

به نام خدا

پاسخنامه تکلیف شماره چهار

پایگاه داده ترم پاییز 1400

1. مراحل دهگانه طراحی یک پایگاه داده را نام برده و هرکدام را مختصراً توضیح دهید .

صفحه 13 اسلاید مربوط به طراحی ER به صورت گام به گام

2. چگونگی پیاده سازی Composite attributes و Multivalued attributes را در سطح Physical توضیح دهید .

اسلاید 6.47 و 6.48 در اسلاید های مربوط به فصل 6

3. تفاوت بین Weak Entity Set و Strong Entity Set را توضیح دهید و با توجه به اینکه می توان هر Weak Entity Set را با افزودن Attribute های مناسب تبدیل به Strong Entity نمود توضیح دهید این کار چه مشکلی ایجاد میکند و چرا ما در طراحی های خود Weak Entity Set ها را داریم ؟

Strong Entity Set برای مشخص نمودن یک انتیتی یکتای خود به موجودیت های دیگر وابسته نیست اما Weak Entity Set برای تعیین یک انتیتی خود وابسته به موجودیت های دیگر است و به تنهایی و با ویژگی های خود نمی تواند یک رکورد یکتا را مشخص کند .

چون با افزودن ستون های کلیدی از جدول قوی به موجودیت ضعیف که باعث یکتا شدن آن میشوند اطلاعات مربوطه هم در Relation بین دو موجودیت و همچنین در موجودیت ضعیف قرار میگیرند باعث افزونگی داده (Redundancy) میشود و در نهایت ممکن است به ناسازگاری (inconsistency) منجر شود. همچنین در صورتی که Relation بین دو جدول را حذف کنیم و صرفاً با استفاده از ویژگی هایی که به موجودیت ضعیف اضافه نمودیم آن را یکتا کنیم دیگر رابطه بین این دو جدول در طراحی ER ملموس و قابل مشاهده نیست .

4. نمودار ER زیر مدلسازی بخشی از فعالیت بانک ها است. هر بانک تعدادی شعبه دارد و در هر شعبه تعدادی شماره حساب و وام وجود دارد.

a. موجودیت های Weak و Strong کدامند؟

موجودیت Bank Branch ضعیف و بقیه Strong هستند .

b. تمام Relation ها و Cardinality های مربوط به آن ها در نمودار زیر را توضیح دهید .

رابطه بین Bank و Bank Branch به نام Branches که یک رابطه یک به چند است یعنی هر بانک مربوط به یک شعبه است و هر بانک چندین شعبه دارد .

رابطه بین Accounts و Bank Branches به نام Accts که یک رابطه یک به چند است یعنی هر شعبه می تواند چندین حساب مرتبط داشته باشد و هر حساب مربوط به یک شعبه بانک است .

رابطه بین Loan و Bank Branches به نام Loans که یک رابطه یک به چند است یعنی هر شعبه می تواند چندین وام مرتبط داده باشد و هر وام مربوط به یک شعبه بانک است .

رابطه بین Loan و Customer به نام L-C که یک رابطه چند به چند است یعنی هر وام می تواند مرتبط با چند مشتری باشد و هر مشتری می تواند چندین وام داشته باشد .

رابطه بین Account و Customer به نام A-C که یک رابطه چند به چند است یعنی هر حساب می تواند مرتبط با چند مشتری باشد و هر مشتری می تواند چندین حساب داشته باشد .

c. این نمودار ER به چه خواسته هایی از کاربران پاسخ می دهد؟ (3 مورد ذکر کنید)

1- وام دهی به مشتریان

2-تعريف حساب

3-مشاهده اطلاعات مشتریان

d. برای پاسخگویی به هر کدام از نیازمندی های مرحله قبل چه پردازش هایی لازمند؟

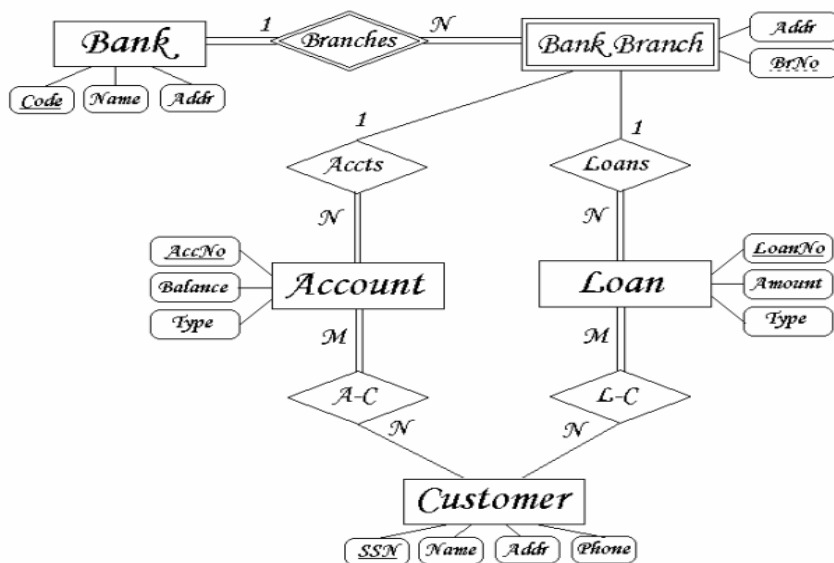
1- برای وام دهی ابتدا لازم است یک رکورد مشتری انتخاب یا ایجاد گردد و کلید اصلی آن یعنی SSN را داشته باشیم سپس یک وام جدید ایجاد کرده و یا یکی از وام های موجود انتخاب و شماره وام که همان کلید اصلی آن است را بدست آورده و سپس با استفاده از این دو فیلد یک رکورد L-C ایجاد نمود .

2- برای تعریف حساب بانکی لازم است ابتدا شماره شعبه بانکی واکنشی شود که همان BrNO است همچنین یک مشتری جدید ایجاد و یا از مشتریان قبلی شماره مشتری که همان SSN است را بدست آورده و سپس با پر کردن اطلاعات حساب بانکی و وارد نمودن شماره شعبه بانکی یک رکورد حساب بانکی جدید ایجاد نمود.

3- برای ویرایش اطلاعات شخصی مشتری کافیت مشتری مورد نظر را واکنشی کرده و شماره مشتری مورد نظر یعنی SSN را بدست آورده و سپس ویژگی های مدنظر را ویرایش نموده و همان انتیتی را بروزرسانی نمود.

e. اگر برای وام دهی نیاز به ثبت یک ضامن نیز وجود داشته باشد چه تغییری در نمودار لازم است ؟

برای این کار میتوان یک موجودیت جدا برای ضامن تعریف نمود و یا آن را در موجودیت مشتریان یا همان Customer در نظر گرفت و در رابطه (Relation) بین وام و مشتری باید یک رابطه مشابه دیگر نیز برای ضامن تعریف نمود که یک طرف آن وام و طرف دیگر آن موجودیت مشتری باشد و یا اگر موجودیت مستقلی برای ضامن در نظر گرفتیم طرف دیگر آن ضامن باشد .



5. یک رستوران قصد تولید یک DBMS برای ثبت داده های خود دارد. در مرحله مهندسی نیازها، به اطلاعات زیر رسیده ایم:

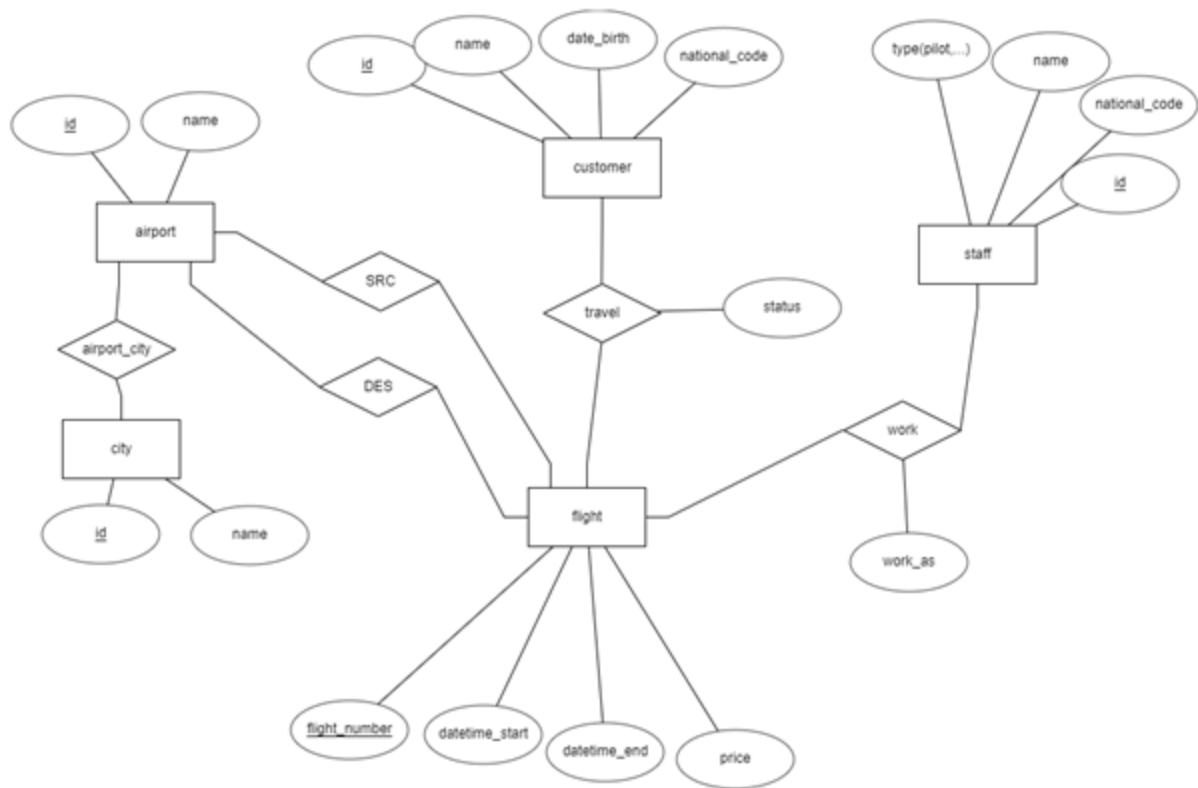
- رستوران شعبات مختلفی در سطح ایران دارد که شعب تهران مستقیماً زیر نظر مدیر شعبه اصلی اداره می شوند و شعب شهرستان هر کدام مدیر مستقلی دارند که تحت نظر مدیر شعبه اصلی فعالیت می کند.
- رستوران با تامین کنندگان مواد غذایی مختلفی قرارداد دارد که محموله های غذایی مختلفی را بر حسب نیاز هر روز یا هر چند روز یک بار به شعب مختلف تحویل می دهند.
- هر شعبه چندین آشپز و پیشخدمت و کارگر و راننده دارد که با توجه به ساعات کاری شان، دستمزد ماهانه می گیرند.
- غذاهای رستوران که نوع و میزان مواد غذایی هر کدام مشخص است، با قیمتهای معین توسط مشتریان (چه مشترک و چه موردی) خریداری می شوند.
- به جز مشتریان مشترک و موردی، چندین سازمان با شعب مختلف رستوران قرارداد ماهیانه دارند و غذاهای مشخصی را هر روز دریافت می کنند.
- شکایات مشتریان از هر نوع (غذا، سرویس و ...) ثبت می شود و در اختیار مدیریت کل رستوران قرار می گیرد و از سوی او بر حسب مورد نمره منفی به شعبه تعلق می گیرد.
- هر مشتری پس از صرف غذا نمره ای بین ۰ تا ۱۰ به غذا و سرویس رستوران می دهد.

- a. ماتریس روابط بین موجودیت ها را رسم نمایید .
- b. یک ERD برای این سیستم رسم کنید و در صورت نیاز مفروضات خود را با ذکر آنها وارد کنید.
- c. برای آن یک پایگاه داده رابطه ای طراحی کنید. (نوشتن جداول و کلید های اصلی و خارجی لازم و کافیست)

6. می خواهیم یک پایگاه داده برای یک شرکت هواپیمایی طراحی نماییم که در آن بتوان اطلاعات مربوط به مشتریان و رزرو آن ها ، پروازها و وضعیت آنها، مسیر پروازها و زمان بندی آن و مهمانداران و خلبان هر پرواز را ذخیره و در فرایند های مختلف استفاده نماییم. پروازها ممکن است بین هر دو فرودگاه در سطح دنیا انجام شوند. یک شهر میتواند بیش از یک فرودگاه داشته باشد، و هنگام جستجو باید بتوانیم وجود یا عدم وجود پرواز بین دو شهر را نیز بررسی کنیم. هر پرواز می تواند تا دو خلبان و تا ده مهماندار و تا 800 مسافر داشته باشد. اطلاعات هر موجودیت را با توجه به نیازها و فرایند هایی که در دنیای واقعی وجود دارند در نظر بگیرید (مفروضات خود را بنویسید) و ماتریس روابط موجودیت ها، یک نمودار E-R و را طراحی کنید و کلیدهای اصلی و کلیدهای خارجی هر موجودیت را مشخص نمایید .

	city	airport	flight	staff	customer
city		has			
airport	Located in		SRC/DES		
flight		SRC/DES		has	has
staff			Work in		

customer			Travel by		
----------	--	--	-----------	--	--



	PK(bold) & FK(underline)
city	id
airport	Id , <u>city_id</u>
flight	Flight_number , <u>src_airport_id</u> , <u>des_airport_id</u>
staff	id
customer	id

travel	<u>customer_id</u> , <u>flight_id</u>
work	<u>staff_id</u> , <u>flight_id</u>

7. با توجه به پایگاه داده معرفی شده در کلاس (University: Large Version) :
a. یک ستون به جدول Student به نام GPA جهت نگهداری معدل کل دانشجو به صورت عددی اعشاری مطابق زیر اضافه نمایید

ALTER TABLE STUDENT ADD GPA DECIMAL(3,2);

b. با توجه به جدول زیر معدل کل هر دانشجو را محاسبه و فیلد GPA وی را آپدیت نمایید (معدل = مجموع (نمره (Score معادل با Grade) درس * واحد آن) برای درس ها / مجموع تعداد واحدها)

Score	Grade
4	A,A+,A-
3	B,B+,B-
2	C+
1	C
0	C-

```
UPDATE STUDENT AS S SET GPA= (SELECT SUM(SCORE)/SUM(credits)
FROM
(SELECT ID,C.course_id,credits,
(CASE WHEN grade LIKE 'A%' THEN 4*credits
      WHEN grade LIKE 'B%' THEN 3*credits
      WHEN grade='C+' THEN 2*credits
      WHEN grade='C' THEN 1*credits
      WHEN grade='C-' THEN 0 END
) AS SCORE
FROM TAKES AS T
INNER JOIN COURSE AS C ON C.course_id=T.course_id
WHERE S.ID=T.ID
) AS STU_SCORES
)
```