

پایگاه داده پیشرفته

محمد داورپناه جزئی

ترم دوم ۹۴-۹۳

گروه مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی فولاد

فصل دوم: یکپارچگی (Integrity), ۲-۱ مقدمه

- یکپارچگی از دید تئوری: شامل مفاهیمی چون

دقت یا accuracy درستی یا correctness بامعنی بودن یا validity
سازگاری یا consistency (یکسانی فیلدهای معادل در جداول مختلف,
شماره درس, معدل رشته و دانشگاه)

- یکپارچگی از دید عملی: گارانتی کردن DB در مقابل update های غلط توسط کاربر و یا برنامه نویس, شکست سیستم, فرابکارهای عمدی (بیشتر مربوط به ایمنی)

- یکپارچگی در سیستمهای چند کاربره: عدم دفالت مستقیم یا غیرمستقیم کاربران در کار یکدیگر, درست کار کردن تراکنشهای تک ولی غلط کار کردن همزمان

- تاکید بیشتر روی جنبه های پژوهشی تا نحوه کار سیستمهای موجود

بالا بودن حجم کار در جنبه های تئوریک, عدم پیاده سازی کامل در محیطهای DBMS

- مثالها: از پایگاه داده تولید کننده و محصولات S&P DB, نمونه ش ۲-۱ ص ۱۷

فصل دوم: یکپارچگی، ۲-۲ قوانین یکپارچگی

- وظایف زیر سیستم یکپارچگی
 - دنبال کردن تراکنشها و یافتن موارد نقض یکپارچگی
 - انجام عملیات تعریف شده در صورت یافتن موارد نقض
 - ارائه گزارش فضا در موارد نقض
 - تصحیح اشتباه در صورت امکان
 - لزوم تعریف قوانین یکپارچگی برای انجام موارد بالا: نمونه شکل ۲-۲ ص ۱۳۷
 - سافتار قوانین یکپارچگی
 - < عبارت else > < عبارت شرط > < عبارت after >
 - یا <violation response> <constraint> <trigger condition>
- اگرچه violation مجاز نیست، برای احتیاط حالت اگر شد راهم در نظر می گیریم

فصل دوم: یکپارچگی, ۲-۲ قوانین یکپارچگی (ادامه)

- مراحل اعمال یکپارچگی: با فرض وجود یک زبان سطح بالا برای بیان قوانین تعریف و ثبت قانون یکپارچگی توسط بخشی از زیر سیستم به مورد اجرا گذاشتن قانون روی DB بلافاصله بعد از ثبت آن تسری قانون به قبل (امکان ایجاد اشکال)

• مزایای عمده:

- ۱- پیشگیری از violation و تضمین validity توسط فود DBMS به جای application های مختلف استفاده کننده
 - ۲- جمع کردن همه قوانین در یکجا, امکان چک کردن مغایرتها و افزودن آنها در قوانین, سهولت تغییر قوانین
 - ۳- امکان سوال در مورد قوانین (در زبان), مثل: قوانین سیستم حقوق؟ قانون SIR1؟
- انواع قوانین یکپارچگی: ۱- قوانین دامنه برای فیلدها به تنهایی و فارغ از روابط,
 - ۲- قوانین روابط برای ایجاد tupleها در یک یا چند جدول و روابط بین آنها

فصل دوم: یکپارچگی, ۲-۳ قوانین یکپارچگی دامنه

- **تعریف:** برای هر $R.A$ (صفت A در رابطه R) روی دامنه D , هر مقدار v که در اثر $update$ یا $insert$ یک tuple مثل t از رابطه R ارائه شده باید در دامنه D باشد.
- **تعریف دامنه:** بیان یک قانون یکپارچگی ساده ولی بسیار مهم, لزوم چک کردن آن در همه ایجادها و اصلاحهای صفات تعریف شده در دامنه.
- **دلیل کنترل:** وجود موارد نقض زیاد و مکرر یکپارچگی دامنه در عملیات روی DB
- **وضعیت فعلی:** حجم بالای کار پژوهشی, پرداختن کمتر در عمل در $DBMS$ ها علیرغم پیشنهادات عملی گوناگون (زیرنویس ص ۳۸)
- **ترتیب تعریف قوانین:** شروع از دامنه های پایه (کارا کتر, عدد صحیح, عدد اعشاری, ...)
- سپس تعریف زیر مجموعه ها با در نظر گرفتن ارث بری
- **مثال:** BNF نمونه ش ۲-۳ ص ۳۹ به عنوان قواعد نحوی ($syntax$) تعریف قوانین
- **کاربرد:** در تعاریف دامنه ش ۲-۴ ص ۴۰ برای $S\&P\ DB$

فصل دوم: یکپارچگی، ۲-۳ قوانین یکپارچگی دامنه (ادامه)

- موارد قابل توجه در مثال: عدم وجود trigger cond, یکسان بودن برای همه, اصلاح یک فیلد یا ایجاد رکوردی حاوی یک فیلد
- عبارت else مشابه یک فضای سیستم, عدم وجود معادل system action
- امکان عمل مشابه نمونه اول, صفر کردن مقادیر عددی غلط, ؟ کردن کارا کتر غلط, وادار کردن به rollback
- عدم نیاز به قانون SIR1 در صورت تعریف STATUS به شکل مزبور
- امکان در نظر گرفتن null value و کنترل با IS_NULL
- امکان کاربرد سایر data type ها و متی picture برای کنترل
- استفاده از دامنه ترکیبی مثل تاریخ و کنترل هر بخش جدا گانه
- وجود ترتیب در مقادیر (برمبنای دامنه پدر و ارث بری), امکان اعلام ترتیب none!
- استفاده از توابع کاربرد در تعریف کنترلها و اعمال پاسفها (violation resp)
- مقایسه بین دامنه ای, درست برای مقادیر مرتبط مثل مایل و کیلومتر, نه رنگ و شهر
- کاربرد مثال: تعریف یک رابطه در ش ۲-۵ ص ۱۴۳

فصل دوم: یکپارگی، ۲-۴ قوانین یکپارگی رابطه

- مفهوم: تعریف یک قانون روی کل یک جدول یا رابطه، پذیرش یک tuple برای ایجاد در یک رابطه، روابط بین tuple های یک رابطه و tuple های دیگر رابطه ها
- مثال: نمونه ساده SIR1 ص ۳۷، نمونه کامل ص ۴۵، قواعد نحوی ش ۲-۷ ص ۴۶
- امکان عمل مشابه نمونه اول، صفر کردن مقادیر عددی غلط، ؟ کردن کارا کتر غلط، وادار کردن به rollback
- عدم نیاز به قانون SIR1 در صورت تعریف STATUS به شکل مزبور
- کاربرد مثال: تعریف ش ۲-۵ ص ۴۳
- توضیحات: $C \rightarrow R$ حالت فاصی از نوع ر کورد R مورد اشاره به وسیله مکان نمای C
- $C \rightarrow R.F$ حالت فاصی از نوع فیلد F در ر کورد R مورد اشاره توسط C
- هر مکان نما مختص یک نوع ر کورد
- هر نوع ر کورد دارای مکان نمای پیش فرض
- استفاده از مکان نمای پیش فرض در صورت عدم اعلام مکان نما

فصل دوم: یکپارچگی، ۲-۴ قوانین یکپارچگی رابطه (ادامه)

توضیحات قواعد نحوی: ش ۲-۷ ص ۴۶، موارد ص ۴۴

- ۱- اعلام شرط trigger با when , before و after.
- ۲- لزوم qualify کردن همه پارامترها با مکان نما، تصریح ر کورد فاص و رفع ابهام.
- ۳- لزوم یکسانی سافتار ر کوردها یا فیلدهای قبل و بعد از from (IR11).
- ۴- در from چند گانه (IR11)، لزوم وجود نام سافتار، لزوم یکسانی سافتارها، تطبیق اسامی قبل و بعد از from، همخوانی before با from و رفع ابهام.
- ۵- اشاره کردن هر مکان نما به ر کورد فاص از DB.
- ۶- امکان وجود پارامترهای مشخص با مکان نما و متغیرهای from در constraintها.
- ۷- امکان حذف شرط trigger اگر constraint حاوی اقلا یک ارجاع با مکان نما باشد، شرط پیش فرض: after updating and after inserting
- ۸ و ۹- مشابهت before/after change با جمله شرطی، ص ۴۷، undo با reject.
- ۱- after و before چند گانه معادل تر کیب شرطها.
- ۱۱- شرط when committing برای تست نتیجه نهایی، مثال انتقال پول و ر کورد حاصل جمع.
- ۱۲- قوانین یکپارچگی روابط شامل چند رابطه، در مورد دامنه شامل فقط یک دامنه.

فصل دوم: یکپارچگی، ۲-۴ قوانین یکپارچگی رابطه (ادامه)

مثالها و توضیحات: ش ۲-۷ ص ۴۸ تا ص ۵۸

- توضیح: در مورد حالات مختلف نمونه اول.
- تکلیف شفاهی: بررسی و درک سایر نمونه ها.

بررسی و درک نکات ص ۵۷ و ۵۸
آمادگی برای ارائه در کلاس برای سایرین
بحث جمعی روی نتیجه در جلسه آینده

فصل دوم: یکپارچگی, ۲-۵ محدودیتهای یکپارچگی fanset

- سافتار fanset: ش ۲-۸ ص ۵۹, وجود دو fanset
- نکته: اهمیت نقش fanset در یکپارچگی در پایگاه داده مدل شبکه
- توضیح: یک fanset, شامل S-SP در آن S پدر و SP فرزند دیگری شامل P-SP در آن P پدر و SP فرزند
- عدم وجود فیلهای P# و S# در نوع رکورد SP
- اصلی (essential) بودن دو fanset, از بین رفتن داده ها در صورت حذف fanset
- غیر اصلی بودن در صورت وجود کلیدها در SP, باقی ماندن داده ها با حذف fanset
- کاربرد اصلی fanset: اعمال یکپارچگی ارجاعی (referential)
- مثال: هیچ SP نمی تواند بدون حضور (بدون اتصال به) S و P متناظر وجود داشته باشد!
- عملیات: با حذف P برای سادگی, عملیات در ص ۵۹ برای فقط S و SP
- شرح عملیات در ادامه

فصل دوم: یکپارچگی، ۲-۵ محدودیتهای یکپارچگی fanset (ادامه)

- (1) **INSERT SP**: حالت کامل در ص ۶، **SP** بدون **S** قابل ایجاد نیست (فرزند بی پدر)
 - (2) **DELETE S**: حالت کامل ص ۶، لزوم نوشتن تو در تو،
نمونه اول **FSDC** برای حذف وابستگان،
نمونه دوم **FSDF** برای عدم حذف در صورت وجود وابسته
نمونه سوم **FSDD** برای قطع وابستگی که به دلیل نقض قواعد یکپارچگی پذیرفته نیست
 - (۳) **UPDATE S.S#**: لزوم نوشتن تو در تو به دلیل عدم وجود **S#** در **SP**
 - (۴) **CONNECT**: ص ۶، عمل غلط به دلیل فرض وجود **SP** بدون **S**
 - (5) **DISCONNECT**: ص ۶، باز هم عمل غلط به دلیل ایجاد **SP** بدون **S**
 - (6) **RECONNECT**: ص ۶، با فرض اتصال **SP** به یک **S** دیگر
- نتیجه: مزیت **fanset** تو در تو شدن رایگان به عناصر زیر دست، نیاز به عملیات اضافی برای ایجاد و حذف

فصل دوم: یکپارچگی، ۲-۶ triggered procedures

- قوانین یکپارچگی: حالت فاصی از triggered procedures, فرم کلی در ص ۶۲
 - تشابه: مقایسه با on unit در برنی زبانها, بلو کهای دینامیک, برای انجام عمل با after و when به جای on, عمل نزدیک به انجام با before
 - مثال: TP1 ص ۶۲ برای حذف آبشاری, آثار جانبی حذف یک رکورد
 - نکته: امکان عدم اطلاع کاربر از آبشاری شدن حذفها, لزوم اعلام, مثال محیط access
 - کاربرد دیگر T P: به روز کردن فیلدهای محاسباتی توسط DBMS و نه کاربر, در سافتن viewها بعد از اصلاح عناصر در گیر به روز رسانی view با T P, مثال ص ۶۳
 - سایر کاربردها ص ۶۳ فیلد مجازی, ایمنی, ارزیابی عملکرد, دیباگ, کنترل ISR, استثناها
 - وضعیت فعلی: موضوع TP در دست تحقیق, کمتر پیاده شده
- تکلیف هفته: یکپارچگی در محیطهای DBMS کتاب ص ۶۴ و حداقل سه محیط دیگر**
- موارد ص ۷۲