

مشاوران نرم افزار اعوان

استاندارد همانندسازی پایگاه های داده

نسخه ۱.۰

مشاوران نرم افزار اعوان

تاریخ گزارش

۱۳۸۶/۱۰/۱۰

فهرست

بخش ۱: مقدمه ای بر REPLICATION در ORACLE	۳
قوانین در محیط همانند سازی	۴
مجموعه قوانین در محیط همانندسازی	۴
زیرمجموعه سازی در محیط همانند سازی	۵
فرایند گرفت در محیط همانندسازی	۶
فرایند انتشار در محیط همانند سازی	۷
فرایند اعمال در محیط همانند سازی	۹
بخش ۲: معماری محیط همانند سازی	۱۱
انواع جداول در معماری Hub-and-Spoke	۱۲

بخش ۱ : مفاهیم Oracle Replication

تعریف Replication: به فرایند تکثیر و به اشتراک گذاری اشیای پایگاه داده و اطلاعات بین دو یا چند پایگاه داده Replication می گویند.

برای این کار روشهای مختلفی وجود دارد که آخرین روش پیشنهادی توسط شرکت Oracle استفاده از Oracle Streams می باشد^۱. روش عملکرد Oracle Streams اینست که باید تمامی تغییراتی که در یک پایگاه داده اعمال می شود را به پایگاه داده دیگر انتقال داد. بدین ترتیب تمامی پایگاه های داده موجود در یک محیط Replication همگام با یکدیگر تغییر پیدا می نمایند. در یک محیط Replication به پایگاه داده ای که تغییرات در آن ایجاد می شود **پایگاه داده مبدا** و به پایگاه داده ای که تغییرات به آن منتقل می شود **پایگاه داده مقصد** می گویند. در این مقاله از کلمه **هماند سازی** برای Replication استفاده می شود.

در یک محیط تکثیر به طور معمول عملیات زیر انجام می شود:

۱. یک Capture Process که وظیفه آن ضبط تغییرات اعمال شده بر روی پایگاه داده در یک صف می باشد که با نام LCR شناخته می شوند. هر LCR بیانگر یک تغییر اعمال شده بر روی پایگاه داده می باشد. اگر تغییر اعمال شده یک تغییر DML باشد سطری از داده که مورد تغییر قرار گرفته و چگونگی تغییر ذخیره می شود و اگر تغییر مربوطه از نوع DDL باشد نوع تغییر به همراه شی مورد تغییر قرار گرفته در LCR ذخیره می شوند. در این مقاله Capture Process را **فرایند گرفت** می نامیم.
۲. یک Propagation Process وظیفه انتقال LCR ها را از صفی که توسط فرایند گرفت تغذیه می شود به یک صف دیگر در پایگاه داده مقصد به عهده دارد. یک LCR می تواند قبل از رسیدن به صف اصلی مقصد بین چند صف میانی توسط فرایندهای انتشار جابجا شود.
۳. در پایگاه داده مقصد Apply Process موظف است که تغییرات را از صف پر شده توسط فرایند انتشار، خوانده و بر روی اشیای مربوطه اعمال نماید. در این مقاله از لفظ فرایند اعمال به جای Apply Process استفاده می شود.

^۱ برای اطلاعات بیشتر به منابع ذکر شده در صفحه آخر مراجعه نمایید.

۴. در محیط های همانندسازی انجام مراحل اول و آخر اجباری و انجام مرحله ۲ اختیاری است، بدین معنی که فرایند گرفت LCR ها را به صورت مستقیم درون صفی قرار دهد که فرایند اعمال از آن تغذیه می شود که در این مدل نیازی به استفاده از فرایند انتشار نیست.

قانون همانندسازی

در هر محیط همانند سازی و در هر کدام از فرایندهای مربوط به آن کاربر می تواند قوانینی را بر روی داده های مورد نظر خود اعمال نماید که با توجه به فرایند مربوط به قوانین به آن Capture Rule، Propagation Rule، Apply Rule می گویند. در واقع قانون (Rule) امکانی است برای اینکه هر کدام از فرایندها در صورت ارضا شدن آن قانون به انجام عمل خاصی پردازد. وظیفه بررسی قوانین در مورد اطلاعات را یک بخش داخلی Oracle به نام Rule Engine به عهده دارد. مدیران پایگاه داده برای اعمال کنترل بر نحوه حرکت اطلاعات درون و بین پایگاههای داده از قوانین استفاده می نمایند.

مجموعه قوانین همانندسازی

تعدادی از قوانین را می توان در یک گروه قرار داد که به آن Rule Set می گویند. و به هر استفاده کننده (فرایند) در محیط همانند سازی می توان (Positive Rule Set) یک مجموعه قوانین مثبت و (Negative Rule Set) یک مجموعه قوانین منفی به هر کدام از فرایندها اختصاص داد. عمل مربوط به آن فرایند فقط در صورتی انجام می شود که تمامی قوانین مثبت مربوط به آن درست بوده و هیچکدام از قوانین منفی مربوط به آن نادرست نباشد. لازم به ذکر است که اگر برای یک فرایند هم قوانین مثبت هم قوانین منفی تعریف شده باشد ابتدا قوانین منفی بررسی شده و سپس قوانین مثبت بررسی می شوند.

به طور خاص شما به عنوان مدیر پایگاه داده ها می توانید به طرق زیر بر روی روند گردش اطلاعات در یک پایگاه داده کنترل ایجاد کنید:

- به کمک قوانینی که برای یک فرایند گرفت تعریف می نمایید. فرایند گرفت در هنگام بروز تغییر در پایگاه داده آن را با قوانین تعریف شده برای خود بررسی نموده و اگر شرایط درست حاصل شود آن تغییر را به صورت یک LCR درون صف مربوطه ذخیره می نماید در غیر اینصورت تغییر مربوطه نادیده گرفته می شود (Discard می شود).

- به کمک قوانینی که برای یک فرایند انتشار تعریف می نمایید. فرایند انتشار در هنگام خواندن LCR ها از صف ورودی خود آنها را با قوانین مربوط به خود مطابقت داده و اگر قوانین مربوطه ارضا نشوند LCR مورد نظر را به صف مقصد (که معمولاً در پایگاه داده مقصد قرار دارد) منتقل می نماید و اگر قوانین مربوطه ارضا نشوند LCR مربوطه نادیده گرفته می شود.
- به کمک قوانینی که برای یک فرایند اعمال تعریف می نمایید. فرایند اعمال LCR ها را از صف مربوطه خوانده و در صورت ارضای قوانین تعریف شده برای خود اقدام به اعمال آنها بر روی اشیای پایگاه داده مقصد می نماید و در غیر اینصورت آنها را نادیده می گیرد.

برای ایجاد قوانین برای یک محیط همانندسازی می توانید از بسته DBMS_STREAMS_ADM

استفاده نمایید. که این قوانین در سطو زیر می توانند وجود داشته باشند:

- Table : که حاوی قوانینی است که در صورت درست بودن تغییرات را بر روی یک جدول اعمال می نماید.
- Schema : که حاوی قوانینی است که در صورت درست بودن تغییرات را بر روی یک شما اعمال می نماید.
- Global : که حاوی قوانینی است که در صورت درست بودن تغییرات را بر روی کل پایگاه داده اعمال می نماید.

علاوه بر مطالب فوق یک LCR تولید شده توسط سیستم یا توسط قوانین مربوط به DDL مقدار درست را بر می گرداند یا توسط قوانین مربوط به DML و نمی تواند هر دو را ارضا نماید بنابراین مثلاً اگر نیاز است که یک جدول خاص هم در مورد DML و هم DDL مورد همانندسازی قرار بگیرد باید یک قانون مربوط به DML و یک قانون دیگر برای DDL برای آن در نظر گرفت.

زیرمجموعه سازی^۲ در محیط همانند سازی

در یک محیط که از سیستم Streams Replication استفاده می نماید می توان تنظیمات را به گونه ی انجام داد که فقط زیر مجموعه مورد نظر از یک جدول خاص همانند سازی شود که این کار با تنظیم قوانین به

^۲ Subsetting

صورتی که فقط برای بخشی از یک جدول ارضا شوند قابل دسترسی است. به عنوان مثال اگر در یک پایگاه داده اطلاعات مربوط به بخشهای مختلف شرکت نگهداری می شود می توان تنظیمات را به گونه ای انجام داد که به پایگاه داده های محلی که در هر بخش شرکت قرار دارد فقط اطلاعات مربوطه همانند سازی شوند و مثلا اطلاعات بخش فروش به بخش پرسنلی وارد نشود. لازم به ذکر است که زیر مجموعه های تعریف شده (که توسط یک Condition مرزهای آنها مشخص می شود) امکان تغییر در طول زمان را ندارند. مثلا داده ای که در یک زمان مربوط به یک شعبه از یک شرکت می باشد نمی تواند در زمان دیگر مربوط به شعبه دیگر تلقی شود. چون در مقصد جدید، اطلاعات قبلی وجود ندارد که بتوان آن را Update نمود. همچنین این تغییر باعث باقی ماندن مقادیری اطلاعات بلا استفاده در مقصد قبلی اطلاعات خواهد شد و اگر در آینده مقصد اطلاعات به حالت اول باز گردد اطلاعات قبلی با اطلاعات جدید دچار تداخل می شود. بنابراین برای استفاده از امکان زیر مجموعه سازی، Condition های استفاده شده برای زیر مجموعه ها باید طوری تعریف شوند که برای هر جدول زیر مجموعه ثابتی از اطلاعات در آن Condition خاص صدق کند یا نکند و این زیر مجموعه امکان تغییر در طول زمان را ندارد.

این کنترل بر روی حرکت اطلاعات می تواند در هر یک از سه بخش ذکر شده بنا به نیاز انجام شود. در مرحله گرفت یا مرحله انتشار یا اعمال می توان قوانینی تعریف نمود که باعث شود فقط بخشی از یک جدول مورد توجه قرار گیرد و بقیه آن نادیده گرفته شود.

فرایند گرفت^۳ در محیط همانندسازی

برای آماده سازی اطلاعات در محیط همانند سازی باید تمامی اطلاعات در پایگاه داده مبدا گرفته شود و سپس به پایگاه داده مقصد انتقال یابد عمل گرفتن (Capture) اطلاعات در پایگاه داده مبدا به دو روش کلی قابل انجام است:

- توسط یک فرایند گرفت
- توسط یک برنامه تنظیم شده به صورت دلخواه

^۳ Capture

نگاهی به فرایند گرفت

تمامی تغییراتی که بر روی یک پایگاه داده اعمال می شود توسط خود نرم افزار پایگاه داده Oracle در Redo Log ذخیره می شود تا در صورت بروز هرگونه اشکال در رسانه ذخیره سازی یا هر بخش دیگر در پایگاه داده بتوان آخرین تغییرات اعمال شده بر روی پایگاه داده را بازیابی نمود. در محیطهای همانندسازی فرایند گرفت از طریق Mining در این Log اطلاعات مورد نیاز را طبق قوانینی که برای این فرایند گرفت خاص تعریف شده است استخراج نموده و از روی آنها LCR ها را ایجاد می نماید. سپس این LCR ها درون یک صف که از قبل برای فرایند گرفت تعریف شده است ذخیره می شوند تا توسط فرایندهای دیگر همانند سازی مورد استفاده قرار بگیرند.

دو نوع از LCR ها وجود دارند: row LCR ها که بر اثر اجرای یک دستور DML بر روی داده ها ایجاد می شود و موضع عمل آن یک سطر از یک جدول می باشد و DDL LCR که بر اثر اجرای یک دستور DDL بر روی پایگاه داده ایجاد می شود و موضع عمل آن بر روی یکی از اشیای پایگاه داده است .

فرایند انتشار^۴ در محیط همانند سازی

پس از گرفته شدن تغییرات توسط فرایند گرفت و ذخیره شدن آنها درون صف مربوطه فرایند دیگری به نام فرایند انتشار LCR های ذخیره شده را به صف دیگری در پایگاه داده مقصد منتقل می نماید. برای فرایند انتشار نیز می توان قوانینی تعریف نمود که توسط یک مجموعه قوانین به فرایند معرفی می شوند. فرایند انتشار LCR های قرار گرفته در صف مبدا را با قوانینی که در مجموعه قوانین مربوطه تعریف شده تطابق می دهد و در صورت تطابق، LCR مورد نظر را به صف مقصد منتقل می نماید. در صورتی که LCR مورد بررسی قوانین مربوطه را ارضا نکند آن را از صف مبدا خارج نموده ولی در صف مقصد قرار نمی دهد یا به اصطلاح آن را نادیده می گیرد.

^۴ Propagation

زمانبندی انتشار

یکی از مزایای تکنولوژی استفاده شده برای همانند سازی اطلاعات عملکرد تقریباً همزمان این سیستم می باشد. بدین معنی که اطلاعات بلافاصله پس از اعمال تغییرات (Insert – Update – Delete) توسط فرایند گرفت به فرایند انتشار و سپس به فرایند اعمال منتقل می شود. بنابراین تغییرات به صورت تقریباً همزمان از مبدا به مقصد منتقل می شود. البته این همزمانی بسته به میزان بار فعلی بر روی سرور پایگاه داده ، تعداد فرایندهای مختلف موجود بر روی سرور و همچنین توانایی شبکه در انتقال اطلاعات متغیر است اما به لحاظ تئوری تغییرات به صورت Real Time منتقل می شوند. این مدل عملکرد، دسترسی فوری به اطلاعات توسط کاربران مختلفی که در مکانهای مختلف از آن استفاده می نمایند را امکانپذیر می نماید. این همگرایی دائمی به وجود آمده در اطلاعات ، خود باعث پیشگیری از به وجود آمدن برخورد میان نسخه های مختلف اطلاعات در مکانهای متفاوت می شود. البته این پیشگیری کامل نیست و نیاز به مکانیزمهای حل برخورد همچنان وجود دارد اما امکان بروز برخورد به میزان بسیار زیادی نسبت به سیستمهای Offline کمتر می باشد. از مزایای دیگر این روش می توان به توزیع بار همانند سازی در طول زمان اشاره نمود که مانند سیستمهای Offline در زمانهای همانند سازی سربار مضاعف ایجاد نمی کند.

همچنین اگر نیاز باشد که در محیطی عملیات همانندسازی در زمانهای خاصی انجام شود می توان از امکان زمانبندی که برای فرایند انتشار تعریف می شود استفاده نمود. هر فرایند انتشار دارای یک زمانبندی (Schedule) برای انجام عملیات انتشار است. که این زمانبندی بر اساس معماری در نظر گرفته شده برای محیط همانند سازی تعیین می شود. در این حالت فرایند گرفت اقدام به ذخیره تغییرات اعمال شده بر روی پایگاه داده در صف مورد نظر می نماید. اما فرایند انتشار فقط در زمانهای خاصی که از قبل تعریف شده اند فعال شده و محتویات صف مبدا در پایگاه داده مبدا را به صف مقصد در پایگاه داده مقصد منتقل می نماید. سپس فرایند اعمال تغییرات را بر روی اشیای پایگاه داده مقصد اعمال می نماید.

فرایند اعمال^۵ در محیط همانند سازی

پس از اینکه فرایند انتشار LCR ها را به صف مربوطه در پایگاه داده مقصد انتقال داد فرایند دیگری به نام فرایند اعمال (Apply) این LCR ها را به جداول مربوطه در پایگاه داده اعمال می نماید. بنابراین اگر در هنگام اعمال تغییرات بر روی جداول مقصد برخوردی^۶ صورت گیرد فرایند اعمال مسئول برخورد مناسب با این برخورد خواهد بود. به طور کلی امکان برخورد در موارد زیر بررسی می شود:

برخورد Update

وقتی رخ می دهد که فرایند اعمال قصد Update کردن یک سطر از یک جدول را دارد ولی مقادیر قبلی که انتظار می رود در جدول موجود باشند موجود نیستند (مثلا مقادیر توسط یک برنامه تغییر پیدا کرده اند) برای حل این نوع از برخورد می توان برای فرایند اعمال دو حالت Overwrite و Discard را تعریف نمود که در حالت اول سطر در حال اعمال، جایگزین سطر درون جدول می شود و در حالت دوم سطر در حال اعمال، نادیده گرفته می شود و فرایند اعمال به LCR های بعدی درون صف می پردازد.

برخورد Uniqueness

وقتی رخ می دهد که فرایند اعمال قصد دارد سطری را به یک جدول بیفزاید که قبلا وجود دارد (Primary Key یا Unique Constraint در حال نقض شدن است). برای این نوع از برخوردها راه حلهای از پیش تعریف شده ای وجود ندارد بلکه باید از بروز آنها در زمان طراحی جلوگیری نمود یا اینکه یک Conflict Handler دستی برای آن ایجاد نمود.

برخورد Delete

وقتی رخ می دهد که فرایند اعمال قصد دارد سطری از جدول را تغییر دهد که قبلا پاک شده است و وجود ندارد. برای حل این نوع از برخورد نیز راه حل از پیش تعریف شده وجود ندارد و باید برای آن یک Conflict Handler دستی ایجاد نمود.

^۵ Apply

^۶ Conflict

برخورد Foreign Key

وقتی رخ می دهد که فرایند اعمال قصد دارد سطری از جدول را تغییر دهد یا ایجاد نماید که یک Foreign Key را نقض می نماید. برای حل این نوع از برخورد نیز راه حل از پیش تعریف شده وجود ندارد و باید برای آن یک Conflict Handler دستی ایجاد نمود.

مدیریت برخوردها^۷

در صورتی که هرکدام از انواع برخوردهای فوق در سیستم طراحی شده محتمل باشد، باید برای آن یک Conflict Handler به گونه ای طراحی نمود که سبب همگونی^۸ داده در پایگاههای داده مختلف گردد. برای برخوردهای Update خود سیستم streams به صورت built-in دارای پروسه های Discard و Overwrite می باشد، که می توان از آنها استفاده نمود. ولی در صورتی که نیاز به عملیات پیچیده تری برای مدیریت برخوردها باشد می توان از custom conflict handler کمک جست. مدیر برخورد دلخواه^۹ یک Stored Procedure است که در زمان برخورد فراخوانی می شود و اطلاعات مربوط به LCR و خطایی که رخ داده نیز در اختیار آن قرار می گیرد. سپس با توجه به استراتژی طراحی شده برای مدیریت برخوردها، اقدام به رفع خطا و اجرای مجدد LCR می نماید. بنابراین برای هر کدام از خطاهای محتمل update, insert, delete می توان یک مدیر برخورد دلخواه ایجاد نمود و آن را به جدول مربوطه متصل ساخت تا در زمان بروز برخورد به صورت خودکار توسط oracle فراخوانی شود. در صورتی که در اجرای آن نیز خطایی رخ دهد یا procedure مربوطه در مدیریت خطا ناتوان باشد، خطای مربوطه به همراه پیغام و کد مربوطه در صف خطاهای مربوط به فرایند اعمال قرار می گیرد.

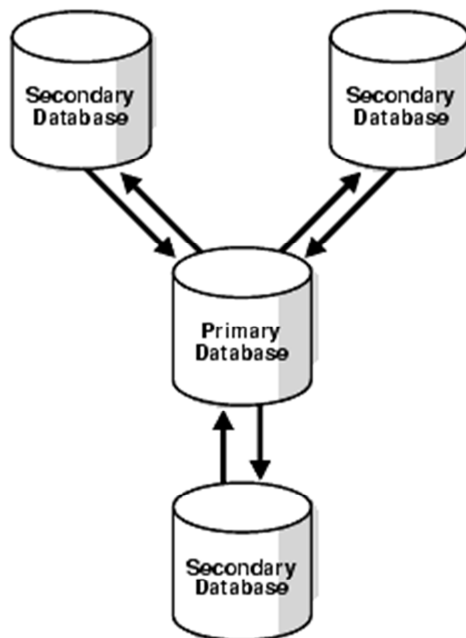
^۷ Conflict Handling

^۸ Convergence

^۹ Custom conflict handler

بخش ۲: معماری محیط همانند سازی

پس از بررسی ساختار موجود ، همچنین امکانات موجود برای ایجاد یک محیط همانند سازی می توان یک معماری مناسب برای محیط همانند سازی طراحی نمود. که تمامی پایگاه های داده موجود در محیط بر اساس جایگاه خود در چهارچوب این معماری تنظیم شده و وظایف مربوط به خود را انجام می دهند. پس از بررسی های معماریهای مختلف قابل استفاده در محیطهای همانند سازی، مناسب ترین معماری برای شرایط موجود معماری Hub and Spoke در نظر گرفته شد. در این معماری یک پایگاه داده مرکزی (Hub) وجود دارد که نقطه اتصال مرکزی در میان پایگاه های داده می باشد و تمامی اطلاعات به صورت تجمیعی در این پایگاه داده موجود می باشد. همچنین در این سیستم تعدادی (Spoke) نیز موجود هستند که مستقیماً به Hub مرکزی متصل می باشند. لازم به ذکر است که در این معماری Spoke ها به طور مستقیم با یکدیگر ارتباط ندارند و فقط امکان ارتباط با Hub را دارا می باشند . همانگونه که از تعریف فوق مشخص است در این معماری دو نقش متفاوت Hub و Spoke برای پایگاههای داده تعریف می شود. در شکل زیر مشاهده می نمایید که یک پایگاه داده Primary نقش Hub و چندین پایگاه داده Secondary نقش Spoke را در این سیستم فرضی ایفا می نمایند.



انواع جداول در معماری Hub-and-Spoke

با توجه به معماری فوق می توان استراتژی های زیر را در مورد نحوه همانند سازی جداول مختلف اتخاذ نمود. که هر جدول در سیستم پایگاه اطلاعاتی در یکی از گروه های زیر قرار می گیرد و با توجه به استراتژی تعیین شده برای آن ، همانند سازی می شود.

• جداول بالا به پایین (Up-Down)

پایگاه داده مرکزی Hub برای این جداول منبع اصلی به حساب آمده و تغییرات به وجود آمده در این جداول فقط از Hub به Spoke منتقل می شود. بنابراین یا این جداول در Spoke ها تغییر پیدا نمی کنند یا اینکه تغییرات اعمال شده بر روی آنها در Spoke ها نادیده گرفته می شود. معمولا جداولی که شامل اطلاعات پایه هستند که در پایگاه داده مرکزی تعریف شده و در پایگاه های داده کناری فقط مورد استفاده قرار می گیرند در این گروه قرار می گیرند.

• جداول دو طرفه (Two-Way)

این گروه از جداول به صورت دو طرفه بین هر Spoke و Hub همانند سازی می شوند. بنابراین اگر تغییری بر روی این جداول در Hub ایجاد شود این تغییر به تمامی Spoke های مربوطه منتقل می شود. همچنین اگر تغییری در یکی از Spoke ها بر روی این نوع جداول اعمال شود ابتدا این تغییر به Hub منتقل شده و سپس توسط Hub به تمامی Spoke های دیگر (به غیر از Spoke مرجع تغییر) انتشار پیدا می کند. بنابراین تمامی تغییرات اعمال شده بر روی این گروه از جداول به تمامی محیط همانند سازی منتشر می شود.

• جداول پایین به بالا (Down-Up)

در این گروه از جداول تمامی تغییرات به وجود آمده در Spoke ها به پایگاه داده Hub منتقل می شود. بنابراین تغییرات اعمال شده در Hub به Spoke ها منتقل نمی شود. این استراتژی معمولاً برای جداول Logging که نیاز به انتقال Log های موجود از تمامی Spoke ها به Hub وجود دارد استفاده می شود.

• جداول بالا به پایین شرطی (Up-Down Conditional)

رفتار این گروه از جداولها در محیط همانند سازی مشابه جداول بالا به پایین است با این تفاوت که می توان از انتقال همه تغییرات Hub به Spoke ها جلوگیری نمود و فقط اطلاعاتی که مورد نیاز Spoke مورد نظر است به آن منتقل شود. این استراتژی برای اطلاعاتی که لزومی ندارد به همه Spoke ها منتقل شود کاربرد دارد.

• جداول دو طرفه شرطی (Two-Way Conditional)

این گروه از جداول نیز مشابه با جداول دو طرفه همانند سازی می شوند با این تفاوت که برای انتقال تغییرات از Hub به Spoke شرایطی ایجاد می شود و در صورت برقراری این شرایط در مورد سطر مورد بررسی، تغییرات مربوط به آن به Spoke مورد نظر منتقل می شود. ولی در انتقال تغییرات از Spoke به سمت Hub همه تغییرات منتقل می شوند.

جدول نمونه تعریف محیط همانند سازی

در جدول زیر نام جداول مختلف موجود در محیط همانندسازی ، استراتژی اتخاذ شده برای آنها و همچنین شروط مختلف برای جداول شرطی آمده است.

Model	Table Name	Condition
Up-Down Conditional	Table1	(Column1<1000) and(Column2 = "Tehran")
Two-Way Conditional	Table2	(CityID = 3)
	Table3	(CustomerID < 20000)
	Table4	(SiteID=5)
Down-Up	Table5	-----
Two-Way	Table6 Table7 Table8 Table9	-----
Up-Down	Table10 Table11 Table12	-----

در جدول بالا مشاهده می کنید که سطرها از جدول Table2 فقط در صورتی که در ستون با نام CityID مقدار برابر ۳ داشته باشد مورد همانند سازی قرار می گیرند. البته می توان برای هر Spoke در سیستم همانند سازی یک شرط مشخص نمود. مثلا اطلاعات مربوط به همه شهرها در پایگاه داده مرکزی موجود است و همچنین اطلاعات هر شهر توسط شرایط تعریف شده برای آن همانند سازی دو طرفه می شود، یعنی اگر در پایگاه داده Hub تغییرات ایجاد شد به پایگاه داده داده Spoke مورد نظر انتقال پیدا می کند و همچنین تغییرات از پایگاه داده Spoke به پایگاه داده مرکزی منتقل می شود.

References:

1. Oracle® Streams - Replication Administrator's Guide [Doc No: B14228-02]
2. Oracle® Streams - Concepts and Administration [Doc No: B14229-02]
3. Oracle® Database - PL/SQL Packages and Types Reference [Doc No: B14258-01]
4. Oracle® Documentation Website: <http://www.oracle.com/pls/db102/db102.homepage>