AUT-QPM: چارچوبی نو در ارزیابی پرس و جو برای تصمیم‌گیری در ایجاد پایگاه داده تحلیلی

نگین دانشپور و احمد عبداله‌زاده بارفروش

چکیده: دلیل اصلی شکست سیستم‌های پایگاه داده تحلیلی، عدم تشخیص لزوم ایجاد آنهاست. تحلیل لزوم ایجاد پایگاه داده تحلیلی دارای اهمیت بسیار زیادی است. در این مقاله چارچوبی با نام AUT-QPM[1] برای بررسی لزوم ایجاد پایگاه داده تحلیلی، بر اساس نوع پرس و جوهای مطرح در آن، ارائه می‌گردد. به این منظور ابتدا انواع پرس و جو دسته‌بندی شده و سپس بر روی یک پایگاه داده عملیاتی و پایگاه داده تحلیلی متناظر با آن با سایزهای مختلف اعمال می‌شود. سپس به منظور ارزیابی پرس و جو، پارامترهای مورد بررسی ارائه می‌گردند که عبارتند از زمان پاسخ پرس و جو و تعداد مراجعات به دیسک. با بررسی این پارامترها به منظور پاسخ‌گویی به پرس و جو، ملاحظه می‌شود که در رابطه با پرس و جوهای چندبعدی و مجتمع، وجود پایگاه داده تحلیلی ضروری بوده و در رابطه با پرس و جوهای تو در تو و پیوندی، استفاده از پایگاه داده تحلیلی مفید بوده و برای پرس و جوهای ساده و محاسباتی، استفاده از پایگاه داده عملیاتی مناسب‌تراست.

کلید واژه: **پایگاه داده تحلیلی، پرس و جو، شبیه‌ساز، طراحی پایگاه داده تحلیلی، متدولوژی، مهندسی نرم‌افزار.**

۱- مقدمه

لزوم ایجاد یک ماشین ارزیاب برای پرس و جو[2]، با بررسی علل شکست پایگاه داده تحلیلی[3] مورد اهمیت قرار گرفت. در منابع مختلف ملاحظه شد که در رابطه با چگونگی ساخت پایگاه داده تحلیلی مسیر مشخصی ارائه شده است که شامل دو مرحله اصلی ساخت آن و انتقال داده‌ها می‌باشد [۱] تا [۴]. خلاصه این مراحل، در زیر آورده شده است:

- انتخاب فرآیند تجاری: این فرآیند، فرآیند عملیاتی اصلی در سازمان مورد نظر است. سازمان مورد نظر شامل چندین سیستم است که داده‌های آن برای پایگاه داده تحلیلی جمع‌آوری می‌شوند.
- انتخاب هدف تحلیلی[4]: رکوردهای ذخیره‌شده در جدول حقیقت[5]، که به شکل یک پرس و جوی کلی نیز مطرح می‌گردد.
- انتخاب بعدها[6] و شناسایی ویژگی‌های آن بعد بر اساس هدف تحلیلی: مانند زمان، محصول و مشتری و شناسایی ویژگی‌های آنها.
- انتخاب و تشخیص پارامترهایی از رکوردهای جدول حقیقت که باید ارزش‌دهی شوند.
- طراحی مدل چندبعدی پایگاه داده تحلیلی.
- تشخیص پارامترهای مطرح در پرس و جوها بر اساس هدف تحلیلی و ایجاد مکعب‌های داده.

برای طراحی منطقی پایگاه داده تحلیلی، ابتدا مدل چندبعدی[7] آن ایجاد می‌شود و سپس با استفاده از آن، مکعب‌های داده، برای تسریع پاسخ‌گویی به پرس و جوهای مطرح، ایجاد می‌شوند. مدل چندبعدی، با نمودارهای ستاره‌ای[8]، دانه برفی[9] و صورت فلکی حقایق[10]، بازنمایی می‌شود که در زیر شرح داده شده‌اند:

- نمودار ستاره‌ای: در نمودار ستاره‌ای، پایگاه داده تحلیلی شامل یک جدول حقیقت و مجموعه‌ای از جداول کوچک‌تر دیگر است که جداول بعد نامیده می‌شوند. جدول حقیقت، یک جدول مرکزی بزرگ است که شامل داده‌های بسیار زیادی است که تکراری نیستند. در گراف این نمودار، بعدها حول جدول حقیقت قرار گرفته و تنها با آن، دارای ارتباط یک به چند می‌باشند. در این نمودار، هر بعد با فقط یک جدول بازنمایی می‌شود و این امر سبب تکرار[11] و افزایش سرعت جستجو می‌شود.
- نمودار دانه برفی: این نمودار، شبیه نمودار ستاره‌ای است، با این تفاوت که در آن برخی از جداول بعد، نرمال شده‌اند. این امر باعث کاهش تکرار و صرفه‌جویی اندکی (در مقایسه با بزرگی جدول حقیقت) در فضای ذخیره‌سازی می‌شود. علاوه بر این، در این طرح، سرعت پرس و جو به علت افزایش تعداد عمل پیوند[12]، کاهش می‌یابد. به همین دلیل، در پایگاه داده تحلیلی، اغلب از دیاگرام ستاره‌ای استفاده می‌شود.
- نمودار صورت فلکی حقایق: برنامه‌های کاربردی پیچیده، ممکن است نیاز به چندین جدول حقیقت داشته باشند که دارای ابعاد مشترک می‌باشند. این نوع دیاگرام، به صورت مجموعه‌ای از ستاره‌ها بوده و طرح کهکشانی[13] یا صورت فلکی حقایق نامیده می‌شود.

فعالیت‌های فوق، منجر به ایجاد چارچوب اصلی پایگاه داده تحلیلی

این مقاله در تاریخ ۱ مهر ماه ۱۳۸۵ دریافت و در تاریخ ۱۹ خرداد ماه ۱۳۸۶ بازنگری شد.

نگین دانش‌پور، آزمایشگاه سیستم‌های هوشمند، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، خیابان حافظ، تهران، کدپستی ۱۵۸۷۵-۴۴۱۳ (email: daneshpour@aut.ac.ir).

احمد عبداله‌زاده بارفروش، استاد، آزمایشگاه سیستم‌های هوشمند، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، خیابان حافظ، تهران، کدپستی ۱۵۸۷۵-۴۴۱۳ (email: ahmad@aut.ac.ir).

1. Amirkabir University of Technology Query Processing Methodology
2. Query
3. Data Warehouse
4. Gain
5. Fact Table
6. Dimension
7. Multidimensional Model
8. Star Schema
9. Snowflake Schema
10. Fact Constellation
11. Redundancy
12. Join
13. Galaxy Schema

می‌شوند. پس از طراحی پایگاه داده تحلیلی و ایجاد چارچوب اصلی آن، مراحل زیر به منظور انتقال داده از پایگاه‌های داده عملیاتی متفاوت و متنوع باید صورت گیرد:

- آماده‌سازی: آماده‌سازی برای هر منبعی انجام می‌شود و شامل استخراج داده‌های آن و ویرایش آنها پس از استخراج است.
- یکپارچگی: این قدم شامل تطبیق دادهای منابع مختلف و پالایش آنها می‌باشد که پاکسازی چندمنبعه نیز نامیده می‌شود.
- تحلیل سطح بالا: در این قدم، محاسبه دیدهای تحلیلی از ابعاد پایه بر مبنای هدف تحلیلی صورت می‌گیرد. در واقع در این قدم پارامترهای تحلیلی ایجاد می‌شوند.
- خصوصی‌سازی: این قدم شامل استخراج و خصوصی‌سازی اطلاعات، در واقع ایجاد پایگاه داده تحلیلی خاص[1] می‌باشد.

مراحل مختلف نشان می‌دهد که ساخت پایگاه داده تحلیلی زمان‌بر و پرهزینه است. تصمیم‌گیری اینکه آیا واقعاً در کاربرد مورد نظر ما پایگاه داده تحلیلی لازم می‌باشد یا خیر، بسیار مورد اهمیت بوده و علت اصلی شکست سیستم‌های دارای پایگاه داده تحلیلی، ایجاد آنها در مواقعی است که واقعاً مورد نیاز نمی‌باشند. تشخیص این مسأله بسیار مورد اهمیت است. در منابع مختلف، بیشتر به نحوه ساخت پایگاه داده تحلیلی و معماری آن اهمیت داده شده است و به تشخیص لزوم ایجاد آن تأکید نشده است [۵] و [۶]. متدولوژی ایجاد پایگاه داده تحلیلی شامل قدم‌های متعددی است که اولین قدم در آن شناسایی کاربران سیستم و در صورت نیاز کاربران به تصمیم‌گیری‌های مدیریتی، شناسایی هدف تحلیلی سیستم و نوع پرس و جوهای درخواستی آنها از سیستم است. در این مقاله لزوم ایجاد پایگاه داده تحلیلی، با توجه به نوع پرس و جو که قرار است در سیستم مورد نظر پاسخ‌گویی شود، مورد بررسی قرار می‌گیرد. در واقع با توجه به اینکه سیستم ما باید پاسخ‌گو به چه نوع پرس و جویی باشد، تشخیص می‌دهیم که آیا پایگاه داده تحلیلی لازم است و یا اینکه پایگاه داده عملیاتی، جهت رفع نیازهای کاربران کفایت می‌کند.

با توجه به مطالب فوق، لازم است یک قدم به مراحل ایجاد پایگاه داده تحلیلی اضافه شود. این قدم که در منابع مختلف به آن پرداخته نشده است و قدم اول در ایجاد پایگاه تحلیلی است، شناسایی پرس و جوهای سیستم آتی است. برای انجام این قدم که به صورت کمی در این مقاله بررسی شده است، ماشینی ایجاد می‌شود که با تشخیص نوع پرس و جوهای مطرح، لزوم ایجاد پایگاه داده تحلیلی را مشخص می‌نماید.

پس از شناسایی پرس و جوهای مورد نیاز سیستم، اگر پاسخ‌گویی به نوعی از پرس و جوهای مطرح در کاربرد مورد نظر، نیاز به پایگاه داده تحلیلی داشت، پایگاه داده تحلیلی باید ایجاد شود.

به این منظور ابتدا به تعریف پرس و جو می‌پردازیم. پرس و جو یک واحد کاری است در سیستم‌های اطلاعاتی که بر خلاف تراکنش در سیستم‌های عملیاتی، شامل درج، بهنگام‌سازی و حذف نمی‌باشد. پرس و جو یک برنامه اجرایی است که به داده‌های مختلف دسترسی پیدا کرده و برای پاسخ‌گویی به نیازهای اطلاعاتی استفاده می‌شود.

در این مقاله ابتدا انواع پرس و جو دسته‌بندی شده و ساختار داده‌ای هر یک ارائه می‌شود. هدف از این دسته‌بندی، تشخیص لزوم ایجاد پایگاه داده تحلیلی با توجه به انواع پرس و جو که به آن اعمال می‌شود، بر اساس پارامترهای ارزیابی می‌باشد. سپس پارامترهای ارزیابی سیستم در رابطه با پرس و جو ارائه می‌شود. در بخش چهارم، یک پایگاه داده عملیاتی و پایگاه داده تحلیلی متناظر با آن به منظور انجام آزمایشات معرفی می‌شوند. پس از آن، نحوه پاسخ‌گویی هر یک از انواع پرس و جو بر روی پایگاه داده عملیاتی و پایگاه داده تحلیلی متناظر با آن در سایزهای مختلف ارزیابی می‌شود. در بخش پایانی، نتایج آزمایشات بررسی می‌شود و خلاصه مقاله ارائه می‌گردد.

۲- دسته‌بندی انواع پرس و جو

پرس و جوها شامل انواع مختلفی هستند که با توجه به تعداد جداولی که تحت تأثیر قرار می‌دهند و نوع تأثیردهی و شرایطی که اعمال می‌کنند به هفت گروه تقسیم می‌شوند [۷] تا [۱۰]. این گروه‌ها عبارتند از: پرس و جوی ساده[2]، پرس و جوی مجتمع[3]، پرس و جوی تو در تو[4]، پرس و جوی پیچیده[5]، پرس و جوی محاسباتی[6]، پرس و جوی پیوندی[7] یا چندبعدی[8] و پرس و جوی بزرگ[9] و در بخش‌های زیر به تعریف و تحلیل هر گروه می‌پردازیم. به این ترتیب که ابتدا ساختار داده‌ای هر گروه با شناسایی ورودی‌ها و خروجی آن گروه، به صورت یک تابع تعریف‌شده و سپس با فرمت دستور *Select* ارائه می‌گردد و نهایتاً با یک مثال، پرس و جوی مورد نظر، تشریح می‌شود.

۲-۱ پرس و جوی ساده

ساختار داده‌ای این نوع پرس و جو به فرم زیر است

$$F(attributes, table, condition(s)) \rightarrow value(s)$$

در این نوع پرس و جو، مقادیر ویژگی‌ها، از سطرهایی که دارای شرایط اعمال‌شده هستند، انتخاب می‌شوند و به عنوان خروجی بازگردانده می‌شوند. ساختار داده‌ای این دسته با فرمت دستور *Select* به صورت زیر می‌باشد

```
Select item(s)-list
From table-name
[Where condition(s)]
 [Order By Asc|Desc column(s)]
```

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

```
Select *
From Damage_Table
Where Given_Value<1000000 And
Total_Cost<10000000
Order By Commitment_Id
```

که در آن، مقادیر تمام ویژگی‌های سطرهای جدول *Damage_Table* که دارای شرایط *Given_Value<1000000* و *Total_Cost<10000000* می‌باشند، به صورت مرتب‌شده بر اساس فیلد *Commitment_Id* ارائه می‌شوند. همان‌طور که ملاحظه می‌شود، این نوع پرس و جوها، ویژگی‌های جداول و شرطهای اعمال‌شده بر آنها را به عنوان ورودی

1. Data Mart
2. Simple Query
3. Aggregate Query
4. Nested Query
5. Complex Query
6. Computational Query
7. Join Query
8. Multidimensional Query
9. Large Query

دریافت کرده و مقادیر آن ویژگی‌ها را که دارای شرایط اعمال‌شده هستند به عنوان خروجی بر می‌گردانند.

۲-۲ پرس و جوی مجتمع

ساختار داده‌ای این نوع پرس و جو به فرم زیر است

F(Aggregate_Function(attributes), table, condition(s)) →value(s)
If Aggregate_Function is: Count, Sum, Avg, Max, Min

در این پرس و جوها، تابع مجتمع‌سازی بر روی مقادیر ویژگی‌های سطرهایی که دارای شرایط اعمال‌شده هستند، اعمال می‌شود و خروجی آن بازگردانده می‌شوند. ساختار داده‌ای این دسته با فرمت دستور *Select* به صورت زیر می‌باشد

Select Aggregate_Function(item(s)-list)
From table-name
[Where condition(s)]
[Group By column(s)]
[Having condition(s)]
[Order By Asc|Desc column(s)]

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

Select Sum(Given_Value), Max(Total_Cost)
From Damage_Table
Where Insurance_Code=47 And
Commitment_Id=17

در این مثال، مجموع *Given_Value* و ماکزیمم *Total_Cost* سطرهای جدول *Damage_Table* که دارای شرایط *Insurance_Code=47* و *Commitment_Id=17* می‌باشند، ارائه می‌شوند. همان‌طور که ملاحظه می‌شود، این نوع پرس و جوها، توابعی مجتمع از بعضی از ویژگی‌های جداول و شرطهای اعمال‌شده بر آنها را به عنوان ورودی دریافت کرده و مقادیر آن توابع از ویژگی‌ها را که دارای شرایط اعمال‌شده هستند به عنوان خروجی بر می‌گرداند.

۲-۳ پرس و جوی تو در تو

ساختار داده‌ای این نوع پرس و جو به فرم زیر است

F(attributes, table, condition(s)) →value(s)
If ∋ condition|condition is a Query

در این نوع پرس و جو، مقادیر ویژگی‌ها، از سطرهایی که دارای شرایط اعمال‌شده هستند، انتخاب می‌شوند و به عنوان خروجی بازگردانده می‌شوند، و در آن حداقل یکی از شرطها، خود یک پرس و جو می‌باشد. ساختار داده‌ای این دسته با فرمت دستور *Select* به صورت زیر می‌باشد

Select item(s)-list
From table-name
Where condition(s)

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

*Select * From Damage_Table*
Where Insurance_Code In (Select Insurance_Code From Person_Table Where Sexuality=1) and
Commitment_Id In (Select Commitment_Id From Commitmrnt_Table Where Contract_Id In (Select Contract_Id From Contract_Table Where End_Date<2004/12/29))

در مثال فوق، سطرهایی از *Damage_Table* انتخاب می‌شوند که مقدار فیلد *Insurance_Code* در آن، از پرس و جوی زیر حاصل می‌شود

Select Insurance_Code From Person_Table Where Sexuality=1

و علاوه بر این، مقدار فیلد *Commitment_Id* نیز، از پرس و جوی زیر حاصل می‌شود

Select Commitment_Id From Commitmrnt_Table Where Contract_Id In (Select Contract_Id From Contract_Table Where End_Date<2004/12/29)

این پرس و جو، یک پرس و جوی تو در تو است.

همان‌طور که ملاحظه می‌شود، این نوع پرس و جوها نیز مانند پرس و جوهای ساده، ویژگی‌های جداول و شرطهای اعمال‌شده بر آنها را به عنوان ورودی دریافت کرده و مقادیر ویژگی‌هایی را که دارای شرایط اعمال‌شده هستند، به عنوان خروجی بر می‌گردانند با این تفاوت که در آنها حداقل یکی از شرطها، یک پرس و جو است.

۲-٤ پرس و جوی پیچیده

ساختار داده‌ای این نوع پرس و جو به فرم زیر است

F(attributes, F(attributes, table, condition(s)), condition(s)) →value(s)

در این نوع، مقادیر ویژگی‌ها، از سطرهایی که دارای شرایط خواسته‌شده هستند، انتخاب می‌شود. ساختار داده‌ای این دسته با فرمت دستور *Select* به صورت زیر می‌باشد

Select item(s)-list
From (Select item(s)-list From table-name) As table-name(item(s)-list)
[Where condition(s)]
[Order By Asc|Desc column(s)]

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

*Select * From (Select Name, Family, Insure_Begin_Date From Person_Table) As P_T(N,F,I) Where I<2004/12/2*

همان‌طور که ملاحظه می‌شود، در این نوع پرس و جوها، ویژگی‌های خروجی یک پرس و جو و شرطهای اعمال‌شده بر آنها، به عنوان ورودی دریافت شده و مقادیر ویژگی‌هایی که دارای شرایط اعمال‌شده هستند، به عنوان خروجی باز گردانده می‌شود.

۲-۵ پرس و جوی محاسباتی

ساختار داده‌ای این نوع پرس و جو به فرم زیر است

[F(G(attributes), table, condition(s)) →value(s)]|G(attributes) is a Computational Function of attributes

در این پرس و جو، مقادیر ویژگی‌ها، از سطرهایی که دارای شرایط اعمال‌شده هستند، انتخاب می‌شوند و پس از اعمال تابع G روی آنها، به عنوان خروجی بازگردانده می‌شوند. ساختار داده‌ای این دسته با فرمت

دستور *Select* به صورت زیر می‌باشد

```
Select G(item(s)-list)
From table-name
[Where condition(s)]
 [Order By Asc|Desc column(s)]
```

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

```
Select (Total_Cost*9/10-Given_Value)*100 +
(2*Insurance_Code +
10*Commitment_Id+5*Damage_Id)/4
From Damage_Table Where (2*Insurance_Code +
10*Commitment_Id+5*Damage_Id)/4>10
```

همان‌طور که ملاحظه می‌شود، این نوع پرس و جوها، تابعی محاسباتی از ویژگی‌های جداول و شرط‌های اعمال‌شده بر آنها را به عنوان ورودی دریافت کرده و مقادیر آن توابع را که دارای شرایط اعمال‌شده هستند، به عنوان خروجی بر می‌گردانند.

۲-۶ پرس و جوی پیوندی یا چندبعدی

ساختار داده‌ای این نوع پرس و جو به فرم زیر است

F(attributes, table(s), condition(s)) →value(s)

در این پرس و جو، مقادیر ویژگی‌ها، از سطرهایی که دارای شرایط اعمال‌شده هستند، انتخاب می‌شوند و به عنوان خروجی بازگردانده می‌شوند. ساختار داده‌ای این گروه، با فرمت دستور *Select* به صورت زیر است

```
Select item(s)-list
From table-name1 Join table-name2 On
condition(s)
Where condition(s)
```

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

```
Select *
From Damage_Table Join Person_Table On
Damage_Table.Insurance_Code =
Person_Table.Insurance_Code
Where Person_Table.Sexuality=1
```

همان‌طور که ملاحظه می‌شود، این نوع پرس و جوها، ویژگی‌های بیش از یک جدول و شرط‌های اعمال‌شده بر آنها را به عنوان ورودی دریافت کرده و مقادیر آن ویژگی‌ها را که دارای شرایط اعمال‌شده هستند، به عنوان خروجی بر می‌گردانند. پرس و جوی چندبعدی نیز بیش از یک بعد را در بر داشته و در پایگاه داده تحلیلی مورد نظر می‌باشد و در پایگاه داده عملیاتی، مشابه پرس و جوی پیوندی می‌باشد.

۲-۷ پرس و جوی بزرگ

این نوع پرس و جو، یک پرس و جوی پیوندی است که تعداد جداول آن بیش از دو جدول است و ساختار داده‌ای آن به فرم زیر است

F(attributes, table(s), condition(s)) →value(s)

مثال زیر، نوعی از این پرس و جو را ارائه می‌کند

```
Select D1.Insurance_Code, D1.Commitment_Id,
D1.Total_Cost, D1.Given_Value, ,Contract_Table
.Begin_Date ,Contract_Table.End_Date,
Com1.Commitment,P1.Family, P1.Birth_Date,
P1.Supervisor_Insurance_Code,
Com1.Franchise_Percent, P1.Name, P1.Address,
P1.Insure_Begin_Date  From  Damage_Table as D1
Join Person_Table as P1  On
D1.Insurance_Code=P1.Insurance_Code ,
Damage_Table as D2 join Commitment_Table as
Com1 on D2.Commitment_Id=
Com1.Commitment_Id, Commitment_Table as Com2
join Contract_Table on
Com2.Contract_Id=Contract_Table.Contract_Id
where  D1.Total_Cost<10000000 and
D2.Total_Cost<10000000 and
P1.Birth_Date>'1976/01/01' and
Com1.Commitment_Id<10 and
Com2.Commitment_Id<10 and
D2.Commitment_Id<10 and D1.Commitment_Id<10
```

همان‌طور که ملاحظه می‌شود، این نوع پرس و جوها، ویژگی‌های بیش از دو جدول و شرط‌های اعمال‌شده بر آنها را به عنوان ورودی دریافت کرده و مقادیر آن ویژگی‌ها را که دارای شرایط اعمال‌شده هستند، به عنوان خروجی بر می‌گردانند.

علاوه بر هفت نوع پرس و جوی ذکرشده در بالا، پرس و جوهای ترکیبی از انواع فوق نیز امکان‌پذیر است. خصوصیات پرس و جوهای ترکیبی، از ترکیب خصوصیات عناصر سازنده‌شان ایجاد می‌شود. بنابراین، خصوصیات آنها را دارا بوده و در این مقاله به طور مجزا بررسی نمی‌شوند. مانند: پرس و جوی مجتمع تو در تو[1] و پرس و جوی مجتمع پیچیده[2] [۱۱] و [۱۲].

۳- پارامترهای ارزیابی سیستم در رابطه با پرس و جو

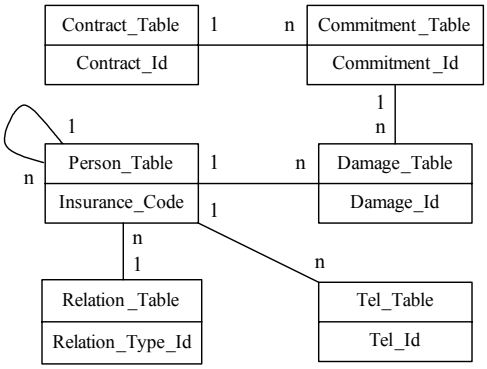
در این مقاله، نحوه پاسخ‌گویی پایگاه داده عملیاتی با پایگاه داده تحلیلی، در رابطه با انواع پرس و جوها، مقایسه می‌شود. به این منظور ابتدا باید پرس و جوها ارزیابی شوند. با بررسی منابع موجود و تحلیل‌های انجام‌شده دریافتیم که به منظور ارزیابی پرس و جوها چندین پارامتر می‌توانند دارای اهمیت باشندکه عبارتند از

۱- زمان واحد پردازش مرکزی[3]: مدت زمان اجرای پرس و جو (زمان پاسخ) [۷] و [۱۳].

۲- تعداد خواندن[4]: تعداد صفحاتی که باید از پایگاه داده خوانده شود، که همان تعداد مراجعات به دیسک است.

۳- کل مدت زمان سپری‌شده[5]: این زمان به میزان بار واحد پردازش مرکزی (برنامه‌های در حال اجرا) بستگی داشته و در زمان‌های مختلف می‌تواند متفاوت باشد.

1. Nested Aggregate Query
2. Complex Aggregate Query
3. CPU Time
4. Logical Read
5. Elapsed Time



شکل ۱: دیاگرام ارتباطی پایگاه داده.

۴- تعداد صفحاتی از دیسک که از پیش خوانده شده است[1]: این تعداد، بستگی به پرس و جوهای اجراشده قبلی داشته و در زمان‌های مختلف می‌تواند متفاوت باشد.

برخی پارامترهای دیگر نیز مانند لزوم محلی‌بودن داده[2] و درج درجا[3] نیز می‌توانند در ارزیابی نحوه پاسخ مد نظر باشند که در این مقاله بررسی نشده‌اند.

با توجه به اینکه پارامترهای ۳ و ۴، تنها وابسته به پرس و جوی مطرح نبوده و به عوامل دیگری نیز بستگی دارند، به منظور ارزیابی انواع پرس و جوی ارائه‌شده در این مقاله، مورد استفاده قرار نمی‌گیرند و تنها از پارامترهای ۱ و ۲ استفاده می‌کنیم.

٤- معرفی پایگاه داده عملیاتی نمونه و پایگاه داده تحلیلی متناظر با آن به منظور انجام آزمایشات

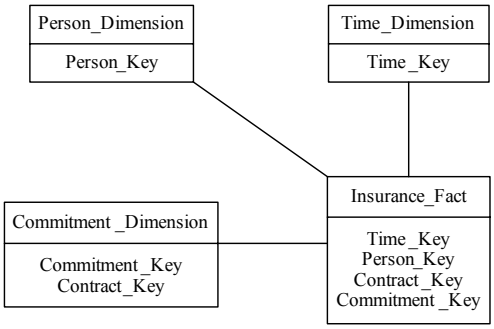
آزمایشات این مقاله، ابتدا بر روی یک پایگاه داده عملیاتی عمومی آزمایشی (یک پایگاه داده عملیاتی بر مبنای داده‌های تولیدشده به صورت تصادفی) و پایگاه داده تحلیلی متناظر با آن انجام شده است و سپس بر روی یک پایگاه داده عملیاتی نمونه، که برای سیستم بیمه سلامتی ایجاد شده است و در شعبات بیمه مورد استفاده قرار می‌گیرد، و پایگاه داده تحلیلی متناظر با آن، انجام شده و نتایج یکسان، حاصل شده‌اند. در این بخش، این پایگاه‌های داده نمونه، معرفی می‌شوند.

در این سیستم، قراردادهای بیمه به صورت سالیانه بوده و حق بیمه هر فرد به صورت ماهیانه مشخص می‌شود و هر شخص از ابتدای یک ماه خاص می‌تواند بیمه شود. در واقع در این سیستم، تعدادی عضو داریم که با حق بیمه‌ای که به طور ماهانه پرداخت می‌کنند، تحت قرارداد هر سال که شامل موارد مختلف است، خسارات خویش را دریافت می‌کنند.

پایگاه داده عملیاتی سیستم بیمه به صورت رابطه‌ای بوده و شامل شش جدول است که در زیر شرح داده می‌شوند:

- جدول اشخاص: این جدول شامل مشخصات شناسنامه‌ای اشخاص، کد بیمه آنها، کد بیمه سرپرست آنها، آدرس، شماره تلفن، وضعیت تأهل، گروه خون، تاریخ شروع بیمه‌شدنشان، وضعیت تحصیل و اشتغال آنها و اینکه هنوز بیمه هستند یا خیر می‌باشد و در پیوست ۱، به طور کامل نشان داده شده است.

نوع رابطه مربوط به جدول شخص، در پیوست ۲ مشخص می‌شود. این نوع عبارت است از: سرپرست، همسر، فرزند پسر، فرزند دختر، مادر، پدر.



شکل ۲: دیاگرام ارتباطی ستاره‌شکل.

شماره تلفن‌های مختلف هر شخص، در پیوست ۳ مشخص می‌شود.

- جدول قرارداد: این جدول شامل شماره قرارداد، تاریخ شروع و خاتمه آن و میزان حق بیمه ماهانه، درصدی از آن که بیمه‌شده باید بپردازد و مسئول انعقاد قرارداد می‌باشد که در پیوست ۴ آورده شده‌اند.
- جدول تعهدات: به ازاء هر قرارداد، سازمان بیمه پرداخت یک سری خسارات را تعهد می‌کند که خصوصیات آنها در پیوست ۵ آورده شده است.
- جدول خسارات: این جدول شامل خسارات انجام‌شده هر شخص می‌باشد که در پیوست ۶ آورده شده است.

دیاگرام ارتباطی جداول فوق در شکل ۱ آورده شده است.

پایگاه داده تحلیلی متناظر با پایگاه داده عملیاتی مورد نظر به صورت چندبعدی بوده و هدف تحلیلی آن عبارت است از:

"چه شخصی در چه زمانی چه هزینه‌ای انجام داده است."

بر این اساس این پایگاه داده دارای یک جدول حقیقت (پیوست ۷) و سه بعد است که در زیر آورده شده‌اند:

- بعد زمان: این بعد زمان انجام خسارات خاص را شرح می‌دهد که از فیلد *Damage_Date* از *Damage_Table* مشتق شده است. در پیوست ۸، فیلدهای آن، آورده شده‌اند.
- بعد شخص: اطلاعات لازم اشخاص برای آنالیز در این جدول آورده شده است. این جدول از *Person_Table* مشتق شده است. این اطلاعات به طور کامل در پیوست ۹ آورده شده‌اند.
- بعد تعهدات: این بعد شامل تعهدات بیمه می‌باشد. از جداول *Commitment_Table* و *Contract_Table* مشتق شده و شامل فیلدهای ارائه‌شده در پیوست ۱۰ است.

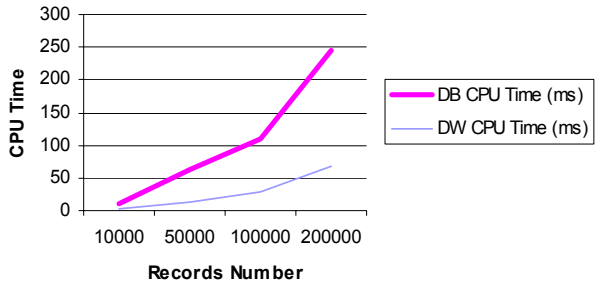
در شکل ۲، دیاگرام ارتباطی جداول فوق آورده شده است.

با توجه به پرس و جوهای مطرح در پایگاه داده مورد نظر، پنج دید تحلیلی نیز در آن پیاده‌سازی می‌شوند، که در پیوست ۱۱، ۱۲، ۱۳، ۱۴ و ۱۵ آورده شده‌اند.

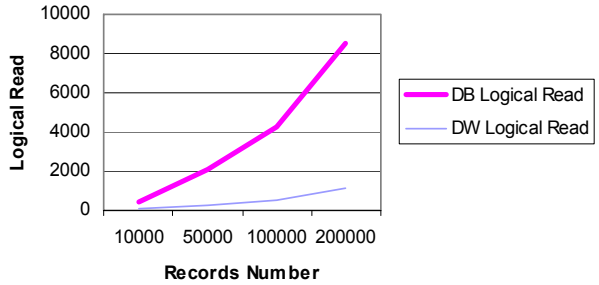
٥- نحوه پاسخ‌گویی پایگاه داده عملیاتی و پایگاه داده تحلیلی به انواع پرس و جو

انواع پرس و جوی ذکرشده در بخش یک، بر روی پایگاه داده عملیاتی و پایگاه داده تحلیلی متناظر با آن که در بخش پیش معرفی شد، اجرا شدند. پرس و جوهای فوق، بر روی پایگاه داده عملیاتی با ۴ سایز مختلف ۱۰۰۰۰، ۵۰۰۰۰، ۱۰۰۰۰۰ و ۲۰۰۰۰۰ رکورد در جداول *Damage_Table* و *Person_Table* آزمایش شده و سپس بر روی پایگاه داده تحلیلی متناظر با آنها نیز آزمایش شد و نتایج آن بر حسب زمان پاسخ (میلی‌ثانیه) و تعداد مراجعه به دیسک [۱۴]، در بخش‌های زیر

1. Read-Ahead
2. Data Locality
3. Inplace Insert



شکل ۳: زمان پاسخ پرس و جو در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۴: تعداد مراجعه به دیسک در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.

آورده شده است. لازم به ذکر است که این نوع پرس و جوها، بر روی پایگاه داده عملیاتی عمومی و پایگاه داده تحلیلی متناظر با آن نیز اجرا شده‌اند و نتایج مشابه حاصل شده‌اند.

۵-۱ نتایج اجرای پرس و جوی ساده

به این منظور پرس و جوی زیر بر روی پایگاه داده عملیاتی با چهار سایز مختلف اعمال شد

```
Select * from Damage_Table where
Given_Value<1000000 and Total_Cost<10000000
order by Commitment_Id
```

سپس پرس و جوی متناظر با آن که در زیر آورده شده است، بر روی پایگاه داده تحلیلی با چهار سایز مختلف اعمال شد

```
Select * from Insurance_Fact where
Given_Value<1000000 and Total_Cost<10000000
order by Commitment_Key
```

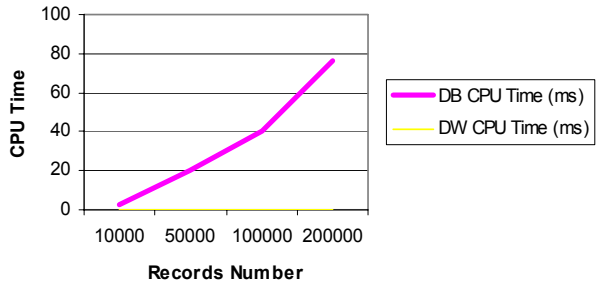
نتایج اعمال این پرس و جوها در شکل‌های ۳ و ۴ آورده شده است.

به نظر می‌رسد که این دو پرس و جو زمان پاسخ و تعداد دسترسی یکسان داشته باشند اما با توجه به اینکه پایگاه داده تحلیلی تنها شامل اطلاعات تحلیلی بوده و فیلدهای اضافه آن حذف شده است، همان‌طور که در نمودارهای ۳ و ۴ می‌بینیم زمان پاسخ پایگاه داده تحلیلی چهار برابر بهتر می‌باشد و تعداد دستیابی به دیسک نیز ۷ برابر بهتر بوده و مستقل از تعداد رکوردها می‌باشد.

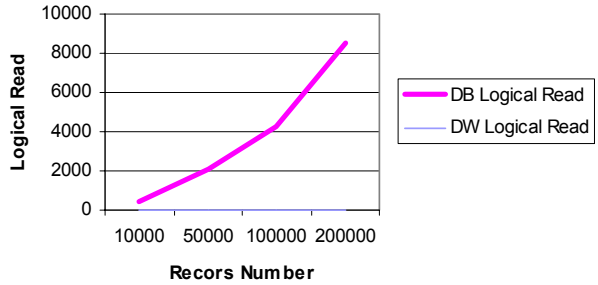
۵-۲ نتایج اجرای پرس و جوی مجتمع

به این منظور پرس و جوی زیر بر روی پایگاه داده عملیاتی با چهار سایز مختلف اعمال شد

```
Select max(Given_Value), max(Total_Cost) From
Damage_Table where Commitment_Id=47
```



شکل ۵: زمان پاسخ پرس و جو در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۶: تعداد مراجعه به دیسک در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.

سپس پرس و جوی متناظر با آن که در زیر آورده شده است، بر روی پایگاه داده تحلیلی با چهار سایز مختلف اعمال شد

```
Select SG_Value,ST_Value from Sum_Fact2 where
Commitment_Key=47
```

با توجه به اینکه پایگاه داده تحلیلی به منظور پاسخ‌گویی به پرس و جوی فوق دارای جدول *Sum_Fact2* از نوع مجتمع می‌باشد، زمان پاسخ پایگاه داده تحلیلی، در هر چهار سایز، صفر میلی‌ثانیه بوده و تعداد مراجعات به دیسک نیز دو بار می‌باشد که در نمودارهای ارائه‌شده در شکل‌های ۵ و ۶ تقریباً صفر بوده و قابل رؤیت نیستند.

۵-۳ نتایج اجرای پرس و جوی تو در تو

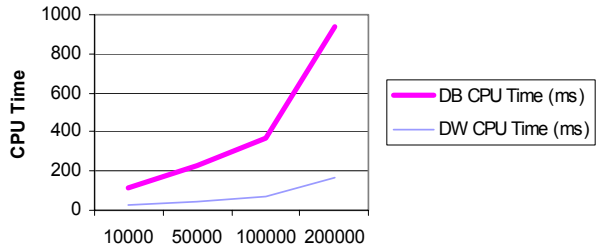
به این منظور پرس و جوی زیر بر روی پایگاه داده عملیاتی با چهار سایز مختلف اعمال شد

```
Select * From Damage_Table Where
Commitment_Id In (Select Commitment_Id From
Commitment_Table Where Contract_Id In  (Select
Contract_Id From Contract_Table Where
End_Date<'2004/12/29'))
```

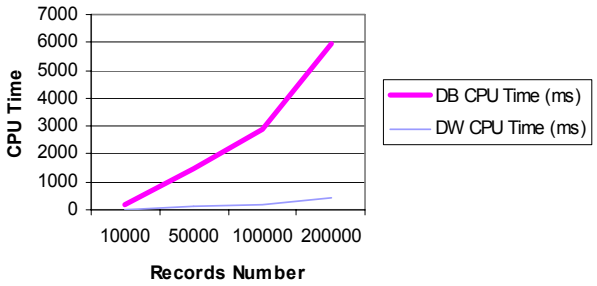
سپس پرس و جوی متناظر با آن که در زیر آورده شده است، بر روی پایگاه داده تحلیلی با چهار سایز مختلف اعمال شد

```
Select * from Year2004_Fact1 where
Year_End_Date<2004
```

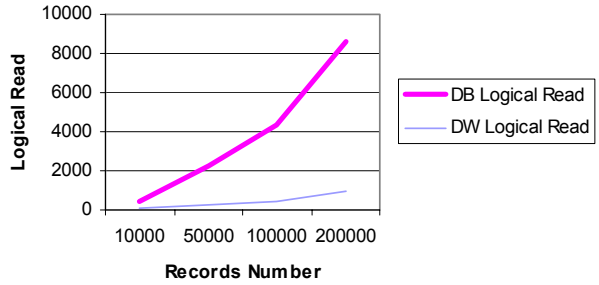
با توجه به اینکه پایگاه داده تحلیلی به منظور پاسخ‌گویی به پرس و جوی فوق دارای جدول *Year2004_Fact1* می‌باشد، که نوعی دید تحلیلی برای آن محسوب می‌شود، [۱۵] تا [۱۹] زمان پاسخ پایگاه داده تحلیلی در مقایسه با پایگاه داده عملیاتی، مستقل از تعداد رکوردها، پنج برابر بهتر می‌باشد و تعداد مراجعات به دیسک نیز مستقل از تعداد رکوردها، نه برابر بهتر می‌باشد و در نمودارهای ارائه‌شده در شکل‌های ۷ و ۸ قابل رؤیت است.



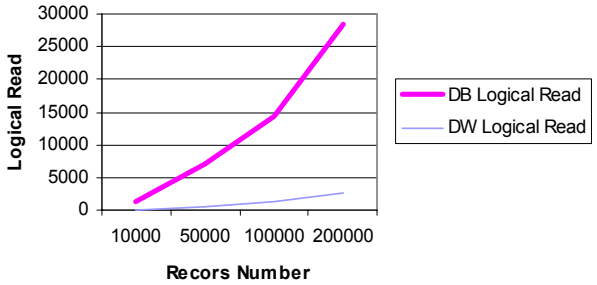
شکل ۷: زمان پاسخ پرس و جو در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۹: زمان پاسخ پرس و جو در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۸: تعداد مراجعه به دیسک در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۱۰: تعداد مراجعه به دیسک در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.

۵-٤ نتایج اجرای پرس و جوی پیوندی

به این منظور پرس و جوی زیر بر روی پایگاه داده عملیاتی با چهار سایز مختلف اعمال شد

*Select * From Damage_Table Join Person_Table*
On Damage_Table.Insurance_Code =
Person_Table.Insurance_Code

سپس پرس و جوی متناظر با آن که در زیر آورده شده است، بر روی پایگاه داده تحلیلی با چهار سایز مختلف اعمال شد

*Select * from FandP*

با توجه به اینکه پایگاه داده تحلیلی به منظور پاسخ‌گویی به پرس و جوی فوق دارای جدول *FandP* می‌باشد، که نوعی دید تحلیلی برای آن محسوب می‌شود، [۱۵] تا [۱۹] زمان پاسخ پایگاه داده تحلیلی در مقایسه با پایگاه داده عملیاتی، مستقل از تعداد رکوردها، چهارده برابر بهتر می‌باشد و تعداد مراجعات به دیسک نیز در آن، مستقل از تعداد رکوردها، یازده بار بهتر می‌باشد، و در نمودارهای اشکال ۹ و ۱۰ قابل رؤیت است.

۵-۵ نتایج اجرای پرس و جوی بزرگ (پرس و جوی دوبعدی)

به این منظور پرس و جوی زیر بر روی پایگاه داده عملیاتی با چهار سایز مختلف اعمال شد

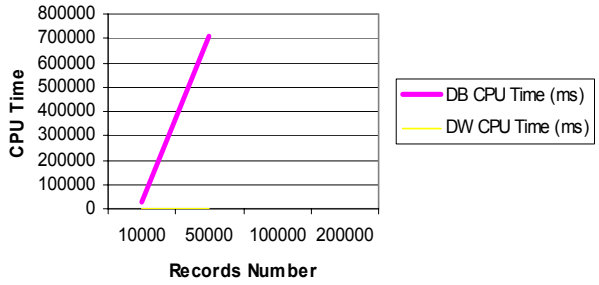
Select D1.Insurance_Code, D1.Commitment_Id,
D1.Total_Cost, D1.Given_Value,
Contract_Table.Contract_Id,
Contract_Table.Begin_Date,
Contract_Table.End_Date, Com1.Commitment,
P1.Family, P1.Birth_Date,
P1.Supervisor_Insurance_Code,
Com1.Franchise_Percent, P1.Name, P1.Address,
P1.Insure_Begin_Date From Damage_Table as
D1 Join Person_Table as P1 On
D1.Insurance_Code =P1.Insurance_Code ,
Damage_Table as D2 join Commitment_Table as
Com1 on D2.Commitment_Id=
Com1.Commitment_Id, Commitment_Table as Com2
join Contract_Table on Com2.Contract_Id=
Contract_Table.Contract_Id where D1.Total_Cost
<10000000 and D2.Total_Cost<10000000 and
P1.Birth_Date>'1976/01/01' and
Com1.Commitment_Id <10 and
Com2.Commitment_Id<10 and
D2.Commitment_Id<10 and D1.Commitment_Id<10

سپس پرس و جوی متناظر با آن که در زیر آورده شده است، بر روی پایگاه داده تحلیلی با چهار سایز مختلف اعمال شد

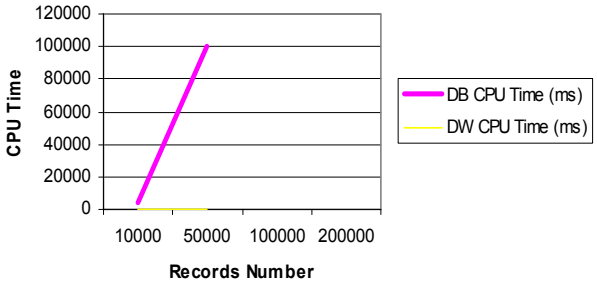
*Select * from Person_Commitment where*
Person_Commitment.Commitment_Key<10

با توجه به اینکه پایگاه داده تحلیلی به منظور پاسخ‌گویی به پرس و جوی دوبعدی فوق دارای جدول *Person_Commitment* می‌باشد، [۱۵] تا [۱۹] با اینکه تعداد خواندن افزایش یافته، زمان پاسخ پایگاه داده تحلیلی بسیار بهتر از زمان پاسخ پایگاه داده عملیاتی بوده و با افزایش تعداد رکوردها و ابعاد، این ضریب افزایش نیز می‌یابد و به علت تفاوت زیاد بین زمان پاسخ پایگاه داده عملیاتی و پایگاه داده تحلیلی، جواب پایگاه داده تحلیلی در نمودار قابل رؤیت نبوده و تقریباً صفر در نظر گرفته شده است. نتایج فوق در شکل‌های ۱۱ و ۱۲ قابل رؤیت است. در واقع زمان پاسخ در پایگاه داده تحلیلی با افزایش سایز و افزایش بعد، نسبت به زمان پاسخ در پایگاه داده عملیاتی با تابع زیر بهبود می‌یابد

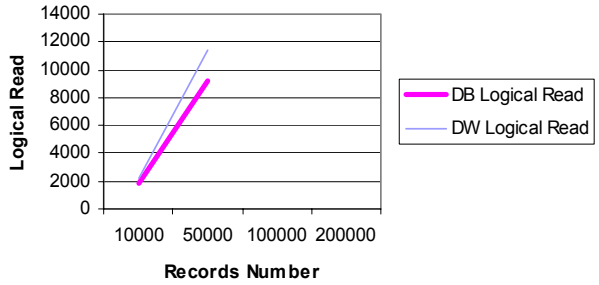
$$\frac{t_k(db)}{t_k(dw)} = (d + \frac{k}{n}) \times \frac{t_n(db)}{t_n(dw)} \qquad (۱)$$



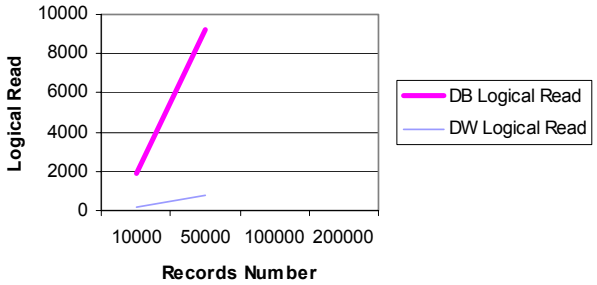
شکل ۱۱: زمان پاسخ پرس و جو در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۱۳: زمان پاسخ پرس و جو در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۱۲: تعداد مراجعه به دیسک در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.



شکل ۱۴: تعداد مراجعه به دیسک در پایگاه داده عملیاتی در مقایسه با پایگاه داده تحلیلی.

در فرمول فوق، $t_k(db)/t_k(dw)$، نسبت زمان پاسخ پایگاه داده عملیاتی به پایگاه داده تحلیلی با تعداد رکورد k (بزرگتر از n) است، و d تعداد ابعاد در پایگاه داده تحلیلی است. k/n، نسبت تعداد رکورد k به n است، و $t_n(db)/t_n(dw)$، نسبت زمان پاسخ پایگاه داده عملیاتی به پایگاه داده تحلیلی با تعداد رکورد n است. این تابع برای پرس و جوهای چندبعدی سازگار بوده و بنابراین، در بخش زیر نیز صادق است.

۵-۶ نتایج اجرای پرس و جوی بزرگ (پرس و جوی سه‌بعدی)

به این منظور پرس و جوی زیر بر روی پایگاه داده عملیاتی با چهار سایز مختلف اعمال شد

```
Select D1.Commitment_Id, D1.Total_Cost,
D1.Given_Value, Contract_Table.Contract_Id,
D1.Damage_Date, P1.Insurance_Code  From
Damage_Table as D1 Join Person_Table as P1  On
D1.Insurance_Code=P1.Insurance_Code ,
Damage_Table as D2 join Commitment_Table as
Com1 on D2.Commitment_Id=
Com1.Commitment_Id, Commitment_Table as Com2
join Contract_Table on
Com2.Contract_Id=Contract_Table.Contract_Id
where  D1.Total_Cost<10000000 and
D2.Total_Cost<10000000 and
P1.Birth_Date>'1976/01/01' and
Com1.Commitment_Id<10 and
Com2.Commitment_Id<10 and
D2.Commitment_Id<10 and D1.Commitment_Id<10
```

سپس پرس و جوی متناظر با آن که در زیر آورده شده است، بر روی پایگاه داده تحلیلی با چهار سایز مختلف اعمال شد

```
Select * from Person_Commitment_Time where
Person_Commitment_Time.Commitment_Key<10
```

با توجه به اینکه پایگاه داده تحلیلی به منظور پاسخ‌گویی به پرس و جوی سه‌بعدی فوق دارای جدول *Person_Commitment_Time* می‌باشد، [۱۵] تا [۱۹] زمان پاسخ پایگاه داده تحلیلی، از تابع ارائه‌شده در بخش پیش، تبعیت کرده و در نمودارهای اشکال ۱۳ و ۱۴ قابل رؤیت است.

با توجه به اینکه دیگر انواع پرس و جو قابل تبدیل به یکی از انواع آزمایش‌شده فوق است، آزمایش بر روی آنها انجام نشده است و بدیهی‌ست که در صورت انجام آزمایش به نتایجی مشابه با نوع قابل تبدیل آن می‌رسیم. به عنوان مثال، پرس و جوی پیچیده قابل تبدیل به پرس و جوی چندبعدی بوده و نتایج ارزیابی مشابه با آن خواهد داشت. پرس و جوی محاسباتی نیز با توجه به اینکه بر روی یک جدول عمل می‌کند، نتایجی مشابه با پرس و جوی ساده خواهد داشت.

۶- خلاصه و نتایج

در مقاله فوق، چارچوبی ارائه شده است که راه‌گشایی جهت تصمیم‌گیری برای ایجاد پایگاه داده تحلیلی می‌باشد. این چارچوب شامل سه مرحله دسته‌بندی انواع پرس و جو، اجرای انواع پرس و جو با در نظر گرفتن دو معیار زمان پاسخ‌گویی و تعداد دسترسی به دیسک، بر روی پایگاه داده عملیاتی و پایگاه داده تحلیلی با چهار سایز مختلف، و سپس ارزیابی نتایج کمی جهت تصمیم‌گیری برای لزوم ایجاد پایگاه داده تحلیلی می‌باشد. به این منظور، یک محیط آزمایشگاهی مناسب، جهت انجام آزمایشات معرفی شد. با بررسی نتایج آزمایشات فوق می‌بینیم که پایگاه داده تحلیلی برای پاسخ به پرس و جوهای از نوع مجتمع و چندبعدی بسیار مناسب بوده و با افزایش ابعاد در رابطه با پرس و جوهای چندبعدی، نتایج بسیار بهتر از قبل شده و لزوم استفاده از پایگاه داده تحلیلی غیر قابل انکار است. در رابطه با پرس و جوهای از نوع تو در تو و پیوندی، با توجه به اینکه دیدهای تحلیلی متناظر با آنها ساخته می‌شود،

جدول ۱: نتایج چارچوب پیشنهادی برای تصمیم‌گیری نوع پایگاه داده لازم.

نوع پرس و جو	استفاده از پایگاه داده تحلیلی	استفاده از پایگاه داده عملیاتی
پرس و جوی بزرگ/پیچیده (پرس و جوی چندبعدی)	توصیه اکید	ابداً توصیه نمی‌شود
پرس و جوی مجتمع	توصیه اکید	توصیه نمی‌شود
پرس و جوی تو در تو	قابل استفاده	قابل استفاده
پرس و جوی پیوندی	قابل استفاده	قابل استفاده
پرس و جوی ساده	ابداً توصیه نمی‌شود	توصیه اکید
پرس و جوی محاسباتی	ابداً توصیه نمی‌شود	توصیه اکید

جدول ۲: جدول شخص (*PERSON_TABLE*).

Field Name	*Data Type*
**Insurance_Code*	*int*
Name	*char(20)*
Family	*char(20)*
Father_Name	*char(20)*
Mother_Name	*char(20)*
Id_Number	*char(20)*
Emission_Place	*char(20)*
Sexuality	*smallint*
Birth_Date	*datetime*
Blood_Group	*char(3)*
Supervisor_Insurance_Code	*int*
Relation_Type_Id	*int*
Address	*text(16)*
Insure_Begin_Date	*datetime*
Marriage_Status	*Smallint*
Active	*Smallint*
Fax_Num	*char(11)*
Office_Tel	*char(11)*
Office_Fax	*char(11)*
Office_Address	*text(16)*
Email_Address	*char(50)*
Homepage_Address	*char(50)*
Mobile_Num	*char(11)*
Study	*char(30)*
Occupation	*char(40)*

باز هم پایگاه داده تحلیلی مفید واقع می‌شود. اما در رابطه با پرس و جوهای از نوع ساده و محاسباتی، با حذف فیلدهای زائد از پایگاه داده عملیاتی و پرس و جوی مورد نظر، به نتایج قابل قبولی خواهیم رسید و دلیلی برای ایجاد پایگاه داده تحلیلی نداریم.

توجه به این نکته ضروریست که ساخت پایگاه داده تحلیلی بسیار پرهزینه و زمان‌بر می‌باشد و بنابراین با توجه به نتایج ارائه‌شده، فقط برای پاسخ به پرس و جوهای چندبعدی و مجتمع، صد در صد توصیه می‌شود و برای پاسخ به پرس و جوهای ساده و محاسباتی ابداً توصیه نمی‌شود و در رابطه با دو نوع دیگر بسته به اینکه در کاربرد مورد نظر، نحوه پاسخ‌گویی خصوصاً زمان آن تا چه حد دارای اهمیت است، قابل استفاده می‌باشد. نتایج فوق به طور خلاصه در جدول ۱ ارائه شده‌اند. خانه‌های این جدول با سه مقدار توصیه اکید، قابل استفاده و ابداً توصیه نمی‌شود پر شده است که به ترتیب به معنی "بسیار مناسب است"، "می‌توان استفاده نمود" و "نباید مورد استفاده قرار گیرد" می‌باشد.

بنابراین با توجه به لزوم تصمیم‌گیری برای ایجاد پایگاه داده تحلیلی، درکاربرد مورد نظر ابتدا پرس و جوهای مورد نیاز سیستم را تشخیص می‌دهیم. سپس با بررسی آن توسط جدول ۱ تصمیم به ایجاد و یا عدم ایجاد پایگاه داده تحلیلی می‌گیریم.

جدول ۳: جدول رابطه (*RELATION_TABLE*).

Field Name	*Data Type*
**Relation_Type_Id*	*int*
Relation_Type	*char(20)*

جدول ۴: جدول شماره تلفن اشخاص (*TEL_TABLE*).

Field Name	*Data Type*
**Tel_Id*	*int*
Insurance_Code	*int*
Tel_Number	*char(12)*

جدول ۵: جدول قرارداد (*CONTRACT_TABLE*).

Field Name	*Data Type*
**Contract_Id*	*int*
Begin_Date	*datetime*
End_Date	*datetime*
Insurance_Premium	*int*
Insurance_Premium_Percent	*int*
Signature	*char(30)*

پیوست

جداول شرح داده شده در بخش چهارم، در این قسمت ارائه می شوند.

۱- جدول شخص.

در این جدول کلید اصلی، فیلد *Insurance_Code* می‌باشد که با علامت * مشخص شده است و مقدار آن از یک شروع شده و به طور اتوماتیک افزایش می‌یابد و تمام فیلدها به جز این فیلد می‌توانند فاقد ارزش[1] باشند. مقدار یک برای فیلد *Sexuality*، نشان‌دهنده جنسیت مرد و دو نشان‌دهنده جنسیت زن می‌باشد. در صورتی که شخصی هنوز بیمه باشد، مقدار فیلد *Active* آن یک و در صورتی که دیگر بیمه نباشد، مقدار این فیلد صفر است. فیلد *Marriage_Status* در صورتی که یک باشد نشان‌دهنده شخص مجرد و در صورتی که دو باشد، نشان‌دهنده شخص متأهل است.

۲- جدول رابطه.

در این جدول کلید اصلی، فیلد *Relation_Type_Id* می‌باشد که با علامت * مشخص شده است و مقدار آن از یک شروع شده و به طور اتوماتیک افزایش می‌یابد و فیلد *Relation_Type* می‌تواند فاقد ارزش باشد.

۳- جدول شماره تلفن اشخاص.

در این جدول کلید اصلی، فیلد *Tel_Id* می‌باشد که با علامت * مشخص شده است و مقدار آن از یک شروع شده و به طور اتوماتیک افزایش می‌یابد و دو فیلد دیگر می‌توانند فاقد ارزش باشند.

۴- جدول قرارداد.

در این جدول کلید اصلی، فیلد *Contract_Id* می‌باشد که با علامت * مشخص شده است و مقدار آن از یک شروع شده و به طور اتوماتیک افزایش می‌یابد و بقیه فیلدها می‌توانند فاقد ارزش باشند.

۵- جدول تعهدات.

در این جدول کلید اصلی، فیلد *Commitment_Id* می‌باشد که با علامت * مشخص شده است و مقدار آن از یک شروع شده و به طور اتوماتیک افزایش می‌یابد و بقیه فیلدها می‌توانند فاقد ارزش باشند. فیلد *Commitment_Type* در صورتی که یک باشد، نشان‌دهنده این است

1. Null

جدول ۶: جدول تعهدات (*Commitment_Table*).

Field Name	Data Type
*Commitment_Id	int
Commitment	char(50)
Contract_Id	int
Max_Person	int
Max_Family	int
Franchise_Percent	int
Commitment_Type	Smallint

جدول ۷: جدول خسارت (*Damage_Table*).

Field Name	Data Type
*Damage_Id	int
Insurance_Code	int
Commitment_Id	int
Total_Cost	int
Given_Value	int
Damage_Date	datetime
Not_Given_Value	int
Damage_Place	char(30)
Damage_place_Address	text(16)
Signature	char(30)
Reason	char(40)
Remedy_Period	char(30)

جدول ۸: جدول حقیقت (*Insurance_Fact*).

Field Name	Data Type
*Time_Key	int
*Person_Key	int
*Commitment_Key	int
Total_Cost	int
Given_Value	int
*Contract_Key	int

جدول ۹: بعد زمان (*Time_Dimension*).

Field Name	Data Type
*Time_Key	int
Year_Field	int
Quarter_Field	smallint
Month_Field	smallint
Day_Field	smallint

جدول ۱۰: بعد شخص (*Person_Dimension*).

Field Name	Data Type
*Insurance_Code	int
Name	char(20)
Family	char(20)
Age	int
Supervisor_Insurance_Code	int
Address	char(10)
Year_Insure_Begin_Date	int

جدول ۱۱: بعد تعهد (*Commitment_Dimension*).

Field Name	Data Type
*Commitment_Key	int
*Contract_Key	int
Year_Begin_Date	int
Year_End_Date	int
Commitment	char(50)
Franchise_Percent	int

که تعهد مورد نظر از نوع بیمارستانی است و در صورتی که دو باشد، از نوع پاراکلینیکی.

۶- جدول خسارات.

در این جدول، کلید اصلی، فیلد *Damage_Id* می‌باشد که با علامت * مشخص شده است و مقدار آن از یک شروع شده و به طور اتوماتیک افزایش می‌یابد و بقیه فیلدها می‌توانند فاقد ارزش باشند.

جدول ۱۲: *FandP* (*Insurance_Fact Join Person_Dimension*).

Field Name	Data Type
Time_Key	int
Person_Key	int
Commitment_Key	int
Total_Cost	int
Given_Value	int
Contract_Key	int
Insurance_Code	int
Name	char(20)
Age	int
Supervisor_Insurance_Code	int
Address	char(10)
Year_Insure_Begin_Date	int
Family	char(20)

جدول ۱۳: *Person_Commitment* (*Insurance_Fact Join Join Commitment_Dimension Person_Dimension*).

Field Name	Data Type
Time_Key	Int
Person_Key	Int
Commitment_Key	Int
Total_Cost	Int
Given_Value	Int
Contract_Key	Int
Year_Begin_Date	Int
Year_End_Date	Int
Commitment	char(50)
Franchise_Percent	Int
Name	char(20)
Family	char(20)
Age	Int
Supervisor_Insurance_Code	Int
Address	char(10)
Year_Insure_Begin_Date	Int

جدول ۱۴: *Person_Commitment_Time* (*Insurance_Fact Join Person_Dimension Join Commitment_Dimension Join Time_Dimension*).

Field Name	Data Type
Time_Key	Int
Year_Field	int
Commitment_Key	int
Contract_Key	int
Person_Key	int
Total_Cost	int
Given_Value	int

جدول ۱۵: *Sum_Fact2* (*Max(Given_Value) and Max(Total_Cost) for each Commitment*).

Field Name	Data Type
*Commitment_Key	int
SG_Value	int
ST_Value	int

۷- جدول حقیقت.

۸- بعد زمان.

۹- بعد شخص.

۱۰- بعد تعهد.

۱۱- دید تحلیلی *FandP*.

۱۲- دید تحلیلی *Person_Commitment*.

۱۳- دید تحلیلی *Person_Commitment_Time*.

۱۴- دید تحلیلی *Sum_Fact2*.

۱۵- دید تحلیلی *Year2004_Fact1*.

جدول ۱۶: *YEAR2004_FACT1*.

Field Name	*Data Type*
**Time_Key*	*int*
**Person_Key*	*int*
**Commitment_Key*	*int*
**Contract_Key*	*int*
Total_Cost	*int*
Given_Value	*int*
Year_End_Date	*int*

مراجع

[1] R. Kimball and M. Ross, *The Data Warehouse Toolkit: the Complete Guide to Dimensional Modeling*, 2nd Edition, John Wiley & Sons, pp. 1-27, 2002.

[2] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, NewYork, NY: Morgan Kaufman, pp. 39-98, 2001.

[3] R. Kimball and J. Caserta, The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data, John Wiley & Sons, pp. 29-52, 2004.

[4] C. Imhoff, N. Galemmo, and J. G. Geiger, *Mastering Data Warehouse Design: Relational and Dimensional Techniques*, John Wiley & Sons, pp. 3-26, 2003.

[5] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite, The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses, John Wiley & Sons, pp. 31-40, 1998.

[6] P. Ponniah, *Data Warehousing Fundamentals*, pp. 63-86, 2001.

[7] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, Fifth ed. MC Graw-Hill, pp. 75-115, 2005.

[8] J. Ullman and J. Widom, *A First Course in Database Systems*, Prentice Hall, New Jersey, 2001.

[9] C. J. Date, *An Introduction to Database Systems*, Eight ed. Addison-Wesley, 2004.

[10] G. Gottlob, C. Koch, and K. U. Schulz et all, *Conjunctive Queries Over Trees*, ACM, PODS, pp. 189-200, 2004.

[11] K. L. Tan, C. H. Goh, and B. C. Ooi, "Online feedback for nested aggregate queries with multi-threading," in *Proc. of the 25th VLDB Conf.*, pp. 18-29, Sep. 1999.

[12] C. Y. Chan and Y. E. Ioannidis, "Hierarchichal prefix cubes for range-sum queris," in *Proc. of the 25th VLDB Conf.*, pp. 675-686, Sep. 1999.

[۱۳] ن. دانشپور، *رویکرد جدید در بهنگام‌سازی پایگاه پردازش تحلیلی*، پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی امیرکبیر، ۱۳۸۱.

[14] B. McGehee, Use SET STATISTICS IO and SET STATISTICS TIME to Help Tune Your SQL Server Queries, Pipelines Newsletter, Feb. 2005.

[15] A. Tsois and T. K. Sellis, "The generalized pre-grouping transformation: aggregate-query optimization in the presence of dependencise," in *Proc. of the 29th VLDB Conf.*, vol. 29, pp. 644-655, Sep. 2003.

[16] R. Kaushik, R. Ramakrishnan, and V. T. Chakaravarthy, *Synopses for Query Optimization: A Space - Complexity Perspective*, ACM, PODS, pp. 201-209, 2004.

[17] F. Scarcello, G. Greco, and N. Leone, *Weighted Hypertree Decompositions and Optimal Query Plans*, ACM, PODS, pp. 210-221, 2004.

[18] R. Chirkova, C. Li, and J. Li, "Answering queries using materialized views with minimum size," *VLDB J.*, vol. 15, no. 3, pp. 191-210, Sep. 2006.

[19] S. Cohen, W. Nutt, and Y. Sagir, "Rewriting queries with arbitrary aggregation functions using views," *ACM Trans. on Database Systems*, vol. 31, no. 2, pp. 672-715, Jun. 2006.

نگین دانشپور تحصیلات خود را در مقطع کارشناسی مهندسی کامپیوتر–سخت‌افزار در سال ۱۳۷۸ از دانشگاه شهید بهشتی و کارشناسی ارشد مهندسی کامپیوتر– نرم‌افزار در سال ۱۳۸۱ از دانشگاه صنعتی امیرکبیر به پایان رسانده است و هم‌اکنون دانشجوی دکتری مهندسی کامپیوتر–نرم‌افزار در دانشگاه صنعتی امیرکبیر می‌باشد. نام‌برده از سال ۱۳۸۴ در دانشکده مهندسی برق دانشگاه شهید رجایی مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: پایگاه داده، پایگاه داده تحلیلی، خصوصاً انتخاب دید مناسب برای ساخت پایگاه داده تحلیلی، سیستم‌های تصمیم یار، داده‌کاوی و مهندسی نرم‌افزار.

احمد عبداله‌زاده بارفروش تحصیلات خود را در مقاطع دکتری علوم کامپیوتر از دانشگاه بریستل انگلستان به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر می‌باشد. نام‌برده از سال ۱۳۷۹ الی ۱۳۸۱ به عنوان استاد مدعو در دانشگاه‌های مریلند آمریکا و ارسی پاریس مشغول به کار بوده است. دکتر عبداله‌زاده کتاب "مقدمه‌ای بر هوش مصنوعی توزیع‌شده" را در سال ۱۳۸۶ تألیف نمود. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: تکنیک‌های هوش مصنوعی، هوش مصنوعی توزیع‌شده، مذاکره خودکار، سیستم‌های خبره، پردازش زبان طبیعی، سیستم‌های تصمیم‌یار، هوش تجاری، پایگاه داده تحلیلی، داده‌کاوی و مهندسی نرم افزار.