数据库工程作业

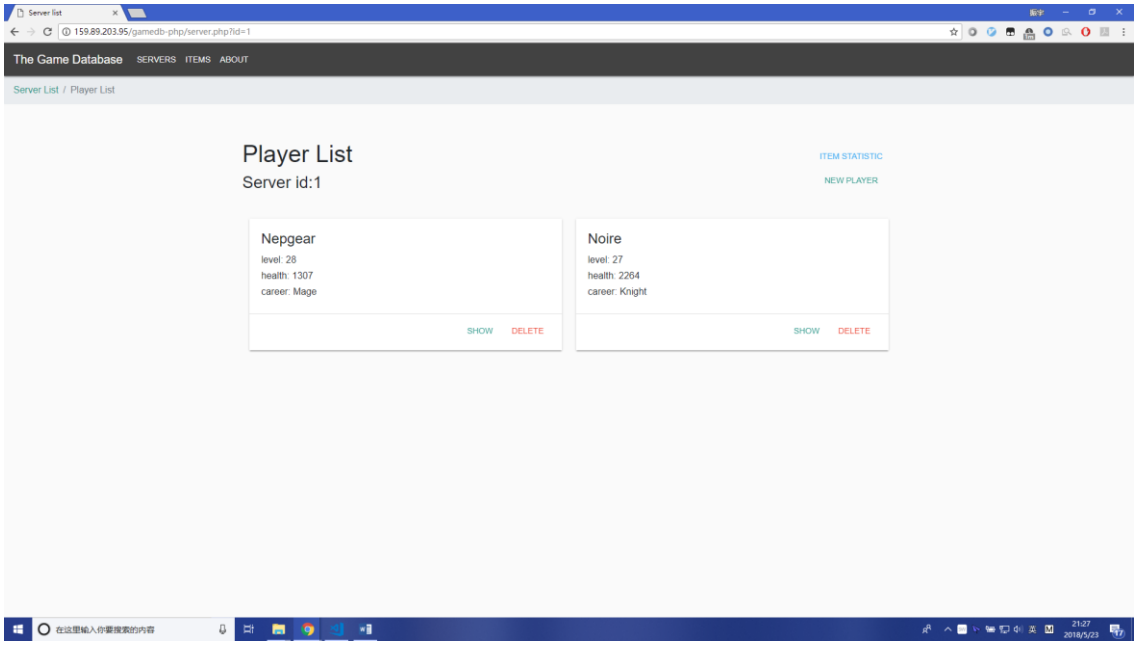
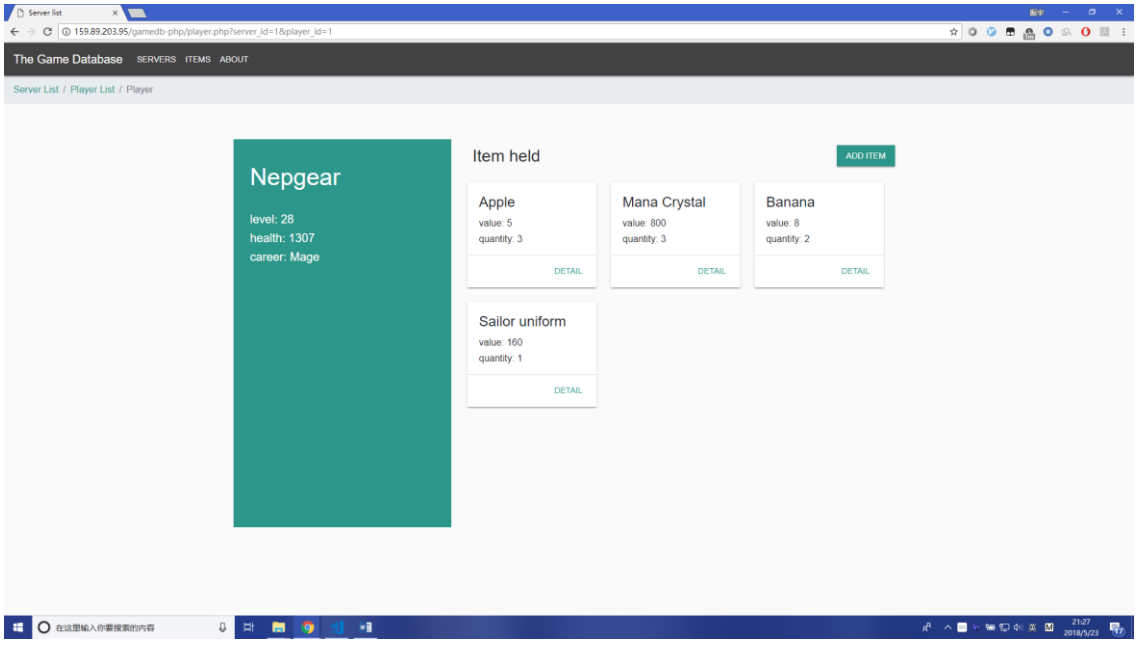
要求：

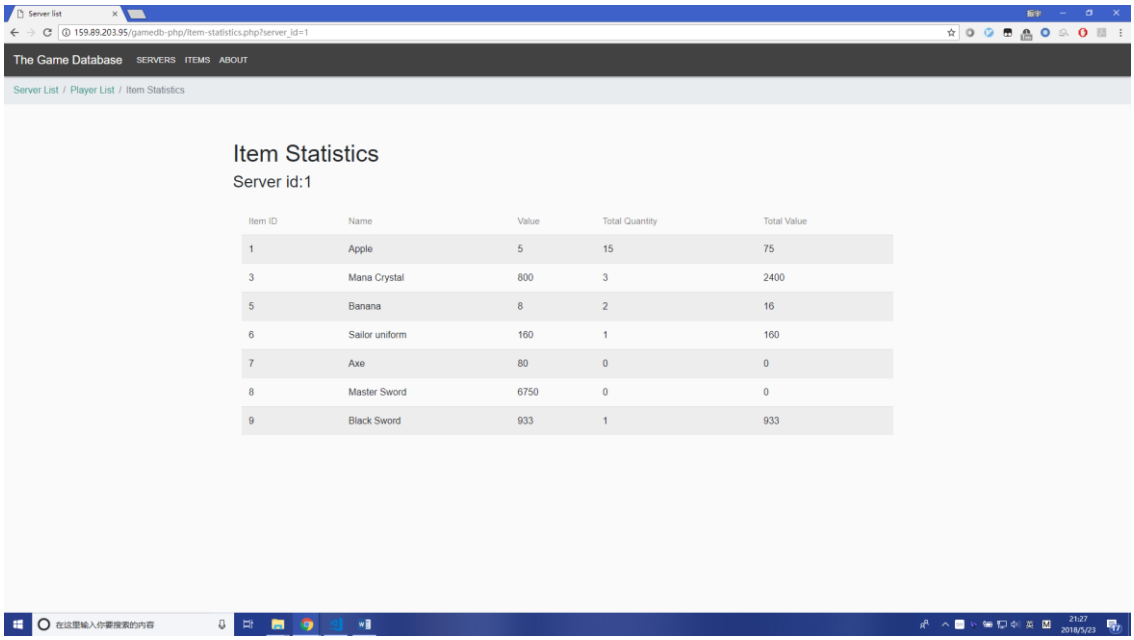
1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
2. 开发模式（B/S 或 C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle 等），不要使用桌面数据库。
3. 报告中所列举的四种操作，每种操作举一个例子即可。
4. 作业成绩按照报告中的标准评分，程序只实现报告中涉及的部分即可。

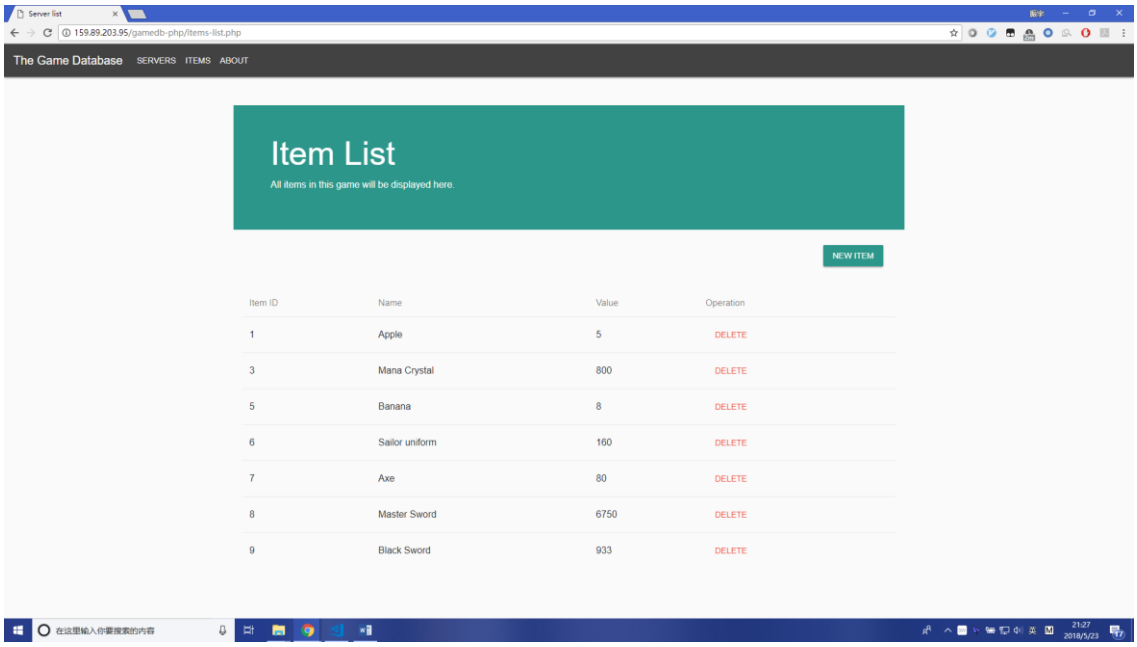
工程作业报告

1. 项目信息（10 分）

学号	1610842	姓名	左振宇	专业	计算机科学与技术
项目名称	Game Database - php				
必备环境	Apache 2.4, PHP 7.2, PostgreSQL 9.6				
系统主要功能简介（4 分）	系统可以管理某一 MMORPG 中服务器、玩家和物品之间的关系。并能够按照需要进行查询、增加、更新和删除操作。 测试用的网页已部署，可以通过 http://159.89.203.95/gamedb-php/servers-list.php 访问。				

系统主要页面截图（6分）	 


Server list
159.89.203.95/gamedb-php/item-statistics.php?server_id=1
The Game Database
SERVERS
ITEMS
ABOUT
Server List / Player List / Item Statistics
Item Statistics
Server id:1
Item ID
Name
Value
Total Quantity
Total Value
1
Apple
5
15
75
3
Mana Crystal
800
3
2400
5
Banana
8
2
16
6
Sailor uniform
160
1
160
7
Axe
80
0
0
8
Master Sword
6750
0
0
9
Black Sword
933
1
933


Server list
159.89.203.95/gamedb-php/items-list.php
The Game Database
SERVERS
ITEMS
ABOUT
Item List
All items in this game will be displayed here.
NEW ITEM
Item ID
Name
Value
Operation
1
Apple
5
DELETE
3
Mana Crystal
800
DELETE
5
Banana
8
DELETE
6
Sailor uniform
160
DELETE
7
Axe
80
DELETE
8
Master Sword
6750
DELETE
9
Black Sword
933
DELETE

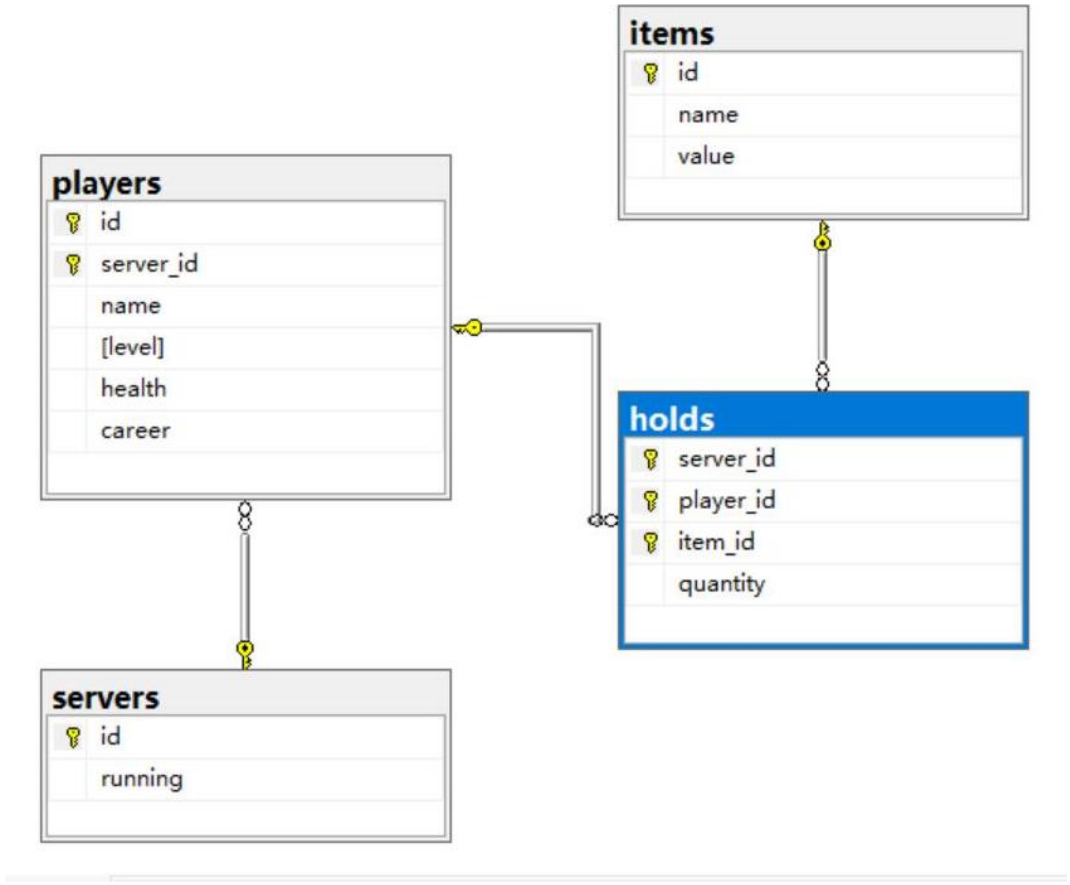
2. 系统配置（10 分）

<table>
<tr><td colspan="2">说明</td><td colspan="4">（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。</td></tr>
<tr><td rowspan="3">配置步骤
2 分</td><td>DBMS</td><td colspan="4">PostgreSQL 9.6</td></tr>
<tr><td rowspan="2">高级语言</td><td colspan="4">1.PHP</td></tr>
<tr><td colspan="4">2.HTML5</td></tr>
<tr><td colspan="2" rowspan="5">连接串分析
（6 分）</td><td>序号</td><td>名称</td><td>功能说明</td><td>取值</td></tr>
<tr><td>1</td><td>$host</td><td>指定数据库地址</td><td>"host=127.0.0.1"</td></tr>
<tr><td>2</td><td>$port</td><td>指定数据库端口</td><td>"port=5432"</td></tr>
<tr><td>3</td><td>$dbname</td><td>指定数据库名称</td><td>"dbname=gamedb"</td></tr>
<tr><td>4</td><td>$credentials</td><td>指定用户名和密码</td><td>"user=gamedbuser password=gPassword"</td></tr>
<tr><td colspan="2">连接串代码
（截屏）
（2 分）</td><td colspan="4">

```
$host        = "host=127.0.0.1";
$port        = "port=5432";
$dbname      = "dbname=gamedb";
$credentials = "user=gamedbuser password=gPassword";
```

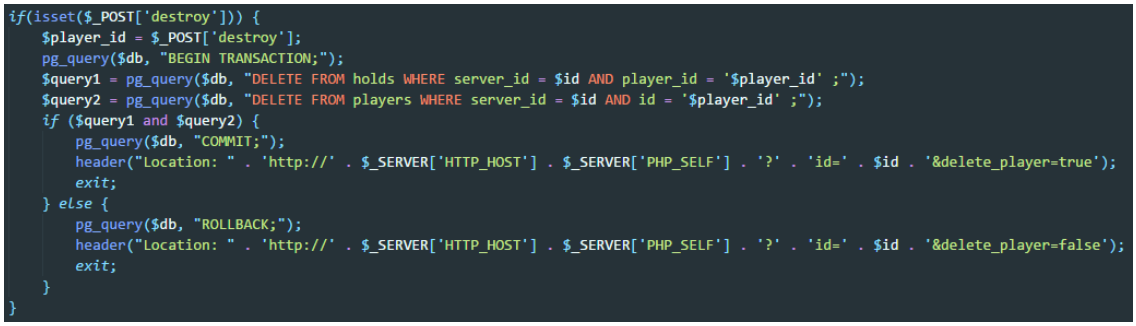
</td></tr>
<tr><td colspan="2">备注</td><td colspan="4"></td></tr>
</table>

3. **数据库设计（14 分）**

<table>
<tr><td>说明</td><td colspan="5">（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。</td></tr>
<tr><td rowspan="5">数据表（10）</td><td>创建顺序</td><td>数据表名称</td><td>主键</td><td>参照属性</td><td>被参照表及属性</td></tr>
<tr><td>1</td><td>servers</td><td>id</td><td></td><td></td></tr>
<tr><td>2</td><td>players</td><td>server_id, id</td><td>server_id</td><td>servers(id)</td></tr>
<tr><td>3</td><td>items</td><td>id</td><td></td><td></td></tr>
<tr><td>4</td><td>holds</td><td>server_id, player_id, item_id</td><td>server_id, player_id, item_id</td><td>players(server_id, id), items(id)</td></tr>
<tr><td>关系图（4）</td><td colspan="5"> 
</td></tr>
<tr><td>备注</td><td colspan="5">PostgreSQL 没有提供生成关系图的功能，故使用了 SQL Server 来生成关系图。</td></tr>
</table>

4. **含有事务应用的删除操作（13 分）**

<table>
<tr><td>说明</td><td colspan="2">（1 分）简要说明该操作所要完成的功能；
（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）
（1 分）表连接涉及字段描述（描述方式为“表 1.属性=表 2.属性”）
（1 分）删除条件涉及的字段描述(以“表名.属性=？”形式给出)
（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）
（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。</td></tr>
<tr><td>功能描述（1 分）</td><td colspan="2">删除一个 player 时，需要同时删除 holds 表中相应的数据。使用事务可以避免在删除操作执行失败时导致的数据库不一致。</td></tr>
<tr><td>涉及的表（2 分）</td><td colspan="2">players, holds</td></tr>
<tr><td>表连接涉及字段（1 分）</td><td colspan="2">holds.player_id = players.id,
holds.server_id = players.server_id</td></tr>
<tr><td rowspan="5">删除条件字段描述（1 分）</td><td>字段</td><td>规则</td></tr>
<tr><td>players.id</td><td>players.id = $player_id</td></tr>
<tr><td>players.server_id</td><td>players.server_id = $id</td></tr>
<tr><td>holds.player_id</td><td>holds.player_id = $player_id</td></tr>
<tr><td>holds.server_id</td><td>holds.server_id = $id</td></tr>
</table>

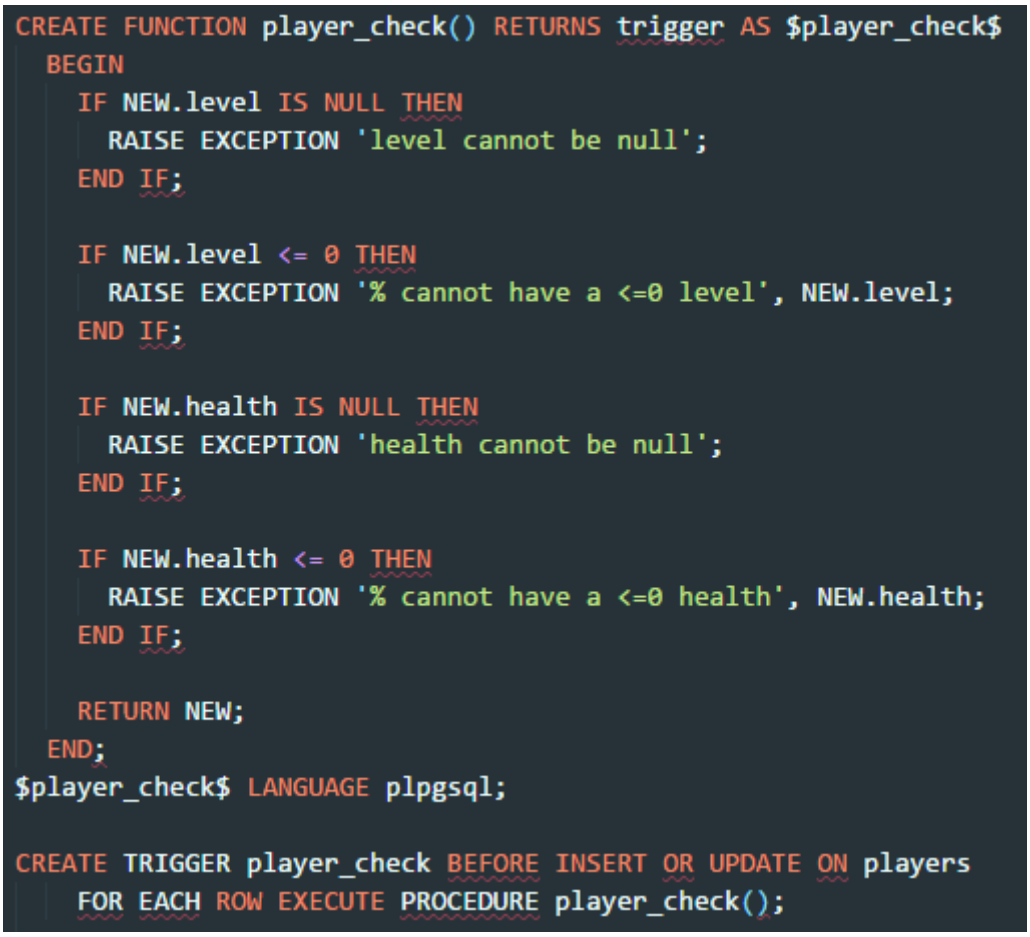
代码（4分）	
程序演示（4分）	
备注	

5. **触发器控制下的添加操作（20 分）**

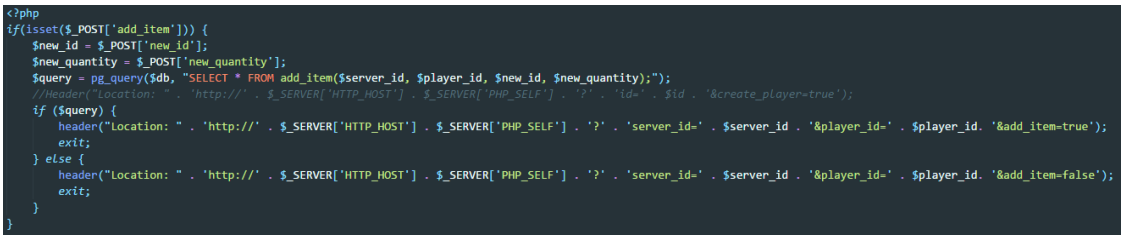
<table>
<tr><td>说明</td><td colspan="2">（1 分）简要说明该操作所要完成的功能；
（2 分）简要说明该触发器所要完成的功能
（1 分）该操作会涉及的表（以“表名”的形式给出）。
（2 分）该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空；
（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；
（8 分）如何执行该操作，按所述方法能够正常演示程序则给分。</td></tr>
<tr><td>功能描述（1 分）</td><td colspan="2">players 表中的条目代表角色的属性，其中 level 和 health 不能小于或等于 0。该触发器的功能是在向 players 表中添加数据时检查上述属性是否小于或等于 0，若如此则拒绝此次操作。</td></tr>
<tr><td>触发器描述（2 分）</td><td colspan="2">依次判断 level 和 health 是否为空或是否小于或等于零，若是，则抛出异常；否则返回新插入的条目。</td></tr>
<tr><td>涉及的表（1 分）</td><td colspan="2">players</td></tr>
<tr><td rowspan="3">输入数据（2 分）</td><td>字段</td><td>规则</td></tr>
<tr><td>NEW.level</td><td>NEW.level != NULL && NEW.level >= 0</td></tr>
<tr><td>NEW.health</td><td>NEW.health != NULL && NEW.health >= 0</td></tr>
<tr><td>插入操作源码（3 分）</td><td colspan="2">

```php
if(isset($_POST['new_player'])) {
    $new_name = $_POST['new_name'];
    $new_level = $_POST['new_level'];
    $new_health = $_POST['new_health'];
    $new_career = $_POST['new_career'];
    $query = pg_query($db, "INSERT INTO players(server_id, name, level, health, career) VALUES($id, '$new_name', $new_level, $new_health, '$new_career');");
    //Header("Location: " . 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] . '?' . 'id=' . $id . '&create_player=true');
    if ($query) {
        header("Location: " . 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] . '?' . 'id=' . $id . '&create_player=true');
        exit;
    } else {
        header("Location: " . 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] . '?' . 'id=' . $id . '&create_player=false');
        exit;
    }
}
```

</td></tr>
</table>

触发器源码（3分）	
程序演示（4分）	说明：不违背触发器能够执行插入操作。
程序演示（4分）	说明：违背触发器要求，不能够执行插入操作，系统报错。
备注	PostgreSQL 中作为触发器的函数并没有显式输入的参数。

6. **存储过程控制下的更新操作（18 分）**

<table>
<tr><td>说明</td><td>（1 分）简要说明该操作所要完成的功能；
（1 分）简要说明该存储过程所要完成的功能；
（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）
（1 分）表连接涉及字段描述（描述方式为“表 1.属性=表 2.属性”）
（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；
（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；
（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。</td></tr>
<tr><td>功能描述（1 分）</td><td>在给玩家添加物品时，需要判断物品 id 是否存在于 items 中，否则拒绝操作；还需判断该玩家是否已持有该物品，若已持有，则更新 holds 表中相应的条目为数值之和，若未持有，则向 holds 表中新添加对应的条目。</td></tr>
<tr><td>存储过程功能描述（1 分）</td><td>输入四个参数 m_server_id,m_plsyer_id,m_item_id 和 m_quantity，首先判断 m_quantity 是否小于等于零，若是，抛出异常；然后判断 m_item_id 是否存在于 items 表中，若否，抛出异常；最后判断 holds 中是否已有相应的条目，若是，则更新条目的 quantity = quantity + m_quantity；若否，则新建对应的条目。</td></tr>
<tr><td>涉及的关系表（2 分）</td><td>holds, items</td></tr>
<tr><td>表连接涉及字段（1）</td><td>holds.item_id = items.id</td></tr>
<tr><td>更改字段（2 分）</td><td><table><tr><td>字段</td><td>规则</td></tr><tr><td>quantity</td><td>IF exist_in_holds(server_id, player_id, item_id)
  quantity + m_quantity
ELSE
  quantity</td></tr></table></td></tr>
<tr><td>更新代码（3 分）</td><td> 
</td></tr>
</table>

<table>
<tr><td>创建存储过程源码（3分）</td><td>

```
CREATE OR REPLACE FUNCTION add_item(m_server_id integer, m_plsyer_id integer, m_item_id integer, m_quantity integer) RETURNS boolean AS $$
  BEGIN
    IF m_quantity <=0 THEN
      RAISE EXCEPTION 'Quantity must greater than 0.';
    END IF;

    IF NOT EXISTS(SELECT * FROM items WHERE id = m_item_id) THEN
      RAISE EXCEPTION 'Item not exist.';
    END IF;

    IF EXISTS(SELECT * FROM holds WHERE m_server_id = server_id AND m_plsyer_id = player_id AND m_item_id = item_id) THEN
      UPDATE holds SET quantity = m_quantity + quantity WHERE m_server_id = server_id AND m_plsyer_id = player_id AND m_item_id = item_id;
    ELSE
      INSERT INTO holds(server_id, player_id, item_id, quantity) VALUES(m_server_id, m_plsyer_id, m_item_id, m_quantity);
    END IF;

    RETURN TRUE;
  END;
$$ LANGUAGE plpgsql;
```

</td></tr>
<tr><td>存储过程执行源码（1分）</td><td>

```
<?php
if(isset($_POST['add_item'])) {
    $new_id = $_POST['new_id'];
    $new_quantity = $_POST['new_quantity'];
    $query = pg_query($db, "SELECT * FROM add_item($server_id, $player_id, $new_id, $new_quantity);");
    //Header("Location: " . 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] . '?' . 'id=' . $id . '&create_player=true');
    if ($query) {
        header("Location: " . 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] . '?' . 'server_id=' . $server_id . '&player_id=' . $player_id. '&add_item=true');
        exit;
    } else {
        header("Location: " . 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] . '?' . 'server_id=' . $server_id . '&player_id=' . $player_id. '&add_item=false');
        exit;
    }
}
```

</td></tr>
<tr><td>程序演示（2分）</td><td>说明：不违背存储过程，能够执行更新操作</td></tr>
<tr><td>程序演示（2分）</td><td>说明：违背存储过程，系统报错；</td></tr>
<tr><td>备注</td><td>PostgreSQL 中，存储过程以函数的形式存储和执行。</td></tr>
</table>

7. **含有视图的查询操作（15 分）**

说明	（1 分）简要说明该操作所要完成的功能； （1 分）简要说明建立的该视图的功能； （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） （1 分）简要说明表连接涉及的字段（以“表 1.属性=表 2.属性”） （6 分）实现该操作的关键代码（高级语言、SQL），截图即可； （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。
操作功能描述（1 分）	查询每个服务器内各种物品的总数以及总价值。
视图功能描述（1 分）	从 holds 表中选择不重复的(server_id, item_id)行，对 quantity 属性求和作为视图的 count 属性，并添加 items 表中对应条目的 name 和 value。
涉及的关系表（2 分）	holds, items
表连接字段（1 分）	holds.item_id = items.id

创建视图代	`CREATE VIEW item_count AS`
	`  SELECT h.server_id, h.item_id, i.name, i.value, Sum(h.quantity) AS "count"`
	`  FROM holds h, items i`
	`  WHERE h.item_id = i.id`
	`  GROUP BY h.server_id, h.item_id, i.name, i.value;`

码（3分）	
查询代码（3分）	
程序演示（4分）	
备注	最终在表格中需要显示在服务器中数量为零的物品，所以使用如上所述的视图并非一个好的实现方法。但为了演示从两个不同表中选择属性，所以最终选择此种写法。