

## سوال (۱) بیمارستان:

### • توضیحات مربوط به هر Attribute:

هر بیمارستان چندتا Attribute دارد: H\_Name (نام بیمارستان)، Dr\_Id (شناسه‌ی دکترهایی که آنجا کار میکنند)، H\_ID (شناسه‌ی آن بیمارستان) (اگر بخوایم گسترش بدیم بعداً، شاید بدرد بخوره به عنوان یک Foreign key ازش استفاده کنیم)، Room\_ID (شناسه‌ی هر اتاقی که داخلش وجود داره)، H\_Address (آدرس) که یک Composite Attribute هستش که خودش شامل استان و شهر و خیابانی هستش که هر بیمارستان داخلش قرار گرفته.

هر دکتر چند تا Attribute دارد: Dr\_Id (شناسه‌ی دکتر و از اون به عنوان ForeignKey در Table های دیگه قراره استفاده بشه و در اینجا یونیکه)، هر دکتر ی lastname و firstname و salary (حقوق) و expertise (تخصص) ای دارد.

هر بیمار چند تا Attribute دارد: p\_id که از اون قراره به عنوان یک foreign key در table های دیگه استفاده بشه و یونیکه، هر بیمار birthdate و lastname و firstname دارد. همچنین nationalcode دارد که unique هستش و اون رو از دیگر بیماران متمایز میکنه برای همین unique کردم. از طرفی هر بیمار ممکنه چندتا شماره تلفن یا چند تا بیماری به صورت همزمان داشته باشه، در نتیجه attribute های phone number و sickness رو براش از نوع multivalued گذاشتم.

هر اتاق هر بیمارستان چندتا attribute داره: breathing equipment که آپشنال گذاشتم و میتونه نداشته باشه (مثلاً اتاق های تصویربرداری تجهیزات تنفسی ندارند)، room\_id که شناسه‌ی اون اتاق هستش و میتونه به عنوان یک foreign key در table های دیگه استفاده بشه و در اینجا یونیک کردم، room\_name که نام اون اتاق هستش (البته بنظرم دلیلی نداشت که با وجود room\_id، room name هم داشته باشه اما بخاطر خواسته‌ی سوال گذاشتمش)، هر اتاقی قاندا یک پرستار رو داره در نتیجه براش nurse\_id گذاشتم که این nurse\_id یک foreign key در اینجا محسوب میشه. همچنین اینکه یک bed\_id داره که شناسه‌ی هر تخت خوابی که داخلش هست رو نشون میده و در اینجا یک foreignkey محسوب میشه.

هر تخت بیمار دو تا Attribute دارد: Bed\_Id که شناسه‌ی تخت هستش و یونیک هستش و برای هر تخت متفاوت و یک name که باز بخاطر خواسته‌ی سوال قرارش دادم و بنظرم دلیلی نداشت وجود داشته باشه، و یک SupplierID که نشوندنده‌ی اینه که این تخت از چه شرکتی تهیه شده و یک foreign key هستش

در ادامه من خودم اوادم دو تا موجودیت دیگه اضافه کردم به دیاگرام برای اینکه بتونم روابط رو بهتر بهم مرتبط کنم و یک جدولم خیلی شلوغ نشه: Hospitalization detail و visits detail

Hospitalization Detail چند تا attribute دارد: H\_ID برای اینکه بفهمیم بستری شدن بیمار در کدام بیمارستان صورت گرفته، Bed\_Id چه تختی به بیمار تخصیص یافته، room\_id برای اینکه بفهمیم چه اتاقی به بیمار تحویل داده شده، p\_id برای اینکه بفهمیم کدام مریض بستری شده، date برای اینکه بفهمیم در چه تاریخی بستری شدن صورت گرفته، cost برای اینکه بفهمیم هر بیمار چه هزینه ای رو پرداخت باید کنه

Visit details هم چندتا attribute براش گذاشتم که مربوط میشه به ویزیت بیمار توسط پزشک:

P\_id شناسه ی بیماری که ویزیت شده، Dr\_ID شناسه ی پزشکی که ویزیت کرده، disease diagnosis برای اینکه بفهمیم تشخیص دکتر چی بوده، و visit date برای اینکه بفهمیم در چه تاریخی دکتر مریض رو ویزیت کرده

#### • روابط:

بیمارستان و پزشک: هر بیمارستان باید به صورت اجباری (Mandatory) دکتر داشته باشه و میتونه بی نهایت دکتر داشته باشه – از طرفی هر دکتر هم باید به صورت اجباری توی یه بیمارستان باشه و میتونه همزمان توی چنتا بیمارستان باشه. (Many To Many)

بیمارستان و اتاق: هر بیمارستان باید به صورت اجباری اتاق داشته باشه و میتونه بی نهایت تا اتاق داشته باشه – از طرفی هر اتاق هم باید حتما توی یه بیمارستان باشه، و باید فقط توی یک بیمارستان باشه، چون شناسه ی هر اتاق منحصر به فرد و نمیتونه توی چنتا بیمارستان باشه (Many To One)

اتاق و تخت: هر تخت باید حتما داخل اتاق باشه و فقط میتونه توی یک اتاق باشه – هر اتاق هم باید به صورت اجباری تخت داشته باشه و میتونه چنتا تخت هم داشته باشه (One To Many)

اتاق و جزئیات بستری: هر اتاقی که قراره به بیماری داده بشه، حتما یه جزئیات بستری شدن داره و فقط یک جزئیات داره چون فقط یک بیمار اون تخت رو تحویل میگیره – همچنین اینکه توی هر بستری شدن، حتما باید یک تخت به بیمار اختصاص پیدا کنه و حتما هم باید یک تخت باشه چون به یک بیمار چندتا تخت نمیدن (One To One)

بیمار و بستری شدن: هر بیماری برای بستری شدن، حتما یک جزئیاتی داره تا بهش تخت و بعد اتاق رو بدن – از طرفی هر بستری شدن، باید حتما یک بیمار درش وجود داشته باشه، وگرنه بستری شدن صورت نمیگیره (One To One)

بیمار و جزئیات ویزیت توسط پزشک: هر بیماری باید حتما توسط پزشک معاینه بشه و همچنین اینکه میتونه توسط چندتا دکتر معاینه بشه – از طرفی هر ویزیت شدنی باید حتما بیمار داشته باشه و حتما یک دونه وجود داشته باشه (Many To One)

پزشک و جزئیات ویزیت: هر پزشکی میتونه بیماری رو ویزیت کنه، میتونه هم نکنه (آپشناله) و اینکه میتونه هم چندتا بیمار رو هم ویزیت کنه – از طرفی هر ویزیتی که انجام میشه، باید حتما توسط یک دکتر صورت گرفته باشه و حتما یک دکتر اون رو انجام داده باشه. (Many To One)

## سوال ۲) فروشگاه:

### • توضیحات مربوط به هر Attribute:

هر فروشگاه چندتا Attribute دارد: Name که نام فروشگاه است، Store\_Id که شناسه ی فروشگاه است و یونیک هست و از آن به عنوان foreign key در جداول دیگر استفاده میشود، address که یک attribute کامپوزیت است و خود شامل استان و شهر و خیابان اصلی است، website\_url که منظور وبسایت فروشگاه است و همانطور که میدانیم همه ی وب سایت ها نیز یونیک هستند. و phonenumner که یک attribute multivalued است و هر فروشگاه میتواند همزمان چندتا شماره تلفن داشته باشد.

هر مشتری چندتا attribute دارد: firstname و lastname که داخل سایت ثبت میکند، address که یک attribute کامپوزیت است و خود شامل استان و شهر و خیابان اصلی است، postalcode که منظور کدپستی مشتری است، phonenumner که یک multivalued attribute است و یک فرد میتونه همزمان چندتا شماره موبایل داشته باشه (همراه اول، ایرانسل، رایتل)، EmailAddress برای فرستادن صورتحساب به ایمیل مشتری، و customer\_id که شناسه ی فرد خریدار است و میتواند به عنوان foreign key در دیگر جداول به کار رود.

هر محصول چند Attribute دارد: product\_ID که شناسه ی محصول است و در این جا یک primary key میباشد و در دیگر جداول میتواند به عنوان یک foreign key بکار رود. Product\_name که منظور نام محصول است. Unit که منظور تعداد هر محصول موجود با یکای اندازه گیری هست (مثلا ۲۴ تا بطری ۱۲ میلی لیتری). price که منظور قیمت محصول است.

هر سفارش چند Attribute دارد: تعداد محصولات یا همان quantity، customer\_id که در اینجا یک foreign key است و منظور شناسه ی فردی است که این سفارش رو انجام داده. Product\_id که منظور شناسه ی محصول است و یک foreign key محسوب میشود. Order\_id شناسه ای هست که به هر سفارش

داده میشود و یک primary key است و از آن در دیگر جداول استفاده میشود. Order\_date تاریخی است که مشتری این سفارش را ثبت کرده است. Total\_cost هزینه ی کلی برای آن سفارش محسوب میشود و یونیک است.

هر تراکنش دارای چند attribute است: store\_id شناسه ی فروشگاه ی است که سفارش در آن ثبت شده، customer\_Id که شناسه ی مشتری ای است که این سفارش را ثبت کرده، date of pay که تاریخ روزی است که سفارش ثبت شده، و order\_ID شناسه ی سفارش است که در آن میتوان جزئیات هر سفارش را مشاهده کرد.

### • روابط:

فروشگاه و مشتری: هر فروشگاه میتواند مشتری داشته باشد و میتواند نداشته باشد (آپشنال است) و همچنین بی نهایت مشتری دارد. - از طرفی هر مشتری میتواند از فروشگاه ی خرید کند میتواند هم نکند (آپشنال) و اینکه اگه بخواد خریدی هم بکند میتواند از چندتا فروشگاه خرید کند. (Many To Many)

فروشگاه و محصول: هر فروشگاه حتما محصولات ی داره (Mandatory) و چندتا محصول میتواند داشته باشد - از طرفی هر محصول حتما باید توی فروشگاه باشد (Mandatory) و همچنین اینکه یک محصول میتواند توی چنتا فروشگاه قرار داشته باشد. (Many To Many)

مشتری و محصول: هر مشتری حتما باید محصول بخره و اینکه میتواند چنتا محصول بخره - از طرفی هر محصول حتما باید توسط مشتری انتخاب بشه، و توسط یک مشتری انتخاب شه که براش صورتحساب درست بشه. (Many To One)

محصول و جزئیات سفارش: محصولاتی که خریده میشن حتما جزئیات سفارشی دارن و فقط یکدونه جزئیات سفارش دارن، از طرفی هر جزئیات سفارشی حتما داخلش محصولاتی وجود داره و میتواند چندتا هم محصول داخلش باشه. (One To Many)

جزئیات سفارش و صورتحساب پرداخت: هر سفارشی که انجام میشه حتما یک صورتحسابی برای پرداخت داره و فقط یکدونه صورتحساب پرداخت داره - از طرفی هر صورتحساب پرداختی که صورت گرفته هم توسط یک جزئیات سفارشی حتما صورت گرفته (One To One)

مشتری و صورتحساب پرداخت: هر مشتری ای که محصولی رو میخواد بخره، حتما باید یه صورتحسابی رو پرداخت کنه (اما خب میتواند هم پرداخت نکند و در نتیجه سفارشی هم اصلا انجام نشه)، در نتیجه من آپشنال

گذاشتمش و اینکه فقط یک صورت حساب از اون فروشگاه رو باید پرداخت کنه. - از طرفی هر صورت حساب پرداختی هم حتما مشتری ای داشته و اینکه میتونه شامل چندتا مشتری باشه (One To Many)