

## Erläuterung mit Screenshots

Das Installationsskript für den "Overwatch Tournament Manager" startet mit dem Löschen vorhandener Tabellen, um sicherzustellen, dass das Datenbankmodell sauber installiert wird. Es erstellt dann die erforderlichen Tabellen und setzt die Integritätsbedingungen mithilfe von Triggern um.

```

4  --Drop-Statements für bestehende Tabellen mit CASCADE CONSTRAINTS
5  DROP TABLE Event CASCADE CONSTRAINTS;
6  DROP TABLE Tournament CASCADE CONSTRAINTS;
7  DROP TABLE Team CASCADE CONSTRAINTS;
8  DROP TABLE Map CASCADE CONSTRAINTS;
9  DROP TABLE Player CASCADE CONSTRAINTS;
10 DROP TABLE Hero CASCADE CONSTRAINTS;
11 DROP TABLE Ranking CASCADE CONSTRAINTS;
12 DROP TABLE Game CASCADE CONSTRAINTS;

```

Die Beziehungen werden durch die Verwendungen von Fremdschlüsseln in den Tabellen realisiert. Diese Beziehung wird durch die FOREIGN KEY-Klauseln in den CREATE TABLE-Anweisungen hergestellt.

**Um die Integritätsbedingungen umzusetzen, werden Trigger verwendet:**

CheckPlayerExperienceTrigger:

Der CheckPlayerExperienceTrigger überprüft, ob die Erfahrung eines Spielers in einem Team das erforderliche Mindestniveau für das Turnier erreicht:

```

102 --Überprüfe für jeden Spieler, ob sein Spielerfahrungsniveau niedriger ist als das Turniererfahrungsniveau.
103 CREATE OR REPLACE TRIGGER CheckPlayerExperienceTrigger
104 BEFORE INSERT ON Player
105 FOR EACH ROW
106 DECLARE
107     v_tournament_experience INTEGER;
108 BEGIN
109     -- Holen Sie sich die vorgesehenen Erfahrungswerte des Turniers
110     SELECT TournamentExperienceLevel INTO v_tournament_experience
111     FROM Tournament t
112     JOIN Team tm ON t.TournamentID = tm.TournamentID
113     WHERE tm.TeamID = :NEW.TeamID;
114
115     -- Überprüfen, ob die Erfahrung des Spielers größer ist als die vorgesehene Turniererfahrung
116     IF :NEW.PlayerExperienceLevel < v_tournament_experience THEN
117         RAISE_APPLICATION_ERROR(-20001, 'Spielererfahrung ist zu wenig, als die vorgesehene Turniererfahrung.');

```

Diese Bedingung wird durch den Trigger "CheckPlayerExperienceTrigger" umgesetzt. Der Trigger prüft vor dem Einfügen eines neuen Spielers und bei einem Update, ob die Erfahrung des Spielers kleiner ist als die vorgesehene Turniererfahrung (In diesem Beispiel: 4000). Hier ist ein Testfall, der den Trigger auslöst:

```

187 -- Füge Spieler zu TeamA ein (sollte erfolgreich sein)
188 INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler1', 4000, 'Info1', 1);
189 INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler2', 4000, 'Info2', 1);
190 INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler3', 4000, 'Info3', 1);
191 INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler4', 4000, 'Info4', 1);
192 INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler5', 4000, 'Info5', 1);
193 --Trigger: CheckPlayerExperienceTrigger wird hier ausgelöst
194 UPDATE Player SET PlayerExperienceLevel = 3800 WHERE PlayerName = 'Spieler5';
195 --Trigger: CheckPlayerExperienceTrigger wird hier ausgelöst
196 INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler6', 200, 'Info5', 1);

```

## Thema: Spiel-Events und Turniere in "Overwatch"

Fehler beim Start in Zeile: 194 in Befehl -

```
UPDATE Player SET PlayerExperienceLevel = 3800 WHERE PlayerName = 'Spieler5'
```

Fehlerbericht -

ORA-20001: Spielererfahrung ist zu wenig, als die vorgesehene Turniererfahrung.

ORA-06512: in "C##FBPOOL230.CHECKPLAYEREXPERIENCETRIGGER", Zeile 12

ORA-04088: Fehler bei der Ausführung von Trigger 'C##FBPOOL230.CHECKPLAYEREXPERIENCETRIGGER'

Fehler beim Start in Zeile: 196 in Befehl -

```
INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler6', 200, 'Info5', 1)
```

Fehlerbericht -

ORA-20001: Spielererfahrung ist zu wenig, als die vorgesehene Turniererfahrung.

ORA-06512: in "C##FBPOOL230.CHECKPLAYEREXPERIENCETRIGGER", Zeile 12

ORA-04088: Fehler bei der Ausführung von Trigger 'C##FBPOOL230.CHECKPLAYEREXPERIENCETRIGGER'

Dieser Testfall demonstriert, dass der Trigger "CheckPlayerExperienceTrigger" korrekt ausgelöst wird, wenn ein Spieler mit einer Erfahrung unterhalb der vorgesehenen Turniererfahrung eingefügt wird.

### CheckTeamFullnessTrigger:

Die Begrenzung der Teamgröße wird durch den Trigger "CheckTeamFullnessTrigger" sichergestellt:

```
122 | --Kontrolliert, ob die Spieleranzahl weniger als die vorausgesetzte maximale anzahl an Spielern pro Team.
123 | CREATE OR REPLACE TRIGGER CheckTeamFullnessTrigger
124 | BEFORE INSERT OR UPDATE ON Player
125 | FOR EACH ROW
126 | DECLARE
127 |     v_team_id Player.TeamID%TYPE; -- Variable, um die TeamID zu speichern
128 |     v_player_count INTEGER;
129 |     v_max_team_size INTEGER;
130 |     v_future_player_count INTEGER; -- Variable für zukünftige Spieleranzahl im Team
131 | BEGIN
132 |     -- Holen Sie sich die TeamID des betroffenen Spielers
133 |     v_team_id := :NEW.TeamID;
134 |
135 |     -- Zähle die Anzahl der Spieler im Team
136 |     SELECT COUNT(*) INTO v_player_count
137 |     FROM Player
138 |     WHERE TeamID = v_team_id;
139 |
140 |     -- Holen Sie sich die maximale Teamgröße aus dem zugehörigen Turnier
141 |     SELECT TournamentMaxTeamSize INTO v_max_team_size
142 |     FROM Tournament t
143 |     JOIN Team tm ON t.TournamentID = tm.TournamentID
144 |     WHERE tm.TeamID = v_team_id;
145 |
146 |     -- Berechne die zukünftige Spieleranzahl im Team
147 |     v_future_player_count := v_player_count;
148 |
149 |     -- Überprüfen, ob das Team nach der Aktion vollständig oder überfüllt wäre
150 |     IF v_future_player_count = v_max_team_size THEN
151 |         RAISE_APPLICATION_ERROR(-20003, 'Das Team mit der TeamID ' || v_team_id || ' wäre nach diesem Vorgang überfüllt.');
```

Der Trigger überprüft vor dem Einfügen oder Aktualisieren eines Spielers, ob das Team nach der Aktion überfüllt wäre. Hier ist ein Testfall:

```
187 | -- Füge Spieler zu TeamA ein (sollte erfolgreich sein)
188 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler1', 4000, 'Info1', 1);
189 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler2', 4000, 'Info2', 1);
190 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler3', 4000, 'Info3', 1);
191 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler4', 4000, 'Info4', 1);
192 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler5', 4000, 'Info5', 1);
193 | --Trigger: CheckTeamFullnessTrigger wird hier ausgelöst
194 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler6', 4000, 'Info6', 1);
```

Fehler beim Start in Zeile: 194 in Befehl -

```
INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler6', 4000, 'Info6', 1)
```

Fehlerbericht -

ORA-20003: Das Team mit der TeamID 1 wäre nach diesem Vorgang überfüllt.

ORA-06512: in "C##FBPOOL230.CHECKTEAMFULLNESSTRIGGER", Zeile 26

ORA-04088: Fehler bei der Ausführung von Trigger 'C##FBPOOL230.CHECKTEAMFULLNESSTRIGGER'

Dieser Testfall zeigt, dass der Trigger "CheckTeamFullnessTrigger" korrekt einen Fehler auslöst, wenn versucht wird, ein sechstes Mitglied zu einem Team hinzuzufügen und dadurch die maximale Teamgröße überschritten wird.

### PreventPlayerDeletionTrigger:

Die rekursive Beziehung wird durch den Trigger "PreventPlayerDeletionTrigger" umgesetzt:

```
156 | --Verhindere, dass Spieler gelöscht werden, solange sie noch in einem aktiven Team sind.
157 | CREATE OR REPLACE TRIGGER PreventPlayerDeletionTrigger
158 | BEFORE DELETE ON Player
159 | FOR EACH ROW
160 | DECLARE
161 |     v_team_count INTEGER;
162 | BEGIN
163 |     SELECT COUNT(*)
164 |     INTO v_team_count
165 |     FROM Team
166 |     WHERE TeamID = :OLD.TeamID;
167 |
168 |     IF v_team_count > 0 THEN
169 |         RAISE_APPLICATION_ERROR(-20004, 'Der Spieler gehört immer noch zu einem aktiven Team und kann nicht gelöscht werden.');
```

Der Trigger verhindert das Löschen eines Spielers, solange dieser noch Mitglied eines aktiven Teams ist. Hier ist ein Testfall:

```
184 | -- Füge Spieler zu TeamA ein (sollte erfolgreich sein)
185 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler1', 1000, 'Info1', 1);
186 |
187 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler2', 3500, 'Info2', 1);
188 |
189 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler3', 800, 'Info3', 1);
190 |
191 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler4', 1500, 'Info4', 1);
192 |
193 | INSERT INTO Player (PlayerName, PlayerExperienceLevel, PlayerInformation, TeamID) VALUES ('Spieler5', 900, 'Info5', 1);
194 |
195 | --Lösche Spieler von der Aktiven Liste: PreventPlayerDeletionTrigger
196 | DELETE FROM Player WHERE TeamID = 1 AND PlayerName = 'Spieler2';
```

Fehler beim Start in Zeile: 201 in Befehl -

```
DELETE FROM Player WHERE TeamID = 1 AND PlayerName = 'Spieler2'
```

Fehlerbericht -

```
ORA-20004: Der Spieler gehört immer noch zu einem aktiven Team und kann nicht gelöscht werden.
```

```
ORA-06512: in "C##FBPOOL230.PREVENTPLAYERDELETIONTRIGGER", Zeile 10
```

```
ORA-04088: Fehler bei der Ausführung von Trigger 'C##FBPOOL230.PREVENTPLAYERDELETIONTRIGGER'
```

Dieser Testfall zeigt, dass der Trigger "PreventPlayerDeletionTrigger" korrekt einen Fehler auslöst, wenn versucht wird, einen Spieler zu löschen, der noch Mitglied eines aktiven Teams ist.