**نشریه علمی- ترویجی محاسبات نرم**
**سال چهارم/ شمارهٔ هشتم/ پاییز و زمستان ۹۴/ صفحهٔ ۱۸ـ۳۱**

**دریافت مقاله: ۹۴/۵/۹**
**پذیرش مقاله: ۹۴/۷/۲۶**

# بهبود فرایند استخراج، تبدیل و بارگذاری در پایگاه داده تحلیلی با کمک پردازش موازی

حامد نجفی[۱]، نگین دانشپور[۲*]

[۱] دانشجوی کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران
h.najafi@srttu.edu
[۲] استادیار، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران
ndaneshpour@srttu.edu

**چکیده: پایگاه داده تحلیلی جهت نگهداری داده‌ها با ساختاری مناسب برای تحلیل بکار می‌رود. فرایند استخراج، تبدیل و بارگذاری عبارت است از تبدیل بعضی از داده‌های منبع به فرم مناسب، و انتقال آن‌ها به پایگاه داده تحلیلی که شامل سه مرحله کلی استخراج، تبدیل و بارگذاری داده است. در طی این مراحل، داده‌ها از یک یا چند پایگاه داده منبع به یک پایگاه داده تحلیلی منتقل می‌شوند. معمولاً ساختار منبع داده (مدل رابطه‌ای و نظایر آن) و پایگاه داده تحلیلی (شمای ستاره‌ای) مشخص است اما فرایند نگاشت داده‌ها دارای ساختار مشخصی نیست و متناسب با داده‌های موجود به روش‌های گوناگون و با ابزارهای مختلف پیاده‌سازی می‌شود. این فرایند از نظر هزینه و زمان، بخش عمده فرایند ساخت پایگاه داده تحلیلی را تشکیل می‌دهد. از این رو روش‌های متعددی جهت بهبود زمان اجرای این رویه ارائه شده است. در این مقاله سعی بر این شده که با استفاده از تکنیک‌های موازی سازی و طراحی بهینه برای مدل استخراج، تبدیل، بارگذاری از زمان اجرای این فرایند کاسته شود. درنتیجه این طراحی زمان اجرای پروسه حدود ۲۹٪ کاهش داشته است.**

**واژه‌های کلیدی: فرایند ETL، پایگاه داده تحلیلی، OLAP، داده‌های حجیم.**

## ۱. مقدمه

در عصر حاضر، شرکت‌ها و مؤسسات بزرگ با حجم عظیمی از داده‌های حرفهٔ خود روبه‌رو هستند که برای بهبود و ارتقای درآمد، نیاز به تحلیل و بررسی این داده‌هاست. پایگاه دادهٔ تحلیلی[1] (DW)، ساختار داده‌ای است که برای نگهداری اطلاعات حرفه و با توجه به نیاز تحلیل‌گران طراحی می‌شود. وظیفهٔ اصلی DW نگهداری داده‌ها به فرمی مناسب است تا پرس‌وجوهای تحلیلی روی آن‌ها در کمترین زمان ممکن اجرا شوند. فرایند استخراج، تبدیل و انتقال اطلاعات از پایگاه دادهٔ منبع به DW، به اختصار ETL[2] نامیده می‌شود و شامل سه بخش کلی خواندن داده‌ها از منبع، اعمال تغییرات روی آن‌ها و بارگذاری در DW است.

فرایند ETL از نظر هزینه و زمان طراحی و اجرا، بخش عمده‌ای از روند ساخت DW را شامل می‌شود. به‌طوری‌که حدود ۷۰ درصد ریسک و کار در ساخت DW مربوط به طراحی و اجرای فرایند ETL است [۱]. این فرایند شامل عملیات مختلفی روی داده‌هاست؛ مانند تبدیل شماها، پاک‌سازی یا تصفیهٔ داده‌ها (مثل حذف مقادیر تکراری)، فیلتر (فیلتر کردن یک سری مقادیر)، مرتب‌سازی، گروه‌بندی، ادغام، و توابع ازپیش‌تعریف‌شده. با توجه به این عملیات و حجم زیاد داده‌ها و پیچیدگی ساختاری فرایند ETL، کارهای متعددی در زمینهٔ بهبود طراحی و اجرای این فرایند انجام شده است [۲ـ۵]. در این مقاله سعی بر بهبود فرایند ETL از طریق موازی‌سازی عملیات شده است.

تاکنون کارهای مختلفی در زمینهٔ بهینه‌سازی فرایند ETL از نظر حافظهٔ مصرفی و زمان اجرا انجام گرفته است. روش‌های ارائه‌شده از تکنیک‌های پیاده‌سازی و الگوریتم‌هایی برای کاهش زمان اجرا استفاده کرده‌اند. بااین‌حال معمولاً تغییرات زمان اجرا در این روش‌ها جزئی بوده یا به منابع سخت‌افزاری بیشتری برای پیاده‌سازی نیاز دارند؛ بنابراین روشی نیاز است تا درصد قابل‌توجهی از زمان اجرا کاهش داده شود و همچنین با منابع سخت‌افزاری موجود قابل اجرا باشد. در این مقاله، با تکیه بر روش‌های موازی‌سازی و با استفاده از روش‌های مدل‌سازی فرایند ETL، سعی بر کاهش زمان اجرای این فرایند شده است. این روش به میزان چشمگیری زمان اجرای فرایند ETL را کاهش می‌دهد. در این مقاله، زمان اجرا تا ۲۹درصد کاهش داده شد و با توجه به توضیحات پیاده‌سازی می‌توان گفت این درصد قابل افزایش است.

در بخش ۲ مقاله، کارهای مرتبط در این زمینه شرح داده می‌شود. بخش ۳ شامل توضیحات روش‌های مدل‌سازی فرایند ETL است. در بخش ۴ روش‌های پیشنهادی ارائه شده و در قسمت ۵، پیاده‌سازی و ارزیابی این روش‌ها شرح داده می‌شود. نتیجهٔ این کار نیز در بخش ۶ آمده است.

## ۲. پیشینهٔ پژوهش

تاکنون کارهای مختلفی در زمینهٔ ETL انجام شده و هریک روی جنبه‌های مختلف طراحی و بهینه‌سازی فرایند ETL تمرکز داشته‌اند. به‌طور کلی، مطالعاتی در زمینهٔ ساخت چارچوب کاری برای فرایند ETL [۶ـ۱۰]، کاهش زمان اجرا [۲ و ۱۱]، کاهش حافظهٔ مصرفی [۲]، کاهش سخت‌افزار مورد نیاز [۴ و ۱۲] و پیاده‌سازی با تکیه بر تکنولوژی‌های نرم‌افزاری [۴، ۵ و ۱۲] انجام گرفته است. گذشته از روش‌های بهینه‌سازی، معیارهایی برای سنجش پیچیدگی ساختاری مدل ETL ارائه شده است [۱۳ و ۱۴]. در مقالهٔ [۱۵] روش‌های مدل‌سازی فرایند مورد بررسی قرار گرفته است. Ying-lan ابزارهایی برای طراحی و مدیریت فرایند ETL طراحی کرد [۱۶]. از دیگر مباحث در این زمینه می‌توان به فرایند ETL در پایگاه دادهٔ تحلیلی بلادرنگ اشاره کرد که YiChuan [۱۷] آن را مورد بررسی قرار داده است. در این مقاله سعی بر این است تا با استفاده از تکنیک‌های موازی‌سازی، زمان اجرای فرایند ETL کاهش داده شود. در ادامه به بررسی کارهای انجام‌شده در زمینهٔ بهینه‌سازی فرایند ETL پرداخته می‌شود.

تعدادی از مقالات، روی طراحی فرایند و سهولت استفاده از آن توسط کاربر تمرکز داشته‌اند. این کار از طریق ساخت چارچوب کاری و روش‌هایی برای طراحی آسان فرایند انجام شده است [۶، ۷، ۸، ۱۸، ۱۹، ۲۰ و ۲۱]. در مقاله‌های دیگری در

1. Data Warehouse
2. Extract - Transform - Load

زمینهٔ کاهش خطا و بهبود قابلیت اطمینان فرایند ETL کار شده است [۹، ۱۰، ۲۲ و ۲۳].

برخی مقالات با ارائهٔ روش‌هایی سعی در بهبود فرایند ETL از طریق کاهش زمان اجرا و حافظهٔ مصرفی داشته‌اند. Karagiannis [۲] در سال ۲۰۱۳ روشی ارائه داد که در آن، فعالیت‌هایی با بیشترین تعداد تاپل اولویت داشتند. در این روش، زمان اجرا تا حدودی کاهش یافت اما تمرکز اصلی روی حجم حافظهٔ مصرفی بوده است. Xavier [۱۱] با استفاده از توابع Powershell سعی در تسریع فرایند ETL نمود. روش پیشنهادی این مقاله در مقایسه با نرم‌افزارهای موجود، کاهش قابل‌توجهی در زمان اجرای فرایند ETL ایجاد کرد.

باید توجه داشت که شمای پایگاه دادهٔ منبع تأثیر زیادی روی زمان استخراج داده‌ها دارد، زیرا بخش عمدهٔ پردازش، مربوط به الحاق جداول است. هرچه مدل رابطه‌ای[1] پیچیده‌تر باشد، تعداد الحاق‌ها زیادتر است و درنتیجه، زمان استخراج بیشتر خواهد بود؛ بنابراین باید روش‌های مورد بررسی روی یک منبع داده با طراحی پیچیده اجرا شوند. به این ترتیب، تفاوت زمان‌های مورد نیاز برای انجام عمل الحاق و خواندن داده‌ها به‌صورت سطر به سطر بسیار واضح‌تر خواهد بود.

Santos [۴] با اجرای فرایند روی محیط توری سعی در استفادهٔ بهینه از منابع سخت‌افزاری موجود در سازمان‌ها نمود. این روش باعث تقسیم بار پردازشی و درنتیجه، کاهش زمان اجرا شد. اما این کاهش مربوط به بخش نگهداری و به‌روزرسانی DW است، درحالی‌که فرایند ETL اولیه برای ساخت DW بسیار سنگین‌تر از حالتی است که DW وجود دارد و تنها به‌روزرسانی می‌شود.

Prema [۵] از ترکیب XML و پایگاه داده تحلیلی اوراکل[2] برای تسریع فرایند بهره گرفته و روش Hyper-ETL را معرفی کرد. ضعف روش Hyper-ETL این است که اولاً تفاوت زمان اجرای آن با روش معمول ناچیز است؛ ثانیاً و مهم‌تر اینکه هرچند با افزایش تعداد رکوردها زمان اجرا افزایش می‌یابد، اما اختلاف زمانی دو روش ثابت است؛ یعنی برای تعداد رکورد زیاد که پردازش آن‌ها زمان زیادی نیاز دارد، این اختلاف زمانی قابل توجه نخواهد بود.

Sun [۱۲] با استفاده از سابروتین‌های[3] زبان Perl، امکان اجرای فرایند به‌صورت خط لوله[4] را فراهم آورده و انعطاف‌پذیری فرایند را بهبود بخشیده است. وی در این پیاده‌سازی زمان اجرا را تا حد مطلوبی کاهش داده است. بااین‌حال در این کار، بخش عمدهٔ پردازش در قسمت تبدیل است و راجع به پیچیدگی ساختاری منبع داده صحبتی نشده است. درنتیجه، با پیچیده‌تر شدن مدل رابطه‌ای و افزایش عملیات الحاق بار پردازشی افزایش قابل توجهی خواهد داشت.

در مجموع، کارهای مختلفی در زمینهٔ فرایند ETL انجام شده که در جدول (۱) به‌صورت خلاصه بیان شده است. در این جدول، کارهای انجام‌شده در سه دستهٔ کلی آمده است. هر گروه از کارها روی پارامترهای مشخصی از فرایند ETL تمرکز داشته‌اند و راه حل‌ها و نتایج به‌دست‌آمده در هر گروه، در دو ستون بیان شده است. تمرکز اصلی این مقاله، بهبود کارایی فرایند ETL از طریق کاهش زمان اجراست. ازاین‌رو با توجه به کارهای انجام‌شده در زمینهٔ بهبود کارایی همچنان نیاز به روشی است تا فرایند ETL را در زمان مناسب اجرا کند و از حداقل منابع سخت‌افزاری به‌صورت بهینه استفاده نماید. همچنین باید نشان داده شود که این روش در شرایطی که مدل رابطه‌ای گسترده بوده و تعداد جداول مورد نیاز بیشتر است، به‌خوبی عمل می‌کند.

## ۳. مدل مفهومی و مدل منطقی

فرایند ETL به‌صورت شماتیک به‌صورت‌های مختلفی قابل نمایش است. تاکنون برای تعریف و نمایش این فرایند، دو روش اصلی مدل‌سازی ارائه شده است که در این بخش به معرفی آن‌ها پرداخته می‌شود. همچنین تفاوت و ارتباط میان این دو روش بیان می‌شود.

1. Rational Model
2. Oracle
3. Subroutine
4. Pipeline

جدول (۱): بررسی کلی کارهای انجام شده

| زمینهٔ کاری | پارامترهای مهم | راه حل | نتایج |
|---|---|---|---|
| راه‌های طراحی فرایند و چارچوب کاری | طراحی آسان فرایند | – ساخت محیط گرافیکی طراحی فرایند ETL [۶]<br>– ارائهٔ روش‌های مدل‌سازی فرایند [۳ و ۲۴] | سهولت و خودکارسازی طراحی فرایند |
| بهبود قابلیت اطمینان فرایند ETL | کاهش خطا | – استفاده از بافرها [۱۷]<br>– مدیریت خطاهای داده‌ای [۲۲] | کاهش خطا و بهبود قابلیت اطمینان فرایند ETL و تصحیح داده‌های ناقص |
| بهبود کارایی ETL | زمان و حافظهٔ مصرفی | – الگوریتم‌های اولویت‌دهی به فعالیت‌ها [۲]<br>– توابع آماده [۱۲]<br>– سیستم‌های توزیع‌شده [۴] | کاهش حافظهٔ مصرفی در طول فرایند و کاهش زمان اجرا |

### ۱.۳. مدل مفهومی

برای طراحی فرایند تبدیل و انتقال داده‌ها نیاز است ابتدا یک طرح اولیه از فرایند رسم شود. مدل مفهومی[1] [۲۴]، نشان‌دهندهٔ بخش‌های مختلفی نظیر داده‌ها، تبدیل‌ها، و شماهاست. این مدل درواقع یک طرح کلی و اولیه از فرایند ETL است که نشان می‌دهد برای تبدیل و انتقال داده‌ها چه عملیاتی باید انجام پذیرد، و چه منابع داده‌ای در این فرایند نقش دارند. در مدل مفهومی هیچ ترتیب و اولویتی برای اجرای عملیات لحاظ نمی‌شود و تنها جریان داده‌ای از منبع داده به DW قابل مشاهده است.

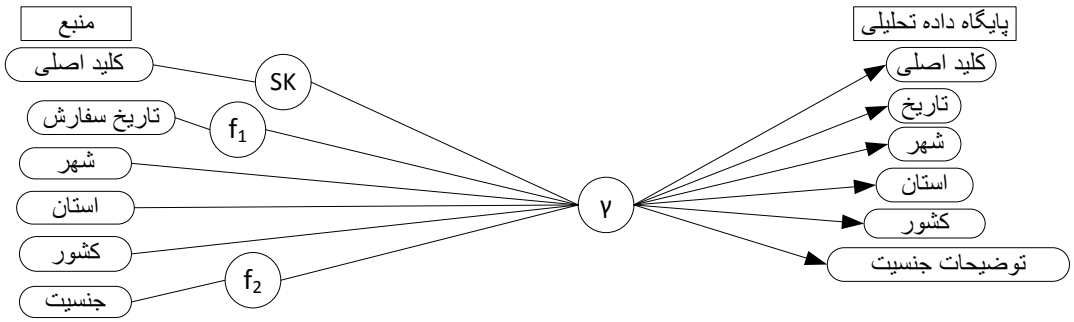
در این مدل، شماها نشان‌دهندهٔ منابع داده‌ای هستند که معمولاً به‌صورت جدول وجود دارند. تبدیل‌ها شامل انواع عملیات تکی[2] یا دوتایی[3] روی داده‌ها می‌شوند؛ مانند فیلتر، الحاق، مرتب‌سازی و کلید جایگزین. شکل (۱) نمونه‌ای از مدل مفهومی را نشان می‌دهد. همان‌طور که مشاهده می‌شود، جریان داده‌ای از منبع داده تا DW و عملیات مختلف روی داده‌ها در این مدل آمده است. سمت چپ نمودار، فیلدهای منبع داده را نشان می‌دهد که در ساخت DW مورد نیازند. در سمت راست نیز فیلدهای DW قرار دارند. یال‌ها که از منبع داده به سمت مقصد رسم شده‌اند، نشان‌دهندهٔ جریان داده‌ای هستند و دایره‌های روی این یال‌ها بیانگر تبدیل‌ها و دیگر عملیات داده‌ای هستند؛ برای مثال، تبدیل $f_1$ کار تبدیل فیلد تاریخ به یک قالب ازپیش‌تعیین‌شده را انجام می‌دهد.

### ۲.۳. مدل منطقی

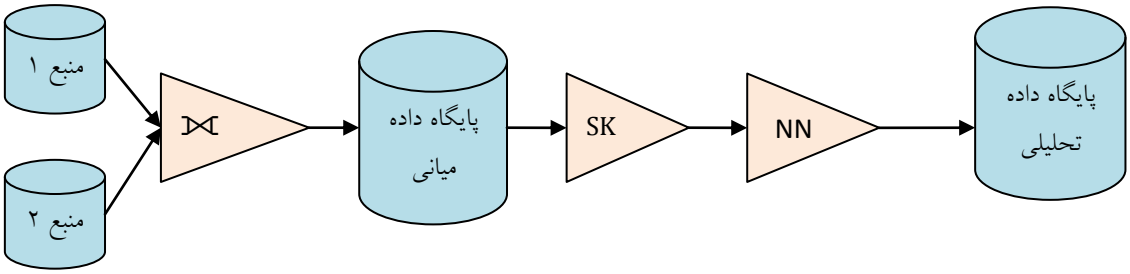
برای بهینه‌سازی اجرای فرایند ETL ابتدا باید روند اجرای عملیات بهینه شود. همان‌طور که گفته شد، در مدل مفهومی هیچ ترتیب اجرایی برای عملیات نشان داده نمی‌شود. بنابراین نیاز به مدلی است که نشان‌دهندهٔ اولویت‌ها و ترتیب اجرایی باشد.

مدل منطقی[4] [۶] یک جریان کاری مبتنی بر داده را نشان می‌دهد. این مدل از یک سری فعالیت‌ها[5]، مجموعه‌های داده[6] و توابع تشکیل می‌شود. مدل منطقی معمولاً به‌صورت یک گراف جهت‌دار نشان داده می‌شود که گره‌های آن منبع داده یا فعالیت هستند و جهت یال‌ها از منبع داده به سمت مقصد است. شکل (۲) نشان‌دهندهٔ یک نمونه از مدل منطقی است. همان‌طور که مشاهده می‌شود، این مدل یک ترتیب از اجرای فعالیت‌های ETL را شامل می‌شود و زمان‌بندی و اولویت اجرای آن‌ها را نیز مشخص می‌کند در سمت چپ این نمودار، منابع داده قرار دارند و مشابه مدل منطقی یال‌ها جهت جریان انتقال داده را نشان می‌دهند. مثلث‌ها بیانگر نوع تبدیل یا عملیات مورد نظر در هر مرحله‌اند.

1. Conceptual Model
2. Unary
3. Binary
4. Logical Model
5. Activity
6. Record set




**شکل (۱): نمونهٔ مدل مفهومی**




**شکل (۲): نمونهٔ مدل منطقی**

### ۳.۳. نگاشت مدل مفهومی به منطقی

همان‌طور که گفته شد، مدل مفهومی طرح اولیهٔ فرایند ETL است و سپس برای مشخص کردن ترتیب اجرای عملیات، به مدل منطقی طراحی رجوع می‌شود؛ بنابراین مدل مفهومی پس از طراحی به مدل منطقی نگاشت می‌شود تا زمان‌بندی شده و آمادهٔ اجرا شود. در این نگاشت، مفاهیم (یا همان شماهای داده‌ای) به مجموعه رکوردها و صفت‌ها به صفات متناظر در مدل منطقی نگاشت می‌شوند. همچنین تبدیل‌ها و فیلترها به فعالیت‌ها نگاشت می‌شود [۳].

## ۴. روش پیشنهادی

یکی از راه‌های پرکاربرد در افزایش سرعت اجرا و کارایی برنامه‌ها استفاده از پردازش موازی است [۲ و ۴]. در این مقاله سعی شده با استفاده از روش‌های مناسب پردازش موازی، زمان اجرای فرایند ETL تا حد مطلوبی کاهش داده شود. نکتهٔ اصلی در این زمینه، مدیریت پردازش‌ها برای استفادهٔ مناسب از منابع سخت‌افزاری، و همچنین مدیریت دسترسی به منابع برای جلوگیری از مشکلاتی مانند بن‌بست[1] یا گرسنگی[2] پردازش‌هاست. این موازی‌سازی در مدل مفهومی مشخص می‌شود، زیرا در این بخش، ترتیب اجرای فعالیت‌ها مد نظر است.

مدل مفهومی شامل سه بخش کلی است که عبارت‌اند از: بخش‌های منبع، میانی[3] و DW. بخش میانی برای نگهداری موقت داده‌ها در ساختاری مناسب استفاده می‌شود. ساختار پایگاه داده منبع و DW متفاوت است و داده‌ها باید از چندین جدول خوانده شده و تغییراتی روی آن‌ها اعمال شود؛ بنابراین به یک جدول میانی برای ذخیره و آماده‌سازی داده‌ها برای بارگذاری در DW نیاز است. با وجود این جدول، هنگام موازی‌سازی فرایند، احتمال ایجاد مشکل گرسنگی نیز افزایش می‌یابد که در این بخش بررسی خواهد شد. در این مقاله، موازی‌سازی فرایند ETL در سه حالت، طراحی و پیاده‌سازی شده است که در ادامه به شرح آن‌ها پرداخته می‌شود.

1. Deadlock
2. Starvation
3. Staging

**موازی‌سازی فرایند استخراج**: همان‌طور که گفته شد، داده‌ها از تعداد n (n≥1) پایگاه داده استخراج می‌شوند. روش پیشنهادی به این صورت است که برای انجام فرایند استخراج، به‌ازای هر منبع داده، یک پردازش در نظر گرفته شده و سپس فرایند استخراج به‌صورت موازی انجام می‌شود. ازآنجایی‌که دو یا چند پردازش قصد درج داده در جدول میانی را دارند، در این حالت، مشکل انحصار متقابل پیش می‌آید، زیرا داده‌ها سطر به سطر درج می‌شوند و زمانی که نخ[1] i در حال درج یک رکورد است، نخ j (i≠j) نمی‌تواند همزمان عمل درج را در همان جدول انجام دهد. برای رفع این مشکل در بخش میانی، به تعداد منابع داده، جدول ساخته می‌شود؛ یعنی به‌ازای n منبع داده به n جدول میانی نیاز است؛ بنابراین هر نخ داده‌های خوانده شده را در یک جدول جداگانه درج می‌کند و تداخلی برای دسترسی به منابع داده‌ای پیش نخواهد آمد.

پس از بارگذاری داده‌ها در بخش میانی، اطلاعات جداول، خوانده شده و در جدول حقیقت درج می‌شوند. در این مرحله خواندن داده‌ها از اولین جدول شروع شده و به‌صورت سطر به سطر انجام می‌شود. در واقع با اینکه داده‌ها در چند جدول ذخیره شده‌اند، هنگام خواندن تفاوتی با یک جدول ندارد. پس از خواندن هر رکورد از بخش میانی، اطلاعات متناظر آن که در جداول بُعد موجود است، جست‌وجو شده و با کلیدهای خارجی یافته شده، در جدول حقیقت درج می‌شود. شکل (۳) تصویر شماتیک روند اجرای عملیات را نمایش می‌دهد. همان‌طور که مشاهده می‌شود، ابتدا به‌ازای هر پایگاه داده منبع یک نخ ایجاد می‌شود که وظیفهٔ استخراج داده‌ها را بر عهده دارد. پس از اجرای موازی این نخ‌ها و بارگذاری کامل داده‌ها در بخش میانی، نوبت به بارگذاری اطلاعات در DW می‌رسد. بارگذاری داده‌ها به‌صورت ترتیبی و سطر به سطر توسط یک نخ انجام می‌شود.

**موازی‌سازی فرایند استخراج و بارگذاری**: در این روش، فرایند استخراج و بارگذاری به‌صورت هم‌زمان انجام می‌گیرند. به این ترتیب که نخ ۱ داده‌ها را استخراج نموده و در جدول میانی درج می‌کند. همزمان نخ ۲ داده‌های جدول میانی را خوانده و در DW بارگذاری می‌کند.

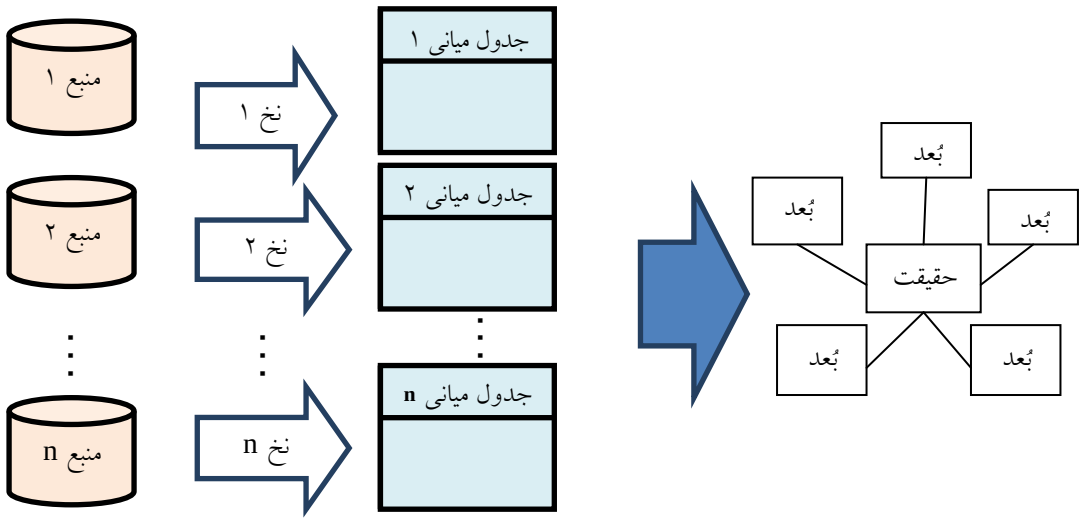
ازآنجایی‌که تنها یک پردازش، داده‌ها را در جدول میانی درج می‌کند، در این روش مشکل انحصار متقابل وجود ندارد؛ بنابراین وجود تنها یک جدول کفایت می‌کند. از طرفی نخ ۲ برای بارگذاری داده‌ها در DW نیاز به داده‌های استخراج شده توسط نخ ۱ دارد؛ بنابراین باید اطمینان حاصل کرد که نخ ۲ زمانی اجرا شود که داده‌های جدیدی برای آن در جدول میانی موجود باشد. با توجه به اینکه در بخش میانی تنها یک جدول وجود دارد با استفاده از یک متغیر شمارنده می‌توان اطمینان حاصل کرد که نخ ۲ زمانی اجرا شود که رکوردهای مورد نیاز آن توسط نخ ۱ در جدول میانی درج شده باشد. شکل (۴) نحوهٔ اجرای فرایند را نشان می‌دهد. نخ ۱ و ۲ مسئول استخراج داده‌ها از منبع و بارگذاری داده‌های جدول میانی در DW هستند. پس از اینکه نخ ۱ اولین سطر را استخراج و در جدول میانی درج نمود، نخ ۲ شروع به کار کرده و داده‌ها را در DW بارگذاری می‌کند. در بخش میانی، وجود یک جدول کافی است. رکوردهای منابع داده‌ای به‌ترتیب خوانده می‌شوند؛ یعنی ابتدا رکوردهای منبع اول استخراج شده و در جدول میانی درج می‌شوند. در ادامه نیز به همین ترتیب داده‌های منبع دوم خوانده می‌شوند.

**روش ترکیبی**: این روش به‌صورت ترکیبی از دو روش فوق عمل می‌کند. به این صورت که برای n منبع داده، n پردازش وظیفهٔ خواندن داده‌ها از منابع را دارند و یک پردازش بارگذاری داده‌ها از بخش میانی به DW را انجام می‌دهد و هر n+1 پردازش به‌صورت موازی انجام می‌گیرند. نخ ۱ تا n استخراج داده را از n منبع به‌صورت موازی انجام می‌دهند. پس از درج اولین رکوردها در جداول میانی، نخ n+1 شروع به بارگذاری داده‌ها در DW می‌کند.

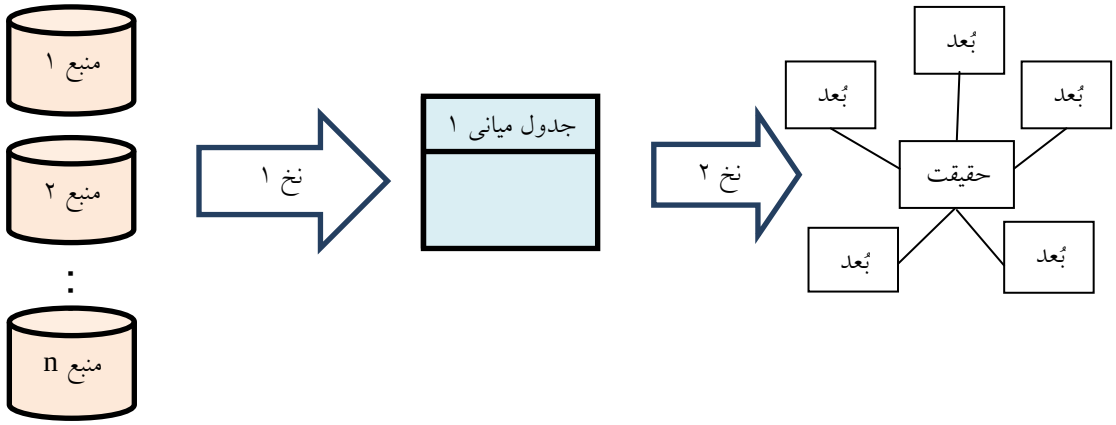
باید توجه داشت که در این روش، مشکل گرسنگی و انحصار متقابل وجود دارد؛ بنابراین مشابه روش اول به‌ازای هر جدول منبع یک جدول در بخش میانی ایجاد می‌شود. همچنین نخ n+1 باید تا استخراج رکورد جدید توسط یکی از نخ‌های ۱ تا

1. Thread

n صبر کند که این قسمت نیز توسط متغیرهای شمارنده مدیریت می‌شود. ازآنجایی‌که n جدول در لایهٔ میانی قرار دارد، نخ n+1 باید هر رکورد را از یک جدول بخواند و به‌صورت مداوم منبع خود را تغییر دهد؛ بنابراین شمارنده‌های جداگانه برای هر جدول در نظر گرفته می‌شود تا مشکل گرسنگی برای نخ n+1 به‌وجود نیاید. شکل (۵) نحوهٔ اجرای این روش را نمایش می‌دهد. نخ ۱ تا n وظیفهٔ استخراج و نخ n+1 وظیفهٔ بارگذاری را بر عهده دارند که به‌صورت موازی اجرا می‌شوند.




**شکل (۳): روش موازی‌سازی استخراج**




**شکل (۴): روش موازی‌سازی استخراج و بارگذاری**

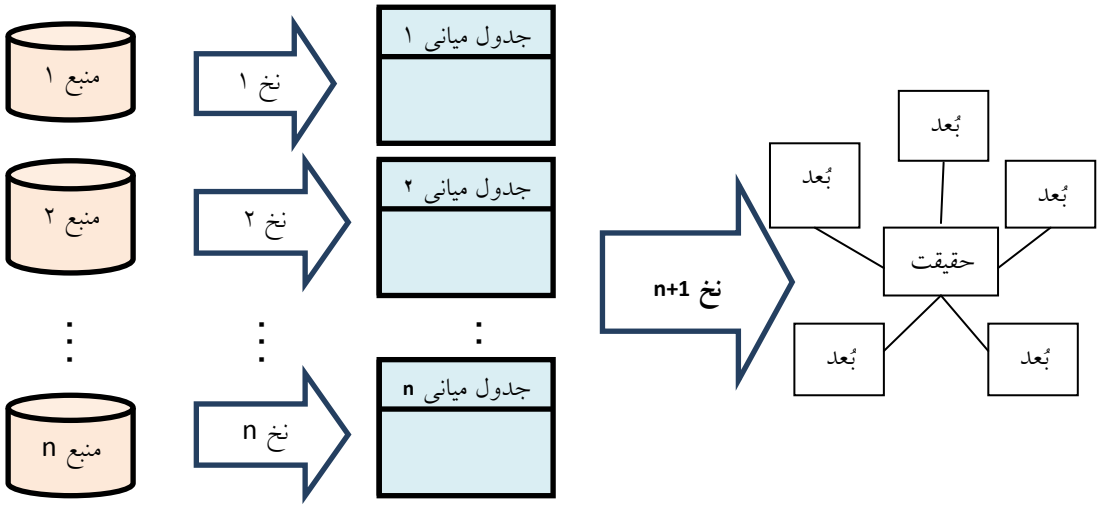
## ۵. پیاده سازی و ارزیابی

در مدل منطقی، ترتیب اجرای فعالیت‌ها مشخص می‌شود. اما قبل از آن به یک مدل مفهومی نیاز است تا مشخص شود در طول فرایند، چه عملیاتی باید اجرا شود. برای پیاده‌سازی روش‌های ارائه‌شده، پایگاه داده‌ای شامل داده‌های یک فروشگاه اینترنتی طراحی شده و با داده‌های مناسب پر شده است.

### ۱.۵. طرح منبع داده و DW

در این کار ابتدا یک پایگاه داده به‌عنوان منبع ایجاد شده و شمای ستاره‌ای برای DW متناظر با آن طراحی شد. مدل ER و شمای ستاره‌ای در شکل (۶) و (۷) آمده است. پایگاه داده منبع برای یک فروشگاه اینترنتی طراحی شده که شامل حجم قابل توجهی از داده است. شمای ستاره‌ای که ساختار داده‌ها

در DW را نشان می‌دهد با توجه به پارامترهای تحلیلی (معیار و ابعاد) طراحی شده است. حال با داشتن مدل رابطه‌ای منبع و شمای ستاره‌ای پایگاه داده تحلیلی، فرایند ETL طراحی می‌شود. برای طراحی مدل مفهومی ابتدا شماهای داده‌ای و عملیات داده‌ای مورد نیاز مشخص می‌شود. در این کار داده‌ها از دو منبع خوانده شده و پس از انجام عملیات لازم، به DW منتقل می‌شوند.




**شکل (۵): روش ترکیبی**

پس از طراحی شمای ستاره‌ای و تعیین معیار سنجش و ابعاد، مشخص می‌شود چه صفاتی برای ساختن DW نیاز هستند. همان‌طور که در شکل (۷) مشاهده می‌شود، شمای ستاره‌ای دارای یک معیار سنجش (مجموع فروش) و ۵ بُعد است. حال باید مشخص شود که صفات موجود در پایگاه داده منبع چگونه به ابعاد در DW نگاشت می‌شوند و همچنین مقدار صفت معیار چگونه محاسبه می‌شود.

به این منظور مدل مفهومی برای فرایند طراحی می‌شود تا تبدیل‌ها و عملیات لازم برای انتقال اطلاعات از منبع داده به DW مشخص شوند.

### ۲.۵. مدل مفهومی

با توجه به ابعاد در شمای ستاره‌ای (مانند زمان و مکان) مشخص می‌شود که چه صفاتی از پایگاه داده منبع نیاز است و کدام جداول باید به هم ملحق شوند تا مقدار معیار برای آن‌ها به‌دست آید. الحاق جداول کاری بس زمان‌بر است؛ بنابراین برای تسریع اجرای فرایند، باید روی مدیریت این بخش تمرکز کرد.

مدل مفهومی فرایند ETL در شکل (۸) آمده است.

همان‌طور که در شکل مشاهده می‌شود، داده‌ها پس از انجام یک سری تغییرات در DW بارگذاری می‌شوند. وظیفهٔ عمل SK ایجاد یک کلید جایگزین برای بارگذاری داده‌ها در جدول مقصد است. عبارات f شامل یک سری تغییرات روی داده‌ها می‌باشند؛ برای مثال، جنسیت در دو پایگاه داده به دو صورت مختلف ذخیره شده است (صفر و یک ـ مرد و زن) و وظیفهٔ $f_2$ تبدیل آن‌ها به یک فرم است. وظیفهٔ فعالیت γ جمع‌آوری داده‌ها و محاسبهٔ مجموع مقدار پرداخت برای چند رکورد است.

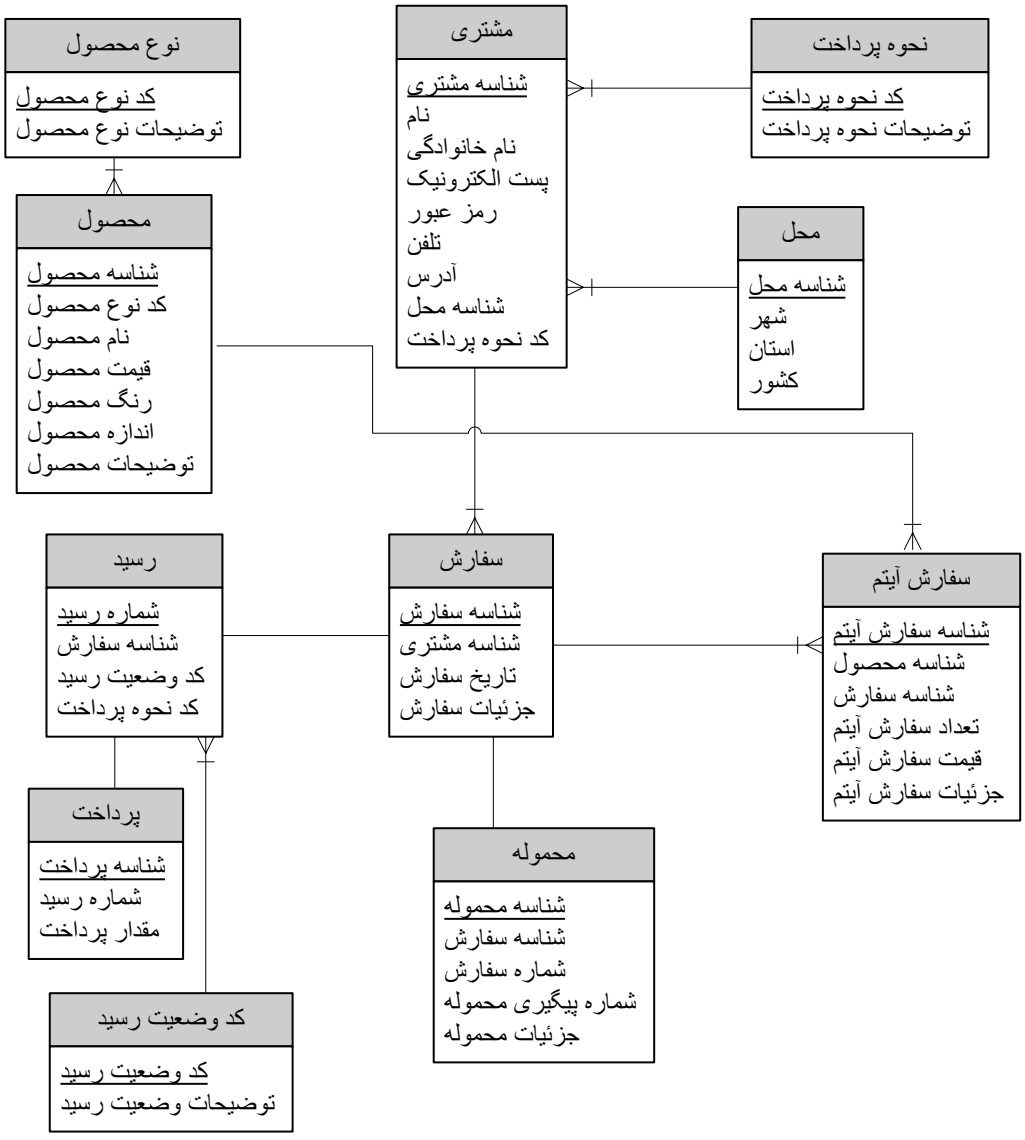
شکل (۸) یک تصویر کلی از فرایند ETL را در اختیار طراح می‌گذارد. پس از این، نوبت به طراحی مدل منطقی می‌رسد که تمرکز اصلی آن روی چند و چون اجرای عملیات و پردازش‌ها روی داده‌هاست.

### ۳.۵. مدل منطقی

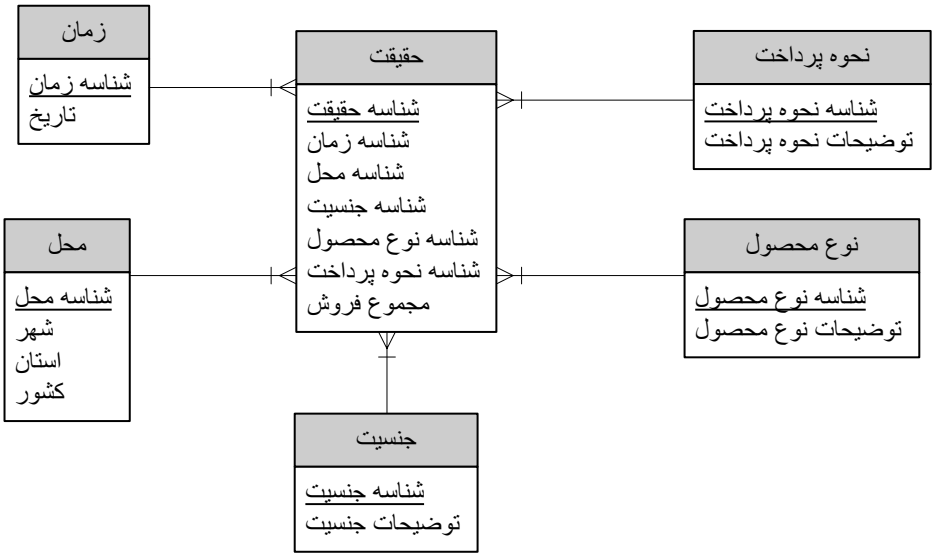
در این مقاله، مدل منطقی به‌طور کلی به سه بخش منبع، میانی و مقصد تقسیم شده است. بخش منبع و مقصد همان پایگاه دادهٔ استخراج‌شده نیاز است تا بعد از آن به DW منتقل شوند. این سه بخش در شکل (۹) نشان داده شده است. در سمت

چپ شکل جداول منبع داده قرار دارند، سمت راست شمای ستاره‌ای یا همان DW و قسمت میانی شامل جدول میانی می‌شود. بخش میانی حاوی یک جدول است که تمامی اطلاعات استخراج‌شده سطر به سطر در آن ذخیره می‌شود. در واقع می‌توان گفت این جدول همان DW است که ساختار دادهٔ آن (مدل رابطه‌ای) نرمال‌سازی نشده است.

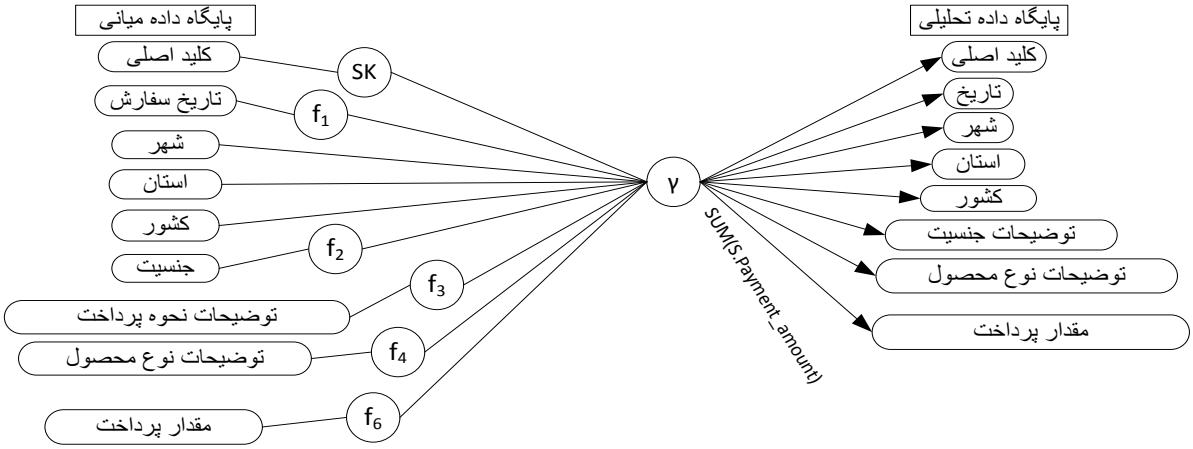
در طی فرایند بارگذاری، هر رکورد از جدول بخش میانی خوانده شده و با توجه به اطلاعاتش که در جداول بُعد قسمت DW موجودند، با کلیدهای خارجی مناسب در جدول حقیقت درج می‌شود. برای بارگذاری اطلاعات در بخش میانی، هنگام خواندن اطلاعات از منبع داده، عمل الحاق روی چند جدول انجام می‌گیرد. به این صورت که برای هر جنس فروخته‌شده که در جدول سفارش آیتم ثبت شده است، سایر اطلاعات آن (زمان خرید، و دسته‌بندی محصول) از الحاق این جدول با جداول دیگر به‌دست می‌آید و به‌صورت یک رکورد در جدول مرحلهٔ میانی درج می‌شود.



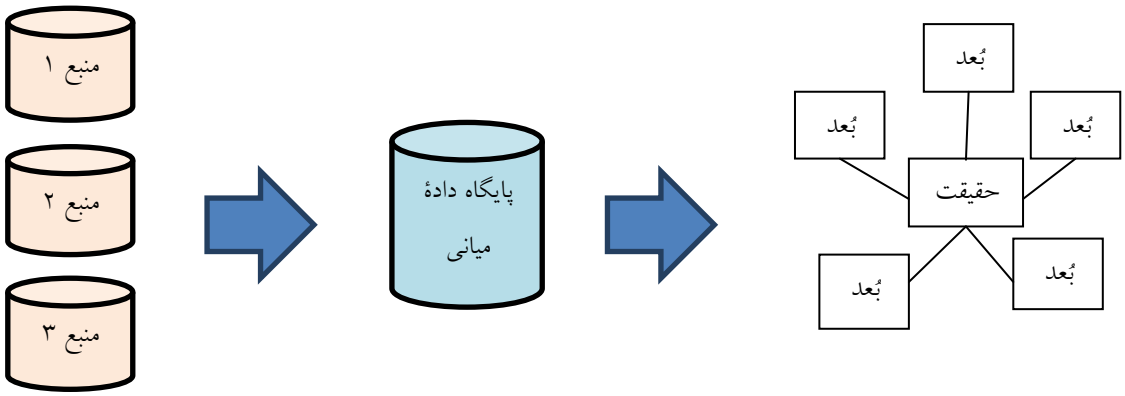

**شکل (۶): مدل ER برای پایگاه دادهٔ منبع**




**شکل (۷): شمای ستاره‌ای DW**




**شکل (۸): مدل مفهومی**




**شکل (۹): مدل کلی فرایند ETL**

برای بارگذاری داده‌ها از مرحلهٔ میانی به DW باید کلیدهای خارجی برای صفت‌های هر رکورد پیدا شوند؛ به این ترتیب که ابتدا یک رکورد از جدول مرحلهٔ میانی خوانده می‌شود، سپس با توجه به مقادیر صفت‌ها، رکوردهای متناظر با آن‌ها در جداول بُعد پیدا می‌شود و مقادیر کلید اصلی آن‌ها انتخاب می‌شود و مقدار صفت معیار به همراه کلیدهای خارجی به‌دست‌آمده در جدول حقیقت درج می‌شود.

برای پیاده‌سازی سه روش موازی‌سازی، از زبان C# استفاده شده است. پایگاه‌های داده در محیط SQL Server 2012 طراحی و داده‌های منبع با استفاده از نرم‌افزار Redgate Data Generator تولید شده است. پیاده‌سازی در محیط Microsoft Visual Studio 2012 انجام شده و پیکربندی سیستم شامل پردازندهٔ Core i5-4200M 2.50GHz، 8GB RAM DDR3، Windows 7 64-bit است.

دو پایگاه داده به‌عنوان منبع طراحی شده که از نظر ساختاری مشابه‌اند (شکل ۶) ولی نحوهٔ ذخیرهٔ داده در بعضی صفات آن‌ها متفاوت است. شایان ذکر است منابع داده از نظر فیزیکی روی یک سیستم قرار دارند، زیرا در این روش، مکان فیزیکی داده‌ها تأثیر چندانی ندارد و هزینه و زمان انتقال داده‌ها مد نظر نیست. درصورتی‌که در کاربردهای تجاری منابع داده از نظر جغرافیایی پراکنده‌اند. پایگاه دادهٔ میانی حاوی دو جدول بوده و پایگاه دادهٔ مقصد به‌صورت شمای ستاره‌ای (شکل ۴) طراحی شده که به‌عنوان DW مورد استفاده قرار می‌گیرد.

برای پیاده‌سازی روش اول (استخراج موازی) دو پایگاه داده منبع و دو جدول بخش میانی لازم‌اند. ابتدا داده‌های دو منبع به‌صورت موازی استخراج می‌شود. پس از بارگذاری تمامی داده‌ها در جداول میانی، رکوردها در جدول حقیقت درج می‌شوند. نخ ۱ و ۲ به‌صورت موازی اجرا می‌شوند.

در روش دوم (استخراج و بارگذاری موازی) یک پردازش مسئول خواندن داده از منبع است و پردازش دیگر مسئول بارگذاری داده از جدول میانی به DW است. هرچند دو منبع داده وجود دارد، داده‌ها به‌صورت ترتیبی خوانده می‌شوند گویی داده‌ها از یک منبع خوانده می‌شوند. همچنین در بخش میانی، یک جدول کفایت می‌کند. روند اجرا به این‌صورت است که نخ ۱ استخراج را انجام داده و همزمان نخ ۲ عمل بارگذاری را انجام می‌دهد. روش ترکیبی نیز به‌صورت ترکیب دو روش قبل عمل می‌کند. به این ترتیب که دو نخ ۱ و ۲ عمل استخراج را انجام داده و نخ ۳ داده‌ها را در DW بارگذاری می‌کند.

منابع دادهٔ هریک شامل ۵۰۰ هزار رکورد اصلی (تعداد رکوردهای جدول سفارش) هستند. بنابراین فرایند ETL باید برای یک میلیون رکورد انجام شود. برای مقایسهٔ روش معمول یعنی اجرای ترتیبی، تمامی مراحل ETL نیز پیاده‌سازی و با سه روش ارائه‌شده مقایسه شده است. این آزمایش با تعداد رکورد متفاوت نیز انجام گرفته تا کارایی هر روش به تناسب حجم داده نیز مشخص شود. همان‌طور که گفته شد، معیار اصلی در سنجش کارایی ETL، مدت زمان اجرای آن است. بنابراین زمان اجرا در حالت‌های مختلف اندازه‌گیری و مقایسه خواهد شد. نمودار (۱) زمان اجرای فرایند با چهار روش را نشان می‌دهد. هر روش با تعداد رکوردهای مختلف (۱۰ هزار، ۲۰۰ هزار و ۱ میلیون رکورد) پیاده‌سازی شده است.

همان‌طور که از نمودار مشاهده می‌شود، زمان اجرا در روش استخراج و بارگذاری موازی و روش ترکیبی، تفاوت چشمگیری (حدود ۲۹درصد) با روش ترتیبی و استخراج موازی دارد. از طرفی مشاهده می‌شود که روش استخراج موازی و ترتیبی از نظر زمانِ اجرا تفاوت چندانی با یکدیگر ندارند و این امر برای دو روشِ دیگر نیز صدق می‌کند؛ بنابراین می‌توان گفت استخراج موازی نسبتاً تأثیر ناچیزی در زمان اجرای فرایند دارد. علت این امر این است که هرچند مرحلهٔ استخراج سریع انجام شود، باز هم کل داده‌ها در مرحله بارگذاری در DW باید سطر به سطر منتقل شوند. در واقع مرحلهٔ بارگذاری در این مدل، گلوگاه فرایند به حساب می‌آید.

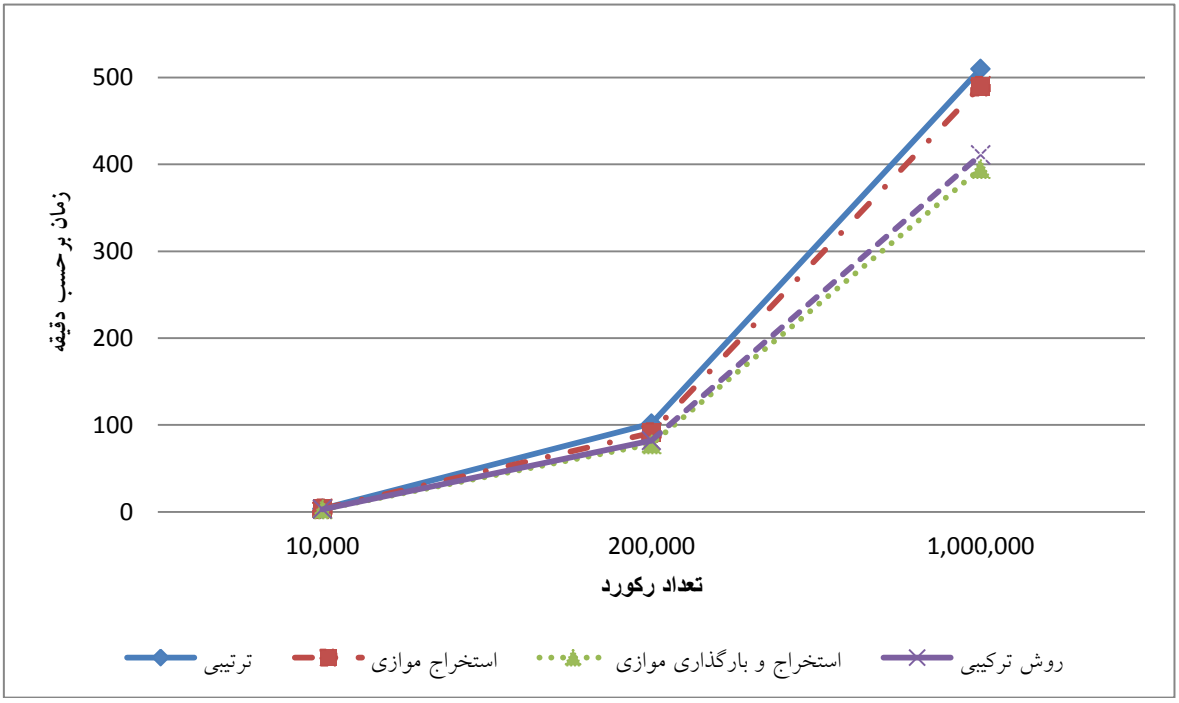
نکتهٔ دیگر که در نمودار مشهود است، اینکه با افزایش تعداد رکورد تفاوت زمان اجرا نیز افزایش می‌یابد. در پایگاه‌های داده تجاری تعداد رکوردها به ده‌ها میلیون می‌رسد. درنتیجه، تفاوت زمان اجرا به نسبت بیشتر خواهد بود و استفاده از روش

پردازش موازی باعث ایجاد کاهش چشمگیری در زمان اجرا می‌شود.

## ۶. نتیجه‌گیری

در روند تولید DW بیشترین بار پردازشی در فرایند ETL اعمال می‌شود. این فرایند به‌دلیل زمان اجرای زیاد حائز اهمیت بوده و برای بهینه‌سازی روند ساخت DW نیاز به کاهش زمان اجرای فرایند است. در این مقاله با استفاده از روش‌های مدل‌سازی فرایند سعی بر بهینه‌سازی زمان اجرای آن شده است. روش‌های ارائه‌شده بر مبنای پردازش موازی هستند که در سه حالت پیاده‌سازی و اجرا شده است.

پس از اجرای فرایند ETL از طریق روش‌های پیشنهادی مشاهده می‌شود که روش موازی‌سازی استخراج و بارگذاری، از نظر زمان اجرا بهتر از دیگر روش‌ها عمل می‌کند. این روش نسبت به روش ترتیبی که به‌طور معمول استفاده می‌شود، کاهش چشمگیری در زمان اجرا ایجاد می‌کند که برای یک منبع داده‌ای با یک میلیون رکورد زمان اجرا را حدود ۲۹درصد کاهش داده است. همچنین ازآنجایی‌که میزان اختلاف زمان اجرا به‌ازای تعداد رکورد به‌صورت خطی افزایش نمی‌یابد، می‌توان نتیجه گرفت به‌ازای تعداد رکورد بیشتر اختلاف زمانی افزایش یابد.




**نمودار (۱): نتیجهٔ اجرای ۴ روش روی تعداد دادهٔ مختلف**

تاکنون روش جامع و کاملی برای نگاشت مدل مفهومی به منطقی ارائه نشده است و معمولاً بخش عمده‌ای از این کار به‌صورت دستی و با کمک کاربر انجام می‌شود. جدای از چگونگی این نگاشت، نحوهٔ اجرای فرایند نیز از اهمیت زیادی برخوردار است. برای ادامهٔ این مبحث در آینده می‌توان روی تسهیل این نگاشت تمرکز کرد تا بخش بیشتری از این روند به‌صورت خودکار و بدون نیاز به دخالت کاربر انجام شود.

## مراجع


[1] Berkani, N., Bellatreche, L., Khouri, S., "*Towards a conceptualization of ETL and physical storage of semantic data warehouses as a service*", cluster computing, Vol. 16, No. 4, pp. 915-931, 2013.

[2] Karagiannis, A., Vassiliadis, P., Simitsis, A., "*Scheduling strategies for efficient ETL execution*", Information Systems, Vol. 38, No. 6, pp. 927–945, 2013.

[3] Simitsis, A., Vassiliadis, P., "*A method for the mapping of conceptual designs to logical blueprints for ETL processes*", Decision Support Systems, Vol. 45, No. 1, pp. 22–40, 2008.

[4] Santosa, V., Oliveiraa, B., Silva, R., Belo, O., "*Configuring and executing etl tasks on grid environments - requirements and specificities*", Procedia Technology, Vol. 1, No. 1, pp. 112–117, 2012.

[5] Prema, A., Pethalakshmi, A., "*Novel approach in ETL*", International Conference on Pattern Recognition, Informatics and Mobile Engineering, pp. 429 - 434, 2013.

[6] Vassiliadisa, P., Simitsis, A., Georgantasb, P., Terrovitisb, M., Skiadopoulos, S., "*A generic and customizable framework for the design of ETL scenarios*", Information Systems, Vol. 30, No. 7, pp. 492–525, 2005.

[7] Awad, M., Abdullah, M. S., "*A Framework for Interoperable Distributed ETL Components Based on SOA*", 2nd International Conference on Software Technology and Engineering, pp. 67-70, 2010.

[8] Awad, M., Abdullah, M. S., Mat Ali, A. B., "*Extending ETL framework using service oriented architecture*", Procedia Computer Science, Vol. 3, No. 1, pp. 110–114, 2011.

[9] Li, L., "*A Framework Study of ETL Processes Optimization Based on Metadata Repository*", 2nd International Conference on Computer Engineering and Technology, pp. 125-129, 2010.

[10] Wang, H., Ye, Z., "*An ETL Services Framework Based on Metadata*", 2nd International Workshop on Intelligent Systems and Applications, pp. 1-4, 2010.

[11] Xavier, C., Moreira, F., "*Agile ETL*", Procedia Technology, Vol. 9, No. 1, pp. 381–387, 2013.

[12] Sun, K., Lan, Y., "*SETL: A Scalable And High Performance ETL System*", 3rd International Conference on System Science, Engineering Design, and Manufacturing Information, pp. 6 - 9, 2012.

[13] Muñoz, L., Mazón, J., Trujillo, J., "*A family of experiments to validate measures for UML activity diagrams of ETL processes in data warehouses*", Information and Software Technology, Vol. 52, No. 11, pp. 1188–1203, 2010.

[14] Muñoz, L., Mazón, J., Trujillo, J., "*Measures for ETL Processes Models in Data Warehouses*", first international workshop on Model driven service engineering and data quality and security, pp. 33-36, 2009.

[15] Ali El-Sappagh, Shaker H., Ahmed, A. M., El Bastawissy, A. H., "*A proposed model for data warehouse ETL processes*", Journal of King Saud University - Computer and Information Sciences, Vol. 23, No. 2, pp. 91–104, 2011.

[16] Ying-lan, F., Bing, H., "*Design and Implementation of ETL Management Tool*", Second International Symposium on Knowledge Acquisition and Modeling, pp. 446 - 449, 2009.

[17] YiChuan, Sh., Yao, X., "*Research of Real-time Data Warehouse Storage Strategy Based on Multi-level Caches*", Physics Procedia, Vol. 25, No. 1, pp. 2315–2321, 2012.

[18] Du, N., Ye, X., Wang, X., "*A schema aware ETL workflow generator*", Information Systems Frontiers, Vol. 16, No. 3, pp. 453-471, 2014.

[19] Kabiri, A., Chiadmi, D., "*A method for modelling and organazing ETL processes*", Second International Conference on Innovative Computing Technology, pp. 138 - 143, 2012.

[20] Dupor, S., Jovanovi, V., "*An approach to conceptual modelling of ETL processes*", 37th International Convention on Information and Communication Technology, pp. 1485 - 1490, 2014.

[21] Ta'a, A., Abdullah, M. S., "*Goal-ontology approach for modeling and designing ETL processes*", Procedia Computer Science, Vol. 3, No. 1, pp. 942–948, 2011.

[22] Huang, J., Guo, C., "*An MAS-based and fault-tolerant distributed ETL workflow engine*", 16th International Conference on Computer Suppurted Cooperative Work in Design, pp. 54 - 58, 2012.

[23] Simitsis, A., Wilkinson, K., Dayal, U., Castellanos, M., "*Optimizing ETL Workflows for Fault-Tolerance*", 26th International Conference on Data

Engineering, pp. 385 - 396, 2010.

[24] Vassiliadis, P., Simitsis, A., Skiadopoulos, S., "*Conceptual modeling for ETL processes*", 5th ACM international workshop on Data Warehousing and OLAP, pp. 14 - 21, 2002.