

فصل چهارم

جبر و حساب رابطه‌ای

هدف‌های کلی

شناخت نظری مدل‌رابطه‌ای با مطالعه دو روش جبر رابطه‌ای و حساب رابطه‌ای دامنه‌ای

هدف‌های یادگیری

۱. مدل پایگاهداده رابطه‌ای از پشتونه نظری بسیار قوی برخوردار است. مفاهیم این مدل به شکل‌های مختلف به صورت نظری بیان و اثبات شده‌اند. خوانندگان گرامی در این فصل با دو روش ریاضی برای بیان پرسش در پایگاهداده رابطه‌ای آشنا می‌شوند.
۲. اولین روش جبر رابطه‌ای نام دارد. در این روش، عملگرهای رابطه‌ای روی رابطه‌ها عمل می‌کنند تا نتایج مورد نظر را به دست دهند.
۳. با عملگرهای رابطه‌ای می‌توان رابطه‌ها را به اشکال مختلف درهم آمیخت و اطلاعات جدیدی استخراج کرد و نیز رابطه‌ها را به روز نگه داشت.
۴. روش دومی که به صورت نظری آموزش داده می‌شود، حساب رابطه‌ای دامنه‌ای است. در این روش، با بیان روابط موجود بین رابطه‌ها، اطلاعات جدید در قالب رابطه‌های جدید استخراج می‌شوند. حساب رابطه‌ای با نیازهای هوش مصنوعی سازگاری بیشتری دارد.

مقدمه

جبر، به معنی نوع داده و عملگرهای^۱ روی آنها است. مثلاً جبر بول^۲، نوع داده‌اش {درست، نادرست} و عملگرهایش {و، یا، نه} است (معادل {false، {and or, true} و {not,}). همچنین جبر اعداد حقیقی که در مدارس تدریس می‌شود، نوع داده‌اش اعداد حقیقی و عملگرهای اصلی آن، جمع و تفریق و ضرب و تقسیم است (عملیاتی از قبیل توان و اتحادها و تجزیه، در واقع برای ساده‌شدن چهار عمل اصلی و تبدیل آنها به یکدیگر است). جبر رابطه‌ای قوی‌ترین مبنای نظری مدل رابطه‌ای است [E. F. Codd, 1972] و با نشانه‌گذاری‌ها^۳ و شکل‌های متفاوتی در متون دانشگاهی آمده است. آنچه در این فصل می‌آید، نشانه‌گذاری استاندارد برگرفته از معتبرترین کتاب‌های دانشگاهی موجود است.

[F. Korth and A. Silberchatz, 2006] و [R. Elmasri, 2004].

۱-۴ جبر رابطه‌ای^۴

جبر رابطه‌ای از دو قسمت تشکیل شده است: نوع داده که رابطه‌ها هستند و عملگرها که روی رابطه‌ها عمل می‌کنند. روی این نکته باید تأکید کرد که در مدل رابطه‌ای (جبر یا SQL یا ...) هیچ‌گاه نمی‌توان نوع داده را از رابطه به چیز دیگری تغییر داد. اجزای رابطه باید از دامنه‌های ساده باشد و حتی رابطه‌های تودرتو مجاز نیست (پرسش‌های تودرتو مفهوم دیگری است).

۱-۱-۴ انواع عملگرها

عملگرها در جبر رابطه‌ای، از چهار نوع زیر است:

- ساده شامل گزینش سطر (با علامت σ) و پرتو یا گزینش ستون (با علامت Π)
- مجموعه‌ای شامل اجتماع، اشتراک و تفاضل (به ترتیب با علامت‌های \cup و \cap و $-$)

1. Operators
2. Boolean Algebra
3. Notation
4. Relational Algebra

- پیوند شامل ضرب دکارتی با علامت \times ، پیوند شرطی^۱ با علامت \times_0 ، پیوند طبیعی^۲ با علامت ∞ و نیمپیوند^۳ با علامت ∞ .
 - نامگذاری عبارت‌ها (با علامت ρ) و جایگزینی (با علامت \leftarrow) و تقسیم (با علامت \div) این عملگرهای به دو دسته اصلی و اضافی تقسیم می‌شوند. عملگرهای اضافی حالت کمکی دارند و می‌توان آن‌ها را با ترکیب عملگرهای اصلی جایگزین کرد.
- شکل ۱-۴ عملگرهای جبر رابطه‌ای و اصلی یا اضافی بودن آن‌ها را نشان می‌دهد. نوع داده در جبر رابطه‌ای فقط و فقط رابطه است، یعنی ورودی و خروجی تمامی عملگرهای رابطه‌ها هستند.

	علامت	اصلی یا اضافی	توضیحات
گزینش	σ	اصلی	انتخاب سطر
پرتو	Π	اصلی	انتخاب ستون
اجتماع	\cup	اصلی	جدول باید همتا باشد
اشتراک	\cap	اضافی	جدول باید همتا باشد
تفاضل	-	اصلی	جدول باید همتا باشد
تقسیم	\div	اضافی	برای پرسش‌های دارای شرط همه زمان و حافظه زیادی می‌گیرد
ضرب دکارتی	\times	اصلی	زیرمجموعه ضرب دکارتی
پیوند شرطی	\times_0	اضافی	هرچند اضافی است ولی کاربرد زیادی دارد
پیوند طبیعی	∞	اضافی	کاربرد اصلی آن در پایگاهداده توزیعی است
نیمپیوند	∞	اضافی	تغییرنام جدول
نامگذاری	ρ	اضافی	کپی کردن جدول
جایگزینی	\leftarrow	اصلی	

شکل ۱-۴. عملگرهای جبر رابطه‌ای

پاسخ به پرسش در جبر رابطه‌ای با استفاده از عملگرهای بالا و ترکیب آن‌ها صورت می‌گیرد. مثلاً اگر بخواهیم ستون‌هایی از یک جدول را برای سطرهای خاصی انتخاب کنیم باید از ترکیب گزینش و پرتو استفاده کنیم. در این فصل از جداول پایگاهداده نشر کتاب که در فصل قبل معرفی شد، استفاده می‌شود.

1. Theta Join
2. Natural Join
3. Semi Join

۱. انتخاب سطر و ستون (گزینش و پرتو)

سطرهای جدول با گزینش انتخاب می‌شود. نام جدول جلوی علامت گزینش در پرانتز و شرط انتخاب سطر در زیر آن می‌آید. همه ستون‌های آن جدول در خروجی پدیدار می‌شوند.

تذکر: در این فصل عیناً از جداول فصل سوم استفاده می‌شود.

مثال: مؤلف‌هایی که آدرس آن‌ها «کیش» است و «متترجم» هستند.

$$\sigma_{\text{adres}=\text{'کیش'} \wedge \text{typ}=\text{'متترجم'}}^{\text{(author)}}$$

پرتو، ستون‌هایی از جدول را انتخاب می‌کند و هیچ‌گونه شرطی اعمال نمی‌شود و سطرهای تکراری از خروجی حذف می‌شود.

نام جدول جلوی علامت پرتو در پرانتز و ستون‌های انتخاب شده زیر آن می‌آید.

مثال: کلید کتاب، نام کتاب و تاریخ چاپ آن.

$$\prod_{\text{bno}, \text{title}, \text{dop}}^{\text{(book)}}$$

از ترکیب گزینش و پرتو می‌توان اطلاعات بیشتری از جداول به دست آورد، ولی ترتیب قرارگرفتن این دو در دستور، چنانکه خواهیم دید، گاهی مهم است و در موقعی اهمیت ندارد.

مثال: کلید مدیر نشر، کلید کتاب، قیمت واحد و تیراژ کتاب‌هایی که بالای ۱۰۰۰ نسخه منتشر شده‌است.

$$\prod_{\text{pubno}, \text{bno}, \text{Uprice}, \text{num}}^{\text{(punish))}}$$

پاسخ معادل:

$$\left(\prod_{\text{pubno}, \text{bno}, \text{Uprice}, \text{num}}^{\text{(punish))}} \right)$$

حال این مثال را اندکی تغییر می‌دهیم تا ببینیم چگونه جابه‌جاکردن گزینش و پرتو ممکن است نتیجه غلط بدهد. برای این‌منظور تیراژ را از خروجی حذف می‌کنیم: مثال: کلید مدیر نشر، کلید کتاب و قیمت واحد کتاب‌هایی که بالای ۱۰۰۰ نسخه منتشر شده‌است.

پاسخ درست:

$$\prod_{\text{pubno}, \text{bno}, \text{Uprice}}^{\text{(punish))}}$$

پاسخ غلط:

$$\sigma_{\text{num} > 10000} \left(\prod_{\text{pubno}, \text{bno}, \text{Uprice}} (\text{punish}) \right)$$

این پاسخ غلط است، زیرا ستون num در خروجی پرتو وجود ندارد و بنابراین شرط اعمال شده $\text{num} > 10000$ معنی ندارد.

۲. عملگرهای مجموعه‌ای

این عملگرها همانند نظریه مجموعه‌ها شامل اجتماع (U) اشتراک (I) و تفاضل (-) با همان معنا هستند. ورودی هر کدام از آن‌ها دو رابطه و خروجی هر کدام یک رابطه است. روابط ورودی، باید دارای خاصیت همتایی^۱ باشند یعنی:

- تعداد صفت‌های دو رابطه (ستون‌های دو جدول) مساوی باشند.
- صفت‌ها به ترتیب دارای دامنه‌های یکسان باشند.

مثال: نام همه کتاب‌ها و مؤلف‌ها

حل: این نام‌ها یا نام کتاب هستند یا نام مؤلف. بنابراین باید نام کتاب‌ها و نام مؤلف‌ها را جداگانه لیست و سپس با هم اجتماع کنیم (اسامی تکراری حذف می‌شود).

$$\prod_{\text{title}} (\text{book}) \bigcup \prod_{\text{name}} (\text{author})$$

مثال: نام کارکنانی که مؤلف نیستند.

حل: ابتدا نام کارکنان را پیدا و سپس نام مؤلفان را از آن تفریق می‌کنیم، یعنی

$$\prod_{\text{name}} \text{personel} - \prod_{\text{name}} (\text{author})$$

مثال: اسامی کارکنانی که مؤلف‌های هم‌نام دارند.

حل: کافی است اشتراک اسامی کارکنان و مؤلفان را پیدا کنیم، یعنی

$$\prod_{\text{name}} \text{personel} \bigcap \prod_{\text{name}} (\text{author})$$

یادآوری چند نکته الزامی است:

- عملگر اشتراک را گاهی می‌توان با عملگرهای دیگر جایگزین کرد، یعنی می‌توان آنچه را که از اشتراک به دست می‌آید، با پیوند دو جدول نیز محاسبه کرد، ولی برای اجتماع به سختی می‌توان از طریق دیگری اقدام کرد و برای تفاضل بسیار دشوار و معمولاً غیرممکن است.

- عملگرهای مجموعه‌ای به تنها یی از توان محاسباتی کافی برخوردار نیستند. اصولاً توان محاسباتی در جبر رابطه‌ای در ترکیب انواع عملگرها نهفته است. در ادامه همین فصل، ضمن معرفی عملگرهای دیگر، مثال‌هایی ارائه خواهد شد.
- ترتیب جداول ورودی فقط در تفاضل، مهم است. به عبارت دیگر:

$$x - y \neq y - x \quad x \cap y = y \cap x \quad x \cup y = y \cup x$$

در فصل گذشته با عملگرهای پیوند آشنا شدیم و دیدیم که معمولاً بسیار گران تمام می‌شوند (زمان و فضای زیادی می‌گیرند) و می‌توان آن‌ها را با عملگرهای دیگر جایگزین کرد. در این بخش عملگرهای پیوند را در جبر رابطه‌ای بررسی می‌کنیم.

۳. ضرب دکارتی

اولین عملگر پیوند، ضرب دکارتی (با علامت \times) است که یکی از گران‌ترین عملگرهای رابطه‌ای است و باید حتی‌الامکان از آن پرهیز کرد. با این‌همه در پاسخ‌گویی به بعضی از پرسش‌ها مجبور هستیم از آن استفاده کنیم. ضرب دکارتی دو جدول، جدولی است که ستون‌هایش، همه ستون‌های دو جدول و سطرهایش تمام ترکیب‌های ممکن سطرهای آن دو جدول است. اگر دو جدول ستون‌های همنام داشته باشند در خروجی از نقطه‌گذاری برای تشخیص آن‌ها استفاده می‌شود (مثلاً $\text{book.title} \times \text{publisher.title}$ و $\text{book.title} \times \text{publisher}$).
مثال: اطلاعات کامل نشر و ناشرین

پاسخ غلط:

$\text{publisher} \times \text{publish}$

توضیح: این پاسخ غلط است زیرا علاوه‌بر سطرهای مربوط به ناشرها و انتشارات آن‌ها، تعداد زیادی سطر مربوط به ناشرها و انتشاراتی که متعلق به آن‌ها نیست ظاهر می‌شود. راه حل این مشکل در پیوندهای دیگر است که در زیر می‌آید.

۴. پیوند شرطی

عملگر دوم، پیوند شرطی (با علامت $\theta \times$) می‌باشد که زیرمجموعه‌ای از ضرب دکارتی است (با اعمال شرط θ روی سطرهای آن). ستون‌های خروجی معادل ستون‌های ضرب دکارتی است.

مثال: کلید و نام کتاب‌ها با قیمت بالای ۱۰۰۰۰

پاسخ:

$$\prod_{\text{bno}, \text{title}} \text{book.bno} = \text{publish.bno} \wedge \text{Uprice} > 10000 \quad (\text{book} \times \text{publish})$$

این راه حل بسیار گران است، زیرا جدول بزرگی شامل همه ترکیب‌های ممکن سطري و ستونی دو جدول book و publish تشکيل مى شود و سپس بخش کوچکی از آن انتخاب و بقیه دور ریخته مى شود. این مشکل را می‌توان با استفاده از دستورهای دیگر حل کرد. به این‌گونه مسائل، یعنی پیدا کردن راه حل بهتر، بهینه‌سازی پرسش گفته می‌شود که در ادامه همین فصل مورد بررسی قرار خواهد گرفت. ناگفته پیدا است که همواره باید بکوشیم ارزان‌ترین پاسخ را بیابیم.

به عنوان نمونه، یک راه حل ارزان‌تر این است که ابتدا شرط قیمت روی جدول publish اعمال شود و جدول کوچک‌تری به دست آید. آنگاه این جدول با جدول book پیوند شرطی‌شده و در پایان، ستون‌های مورد نظر انتخاب شود.

۵. پیوند طبیعی

عملگر سوم، پیوند طبیعی (با علامت \circ) از معروف‌ترین و کارامد‌ترین عملگرهای است.

پیوند طبیعی از عملگرهای اصلی جبر رابطه‌ای نیست، زیرا مشابه پیوند شرطی است، با تفاوت‌های زیر:

۱. خود به خود شرط تساوی روی همه ستون‌های همنام دو جدول اعمال می‌شود، یعنی فقط سطرهایی از دو جدول انتخاب می‌شوند، که همه ستون‌های همنام آن‌ها، مقادیر مساوی داشته باشند. در صورتی که دو جدول، ستون همنام نداشته باشند نتیجه معادل ضرب دکارتی است.

۲. ستون‌های تکراری فقط یک بار در خروجی ظاهر می‌شوند.

۳. برخلاف پیوند شرطی، جدول ترکیبی تشکیل نمی‌شود و در نتیجه پیوند طبیعی گران نیست و زمان و فضای آن نیز قابل مقایسه با ضرب دکارتی و پیوند شرطی نیست. این‌گونه بهینه‌سازی‌ها در پیوند طبیعی و موارد مشابه با استفاده از تکنیک‌هایی مانند شاخص‌گذاری توسط سیستم انجام می‌شود. در فصل پایانی کتاب به این‌گونه مسائل که به لایه داخلی پایگاه‌داده‌ها مربوط می‌شود، خواهیم پرداخت.

مثال: همان مثال پیوند شرطی: کلید و نام کتاب‌ها با قیمت بالای ۱۰۰۰۰

$$\left(\prod_{bno} \sigma_{Uprice > 10000}^{(\text{publish})} \right) \circ \left(\prod_{bno, \text{title}} (\text{book}) \right)$$

مثال: مشخصات کامل مدیرهای نشر
باید توجه کرد که مشخصات کامل مدیرهای نشر در جدول personnel با کلید آن‌ها آمده است.

$$\left(\prod_{pno} (\text{publisher}) \right)^\infty \text{personel}$$

مثال: خروجی دستور $\text{personel} \bowtie \text{author}$ چیست؟
حل: در طراحی این دو جدول کلید کارکنان (pno) و مؤلف (ano) با دو نام مختلف آمده است و در پیوند طبیعی (در جبر رابطه‌ای) آن‌ها مشارکت نخواهد داشت. این درحالی است که داده‌های دو جدول به ندرت مشترک هستند (فقط برای کارکنانی که مؤلف نیز هستند و پیوند طبیعی باید این افراد را نشان دهد). از سوی دیگر چند ستون هم‌نام مانند ...، name، typ دارند. آیا این ضعف در طراحی است؟! خیر، زیرا در طراحی یک جدول باید استقلال را رعایت کرد. در واقع جبر رابطه‌ای ضعف دارد که پیوند طبیعی جداول را بر مبنای ستون‌های هم‌نام قرار داده است. چنان‌که در فصل گذشته دیدیم زبان SQL که بسیار به جبر رابطه‌ای شبیه و از نظر تاریخی بعد از آن به وجود آمده است، مسئله ارتباط جداول را به‌طور مستقل با استفاده از کلید خارجی¹ حل کرده است.

با توجه به این‌که دو جدول تعداد زیادی ستون هم‌نام دارند، ولی کلید آن‌ها هم‌نام نیست، خروجی این مثال جدولی با همه ستون‌های مشترک (بدون تکرار) خواهد بود و کارکنانی را در بر می‌گیرد که یا مؤلف هستند یا نام و نوع و تاریخ تولد و جنسیت و آدرس یکسان با یک مؤلف دارند!

۶. نیم‌پیوند

عملکر نیم‌پیوند (با علامت \bowtie) می‌باشد. این عملکر مشابه پیوند طبیعی است، با این تفاوت که فقط ستون‌های جدول اول را می‌دهد.

مثال: مشخصات کامل مدیرهای نشر
توجه: در جدول publisher همه ستون‌ها مربوط به مؤسسه نشر است، به جز pno که کلید مدیر نشر است.

$$\text{personel} \bowtie \left(\prod_{pno} (\text{publisher}) \right)$$

توضیح: در پاسخ بالا نیم‌پیوند با پیوند طبیعی یکسان عمل می‌کند. پاسخ زیر فقط کلید مدیر نشر را می‌دهد T زیرا در نیم‌پیوند فقط ستون‌های جدول اول در خروجی ظاهر می‌شود:

$$\left(\prod_{\text{pno}} (\text{publisher}) \right)^\infty \text{personel}$$

مثال: خروجی دستور زیر چیست؟

$$\sigma_{\text{typ} = \text{'آموزشی'} \wedge \text{Uprice} > 1000}^{\text{(book} \bowtie \text{publish)}}$$

پاسخ: این دستور غلط^۱ است، ظاهراً مشخصات کتابی‌هایی از نوع آموزشی را می‌دهد که قیمت آنها بیش از ۱۰۰۰ است، ولی واقعاً چنین نیست، زیرا ستون Uprice در جدول book نیست و پس از نیم‌پیوند حذف می‌شود و نمی‌تواند در شرط گزینش به کار رود. در صورتی که به جای نیم‌پیوند، از پیوند طبیعی استفاده کنیم پاسخ درست می‌شود، ولی ستون‌های اضافی خواهد داشت.

در ارتباط با نیم‌پیوند باید به نکات زیر توجه کرد:

- علاوه بر تعداد ستون‌ها، تعداد سطرهای خروجی نیز می‌تواند کمتر از پیوند طبیعی باشد، زیرا با کنار رفتن چند ستون، سطرهای تکراری حذف می‌شوند.
- مثال: نام، نوع، و تاریخ تولد کارکنانی که ناشر هستند.

$$\left(\prod_{\text{name, typ, dob}} (\text{personel} \bowtie \text{publisher}) \right)$$

اگر خروجی دستور فوق دو سطر به صورت زیر داشته باشد:

name	Typ	Dob
ابوذر	مترجم	۱۳۷۳
ابوذر	مؤلف	۱۳۷۴

آنگاه خروجی دستور زیر که فقط ستون name را دارد و مقدار آن در هر دو سطر جدول بالا یکسان است تنها یک سطر خواهد داشت: ابوذر

$$\left(\prod_{\text{name}} (\text{personel} \bowtie \text{publisher}) \right)$$

- ترتیب جداول ورودی در نیم‌پیوند مهم است ($x \bowtie y \neq y \bowtie x$)، زیرا این عملگر همیشه ستون‌های جدول اول را می‌دهد.

مثال: عبارت $\text{publisher} \bowtie \text{publish}$ مشخصات مدیران نشر را می‌دهد، ولی عبارت $\text{publish} \bowtie \text{publisher}$ مشخصات ناشر (انتشارات) را می‌دهد. این در حالی است که پیوند طبیعی آن‌ها، مشخصات کامل مدیر نشر و انتشارات را با هم می‌دهد.

کاربرد اصلی نیم‌پیوند در پایگاه‌داده‌های نامتمركز (توزیعی) است. اگر جدول A در سایت الف و جدول B در سایت ب ذخیره شده باشد و دستور $A \bowtie B$ را در سایت الف صادر کنیم، ضمن اینکه عمل پیوند طبیعی انجام می‌گیرد، از انتقال داده‌های جدول B از سایت ب به سایت الف پرهیز می‌شود.

۷. نام‌گذاری

نام‌گذاری (با علامت $\rho_{\text{new_name}}^{\text{old_name}}$) نام جدیدی برای یک جدول تعریف می‌کند، در حالی که نام قدیم نیز به قوت خود باقی است (اشارة‌گر جدیدی تعریف می‌شود). در واقع بدون ذخیره‌سازی مجدد یک جدول، می‌توان از آن دو بار استفاده کرد. محدوده عملکرد آن در درون دستور مربوطه است، یعنی پس از اتمام آن دستور، نام جدید دیگر وجود ندارد.

مثال: ناشرانی که آدرس آن‌ها یکی است.

حل: باید دو بار از جدول ناشر استفاده کنیم. دستور زیر پاسخ این پرسش است:

$$\rho_{P1}^{\text{publisher}} \quad P1.adres = P2.adres \wedge P1.pno \neq P2.pno \quad (\rho_{P2}^{\left(\prod_{pno, adres} (\text{publisher}) \right)})$$

جدول ناشر با نام P1 و نیز ستون‌های کلید و آدرس ناشر جداگانه با نام P2 نام‌گذاری می‌شوند. سپس سطراهایی از این دو جدول که آدرس یکسان ولی کلید متفاوتی دارند، انتخاب می‌شوند. باید توجه داشت که اگر شرط $P1.pno \neq P2.pno$ اعمال نشود، نتیجه غلط خواهد بود، زیرا هر ناشری با خودش آدرس مشترک دارد.

مثال: کتاب‌هایی که نام یکسان، ولی نوع متفاوت دارند.

توضیح: فقط نام کتاب‌ها و نوع آن‌ها را نیاز داریم، ولی هر کدام دو بار.

$$\rho_{B1}^{\prod_{title, typ} \text{book}} \quad B1.title = B2.title \wedge B1.typ \neq B2.typ \quad \rho_{B2}^{\left(\prod_{title, typ} (\text{book}) \right)}$$

۸. جایگزینی

جایگزینی (با علامت $\text{new_name} \leftarrow \text{old_name}$) جدول جدیدی ذخیره می‌کند. اگر پاسخ پرسش از چند دستور تشکیل شود یا طولانی باشد، می‌توان با استفاده از جایگزینی آن را در چند مرحله نوشت.

مثال: کتاب‌هایی که توسط «قربانی» ترجمه شده است.

حل: ابتدا مشخصات مترجم را از جدول author استخراج و ذخیره می‌کنیم.

سپس جدول book را با آن پیوند می‌دهیم (نیم‌پیوند).

$$\text{Temp} \leftarrow \sigma_{\text{name} = 'ترجم' \wedge \text{typ} = 'قربانی'}^{\text{(author)}} \text{book} \propto \text{Temp}$$

مثال: کارکنانی که هم نویسنده و هم مدیر نشر هستند.

حل: ابتدا کلید نویسنده‌ها را از جدول author با نام Temp1 ذخیره می‌کنیم.

سپس این جدول را با جدول کارکنان پیوند شرطی می‌دهیم تا کارکنانی که نویسنده هستند به دست آیند و در Temp2 ذخیره می‌کنیم. در پایان جدول ناشر را با این جدول دوم پیوند می‌دهیم.

$$\text{Temp1} \leftarrow \prod_{\text{ano}} \sigma_{\text{typ} = 'نویسنده'}^{\text{(author)}}$$

$$\text{Temp2} \leftarrow \text{personel} \times \text{Temp1} \\ \text{p.pno} = \text{Temp1.ano}$$

$$\text{Temp2} \propto \text{publisher}$$

۹. تقسیم

تقسیم (با علامت \div) برای بررسی همهٔ حالت‌های یک اتفاق به کار می‌رود. ابتدا بخشی که شامل شرط «همه» می‌شود (مقسوم‌علیه) را پیدا می‌کنیم، سپس بخش دیگر را بر آن تقسیم می‌کنیم. در مقسوم، حتماً باید صفت‌های مقسوم‌علیه و نیز تعدادی صفت اضافی برای خروجی وجود داشته باشد.

مثال: مدیران نشر که همهٔ کتاب‌های درسی را نشر داده‌اند.

$$\text{Temp1} \leftarrow \prod_{\text{bno}} \sigma_{\text{typ} = 'درسی'}^{\text{(book)}}$$

$$\text{Temp2} \prod_{\text{pno}, \text{bno}} (\text{publish}) \div \text{Temp1}$$

$$\text{personel} \propto \text{Temp2}$$

ابتدا کلید کتاب‌های درسی temp1 را ذخیره کردیم. سپس با عملگر تقسیم کلید ناشرهایی که همهٔ این کتاب‌ها را نشر داده‌اند، به دست آوردیم. آنگاه مشخصات کامل ناشر را از جدول personnel استخراج کردیم (با پیوند طبیعی نیز صحیح است).

توضیح: اینکه یک مدیر نشر همهٔ کتاب‌های درسی را نشر داده باشد بستگی به پایگاه‌داده ما دارد. مثلاً در دانشگاه پیام‌نور چنین گزینه‌ای بعید نیست (انتشارات پیام‌نور)، اما در وزارت علوم دور از ذهن است.

۲-۱-۴ بهینه‌سازی پرسش

یک پرسش ممکن است چند پاسخ داشته باشد. مثلاً عملگرهای پیوند طبیعی و نیم‌پیوند بسیار مشابه‌اند. همچنین می‌توان پیوند طبیعی را با پیوند شرطی معادل‌سازی کرد. بسیاری از عملگرهای معادل‌های دیگر دارند. آیا همه پاسخ‌های یک پرسش از نظر زمان و فضا معادل هستند؟ البته نه!

بعضی از عملگرها گران‌تر از بعضی دیگر هستند و باید از آن‌ها پرهیز کرد، ولی مسئله به همین‌جا ختم نمی‌شود. با جایه‌جاکردن عملگرها به پاسخ‌های مساوی متفاوتی می‌رسیم که همگی صحیح هستند. آیا از نظر زمان و فضای اشغالی حافظه نیز معادل هستند؟

مثال: کلید و نام کارکنانی که هرگز نویسنده نبوده‌اند.

پاسخ غلط:

$$\prod_{pno, personel.name} \sigma_{pno=ano \wedge author.typ='نویسنده'} (personel \times author)$$

این پاسخ غلط است، زیرا افرادی را می‌دهد که کاری جز نویسنندگی کرده‌اند (مانند مترجم) ولی ممکن است نویسنده نیز باشند. در پاسخ به چنین پرسشی فقط و فقط تفاضل (-) پاسخ‌گو است.

پاسخ ۱:

$$Temp \leftarrow \prod_{ano, name, typ, dob, sex, adres} \sigma_{typ='نویسنده'} (author) \\ \prod_{pno, personel.name} (personel - Temp)$$

پاسخ بهینه:

$$Temp \leftarrow \prod_{ano, name} \sigma_{typ='نویسنده'} (author) \\ \prod_{pno, name} (personel) - Temp$$

پاسخ‌ها را مقایسه کنید: می‌بینید که در پاسخ بهینه دو جدول کوچک‌تر، از هم منها شده‌اند.

انتظار نوشتن دستور بهینه از کاربران پایگاهداده‌ها که افرادی عادی هستند و فقط نحوه استفاده از زبان‌های پرسش مانند SQL را آموزش دیده‌اند، منطقی نیست. از سوی دیگر آمارهای پایگاهداده‌ها (اندازه جداول و غیره) در بهینه‌سازی پرسش بسیار مؤثر

است و کاربران از آن‌ها بی‌اطلاع هستند. این در حالی است که دستور غیرفنی بسیار گران تمام می‌شود و سرعت را پایین می‌آورد و بعضاً ممکن است به دلیل نیاز به فضای زیاد قابل انجام نباشد. راه حل این مشکل در دست‌های توانمند نرم‌افزار غول‌پیکر نظام مدیریت بانک اطلاعات (DBMS) نهفته است به شرطی که بتوان قواعدی برای بهینه‌سازی پرسش پیدا کرد و به صورت الگوریتم پیاده‌سازی کرد. یکی از بزرگ‌ترین امتیازهای مدل رابطه‌ای که پشتوانه نظری بسیار قوی دارد، همین است که می‌توان قواعدی برای بهینه‌سازی پرسش بیان و سپس به الگوریتم تبدیل کرد. آنچه کاربر وارد می‌کند بدون اطلاع او، ابتدا توسط این الگوریتم‌ها به دستور بهینه تبدیل و سپس به صورت بهینه اجرا می‌شود.

۱. قواعد بهینه‌سازی پرسش

در مدل رابطه‌ای برای بهینه‌سازی پرسش قواعدی وجود دارد که در زیر خلاصه می‌شود [E. F. Codd, 1972]:

اولویت ۱: گزینش را هرچه ممکن است زودتر انجام دهید.

اولویت ۲: شرط‌های ترکیبی را به شرط‌های متوالی تبدیل کنید. مثلاً می‌توان

$\sigma_{p1 \wedge p2}(e)$ را با $(\sigma_{p1}(e)) \sigma_{p2}(e)$ جایگزین کرد.

اولویت ۳: پرتو را زود انجام دهید، ولی دیرتر از گزینش.

اولویت ۴: عملگرهای ترکیبی مانند عملگرهای مجموعه‌ای و پیوند در انتها بیایند.

مثال: نام و قیمت و نوع کتاب‌های پایگاهداده که ارزان‌تر از ۱۰۰۰ هستند.

پاسخ بد:

$$\prod_{\text{title, Uprice, type}} \left(\begin{array}{c} \text{book} \times \text{publish} \\ \text{book.bno} = \text{publish.bno} \end{array} \right) \sigma_{\text{typ}='پایگاه داده' \wedge \text{Uprice} < 1000}$$

پاسخ بهتر:

$$\prod_{\text{typ, title, Uprice}} \sigma_{\text{typ}='پایگاه داده' \wedge \text{Uprice} < 1000}^{(\text{book} \times \text{publish})}$$

دقیق کنید که جداول book و publish دو ستون همنام دارند: bno و dop. اما چون مقدار هر دو ستون در دو جدول یکسان است پاسخ غلط نیست. یکبار دیگر تأکید می‌کنیم که ستون همنام به عنوان کلید خارجی، ضعف جبر رابطه‌ای است. در SQL کلید خارجی که می‌تواند همنام باشد یا نباشد، جداگانه تعریف می‌شود.

پاسخ بهینه:

$$\prod_{typ,bno,title} \sigma_{title='پایگاه داده'}^{(book)} \infty \prod_{bno,Uprice} \sigma_{Uprice < 1000}^{(publish)}$$

عملگرهای معادل در جبر رابطه‌ای بسیار هستند و جایگزینی آن‌ها تأثیر شگفتی در عملکرد پرسش دارد. در اینجا عمده‌ترین فرمول‌های معادل را می‌آوریم [H. F. Korth, A. Silberchatz 2006]

الف) تعویض‌پذیری (جایگشتی):

$$a \infty b = b \infty a$$

$$a \times b = b \times a$$

$$a \cup b = b \cup a$$

$$a \cap b = b \cap a$$

ب) شرکت‌پذیری:

$$(a \infty b) \infty c = a \infty (b \infty c)$$

$$(a \times b) \times c = a \times (b \times c)$$

$$(a \cup b) \cup c = a \cup (b \cup c)$$

$$(a \cap b) \cap c = a \cap (b \cap c)$$

معادلهای زیاد دیگری نیز وجود دارد، مانند:

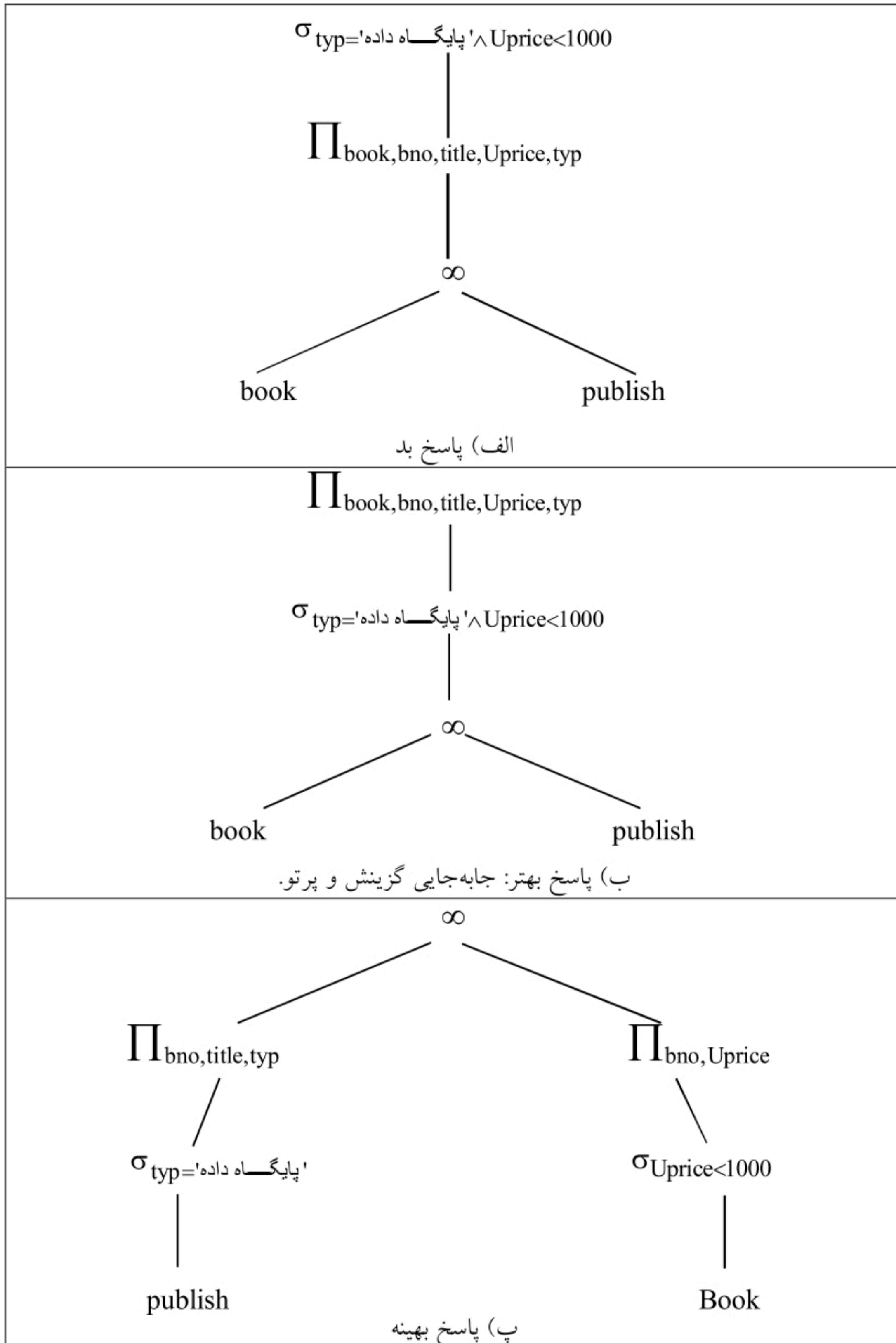
$$\sigma_p(a \cap b) = \sigma_p(a) \cap \sigma_p(b)$$

$$\sigma_p(a - b) = \sigma_p(a) - b = \sigma_p(a) - \sigma_p(b)$$

۲. الگوریتم بهینه‌سازی پرسش

پرسش‌های دریافتی از کاربران را باید ابتدا تا حد ممکن بهینه و سپس اجرا کرد. سؤال اساسی این است که چگونه می‌توان این همه اولویت و معادلهای را به صورت الگوریتم بیان کرد تا زبان‌هایی مانند SQL آن را اجرا کنند. امتیاز بزرگ جبر رابطه‌ای در ساخت این الگوریتم نهفته است. یکی از روش‌های اصلی این کار استفاده از درخت بهینه‌سازی پرسش است. برخلاف جبر رابطه‌ای، در زبان‌هایی مانند SQL عملگرهای گزینش، پرتو و پیوند و بسیاری از عملگرهای دیگر ادغام شده‌اند (در دستور SELECT) تا از تشکیل جداول میانی غیر لازم پرهیز شود. بنابراین، نمی‌توان به سادگی آن‌ها را از هم مجزا کرد و در درخت بهینه‌سازی پرسش به کار برد.

برای ساختن درخت بهینه می‌توان دستور SQL را به جبر رابطه‌ای تبدیل کرد، سپس درخت آن را رسم و آنگاه درخت را بهینه کرد و در پایان نتیجه به دست آمده را به SQL برگرداند. این روش معمول است.



شکل ۴-۲. مثال برای بهینه‌سازی پرسش

در درخت بهینه‌سازی پرسش، رابطه‌ها در برگ‌ها قرار می‌گیرند، عملگرها گره‌های میانی را تشکیل می‌دهند و درنهایت ریشه درخت که عملگر نهایی است، نتیجه پرسش را نمایش می‌دهد. با جایگزینی عملگرهاي معادل و به کارگيري اولويت‌هاي بهينه‌سازی پرسش، می‌توان عملگرهاي با اولويت بيشتر را به سمت ریشه و عملگرهاي با اولويت کمتر را به سمت برگ‌ها سوق داد و به تدریج آن را به درخت‌هاي معادل تبدیل کرد و درنهایت درخت معادل بهینه را به دست آورد.

درخت بهینه‌سازی پرسش را با استفاده از عملگرهاي ∞ (پیوند طبیعی) و Π و σ در یک مثال بررسی می‌کنیم. عملگرهاي مشابه در منابع و مراجع کتاب آمده‌اند. مثال: شماره و نام و قيمت کتاب‌هاي پایگاه‌داده که ارزان‌تر از ۱۰۰۰ هستند.

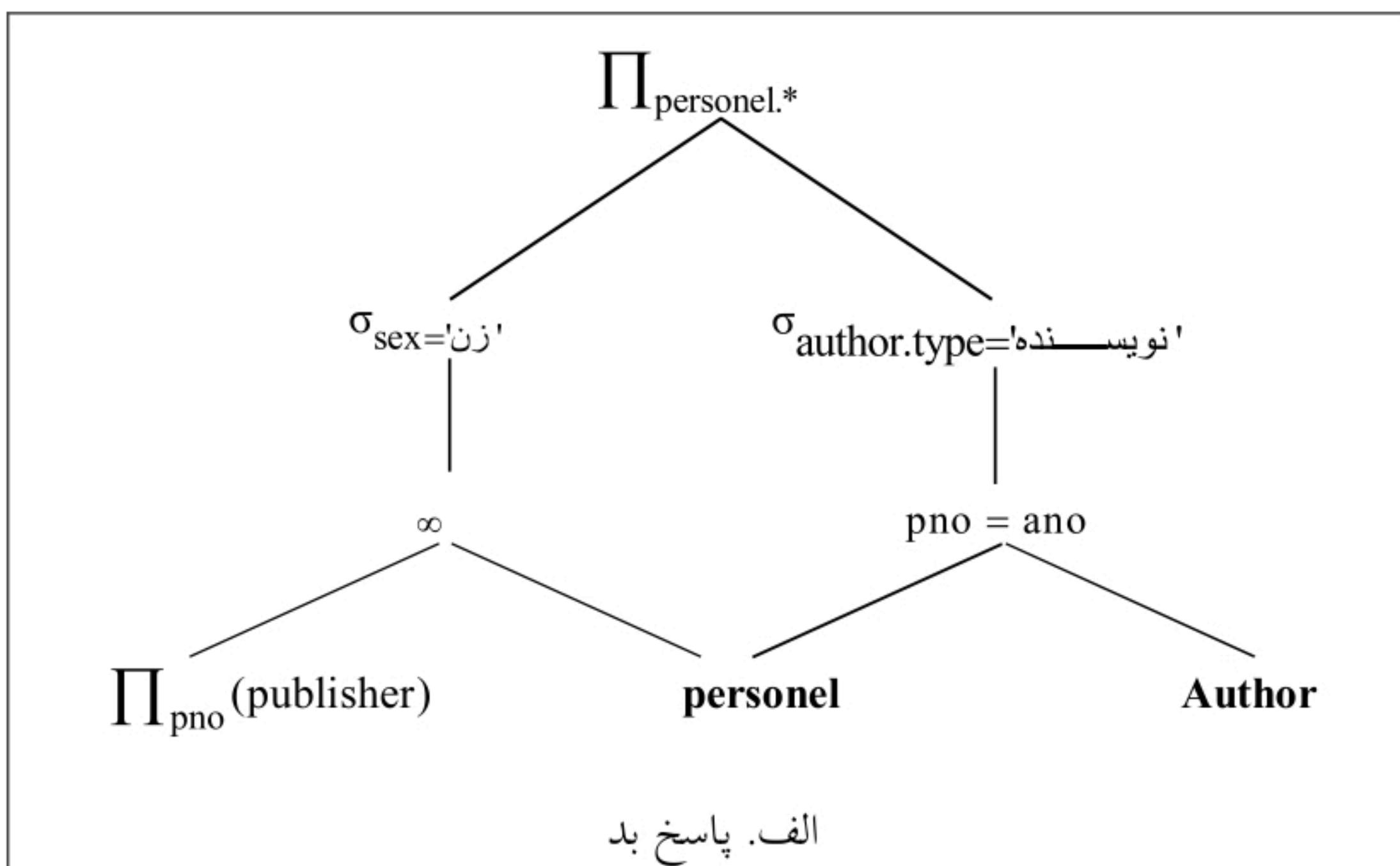
پاسخ: با توجه به خواص پیوند طبیعی، می‌توان معادل‌های مختلف با درخت‌هاي پرسش متفاوت برای اين مثال نوشت. ما از یک پاسخ بد، شروع می‌کنیم و به پاسخ بهینه می‌رسیم. شکل ۳-۴ اين مراحل را نشان می‌دهد. با توجه به گران‌بودن پیوند، باید جداول میانی را کوچک کرد و سپس پیوند داد. به عبارت دیگر باید ابتدا در سطوح پایین‌تر درخت، از هر جدول فقط ستون‌ها و سطرهایی را جدا کنیم که به آن‌ها نیاز داریم و سپس در سطوح بالاتر درخت، پیوند طبیعی را بیاوریم. این عملیات قدم‌به‌قدم انجام می‌شود تا به تدریج درخت نهایی به دست آید.

۳. بهینه‌سازی پرسش در زمان اجرا

درست است که روش به دست آوردن معادل بهینه یک پرسش را آموختیم و می‌توانیم از بین فرمول‌های معادل فرمول بهینه را انتخاب کنیم، اما باز هم چگونگی اجرای آن می‌تواند تأثیر بسیار زیادی بر زمان اجرای برنامه داشته باشد. به عنوان نمونه با استفاده از شرکت‌پذیری و تعویض‌پذیری، پیوند سه جدول را می‌توان به چندین ترتیب متفاوت محاسبه کرد:

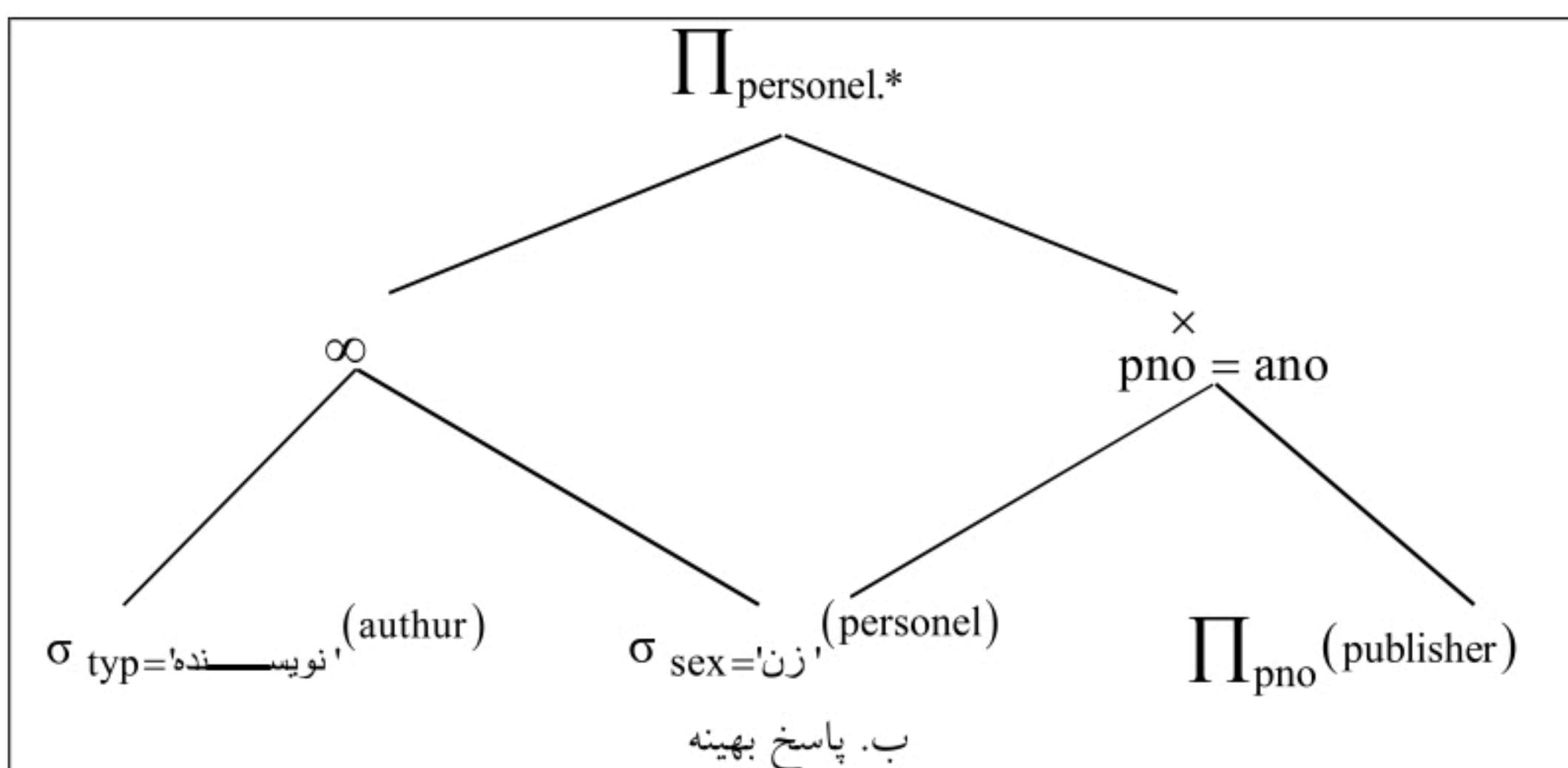
$$\begin{aligned} a \infty b \infty c &= a \infty c \infty b = \dots = c \infty b \infty a \\ a \times b \times c &= a \times c \times b = \dots = c \times b \times a \end{aligned}$$

مثال: کارکنان زن که هم نویسنده هستند و هم مدیر نشر. (شکل ۳-۴)



شکل ۴-۴. مثال دیگر برای بهینه‌سازی پرسش

عملکرد این ترتیب‌ها می‌تواند با یکدیگر بسیار متفاوت باشد، یعنی یکی، ده‌ها برابر گران‌تر از دیگری تمام شود.



ادامه شکل ۴-۴. مثال دیگر برای بهینه‌سازی پرسش

مثال: شماره کارکنی که با دو مؤلف، آدرس مشترک دارند:
یک پاسخ:

$$\text{Temp} \leftarrow \text{personel} \quad \text{personel.pno} \neq \text{author.ano} \wedge \text{personel.adres} = \text{author.adres} \quad \text{author}$$

$$\text{Temp2} \leftarrow \text{Temp} \quad \text{Temp.ano} \neq \text{author.ano} \wedge \text{Temp.adres} = \text{author.adres} \quad \text{author}$$

$$\prod_{\text{pno}} (\text{Temp2})$$

با توجه به خاصیت جابه‌جایی در پیوند، می‌توان معادل‌های دیگری نیز برای این پرسش نوشت که از جابه‌جایی جداول در پیوند بالا به دست می‌آیند. هر چند این معادل‌ها نتیجه یکسان دارند، از نظر مدت زمان اجرا می‌توانند بسیار متفاوت باشند، زیرا سایز جدول کارکنان (در یک انتشارات بزرگ) می‌تواند چندین برابر سایز جدول مؤلف باشد. بهینه‌سازی پرسش در زمان اجرا نیاز به دانستن سایز جداول دارد که مرتب‌اً عوض می‌شود. نظام مدیریت بانک اطلاعات (DBMS) می‌تواند آمار جداول را نگهداری و به‌هنگام و از آن در بهینه‌سازی پرسش استفاده کند. سؤال این است که آیا این نوع بهینه‌سازی مقرر و به صرفه است؟ پاسخ مثبت است. دستورها آنقدر با هم متفاوت هستند که نه تنها نگهداشت آمار جداول و به‌هنگام کردن آنها می‌صرفد، بلکه عملیات دیگری از قبیل شاخص‌گذاری روی جداول نیز ضروری است. نکته اصلی در انتقال داده‌ها از حافظه جانبی به حافظه اصلی و سایز بافر در کامپیوتر مربوطه نهفته است.

۳-۱-۴ به روز در آوردن داده‌ها

تغییر شمای جداول مانند اضافه کردن و حذف کردن یک یا چند ستون و تغییر دامنه ستون‌ها به جبر رابطه‌ای مربوط نمی‌شود، زیرا این قبیل تغییرات در واقع تغییر تصویر ادراکی بانک اطلاعات است. تغییر داده‌های جداول در قالب جبر رابطه‌ای می‌گنجد و سه مورد زیر را شامل می‌شود:

۱. اضافه کردن داده به جدول

این عمل معادل دستور ... SQL در `INSERT INTO` است و در جبر رابطه‌ای با استفاده از عملگرهای `U` و `←` قابل انجام است.

مثال: افزودن تمام مترجم‌ها به جدول کارکنان.

پاسخ: با توجه به ستون‌های مشترک در دو جدول کارکنان و مؤلف:

$\text{personel} \leftarrow \text{personel} \prod_{\text{bno}, \text{name}, \text{type}, \text{dob}, \text{sex}, \text{adres}} \sigma_{\text{typ} = 'مترجم'}^{(\text{author})}$

۲. حذف داده از جدول

این عمل معادل دستور `DELETE FROM ...` در SQL است و با دستورات تفاضل و جایگزینی قابل انجام است (نیاز به عملگر جدیدی ندارد). با یک دستور ممکن است یک یا چند سطر از جدول حذف شود و یا حذفی صورت نگیرد (بستگی به داده‌ها دارد).

مثال: حذف صحاف‌ها از جدول `personel`

پاسخ:

$\text{personel} \leftarrow \text{personel} - \sigma_{\text{typ} = 'صحاف'}^{(\text{personel})}$

نکته: چند بار استفاده از یک جدول در این دستور مشکلی ایجاد نمی‌کند، زیرا عملیات مختلف آن به ترتیب انجام می‌شوند و جداول میانی تشکیل می‌شود.

۳. تغییر داده‌های جدول

این عمل معادل دستور `UPDATE ... SET` در SQL است. نه سطری به جدول افزوده و نه از آن حذف می‌شود، بلکه داده‌های سطرهای موجود تغییر می‌کند. تغییر داده‌های جدول نیز نیاز به دستور جدیدی در جبر رابطه‌ای ندارد. می‌توان با تغییر شکل دستور گزینش و دستور جایگزینی این عمل را انجام داد.

مثال: افزودن هزار تومان به قیمت هر کتاب.

پاسخ:

$\sigma_{\text{Uprice} \leftarrow \text{Uprice} + 1}^{(\text{publish})}$

مثال: تغییر نام 'کرج' به 'البرز' در جداول کارکنان و فروش.

پاسخ:

$\sigma_{\text{adres} \leftarrow 'البرز'}^{(\sigma_{\text{adres} \leftarrow 'کرج'}^{(\text{personel})})}$

$\sigma_{\text{adres} \leftarrow 'البرز'}^{(\sigma_{\text{adres} \leftarrow 'کرج'}^{(\text{sell})})}$

یادآوری: نقش عملگر جایگزینی (\leftarrow) در بهروز درآوردن داده‌ها بسیار مهم است. در صورت عدم استفاده از آن، جدول مورد نظر تغییر نمی‌کند.

۲-۴ حساب رابطه‌ای دامنه‌ای

۱-۲-۴ معرفی

حساب رابطه‌ای بر دو نوع است: تاپلی و دامنه‌ای. حساب رابطه‌ای تاپلی^۱ که توسط خود آقای کاد ارائه شد [E. F. Codd 1972]، در سایر درس‌ها مانند ساختمان‌های گسته با نام‌هایی چون predicate calculus آمده‌است و نیازی به تکرار آن‌ها نیست. حساب رابطه‌ای دامنه‌ای^۲ بعدها توسط شخص دیگری ارائه شد [M. Lacroix and A. piotte, 1977]. هر دو روش از قدرت محاسباتی مساوی با جبر رابطه‌ای برخوردار هستند، ولی حساب رابطه‌ای دامنه‌ای، ساده‌تر می‌کند.

شكل کلی عبارت در حساب رابطه‌ای دامنه‌ای چنین است:

$$\{ < c_1, c_2, \dots, c_n > | P(c_1, c_2, \dots, c_n, c_{n+1}, \dots) \}$$

این دستور ستون‌های c_1, c_2, \dots, c_n را می‌دهد به‌طوری‌که شرط $P(c_1, c_2, \dots, c_n, c_{n+1}, \dots)$ برقرار باشد.

رعایت قواعد زیر در حساب رابطه‌ای دامنه‌ای الزامی است:

- خروجی دستور شامل همه ستون‌های $< c_1, c_2, \dots, c_n >$ خواهد بود.
- در شرط $(c_1, c_2, \dots, c_n, c_{n+1}, \dots) P$ می‌توان از متغیرهای دیگری نیز استفاده کرد که با \exists یا \forall تعریف می‌شوند (همانند ساختمان‌های گسته). استفاده از ثابت‌ها نیز در صورتی که متعلق به دامنه متغیرهای مربوطه باشند، ایرادی ندارد.
- ارتباط متغیرها به جداول با علامت \in انجام می‌شود و شرط همتایی باید رعایت شود.
- ترکیب شرط‌ها همانند جبر بول با \wedge و \vee و \neg و \Rightarrow و ... انجام می‌شود.
- علامت عدم تعلق (\notin) تعریف نشده‌است، ولی می‌توان با استفاده از \in و معادل‌سازی کرد.

- همانند جبر رابطه‌ای، برای پیوند جداول از متغیرهای همنام در آنها استفاده می‌شود.
 - گاهی می‌توان با تغییر جملات پرسش، پاسخ‌گویی به آن را آسان‌تر کرد.
 - از فرمول‌های معادل در نظریه مجموعه‌ها برای تبدیل عبارت‌ها استفاده می‌شود. مثلاً برای معادل‌سازی $A - B$ از $A \cap B$ و برای $A \rightarrow B$ از $A \cup B$ استفاده می‌شود. در این معادل‌سازی، عملگرهایی مانند تفریق که مربوط به جبر رابطه‌ای هستند و در حساب رابطه‌ای جایی ندارند، حذف می‌شوند و به جای آنها عملگرهای معادل می‌آیند.
- توضیح: شناخت علائم استفاده شده در بالا، دانسته فرض می‌شود.

۴-۲-۴ انواع دستورها

مثال: کارکنانی که تاریخ تولد آنها بعد از سال ۱۳۵۰ است.

$$\{<pn, n, t, d, s, a> | <pn, n, t, d, s, a> \in \text{personel} \wedge d > 1350\}$$

مثال: نام کارکنانی که تاریخ تولد آنها بعد از سال ۱۳۵۰ است.

$$\{<n> | \exists pn, t, d, s, a (<pn, n, t, d, s, a> \in \text{personel} \wedge d > 1350)\}$$

توضیح: شرط همتایی، یعنی تعلق سطرهای شش‌ستونی به جدول personel، با معرفی متغیرهای جدید رعایت شده است.

مثال: مترجم‌هایی که مدرک دکتری دارند.

$$\left\{ <an, n, t, d, s, ad, de, m, r> | <an, n, t, d, s, ad, de, m, r> \in \text{author} \wedge t = \text{'دکتری'} \wedge de = \text{'مترجم'} \right\}$$

مثال: کتاب‌هایی که قیمت آنها بالای ۱۰۰۰ است.

$$\{<b, t, d, ty, v, a> | \{<b, t, d, ty, v, a> \in \text{book} \wedge \exists pu, p, dl, u, n (<pu, p, b, dl, u, n> \text{publish} \wedge u > 1000)\}$$

توضیح: به دو جدول کتاب و انتشار نیاز داریم. متغیر مشترک b دو جدول را به هم پیوند می‌دهد. به محل باز و بسته‌شدن پرانتز دقت شود. متغیرها باید در نزدیک‌ترین محلی که مورد استفاده قرار می‌گیرند، تعریف شوند و حوزه عملکرد آنها با پرانتز مشخص می‌شود. حوزه عملکرد متغیرهای خروجی تمام دستور است.

دستور:

مثال: کتاب‌هایی که توسط انتشارات امیرکبیر منتشر شده‌اند.

$$\{< b, t, d, ty, v, a > | < b, t, d, ty, v, a > \in book \wedge \exists pu, p, dl, u, n$$

$$(< pu, p, b, dl, u, n > \in publish \wedge \exists t2, ty2, d2, a2$$

$$(< p, t2, ty2, d2, a > \in publisher) \wedge t2 = 'امیرکبیر'$$

توضیح: چون دو جدول کتاب و ناشر در یکدیگر کلید خارجی ندارند، باید از جدول انتشار نیز برای پیوند آن‌ها استفاده کرد. به محل بسته‌شدن پرانتز اول دقت کنید. چون p برای این پرانتز تعریف شده‌است، باید در اولین جای ممکن که دیگر مورد استفاده قرار نمی‌گیرد، پرانتز را بیندیم.

مثال: نویسنده‌گانی که همه کتاب‌های فلسفه را نوشته‌اند.

حل: این پرسش با قید «همه» است و می‌توان آن را به دو صورت پاسخ داد: با معادل‌سازی یا استفاده از \forall .

پاسخ ۱: با معادل‌سازی

نویسنده‌گانی که هیچ کتاب فلسفه‌ای در این پایگاهداده وجود ندارد که مؤلف دیگری نوشته باشد.

$$\{ < an, n, t, d, s, ad, de, m, r > | < an, n, t, d, s, ad, de, m, r >$$

$$\in author \wedge t = 'نویسنده' \wedge \neg \exists b, t, d, ty, v, al$$

$$(< b, t, d, ty, v, al > \in book \wedge al \neq an \wedge ty = 'فلسفه')$$

توضیح: در اینجا به دنبال نویسنده‌گانی می‌گردیم که کتاب فلسفه‌ای وجود نداشته باشد که مؤلف دیگری تألیف کرده باشد. معادل‌سازی در حساب رابطه‌ای خیلی دقیق و ظریف است و نباید از معادلهای غلط مشابه استفاده کنیم. ترکیب \neg به جای علامت \neq برای عدم تعلق به کار رفته است. \neg در جبر رابطه‌ای جایی ندارد، زیرا با داده‌هایی که در جدول وجود ندارند کاری نداریم.

پاسخ ۲: با استفاده از \forall . به کاربرد \Rightarrow دقت شود.

$$\{ < an, n, t, d, s, ad, de, m, r > | < an, n, t, d, s, ad, de, m, r >$$

$$\in author \wedge t = 'نویسنده' \wedge \forall b, t, d, ty, v, al$$

$$((< b, t, d, ty, v, al > \in book \wedge ty = 'فلسفه') \Rightarrow al = an)$$

مثال: کتاب‌هایی که همه ناشرها آن را چاپ کرده‌اند.

$$\{< b, t, d, ty, v, a > | < b, t, d, ty, v, a > \in book \wedge \forall pu, p, dl, u, n \\ (tl, tyl, dl, al \in publisher)))\}$$

۴-۲-۳ خطر حلقة بی‌پایان

عبارت‌های حساب رابطه‌ای دامنه‌ای، که در آن‌ها از \neg استفاده شود، می‌توانند حلقة بی‌انتها تولید کنند و خروجی آن‌ها بی‌پایان باشد. به عبارت زیر توجه کنید:

$$\{< b, t, d, ty, v, a > | \neg(< b, t, d, ty, v, a > \in book)\}$$

این عبارت می‌گوید «تمام جداول شش‌ستونی ممکن، به غیر از جدول book را بدء، که ستون‌های آن‌ها هم‌دامنه باشند». خروجی این دستور، بی‌پایان است.

در فرمول‌هایی که در آن‌ها از \exists و \forall استفاده می‌شود، دو خطر وجود دارد:

۱. تعداد بی‌پایان مقایسه و ...

۲. خروجی بی‌پایان

به عبارت زیر توجه کنید:

$$\{< b, t, d, ty, v > | \exists a (\neg(< b, t, d, ty, v, a > \in book))\}$$

این عبارت علاوه‌بر اینکه خطر خروجی بی‌پایان را شبیه مثال فوق دارد، باید برای تعداد بی‌پایانی رشته که می‌توانند به جای نام مؤلف بنشینند، مقایسه را انجام دهد. در نوشتن عبارت‌های حساب رابطه‌ای دامنه‌ای، باید به این مهم توجه کرد.

برای پیاده‌سازی حساب رابطه‌ای دامنه‌ای، کوشش‌هایی صورت گرفته که قدیمی‌ترین آن‌ها^۱ M. M. Zloof] QBE [1977] و موفق‌ترین آن‌ها زبان پرسش^۲ Datalog است. در زبان Datalog، رابطه‌ها و ارتباط بین آن‌ها مشابه آنچه در این بخش گفته شد، با فرمی متفاوت بیان می‌شوند.

۴-۲-۴ مقایسه با جبر رابطه‌ای و SQL

جبر و حساب رابطه‌ای از نظر قدرت محاسباتی معادل یک‌دیگر هستند. زبان بیانی (اعلانی)^۳ SQL که پیاده‌سازی آزاد جبر رابطه‌ای است، از نظر محاسباتی کامل

1. Query by Example

2. Database Logic

3. Declarative

نیست، یعنی اطلاعاتی در جداول وجود دارد که نمی‌توان آن‌ها را استخراج کرد (SQL3 از نظر محاسباتی کامل است). با این همه، جبر و حساب رابطه‌ای از آن هم ضعیفتر هستند. مهم‌ترین کاستی آن‌ها در مقایسه با SQL توابع محاسباتی و گروه‌بندی داده‌ها است.

مثال: کتاب‌هایی که در هیچ تاریخی هزینه چاپ کمتر از ۱۰۰۰۰۰ نداشته‌اند.

برای پاسخ‌گویی به این پرسش به دو مورد زیر نیاز است:

۱. گروه‌بندی کتاب‌ها بر اساس تاریخ انتشار

۲. محاسبه هزینه در هر تاریخ

هیچ‌کدام از این دو مورد در جبر و حساب رابطه‌ای مستقیماً پیش‌بینی نشده‌است. بزرگ‌ترین امتیاز حساب رابطه‌ای دامنه‌ای نسبت به جبر رابطه‌ای و SQL پشتیبانی مستقیم از بانک دانش^۱ است.

خوانندگان علاقه‌مند به حساب رابطه‌ای تاپلی را به [روحانی، ۱۳۹۲] ارجاع می‌دهیم و یادآور می‌شویم که این روش فرق زیادی با predicate calculus که در دوره‌های مهندسی کامپیوتر تدریس می‌شود، ندارد.

خلاصه فصل چهارم

در این فصل به مبانی نظری مدل رابطه‌ای می‌پردازیم. آقای کاد از ابتدا مدل رابطه‌ای را در قالب مفاهیم نظری بیان کرد و پس از آن پژوهش‌گران بی‌شماری به این مقوله علاقه نشان دادند. جبر رابطه‌ای اولین و مهم‌ترین قالبی بود که مدل رابطه‌ای توسط آقای کاد در آن ریخته شد. سپس حساب رابطه‌ای به کار گرفته شد و به شکل‌های متفاوتی مفاهیم مدل رابطه‌ای را ارائه کرد. عمده‌ترین هدف بیان نظری یک مدل، اثبات صحت ویژگی‌های آن است. مثلاً اینکه گفته می‌شود تراکنش‌های ACID جامعیت پایگاه‌داده را تضمین می‌کنند، باید اثبات شود. این اثبات‌ها تنها و تنها از طریق مفاهیم نظری انجام‌پذیر است. در این بخش عمده‌ترین ساختارها و مفاهیم مدل رابطه‌ای در قالب جبر رابطه‌ای تعریف و بهینه‌سازی آن‌ها ارائه شده‌است. این مفاهیم در واقع عملگرهای مدل رابطه‌ای هستند که ساختار رابطه‌ها، بارگذاری آن‌ها، استخراج اطلاعات از آن‌ها و

به روز نگهداری آنها را شامل می‌شود. حساب رابطه‌ای دامنه‌ای نیز ارائه شده که از دیدگاه متفاوتی به مدل رابطه‌ای می‌نگرد و برخلاف جبر رابطه‌ای و SQL، عملگرهای معمول رابطه‌ای در آن جایگاهی ندارند و توسط عملگرهای ساده ریاضی، شبیه‌سازی می‌شوند. جبر و حساب رابطه‌ای از قدرت محاسباتی یکسانی برخوردارند.

خودآزمایی‌های تشریحی فصل چهارم

در این بخش با استفاده از جدول‌های مربوط به آموزش رانندگی که در فصل‌های گذشته معرفی شد، چند مثال مطرح و پاسخ آنها در جبر و حساب رابطه‌ای ارائه می‌شود.
الف) آموزش رانندگی

مثال: کارآموزان و کد آموزشگاه‌هایی که در آن ثبت نام کرده‌اند.

جبر رابطه‌ای: چون دو جدول student و academy و کلید خارجی در یکدیگر ندارند، به جدول admission برای برقراری ارتباط بین آنها نیاز است.

پاسخ غلط: $\prod_{\text{student} \cdot \text{ac} \#} (\text{academy} \bowtie \text{admission} \bowtie \text{student})$

این پاسخ غلط است، زیرا ستون‌های زیادی همنام هستند؛ مانند name و doe و adres

پاسخ صحیح:

$\text{student} \bowtie \prod_{\text{st}\#, \text{ac}\#} (\text{admission}) \bowtie \prod_{\text{ac}\#} (\text{academy})$

حساب رابطه‌ای:

$\{\langle \text{st}, \text{n}, \text{do}, \text{s}, \text{ad} \rangle | \langle \text{st}, \text{n}, \text{do}, \text{s}, \text{ad} \rangle \in \text{student} \wedge \exists \text{as}, \text{ac}, \text{tr}, \text{d2}, \text{do2}, \text{du}, \text{c} (\langle \text{as}, \text{st}, \text{ac}, \text{tr}, \text{d2}, \text{do2}, \text{du}, \text{c} \rangle \in \text{admission} \wedge \exists \text{co}, \text{n3}, \text{d3}, \text{ad3} (\langle \text{ac}, \text{co}, \text{n3}, \text{d3}, \text{ad3} \rangle \in \text{academy}))\}$

مثال: کارآموزانی که بیش از ۳ ثبت نام داشته‌اند.

پاسخ: این پرسش و هر پرسش دیگری که شمارش یا محاسبه‌ای در آن باشد را در جبر و حساب رابطه‌ای نمی‌توان پاسخ داد.

مثال: کارآموزانی که در بیش از یک آموزشگاه ثبت نام کرده‌اند.

جبر رابطه‌ای:

$$\begin{aligned} t1 &\leftarrow \prod_{\text{st}, \text{ac}\#} (\text{admission}) \\ t2 &\leftarrow (\rho_s^{\text{student}} \bowtie t1) \underset{\text{s.st}\# = \text{t2.st}\# \wedge \text{t1.ac}\# \neq \text{t2.ac}\#}{\times} t2 \end{aligned}$$

حساب رابطه‌ای:

$$\{<st, n, do, s, ad> | <st, n, do, s, ad> \in student \wedge \exists as, ac, tr, d, do, du, c \\ (<as, st, ac, tr, d, do, du, c> \in admission \wedge \exists as2, ac2, tr2, d2, do2, du2, c2 \\ (<as2, st, ac2, tr2, d2, do2, du2, c2> \in admission \wedge ac \neq ac2))\}$$

مثال: کارآموزانی که فقط در یک آموزشگاه ثبت نام کرده‌اند.

جبر رابطه‌ای: کافی است نتیجه پرسش بالا را از جدول کارآموز منها کنیم.
یادآوری می‌شود که عملگر منها قابل جایگزینی با هیچ عملگر دیگری نیست.

حساب رابطه‌ای:

$$\{<st, n, do, s, ad> | <st, n, do, s, ad> \in student \wedge \exists as, ac, tr, d, do, du, c \\ (<as, st, ac, tr, d, do, du, c> \in admission \wedge \neg(\exists as2, ac2, tr2, d2, do2, du2, c2 \\ (<as2, st, ac2, tr2, d2, do2, du2, c2> \in admission \wedge ac \neq ac2)))\}$$

تمرین‌های تشریحی فصل چهارم

۱. با استفاده از جداول پایگاهداده نشر کتاب به پرسش‌های زیر در جبر و حساب رابطه‌ای پاسخ گویید:

- کارکنان زن که بعد از سال ۱۳۷۰ متولد شده‌اند.
- کتاب‌های از نوع «اجتماعی» که مؤلف آن‌ها نامشخص است.
- ناشرهایی که بعد از سال ۱۳۷۰ کتابی نشر نداده‌اند.
- تاریخ انتشار کتاب‌هایی که مؤلف آن «اکبر عبدی» است.
- مؤلفهایی که در رشته «نرم‌افزار» دارای مدرک «دکتری» هستند، ولی رزومه آن‌ها تهی است.
- کتاب‌های «اکبر عبدی» که توسط انتشارات امیرکبیر منتشر شده‌است.
- کتاب‌های از نوع «تاریخ» که توسط انتشارات امیرکبیر منتشر شده‌است.
- انتشارات واقع در «تهران» که در خارج از این شهر، انتشارات همنام دارند.
- کتاب‌هایی که در نوع مختلف با نام یکسان و مؤلف متفاوت منتشر شده‌اند.
- کتاب‌هایی که بعد از سال ۱۳۹۰ با قیمت بالای ۱۰۰۰۰۰ توسط انتشارات امیرکبیر منتشر شده‌است.

۲. با استفاده از جداول آموزش رانندگی به پرسش‌های زیر در جبر و حساب رابطه‌ای پاسخ گویید:

- اطلاعات آموزشگاه‌های رانندگی که تاریخ تأسیس آن‌ها بین ۱۳۸۰ تا ۱۳۹۰ بوده است.

- نام کارآموزان و آموزشگاه‌هایی که در آن ثبت نام کرده‌اند.
- نام مربيان و آموزشگاه‌ها و کد آموزشگاه.
- نام مربيانی که فقط در یک آموزشگاه آموزش داده‌اند.
- مربيانی که در آموزشگاه‌های مختلف آموزش داده‌اند.
- آموزشگاه‌هایی که قبل از سال ۹۴ تأسیس شده‌اند.
- مربيانی که طول دوره آن‌ها بیش از ۳۰ ساعت طول کشیده است.
- تعداد کارآموزانی که ساکن «خیابان پروین» هستند.
- بیشینه طول دوره در پذیرش کارآموزان.
- نام کارآموزانی که با «علی محمدی» آموزش داشته‌اند.

۳. با استفاده از جداول کلینیک دندانپزشکی به پرسش‌های زیر در جبر و حساب رابطه‌ای پاسخ گویید:

- دندانپزشکانی که در سال‌های ۱۳۷۰ تا ۱۳۸۰ استخدام و ترک کار داشته‌اند.
- منشی‌هایی که برای «دکتر داودی» پذیرش انجام داده‌اند.
- بیمارانی که در این «کلینیک خانواده» پذیرش شده‌اند.
- بیمارانی که فقط توسط یک پزشک پذیرش شده‌اند.
- بیمارانی که توسط پزشک‌های مختلف پذیرش شده‌اند.