

عملیات ذخیره سازی در دیدها

View Updating

ارائه شده به:

دکتر روحانی رانکوهی

توسط:

آبتین رسولیان

پایگاه داده

ترم دوم سال تحصیلی 84-85

دانشکده کامپیوتر

دانشگاه صنعتی شریف

پیشگفتار

به نظر می آید مفوله نخیره سازی در دیده‌ها مانند دیگر مباحث پایگاه داده‌ها چنان پیچیده باشد که نگارش یک مطلب جامع درباره آن بسیار دشوار باشد. من بر حسب جستجو‌هایی که انجام دادم به این نتیجه رسیدم که پیگیری یک راه حل ارائه شده و منطقی در راستای این مسئله بسیار جذاب‌تر و مفیدتر خواهد بود گرچه قضاوت درباره عملکرد اینجانب بر عهده استاد ارجمند می باشد. راه حلی که در زیر آورده شده با استفاده از جداول شرطی به حل مسئله می پردازد. اینجانب تمام تلاش خود را کردم که حق مطلب را بیان کنم اگرچه به دلیل کمبود وقت به نگارش بسیاری از مطالب مهم نپرداختم.

فهرست

2	پیشگفتار
4	چکیده
4	مقدمه
4	روش VIEW-DEPENDENCY
5	یک مثال
5	مقدمات
5	جدول شرطی
6	بعضی اعمال روی جدول شرطی
6	ارزشگذاری جدول شرطی
7	چند تعریف
8	تعریف ترجمه درست
9	CSP
9	الگوریتم حل مسئله
10	مراجع

چکیده

در این تحقیق در ابتدا مواردی از انواع روشهای ذخیره سازی در دیدها بیان می شود همچنین رویکرد اصلی مورد توجه بیشتر قرار می گیرد. در ابتدا جدول شرطی تعریف می شوند و سپس با استفاده از اعمالی که روی این جدولها می توان انجام داد درخواست ذخیره سازی به یک درخواست در سطح داده های مبنا ترجمه می شود.

مقدمه

مسئله ذخیره سازی در دیدها به ترتیب اثر دادن به درخواست غیر مبهم ذخیره سازی برای تغییر داده های یک دید که روی چندین جدول مبنا تعریف شده ، اطلاق می شود. یک درخواست ذخیره سازی ممکن است به یک تغییر یکتا در داده های مبنا منجر نشود.

در برخورد با یک درخواست ذخیره سازی چند رویکرد را می توان دنبال کرد. می توان فرض کرد که دید یک موجود مجرد است و تغییرت روی آن مفهومی ندارد. رویکرد دیگر این است که دید را به صورت فیزیکی ذخیره کنیم¹. آنگاه تغییر در داده های دید بدون ایجاد تغییرات در داده های مبنا ممکن خواهد بود. اگرچه با تغییرات در داده های مبنا دید نیز باید تغییر کند. اما استراتژی محافظه کارانه ای وجود دارد که ترجمه را در حالت های خاص ممکن می داند:

- اگر دید روی یک جدول تعریف شده باشد و کلید جدول در دید باشد ، ذخیره سازی ممکن خواهد بود.
 - اگر دید روی چندین جدول و با عملیات join درست شده باشد ، ذخیره سازی ممکن نخواهد بود.
 - اگر دید با استفاده از توابع جمعی تعریف شده باشد، ذخیره سازی ممکن نیست.
- رویکرد دیگر ترجمه درخواست ذخیره سازی به یک عمل ذخیره سازی در جدولهای مبنا است. چنین ترجمه ای در صورتی که دید از مجموعه پیچیده ای از جبر رابطه ای روی جدولهای مبنا تشکیل شده باشد ، بسیار دشوار خواهد بود. در ضمن ممکن است چنین تغییری ممکن نباشد و یا روی دیدهای دیگر side effect داشته باشد. DBMS ها پشتیبانی چندانی از این روش نمی کنند.

روش view-dependency

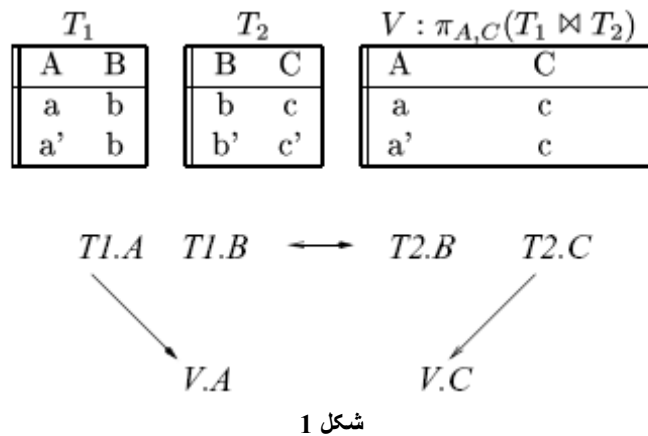
یکی از روشهای ترجمه درخواست ذخیره سازی به سطوح پایین تر روش view-dependency است. در این روش ابتدا گراف وابستگیها کشیده می شود. یک رابطه مبنا برای یک دید ، clean source است اگر برای هر صفت از دید ، صفتی در رابطه مبنا باشد که [در گراف وابستگیها] به آن صفت مسیر داشته باشد. اگر چنین باشد درخواست ذخیره سازی در دید به ذخیره سازی در آن رابطه مبنا منجر می شود.

از مضرات این روش کامل نبودن آن و همچنین محدودیت آن به ترجمه یک درخواست به درخواست یکسان در سطوح پایین تر است (به طور مثال delete به delete ترجمه می شود).

¹ View Materialization

یک مثال

همان طور که در شکل 1 می بینید دید V روی دو جدول T_1, T_2 تعریف شده است. اگر درخواست حذف تاپل $\langle a, c \rangle$ از دید V را داشته باشیم، آنگاه طبق روش view-dependency نمی توان حذفی داشت. چون اگر $\langle a, b \rangle$ از جدول T_1 و $\langle b, c \rangle$ از جدول T_2 برای این تاپل source محسوب شوند؛ نمی توان هیچکدام از این تاپل ها را clean-source به حساب آورد. چون در گراف شکل 1 به طور مشخص از $T1.B$ به $V.A$ و از $T2.B$ به $V.C$ مسیری وجود ندارد. اگرچه این روش در مورد حذف $\langle b, c \rangle$ راهنمایی درستی ارائه کرده، اما حذف $\langle a, b \rangle$ از T_1 بدون هیچ مشکلی امکان پذیر بوده.



در بخشهای بعدی روشی را که مبتنی بر استفاده از جدول شرطی است را بازگو می کنیم :

مقدمات

جدول شرطی¹

جدول شرطی یک سه تایی است به صورت $T_C = (T, \Phi_T, \phi)$ که T یک رابطه است. Φ_T یک شرط کلی برای همه جدول است و عنوان آن شرط عمومی² است. ϕ که شرط محلی³ نامیده می شود نیز برای هر تاپل تعریف می شود، به این صورت که $\phi(t)$ شرطی است که روی تاپل t از رابطه T تعریف می شود. شرط محلی زیر یک ستون با عنوان con معرفی می شود. ممکن است متغیرهایی که در شرط محلی ظاهر می شوند در قسمتهای دیگر جدول حضور نداشته باشند. شکل 2 یک نمونه از این جدول را نشان می دهد.

¹ Conditional Table

² Global Condition

³ Local Condition

$$u \neq v$$

A	C	con
t	c	$x = u$
s	c	$y = u$
t	c'	$x = v$
s	c'	$y = v$

شکل 2

بعضی اعمال روی جدول شرطی

Projection: پرتو روی $T_C = (T, \Phi_T, \phi)$ به صورت $\pi_Y(T_C) = (T, \Phi_T, \phi')$ نمایش داده می شود که ϕ' یک تابع روی تاپلهای $\pi_Y(T)$ است که برای هر تاپل t که شرط $t = t'[Y]$ را داشته باشد، داریم: $\phi'(t) = \phi(t)$.

Selection: اگر X مجموعه صفات T باشد داریم $\sigma_E(T) = (\sigma_E(T), \Phi_T, \phi')$. برای هر تاپل جواب داریم $\phi'(t) = \phi(t) \wedge E(t)$ که $E(t)$ نتیجه جابجایی $t(A)$ برای A های در E است ($A \in X$). نتیجه اعمال دیگر از قبیل اجتماع و پیوند را در اینجا نیاورده ایم.

ارزشگذاری جدول شرطی

برای مجموعه متناهی V از متغیرها، ارزیابی v یک تناظر است از متغیرهای V به دامنه متغیرها. ارزیابی v ناکامل است اگر تناظر ناکامل باشد و یا برای بعضی از متغیرها داشته باشیم $x \in v, v(x) = \eta$ که به معنای وجود نداشتن¹ است.

ارزشگذاری شرط C با توجه به ارزیابی v به صورت $V^v(C)$ نمایش داده می شود که سه مقدار $T(true)$, $F(false)$, $U(undefined)$ می پذیرد و به صورت زیر محاسبه می شود:

1. برای جزء $x=y$

اگر $v(x) = \eta$ یا $v(y) = \eta$ آنگاه $V^v(x=y) = U$

اگر $v(x) \neq \eta$ و $v(y) \neq \eta$ آنگاه $V^v(x=y) = T$

در غیر این صورت $V^v(x=y) = F$

2. برای عبارات بولی می توان به صورت زیر عمل کرد:

$$V^v(\neg p) = \neg V^v(p)$$

$$V^v(p \wedge q) = V^v(p) \wedge V^v(q)$$

$$V^v(p \vee q) = V^v(p) \vee V^v(q)$$

جدولهای ارزشگذاری در شکل 3 آمده است.

¹ Not Existing

p	$\neg p$	\wedge	T	F	U	\vee	T	F	U
T	F	T	T	F	U	T	T	T	T
F	T	F	F	F	F	F	T	F	U
U	U	U	U	F	U	U	T	U	U

شکل 3

ارزشگذاری جدول T_C با توجه به تناظر v به صورت $v(T_C)$ نمایش داده می شود و به صورت زیر تعریف می شود :

$$v(T_C) = \{v(t) | t \in T, V^v(\phi(t)) = T, v(x) = \eta : x \in t\} \text{ اگر } V^v(\Phi_T) = T$$

در غیر این صورت : $v(T_C) = \{\}$

بنابراین با احراز شرایط شرط عمومی : $v(T_C)$ تمام $v(t)$ هایی است که v شرایط شرط عمومی تاپل مربوطه را تامین کند. $v(T_C)$ حالت¹ T_C است با توجه به v .

مجموعه تمام حالت‌های T_C به صورت زیر تعریف می شود:

$$rep(T_C) = \{v(T_C) | \Phi_T(v) = T\}$$

اگر T_C یک جدول شرطی و r_1, r_2 دو حالت برای T_C باشند به صورتی که $r_1 = v_2(T_C)$ و $r_2 = v_2(T_C)$ ، می گوییم r_2 توسط r_1 قابل دستیابی است اگر عملیات U متشکل از سه عملیات ذخیره سازی² وجود داشته باشد که طبق آن داشته باشیم : $r_2 = U(r_1)$.

چند تعریف

تعریف. اگر I یک پایگاه داده و I_C مجموعه ای از جدولها باشد که داشته باشیم $I = v(I_C)$ برای یک v ، متغیرهای موجود در I_C برای دید q که روی I تعریف شده ، کافی هستند اگر برای هر مقدار a از صفتی که در مجموعه صفات گسترده³ دید قرار دارد ، متغیر x موجود باشد که

$$v(x) = a$$

مجموعه صفات گسترده یک دید از اجتماع دو مجموعه به دست می آید :

1. مجموعه صفات متمایز⁴ که صفاتی هستند که در بخش select از تعریف دید آمده اند.

2. مجموعه صفات exposed که در شروط where از تعریف دید آمده اند.

¹ state

² IDM : Insert, Delete, Modification

³ Extended Attribute Set

⁴ Distinguished Attribute

عملیات ذخیره سازی در یک رابطه مبنا ، بر یک دید موثر است اگر عملیات موجب تغییرات در مقدار بعضی صفات موجود در مجموعه صفات گسترده دید شود. برای دریافت این تغییرات کافی است متغیرهایی را برای معرفی مقادیر جایگزین کنیم. این متغیرها برای دید باید "کافی" باشند.

تعریف. اگر I یک پایگاه داده و I_C مجموعه ای از جدول باشد که داشته باشیم $I = v(I_C)$ و متغیرهای I_C برای دید q کافی باشند ، اگر برای هر صفت x که مقدار a از صفت A را توصیف می کند داشته باشیم : $D(x) = Dom(A)$ آنگاه I_C برای I ، M-version است با توجه به دید q . اگر $D(x) = Dom(A) \cup \{\eta\}$ آنگاه I_C برای I ؛ DM-version است با توجه به دید q .

DM-version برای یک پایگاه داده می تواند تاثیر حذف در پایگاه I را بر دید q نشان دهد. عمل درج در یک رابطه مبنا از یک دید بر دید موثر است اگر عمل درج تاپلهایی را در رابطه مبنا درج کند که حداقل یک صفت از رابطه در مجموعه صفات گسترده دید باشد. بنابراین برای دریافت نتیجه درج کافی است یک تاپل آزاد را اضافه کنیم.

تعریف. اگر I یک پایگاه داده باشد و دید q روی آن تعریف شده باشد، ما تاپل آزاد را به هر رابطه در I با صفات جزو صفات گسترده دید اضافه می کنیم. هر تاپل آزاد به صورت زیر ساخته می شود:

برای هر صفت A که جزو مجموعه صفات گسترده دید است متغیر x را (که $D(x) = Dom(A) \cup \{\eta\}$) قرار می دهیم و برای بقیه $null$ می گذاریم.

مجموعه جدول جواب I-version پایگاه I نامیده می شود.

تعریف. اگر I_C با توجه به q برای I ، M-version باشد و به آن تاپل آزاد اضافه کنیم مجموعه جدول جواب IM-version است برای پایگاه I .

تعریف ترجمه درست

اگر I یک پایگاه داده باشد و q یک دید باشد که با جبر رابطه ای روی I تعریف شده باشد ، حالت q روی I به صورت $q(I)$ نمایش داده می شود. عملیات ذخیره سازی μ روی q به صورت $\mu(q(I))$ تعریف می شود. مسئله اصلی این است که چگونه μ را به یک عملیات IDM روی I ترجمه کنیم. یک ترجمه درست (U) روی پایگاه داده I باید شرایط زیر را داشته باشد:

1. سازگاری را باید حفظ کند. (منظور قواعد جامعیتی و محدودیتها است)

2. U نباید side effect داشته باشد یعنی باید داشته باشیم $q(U(I)) = \mu(q(I))$

مسئله side effect زمانی مطرح است که دیدها از عمل join برای ساخته شدن ، بهره برده باشند.

CSP

برای فرموله کردن مسئله ذخیره سازی از یک برآورد محدودیت استفاده می کنیم. یک مسئله برآورد محدودیت¹ CSP یک سه تایی به صورت (Γ, D, C) است که Γ مجموعه ای است از متغیرها و D تابعی است که هر متغیر x در Γ را به دامنه آن می برد. C مجموعه ای از محدودیتها است روی متغیرهای Γ . اگر همه دامنه ها متناهی باشند آنگاه CSP را متناهی می نامیم. یک "راه حل" برای یک CSP، یک تابع مناسب D است به صورتی که شرایط C تامین شود. یک CSP قابل حل است اگر یک راه حل برای آن وجود داشته باشد. مقدار دهی کامل است اگر به تمام متغیرهای CSP مقدار داده شود. برای حل CSP لازم نیست مقدار دهی کامل باشد. یک CSP ممکن است چندین راه حل داشته باشد.

الگوریتم حل مسئله

1. جدولی را که فضای جستجوی تمام ترجمه های ممکن را با توجه به درخواست ذخیره سازی ایجاد می کنیم.

اگر درخواست ، حذف بود DM-version

اگر درخواست ، درج بود و قرار بر ترجمه به بهنگام سازی بود I-version

اگر درخواست ، درج بود و قرار بر ترجمه به درج بود M-version

اگر درخواست ، بهنگام سازی بود IM-version را تولید کن

اکنون I_C را داریم. برای وابستگیهای تابعی باید شروط لازم را روی I_C قرار دهیم. v یک ارزشگذاری است برای I_C باشد که داریم $v(I_C) = I$. ما چندتایی (I, q, μ, I_C) را مشخصه ذخیره سازی دید² می نامیم.

2. دید را روی I_C محاسبه می کنیم و جدول T_C را به دست می آوریم. داریم $T_C = q(I)$. با توجه به خصوصیات جدول شرطی داریم $q(I) = q(v(I_C)) = q(v(q(I_C))) = v(T_C)$. که به این معنی است که حالت $q(I_C)$ با توجه به ارزشگذاری v همان حالت $q(I)$ است. اگر تاثیر ذخیره سازی روی دید را V' بنامیم آنگاه $V' = \mu(v(T_C)) = v'(T_C)$ برای یک v' .

3. CSP را با توجه به V' بساز. بنابراین برای هر ارزشگذاری v از T_C داریم: $V' = v(T_C)$ اگر و فقط اگر v یک "راه حل" برای CSP ساخته شده باشد.

4. CSP را حل کن.

5. ترجمه لازم را با توجه به راه حل CSP انجام بده. به این معنی که برای هر v' که جواب است داریم $V' = v'(T_C)$. بنابراین ترجمه یک IDM است به نام U که طبق آن $v'(I_C)$ از $v(I_C)$ قابل دستیابی است.

¹ Constraint Satisfaction Problem

² View Update Specification

[1] Shu,Hua , Formulating view update Translation as constrain satisfaction,
Hannover University.

[2] Elmasri, Ramez, Fundamental of Database System, US, 2000.