

طراحی پایگاه داده ها

دانشکده مهندسی کامپیوتر

مریم رمضانی
بهار ۱۴۰۴



تمرین سوم - بخش تئوری

تاریخ انتشار: ۰۳ اردیبهشت ۱۴۰۴

۱. پرسش‌های خود درمورد این تمرین را در سامانه کوئرا مطرح کنید.

۲. سیاست ارسال با تاخیر: شما در مجموع در طول نیم‌سال می‌توانید از ۹ روز تاخیر استفاده کنید. همچنین هر تمرین را می‌توانید تا حداکثر ۴ روز با تاخیر تحویل دهید. تاخیرها با مقیاس ساعت محاسبه شده و به بالا گرد می‌شوند.

۳. سیاست مشارکت دانشجویان در حل کردن تمرین: دانشجویان می‌توانند در حل تمرین برای رفع ابهام و یا به‌دست آوردن ایده‌ی کلی با یک‌دیگر مشورت و همفکری کنند. این کار مورد تایید و تشویق تیم ارائه‌ی درس می‌باشد؛ چرا که هم‌فکری و کار گروهی می‌تواند موجب تقویت یادگیری شود. اما به‌دست آوردن جزئیات راه‌حل و نگارش پاسخ باید تماماً توسط خود دانشجو انجام شود. حتماً در انتهای پاسخ‌های ارسالی خود نام افرادی که با آن‌ها همفکری کردید را ذکر کنید.

۴. این تمرین (بخش تئوری و عملی) مجموعاً ۱ نمره دارد.

سوالات تئوری (۵۰ نمره)

تاریخ تحویل: ۱۷ اردیبهشت ۱۴۰۴

پرسش ۱ (۱۲ نمره) در پایگاه داده یک شرکت دو جدول زیر را داریم:

Employees(employee_id, name VARCHAR(100), department_id INT, salary NUMERIC, hire_date DATE)
Departments(department_id, name VARCHAR(100))

(آ) یک VIEW ایجاد کنید که نام کارمند، نام دپارتمان و حقوق را نمایش دهد. این VIEW در چه سناریو هایی میتواند کاربردی و مفید باشد؟

(ب) در VIEW فوق اگر یک دپارتمان حذف شود، چه اتفاقی برای داده‌های VIEW خواهد افتاد؟ (سناریو های مختلف را شرح دهید.)

(ج) یک VIEW مشابه ایجاد کنید که فقط کارمندانی را نمایش دهد که حقوقشان از ۵۰ میلیون بیشتر است.

(د) استفاده از WITH CHECK OPTION در ساخت این VIEW چه تاثیری دارد؟ آیا استفاده از آن الزامی است؟

پرسش ۲ (۱۰ نمره) سه جدول زیر در اختیار دارید:

Student(student_id, student_name, has_outstanding_fee BOOLEAN)
Course(course_id, course_name, prerequisite_course_id, max_seats INT)
Enrollment(student_id, course_id, enrollment_date)

(آ) در این محیط کدام قیود یکپارچگی ممکن است نقض شوند؟ به‌خصوص به دو قید «تکمیل prerequisite پیش از ثبت نام» و «رعایت محدودیت max_seats برای هر Course» توجه کنید و شرح دهید در صورت نقض هر یک، چه پیامدهای عملی و آموزشی به‌وجود می‌آید.

(ب) یکی از این قیود را به دلخواه خود انتخاب کنید و یک ASSERTION در SQL بنویسید که نقض آن قید را جلوگیری کند. سپس توضیح دهید اگر در حین اجرای یک تراکنش این ASSERTION نقض شود، هنگام ثبت داده چه اتفاقی برای تراکنش و عملیات COMMIT رخ می‌دهد.

(ج) می‌دانیم که امکان تعریف ASSERTION در عمل وجود ندارد. نشان دهید با استفاده از TRIGGER در PostgreSQL چگونه می‌توان همان قید انتخاب‌شده در بخش «ب» را به‌طور کامل در سطح پایگاه داده اعمال کرد.

پرسش ۳ (۱۸ نمره) سه جدول زیر را در نظر بگیرید:

orders(order_id, customer_id, order_date)
order_items(order_id, product_id, quantity, price)
products(product_id, product_name, unit_price)

(آ) یک view برای order_summary طراحی کنید که order_id, customer_id, order_date و total_amount (محاسبه‌شده با SUM روی order_items) را نمایش دهد.

(ب) یک تابع trigger از نوع INSTEAD OF روی order_summary view بنویسید که به‌روزرسانی مقدار total_amount را مدیریت کند. این تابع باید:

• total_amount فعلی هر سفارش را محاسبه کند.

- اگر مقدار جدید بزرگ‌تر باشد، یک رکورد «bonus» در جدول order_items (با product_id دلخواه و quantity مناسب) درج کند.
- اگر مقدار جدید کوچک‌تر باشد، مقدار یکی از رکوردهای order_items را به‌صورت نسبی کاهش دهد (به‌گونه‌ای که هیچ quantity منفی نشود).
- اگر یک رکورد در order_items حذف شود یا مقدار quantity آن تغییر کند، view به‌طور خودکار به‌روزرسانی شود و total_amount جدید محاسبه گردد.

(ج) چالش‌های طراحی هنگام به‌روزرسانی ستون‌های محاسبه‌شده که به چند جدول وابسته‌اند (از جمله همگام‌سازی حذف یا تغییر داده‌ها) را تحلیل کنید.

پرسش ۴ (۱۰ نمره) یک پایگاه داده برای ذخیره اطلاعات فیلم‌های سینمایی به همراه بازیگران و تولیدکنندگان آن شامل جداول زیر است:

Movie(title, year, producerID)

Producer(pID, pName, netWorth)

با توجه به نیاز مکرر به دسترسی به نام تولیدکننده یک فیلم، یک دید ذخیره شده با دستور SQL زیر ساخته شده است:

CREATE MATERIALIZED VIEW MovieProd **AS**

SELECT title, year, pName **FROM** Movie, Producer

WHERE producerID = pID

در ابتدای کار، برای ساخت دید ذخیره شده، کل دستور ساخت آن اجرا می‌شود. برای حفظ سازگاری دید با جدول مبنا نیز می‌توان بعد از هر تغییر جدول مبنا، دید را از نو ساخت اما این کار سربار زیادی دارد. در عوض می‌توان با اجرای دستورات ساده‌تر، سازگاری را در هر گام حفظ کرد. برای هر کدام از تغییرات زیر، کمترین دستورات لازم برای حفظ سازگاری دید با جدول مبنا را ارائه دهید.

(آ) **INSERT INTO** Movie **VALUES** ('New DB', 1404, 123)

(ب) **DELETE FROM** Movie **WHERE** title = 'Old DB' **AND** year = 1403

(ج) **INSERT INTO** Producer **VALUES** (456, 'Rox', 100)

(د) **DELETE FROM** Producer **WHERE** pName = 'Bob'

(ه) **UPDATE** Producer **SET** netWorth = netWorth * 5 **WHERE** netWorth < 10