دانشگاه اصفهان

دانشکده‌ی مهندسی کامپیوتر

آزمایشگاه پایگاه داده

فاز اول پروژه

استاد: دکتر مهدوی‌نژاد

سجاد یزدان پرست - ۹۶۳۶۱۳۱۰۵

هادی حیدری راد - ۹۶۳۶۱۳۰۲۸

پاییز ۹۹

سیستم آزمایشگاه سیار

- Project Description

پروژه‌ی انتخابی ما یک ایده‌ی جدید است تحت عنوان آزمایشگاه سیار، اما سیار به معنای سیار بودن خود آزمایشگاه نیست بلکه به معنای سیار بودن فرآیند نمونه‌گیری است. روال به این صورت است که سیستمی ارائه می‌دهیم که بوسیله‌ی آن مشتریان بتوانند آزمایش‌های دلخواه خود (فعلا فقط آزمایش‌های آزاد و بدون نیاز به نسخه) را در اپلیکیشن انتخاب کنند، و سیستم به آن‌ها لیستی از آزمایشگاه‌هایی که تمام آن آزمایش‌ها را ساپورت می‌کنند به کاربر نشان دهد. سپس کاربر سفارش خود را ثبت می‌کند و آدرسی نیز مشخص می‌کند. نمونه‌گیرهایی که برای آزمایشگاه‌ها کار می‌کنند در تاریخ و آدرس مشخص شده نزد کاربر حضور می‌یابند و از وی نمونه‌گیری‌های لازم را انجام می‌دهند و تحویل آزمایشگاه می‌دهند. سپس آزمایشگاه پس از آماده شدن نتیجه آن را به سیستم ما ارسال می‌کند و ما نتیجه را بصورت آنلاین به مشتری نمایش خواهیم داد.

---

1. Purpose of our Database

هدف ما از طراحی این پایگاه‌داده، نگهداری اطلاعات مربوط به پشتیبانی آزمایش‌ها توسط آزمایشگاه‌ها و نمونه‌گیرهای مربوط به آن، و اطلاعات سفارش‌های ثبت شده‌ی کاربران است.

2. Finding and organizing the information required

به طور کلی اطلاعاتی که برای این پروژه نیاز است انواع آزمایش‌های آزاد، آزمایشگاه‌ها و پشتیبانیشان از آزمایش‌ها، نمونه‌گیرها، مشتریان، و آدرسشان است. این اطلاعات به طور کلی بیان شدند و به تبع جزئیات بیشتری از هر کدامشان (attribute ها) نیز نیاز به ذخیره شدن دارند.

3. Dividing the information into tables

پس می‌توان موجودیت‌های پایگاه داده را به صورت زیر تعریف کرد:

User: کاربر برنامه را مشخص می‌کند.

BloodExpert: نمونه‌گیر را مشخص می‌کند. از قبل می‌دانیم این نمونه‌گیرها هر کدام برای یک آزمایشگاه مشخص کار می‌کنند.

Lab: آزمایشگاه را مشخص می‌کند.

TestType: انواع آزمایش‌های آزاد که در سیستم پشتیبانی می‌شوند هستند. به طور مثال: آزمایش قند خون، چک‌آپ، HIV و ...

Test: آزمایش‌های انتخابی کاربر هستند. نتیجه‌ی آزمایش نیز پس از انجام، زیر مجموعه‌ی همین موجودیت قرار خواهد گرفت.

Order: مشخص کننده‌ی یک سفارش کاربر است. معدل سبد خرید در یک سیستم فروش آنلاین. چون می‌دانیم کاربر در یک نوبت ثبت سفارش می‌تواند سفارش چند آزمایش دهد.

TimeService: موجودیتی است برای مدل کردن زمان‌های ارائه‌ی سرویس توسط نمونه‌گیرها (زمان‌های دقیق حضور نمونه‌گیر در مکان مشتری‌ها)

4. Turning information items into columns

حال به ازای تمام موجودیت‌های استخراج شده، ستون‌های مورد نیاز آن موجودیت (attribute) را در می‌آوریم:

User: نیاز داریم به ازای هر کاربر (معادل هر شخص) نام و نام خانوادگی، کد ملی، جنسیت، شماره موبایل و آدرس ایمیل وی را در سیستم ذخیره نماییم.

BloodExpert: نمونه‌گیر را می‌توان دقیقا همان کاربر فرض کرد منتهی با دو attribute اضافه‌ی طول و عرض جغرافیایی! چون نیاز داریم برای پیگیری سفارش‌ها بدانیم که هر نمونه‌گیر در لحظه کجا است. (با توجه به قابل پیش بینی نبودن مسیر، امکان محاسبه‌ای بودن این attribute ها وجود ندارد و مجاب هستیم در لحظات جابجایی مدام این attribute ها را بروز کنیم پس calculated نیستند)

Lab: به ازای هر آزمایشگاه نیاز است یک اسم (name)، و یک end point و api key برای ارتباط با api آزمایشگاه برای دریافت قیمت‌ها را ذخیره کنیم.

TestType: دانستن فقط نام یک آزمایش برای این موجودیت کفایت می‌کند. چون درحال ذخیره‌ی تایپ های آزمایش‌ها هستیم. به طور مثال آزمایش قند.

Test: نیاز داریم علاده بر دانستن نوع آزمایش، نتیجه‌ی آزمایش را نیز ذخیره کنیم پس یک ستون result اضافه می‌شود.

Order: هر سفارش نیاز دارد برای شناسایی یکتا، یک ID داشته باشد. جدا از آن برای هر سفارش باید وضعیت، قیمت و طول و عرض جغرافیایی (مشخص کننده‌ی لوکیشن یا آدرس) را نیز مشخص نماییم.

TimeService: نیاز داریم بدانیم یک زمان ارائه‌ی سرویس مختص به کدام نمونه‌گیر است (expert_id)، تاریخ آن کی است(date)، و بازه‌ی زمانی آن کی است که این بازه‌ی زمانی را با زمان شروع (stime) و زمان پایان (etime) مدل می‌کنیم.همچنین نیاز داریم بدانیم یک TimeService آیا موجود است یا پر شده است (is_available)

5. Specifying primary keys

User: با توجه به محدودیت یکتا بودن کدملی، می‌توان آن را به عنوان primary key در نظر گرفت.

BloodExpert: نمونه‌گیر نیز با توجه به آن‌که یک کاربر برنامه است (منتهی با دسترسی‌های متفاوت) با همان کدملی می‌تواند بصورت یکتا شناسایی شود پس کدملی primary key است.

TestType: این موجودیت فقط یک attribute دارد که نام آزمایش است. و همان نیز با توجه به یکتا بودنش در هر آزمایش، می‌تواند به عنوان primary key انتخاب شود.

Test: با توجه به آنکه آزمایش‌های ثبت شده‌ی مشتریان نیاز به ثبت نتیجه نیز دارد، پس هر آزمایش هر کاربر بصورت یکتا توسط یک ID شناسایی می‌شود (primary key)

Order: به هر سفارش یک ID برای شناسایی یکتایشان اختصاص می‌دهیم. چون بوسیله‌ي مابقی attribute ها و یا ترکیبشان امکان شناسایی نیست.

TimeService: ترکیب شناسه‌ی نمونه‌گیر، تاریخ، و زمان شروع و پایان می‌تواند به عنوان primary key برای شناسایی یکتای زمان‌های ارائه‌ی سرویس انتخاب شود.

6. Setting up the table relationships

رابطه‌ی BloodExpert با user از نوع ISA است و BloodExpert تمام ویژگی‌های یک User را ارث می‌برد چون خود نیز کاربری از سیستم است.

رابطه‌ی User و Test از نوع یک به چند (صفر یا بیشتر) است. به این دلیل که هر کاربر می‌تواند تعداد صفر و یا بیشتر آزمایش ثبت کرده باشد. و هر آزمایش ساخته شده نیز متعلق به دقیقا یک کاربر است.

رابطه‌ی Test و Order از نوع یک به چند (یک یا بیشتر) است. به این دلیل که هر سفارش می‌تواند یک یا بیشتر آزمایش درون خود داشته باشد و هر آزمایش نیز دقیقا متعلق به یک سفارش است.

رابطه‌ی Test و Lab از نوع یک به چند (صفر و یا بیشتر) است به این دلیل که هر آزمایش ثبت شده دقیقا توسط یک آزمایشگاه انجام می‌شود و یک آزمایشگاه می‌تواند تعداد صفر و یا بیشتر آزمایش را انجام دهد.

رابطه‌ی Lab و TestType از نوع چند به چند است به این دلیل که هر آزمایشگاه می‌تواند چند نوعِ آزمایش را انجام دهد و هر نوعِ آزمایش می‌تواند توسط چند آزمایشگاه انجام شود.

رابطه‌ي BloodExpert و Lab از نوع یک به چند (صفر و یا بیشتر) است. به این دلیل که هر نمونه‌گیر برای دقیقا یک آزمایشگاه کار می‌کند و هر آزمایشگاه می‌تواند تعداد صفر و یا بیشتر نمونه‌گیر داشته باشد.

رابطه‌ی Test و TestType از نوع ISA است به این دلیل که هر آزمایش ثبت شده توسط کاربر یک نوعِ آزمایش نیز هست منتهی به همراه result.

رابطه‌ی BloodExpert و TimeService از نوع یک به چند (صفر و یا بیشتر) است به این دلیل که هر زمان ارائه‌ی سرویس متعلق به دقیقا یک نمونه‌گیر است و هر نمونه‌گیر تعداد صفر و یا بیشتر زمان ارائه‌ی سرویس دارد. برای این ارتباط از شناسه‌ی نمونه‌گیر به عنوان foreign key استفاده شده‌است.

7. Refining our design

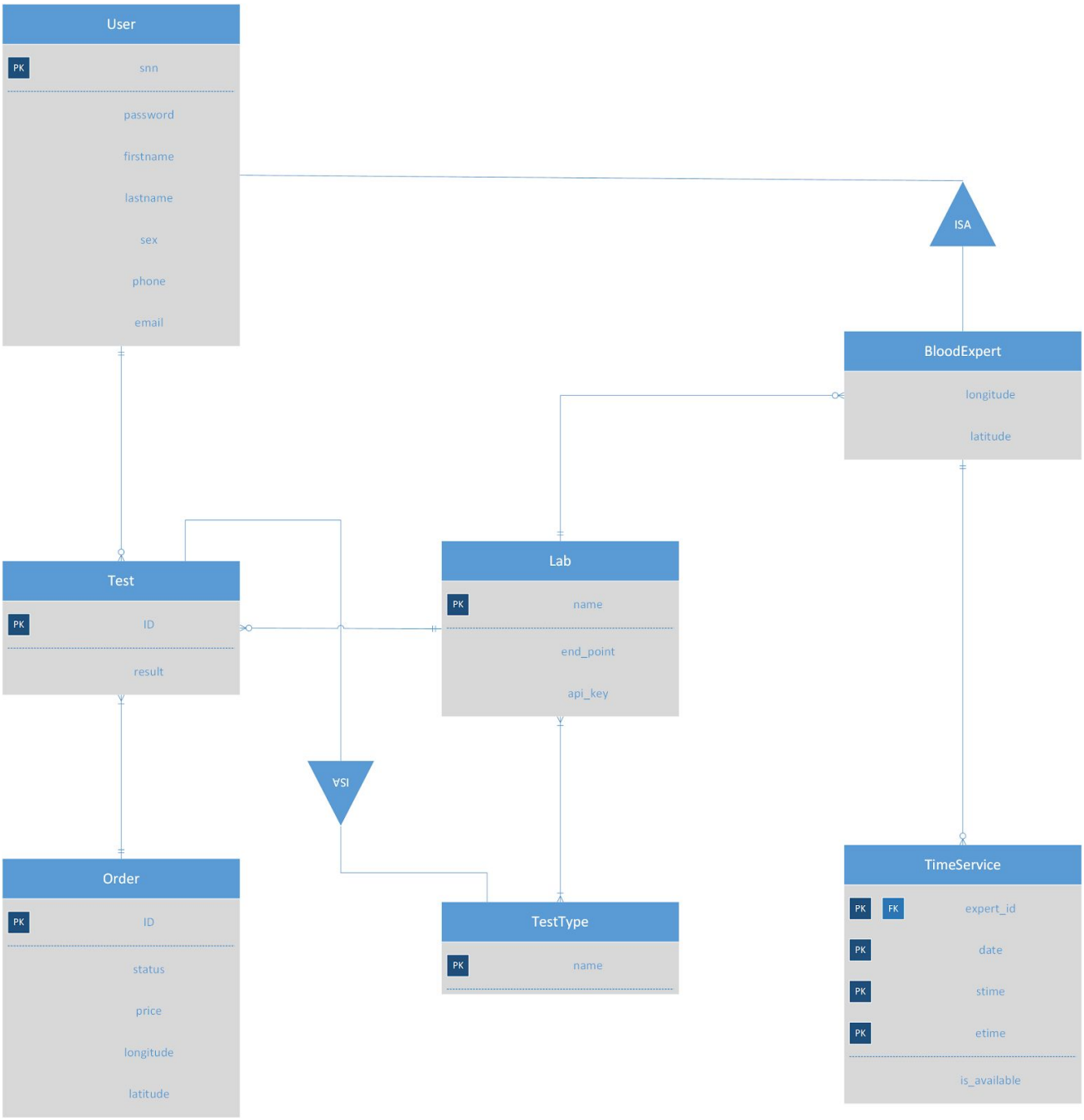
به طور مثال در طراحی اولی که داشتیم هر دوی Test و TestType را در یک موجودیت لحاظ کرده بودیم که طراحی خوبی نبود. و یا به‌جای طول و عرض جغرافیایی از location استفاده کرده بودیم که قابل تجزیه بود. در موجودیت User علاوه بر ستون کدملی ستون username نیز تعریف کرده بودیم که از آن به عنوان کلید اصلی استفاده میشد. چون کدملی برای هر فرد یکتاست، ستون username را حذف کردیم. بنابراین کاربران برنامه با استفاده از کدملی و password شان احراز هویت خواهندشد.

8. Applying the normalization rules

- نرمال فرم۱: همه المان‌ها از تمامی جدول ها غیر قابل تجزیه هستند (atomic). در هر جدول اسامی تمام ستون ها یکتاست.
- نرمال فرم۲: این فرم تنها برای جداولی قابل پیاده‌سازی است که composite key داشته باشند. تنها جدولی که composite key دارد جدول TimeService است که ستون is_available به تمامی دیگر ستون‌ها وابسته است.
- نرمال فرم۳: در همه جداول ستون (های) غیر کلید فقط و فقط به کلید(های) اصلی وابسته هستند. قبلا در موجودیت Order ستون آدرس از جنس string و طول و عرض جغرافیایی باهم حضور داشتند. واضح است که ستون آدرس کاملا به طول و عرض جغرافیایی وابسته است؛(طول و عرض جغرافیایی جز کلید اصلی موجودیت نیستند) همین امر جدول را از فرم نرمال ۳ خارج می‌کرد. بنابراین با حذف فیلد آدرس جدول فرم نرمال سوم را پیداکرد. لازم به ذکر است چون فیلد آدرس از جنس string بود مشکلی از جهت فرم نرمال ۱ به وجود نمی آمد. در ابتدا ما تصمیم داشتیم فیلد آدرس را صرفا جهت راهنمایی دقیق‌تر BloodExpert ذخیره‌کنیم؛ یعنی کاربران به دلخواه خود می‌توانستند هر چیزی را به ما به عنوان آدرس بدهند. مثلا کاربر ۱ صرفا اسم کوچه را قید می‌کرد اما کاربر ۲اسم خیابان اصلی، اسم خیابان فرعی و سپس اسم کوچه را به عنوان آدرس ثبت می‌کرد. با ذخیره‌سازی طول و عرض جغرافیایی در هر لحظه می‌توانیم با تعامل با دیگر سیستم‌ها (Google Map) آدرس دقیق را محاسبه کنیم.

نمودار موجودیت-رابطه زیر با استفاده از نرم‌افزار visio با استفاده از Crow's Foot database notation رسم کردیم.

هر خط در این نمودار نشان‌دهنده یک رابطه است(به جز خط‌های متصل به ISA. ISA خود رابطه specialization درست کرده‌است)


User
PK
snn
password
firstname
lastname
sex
phone
email
ISA
BloodExpert
longitude
latitude
Test
PK
ID
result
Lab
PK
name
end_point
api_key
∀SI
Order
PK
ID
status
price
longitude
latitude
TestType
PK
name
TimeService
PK
FK
expert_id
PK
date
PK
stime
PK
etime
is_available