

## مشکلات بهنگام سازی دیدهای رابطه‌ای

سیدمحمد تقی روحانی رانکوهی

عضو هیئت علمی دانشکده مهندسی برق و کامپیوتر دانشگاه شهید بهشتی

### چکیده

در این مقاله، مشکلات موجود در عملیات بهنگام سازی دیدهای رابطه‌ای<sup>۱</sup> و موارد ناممکن این عملیات، مورد تحلیل قرار می‌گیرند. ضمن مرور اجمالی کارهای پژوهشگران در این زمینه، تعریف صوری<sup>۲</sup> دید ارائه و مزایای این مفهوم یادآوری می‌شود. در این بررسی، دیدهای رابطه‌ای به سه رده اساسی تقسیم می‌شوند: دیدهای بی‌مشکل، دیدهای دارای مشکل و دیدهای غیرقابل بهنگام سازی. با توجه به تنوع مکانیسمهای اشتقاق دید از رابطه(های) مبنا<sup>۳</sup>، نشان داده می‌شود که علیرغم پژوهشهای انجام شده، بیشتر دیدها از رده دوم یا سوم هستند. عدم قابلیت بهنگام سازی<sup>۴</sup> اکثر دیدها نقطه ضعف مهمی در مدل رابطه‌ای است که هنوز برطرف نشده است.

مفاهیم اصلی: مدل رابطه‌ای، دید، شمای ادراکی (مفهومی)، شمای خارجی، کلید اصلی، کلید خارجی، جبر رابطه‌ای، قواعد جامعیت، وابستگی تابعی<sup>۵</sup>.

### ۱- مقدمه

مشکلات جدی دارد که هنوز به تمامی برطرف نشده‌اند (DELO 95). این مشکلات نقطه ضعف قابل توجهی در مدل رابطه‌ای و طبیعتاً در سیستمهای رابطه‌ای است.

مشکل بهنگام سازی دیدهای رابطه‌ای از همان اوان مطرح شدن مدل رابطه‌ای و سپس با توسعه سیستمهای رابطه‌ای، توسط پژوهشگران بررسی شده است. البته در متون درسی پایگاه داده‌ها به این مشکلات یا اساساً پرداخته نشده و یا به اختصار به آن اشاره شده است. غیر از دید در (DATE 86, DATE 90 و DATE 95) و کساد در (CODD 90)، دیگران چندان توجهی به موضوع در کتب درسی نداشته‌اند. در این مراجع راه‌حلهایی ارائه شده است. با توجه به همین راه‌حلهاست که بعضی از دیدها در رده دیدهای بی‌مشکل قرار می‌گیرند. اما علیرغم پژوهشهای انجام شده، مشکل بهنگام سازی

دید یا نما مفهومی است همگانی<sup>۶</sup> در معماری پایگاه داده‌ها و هدف اصلی از پیاده سازی آن تامین سطح خارجی برای کاربران است آنگونه که ANSI پیشنهاد کرده است. این مفهوم مهم مزایایی دارد که در جای خود یادآوری خواهد شد.

در مدل رابطه‌ای انجام عملیات از سطح خارجی در دید کاربر، تا آنجا که به عملیات بازیابی مربوط می‌شود (اقتلاً از نظر تئوریک) با دشواری عمده‌ای مواجه نیست. به عبارت دیگر تبدیل (نگاشت)<sup>۷</sup> احکام بازیابی از دید به احکام عمل‌کننده در سطح ادراکی<sup>۸</sup>، مشکلی ندارد. البته در سیستمهای مدیریت پایگاه داده‌های رابطه‌ای موجود، محدودیتهایی (حتی در خود SQL استاندارد موسوم به SQL/92 (DATE 95)) در عمل بازیابی از دید وجود دارد. اما انجام عملیات بهنگام سازی (درمعنای عام یعنی درج، حذف و تغییر) در دیدهای رابطه‌ای

1 - Relational Views Updating

2 - Formal

3 - Base Relation

4 - Updatability

5 - Functional Dependency

6 - Universal

7 - Mapping

8 - Conceptual

**تعریف:** فرض میکنیم  $S$  یک شمای رابطه‌ای باشد یعنی مجموعه‌ای از چکیده<sup>۱</sup> (مجموعه عنوان) تعدادی رابطه مبنا با نامهای  $R_1, R_2, \dots, R_m$  به انضمام مجموعه‌ای از قواعد جامعیت ناظر بر رابطه‌ها. اگر  $J = 1, 2, \dots, n, A_i$  اسامی صفات خاص<sup>۴</sup> رابطه  $R_i$  باشند، داریم:

$$R_i = \{A_j \mid J = 1, 2, \dots, n\}, i=1, 2, \dots, m$$

$$S = \{R_i, C_{ik} \mid i=1, 2, \dots, m; K=1, 2, \dots, K\}$$

که  $m, n, k \in \mathbb{N}$  و  $C_{ik}$  مجموعه تمام قواعد جامعیت ناظر بر رابطه  $R_i$  است.

اگر  $r_i$  بسط رابطه  $R_i$  در یک لحظه خاص باشد، این بسط در واقع حالت یا وضعیت رابطه  $R_i$  را در یک لحظه از حیات پایگاه نشان می‌دهد.

اگر  $S_q$  حالت یا وضعیت پایگاه در یک لحظه خاص باشد، به ازاء  $q \in \mathbb{N}$  داریم:

$$S_q = \{r_{iq} \mid i = 1, \dots, m, m \in \mathbb{N}\}$$

توجه داریم که پایگاه در هر لحظه یک حالت دارد یعنی  $s_1, s_2, \dots, s_q$ ، زیرا هر رابطه در هر لحظه یک بسط خاص دارد:  $r_{i1}, r_{i2}, \dots, r_{iq}$  و بطور کلی  $r_{iq}$  نمایشگر تمام بسطهای ممکن رابطه  $r_i$  است. بنابراین:

$$S_1 = \{r_{i1} \mid i = 1, 2, \dots, m, m \in \mathbb{N}\}$$

$$S_2 = \{r_{i2} \mid i = 1, 2, \dots, m, m \in \mathbb{N}\}$$

حال فرض می‌کنیم  $V$  نمایشگر شمای خارجی باشد، یعنی مجموعه تعریف دیدهای کاربران می‌توان نوشت:

$$V = \{V_i, C_{ik} \mid i = 1, 2, \dots, m \\ K = 1, 2, \dots, K, m, K \in \mathbb{N}\}$$

که در آن  $V_i$  چکیده (مجموعه عنوان) یک رابطه مجازی یا دید است و  $C_{ik}$  مجموعه قواعد جامعیت ناظر بر  $V_i$ ، که معمولاً زیرمجموعه‌ای از  $C_{ik}$  است. به عبارت دیگر دید به عنوان یک رابطه مشتق شده از یک

دیدها در همه انواع آن هنوز حل نشده است. پژوهشهای انجام شده را می‌توان در مقالات و معدود کتابهای پژوهشگران زیر یافت:

بانسیلهون، اسپراتوس، دایال و پرایس، کالر، کاسماداکیس و پاپامتریو، چنگ و یاناکوداکیس، شیت و لارسون، لانگراک، دیت، دایال و برنستاین، کاد.

در این پژوهشها، عمدتاً به مواردی از بهنگام سازی دیدهای پرداخته شده است که در تعریف آنها از عملگر<sup>۱</sup> تحدید<sup>۲</sup> یا پرتو<sup>۳</sup> به شرط حفظ کلید، پیوند<sup>۴</sup> PK-PK<sup>۵</sup> و PK-FK<sup>۶</sup> استفاده می‌شود. هرچند در این دو حالت پیوند بویژه حالت PK-FK نیز مشکلاتی وجود دارد که خواهیم دید. اخیراً کاد و بویژه دیت، هریک مستقلاً راه‌حلهایی برای حالاتی که در آنها عملگر در تعریف دید عملگر اجتماع، اشتراک و تفاضل باشد برنهاده‌اند که مشکل بهنگام سازی این گونه دیدها را در بعض حالات خاص و با رعایت بعض اصول پیشنهادی، رفع می‌کنند (CODD 90, DATE 94, DATE 95) ولی راه‌حلها عمومیت ندارند.

## ۲- تعریف دید در مدل رابطه‌ای و یادآوری مزایای آن

در مدل رابطه‌ای، دید خود نوعی رابطه است، البته یک رابطه مجازی به این معنا که فاقد استقلال وجودی است و حتماً باید روی یک یا بیش از یک رابطه مبنا تعریف شود. در واقع هر دید یک مکانیسم اشتقاق مجازی دارد که در اساس به کمک عباراتی از جبر رابطه‌ای یا محاسبات رابطه‌ای تعریف می‌شود. این عبارت یا عبارات منطقی، ماهیتاً نوعی پرس و جو<sup>۷</sup> هستند که روی رابطه یا رابطه‌های مبنا مطرح می‌شوند (البته این پرس و جو در مرحله تعریف دید<sup>۸</sup> به اجرا در نمی‌آید بلکه صورت درونی آن در کاتالوگ سیستم نگهداری می‌شود تا دیرتر مورد مراجعه و استفاده قرار گیرد).

1 - Operator

2 - Select

3 - Project

4 - Join

5 - Primary Key-Primary Key

6 - Primay Key-Foreign Key

7 - Query

8 - View Definition Time

9 - Intention

یا چند رابطه مبنا، قواعد جامعیت ناظر به آن رابطه‌ها را به ارث می‌برد، هرچند ممکن است قواعد خاص خود را نیز داشته باشد که طبعاً نباید متناقض با مجموعه قواعد جامعیت ناظر بر رابطه (های) مبنا باشند. این قواعد جامعیت یه به عبارت دیگر محدودیتها ممکن است پویا یا ایستا باشند (BANCILHON 81).

می‌توان نوشت:

$$V_i = OP(s'), s' \subseteq s \\ i = 1, 2, \dots, m \quad m \in N$$

یعنی هر دید حاصل عملکرد یک عملگر یا ترکیبی از عملگرهای جبر رابطه‌ای یا محاسبات رابطه‌ای روی بخشی از شمای ادراکی یا همه آنست.

اگر  $V_i$  حالت یا وضعیت رابطه مجازی  $V_i$  در یک لحظه خاص باشد، به ازاء  $q \in N$  داریم:

$$V = OP(s'_q), s'_q \subseteq s \\ i = 1, 2, \dots, m \quad m \in N$$

که  $op$  یک عملگر تعریف شده است. بنابراین رابطه مجازی  $V_i$  (بسط  $V_i$ )، یک اشتقاق از بسط یک یا بیش از یک رابطه میناست در یک لحظه خاص و هر رابطه در آن لحظه بسط خاص دارد (این بسط در زمان تعریف با بسط در زمان اجرای درخواست کاربر می‌تواند متفاوت باشد).

عملگر  $op$  در تعریف دید می‌تواند یکی از عملگرهای زیر یا هر ترکیب ممکن از آنها باشد:

- \* تابع جمعی<sup>۱</sup> که روی مقادیر یک یا بیش از یک  $A_i$  از رابطه  $R_i$  عمل می‌کند.
- \* عملگر گزینش یا تحدید جبری یک رابطه
- \* عملگر پرتو جبری از یک رابطه
- \* عملگر پیوند دو رابطه
- \* عملگر اجتماع دو رابطه
- \* عملگر اشتراک دو رابطه
- \* عملگر تفاضل دو رابطه

- \* عملگر ضرب دو رابطه
- \* عملگر تقسیم دو رابطه
- \* سوره‌های<sup>۲</sup> وجودی و عمومی موجود در محاسبات رابطه‌ای

و بطور کلی همه عملگرهای معمولی و پیشرفته که کاد در مدل رابطه‌ای خود تعریف کرده است (90 CODD) اعم از عادی یا بیرونی<sup>۳</sup> و هنوز بسیاری از آنها در همان حد پیشنهاد تئوریک باقی مانده‌اند و در هیچ RDBMS واقعی پیاده سازی نشده‌اند.

در بحث ما عملگر  $op$  از اهمیت خاصی برخوردار است، زیرا با توجه به ماهیت  $op$ ، نوع مشکلی که، در بهنگام سازی دید بروز می‌کند، می‌تواند متفاوت باشد و در بعضی از انواع  $op$ ، اساساً دید غیرقابل بهنگام سازی می‌شود.

و اما دید مزایایی دارد از جمله:

\*تامین کننده محیط انتزاعی برای کاربران خارجی

\*تسهیل کننده واسط کاربری با پایگاه

\*تامین کننده پویایی بالا در تعریف پایگاه توسط کاربر

\*تامین کننده وضوح کاربری

\*تامین کننده اشتراک داده‌ای

\*تامین کننده استقلال داده‌ای منطقی<sup>۴</sup> (البته نه، به طور کامل، به دلیل وجود مشکلات، در بهنگام سازی)

\*تامین کننده نوعی مکانیسم خودکار ایمنی پایگاه بخاطر مزایای قابل توجه مفهوم دید، اکثر سیستمهای رابطه‌ای، آنرا به نحوی ارائه کرده‌اند. اما نکته اساسی اینست که برای آنکه کاربران بتوانند از مزایای دید به طور کامل استفاده نمایند باید امکان انجام عملیات بازیابی و ذخیره سازی از یا در دید را به نحو گسترده داشته باشند و تنها در این صورت است که این مفهوم می‌تواند کاملاً مفید باشد.

همانطور که پیشتر اشاره شد، در عملیات بازیابی از دید مشکل مهمی وجود ندارد، اما بعلت وجود دشواریهای جدی در عملیات بهنگام سازی اکثر انواع

متناظر با یک ستون از جدول مبنا باشد. توضیح اینکه در سیستم R، بجای مفهوم رابطه، از اصطلاح جدول استفاده می شود). در تعریف چنین دیدی،  $op$ ، عملگر تحدید یا ترکیبی است از تحدید و پرتو، یعنی اگر A مجموعه عنوان یک رابطه باشد، به ازاء  $i \in N$  دلخواه داریم:

$$V_i = R_i \text{ Where Conditions}$$

یا

$$V_i = (R_i \text{ Where Conditions}) [A'_j] \quad A'_j \subseteq A', \quad A' \subseteq A, \quad j \in N$$

در نمایش جدولی می گوئیم که بسط  $V_i$  از  $V_j$ ، زیر مجموعه ای افقی - عمودی از یک جدول مبنا (از یک  $r$ ) است و فرض بر این است که کلید اصلی  $R_i$  در دید وجود دارد.

حتی این دید ساده هم مشکل دارد زیرا مثلاً در عمل درج، پدیده هیچمقدار<sup>۱</sup> در رابطه مبنایی که دید روی آن تعریف شده است، بروز می کند و می دانیم که این پدیده در سیستمهای رابطه‌ای باعث بروز مشکلاتی می شود. (مطلب ۲-۲-۱ را ببینید)

### ۲-۳- دیدهای دارای مشکل

دیدهای دارای مشکل آنهایی هستند که در تعریف آنها،  $op$ ، عملگر تحدید، پرتو، پیوند، اجتماع، اشتراک و تفاضل یا ترکیبی از این عملگرها، با شرایط خاص و محدود کننده، نباشد. با توجه به پویایی و گوناگونی مکانیسم اشتقاق دید، می توان گفت که بسیاری از دیدها از رده دیدهای دارای مشکل هستند. اهم مشکلاتی که در بهنگام سازی دیدها بروز می کند، عبارتند از:

- ۱ - بروز پدیده هیچمقدار
- ۲ - بروز عوارض جانبی<sup>۲</sup> در دید تحت عمل بهنگام سازی
- ۳ - نقص قاعده جامعیت و بروز پدیده ناسازگاری داده‌ای<sup>۳</sup> در رابطه (های) مبنا
- ۴ - تاثیر نامطلوب در دیدهای دیگر
- ۵ - غیر اتومیک شدن عمل اتومیک مورد نظر کاربر

دیدها، استقلال داده های منطقی که هم از دلایل و هم از اهداف اساسی تکنولوژی پایگاه داده هاست، بطور کامل تامین نمی شود، و پویایی موجود در تعریف شمای خارجی نیز تضعیف می شود. بعلاوه تنوع دیدهای پذیرایی عملیات بهنگام سازی، از ضوابط مهم در انتخاب RDBMS هاست. هرچه محدودیت‌های یک سیستم در این عملیات کمتر باشد، آن سیستم از این نقطه نظر مطلوب تر است.

### ۳ - تحلیل مشکلات و موارد ناممکن در بهنگام سازی

دیدها را از نظر پذیرش عملیات بهنگام سازی می توان به سه رده اساسی تقسیم کرد:

- ۱ - رده اول: دیدهای بی مشکل
- ۲ - رده دوم: دیدهای قابل بهنگام سازی، اما دارای مشکل
- ۳ - رده سوم: دیدهای غیرقابل بهنگام سازی

#### ۳-۱- دیدهای بی مشکل

واقعیت اینست که علیرغم تلاشهای پژوهشی انجام شده در جهت حل مشکلات موجود در بهنگام سازی دیدها، انواع دیدهای بی مشکل معدود است. حداقل از لحاظ نظری می توان گفت که تنها دیدهایی که در تعریف آنها،  $op$  فقط عملگر تحدید باشد و برخی انواع خاص دیدها که در تعریف آنها  $op$ ، عملگر پیوند، اجتماع، اشتراک یا تفاضل باشد آنهم در شرایط خاص، فاقد دشواری هستند. اگر شرایط خاص و محدود کننده را در نظر بگیریم، حتی دیدی که در اکثر سیستمهای رابطه‌ای کلاسیک ظاهراً قابل بهنگام سازی است و در تعریف آن قواعد محدودکننده وضع شده توسط چمبرلن در سیستم R رعایت شده باشد، نیز در مواردی مشکل دارد. (از نظر چمبرلن دیدی قابل بهنگام سازی است که اولاً روی یک جدول تعریف شده باشد و ثانیاً هر سطر آن متناظر با یک سطر از جدول مبنا و هر ستون آن

می‌گوییم عارضه جانبی در آن دید بروز می‌کند. این مشکل از جمله در دیدهای حاصل پیوند بروز می‌کند که ذیلاً می‌بینیم.

۳-۲-۱- دید حاصل پیوند PK-FK

چنین دیدی ظاهراً نباید مشکل داشته باشد، اما مثال زیر وجود یک مشکل جدی را نشان می‌دهد:

مثال ۱: رابطه های R1 و R2 را در نظر می‌گیریم:

R1 (E#,D#)                      R2(D#,MGR#)

دید V1 را چنین تعریف می‌کنیم:

V1 = JOIN(R1,R2)

توجه داریم که D# در R2 کلید اصلی است پس در R1 کلید خارجی است و پیوند مزبور از نوع PK-FK است. فرض می‌کنیم بسط دورابطه چنین باشد:

r <sub>1</sub>		r <sub>2</sub>	
E#	D#	D#	MGR#
E1	D1	D1	E5
E2	D3	D3	E6
E3	D3	D4	E7
E4	NULL		

با توجه به این بسطها، بسط دید V1 چنین است:

V <sub>1</sub>		
E#	D#	MGR#
E1	D1	E5
E3	D3	E6

حال درخواست می‌کنیم که درج زیر در V1 انجام شود:

INSERT INTO V1

VALUES (E9,D2,E10);

به این معنا که 'کارمند E9 جزء اداره D2 است و مدیر آن اداره E10 است.

آیا این درج باید انجام شود؟ برای پاسخگویی به این سؤال باید ببینیم آیا این درج انجام شدنی است؟ پاسخ مثبت است. زیرا منجر به درج تاپل <E9,D2>

و در نتیجه ایجاد فزون کاری<sup>۱</sup> در سیستم ۶- تغییر ماهیت عمل درخواست شده از سوی کاربر

۷- تعدد تبدیلات بخاط وجود ابهام سمتیک و بروز مشکل تصمیم گیری در سیستم

۳-۲-۱- بروز پدیده هیچمقدار

این پدیده در بهنگام سازی دیدهایی بروز می‌کند که در آنها op، یا عملگر پرتو باشد و یا ترکیبی از عملگرهای پرتو و گزینش و یا عملگری که نتیجه عملکرد آن معادل نتیجه عملکرد این عملگرها باشد و ظاهراً دید قابل بهنگام سازی هم باشد، یعنی دید حافظه کلید اصلی<sup>۲</sup> باشد. در این حالات چون مجموعه عنوان دید زیرمجموعه‌ای از مجموعه عنوان رابطه مبناست، بروز این مشکل در عمل درج اجتناب‌ناپذیر است. به بیان دیگر درج تاپلی در دید (با فرض امکان پذیر بودن عمل) که با توجه به محدوده دید، تاپل کاملی است، منجر به درج تاپلی در رابطه مبنای زیرین می‌شود که ناقص است و در نتیجه پدیده هیچمقدار در این رابطه بروز می‌کند. بروز این پدیده می‌تواند جامعیت پایگاه را نقض کند زیرا هیچمقدار ممکن است برای صفتی درج شود که برای آن قاعده جامعیت عدم پذیرش<sup>۳</sup> هیچمقدار وضع شده باشد. حتی ممکن است قاعده جامعیت ناشی از وابستگی‌های تابعی نیز نقض گردد. به علاوه صرف وجود هیچمقدار در پایگاه مشکلاتی را ایجاد می‌کند (DATE 86, IMIELINSKI 84 و بویژه 95 DATE که در آن دیت نظر می‌دهد که پدیده هیچمقدار، یک انحراف از اصول مدل رابطه‌ای است و مدل را تباه می‌سازد! و اساساً نباید در سیستمهای رابطه‌ای مطرح شود).

۳-۲-۲- بروز عوارض جانبی در دید تحت عمل بهنگام سازی ممکن است انجام عمل بهنگام سازی در یک دید امکان پذیر باشد، اما در پی انجام عمل، تغییراتی در همان دید پدید آید که نامطلوب است. اصطلاحاً

$V2 = JOIN(R3,R4)[A,C]$

این دید حاصل پیوند دو رابطه و سپس پرتو روی صفات خاصه A و C است. بسط رابطه های R3 و R4 را چنین فرض می کنیم:

$r_3$		$r_4$	
1		1	
A	B	B	C
a	b	b	c
$a^2$	b		

با توجه به این بسطها، بسط دید  $V2$  چنین است:

$V_2$	
1	
A	C
a	c
$a^2$	c

\* کاربر این دید درخواست می کند:

DELETE <a,c> FROM V2

اگر این حذف تبدیل شود به

DELETE <b,c> FROM R4

با این حذف، تاپل <a',c> نیز از دید  $V2$  حذف می شود که کاربر نخواسته است.

\* کاربر این دید درخواست می کند

MODIFY <a,c> TO <a,c'>

اگر این تغییر تبدیل شود به:

MODIFY <b,c> IN R4 TO <b,c'>

در اینصورت تاپل <a',c> در دید  $V2$  نیز تغییر می کند که کاربر درخواست نکرده است.

### ۳-۲-۳- نقص قواعد جامعیت

بهنگام سازی دید ممکن است منجر به نقض قواعد جامعیت شود. یک مورد همان بروز پدیده هیچمقدار است برای صفاتی که مجاز به پذیرش آن نیست. مورد دیگر نقض یکتایی مقدار کلید اصلی است که ممکن است در پی انجام عمل درج پیش آید.

مثال ۳: همان دو رابطه R1 و R2 از بند ۳-۲-۲-۱ را

در رابطه R1 و تاپل <D2,E10> در رابطه R2 می شود و تا اینجا مشکلی وجود ندارد. اما پس از درج، کاربر دارای دید  $V1$  انتظار دارد که بسط دیدش چنین باشد:

$V_1$		
2		
E#	D#	MGR#
E1	D1	E5
E3	D3	E6
E9	D2	E10

حال آنکه با پیوند R1 و R2 پس از درج بالا، چنین داریم:

$V_1$		
2		
E#	D#	MGR#
E1	D1	E5
E1	D2	E10
E3	D3	E6
E9	D2	E10

تاپل افزونه (حشو)

تاپل دوم یعنی <E2,D2,E10> اساساً از نظر کاربر دارای دید  $V1$  وجود ندارد. در واقع یک تاپل افزونه یا حشو در پایگاه کاربر پدید آمده است. این عارضه جانبی پدید آمده در  $V1$  نهایتاً سبب خدشه دار شدن صحت و دقت پایگاه تعریف شده برای کاربر دید  $V1$  می شود. اطلاعاتی در پایگاه کاربر وجود دارد که در 'خرد جهان واقع' مورد نظرش وجود ندارد. عارضه جانبی لزوماً همیشه به صورت پدید آمدن تاپل حشو بروز نمی کند. ممکن است حذف یا تغییر تاپلی ناخواسته از دید کاربر باشد. این مشکل از جمله در دید حاصل پیوند و پرتو بروز می کند که در زیر نشان داده می شود.

۳-۲-۲-۳- دید حاصل پیوند و پرتو

مثال ۲: رابطه های زیر را در نظر می گیریم:

$R3(A,B)$        $R4(B,C)$

دید  $V2$  را چنین تعریف می کنیم:

کاربر این دید درخواست می کند:  
MODIFY <a,b,c,> TO <a,b,c'>

برای این تغییر، نیازی به انجام عمل خاص در R3 نیست زیرا تاپل <a,b> در R3 وجود دارد. اما انجام این تغییر منجر می شود به درج تاپل <b,c'> در رابطه R4، و بسط جدید R4 چنین می شود:

$$r_4$$

B	C
b	c
b	c'

و در نتیجه وابستگی تابعی C  $\rightarrow$  B در R4 نقض می شود. جالب اینست که نمی توان بجای درج تاپل <b,c'> در R4، تاپل <a,c> موجود در R4 را به تاپل <b,c'> تغییر داد زیرا با اینکار بسط جدید V3 به گونه ای تغییر می کند که موردنظر کاربر نبوده است:

$$V_3$$

A	B	C
a	b	c'
a'	b	c'

یعنی علاوه بر تغییر تاپل <a',b,c'> به تاپل <a,b,c'> که موردنظر کاربر بوده است، تاپل <a',b,c'> نیز به تاپل <a,b,c'> تغییر می یابد. که اساساً موردنظر کاربر نبوده است (عارضه جانبی در دید تحت عمل). بعلاوه مشکل تغییر ماهیت عمل نیز بروز می کند (ر ک ب ۳-۲-۶)

### ۳-۲-۴- تاثیر در دیدهای دیگر

بهنگام سازی در یک دید می تواند در دید(های) دیگر تاثیر بگذارد زیرا عمل درخواست شده روی دید، نهایتاً باید، در صورت امکان پذیر بودن، در رابطه (های) مبنا انجام شود. این کار روی دیدهای دیگر تعریف شده روی رابطه (های) مبنا تاثیر می گذارد.

مثال ۵: رابطه R5 حاوی اطلاعاتی در مورد کارمندان

درنظر می گیریم. فرض می کنیم که کاربر می خواهد تاپل <E4,D1,E5> را در دید V1 درج کند. انجام این درج می تواند منجر به انجام یک عمل تغییر در R1 و یک درج در R2 شود:

```
UPDATE R1      INSERT
SET D#=D1      INTO R2
WHERE E# = E4;  VALUES <D1,E5>
```

با انجام این درج در R2، یکتایی مقدار D# در این رابطه نقض می شود (از طریق پدید آمدن تاپل تکراری در همین رابطه، زیرا و D# در R2 کلید اصلی است). مشخص است که این درج نباید انجام شود، حال آنکه دید V1 حاصل پیوند PK-FK است.

مورد دیگر نقض قاعده جامعیت ناشی از وابستگی های تابعی بین صفات رابطه است. وابستگی های تابعی از قواعد سمنتیک ناظر بر خرد جهان واقع ناشی می شوند و لذا جزء قواعد جامعیت پایگاه هستند. در اثر بهنگام سازی دید، احتمال نقض شدن چنین قواعدی وجود دارد.

مثال ۴: رابطه های R3 و R4 از مثال ۲ را در نظر می گیریم:

R3(A,B)      R4(B,C)

با بسطهای زیر:

$$r_3$$

A	B
a	b
a'	b

$$r_4$$

B	C
b	c

و فرض می کنیم وابستگی های تابعی زیر برقرار باشند: B  $\rightarrow$  A در R3      C  $\rightarrow$  B در R4.  
کاربری دید V3 را چنین تعریف کرده است:

V3 = JOIN (R3,R4)

این دید هم حاصل پیوند PK-FK است که ظاهراً نباید مشکل داشته باشد.

بسط V3، با توجه به بسطهای R3 و R4 چنین است:

$$V_3$$

A	B	C
a	b	c
a'	b	c

به چندین عمل منطقاً متفاوت و نهایتاً منجر به عملیات غیراتومیک شود، به عبارت دیگر، تناظر یک به یک بین عمل درخواست شده و عملی که باید در پایگاه انجام شود، وجود ندارد. بدیهی است که هرچه تعداد عملی که باید نهایتاً انجام شود، بیشتر باشد، فزون کاری بیشتری در سیستم ایجاد می گردد. لازم به ذکر است که اجتناب از چنین مشکلاتی عملاً امکان ناپذیر است. کاد (CODD 90) پذیرفته است که مثلاً یک عمل واحد درج می تواند به چند عمل اما کماکان درج تبدیل شود و دیت نیز چنین نظری برای درج و حذف، ولی نه برای بهنگام سازی (تغییر) دارد (DATE 90,94)، البته به شرط آنکه، ماهیت عمل عوض نشود (نه برای عمل تغییر، زیرا در پیشنهاد دیت، همه جا عمل تغییر (UPDATE)، به کمک دو عمل حذف و درج باید انجام شود). در قسمت ۳-۲-۲ مثالی از این مورد دیده می شود.

### ۳-۲-۶- تغییر ماهیت عمل

از اصول برنهاده توسط اکثر پژوهشگران در بهنگام سازی دیدها اینست که در تبدیل عمل درخواست شده توسط کاربر به عمل یا عملیاتی در رابطه (های) مبنا، ماهیت عمل نباید تغییر کند، یعنی عمل درج نباید منجر به عمل تغییر شود و یا حذف نباید منتهی به عمل تغییر گردد. در مثال ۳ از بند ۳-۲-۲، عمل درج به عمل بهنگام سازی و در مثال ۵ از همین بند عمل MODIFY به عمل INSERT در رابطه مبنا تبدیل شد. این تغییر ماهیت عمل در بهنگام سازی دیدها نباید پدید آید زیرا با طبیعت عمل درخواست شده توسط کاربر منافات دارد. برای روشن تر شدن مطلب مثال دیگری قید می کنیم:

مثال ۶: رابطه R5 از مثال ۵ را در نظر می گیریم. فرض می کنیم مربی یک تیم ورزشی درخواست کند که کارمند E12 از دید V5 حذف شود. با توجه به سمنتیک رابطه منطقی نیست که این حذف را تبدیل به حذفی در رابطه R5 بکنیم، بلکه باید مقدار صفت TEAM در رابطه R5 را برای این کارمند از "YES" به "NO" تغییر

یک سازمان را در نظر می گیریم:

R5 (E#,ENAME,ELCO,TEAM)

صفات این رابطه عبارتند از: شماره کارمند، نام کارمند، شهر محل خدمت کارمند و کد نشان دهنده اینکه کارمند عضو یک تیم ورزشی هست یا خیر. سازمان در شهرهای c1 و c2 و ... اداره دارد. اعضاء تیمهای ورزشی همه از کارمندان سازمان هستند و یک کارمند فقط عضو یک تیم ورزشی است. فرض می کنیم مدیر امور پرسنلی دیدی چنین داشته باشد:

(علامت \* یعنی تمام صفات رابطه مبنا)

V4 = (R5 Where ELCO = 'c1')[\*]

این دید حاصل عملگر ساده تحدید است که اصولاً باید جزء دیدهای بی مشکل باشد. این کاربر درخواست می کند تا کارمند E10 از دیدش حذف شود (معنایش این نیست که اساساً از سازمان اخراج شود). این حذف منجر به حذف یک تاپل از رابطه مبنای R5 می شود و مشکلی ندارد، اما اگر کارمند E10 عضو یک تیم ورزشی باشد، از آن تیم نیز حذف می شود. مثلاً فرض می کنیم که مربی یک تیم ورزشی در سازمان، دیدی چنین داشته باشد:

V5 = (R5 Where TEAM = 'YES')[\*]

با توجه به حذف E10 از دید V4 و با این فرض که E10 عضو تیم ورزش باشد، حذف تاپلی از دید V4 منجر به حذف تاپلی از دید V5 نیز می شود بی آنکه مربی تیم ورزشی چنین درخواستی کرده باشد. ممکن است تصور شود که می توان E10 را از شهر c1 به شهر دیگری منتقل کرد (بجای حذف آن از رابطه R5، شهرش را تغییر داد) تا دید V5 تاثیر نپذیرد. چنین کاری منطقی نیست، مخصوصاً که کاربر دارای دید V4 حق چنین انتقالی را ندارد.

### ۳-۲-۵- غیراتومیک شدن عمل اتومیک موردنظر کاربر

این مشکل که باعث ایجاد فزون کاری در سیستم می شود هنگامی بروز می کند که عمل منطقاً واحد از دید کاربر (مثلاً یک درج، یک حذف یا یک تغییر) در فرآیند تبدیل به عملیات روی رابطه (های) مبنا، تبدیل



دریافت کمک هزینه را دارد یا خیر و مقدار کمک هزینه، (توجه داریم که این دورابطه را می توانستیم در یک رابطه ادغام کنیم، ولی فرض بر این است که طراح پیاگاه در سطح ادراکی، بخاطر افزایش کارایی، آن رابطه واحد را به این دو رابطه تجزیه کرده است، یعنی با نرمالترسازی رابطه، البته از طریق تجزیه افقی رابطه واحد با استفاده از عملگر تحدید و نه براساس تئوری وابستگی، کاری که دیت به عنوان یکی از راههای نرمالترسازی پیشنهاد کرده است (DATE 90).

یک بسط ممکن از دورابطه چنین است:

r <sub>6</sub>				r <sub>7</sub>			
ST#	NAME	CODE	AID	ST#	NAME	CODE	AID
1	N1	NO	500	3	N3	YES	1000
2	N2	NO	0				

ضمناً می بینیم که در رابطه R6 دانشجوی شماره ۱ شرایط دریافت کمک هزینه را نداشته، اما مبلغی دریافت کرده است. این وضع در عمل می تواند پیش آید و مسئول امر مثلاً بعد از یک یا دو ترم درمی یابد که دانشجویی از یک یا دو ترم پیش دیگر شرایط دریافت کمک هزینه را نداشته است، اما کمک هزینه دریافت کرده است!

کاربر یا کاربرانی دو دید به نامهای V6 و V7 تعریف کرده اند به شرح زیر:

V6 = UNION(R6,R7)      V7=(V6 Where CODE='NO' AND AID>0)[\*]

دید V7 دیدی است تعریف شده روی یک دید دیگر که خود حاصل اجتماع دورابطه است. دید V6 حاوی اطلاعاتی در مورد تمام دانشجویان است. دید V7 مشخص می کند چه دانشجویانی خلافکار بوده اند (کمک هزینه دریافت داشته اند حال آنکه شرایط دریافت را نداشته اند).

با توجه به بسط رابطه های R6 و R7، بسط این

دهیم تا دیگر در دید V5 وجود نداشته باشد. یعنی عمل حذف درخواست شده توسط کاربر تبدیل به عمل تغییر یا جایگزینی در رابطه مبنا می شود. ضمناً توجه داریم که بجای حذف کارمند E12 از دید V5 نمی توان درخواست تغییر یا جایگزینی مقدار صفت خاصه TEAM در خود این دید را کرد یعنی مثلاً نمی توان نوشت:

UPDATE V5

SET TEAM = 'NO'

WHERE E# = 'E12'

زیرا با شرط تعریف V5 تناقض دارد.

### ۳-۲-۷- تعدد تبدیلات بخاطر وجود ابهام سمنتیک

گاه پیش می آید که برای تبدیل عمل درخواست شده توسط کاربر، چندین راه مطرح باشد. در چنین وضعی سیستم باید از 'شناخت سمنتیک' کافی برخوردار باشد تا بتواند عمل درخواست شده را انجام دهد. وجود ابهام سمنتیک در انجام عمل درخواست شده (از جمله ناشی از عدم کفایت قواعد جامعیت کاربری) می تواند تصمیم گیری سیستم را با مشکل مواجه سازد. کلر بر این نظر است که برای حل مشکل ابهام سمنتیک در بهنگام سازی برخی از انواع دیدها، داشتن همه قواعد سمنتیک لازم در زمان تعریف دید کفایت می کند (KELLER 85)، حال آنکه همیشه لزوماً چنین نیست و باید 'منظور' کاربر را در لحظه صدور حکم، دریافت و با توجه به آن عمل کرد. برای درک بهتر این مشکل مثالی قید می کنیم:

مثال ۷: رابطه های زیر را در نظر می گیریم:

R6(ST#,NAME,CODE,AID)      R7(ST#,NAME,CODE,AID)

رابطه R6 حاوی اطلاعاتی است در مورد دانشجویان دوره کارشناسی و R7 حاوی همان اطلاعات است در مورد دانشجویان دوره کارشناسی ارشد، صفات هر دو رابطه عبارتند از: شماره دانشجویی، نام دانشجو، کد نشان دهنده اینکه دانشجو شرایط

دیدها چنین است :

V <sub>6</sub>				V <sub>7</sub>			
1				1			
ST#	NAME	CODE	AID	ST#	NAME	CODE	AID
1	N1	NO	500	1	N1	NO	500
2	N2	NO	0				
3	N3	YES	1000				

عملاً نباید در آنها بهنگام سازی انجام شود). در اینجا باتوجه به ماهیت عملگر OP، بعضی از این دیدها را بررسی می کنیم :

### ۳-۳-۱- عملگر پرتو و دید غیرحافظ کلید

در تعریف دید، OP فقط عملگر پرتو باشد و دید حافظ کلید اصلی نباشد (یعنی کلید اصلی رابطه مبنا در دید وجود نداشته باشد). این دید در واقع به گونه ایست که حتی تمام شرایط برنهاده توسط چمبرلن را هم حائز نیست. به دلیل عدم وجود کلید اصلی در دید، امکان نگاشت عمل درخواست شده در دید، در رابطه ای که دید روی آن تعریف شده است، وجود ندارد.

حال فرض می کنیم کاربری بخواهد عمل زیر را انجام دهد :

DELETE <1, N1, 0, 500> FROM V7

برای تبدیل این عمل به عملی در رابطه (های) مبنا، چند امکان وجود دارد :

الف : REPLACE < 1, N1, NO, 500 > WITH < 1, N1, YES, 500 > IN R6

ب : REPLACE < 1, N1, NO, 500 > WITH < 1, N1, NO, 0 > IN R6

پ : DELETE < 1, N1, NO, 500 > FROM R6

۳-۳-۲- ترکیب عملگرهای پرتو و تحدید و دید غیرحافظ کلید در تعریف دید، OP ترکیبی از عملگرهای پرتو و تحدید (گزینش) باشد و دید حافظ کلید اصلی نباشد، چنین دیدی حالت خاصی از دید مورد ۳-۳-۱ است و همان وضع را دارد.

البته چون دید V7 روی دید دیگر یعنی V6 تعریف شده است، برای تبدیل عمل درخواست شده، ابتدا باید این عمل تبدیل به عملی روی V6 شود و سپس به یکی از احکام بالا تبدیل گردد، آنهم به شرطی که سیستم بداند در کدامیک از دو رابطه R6 یا R7 باید این عمل انجام شود (V6 حاصل اجتماع این دو رابطه است). در این وضع، سیستم برای تصمیم گیری نیاز به شناخت سمنتیک بیشتری در زمان اجرا دارد. مثلاً اگر یک قاعده سمنتیک چنین باشد کاربر اداره وام دانشجویی فقط می تواند پایگاه خود را بهنگام درآورد و حق حذف ندارد، با این قاعده امکان سوم منتفی می شود. تعدد تبدیلات سبب می شود تا سیستم با مشکل تصمیم گیری مواجه گردد.

### ۳-۳-۳- تابع جمعی

در تعریف دید، OP تابع جمعی باشد مثل SUM, MAX, MIN و AVG و ... که معمولاً ماهیت آماری دارند، در واقع اگر در تعریف دید از تابع آماری استفاده شود مثلاً  $V_i = \text{MAX}(A_j) A_j \in A$  که در آن A مجموعه صفات رابطه R<sub>i</sub> است، معنایش اینست که صفت یا صفاتی از دید، مجازی هستند (در نمایش جدولی، دید حاوی ستون یا ستونهایی است که متناظر در رابطه مبنا ندارند). انجام عمل بهنگام سازی روی صفت مجازی ناممکن است، به این دلیل بدیهی است که امکان نگاشت عمل موردنظر، در رابطه مبنا وجود ندارد، بعلاوه عمل بهنگام سازی روی صفت مجازی، اساساً فاقد سمنتیک است. مثلاً تغییر ماکزیمم، می نیمم یا مجموع مقادیر یک ستون از دید نمی تواند معنا داشته باشد.

### ۳-۳-۴- عملگر پیوند در حالت NK-NK

### ۳-۳- دیدهای غیرقابل بهنگام سازی

بسیاری از دیدها ذاتاً غیرقابل بهنگام سازی هستند (صرفنظر از بعضی دیدهای دارای مشکل که

اجتماع چند رابطه ، مشکل تشدید می شود، از نظر کاد، عمل حذف ، به شرط اینکه معلوم شود تاپل حذف شدنی متعلق به کدام رابطه میناست ، می تواند در آن رابطه انجام شود اما عمل درج فقط در شرایط خاصی امکان پذیر است (90 CODD) ، پس در این حالت نیز گاه عمل بهنگام سازی در دید ناممکن است و گاه با دشواری مواجه است مخصوصاً اگر عملگر اجتماع با عملگر دیگری مثل پرتو یا تحدید ترکیب شود(ازجمله به ۲-۲-۷ رجوع شود).

### ۳-۳-۶- عملگر اشتراک یا تفاضل

در تعریف دید، OP اشتراک یا تفاضل دو یا بیش از دو رابطه باشد، در این حالت نیز با توجه به اینکه راه حل مشابه راه حل اجتماع رابطه ها، ارائه شده است ، مشکل مطرح شده در ۲-۳-۵ در این حالت نیز وجود دارد.

### ۳-۳-۷- عملگر ضرب $V = MULTIPLY(R_i, R_j)$

عمل ضرب دو رابطه (یا بیش از دو رابطه) را می توان حالتی از عمل پیوند دانست که در آن دو رابطه عملوند، صفت مشترک ندارند و رابطه حاصله ، حاصل ضرب کارتیزین دو رابطه  $R_i$  و  $R_j$  است . در این حالت نیز انجام عمل بهنگام سازی دید، به سبب بروز مشکلات اساسی ، عملاً ناممکن می شود. مثال ۸: رابطه های  $R_8$  و  $R_9$  با بسطهای زیر را در نظر می گیریم :

$R_8(A,B,C)$			$R_9(D,E)$	
$r_8$			$r_9$	
A	B	C	D	E
a1	b1	c1	d1	e1
a1	b2	c2	d2	e2
a2	b2	c2		

دید  $V_8$  را چنین تعریف کنیم :

$$V_8 = MULTIPLY (R_8, R_9)$$

در تعریف دید ، OP عملگر پیوند دویا بیش از دو رابطه باشد در حالتی که پیوند PK-PK یا PK-FK نباشد (دیدیم که در حالت پیوند PK-PK یا PK-FK نیز مشکلاتی وجود دارد، ازجمله به بحث ذیل ۲-۲-۲ رجوع شود ) یعنی در حالات زیر عمل بهنگام سازی ناممکن است :

$$NPK - NPK , FK - FK , NFK - NFK$$

ضمناً اگر OP ترکیبی از پیوند دو یا بیش از دو رابطه و سپس پرتو رابطه حاصل از پیوند باشد، حتی در حالت پیوند PK-PK یا PK-FK حداقل مشکلی که پیش می آید اینست که در رابطه (های) زیرین پدیده هیچمقدار بروز می کند . در حالتی که پیوند PK-PK یا PK-FK نباشد و عملگر پرتو هم اعمال شود، باز هم عمل بهنگام سازی دید ناممکن می شود.

### ۳-۳-۵- عملگر اجتماع

در تعریف دید، OP عملگر اجتماع دو یا بیش از دو رابطه باشد. مشکل بهنگام سازی چنین دیدی کاملاً حل نشده است . دیت در یکی از آخرین کارهایش (DATE 94) راه حلی پیشنهاد کرده است ، مبنی براینکه سیستم با داشتن قواعد جامعیت (محدودیتهای) ناظر به رابطه های مبنا، تقریباً می تواند تشخیص دهد که عمل درخواست شده در کدام رابطه باید انجام شود. این قواعد جامعیت به صورت یک (یاچند) محمول (مسند) موسوم به محمول رابطه بیان می شوند(از نظر دیت ، برای هر رابطه می توان یک یا چند محمول تعریف کرد (DATE 95) ). اما این راه حل در شرایطی که امکان چنین تشخیصی وجود نداشته باشد، خواه به دلیل یکسان بودن مجموعه قواعد جامعیت ناظر به دو رابطه که (محتمل است ) و خواه به دلیل دشواری در بیان بعضی از قواعد جامعیت به سیستم به کمک محمول (ها) و خواه به دلیل اینکه دو رابطه عملوند در عمل اجتماع خود زیرمجموعه هایی از یک رابطه باشند و با یکدیگر اشتراک تاپل داشته باشند، قابل اعمال نیست . روشن است که در حالت

بسط  $V_8$  چنین است :

$V_8$				
1				
A	B	C	D	E
a1	b1	c1	d1	e1
a1	b1	c1	d2	e2
a1	b2	c2	d1	e1
a2	b2	c2	d2	e2
a2	b2	c2	d1	e1
a2	b2	c2	d2	e2

می دانیم که در حالت کلی، کلید اصلی این رابطه، اجتماع مجموعه عنوان دو رابطه است. حال عملیات زیر را در  $V_8$  انجام می دهیم :

DELETE FROM  $V_8$  TUPLE: <a1,b1,c1,d2,e2>

با این حذف،  $R_8$  از <a1,b1,c1> و  $R_9$  از <d2,e2> حذف می شود و در نتیجه چهار تاپل از دید  $V_8$  خارج می شوند که منظور کاربر این نبوده است. در بهنگام سازی نیز مشکل مشابهی بروز می کند.

INSERT INTO  $V_8$

در درج :

TUPLE : <a3,b2,c1,d3,e4>

این درج منجر به درج تاپل <a3,b3,c1> در  $R_8$  و تاپل <d3,e4> در  $R_9$  می شود. در نتیجه در دید  $V_8$  تاپلهای دیگری هم پدید می آیند که اساساً موردنظر کاربر نبوده اند.

نتیجه اینکه در این حالت نیز عمل بهنگام سازی دید ناممکن می شود.

۳-۳-۸- عملگر تقسیم ( $R_i, R_j$ )  $V = \text{DIVIDE}$

در تعریف دید،  $OP$  عملگر تقسیم باشد، دید حاصل تقسیم دو رابطه اساساً رفتار مشخصی ندارد. انجام عمل بهنگام سازی در چنین دیدی در حالت کلی ناممکن است. چنین دیدی ماهیتاً مثل دیدی است که  $OP$  در تعریف آن ترکیبی از عملگرهای پرتو و

گزینش باشد که لزوماً حافظ کلید نیست. حذف یک تاپل از این دید می تواند به حذف تعدادی تاپل از رابطه مقسوم منجر شود و نیز بهنگام سازی (تغییر یک تاپل) همین مشکل را دارد. عمل درج اساساً ناممکن است زیرا با عملکرد تقسیم منافات دارد، ضمن اینکه پدیده هیچمقدار هم بروز می کند. بنابراین می توان گفت که در حالت عملگر تقسیم نیز بهنگام سازی ناممکن است.

مثال ۹: فرض می کنیم که  $R_{10}$ ، رابطه مقسوم و  $R_{11}$  رابطه مقسوم علیه باشد:

$R_{10}(X,Y,Z)$        $R_{11}(Z)$

صفحات  $X$ ،  $Y$  و  $Z$  می توانند ساده یا مرکب باشند. بسط این دو رابطه را در یک لحظه خاص چنین برمی نهم:

$r_{10}$			$r_{11}$
X	Y	Z	Z
x1	y1	z1	z1
x1	y1	z2	z2
x1	y1	z3	z3
x1	y1	z4	
x2	y1	z1	
x2	y1	z2	
x3	y1	z3	
x3	y2	z1	
x1	y2	z2	

و داشته باشیم:  $V_9 = \text{DIVIDE}(R_{10}, R_{11})$   
بسط این دید چنین است:

$V_9$	
X	Y
x1	y1
x2	y1

حال اگر عمل زیر را بخواهیم در این دید انجام دهیم:

DELETE

FROM  $V_9$

WHERE  $x = x1$

این حذف تبدیل می شود به حذف زیر در رابطه  $R_{10}$ :

دیدها، نقطه ضعف مهمی در پایگاههای رابطه ای است و سبب می شود تا مزایایی مفهوم دید به تمامی حاصل نشود.

### ۵ - خلاصه

دید مفهومی همگانی در مدل رابطه ای و در معماری پایگاه داده های رابطه ای است و مزایای قابل توجهی دارد که اهم آن عبارتند از تامین استقلال داده ای، اشتراکی شدن داده ها، تامین محیط انتزاعی و تامین نوعی مکانیسم اتوماتیک ایمنی پایگاه. کاربران باید بتوانند از محدوده دید خود عملیات بازایی و ذخیره سازی را انجام دهند. عملیات بازایی از دید اقلأ از نظر تئوریک مشکلی ندارد، اما همه انواع دیدها عملیات ذخیره سازی را نمی پذیرند. در تعریف دید معمولاً از عبارات منطقی یا محمولها استفاده می شود و هر دید یک مکانیسم اشتقاق مشخص دارد. در تعریف دید از عملگرهای مختلفی از جبر رابطه ای یا محاسبات رابطه ای می توان استفاده کرد و با پویایی بسیار بالایی، انواع دیدها را می توان تعریف کرد. اما از انواع مختلف دیدها، تنها معدودی در عملیات بهنگام سازی بی مشکل هستند. برخی اساساً غیرقابل بهنگام سازی بوده و در بسیاری از آنها مشکلاتی وجود دارد از جمله بروز پدیده هیچمقدار، بروز عوارض جانبی در دید، نقض قواعد جامعیت، بروز عوارض جانبی در دیدهای دیگر، غیراتومیک شدن عمل موردنظر کاربر، تغییر ماهیت عمل و تعدد تبدیلات به سبب وجود ابهام سمنتیک، به دلیل وجود این مشکلات در بسیاری از انواع دیدها، امکان تبدیل عمل درخواست شده در رابطه (های) مبنا یا وجود ندارد و یا منطقی نیست. بسیاری از مشکلات موجود هنوز ناگشوده مانده اند.

DELETE

FROM R10

WHERE  $x = x1$

یعنی پنج تاپل از رابطه R10 حذف می شوند (و نه یک تاپل مشخص) و از جمله تاپل  $\langle x1, y2, z3 \rangle$  که از آن، قسمت  $\langle x1, y2 \rangle$  اصلاً در رابطه خارج قسمت یعنی دید V9 نیست. همین مشکل در بهنگام سازی (تغییر) نیز وجود دارد. واضح است که درج مثلاً تاپل  $\langle x2, y3 \rangle$  در دید V9 بی معنا و در نتیجه ناممکن است، زیرا آنچه که در V9 وجود دارد باید حاصل تقسیم دو رابطه R10 و R11 باشد.

### ۳-۳-۹- سورها

چنانچه در تعریف دید از سورهای وجودی یا همگانی استفاده شود، از آنجا که جبر رابطه ای و محاسبات رابطه ای معادلند، لذا می توان معادل چنین دیدی را با استفاده از عملگرهای جبر رابطه ای تولید کرد و سپس آنرا مورد تحلیل قرار داد تا امکان پذیر بودن عمل بهنگام سازی یا ناممکن بودن آن و مشکلات موجود در این کار، مشخص گردد.

### ۴ - نتیجه

به سبب اهمیت مفهوم دید در مدل رابطه ای و مزایایی آن، امکان پذیر ساختن عملیات بهنگام سازی در دیدها یک الزام است. به رغم تلاشهای پژوهشگران در سالهای اخیر، هنوز انواع زیادی از دیدها یا غیرقابل بهنگام سازی هستند و یا مشکلات قابل توجهی دارند. این مشکلات بسته به نوع مکانیسم اشتقاق دید (عملگرهای) بکار رفته در تعریف دید و نیز مقدار و ماهیت ابهامات سمنتیک، متفاوت اند. در مواردی حتی دیدهای بسیار ساده هم در بهنگام سازی مشکل دارند. وجود مشکلات در بهنگام سازی

## مراجع

- 1 - Bancilhon, F. , and Spyrtos, N.(1981). "Update semantics of relational views." *ACM transactions on database systems*, 6(4), Dec.
- 2 - Codd, E. F. (1990). *The relational model for database management*, Version 2, Addison-Wesley.
- 3 - Cosmadakis, S. S. (1984). "Update of relational views." *Journal of The ACM*, 31(4), Oct.
- 4 - Date, C. J.(1983). *An introduction to database systems*, Vol. I, 3rd ed., Addison-Wesley.
- 5 - Date, C. J.(1990). *An introduction to database systems*, Vol. I, 5th ed., Addison-Wesley.
- 6 - Date, C. J.(1995). *An introduction to database systems*, Vol. I, 6th ed., Addison-Wesley.
- 7 - Date, C. J., and Mcgoveran, D.(1994). "Updating union, intersection, and difference views." *Database programming & design*, 7(6), Jun.
- 8 - Date, C. J.(1982). *Relational database : selected writings*, Addison-Wesley.
- 9 - Dayal, U.(1982). "On the correct transaction of update operations on relational views." *ACM transactions on database systems*, 8(3), Sep.
- 10 - Delobel, C. et al.(1995). *Database : from relational to object oriented systems*, London , THOMSON Publishing.
- 11 - Imielinski, T ., and Lipski, W.(1984). "Incomplete information in relational database." *Journal of ACM*, 31(4), Oct.
- 12 - Keller, A.(1985). "Algorithm for translating view updates to database updates for views involving selections, projections and joins." *Proceedings of the 4th ACM symposium on principles of database systems*. ACM, New York, 154-168.
- 13 - Keller, A.(1987). "Comment on bancilhon and syrtos update semantics and relational views." *ACM transactions on database systems*, 12(3), Sep.
- 14 - Korth, H., and Silberschatz, A.(1990). *Data base system concepts*, 2nd Ed., McGraw-Hill.
- 15 - Langerak, Rom.(1990). "View update in relational databases with an independent schema." *ACM transactions on database systems*, 15(4), Mar.
- 16 - Ulliman, J. D.(1988). *Database and knowledge-base systems*, Vol. 1, Computer Service Press.