

פרויקט גמר - SQL

טורניר תחרויות רובוטיקה

שם המרצה: איריס
שם המגיש: איתן רפאל צ'רטוף
ת.ז.: 215310715

תוכן עניינים:

2	תוכן עניינים:
4	מבוא ותיאור מערכת
5	מטרות המערכת
5	משתמשי המערכת
6	ישויות
6	אילוצים
6	ישויות קשר הנובעות מאילוצים:
6	מימוש
8	ERD - Entity-Relationship Model
9	DSD - Data Structure Diagram
10	טבלאות
10	Alliances
10	Awards
11	Awards_in_Competitions
11	Cities
12	Competitions
12	Countries
13	Districts
13	Groups
14	Matches
14	Matches_has_Teams
15	Match_Type
15	Pits
16	Pits_in_Competition
16	Teams
16	Teams_in_Alliance
17	Teams_in_Competition
18	רשימת שאלות
18	1. AddCompetition
18	2. UpdateTeamCity
19	3. GetTopScoringTeamInGroups
19	4. GetTeamWithMostWinsInDistrict
20	5. GetSharedCompetitions

21.....	PreventDuplicateTeamInAlliance .6
22.....	GetCompetitionAwardsWithTeams .7
22.....	PredictWinner .8
24.....	PredictWinnerWithPrecent .9
25.....	GetTeamWinRate .10
26.....	CreateChampionship .11

מבוא ותיאור מערכת

חברת "First", חברה העוסקת בניהול תחרויות רובטיקה בכל רחבי העולם, מעוניינת לשמור נתונים על התחרויות שלה, בשביל לנהל את העסק שלהם יותר טוב. החברה מפיקה תחרויות מקומיות ועולמיות בכל רחבי העולם, בין עם זה קליפורניה או רעננה. בכל תחרות משחקים מספר קבוצות, אם זה תחרות מקומית אזי רוב הקבוצות יהיו מהאזור, ואם זה תחרות עולמית הקבוצות יכולות להיות מכל מקום. לכל קבוצה יש רובוט שבנו לפני התחרות בשביל להתמודד עם האתגר הספציפי של אותה תחרות. בדר"כ לכל אתגר יהיו שני סוגים של מטרות, אחת המטרה הראשית, שיכולה להיות משהו רציף, כמו לקלוע לסל. המטרה המשנית הינה הרבה מטרות חד-פעמיות, כמו לטפס על קיר. מקבלים נקודות בכך שמשלימים את מטרות המשחק. המשחקים משוחקים בבריתות, כאשר בכל ברית מספר רובוטים של קבוצות שונות, ומטרת הברית הינה להשיג יותר נקודות מהברית השנייה.

- החברה החליטה שלכל טורניר יהיה מבנה מוסכם, שיהיה פייר אך תחרותי. מהלך טורניר יראה כך:
1. יוחלטו על מספר משחקים שבהם בריתות ישחקו אחד בשני. זה לצורך צבירת נקודות והיבת ליכולות שאר הקבוצות. כל ברית בכל משחק כן היא שונה, כלומר במשחק אחד קבוצה א יכולה לשחק ביחד עם קבוצה ב נגד קבוצות ג ו-ד, ומשחק הבא קבוצה ג ביחד עם קבוצה ה תשחק נגד קבוצות ד ו-ו.
 2. לאחר שכל המשחקים שוחקו, חבורה של הקבוצות הכי מצליחות בוחרות בקבוצות אחרות ליצור איתם ברית ממושכת (alliance), ואם אותם בריתות ממושכות ישחקו שאר המשחקים בטורניר.
 3. מוחלט עוד פעם על משחקים הבריתות החדשות, אך הבריתות המשחקות אחת נגד השנייה הם הבריתות הממושכות. משחקים אלו משוחקים בסוג טורניר בסיסי, כלומר שני בריתות במשחק, אחת מנצחת ונשארת, השנייה מפסידה ויוצאת, ומנצחים משחקים אחד נגד השני עד שנשארה קבוצה אחת. בדר"כ מספר הבריתות יהיה חזקה של שניים בשביל מבנה טורניר נחמד, ורוב הזמן מספר הבריתות יהיה 8.

בסוף כל טורניר קבוצות יכולות לקבל פרסים על דברים שעשו, כמו פרס התכנות הכי טוב, הנדסת הרובוט הכי טובה, ניצחון בטורניר ועוד... בנוסף, לכל קבוצה יהיה מקום פרטי משלה שתוכל לתקן את הרובוט עם יש בו תקלה, בשביל שיוכלו להמשיך לשחק.

בשביל להשתתף באותם טורנירים, אנשים צריכים להצטרף לקבוצה. הקבוצות הם גופים שאושרו ע"י החברה להשתתף בתחרויות. לכל קבוצה יש שם, מספר, כתובת ועוד... בנוסף לכך כל טורניר חייב להיות מנוהל ע"י החברה, ואי אפשר לנהל טורניר בלעדיהם. בכל תחרות מוקצה לקבוצות מקום באזור התחרות, בשביל תיקון הרובוט כאשר יש תקלה. האזור הזה נקרא הפיט.

מטרות המערכת

להריץ טורנירים ברמה כזאת בכל רחבי העולם זה לא קל, ובשביל לנהל אותם כך שהם יהיו מצליחים ובלי בעיות, החברה חייבת מידע ניהולי מאורגן ובשפע. מסד נתונים גדול ונוח, המעוצב לאותם שימושים יכול לשנות את השיטה הניהולית של הטורנירים ולהפיק ייתרונות רבים בשביל החברה:

- **מידע ניהולי מרוכז:** כל המידע על טורנירים, קבוצות, משחקים ועוד מאוחסן במקום אחד, ועוזר למנהלי הטורנירים לגשת למידע שהם צריכים בקלות.
- **אוטומציה:** ארגון נתונים על טורנירים יכול לעזור למפתחים ליצור מערכות שיעשו תהליכים אוטומטיים בתוכנה, כמו ליצור משחקים בין בריתות.
- **אנליזה:** ארגון כמות מידע גדולה כמו זו יכולה להפיק מסקנות על חלקים שונים של התחרות, וניתוח המידע יפיק תוצאות שישנו את התחרויות לטובה.

בנוסף, מנהלי הטורניר רוצים שקבוצות יוכלו לעקוב אחרי תהליך הטורניר. ארגון המידע במסד נתונים קריא לקהילה יפשט את תהליך התקשורת בין החברה והקבוצות.

- **עדכונים עכשוויים:** משתתפים יקבלו עדכוני מידע לגבי תחרויות, משחקים וקבוצות בקלות.
- **הפשטת ניהול קבוצה:** מקום ריכוזי בו קבוצות יקבלו מידע על ההשתתפות שלהם בתחרויות קודמות, ומקום שמנהלים יודעים מה קורה לכל קבוצה. קבוצות לא ייצתרו ליצור מאגרי נתונים משלהם ולבזבז משאבים.
- **שיתוף פעולה:** ארגון המידע במקום שכולם יכולים לקרוא יגרום לקבוצות להבחין בקבוצות דומות, מה שייגרום לשיתוף פעולה וקהילה שיתופית יותר. בנוסף עוד אנשים שהם לא בהכרח קשורים לתחרויות יוכלו להשתמש בנתונים. לדוגמה, אם תחרות מתקיימת ביום מסוים באולם, מנהל האולם יוכל להתעדכן על מה שקורה באותו אולם.
- **שיפור שקיפות:** קבוצות יוכלו להבין בקלות את תהליך התחרויות ואיך מנהלים אותם, מה שיוריד מבלבול שיכול להגיע ממורכבות התחרות.
- **אסטרטגיה:** קבוצות יוכלו לבנות אסטרטגיות יותר חכמות בהתחשב למידע שיש להם, מה שיגרום לטורנירים תחרותיים יותר וחשיבה אסטרטגית וחכמה.

משתמשי המערכת

כפי שנאמר מקודם, הרבה אנשים יכולים להשתמש במערכת זו בשביל לשפר את תהליך טורנירי הרובוטיקה. האנשים שיישתמשו במערכת כזאת יהיו:

- חברת first: יהיו אחראים על המידע הכתוב עם גישה לכל המידע
- קבוצות: גישה למידע שחברת first נותנת להם, וגישות קריאה בלבד.
- הציבור: גישה למידע שיהיה חשוב לדעת מחוץ לנתוני משחקים, כמו מתי קוראת תחרות, ואיפה.

ישויות

- **קבוצות (Teams)** - קבוצות הקיימות בארגון
- **משחקים (Matches)** - כל המשחקים ששוחקו
- **תחרויות (Competitions)** - כל התחרויות
- **פרסים (Awards)** - כל סוגי הפרסים שחולקו
- **בריתות (Alliances)** - כל הבריתות שהוקמו (בחלק השני של התחרות)
- **סוג משחק (Match Types)** - סוגי המשחק שיכולים להיות
- **חבורות (Groups)** - סוגי קבוצות של קבוצות במשחק
- **פיטים (Pits)** - פיטים שהיו בתחרות
- **ערים (Cities)**
- **אזורים (Districts)**
- **מדינות (Countries)**

אילוצים

- כל תחרות ומשחק וברית יכולים להכיל הרבה קבוצות. הקבוצות באותם ישויות יהיו שונות אחת מהשנייה.
- קבוצה אחת הייתה יכולה להיות בהרבה משחקים.
- בתחרות יכול להיות הרבה פיטים, שכל אחד מהפיטים שייך לקבוצה שונה בתחרות
- תחרות יכולה להיות רק בעיר אחת
- קבוצה בתחרות יכולה להיות רק בברית אחת.
- בתחרות אפשר לחלק מספר רב של פרסים, שחולקו להרבה קבוצות.
- שנת ההתחלה של קבוצות, משחקים ותחרויות לא יכול להיות קטן יותר מן 1991.
- הקבוצות במשחק או בתחרות חייבים להיות שונים.
- לכל קבוצה במשחק יהיה את סוג החבורה שהייתה בא, ולכל משחק יהיה סוג משחק.

ישויות קשר הנובעות מאילוצים:

- **קבוצות במשחק**
- **קבוצות בתחרות**
- **קבוצות בברית**
- **פיטים בתחרות**
- **פרסים בתחרות**

מימוש

-- Checks if rookie_year is more then 1991, because the first team

was created in 1992.

```
ALTER TABLE Teams
```

```
ADD CONSTRAINT check_rookie_year
```

```
CHECK (rookie_year > 1991);
```

```
-- Check if each team in match is different. Can't have one team in  
one match twice.
```

```
ALTER TABLE Matches_has_Teams
```

```
ADD CONSTRAINT unique_team_in_same_match
```

```
UNIQUE (matches_id, teams_id, groups_id);
```

```
-- Check if each team in same comp is different.
```

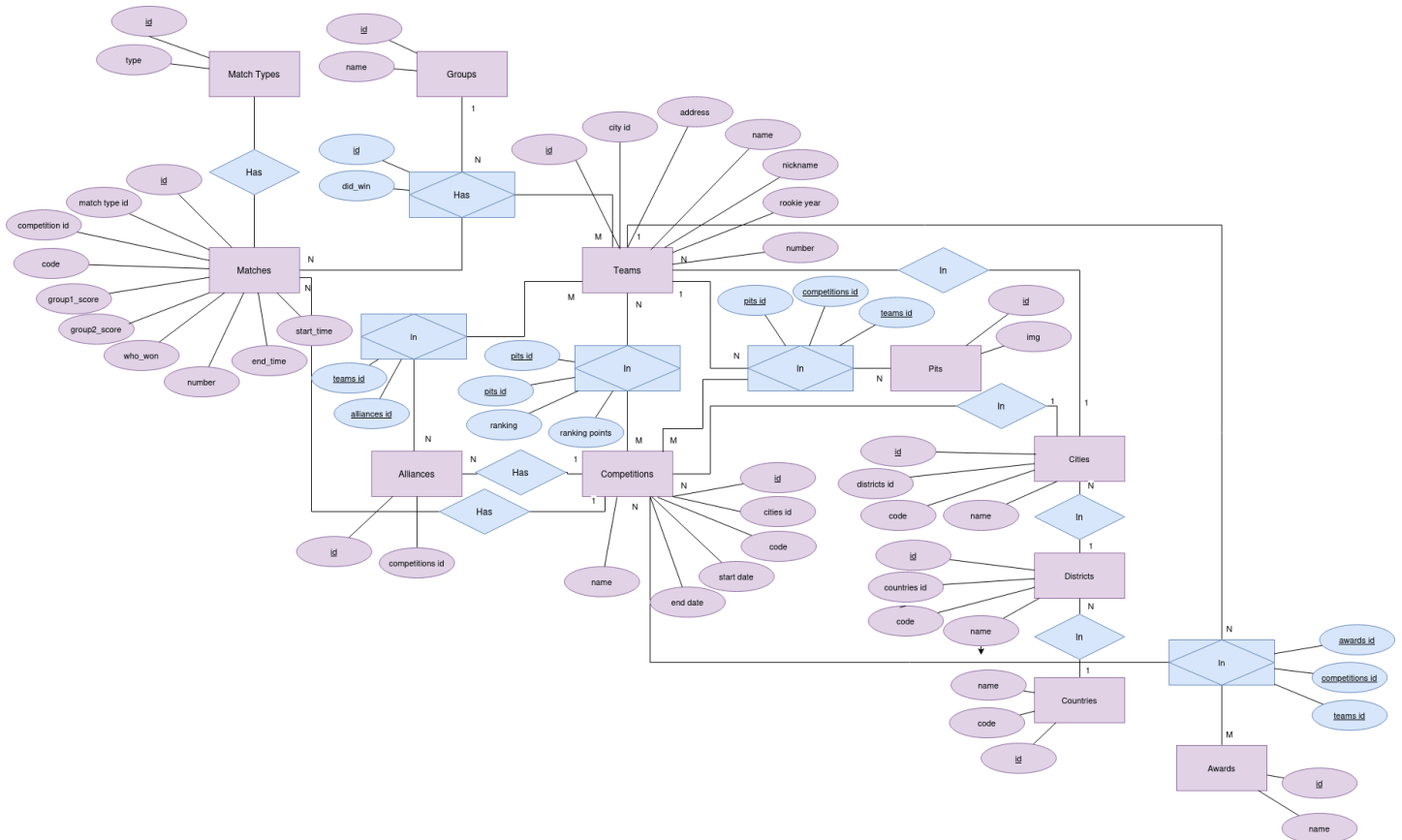
```
ALTER TABLE Teams_in_Competitions
```

```
ADD CONSTRAINT unique_team_in_same_competition
```

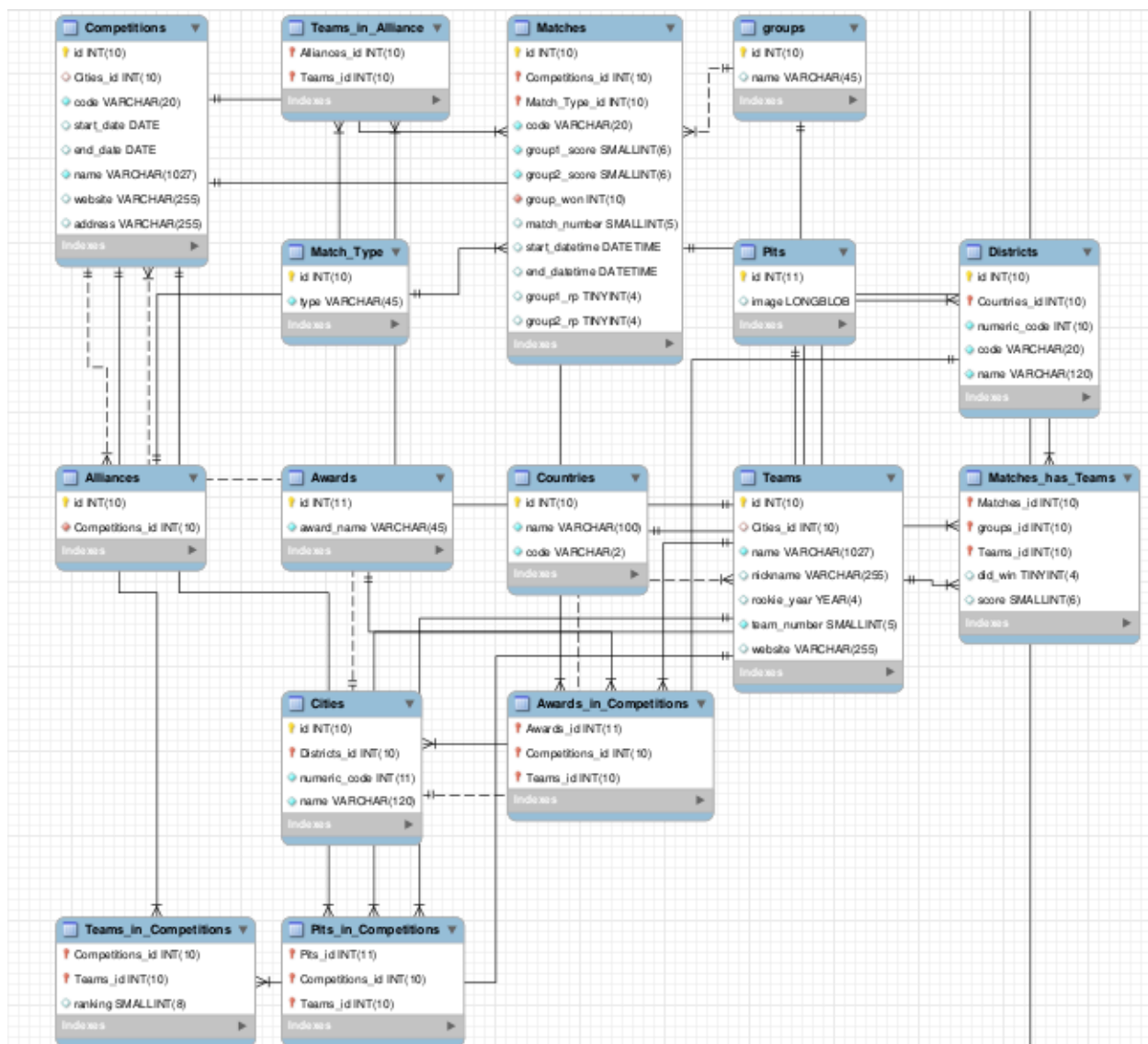
```
UNIQUE (competitions_id, teams_id);
```

שאר המימושים מתוארים במהלך הספר (בתוך תרשימים, וכו...).

ERD - Entity-Relationship Model



DSD - Data Structure Diagram



טבלאות

Alliances

id	int(10) unsigned	NO		select,insert,update,reference auto_incremer
Competitions_id	int(10) unsigned	NO		select,insert,update,reference

id	Competitions_i
3	3515
4	3515
5	3515
6	3515
7	3515
8	3515
9	3515
10	3515
11	3648

Awards

id	int(11)	NO		select,insert,update,reference auto_incremer	
award_name	varchar(45)	NO	latin1	latin1_swedish_ci	select,insert,update,reference

an award is given to a team who has shown:
 - extraordinary effort in a given field
 - a unique and out of the box result
 - success in the tournament
 - and more...

each award has a unique purpose.

50	AGAINST_ALL_ODDS
34	AUTODESK_INVENTOR
72	AUTONOMOUS
8	BART_KAMEN_MEMORIAL
62	BEST_CRAFTSMANSHIP
63	BEST_DEFENSIVE_MATCH
44	BEST_OFFENSIVE_ROUND

Awards_in_Competitions

◆ Awards_id	int(11)	NO	select,insert,update,reference	
◆ Competitions_id	int(10) unsigned	NO	select,insert,update,reference	
◆ Teams_id	int(10) unsigned	NO	select,insert,update,reference	the team that is given the award

Awards_id	Competitions_id	Teams_id
1	3321	15541
2	3321	15492
3	3321	15422
3	3321	15479
3	3321	15540
20	3321	15479
21	3321	15492
31	3321	15541
44	3321	15496

Cities

◆ id	int(10) unsigned	NO	select,insert,update,reference	auto_increment	
◆ Districts_id	int(10) unsigned	NO	select,insert,update,reference		
◆ numeric_code	int(11)	NO	select,insert,update,reference		each city is given a unique code, so they can be easily identified by users.
◆ name	varchar(120)	NO	utf8 utf8_general_ci select,insert,update,reference		official name of the city

id	Districts_id	numeric_code	name
30...	10169	52	Ashkāsham
30...	10169	68	Fayzabad
30...	10169	78	Jurm
30...	10169	84	Khandūd
30...	10169	115	Rāghistān
30...	10169	131	Wākhān
30...	10170	72	Ghormach
30...	10170	108	Qala i Naw
30...	10171	54	Baghlān

Competitions

id	int(10) unsigned	NO			select,insert,update,reference auto_incremer	
Cities_id	int(10) unsigned	YES			select,insert,update,reference	
code	varchar(20)	NO	latin1	latin1_swedish	select,insert,update,reference	unique code given to a comp, so it can be easily identified by a name. codes will have a naming convention, for exam - isr2019 so it can be easily identified by users.
start_date	date	YES			select,insert,update,reference	starting date of a comp
end_date	date	YES			select,insert,update,reference	ending date of a comp
name	varchar(1027)	NO	utf8	utf8_general_c	select,insert,update,reference	name if comp
website	varchar(255)	YES	utf8	utf8_general_c	select,insert,update,reference	link for a website that is realted to the competition
address	varchar(255)	YES	utf8	utf8_general_c	select,insert,update,reference	address of a competition. not constrained by a key so it can be very specific for local purposes.

id	Cities_id	code	start_date	end_date	name	website	address
3503	434108	2006ga	2006-03-16	2006-03-18	Peachtree Regional	http://...	2400 Chattahoochee Dr, Duluth, G...
3504	433973	2006gal	2006-04-12	2006-04-14	Galileo Division	http://...	Atlanta, GA, USA
3505	439039	2006gl	2006-03-09	2006-03-11	Great Lakes Regional	NULL	2064 Whittaker Rd, Ypsilanti, MI 48...
3506	434880	2006il	2006-03-16	2006-03-18	Midwest Regional	http://...	1455 S Wabash Ave, Chicago, IL 6...
3507	436022	2006in	2006-03-16	2006-03-18	Boilermaker Regional	http://...	128 Memorial Mall, West Lafayette,...
3508	NULL	2006is	2006-03-28	2006-03-30	GM/Technion Israel Regional		95 Rokah Blvd, תל אביב יפו, Israel
3509	442107	2006li	2006-03-23	2006-03-25	SBPLI Long Island Regional	http://...	173A Fulton Ave, Hempstead, NY 1...
3510	438163	2006ma	2006-03-23	2006-03-25	Boston Regional	http://...	303 Boylston St, Brookline, MA 024...
3511	437755	2006md	2006-03-16	2006-03-18	Chesapeake Regional	http://...	303 2nd St, Annapolis, MD 21403, ...

Countries

id	int(10) unsigned	NO			select,insert,update,reference auto_incremer	
name	varchar(100)	NO	utf8	utf8_general_c	select,insert,update,reference	official name of country
code	varchar(2)	NO	latin1	latin1_swedish	select,insert,update,reference	each country is given a unique code, so they can be easily identified by users.

id	name	code
501	Afghanistan	AF
502	Aland Islands	AX
503	Albania	AL
504	Algeria	DZ
505	American Sa...	AS
506	Andorra	AD
507	Angola	AO
508	Anguilla	AI
509	Antarctica	AQ

Districts

id	int(10) unsigned	NO		select,insert,update,reference auto_increment	
Countries_id	int(10) unsigned	NO		select,insert,update,reference	
numeric_code	int(10) unsigned	NO		select,insert,update,reference	each district is given a unique numeric code, so they can be easily identified by users.
code	varchar(20)	NO	latin1	latin1_swedish_ci select,insert,update,reference	each district is given a unique code, so they can be easily identified by users.
name	varchar(120)	NO	utf8	utf8_general_ci select,insert,update,reference	official name of the district

id	Countries_id	numeric_code	code	name
10...	501	3901	BDS	Badakhshan
10...	501	3871	BDG	Badghis
10...	501	3875	BGL	Baghlan
10...	501	3884	BAL	Balkh
10...	501	3872	BAM	Bamyan
10...	501	3892	DAY	Daykundi
10...	501	3899	FRA	Farah
10...	501	3889	FYB	Faryab
10...	501	3870	GHA	Ghazni

Groups

id	int(10) unsigned	NO		select,insert,update,reference auto_increment	
name	varchar(45)	YES	latin1	latin1_swedish_ci select,insert,update,reference	a group is an identifier for a set of teams in a match. in a match there is two groups/alliances fighting each other. groups and alliances are different from each other: - groups are formed by the competition managers beforehand, and they represent sets of teams that fight each other in the first part of the comp, and they are the sets of teams that fight each other in the second part of the comp. - alliances are formed by teams in the second part of the comp, and they are the sets of teams that fight each other in the first part of the comp.

id	name
3	blue
4	red

Matches

id	int(10) unsigned	NO			select,insert,update,reference auto_incremer	
Competitions_id	int(10) unsigned	NO			select,insert,update,reference	
Match_Type_id	int(10) unsigned	NO			select,insert,update,reference	
code	varchar(20)	NO	latin1	latin1_swedish_ci	select,insert,update,reference	a unique code given to a match. matches in the same competition will have a naming convention.
group1_score	smallint(6)	NO			select,insert,update,reference	score of group/alliance 1 in the match.
group2_score	smallint(6)	NO			select,insert,update,reference	score of group/alliance 2 in the match.
group_won	tinyint(4)	NO			select,insert,update,reference	identified which group won
match_number	smallint(5) unsigned	YES			select,insert,update,reference	the number of the match in a comp.
start_datetime	datetime	YES			select,insert,update,reference	
end_datetime	datetime	YES			select,insert,update,reference	
group1_rp	tinyint(4)	YES			select,insert,update,reference	ranking points got by group/alliance 1 in the match.
group2_rp	tinyint(4)	YES			select,insert,update,reference	ranking points got by group/alliance 2 in the match.

id	Competitions_id	Match_Type_id	code	group1_score	group2_score	group_won	match_number	start_datetime	end_datetime	group1_rp	group2_rp
344631	5135	10	2019casd_qm15	42	30	3	15	2019-03-08 ...	2019-03-08...	2	0
344632	5135	10	2019casd_qm16	38	57	4	16	2019-03-08 ...	2019-03-08...	1	3
344633	5135	10	2019casd_qm17	48	50	4	17	2019-03-08 ...	2019-03-08...	1	3
344634	5135	10	2019casd_qm18	34	63	4	18	2019-03-08 ...	2019-03-08...	0	3
344635	5135	10	2019casd_qm19	43	24	3	19	2019-03-08 ...	2019-03-08...	3	0
344636	5135	10	2019casd_qm20	24	14	3	2	2019-03-08 ...	2019-03-08...	2	0
344637	5135	10	2019casd_qm21	36	66	4	20	2019-03-08 ...	2019-03-08...	1	3
344638	5135	10	2019casd_qm22	46	40	3	21	2019-03-08 ...	2019-03-08...	2	0
344639	5135	10	2019casd_qm23	56	43	3	22	2019-03-08 ...	2019-03-08...	3	0
344640	5135	10	2019casd_qm24	38	29	3	23	2019-03-08 ...	2019-03-08...	2	0
344641	5135	10	2019casd_qm25	42	35	3	24	2019-03-08 ...	2019-03-08...	2	0
344642	5135	10	2019casd_qm26	36	55	4	25	2019-03-09 ...	2019-03-09...	1	3

Matches_has_Teams

Matches_id	int(10) unsigned	NO			select,insert,update,reference	
groups_id	int(10) unsigned	NO			select,insert,update,reference	
Teams_id	int(10) unsigned	NO			select,insert,update,reference	
did_win	tinyint(4)	YES			select,insert,update,reference	did group/alliance of the team in match won.
score	smallint(6)	YES			select,insert,update,reference	score of the whole group/alliance in the match.

Matches_id	groups_id	Teams_id	did_win	score
205466	3	15404	0	20
205466	3	15422	0	20
205466	4	15479	1	32
205466	4	15529	1	32
205467	3	15422	1	37
205467	3	15662	1	37
205467	4	15453	0	31

Match_Type

id	int(10) unsigned	NO		select,insert,update,reference auto_increment	
type	varchar(45)	NO	latin1	latin1_swedish_ci select,insert,update,reference	each match has a type. a type can represent if a match is a match in the first part, the second part, the finals and mon

id	type
7	ef
9	f
6	qf
10	qm
8	sf

Pits

id	int(11)	NO	select,insert,update,reference auto_increment
image	longblob	YES	select,insert,update,reference

id	image
1	BLOB
2	BLOB
3	BLOB
4	BLOB
5	BLOB
6	BLOB
7	BLOB

Pits_in_Competition

Pits_id	int(11)	NO	select,insert,update,reference
Competitions_id	int(10) unsigned	NO	select,insert,update,reference
Teams_id	int(10) unsigned	NO	select,insert,update,reference

Pits_id	Competitions_id	Teams_id
37	3515	15465
44	3534	16816
27	3588	17090
71	3589	15969
86	3589	16727
79	3603	16302
57	3680	16042

Teams

id	int(10) unsigned	NO	select,insert,update,reference	auto_incremer
Cities_id	int(10) unsigned	YES	select,insert,update,reference	
name	varchar(1027)	NO	utf8 utf8_general_c	select,insert,update,reference
nickname	varchar(255)	YES	utf8 utf8_general_c	select,insert,update,reference each team can decide on a nickname for themselves
rookie_year	year(4)	YES	select,insert,update,reference	the first year the team participated in a competition
team_number	smallint(5) unsigned	NO	select,insert,update,reference	each team has a unique number. teams can forfeit, be deleted, and change number, so its best to not use it as an iden
website	varchar(255)	YES	utf8 utf8_general_c	select,insert,update,reference

id	Cities_id	name	nickname	rookie_year	team_number	website
15...	435373	FCA Foundation/Hirotec America/...	The Juggernauts	1997	1	http://www.juggernautteam1.com
15...	432491	Gene HAAS Foundation/Raytheon/...	Team 4 ELEMENT	1997	4	http://www.team4element.com
15...	438833	Ford FIRST Robotics & Melvindale ...	Robocards	1998	5	http://www.robocards.org/
15...	428611	ATK (Alliant Techsystems) & Hopki...	The CogSquad	1994	6	http://www.cogsquad.com/
15...	437765	Lockheed Martin/Northrop Grumm...	Team007	1997	7	http://www.team007.org

Teams_in_Alliance

Alliances_id	int(10) unsigned	NO	select,insert,update,reference
Teams_id	int(10) unsigned	NO	select,insert,update,reference

Alliances_id	Teams_id
3	15510
3	15555
3	16255
4	15419
4	15421
4	16216
5	15499

Teams_in_Competition

Competitions_id	int(10) unsigned	NO	select,insert,update,reference	
Teams_id	int(10) unsigned	NO	select,insert,update,reference	
ranking	smallint(8)	YES	select,insert,update,reference	at what place the team was at the end of the comp
3515	15419	24		
3515	15421	12		
3515	15433	3		
3515	15465	47		
3515	15467	21		
3515	15491	33		
3515	15496	18		

רשימת שאילות

1. AddCompetition

פרוצדורה המקבלת את כל הפרמטרים של תחרות ומסופיה תחרות מתאימה לטבלת התחרויות.

```
CREATE PROCEDURE AddCompetition(
    IN city_id INT,
    IN code VARCHAR(20),
    IN start_date DATE,
    IN end_date DATE,
    IN name VARCHAR(1027),
    IN website VARCHAR(255),
    IN address VARCHAR(255)
)
BEGIN
    INSERT INTO Competitions (Cities_id, code, start_date, end_date,
name, website, address)
    VALUES (city_id, code, start_date, end_date, name, website,
address);
END $$
```

2. UpdateTeamCity

פרוצדורה המקבלת מפתח קבוצה ומפתח עיר, ומעדכנת את העיר של אותה קבוצה לאותה עיר.

```
CREATE PROCEDURE RegisterTeamForCompetition(
    IN team_id INT,
    IN competition_id INT
)
BEGIN
    INSERT INTO Teams_in_Competitions (Competitions_id, Teams_id)
    VALUES (competition_id, team_id);
END $$
```

3. GetTopScoringTeamInGroups

פונקציה המקבלת מפתח של תחרות, ומחזירה את המפתח של הקבוצה שהבריתות ביחד איתה עשו את הסכום הכי גדול של נקודות בתחרות. הפונקציה הזאת שונה מפונקציה שמחזירה את הקבוצה שעשתה הכי הרבה נקודות לבדה בתחרות.

```
CREATE FUNCTION GetTopScoringTeamInGroups(
    competition_id INT
)
RETURNS INT
BEGIN
    DECLARE top_scorer_id INT DEFAULT NULL;
    DECLARE top_score INT DEFAULT 0;

    SELECT Teams_id, SUM(CASE WHEN mht.groups_id =
        (SELECT g.id FROM groups g WHERE g.`name` = "blue")
        THEN m.group1_score ELSE m.group2_score END) AS total_score
    FROM Teams t
    INNER JOIN Matches_has_Teams mht ON t.id = mht.Teams_id
    INNER JOIN Matches m ON mht.Matches_id = m.id
    WHERE m.Competitions_id = competition_id
    GROUP BY Teams_id
    ORDER BY total_score DESC
    LIMIT 1
    INTO top_scorer_id, top_score;

    RETURN top_scorer_id;
END $$
```

4. GetTeamWithMostWinsInDistrict

פונקציה המקבלת מפתח של דיסטריקט, ומחזירה את מפתח הקבוצה שניצחה הכי הרבה בתוך הדיסטריקט.

```
CREATE FUNCTION GetTeamWithMostWinsInDistrict(districtId INT(10)
UNSIGNED) RETURNS INT(10) UNSIGNED
BEGIN
    DECLARE teamId INT(10) UNSIGNED;
```

```
-- Get the team with the most overall match wins in the
district
SELECT
    Teams.id INTO teamId
FROM
    robotics.Matches_has_Teams
JOIN robotics.Matches ON Matches_has_Teams.Matches_id =
Matches.id
JOIN robotics.Teams ON Matches_has_Teams.Teams_id = Teams.id
WHERE
    Teams.Cities_id IN (
        SELECT
            Cities_id
        FROM
            robotics.Districts
        WHERE
            Districts.id = districtId
    )
AND Matches_has_Teams.did_win = 1
GROUP BY
    Teams.id
ORDER BY
    COUNT(Matches_has_Teams.Teams_id) DESC
LIMIT 1;

RETURN teamId;
END $$
```

5. GetSharedCompetitions

פרוצדורה המקבלת שני מפתחות של קבוצות, ומחזירה את כל התחרויות שהיו בהן יחד.

```
CREATE PROCEDURE GetSharedCompetitions(
    IN team1_id INT,
    IN team2_id INT
)
BEGIN
    SELECT DISTINCT c.name AS competition_name, c.start_date,
```

```

c.end_date
FROM Competitions c
INNER JOIN Teams_in_Competitions tic1 ON c.id =
tic1.Competitions_id
INNER JOIN Teams_in_Competitions tic2 ON c.id =
tic2.Competitions_id
WHERE tic1.Teams_id = team1_id
AND tic2.Teams_id = team2_id
AND tic1.Teams_id != tic2.Teams_id -- Avoid self-joins (team
competing against itself)
ORDER BY c.start_date ASC;
END $$

```

6. PreventDuplicateTeamInAlliance

```

CREATE TRIGGER PreventDuplicateTeamInAlliance
BEFORE INSERT ON Teams_in_Alliance
FOR EACH ROW
BEGIN
    DECLARE existing_team_id INT;

    /* Check if a team with the same ID already exists in the alliance
    */
    SELECT TOP 1 id
    FROM Teams_in_Alliance
    WHERE Alliances_id = NEW.Alliances_id
    AND Teams_id = NEW.Teams_id
    INTO existing_team_id;

    /* If a duplicate is found, raise an error and rollback the
    transaction */
    IF existing_team_id IS NOT NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Team with ID ' +
CONVERT(VARCHAR(10), NEW.Teams_id) + ' already exists in this
alliance.';
    END IF;
END $$

```

7. GetCompetitionAwardsWithTeams

פרוצדורה המקבלת מפתח של תחרות ומראה את שם התחרות, שם פרס מתוך התחרות ואת הקבוצה שהפרס הגיע אליה.

```
CREATE PROCEDURE GetCompetitionAwardsWithTeams(
    IN competition_id INT
)
BEGIN
    SELECT
        c.name AS competition_name,
        a.award_name AS award_name,
        t.name AS team_name,
        COUNT(*) AS awarded_count
    FROM Competitions c
    INNER JOIN Awards_in_Competitions aic ON c.id = aic.Competitions_id
    INNER JOIN Awards a ON aic.Awards_id = a.id
    INNER JOIN Teams_in_Competitions tic ON c.id = tic.Competitions_id
    INNER JOIN Teams t ON tic.Teams_id = t.id
    WHERE tic.Teams_id = aic.teams_id -- Team awarded based on
    Awards_in_Competitions
    AND c.id = competition_id -- Filter by competition ID
    GROUP BY c.id, a.id, t.id
    ORDER BY c.name ASC, a.award_name ASC, t.name ASC;
END $$
```

8. PredictWinner

פונקציה המקבלת שתי קבוצות, ובעזרת חישוב ממוצעי הנקודות שלהם, יחליט מי ינצח מבין שניהם. אם יש שגיאה או תיקו, הפונקציה תחזיר 0.

```
CREATE FUNCTION PredictWinner(
    team1_id INT(10) UNSIGNED,
    team2_id INT(10) UNSIGNED
)
RETURNS INT(10)
```

BEGIN**DECLARE** team1_avg **DECIMAL**(10,2);**DECLARE** team2_avg **DECIMAL**(10,2);

/* Calculate average score for team 1 */

SELECT **AVG**(score) **INTO** team1_avg**FROM** Matches_has_Teams mht**JOIN** Matches m **ON** mht.Matches_id = m.id**WHERE** mht.Teams_id = team1_id**AND** mht.score **IS NOT NULL**; /* Exclude matches with no score */

/* Calculate average score for team 2 */

SELECT **AVG**(score) **INTO** team2_avg**FROM** Matches_has_Teams mht**JOIN** Matches m **ON** mht.Matches_id = m.id**WHERE** mht.Teams_id = team2_id**AND** mht.score **IS NOT NULL**; /* Exclude matches with no score */

/* Check if average scores exist for both teams */

IF (team1_avg **IS NULL** **OR** team2_avg **IS NULL**) **THEN** **RETURN** 0;**END IF**;

/* Determine winner based on higher average score */

IF team1_avg > team2_avg **THEN** **RETURN** team1_id;**ELSEIF** team1_avg < team2_avg **THEN** **RETURN** team1_id;**ELSE** **RETURN** 0;**END IF**;

/* Return winner team name */

RETURN winner_name;**END \$\$**

PredictWinnerWithPrecent .9

פרוצדורה המקבלת מפתח של שני קבוצות, ושני פרמטרים להוציא:

- שם קבוצה
- אחוזים לניצחון

הפרוצדורה מחזירה לשם הקבוצה את הקבוצה אם הסיכוי היותר גבוהה לנצח, ולאחוזי הניצחון מחזירה את אחוז הניצחון שתנצח ביחס לממוצאי נקודות המשחקים ששיחקו. הפרוצדורה מתמודדת גם אם שגיאות במידת הצורך:

```
CREATE PROCEDURE PredictWinnerWithPrecent(
    IN team1_id INT(10) UNSIGNED,
    IN team2_id INT(10) UNSIGNED,
    OUT winner_name VARCHAR(1027),
    OUT win_probability DECIMAL(5,2)
)
PWWWP:BEGIN
    DECLARE team1_avg DECIMAL(10,2);
    DECLARE team2_avg DECIMAL(10,2);

    SELECT AVG(score) INTO team1_avg
    FROM Matches_has_Teams mht
    JOIN Matches m ON mht.Matches_id = m.id
    WHERE mht.Teams_id = team1_id
    AND mht.score IS NOT NULL;

    SELECT AVG(score) INTO team2_avg
    FROM Matches_has_Teams mht
    JOIN Matches m ON mht.Matches_id = m.id
    WHERE mht.Teams_id = team2_id
    AND mht.score IS NOT NULL;

    /* Check if average scores exist for both teams */
    IF (team1_avg IS NULL OR team2_avg IS NULL) THEN
        SET winner_name = 'Insufficient data';
        SET win_probability = NULL;
        LEAVE PWWWP;
    END IF;

    /* Determine winner based on higher average score */
```



```

IF team1_avg > team2_avg THEN
    SET winner_name = (SELECT name FROM Teams WHERE id = team1_id);
    SET win_probability = (team1_avg / (team1_avg + team2_avg)) *
100;
ELSEIF team1_avg < team2_avg THEN
    SET winner_name = (SELECT name FROM Teams WHERE id = team2_id);
    SET win_probability = (team2_avg / (team1_avg + team2_avg)) *
100;
ELSE
    SET winner_name = 'Teams have the same average score';
    SET win_probability = 50.00;
END IF;
END$$

```

10. GetTeamWinRate

פונקציה המקבלת מפתח של קבוצה ושנה, ובודקת את אחוז הניצחונות שלה באותה שנה.

```

CREATE FUNCTION GetTeamWinRateByYear(
    team_id INT(10) UNSIGNED,
    `year` INT(4)
)
RETURNS DECIMAL(5,2)
BEGIN
    DECLARE total_matches INT UNSIGNED;
    DECLARE wins INT UNSIGNED;
    DECLARE win_rate DECIMAL(5,2);

    SELECT COUNT(*),
        SUM(CASE WHEN m1.did_win = 1 THEN 1 ELSE 0 END) INTO
total_matches, wins
    FROM Matches_has_Teams m1
    INNER JOIN Matches m ON m1.Matches_id = m.id
    WHERE m1.Teams_id = team_id
    AND YEAR(m.start_datetime) = year;

    SET win_rate = (wins / total_matches);

```

```

RETURN win_rate;
END$$

```

11. CreateChampionship

פרוצדורה המקבלת שנה ויוצרת טורניר של מנצחים בשנה, כלומר, טורניר שיש בוא רק קבוצות שזכו בפרס המגדיר אותם כמנצחים.

```

CREATE PROCEDURE CreateChampionship(
  IN `year` YEAR
)
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE team_id INT(10) UNSIGNED;
  DECLARE comp_id INT(10) UNSIGNED;

  DECLARE cur CURSOR FOR
    SELECT DISTINCT distinct t.id
  FROM Teams t
  INNER JOIN Awards_in_Competitions aic ON t.id = aic.teams_id
  INNER JOIN Competitions c ON aic.competitions_id = c.id
  WHERE YEAR(c.start_date) = `year`
  AND aic.Awards_id < 4
  AND t.id = aic.Teams_id;

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  INSERT INTO `robotics`.`Competitions`
    (`Cities_id`,
    `code`,
    `start_date`,
    `end_date`,
    `name`,
    `website`,
    `address`)
  VALUES
    (353747,

```

```
"2024champs",
NOW(),
NOW(),
"The absolute winners",
NULL,
"ORT hermlin");

SET comp_id = LAST_INSERT_ID();

OPEN cur;

loopi: LOOP
    FETCH cur INTO team_id;

    IF done THEN
        LEAVE loopi;
    END IF;

    INSERT INTO Teams_in_Competitions
    (`Competitions_id`,
    `Teams_id`,
    `ranking`)
    VALUES
    (comp_id,
    team_id,
    NULL);

END LOOP;

CLOSE cur;
END $$
```