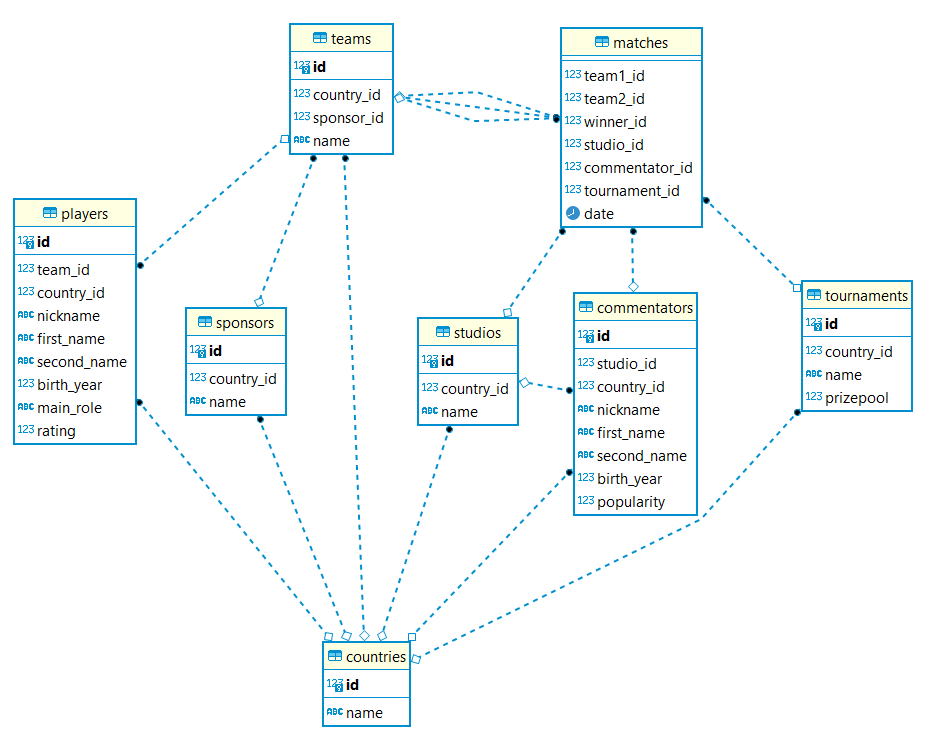
Сущности базы данных



1. Создание В SWI-prolog нету создания таблиц

|  |  |
| --- | --- |
|  | **create** **table** **if** **not** **exists** Countries  (  id serial **primary** **key**,  **name** **text**  ); |
|  | **create** **table** **if** **not** **exists** Sponsors  (  id serial **primary** **key**,  country\_id **int**,  **FOREIGN** **KEY** (country\_id) **REFERENCES** public.Countries (id),  **name** **text**  ); |
|  | **create** **table** **if** **not** **exists** Teams  (  id serial **primary** **key**,  country\_id **int**,  sponsor\_id **int**,  **FOREIGN** **KEY** (country\_id) **REFERENCES** public.Countries (id),  **FOREIGN** **KEY** (sponsor\_id) **REFERENCES** public.Sponsors (id),  **name** **text**  ); |
|  | **create** **table** **if** **not** **exists** Players  (  id serial **primary** **key**,  team\_id **int**,  country\_id **int**,  **FOREIGN** **KEY** (team\_id) **REFERENCES** public.Teams (id),  **FOREIGN** **KEY** (country\_id) **REFERENCES** public.Countries (id),  nickname **text**,  first\_name **text**,  second\_name **text**,  birth\_year **int**,  main\_role **text**,  rating **int**  ); |
|  | **create** **table** **if** **not** **exists** Studios  (  id serial **primary** **key**,  country\_id **int**,  **FOREIGN** **KEY** (country\_id) **REFERENCES** public.Countries (id),  **name** **text**  ); |
|  | **create** **table** **if** **not** **exists** Commentators  (  id serial **primary** **key**,  studio\_id **int**,  country\_id **int**,  **FOREIGN** **KEY** (studio\_id) **REFERENCES** public.Studios (id),  **FOREIGN** **KEY** (country\_id) **REFERENCES** public.Countries (id),  nickname **text**,  first\_name **text**,  second\_name **text**,  birth\_year **int**,  popularity **int**  ); |
|  | **create** **table** **if** **not** **exists** Tournaments  (  id serial **primary** **key**,  country\_id **int**,  **FOREIGN** **KEY** (country\_id) **REFERENCES** public.Countries (id),  **name** **text**,  prizepool **int**  ); |
|  | **create** **table** **if** **not** **exists** Matches  (  team1\_id **int**,  team2\_id **int**,  winner\_id **int**,  studio\_id **int**,  commentator\_id **int**,  tournament\_id **int**,  **FOREIGN** **KEY** (team1\_id) **REFERENCES** public.Teams (id),  **FOREIGN** **KEY** (team2\_id) **REFERENCES** public.Teams (id),  **FOREIGN** **KEY** (winner\_id) **REFERENCES** public.Teams (id),  **FOREIGN** **KEY** (studio\_id) **REFERENCES** public.Studios (id),  **FOREIGN** **KEY** (commentator\_id) **REFERENCES** public.Commentators (id),  **FOREIGN** **KEY** (tournament\_id) **REFERENCES** public.Tournaments (id),  **date** **date**  ); |
|  | **create** **table** **if** **not** **exists** UserRoles  (  id serial **primary** **key**,  **name** **text**  );); |
|  | **create** **table** **if** **not** **exists** Users  (  id serial **primary** **key**,  role\_id **int**,  **FOREIGN** **KEY** (role\_id) **REFERENCES** public.UserRoles (id),  --email text,  login **text**,  **password** **text**  ); |

1. Заполнение

|  |  |
| --- | --- |
| country(1, "Russia").  country(2, "USA").  country(3, "Ukraine"). | **insert** **into** countries **values**(1, 'Russia');  **insert** **into** countries **values**(2, 'USA');  **insert** **into** countries **values**(3, 'Ukraine'); |
| sponsor(1, 1, "GGBet").  sponsor(2, 2, "Red Bull").  sponsor(3, 2, "Coca-cola"). | **insert** **into** sponsors **values**(1, 1, 'GGBet');  **insert** **into** sponsors **values**(2, 2, 'Red Bull');  **insert** **into** sponsors **values**(3, 2, 'Coca-cola'); |
| team(1, 1, 1, "Virtus Pro").  team(2, 2, 2, "Evil Geniuses").  team(3, 3, 3, "Natus Vincere"). | **insert** **into** teams **values**(1, 1, 1, 'Virtus Pro');  **insert** **into** teams **values**(2, 2, 2, 'Evil Geniuses');  **insert** **into** teams **values**(3, 3, 3, 'Natus Vincere'); |
| player(1, 1, 1, "GPK", "Danil", "Skutin", 2001, "Midlaner", 12000).  player(2, 2, 1, "Nightfall", "Egor", "Grigorenko", 2002, "Offlaner", 11000).  player(3, 3, 3, "Noone", "Vladimir", "Minenko", 1997, "Midlaner", 10000). | **insert** **into** players **values**(1, 1, 1, 'GPK', 'Danil', 'Skutin', 2001, 'Midlaner', 12000);  **insert** **into** players **values**(2, 2, 1, 'Nightfall', 'Egor', 'Grigorenko', 2002, 'Offlaner', 11000);  **insert** **into** players **values**(3, 3, 3, 'Noone', 'Vladimir', 'Minenko', 1997, 'Midlaner', 10000); |
| studio(1, 1, "RuHub").  studio(2, 2, "Beyond the Summit").  studio(3, 3, "Maincast"). | **insert** **into** studios **values**(1, 1, 'RuHub');  **insert** **into** studios **values**(2, 2, 'Beyond the Summit');  **insert** **into** studios **values**(3, 3, 'Maincast'); |
| commentator(1, 1, 1, "4ce", "Dmitriy", "Filinov", 1991, 10000).  commentator(2, 2, 2, "Forsaken Oracle", "Kyle", "Freedman", 1993, 11000).  commentator(3, 3, 3, "ALWAYSWANNAFLY", "Andrey", "Bondarenko", 1991, 12000). | **insert** **into** commentators **values**(1, 1, 1, '4ce', 'Dmitriy', 'Filinov', 1991, 10000);  **insert** **into** commentators **values**(2, 2, 2, 'Forsaken Oracle', 'Kyle', 'Freedman', 1993, 11000);  **insert** **into** commentators **values**(3, 3, 3, 'ALWAYSWANNAFLY', 'Andrey', 'Bondarenko', 1991, 12000); |
| tournament(1, 2, "The Inernational 2021", 40018195). | **insert** **into** tournaments **values**(1, 2, 'The Inernational 2021', 40018195); |
| match(1, 2, 1, 1, 1, 1, "2021-11-07").  match(2, 3, 2, 2, 2, 1, "2021-11-08").  match(3, 1, 3, 3, 3, 1, "2021-11-09"). | **insert** **into** matches **values**(1, 2, 1, 1, 1, 1, '2021-11-07');  **insert** **into** matches **values**(2, 3, 2, 2, 2, 1, '2021-11-08');  **insert** **into** matches **values**(3, 1, 3, 3, 3, 1, '2021-11-09'); |

1. Правила – их нет в SQL, но они необходимы для задания вопроса в Prolog

|  |  |
| --- | --- |
| get\_matches(CID, RES):-findall((T1ID, T2ID, WID, SID, CID, TID, DATE), match(T1ID, T2ID, WID, SID, CID, TID, DATE), RES). |  |
| team\_winner(ID, NAME):-team(ID, \_, \_, NAME), match(\_, \_, ID, \_, \_, \_, \_).  team\_winners(RES):-findall((ID, NAME), team\_winner(ID, NAME), RES). |  |
| best\_player(Nickname, Rating):-player(\_, \_, \_, Nickname, \_, \_, \_, \_, Rating), Rating>=11000.  best\_players(RES):-findall((N, R), best\_player(N, R), RES). |  |
| year(Year) :-  get\_time(Stamp),  stamp\_date\_time(Stamp, DateTime, local),  date\_time\_value(year, DateTime, Year).  sum\_r([], Sum0, Sum, \_):- Sum = Sum0, !.  sum\_r([H|T], Sum0, Sum, Y):-Sum1 is Sum0 + Y - H, sum\_r(T, Sum1, Sum, Y).  sum(List, Sum):-year(Year), sum\_r(List, 0, Sum, Year).  get\_avgage(Role, Avg):-findall(Age, player(\_, \_, \_, \_, \_, \_, Age, Role, \_), Res), sum(Res, Sum), proper\_length(Res, Len), Avg is Sum/Len.  get\_maxage(Role, Max):-findall(Age, player(\_, \_, \_, \_, \_, \_, Age, Role, \_), Res), min\_list(Res, MaxY), year(Y), Max is Y-MaxY.  get\_minage(Role, Min):-findall(Age, player(\_, \_, \_, \_, \_, \_, Age, Role, \_), Res), max\_list(Res, MinY), year(Y), Min is Y-MinY.  get\_age\_stats(RES):-findall(("Midlaner", Avgage, Minage, Maxage), (get\_avgage("Midlaner", Avgage),get\_maxage("Midlaner", Maxage),get\_minage("Midlaner", Minage)), R1),  findall(("Offlaner", Avgage, Minage, Maxage), (get\_avgage("Offlaner", Avgage),get\_maxage("Offlaner", Maxage),get\_minage("Offlaner", Minage)), R2),  append(R1, R2, RES). |  |

1. Поиск

|  |  |
| --- | --- |
| get\_matches(2, MATCHES). | --Все матчи с конкретным комментатором  **select** \*  **from** matches  **where** commentator\_id = 2 |
| team\_winners(TEAMS). | --Все команды которые выиграли хотябы 1 матч  **select** id, **name**  **from** teams t  **where** **exists** (**select** id  **from** matches m  **where** t.id = m.winner\_id) |
| best\_players(PLAYERS). | **select** nickname, rating  **from** players  **where** rating > 11000 |
| get\_age\_stats(STATS). | --Средние, минимальные и максимальные возраста для ролей  **select** **distinct** main\_role,  **avg**(**extract**(**year** **from** **CURRENT\_DATE**) - birth\_year) **over**(**partition** **by** p.main\_role) **as** AvgAge,  **min**(**extract**(**year** **from** **CURRENT\_DATE**) - birth\_year) **over**(**partition** **by** p.main\_role) **as** MinAge,  **max**(**extract**(**year** **from** **CURRENT\_DATE**) - birth\_year) **over**(**partition** **by** p.main\_role) **as** MaxAge  **from** players p |