

مرحله اول: ER

تقریباً دو روز طول کشید تا مدل اولیه رو بسازیم. بعد از چند جلسه هم توی یکی دو روز بعدی 6 بار ادیت کردیم تا به یه مدل تقریباً نهایی رسیدیم.

مرحله دوم: RM

با استفاده از ابزار dbschema تمامی جداول و ستون ها رو پیاده سازی کردیم. یک سری از ریلیشن ها مجبور شدیم کلاً به جدول جداگانه درست کنیم هر چند که رابطه یک به چند داشتند. در نهایت بعد از ساختن SQL و ... به این نتیجه رسیدیم که به جای جدول game میشد یه جدول دیگه گذاشت به نام event که یک event type داشته باشیم نشان دهنده اینکه آیا خطا بوده آیا گل بوده و ... و زمان این رویداد رو هم به عنوان یک صفت در اون جدول بزاریم. که این ایده دیر به ذهنمون رسید و عملی نشد.

مرحله سوم: ساخت Relational Algebra

در این مرحله با استفاده از وبسایت <https://dbis-uibk.github.io/relax/calc/local/uibk/local/0> هر کدوم از اعضای تیم یکی یا دو تا از گزارش گیری ها را به عهده گرفت و در نهایت جبر رابطه ای های نهایی برای هر گزارش گیری به وجود اومد.

مرحله چهارم: ساخت ساختار SQL

تیم ما از پایگاه داده MySQL استفاده کرد و تمامی جداول و صفات که داخل مدل رابطه ای ساخته شده بود را به یک schema در این نرم افزار انتقال دادیم. کدهای ساخت جداول و کلید های خارجی و اصلی و ... داخل فایل INITSQL2.sql وجود دارد.

مرحله پنجم: افزودن داده

برای داده ماک، در نرم افزار dbschema یک گزینه ای به نام data generator وجود داره که با توجه به generator، میتونه چندین سطر داده برای هر جدول تولید بکنه. یک سری از داده هامون رو مثلاً داده های player و staff و referee_observer رو با استفاده از این قابلیت نرم افزار ساختیم.

یه سری از داده های دیگه که شرایط خاصی داشت رو با استفاده از ChatGPT ساختیم. مثلاً داده های team.

یه سری دیگه از داده ها رو دستی وارد کردیم و تعداد خیلی محدودی دارن در حد 10 الی 20 تا سطر داده

چالش: موقع افزودن داده به یه سری از جداولی که Unique key داشتن و ما نیاز داشتیم که توی یک جدول بتونیم دو تا از همین داده های یکسان داشته باشیم به مشکل برمیخوریم که با استفاده از مراحل زیر این مشکل رفع میشد.

```
SHOW CREATE TABLE game
```

```
ALTER TABLE game DROP FOREIGN KEY fk_game_match_id
```

```
ALTER TABLE game DROP INDEX uniq_game_match_id
```

```
ALTER TABLE game ADD CONSTRAINT fk_game_match_id FOREIGN KEY (match_id) REFERENCES matches(match_id)
```

```
INSERT INTO game (team_id, match_id, player_id, league_id, g_id) VALUES(7, 1, 130, 2, 1)
```

مرحله ششم: ساختن کوئری ها

این مرحله، مرحله نهایی کار بود و ثمره تمام زحماتمون توی این مرحله دیده میشد. برای بیشتر کوئری ها مجبور بودیم از جداول زیادی استفاده بکنیم و جوین های زیادی بزنینم که احتمالا میشد با جدول های کمتری هم به همین نتیجه رسید و این به خاطر اشکالات مرحله اول و دوم بود.

گاهی مواقع مجبور میشدیم ساختار داده ای رو تغییر بدیم و یه سری صفت و ستون از یه سری جداول کم و زیاد بکنیم و این باعث شد نسخه های زیادی از مدل رابطه ای به وجود بیاد و هی مجبور بشیم به مرحله دو بریم و مدل رابطه ای رو تغییر بدیم.

در نهایت تمامی کوئری ها ساخته شدن و تلاش هامون ثمره داشت.

فایل های SQL تمامی کوئری ها داخل فولدر SQL هستند.

فرضایت اضافه شده:

چند تا جدول وجود داره که صرفا باید تمامی مقادیر ها به جز کلید های خارجی، محاسبه بشوند. مثلا در داخل جدول match ما تعداد گل های هر تیم رو داریم که این میشه با استفاده از یه کوئری که player_in_game و game رو به هم وصل میکنه، یه جمع بزنینم و در اون صفت قرار بدیم و بعد از هر بازی این UPDATE بشه. کد های این قسمت موقع ارائه، به TA نشان داده خواهد شد.

برای جدول لیگ هم ما یه سری صفت جدید مثل Win, Loss, Draw, Point و ... اضافه کردیم که این ها هم باید محاسبه بشوند و بعد از هر بازی این جدول UPDATE شود.

گزارش ها و کوئری ها:

این قسمت در داخل فایل RA.pdf در پوشه Relational Algebra کامل بررسی شده.