放资源 * @param resultSet 结果集对象 * @param statement sql执行语句对象 * @param connection 连接对象 */ public static void close(ResultSet resultSet, Statement statement, Connection connection) { if (resultSet != null) { try { resultSet.close(); } catch (SQLException e) { e.printStackTrace(); } } else if (statement != null) { try { statement.close(); } catch (SQLException e) { e.printStackTrace(); } } if (connection != null) { try { connection.close(); } catch (SQLException e) { e.printStackTrace(); } } } </pre>
备注	<p>JDBC 工具类的设计实现了在其他类中直接调用获取连接与关闭。 关闭时采用了 finally 结构：以防在 conn close 之前出现异常无法关闭 }finally{//以防在conn close之前出现异常无法关闭 JDBCUtils.close(rs, ps, conn); }</p>

3. 数据库设计（14分）

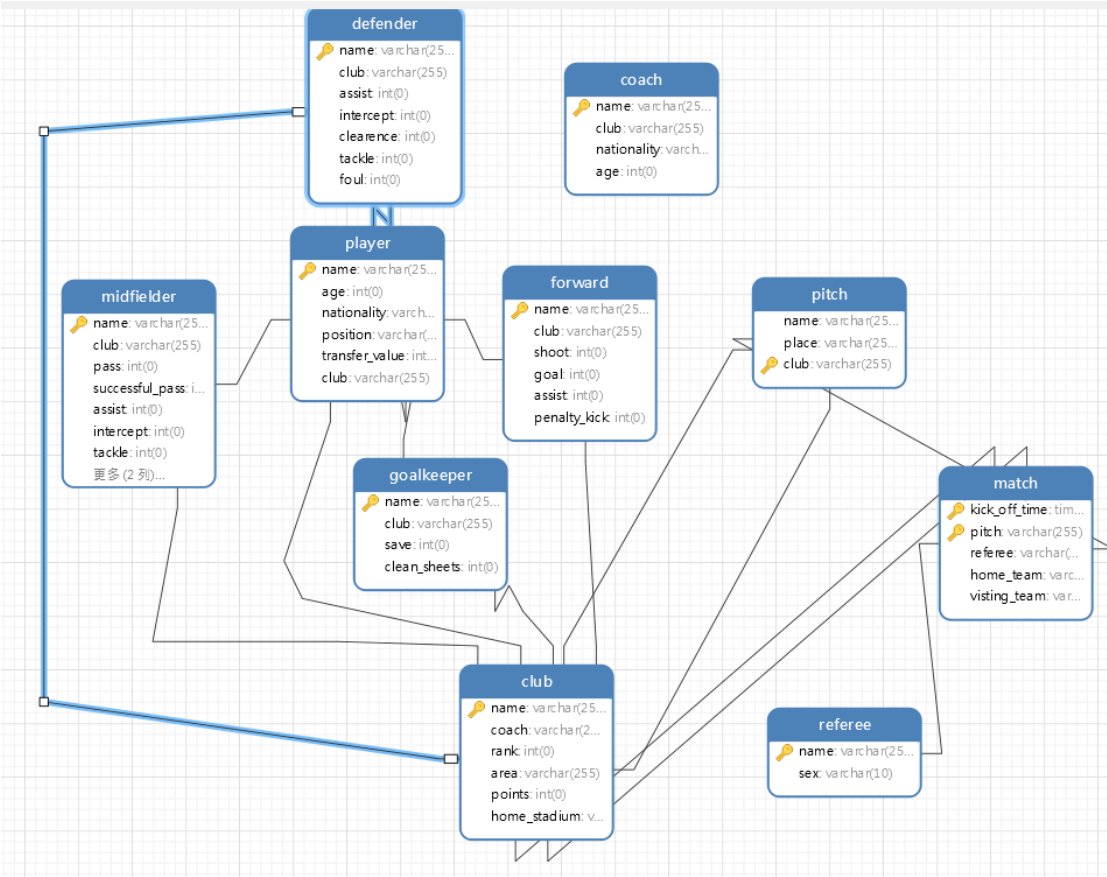
说明	<p>（10分）按照数据表的创建顺序，依次给出所涉及数据表的信息，其中参照字段以“（字段1，字段2，……，字段n）”的形式给出，被参照字段以“表名（字段1，字段2，……，字段n）”的形式给出； （4分）一般DBMS都可以为数据库生成关系图，请将该图片截屏并粘贴到表格中。</p>				
数据表 (10)	创建顺序	数据表名称	主键	参照属性	被参照表及属性
	1	club	name	home_stadium	pitch(name) 【 Delete:SET NULL Update:CASCADE 】
	2	coach	name	无	无
	3	player	name	club	club(name) 【 Delete:CASCADE Update:CASCADE

					】
	4	forward	name	1. name 2. club	1. player (name) 【 Delete:CASCADE Update:CASCADE 】 2. club (name) 【 Delete:CASCADE Update:CASCADE 】
	5	midfielder	name	1. name 2. club	1. player (name) 【 Delete:CASCADE Update:CASCADE 】 2. club (name) 【 Delete:CASCADE Update:CASCADE 】
	6	defender	name	1. name 2. club	1. player (name) 【 Delete:CASCADE Update:CASCADE 】 2. club (name) 【 Delete:CASCADE Update:CASCADE 】
	7	goalkeeper	name	1. name 2. club	1. player (name) 【 Delete:CASCADE Update:CASCADE 】 2. club (name) 【 Delete:CASCADE Update:CASCADE 】

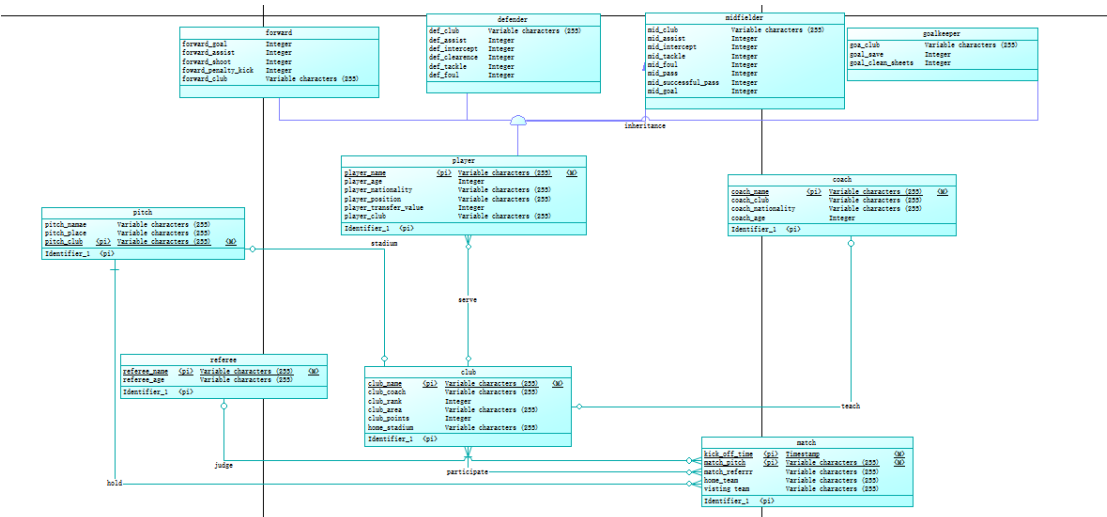
	8	pitch	name	club	club(name) 【 Delete:CASCADE Update:CASCADE 】
	9	match	(kick_off_time, pitch)	1. home_team 2. visiting_tea m 3. referee 4. pitch	1. club(name) 【 Delete:CASCADE Update:CASCADE 】 2. club(name) 【 Delete:CASCADE Update:CASCADE 】 3. referee(name) 【 Delete:SETNULL Update:CASCADE 】 4. pitch(name) 【 Delete:CASCADE Update:CASCADE 】
关系 图(4)	(截屏) 1. Navicat: 【无法体现继承关系】 (1) 整体关系更加清晰: (缩小版)				



(2) 每张表内容更加清晰：（放大版）



2. PowerDesigner: 【可以体现继承关系】



备注

1. 为了保证数据的真实性，我从英超官网和懂球帝 app 上获取了如下多条真实存在的信
息，并放入了如下的表中：

(1) player:
选择了英超的 Big6 即最大的 6 家俱乐部 Manchester City, Manchester United, Liverpool, Chelsea, Arsenal, Tottenham Hotspur 以及 2022-2023 赛季球队整体联赛表现相当不错的另外 3 家俱乐部 Newcastle United、Brighton & Hove Albion、Aston Villa。其中 Big6 俱乐部按照最经典的足球 4-3-3 战术阵型加入了 4 名后卫(defender)、3 名中场 (midfielder)、3 名前锋 (forward)。以及一些夹带私货的其它球员。总共加入了 71 名真实存在的球员

	<p>全部真实数据信息。</p> <p>(2) club: 英超 Big6 俱乐部 Manchester City, Manchester United, Liverpool, Chelsea, Arsenal, Tottenham Hotspur 以及 Newcastle United、Brighton & Hove Albion、Aston Villa 的一共 9 家球队教练以及球场地区积分等真实信息。</p> <p>(3) coach: 加入了上述 9 家俱乐部的对应教练的国籍等真实信息。</p> <p>(4) referee: 加入了 10 名最著名的英超裁判的真实信息。</p> <p>(5) pitch: 加入了上述 9 家俱乐部的主场球场的真实信息。</p> <p>(6) match: 加入了 2022-2023 赛季真实发生的 4 场著名比赛。</p> <p>(7) forward、midfielder、defender、goalkeeper: 将 player 中对应位置的球员加入对应的表中。之后又加入了 forward 的 goal 和 assist 数据, midfielder 的 assist 数据, goalkeeper 的 save 和 clean_sheets 数据。</p> <p>2. 关于包含的继承关系: 由于有些属性 (比如零封 clean sheets 对于非门将位置球员就没有统计的必要), 同时为了避免冗余, 采用 E-R 图的方式, 而没有采用 0-0 法或者空值法。由此设计了继承自 player 的四张关系模式 forward, midfielder, defender, goalkeeper。这四张关系模式各自都继承下来了 player 的主键属性: name, 同时包含 club 和其它统计数据。</p> <p>3. 为了实现事务应用的删除操作, 上述有一个外键理应设置, 但没有设置 (只是为了完成报告要求而实现), 其中 coach 的 club 属性本应也参照 club 里面的 name 属性。但如果设置外键并设置级联删除 CASCADE, 就可以实现事务应用的删除操作了, 便暂时没有设置该外键。未来我在完善该作品时会将该外键补充完整。</p> <p>4. 本数据库中所有已经加入的数据均截止到 2022-2023 赛季英超足球联赛的 2023. 5. 22 日。之后的数据没有进一步更新。</p>
--	---

4. 含有事务应用的删除操作 (13 分)

说明	<p>(1 分) 简要说明该操作所要完成的功能;</p> <p>(2 分) 该操作会涉及的表 (必须含有两张或两张以上的关系表, 同时以“表名”的形式给出)</p> <p>(1 分) 表连接涉及字段描述 (描述方式为“表 1. 属性=表 2. 属性”)</p> <p>(1 分) 删除条件涉及的字段描述 (以“表名. 属性=? ”形式给出)</p> <p>(4 分) 实现该操作的关键代码 (高级语言、SQL), 截图即可; (其中如果删除语句中不包含任何形式的事务应用将扣除 3 分)</p> <p>(4 分) 如何执行该操作, 按所述方法能够正常演示程序则给分。</p>
功能描述 (1 分)	<p>在英足总模式 (The Football Association) 下:</p> <p>当删除一家存在的俱乐部时候 (令该俱乐部的降级或解散), 同时删除该俱乐部的教练。(俱乐部降级: 该教练不再属于英超管理; 解散同理。)</p>
涉及的表 (2 分)	club、coach

分)		
表 连 接 涉 及 字 段 (1 分)	club.name=coach.club	
删 除 条 件 字 段 描 述 (1 分)	字段	规则
	1. DELETE FROM club WHERE `name`=?; 删除某俱乐部的全部信息 2. DELETE FROM coach WHERE club=?; 删除属于该俱乐部的教练的全部信息	使用 JDBC 占位符, 当 FA 英足总模式下, 输入想要降级的俱乐部名字 (name 作为 club 的主键), 点击 "Relegation" 按钮。即可删除该俱乐部信息。该俱乐部的教练信息也会同时被删除。
代 码 (4 分)	(截屏)	

	<pre> public void actionPerformed(ActionEvent e) { Connection conn=null; PreparedStatement ps1=null; PreparedStatement ps2=null; try { conn=JDBCUtils.getConnection(); conn.setAutoCommit(false); String c1=ClubNameTextField2.getText(); ClubNameTextField2.setText(""); //删除语句1 String sql1="DELETE FROM club WHERE `name`=?"; ps1=conn.prepareStatement(sql1); ps1.setString(1, c1); //删除语句2 String sq2="DELETE FROM coach WHERE club=?"; ps2=conn.prepareStatement(sq2); ps2.setString(1, c1); //执行删除语句 ps1.executeUpdate(); ps2.executeUpdate(); //提交事务 conn.commit(); } catch (Exception e1) { //遇到异常，回滚事务 try { conn.rollback(); } catch (SQLException e2) { e2.printStackTrace(); } e1.printStackTrace(); }finally{//以防在conn close之前出现异常无法关闭 JDBCUtils.close(ps1, conn); JDBCUtils.close(ps2, null); } } }); return RelegationButton; </pre>
程序演示 (4分)	1. 插入一个新俱乐部名为“小元的家”，其教练为“小元”

name	coach	rank	area	points	home_stadium
Arsenal	Mikel Arte	2	London	81	Emirates Stadium
Aston Vill	Unai Emer	7	Birmingl	58	Villa Park
Brighton	Roberto C	6	Brightor	61	Amex Stadium
Chelsea	Frank Lam	12	London	43	Stamford Bridge
▶ Liverpool	Jürgen Klo	5	Liverpoc	66	Anfield
Manchester	Pep Guar	1	Manche	88	Etihad Stadium
Manchester	Eric ten H	4	Manche	69	Old Trafford
Newcastl	Edward H	3	Newcast	69	St.Jame's Park
Tottenham	Ryan Mas	8	London	57	Tottenham Hotspur
小元的家	小元	(Null)	(Null)	(Null)	(Null)

插入一个新教练“小元”，所属于俱乐部为“小元的家”

name	club	nationality	age
Edward H	Newcast	England	45
Eric ten H	Manche	Netherland	53
Frank Lan	Chelsea	England	44
Jürgen Kl	Liverpoc	Germany	55
▶ Mikel Art	Arsenal	Spain	41
Pep Guar	Manche	Spain	52
Roberto I	Brightor	Italy	43
Ryan Mas	Tottenh	England	31
Unai Eme	Aston Vi	Spain	51
小元	小元的家	(Null)	(Null)

2. 在程序中进入 FA 模式界面，输入俱乐部名字“小元的家”，然后点击“Relegation”降级——

Premier League Database:The Football Association

name	coach	rank	area	points	home_stadium
小元的家	小元	0		0	
Tottenham Hotspur	Ryan Mason	8	London	57	Tottenham Hotspur Stadi...
Newcastle United	Edward Howe	3	Newcastle	69	StJame's Park
Manchester United	Eric ten Hag	4	Manchester	69	Old Trafford
Manchester City	Pep Guardiola	1	Manchester	88	Ethad Stadium
Liverpool	Jürgen Klopp	5	Liverpool	66	Anfield
Chelsea	Frank Lampard	12	London	43	Stamford Bridge
Brighton & Hove Albion	Roberto De Zerbi	6	Brighton	61	Amex Stadium
Aston Villa	Unai Emery	7	Birmingham	58	Villa Park
Arsenal	Mikel Arteta	2	London	81	Emirates Stadium

for_info: mid_info: def_info: goa_info:

match: club: coach: referee:

club_name:

3.重新点击 show club, 可以发现“小元的家”俱乐部已经不再显示了。证明成功删除。

Premier League Database:The Football Association

name	coach	rank	area	points	home_stadium
Tottenham Hotspur	Ryan Mason	8	London	57	Tottenham Hotspur Stadi...
Newcastle United	Edward Howe	3	Newcastle	69	StJame's Park
Manchester United	Eric ten Hag	4	Manchester	69	Old Trafford
Manchester City	Pep Guardiola	1	Manchester	88	Ethad Stadium
Liverpool	Jürgen Klopp	5	Liverpool	66	Anfield
Chelsea	Frank Lampard	12	London	43	Stamford Bridge
Brighton & Hove Albion	Roberto De Zerbi	6	Brighton	61	Amex Stadium
Aston Villa	Unai Emery	7	Birmingham	58	Villa Park
Arsenal	Mikel Arteta	2	London	81	Emirates Stadium

此时点击 show coach, 可以发现教练“小元”也已经不在了, 证明成功删除。

Premier League Database:The Football Association

name	club	nationality	age
Unai Emery	Aston Villa	Spain	51
Ryan Mason	Tottenham Hotspur	England	31
Roberto De Zerbi	Brighton & Hove Albion	Italy	43
Pep Guardiola	Manchester City	Spain	52
Mikel Arteta	Arsenal	Spain	41
Jürgen Klopp	Liverpool	Germany	55
Frank Lampard	Chelsea	England	44
Eric ten Hag	Manchester United	Netherlands	53
Edward Howe	Newcastle United	England	45

由此可知, 能够成功实现含有事务应用的删除操作。

- 备注
1. 实际上使用外键约束的 CASCADE 删除也可以实现, 但不包含事务。
 2. 实际上凭借 player.name 对 club.name 的参照以及 DELETE ON CASCADE 设置, 这个过程令一个俱乐部降级后, 该球队的所有球员信息也会被同时删除。同时由于 forward 等四个位置的球员表中对应的 club 字段也参照了 club.name 以及 DELETE ON CASCADE 设置, 因此也都会

	被同时清空。同理还有 match 的表中的 home_team、visiting_team 字段。这样就实现了所有该球队涉及的表的信息的清空。
--	---

触发器控制下的添加操作（20分）

说明	<p>(1分) 简要说明该操作所要完成的功能；</p> <p>(2分) 简要说明该触发器所要完成的功能</p> <p>(1分) 该操作会涉及的表（以“表名”的形式给出）。</p> <p>(2分) 该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空；</p> <p>(6分) 实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>(8分) 如何执行该操作，按所述方法能够正常演示程序则给分。</p>	
功能描述 (1分)	<p>在俱乐部经理模式 (Club Manager) 下：</p> <p>实现买入一个位置的球员时，将该位置的球员信息插入 player 表的同时，也插入对应位置的表中。</p>	
触发器描述 (2分)	<p>1. 触发器名：BuyPlayerTrig</p> <p>2. 事件：在像 player 表插入数据之前</p> <p>3. 条件：</p> <p>(1) 插入的球员俱乐部在已存在的俱乐部中</p> <p>(2) 插入的球员位置是属于 forward、midfielder、defender、goalkeeper 其中之一。</p> <p>4. 动作：根据插入的球员位置，将其的 name 和 club 信息插入到对应的位置的表中。</p>	
涉及的表 (1分)	<p>player、forward(或 midfielder 或 defender 或 goalkeeper)</p>	
输入数据 (2分)	字段	规则
	player.name	无
	player.club	输入的俱乐部必须是已经存在的俱乐部；即在 club 表中。因为作为某俱乐部的经理，你只能向你的已经存在的俱乐部买入球员。
	player.age	无
	player.nationality	无
	player.position=forward	实际上不用在俱乐部经理模式手动输入，因为一个俱乐部教练购买球员后，其实际位置往往是由教练确定，因此先默认为 forward。
插入操作源码 (3分)	<p>(截屏)</p>	

	<pre> @Override public void actionPerformed(ActionEvent e) { // TODO Auto-generated method stub Connection conn=null; PreparedStatement ps=null; ResultSet rs=null; try { conn=JDBCUtils.getConnection(); String na=NamTextField.getText(); NamTextField.setText(""); String cl=ClubTextField.getText(); ClubTextField.setText(""); String ag=AgeTextField.getText(); AgeTextField.setText(""); String nat=NatTextField.getText(); NatTextField.setText(""); String sql="INSERT INTO player(name,age,nationality,position,transfer_value,club) VALUES(?,?,?,?);"; ps=conn.prepareStatement(sql); ps.setString(1, na); ps.setInt(2, Integer.parseInt(ag)); ps.setString(3, nat); ps.setString(4, "forward");//默认position=forward ps.setInt(5, -1); ps.setString(6, cl); ps.executeUpdate(); JOptionPane.showMessageDialog(null, "Buy successfully!", "Buy successfully", JOptionPane.INFORMATION_MESSAGE); } catch (Exception e1) { //若违反了触发器的条件,即输入的俱乐部不存在,则违反触发器,弹出窗口报错 JOptionPane.showMessageDialog(null,"Violated Trigger:BuyPlayerTrig","Insert failed!", JOptionPane.ERROR_MESSAGE); e1.printStackTrace(); }finally{//以防在conn close之前出现异常无法关闭 JDBCUtils.close(rs, ps, conn); } showPlayer();//刷新界面并显示 } </pre>
<p>触发器源代码 (3分)</p>	<p>(截屏)</p>  <pre> 1 DROP TRIGGER IF EXISTS BuyPlayerTrig; 2 delimiter // 3 CREATE TRIGGER BuyPlayerTrig 4 AFTER INSERT ON player FOR EACH ROW 5 BEGIN 6 IF NEW.club IN (SELECT `name` FROM club) AND NEW.position IN ('forward','midfielder','defender','goalkeeper') THEN 7 IF NEW.position='forward' THEN 8 INSERT INTO forward(`name`,club) VALUES(NEW.`name`,NEW.club); 9 END IF; 10 IF NEW.position='midfielder' THEN 11 INSERT INTO midfielder(`name`,club) VALUES(NEW.`name`,NEW.club); 12 END IF; 13 IF NEW.position='defender' THEN 14 INSERT INTO defender(`name`,club) VALUES(NEW.`name`,NEW.club); 15 END IF; 16 IF NEW.position='goalkeeper' THEN 17 INSERT INTO goalkeeper(`name`,club) VALUES(NEW.`name`,NEW.club); 18 END IF; 19 ELSE 20 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Violated TRIGGER:BuyPlayerTrig'; 21 END IF; 22 END; // 23 delimiter ; </pre>
<p>程序演示 (4分)</p>	<p>1. 向一个已经存在的俱乐部” Arsenal”中插入球员“小元”，因为俱乐部存在，并且默认插入的位置为“forward”，因此满足触发器条件。点击 Buy 按钮。可以看到弹出一个对话框提示” Buy successfully! ”</p>

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal
Marcus Rashford	25	England	forward	80000000	Manchester United

club: Arsenal name: 小元 age: 20 nationality: China Buy

club: name: Sell club: Query

club: name: pos: value: Update

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal
Marcus Rashford	25	England	forward	80000000	Manchester United

Buy successfully!

Buy successfully!

确定

club: name: age: nationality: Buy

club: name: Sell club: Query

club: name: pos: value: Update

3. 在足球经理表中查看 Arsenal 俱乐部的全部球员：可以发现“小元”在其中——

name	age	nationality	position	transfer_value	club
小元	20	China	forward	-1	Arsenal
William Saliba	22	France	defender	55000000	Arsenal
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Oleksandr Zinchen...	26	Ukraine	defender	40000000	Arsenal
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal
Jorge Filho(Jorgin...	31	Italy+Brazil	midfielder	35000000	Arsenal
Granit Xhaka	30	Swiss	midfielder	28000000	Arsenal
Gabriel Martinelli	21	Brazil	forward	70000000	Arsenal
Gabriel Magalhães	25	Brazil	defender	50000000	Arsenal
Gabriel Jesus	26	Brazil	forward	75000000	Arsenal
Bukayo Saka	21	England	forward	110000000	Arsenal

club: name: age: nationality:

club: name: club:

club: name: pos: value:

4. 切换到 FA 模式，选择查看所有前锋信息——forward_info，可以发现小元也在所有前锋信息表中，由此可知，不违背触发器的条件情况下可以成功插入！

Premier League Database: The Football Association

name	club	age	nationality	transfer_value	shoot	goal	assist	penalty_kick
小元	Arsenal	20	China	-1	0	0	0	0
Wout Weghorst	Manchester Uni...	30	Netherlands	14000000	13	0	1	0
Son Heung-Min	Tottenham Hots...	30	Korea	60000000	80	10	5	0
Richarlison	Tottenham Hots...	26	Brazil	55000000	25	1	3	0
Raheem Sterling	Chelsea	28	England	60000000	37	6	3	0
Pierre-Emerick ...	Chelsea	33	Gabonese	7000000	16	1	0	0
Mohamed Salah	Liverpool	30	Egypt	70000000	117	19	11	2
Marcus Rashford	Manchester Uni...	25	England	80000000	102	17	5	0
Kaoru Mitoma	Brighton & Hove...	26	Japan	22000000	52	7	7	0
Kai Havertz	Chelsea	23	Germany	60000000	71	7	1	1
Julián Álvarez	Manchester City	23	Argentina	50000000	42	9	0	1
Jack Grealish	Manchester City	27	England	70000000	41	5	7	0
Harry Kane	Tottenham Hots...	29	England	90000000	127	28	3	5
Gabriel Martinelli	Arsenal	21	Brazil	70000000	79	15	5	0
Gabriel Jesus	Arsenal	26	Brazil	75000000	76	10	5	1
Erling Haaland	Manchester City	22	Norway	170000000	123	36	8	7
Darwin Núñez	Liverpool	23	Uruguay	70000000	84	9	3	0
Cody Gakpo	Liverpool	24	Netherlands	60000000	36	6	2	0
Bukayo Saka	Arsenal	21	England	110000000	86	13	11	2
Antony Santos	Manchester Uni...	23	Brazil	70000000	72	4	2	0
Alexander Isak	Newcastle United	23	Sverige	50000000	48	10	1	2

for_info: mid_info: def_info: goa_info:

程序
演示
(4

1. 向一个不存在的俱乐部“小元的家”中试图插入新球员“小元的哥哥”，因为俱乐部不存在，因此不满足触发器条件。点击 Buy 按钮。可以看到弹出一个对话框提示“Violated Trigger:BuyPlayerTrig”。

分)

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
小元	20	China	forward	-1	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	7000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

club: name: age: nationality:

club: name: club:

club: name: pos: value:

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
小元	20	China	forward	-1	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	2500000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	7000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

club: name: age: nationality:

club: name: club:

club: name: pos: value:

Insert failed!
Violated Trigger:BuyPlayerTrig

2. 同时此时查看所有俱乐部球员信息和 forward 表，都没有球员“小元的哥哥”的存在，只有之前满足触发器条件插入的球员“小元”：证明了违背触发器条件的情况下不能成功插入，系统报错。

name	club	age	nationality	transfer_value	shoot	goal	assist	penalty_kick
小元	Arsenal	20	China	-1	0	0	0	0
Wout Weghorst	Manchester Uni...	30	Netherlands	14000000	13	0	1	0
Son Heung-Min	Tottenham Hots...	30	Korea	60000000	80	10	5	0
Richarlison	Tottenham Hots...	26	Brazil	55000000	25	1	3	0
Raheem Sterling	Chelsea	28	England	60000000	37	6	3	0
Pierre-Emerick ...	Chelsea	33	Gabonese	7000000	16	1	0	0
Mohamed Salah	Liverpool	30	Egypt	70000000	117	19	11	2
Marcus Rashford	Manchester Uni...	25	England	80000000	102	17	5	0
Kaoru Mitoma	Brighton & Hove...	26	Japan	22000000	52	7	7	0
Kai Havertz	Chelsea	23	Germany	60000000	71	7	1	1
Julián Álvarez	Manchester City	23	Argentina	50000000	42	9	0	1
Jack Grealish	Manchester City	27	England	70000000	41	5	7	0
Harry Kane	Tottenham Hots...	29	England	90000000	127	28	3	5
Gabriel Martinelli	Arsenal	21	Brazil	70000000	79	15	5	0
Gabriel Jesus	Arsenal	26	Brazil	75000000	76	10	5	1
Erling Haaland	Manchester City	22	Norway	170000000	123	36	8	7
Darwin Núñez	Liverpool	23	Uruguay	70000000	84	9	3	0
Cody Gakpo	Liverpool	24	Netherlands	60000000	36	6	2	0
Bukayo Saka	Arsenal	21	England	110000000	86	13	11	2
Antony Santos	Manchester Uni...	23	Brazil	70000000	72	4	2	0
Alexander Isak	Newcastle United	23	Sverige	50000000	48	10	1	2

for_info:

Show

mid_info:

Show

def_info:

Show

goa_info:

Show

对象

forward @premierleague (l...

开始事务

文本

筛选

排序

导入

导出

name	club	shoot	goal	assist	penalty_kick
Alexande	Newcast	48	10	1	2
Antony S.	Manche	72	4	2	0
Bukayo S.	Arsenal	86	13	11	2
Cody Gak	Liverpoc	36	6	2	0
Darwin Ni	Liverpoc	84	9	3	0
Erling Ha	Manche	123	36	8	7
Gabriel Je	Arsenal	76	10	5	1
Gabriel M	Arsenal	79	15	5	0
▶ Harry Ka	Tottenh	127	28	3	5
Jack Grea	Manche	41	5	7	0
Julián Álv	Manche	42	9	0	1
Kai Haver	Chelsea	71	7	1	1
Kaoru Mi	Brightor	52	7	7	0
Marcus R	Manche	102	17	5	0
Mohame	Liverpoc	117	19	11	2
Pierre-Er	Chelsea	16	1	0	0
Raheem	Chelsea	37	6	3	0
Richarliso	Tottenh	25	1	3	0
Son Heur	Tottenh	80	10	5	0
Wout We	Manche	13	0	1	0
小元	Arsenal	(Null)	(Null)	(Null)	(Null)

备注

本次实验演示中所使用的 forward_info 是使用 player 与 forward 按照 player.name=forward.name 连接后所建立的视图，详见 6. 含有视图的查询。

5. 存储过程控制下的更新操作（18分）

说明	<p>(1分) 简要说明该操作所要完成的功能；</p> <p>(1分) 简要说明该存储过程所要完成的功能；</p> <p>(2分) 说明该操作涉及操作的表（必须包含两张或两张以上的关系表，以“表名形式”描述）</p> <p>(1分) 表连接涉及字段描述（描述方式为“表1.属性=表2.属性”）</p> <p>(2分) 该操作会修改字段（以“表名.字段名”的形式给出），以及修改规则，如新数值的计算方法、在何种条件下予以修改等；</p> <p>(6分) 实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>(5分) 如何执行该操作，按所述方法能够正常演示程序则给分。</p>	
功能描述 (1分)	<p>在俱乐部经理(Club Manager)模式下：</p> <p>当球队的教练已经为球员设定了常驻位置后，由俱乐部经理对其位置信息进行更新。</p> <p>实现：更新一个球员 player 表中的位置 position 信息后，将其从原来的 position 对应的表中删除，移动至新的 position 表中。</p>	
存储过程功能描述 (1分)	<p>1. 输入参数：球员的俱乐部 player_club，球员的名字 player_name，球员更新后的新位置 new_po；</p> <p>2. 规则：</p> <p>(1) 更新后的新位置 new_pos 必须属于 forward、midfielder、defender、goalkeeper 其中之一；</p> <p>(2) 球员的名字 player_name 必须是已经存在的球员名字；</p> <p>(3) 球员的俱乐部 player_club 必须是其实际属于的俱乐部，否则不能进行更新。</p> <p>3. 具体功能：</p> <p>根据球员的曾经位置 old_pos（定义的局部变量），先将球员从旧的所属于位置表中删除，然后再插入新属于的位置表 new_pos，插入数据包括 player.name 与 player.club。</p>	
涉及的关系表 (2分)	<p>player 与 forward(midfielder、defender、goalkeeper) 中的一个或两个表。</p>	
表连接涉及字段 (1)	<p>player.name=forward.name(midfielder、defender、goalkeeper)</p>	
更改字段 (2分)	字段	规则
	player.position	更新后的新位置 new_pos 必须属于 forward、midfielder、defender、goalkeeper 其中之一；

更新
代码
(3
分)

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    Connection conn=null;
    PreparedStatement ps=null;
    CallableStatement cs=null;
    ResultSet rs=null;
    try {
        conn=JDBCUtils.getConnection();
        String c1=ClubTextField3.getText();
        ClubTextField3.setText("");
        String na=NameTextField2.getText();
        NameTextField2.setText("");
        String po=PositionTextField.getText();
        PositionTextField.setText("");
        String ta=TransferTextField.getText();
        TransferTextField.setText("");

        String sql1="CALL UpdatePlayerPosition1(?,?,?)";
        cs=conn.prepareCall(sql1);
        cs.setString(1, c1);
        cs.setString(2, na);
        cs.setString(3, po);

        cs.execute();

        JOptionPane.showMessageDialog(null,"Update successfully!","Update successfully!",JOptionPane.INFORMATION_MESSAGE);

        String sql2="UPDATE player SET transfer_value=? WHERE club=? AND name=?";
        ps=conn.prepareStatement(sql2);
        ps.setString(1, ta);
        ps.setString(2, c1);
        ps.setString(3, na);

        ps.executeUpdate();

    } catch (Exception e1) {
        // TODO Auto-generated catch block
        JOptionPane.showMessageDialog(null,"Violated Procedure:UpdatePlayerPosition","Update failed!",JOptionPane.ERROR_MESSAGE);
        e1.printStackTrace();
    }finally{//以防在conn close之前出现异常无法关闭
        JDBCUtils.close(rs, cs, conn);
        JDBCUtils.close(ps, null);
    }
    showPlayer();
}
```

创建
存储
过程
源码
(3
分)



```
localhost_3306 | premierleague | 运行 | 停止 | 解释
1 DROP PROCEDURE IF EXISTS UpdatePlayerPosition1
2 delimiter //
3 CREATE DEFINER='root'@'localhost' PROCEDURE `UpdatePlayerPosition1`(IN player_club VARCHAR(255),IN player_name VARCHAR(255)
4 )
5 BEGIN
6 DECLARE old_pos VARCHAR(255);
7 DECLARE c1 VARCHAR(255);# 俱乐部信息
8 IF new_pos IN ('forward','midfielder','defender','goalkeeper') AND player_name IN (SELECT `name` FROM player)
9 AND player_club IN (SELECT club FROM player WHERE `name`=player_name) THEN
10 SELECT position INTO old_pos FROM player WHERE `name`=player_name;
11 SELECT club INTO c1 FROM player WHERE `name`=player_name;
12 UPDATE player SET position=new_pos WHERE `name`=player_name;
```

	<pre> 12 IF old_pos='forward' THEN 13 DELETE FROM forward WHERE `name`=player_name; 14 END IF; 15 IF old_pos='midfielder' THEN 16 DELETE FROM midfielder WHERE `name`=player_name; 17 END IF; 18 IF old_pos='defender' THEN 19 DELETE FROM defender WHERE `name`=player_name; 20 END IF; 21 IF old_pos='goalkeeper' THEN 22 DELETE FROM goalkeeper WHERE `name`=player_name; 23 END IF; 24 IF new_pos='forward' THEN 25 INSERT INTO forward(`name`,club) VALUES(player_name,c1); 26 END IF; 27 IF new_pos='midfielder' THEN 28 INSERT INTO midfielder(`name`,club) VALUES(player_name,c1); 29 END IF; 30 IF new_pos='defender' THEN 31 INSERT INTO defender(`name`,club) VALUES(player_name,c1); 32 END IF; 33 IF new_pos='goalkeeper' THEN 34 INSERT INTO goalkeeper(`name`,club) VALUES(player_name,c1); 35 END IF; 36 ELSE 37 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Violated PROCEDURE:UpdatePlayerPosition1'; 38 END IF; 39 END; // 40 delimiter ; </pre>
<p>存储过程执行源码 (1分)</p>	<p>(截屏)</p> <pre> public void actionPerformed(ActionEvent e) { // TODO Auto-generated method stub Connection conn=null; PreparedStatement ps=null; CallableStatement cs=null; ResultSet rs=null; try { conn=JDBCUtils.getConnection(); String c1=ClubTextField3.getText(); ClubTextField3.setText(""); String na=NameTextField2.getText(); NameTextField2.setText(""); String po=PosTextField.getText(); PosTextField.setText(""); String ta=TransferTextField.getText(); TransferTextField.setText(""); String sql1="CALL UpdatePlayerPosition(?,?,?);"; cs=conn.prepareCall(sql1); cs.setString(1, c1); cs.setString(2, na); cs.setString(3, po); cs.execute(); JOptionPane.showMessageDialog(null,"Update successfully!","Update successfully!",JOptionPane.INFORMATION_MESSAGE); String sql2="UPDATE player SET transfer_value=? WHERE club=? AND name=?"; ps=conn.prepareStatement(sql2); ps.setString(1, ta); ps.setString(2, c1); ps.setString(3, na); ps.executeUpdate(); } catch (Exception e1) { // TODO Auto-generated catch block JOptionPane.showMessageDialog(null,"Violated Procedure:UpdatePlayerPosition","Update failed!",JOptionPane.ERROR_MESSAGE); e1.printStackTrace(); }finally{//以防在conn close之前出现异常无法关闭 JDBCUtils.close(rs, cs, conn); JDBCUtils.close(ps, null); } showPlayer(); } </pre>
<p>程序演示</p>	<p>1. 更新在含有触发器的添加操作时添加的数据“小元”的位置，其旧位置为 forward，俱乐部为 Arsenal。因此其在对应的 forward 表中，如下图所示：（用前端 forward_info 显</p>

(2 分)
示同理)

对象 forward @premierleague (l...

开始事务 文本 筛选 排序 导入 导出

name	club	shoot	goal	assist	penalty_kick
Alexande Newcast		48	10	1	2
Antony S.Manche		72	4	2	0
Bukayo S.Arsenal		86	13	11	2
Cody Gak Liverpoc		36	6	2	0
Darwin Ni Liverpoc		84	9	3	0
Erling Ha:Manche		123	36	8	7
Gabriel Je Arsenal		76	10	5	1
Gabriel M Arsenal		79	15	5	0
Harry Kar Tottenh		127	28	3	5
Jack Grea Manche		41	5	7	0
Julián Álv Manche		42	9	0	1
Kai Haver Chelsea		71	7	1	1
Kaoru Mi Brightor		52	7	7	0
Marcus R Manche		102	17	5	0
Mohame: Liverpoc		117	19	11	2
Pierre-Er Chelsea		16	1	0	0
Raheem S Chelsea		37	6	3	0
Richarlisc Tottenh		25	1	3	0
Son Heur Tottenh		80	10	5	0
Wout We Manche		13	0	1	0
小元	Arsenal	(Null)	(Null)	(Null)	(Null)

Premier League Database:The Football Association

name	club	age	nationality	transfer_value	shoot	goal	assist	penalty_kick
小元	Arsenal	20	China	-1	0	0	0	0
Wout Weghorst	Manchester Uni...	30	Netherlands	14000000	13	0	1	0
Son Heung-Min	Tottenham Hots...	30	Korea	60000000	80	10	5	0
Richarlison	Tottenham Hots...	26	Brazil	55000000	25	1	3	0
Raheem Sterling	Chelsea	28	England	60000000	37	6	3	0
Pierre-Emerick ...	Chelsea	33	Gabonese	70000000	16	1	0	0
Mohamed Salah	Liverpool	30	Egypt	70000000	117	19	11	2
Marcus Rashford	Manchester Uni...	25	England	80000000	102	17	5	0
Kaoru Mitoma	Brighton & Hove...	26	Japan	22000000	52	7	7	0
Kai Havertz	Chelsea	23	Germany	60000000	71	7	1	1
Julián Alvarez	Manchester City	23	Argentina	50000000	42	9	0	1
Jack Grealish	Manchester City	27	England	70000000	41	5	7	0
Harry Kane	Tottenham Hots...	29	England	90000000	127	28	3	5
Gabriel Martinelli	Arsenal	21	Brazil	70000000	79	15	5	0
Gabriel Jesus	Arsenal	26	Brazil	75000000	76	10	5	1
Erling Haaland	Manchester City	22	Norway	170000000	123	36	8	7
Darwin Núñez	Liverpool	23	Uruguay	70000000	84	9	3	0
Cody Gakpo	Liverpool	24	Netherlands	60000000	36	6	2	0
Bukayo Saka	Arsenal	21	England	110000000	86	13	11	2
Antony Santos	Manchester Uni...	23	Brazil	70000000	72	4	2	0
Alexander Isak	Newcastle United	23	Sverige	50000000	48	10	1	2

for_info: Show mid_info: Show def_info: Show goa_info: Show

2.在俱乐部经理模式下，输入“小元”所属于的俱乐部 Arsenal，更新球员“小元”的位置为 goalkeeper。因此满足存储过程条件。点击 Update 按钮。可以看到弹出一个对话框提示” Update successfully! ”

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
小元	20	China	forward	-1	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

club: name: age: nationality:

club: name: club:

club: Arsenal name: 小元 pos: goalkeeper value: 11

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
小元	20	China	forward	-1	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

Update successfully!

Update successfully!

确定

club: name: age: nationality:

club: name: club:

club: name: pos: value:

3.

(1)此时 player 表, 显示所有球员, 可以看到其位置与身价已经进行更新。

Premier League Database:Manager

name	age	nationality	position	transfer_value	club
小元	20	China	goalkeeper	11	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur

2, 此时的 forward 表中不再存在“小元”的球员信息：如下表所示——（在前端使用 forward_info 表也可以显示验证）

对象 forward @premierleague (l...

开始事务 文本 筛选 排序 导入 导出

name	club	shoot	goal	assist	penalty_kick
Alexander	Newcast	48	10	1	2
Antony S.	Manche	72	4	2	0
Bukayo S.	Arsenal	86	13	11	2
Cody Gak	Liverpoc	36	6	2	0
Darwin N.	Liverpoc	84	9	3	0
Erling Ha.	Manche	123	36	8	7
Gabriel J.	Arsenal	76	10	5	1
Gabriel M.	Arsenal	79	15	5	0
Harry Kar	Tottenh	127	28	3	5
Jack Grea	Manche	41	5	7	0
Julián Álv	Manche	42	9	0	1
Kai Haver	Chelsea	71	7	1	1
Kaoru Mi	Brighton	52	7	7	0
Marcus R.	Manche	102	17	5	0
Mohamed	Liverpoc	117	19	11	2
Pierre-Er	Chelsea	16	1	0	0
Raheem S.	Chelsea	37	6	3	0
Richarlison	Tottenh	25	1	3	0
Son Heun	Tottenh	80	10	5	0
Wout We	Manche	13	0	1	0

Premier League Database: The Football Association

name	club	age	nationality	transfer_value	shoot	goal	assist	penalty_kick
Wout Weghorst	Manchester Uni...	30	Netherlands	14000000	13	0	1	0
Son Heung-Min	Tottenham Hots...	30	Korea	60000000	80	10	5	0
Richarlison	Tottenham Hots...	26	Brazil	55000000	25	1	3	0
Raheem Sterling	Chelsea	28	England	60000000	37	6	3	0
Pierre-Emerick ...	Chelsea	33	Gabonese	7000000	16	1	0	0
Mohamed Salah	Liverpool	30	Egypt	70000000	117	19	11	2
Marcus Rashford	Manchester Uni...	25	England	80000000	102	17	5	0
Kaoru Mitoma	Brighton & Hove...	26	Japan	22000000	52	7	7	0
Kai Havertz	Chelsea	23	Germany	60000000	71	7	1	1
Julián Álvarez	Manchester City	23	Argentina	50000000	42	9	0	1
Jack Grealish	Manchester City	27	England	70000000	41	5	7	0
Harry Kane	Tottenham Hots...	29	England	90000000	127	28	3	5
Gabriel Martinelli	Arsenal	21	Brazil	70000000	79	15	5	0
Gabriel Jesus	Arsenal	26	Brazil	75000000	76	10	5	1
Erling Haaland	Manchester City	22	Norway	170000000	123	36	8	7
Darwin Núñez	Liverpool	23	Uruguay	70000000	84	9	3	0
Cody Gakpo	Liverpool	24	Netherlands	60000000	36	6	2	0
Bukayo Saka	Arsenal	21	England	110000000	86	13	11	2
Antony Santos	Manchester Uni...	23	Brazil	70000000	72	4	2	0
Alexander Isak	Newcastle United	23	Sverige	50000000	48	10	1	2

for_info: mid_info: def_info: goa_info:

3, 此时的 goalkeeper 表中出现了“小元”的球员信息：如下表所示——（在前端使用 goalkeeper_info 也可以显示并验证）

对象

name	club	save	clean_sheets
Aaron Ra	Chelsea	95	13
Alisson B	Liverpoc	108	14
David de	Manche	99	17
Ederson I	Manche	43	11
Emiliano I	Aston Vi	95	11
Hugo Llo	Tottenh	79	7
Kepa Arri	Chelsea	88	9
小元	Arsenal	(Null)	(Null)

Premier League Database:The Football Association

name	club	age	nationality	transfer_value	save	clean_sheets
小元	Arsenal	20	China	11	0	0
Kepa Arrizabalaga	Chelsea	26	Spain	15000000	86	9
Hugo Lloris	Tottenham Hotspur	26	France	4000000	79	7
Emiliano Martínez	Aston Villa	30	Argentina	28000000	95	11
Ederson Moraes	Manchester City	29	Brazil	45000000	43	11
David de Gea	Manchester United	32	Spain	15000000	99	17
Alisson Becker	Liverpool	30	Brazil	45000000	108	14
Aaron Ramsdale	Chelsea	25	England	38000000	95	13

for_info: mid_info: def_info: goa_info:

4.不改变球员的位置，而是直接改变身价：

Premier League Database:Manager


name	age	nationality	position	transfer_value	club
小元	20	China	goalkeeper	11	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

club: name: age: nationality:

club: name: club:

club: Arsenal name: 小元 pos: goalkeeper value: 999

name	age	nationality	position	transfer_value	club
小元	20	China	goalkeeper	11	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

Update successfully!
 Update successfully!
 确定

club: name: age: nationality:

club: name: club:

club: name: pos: value:

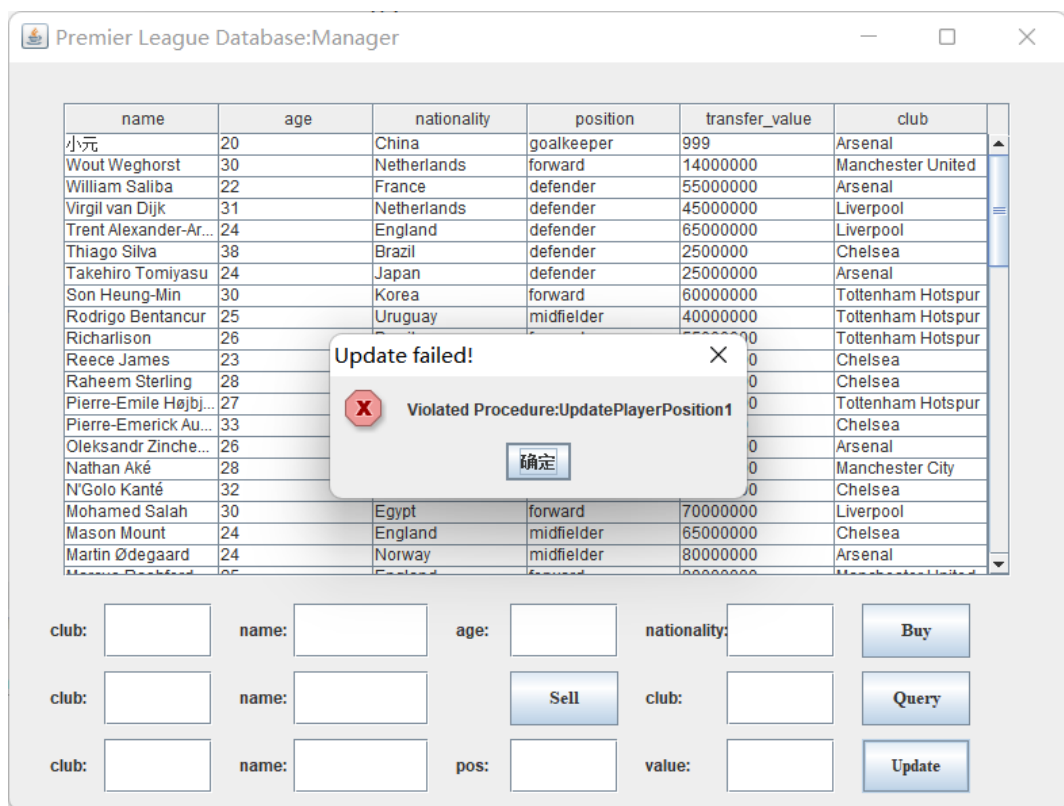
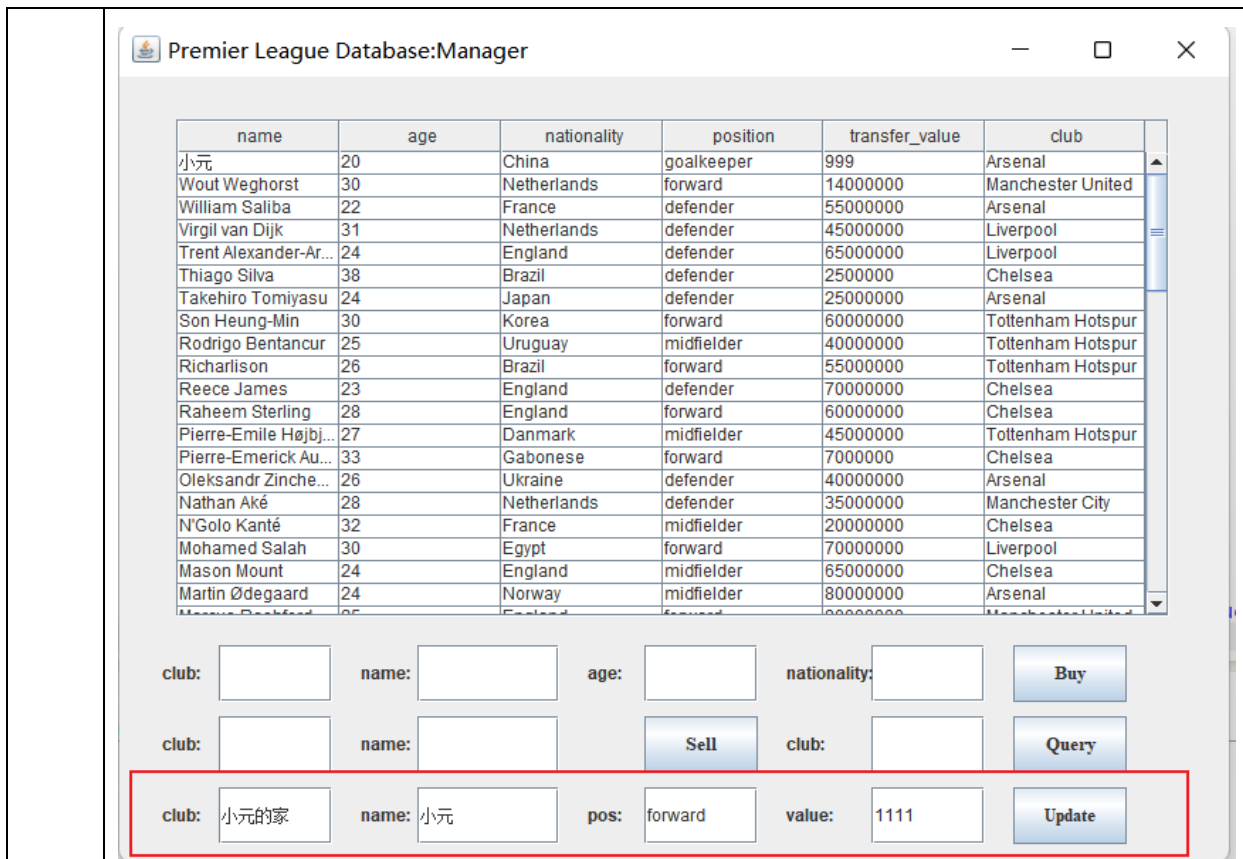
Premier League Database:Manager

name	age	nationality	position	transfer_value	club
小元	20	China	goalkeeper	999	Arsenal
Wout Weghorst	30	Netherlands	forward	14000000	Manchester United
William Saliba	22	France	defender	55000000	Arsenal
Virgil van Dijk	31	Netherlands	defender	45000000	Liverpool
Trent Alexander-Ar...	24	England	defender	65000000	Liverpool
Thiago Silva	38	Brazil	defender	25000000	Chelsea
Takehiro Tomiyasu	24	Japan	defender	25000000	Arsenal
Son Heung-Min	30	Korea	forward	60000000	Tottenham Hotspur
Rodrigo Bentancur	25	Uruguay	midfielder	40000000	Tottenham Hotspur
Richarlison	26	Brazil	forward	55000000	Tottenham Hotspur
Reece James	23	England	defender	70000000	Chelsea
Raheem Sterling	28	England	forward	60000000	Chelsea
Pierre-Emile Højbj...	27	Danmark	midfielder	45000000	Tottenham Hotspur
Pierre-Emerick Au...	33	Gabonese	forward	70000000	Chelsea
Oleksandr Zinche...	26	Ukraine	defender	40000000	Arsenal
Nathan Aké	28	Netherlands	defender	35000000	Manchester City
N'Golo Kanté	32	France	midfielder	20000000	Chelsea
Mohamed Salah	30	Egypt	forward	70000000	Liverpool
Mason Mount	24	England	midfielder	65000000	Chelsea
Martin Ødegaard	24	Norway	midfielder	80000000	Arsenal

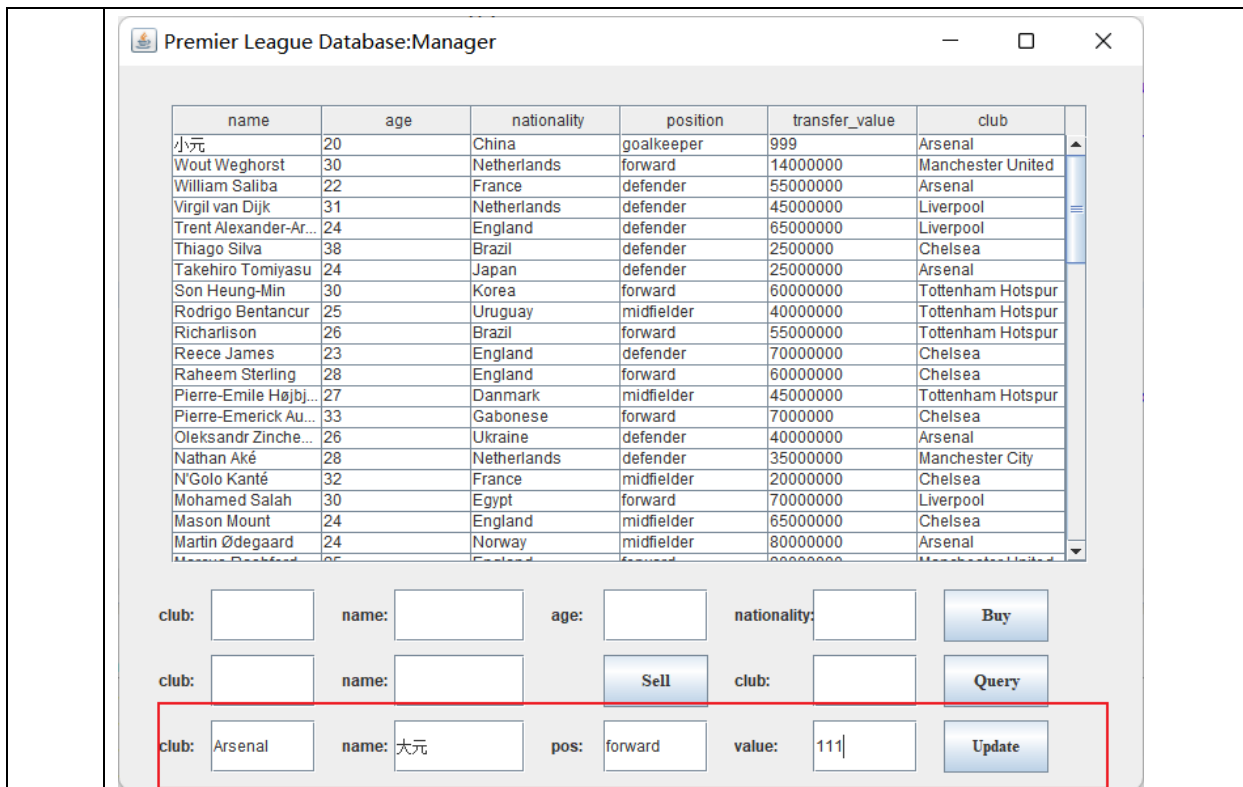
证明不违背存储过程可以成功完成更新。（不论是改变位置，还是不改变位置，直接改变身价）

程序
演示
(2
分)

1. **违反存储过程条件一**：球员的俱乐部必须是其实际属于的俱乐部，否则不能进行更新：
（输入不存在的俱乐部“小元的家”，会报错，弹出对话框提示“Violated Procedure:Update PlayerPosition1”）

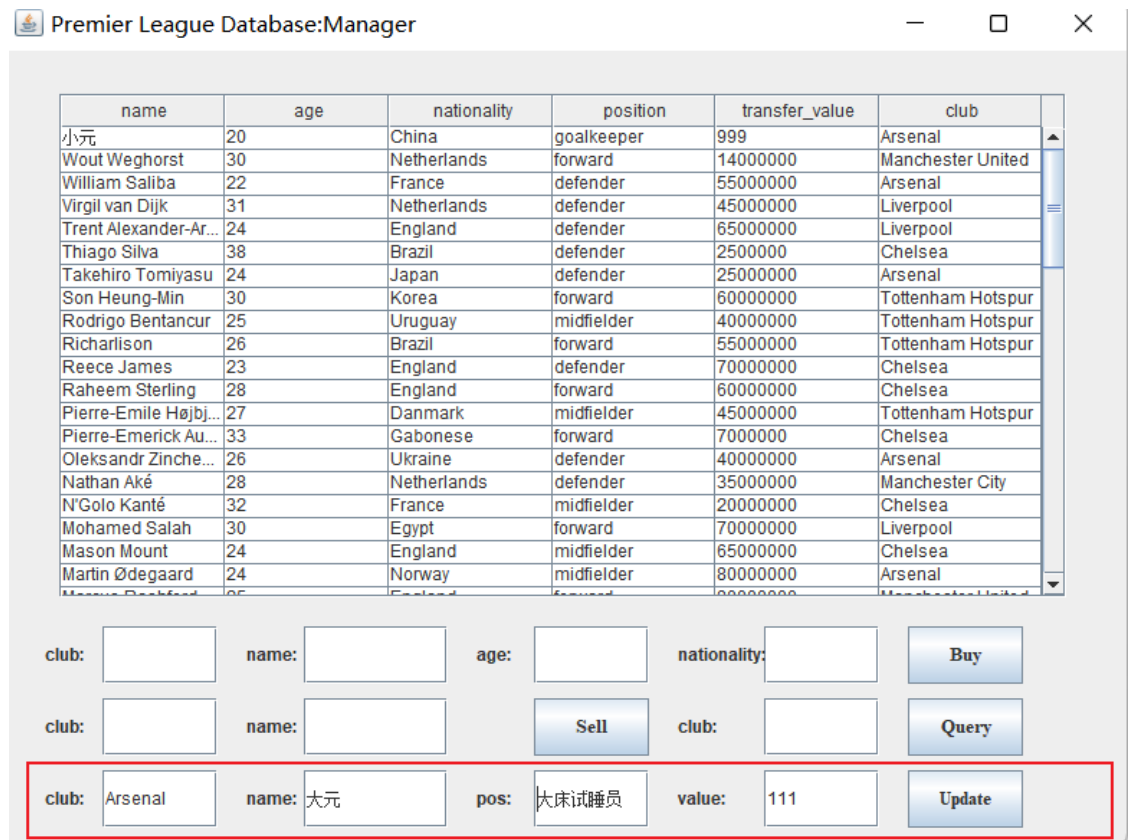


2. 违背存储过程条件二：球员名字必须是已经存在的球员名字——同理报错。（对话框错误提示如上图，不再举图显示）



3. 违背存储过程条件三：更新后的新位置 new_pos 必须属于 forward、midfielder、defender、goalkeeper 其中之一——同理报错。（对话框错误提示如上图，不再举图显示）

4.



备注 1. 之所以在定义存储过程时，先删除旧表数据再插入新表数据是因为：在俱乐部经理模式

下，更新球员位置和球员身价只能同时进行，因此需要保证即使不更新球员位置，也能更新球员身价。若先插入新表再删除旧表由于 name 是四个位置表的主键，因此会出现重复主键无法插入的问题。而先删除再插入即可解决该问题。

2. 本次程序演示所使用的 forward_info 和 goalkeeper_info 是将 player 与 forward (或 goalkeeper)表按照主键连接起来建立的视图。在上面的演示中，因为更新球员“小元”的位置数据后，“小元”还在表 player 中，但 forward_info 中不在，goalkeeper_info 中在。因为也侧面验证“小元”从 forward 表中删除到了 goalkeeper 表中。

6. 含有视图的查询操作 (15 分)

说明	<p>(1 分) 简要说明该操作所要完成的功能;</p> <p>(1 分) 简要说明建立的该视图的功能;</p> <p>(2 分) 简要说明该操作涉及的关系数据表 (以“表名”的形式给出)</p> <p>(1 分) 简要说明表连接涉及的字段 (以“表 1. 属性=表 2. 属性”)</p> <p>(6 分) 实现该操作的关键代码 (高级语言、SQL), 截图即可;</p> <p>(4 分) 如何执行该操作, 按所述方法能够正常演示程序则给分。</p>
操作功能描述 (1 分)	<p>在英足总模式 (The Football Association) 下:</p> <p>查询前锋的所有详细信息 (包含其 name、club、transfer_value、goal、assist 等)</p> <p>查询门将的所有详细信息 (包含其 name、club、transfer_value、save、clean_sheets 等)</p>
视图功能描述 (1 分)	<p>1. forward_info: 显示前锋的所有详细信息 (包含其 name、club、transfer_value、goal、assist 等)</p> <p>2. Goalkeeper_info: 显示门将的所有详细信息 (包含其 name、club、transfer_value、save、clean_sheets 等)</p>
涉及的关系表 (2 分)	<p>player、forward、goalkeeper</p>
表连接字段 (1 分)	<p>player.name=forward.name(goalkeeper.name)</p>
创建视图代码	<p>(截屏)</p> <p>1. forward_info:</p>

<p>码 (3 分)</p>	<div data-bbox="272 197 1410 515"> <p>对象 Create_View @premierleag...</p> <p>保存 查询创建工具 美化 SQL () 代码段 文本 导出结果</p> <p>localhost_3306 premierleague 运行 停止 解释</p> <pre> 1 DROP VIEW IF EXISTS forward_info; 2 CREATE VIEW forward_info 3 AS SELECT player.`name`,player.club,player.age,player.nationality, 4 player.transfer_value,forward.shoot,forward.goal,forward.assist,forward.penalty_kick 5 FROM player,forward 6 WHERE player.`name`=forward.`name`;</pre> </div> <p>2. goalkeeper_info:</p> <div data-bbox="272 571 1410 936"> <p>对象 Create_View @premierleag... forward_info @premierleag... goalkeeper_info @premierl...</p> <p>保存 查询创建工具 美化 SQL () 代码段 文本 导出结果</p> <p>localhost_3306 premierleague 运行 停止 解释</p> <pre> 1 DROP VIEW IF EXISTS goalkeeper_info; 2 CREATE VIEW goalkeeper_info 3 AS SELECT player.`name`,player.club,player.age,player.nationality, 4 player.transfer_value,goalkeeper.save,goalkeeper.clean_sheets 5 FROM player,goalkeeper 6 WHERE player.`name`=goalkeeper.`name`;</pre> </div>
<p>查询 代码 (3 分)</p>	<p>(截屏)</p> <p>1. forward_info:</p> <pre> @Override public void actionPerformed(ActionEvent e) { // TODO Auto-generated method stub Integer dataSize=0; contextList.clear(); Vector<Object> newTitleList = new Vector<>(); newTitleList.add("name"); newTitleList.add("club"); newTitleList.add("age"); newTitleList.add("nationality"); newTitleList.add("transfer_value"); newTitleList.add("shoot"); newTitleList.add("goal"); newTitleList.add("assist"); newTitleList.add("penalty_kick"); model.setColumnIdentifiers(newTitleList); Connection conn=null; Statement stat=null; ResultSet rs=null; //获取连接 try { conn=JDBCUtils.getConnection(); String sql="SELECT * FROM forward_info;"; stat=conn.createStatement(); rs=stat.executeQuery(sql); Object showData[][] =new Object[200][200]; while(rs.next()){</pre>

```

String na=rs.getString("name");
showData[dataSize][0]=na;
String cl=rs.getString("club");
showData[dataSize][1]=cl;
Integer ag=rs.getInt("age");
showData[dataSize][2]=ag;
String nat=rs.getString("nationality");
showData[dataSize][3]=nat;
Integer tra=rs.getInt("transfer_value");
showData[dataSize][4]=tra;
Integer sh=rs.getInt("shoot");
showData[dataSize][5]=sh;
Integer go=rs.getInt("goal");
showData[dataSize][6]=go;
Integer as=rs.getInt("assist");
showData[dataSize][7]=as;
Integer pen=rs.getInt("penalty_kick");
showData[dataSize][8]=pen;
dataSize++;
}
Integer dataSize_=dataSize-1;
while(dataSize_>=0){
    Vector<Object> list = new Vector<>();
    int j=0;
    while(j<=8){
        list.addElement(showData[dataSize_][j]);
        j++;
    }
    //将数据放入数据容器中
    model.addRow(list);
    dataSize_--;
}
} catch (Exception e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}finally{//以防在conn close之前出现异常无法关闭
    JDBCUtils.close(rs, stat, conn);
}
}

```

2. goalkeeper_info:

```
@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    Integer dataSize=0;
    contextList.clear();
    Vector<Object> newTitleList = new Vector<>();
    newTitleList.add("name");
    newTitleList.add("club");
    newTitleList.add("age");
    newTitleList.add("nationality");
    newTitleList.add("transfer_value");
    newTitleList.add("save");
    newTitleList.add("clean_sheets");
    model.setColumnIdentifiers(newTitleList);

    Connection conn=null;
    Statement stat=null;
    ResultSet rs=null;
    //获取连接
    try {
        conn=JDBCUtils.getConnection();
        String sql="SELECT * FROM goalkeeper_info";
        stat=conn.createStatement();
        rs=stat.executeQuery(sql);
        Object showData[][] =new Object[200][200];
        while(rs.next()){
            String na=rs.getString("name");
            showData[dataSize][0]=na;
            String cl=rs.getString("club");
            showData[dataSize][1]=cl;
            Integer ag=rs.getInt("age");
            showData[dataSize][2]=ag;
            String nat=rs.getString("nationality");
            showData[dataSize][3]=nat;
            Integer tra=rs.getInt("transfer_value");
            showData[dataSize][4]=tra;
            Integer sa=rs.getInt("save");
            showData[dataSize][5]=sa;
            Integer cle=rs.getInt("clean_sheets");
            showData[dataSize][6]=cle;
            dataSize++;
        }
    }
```



```

Integer dataSize_ = dataSize - 1;
while(dataSize_ >= 0){
    Vector<Object> list = new Vector<>();
    int j = 0;
    while(j <= 6){
        list.addElement(showData[dataSize_][j]);
        j++;
    }
    //将数据放入数据容器中
    model.addRow(list);
    dataSize_--;
}
} catch (Exception e1) {
    e1.printStackTrace();
} finally{//以防在conn close之前出现异常无法关闭
    JDBCUtils.close(rs, stat, conn);
}
}

```

1. forward_info:

Premier League Database: The Football Association

name	club	age	nationality	transfer_value	shoot	goal	assist	penalty_kick
Wout Weghorst	Manchester Uni...	30	Netherlands	14000000	13	0	1	0
Son Heung-Min	Tottenham Hots...	30	Korea	60000000	80	10	5	0
Richarlison	Tottenham Hots...	26	Brazil	55000000	25	1	3	0
Raheem Sterling	Chelsea	28	England	60000000	37	6	3	0
Pierre-Emerick ...	Chelsea	33	Gabonese	7000000	16	1	0	0
Mohamed Salah	Liverpool	30	Egypt	70000000	117	19	11	2
Marcus Rashford	Manchester Uni...	25	England	80000000	102	17	5	0
Kaoru Mitoma	Brighton & Hove...	26	Japan	22000000	52	7	7	0
Kai Havertz	Chelsea	23	Germany	60000000	71	7	1	1
Julián Alvarez	Manchester City	23	Argentina	50000000	42	9	0	1
Jack Grealish	Manchester City	27	England	70000000	41	5	7	0
Harry Kane	Tottenham Hots...	29	England	90000000	127	28	3	5
Gabriel Martinelli	Arsenal	21	Brazil	70000000	79	15	5	0
Gabriel Jesus	Arsenal	26	Brazil	75000000	76	10	5	1
Erling Haaland	Manchester City	22	Norway	170000000	123	36	8	7
Darwin Núñez	Liverpool	23	Uruguay	70000000	84	9	3	0
Cody Gakpo	Liverpool	24	Netherlands	60000000	36	6	2	0
Bukayo Saka	Arsenal	21	England	110000000	86	13	11	2
Antony Santos	Manchester Uni...	23	Brazil	70000000	72	4	2	0
Alexander Isak	Newcastle United	23	Sverige	50000000	48	10	1	2

程序演示 (4分)

2. goalkeeper_info:

Premier League Database: The Football Association

name	club	age	nationality	transfer_value	save	clean_sheets
小元	Arsenal	20	China	999	0	0
Kepa Arrizabalaga	Chelsea	28	Spain	15000000	88	9
Hugo Lloris	Tottenham Hotspur	26	France	40000000	79	7
Emiliano Martínez	Aston Villa	30	Argentina	28000000	95	11
Ederson Moraes	Manchester City	29	Brazil	45000000	43	11
David de Gea	Manchester United	32	Spain	15000000	99	17
Alisson Becker	Liverpool	30	Brazil	45000000	108	14
Aaron Ramsdale	Chelsea	25	England	38000000	95	13

备注

除了报告要求的视图外，我还创建了其它视图。比如 midfielder_info, defender_info, 但由于实际数据没有输入进去的原因，并没有在前端显示。
除此之外还有别的视图，虽然不满足报告中的要求，但是作为一个热爱足球的人，我觉得以

下视图具有别样的意义:

1. 进球榜 (top goal scorers):

(1) Navicat 中创建代码:

```
localhost_3306 premierleague ▶ 运行 ◻ 停止 解释
1 CREATE VIEW top_goal_scorers
2 AS SELECT f1.`name`,f1.club,f1.goal,
3 (SELECT COUNT(*) FROM forward f2 WHERE f2.goal>f1.goal)+1 AS ranking
4 FROM forward f1
5 ORDER BY ranking;
6
7
```

(2) Navicat 中显示:

name	club	goal	ranking
▶ Erling Ha	Manche	36	1
Harry Kar	Tottenh	28	2
Mohame	Liverpoc	19	3
Marcus R	Manche	17	4
Gabriel M	Arsenal	15	5
Bukayo S	Arsenal	13	6
Alexande	Newcast	10	7
Gabriel J	Arsenal	10	7
Son Heur	Tottenh	10	7
Darwin N	Liverpoc	9	10
Julián Á	Manche	9	10
Kai Haver	Chelsea	7	12
Kaoru Mi	Brighton	7	12
Cody Gak	Liverpoc	6	14
Raheem S	Chelsea	6	14
Jack Grea	Manche	5	16
Antony S	Manche	4	17
Pierre-Er	Chelsea	1	18
Richarlisc	Tottenh	1	18
Wout We	Manche	0	20

(3) GUI 界面中的教练模式下的展示:

Premier League Database:Coach

name	club	goal	ranking
Erling Haaland	Manchester City	36	1
Harry Kane	Tottenham Hotspur	28	2
Mohamed Salah	Liverpool	19	3
Marcus Rashford	Manchester United	17	4
Gabriel Martinelli	Arsenal	15	5
Bukayo Saka	Arsenal	13	6
Alexander Isak	Newcastle United	10	7
Gabriel Jesus	Arsenal	10	7
Son Heung-Min	Tottenham Hotspur	10	7
Darwin Núñez	Liverpool	9	10
Julián Álvarez	Manchester City	9	10
Kai Havertz	Chelsea	7	12
Kaoru Mitoma	Brighton & Hove Albion	7	12
Cody Gakpo	Liverpool	6	14
Raheem Sterling	Chelsea	6	14
Jack Grealish	Manchester City	5	16
Antony Santos	Manchester United	4	17
Pierre-Emerick Aubameyang	Chelsea	1	18
Richarlison	Tottenham Hotspur	1	18
Wout Weghorst	Manchester United	0	20

Top Goal Scorers:

2. 助攻榜 (top assist provider):

(1) Navicat 中创建代码:

```
localhost_3306 premierleague 运行 停止 解释
1 CREATE VIEW top_assist_providers
2 AS SELECT m1.`name`,m1.club,m1.assist,
3 (SELECT COUNT(*) FROM midfielder m2 WHERE m2.assist>m1.assist)+1 AS ranking
4 FROM midfielder m1
5 ORDER BY ranking,m1.`name`;
```

(2) Navicat 中显示:

name	club	assist	ranking
▶ Kevin De Bruyne	Manchester City	16	1
Bruno Fernandes	Manchester United	8	2
Christian Erikson	Manchester United	8	2
Ivan Perišić	Tottenham Hotspur	8	2
Granit Xhaka	Arsenal	7	5
Martin Ødegaard	Arsenal	7	5
Bernardo Silva	Manchester City	5	7
Bruno Guimarães	Newcastle United	5	7
Pierre-Emile Højbjerg	Tottenham Hotspur	5	7
İlkay Gündoğan	Manchester City	4	10
Carlos Casimiro	Manchester United	3	11
Alexis Mac Allister	Brighton & Hove Albion	2	12
Enzo Fernández	Chelsea	2	12
Harvey Elliott	Liverpool	2	12
Jordan Henderson	Liverpool	2	12
Mason Mount	Chelsea	2	12
Rodrigo Bentancur	Tottenham Hotspur	2	12
Fabio Tavares(Fabinho)	Liverpool	1	18
Jorge Filho(Jorginho)	Arsenal	1	18
N'Golo Kanté	Chelsea	1	18

3. 零封榜 (clean sheets table):

(1) Navicat 中创建代码:

```
localhost_3306 premierleague ▶ 运行 ◻ 停止 解释
1 CREATE VIEW clean_sheets_table
2 AS SELECT g1.`name`,g1.club,g1.clean_sheets,
3 (SELECT COUNT(*) FROM goalkeeper g2 WHERE g2.clean_sheets>g1.clean_sheets)+1 AS ranking
4 FROM goalkeeper g1
5 ORDER BY ranking,g1.`name`;
```

(2) Navicat 中显示:

name	club	clean_sheets	ranking
▶ David de Gea	Manchester United	17	1
Alisson Becker	Liverpool	14	2
Aaron Ramsdale	Chelsea	13	3
Ederson Moraes	Manchester City	11	4
Emiliano Martínez	Aston Villa	11	4
Kepa Arrizabalaga	Chelsea	9	6
Hugo Lloris	Tottenham Hotspur	7	7

4. 球队积分榜: (League table):

name	points	rank
▶ Manchester City	88	1
Arsenal	81	2
Newcastle United	69	3
Manchester United	69	4
Liverpool	66	5
Brighton & Hove Alb	61	6
Aston Villa	58	7
Tottenham Hotspur	57	8
Chelsea	43	12

（由于我并没有插入全部的英超联赛球队，因此视图中的 rank 属性是真实的 20 家俱乐部的排名，而不是这几个球队的部分排名）