

-1

شناسایی موجودیت ها: یافتن تمامی نقش ها، افراد و ... که لازم است اطلاعات آن ها ذخیره شود

شناسایی روابط: یافتن روابط بین جفت موجودیت ها با استفاده از ماتریکس روابط

رسم یک ERD اولیه: قرار دادن موجودیت ها در مستطیل ها و اتصال آن ها بر اساس روابط

مشخص کردن کاردینالیتی: تعیین تعداد دفعات تکرار یک موجودیت در رابطه با یک موجودیت دیگر

مشخص کردن کلید اصلی: شناسایی دیتایی که به صورت یکتا، تنها و فقط تنها یک نمونه از موجودیت را تمیز میدهد

رسم یک ERD بر اساس کلید ها: پاک کردن روابط many-to-many و اضافه کردن کلید های اصلی و خارجی به هر موجودیت

شناسایی خواص: یافتن اطلاعاتی و جزئیاتی که برای سیستم در حال طراحی ضروری هستند

نگاشت خواص: تخصیص هر یک از خواص به دقیقاً یک موجودیت که آن را توصیف میکند

رسم یک ERD با تمامی خواص: تغییر و بروزرسانی ERD با استفاده از خواص جدید

چک کردن نتیجه نهایی: آیا ERD به دست آمده دقیقاً نیاز های سیستم را برآورده میکند؟

-2

خواص مرکب: برای هر بخش از آن خاصیت، یک خاصیت جدا در نظر گرفته میشود، به عبارت دیگر، یک خاصیت مرکب باز میشود به چندین خاصیت دیگر، به طور مثال

Name

First\_name

Last\_name

به صورت name\_first\_name و name\_last\_name در می آید

خواص چند ارزشی: برای این خواص، یک جدول جدا در نظر گرفته میشود، فرض کنیم خاصیت M از موجودیت E چند ارزشی باشد، یک جدول که کلید اصلی آن، کلید خارجی به E است و یک ستون برای مقادیر آن خاصیت چند ارزشی دارد

-3

یک موجودیت که خواص لازم برای تشکیل یک کلید اصلی ندارد را موجودیت ضعیف نامیده میشود و موجودیتی که کلید اصلی دارد را موجودیت قوی مینامند، به عبارت دیگر موجودیت ضعیف، وابسته به یک موجودیت دیگر است

با افزودن attribute به عنوان کلید اصلی اطلاعات دچار redundancy میشود

-4

a. موجودیت های قوی: bank, account, customer, loan      موجودیت ضعیف: bank branch

b. بانک و شعبه: هر بانک چندین شعبه دارد ولی هر شعبه مربوط به فقط یک بانک است، هر دو سمت رابطه total participation هستند (یعنی کاملاً در relationship set حضور دارند)

شعبه و حساب: هر شعبه چندین حساب دارد ولی هر حساب فقط در یک شعبه است، همینطور همه حساب ها حتماً یک شعبه دارد (total participation) ولی لزوماً هر شعبه ای دارای حساب نیست (partial)

شعبه و وام: هر شعبه چندین وام ارائه میدهد ولی هر وام فقط در یک شعبه ارائه میشود، همینطور همه وام ها حتماً یک شعبه مربوطه دارند، ولی لزوماً هر شعبه ای وام ارائه نمیدهد

حساب و مشتری: هر مشتری صفر تا چندین حساب دارد و هر حساب میتواند مربوط به یک یا چند نفر باشد  
مشتری و وام: هر مشتری صفر تا چندین وام میتواند گرفته باشد و هر وام توسط یک تا چند مشتری دریافت شده است

c. مشخص کردن مشخصات مشتریان و وام های هر یک از آن ها  
مشخص کردن اینکه هر وام مربوط به کدام شعبه و کدام بانک است  
مشخص کردن اطلاعات هر مشتری و حساب هایی که به نام آنان است  
d.

1- انتخاب ستون های نام، آدرس، تلفن، مقدار و نوع وام از ضرب دکارتی جدول مشتری و وام با استفاده از آی دی مشتری

2- انتخاب ستون های مقدار و نوع وام، نام بانک و آدرس شعبه از ضرب دکارتی جدول وام و شعبه با استفاده از آی دی شعبه و ضرب آن در بانک با استفاده از آی دی بانک

3- انتخاب ستون های نام و آدرس و شماره تلفن مشتری، اعتبار و نوع حساب از ضرب دکارتی جدول حساب ها و مشتری با استفاده از آی دی مشتری

e. از آنجایی که در صورت مسئله گفته شده فقط یک ضامن، میتوان اطلاعات آن ها را به صورت attributes مستقیماً در جدول وام ها ذخیره کرد

-5

a.

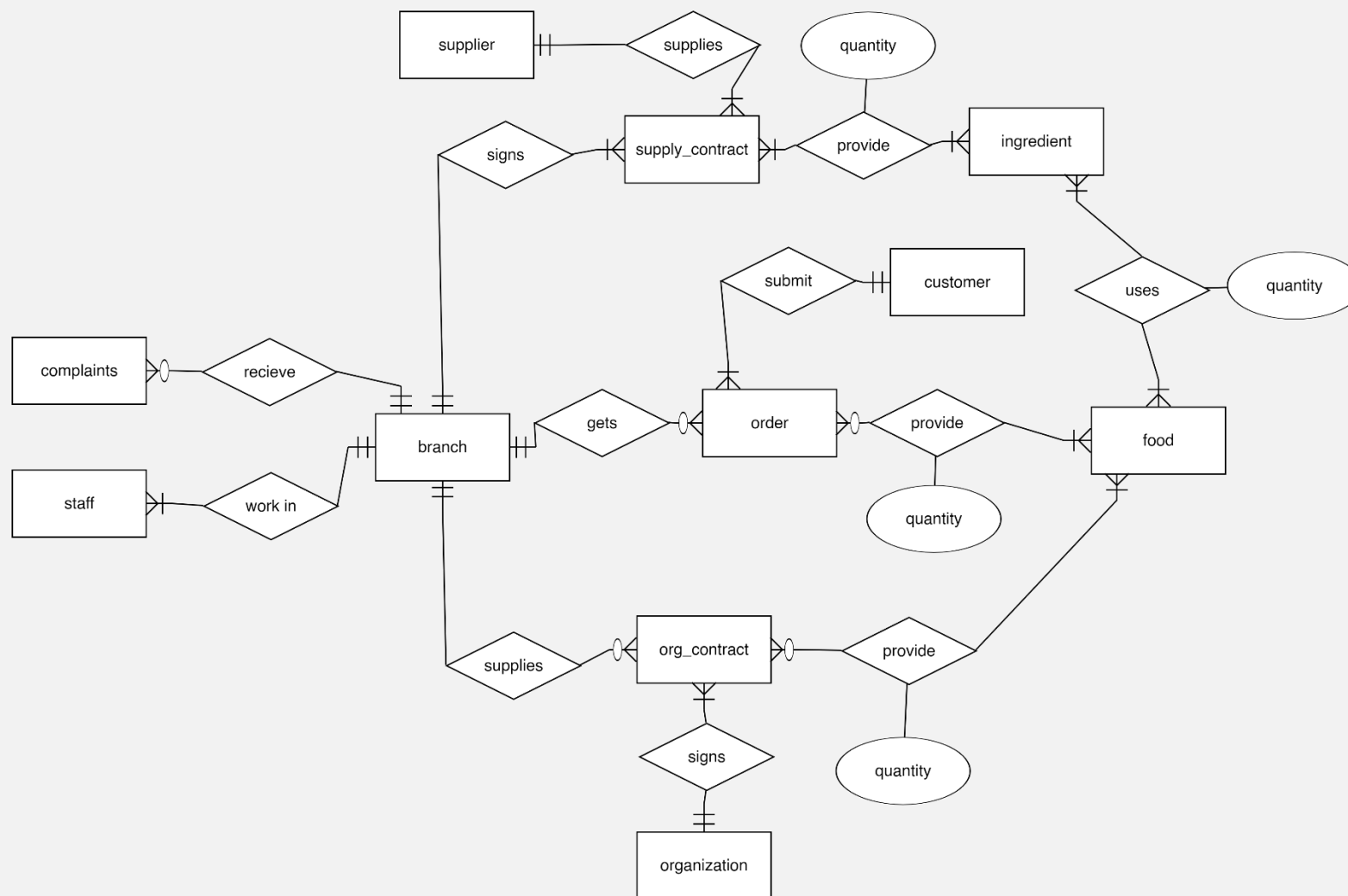
فرض میکنیم که بین توزیع کننده و شعبه قرارداد هایی برقرار است

-----	branch	supplier	Supply_contract	staff	food	ingredient	order	customer	organization	Org_contract	Complaint
branch			signs	uses			gets			supplies	recieves
supplier			supplies								
Supply_contract	Get signed by	Get supplied by				provide					
staff	Works in										
food						uses	Provided in			Provided in	
ingredient			Provided in		Used in						
order	Sent to				provide			Get submitted by			
customer							submit				
organization										signs	
Org_contract	Get supplied by				provide				Get signed by		
Complaint	Sent to										

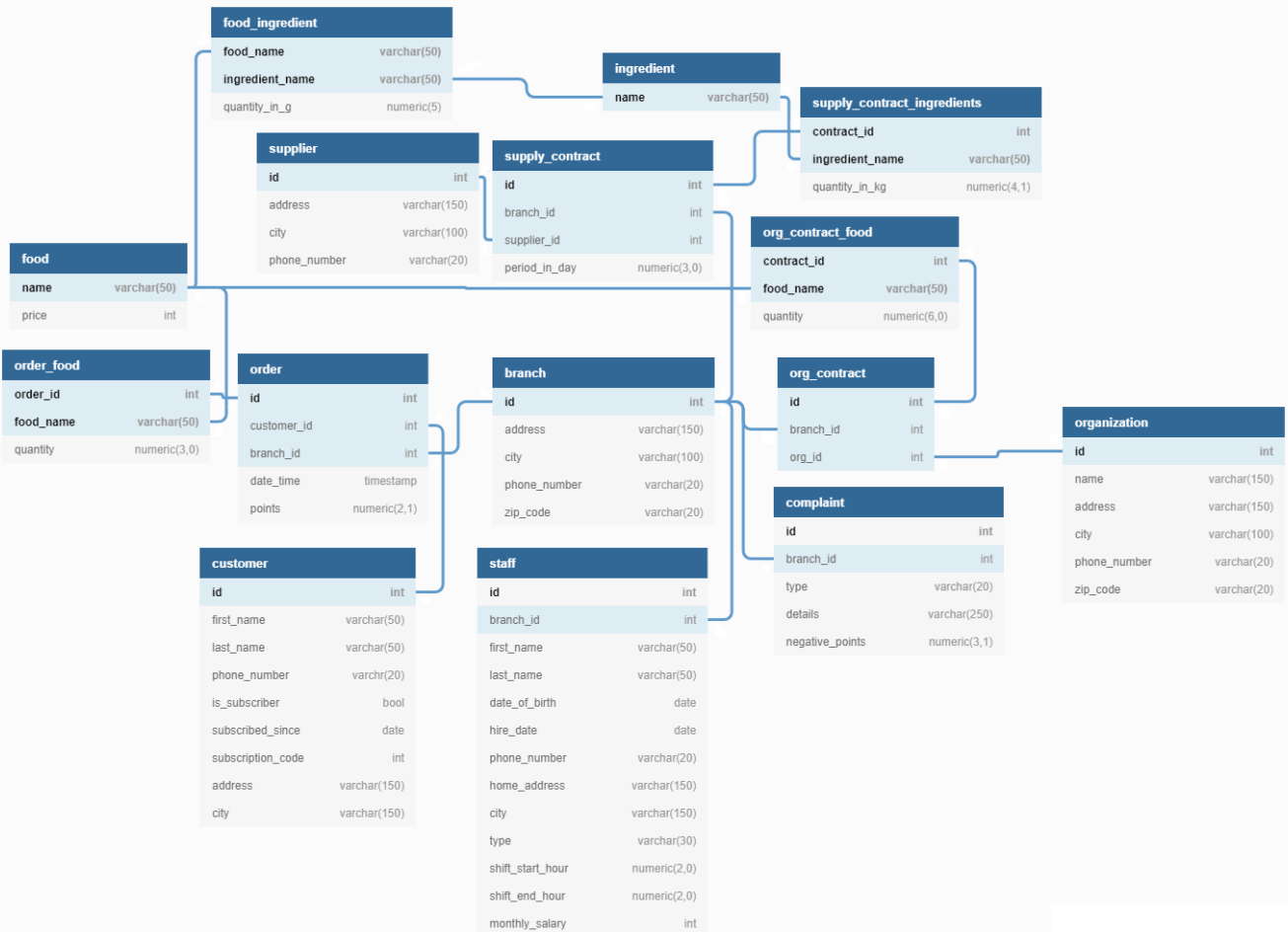
b.

مفروضات: هر شعبه حداقل یک قرار داد با یک توزیع دهنده دارد. هر فرد استخدامی فقط در یک شعبه کار میکند. همه غذاها در همه شعبه ها با قیمت معین ارائه میشود و نیاز نیست که رابطه خاصی میان این دو باشد. در ابتدای کار یک شعبه هیچ سفارش مشتری یا قرار داد با سازمان ممکن است نداشته باشد. فرض میکنیم شکایات مشتریان به صورت ناشناس ثبت میشود. توزیع کننده یا سازمان بدون قرارداد در سیستم ثبت نمیشود. هر ماده اولیه حداقل از طریق یک قرارداد

توزیع تعیین میشود و حتما و حداقل در یک نوع غذا رستوران استفاده میشود. ممکن است غذایی که تازه در سامانه ثبت شده در هیچ سفارشی نباشد. مشتری باید حداقل یک سفارش ثبت کرده باشد تا در سامانه ذخیره شود.

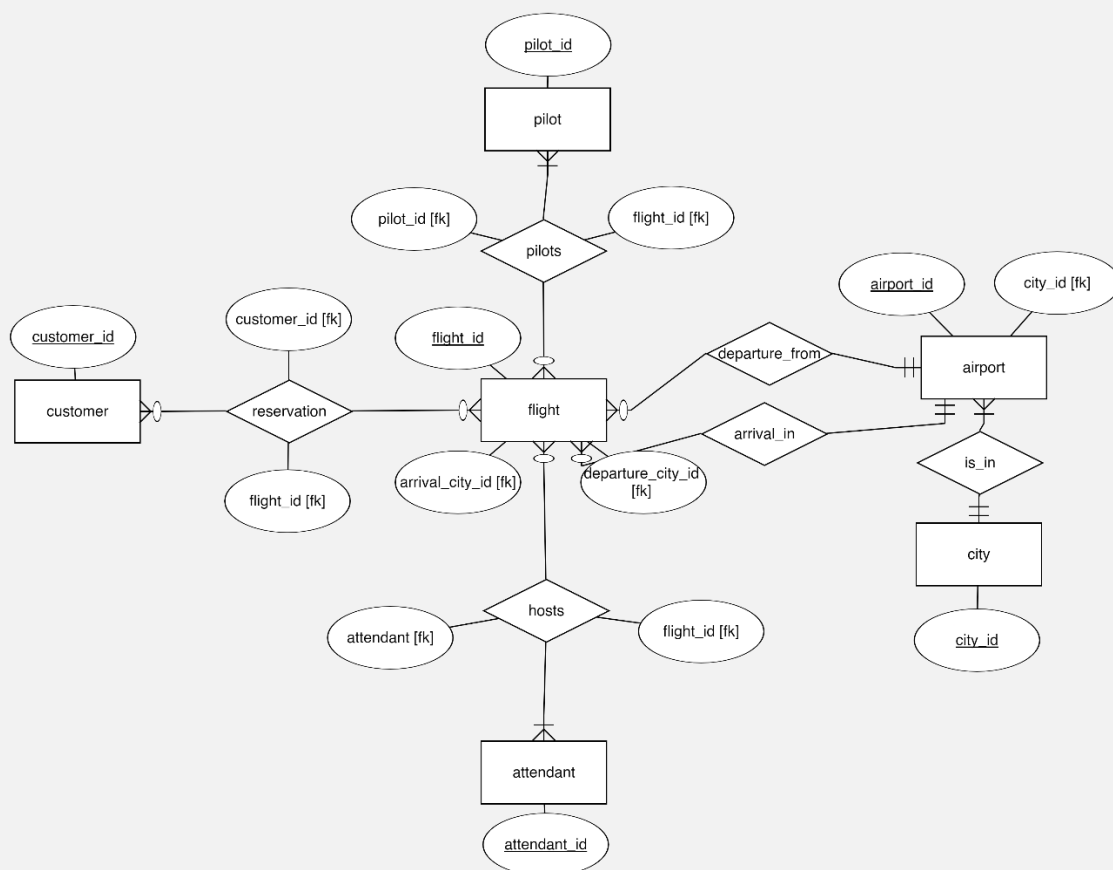


c. مدیر هر شعبه در جدول staff ذخیره میشود. نمره منفی در کنار هر شکایت ذخیره میشود.



	customer	flight	attendant	pilot	airport	city
customer		reserves				
flight	Is reserved by		Has some	Piloted by	departures from and arrives in	
attendant		hosts				
pilot		pilots				
airport		Departure from and arrival of				Is in
city					Has some	

مفروضات: مشتری میتواند قبل از رزرو هم حساب داشته باشد. یک مهماندار یا خلبان موقع استخدام سابقه در سیستم ندارد. شهر بدون فرودگاه در سیستم ثبت نمیشود



id	name	dept_name	tot_cred	gpa
25525	Moreira	Comp. Sci.	107	2.6829268292682927
75040	Kruglyak	Pol. Sci.	99	1.9666666666666667
31624	Keuk	Geology	107	3.1034482758620690
64893	Katzenb	Statistics	78	2.5121951219512195
59920	Dano	Civil Eng.	16	2.3333333333333333
38602	Torres	Cybernetics	108	3.1891891891891892
21401	Garze	Physics	92	2.2127659574468085
93004	Gibbs	Finance	129	2.8500000000000000
38899	Murphy	Marketing	30	2.6538461538461538
75241	Hughes	Languages	79	2.8771929824561404
98830	Frolova	Comp. Sci.	13	3.2200000000000000
37101	Falconer	Physics	78	3.0444444444444444
4508	Cochran	English	127	2.4925373134328358
32130	Bannac	Biology	56	2.7454545454545455
56755	Paul	Mech. Eng.	62	2.6666666666666667
11510	Rakoj	Mech. Eng.	37	3.1320754716981132
3335	Otterm	Finance	73	1.9848484848484848
61081	Bai	Geology	100	2.5789473684210526
33651	Seike	Geology	2	3.2857142857142857
8251	Kite	History	47	2.9552238805970149
10705	Terauchi	Physics	48	2.2542372881355932
64082	Boyle	Languages	87	2.7500000000000000
59517	Harrison	Elec. Eng.	81	3.3823529411764706
91569	Pavlovico	Marketing	90	3.1395348837209302