

طراحی پایگاه داده منطقی با مدل های مشخص ارتباطی نهادی از منظر هستی

شناختی

چکیده

استدلال بر این است که مدل های ارتباطی نهاد روشن از نقطه نظر هستی شناسی در مقایسه با مدل های نامشخص هستی شناسی دامنه های جهان واقعی را دقیق تر مدل سازی می کند. با این حال، تحول این مدل ها به مدل ارتباطی در سطح منطقی تا بدین جا به اندازه ای مطالعه نشده است تا قوانین تحول جدید را تدوین کند. این مقاله مجموعه ای از قوانین تحول را ارائه می کند تا مدل های هستی شناسی مفهومی روشن را به مدل های منطقی تبدیل کند. در نهایت، مقایسه ای از دو مدل منطقی را انجام می دهیم که از مدل های مشخص و نامشخص هستی شناسی با استفاده از یک معیار کیفیت ترکیبی از منابع موجود توسعه می یابد. نتایج اولیه از این تحقیق در حال انجام حاکی از آن است که کیفیت مدل منطقی توسعه یافته از مدل مفهومی مشخص هستی شناسی نسبت به هم تالی نامشخص هستی شناسی خودش برتری دارد.

1. مقدمه

مدلسازی مفهومی فعالیتی است که در طول مراحل ابتدائی کار توسعه سیستم های اطلاعاتی انجام می شود، که در آن یک نمودار گرافیکی تولید می شود که در حال ارائه پدیده کلمه واقعی از یک دامنه کاربرد است. محققان توانائی بیان متدولوژی مدلسازی مفهومی خوب شناخته شده، مدل ارتباطی موجودیت (ER) [4]، را با استفاده از هستی-

شناسی [7] بهبود می‌بخشند که ساختار و رفتار جهان واقعی را تشریح می‌کند. این مدل‌ها نمودارهای ارتباطی نهاد مشخص هستی‌شناسی (OC-ERD) نام دارد [8].

به منظور استفاده از مزیت OC-ERD ها، چنین مدل‌هایی را می‌بایست بصورت مناسبی به طرح منطقی پایگاه داده (RDS) [9, 10] بدون از دست دادن شماتیک‌های OC-ERD تبدیل نمود. مجموعه کنونی از قوانین توسعه یافته برای تبدیل ERD های عمومی برای OC-ERD ها کاملاً قابل اجرا نیست. بر این اساس، این مقاله برخی از نتایج ابتدائی از یک مطالعه تحقیقاتی در حال انجام را ارائه می‌کند تا قوانین تحول جدید و روش ارزیابی کیفیت مدل منطقی حاصله از این قوانین توسعه یابد.

بقیه مقاله به شرح زیر است. بر این اساس، بخش 2 یک ERD نامشخص هستی‌شناسانه (OUC-ERD) از یک سناریوی جهان واقعی مخصوص و تحول آن را همراه با موضوعات تحول نشان می‌دهد. بخش 3 نسخه مشخص هستی‌شناسانه از OUC-ERD، OC-ERD، و موضوعات تحول آن با استفاده از الگوریتم موجود را ارائه می‌کند. همچنین این بخش یک الگوریتم جدید ارائه شده و تحول با استفاده از آن را ارائه می‌کند. در بخش 4، یک معیار کیفیت برای ارزیابی کیفیت دو نوع از RDS های منتج از هر دو رویکرد ارائه می‌شود و کیفیت مقایسه نشان داده می‌شود. در نهایت، نتایج مقدماتی و لثر آتی از این مطالعه تحقیقاتی در حال انجام در بخش 5 ارائه می‌شود.

2. موضوعات طراحی پایگاه داده منطقی با OUC-ERD ها

اکنون یک OUC-ERD و موضوعات تبدیل آن به مدل ارتباطی با استفاده از ER موجود به الگوریتم تحول ارتباطی را ارائه میکنیم.

شکل 1 یک OUC-ERD است که یک شرکت را در دنیای واقعی ارائه میکند. نمودار شامل یک دودویی 1:1 (یک به یک) و «مدیران» نوع ارتباط بهینه بین دو نهاد از نوع «کارکنان» و «دپارتمان» است. نسبت کاردینالیتی (0، 1) بین نوع نهاد کارکنان و نوع رابطه دو معنا دارد یعنی یک کارمند ممکن نیست یک بخش دپارتمان را مدیریت کند یا

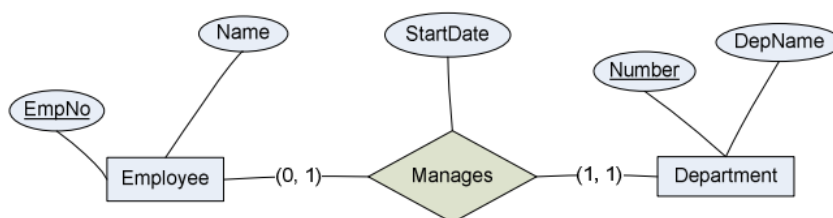
کارمندی که دپارتمان را مدیریت می کند تنها یک دپارتمان را می تواند مدیریت کند. مشخصه «StartDate» نشان دهنده مورخه ای است که یک کارمند شروع به اداره یک دپارتمان مخصوص می کند.

اعتقاد بر این است که روابط اختیاری مانند رابطه فوق از جانب کاربران معمول حوزه به سختی درک شود. هستی-شناسی مانع استفاده از انواع روابط اختیاری است و استفاده از انواع روابط اجباری با ساب تایپینگ را توصیه می کند [8]. بدین ترتیب، نمودار نشان داده شده در شکل 1 یک نمودار ER نامشخص هستی شناسانه (OUC-ERD) است.

الگوریتم تحول موجود ارائه شده توسط المصری و ناواته [9] در زیر داده شده است.

برای هر نوع موجودیت منظم E در طرح ER، یک رابطه L ایجاد کنید که شامل تمام خصوصیات E باشد. یکی از ویژگیهای اصلی E را به عنوان کلید اصلی L انتخاب کنید.

برای هر نوع رابطه باینری 1:1 در طرح ER، روابط S و T شناسایی می شود که مربوط به انواع موجودیت مشارکت کننده در R است. یکی از روابط S را انتخاب کنید که شامل یک کلید خارجی در کلید اصلی S از T است. تمامی ویژگیهای ساده R را بعنوان ویژگیهای S به حساب آورید.



شکل 1. ERD نامشخص هستی شناسانه (OUC-ERD)

شکل 2 RDS به دست آمده توسط تبدیل OIC-ERD با استفاده از الگوریتم فوق را نشان می دهد.

Employee | EmpNo | Name

Department | DepNo | DepName | EmpNo | StartDate

شکل 2. RDS به دست آمده با تبدیل OUC-ERD در شکل 1 با استفاده از ERD موجود برای الگوریتم تبدیل

مدل منطقی.

از آنجا که RDS در شکل 2 توسط قوانین تحول دقیق زیر به دست می‌آید، پس آن را تحول درون الگوریتمی می‌نامیم.

با اینحال، برخی ابهامات غالب بر RDS وجود دارد. دلایل برای داشتن کلید ابتدائی (PK)، EmpNo کارکنان بعنوان یک کلید خارجی (FK) در رابطه دپارتمان به خاطر موارد مزبور است: در ابتدا به منظور ارائه ارتباط «مدیران» OUC-ERD، و سپس به منظور اشاره بر کارکنانی است که قرار است دپارتمان را اداره کنند. با اینحال، بطور کلی، این ممکن است برای شخصی غیر از طراح RDS به منظور درک این دلایل سخت باشد.

بطور مشابه، معنای ورود مشخصه «StartDate» در رابطه دپارتمان درک نمی‌شود. کلمه «مدیران» در ERD در طول تحول حذف می‌شود و در نتیجه این در RDS شامل نمی‌شود. این احتمالاً باید به ERD رجوع کند و گاهی اوقات حتی به سناریوی دامنه برای یافتن دلیل اشاره دارد. راه‌حل اسفاده شده در عمل فعلی به منظور حل این ابهامات تغییر نام ویژگی‌ها با استفاده از پیشوندهای مناسب است. بر این اساس، EmpNo و StartDate در دپارتمان به ترتیب به Mgr_EmpNo و Mgr_StartDate تغییر می‌یابد. پیشوند «Mgr» کلمه «مدیر» را نشان می‌دهد. RDS اصلاح شده در شکل 3 داده شده است.

با اینحال، معتقدیم که این تنظیمات خارج از قوانین است. هر یک از قوانین تحول یا تبدیل ویژگی‌های نامگذاری مجدد، روش نامگذاری مجدد و چگونگی تصمیم‌گیری درباره پیشوندها برای نامگذاری مجدد را ارائه نمی‌کنند. از آنجا که این تنظیمات بیرون از الگوریتم تبدیل قانونی ایجاد می‌شوند، پس آن را تنظیمات الگوریتم اضافه می‌نامیم. این فرایند اضافه الگوریتم از تخصص دامنه طراح، بینش، داوری، و نظر شخصی استفاده می‌کند.

Employee | EmpNo | Name

Department | DepNo | DepName | Mgr_EmpNo | Mgr_StartDate

شکل 3. تنظیمات الگوریتم اضافی ایجاد شده برای RDS تبدیل شده در شکل 1. ویژگی‌های «EmpNo» و

«StartDate» به «Mgr_EmpNo» و «Mgr_StartDate» نامگذاری مجدد می‌شود. پیشوند اختصاری

«Mgr» حاکی از کلمه «مدیر» است.

تنظیمات اضافه الگوریتم چندین موضوع را به شرح زیر ایجاد می‌کند.

a. نامگذاری دوباره مشخصات خارج از الگوریتم،

b. چگونه یک کلمه خاص به منظور استفاده تصمیم‌گیری می‌شود تا پیشوندی را برای نامگذاری مجدد ایجاد کند، برای مثال «مدیر» در این مورد،

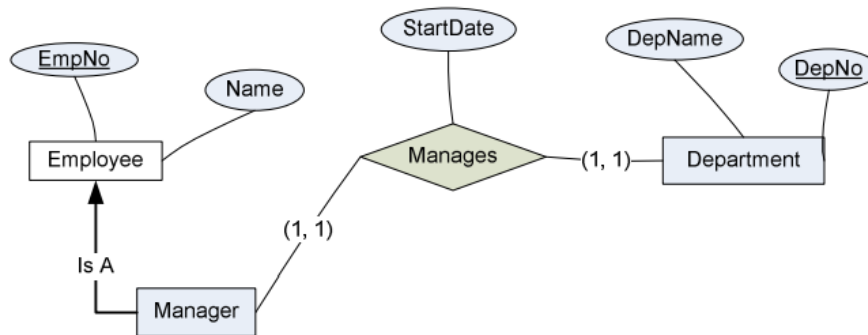
c. چگونه پیشوند، برای مثال «Mgr» ایجاد می‌شود.

حتی پس از تنظیمات الگوریتم اضافی، به نظر می‌رسد برخی از سناریوها هنوز هم مبهم هستند. برای مثال، OUC-ERD نشان می‌دهد که یک کارمند تنها یک دپارتمان را اداره می‌کند یا اصلاً دپارتمانی را مدیریت نمی‌کند، و این باید در RDS در شکل 3 منعکس شود. با اینحال، این مشخص نیست که آیا کارکنان یک مدیر ذکر شده توسط ویژگی «Mgr_EmpNo» مجدداً نامگذاری شده می‌توانند تنها یک یا چند دپارتمان را اداره کنند یا خیر، به این دلیل که همان Mgr_EmpNo در چند تائی‌های مختلف تکرار می‌شوند. این نشان می‌دهد که برخی از اطلاعات معنای در طول مدت تبدیل از دست رفته است و حتی با فرایند تنظیم الگوریتم اضافی نیز مجدداً ایجاد نمی‌شود. فرض کنید یک طراحی با دامنه آشنا نیست، دامنه‌ای همچون OUC-ERD که به منظور طراحی یک RDS داده شده است. احتمالاً، طراح تبدیل درون الگوریتم را انجام دهد که با RDS نشان داده شده در شکل 2 یافته می‌شود. با اینحال، وی گاهی اوقات ممکن است کل طراحی را بهم بریزد زمانی که ورود می‌کند تا تنظیمات الگوریتم اضافی را با توجه به نبود دانش دامنه انجام دهد.

3. OC-ERD و تبدیل آن

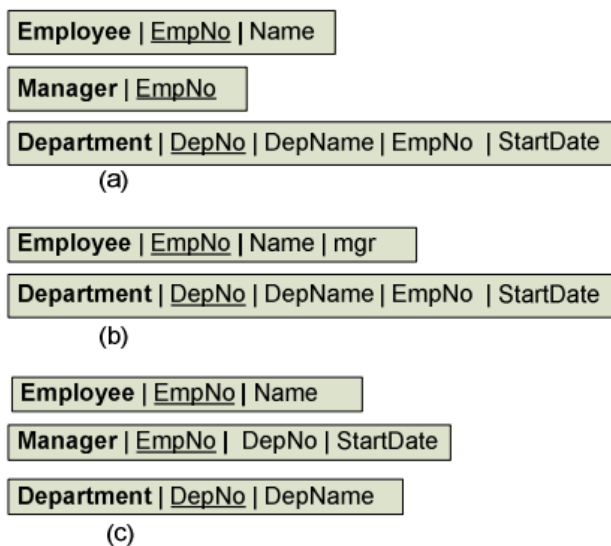
در OC-ERD نشان داده شده در شکل 4، رابطه بهینه با یک رابطه اجباری بین دپارتمان و یک زیر نوع جدیداً خلق شده به نام «مدیر» کارکنان حل می‌شود؛ کارمند مبدل به یک زیر نوع می‌شود.

اکنون نسبت‌های کاردینالیته بسیار روشن و آشکار است، و نشان‌دهنده‌ی این موضوع هستند که هر «مدیر» بطور قطعی یک «دپارتمان» را «مدیریت می‌کند» و یک و تنها یک «دپارتمان» در هر زمان معین را «مدیریت می‌شود».



شکل 4. ERD مشخص هستی شناسانه (OC-ERD)

تبدیل OC-ERD با استفاده از قوانین موجود سه RDS بهینه مجزا را ارائه می‌کند همانطور که در شکل 5 داده شده است. انواع مختلف ابهامات در هر RDS ارائه شده است، برای مثال، در شکل 5(الف)، افزودگی‌های اطلاعات، عدم توانائی به منظور شناسایی ارتباط زیر نوع سوپر نوع و نهاد یا موجودیت دقیقی که در رابطه FK مشارکت می‌کند، و فقدان توانائی به منظور شناسایی هدف ارتباط FK و غیره. همچنین شکل 5(ب) و (ج) حاوی برخی از ابهامات است.



شکل 5. تبدیل OC-ERD در شکل 4 با استفاده از قوانین موجود، (الف) RDS - 1 بهینه، (ب) RDS-2 بهینه،

mgr یک مشخصه نوع بولی است که حاکی از این است که چندتائی متعلق به آن است، (ج) RDS-3 بهینه

مثال ساده فوق نشان می‌دهد که قوانین تبدیل موجود برای هر دو OUC-ERD ها و OC-ERD ها کارائی مناسبی ندارد. در نتیجه، این امر امکانپذیر نیست که یک RDS غیر مستقیم، سراسر است و کاملاً قابل اعتماد به دست آید که تمامی اطلاعات ارائه شده در هر یک از RDS ها را ارائه کند. در نتیجه، این موضوع ضروری است که مجموعه جدید یا تغییر یافته‌ای از قوانین به منظور تطبیق با OC-ERD ها و همچنین OUC-ERD ها توسعه یابد.

مشاهده می‌شود که OC-ERD ها پشتیبانی نسبتاً بیشتری را برای جبران یا رفع ابهامات در فرایند تبدیل ارائه می‌کنند. برای مثال، موضوع حل پیشوند «Mgr» برای نام‌گذاری مجدد ویژگی‌ها. در مورد OUC-ERD، این کاملاً مبتنی بر نظر و تخصص طراح است. با اینحال، در مورد OC-ERD، خود ERD کلمه «مدیر» را بعنوان یک زیر نوع در حال مشارکت در ارتباط مربوط به «مدیران» اشاره می‌کند، به طوری که کلمه را می‌توان به منظور فرم‌دهی یا شکل‌دهی پسوند اختصاری کرد یا کلمه کامل را می‌توان بعنوان پسوند به کار برد اگر که مطلوب باشد. وضعیت موجود این انگیزه را به ما می‌دهد تا یک رویکرد منطقی را برای فرایند نامگذاری مجدد معرفی کنیم که OUC-ERD را به RDS نهایی پیوند می‌دهد.

الگوریتم جدید ارائه شده که OC-ERD ها را تحت پوشش قرار می‌دهد دارای جنبه‌های زیر است.

1. یک مجموعه نامگذاری برای مشخصات، روابط و انواع نهاد در OUC-ERD.
 2. یک مجموعه جدید از قوانین تحول برای تبدیل OC-ERD ها به مدل ارتباطی.
 3. قوانین برای نامگذاری مجدد اسامی ویژگی‌ها در طرح ارتباطی پاسگاه داده.
- الگوریتم ارائه شده که دارای چندین مرحله است و همچنین در فرمی جدید یا اتخاذ شده است [9] در زیر ارائه می‌شود.

مرحله 1: نقشه‌برداری از انواع نهاد منظم

1. برای هر نهاد منظم نوع E در ERID، یک رابطه L را ایجاد کنید که دربرگیرنده‌ی تمامی ویژگی‌های E باشد.

2. یکی از ویژگی‌های کلیدی E را بعنوان کلید اصلی L یعنی PK(L) انتخاب کنید.

3. PK را بعنوان نخستین مشخصه رابطه جدید L در حال تداوم توسط ویژگی‌های باقیمانده E نهاد قراردعی کنید

مرحله 2: نقشه‌برداری بخش‌های زیر نوع / سوپر نوع

هر بخش زیر نوع / سوپر نوع با تعداد m از زیر نوع‌های $\{S_1, S_2, \dots, S_m\}$ و همچنین یک سوپر نوع C را به طرح‌های رابطه از طریق یکی از گزینه‌های زیر تبدیل کنید:

گزینه 1:

1. ایجاد یک رابطه L برای C با ویژگی‌های و کلید اصلی، .

2. ایجاد یک رابطه مجزای برای هر زیر کلاس ، با ویژگی‌های و .

گزینه 2:

1. ایجاد یک رابطه مستقل L با مشخصات و . هر ، ، یک ویژگی نوع بولی است که نشان می‌دهد آیا چندتائی متعلق به زیر کلاس است یا نیست.

2. در L، ویژگی‌های یک زیر نوع مخصوص باید مشتمل بر مشخصه نوع بولی مرتبط آن باشد.

مرحله 3: نامگذاری روابط تبدیل شده

مورد 1: نام سوپر نوع C در حال حاضر در نام زیر نوع S بعنوان آخرین کلمه آن به شرح زیر به حساب آورده می‌شود

$S = XC$ که در آن X یک کلمه جدید مورد استفاده برای شکل‌دهی بخشی از نام زیر نوع است.

در این مورد، خط زیرین یا زیرین خط بایستی در بین X و C برای شکل‌دهی X_C قرار گیرد، به طوری که

$$L_x = X_C$$

مورد 2: زمتمنی که نام سوپر نوع در نام زیر نوع به حساب نمیآید، کل نام زیر نوع بایستی به نام سوپر نوع توسط یک خط زیرین متصل شود تا S_C را شکل دهد، که در آن S نام زیر نوع است و C نام سوپر نوع است.

مرحله 4: نامگذاری یک ویژگی نوع بولی استفاده شده برای تبدیل یک زیر نوع

نام مشخصه نوع بولی، به اصطلاح t ، بایستی با پیوند نام زیر نوع، به اصطلاح S ، توسط یک خط تیره همراه با ردیف متن «Flag» ایجاد شود که کلمه «Flag» را نشان می‌دهد، به طوری که $t = S\text{-Flag}$.

مرحله 5: نقشه‌برداری یک رابطه دودویی نوع R در بین یک نوع نهاد منظم و یک زیر نوع از یک سوپر نوع

فرض کنید که یک زیر نوع S از یک سوپر نوع C دارای یک نوع ارتباط دودویی $1:1$ با یک نهاد منظم نوع E باشد. مورد 1: زیر نوع به یک رابطه جداگانه تبدیل می‌شود.

به منظور تبدیل رابطه نوع R روابط مجزای L_* و L_e را شناسایی کنید که مربوط به نهاد نوع S (زیر نوع) و E (منظم) است که در R مشارکت دارند.

گزینه 1

1. بعنوان یک کلید بیرونی در کلید اولیه، شامل می‌شود و آن بعنوان مجددا نامگذاری می‌شود.

2. یک پسوند اضافه میشود که بعنوان ظاهر می‌شود. حرف (U) نشان میدهد که کلید خارجی منحصر بفرد است.

3. هر ویژگی ساده، به اصطلاح A ، از رابطه نوع R در به حساب می‌آید که پیرو کلید خارجی مشتمل است و آن بعنوان S_A نامگذاری مجدد می‌شود.

4. اگر زیر نوع S در طرح ER رابطه مجزای ویژگی‌های خود آن را در بر نگیرد، که زیر نوع S را نشان می‌دهد بایستی در این گزینه نادیده گرفته شود.

گزینه 2

1. بعنوان یک کلید خارجی در کلید اصلی به حساب می‌آید، همراه با یک پسوند U بعنوان ظاهر می‌شود. کلمه «U» شاه بر این دارد که کلید بیرونی محصر بفرود است.

2. تمامی ویژگی‌های رابطه نوع R در L به تبعیت از کلید بیرونی شامل می‌شود.

مورد 2: زیر نوع با استفاده از یک ویژگی نوع بولی به یک رابطه مستقل همراه با سوپر نوع آن تبدیل می‌شود.

برای هر رابطه نوع R، رابطه واحد L شناسایی می‌شود که مربوط به بخش S/C زیر نوع/ سوپر نوع و رابطه L از نهاد منظم نوع E است که در آن S و E در حال مشارکت در R هستند.

1. شامل کلید خارجی در کلید اصلی از L، است و آن بعنوان مجددا نامگذاری می‌شود.

2. یک پسوند (U) اضافه می‌شود که بعنوان ظاهر می‌شود. کلمه «U» نشان می‌دهد که کلید بیرونی منحصر بفرود است.

3. شامل تمامی ویژگی‌های ساده R بعنوان ویژگی‌های است که به تبعیت از کلید بیرونی مشتمل می‌باشد.

4. شامل هر مشخصه ساده، به اصطلاح A، از رابطه نوع R در است به تبیین از این کلید بیرونی که مشتمل است و بعنوان S_A نامگذاری مجدد می‌شود.

5. اگر زیر نوع S در طرح ER شامل مشخصات خود آن نباشد پس مشخصه نوع بولی شامل در رابطه L می‌بایست برای این گزینه حذف شود.

مرحله 6: نقشه برداری یک رابطه دودویی 1:1 نوع R بین دو زیر نوع مختلف از سوپر نوع‌های متفاوت

فرض کنید که A و B دو زیر نوع متفاوت از دو سوپر نوع مختلف از یک طرح ER باشند و فرض کنید که R یک نوع رابطه دودویی 1:1 است که در بین A و B موجود است.

1. تبدیل یکی از زیر نوع‌ها، به اصطلاح A، به یک رابطه مجزای .

2. با فرض بر این که یک رابطه جداگانه مربوط به زیر نوع B است، و بعنوان یک کلید اصلی از کلید بیرونی ، شامل می‌شود، و آن بعنوان نامگذاری مجدد می‌گردد. این کلید خارجی بایستی به تبعیت از آخرین ویژگی موجود از شامل شود.

3. شامل تمامی ویژگی‌های ساده R در به دنبال کلید بیرونی است.

4. اگر زیر نوع B حاوی ویژگی‌های متعلق به خودش باشد، پس این باید همراه با سوپر نوع خودش با استفاده از یکی از عملیات مناسب زیر تبدیل شود.

(1) دو رابطه جداگانه برای هر زیر نوع B و سوپر نوع آن (مرحله 2 گزینه 1 فوق)

(2) یک رابطه مستقل برای هر دو زیر نوع B و سوپر نوع آن (مرحله 2 گزینه 2 فوق).

5. اگر زیر نوع B حاوی ویژگی‌های متعلق به خودش نباشد پس سوپر نوع آن بایستی تبدیل شود.

مرحله 7: نقشه‌برداری یک نوع ارتباط 1:N: دودویی بین دو زیر نوع متفاوت از سوپر نوع‌های مختلف

فرض کنید که A و B دو زیر نوع مختلف از دو سوپر نوع متفاوت از طرح ER هستند که در آن R یک نوع رابطه 1:N دودویی موجود در بین دو زیر نوع است و همچنین فرض کنید که A در جانب 1 و B در جانب N از رابطه است، به طوری که یک نمونه از A با بسیاری از نمونه‌های B در ارتباط است.

1. تبدیل زیر نوع B که در جانب N با یک رابطه مجزای $B \rightarrow LA$ است.

2. فرض کنید که $B \rightarrow LA$ رابطه مربوط به زیر نوع A باشد، بعنوان کلید خارجی کلید اصلی $LA \rightarrow B$ را به حساب آورید و آن را بعنوان $A_PK(LA \rightarrow B)$ مجدداً نامگذاری کنید. این کلید خارجی می‌بایست شامل آخرین مشخصه موجود در $B \rightarrow LA$ باشد.

3. در $B \rightarrow LA$ تمامی ویژگی‌های ساده رابطه نوع R را به حساب آورید که به تبعیت از کلید خارجی است.

4. اگر زیر نوع A حاوی ویژگی‌های متعلق به خودش باشد، پس A و زیر نوع آن می‌بایست با استفاده از هر یک از روش‌های بهینه زیر تبدیل شود.

الف) دو رابطه جداگانه برای هر زیر نوع A و زیر نوع آن (مرحله 2 گزینه 1 فوق)

ب) یک رابطه مستقل برای هر دو زیر نوع A و زیر نوع آن (مرحله 2 گزینه 2 فوق)

5. اگر زیر نوع A شامل ویژگی‌های خودش نباشد پس سوپر نوع می‌بایست تنها به یک رابطه مجزا تبدیل شود.

شکل 6 الف) و ب) نتایج تبدیل OC-ERD داده شده در شکل 4 با استفاده از الگوریتم جدید ارائه شده در بالا را

نشان می‌دهد. بر این اساس، RDS بهینه ب) نتیجه‌ای از مورد 1 گزینه 1.1 است؛ RDS بهینه الف) نتیجه‌ای از

مورد 1 گزینه 1.2 است. نتیجه مورد 2 مشابه با RDS ب) است. در نتیجه تنها دو RDS بهینه تولید می‌شود هر

چند که سه روش بهینه مجزا برای تبدیل OC-ERD وجود دارد.

Employee | EmpNo | Name

Manager_Employee | EmpNo | DepNo (U) | StartDate

Department | DepNo | DepName

(a)

Employee | EmpNo | Name

Department | DepNo | DepName | Manager_EmpNo (U) | Manager_StartDate

(b)

شکل 6. تبدیل OC-ERD در شکل 4 با استفاده از الگوریتم جدید، الف) OC-RDS-1 مطلوب ب) OC-

RDS-2 مطلوب

حتی در این تحول، برخی از ویژگی‌های نیاز به نامگذاری مجدد دارند، برای مثال، EmpNo به

Manager_EmpNo نامگذاری مجدد می‌شود. تفاوت بین این فرایند نامگذاری مجدد و فرایند نامگذاری مجدد به

دلیل OUC-ERD است که در این مورد این امر توسط خود فرایند تحول رسیدگی می‌شود بدون این که از بینش

طراح گذر کند. این رابطه نظام‌مند قبلاً بعنوان یک قانون کلی ایجاد شده است و در الگوریتم جدید در مرحله 5،

مورد 1، گزینه 1.1 زیر مرحله 2 و 3 داده شده است. مشخصه FK دوباره نامگذاری شده یک پسوند U را داده است

که حاکی از این است که کلید بیرونی منحصر بفرد در حال پیشگیری از هر نوع تکرار غیر معمول از مقادیر FK در چندتائی‌های مختلف است همانطور که این مورد در شکل 3 رخ داده است. بر این اساس، تمامی ابهامات تجربه شده در تبدیل OUC-ERD توسط یک قانون بخصوص حل میشود.

4. کیفیت طرح پایگاه داده طراحی شده با استفاده از الگوریتم جدید

ارزیابی کیفیت RDS‌های منتج از OC-ERD‌ها از طریق الگوریتم تبدیل جدید حائز اهمیت است. بخصوص این شایان توجه و اهمیت است که، کیفیت نسبی RDS‌ها از طریق مدل نامشخص هستی‌شناسانه تولید می‌شود و مدل‌های مشخص هستی‌شناسانه نیاز به ارزیابی دارند. پس تنها موفقیت الگوریتم جدید را می‌تواند ادعا نمود. چند مطالعه قبلی [1، 2، 5] ابعاد کیفیت و طرح‌های اندازه‌گیری کیفیت را برای طرح‌های پایگاه داده و داده‌ها ارائه کرده‌اند.

جدول 1 مجموعه‌ای از ابعاد کیفیت ترکیب شده از منابع موجود را نشان می‌دهد. بر این باوریم که ارزیابی کیفیت ابتدائی را می‌توان با استفاده از این ابعاد انجام داد.

No	Quality Dimension (QD)	Description
1	Accuracy	A RDS should accurately represent information modeled in the ERD.
2	Complete	The RDS should preserve all the information represented in its predecessor ERD without losing anything(adapted from[1, 2])
3	Straightforward	The entire RDS including any tiny part of it must be a one logically and directly derived from its predecessor ERD according to a specific procedure
4	Trustworthy	The RDS should reflect only the information represented in its predecessor ERD and nothing outside it (adapted from [2])
5	Clear	The meaning intended by each item in the schema must be clear
6	Minimality	The RDS must be free from all the redundancies. Each aspect of the ERD must appear only ones in the RDS (adapted from [1, 5, 6])
7	Normalization	The RDS must satisfy the well-known normal forms as much as possible (adapted from[2])
8	Expressiveness	The RDS should be easily understood through its constructs(adapted from [1])
9	Reversible	The respective ERD could be able to re-produced only from the information presented in the RDS following a specific logic.

همچون تست کیفیت ابتدائی RDS در شکل 3 را همراه با دو RDS بهینه در شکل 6 با استفاده از جدول 1 فوق مقایسه می‌کنیم. مشاهده می‌کنیم که تمامی روابط RDS های شکل 6 در سومین فرم معمول هستند. در عین حال، همه روابط در شکل 3 در فرم‌های اول و دوم معمول هستند. همچنین، می‌توان نتیجه‌گیری کرد که طرح‌های پایگاه داده منطقی از ERD های مشخص هستی‌شناسانه تولید می‌شوند که در تبعیت از الگوریتم جدید هستند که کیفیت بالاتر را با توجه به سطح عادی‌سازی در مقایسه با همایان نامشخص هستی‌شناسانه خودش نشان می‌دهد.

5. بحث و کار آتی

با استفاده از یک مثال ساده از زندگی واقعی، نشان داده‌ایم که نمودارهای ER نامشخص هستی‌شناسانه مدل‌های ارتباطی کیفیت بالا را در مقایسه با همتای خودش یعنی مدل نامشخص هستی‌شناسانه تولید می‌کند. این امر با اصلاح قوانین تبدیل موجود حاصل می‌شود. با اینحال، به منظور دستیابی به نتایج مطلوب، این الگوریتم تبدیل باید از بهبود بیشتری برخوردار شود. همچون مطالعه اولیه تنها یک بخش از معیار کیفیت برای ارزیابی کیفیت مدل‌های منطقی تولید شده استفاده می‌شود.

اثر آتی از این تحقیق در حال انجام شامل موارد زیر است:

- توسعه قوانین جدید به منظور پوشش تمامی ساختارهای هستی‌شناسی
- اصلاح قوانین موجود به منظور غلبه بر معایب موجود با توجه به نمودارهای هستی‌شناختی نامشخص
- توسعه ارزیابی کیفیت به منظور غلبه بر سایر معیارهای کیفیت
- بهبود معیار کیفیت در این اثناء

انتظار بر این است که این مطالعه بر محبوبیت و کاربرد مدل‌های ER مفهومی مشخص هستی‌شناختی در صنعت تاثیرگذار باشد.

REFERENCES

- [1] C. Batini, S. Ceri, and S. B. Navathe, Conceptual database design: an Entity-relationship approach: Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA ©1992, 1992.
- [2] R. Y. Wang, M. P. Reddy, and H. B. Kon, "Toward quality data: An attribute-based approach," Decision Support Systems, vol. 13, pp. 349-372, 1995.
- [3] R. Weber, "Research review paper conceptual modeling and ontology: possibilities and pitfalls," Journal of Database Management, vol. 14, pp. 1-20, 2003.
- [4] P. P.-S. Chen, "The entity-relationship model: toward a unified view of data," ACM Trans. Database Syst., vol. 1, pp. 9-36, 1976.
- [5] C. Fahrner and G. Vossen, "A survey of database design transformations based on the Entity-Relationship model," Data & Knowledge Engineering, vol. 15, pp. 213-250, 1995.
- [6] Y. Wand and R. Weber, "On the ontological expressiveness of information systems analysis and design grammars," Information Systems Journal, vol. 3, pp. 217-237, 1993.
- [7] M. Bunge, Treatise on basic philosophy: Vol3: ontology I: The furniture of the world.: D. Reidel Publishing Co., Inc., New York, NY., 1977.
- [8] Y. Wand, V. C. Storey, and R. Weber, "An ontological analysis of the relationship construct in conceptual modeling," ACM Transactions on Database Systems, vol. 24, pp. 494-528, December 1999.
- [9] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems," in Fundamentals of Database Systems, 5th Edition ed New York: Addison Wesley, 2007, pp. 223-239
- [10] E. F. Codd, "A relational model of data for large shared data banks," Commun. ACM, vol. 13, pp. 377-387, 1970.