Relationales Schema: Design

Felix Höffken, Sebastian Raitza, Nico von Geyso

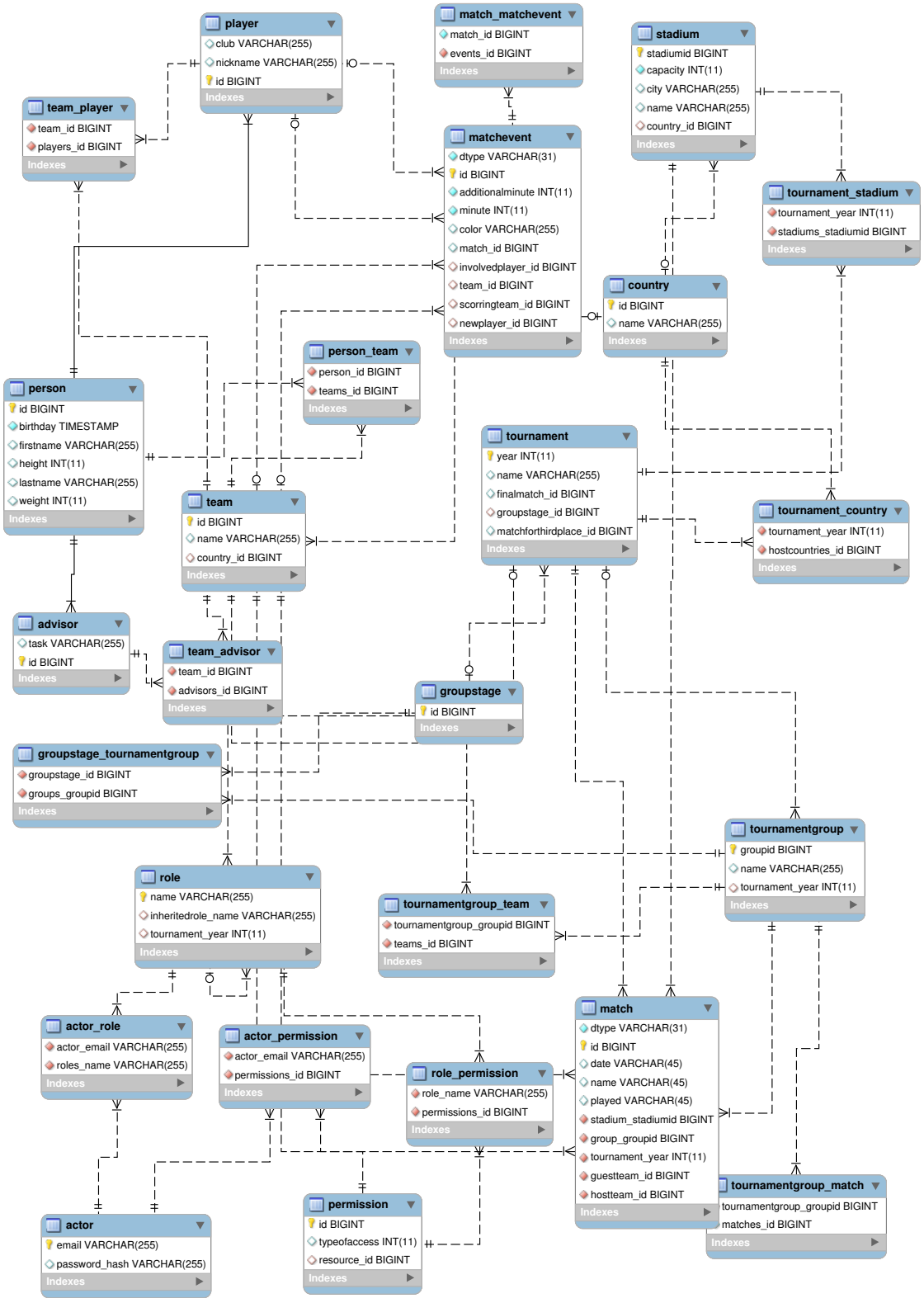
OR-Mapper Vorgehensweise

Wir haben uns entschieden mit *Hibernate* zu Arbeiten. Dabei wurde ausgehend von einem Java-Objektmodell ein relationales Modell erstellt. In unserem Build-Prozess (mittels *Maven*) wird das Schema jedesmal regeneriert und in die Datenbank exportiert. Ein Snapshot der Datenbank inklusive *Stored Procedure* zur Turniererstellung folgt unten.

Alle Testdaten werden von unserer GUI-Anwendung automatisch generiert und in der Datenbank persistiert. Die Turniererstellung aus dem Programm heraus ist sowohl über die *Stored Procedure* als über Java-Methoden möglich.

relationales Schema

Grafisches Model



SQL (Postgres)

```
--
-- PostgreSQL database dump
--

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

SET search_path = public, pg_catalog;

ALTER TABLE ONLY public.role_permission DROP CONSTRAINT fkf8a569386ac4edcc;
ALTER TABLE ONLY public.role_permission DROP CONSTRAINT fkf8a56938401023a7;
ALTER TABLE ONLY public.stadium DROP CONSTRAINT fkf21d53ddfcf4fc9d;
ALTER TABLE ONLY public.matchevent DROP CONSTRAINT fke491d7f5df1dd7b0;
ALTER TABLE ONLY public.matchevent DROP CONSTRAINT fke491d7f5db6578d7;
ALTER TABLE ONLY public.matchevent DROP CONSTRAINT fke491d7f5c7168977;
ALTER TABLE ONLY public.matchevent DROP CONSTRAINT fke491d7f56a9e6294;
ALTER TABLE ONLY public.matchevent DROP CONSTRAINT fke491d7f560766fd;
ALTER TABLE ONLY public.match_match DROP CONSTRAINT fkd7c3a68b884bbd1;
ALTER TABLE ONLY public.match_match DROP CONSTRAINT fkd7c3a68b1fd58223;
ALTER TABLE ONLY public.team_player DROP CONSTRAINT fkd4e5f703db6578d7;
ALTER TABLE ONLY public.team_player DROP CONSTRAINT fkd4e5f70318f27aa6;
ALTER TABLE ONLY public.actor_permission DROP CONSTRAINT fkcf243699d36f2a65;
ALTER TABLE ONLY public.actor_permission DROP CONSTRAINT fkcf243699401023a7;
ALTER TABLE ONLY public.match_matchevent DROP CONSTRAINT fkbfc93acfdcc26853;
ALTER TABLE ONLY public.match_matchevent DROP CONSTRAINT fkbfc93acf60766fd;
ALTER TABLE ONLY public.tournamentgroup_team DROP CONSTRAINT fkb7678b26d57db48a;
ALTER TABLE ONLY public.tournamentgroup_team DROP CONSTRAINT fkb7678b26cdc8c6de;
ALTER TABLE ONLY public.tournament_stadium DROP CONSTRAINT fkabf317a7db034f8f;
ALTER TABLE ONLY public.tournament_stadium DROP CONSTRAINT fkabf317a774b12939;
ALTER TABLE ONLY public.team_advisor DROP CONSTRAINT fka1d587deea5528a;
ALTER TABLE ONLY public.team_advisor DROP CONSTRAINT fka1d587dedb6578d7;
ALTER TABLE ONLY public.player DROP CONSTRAINT fk8ea387019b9d8d6d;
ALTER TABLE ONLY public.tournamentgroup DROP CONSTRAINT fk6372df674b12939;
ALTER TABLE ONLY public.groupstage_tournamentgroup DROP CONSTRAINT fk62cd5967d361fb7;
ALTER TABLE ONLY public.groupstage_tournamentgroup DROP CONSTRAINT fk62cd596526daa28;
ALTER TABLE ONLY public.permission DROP CONSTRAINT fk57f7a1ef94470e9c;
ALTER TABLE ONLY public.tournament_country DROP CONSTRAINT fk5625b3409a553667;
ALTER TABLE ONLY public.tournament_country DROP CONSTRAINT fk5625b34074b12939;
ALTER TABLE ONLY public.person_team DROP CONSTRAINT fk49fd4907ce781a97;
ALTER TABLE ONLY public.person_team DROP CONSTRAINT fk49fd4907cdc8c6de;
ALTER TABLE ONLY public.match DROP CONSTRAINT fk46ae9a5f0ec562f;
ALTER TABLE ONLY public.match DROP CONSTRAINT fk46ae9a5e2487ff;
ALTER TABLE ONLY public.match DROP CONSTRAINT fk46ae9a5d4eaebd3;
ALTER TABLE ONLY public.match DROP CONSTRAINT fk46ae9a574b12939;
ALTER TABLE ONLY public.match DROP CONSTRAINT fk46ae9a515c9d2b6;
ALTER TABLE ONLY public.tournament DROP CONSTRAINT fk3b743609fcd043a4;
ALTER TABLE ONLY public.tournament DROP CONSTRAINT fk3b7436097d361fb7;
ALTER TABLE ONLY public.tournament DROP CONSTRAINT fk3b74360966de6d99;
ALTER TABLE ONLY public.tournamentgroup_match DROP CONSTRAINT fk3525aa1cd57db48a;
ALTER TABLE ONLY public.tournamentgroup_match DROP CONSTRAINT fk3525aa1c2801c0aa;
ALTER TABLE ONLY public.team DROP CONSTRAINT fk27b67dfcf4fc9d;
ALTER TABLE ONLY public.role DROP CONSTRAINT fk26f496da563832;
ALTER TABLE ONLY public.role DROP CONSTRAINT fk26f49674b12939;
ALTER TABLE ONLY public.actor_role DROP CONSTRAINT fk26e2d0c0e4a0b3e5;
ALTER TABLE ONLY public.actor_role DROP CONSTRAINT fk26e2d0c0d36f2a65;
ALTER TABLE ONLY public.advisor DROP CONSTRAINT fk1fc9f7a09b9d8d6d;
ALTER TABLE ONLY public.tournamentgroup_team DROP CONSTRAINT
    tournamentgroup_team_teams_id_key;
ALTER TABLE ONLY public.tournamentgroup DROP CONSTRAINT tournamentgroup_pkey;
ALTER TABLE ONLY public.tournamentgroup_match DROP CONSTRAINT
    tournamentgroup_match_matches_id_key;
ALTER TABLE ONLY public.tournament DROP CONSTRAINT tournament_pkey;
```

```
65 ALTER TABLE ONLY public.team DROP CONSTRAINT team_pkey;
66 ALTER TABLE ONLY public.team_advisor DROP CONSTRAINT team_advisor_advisors_id_key;
67 ALTER TABLE ONLY public.stadium DROP CONSTRAINT stadium_pkey;
68 ALTER TABLE ONLY public."role" DROP CONSTRAINT role_pkey;
69 ALTER TABLE ONLY public.role_permission DROP CONSTRAINT
       role_permission_permissions_id_key;
70 ALTER TABLE ONLY public.resource DROP CONSTRAINT resource_pkey;
71 ALTER TABLE ONLY public.player DROP CONSTRAINT player_pkey;
72 ALTER TABLE ONLY public.person DROP CONSTRAINT person_pkey;
73 ALTER TABLE ONLY public.permission DROP CONSTRAINT permission_pkey;
74 ALTER TABLE ONLY public.matchevent DROP CONSTRAINT matchevent_pkey;
75 ALTER TABLE ONLY public."match" DROP CONSTRAINT match_pkey;
76 ALTER TABLE ONLY public.match_matchevent DROP CONSTRAINT match_matchevent_events_id_key;
77 ALTER TABLE ONLY public.groupstage_tournamentgroup DROP CONSTRAINT
       groupstage_tournamentgroup_groups_groupid_key;
78 ALTER TABLE ONLY public.groupstage DROP CONSTRAINT groupstage_pkey;
79 ALTER TABLE ONLY public.country DROP CONSTRAINT country_pkey;
80 ALTER TABLE ONLY public.advisor DROP CONSTRAINT advisor_pkey;
81 ALTER TABLE ONLY public.actor_role DROP CONSTRAINT actor_role_roles_name_key;
82 ALTER TABLE ONLY public.actor DROP CONSTRAINT actor_pkey;
83 ALTER TABLE ONLY public.actor_permission DROP CONSTRAINT
       actor_permission_permissions_id_key;
84 DROP TABLE public.tournamentgroup_team;
85 DROP TABLE public.tournamentgroup_match;
86 DROP TABLE public.tournamentgroup;
87 DROP TABLE public.tournament_stadium;
88 DROP TABLE public.tournament_country;
89 DROP TABLE public.tournament;
90 DROP TABLE public.team_player;
91 DROP TABLE public.team_advisor;
92 DROP TABLE public.role_permission;
93 DROP TABLE public."role";
94 DROP TABLE public.resource;
95 DROP TABLE public.person_team;
96 DROP TABLE public.person;
97 DROP TABLE public.permission;
98 DROP TABLE public.matchevent;
99 DROP TABLE public.match_matchevent;
100 DROP TABLE public.match_match;
101 DROP TABLE public."match";
102 DROP SEQUENCE public.hibernate_sequence;
103 DROP TABLE public.groupstage_tournamentgroup;
104 DROP TABLE public.advisor;
105 DROP TABLE public.actor_role;
106 DROP TABLE public.actor_permission;
107 DROP TABLE public.actor;
108 DROP FUNCTION public.getstadiumsforcountry(bigint);
109 DROP TABLE public.stadium;
110 DROP FUNCTION public.getplayer();
111 DROP TABLE public.player;
112 DROP FUNCTION public.getnextsequence();
113 DROP FUNCTION public.getcountry();
114 DROP TABLE public.country;
115 DROP FUNCTION public.generateteam();
116 DROP TABLE public.team;
117 DROP FUNCTION public.generatematch(bigint, bigint, bigint);
118 DROP FUNCTION public.generateknockouttree("integer", bigint);
119 DROP FUNCTION public.generategroupstage();
120 DROP TABLE public.groupstage;
121 DROP FUNCTION public.generategroupmatches(bigint);
122 DROP FUNCTION public.createchampionship("integer", text);
123 DROP FUNCTION public.concat("character varying", bigint);
124 DROP FUNCTION public.concat("character varying", "integer");
125 DROP PROCEDURAL LANGUAGE plpgsql;
126 DROP SCHEMA public;
127 --
128 -- Name: public; Type: SCHEMA; Schema: -; Owner: postgres
129 --
```

```
130
131 CREATE SCHEMA public;
132
133
134 ALTER SCHEMA public OWNER TO postgres;
135
136 --
137 -- Name: plpgsql; Type: PROCEDURAL LANGUAGE; Schema: -; Owner: postgres
138 --
139
140 CREATE PROCEDURAL LANGUAGE plpgsql;
141
142
143 ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;
144
145 SET search_path = public, pg_catalog;
146
147 --
148 -- Name: concat(character varying, integer); Type: FUNCTION; Schema: public; Owner:
        postgres
149 --
150
151 CREATE FUNCTION concat(character varying, integer) RETURNS character varying
152     LANGUAGE plpgsql
153     AS $_$
154 BEGIN
155     return $1 || ' ' || chr(49 + ($2%119));
156 END
157 $_$;
158
159
160 ALTER FUNCTION public.concat(character varying, integer) OWNER TO postgres;
161
162 --
163 -- Name: concat(character varying, bigint); Type: FUNCTION; Schema: public; Owner:
        postgres
164 --
165
166 CREATE FUNCTION concat(character varying, bigint) RETURNS character varying
167     LANGUAGE plpgsql
168     AS $_$
169 BEGIN
170     return $1 || ' ' || chr(CAST(49 + ($2%119) AS INT));
171 END
172 $_$;
173
174
175 ALTER FUNCTION public.concat(character varying, bigint) OWNER TO postgres;
176
177 --
178 -- Name: createchampionship(integer, text); Type: FUNCTION; Schema: public; Owner:
        postgres
179 --
180
181 CREATE FUNCTION createchampionship(integer, text) RETURNS void
182     LANGUAGE plpgsql
183     AS $_$
184 DECLARE
185     yearParam ALIAS FOR $1;
186     nameParam ALIAS FOR $2;
187     host country%ROWTYPE;
188     currentStadium stadium%ROWTYPE;
189     groupStage groupstage%ROWTYPE;
190     finalId bigint;
191 BEGIN
192     RAISE NOTICE 'Creating a new tournament';
193
194   --Generiert die K.O.-Phase
```

```
195     finalId := getNextSequence();
196     INSERT INTO match(id, name, played, dtype)
197     VALUES (finalId, 'Finale', false, 'KnockoutMatch');
198     PERFORM generateKnockoutTree(1, finalId);
199 
200     --Generiert die Gruppenphase
201     groupStage := generateGroupStage();
202 
203     --Speichert das Turnier ab
204     INSERT INTO tournament (year, name, finalmatch_id, groupstage_id)
205     VALUES (yearParam, nameParam, finalId, groupStage.id);
206 
207     -- Set a random host country
208     host := getCountry();
209     INSERT INTO "tournament_country" VALUES (yearParam, host.id);
210 
211     -- Set 8 random stadiums
212     FOR currentStadium IN SELECT * FROM getStadiumsForCountry(host.id) LOOP
213         INSERT INTO tournament_stadium VALUES(yearParam,currentStadium.stadiumid);
214     END LOOP;
215   RETURN;
216 END;
217 $_$;
218 
219 
220 ALTER FUNCTION public.createchampionship(integer, text) OWNER TO postgres;
221 
222 --
223 -- Name: generategroupmatches(bigint); Type: FUNCTION; Schema: public; Owner: postgres
224 --
225 
226 CREATE FUNCTION generategroupmatches(bigint) RETURNS void
227     LANGUAGE plpgsql
228     AS $_$
229 DECLARE
230     groupId ALIAS FOR $1;
231     numberOfTeams int;
232     currentTeam team%ROWTYPE;
233     teams team[];
234     i int;
235     j int;
236 BEGIN
237 
238     SELECT COUNT(*) INTO numberOfTeams
239     FROM tournamentgroup_team
240     WHERE tournamentgroup_groupid = groupId;
241 
242   -- Test ob genuegend Teams in der Gruppe sind
243     if(numberOfTeams < 4) THEN
244         RAISE EXCEPTION 'at least 4 teams have to be in a group';
245         RETURN;
246     END IF;
247 
248 
249     -- Erstellt ein Array aus dem Teams der Gruppe
250     teams := '{}';
251     FOR currentTeam IN
252         SELECT t.*
253         FROM team t
254         JOIN tournamentgroup_team g ON (g.teams_id = t.id)
255         WHERE tournamentgroup_groupid = groupId
256     LOOP
257         teams := array_append(teams, currentTeam);
258     END LOOP;
259 
260   -- Laesst jede Mannschaft einmal gegen alle anderen Manschaften antreten
261     FOR i IN 1..4 LOOP
262         FOR j IN (i+1)..4 LOOP
```

```
263             PERFORM generateMatch(teams[i].id, teams[j].id, groupId);
264         END LOOP;
265     END LOOP;
266
267     return;
268
269 END
270 $_$;
271
272
273 ALTER FUNCTION public.generategroupmatches(bigint) OWNER TO postgres;
274
275 SET default_tablespace = '';
276
277 SET default_with_oids = false;
278
279 --
280 -- Name: groupstage; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
281 --
282
283 CREATE TABLE groupstage (
284     id bigint NOT NULL
285 );
286
287
288 ALTER TABLE public.groupstage OWNER TO postgres;
289
290 --
291 -- Name: generategroupstage(); Type: FUNCTION; Schema: public; Owner: postgres
292 --
293
294 CREATE FUNCTION generategroupstage() RETURNS groupstage
295     LANGUAGE plpgsql
296     AS $$
297 DECLARE
298     stageId int;
299     stage groupstage;
300     currentTeam team;
301     currentGroup tournamentgroup;
302     currentGroupId bigint;
303     i int;
304     j int;
305 BEGIN
306     stageId := getNextSequence();
307
308     INSERT INTO groupstage VALUES (stageId);
309
310     -- Fuer alle 8 Gruppen
311     FOR i IN 1..8 LOOP
312         currentGroupId := getNextSequence();
313
314         INSERT INTO tournamentgroup (groupid, name)
315         VALUES (currentGroupId, concat('Gruppe', 10));
316
317         -- generiere 4 Mannschaften
318         FOR j IN 1..4 LOOP
319             currentTeam := generateTeam();
320
321             INSERT INTO tournamentgroup_team (tournamentgroup_groupid, teams_id)
322             VALUES (currentGroupId, currentTeam.id);
323         END LOOP;
324
325         INSERT INTO groupstage_tournamentgroup VALUES (stageId, currentGroupId);
326
327     -- und trage die Gruppenspiele ein
328         PERFORM generateGroupMatches(currentGroupId);
329
330     END LOOP;
```

```
331
332     SELECT * INTO stage FROM groupstage WHERE id = stageId;
333
334     return stage;
335
336 END
337 $$;
338
339
340 ALTER FUNCTION public.generategroupstage() OWNER TO postgres;
341
342 --
343 -- Name: generateknockouttree(integer, bigint); Type: FUNCTION; Schema: public; Owner:
        postgres
344 --
345
346 CREATE FUNCTION generateknockouttree(integer, bigint) RETURNS void
347     LANGUAGE plpgsql
348     AS $_$
349 DECLARE
350     height ALIAS FOR $1;
351     nodeId ALIAS FOR $2;
352     matchId1 bigint;
353     matchId2 bigint;
354     newHeight int;
355     knockoutMatchType varchar;
356 BEGIN
357   --Rekursionsanker
358     IF(height > 3) THEN
359         RETURN;
360     ELSIF (height = 1) THEN
361         knockoutMatchType := 'Halbfinale';
362     ELSIF (height = 2) THEN
363         knockoutMatchType := 'Viertelfinale';
364     ELSIF (height = 3) THEN
365         knockoutMatchType := 'Achtelfinale';
366     END IF;
367
368
369   -- Erstellen zweier Kindspiele
370     matchId1 := getNextSequence();
371     INSERT INTO match(id, name, played, dtype)
372     VALUES (matchId1, knockoutMatchType, false, 'KnockoutMatch');
373
374     matchId2 := getNextSequence();
375     INSERT INTO match(id, name, played, dtype)
376     VALUES (matchId2, knockoutMatchType, false, 'KnockoutMatch');
377
378   -- Hinzufuegen zum Baum
379     INSERT INTO match_match(match_id, childs_id) VALUES (nodeId, matchId1);
380     INSERT INTO match_match(match_id, childs_id) VALUES (nodeId, matchId2);
381
382   -- rekursiver Aufruf
383     newHeight := height + 1;
384     PERFORM generateKnockoutTree(newHeight, matchId1);
385     PERFORM generateKnockoutTree(newHeight, matchId2);
386
387     RETURN;
388 END;
389 $_$;
390
391
392 ALTER FUNCTION public.generateknockouttree(integer, bigint) OWNER TO postgres;
393
394 --
395 -- Name: generatematch(bigint, bigint, bigint); Type: FUNCTION; Schema: public; Owner:
        postgres
396 --
```

```
397
398 CREATE FUNCTION generatematch(bigint, bigint, bigint) RETURNS void
399     LANGUAGE plpgsql
400     AS $_$
401 DECLARE
402     hostTeam ALIAS FOR $1;
403     guestTeam ALIAS FOR $2;
404     groupId ALIAS FOR $3;
405     matchId bigint;
406     i int;
407 BEGIN
408     matchId := getNextSequence();
409
410     INSERT INTO match(id, hostteam_id, guestteam_id, played, dtype, group_groupid)
411     VALUES (matchId, hostTeam, guestTeam, false,'GroupMatch', groupId);
412
413     INSERT INTO tournamentgroup_match
414     VALUES (groupId, matchId);
415 END
416 $_$;
417
418
419 ALTER FUNCTION public.generatematch(bigint, bigint, bigint) OWNER TO postgres;
420
421 --
422 -- Name: team; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
423 --
424
425 CREATE TABLE team (
426     id bigint NOT NULL,
427     name character varying(255),
428     country_id bigint
429 );
430
431
432 ALTER TABLE public.team OWNER TO postgres;
433
434 --
435 -- Name: generateteam(); Type: FUNCTION; Schema: public; Owner: postgres
436 --
437
438 CREATE FUNCTION generateteam() RETURNS team
439     LANGUAGE plpgsql
440     AS $$
441 DECLARE
442     i int;
443     j int;
444     sequenceValue int;
445     playerId int;
446     selectedTeam Team%ROWTYPE;
447 BEGIN
448     SELECT id INTO i FROM getCountry();
449
450     sequenceValue := getNextSequence();
451
452     INSERT INTO team VALUES (sequenceValue, concat('Musterteam ', sequenceValue), i);
453     SELECT * INTO selectedTeam FROM team WHERE id = sequenceValue;
454
455     FOR j IN 1..23 LOOP
456         SELECT id INTO playerId FROM getPlayer();
457         INSERT INTO team_player VALUES (selectedTeam.id, playerId);
458         INSERT INTO person_team VALUES (playerId, selectedTeam.id);
459     END LOOP;
460
461     return selectedTeam;
462 END
463 $$;
464
```

```
465
466 ALTER FUNCTION public.generateteam() OWNER TO postgres;
467
468 --
469 -- Name: country; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
470 --
471
472 CREATE TABLE country (
473     id bigint NOT NULL,
474     name character varying(255)
475 );
476
477
478 ALTER TABLE public.country OWNER TO postgres;
479
480 --
481 -- Name: getcountry(); Type: FUNCTION; Schema: public; Owner: postgres
482 --
483
484 CREATE FUNCTION getcountry() RETURNS SETOF country
485     LANGUAGE plpgsql
486     AS $$
487 DECLARE
488     selectedRow Country%ROWTYPE;
489     n int := 0;
490 BEGIN
491     SELECT COUNT(*) INTO n FROM Country;
492     IF(n < 1) THEN
493         INSERT INTO Country VALUES (getNextSequence(), 'DummyLand');
494     END IF;
495
496     SELECT * INTO selectedRow FROM Country ORDER BY RANDOM() LIMIT 1;
497     RETURN NEXT selectedRow;
498 END
499 $$;
500
501
502 ALTER FUNCTION public.getcountry() OWNER TO postgres;
503
504 --
505 -- Name: getnextsequence(); Type: FUNCTION; Schema: public; Owner: postgres
506 --
507
508 CREATE FUNCTION getnextsequence() RETURNS bigint
509     LANGUAGE sql
510     AS $$
511     SELECT nextval('hibernate_sequence') FROM hibernate_sequence;
512 $$;
513
514
515 ALTER FUNCTION public.getnextsequence() OWNER TO postgres;
516
517 --
518 -- Name: player; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
519 --
520
521 CREATE TABLE player (
522     club character varying(255),
523     nickname character varying(255),
524     id bigint NOT NULL
525 );
526
527
528 ALTER TABLE public.player OWNER TO postgres;
529
530 --
531 -- Name: getplayer(); Type: FUNCTION; Schema: public; Owner: postgres
532 --
```

```
533
534 CREATE FUNCTION getplayer() RETURNS player
535     LANGUAGE plpgsql
536     AS $$
537 DECLARE
538     createdPlayer Player%ROWTYPE;
539     sequenceValue bigint;
540 BEGIN
541     sequenceValue := getNextSequence();
542
543     INSERT INTO person (id, firstname, lastname)
544     VALUES (sequenceValue, concat('Vorname', sequenceValue), concat('Nachname',
            sequenceValue));
545
546     INSERT INTO player (id, nickname, club)
547     VALUES (sequenceValue, concat('Nick', sequenceValue), 'FC Seehaeusl');
548
549     SELECT * INTO createdPlayer FROM player WHERE id = sequenceValue;
550
551     return createdPlayer;
552 END
553 $$;
554
555
556 ALTER FUNCTION public.getplayer() OWNER TO postgres;
557
558 --
559 -- Name: stadium; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
560 --
561
562 CREATE TABLE stadium (
563     stadiumid bigint NOT NULL,
564     capacity integer NOT NULL,
565     city character varying(255),
566     name character varying(255),
567     country_id bigint
568 );
569
570
571 ALTER TABLE public.stadium OWNER TO postgres;
572
573 --
574 -- Name: getstadiumsforcountry(bigint); Type: FUNCTION; Schema: public; Owner: postgres
575 --
576
577 CREATE FUNCTION getstadiumsforcountry(bigint) RETURNS SETOF stadium
578     LANGUAGE plpgsql
579     AS $_$
580 DECLARE
581     countryId ALIAS FOR $1;
582     selectedRow Stadium%ROWTYPE;
583     n int := 0;
584     i int;
585 BEGIN
586     SELECT COUNT(*) INTO n FROM Stadium WHERE country_id = countryId;
587     IF(n < 8) THEN
588         FOR i IN 1..(8-n) LOOP
589             INSERT INTO Stadium VALUES (getNextSequence(), 500, concat('Dummystadt',i) ,
                    concat('Dummystadion',i), countryId);
590         END LOOP;
591     END IF;
592
593     FOR selectedRow IN SELECT * FROM stadium ORDER BY RANDOM() LIMIT 8 LOOP
594         return next selectedRow;
595     END LOOP;
596
597     return;
598 END
```

```
$_$;


ALTER FUNCTION public.getstadiumsforcountry(bigint) OWNER TO postgres;

--
-- Name: actor; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE actor (
    email character varying(255) NOT NULL,
    password_hash character varying(255)
);


ALTER TABLE public.actor OWNER TO postgres;

--
-- Name: actor_permission; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE actor_permission (
    actor_email character varying(255) NOT NULL,
    permissions_id bigint NOT NULL
);


ALTER TABLE public.actor_permission OWNER TO postgres;

--
-- Name: actor_role; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE actor_role (
    actor_email character varying(255) NOT NULL,
    roles_name character varying(255) NOT NULL
);


ALTER TABLE public.actor_role OWNER TO postgres;

--
-- Name: advisor; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE advisor (
    task character varying(255),
    id bigint NOT NULL
);


ALTER TABLE public.advisor OWNER TO postgres;

--
-- Name: groupstage_tournamentgroup; Type: TABLE; Schema: public; Owner: postgres;
    Tablespace:
--

CREATE TABLE groupstage_tournamentgroup (
    groupstage_id bigint NOT NULL,
    groups_groupid bigint NOT NULL
);


ALTER TABLE public.groupstage_tournamentgroup OWNER TO postgres;

--
-- Name: hibernate_sequence; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--

CREATE SEQUENCE hibernate_sequence
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;


ALTER TABLE public.hibernate_sequence OWNER TO postgres;

--
-- Name: match; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE match (
    dtype character varying(31) NOT NULL,
    id bigint NOT NULL,
    date timestamp without time zone,
    name character varying(255),
    played boolean NOT NULL,
    guestteam_id bigint,
    hostteam_id bigint,
    stadium_stadiumid bigint,
    tournament_year integer,
    group_groupid bigint
);


ALTER TABLE public.match OWNER TO postgres;

--
-- Name: match_match; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE match_match (
    match_id bigint NOT NULL,
    childs_id bigint NOT NULL
);


ALTER TABLE public.match_match OWNER TO postgres;

--
-- Name: match_matchevent; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE match_matchevent (
    match_id bigint NOT NULL,
    events_id bigint NOT NULL
);


ALTER TABLE public.match_matchevent OWNER TO postgres;

--
-- Name: matchevent; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE matchevent (
    dtype character varying(31) NOT NULL,
    id bigint NOT NULL,
    additionalminute integer NOT NULL,
    minute integer NOT NULL,
    color character varying(255),
    match_id bigint,
    involvedplayer_id bigint,
```

```
    team_id bigint,
    scorringteam_id bigint,
    newplayer_id bigint
);


ALTER TABLE public.matchevent OWNER TO postgres;

--
-- Name: permission; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE permission (
    id bigint NOT NULL,
    typeofaccess integer,
    resource_id bigint
);


ALTER TABLE public.permission OWNER TO postgres;

--
-- Name: person; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE person (
    id bigint NOT NULL,
    birthday timestamp without time zone,
    firstname character varying(255),
    height integer,
    lastname character varying(255),
    weight integer
);


ALTER TABLE public.person OWNER TO postgres;

--
-- Name: person_team; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE person_team (
    person_id bigint NOT NULL,
    teams_id bigint NOT NULL
);


ALTER TABLE public.person_team OWNER TO postgres;

--
-- Name: resource; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE resource (
    id bigint NOT NULL,
    key bytea,
    name character varying(255)
);


ALTER TABLE public.resource OWNER TO postgres;

--
-- Name: role; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE role (
    name character varying(255) NOT NULL,
```

```
    inheritedrole_name character varying(255),
    tournament_year integer
);


ALTER TABLE public.role OWNER TO postgres;

--
-- Name: role_permission; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE role_permission (
    role_name character varying(255) NOT NULL,
    permissions_id bigint NOT NULL
);


ALTER TABLE public.role_permission OWNER TO postgres;

--
-- Name: team_advisor; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE team_advisor (
    team_id bigint NOT NULL,
    advisors_id bigint NOT NULL
);


ALTER TABLE public.team_advisor OWNER TO postgres;

--
-- Name: team_player; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE team_player (
    team_id bigint NOT NULL,
    players_id bigint NOT NULL
);


ALTER TABLE public.team_player OWNER TO postgres;

--
-- Name: tournament; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE tournament (
    year integer NOT NULL,
    name character varying(255),
    finalmatch_id bigint,
    groupstage_id bigint,
    matchforthirdplace_id bigint
);


ALTER TABLE public.tournament OWNER TO postgres;

--
-- Name: tournament_country; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE tournament_country (
    tournament_year integer NOT NULL,
    hostcountries_id bigint NOT NULL
);


```

```
870 ALTER TABLE public.tournament_country OWNER TO postgres;
871
872 --
873 -- Name: tournament_stadium; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
874 --
875
876 CREATE TABLE tournament_stadium (
877     tournament_year integer NOT NULL,
878     stadiums_stadiumid bigint NOT NULL
879 );
880
881
882 ALTER TABLE public.tournament_stadium OWNER TO postgres;
883
884 --
885 -- Name: tournamentgroup; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
886 --
887
888 CREATE TABLE tournamentgroup (
889     groupid bigint NOT NULL,
890     name character varying(255),
891     tournament_year integer
892 );
893
894
895 ALTER TABLE public.tournamentgroup OWNER TO postgres;
896
897 --
898 -- Name: tournamentgroup_match; Type: TABLE; Schema: public; Owner: postgres; Tablespace
        :
899 --
900
901 CREATE TABLE tournamentgroup_match (
902     tournamentgroup_groupid bigint NOT NULL,
903     matches_id bigint NOT NULL
904 );
905
906
907 ALTER TABLE public.tournamentgroup_match OWNER TO postgres;
908
909 --
910 -- Name: tournamentgroup_team; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
911 --
912
913 CREATE TABLE tournamentgroup_team (
914     tournamentgroup_groupid bigint NOT NULL,
915     teams_id bigint NOT NULL
916 );
917
918
919 ALTER TABLE public.tournamentgroup_team OWNER TO postgres;
920
921 --
922 -- Name: actor_permission_permissions_id_key; Type: CONSTRAINT; Schema: public; Owner:
        postgres; Tablespace:
923 --
924
925 ALTER TABLE ONLY actor_permission
926     ADD CONSTRAINT actor_permission_permissions_id_key UNIQUE (permissions_id);
927
928
929 --
930 -- Name: actor_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
931 --
932
933 ALTER TABLE ONLY actor
934     ADD CONSTRAINT actor_pkey PRIMARY KEY (email);
935
```

```
936
937 --
938 -- Name: actor_role_roles_name_key; Type: CONSTRAINT; Schema: public; Owner: postgres;
        Tablespace:
939 --
940
941 ALTER TABLE ONLY actor_role
942     ADD CONSTRAINT actor_role_roles_name_key UNIQUE (roles_name);
943
944
945 --
946 -- Name: advisor_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
947 --
948
949 ALTER TABLE ONLY advisor
950     ADD CONSTRAINT advisor_pkey PRIMARY KEY (id);
951
952
953 --
954 -- Name: country_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
955 --
956
957 ALTER TABLE ONLY country
958     ADD CONSTRAINT country_pkey PRIMARY KEY (id);
959
960
961 --
962 -- Name: groupstage_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
963 --
964
965 ALTER TABLE ONLY groupstage
966     ADD CONSTRAINT groupstage_pkey PRIMARY KEY (id);
967
968
969 --
970 -- Name: groupstage_tournamentgroup_groups_groupid_key; Type: CONSTRAINT; Schema: public
        ; Owner: postgres; Tablespace:
971 --
972
973 ALTER TABLE ONLY groupstage_tournamentgroup
974     ADD CONSTRAINT groupstage_tournamentgroup_groups_groupid_key UNIQUE (groups_groupid)
            ;
975
976
977 --
978 -- Name: match_matchevent_events_id_key; Type: CONSTRAINT; Schema: public; Owner:
        postgres; Tablespace:
979 --
980
981 ALTER TABLE ONLY match_matchevent
982     ADD CONSTRAINT match_matchevent_events_id_key UNIQUE (events_id);
983
984
985 --
986 -- Name: match_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
987 --
988
989 ALTER TABLE ONLY match
990     ADD CONSTRAINT match_pkey PRIMARY KEY (id);
991
992
993 --
994 -- Name: matchevent_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
995 --
996
997 ALTER TABLE ONLY matchevent
998     ADD CONSTRAINT matchevent_pkey PRIMARY KEY (id);
999
```

```
1000
1001 --
1002 -- Name: permission_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
1003 --
1004
1005 ALTER TABLE ONLY permission
1006     ADD CONSTRAINT permission_pkey PRIMARY KEY (id);
1007
1008
1009 --
1010 -- Name: person_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
1011 --
1012
1013 ALTER TABLE ONLY person
1014     ADD CONSTRAINT person_pkey PRIMARY KEY (id);
1015
1016
1017 --
1018 -- Name: player_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
1019 --
1020
1021 ALTER TABLE ONLY player
1022     ADD CONSTRAINT player_pkey PRIMARY KEY (id);
1023
1024
1025 --
1026 -- Name: resource_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
1027 --
1028
1029 ALTER TABLE ONLY resource
1030     ADD CONSTRAINT resource_pkey PRIMARY KEY (id);
1031
1032
1033 --
1034 -- Name: role_permission_permissions_id_key; Type: CONSTRAINT; Schema: public; Owner:
        postgres; Tablespace:
1035 --
1036
1037 ALTER TABLE ONLY role_permission
1038     ADD CONSTRAINT role_permission_permissions_id_key UNIQUE (permissions_id);
1039
1040
1041 --
1042 -- Name: role_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
1043 --
1044
1045 ALTER TABLE ONLY role
1046     ADD CONSTRAINT role_pkey PRIMARY KEY (name);
1047
1048
1049 --
1050 -- Name: stadium_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
1051 --
1052
1053 ALTER TABLE ONLY stadium
1054     ADD CONSTRAINT stadium_pkey PRIMARY KEY (stadiumid);
1055
1056
1057 --
1058 -- Name: team_advisor_advisors_id_key; Type: CONSTRAINT; Schema: public; Owner: postgres
        ; Tablespace:
1059 --
1060
1061 ALTER TABLE ONLY team_advisor
1062     ADD CONSTRAINT team_advisor_advisors_id_key UNIQUE (advisors_id);
1063
1064
1065 --
```

```
-- Name: team_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY team
    ADD CONSTRAINT team_pkey PRIMARY KEY (id);


--
-- Name: tournament_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY tournament
    ADD CONSTRAINT tournament_pkey PRIMARY KEY (year);


--
-- Name: tournamentgroup_match_matches_id_key; Type: CONSTRAINT; Schema: public; Owner:
    postgres; Tablespace:
--

ALTER TABLE ONLY tournamentgroup_match
    ADD CONSTRAINT tournamentgroup_match_matches_id_key UNIQUE (matches_id);


--
-- Name: tournamentgroup_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres;
    Tablespace:
--

ALTER TABLE ONLY tournamentgroup
    ADD CONSTRAINT tournamentgroup_pkey PRIMARY KEY (groupid);


--
-- Name: tournamentgroup_team_teams_id_key; Type: CONSTRAINT; Schema: public; Owner:
    postgres; Tablespace:
--

ALTER TABLE ONLY tournamentgroup_team
    ADD CONSTRAINT tournamentgroup_team_teams_id_key UNIQUE (teams_id);


--
-- Name: fk1fc9f7a09b9d8d6d; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY advisor
    ADD CONSTRAINT fk1fc9f7a09b9d8d6d FOREIGN KEY (id) REFERENCES person(id);


--
-- Name: fk26e2d0c0d36f2a65; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY actor_role
    ADD CONSTRAINT fk26e2d0c0d36f2a65 FOREIGN KEY (actor_email) REFERENCES actor(email);


--
-- Name: fk26e2d0c0e4a0b3e5; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY actor_role
    ADD CONSTRAINT fk26e2d0c0e4a0b3e5 FOREIGN KEY (roles_name) REFERENCES role(name);


--
-- Name: fk26f49674b12939; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--

ALTER TABLE ONLY role
    ADD CONSTRAINT fk26f49674b12939 FOREIGN KEY (tournament_year) REFERENCES tournament(year);


--
-- Name: fk26f496da563832; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY role
    ADD CONSTRAINT fk26f496da563832 FOREIGN KEY (inheritedrole_name) REFERENCES role(name);


--
-- Name: fk27b67dfcf4fc9d; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY team
    ADD CONSTRAINT fk27b67dfcf4fc9d FOREIGN KEY (country_id) REFERENCES country(id);


--
-- Name: fk3525aa1c2801c0aa; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournamentgroup_match
    ADD CONSTRAINT fk3525aa1c2801c0aa FOREIGN KEY (matches_id) REFERENCES match(id);


--
-- Name: fk3525aa1cd57db48a; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournamentgroup_match
    ADD CONSTRAINT fk3525aa1cd57db48a FOREIGN KEY (tournamentgroup_groupid) REFERENCES tournamentgroup(groupid);


--
-- Name: fk3b74360966de6d99; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournament
    ADD CONSTRAINT fk3b74360966de6d99 FOREIGN KEY (finalmatch_id) REFERENCES match(id);


--
-- Name: fk3b7436097d361fb7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournament
    ADD CONSTRAINT fk3b7436097d361fb7 FOREIGN KEY (groupstage_id) REFERENCES groupstage(id);


--
-- Name: fk3b743609fcd043a4; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournament
    ADD CONSTRAINT fk3b743609fcd043a4 FOREIGN KEY (matchforthirdplace_id) REFERENCES match(id);


--
```

```
1194 -- Name: fk46ae9a515c9d2b6; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1195 --
1196
1197 ALTER TABLE ONLY match
1198     ADD CONSTRAINT fk46ae9a515c9d2b6 FOREIGN KEY (stadium_stadiumid) REFERENCES stadium(
         stadiumid);
1199
1200
1201 --
1202 -- Name: fk46ae9a574b12939; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1203 --
1204
1205 ALTER TABLE ONLY match
1206     ADD CONSTRAINT fk46ae9a574b12939 FOREIGN KEY (tournament_year) REFERENCES tournament
         (year);
1207
1208
1209 --
1210 -- Name: fk46ae9a5d4eaebd3; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1211 --
1212
1213 ALTER TABLE ONLY match
1214     ADD CONSTRAINT fk46ae9a5d4eaebd3 FOREIGN KEY (group_groupid) REFERENCES
         tournamentgroup(groupid);
1215
1216
1217 --
1218 -- Name: fk46ae9a5e2487ff; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1219 --
1220
1221 ALTER TABLE ONLY match
1222     ADD CONSTRAINT fk46ae9a5e2487ff FOREIGN KEY (guestteam_id) REFERENCES team(id);
1223
1224
1225 --
1226 -- Name: fk46ae9a5f0ec562f; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1227 --
1228
1229 ALTER TABLE ONLY match
1230     ADD CONSTRAINT fk46ae9a5f0ec562f FOREIGN KEY (hostteam_id) REFERENCES team(id);
1231
1232
1233 --
1234 -- Name: fk49fd4907cdc8c6de; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1235 --
1236
1237 ALTER TABLE ONLY person_team
1238     ADD CONSTRAINT fk49fd4907cdc8c6de FOREIGN KEY (teams_id) REFERENCES team(id);
1239
1240
1241 --
1242 -- Name: fk49fd4907ce781a97; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1243 --
1244
1245 ALTER TABLE ONLY person_team
1246     ADD CONSTRAINT fk49fd4907ce781a97 FOREIGN KEY (person_id) REFERENCES person(id);
1247
1248
1249 --
1250 -- Name: fk5625b34074b12939; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1251 --
1252
1253 ALTER TABLE ONLY tournament_country
1254     ADD CONSTRAINT fk5625b34074b12939 FOREIGN KEY (tournament_year) REFERENCES
         tournament(year);
1255
1256
1257 --
```

```
1258 -- Name: fk5625b3409a553667; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1259 --
1260
1261 ALTER TABLE ONLY tournament_country
1262     ADD CONSTRAINT fk5625b3409a553667 FOREIGN KEY (hostcountries_id) REFERENCES country(
         id);
1263
1264
1265 --
1266 -- Name: fk57f7a1ef94470e9c; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1267 --
1268
1269 ALTER TABLE ONLY permission
1270     ADD CONSTRAINT fk57f7a1ef94470e9c FOREIGN KEY (resource_id) REFERENCES resource(id);
1271
1272
1273 --
1274 -- Name: fk62cd596526daa28; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1275 --
1276
1277 ALTER TABLE ONLY groupstage_tournamentgroup
1278     ADD CONSTRAINT fk62cd596526daa28 FOREIGN KEY (groups_groupid) REFERENCES
         tournamentgroup(groupid);
1279
1280
1281 --
1282 -- Name: fk62cd5967d361fb7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1283 --
1284
1285 ALTER TABLE ONLY groupstage_tournamentgroup
1286     ADD CONSTRAINT fk62cd5967d361fb7 FOREIGN KEY (groupstage_id) REFERENCES groupstage(
         id);
1287
1288
1289 --
1290 -- Name: fk6372df674b12939; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1291 --
1292
1293 ALTER TABLE ONLY tournamentgroup
1294     ADD CONSTRAINT fk6372df674b12939 FOREIGN KEY (tournament_year) REFERENCES tournament
         (year);
1295
1296
1297 --
1298 -- Name: fk8ea387019b9d8d6d; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1299 --
1300
1301 ALTER TABLE ONLY player
1302     ADD CONSTRAINT fk8ea387019b9d8d6d FOREIGN KEY (id) REFERENCES person(id);
1303
1304
1305 --
1306 -- Name: fka1d587dedb6578d7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1307 --
1308
1309 ALTER TABLE ONLY team_advisor
1310     ADD CONSTRAINT fka1d587dedb6578d7 FOREIGN KEY (team_id) REFERENCES team(id);
1311
1312
1313 --
1314 -- Name: fka1d587deea5528a; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1315 --
1316
1317 ALTER TABLE ONLY team_advisor
1318     ADD CONSTRAINT fka1d587deea5528a FOREIGN KEY (advisors_id) REFERENCES advisor(id);
1319
1320
1321 --
```

```
-- Name: fkabf317a774b12939; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournament_stadium
    ADD CONSTRAINT fkabf317a774b12939 FOREIGN KEY (tournament_year) REFERENCES
        tournament(year);


--
-- Name: fkabf317a7db034f8f; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournament_stadium
    ADD CONSTRAINT fkabf317a7db034f8f FOREIGN KEY (stadiums_stadiumid) REFERENCES
        stadium(stadiumid);


--
-- Name: fkb7678b26cdc8c6de; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournamentgroup_team
    ADD CONSTRAINT fkb7678b26cdc8c6de FOREIGN KEY (teams_id) REFERENCES team(id);


--
-- Name: fkb7678b26d57db48a; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY tournamentgroup_team
    ADD CONSTRAINT fkb7678b26d57db48a FOREIGN KEY (tournamentgroup_groupid) REFERENCES
        tournamentgroup(groupid);


--
-- Name: fkbfc93acf60766fd; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY match_matchevent
    ADD CONSTRAINT fkbfc93acf60766fd FOREIGN KEY (match_id) REFERENCES match(id);


--
-- Name: fkbfc93acfdcc26853; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY match_matchevent
    ADD CONSTRAINT fkbfc93acfdcc26853 FOREIGN KEY (events_id) REFERENCES matchevent(id);


--
-- Name: fkcf2436994 01023a7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY actor_permission
    ADD CONSTRAINT fkcf243699401023a7 FOREIGN KEY (permissions_id) REFERENCES permission
        (id);


--
-- Name: fkcf243699d36f2a65; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY actor_permission
    ADD CONSTRAINT fkcf243699d36f2a65 FOREIGN KEY (actor_email) REFERENCES actor(email);


--
```

```
-- Name: fkd4e5f70318f27aa6; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY team_player
    ADD CONSTRAINT fkd4e5f70318f27aa6 FOREIGN KEY (players_id) REFERENCES player(id);


--
-- Name: fkd4e5f703db6578d7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY team_player
    ADD CONSTRAINT fkd4e5f703db6578d7 FOREIGN KEY (team_id) REFERENCES team(id);


--
-- Name: fkd7c3a68b1fd58223; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY match_match
    ADD CONSTRAINT fkd7c3a68b1fd58223 FOREIGN KEY (match_id) REFERENCES match(id);


--
-- Name: fkd7c3a68b884bbd1; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY match_match
    ADD CONSTRAINT fkd7c3a68b884bbd1 FOREIGN KEY (childs_id) REFERENCES match(id);


--
-- Name: fke491d7f560766fd; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY matchevent
    ADD CONSTRAINT fke491d7f560766fd FOREIGN KEY (match_id) REFERENCES match(id);


--
-- Name: fke491d7f56a9e6294; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY matchevent
    ADD CONSTRAINT fke491d7f56a9e6294 FOREIGN KEY (scorringteam_id) REFERENCES team(id);


--
-- Name: fke491d7f5c7168977; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY matchevent
    ADD CONSTRAINT fke491d7f5c7168977 FOREIGN KEY (newplayer_id) REFERENCES player(id);


--
-- Name: fke491d7f5db6578d7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY matchevent
    ADD CONSTRAINT fke491d7f5db6578d7 FOREIGN KEY (team_id) REFERENCES team(id);


--
-- Name: fke491d7f5df1dd7b0; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY matchevent
```

```
1454      ADD CONSTRAINT fke491d7f5df1dd7b0 FOREIGN KEY (involvedplayer_id) REFERENCES player(
              id);
1455
1456
1457 --
1458 -- Name: fkf21d53ddfcf4fc9d; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1459 --
1460
1461 ALTER TABLE ONLY stadium
1462      ADD CONSTRAINT fkf21d53ddfcf4fc9d FOREIGN KEY (country_id) REFERENCES country(id);
1463
1464
1465 --
1466 -- Name: fkf8a56938401023a7; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1467 --
1468
1469 ALTER TABLE ONLY role_permission
1470      ADD CONSTRAINT fkf8a56938401023a7 FOREIGN KEY (permissions_id) REFERENCES permission
              (id);
1471
1472
1473 --
1474 -- Name: fkf8a569386ac4edcc; Type: FK CONSTRAINT; Schema: public; Owner: postgres
1475 --
1476
1477 ALTER TABLE ONLY role_permission
1478      ADD CONSTRAINT fkf8a569386ac4edcc FOREIGN KEY (role_name) REFERENCES role(name);
1479
1480
1481 --
1482 -- PostgreSQL database dump complete
1483 --
```