



گسترش یک زبان پرسش زمانی برای کار با داده‌های زمانی نادقیق

سید محمد تقی روحانی رانکوهی مرجان نظریان نائینی

دانشکده مهندسی برق و کامپیوتر، دانشگاه شهید بهشتی، تهران، ایران

چکیده

عدم دقت، ویژگی ذاتی اطلاعات در بسیاری از کاربردهای پایگاه‌داده‌ها محسوب می‌شود. به همین دلیل در پایگاه‌داده‌های متعارف شیوه‌های متعددی برای کار با داده‌های نادقیق ابداع شده است. با این حال پایگاه‌داده‌های زمانی، همچنان در کار با داده‌هایی که دارای ابعاد زمانی نادقیق هستند دچار ضعف می‌باشند. در این مقاله پس از بررسی مدل‌ها و زبان‌های پرسش زمانی، قواعد نحوی زبان پرسش زمانی ATSQL را به گونه‌ای گسترش خواهیم داد که از شیوه‌های فازی برای کار با داده‌هایی از این دست استفاده کند.

کلمات کلیدی: پایگاه‌داده‌های زمانی، زبان پرسش زمانی، عدم دقت زمانی، مدل‌سازی فازی، زبان پرسش زمانی فازی

۱- مقدمه

زبان‌های پرسش TSQL2 و TQUEL، برای بکارگیری داده‌های زمانی نامعین گسترش داده شده‌اند.

یکی از متداول‌ترین شیوه‌های کار با عدم قطعیت در اطلاعات استفاده از شیوه‌های مبتنی بر مدل‌سازی فازی است. این شیوه علاوه بر ارائه راه‌حلی برای مشکل داده‌های نادقیق، امکان مدل‌کردن مفاهیمی را می‌دهد که عدم دقت موجود در آنها ریشه در خطای سنجش و اندیشه انسانی دارد. با توجه به ارتباط تنگاتنگ پایگاه‌داده‌ها با کاربردهایی متکی بر دانش نادقیق بشری استفاده از شیوه‌های مدل‌سازی فازی می‌تواند مزیت مهمی محسوب شود.

در زمینه پایگاه‌داده‌های رابطه‌ای فازی و زبان‌های پرسش فازی پژوهش‌هایی در [۸] و [۹] انجام شده است. زبان‌های پرسش فازی مانند FQL و FSQL (که گسترش یافته زبان SQL است) با استفاده از تئوری فازی طراحی شده‌اند.

پایگاه‌داده‌های زمانی موجود در کار با داده‌های زمانی فازی کاستی‌هایی دارند. اگرچه مفهوم فازی در زبان‌های پرسش غیر زمانی بکار گرفته شده است، اما در هیچیک از زبان‌های پرسش زمانی مورد استفاده قرار نگرفته است. هدف اصلی ما در این مقاله اضافه کردن مفهوم فازی به پایگاه‌داده‌های زمانی و زبان پرسش زمانی است. با این گسترش می‌توان داده‌هایی را در پایگاه ذخیره کرد که دارای زمان اعتبار فازی هستند.

زمان جنبه مهمی در تمام پدیده‌های جهان واقعی است. رویدادها در لحظات مشخصی از زمان اتفاق می‌افتند و واقعیات وابسته به زمان هستند [۱]. بنابراین باید امکان ذخیره‌سازی و پردازش رویدادها و واقعیات‌ها بطور زمانمند وجود داشته باشد [۲]. پایگاه‌داده‌های زمانی، پایگاهی است که در آن اطلاعات متغیر در زمان نگهداری و پردازش می‌شود [۳]. به بیان دیگر پایگاه‌داده‌های زمانی به جای داده‌های جاری، شامل داده‌های تاریخی است [۴].

ابهام در زمان یک رویداد به خاطر استفاده از زبان طبیعی در بیان اطلاعات و یا کمبود دانش زمانی در مورد رویدادهای گذشته یا مورد انتظار در آینده همواره وجود دارد. بنابراین پایگاه‌داده‌های زمانی باید بتوانند داده‌های زمانی ناقص را ذخیره و پردازش کنند. اما یکی از کاستی‌های بارز مدل‌ها و زبان‌های پرسش زمانی موجود، ناتوانی آنها در کار با مقادیر نادقیق زمانی است.

مفهوم زمان اعتبار نامعین در [۵]، [۶] و [۷] مورد بررسی قرار گرفته است. در پایگاه‌داده‌های زمانی نامعین، هر واقعیت به مجموعه‌ای از لحظات با یک توزیع احتمال، منتسب است که تنها می‌تواند در یکی از این لحظات اعتبار داشته باشد. در [۵] زبان‌های SQL و SQL3 و در [۶] و [۷]

- لحظه‌ای یا بازه‌ای بودن مدل: در مدل لحظه‌ای، زمان به صورت لحظات هستند و اگر به صورت بازه نمایش داده شوند، تنها بازه‌ها ابزار نمایش هستند و بازه‌ها به صورت فشرده ذخیره می‌شوند.

۴- زبان‌های پرسش زمانی

تنظیم پرسش‌های زمانی با زبان پرسش غیر زمانی پیچیده و مشکل است. با وجود زبان پرسش زمانی، تنظیم پرسش‌ها با دخالت دادن زمان، ساده می‌شود. زبان‌های پرسش از مدل داده‌ای‌شان تاثیر می‌پذیرند؛ در طراحی زبان زمانمند باید طبیعت متغیر در زمان بودن داده‌ها روی تمام جنبه‌های زبان در نظر گرفته شود [۱۰].

در مبحث زبان‌های زمانمند، اصطلاحاتی وجود دارد که معرفی می‌کنیم.

- (۱) ارجاع زمانی^۴: این اصطلاح دلالت به تابعی در زبان دارد که برای ارجاع به مقادیر زمانمهر تاپل‌ها استفاده می‌شود.
- (۲) واکنشی لحظه^۵: به کمک این تابع می‌توان لحظات آغازین و پایانی بازه را بدست آورد.
- (۳) ساختگر لحظه^۶: ساختگر لحظه چندین لحظه را به عنوان ورودی می‌گیرد و یک لحظه بازمی‌گرداند.
- (۴) ساختگر بازه^۵: این ساختگر یا بازه‌ای از روی دو لحظه تولید می‌کند و یا بازه‌ای از اجتماع بازه‌های دیگر می‌سازد.
- (۵) مسند مقایسه کننده بازه یا لحظه^۶: باید مسندهایی برای مقایسه مقادیر زمان، لحظه‌ها یا بازه‌ها، در زبان وجود داشته باشد. مقایسه بازه‌ای می‌تواند بر اساس مقایسه لحظه‌ای بدست آید و وجود آن تنها خوانایی زبان را بالا می‌برد.

(۶) مسند زمانی^۷: مسند زمانی یک عبارت بولی است که مبتنی بر مسندهای مقایسه‌ای بوجود آمده است. از این مسند برای انتخاب تاپل بر اساس زمانمهرها استفاده می‌شود.

(۷) عملگر انتخاب زمانی: بازایی داده بر اساس شرایط زمانی، انتخاب زمانی است. عملگر انتخاب زمان اعتبار، اطلاعاتی را که در زمان مشخص شده معتبر هستند، بازایی می‌کند.

ما با توجه به بررسی‌هایی که در [۱۱] روی مدل‌های داده‌ای زمانی و زبانهای پرسش زمانی انجام داده‌ایم، زبان ATSQL را که یک زبان جدید و مبتنی بر زبان پرسش SQL است و نیز دارای مجموعه کاملی از امکانات کار با داده‌های زمانی می‌باشد، به عنوان یک زبان زمانی انتخاب کرده‌ایم و آن را به منظور پشتیبانی از داده‌های نادقیق زمانی تعمیم داده‌ایم. زبان تعمیم یافته که آن را FATSQL نام نهاده‌ایم، امکانات مناسبی را جهت کار با داده‌های نادقیق مهیا ساخته‌است بی آنکه بخشی از قابلیت‌های ATSQL را از دست بدهد.

۴-۱ ATSQL

در ادامه مطلب مختصری در مورد زبان ATSQL بیان می‌کنیم [۱۲] و [۱۳]:

زبان پرسش زمانی ATSQL گسترش یافته زبان SQL2 است. گسترش زمانی این زبان در چهار سطح طراحی شامل سازگاری رو به بالا، سازگاری رو به بالای زمانی^۸، کاهش‌پذیری لحظه‌ای مشابه نحوی^۹ و عدم محدودیت^{۱۰} انجام می‌شود [۱۲]. در زبان ATSQL، با افزودن دگرسازهای^{۱۱} جمله به SQL2، می‌توان پرسش‌ها را زمانمند کرد.

• سازگاری رو به بالا: زبان ATSQL شامل تمام عبارات پرسش SQL2 است و نتایج پرسش‌های قابل بیان در این زبان با نتایج همان پرسش‌ها در SQL2 یکسان است.

• سازگاری رو به بالای زمانی: در این زبان می‌توان رابطه‌های زمانی را تعریف کرد.

در این مقاله در بخش ۲ به بررسی مفاهیم پایه‌ای پایگاه‌داده‌های زمانی می‌پردازیم. در بخش ۳ روش‌های مدل‌سازی زمانی را مرور می‌کنیم و در بخش ۴ به معرفی یکی از جدیدترین زبان‌های پرسش زمانی به نام ATSQL می‌پردازیم. پس از آن در بخش ۵ مفاهیم فازی، چگونگی نمایش زمان فازی (و عملیات روی لحظه‌ها و بازه‌های زمانی فازی) را شرح می‌دهیم. سپس زبان ATSQL را به کمک شیوه‌های مدل‌سازی فازی تعمیم می‌دهیم و زبان پرسش توسعه یافته جدیدی را معرفی می‌کنیم که امکانات مناسبی برای کار با داده‌های زمانی نادقیق دارد. در این گسترش، قواعد نحوی زبان گسترش یافته و معنای آن را بیان می‌کنیم.

۲- مفاهیم پایه

انواع داده‌ای زمانی عبارتند از: لحظه، پرپود یا بازه زمانی و عنصر زمانی.

- **لحظه**: لحظه کوچکترین واحد غیر قابل تجزیه از زمان است که در سیستم مدیریت پایگاه‌داده وجود دارد.
- **پرپود (بازه) زمانی**: پرپود زمانی یا بازه زمانی مجموعه‌ای است از لحظات متوالی، با زمان آغاز و پایان مشخص، که به صورت [s, e] نمایش داده می‌شود (s لحظه آغازین و e لحظه پایانی است).
- **عنصر زمانی**: عنصر زمانی به صورت مجموعه محدودی از بازه‌ها (یا مجموعه دلخواه از لحظات) است.
- **ابعاد زمانی**: در پایگاه‌داده‌های زمانی دو بعد زمانی برای هر واقعیت مطرح است: بعد زمان اعتبار و بعد زمان تراکنش. زمان اعتبار یک واقعیت، مدت زمانی است که طی آن واقعیت در جهان واقعی صحت دارد. زمان تراکنش یک واقعیت، زمانی است که واقعیت در پایگاه‌داده‌ها ثبت می‌شود.

۳- مدل‌های داده‌ای زمانی

مدل داده‌ای زمانی شامل ساختارهایی برای نمایش داده‌های زمانی در سطح طراحی منطقی و عملیاتی روی ساختارهای زمانی و نیز امکانات کنترل جامعیت داده‌ها است.

در طراحی مدل داده‌ای زمانی، چگونگی نمایش زمان اعتبار و تراکنش، چگونگی انتساب زمان به واقعیت‌ها و فشرده‌بودن^۱ زمان بررسی می‌شود [۲]:

- زمان اعتبار و تراکنش می‌توانند با لحظه، با بازه و با عنصر زمانی نمایش داده شود.
- زمان می‌تواند به یک صفت، گروهی از صفات، یک تاپل یا شیء منتسب شود.
- اگر در مدل داده‌ای، زمان به صورت فشرده ذخیره شود، اجتماع بازه‌ها جایگزین مجموعه بازه‌های همپوشان یا مجاوری می‌شود که منتسب به تاپل‌هایی هستند که مقادیر صفاتشان یکسان است. مدل‌های زمانی می‌توانند بر اساس سه ضابطه تقسیم شوند.
- نوع زمان مورد استفاده در مدل‌ها: بر این اساس، سه مدل زمان اعتبار، زمان تراکنش و دو زمانی وجود دارد.
- روش انتساب زمان در مدل‌ها: بر این اساس، دو نوع مدل با زمانمهر تاپلی و مدل با زمانمهر صفتی وجود دارد.
- **§** مدل با زمانمهر تاپلی: در این مدل زمان اعتبار یا زمان تراکنش و یا هر دو، و به صورت لحظه زمانی یا بازه زمانی به تاپل‌ها منسوب می‌شوند.
- **§** مدل با زمانمهر صفتی: در این مدل زمان اعتبار یا زمان تراکنش به صفات منسوب می‌شوند.

مسندهای زمانی عبارت‌های بولینی هستند که بر اساس مسندهای مقایسه‌ای بوجود آمده‌اند. فرض می‌کنیم c_1 و c_2 دو مسند مقایسه‌ای باشند، c_1 AND c_2 ، c_1 OR c_2 و c_1 NOT c_2 مسندهای زمانی در ATSQL هستند، که نتیجه آنها از قوانینی که روی عبارت‌های بولی وجود دارد، تبعیت می‌کند.

فشرده‌سازی در زبان ATSQL:

زبان ATSQL یک زبان بازه‌ای است که فشرده‌سازی در آن به صورت دستی انجام می‌گیرد و در تنظیم پرسش‌ها از کلارز COAL برای مشخص کردن انجام عمل فشرده‌سازی روی رابطه‌ها استفاده می‌شود. برای فشرده کردن بازه‌های زمانی دقیق می‌توان از عملگر اجتماع بازه‌ها استفاده کرد.

دگرسازهای زبان در ATSQL:

زبان ATSQL، دگرسازهای SEQ VT و NONSEQ VT را برای اجرای پرسش‌های متوالی و غیر متوالی و دگرساز SET VT را برای تعیین برد زمان به زبان SQL2 افزوده است. علاوه بر این دگرسازها، ATSQL دگرساز دامنه زمان را نیز معرفی کرده است. این دگرساز ثابت پریودی است که بعد از عبارت NONSEQ VT ذکر می‌شود و رابطه‌ها را به زمان مشخص شده محدود می‌کند.

عملیات ریاضی در زبان ATSQL:

در این زبان عملیات جمع، تفریق، \min و \max روی لحظه‌ها و بازه‌های دقیق تعریف شده است. این عملیات، عملیات ساده ریاضی هستند. به‌علاوه در زبان ATSQL، تابع $DURATION(I, granule)$ ، فاصله زمانی میان لحظه آغازی و پایانی بازه را برحسب واحد زمانی تعیین شده محاسبه می‌کند.

بازیابی، درج، حذف و بهنگام‌سازی در زبان ATSQL:

دستورهای بازیابی، درج، حذف و بهنگام‌سازی در زبان ATSQL، بر اساس دستورهای SQL هستند. رفتار زمانی این دستورها، با دگرسازها و عملیات جدید زبان از قبیل واكشی لحظه، مقایسه لحظه‌ای به دست می‌آید.

۵-۱ FATSQL: زبان پرسش زمانی فازی

در این بخش، به توصیف مشخصات نحوی ATSQL و نحوه تعمیم آن به FATSQL خواهیم پرداخت، اما نخست مفاهیم مورد نیاز برای نمایش زمان فازی را تعریف می‌کنیم.

۵-۱ مفاهیم فازی

مفاهیم جدید متنوعی در تئوری فازی مطرح می‌شوند که اساسی‌ترین آنها عبارتند از: مجموعه فازی^{۱۳}، تابع عضویت^{۱۴} یا درجه عضویت^{۱۵} و اعداد فازی^{۱۶}. اینجا این مفاهیم را یادآور می‌شویم:

فرض می‌کنیم X مجموعه عناصر x باشد، مجموعه $A \subseteq X$ ، یک مجموعه کلاسیک یا غیر فازی است اگر، هر عنصر x یا به مجموعه A تعلق داشته باشد یا نداشته باشد. مجموعه $\tilde{A} \subseteq X$ یک مجموعه فازی است اگر به صورت $\tilde{A} = \{x, \mu_{\tilde{A}}(x) \mid x \in X\}$ تعریف شده باشد که در آن $\mu_{\tilde{A}}(x)$ تابع عضویت یا درجه عضویت (درجه صحت) x نام دارد و میزان عضویت x را در A نشان می‌دهد.

مثال:

فرض می‌کنیم

$X = \{1, 2, 3, 4, 5, 6\}$ و $\tilde{A} = \{(1, 0.2), (2, 0.5), (5, 1)\}$ باشد،

بنابراین عدد 1 با درجه امکان 0.2 به مجموعه A تعلق دارد.

اعداد فازی، ماهیت فازی داده‌هایی عددی (از قبیل ABOUT 20) را نشان می‌دهند. این داده‌ها توسط مجموعه‌های فازی که شکل دوزنقه‌ای دارند، نمایش داده می‌شوند (فرض دوزنقه‌ای بودن به منظور حفظ سادگی است). تابع عضویت یک عدد فازی به صورت زیر تعریف می‌شود:

جملات SQL2، روی رابطه‌های زمانی همانند رابطه‌های لحظه‌ای عمل می‌کنند.

• کاهش‌پذیری لحظه‌ای مشابه نحوی: زبان ATSQL با افزودن دگرساز جمله SEQ VT به پرسش‌های SQL2، نسخه زمانی مشابه نحوی آنها را می‌سازد. نتیجه این پرسش‌ها روی رابطه زمانی با نتیجه پرسش غیر زمانی نظیرشان روی مجموعه‌ای از تصاویر لحظه‌ای رابطه زمانی یکسان است.

• عدم محدودیت: در این زبان، با اضافه شدن دگرساز NONSEQ VT به پرسش‌های SQL2، پرسش‌های غیر متوالی را می‌توان ساخت. در این پرسش‌ها دستیابی به زمان به صورت دستیابی صریح به صفت زمانمهر در رابطه‌ها انجام می‌شود. دگرساز SET VT، برد زمان (زمان اعتبار) رابطه نتیجه را تعیین می‌کند. ثابت زمانی که بعد از VT ذکر می‌شود، دامنه زمانی رابطه‌ها را مشخص می‌کند.

زبان ATSQL امکان تعریف رابطه‌های زمانی و عملیات درج، حذف، تغییر و پرسش روی رابطه‌های زمانی را دارد. در این زبان، زمان به صورت لحظه زمانی و پریود زمانی دیده می‌شود. این زبان توابعی برای واكشی لحظه، ساختگر لحظه، ساختگر بازه، مقایسه زمانی دارد و در تنظیم جملات پرسش و بهنگام‌سازی از آنها استفاده می‌شود:

• انواع داده‌ای زمانی در ATSQL:

انواع داده‌ای زمانی این زبان عبارتند از:

- انواع لحظه‌ای: TIME، DATE و TIMESTAMP
- نوع بازه‌ای: PERIOD

تعریف شما در ATSQL:

تعریف رابطه‌های زمانی در این زبان به کمک دستور CREATE انجام می‌گیرد که ساختار نحوی آن به صورت زیر است:

CREATE TABLE ... AS VT

در این دستور وجود کلارز AS VT به این معناست که رابطه‌ای که ایجاد می‌شود باید از نوع زمانی باشد.

همچنین امکان تغییر یک رابطه غیر زمانی به رابطه زمانی توسط دستور ALTER و به کمک کلارز ADD VT ارائه می‌شود:

ALTER TABLE ... ADD VT

واكشی لحظه در ATSQL:

در ATSQL از دو کلارز BEGIN(period) و END(period) برای واكشی لحظه‌های آغازین و پایانی یک بازه دقیق، استفاده می‌شود.

مقایسه لحظه‌ای در زبان ATSQL:

در این زبان عملگرهای <، =، > و <=، > و <> برای مقایسه دو لحظه زمانی دقیق وجود دارد.

مقایسه بازه‌ای در زبان ATSQL:

در زبان ATSQL بین دو بازه زمانی دقیق عملگرهای مقایسه‌ای =، CONTAINS، MEETS، PRECEDES و OVERLAPS وجود دارد.

ساختگر بازه در زبان ATSQL:

در زبان ATSQL دو ساختگر بازه وجود دارد: کلارز PERIOD(p_1, p_2) که از دو لحظه p_1 و p_2 ، بازه $[p_1, p_2]$ را می‌سازد و کلارز INTERSECT(I_1, I_2) که اشتراک دو بازه I_1 و I_2 را بدست می‌آورد.

ساختگر لحظه در زبان ATSQL:

در زبان ATSQL دو ساختگر لحظه وجود دارد: FIRST(p_1, p_2) و LAST(p_1, p_2) که به ترتیب لحظه کوچکتر و لحظه بزرگتر را از بین دو لحظه مشخص شده تعیین می‌کنند.

مسندهای زمانی در زبان ATSQL:

FUZZY PERIOD و نوع بازه‌های فازی FUZZY TIMESTAMP را تعریف می‌کنیم.

قواعد نحوی مربوط به تعریف این انواع زمانی عبارتند از:

```
<data type> ::= <period type> | <datetime type>
<period type> ::=
    • [ <fuzzy type> ] PERIOD [ <left paren> <data type>
      <right paren> ]
<datetime type> ::=
    • [ <fuzzy type> ] DATE
    • [ [ <fuzzy type> ] TIME [ <left paren> <time
      precision>
      <right paren> ] [ WITH TIME ZONE ]
    • [ [ <fuzzy type> ] TIMESTAMP [ <left paren>
      <timestamp precision> <right paren> ] [ WITH TIME
      ZONE ]
<fuzzy type> ::=
    • FUZZY
```

۲-۳-۵ تعریف شمای رابطه فازی

دستورات مربوط به تعریف و تغییر رابطه‌ها در ATSQL عبارتند از CREATE و ALTER. از سوی دیگر رابطه زمانی فازی، رابطه‌ای است که در صفت زمان اعتبار آن می‌توان پیروی (بازه) فازی درج کرد. برای تعریف رابطه‌های زمانی فازی در FATSQ دستور CREATE را به شکل زیر تعمیم می‌دهیم:

```
<table definition> ::=
    CREATE [ { GLOBAL | LOCAL } TEMPORARY ]
    TABLE <table name> <table element list>
    <temporal definition>
    [ ON COMMIT { DELETE | PRESERVE } ROWS ]
```

دستور تغییر رابطه (ALTER) را نیز برای فازی کردن یک رابطه غیر زمانی موجود، به شکلی گسترش می‌دهیم که بتوان از آن برای افزودن صفت زمان فازی به یک رابطه غیر زمانی استفاده کرد:

```
<alter table statement> ::=
    ALTER TABLE <table name> <alter table action>
<alter table action> ::= <add valid definition>
<add valid definition> ::=
    • ADD [ <fuzzy type> ] VT [ <period type> ]
```

۳-۳-۵ تعریف زمان فازی

در ATSQL قواعد نحوی زیر مقادیر نشان دهنده لحظه‌ها و بازه‌ها را تولید می‌کند.

```
<date string> ::= <quote> <date value> <quote>
<date value> ::= <years value> <minus sign> <months
value>
    <minus sign> <days value>
<years value> ::= <unsigned integer>
<months value> ::= <unsigned integer>
<days value> ::= <unsigned integer>
<time string> ::= <quote> <time value> <quote>
<time value> ::= <hours value> <colon> <minutes value>
    <colon>
    <seconds value>
<hours value> ::= <unsigned integer>
<minutes value> ::= <unsigned integer>
<seconds value> ::= <unsigned integer>
<timestamp string> ::= <quote> <date value> <space>
    <time value> <quote>
<period string> ::=
    <quote> <beginning value> <minus sign> <ending
value>
    <quote>
```

$$T(x: a, b, c, d) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \leq x \leq b \\ 1 & b < x < c \\ (d-x)/(d-c) & c \leq x \leq d \\ 0 & x > d \end{cases}$$

در مدل‌سازی زمان فازی از اعداد فازی استفاده می‌شود.

۲-۵ مفهوم زمان فازی

لحظه فازی^{۱۶}: فرض می‌کنیم N مجموعه اعداد طبیعی منطبق بر لحظات محور زمان و $[0, 1]^N$ مجموعه زیرمجموعه‌های فازی تعریف شده روی N و POINT مجموعه جهانی لحظه‌های زمانی باشد. تابع $\tau: POINT \rightarrow [0, 1]^N$ ، تفسیر معنایی لحظه زمانی P، به صورت زیر تعریف می‌شود:

$$\tau(P) = \{ \mu_\tau(P)(n) / n \mid n \in N, \mu_\tau(P) = \mu(n) \} \quad (۱)$$

لحظه فازی مجموعه‌ای از لحظات زمانی است که یک رویداد ممکن است در آنها روی دهد.

بازه فازی^{۱۷}: بازه زمانی فازی، یک بازه زمانی با مرزهای زمانی فازی است. بازه فازی به صورت $[s, e]$ نمایش داده می‌شود که در آن مرزهای زمانی s و e می‌توانند مبهم و فازی باشد. مرزهای زمانی با اعداد طبیعی فازی که با توابع عضویت دوزنقه‌ای مشخص می‌شوند، مدل می‌شوند [۱۴]:

$$\mu_e(n) = T(n: a_e, b_e, c_e, d_e) \quad \mu_s(n) = T(n: a_s, b_s, c_s, d_s) \quad (۲)$$

فرض می‌کنیم N مجموعه اعداد طبیعی منطبق بر لحظات محور زمان و $[0, 1]^{N*N}$ مجموعه زیرمجموعه‌های فازی تعریف شده روی $N*N$ و INTERVAL مجموعه جهانی بازه‌های زمانی باشد. تابع $\eta: INTERVAL \rightarrow [0, 1]^{N*N}$ ، تفسیر معنایی بازه زمانی I، به صورت زیر تعریف می‌شود:

$$\eta(I) = \{ \mu_\eta(I) ([n_s, n_e]) / [n_s, n_e] \mid [n_s, n_e] \in N * N, \mu_\eta(I) = \min(\mu_s(n_s), \mu_e(n_e)) \} \quad (۳)$$

بازه فازی مجموعه‌ای از بازه‌های زمانی ممکن (همراه با درجه امکان) است.

۳-۵ گسترش زبان ATSQL

در نسخه فازی ATSQL که آن را FATSQ (Fuzzy ATSQL) می‌نامیم، زبان پرسش زمانی ATSQL، با استفاده از تئوری فازی به شکلی گسترش داده می‌شود که بتواند از انواع داده‌های زمانی فازی پشتیبانی کند. در ادامه معنای قواعد نحوی جدید یا تغییر یافته را برای هر یک از دستورات اصلی ATSQL بیان می‌کنیم. در سراسر این فصل برای مشخص کردن قواعد نحوی جدید یا تغییر یافته (نسبت به زبان ATSQL) از علامت “•” استفاده خواهیم کرد.

۱-۳-۵ تعریف نوع داده‌ای فازی

انواع داده‌ای زمانی ATSQL عبارتند از انواع لحظه‌های DATE، TIME و TIMESTAMP و نوع بازه‌های PERIOD. برای تعریف انواع لحظه‌ای فازی و بازه‌ای فازی باید نسخه‌های فازی انواع لحظه و پیروی تعریف شود. به همین منظور در FATSQ انواع لحظه‌ای فازی FUZZY DATE، FUZZY TIME و

- | <fuzzy time name>

<date literal> ::= DATE <date string>

<date string> ::=

- | <quote> <fuzzy time name> <quote>

<time literal> ::= TIME <time string>

<time string> ::=

- | <quote> <fuzzy time name> <quote>

<timestamp literal> ::= TIMESTAMP <timestamp string>

<timestamp string> ::=

- | <quote> <fuzzy time name> <quote>

۵-۳-۴ واکنشی لحظه

در ATSQL از دو کلاز BEGIN(period) و END(period) برای واکنشی لحظه‌های آغازین و پایانی یک بازه دقیق، استفاده می‌شود ولی با فازی شدن زمان، معنای BEGIN و END به شکلی تغییر می‌کند که بتوانند روی بازه‌های فازی عمل کنند. نسخه فازی این دو تابع را به صورت زیر تعریف می‌کنیم:

$$begin(I) = \{m_{begin(I)}(n_s) / n_s \} \quad (۴)$$

$$m_{begin(I)}(n_s) = \sup_{n_e} m_{h(I)}([n_s, n_e])$$

$$end(I) = \{m_{end(I)}(n_e) / n_e \} \quad (۵)$$

$$m_{end(I)}(n_e) = \sup_{n_s} m_{h(I)}([n_s, n_e])$$

در این تعریف، I بازه فازی و n_s و n_e لحظات ابتدایی و انتهایی بازه‌های دقیق متعلق به بازه I هستند.

۵-۳-۵ مقایسه زمانی

در ATSQL عملگرهای <, =, >, <=, >= و <> برای مقایسه لحظات زمانی دقیق وجود دارد. با فازی شدن لحظات زمانی، معنای عملگرهای مقایسه‌ای تغییر می‌کند. معنای این عملگرها روی لحظات فازی، با توجه به تعریف معیار امکان‌پذیری^{۱۸} و معیار الزام‌پذیری^{۱۹} در [۱۵] بدست می‌آید. معیار امکان‌پذیری مشخص می‌کند که شرط مقایسه‌ای مشخص شده بین دو لحظه به چه میزان امکان‌پذیر است و معیار الزام‌پذیری مشخص می‌کند که تا چه اندازه شرط مقایسه‌ای مشخص شده بین دو لحظه لزوماً برقرار است [۱۶]. باید توجه داشت که در صورت وجود دو لحظه دقیق مقدار دو معیار مذکور صفر یا یک است، در صورتی که با فرض وجود دو لحظه فازی این دو معیار، مقداری بین صفر و یک دارند. فرض می‌کنیم p_1 و p_2 دو لحظه زمانی فازی و $n_1, n_2 \in \mathbb{N}$ به ترتیب لحظات دقیق متعلق به این دو لحظه باشند. معنای عملیات مقایسه لحظه‌ای را که روی لحظه‌های فازی عمل می‌کنند، به صورت زیر تعریف می‌کنیم:

$$Poss(p_1 < p_2) = \quad (۶)$$

$$\sup_{n_1 < n_2} \min(m_{h(p_1)}(n_1), m_{h(p_2)}(n_2))$$

$$Nec(p_1 < p_2) = \quad (۷)$$

$$\inf_{n_1 > n_2} \max(1 - m_{h(p_1)}(n_1), 1 - m_{h(p_2)}(n_2))$$

معنای سایر عملگرها (>, <=, >=, <>) نیز به همین صورت تعریف می‌شود.

<beginning value> ::=

<date value> <space> <time value> | <date value>

|

<time Value>

<ending value> ::=

<date value> <space> <time value> | <date value>

|

<time value>

برای اینکه بتوانیم در FATSOL به صورت پویا زمان فازی ایجاد کنیم، جمله CREATE را به شکلی گسترش می‌دهیم که به کمک آن بتوان یک مقدار زمانی فازی را به عنوان یک صفت از یک رابطه ایجاد کرد. قاعده نحوی تعمیم یافته این دستور به صورت زیر است:

<create fuzzy time> ::=

- CREATE FUZZYTIME <fuzzy time name>
<left paren> <fuzzy value> <comma> <fuzzy value>
<comma> <fuzzy value> <comma> <fuzzy value>
<right paren>

<fuzzy value> ::=

- <unsigned integer>
- | <quote> <date value> <space> <time value>
<quote>
- | <quote> <date value> <quote>
- | <quote> <time value> <quote>

<fuzzy time name> ::=

- <character representation>

بطوریکه دیده می‌شود، زمان فازی با یک نام مستعار و چهار عدد که معرف زمان هستند و یا چهار مقدار زمانی مشخص می‌شود. چهار عدد یا چهار زمان، یک تابع عضویت فازی را تعریف می‌کند که به شکل یک عدد فازی دوزنقه‌ای است. به منظور تغییر مقدار یک زمان فازی ایجاد شده دستور ALTER نیز به این شکل تعمیم می‌یابد:

<alter fuzzy time> ::=

- ALTER FUZZYTIME <fuzzy time name>
<left paren> <fuzzy value> <comma> <fuzzy value>
<comma> <fuzzy value> <comma> <fuzzy value>
<right paren>

در این دستور به منظور تخصیص مقدار زمانی فازی جدیدی به یک نام مستعار موجود ابتدا نام مذکور که قبلاً ایجاد شده است و سپس مقادیر تابع عضویت جدید ذکر می‌شود.

عمل حذف یک زمان فازی در FATSOL توسط ویرایش تعمیم یافته DROP با قاعده نحوی زیر انجام می‌شود:

<drop fuzzy time> ::= DROP FUZZYTIME <fuzzy date name>

به منظور تکمیل امکانات کار با مقادیر و متغیرهای فازی، باید متغیرهای زمانی زبان را به گونه‌ای گسترش دهیم که امکان قرار دادن زمان فازی در آنها وجود داشته‌باشد:

<period literal> ::= PERIOD <period string>

<period string> ::= <quote> <beginning value> <minus sign>

<ending value> <quote>

<beginning value> ::=

- | <fuzzy time name>

<ending value> ::=

فازی عمل می کنند، تعریف مسندهای زمانی با قبل تفاوت پیدا می کند. لازم است نسخه های فازی این مسندها بر اساس مقادیر میزان امکان پذیری و میزان الزام پذیری تعریف شوند. میزان امکان پذیری و میزان الزام پذیری مسندهای زمانی c_1 و c_2 را به صورت زیر تعریف می کنیم:

$$\text{Poss}(c_1 \text{ AND } c_2) = \min(\text{Poss}(c_1), \text{Poss}(c_2)) \quad (12)$$

$$\text{Nec}(c_1 \text{ AND } c_2) = \min(\text{Nec}(c_1), \text{Nec}(c_2)) \quad (13)$$

$$\text{Poss}(c_1 \text{ OR } c_2) = \max(\text{Poss}(c_1), \text{Poss}(c_2)) \quad (14)$$

$$\text{Nec}(c_1 \text{ OR } c_2) = \max(\text{Nec}(c_1), \text{Nec}(c_2)) \quad (15)$$

$$\text{Poss}(\text{NOT } c_1) = 1 - \text{Nec}(c_1) \quad (16)$$

$$\text{Poss}(\text{NOT } c_1) = 1 - \text{Nec}(c_1) \quad (17)$$

۹-۳-۵ فشرده سازی

زبان ATSQL یک زبان بازه ای است که انجام فشرده سازی روی رابطه ها با کلاز COAL در پرسش ها مشخص می شود. برای فشرده کردن بازه های زمانی دقیق می توان از عملگر اجتماع بازه ها استفاده کرد. زبان FATSQUAL نیز یک زبان بازه ای است که در آن نیز برای مشخص کردن اجرای فشرده سازی از کلاز COAL استفاده می شود. از آنجایی که برای فشرده کردن تاپل های فازی (تاپل هایی با زمان فازی) باید از عملگر اجتماع روی بازه های فازی استفاده کرد، لازم است معنای اجتماع فازی را مجدداً تعریف کنیم. معنای نسخه فازی عملگر اجتماع را به صورت زیر تعریف می کنیم:

$$m_{h(I_1 \cup I_2)}(i) = \sup_{i=i_1 \cup i_2} \min(m_{h(I_1)}(i_1), m_{h(I_2)}(i_2)) \quad (18)$$

۱۰-۳-۵ دگرسازهای زبان

معنای دگرسازهای SEQ VT, SET VT و NONSEQ VT در زبان FATSQUAL بدون تغییر باقی می ماند. اما لازم است معنای دگرساز دامنه زمان در هنگام فازی شدن زمان تغییر کند. معنای دگرساز دامنه زمان مشابه معنای انتخاب زمانی فازی است، زیرا در حقیقت با مشخص کردن ثابت پررود فازی، انتخاب تاپل ها بر اساس شرط زمانی مشخص شده انجام می گیرد. معنایی را که برای انتخاب فازی تعریف کرده ایم در ادامه شرح خواهیم داد.

۱۱-۳-۵ عملیات ریاضی

در ATSQL عملیات جمع، تفریق، \min و \max روی لحظه ها و بازه های دقیق تعریف شده است. همچنین تابع $\text{Duration}(I, \text{granule})$ فاصله زمانی میان لحظه آغازی و پایانی بازه را برحسب واحد زمانی تعیین شده بدست می آورد. با فازی شدن لحظه ها و بازه ها لازم است که معنای عملیات ریاضی مجدداً تعریف شود. به این منظور عمل جمع بین دو بازه فازی I_1 و I_2 را به صورت زیر تعریف می کنیم. \dot{I}_1 و \dot{I}_2 به ترتیب بازه های دقیق بازه های I_1 و I_2 هستند:

$$m_{h(I_1 + I_2)}(i) = \sup_{i=i_1 + i_2} \min(m_{h(I_1)}(i_1), m_{h(I_2)}(i_2)) \quad (19)$$

همچنین عمل جمع بین دو لحظه فازی P_1 و P_2 را به شکل زیر خواهد بود (P_1) لحظه های دقیق لحظه P_1 و P_2 لحظه های دقیق لحظه P_2 هستند):

در زبان ATSQL بین دو بازه زمانی دقیق عملگرهای مقایسه ای $=$ ، CONTAINS ، MEETS ، PRECEDES و OVERLAPS وجود دارد. در FATSQUAL فرض می کنیم به طور کلی θ عملگر مقایسه ای بین دو بازه فازی I_1 و I_2 و $\bar{\theta}$ عملگر معکوس θ باشد. همچنین فرض می کنیم \dot{I}_1 و \dot{I}_2 به ترتیب دو بازه از مجموعه بازه های دقیق متعلق بازه های فازی I_1 و I_2 باشند. درجه امکان پذیری و درجه الزام پذیری عملگر مقایسه بازه ای را به صورت زیر تعریف می کنیم:

$$\text{Poss}(I_1 \theta I_2) = \quad (8)$$

$$\sup_{i=i_1 \theta i_2} \min(m_{h(I_1)}(i_1), m_{h(I_2)}(i_2))$$

$$\text{Nec}(I_1 \theta I_2) =$$

$$\inf_{i=i_1 \theta i_2} \max(1 - m_{h(I_1)}(i_1), 1 - m_{h(I_2)}(i_2)) \quad (9)$$

$\text{Poss}(I_1 \theta I_2)$ مشخص می کند که به چه میزان شرط مقایسه ای مشخص شده بین دو بازه فازی I_1 و I_2 امکان پذیر است و $\text{Nec}(I_1 \theta I_2)$ مشخص می کند که به چه میزان شرط مقایسه ای مشخص شده بین دو بازه فازی لزوماً برقرار است.

۶-۳-۵ ساختگر بازه

در زبان ATSQL دو ساختگر بازه وجود دارد: کلاز $\text{PERIOD}(p_1, p_2)$ که از دو لحظه p_1 و p_2 بازه $[p_1, p_2]$ را می سازد و کلاز $\text{INTERSECT}(I_1, I_2)$ که اشتراک دو بازه I_1 و I_2 را بدست می آورد. به منظور تعمیم معنای $\text{PERIOD}(p_1, p_2)$ ، هنگامی که p_1 و p_2 دو لحظه فازی هستند، نسخه فازی این معنا را به صورت زیر تعریف می کنیم:

$$\eta \text{ period}(p_1, p_2) = \{ \mu \eta(\text{period}(p_1, p_2)) ([p_1, p_2]) / [p_1, p_2] \}$$

$$[p_1, p_2] \in \mathbb{N} * \mathbb{N}, \mu \eta(\text{period}(p_1, p_2)) =$$

$$\min(\mu_r(p_1), \mu_r(p_2)) \quad (10)$$

فرض می کنیم I_1 و I_2 دو بازه فازی و \dot{I}_1 و \dot{I}_2 به ترتیب دو بازه از مجموعه بازه های دقیق متعلق به بازه های فازی I_1 و I_2 باشند. معنای نسخه فازی $\text{INTERSECT}(I_1, I_2)$ را به صورت زیر تعریف می کنیم:

$$m_{h(I_1 \cap I_2)}(i) = \quad (11)$$

$$\sup_{i=i_1 \cap i_2} \min(m_{h(I_1)}(i_1), m_{h(I_2)}(i_2))$$

۷-۳-۵ ساختگر لحظه

در زبان ATSQL دو ساختگر لحظه وجود دارد: $\text{FIRST}(p_1, p_2)$ و $\text{LAST}(p_1, p_2)$ که به ترتیب لحظه کوچکتر و لحظه بزرگتر را از بین دو لحظه مشخص شده تعیین می کنند. در FATSQUAL توابع FIRST و LAST را به گونه ای تعریف می کنیم که برای بدست آوردن کوچکترین و بزرگترین لحظه از بین دو لحظه فازی به ترتیب از عملگرهای $<$ و $>$ که بین دو لحظه زمانی فازی تعریف شده اند، استفاده کنند. به منظور استفاده از این دو ساختگر لازم است که میزان امکان پذیری یا الزام پذیری مورد نظر مشخص شود.

۸-۳-۵ مسندهای زمانی

مسندهای زمانی عبارات بولینی هستند که بر اساس مسندهای مقایسه ای بوجود آمده اند. با توجه به این که در زبان FATSQUAL مسندهای مقایسه ای روی انواع

- **<with clause> ::= WITH** [<possibility cond>] [AND] [<necessity cond>]
- **<possibility cond> ::=** [<degree> <comp op>] POSSIBILITY <comp op> <degree>
- **<necessity cond> ::=** [<degree> <comp op>] NECESSITY <comp op> <degree>
- **<degree> ::=** <unsigned integer>

به این ترتیب در اجرای پرسش ابتدا مقادیر Poss و Nec مربوط به شرط WHERE بدست می‌آید و سپس نتیجه پرسش با اعمال شرطی که روی Poss و Nec در کلاز WITH مشخص شده روی مقادیر مزبور، تعیین می‌شود. مقادیر مربوط به میزان امکان پذیری و میزان الزام‌پذیری کلاز WITH به طور پیش‌فرض برابر یک در نظر گرفته می‌شود.

۵-۳-۱۳ مثال

در این بخش دو مثال از زبان FATSQZ ذکر می‌کنیم:

مثال:

با کمک دستورات زیر رابطه زمانی فازی کارمند ایجاد می‌شود و یک تاپل در آن درج می‌گردد.

```
CREATE TABLE Emp
Name: CHAR(20),
Mgr: CHAR(20),
Salary: INTEGER,
Dept: CHAR(10)
AS FUZZY VT;
CREATE about2003/1/30 (2003/1/27,2003/1/30, 2003/1/30,
2003/2/2)
CREATE about2003/5/1 (2003/4/28,2003/5/1, 2003/5/1, 2003/5/4)
SET VT PERIOD[about2003/1/30, about2003/5/1]
INSERT INTO Emp
VALUES ('a', 'b', 1000, 'd1');
```

مثال:

کارمندی را بیابید که با امکان بیشتر از ۰.۵ قبل از تاریخ ۲۰۰۳/۱/۱ به استخدام در آمده باشند.

```
NONSEQ VT
SELECT Name
FROM Emp(VT) AS E
WHERE BEGIN(VTIME(E)) <= DATE '2003/1/1'
WITH POSSIBILITY > 0.5 0 AND NECESSITY <= 1
```

۵-۳-۱۴ کاهش پذیری پرسش

برای تامین کاهش پذیری باید نتیجه اجرای جملات پرسش فازی (با معانی فازی) روی رابطه‌های غیر فازی با نتیجه اجرای جملات پرسش قدیمی (با معانی غیر فازی) یکسان باشد. بدین منظور باید توابعی که در معانی گسترش یافته زبان فازی مجدداً تعریف شده‌اند، هنگام عمل روی رابطه‌های غیر فازی به تعریفشان در معانی قبلی برگردند.

$$m_{t(P_1+P_2)}(p) = \sup_{p=P_1+P_2} \min(m_{t(P_1)}(p_1), m_{t(P_2)}(p_2)) \quad (20)$$

تعریف عمل تفریق فازی بین لحظه‌ها و بازه‌های فازی نیز به همین صورت بدست می‌آید.

امکان افزودن فاصله زمانی d به لحظه فازی P و بازه فازی I را به صورت زیر تعریف می‌کنیم:

$$Poss(dur(I, g)) = \sup_{dur(i, g)} (m_{h(I)}(i)) \quad (21)$$

$$Nec(dur(I, g)) = \inf_{dur(i, g)} (1 - m_{h(I)}(i)) \quad (22)$$

برای بدست آمدن min و max لحظه‌ها و بازه‌های فازی از عملگر < و > استفاده می‌شود. برای انجام توابع min و max باید میزان امکان‌پذیری یا میزان الزام‌پذیری مورد نظر مشخص باشد.

معنی تابع Duration(I, granule) نیز هنگامی که روی یک بازه فازی عمل می‌کند، تغییر می‌کند. میزان امکان‌پذیری و میزان الزام‌پذیری این تابع را در FATSQZ به صورت زیر تعریف می‌کنیم:

$$m_{t(P+d)}(p) = m_{t(P)}(p) \quad (23)$$

$$m_{h(I+d)}(i) = m_{h(I)}(i) \quad (24)$$

روشن است که خروجی این تابع، مجموعه‌ای از فاصله‌های زمانی با درجه امکان معین خواهد بود.

۵-۳-۱۲ بازبایی، درج، حذف و بهنگام‌سازی

دستورهای بازبایی، درج، حذف و بهنگام‌سازی در زبان ATSQL، بر اساس دستورهای SQL هستند. از آنجایی که در درج فازی کافی است مقدار زمان که در صفت زمان قرار می‌گیرد، مقداری فازی باشد، قاعده نحوی دستور درج در FATSQZ نسبت به ATSQL بدون تغییر باقی می‌ماند:

```
<insert statement> ::= INSERT INTO <table name>
<values list>
```

با توجه به این که در دستورهای حذف، بهنگام‌سازی و انتخاب، لازم است امکان مقایسه لحظه‌ای و بازه‌ای فازی در کلاز WEHRE وجود داشته باشد، قواعد نحوی و معنای این دستورها باید مجدداً تعریف شوند. همان طور که می‌دانیم نتیجه انجام مقایسه‌های فازی، مقادیر مربوط به میزان امکان پذیری و میزان الزام‌پذیری شرط آن مقایسه است، بنابراین لازم است که در دستورها، میزان امکان پذیری و میزان الزام‌پذیری مورد نظر (که مقداری بین صفر و یک است) مشخص شود. به این منظور کلاز WITH را برای تعیین مقادیر مزبور به دستورهای حذف، بهنگام‌سازی و بازبایی اضافه کرده‌ایم.

قواعد نحوی پرسش‌های گسترش یافته در زیر آورده شده است:

```
<update statement: searched> ::= UPDATE <table name>
SET <set clause list> [ WHERE <search condition> ]
```

- [<with clause>]

```
<delete statement: searched> ::=
DELETE FROM <table name> [ WHERE <search condition> ]
```

- [<with clause>]

```
<query specification> ::= SELECT [<set quantifier>]
<select list> <table expression>
```

- [<with clause>]

- [2] B. Skjellavg, "Temporal Data : Time and Relational Databases Research," Report 246, Department of Informatics, University of Oslo, 1997.
- [3] G. Ozsoyglu and R. T. Snodgrass, "Temporal and Realtime Database : A Survey," *IEEE Transaction On Knowledge and Data Engineering*, vol. 7, no. 4, 1997.
- [4] C. S. Jensen, R. T. Snodgrass, et.al. , "Temporal Data Management," *IEEE Transaction On Knowledge and Data Engineering*, vol. 11, no. 1 ,1997.
- [5] C. E. Dyreson and R.T. Snodgrass, "Supporting Valid-time Indeterminacy," *ACM Transactions on Database system*, vol. 23, no. 1, 1998.
- [6] C. E. Dyreson, *Valid-time Indeterminacy*, Ph.D. dissertation, University of Arizona. 1994.
- [7] C. E. Dyreson and R.T. Snodgrass, "Temporal Indeterminacy". Chapter 18 in *The TSQL2 Temporal Query Language*, 1995.
- [8] L. Doungbo and D. Liv, *A Fuzzy Prolog Database System*, Research Press Ltd, 1990.
- [9] H. Nakajima and, D. Sogoh, *Fuzzy Database Language and Library Fuzzy Extension to SQL*, OMRAN CORPORATION, 1993.
- [10] C. S. Jensen, R. T. Snodgrass, and M. D. Soo, "The Tsql2 Data Model," Chapter 10, in *The TSQL2 Temporal Query Language*, 1995.
- [۱۱] م. نظریان، بررسی پایگاه داده های زمانی و ارائه طرح یک زبان پرسش زمانی، پایان نامه کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه شهید بهشتی، ۸۲.
- [12] M. H. Bohlen and C. S. Jensen, "Temporal Statement Modifier", *ACM Transactions on Database system*, vol. 25, no. 4 , 2000.
- [13] J. Chomocki, M. H. Bohlen, and D. Toman, "Querying ATSQL with Temporal Logic," *ACM Transactions on Database system*, vol. 26, no. 2 , 2001.
- [14] D. Dubios, and H. Prade, "Processing Fuzzy Temporal Knowledge," *IEEE Transaction on Systems*, vol. 19, no. 4, 1989.
- [15] H. J. Zimmermann, *Fuzzy Set Theory and its Applications*, Prentice-Hall, 1996.
- [16] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility, " *Fuzzy Sets and Systems*, vol. 1, no. 1, 1978.

ما فرض می کنیم که در تمام لحظات و بازه های غیر فازی درجه عضویت برابر یک است (به طور مثال در بازه $[۱,۵]$ مقدار μ برابر یک است). با این فرض پرسش های فازی می توانند بر روی زمان های دقیق عمل کنند.

اگر I_1 و I_2 بازه های دقیق و P_1 و P_2 لحظات دقیق باشند و توابع باز تعریف شده بر روی آنها عمل کند، نتیجه $(\sup(\min(\mu(I_1), \mu(I_2)))$ و $\sup(\min(\mu(P_1), \mu(P_2)))$ یا میزان امکان پذیری، برابر یک است، زیرا داریم $\sup(\min(1, 1)) = 1$ و نتیجه عبارات $(\inf(\max(1-\mu(I_1), 1-\mu(I_2)))$ و $\inf(\max(1-\mu(P_1), 1-\mu(P_2)))$ یا میزان الزام پذیری برابر یک است، زیرا داریم $\inf(\max(1-0, 1-0)) = 1$. به این ترتیب می توان گفت که هر پرسش غیر فازی می تواند به صورت یک پرسش فازی با مقادیر امکان پذیری و الزام پذیری برابر یک نیز تعریف شود.

از سوی دیگر در زبان FATSQ در پرسش های فازی مقادیر مربوط به میزان امکان پذیری و الزام پذیری را به طور پیش فرض برابر یک فرض می کنیم، لذا چنانچه این پرسش ها بر روی زمان غیر فازی نیز اعمال شوند با توجه به توضیحات بالا نتیجه پرسش دقیقاً معادل نتیجه حاصل از اعمال پرسش غیر فازی خواهد بود. به این ترتیب می توان ادعا کرد که ATSQL حالت خاص و محدودی از FATSQ است.

از سوی دیگر هر کجا در زبان ATSQL صحبت از مقادیر و متغیرهای زمانی می شود نسخه گسترش یافته فازی دستورات در FATSQ امکان کار با شکل فازی مقادیر و متغیرهای زمانی مزبور را فراهم می آورد. به عبارت دیگر FATSQ نه تنها همه قابلیت های ATSQL را در زمینه تعریف و پردازش داده های زمانی را داراست، بلکه تمامی این قابلیت ها را در مورد داده های زمانی فازی نیز ارائه می کند بدون اینکه از هیچ یک از کارایی های ATSQL چشم پوشی کند.

۶- نتیجه گیری

شیوه های متعددی برای کار با عدم قطعیت در اطلاعات وجود دارد و یکی از متداول ترین آنها استفاده از شیوه های مبتنی بر مدل سازی فازی است. این شیوه علاوه بر ارائه راه حلی برای مشکل داده های نادقیق، امکان مدل کردن مفاهیمی را می دهد که عدم دقت موجود در آنها ریشه در خطای سنجش و اندیشه انسانی دارد. با توجه به ارتباط تنگاتنگ پایگاه داده ها با کاربردهایی متکی بر دانش نادقیق بشری استفاده از شیوه های مدل سازی فازی می تواند مزیت مهمی محسوب شود.

در این مقاله زبان پرسش ATSQL به نحوی تعمیم داده شد که توانایی کار با داده های زمانی نادقیق را پیدا کند و بدین منظور تعدادی از مفاهیم فازی در قواعد نحوی این زبان گنجانیده شد. به نظر می رسد زبان تعمیم یافته که آن را FATSQ نام نهاده ایم، امکانات مناسبی را جهت کار با داده های زمانی مبهم مهیا ساخته است بی آنکه بخشی از قابلیت های ATSQL را از دست بدهد. به عبارت دیگر تمامی کاربردهای ATSQL با FATSQ نیز کاملاً سازگار است و این امر تضمین می کند که ATSQL بتواند بدون ایجاد مشکل با FATSQ جایگزین شود.

زمینه نسبتاً نوپای پایگاه داده های زمانی جای پژوهش و نوآوری بسیار دارد که می تواند در ادامه این تحقیق و یا به صورت مستقل از آن انجام پذیرد. از جمله این فرصت های پژوهشی می توان به پیاده سازی زبان FATSQ به عنوان نخستین زبان پرسش زمانی فازی، افزودن امکان استفاده از سایر شیوه های کار با داده های زمانی نادقیق به FATSQ و افزودن مفاهیم فازی به سایر مدل های داده ای و زبانهای پرسش زمانی از جمله مدل ها و زبان های زمانی شیءگرا اشاره کرد.

مراجع

- [1] M. H. Bohlen, R.T. Snodgrass, et.al. , "Coalescing in Temporal Databases," *VLDB* , 1996.

¹ coalesce

² timestamp referencing

³ instant extraction

⁴ instant constructor

⁵ interval constructor

⁶ instant or interval comparison predicate

- 7 temporal predicate
- 8 temporal upward compatibility
- 9 syntactically similar snapshot-reducibility
- 10 not restrictiveness
- 11 statement modifier
- 12 fuzzy set
- 13 membership function
- 14 membership degree
- 15 fuzzy number
- 16 fuzzy point
- 17 fuzzy interval
- 18 possibility measure
- 19 necessity measure



روحانی رانکوهی از سال ۱۳۵۸ عضو هیات علمی و در حال حاضر استادیار گروه مهندسی کامپیوتر دانشگاه شهید بهشتی است. او تحصیلاتش را در رشته ریاضی در دانشگاه تهران و سپس در دانشگاه پیر و ماری کوری فرانسه در رشته انفورماتیک انجام داده است. تاکنون ۹ کتاب دانشگاهی ترجمه و/یا تألیف کرده است.

از آن میان کتاب مقدمه *ای بر پایگاه داده ها* در سال ۱۳۷۲ کتاب تشویقی در مراسم کتاب سال شد و نیز کتاب *مفاهیم بنیادی پایگاه داده ها* که در سال ۱۳۸۱ کتاب برگزیده دانشگاهی در دانشگاه تهران و کتاب سال جمهوری اسلامی ایران شد. روحانی مقالاتی نیز در مجلات علمی-فنی داخلی دارد. زمینه های پژوهشی مورد علاقه نامبرده عبارتند از: مهندسی فایلها، سیستمهای پایگاهی (متمکز و نامتمکز)، سیستمهای داده کاوی و کشف دانش.



مرجان نظریان نائینی در سال ۱۳۵۶ در تهران متولد شد. نامبرده در سال ۱۳۷۸ دوره کارشناسی مهندسی نرم افزار را در دانشگاه صنعتی شریف به اتمام رساند و در سال ۱۳۷۹ وارد دوره کارشناسی ارشد مهندسی نرم افزار در دانشگاه شهید بهشتی گردید و در سال ۱۳۸۲ از این برنامه فارغ التحصیل شد. زمینه های مورد علاقه نامبرده پایگاه های داده زمانی و فازی می باشد.