تمرین اول اصول طراحی پایگاه داده‌ها

اشکان شکیبا (9931030)

سوال اول

پنج قسمت اصلی یک سیستم مدیریت پایگاه داده عبارت‌اند از:

موتور پایگاه داده: این بخش مسئول ذخیره‌سازی، مدیریت و دسترسی به داده‌ها است. همچنین این بخش شامل قابلیت‌های مختلفی مانند جستجو، ترتیب‌بندی، فیلتر کردن و ... می‌باشد.

زبان پایگاه داده: زبانی است که توسط کاربران و برنامه‌ها برای دسترسی به داده‌ها و انجام عملیات مختلف بر روی آنها استفاده می‌شود. مانند SQL.

مدیریت حریم خصوصی و امنیت: این بخش مسئول مدیریت حریم خصوصی و امنیت داده‌ها است. این بخش شامل مدیریت دسترسی، رمزنگاری، پشتیبانی از تصدیق هویت و سایر امکانات امنیتی است.

مدیریت تراکنش‌ها: مدیریت تراکنش‌ها در DBMS به معنای مدیریت فرایند انجام تراکنش‌هایی است که به صورت همزمان توسط چند کاربر یا برنامه انجام می‌شود.

ابزارهای مدیریتی: ابزارهای مدیریتی، ابزارهایی هستند که برای مدیریت، پشتیبانی و مانیتورینگ پایگاه داده‌ها استفاده می‌شود. به عنوان مثال، ابزارهای مانیتورینگ عملکرد، ابزارهای پشتیبانی و بازیابی و سایر ابزارهای مدیریتی می‌باشد.

سوال دوم

استفاده از فایل به صورت مستقیم برای ثبت داده‌ها، به جای استفاده از یک سیستم مدیریت پایگاه داده، مشکلات و معایبی دارد که به طور کلی به شرح زیر هستند:

نامنظم بودن داده‌ها: در یک فایل، داده‌ها به صورت نامنظم و بدون ترتیب واضحی قرار دارند. این باعث می‌شود که پردازش و جستجوی داده‌ها به صورت موثر و با کارایی بالا انجام نشود.

عدم پشتیبانی از تراکنش‌ها: در یک فایل، تراکنش‌ها به صورت مناسب پشتیبانی نمی‌شوند. این موضوع می‌تواند باعث ایجاد مشکلاتی مانند از دست دادن داده‌ها، یا به وجود آمدن داده‌های تکراری یا نامرتبط شود.

عدم قابلیت اطمینان: در یک فایل، در صورت بروز خطایی، امکان بازیابی داده‌های از دست رفته، بسیار کم است. بنابراین، این موضوع می‌تواند باعث از دست دادن داده‌های مهم شود.

عدم امنیت: در یک فایل، امنیت داده‌ها به شکل مناسبی تامین نمی‌شود. اطلاعات در فایل به صورت روشن و در دسترس قرار دارند، که می‌تواند باعث سرقت اطلاعات شود.

عدم قابلیت بهینه سازی: در یک فایل، امکان بهینه سازی داده‌ها و کوئری‌ها به صورت مناسب وجود ندارد. به عبارت دیگر، این موضوع می‌تواند باعث کاهش کارایی و کندی در اجرای کوئری‌ها و پردازش داده‌ها شود.

سوال سوم

Data Integrity به معنای صحت و صداقت داده‌های موجود در پایگاه داده است. به عبارت دیگر، هرگونه تغییر در داده‌های موجود باید در چارچوب قوانین و محدودیت‌هایی که برای داده‌ها تعریف شده است انجام شود، تا دیتابیس دچار مشکلاتی نشود.

در دیتابیس، Data Integrity می‌تواند به صورت مفهومی مشخص شده و قوانینی برای صحت داده‌ها تعریف شود، به طوری که به صورت خودکار و در هنگام ورود داده‌های جدید یا تغییرات در داده‌های موجود، قوانین اعمال شود و هرگونه نقض این قوانین به صورت خودکار از طریق مکانیزم‌های مختلفی که در سیستم مدیریت پایگاه داده وجود دارد، تشخیص داده شود.

در فایل سیستم، Data Integrity به صورت مشخص و قابل اجرا تعریف نمی‌شود و امکان بروز خطاها در داده‌های موجود بسیار زیاد است. به علاوه، در فایل سیستم، تغییر در داده‌ها بسیار آسان است و هرگونه نقض در قوانین محدودیت‌های داده‌ها نمی‌تواند به صورت خودکار تشخیص داده شود. این موضوع باعث می‌شود که در فایل سیستم، Data Integrity به صورت دستی و با استفاده از ابزارهای خاص انجام شود که کار آن‌ها بسیار زمان‌بر و پر هزینه است.

بنابراین، استفاده از یک سیستم مدیریت پایگاه داده که Data Integrity را به صورت خودکار مدیریت می‌کند، در مقایسه با فایل سیستم، مزایای بسیاری دارد.

سوال چهارم

Database Administrator یا مدیر پایگاه داده، شخصی است که مسئولیت مدیریت، نظارت و اجرای سیستم پایگاه داده را دارد. بعضی از وظایف و مسئولیت‌های یک DBA عبارتند از:

طراحی و پیاده‌سازی پایگاه داده: DBA باید قادر باشد تا پایگاه داده‌ای که مورد نیاز سازمان است را طراحی کند و پیاده‌سازی کند.

نظارت و مانیتورینگ پایگاه داده: DBA باید به صورت مداوم پایگاه داده را بررسی کرده و مشکلات را رفع کند. این شامل بررسی عملکرد پایگاه داده، رفع خطاهای سیستم، پشتیبان گیری و بازیابی اطلاعات و مدیریت دسترسی‌ها می‌شود.

به‌روزرسانی و پشتیبان‌گیری از پایگاه داده: DBA باید به صورت منظم پایگاه داده را به‌روزرسانی کند و از پایگاه داده پشتیبان گیری منظم داشته باشد تا در صورت بروز خطا، بتواند اطلاعات را بازیابی کند.

مدیریت امنیت پایگاه داده: DBA باید برای حفاظت از اطلاعات حساس در پایگاه داده، امنیت سیستم را مدیریت کند. این شامل مدیریت دسترسی‌ها، اجرای امنیتی پایگاه داده، مانیتور کردن حملات سایبری و پیشگیری از آن‌ها می‌شود.

ارائه راهکارهای بهینه‌سازی: DBA باید راهکارهای بهینه‌سازی برای پایگاه داده ارائه دهد تا عملکرد و بهره‌وری پایگاه داده بالا برود. شامل بهینه سازی پرس‌وجوها، شناسایی و رفع مشکلات عملکرد.

سوال پنجم

انتزاع داده در طراحی پایگاه داده به معنای پنهان کردن جزئیات پیاده‌سازی و ارائه یک دید ساده و چیدمانی شده از داده به کاربران است. این کار به کاربران اجازه می‌دهد که با داده‌ها با یک زبان بالاترین سطح یا یک مدل مفهومی کار کنند که به راحتی قابل فهم و استفاده است.

سه سطح مختلف انتزاع داده در طراحی پایگاه داده وجود دارد که به شرح زیر است:

۱. سطح فیزیکی: این سطح نحوه‌ی ذخیره سازی داده‌ها در پایگاه داده را نشان می‌دهد. شامل جزئیاتی مانند ساختارهای ذخیره‌سازی داده، سازماندهی پرونده‌ها، روش‌های دسترسی و تکنیک‌های فشرده‌سازی داده است. سطح فیزیکی پایین‌ترین سطح انتزاع است و بیشتر به مسائل مربوط به ذخیره و بازیابی بهینه داده توجه دارد.

۲. سطح منطقی: این سطح داده را با استفاده از سیستم مدیریت پایگاه داده نشان می‌دهد. شامل جزئیاتی مانند جداول، نمایش‌ها، فهرست‌ها و محدودیت‌ها است. سطح منطقی یک نمای مفهومی از داده‌ها ارائه می‌دهد و مستقل از جزئیات فیزیکی ذخیره‌سازی است.

۳. سطح دید: این سطح یک زیرمجموعه از داده‌های موجود در سطح منطقی را که مربوط به یک کاربر یا برنامه خاص است، نشان می‌دهد و شامل جزئیاتی مانند رابط کاربری، گزارش‌ها و فرم‌ها است.

سوال ششم

دو دسته کلی اصلی از زبان‌های manipulation data شامل زبان‌های declarative و procedural هستند.

زبان‌های declarative:

این زبان‌ها مانند SQL، به شما اجازه می دهند که بگویید چه کاری را باید انجام دهید، اما روش انجام کار را به آنها بسپارید. به عبارت دیگر، شما به آنها می گویید چه نتایجی را می خواهید و این زبان‌ها با استفاده از الگوریتم‌های خود، کار را انجام می دهند. برای مثال، در SQL، شما با استفاده از دستور SELECT می توانید داده‌های خود را از دیتابیس خوانده و به دست آورید.

زبان‌های procedural:

در این نوع از زبان‌ها مانند Python و R، شما به آنها می گویید که چه کاری را انجام دهند و چگونه آن را انجام دهند. به عبارت دیگر، شما یک الگوریتم خاص را برای انجام کار مشخص می کنید. برای مثال در Python، شما می توانید با استفاده از دستوراتی مانند for و if، داده‌های خود را مورد بررسی و تحلیل قرار دهید.

بنابراین، تفاوت اصلی بین این دو نوع زبان، در روش استفاده از آنها برای کار با داده‌ها است. در زبان های declarative، شما می گویید چه کاری باید انجام شود و در زبان‌های procedural، شما می گویید چگونه این کار باید انجام شود.

سوال هفتم

تراکنش در پایگاه داده‌ها به معنای یک دستور یا یک سری از دستورات است که به صورت اتمی و قابل بازگشت اجرا می‌شود. به عبارت دیگر، تراکنش یک واحد کاری است که یا به صورت کامل انجام می‌شود و تغییراتی که در پایگاه داده صورت گرفته‌اند به صورت پایدار ثبت می‌شوند، یا در صورت عدم توانایی در انجام تمامی دستورات، هیچ تغییری اعمال نمی‌شود.

ویژگی‌های ACID یا Atomicity، Consistency، Isolation و Durability، چهار ویژگی اصلی تراکنش‌ها هستند.

اتمیک بودن (Atomicity):

این ویژگی در تراکنش‌ها به همه یا هیچ معروف است، در واقع تراکنش در صورتی موفق است که تمام دستوراتی که در آن اجرا می شوند موفق باشند و بر اساس این قاعده دیتابیس باید قادر باشد که در صورت عدم موفقیت هر دستوری، تمامی دستورات قبلی را بازگردانی کند.

سازگاری (Consistency):

یک تراکنش در واقع بایستی در هنگام تغییر وضعیت، اطلاعات را از یک وضعیت صحیح به یک وضعیت صحیح دیگر ببرد. در این تغییر وضعیت دیتابیس هم سعی می کند که هیچ کدام از کلیدها و نوع‌های داده‌ای و Triggerها نباید نقض شوند.

انزوا (Isolation): این ویژگی برای جلوگیری از همزمانی‌های ناخواسته در پایگاه داده است. با این ویژگی، هر تراکنش باید به صورت جداگانه اجرا شود و تداخل با تراکنش‌های دیگر را به حداقل برساند.

دوام‌پذیری (Durability): یکی از ویژگی‌های پایگاه داده است که به معنی اطمینان از دوام داده‌ها در پایگاه داده پس از اعمال تراکنش است. به طور دقیق‌تر، این ویژگی به معنی تضمین این است که هر تراکنش که با موفقیت اعمال شده است، تغییراتی که در داده‌ها ایجاد کرده است به صورت دائمی در پایگاه داده ذخیره می‌شوند و در صورت قطعی یا خرابی سیستم، این داده‌ها بازیابی خواهند شد.

سوال هشتم

عملیاتی که سیستم مدیریت پایگاه داده (DBMS) برای جلوگیری از خطاها انجام می‌دهد، با نام کنترل همروندی (Concurrency Control) شناخته می‌شود. این مکانیزم مدیریت دسترسی به منابع مشترک مانند پایگاه داده را مدیریت می‌کند تا اطمینان حاصل شود که تراکنش‌های همروند با یکدیگر تداخل نداشته باشند.

در این سناریو، هنگامی که دو کاربر به صورت همزمان تلاش می‌کنند تا صندلی هواپیما را خریداری کنند، کنترل همروندی مطمئن می‌شود که DBMS اجازه نمی‌دهد که هر دو تراکنش به طور همزمان به داده‌های مشابه دسترسی یافته و آنها را تغییر دهند که می‌تواند به ناهماهنگی و خطاها منجر شود. به جای اینکه این مکانیزم به دو تراکنش اجازه دهد که به صورت همزمان به داده‌ها دسترسی پیدا کنند و تغییراتی در آن‌ها ایجاد کنند، تضمین می‌کند که تنها یک تراکنش می‌تواند در هر زمان به داده‌ها دسترسی پیدا کند و تراکنش دیگر باید منتظر تراکنش اولیه بماند یا به دلیل خطا لغو شود. به این ترتیب، کنترل همروندی از ایجاد ناهماهنگی در داده‌ها جلوگیری می‌کند و اطمینان حاصل می‌شود که صحت داده‌ها حفظ می‌شود.

سوال نهم

داده‌های بدون ساختار: داده‌هایی هستند که در قالبی غیر ساختارمند به صورت دودویی (binary) در کامپیوتر ذخیره می‌شوند و در آن‌ها هیچ نوع سازماندهی و روابط مشخص وجود ندارد. به‌عنوان مثال فایل‌های تصویری، ویدئویی، موسیقی و متنی که در فرمت‌هایی مانند txt، pdf، docx، و mp3 ذخیره می‌شوند داده‌های بدون ساختار هستند.

داده‌های نیمه ساختارمند: داده‌هایی هستند که در قالبی نیمه ساختارمند و با استفاده از تگ‌های مشخص و دستورات مانند XML، JSON، و YAML سازمان‌دهی می‌شوند. این داده‌ها دارای یک ساختار معین هستند، اما این ساختار می‌تواند گسترش یابد. به‌عنوان مثال، فایل‌های سبک اطلاعاتی (metadata) عکس‌ها، فیلم‌ها، موسیقی و دیگر فرمت‌های دیجیتالی نیمه ساختارمند هستند.

داده‌های ساختارمند: داده‌هایی هستند که در قالبی ساختارمند سازمان‌دهی می‌شوند و شامل جداول و روابط بین جداول هستند. این داده‌ها دارای یک ساختار مشخص و دقیق برای فیلدهای داده‌ای هستند که به وسیله‌ی داده‌های دیگر وابسته هستند. به‌عنوان مثال، پایگاه‌داده‌های مشتریان، فروش، انبار و سفارشات ساختارمند هستند.

سوال دهم

جدول کالاها، شامل اطلاعات کالا از جمله اسم، تصویر، مشخصات فیزیکی و فنی، قیمت و...

جدول فروشندگان، شامل اطلاعات فروشنده از جمله نام، مشخصات تماس، کالاهای به فروش گذاشته شده و...

جدول مشتریان، شامل اطلاعات مشتری از جمله نام، مشخصات تماس، تاریخچه خریدها و جست‌وجوها و...

سوال یازدهم

1) $\pi_{account\_id}(\sigma_{type=passenger \wedge first\_name="John"}(Accounts))$

2) $Accounts \bowtie_{type=driver \wedge account\_id=driver\_id \wedge banned=true} Rides$

3) $Rides \bowtie_{driverId=driver\_id \wedge passengerId=passenger\_id \wedge driverFirstName=passengerFirstName} (\sigma_{type=passenger}(\rho_{p(passengerId, passengerFirstName)}(Accounts)) \times \sigma_{type=driver}(\rho_{d(driverId, driverFirstName)}(Accounts)))$

4) $G_{avg(price)}(\sigma_{request\_date.year=1401} Rides)$

5) $G_{avg(rating)}(Rides \bowtie_{account\_id=passenger\_id} \sigma_{first\_name="Simon" \wedge last\_name="Cowell"}(Accounts))$

6) $ValidRides \leftarrow (\pi_{driverId}(\sigma_{banned=false \wedge type=driver}(\rho_{d(driverId)}(Accounts))) \times \pi_{passengerId}(\sigma_{banned=false \wedge type=passenger}(\rho_{p(passengerId)}(Accounts)))) \bowtie_{driverId=driver\_id \wedge passengerId=passenger\_id} Rides$

$\rho_{ratio(count/G\ count\ (ValidRides))}(G_{count}(\sigma_{status=canceled}(ValidRides)))$