

به نام خدا



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

# اصول طراحی پایگاه داده

## پیاده سازی سیستم بانکی (فاز 1)

استاد درس:

دکتر پوربهمن

مهلت ارسال پاسخ:

چهارشنبه 1403/09/28 ساعت 23:59

## مقدمه:

هدف این مرحله از پروژه، به‌طور کلی، پیاده‌سازی ساختار کلی سامانه و ایجاد جداول موردنیاز است. این کار نیازمند درک عمیق‌تری از جداول پایگاه داده و روابط میان آن‌ها می‌باشد. برای دستیابی به این هدف، از یک اسکیمای استفاده خواهیم کرد که روابط بین جداول را به‌طور شفاف نمایش می‌دهد و در توسعه پایگاه داده نقش موثری ایفا می‌کند. سپس یک نمودار ER طراحی خواهیم کرد و در ادامه جداول را نرمال‌سازی می‌کنیم.

در نهایت، با اجرای کوئری‌ها می‌توانیم به داده‌های ذخیره‌شده در پایگاه داده دسترسی پیدا کنیم. این دسترسی به ما امکان می‌دهد اطلاعات کامل‌تر و دقیق‌تری را از داخل پایگاه داده استخراج کرده و از آن‌ها در تحلیل‌ها و گزارش‌های متنوع استفاده کنیم. این فرآیند می‌تواند به ما کمک کند تا دید بهتری از وضعیت سامانه و فعالیت‌های کاربران به دست آورده و برنامه‌ها و استراتژی‌های خود را بر اساس این اطلاعات بهبود بخشیم.

## تعریف کلی از پروژه:

در این فاز از پروژه، ابتدا لازم است اسکیمای مربوط به تمامی جداول سامانه را طراحی کنید. در ادامه، ER-Diagram مربوط به سامانه را طراحی و رسم کنید. پس از آن باید جدولی که دارید را تا سطح BCNF نرمال‌سازی کنید، سپس باید جداول بدست آمده را پیاده‌سازی کرده (با کد SQL) و با توجه به ساختار کلی سامانه، داده‌های اولیه (آزمایشی) را در هر جدول وارد (insert) کنید.

توجه داشته باشید که این فاز به‌عنوان ادامه‌ای بر فاز صفر پروژه طراحی شده است. در صورتی که فاز صفر را انجام نداده باشید و صرفاً به اجرای این فاز بپردازید، پروژه شما ناقص تلقی شده و نمره‌ای دریافت نخواهید کرد. بنابراین، در صورتی که فاز صفر را تکمیل نکرده‌اید، ابتدا لازم است آن را به پایان رسانده و ارسال کنید. با این حال، به دلیل تأخیر در تکمیل فاز صفر، نمره‌ای برای آن دریافت نخواهید کرد. اما همچنان می‌توانید با تکمیل و ارسال این فاز، نمره مربوط به آن را کسب کنید.

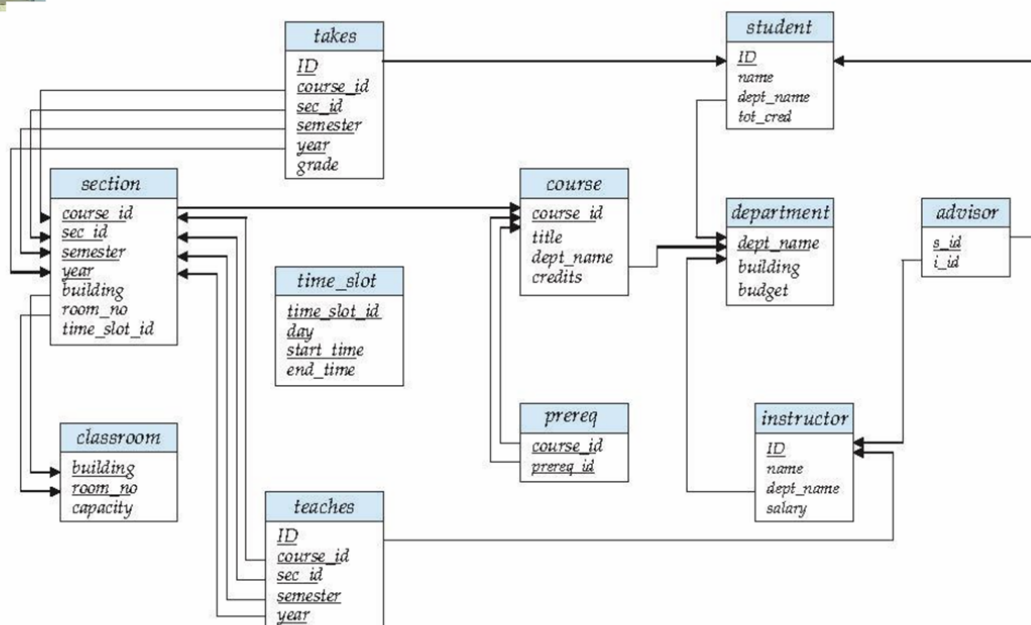
## شرح پروژه:

### قسمت اول

در اولین مرحله از این فاز، باید یک اسکیمای (schema) طراحی کنید. این اسکیمای به نمایش ساختار جداول و روابط میان آنها می‌پردازد و یکی از الزامات اصلی برای طراحی یک پایگاه داده اصولی است. برای مثال یک اسکیمای که در اسلاید درس نیز وجود دارد را می‌بینیم.



### Schema Diagram for University Database



در طراحی اسکیمای باید موارد زیر به دقت انجام شود:

تعریف کلیدهای اصلی (Primary Keys): این کلیدها برای شناسایی یکتای رکوردها در هر جدول استفاده می‌شوند.

تعریف کلیدهای خارجی (Foreign Keys): این کلیدها روابط میان جداول مختلف را مشخص می‌کنند و کاربرد آن در برقراری ارتباطات منطقی بین روابط است.

تعریف سایر خصوصیات (Attributes): باید تمام فیلدها و ویژگی‌های ضروری برای هر جدول را مشخص کنید تا نیازهای سامانه به درستی پوشش داده شود.

در بخش اول این فاز، هدف اصلی طراحی و رسم اسکیماست تا ساختار کلی پایگاه داده را به صورت دقیق و شفاف مشخص کند. این کار نه تنها به درک بهتر داده ها و روابط کمک می کند، بلکه فرآیند پیاده سازی را نیز بهینه تر می سازد.

ما برای سامانه خود نیازمند طراحی یک پایگاه داده هستیم که ویژگی های زیر را به درستی پوشش دهد. طراحی دقیق و اصولی این بخش، نقش کلیدی در ارزیابی و کسب نمره شما دارد؛ بنابراین توجه ویژه ای به آن داشته باشید. لازم به ذکر است که نیازمندی های ارائه شده صرفاً بخشی از ویژگی های مورد نظر هستند و لزوماً به معنای معادل سازی هر نیازمندی با یک جدول در پایگاه داده نیستند. هر یک از این نیازمندی ها می توانند در قالب یک یا چند جدول طراحی شوند، با سایر نیازمندی ها ترکیب شوند یا حتی جداول واسط جدیدی برای ایجاد ارتباط میان جداول ایجاد شوند. در نتیجه، طراحی شما می تواند بسیار متنوع و خلاقانه باشد و به نگرش و درک شما از مسأله بستگی دارد.

#### • اطلاعات اشخاص:

افراد مرتبط با سامانه بانکی شامل دو گروه اصلی کارمندان و مشتریان هستند. اطلاعات مشترک این افراد مانند شماره شناسایی، نام، نام خانوادگی، تاریخ تولد، شماره تلفن، ایمیل و آدرس باید ذخیره شوند.

1. کارمندان بانک: این افراد علاوه بر ویژگی های مشترک، دارای مشخصه های اختصاصی مانند شناسه کارمندی و سمت شغلی هستند که در جدول Employee ذخیره می شوند.
2. مشتریان بانک: مشتریان نیز دارای مشخصه هایی مانند شناسه مشتری و نوع مشتری هستند که در جدول Customer ذخیره می شوند. نوع مشتری می تواند اطلاعاتی مانند حقیقی یا حقوقی بودن فرد را مشخص کند.

#### • حساب کاربری:

هر مشتری می تواند دارای یک یا چند حساب بانکی باشد.

هر حساب بانکی مشخصاتی شامل شماره حساب، نوع حساب، موجودی حساب و وضعیت حساب دارد. وضعیت حساب می تواند فعال، معلق یا بسته شده باشد. اطلاعات دیگری مانند تاریخ باز شدن حساب و تاریخ بسته شدن نیز ذخیره می شوند.

- عملیات بانکی:

هر تراکنش دارای یک شناسه منحصر به فرد، حساب مبدا، حساب مقصد (در صورت نیاز)، مبلغ تراکنش و تاریخ تراکنش است.

تراکنش‌ها می‌توانند نشان‌دهنده برداشت، واریز یا انتقال وجه باشند. این بخش کمک می‌کند تا تاریخچه تمام فعالیت‌های مالی مشتریان به دقت ثبت و قابل پیگیری باشد.

- وام:

در سیستم بانکی، چندین نوع وام ارائه می‌شود که مشتریان می‌توانند برای دریافت آن‌ها درخواست بدهند.

هر وام دارای ویژگی‌هایی مانند شناسه وام، شناسه متقاضی وام، نوع وام، مبلغ وام، نرخ بهره، دوره بازپرداخت، تاریخ شروع، تاریخ پایان و وضعیت وام است.

هر وام می‌تواند شامل چندین قسط باشد. اطلاعاتی مانند تاریخ برنامه‌ریزی شده برای پرداخت قسط، تاریخ پرداخت شده، مبلغ پرداخت شده، سود پرداخت شده و مجموع مبلغ پرداخت شده باید ذخیره شوند. همچنین هر مشتری می‌تواند چند وام از انواع مختلف درخواست کند. توجه داشته باشید که درخواست وام با عملیات بانکی، متفاوت است.

پس از طراحی اسکیمای، در مرحله بعد باید جداول مربوطه را با استفاده از دستورات SQL به صورت دقیق و استاندارد پیاده‌سازی کنید. در این مرحله لازم است تمامی جزئیات تعریف شده در اسکیمای، از جمله نوع داده‌ها (Data Types)، محدودیت‌ها (Constraints) و روابط میان جداول به درستی اعمال شوند.

## قسمت دوم

در این قسمت وظیفه شما رسم نمودار ER است.

لازمه هر پروژه پایگاه داده، طراحی ساختار و روابط داده‌ها است. برای این کار، از یک نمودار با نام Entity-Relationship استفاده می‌شود. در این نمودار، اجزای اصلی سیستم به عنوان Entityها مشخص می‌شوند. ارتباطات بین این Entityها و ویژگی‌های آنها با استفاده از روابط مانند one-to-one، one-to-many و many-to-many نشان داده می‌شوند.

برای مثال، در یک فروشگاه آنلاین، ممکن است Entityهایی مانند محصول، مشتری، سفارش و ... وجود داشته باشند. این Entityها با یکدیگر ارتباطاتی دارند که در نمودار ER به وضوح نشان داده می‌شوند. مثلاً یک مشتری می‌تواند چندین سفارش داشته باشد (روابط one-to-many) یا یک سفارش ممکن است شامل چندین محصول باشد (روابط many-to-many).

پس، در این مرحله، ما نیاز داریم که نمودار ER را طراحی کنیم تا ساختار اصلی داده‌ها و ارتباطات بین آنها را مشخص کنیم. این نمودار به توسعه‌دهندگان و مدیران پروژه کمک می‌کند تا ساختار داده‌ها را به صورت تری و قابل فهم طراحی کنند.

## قسمت سوم

در قسمت های قبل شما به طراحی اسکیمای برای پروژه خود پرداختید. در این مرحله، لازم است که جداول خود را تا سطح BCNF نرمال سازی کنید. این کار باعث می‌شود ساختار پایگاه داده بهینه تر شده و از افزونگی (redundancy) داده‌ها جلوگیری شود.

پس در این قسمت شما باید با استفاده از دانشی که در طول ترم درمورد BCNF و روش‌های نرمال سازی بدست آورده‌اید، تمام جدول های خود را تا حد BCNF نرمال سازی کنید و فرایندی که در هر مرحله انجام می‌دهید را به خوبی توضیح دهید و بگویید چه تفاوت‌هایی با طراحی اولیه (قسمت‌های قبل) شما داشته است و تفاوت ها را ذکر کنید. توجه داشته باشید که ممکن است طراحی اولیه شما از ابتدا در سطح BCNF بوده و نیازی به تغییر نداشته باشد.

## قسمت چهارم

در این بخش، پس از طراحی اسکیمای کلی و ساختار پایگاه داده، باید جداول مربوطه را با استفاده از دستورات SQL به صورت دقیق و استاندارد پیاده‌سازی کنید. در این مرحله، ضروری است که تمامی جزئیات تعریف‌شده در اسکیمای (پس از نرمال‌سازی)، از جمله انواع داده‌ها (Data Types)، محدودیت‌ها (Constraints) و روابط میان جداول، به طور کامل و صحیح اعمال شوند. سپس، در هر جدول تعدادی رکورد نمونه به صورت دلخواه وارد کنید. این رکوردها برای آزمایش عملکرد جداول و اطمینان از صحت پیاده‌سازی پایگاه داده استفاده می‌شوند. همچنین، می‌توانید از ابزارها یا وبسایت‌هایی که داده‌های تصادفی (Random Data) تولید می‌کنند برای تست کردن و تکمیل فرآیند استفاده کنید.

```
1 CREATE TABLE Customers (  
2     CustomerID INT PRIMARY KEY AUTO_INCREMENT,  
3     FirstName VARCHAR(50) NOT NULL,  
4     LastName VARCHAR(50) NOT NULL,  
5     PhoneNumber VARCHAR(15) UNIQUE,  
6     Email VARCHAR(100) UNIQUE,  
7     Address TEXT,  
8     DateOfBirth DATE,  
9     CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
10 );
```

1- نمونه از ساخت جدول

```

1 INSERT INTO Customers (FirstName, LastName, PhoneNumber, Email, Address, DateOfBirth)
2 VALUES
3 ('Ali', 'Ahmadi', '09123456789', 'ali.ahmadi@gmail.com', 'Tehran, Iran', '1990-05-14'),
4 ('Sara', 'Rezai', '09121234567', 'sara.rezai@yahoo.com', 'Shiraz, Iran', '1995-11-23'),
5 ('Reza', 'Karimi', '09351231234', 'reza.karimi@hotmail.com', 'Esfahan, Iran', '1988-07-08');
6

```

2- نمونه‌ای از داده‌های تصادفی وارد شده در جدول

```

1 SELECT CustomerID, FirstName, LastName, Email, PhoneNumber
2 FROM Customers
3 WHERE DateOfBirth > '1990-01-01';
4

```

3- نمایش خروجی دستورات SQL (یک SELECT ساده)

```

1 CREATE TABLE BankAccounts (
2     AccountID INT PRIMARY KEY AUTO_INCREMENT,
3     CustomerID INT NOT NULL,
4     AccountNumber VARCHAR(20) NOT NULL UNIQUE,
5     Balance DECIMAL(15, 2) DEFAULT 0 CHECK (Balance >= 0),
6     AccountStatus ENUM('Active', 'Suspended', 'Closed') DEFAULT 'Active',
7     CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
8     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE
9 );

```

4- مثالی از محدودیت‌ها (Constraints)



## نکات مربوط به تحویل:

- این پروژه در انتها، تحویل به صورت مجازی در google meet یا skype خواهد داشت و عدم تحویل پروژه به منظور از دست دادن تمام نمره‌ی آن خواهد بود.
- تمرین شما تحویل آنلاین خواهد داشت؛ بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح دادن عملکرد آن‌ها نیستید، بپرهیزید.
- ابهامات خود را با تدریس‌یاران درس مطرح کنید تا آن‌ها در سریع‌ترین زمان ممکن به شما پاسخ دهند. همین‌طور می‌توانید مشکلات خود را با طراحان پروژه با آیدی @HolyBardia، @roza\_gp و @ZeinabKarakani مطرح کنید.
- تاکید می‌شود که به هیچ وجه نباید از ORM ها استفاده کنید.

## مواردی که باید ارسال شوند:

- یک تصویر از اسکیمای طراحی شما از پایگاه داده
- نمودار ER
- توضیحات نرمال‌سازی و نمودار نهایی (در صورت نیاز)
- فایل‌های SQL شما که جداول را در آن ساخته‌اید
- یک فایل زیپ با نام studentID\_PRJ\_P1.zip که شامل تمام موارد بالا و گزارش شماست.