فاز دوم پروژه پایگاه داده

استاد درس: دکتر شهریاری

بهار 1405 – دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

نکاتی در مورد این تمرین نیاز به توجه و دقت دوستان دارد.

1 -هرگونه کپی کردن باعث عدم تعلق نمره به تمامی افراد مشارکت کننده در آن میشود.

2 -تمامی فایلهای خود (یک فایل pdf و فایل‌های sql) را به صورت یک فایل آرشیو zip که با قالب زیر نام گذاری شده است؛ آپلود بفرمایید.

phase2_Name_LastName_StudentNumber.zip

پاسخ به هر کدام از فازها را به صورت فایلی با فرمت و اسم مطرح شده در قسمت ارائه‌ی فاز در zip نهایی خود قرار دهید و آپلود کنید.
برای برخی از فازها گزارش لازم است. تمام گزارش‌ها را در قالب یک فایل pdf به نام زیر آپلود کنید و نام header هر فاز را به شکل خواسته شده قرار دهید.
phase2_Name_LastName_StudentNumber.pdf

3 -در صورت عدم رعایت موارد فوق تصحیح انجام نخواهد شد.

مقدمه

این پروژه به 10 گام تقسیم شده است که در گام صفر لازم است دانشجویان نمودار ER برای سامانه مدیریت سفارش و ارسال آنلاین طراحی نمایند و در گامهای بعدی طراحی آن کامل‌تر شده و پیاده‌سازی می‌شود.

نصب نرم‌افزارهای مورد نیاز

در مرحله اول لازم است که DBMS بر روی سیستم خود نصب کنید. برای این پروژه استفاده از Postgresql اجباری می‌باشد.

پس از نصب Postgresql باید یک database جدید ایجاد کنید که در ادامه پروژه تمام جدول‌ها، اطلاعات و query ها در آن ذخیره و اجرا خواهند شد.

در صورت بروز مشکل در نصب یا تنظیمات، می‌توانید از مستندات رسمی Postgresql ،آموزش‌های آنلاین یا تدریس یاران کمک بگیرید.

نوشتن کد SQL

در اکثر گامهای این پروژه لازم است که کدهای SQL را داخل فایل‌هایی با فرمت sql ذخیره کنید.

در این پروژه شما نباید تنها روی محیط خط فرمان (REPL) تکیه کنید، بلکه باید فایل SQL ایجاد کنید، query ها و دستورات را داخل آن بنویسید، و سپس آن فایل را توسط DBMS اجرا کنید. این روش بهتر این است که کد را در پنجره اجرای SQL داخل DBMS کپی کرده و اجرا کنید. زیرا کدها قابل ذخیره و مستندسازی هستند، می‌توانید در آینده آنها را مجدد اجرا یا اصلاح کنید و روند کار شما قابل بررسی خواهد بود.

گام صفر - طراحی ER Diagram

در این گام ابتدا توصیفی از نرم افزار ارائه میشود که شما باید برای آن یک Diagram-ER طراحی کنید.

این سامانه یک سیستم مدیریت سفارش و ارسال آنلاین برای مشتریان، فروشندگان و تیم عملیات فراهم می کند. مشتریان می توانند ثبت نام کنند، آدرس های مختلف داشته باشند، محصولات را جست وجو کنند، سفارش ثبت کنند و پرداخت انجام دهند. فروشندگان می توانند کالاها، قیمت و موجودی را مدیریت کنند. واحد انبار و ارسال نیز باید وضعیت بسته بندی، تحویل به پیک و تحویل نهایی را کنترل کند.

ویژگی‌های سامانه:

- **مشتریان:** هر مشتری اطلاعات شخصی شامل نام، نام خانوادگی، شماره تماس، ایمیل، تاریخ ثبت نام، وضعیت حساب کاربری و چند آدرس ارسال دارد .
- **فروشندگان:** هر فروشنده شامل نام تجاری، کد اقتصادی یا شناسه ثبت، شماره تماس، ایمیل، وضعیت تأیید و تاریخ عضویت است.
- **محصولات:** هر محصول شامل نام، توضیحات، قیمت، وضعیت فعال بودن، فروشنده، دسته بندی و مشخصات عمومی است .
- **دسته بندی ها:** محصولات در دسته بندی های مختلف قرار می گیرند و هر دسته بندی می تواند زیر دسته داشته باشد .
- **موجودی انبار:** برای هر محصول، تعداد موجود، حد هشدار، محل نگهداری و تاریخ آخرین به روزرسانی نگهداری می شود.
- **سفارش ها:** هر سفارش به یک مشتری و یک آدرس ارسال متصل است و شامل وضعیت سفارش، تاریخ ثبت، مبلغ کل، هزینه ارسال و تخفیف است.
- **آیتم های سفارش:** هر سفارش از چند آیتم تشکیل می شود و هر آیتم شامل محصول، تعداد، قیمت واحد در زمان خرید و مبلغ نهایی است.
- **پرداخت ها:** پرداخت هر سفارش ثبت می شود و شامل مبلغ، روش پرداخت، وضعیت پرداخت، زمان پرداخت و کد رهگیری است.

- **ارسال ها:** برای هر سفارش اطلاعات ارسال شامل شرکت حمل، کد رهگیری، زمان ارسال، زمان تحویل و وضعیت ارسال ذخیره می شود.
- **تخفیف ها:** کدهای تخفیف می توانند درصدی یا مبلغی باشند و محدودیت تاریخ، سقف استفاده و حداقل مبلغ سفارش داشته باشند.

روابط اصلی بین موجودیت‌ها:

- هر مشتری می تواند چند آدرس و چند سفارش داشته باشد.
- هر فروشنده می تواند چندین محصول تعریف کند.
- هر محصول متعلق به یک فروشنده و حداقل یک دسته بندی اصلی است.
- هر سفارش متعلق به یک مشتری است و می تواند شامل چند آیتم سفارش باشد.
- هر آیتم سفارش دقیقا به یک محصول و یک سفارش مرتبط است.
- هر سفارش می تواند یک یا چند رکورد پرداخت داشته باشد، اما فقط پرداخت موفق آن برای نهایی شدن سفارش معتبر است.
- هر سفارش حداکثر یک رکورد ارسال فعال دارد.
- هر محصول یک رکورد موجودی اصلی دارد و تغییرات مهم موجودی باید قابل رهگیری باشند

نکات مهم:

1. قبل از شروع طراحی، کل این سند را کامل مطالعه کنید. درک نادرست از نیازمندی‌ها باعث بازطراحی، اصلاح‌های سنگین و اتلاف زمان خواهد شد.
هدف این گام این است که شما تصویر کامل از سیستم داشته باشید.

2. فیلدهای ذکر شده برای موجودیت‌ها فقط اطلاعات خام اولیه هستند. شما **باید** فیلدهای عملیاتی و فنی لازم برای تبدیل موجودیت‌ها به جدول‌های واقعی دیتابیس را اضافه کنید.

3. برای موجودیت های عملیاتی مانند سفارش، پرداخت و ارسال، وضعیت ها را به شکل قابل کنترل طراحی کنید و برای آنها محدودیت مناسب در نظر بگیرید.

تحویل گام صفر

همانگونه که در ابتدا نیز ذکر شده برای این گام صرفا یک Diagram ER باید تحویل دهید که این ER Diagram را میتوانید به صورت دستی (خوانا و بدون خط خوردگی) یا با استفاده از نرم افزار هایی همچون creately یا draw.io طراحی کنید و بعد به صورت عکس در zip نهایی خود با اسم phase0-ER قرار دهید.

گام اول – نرمال‌سازی پایگاه داده تا BCNF

پس از طراحی Diagram ER سامانه مدیریت سفارش و ارسال آنلاین، پایگاه داده را تا سطح BCNF نرمال سازی کنید.

هدف این گام این است که پایگاه داده بهینه و استاندارد باشد؛ یعنی جداول به گونه ای طراحی شوند که از افزونگی داده جلوگیری شود و اطلاعات تکراری به حداقل برسد. همچنین تحلیل و گزارش گیری آسان شود؛ یعنی تیم تحلیل داده بتواند بدون پیچیدگی به اطلاعاتی مثل وضعیت سفارش ها، فروش هر فروشنده، محصولات کم موجودی و پرداخت های ناموفق دسترسی داشته باشد و داده ها را به راحتی پردازش کند.

صحت داده ها نیز باید حفظ شود. برای مثال نباید آدرس ارسال بدون مشتری، آیتم سفارش بدون سفارش، یا پرداخت بدون سفارش معتبر وجود داشته باشد.

در این گام لازم است حداقل موارد زیر را بررسی کنید:

- جداسازی اطلاعات مشتری از آدرس های ارسال.
- جداسازی اطلاعات محصول از موجودی انبار.
- جداسازی سفارش از آیتم های سفارش.
- جداسازی پرداخت ها و ارسال ها از جدول سفارش.
- بررسی وابستگی های تابعی برای کد تخفیف، دسته بندی، فروشنده و محصول

تحویل گام اول

در این قسمت شما باید با استفاده از دانشی که در طول ترم در مورد BCNF و روشهای

normalization بدست آورده اید، تمام جدول های خود را تا حد BCNF نرمالایز کرده و گزارشی از روند normalization خود تهیه کنید و با عنوان phase1-normalization در فایل گزارش خود قرار دهید.

نکته مهم: بعد از نرمال‌سازی، لازم است که تمام وابستگی‌های تابعی (Functional Dependencies) که در طول فرآیند شناسایی شده‌اند، نیز ارسال شوند.

گام دوم – پیاده‌سازی جداول پایگاه داده در سامانه مدیریت سفارش و ارسال آنلاین

در گامهای قبلی شما پایگاه داده سامانه مدیریت سفارش و ارسال آنلاین را طراحی کرده و Diagram ER اولیه را ایجاد کردید.
در این مرحله باید ER Diagram خود را به جداول رابطه‌ای تبدیل کنید و آن‌ها را در محیط پایگاه داده بسازید.

در این گام لازم است:

- موجودیت‌های طراحی شده در ER را به جداول تبدیل کنید.
- کلیدهای اصلی (Primary Keys) هر جدول را تعیین و تعریف کنید.
- کلیدهای خارجی (Foreign Keys) را براساس روابط بین موجودیت‌ها مشخص کرده و در SQL پیاده‌سازی کنید.
- انواع داده هر ستون را مناسب با اطلاعات موجود انتخاب کنید (مثل INT، VARCHAR، DATE و ...).
- محدودیت‌های لازم را اعمال کنید (NOT NULL، UNIQUE، CHECK و ...).
- برای ستون های وضعیت سفارش، پرداخت و ارسال، مقادیر مجاز را با CHECK یا روش مناسب دیگر کنترل کنید.
- کلیدهای خارجی به گونه‌ای تعریف شوند که صحت داده تضمین شود (در صورت نیاز با ON DELETE / ON UPDATE).

پیشنهاد می شود حداقل جدول های زیر در طراحی شما وجود داشته باشند: مشتری، آدرس مشتری، فروشنده، دسته بندی، محصول، موجودی، سفارش، آیتم سفارش، پرداخت، ارسال و تخفیف. با توجه به طراحی خود می توانید جدول های دیگری نیز اضافه کنید.

تحویل گام دوم

در این مرحله باید تمام کوئری‌های مربوط به ایجاد جدول‌های پایگاه داده (CREATE TABLE) را در یک فایل SQL قرار دهید. کلیدهای اصلی و خارجی باید در داخل همین کوئری‌ها تعریف شده باشند.
فایل خروجی با نام phase2-create.sql در فایل zip نهایی پروژه قرار داده شود.

گام سوم – درج داده‌ها و تست پایگاه داده

در گامهای قبل جداول سامانه مدیریت سفارش و ارسال آنلاین طراحی و ساخته شدند.
در این مرحله، هدف شما پر کردن جداول با داده‌های نمونه و اجرای چند پرس‌وجوی تستی است تا مطمئن شوید ساختار پایگاه داده به‌درستی کار می‌کند.

درج داده در تمام جداول

تمام جداول شما باید حداقل ۱۰ ردیف داده داشته باشند. هدف ایجاد یک پایگاه داده است که بتوان روی آن عملیات عادی ثبت سفارش، پرداخت، کاهش موجودی و ارسال را تست کرد. داده‌ها می‌توانند توسط خودتان وارد شوند یا با استفاده از Fake Data Generators مثل:

Mockaroo

GenerateData

Online Random CSV tools

Power Query

یا هر ابزار مشابه دیگر

مهم است که داده ها منطقی باشند؛ مثلا تاریخ ثبت سفارش در آینده نباشد، موجودی محصول منفی نباشد، مبلغ پرداخت با مبلغ سفارش سازگار باشد و سفارش ارسال شده حتما آدرس معتبر داشته باشد.

پرس‌وجوهای تست

- نمایش اطلاعات تمام سفارش های متعلق به یک مشتری مشخص.
- نمایش ۵ سفارش اخیر یک مشتری بر اساس تاریخ ثبت سفارش.
- نمایش محصولاتی که موجودی آنها کمتر از حد هشدار تعریف شده است.
- نمایش فروشندگانی که در ماه اخیر بیش از یک مقدار مشخص فروش داشته اند.
- نمایش سفارش هایی که پرداخت ناموفق یا در انتظار پرداخت دارند.
- نمایش تعداد سفارش های انجام شده در یک بازه زمانی مشخص (مثلا یک هفته گذشته یا یک ماه گذشته)

تحویل گام سوم

برای این مرحله باید تمام دستورهای INSERT خود برای درج داده‌ها را در یک فایل SQL قرار دهید و پرس‌وجوهای تست را نیز در انتهای همان فایل SQL قرار دهید. فایل خروجی با نام phase3-insert.sql در فایل zip نهایی پروژه قرار داده شود.

گام چهارم - طراحی و پیاده سازی index

در این گام قرار است کیفیت عملکرد دیتابیس را با طراحی ایندکس‌های مناسب بهبود دهید. اگرچه ایندکس‌ها ابزار قدرتمندی برای افزایش سرعت جست‌وجو و واکشی داده هستند، اما استفاده‌ی غلط از آن‌ها باعث افت کارایی، افزایش سایز دیتابیس، و هزینه‌های اضافه در عملیات درج/به‌روزرسانی می‌شود. بنابراین این مرحله قرار نیست فقط حدسی باشد، بلکه باید تحلیلی و مبتنی بر تست عملکرد واقعی باشد.

در این گام هدف شما این است که بر اساس Query هایی که تست کرده‌اید مشخص کنید کدام ستون‌ها نیاز به ایندکس دارند.

از ایجاد ایندکس صرفاً روی primary key و foreign key و unique key constraints پرهیز کنید — آن‌ها به‌صورت پیش‌فرض ایندکس می‌شوند و هدف این گام تحلیل فراتر از موارد بدیهی است.

برای اینکه ایندکس های خوبی بسازید، سوال های زیر را باید تحلیل کنید:

- کدام ستون‌ها بیشترین تکرار در شرط‌های WHERE را دارند؟
- کدام ستون‌ها برای فیلتر، جست‌وجو یا sort استفاده می‌شوند؟
- کدام Queryها روی داده‌های زیاد اجرا می‌شوند و کند هستند؟

- یا ایندکس ترکیبی روی ستون هایی مثل شناسه مشتری و تاریخ سفارش، یا شناسه فروشنده و تاریخ ثبت سفارش، بهتر از چند ایندکس جداگانه عمل می کند؟
- آیا آن ستون تنوع داده‌ای مناسب دارد؟

(ستونی با مقدار تکراری زیاد، مثلا جنسیت، ایندکس خوبی نیست)

نکته مهم: ایندکس باید بر اساس کاربرد باشد، نه صرفاً تعداد ردیف‌های جدول.

تحویل گام چهارم

در این مرحله باید گزارش لیست ستون‌هایی که ایندکس شده‌اند و دلیل دقیق انتخاب آن‌ها را به گزارش خود با عنوان phase4-index اضافه کنید. فایل خروجی با نام phase4-index.sql شامل تمام دستورات ایجاد ایندکس نیز در فایل zip نهایی پروژه قرار داده شود.

حداقل ۳ ایندکس باید طراحی و تحلیل شوند.

گام پنجم - طراحی و پیاده سازی view ها

قرار است به یکی از تیم های backend برای اضافه کردن چند feature جدید به اپلیکیشن مدیریت سفارش، به داده دسترسی بدهیم. از آنجایی که نمی خواهیم در مراحل develop به تمام اطلاعات مالی و عملیاتی دسترسی داشته باشند، می خواهیم view های زیر را طراحی کنیم تا تنها با استفاده از این موارد feature های جدید را طراحی کنند. دقت کنید که این تیم فقط به داده های ویوهای زیر دسترسی دارد و باید بتواند تمام اطلاعات لازم مشتری، سفارش، فروشنده و وضعیت ارسال را در اپلیکیشن نشان دهد؛ پس تشخیص ستون های لازم به عهده ی شماست. می توانید هوشمندانه عمل کنید و برای کم کردن ستون های هر view از view چهارم استفاده کنید.

۱- خلاصه ی سفارش های هر مشتری

نام این view را vw_summary_orders_customer بگذارید. مجموع مبلغ سفارش ها، تعداد سفارش ها و آخرین تاریخ سفارش مهم هستند.

۲- فروش و وضعیت فروشندگان

نام این view را vw_status_sales_seller بگذارید. تعداد سفارش های موفق، مجموع فروش و تعداد محصولات فعال فروشنده مهم هستند.

۳- سفارش های ماه اخیر هر مشتری
نام این view را vw_recent_orders_30days بگذارید. دقت کنید که در اینجا ریز سفارش ها و آیتم های سفارش مهم هستند نه فقط مجموع آنها.

۴- طراحی این view به انتخاب شماست. از این view برای ستون‌های پر تکرار استفاده کنید. پیشنهاد می شود نام آن را vw_alerts_inventory بگذارید و کالاهای کم موجودی را نمایش دهید.

فایل خروجی با نام phase5-view.sql در فایل zip نهایی پروژه قرار داده شود. شمایی از viewها را با هدر phase5-view در فایل pdf قرار دهید.

گام ششم - طراحی و پیاده سازی procedure ها

در این گام می خواهیم چند Procedure Stored طراحی و پیاده سازی کنیم تا عملیات متداول سامانه سفارش و ارسال به صورت ایمن و قابل استفاده ی مجدد باشد.
برای هر procedure چند خطای ممکن را در نظر بگیرید و رفتار مربوطه (مانند RAISE EXCEPTION) را برای آن در نظر بگیرید. نام هر یک را دقیقا چیزی که تعریف شده قرار دهید. procedure ها را به گونه‌ای تعریف کنید که در گام بعدی از آن‌ها به راحتی استفاده کنید.

۱- ثبت سفارش جدید - pr_place_order
انجام عملیات ثبت سفارش برای یک مشتری به همراه ایجاد رکورد سفارش و آیتم های سفارش، محاسبه مبلغ نهایی، اعمال کد تخفیف معتبر در صورت وجود و کاهش موجودی محصولات. در این procedure باید بررسی شود که مشتری معتبر و فعال باشد، محصولات فعال باشند و موجودی کافی وجود داشته باشد

۲- ثبت پرداخت سفارش - pr_register_payment
انجام عملیات پرداخت برای یک سفارش مشخص، ثبت اطلاعات پرداخت، بررسی مبلغ پرداخت، کنترل وضعیت سفارش و تغییر وضعیت سفارش در صورت موفق بودن پرداخت

۳- لغو سفارش : pr_cancel_order

لغو سفارش در حال انتظار یا در انتظار پرداخت، ثبت علت لغو، برگرداندن موجودی محصولات و تغییر وضعیت سفارش. سفارش ارسال شده یا تحویل شده نباید قابل لغو باشد

تحویل گام ششم

فایل خروجی با نام phase6-procedures.sql در فایل zip نهایی پروژه قرار داده شود و قبل از هر procedure نام آن را در کامنت بنویسید. توضیحی خلاصه از منطق و مراحل لازم هر procedure را با هدر phase6-procedure در فایل pdf قرار دهید. دقت کنید که نوشتن مراحل هر procedure در این گام مهم است چون در گام بعدی باید از این مراحل استفاده کنید.

گام هفتم - طراحی و پیاده سازی transaction

در این گام می خواهیم روی همان سامانه مدیریت سفارش و ارسال آنلاین، چند تراکنش طراحی کنیم که با به) order_place_pr، payment_register_pr، order_cancel_pr) ششم گام های procedure از استفاده صورت اتمیک اجرا شوند؛ یعنی اگر هر کدام از مراحل داخلی آنها با مشکلی روبه رو شد، کل تغییرات به وضعیت قبل از شروع تراکنش برگردد. نام هر transaction را در قالب کامنت قبل از آن بنویسید.

تراکنش‌ها را به صورت زیر تعریف کنید:

1- تراکنش ثبت سفارش جدید - tr_place_order

transaction ای بنویسید که برای یک مشتری، با استفاده از pr_place_order یک سفارش جدید ثبت کند. اگر موجودی کافی نباشد یا محصول نامعتبر باشد، تمام تغییرات باید به وضعیت قبل از اجرای تراکنش بازگردانی شود

۲- تراکنش پرداخت سفارش - tr_payment_order

transaction ای بنویسید که با استفاده از pr_register_payment پرداخت یک سفارش را ثبت کند. اگر مبلغ پرداخت یا وضعیت سفارش نامعتبر باشد، وضعیت تمام جداول باید به حالت قبل از شروع

تراکنش برگردد

۳- تراکنش لغو سفارش - tr_cancel_ord

transaction ای بنویسید که با استفاده از pr_cancel_order یک سفارش مجاز را لغو کند و موجودی کالاها را برگرداند. اگر سفارش قبلا ارسال یا تحویل شده باشد، کل تراکنش باید لغو گردد و تمامی داده ها به وضعیت قبل از اجرا بازگردد

تحویل گام هفتم

فایل خروجی را با نام phase7-transaction.sql در فایل zip نهایی پروژه قرار دهید و قبل از هر transaction نام آن را در کامنت بنویسید. دقت کنید که باید برای هر transaction باید ۳ حالت بنویسید که شامل یک حالت موفق و ۲ حالت ناموفق باشد. برای هر یک اسکرین شات خروجی‌ها را با هدر phase7-transaction در فایل pdf قرار دهید.

گام هشتم - پیاده سازی Roles و Security

در این گام هدف این است که با استفاده از Role ها و مجوزهای دسترسی (Privileges (سطح دسترسی کاربران مختلف سامانه مدیریت سفارش و ارسال آنلاین را کنترل کنید. این گام باید نشان دهد که هر کاربر فقط به داده ها و عملیات متناسب با نقش خود دسترسی دارد و از دسترسی های غیرمجاز جلوگیری می شود. برای هر نقش یک role و کاربر نمونه بسازید و دسترسی های لازم را به او بدهید

1)نقش مدیر سیستم فروشگاه role_shop_admin

این نقش بالاترین سطح دسترسی را دارد و معمولا برای مدیر سیستم یا مدیر ارشد سامانه در نظر گرفته می شود. نیاز است که دسترسی کامل به همه ی جداول اصلی سامانه را داشته باشد.

2) نقش فروشنده role_seller

این نقش برای فروشندگانی است که محصولات خود را مدیریت می کنند. دسترسی ها باید محدود به مشاهده و ویرایش اطلاعات محصولات، قیمت و موجودی مرتبط با فروشنده باشد. فروشنده نباید به اطلاعات کامل پرداخت مشتریان دسترسی داشته باشد.

3) نقش پشتیبانی سفارش role_support

این نقش برای کارمندانی است که وضعیت سفارش ها، پرداخت های ناموفق و شکایات مشتریان را بررسی می کنند. دسترسی مشاهده روی مشتری، سفارش، آیتم سفارش، پرداخت و ارسال لازم است، اما ویرایش باید محدود باشد.

4)نقش عملیات انبار role_warehouse

این نقش برای افرادی است که بسته بندی، کاهش یا افزایش موجودی و تحویل سفارش به شرکت حمل را انجام می دهند. دسترسی مشاهده سفارش و ویرایش محدود روی موجودی و ارسال لازم است.

وظایف و دسترسی ها را با دقت تعریف کنید. به عنوان مثال، نقش های غیرمدیر نباید بتوانند جدول های پایه را حذف کنند یا ساختار دیتابیس را تغییر دهند. همچنین بهتر است دسترسی تیم backend به اطلاعات حساس از طریق view های گام پنجم انجام شود

تحویل گام هشتم

فایل خروجی را با نام sql.phase8-roles در فایل zip نهایی پروژه قرار دهید. این فایل باید شامل ساخت role ها، ساخت کاربران نمونه، دستورهای GRANT و در صورت نیاز REVOKE باشد.

گام نهم (امتیازی) - پیاده سازی سیستم Audit و Log

در این گام امتیازی، هدف این است که برای سامانه مدیریت سفارش و ارسال آنلاین، یک سیستم ثبت رخداد Audit / Log طراحی کنید تا تغییرات مهم داده ها و عملیات حساس در پایگاه داده ثبت و قابل رهگیری باشد. این سیستم باید کمک کند که بتوانیم بعدا بررسی کنیم چه کسی، چه تغییری، در چه زمانی و روی کدام رکورد انجام داده است.

۱) طراحی جداول Audit / Log

برای ثبت رخدادها، حداقل یک جدول مخصوص Audit طراحی و ایجاد کنید. به عنوان مثال: جدول audit_log برای ثبت رخدادهای کلی مثل تغییر روی مشتری، محصول، سفارش، موجودی، پرداخت و غیره. پیشنهاد می شود در جدول audit_log ستون های زیر یا معادل آنها وجود داشته باشد:

- شناسه رخداد (کلید اصلی)
- نام جدول (مثلاً: مشتری، محصول، سفارش، پرداخت و ...)
- نوع عملیات (DELETE، UPDATE، INSERT)
- شناسه رکورد تحت تأثیر (مثلاً شناسه سفارش یا شناسه محصول)
- مقدار قدیمی داده‌ها (در صورت نیاز، مثلاً به صورت JSON یا متن)
- مقدار جدید داده‌ها (در صورت نیاز)
- نام کاربر پایگاه داده‌ای که تغییر را انجام داده (با استفاده از CURRENT_USER)
- تاریخ و زمان انجام عملیات (TIMESTAMP)

در صورت تمایل می توانید جداول جداگانه برای انواع رخدادهای خاص مثل تغییرات موجودی، تغییرات قیمت محصول یا تغییر وضعیت سفارش تعریف کنید، اما داشتن یک جدول کلی audit_log کافی است.

۲) تعریف Trigger برای ثبت رخدادها

برای حداقل دو جدول اصلی سامانه باید Trigger هایی تعریف کنید که در زمان انجام عملیات زیر اجرا شوند:

- بعد از INSERT
- بعد از UPDATE
- بعد از DELETE

و هر بار که این عملیات انجام شد، یک رکورد مناسب در جدول audit_log ثبت شود.

Trigger باید به یک تابع (Function) متصل باشد که عمل ثبت اطلاعات را در جدول audit_log انجام می‌دهد.

۳) ثبت اطلاعات کاربر و زمان انجام عملیات

در پیاده‌سازی Trigger ها و Function های مربوط به Audit، لازم است حداقل موارد زیر ثبت شوند:

- کاربری که عملیات را انجام داده
- تاریخ و زمان دقیق انجام عملیات
- نام جدول و نوع عملیات

- مقدار قدیمی و جدید برای تغییرات حساس مانند قیمت محصول، وضعیت سفارش و موجودی انبار.

این اطلاعات برای پیگیری امنیتی و بررسی‌های بعدی ضروری هستند.

۴) طراحی View یا گزارش برای Audit

برای راحت‌تر دیدن و تحلیل لاگ‌ها، حداقل یک view تعریف کنید، مثلاً:

- view برای نمایش آخرین تغییرات روی سفارش ها.
- view برای نمایش تاریخچه تغییر قیمت محصولات.
- view برای نمایش تغییرات موجودی کالاها

تمام دستورات SQL مربوط به این گام را در فایل phase9_audit.sql قرار دهید. این فایل sql باید شامل تست‌های شما هم باشد. به طور مختصر فرایند را توضیح دهید و اسکرین‌شات‌های لاگ‌ها را با هدر phase9_audit در فایل pdf قرار دهید.

این گام امتیازی است و پیاده‌سازی تمیز و مستند آن می‌تواند در نمره‌ی پروژه تأثیر مثبت داشته باشد.