



## تلاقی هوش تجاری عملیاتی با فرآیند تجارت

فاطمه جزمحمدپور<sup>۱</sup>

دانشگاه آزاد اسلامی واحد زنجان(سایت امام(ره))

### چکیده

از زمانی که معمارهای هوش تجاری از تصمیم گیری راهبردی آفلاین به تصمیم گیری راهبردی موثر آنلاین تغییر رویه دادند، طراحی ثبت خط فرآیند ETL (استخراج، تغییر شکل و بارگذاری) بسیار پیچیده تر از قبل شد. در این مقاله مشکل پیش رو در بخش طراحی و اجرای ETL را توصیف می کنیم و درباره نتیجه حاصله آن صحبت کرده و با این مشکلات درگیر خواهیم شد. سرانجام روی هوش شناسی لایه لایه توجه خواهیم کرد که با مدل کردن مناسب عملیات فرآیند تجارت، اطلاعات مورد نیاز و هدفهای سطح خدمت دهی مطلوب حاصل خواهد شد و همچنین به پردازش سیستمی توسط طراحی منطقی ناشی از اجرای فیزیکی می پردازیم. عنصر اصلی بدست آمده خصوصیات آشکار کیفیت انواع هدفهاست (که به این مجموعه هدفها QOX می گوئیم) که در سطح تجارت معین است و استفاده از این هدفها برای بدست آوردن حد مطلوب طراحی منطقی و فیزیکی سطح هاست.

کلید واژه: هوش تجاری، فرآیند ETL، پایگاه داده OLTP، مدل BPMN، اهداف QOX.

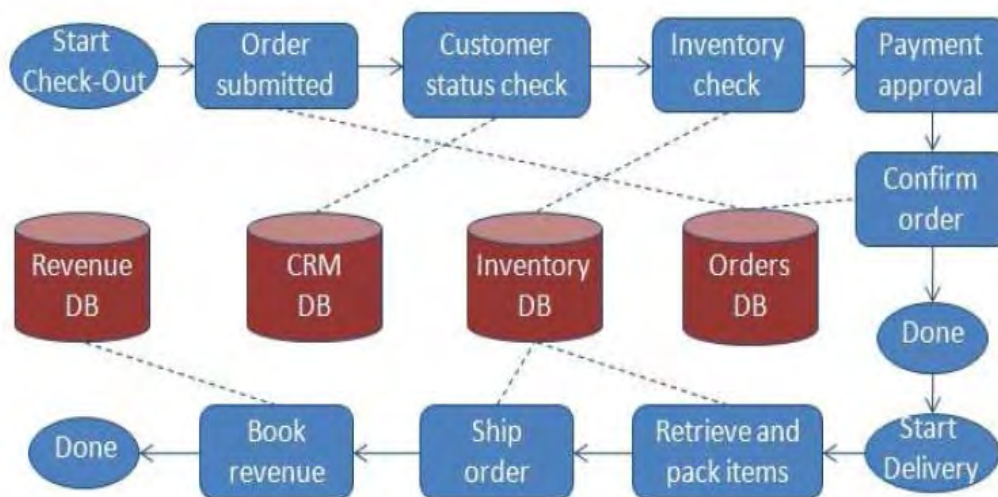
### ۱- مقدمه

معماری هوش تجاری امروزه به طور نمونه شامل مخزن اطلاعات است که اطلاعات را از چندین منبع اطلاعات مهم جمع می کند و خدماتی متنوع از قبیل آزمایش، گزارش و ابزارهایی جهت تحلیل را ارائه می دهد. در نهایت در این معماری یک خط لوله برای مجموع اطلاعات وجود دارد که برای محاسبه مخزن اطلاعات توسط خارج کردن اطلاعات از منابع توزیع شده و معمولاً منابع موثر نامتوازن استفاده می شود که: پاک می کند، جمع آوری می کند و اطلاعات را تغییر می دهند و آن را در داخل مخزن اطلاعات بارگذاری می کند. خط لوله های جمع کننده اطلاعات قدیمی یک فرآیند ناپیوسته است و معمولاً توسط ابزارهای ETL تکمیل می شود (۱ و ۲). در زمان های گذشته، سیستم های BI برای حمایت از تصمیم گیری های آفلاین و استراتژیک بود که نیازمندیهای اطلاعات با گزارشهای دوره ای و آزمایش های تحلیلی تاریخی ارضا می شد. عملیات فرآیند تجارت و ابزارهای تحلیلی جدا نگهداشته

<sup>۱</sup> دانشجوی کارشناسی مهندسی تکنولوژی نرم افزار، تلفن ۰۹۱۲۲۴۱۷۰۹۸، پست الکترونیکی fjozmohammadpour@yahoo.com

می شد. OLTP تشکیل دهنده پایگاه داده ای وارد عمل شده و سپس مخزن اطلاعات را اجرا می کرد و ETL ساختن نقشه بین این دو را اجرا می کرد. از بررسی های به عمل آمده در زمینه پروژه های BI دریافته می شود که اغلب ۶۰-۷۰٪ تلاش ها به سمت طراحی و پیاده سازی ETL سوق پیدا می کند. از زمانی که سرمایه گذارها بیشتر بصورت خودکار درآمد، چرخه اطلاعات و معماری های BI بلادرنگ مجبور شدند تا نتیجه بگیرند که از هوش تجاری موثر استفاده کنند که آن بصورت آنلاین و پیوسته تصمیم گیری ها را به فرآیند تجاری موثری از سرمایه گذاری تبدیل می کرد (۳). این تحمیل کردن بیش از همه چیز به چالش هایی در خط لوله های یکپارچه احتیاج داشت. تعدادی از این چالش ها را مطرح کرده و راه حل های جدید برای آدرس دهی طراحی ETL ها پیشنهاد می کنیم. برای متوجه شدن راه حل، از یک مثال ساده استفاده می کنیم، گردش کار، یک سرمایه گذاری فرضی، پیوسته و جزئی را فرض کنید. فرآیند تجاری برای پذیرش درخواستهای مشتریان، تکمیل کردن و بار گذاری درخواست ها و ثبت کردن سود آنها را در نظر بگیرید. اینچنین فرآیند درخواست سود تعداد گامها، بهره برداری از مخازن اطلاعاتی مختلف موثر و منابع اطلاعاتی سرمایه گذاری را درگیر می کند. (شکل ۱ الف) فرض کنید یک مصرف کننده یک سری سایتهای کوچک را جستجو می کند و مواردی را به کارت خرید خود اضافه می کند.

سرانجام، مشتری آماده است که خرید کند که فرآیند واریسی را شروع می کند. این تبعیت باعث ورود به مخزن اطلاعات درخواستها می شود و سپس وضعیت مشتری برای تایید اعتبار درخواست چک می شود. سپس مخزن اطلاعات چک می کند تا مطمئن شود محصول در مخزن وجود دارد. در این لحظه، درخواست می تواند انجام شود بنابراین وجه مشتری پردازش می شود. یکبار که تأیید شد، در اولین قدم فرآیند تحویل دادن شروع می شود. این موارد از موجودی کسر شده و بسته بندی می شود. نهایتاً درخواست ارسال می شود و سود درخواست به مخزن اطلاعات سود مالی اضافه می شود. در مثال ما سود محاسبه نمی شود تا درخواست ارسال شود.



(شکل ۱ الف)

BI عملیاتی و موثر احتیاجات جدیدی را به ETL تحمیل می کند که به سختی می توان در قراردادهای امروزی دید. ما ۳ چالش را مورد بحث قرار می دهیم.

## ۱.۱- نگاه پیوسته موثر به سرمایه گذاری :

در معماری مرسوم BI، مخزن اطلاعات یک نگاه تاریخی به سرمایه گذاری فراهم می کند. برای مثال برای فراهم کردن گزارش فروش هفتگی، موارد فروش زیاد، گزارش فصلی یا ساختن قطعات مدلها برای مشتریان می تواند استفاده شود. این معماری معمولاً مخزن اطلاعات موثر را یکی می کند. تا یک نگاه فوری از اطلاعات ثبت شده در پایگاه داده OLTP فراهم کند. هنوز، این معماری یک نگاه کامل و بلادرنگ از تمام سرمایه گذاری ها که توسط مدل های جدید نرم افزار های کاربردی BI لازم است را فراهم نمی کند. برای مثال: فرض کنید ما می خواهیم پیشنهاد های ویژه ای به برخی از مشتریان مخصوص برحسب گذشته خریدشان، فعالیت های اخیر در جستجو کردن، سود امروز و موجودی اخیرمان بدهیم. این مقصود احتیاج به اطلاعات از پایگاه داده OLTP (موجودی اخیر، جستجو های اخیر مشتریان)، مخزن اطلاعات (بخش مشتریان، پیشینه خرید) و اطلاعات پیوسته (سود امروز، شامل درخواست هایی که هنوز ارسال نشده) دارد که می تواند در مسیر خودش به مخزن داده ها قرار بگیرد. این نگاه های سرمایه گذاری بسیار پیچیده از جهت طراحی، اجرا و نگهداشتن می باشد و همچنین ابزار متداول BI حمایت کوچکی برای آنها را ترتیب می دهند.

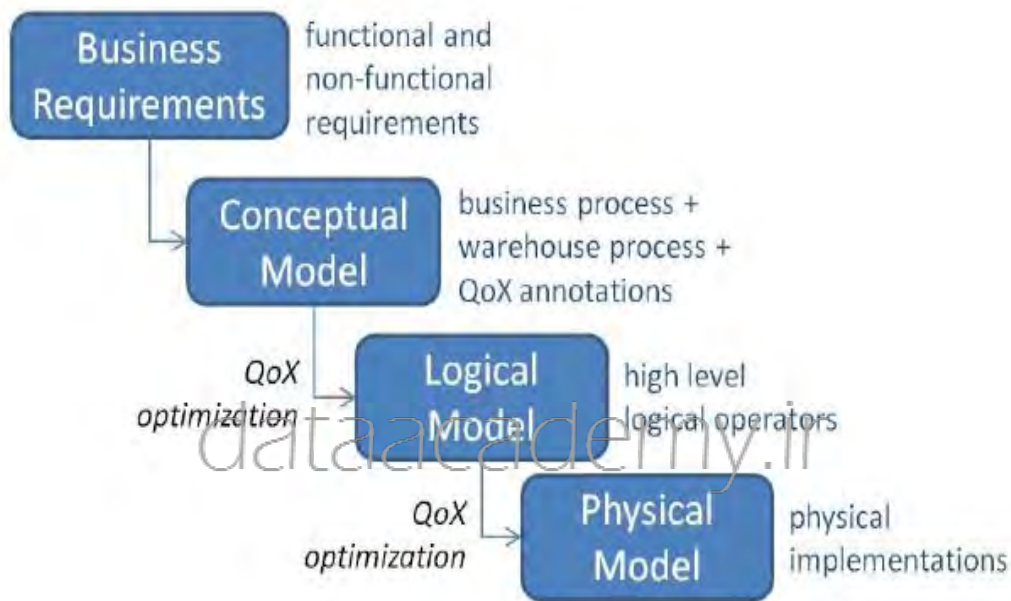
## ۱.۲- طراحی با توجه به هدف :

توجه به ETL به طرز کارآمدی صحیح و اجرای کافی خواهد بود که نشانگر این است که طراحی نقشه از منبع اطلاعات به مخزن باید به طور صحیح باشد و اجرای آنها باید در مدت زمان کوتاهی صورت گیرد. اگر چه، توجه فقط به سودمندی و اجرا بقیه فاکتور های مهم تجارت را از دست می دهد (مثل دوباره بدست آوردن، قابلیت نگهداشتن، قابلیت اطمینان) که باعث می شود بدست آوردن کیفیت سخت تر شود که برای گسترش موفق ETL لازم است. این موضوع بخصوص یک حقیقت برای BI های عملکردی است که برای آنها باید یک محدوده بزرگ برای رقابت موضوعی فراهم شود. قابلیت اطمینان بالا شاید برای اجزای گردش مربوط به سود احتیاج باشد مثلاً از دست دادن دیتاهایی که لیست لینک های ثبت شده را به کامپیوتر دارند و با یک کلیک به آنها دسترسی پیدا می کنیم قابل قبول است در حالیکه از دست دادن وجه بدین صورت نیست. نتیجتاً آنچه احتیاج است انجام شود طراحی ETL آن با هدفهایش که بسیار ساده قابل دسترسی باشد و بتواند رسیدگی به داد و ستدهایش را بهینه سازی کند.

## ۳.۱- سیر تکاملی طراحی:

بکار گیری نوعی ETL ماههای بسیار زیادی را برای شروع با احتیاجات تجارت و هدفها طراحی، زمینه یابی زیر ساختها، طراحی منطقی و ذهنی و به حداکثر رسیدن در طراحی فیزیکی و اجرا صرف می کند. در جهان ایده آل، احتیاجات هیچ وقت تغییر نمی کند. در جهان واقعی و مخصوصاً در BI عملیاتی، همزمان با اینکه تجارت رشد و نمو می کند احتیاجات به سرعت تغییر می کند. برای مثال یک فرآیند درخواست سود را مدنظر بگیرید که با یک تاخیر برای فرآیند تأیید وجه انتظار می رود ولی چند ماه بعد عاملیت اعتبار تاخیر را دو برابر می کند. این تاثیرات تمام خط لوله ETL را به پایین می راند و شاید بصورت قابل توجهی دوباره طراحی شود تا هدفهای سطح سرویس دهی را باقی نگه دارد. روش شناسی که ماههای اضافی زیادی را برای وفق دادن احتیاج دارد در BI عملکردی قابل استفاده نخواهد بود.

ما روش شناسی لایه لایه را توصیف می کنیم که در حالت متوالی ، پالایش کم کم از نیازمندیهای سطح بالای تجارت را بصورت درآمد درمی آورند که توسط چندین سطح از مشخصات واقعی طبق مدل‌های اجرایی انجام می شود. (شکل اب) در این سطح طراحی ، کیفیتهای مختلف (هدفهای QOX) از سطحهای بالا پالایش یا جدا می شوند. این لایه لایه ها موقعیتهای دستیابی مناسبی را برای گرداننده QOX در بهینه سازی این سطح متوالی فراهم می کند. با تلفیق طراحیها و هدفها در سطح های متوالی پالایش ، ما بهتر می توانیم هدفها را پیگیری کرده و به سرعت طراحی جدید را بعنوان نتیجه تجارت تکمیل کنیم.



(شکل اب)

یک ویژگی مهم در طراحی مدل استفاده از مدل های فرآیند تجارت برای طراحی مفهومی (سطح بالا) است. این دارای چندین مزیت است . این مدلها یک رسمیت متحد برای مدل کردن فرآیند محصولات (عملیاتی) مثل درخواست به سود به خوبی فرآیندهایی که مخزن اطلاعات و سرمایه گذاری متوسط را فراهم می کنند و آنها را محاسبه می کند. این مدل طراحی ETL را از دید تجاری ممکن می کند که جزئیات اجرای سطح پایین را مخفی کند بنابراین تشخیص (موافقتهای سطح سرویس دهی<sup>۱</sup>) و استاندارد ها توسط تحلیل گره های تجاری را آسان می کند. بطور خلاصه ، مدل های بدست آمده فرآیند تجاری بصورت اهرمی کار می کنند تا هوش تجاری عملیاتی را فعال کنند. این مدل از دیتاهای سرمایه گذاری یک نگاه پیوسته بدست می آورد و آنها را با هدفهای طراحی سطح بالا پیوند می زند که برای بهینه ساختن فرآیند های ETL و اجرای آنها استفاده می شود.

<sup>1</sup> - SLAs

## ۲- نگاهی تجاری جامع بر چرخه QOX :

این بخش نظر ما را برای بدست آوردن نگاه جامع، پیوسته از اعمال در سطح فکری جلب می کند و از اینکه چگونه به طراحی منطقی ETL دست بیابیم بحث می کند. واقعیت این است در مخزن دیتا هدفهای تجاری و رویدادهایی از سود تصمیم گیری را معین می کند. منابع دیتا برای این حقیقت ها کاملاً مربوط به پایگاه های داده OLTP می باشد که توسط فرآیند های عملکردی تجارت تغییر می کنند. برای نمونه بازرسی، تحویل دادن، گردش ETL طرح نقشه بین منابع هدف ها و محصولات نهایی در مخزن ها را معین می کند. امروزه اگر چه، ابزار ETL مدل کردن این طراحی نقشه ها را در سطح مفهومی حمایت نمی کند (در بخشهای هدفهای تجاری) ترجیحاً، آنها فقط طراحی منطقی ETL را در شرح هدفهای مثل جدولها، اندیسها، فایلها، لینکهای ارتباطی را حمایت می کنند. ما براین باوریم که مدل کردن ETL در سطح مفهومی می تواند به BI عملکردی در تعداد روشهای سود دهد یا کمک کند. اولاً مدل کردن ETL در سطح مفهومی استفاده کننده های مخزن را فعال می کند تا منشا مخزن دیتا در مدت تجارت که آنها به آن توجه دارند را ببینند. دوماً این مدل کردن از دیتا ها یک نگرشی از سرمایه گذاری بروز شده را فراهم می کند که توسط مکان میانی خط لوله ETL مشخص می شود. این نگاه میانی موقعیت جدید برای ابزارهای عملکردی بوجود می آورد که بلادرنگ بتوانند در هر زمانی برای تصمیم گیری راهبردی استفاده شوند یا از تلف کردن زمانی که برای رفرش کردن مخزن لازم است اجتناب شود. سوماً این مدل مفهومی می تواند برای چرخه مدل منطقی برای ETL استفاده شود.

دست آوردهای ما برای استفاده BPMN برای مدل مفهومی است. چون BPMN می تواند برای فرآیند های مدل سازی تجاری عملکردی استفاده شود، این می تواند یک شکل دهی معمول به مدل ها فراهم کند که منبع اطلاعات برای سرمایه گذاری را فراهم کند. برای این منظور (از قبیل اندازه و ابعاد، ویو و ...) شیء در مخزن است که حقیقت فرآیند تجاری BPMN را بهم مربوط می کند که نشان می دهد که شیء چگونه در خارج از فرآیند تجارت مؤثر ساخته می شود. تحقیقات در زمینه فرآیند تجارت راهبردی داخل شده و از آن برای فرستادن پیام برای تمام فرآیند های واقعی که از دیتای آن در این فرآیند استفاده می شود. اما ۳ چالش در استفاده از BPMN وجود دارد.

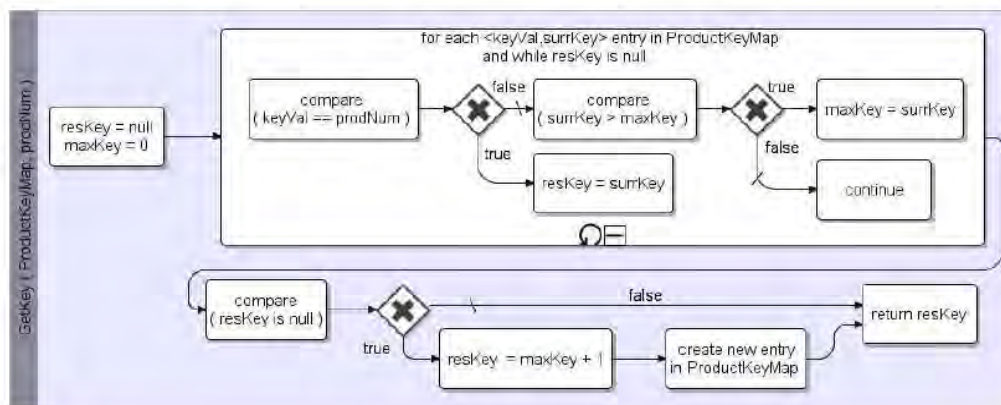
۱.۲- چالش اول این است که برای گردش ETL منطقی ما یک راه احتیاج داریم تا قطعات فرآیند BPMN واقعی را به اپراتور های منطقی ETL مشابه مربوط کند. برای انجام این کار، ما از ۳ تکنیک استفاده می کنیم.

۱-۱-۲- استفاده از زبان اصطلاحات برای بیان روشها در زمان دشواری فرآیند واقعی: اصطلاحات به راحتی می توانند به اپراتورهای ETL منطقی وصل شوند.

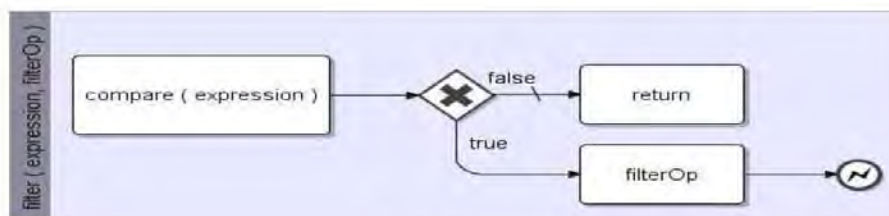
۲-۱-۲- توسعه ماکرو مثل (شکل ۲ الف) که نمونه ای برای عملکرد اخیر ETL از تولید مهم آن نمایش می دهد.

۲-۱-۳- الگوهای برای وصل کردن نمونه های مشخص به عملکرد ETL: برای مثال روش مقایسه کردن که توسط شاخه های صحیح / غلط مقایسه می شوند جایی که یک شاخه گردش را بعنوان یک فیلتر ETL تشخیص داده و تغییر می دهد. (شکل ۲ ب)

به این موضوع توجه شود که شرحی بر BPMN<sup>۱</sup> و حاشیه نویسی بر آن در (شکل ۳) فراهم شده است.



(شکل ۲ الف) توسعه ماکروبرای اپراتور sk



(شکل ۲ ب) الگویی برای فیلترها

dataacademy.ir

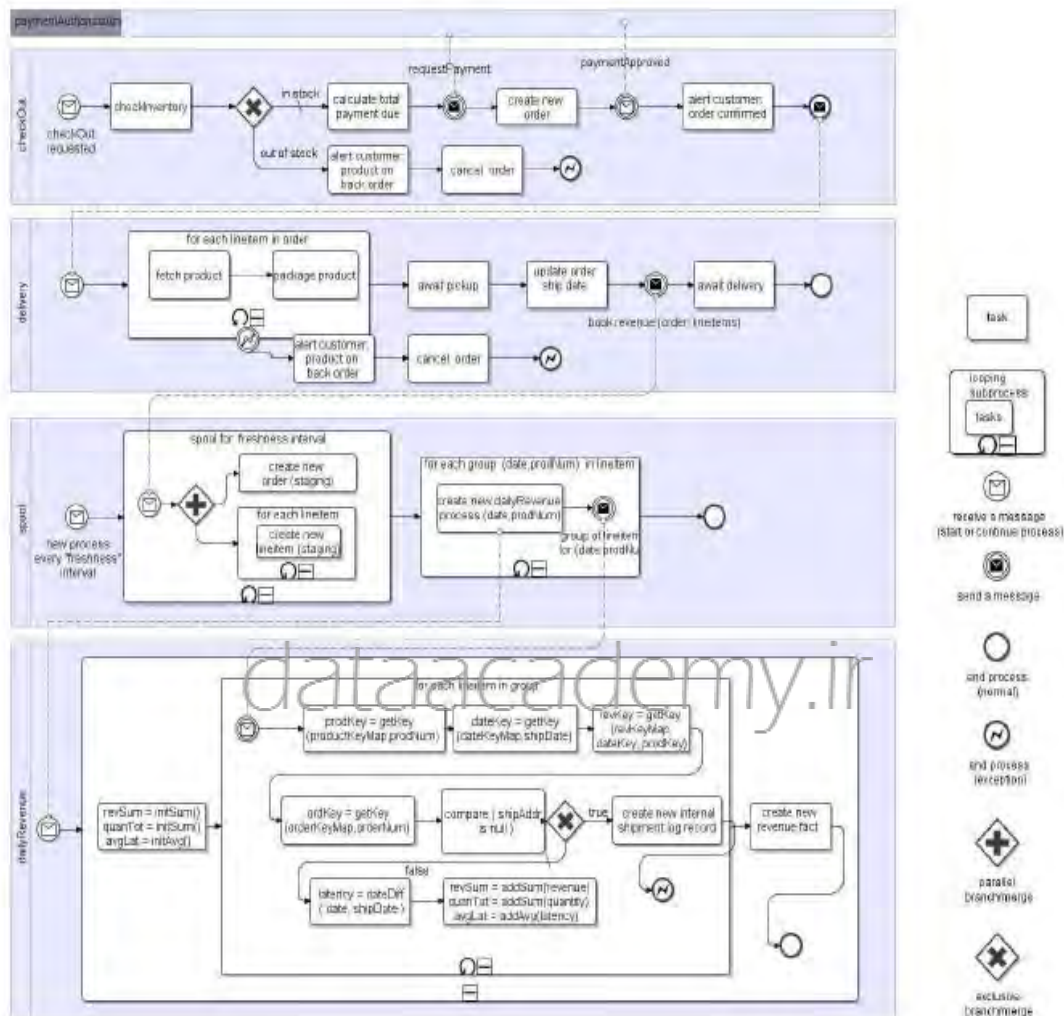
۲.۲- چالش دوم: گردش فرآیند مدل های BPMN است. اما ETL بصورت بنیادی یک گردش دیتا است.

بنابراین ما احتیاج داریم تا دیگرام های BPMN خودمان را تکمیل کنیم تا به اطلاعات گردش دیتای لازم برسیم ( مشخصاً ورودی ، خروجی ، و پارامترهای اپراتور های ETL ).

برای انجام این کار ، ما فرآیند BPMN واقعی خودمان را با ورودی ، خروجی و الگوهای پارامتری مثل این ها تکمیل می کنیم. ما فرض می کنیم که هر پیام BPMN توسط یک الگوی XML توصیف می شود. وقتی یک پیام توسط یک روش دریافت می شود این بعنوان قسمتی از الگوی ورودی گره محسوب می شود . ما از خاصیت حاشیه نویسی BPMN استفاده می کنیم تا روشها را با الگوی XML برای دیگر ورودیها، خروجی ها و هر پارامتر دیگر حاشیه نویسی کنیم. در محدوده گردش های فرآیند تا گردش های دیتا ، ما معمولاً باید به این نکته توجه کنیم که فرآیند تجارت نوعاً از گردش تک دیتا به فرآیندی که شاخه ای از ثبت شده هاست استفاده می کند. در حقیقت برای ساختن شاخه ای از شیء ها و هدفها ، مایک روش مدرن معرفی می کنیم که توالی اشیاء به عنوان ورودی می دهد و خروجی نوعی از همان شیء خواهد بود.

۲.۳- چالش سوم از هدف نهایی ما در معرفی اجرای بهینه فیزیکی ETL گرفته شده است. برای این منظور ، ما مجبور هستیم تا هدفهای QOX را با فرآیند BPMN واقعی یکی کنیم و گردش های منطقی ETL را بدست آوریم .

مثال: ما ایده های خودمان را با نشان دادن دیاگرام BPMN (دیاگرام BPMN) نمای طراحی شده اند و به صورت لازم و کامل با خصوصیات سازگار نیستند. برای فرآیند واقعی سود روزانه تشریح کردیم. (شکل ۳) را ببینید.

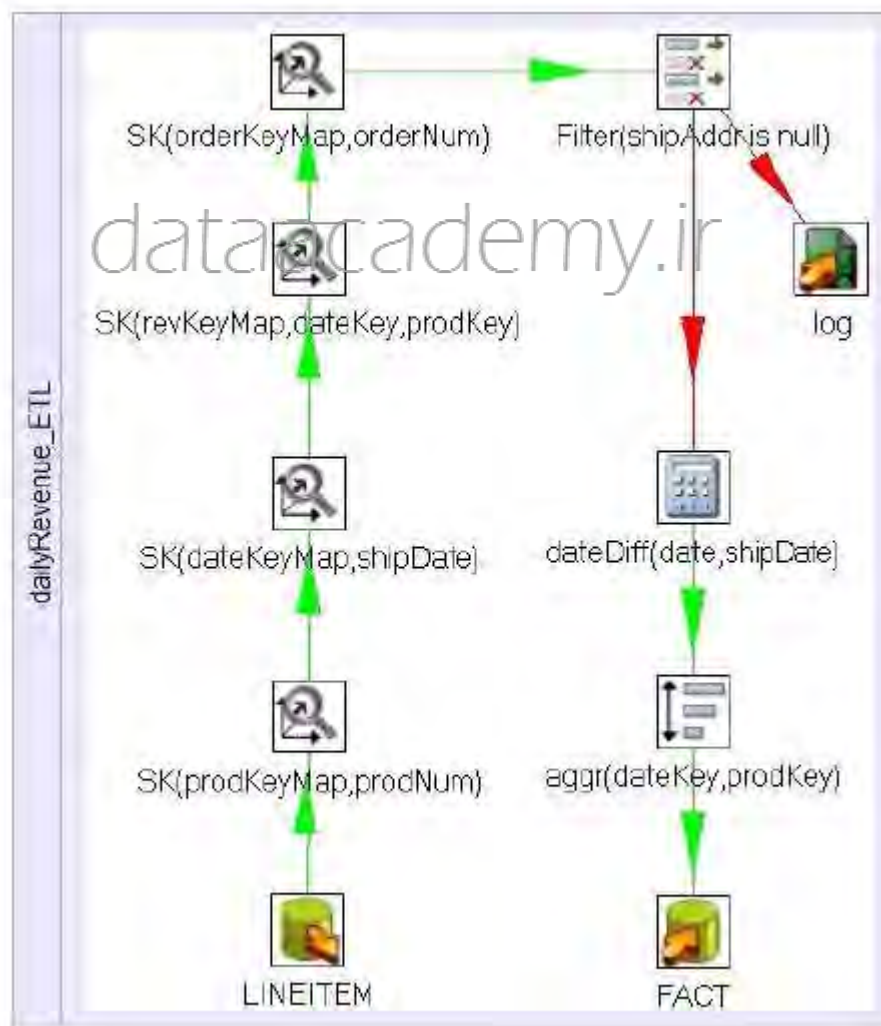


(شکل ۳) مثالی از فرآیند تجارت مؤثر (بازرسی، تحویل دادن) و فرآیند تجارت واقعی (سود روزانه، مدل قرقره وار)

این فرآیند سود هر محصولی که در روز فروخته می شود را محاسبه می کند. ما فرض می کنیم که احتیاجات تجارت شامل اندازه QOX برای بروز کردن باشد. این نشان می دهد که چگونه بارها، مخزن آپدیت می شود (برای مثال، روزانه برای تازه شدن سطح بالا، ماهانه برای تازه شدن سطح پایین و...) بنا براین فرآیند سود روزانه از فاصله یکبار به ازای فرش شدن آغاز می شود. یادآوری می کنیم که سود زمانی به حساب می آید که محصول فرستاده شده باشد. بنابراین تحقیق باید به فرآیند تجارت تحویلی اضافه شود. برای اینکه جزئیات درخواست را به فرآیند سود روزانه ارسال کند. متد مدرن (قرقره وار) جزئیات درخواست را از خلاصه درخواست و جزء اصلی موجود در خط جدا می کند و این دیتاها را برای دوره تازه شدن انباشته می کند. پس از این، گروهی از این آیتم های قرقره وار به خط شده به بخش تفکیک کردن فرستاده می شوند که گروه های آیتم های به صف شده بر حسب تاریخ و شماره محصول می

باشد. برای این چنین گروهی ، این متد یک فرآیند سود روزانه نمونه تهیه می کند و آن را برای آیتم های به صف شده ارسال می کند.

فرآیند سود روزانه کار ساختن عامل یا شیء جدید را انجام می دهد. این فرآیند توسط آیتمهایی به صف شده دوباره تکرار شده و جزئیات خود را جمع می کنند. روش `getKey` کلیدهای محصولات را به کلیدهای جانشین تبدیل می کند. توجه کنید که روش `getKey` یک توسعه ماکرو می باشد و دیاگرام BPMN متناظر آن در (شکل ۲ الف) نمایش داده شده است. درخواست های داخلی ، با آدرس های درخواست خالی مشخص می شود که بهتر است بعنوان سود محاسبه نشوند تا زمانیکه از صافی داخلی عبور نکرده اند. الگویی که استفاده می شود تا این شیوه را تشخیص دهد بعنوان یک فیلتر همانطور که در (شکل ۲ ب) نشان داده شده است استفاده می شود تا عمل کند. زمانی که تمام آیتمهای به صف شده در یک گروه پردازش شده مجموع ها به مخزن به عنوان عامل یا شیء جدید اضافه می شوند. گرفتن شرح فرآیند سود روزانه همزمان است با حاشیه نویسی گردش دیتا ، گردش منطقی ETL ، سود روزانه ETL که می تواند نسبتاً رابطه مستقیم با انتقال آنها داشته باشد . گردش منطقی ETL در زیر نمایش داده شده است .



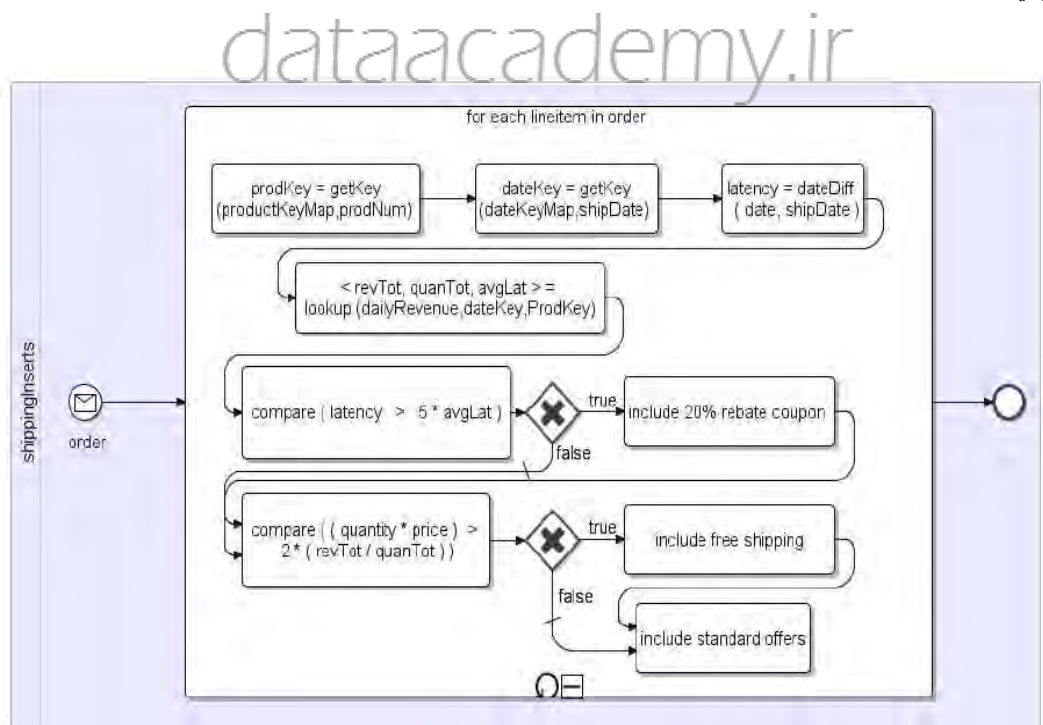
(شکل ۴ الف) گردش ETL منطقی



اینجا ما از نشانه گذاری ابزار منابع ETL استفاده کردیم . زبان گردش منطقی ETL به خودی خود یک چالش مهم است .

نمونه BI عملیاتی: همانطور که بحث شد ، فرآیند BPMN موقعیتهای جدید برای تصمیم گیریهای آنی فعال می کند بدون اینکه برای تازه شدن مخزن منتظر بمانیم. بعنوان مثال ، فرض کنید یک خرده فروش آنلاین بخواهد پیشنهادهای ویژه ای در پکیج ارسال شده قرار بدهد مثل تخفیف محصول ، ارسال رایگان یا تخفیف به محصولات جدید.و فرض کنید این پیشنهادهای به سود روزانه امروز بستگی داشته باشد. سود روزانه جاری در مخزن موجود نمی باشد.

□ برای انجام دادن این ، ما احتیاج داریم که روش اصلاح و پاک □ در فرآیند تحویل دادن به یک فرآیند جدید (شکل ۴ ب) متصل کنیم. این فرآیند جدید تعدادی از پیشنهادهای را برمی گرداند تا در بسته تحویلی طبق قوانین تجارت قرار دهد. در مثال ما فرض کردیم تخفیف برای درخواست ها پیشنهاد شده است که تحویل استثنائی و سریع و ارسال رایگان برای درخواست هایی فراهم شده است که بیش از دو برابر میزان میانگین درخواست ها است. ما احتیاج داریم مدت زمان رفرش شدن را تنظیم کنیم تا مطمئن شویم که سود روزانه در هر ساعت آپدیت می شود (یا حتی الامکان در تکرارهای زیاد) مجموع در حال کم شدن می تواند دنبال شود. توجه کنید که این احتیاج به یک تغییر اندک در فرآیند سود روزانه واقعی دارد برای اینکه مجموع در حال کم شدن را ثابت نگه دارد. بنابراین این فرآیند می تواند چندین گروه فرقه وار تشکیل دهد و فقط مخزن را یکبار در هر چرخه تازه کردن آپدیت کند.



(شکل ۴ ب) درخواست های آنی

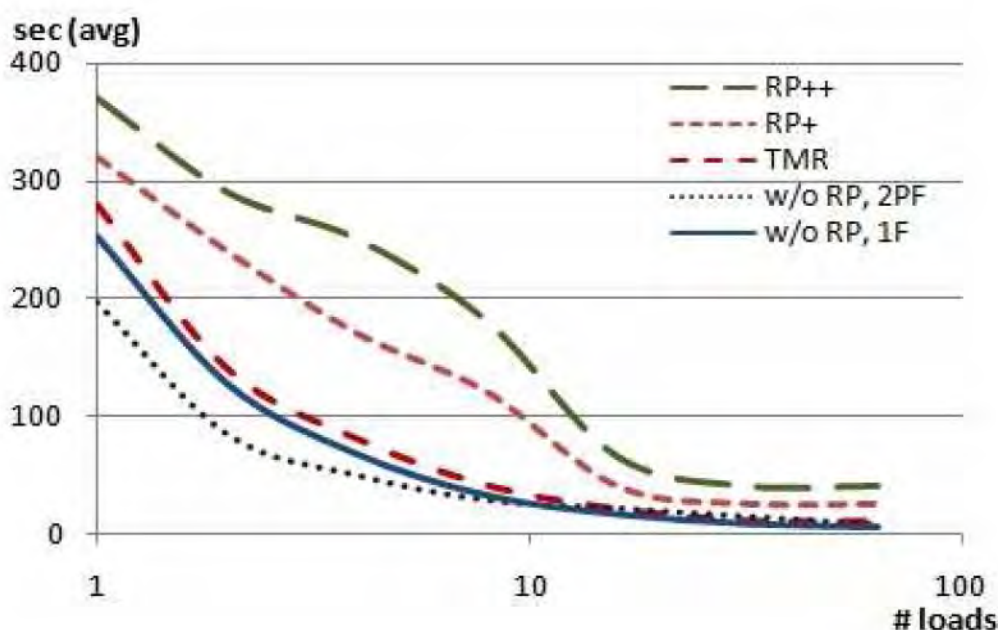
RetrieveAndPack ( )  
shipping inserts ( )

### ۳. بهینه سازی چرخه QOX .

بعد از گرفتن احتیاجات تجارت و فراهم کردن طراحی ETL منطقی مناسب ، گام بعدی در گیر کردن بهینه سازی طراحی ETL بر حسب معیارهای QOX است . چالش ها در انجام این ، شامل تعریف مدل های هزینه ها برای محاسبه معیارهای QOX ، تعریف فضای طراحی و الگوریتم ها برای جستجوی فضای طراحی تا طراحی مطلوب فراهم گردد. اینجا، ما بطور خلاصه تکنیکهای بهینه سازی و داد و ستد کالا را مطرح می کنیم.

بهینه سازی برای اجرا (که زمان اجرا را برای گردش ETL ارتقاء می دهد) نوعاً دوباره نویسی جبری را در پی دارد (برای مثال تقسیم کردن روش `getkey` برای `revkey` تا زمانی بعد از گردآوری برای کم کردن حجم دیتا ) یا بازسازی دیتا را در پی دارد. (مثل : کی و کجا تقسیم کردن انجام شود). دنباله کار ETL می تواند خراب شود بدلیل خطاهای سیستمی یا عملکردی . طراحی برای توانایی بازیابی دوباره ، نوعاً علاوه بر نقاط بازیابی در مکانهای مختلف در گیر می شود در دنباله کار که فرآیند ETL بعد از خطا ادامه می یابد و عملکرد خود را ادامه می دهد. اگر چه هزینه های I/O برای نگهداری نقاط بازیابی استفاده می شود و از اینرو داد و ستد های بین توانایی بازیابی دوباره و اجرا وجود دارد. بهینه ساز باید روی تعداد و مکان یابی نقاط بازیابی تصمیم گیری کند. بعضی وقتها ، ما نمی توانیم از نقاط بازیابی استفاده کنیم ، بعنوان مثال زمانی که تازه کردن در سطح بالا مورد نیاز است . در این گونه موارد ، بهترین شرایط ممکن این است که برای تلورانس خطا چیزی طراحی کنیم که بوسیله آن از یک نسخه از موارد مورد استفاده تهیه کنیم (که نشان می دهد پاسخ ، کم کردن خطا ، تنوع ). چالش های بسیار زیادی وجود دارد که قسمتی از دنباله کار است تا تکرار کند و به یک بالانس بین استفاده از نقطه بازیابی و نسخه برداری برسد. توانایی تازه کردن یک احتیاج حیاتی است برای BI عملیاتی و طراحی برای توانایی آپدیت یک ناحیه مهم از تحقیقات است. تکنیک جایگزین اینجا شامل استفاده از جزئیتهای موازی ، جلوگیری از مسدود کردن عملگرها و نقاط بازیابی ، گردش اجرایی از اپراتورهای مدل مثل متصل کننده ها ( ۸ ) یا فاز لود کننده ( ۹ ) می باشد. معمولاً زمانبندی گردش ETL و اجرای درخواست تغییر ساختارها فاکتور حیاتی می باشد ( ۱۰ ). بهینه سازی برای اینگونه معیارهای QOX به خودی خود چالش برانگیز است به دلیل فضای زیاد طراحی . اگر چه ، چالش اصلی بیشتر به این توجه می کند که جایگزین های اجرا کنار هم قرار می گیرد اینکه در مقابل ترکیب هدفهای QOX بهینه سازی شوند که بیشتر به خاطر احتیاجات تجارت مشخص می شوند.

(شکل ۵) بعضی از داد و ستدها را در بهینه سازی برای تازه کردن ، اجرا، امکان بازیابی دوباره ، و تلورانس خطا برای گردش های خاص نشان می دهد. خط آبی ثابت خط معیار اجرای گردش اصلی را نشان می دهد. برای افزایش توانایی تازه کردن (که آن کم کردن تأخیر لحظه آپدیت در سایتهای بزرگ با محور Y است) ما به افزایش تعداد بارها احتیاج داریم (محور X) . در انجام دادن اینچنینی ، بهترین اجرا با کمترین تأخیر ممکن است با موازی سازی حاصل شود (خط سیاه نقط چین). استفاده از نقطه بازیابی، به تازه کردن ، به میزان کم یا زیاد آسیب می زند که بستگی به این دارد که ما استفاده تعداد بالا (خط سبز با خط فاصله بزرگ) یا تعداد پایین (خط قرمز با خط فاصله کوچک) از نقاط بازیابی به ترتیب داشته باشیم. روش جایگزینی برای ۳ نسخه بالا (خط قرمز با فاصله بزرگ) نسخه پیمانه ای است که برای رسیدن تلورانس خطا یک سطح یکسان نزدیک از تازه کردن بعنوان طراحی اصلی استفاده می شود.



(شکل ۵)

#### ۴- نتیجه گیری

ما یک روش شناسی لایه لایه برای طراحی فرآیند ETL را توصیف کردیم که در سیستم های هوشمند تجارت مؤثر می باشد. فاکتور کلیدی این روش شناسی استفاده از شکل دهی متحد برای مدل کردن فرآیند تجاری عملکردی است که از سرمایه گذاری به خوبی فرآیند ها استفاده می کند تا یک نگاه پیوسته برای اطلاعات ارائه دهد. (مثل عامل های تجارت) که توسط تصمیم گیریهای عملکردی احتیاج است. این روش شناسی با یک خصوصیات مفهومی شروع می شود که یک تعریف منطقی دارد و اجرای فیزیکی آن بصورت معین نتیجه می شود. در مدل مفهومی تشخیص هدفهای QOX وجود دارد که طراحی را سبب شده و بهینه سازی در سطح های منطقی و فیزیکی را شامل می شود. تحقیقات مداوم ما مشکلات زیر را نشان می دهند:

۱) شکل دهی مدل کردن مفهومی: ما نشان دادیم محصولات ما از BPMN استفاده می کنند. اگر چه ، همانطور که ما بحث کردیم ، BPMN مخصوصاً به خوبی نمی تواند درخواست کند. تا طراحی نقشه گردش دیتا برای ساختن دیدهای اطلاعات را نشان دهد. بنابراین ، شکل دهی مدلهای باید حاشیه نویسی فرآیند و گردش دیتا با کیفیت هدف را حمایت کند.

۲) شکل دهی مدل کردن منطقی : این باید شامل اپراتورهای نوعی که توسط طراحی نقشه ها مورد نیاز است، باشد اما باید مجهز به هر نوع موتور اجرا باشد. و آن باید بتواند بهینه سازی چرخه QOX را فعال کند.

۳) نتیجه گیری اتوماتیک مدل منطقی از مدل مفهومی

۴) بهینه سازی گردش QOX : این شامل مدل هزینه برای نشان دادن معیارهای QOX است و نمایانگر الگوریتم هایی برای بهینه سازی در مقابل این معیارهاست.

۵) تکنیکهایی برای معتبر کردن طراحی در مقابل خصوصیات هر سطح تجارت

۶) تکنیک‌هایی برای متوجه شدن طراحی زمانی که احتیاجات سطح تجارت تغییر می کند.

## ۵- مراجع

- [1] W. Inmon, *Building the Data Warehouse*. John Wiley, 1993.
- [2] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. John Wiley, 2004.
- [3] C. White, "The Next Generation of Business Intelligence: Operational BI," *DM Review Magazine*, May 2005.
- [4] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson, "Data Integration Flows for Business Intelligence," in *EDBT*, 2009, pp. 1–11.
- [5] A. Simitsis, K. Wilkinson, M. Castellanos, and U. Dayal, "QoX-driven ETL Design: Reducing the Cost of ETL Consulting Engagements," in *SIGMOD Conference*, 2009, pp. 953–960.
- [6] D. Agrawal, "The Reality of Real-time Business Intelligence," in *BIRTE (Informal Proceedings)*, 2008.
- [7] P. Vassiliadis and A. Simitsis, *New Trends in Data Warehousing and Data Analysis*. Springer, 2008, ch. Near Real Time ETL, pp. 1–31.
- [8] N. Polyzotis, S. Skiadopoulos, P. Vassiliadis, A. Simitsis, and N.-E. Frantzell, "Supporting Streaming Updates in an Active Data Warehouse," in *ICDE*, 2007, pp. 476–485.
- [9] C. Thomsen, T. B. Pedersen, and W. Lehner, "RiTE: Providing On-Demand Data for Right-Time Data Warehousing," in *ICDE*, 2008, pp. 456–465.
- [10] L. Golab, T. Johnson, and V. Shkapenyuk, "Scheduling Updates in a Real-Time Stream Warehouse," in *ICDE*, 2009, pp. 1207–1210.

dataacademy.ir