



پایگاه داده

تمرین سری اول

۴۰۲۱۷۱۰۷۵

پارسا ملکیان

تمارین تئوری

سوال ۱

(آ) اگر رابطه S خالی باشد، آن گاه $R \div S$ به یک رابطه خالی منجر خواهد شد.

نادرست - اگر رابطه S خالی باشد، عملیات تقسیم $(R \div S)$ به یک رابطه شامل همه تاپل‌های ممکن از دامنه مربوطه منجر می‌شود، نه یک رابطه خالی. این به دلیل تعریف عملگر تقسیم است که در آن برای هر تاپل r از R باید به ازای تمام تاپل‌های s از S یک تاپل در نتیجه وجود داشته باشد. وقتی S خالی است، این شرط به طور پیش فرض برای همه تاپل‌های ممکن برقرار است.

(ب) یک موجودیت که از لحاظ مفهومی وابسته به وجود موجودیتی دیگر است همیشه موجودیتی ضعیف است.

نادرست - اگر خودش ویژگی‌ای داشته باشد که آن را بدون اطلاع از موجودیت قوی وابسته به آن، متمایز کند، آنگاه موجودیت ضعیف به حساب نمی‌آید.

(ج) می‌توان از هر سه رابطه دو گانی یک رابطه سه گانی را نتیجه گرفت.

نادرست - لزوماً نمی‌توان از سه رابطه دوتایی یک رابطه سه‌تایی را استخراج کرد. مثال نقض: سه رابطه دوتایی $R(A,B)$ ، $S(B,C)$ و $T(A,C)$ لزوماً به معنای وجود یک رابطه سه‌تایی $U(A,B,C)$ نیست، زیرا ممکن است ترکیب‌هایی از مقادیر وجود داشته باشند که در رابطه‌های دوتایی ظاهر شوند اما در رابطه سه‌تایی مورد نظر قرار نگیرند.

(د) در مدل ER اگر دو رابطه بین دو موجودیت یکسان وجود داشته باشد، می‌توان هر دو را در یک رابطه واحد در مدل رابطه‌ای نگاشت کرد.

نادرست - در صورتی که دو رابطه متفاوت بین دو موجودیت وجود داشته باشد، نمی‌توان آنها را در یک رابطه واحد نگاشت کرد، زیرا هر کدام معنای متفاوتی دارند و ممکن است ویژگی‌های متفاوتی داشته باشند. مثال نقض: رابطه «استخدام» و «نظارت» بین دو موجودیت «کارمند» و «بخش» دو معنای متفاوت دارند و باید به دو جدول مجزا نگاشت شوند.

(ه) تفاوت اصلی عملیات‌های Union و Join در جبر رابطه‌ای این بوده که در عملیات Join باید همه خصیصه‌ها یکسان باشند.

نادرست - این گزاره برعکس است. در عملیات Union باید همه خصیصه‌ها (ستون‌ها) یکسان باشند، در حالی که در Join چنین محدودیتی وجود ندارد. Join برای ترکیب دو رابطه بر اساس شرط مشخصی استفاده می‌شود، در حالی که Union برای اجتماع دو رابطه با ساختار یکسان است.

و) عملیات جبر رابطه ای $\pi_{E(\sigma_{D=1}(T \bowtie S))}$ با عملیات $\pi_{B,E(\sigma_{D=1}(T))}$ برابر است.

نادرست - این دو عبارت به طور کلی معادل نیستند. در سمت چپ، ابتدا T و S پیوند می‌شوند، سپس شرط $D=1$ اعمال می‌شود و در نهایت ستون E انتخاب می‌شود. در سمت راست، ابتدا شرط $D=1$ روی T اعمال می‌شود، سپس ستون‌های B و E انتخاب می‌شوند و همزمان فقط ستون B از S انتخاب شده و سپس این دو نتیجه پیوند می‌شوند. نتایج می‌توانند متفاوت باشند.

زا) بهترین راه پیاده سازی رابطه IS A در یک نمودار EER به یک مدل رابطه ای این است که، کلید اصلی موجودیت های فرزند را به عنوان کلید خارجی به موجودیت پدر اضافه کنیم.

نادرست - این روش برعکس است. در پیاده سازی رابطه IS-A، کلید اصلی موجودیت پدر به عنوان کلید خارجی در موجودیت فرزند قرار می‌گیرد، نه برعکس. این مدل سازی نشان می‌دهد که موجودیت فرزند نوع خاصی از موجودیت پدر است و باید کلید اصلی پدر را به عنوان کلید خارجی (و معمولاً به عنوان بخشی از کلید اصلی خود) داشته باشد.

سوال ۲

(آ)

``πMajor Name, (σGPA > 3.88 ∧ EnrollmentYear < 2018(Students))``

این عبارت نام رشته تحصیلی (Major Name) دانشجویانی را برمی‌گرداند که معدل آنها بیشتر از ۳٫۸۸ است و پیش از سال ۲۰۱۸ ثبت نام کرده‌اند.

(ب)

``(σCredits > 4(Courses)) ⋈ Professors.Department
= Courses.Department πRank Name, (σTenureStatus
= No(Professors))``

این عبارت اطلاعات درس‌هایی با بیش از ۴ واحد را با اطلاعات استادانی که وضعیت تصدی آنها "No" است (یعنی استادان غیررسمی) و در همان دپارتمان هستند، پیوند می‌دهد. نتیجه شامل رتبه علمی و نام استادان غیررسمی است که درس‌های با بیش از ۴ واحد را در دپارتمان خود ارائه می‌دهند.

(ج)

$$\pi_{Name ProfessorID, (\sigma_{ProfessorID \in \pi_{LeadProfessorID}(\sigma_{Funding > 20000}(ResearchProjects))}(Professors))}$$

این عبارت نام و شناسه استادانی را برمی‌گرداند که رهبری پروژه‌های تحقیقاتی با بودجه بیش از ۲۰,۰۰۰ را بر عهده دارند. به عبارت دیگر، فهرست استادانی که مسئول پروژه‌های تحقیقاتی با بودجه بالا هستند.

(د)

$$\pi_{Name StudentID, (\sigma_{StudentID \notin \pi_{StudentID}(Enrollments)}(Students))}$$

این عبارت نام و شناسه دانشجویانی را برمی‌گرداند که در هیچ درسی ثبت‌نام نکرده‌اند (یعنی شناسه آنها در جدول Enrollments وجود ندارد). به عبارت دیگر، فهرست دانشجویانی که فعلاً در هیچ درسی شرکت نمی‌کنند.

(ه)

$$\pi_{Department Name, (\sigma_{ProfessorID \in \pi_{ProfessorID}(Courses \bowtie Enrollments \sigma_{Grade < 2.5}(Enrollments))}(Professors))}$$

این عبارت نام دپارتمان و استادانی را برمی‌گرداند که دروسی ارائه می‌دهند که دانشجویان در آن درس‌ها نمره کمتر از ۲/۵ گرفته‌اند.

(و)

$$\pi_{Name StudentID, (\sigma_{StudentID \in \pi_{StudentID}(Enrollments \bowtie Courses \sigma_{Department \in \pi_{Department}(\sigma_{Funding > 50000}(ResearchProjects))}(Courses))}(Students))}$$

این عبارت نام و شناسه دانشجویانی را برمی‌گرداند که در دروسی ثبت‌نام کرده‌اند که مربوط به دپارتمان‌هایی هستند که پروژه‌های تحقیقاتی با بودجه بیش از ۵۰,۰۰۰ دارند. به عبارت دیگر، فهرست دانشجویانی که در دروسی شرکت می‌کنند که دپارتمان ارائه‌دهنده آنها دارای پروژه‌های تحقیقاتی پر بودجه است.

سوال ۳

موجودیت‌های اصلی

۱. دانشجو (STUDENT): شامل اطلاعات دانشجویان با شناسه یکتا
۲. استاد (PROFESSOR): شامل اطلاعات اساتید با شناسه یکتا
۳. درس (COURSE): اطلاعات پایه درس‌ها شامل شماره یکتا، عنوان، تعداد واحد، سرفصل و پیش‌نیازها
۴. ارائه درس (COURSE_OFFERING): هر نمونه از ارائه یک درس در یک نیمسال خاص
۵. ثبت‌نام (ENROLLMENT): ارتباط بین دانشجو و درس ارائه شده که شامل نمره نیز می‌شود
۶. دانشکده (DEPARTMENT): اطلاعات دانشکده‌ها

روابط

۱. دانشجو در درس‌های ارائه شده ثبت‌نام می‌کند (ثبت‌نام)
۱. در هر فرم ثبت‌نام یک دانشجو حضور دارد ولی هر دانشجو میتواند بیش از ۱ فرم ثبت‌نام داشته باشد
۲. فرم ثبت‌نام موجودیت ضعیف و این رابطه رابطه وابستگی است زیرا هر فرم ثبت‌نام به حضور دانشجو وابسته است
۲. استاد درس‌های ارائه شده را تدریس می‌کند
۱. در هر ارائه درس، چند استاد میتوانند تدریس کنند و از طرفی هر استاد میتواند چند درس را ارائه دهد
۲. هر ارائه درس حداقل یک استاد نیاز دارد پس این رابطه، وابستگی میشود
۳. درس‌ها در نیمسال‌های مختلف ارائه می‌شوند (ارائه درس)
۱. در هر ارائه درس، فقط یک درس ارائه میشود اما یک درس میتواند در ترم‌های متفاوت و توسط اساتید متفاوت ارائه شود
۲. هر ارائه درس نیاز به یک درس دارد و بدون آن موجودیت آن معنا ندارد پس این رابطه نیز وابستگی میشود.

۴. دانشکده، اساتید را استخدام می‌کند

۱. هر دانشکده میتواند چندین استاد داشته باشد اما هر استاد فقط در یک دانشکده کار میکند

۵. برخی از دروس، پیش‌نیاز دروس دیگر هستند

۱. هر درس میتواند بیش از یک درس پیش‌نیاز داشته باشد یا خودش پیش‌نیاز بیش از یک درس باشد

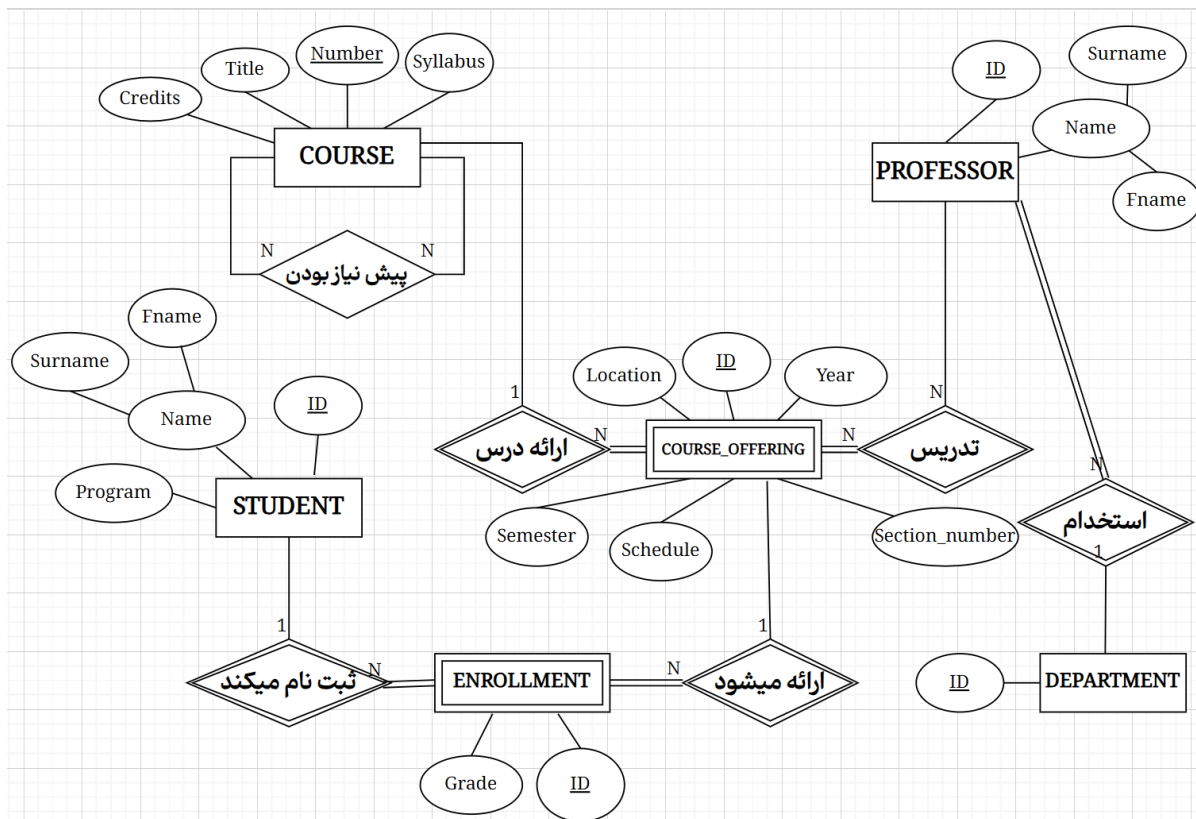
۲. یک درس میتواند پیش‌نیاز درسی نباشد و همچنین پیش‌نیاز نداشته باشد پس رابطه ساده است

۶. در یک پروسه ثبت‌نام، درس ارائه شده وجود دارد (ارائه میشود)

۱. هر ارائه درس میتواند در چند فرم ثبت‌نام شود اما هر فرم فقط یک ارائه درس دارد

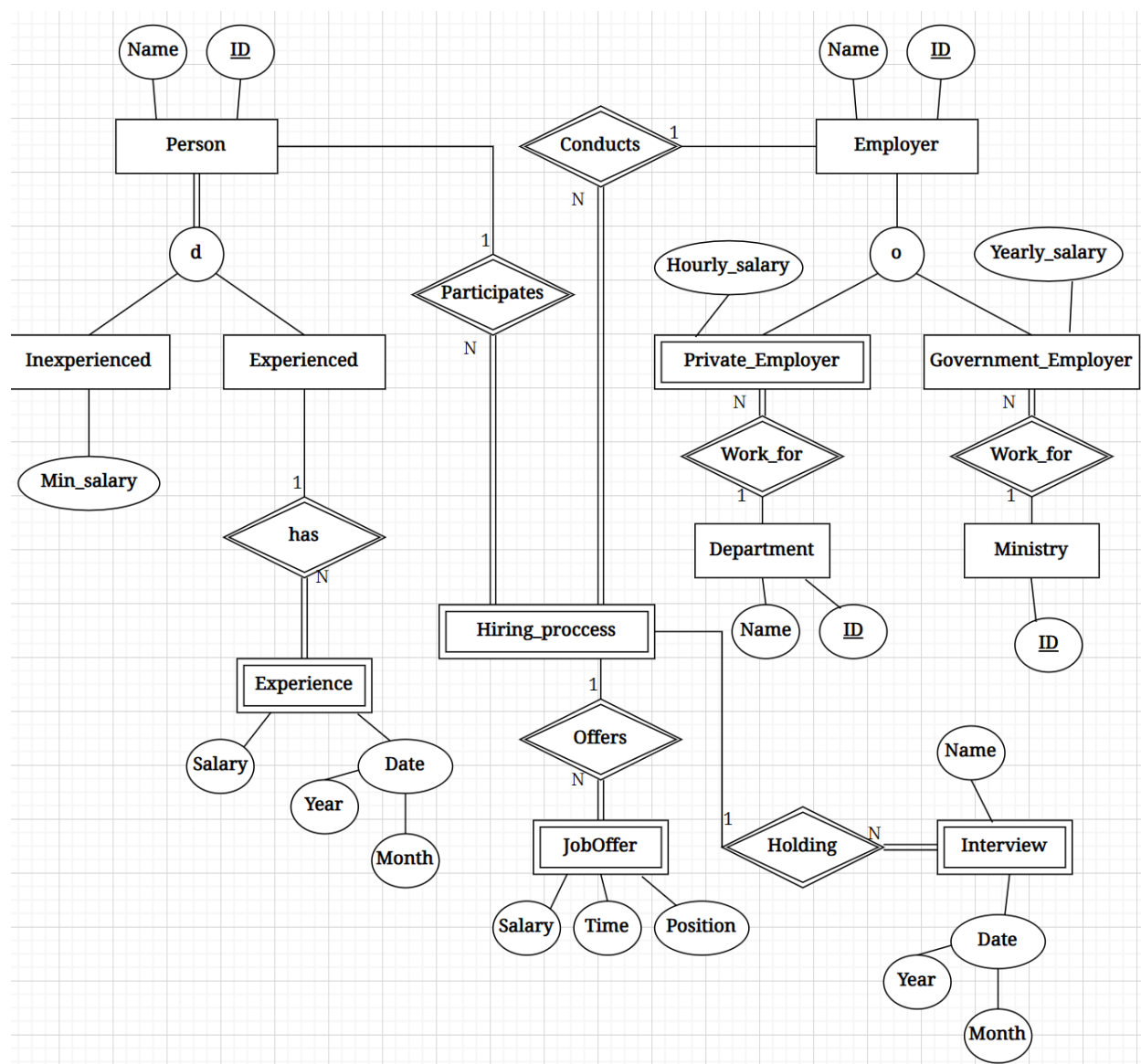
۲. در هر پروسه ثبت‌نام حتما یک ارائه درس وجود دارد و بدون آن ثبت نام بی‌معنی میشود

نمودار نهایی



فایل این نمودار به صورت HW1Q3 در فایل zip موجود است.

سوال ۴

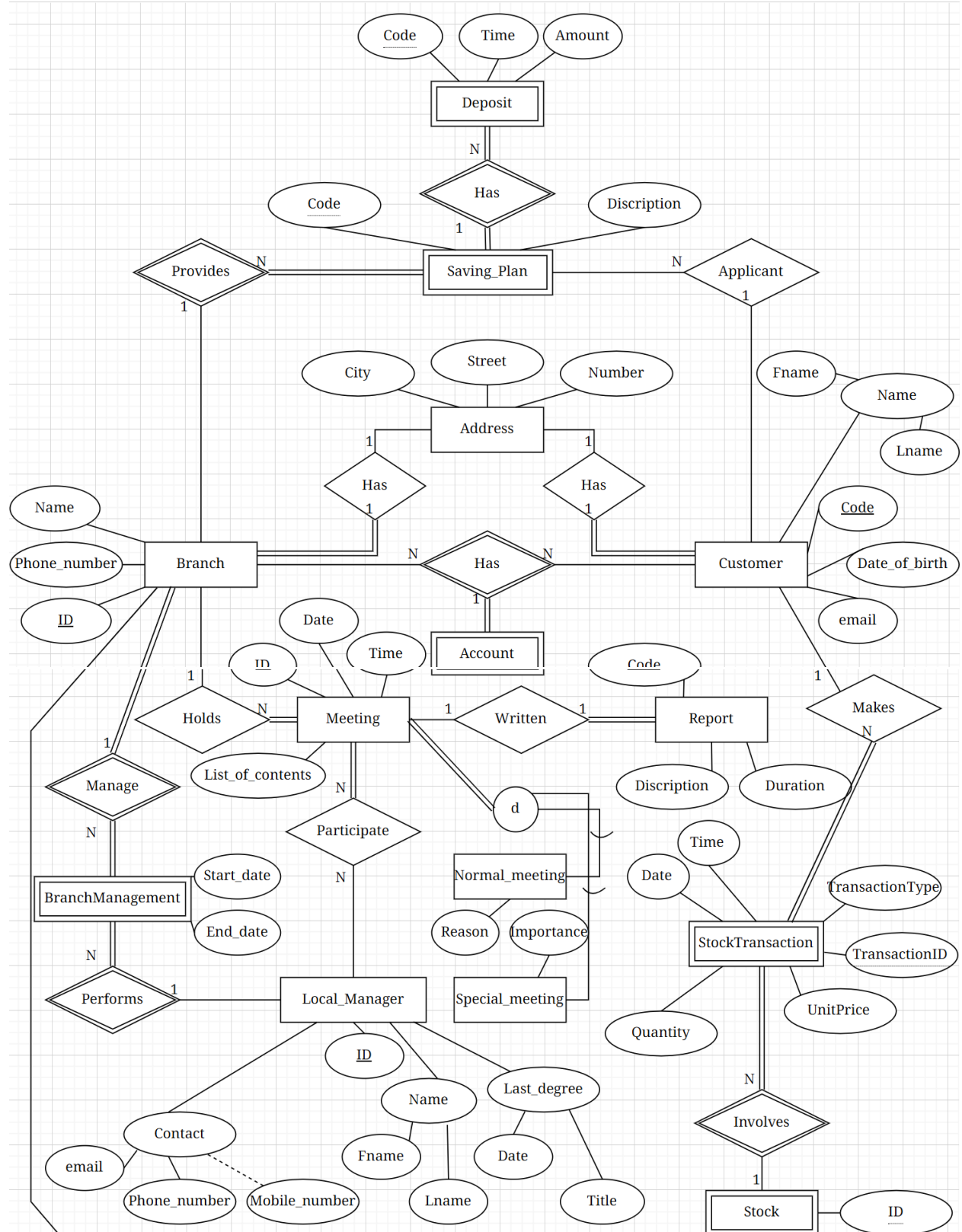


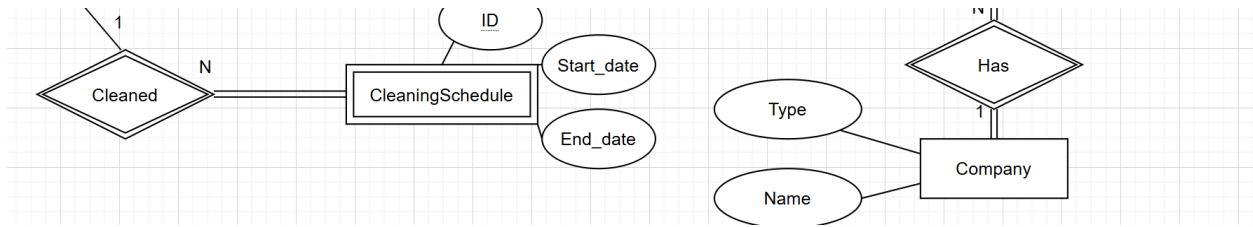
فایل این طراحی به نام HW1Q4 در فایل zip قرار دارد

سوال ۵

(الف)

فایل این طراحی به نام HW1Q5 در فایل zip قرار دارد





(ب)

شمای منطقی پایگاه داده بانک

جداول و ویژگی‌ها

شعبه (Branch)

- **branch_id** (PK)
- name
- address(FK -> Address)
- phone_number

مشتری (Customer)

- code (PK)
- name(first_name and last_name)
- birth_date
- address(FK -> Address)
- Email

آدرس (Address)

- City
- Street
- Number

حساب (Account)

- **account_id** (PK)
- *branch_id* (FK -> Branch)
- *customer_ssn* (FK -> Customer)

مدیر محلی (Local_Manager)

- **manager_id** (PK)

- name(fname and lname)
- contact(email and phone_number and mobile_number)
- last_degree(title and date)

جلسه (Meeting)

- **meeting_id** (PK)
- *branch_id* (FK -> Branch)
- date
- time
- list_of_contents
- meeting_type (عادی/فوق العاده)
- importance (برای جلسات فوق العاده)
- reason (برای جلسات عادی)

گزارش جلسه (Meeting_Report)

- **report_code** (PK)
- *meeting_id* (FK -> Meeting)
- description
- duration

طرح پس انداز (Saving_Plan)

- **plan_code** (PK)
- *branch_id* (FK -> Branch)
- *customer_ssn* (FK -> Customer)
- description

سپرده (Deposit)

- **deposit_code** (PK)
- *plan_code* (FK -> Saving_Plan)
- time
- amount

سهام (Stock)

- **stock_id** (PK)

- company_name_type (FK -> Company)

کمپانی (Company)

- name (بخشی از کلید اصلی)
- type (بخشی از کلید اصلی)

تراکنش سهام (Stock_Transaction)

- transaction_id (PK)
- stock_id (FK -> Stock)
- customer_ssn (FK -> Customer)
- transaction_time
- transaction_date
- transaction_type (خرید/فروش)
- unit_price
- quantity

مدیریت شعبه (Branch_Management)

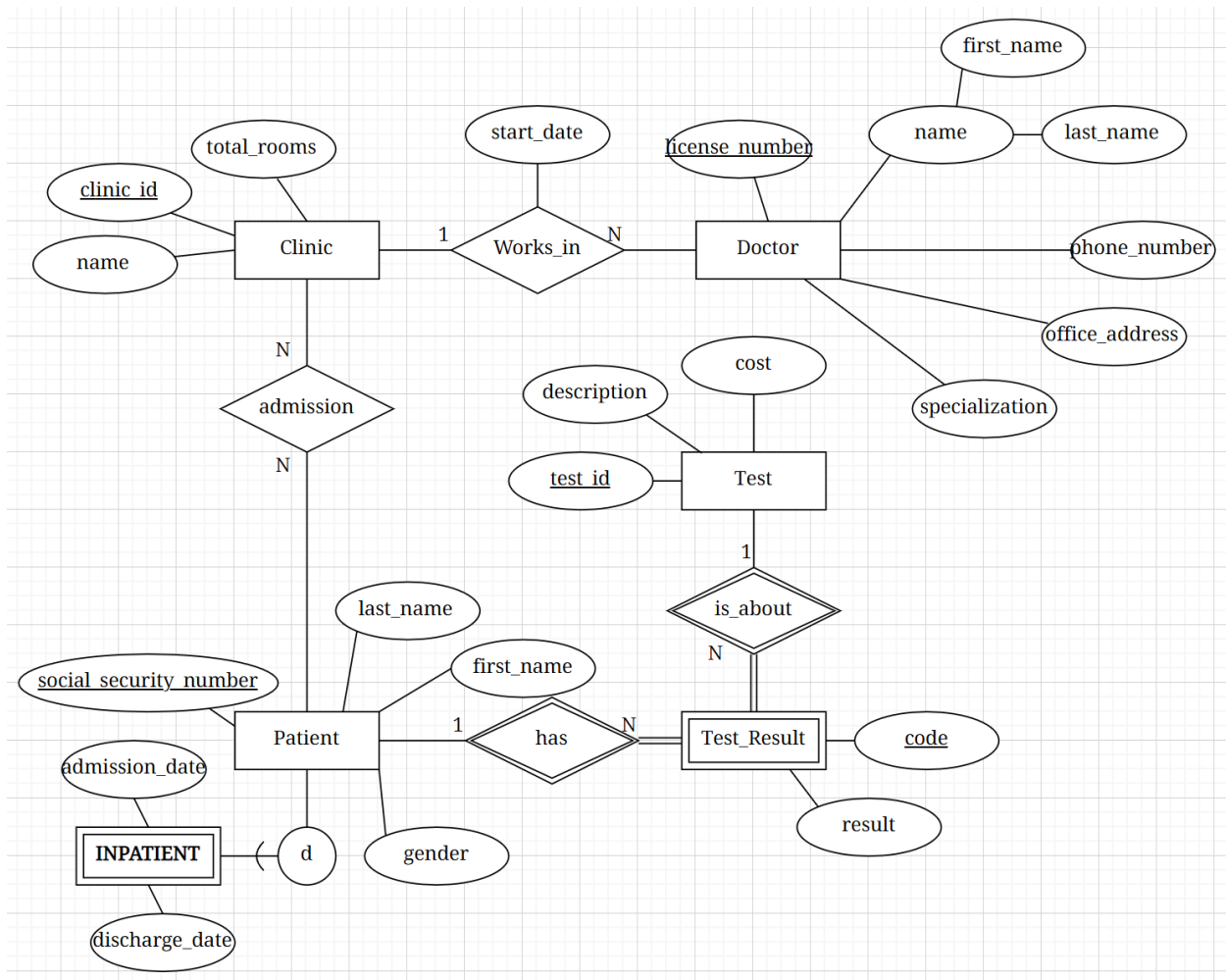
- branch_id (بخشی از کلید اصلی، کلید خارجی مربوط به Branch)
- manager_id (بخشی از کلید اصلی، کلید خارجی مربوط به Local_Manager)
- start_date
- end_date (باشد null می تواند)

برنامه نظافت (Cleaning_Schedule)

- branch_id (بخشی از کلید اصلی، کلید خارجی مربوط به Branch)
- id (بخشی از کلید اصلی)
- start_time
- end_time

سوال ۶

آ) مدل ER بیمارستان



فایل این طراحی به نام HW1Q6 در فایل zip قرار دارد

ب) تبدیل مدل ER به مدل منطقی (Logical Design)

با توجه به مدل ER بالا، مدل منطقی به صورت زیر است:

۱. کلینیک (CLINIC)

- نام رابطه CLINIC :
- ویژگی‌ها {clinic_id, name, total_rooms} :
- کلید اصلی clinic_id :

۲. پزشک (DOCTOR)

- نام رابطه DOCTOR :
- ویژگی‌ها {license_number, first_name, last_name, phone_number, office_address, specialization, clinic_id}
- کلید اصلی license_number :
- کلید خارجی clinic_id: ارجاع به CLINIC

۳. بیمار (PATIENT)

- نام رابطه PATIENT :
- ویژگی‌ها {social_security_number, first_name, last_name, gender, clinic_id}
- کلید اصلی patient_id :
- کلید خارجی clinic_id: ارجاع به CLINIC

۴. بیمار بستری (INPATIENT)

- نام رابطه: INPATIENT
- ویژگی‌ها: {patient_id, admission_date, discharge_date}
- کلید اصلی: patient_id
- کلید خارجی: patient_id ارجاع به PATIENT (ارث‌بری از PATIENT)

۵. آزمایش (TEST)

- نام رابطه TEST :
- ویژگی‌ها {test_id, description, result_id}
- کلید اصلی test_id :
- کلید خارجی :

▪ result_id ارجاع به TEST_RESULT

۶. نتیجه آزمایش (TEST_RESULT)

- نام رابطه: TEST_RESULT
- ویژگی‌ها: {code, cost, result, patient_id, test_id}

- کلید اصلی: code
- کلید خارجی:
- patient_id ارجاع به PATIENT
- test_id ارجاع به TEST

روابط بین جداول

۱. رابطه بین پزشک و کلینیک:

- نام رابطه: Works_In (کار می کند در)
- نوع رابطه: چند به یک (هر پزشک فقط در یک کلینیک کار می کند)
- اجرای رابطه: فیلد clinic_id به عنوان کلید خارجی در جدول Doctor
- ویژگی start_date برای این رابطه نشان دهنده زمان شروع کار است

۲. رابطه بین بیمار و کلینیک:

- نام رابطه: Has_Admission (دارد پذیرش)
- نوع رابطه: چند به چند (هر بیمار می تواند در چند کلینیک پذیرش داشته باشد)

۳. رابطه بین بیمار و نتایج آزمایش:

- نام رابطه: Has_Tests (دارد آزمایش ها)
- نوع رابطه: یک به چند (هر بیمار می تواند چندین نتیجه آزمایش داشته باشد)
- اجرای رابطه: فیلد patient_id به عنوان کلید خارجی در جدول TestResult

۴. رابطه بین آزمایش و نتایج آزمایش:

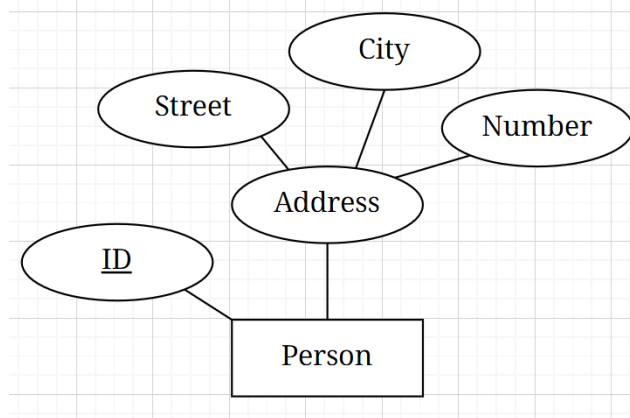
- نام رابطه: Generates (تولید می کند)
- نوع رابطه: یک به چند (هر آزمایش می تواند چندین نتیجه داشته باشد)
- اجرای رابطه: فیلد test_id به عنوان کلید خارجی در جدول TestResult

سوال ۷

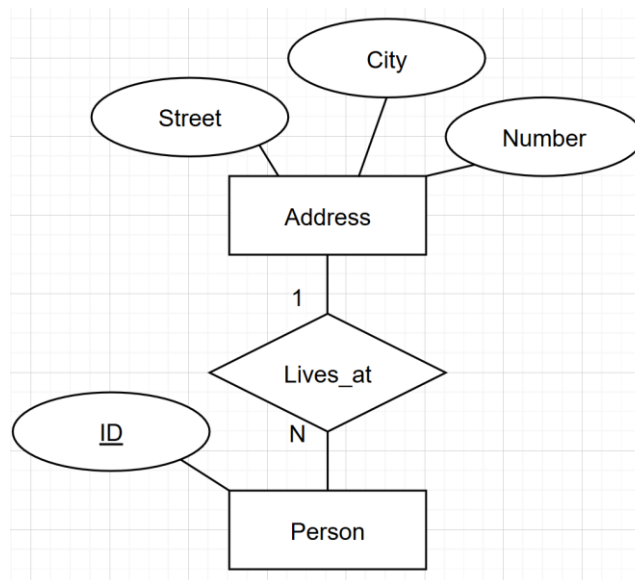
آ مدل سازی افراد و آدرس های خانه

i. دو مدل ER مختلف برای نمایش Persons و آدرس های آنها

مدل اول: آدرس به عنوان یک ویژگی از موجودیت Person در نظر گرفته شده است.



مدل دوم: آدرس به عنوان یک موجودیت مجزا (Address) در نظر گرفته شده است و با ارتباط Lives_at به Person متصل شده است.



ii. چرا یکی را به جای دیگری انتخاب می کنیم؟

دلایل انتخاب مدل اول (آدرس به عنوان ویژگی):

- سادگی: پیاده سازی و کوئری های ساده تر
- کارایی بهتر: نیاز به عملیات JOIN کمتر هنگام بازیابی اطلاعات

- مناسب برای سیستم‌های ساده: اگر جزئیات آدرس یا عملیات پیچیده روی آدرس‌ها نیاز نیست

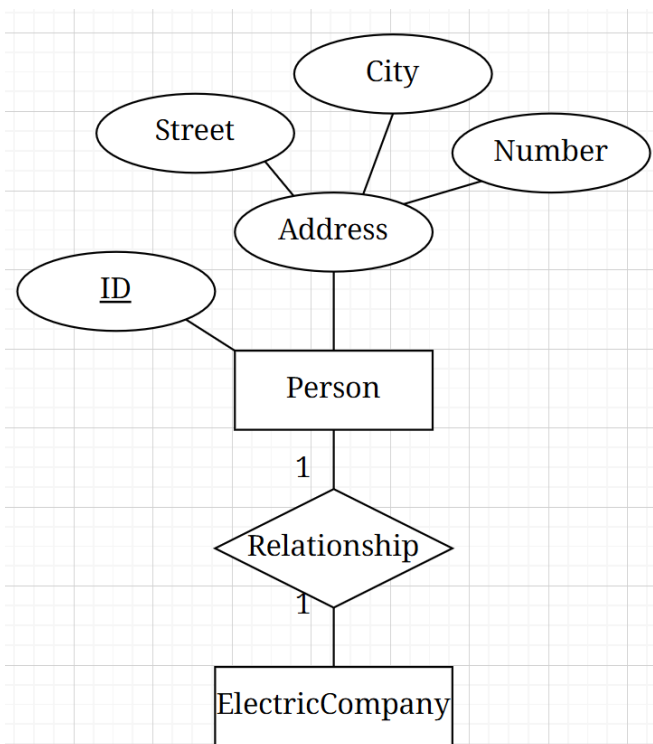
دلایل انتخاب مدل دوم (آدرس به عنوان موجودیت):

- جلوگیری از افزونگی داده: اگر چندین فرد در یک آدرس زندگی کنند
 - ساده شدن عمل تغییر: اگر نام خیابان یا شهری عوض شود نیازی به ایجاد تغییر در $O(n)$ نیست و در $O(1)$ تغییرات انجام میشود
 - امکان ذخیره جزئیات بیشتر: می‌توان اطلاعات بیشتری مانند کد پستی، استان، و غیره را ذخیره کرد
 - انعطاف‌پذیری بیشتر: امکان ثبت سابقه آدرس‌های قبلی افراد
 - پشتیبانی از ارتباط چند به چند: هر فرد می‌تواند چندین آدرس داشته باشد و هر آدرس می‌تواند محل سکونت چندین فرد باشد
- با توجه به این که "بیش از یک نفر می‌تواند در یک آدرس خانه زندگی کند"، مدل دوم مناسب‌تر است.

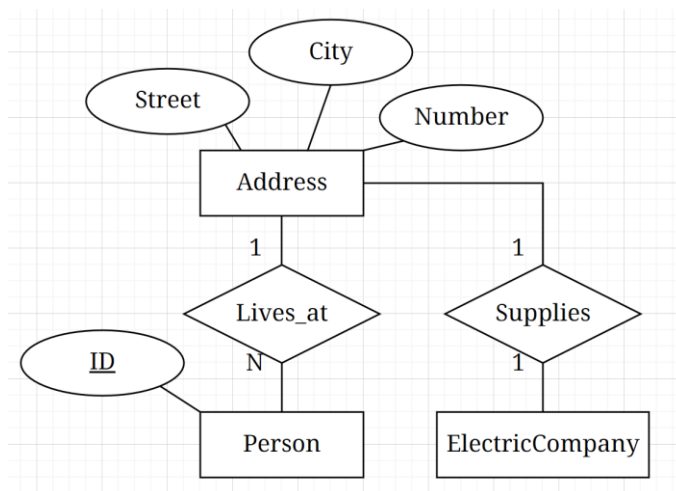
iii. افزودن ElectricCompany به نمودار ER

با توجه به اینکه "فقط یکی از این شرکت‌ها به هر آدرس خانه برق می‌دهد"، رابطه بین آدرس و شرکت برق یک به یک است (یک آدرس فقط از یک شرکت برق سرویس می‌گیرد).

اولی: آدرس به عنوان کلید خارجی برای ElectricCompany استفاده میشود



دومی:



(ب) مقایسه نمودارهای ER ارائه شده

بر اساس نمودارهای ER نشان داده شده در تصویر، به تحلیل و مقایسه روابط می‌پردازم.

تحلیل نمودارهای ER

نمودار سمت چپ

- دو رابطه داریم: $R1$ بین A و B ، و $R2$ بین A ، B و C
- $R1$: رابطه باینری (دوتایی) با نسبت $N:1$ بین A و B
- $R2$: رابطه باینری بین A و B با نسبت $N:1$ که همچنین با C با نسبت $1:1$ مرتبط است
- ویژگی‌ها A : دارای ویژگی‌های $a1$ و $a2$ ، B : دارای ویژگی $b1$ ، C : دارای ویژگی‌های $c1$ (دو بار)، و $R1$ دارای ویژگی $r1$

نمودار سمت راست

- دو رابطه داریم: $R1$ بین A و B ، و $R2$ بین A ، B و C
- $R1$: رابطه باینری (دوتایی) با نسبت $N:1$ بین A و B
- $R2$: رابطه باینری بین A و B با نسبت $N:1$ که همچنین با C با نسبت $M:1$ مرتبط است
- ویژگی‌ها: مشابه نمودار سمت چپ

روابط متناظر (نگاشت به مدل رابطه‌ای)

نمودار سمت چپ

- $a1 // A(a1, a2)$ به عنوان کلید اصلی
- $b1 // B(b1)$ به عنوان کلید اصلی
- $c1 // C(c1, c1)$ به عنوان کلید اصلی
- $a1 // R1(r1, a1, b1)$ و $b1$ به عنوان کلید خارجی
- $R2(a1, b1, c1)$ هر سه به عنوان کلید خارجی و $b1$ کلید اصلی

نمودار سمت راست

- $R2(a1, b1, c1)$ هر سه به عنوان کلید خارجی و $b1$ و $c1$ مجموعاً کلید اصلی

مقایسه دو نمودار

۱. شباهت‌ها:

- هر دو نمودار دارای سه موجودیت A ، B و C هستند
- هر دو نمودار دارای دو رابطه R1 و R2 هستند
- R1 در هر دو نمودار رابطه ۱:N بین A و B است
- ویژگی‌های موجودیت‌ها در هر دو نمودار یکسان هستند

۲. تفاوت‌های کلیدی:

- در نمودار سمت چپ، رابطه R2 بین B و C دارای نسبت ۱:۱ است، اما در نمودار سمت راست این نسبت M:1 است
- این تفاوت باعث می‌شود که در نمودار سمت چپ هر B با حداکثر یک C مرتبط باشد، اما در نمودار سمت راست هر B می‌تواند با چندین C مرتبط باشد
- این تفاوت در قیود جامعیت (integrity constraints) نمایان می‌شود. در نمودار سمت چپ، محدودیت یکتایی (uniqueness constraint) روی b1 در رابطه R2 وجود دارد، اما در نمودار سمت راست این محدودیت وجود ندارد.

سوال ۸

- Flight(flno, from, to, distance, departs, arrives)
- Aircraft(aid, aname, cruisingRange)
- Certified(eid, aid)
- Employees(eid, ename, salary)

آشناسه‌های خلبانانی که مجوز پرواز با هواپیمای Boeing747 را دارند:

$$\pi_{eid}(\sigma_{aname = 'Boeing747'}(Aircraft) \bowtie Certified)$$

ابتدا هواپیماهای Boeing747 را با استفاده از عملگر انتخاب (σ) پیدا می‌کنیم، سپس با جدول Certified پیوند طبیعی (\bowtie) می‌دهیم تا خلبانانی که با این نوع هواپیما مجوز دارند مشخص شوند، و در نهایت با عملگر پروجکشن (π) فقط شناسه‌های (eid) آنها را استخراج می‌کنیم.

ب) نام خلبانانی که مجوز پرواز با هواپیمای Boeing747 را دارند:

$\pi_{ename}(Employees \bowtie (\pi_{eid}(\sigma_{aname = 'Boeing747'}(Aircraft) \bowtie Certified)))$

ابتدا همان کار قسمت (آ) را انجام می‌دهد، سپس نتیجه را با جدول Employees پیوند می‌دهد تا نام‌ها استخراج شوند.

ج) شناسه‌های تمام هواپیماهایی که می‌توانند برای پروازهای بدون توقف از نیویورک به لس‌آنجلس استفاده شوند:

$\pi_{aid}(Aircraft \bowtie _cruisingRange \geq distance (\sigma_{from = 'New York' \wedge to = 'Los Angeles'}(Flight)))$

ابتدا پروازهای از نیویورک به لس‌آنجلس را با عملگر انتخاب پیدا می‌کنیم. سپس این نتیجه را با جدول Aircraft با شرط پیوند $cruisingRange \geq distance$ ترکیب می‌کنیم تا هواپیماهایی که برد پروازی کافی دارند را پیدا کنیم. در نهایت فقط شناسه‌های هواپیما (aid) را نمایش می‌دهیم.

د) نام خلبانانی که می‌توانند هواپیماهای با محدوده پرواز بیش از ۳۰۰۰ کیلومتر را هدایت کنند، اما مجوز پرواز با Boeing747 را ندارند:

$\pi_{ename}(Employees \bowtie (\pi_{eid}(Certified \bowtie \sigma_{cruisingRange > 3000}(Aircraft)) - \pi_{eid}(Certified \bowtie \sigma_{aname = 'Boeing747'}(Aircraft))))$

ابتدا دو مجموعه زیر را محاسبه می‌کنیم:

۱. خلبانانی که مجوز پرواز با هواپیماهای با محدوده پرواز بیش از ۳۰۰۰ کیلومتر را دارند

۲. خلبانانی که مجوز پرواز با Boeing747 را دارند

سپس از عملگر تفاضت (-) استفاده می‌کنیم تا خلبانانی که در مجموعه اول هستند اما در مجموعه دوم نیستند را پیدا کنیم. در آخر با پیوند با جدول Employees، نام این خلبانان را استخراج می‌کنیم.

ه) مجموع مبلغ پرداختی به کارمندان به عنوان حقوق:

$\Sigma_SUM(salary)(Employees)$

از عملگر تجمیع (Σ) با تابع SUM برای محاسبه مجموع ستون salary استفاده می‌کنیم.

و) شناسه کارمندانی که بیشترین حقوق را دریافت می‌کنند:

$\pi_{eid}(\sigma_{salary = MAX_salary}(Employees)(Employees))$

ابتدا باید بیشترین حقوق را پیدا کنیم ($MAX_salary(Employees)$) و سپس کارمندانی که حقوق آنها برابر با این مقدار است را انتخاب کنیم.

ز) یافتن شناسه‌های کارمندانی که دومین بیشترین مقدار حقوق را دریافت می‌کنند:

$$\pi_{eid}(\sigma_{salary = MAX_salary(\sigma_{salary} < MAX_salary(Employees)(Employees))(Employees))$$

ابتدا بیشترین حقوق را پیدا می‌کنیم. سپس کارمندانی که حقوق کمتر از این مقدار دارند را انتخاب می‌کنیم. از بین این کارمندان، بیشترین حقوق (که در واقع دومین بیشترین حقوق کل است) را پیدا می‌کنیم و کارمندانی که این حقوق را دریافت می‌کنند را استخراج می‌کنیم.

ح) بررسی وجود مسیر هوایی بدون توقف یا با یک توقف از سیدنی به نیویورک:

برای پروازهای مستقیم:

$$\sigma_{from = 'Sydney' \wedge to = 'New York'}(Flight)$$

برای پروازهای با یک توقف:

$$\pi_{f1.flno, f1.from, f2.to, f1.to}(\rho_{f1}(Flight) \bowtie_{f1.to = f2.from} \rho_{f2}(\sigma_{from = 'Sydney' \wedge to = 'New York'}(Flight)))$$

از عملگر تغییر نام (ρ) برای ایجاد دو نسخه از جدول Flight استفاده می‌کنیم و آنها را با شرط $f1.to = f2.from$ پیوند می‌دهیم تا پروازهایی که از سیدنی شروع می‌شوند و در نیویورک پایان می‌یابند را با یک توقف در میان پیدا کنیم.

ط) نام و حقوق کارمندانی که مجوز پرواز با بیشترین هواپیما ها را دارند:

$$\pi_{ename, salary}(Employees \bowtie (\pi_{eid}(G_{eid}, COUNT(aid) \rightarrow count(Certified)) \bowtie \sigma_{count = MAX_count(G_{eid}, COUNT(aid)) \rightarrow count(Certified)}(G_{eid}, COUNT(aid) \rightarrow count(Certified))))$$

از عملگر گروه‌بندی (G) استفاده می‌کنیم تا برای هر خلبان، تعداد هواپیماهایی که مجوز پرواز با آنها را دارد محاسبه کنیم. سپس بیشترین تعداد را پیدا می‌کنیم و خلبانانی که دقیقاً این تعداد هواپیما را دارند همراه با نام و حقوقشان برمی‌گردانیم.

ی) شناسه کارمندانی که دقیقاً با ۳ هواپیما مجوز پرواز دارند:

$$\pi_{eid}(G_{eid}, COUNT(aid) \rightarrow count(Certified) \bowtie \sigma_{count = 3}(G_{eid}, COUNT(aid) \rightarrow count(Certified)))$$

از عملگر گروه‌بندی استفاده می‌کنیم تا برای هر خلبان، تعداد هواپیماهای مجوز داده شده را بشماریم. سپس کسانی را که دقیقاً ۳ هواپیما دارند انتخاب می‌کنیم.

سوال ٩

(الف)

١.

$$\{p.PID \mid Player(p) \wedge p.Level > 100 \wedge \forall s (Session(s) \wedge s.PID = p.PID \rightarrow \exists d (d \text{ is a date} \wedge s.StartDate.date = d \wedge s.EndDate.date = d))\}$$

٢.

$$\{p.PID \mid Player(p) \wedge \exists l (Leaderboard(l) \wedge l.PID = p.PID \wedge l.Score \geq 3000 \wedge l.Score \leq 5000) \wedge$$

$$\exists s (Session(s) \wedge s.PID = p.PID \wedge s.Length > 6) \wedge$$

$$\exists r (Player(r) \wedge r.PID = p.PID \wedge r.RegisterDate.year \leq 2019)\}$$

٣.

$$\{p.Username, p.Email \mid Player(p) \wedge$$

$$(\exists s (Session(s) \wedge s.PID = p.PID \wedge (s.EndDate - s.StartDate) > 1 \text{ day}) \wedge$$

$$\exists l (Leaderboard(l) \wedge l.PID = p.PID \wedge l.Year = 2022 \wedge l.Rank \leq 100)) \vee$$

$$(\exists i, q, l (Interaction(i) \wedge i.SenderID = p.PID \wedge i.ReceiverId = q \wedge$$

$$Leaderboard(l) \wedge l.PID = q \wedge l.Year = current_year \wedge l.Rank \leq 100 \wedge l.Score \geq 5000))\}$$

(ب)

١.

$$\{m.MID \mid Matches(m) \wedge ((m.HomeTeamGoals - m.AwayTeamGoals \geq 3) \vee$$

$$(m.AwayTeamGoals - m.HomeTeamGoals \geq 3))\}$$

٢.

$$\{p.PlayerName, p.Age \mid Players(p) \wedge p.Position = 'Goalkeeper' \wedge$$

$$\exists t, l (Teams(t) \wedge t.TID = p.TID \wedge Leagues(l) \wedge l.LID = t.LID \wedge$$

$$l.LeagueName = 'Premier League' \wedge l.Country = 'England')\}$$

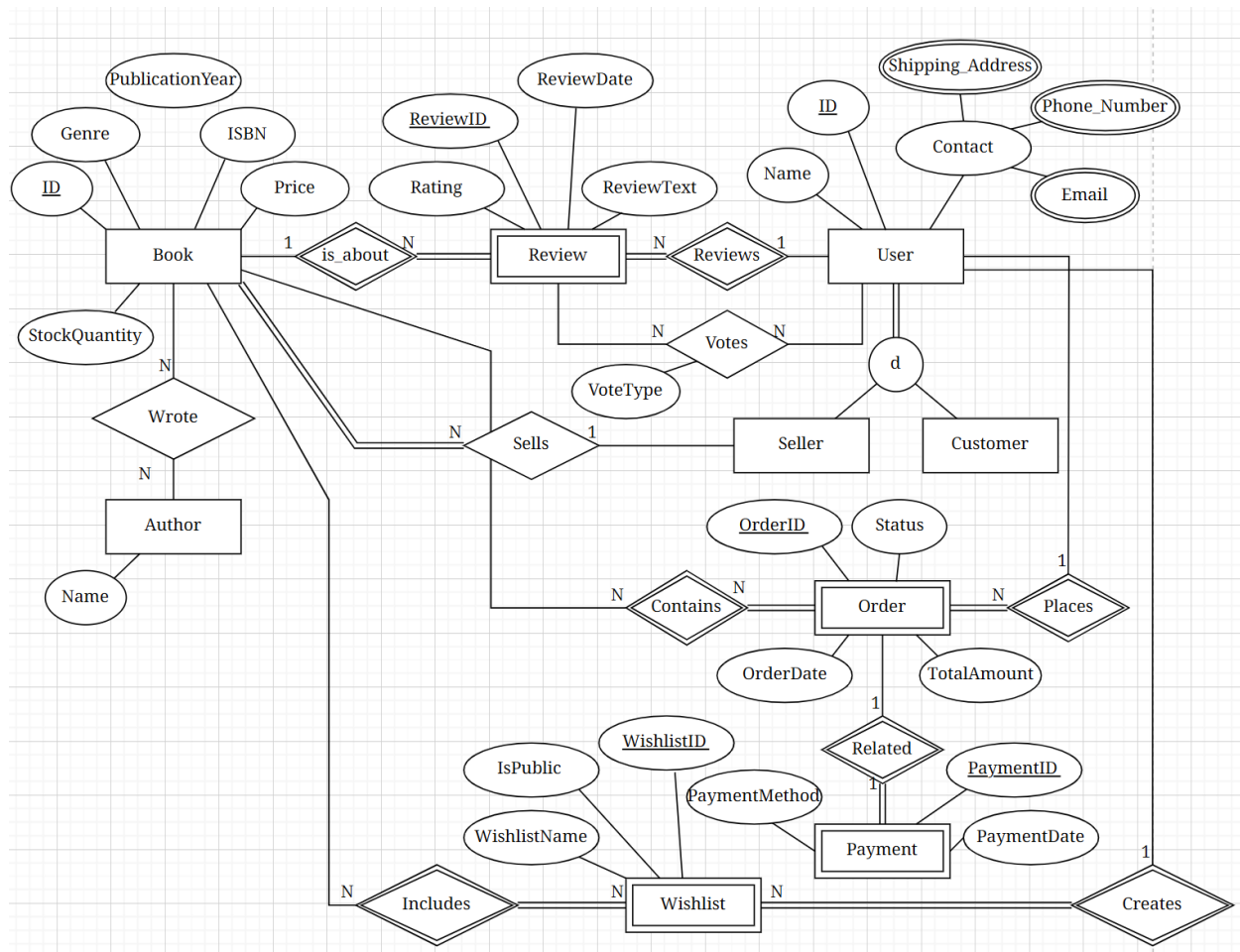
٣.

$$\{t.TeamName \mid Teams(t) \wedge \exists m, l (Matches(m) \wedge m.AwayTeamID = t.TID \wedge$$

$$m.AwayTeamGoals > m.HomeTeamGoals \wedge \\ Leagues(l) \wedge l.LID = m.LID \wedge \\ l.Country = 'Germany' \wedge l.Year = 2023)\}$$

.f

$$\{t.TeamName \mid Teams(t) \wedge \\ (\forall p (Players(p) \wedge p.TID = t.TID \rightarrow (p.Age \geq 20 \wedge p.Age \leq 35))) \wedge \\ \exists m, l (Matches(m) \wedge m.HomeTeamID = t.TID \wedge m.HomeTeamGoals \\ > m.AwayTeamGoals \wedge \\ Leagues(l) \wedge l.LID = m.LID \wedge l.Country = 'Italy' \wedge \\ l.LeagueName = 'Serie A' \wedge (l.Year = 2022 \vee l.Year = 2023))\}$$



فایل این طراحی به نام HW1Q10 در فایل zip قرار دارد

(ب) موجودیت‌ها، روابط و ویژگی‌ها برای نمودار

موجودیت‌ها:

۱. کاربر (User)

ویژگی‌ها: UserID (کلید اصلی)، Name، Email، PhoneNumber،
ShippingAddress، UserType (مشتری/فروشنده)

۲. کتاب (Book)

ویژگی‌ها: ISBN (کلید اصلی)، Title، Price، Genre، StockQuantity،
PublicationYear

۳. نویسنده (Author)

ویژگی‌ها: AuthorID (کلید اصلی)، AuthorName

۴. سفارش (Order)

ویژگی‌ها: OrderID (کلید اصلی)، OrderDate، Status، TotalAmount

۵. پرداخت (Payment)

ویژگی‌ها: PaymentID (کلید اصلی)، PaymentDate، PaymentMethod، Amount

۶. بررسی (Review)

ویژگی‌ها: ReviewID (کلید اصلی)، ReviewDate، ReviewText، Rating

۷. لیست علاقه‌مندی (Wishlist)

ویژگی‌ها: WishlistID (کلید اصلی)، IsPublic، WishlistName (خصوصی/عمومی)

روابط:

۱. نوشتن (Writes)

بین: Author و Book (چند به چند)

توضیح: هر کتاب می‌تواند چندین نویسنده داشته باشد و هر نویسنده می‌تواند چندین کتاب نوشته باشد

۲. ثبت سفارش (Places)

بین: User و Order (یک به چند)

توضیح: هر کاربر می‌تواند چندین سفارش ثبت کند، اما هر سفارش فقط توسط یک کاربر ثبت می‌شود

۳. شامل بودن (Contains)

بین: Order و Book (چند به چند)

۴. مربوط بودن (Related)

بین: Order و Payment (یک به یک)

- توضیح: هر سفارش مربوط به یک پرداخت است، و هر پرداخت مربوط به یک سفارش است

۵. نظردادن (Reviews)

- بین: User و Review (یک به چند)
- توضیح: هر کاربر می تواند چندین نظر بدهد و هر نظر می تواند مربوط به یک نفر باشد

۶. رأی دادن (Votes)

- بین: User و Review (چند به چند)
- ویژگی های رابطه: VoteType (مثبت/منفی)

۷. ایجاد لیست (Creates)

- بین: User و Wishlist (یک به چند)

۸. اضافه کردن به لیست (Includes)

- بین: Book و Wishlist (چند به چند)
- توضیح: هر لیست می تواند شامل چندین کتاب باشد و هر کتاب می تواند در چندین لیست قرار گیرد

۹. فروش (Sells)

- بین: User (فروشنده) و Book (یک به چند)
- توضیح: هر فروشنده می تواند چندین کتاب بفروشد و هر کتاب می تواند توسط یک فروشنده فروخته شود

تبدیل نمودار ER به مدل رابطه ای

1. User (کاربر)

- UserID (PK)
- Name
- Email
- PhoneNumber

- ShippingAddress
- UserType (مشتري/فروشنده)

2. Book (کتاب)

- ISBN (PK)
- Title
- Price
- Genre
- StockQuantity
- PublicationYear
- SellerID (FK to User.UserID) – "با توجه به رابطه فروش"

3. Author (نویسنده)

- AuthorID (PK)
- AuthorName

4. Book_Author ("جدول میانی برای رابطه چند به چند نوشتن")

- ISBN (PK, FK to Book.ISBN)
- AuthorID (PK, FK to Author.AuthorID)

5. Order (سفارش)

- OrderID (PK)
- OrderDate
- Status
- TotalAmount
- UserID (FK to User.UserID) – "با توجه به رابطه ثبت سفارش"

6. Order_Book ("جدول میانی برای رابطه چند به چند شامل بودن")

- OrderID (PK, FK to Order.OrderID)
- ISBN (PK, FK to Book.ISBN)
- Quantity
- UnitPrice

7. Payment (پرداخت)

- PaymentID (PK)
- PaymentMethod
- PaymentDate
- Amount
- OrderID (FK to Order.OrderID, Unique) – "مربوط به توجه به رابطه یک به یک" بودن

8. Review (بررسی)

- ReviewID (PK)
- Rating
- ReviewText
- ReviewDate
- UserID (FK to User.UserID) – "نظردادن به توجه به رابطه"
- ISBN (FK to Book.ISBN) – "برای مشخص کردن کتاب مورد بررسی"

9. Review_Vote ("جدول میانی برای رابطه چند به چند" رأی دادن)

- UserID (PK, FK to User.UserID)
- ReviewID (PK, FK to Review.ReviewID)
- VoteType (مثبت/منفی)

10. Wishlist (لیست علاقه‌مندی)

- WishlistID (PK)

- WishlistName
- IsPublic
- UserID (FK to User.UserID) – "با توجه به رابطه "ایجاد لیست"

11. Wishlist_Book ("جدول میانی برای رابطه چند به چند "اضافه کردن به لیست")

- WishlistID (PK, FK to Wishlist.WishlistID)
- ISBN (PK, FK to Book.ISBN)
- DateAdded

ج) عبارات جبر رابطه‌ای

۱. یافتن تمام کتاب‌هایی که توسط یک نویسنده مشخص نوشته شده‌اند

$$\pi_{Title, ISBN} (Books \bowtie (\sigma_{AuthorName = '[موردنظر نویسنده نام]'} (Authors) \bowtie BookAuthors))$$

۲. بازیابی تمام سفارش‌های ثبت‌شده توسط یک کاربر خاص

$$\pi_{OrderID, OrderDate, Status, TotalAmount} (\sigma_{UserID = '[موردنظر کاربر ID]'} (Orders))$$

۳. یافتن کتاب‌هایی که بالاترین امتیاز را در یک ژانر خاص دارند

$$\begin{aligned} &\pi_{Title, ISBN} (\\ &\sigma_{Rating = MAX_Rating} (\\ &Books \bowtie \sigma_{Genre = '[موردنظر ژانر]'} (Books) \bowtie Reviews \bowtie \\ &(\pi_{Genre, MAX(Rating)} AS MAX_Rating (Books \bowtie Reviews) Group By Genre) \\ &) \\ &) \end{aligned}$$

۴. یافتن کاربرانی که کتاب خریده‌اند اما هیچ بررسی‌ای ثبت نکرده‌اند

$$\begin{aligned} &\pi_{UserID, Name, Email} (\\ &Users \bowtie (\\ &(\pi_{UserID} (Orders \bowtie OrderItems)) - \\ &(\pi_{UserID} (Reviews))) \end{aligned}$$

۵. بازیابی کتاب‌هایی که در لیست علاقه‌مندی یک کاربر هستند اما هنوز خریداری نشده‌اند.

$$\pi_{Title, ISBN} ($$

$$Books \bowtie ($$

$$(\pi_{ISBN} (WishlistItems \bowtie \sigma_{UserID = '[ID \text{ مورد نظر کاربر}]' (Wishlists)))) -$$

$$(\pi_{ISBN} (OrderItems \bowtie Orders \bowtie \sigma_{UserID = '[ID \text{ مورد نظر کاربر}]' (Users))))$$

$$)$$

$$)$$