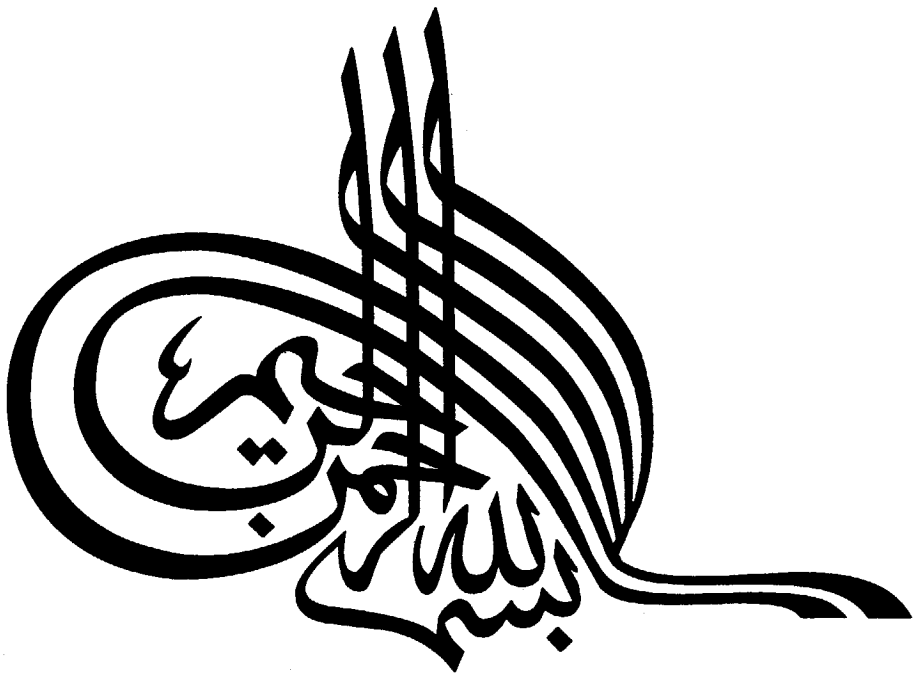

٣٠٠ ٧٨

۱۳۷۹/۵/۲۵

مرکز اطلاعات و مدارک علمی ایران
تقدیم مدارک

دانشگاه علم و صنعت ایران
دانشکده مهندسی کامپیوتر

ابزار طراحی بانک اطلاعات شیئ‌ـرابطه‌ای(در راستای کلان تراکنش)

بابک سپاهی بروجنی

پایان‌نامه برای اخذ درجه کارشناسی ارشد

در رشته

مهندسی کامپیوتر نرم‌افزار

استاد راهنما:
دکتر مصطفی حق‌جو

۱۷۱۳۹

۱۳۷۸

۳۰۰۷۸

تقدیم به

پدر و مادر

بزرگوارم

چکیده

پژوهش‌های بسیاری در مقولهٔ بانک اطلاعات، چه مدل رابطه‌ای و چه دیگر مدل‌ها صورت پذیرفته است. در بین مدل‌های غیر رابطه‌ای ، مدل شیئ‌گرا حرف اول را می‌زند. مدل رابطه‌ای با پشتوانهٔ تئوریک بسیار قوی و مقبولیت عام بی‌نظیر، همچنان اهمیت خود را حفظ نموده و در کاربردهای سنتی ، یکه‌تاز باقی مانده است ولی در کاربردهای پیشرفته فاقد کارایی لازم است. درمورد آیندهٔ بانک‌های اطلاعات، سه دیدگاه کلی مطرح می‌شود:

الف) جایگزینی مدل رابطه‌ای با مدل شیئ‌گرا.

ب) حفظ استقلال دو مدل در محدودهٔ کاری خود.

ج) غنی‌سازی مدل رابطه‌ای با مفاهیم شیئ‌گرایی

بانک اطلاعات شیئ_رابطه‌ای نتیجهٔ دیدگاه سوم است.

کاربردهای سنتی بانک اطلاعات به سمت نامتمرکز شدن پیش می‌روند.موفقترین مدل تراکنش غیرسنتی مدل"تراکنش‌های تودرتو(Nested Transactions)" است. مدل‌های"تراکنش‌های تودرتوی باز"(Open Nested Transactions)جهت رفع کاستی‌های مدل تراکنش‌های تودرتو پدید آمده‌اند.کلان تراکنش (MegaTransaction)،مدل تراکنش تودرتوی باز است که مدل رابطه‌ای،شیئ‌گرا و شیئ_رابطه‌ای را پشتیبانی می‌کند.

در این پروژه "ابزار طراحی بانک اطلاعات شیئ_رابطه‌ای" طراحی و پیاده‌سازی شده‌است.در این ابزار شمای بانک اطلاعات، با استفاده از یک محیط گرافیکی بفرمی مشابه دیاگرام موجودیت_رابطه توسعه‌یافته ایجاد می‌گردد و درنهایت کد پیاده‌سازی شمای طراحی شده در یکی از قالب‌های SQL[1] ، ODL[2] یا Unisql[3] تولید می‌شود.

---

1. Structured Query Language

2. Object Definition Language

3. یکی از محصولات موفقی که در زمینهٔ مدیریت بانک‌های اطلاعات شیئ_رابطه‌ای توسط شرکت Unisql عرضه گشته است

تقدیر و تشکر:

ضمن سپاس بیکران خداوند، بر خود لازم می‌دانم، از استاد محترم جناب آقای دکتر مصطفی حق‌جو که با ارائه راهنمایی‌های مدبرانهٔ خود، نظارت و سرپرستی این پروژه را به عهده داشته‌اند، صمیمانه تشکر و قدردانی نمایم. همچنین از اعضای هیأت داوری، آقایان دکتر محمدرضا کنگاوری و دکتر علی‌اصغر صیفی‌کار، بخاطر حضور در جلسهٔ دفاعیه صمیمانه تشکر نموده و سپاس خود را حضور کلیهٔ کسانی که بنحوی در انجام این پروژه مساعدت نمودند، تقدیم می‌دارم.

بهمن ماه ۱۳۷۸

فهرست

عنوان	صفحه
۱) مدل‌های متداول بانک اطلاعات	۱
۱_۱) نقاط قوت و ضعف بانک‌های اطلاعات رابطه‌ای	۴
۱_۲) خصوصیات بانک‌های اطلاعات شیئ‌گرا	۵
۱_۲_۱) خصوصیات مدیریت داده‌ها	۶
۱_۲_۲) خصوصیات مدیریت اشیاء	۶
۱_۳) مزایای بانک‌های اطلاعاتی شیئ‌گرا	۸
۱_۴) توسعه و تکامل مدل رابطه‌ای	۱۰
۱_۴_۱) کلاس‌بندی در مدل رابطه‌ای	۱۰
۱_۴_۲) انواع دادهٔ انتزاعی در مدل رابطه‌ای	۱۳
۱_۴_۳) مدل‌سازی اشیاء مرکب در XSQL	۱۴
۱_۴_۴) مدل رابطه‌ای تودرتو (NF2)	۱۶
۲) بانک اطلاعات شیئ_رابطه‌ای	۲۰
۲_۱) مقایسه مدل رابطه‌ای و مدل شیئ‌گرایی	۲۱
۲_۲) بزرگنمایی‌های مدل شیئ‌گرایی	۲۵
۲_۲_۱) برطرف سازی نیاز به JOIN	۲۵
۲_۲_۲) برطرف سازی نیاز به کلید	۲۵
۲_۲_۳) برطرف سازی نیاز به یک زبان غیررویه‌ای	۲۵
۲_۲_۴) نقض خاصیت دربرگیری با پرس‌وجو	۲۶
۲_۲_۵) پشتیبانی کامل از بانک اطلاعات چندرسانه‌ای	۲۶
۲_۳) بانک اطلاعات شیئ_رابطه‌ای (ORDB)	۲۷
۲_۳_۱) قابلیت روابط تودرتو(Nested Relation)	۲۸
۲_۳_۲) قابلیت دربرگیری (Encapsulation)	۳۰
۲_۳_۳) قابلیت وراثت (Inheritance)	۳۰
۲_۳_۴) ارتباط با کلان تراکنش	۳۳

عنوان | صفحه

۴_۲) مطالعهٔ بانک اطلاعات شیئ_رابطه‌ای اوراکل ۸ ۳۶

۱_۴_۲) نوع‌های دادهٔ تعریف شده توسط کاربر ۳۶

۲_۴_۲) ارث‌بری ۳۸

۳_۴_۲) تایپ ارجاع (REF) ۳۸

۴_۴_۲) تایپ Collection ۳۹

۵_۴_۲) مجتمع‌سازی اشیاء و رابطه‌ها ۴۰

۶_۴_۲) به‌روز رسانی از طریق جدول مجازی ۴۱

۷_۴_۲) مترجم تایپ اوراکل ۴۲

۸_۴_۲) دسترسی به داده‌های رابطه‌ای با جاوا ۴۲

۵_۲) مطالعهٔ بانک اطلاعات شیئ_رابطه‌ای PostgresSQL ۴۴

۱_۵_۲) معماری Postgres ۴۴

۲_۵_۲) مدیریت بانک اطلاعات در Postgres ۴۵

۳_۵_۲) قابلیت‌های پیشرفتهٔ Postgres ۴۶

۴_۵_۲) توسعه‌پذیری زبان پرس‌وجوی PostgresSQL ۴۷

۳) طراحی "ابزار طراحی بانک اطلاعات شیئ_رابطه‌ای" ۵۴

۱_۳) نگاشت مدل توسعه‌یافتهٔ پدیده_ارتباط به مدل شیئ‌گرایی ۵۵

۱_۱_۳) مفاهیم مشترک در دو دیدگاه ۵۶

۲_۱_۳) تفاوت‌های اساسی دو دیدگاه ۵۶

۳_۱_۳) مدل توسعه‌یافتهٔ موجودیت_رابطه ۵۷

۴_۱_۳) مدل شیئ‌گرایی ۵۷

۵_۱_۳) قواعد نگاشت ۵۸

۲_۳) نگاشت مدل شیئ‌گرایی به مدل رابطه‌ای ۶۲

۱_۲_۳) نگاشت اشیاء به جداول ۶۲

۲_۲_۳) نگاشت رابطهٔ توارث به جدول ۶۳

۳_۲_۳) نگاشت روابط غیرسلسله‌مراتبی ۶۶

عنوان	صفحه

۳_۳) طراحی المان‌های ابزار طراحی ۷۱

۱_۳_۳) طراحی المان پدیده (Entity) ۷۲

۲_۳_۳) طراحی المان ارتباط (Relation) ۷۳

۳_۳_۳) طراحی المان رابطهٔ ارث‌بری (Inheritance) ۷۴

۴_۳_۳) طراحی المان رابطهٔ تودرتو (Nested_Relation) ۷۵

۴) پیاده‌سازی "ابزار طراحی بانک اطلاعات شیئ_رابطه‌ای" ۷۶

۱_۴) پیاده‌سازی المان‌های ابزار طراحی ۷۷

۱_۱_۴) پیاده‌سازی المان پدیده (Entity) ۷۷

۲_۱_۴) پیاده‌سازی المان ارتباط (Relation) ۷۹

۳_۱_۴) پیاده‌سازی المان رابطهٔ ارث‌بری (Inheritance) ۸۱

۴_۱_۴) پیاده‌سازی المان رابطهٔ تودرتو (Nested Relations) ۸۲

۲_۴) نام‌گذاری المان‌های ابزار طراحی ۸۳

۳_۴) ساختارهای مورداستفاده درذخیره‌وبازیابی شمای طراحی‌شده ۸۴

۴_۴) نحوهٔ ذخیره‌سازی یک شمای طراحی شده ۸۹

۵_۴) نحوهٔ بازیابی یک شمای موجود ۹۰

۶_۴) خروجی ابزار طراحی بانک اطلاعات ۹۲

۱_۶_۴) ساختار SQL ۹۲

۲_۶_۴) ساختار ODL ۹۴

۳_۶_۴) ساختار UniSQL ۹۷

۵) راهنمای کاربر ابزار ۱۰۰

۱_۵) المان‌های قابل استفاده در ابزار طراحی ۱۰۲

۱_۱_۵) ایجاد المان پدیده ۱۰۳

۲_۱_۵) ایجاد المان ارتباط ۱۰۴

۳_۱_۵) ایجاد المان رابطهٔ ارث‌بری ۱۰۶

۴_۱_۵) ایجاد المان رابطهٔ تودرتو ۱۰۷

عنوان	صفحه
۵_۲) چند راهنمایی کلیدی	۱۰۸
۵_۳) اجرای یک مثال	۱۰۹
۵_۴) خروجی ابزار طراحی بانک اطلاعات	۱۱۰
۵_۴_۱) کد خروجی با فرمت UniSQL	۱۱۰
۵_۴_۲) کد خروجی با فرمت ODL	۱۱۴
۵_۴_۳) کد خروجی با فرمت SQL	۱۱۸
جمع‌بندی و نتیجه‌گیری	۱۲۱
فهرست منابع	۱۲۴

فهرست شکل‌ها

فصل اول

شکل ۱_۱ شیوهٔ سازماندهی اطلاعات در مدل شیئ‌گرایی و رابطه‌ای ۹

شکل ۱_۲ کلاس‌بندی در مدل رابطه‌ای ۱۰

شکل ۱_۳ نمونه‌ای از کلاس‌بندی مدل رابطه‌ای ۱۲

شکل ۱_۴ ساختار سلسله‌مراتبی چندسطحی ۱۵

شکل ۱_۵ ساختار اساسی مدل داده‌ای رابطه‌ای ۱۶

شکل ۱_۶ ساختار اساسی مدل داده‌ای تودرتو (NF2) ۱۷

شکل ۱_۷ فرم درختی مدل داده‌ای تودرتو ۱۸

شکل ۱_۸ فرم جدولی مدل داده‌ای تودرتو ۱۸

شکل ۱_۹ جدول اطلاعات مدارک در مدل تودرتو ۱۹

فصل دوم

شکل ۲_۱ نمایش اطلاعات در بانک شیئ‌گرا و رابطه‌ای ۲۳

شکل ۲_۲ نمایش اشیاء در بانک اطلاعات و حافظه ۲۴

شکل ۲_۳ نمونه‌ای از بانک اطلاعات شیئ_رابطه‌ای ۲۹

فصل سوم

شکل ۳_۱ ارتباط یک به چند ۵۹

شکل ۳_۲ رابطهٔ ارث‌بری ۶۰

شکل ۳_۳ نگاشت شیئ به جدول با ID افزوده ۶۲

شکل ۳_۴ توارث اجتماع ۶۳

فصل سوم

شکل ۳_۵	نگاشت رابطهٔ‌ارث‌بری به جدول	۶۴
شکل ۳_۶	رابطهٔ چند به چند	۶۷
شکل ۳_۷	نگاشت رابطهٔ چند به چند به جداول	۶۷
شکل ۳_۸	رابطهٔ یک به چند	۶۸
شکل ۳_۹	نگاشت رابطهٔ یک به چند به جداول	۶۸

فصل چهارم

شکل ۴_۱	فیلدهای جدول Entity_Table	۸۴
شکل ۴_۲	فیلدهای جدول Has_Table	۸۵
شکل ۴_۳	فیلدهای جدول Is_a_Table	۸۶
شکل ۴_۴	فیلدهای جدول Rel_Table	۸۷
شکل ۴_۵	فیلدهای جدول Last_names	۸۸

فصل پنجم

شکل ۵_۱	نمایی از ابزار طراحی	۱۰۱
شکل ۵_۲	المان‌های قابل استفاده در ابزار طراحی	۱۰۲
شکل ۵_۳	پدیدهٔ Engineer و لیست پردازش	۱۰۳
شکل ۵_۴	فرم انتخاب یک ارتباط برای ویرایش	۱۰۴
شکل ۵_۵	فرم تنظیم یک ارتباط	۱۰۵
شکل ۵_۶	فرم تنظیم رابطهٔ ارث‌بری	۱۰۶
شکل ۵_۷	فرم تنظیم رابطهٔ تودرتو	۱۰۷

فصل اول

مدل‌های متداول بانک اطلاعات

مقدمه:

سیستم‌های مدیریت بانک اطلاعات (*DBMS*)، سیستم‌های نرم‌افزاری می‌باشند که امکاناتی را جهت مدل‌سازی، پردازش و استخراج اطلاعات در مقیاس وسیع از داده‌های مرتبط با هم فراهم می‌کنند[۱]. این حجم وسیع اطلاعات عموماً بصورت اشتراکی بین چندین کاربر و در یک بازهٔ زمانی طولانی وجود دارند. در واقع سیستم مدیریت بانک اطلاعات یک پیاده سازی از مدل داده‌ای (*Data Model*) به همراه مجموعه‌ای از امکانات برای کمک به نگهداری بانک اطلاعات است.

امکاناتی که یک سیستم مدیریت بانک اطلاعات فراهم می‌کند عبارتند از:

- **مانایی (*Persistency*)** : کاربران می‌توانند بدون ایجاد فایلها، که محتوی داده‌ها می‌باشند و توسط نامشان به آنها ارجاع می‌شود، اجزاء داده‌ای ایجاد کنند که در حافظه مانا مقیم شوند (به جای حافظهٔ فرار *Volatile*).
- **کنترل همروندی (*Concurrency Control*)** : کاربران متعدد می‌توانند به صورت همزمان به یک بانک اطلاعات دسترسی داشته باشند بدون اینکه به علت تأثیرات متقابل عملگرهای همزمان، نتایج ناسازگار و نامطلوب ایجاد شود.
- **مدیریت تراکنش (*Transaction Management*)** : کاربران می‌توانند از مدیریت بانک اطلاعات بخواهند که یک دنباله از عملگرهای بانک اطلاعاتی در قالب یک واحد تفکیک‌ناپذیر اجرا شوند که این واحد یا بصورت کامل اجرا شود یا اصلاً اجرا نشود. بانک اطلاعات سازگاری و انسجام خود را قبل و بعد از اجرای این تراکنش حفظ می‌کند.
- **بازگرد (*Recovery*)** : اطلاعات به علت بروز خرابی سیستم، از بین نمی‌روند و سازگاری خود را نیز حفظ می‌کنند. اگر در وسط یک تراکنش خرابی رخ داد، تغییرات ایجاد شده توسط تراکنش بازگرد داده می‌شوند و بانک اطلاعات به وضعیت قبل از اجرای تراکنش برمی‌گردد.

- **زبان پرس و جو (*Query Language*)** : سیستم مدیریت بانک اطلاعات زبانهای پرس و جوی سطح بالا و ساده به عنوان رابط دسترسی به اطلاعات فراهم می‌کند.
- **امنیت (*Security*)** : سیستم مدیریت بانک اطلاعات برای محافظت اطلاعات از دسترسی‌های غیر مجاز امکاناتی فراهم می‌کند.

موفق‌ترین مدل بانک اطلاعات تاکنون مدل رابطه‌ای بوده است که داده‌ها را در قالب جداول مرتبط سامان می‌دهد. مجموعهٔ ساختارها، عملگرها و محدودیتها در مدل رابطه‌ای محدود و ثابت می‌باشد بنابراین هر ساختار و عملگری که در یک کاربرد مورد نیاز باشد باید به این مجموعهٔ ثابت نگاشت شود. هرچه برنامه کاربردی پیچیده‌تر و کاملتر شود این نگاشت نیز پیچیده‌تر می‌شود و نیاز به مقدار زیادی کد خواهد داشت. برنامه کاربردی با حجم کد بالا، نگهداری و توسعه برنامه را مشکل و پرهزینه می‌کند و علاوه بر آن به علت نیاز به اجرای عملگرهایی روی بانکهای اطلاعات چندگانه، دسترسی خیلی کند می‌شود.

سیستمهای مدیریت بانک اطلاعات شیئ‌گرا برای حل چنین مشکلاتی قدم به عرصه بانک اطلاعات گذاشته‌اند. سیستم مدیریت بانک اطلاعات شیئ‌گرا یک سیستم مدیریت است که علاوه بر امکانات فوق خواص زیر را نیز داراست [۲] :

۱) شناسهٔ شیئ (*Object Identity*)

۲) داده فعال (*Active Data*)

۳) کلاس‌بندی (*Classification*)

۴) عمومی‌سازی (*Generalization*)

۵) درپرگیری (*Encapsulation*)

۶) اشیاء مرکب (*Composition*)

۷) توسعه‌پذیری (*Extensibility*)

در ادامه به بررسی و تحلیل نقاط ضعف و قوت مدل داده‌ای رابطه‌ای و بررسی مدل داده‌ای شیئ‌گرا و تحلیل آن می‌پردازیم.

۱_۱) نقاط قوت و ضعف بانک‌های اطلاعات رابطه‌ای :

بانک اطلاعات رابطه‌ای نسبت به روش‌های سلسله مراتبی و شبکه‌ای، امتیازهای زیادی دارد، از آن جمله می‌توان به موارد زیر اشاره کرد [۸] :

۱. درک نمایش جدولی مورد استفاده در روش رابطه‌ای برای کاربران و پیاده‌سازی فیزیکی آن بسیار ساده‌تر است.

۲. تبدیل هر نوع ساختار بانک اطلاعاتی به صورت مجازی به نوع رابطه‌ای به نسبت ساده است، بنابراین به این روش می‌توان به عنوان یک روش عام (*Universal*) برای نمایش نگاه کرد.

۳. پیاده سازی عملیات تجزیه، ترکیب و در نهایت ایجاد رابطه‌های جدید مورد نیاز برای کاربردهای ویژه به سادگی انجام می‌شود.

۴. پیاده سازی روش‌های مربوط به کنترل دسترسی به داده‌های حساس به سادگی و بوسیلهٔ روش‌های دسترسی یا اجازه‌دهی (*Authority*) انجام می‌گیرد.

۵. برخی از انواع جست‌وجو (*Search*) می‌تواند به مراتب سریع‌تر از طرح‌هایی که در آن به تعقیب یک سلسله طولانی از اشاره‌گرها نیازمند است، صورت گیرد.

۶. اصلاح (*Modify*) ساختار رابطه‌ای به مراتب ساده‌تر از ساختارهای سلسله مراتبی و شبکه‌ای صورت می‌گیرد. این مطلب در محیط هایی که مسألهٔ انعطاف‌پذیری دارای اهمیت بسیار است، حیاتی است.

۷. وضوح (*Clarity*) و مشاهده‌پذیری (*Visibility*) بانک اطلاعات به وسیلهٔ ساختار رابطه‌ای بیشتر می‌شود. جست‌وجو در جدول‌های داده به مراتب ساده‌تر از باز کردن ارتباط‌های داخلی پیچیدهٔ ممکن و دلخواه بین عناصر داده در مکانیزمی است که اشاره‌گرها آن را ایجاد می‌کنند.

سیستم‌های بانک اطلاعات رابطه‌ای اولیه از ضعف کارایی شدیدی رنج می‌بردند. با ایجاد و توسعهٔ برخی از ویژگیها در این سیستم‌ها، با گذشت زمان از ضعف کارایی این سیستم‌ها کاسته شد. رشد بسیار سریع تکنولوژی سخت‌افزار نیز کمکی به این امر بود. با وجود این و برغم این که مدل رابطه‌ای و روش‌های نرمال‌سازی (*Normalization*) بسیاری از مشکلات مرحلهٔ طراحی بانک اطلاعات را حل کرده است و نیز