

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

قرینه سازی - حالت های اجرایی مختلف

مدرس : مهندس افشین رفوآ

دوره آموزش [MVC](#)

قرینه سازی - حالت های اجرایی مختلف (Database Mirroring Operating Modes)

در این مبحث شما را با حالت های اجرایی مختلف (همزمان و ناهمزمان) برای **session** قرینه سازی (گرفت رونوشت عینی) از پایگاه داده آشنا می سازیم.

در ابتدا به شرح چندین واژه که جز بخش اصلی مقاله ی پیش رو هستند، می پردازیم:

High-performance mode (حالت کارایی بهینه): در این حالت **session** قرینه سازی به صورت ناهمزمان اجرا شده و تنها از سرور اصلی و سرور قرینه استفاده می کند. همچنین در این حالت تنها تعویض نقشی (**role-switching**) که صورت می گیرد از نوع **forced service** / سرویس اجباری (با از دست رفت احتمالی اطلاعات) می باشد.

High-safety mode (حالت امنیت بهینه): **Database mirroring session** به صورت همزمان اجرا می شود و در صورت نیاز علاوه بر سرور اصلی و جایگزین یک **witness** نیز بکار می برد.

Transaction safety (امنیت تراکنش): یک خاصیت پایگاه داده مختص **mirroring** (قرینه سازی) هست که مشخص می کند آیا **mirroring session** (گرفتن رونوشت عینی یا قرینه از پایگاه داده) به طور همزمان (**synch**) اجرا شود یا ناهمزمان (**asynch**). دو لایه ی امنیتی وجود دارد: **FULL** و **OFF**.

Witness: یک نمونه یا **instance** اختیاری که به سرور جایگزین (قرینه) این امکان را می دهد که زمان مناسب برای راه اندازی **automatic failover** را بشناسد. بر خلاف دو **failover partner** (همراهانی که مکمل هم هستند، **principal** و **mirror**)، **witness** به پایگاه داده سرویس نمی دهد. پشتیبانی

automatic failover تنها نقشی است که **witness** ایفا می کند. **witness** ویژه ی کار با حالت امنیت بهینه طراحی شده است.

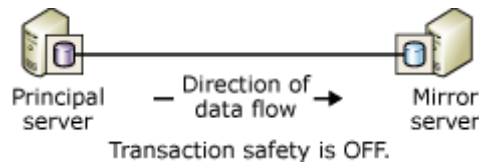
قرینه سازی ناهمزمان (حالت high-performance)

این قسمت نحوه ی کارکرد قرینه سازی ناهمزمان (**asynchronous mirroring**)، اینکه چه زمانی برای استفاده از حالت **high-performance** مناسب می باشد و نیز به هنگام از کار افتادگی سرور اصلی چگونه باید اقدام کرد، را برای شما شرح می دهد.

هنگامی که **transaction safety** بر روی **OFF** تنظیم شده باشد، **session** قرینه سازی به صورت ناهمزمان اجرا می شود. عملیات ناهمزمان تنها از یک حالت پشتیبانی می کند و آن حالت **high-performance** می باشد. این حالت با قربانی کردن دسترس پذیری بالا، کارایی بهینه را ارائه می دهد (کارایی را بهبود می بخشد ولی دسترس پذیری را کاهش می دهد). حالت کارایی بهینه یا **high-performance** فقط از دو سرور جانشین (**mirror**) و اصلی (**principal**) استفاده می کند. مشکلاتی که در سمت سرور جانشین رخ می دهد، هیچگاه بر روی سرور اصلی تاثیر نمی گذارد. به مجرد از کار افتادگی سرور اصلی، پایگاه داده ی جانشین **DISCONNECTED** علامت گذاری شده ولی به عنوان یک نسخه ی ذخیره ی آماده از پایگاه داده (**warm standby**) در دسترس و قابل استفاده می باشد.

حالت **high-performance** تنها از یک نوع **role-switching** (تعویض نقش) پشتیبانی می کند: سرویس اجباری یا **forced service** (با احتمال از دست رفت جزئی اطلاعات) که سرور جانشین را به عنوان یک ذخیره یا پشتیبان آماده مورد استفاده ی خود قرار می دهد. **Forced service** تنها یکی از پاسخ های ممکن به رویداد از کار افتادگی سرور اصلی است. از آنجایی احتمال از دست رفت اطلاعات و داده ها وجود دارد، توصیه می شود پیش از بکار بردن و راه اندازی **forced service** (تحمیل سرویس به سرور جانشین) دیگر گزینه های در دست را نیز امتحان کنید.

نگاره ی زیر یک **session** را به تصویر می کشد که با استفاده از حالت **high-performance** پیکربندی شده است.



در حالت **high-performance**، به محض اینکه سرور اصلی (**principal**) گزارش مربوط به تراکنش را به سرور جانشین (**mirror**) می فرستد، سرور اصلی بدون اینکه منتظر دریافت تصدیق از سرور جانشین شود، یک تأییدیه به سرویس گیرنده (**client**) ارسال می کند. تراکنش ها بدون اینکه منتظر بمانند تا سرور جانشین گزارشات را بر روی دیسک خود ذخیره کند، تایید ثبت (**commit**) می شوند. عملیات ناهمزمان به سرور اصلی این امکان را می دهد که با کمترین میزان نهمتگی تراکنش (**transaction latency**) اجرا و راه اندازی شود.

سرور جانشین تلاش می کند تا با سطرهای گزارش ارسال شده توسط سرور اصلی خود را بروز و هماهنگ نگه دارد، با این حال امکان عقب افتادن آن (پایگاه داده ی جانشین) از پایگاه داده ی اصلی وجود دارد. اما لازم به ذکر است که فاصله یا اختلاف بین این دو اغلب بسیار ناچیز و محدود است. باید توجه داشته باشید که این اختلاف در شرایطی که سرور اصلی زیر کار باری بسیار سنگین قرار دارد و یا سرور جانشین به هر دلیلی درگیر انجام کارهای سنگین می باشد، بسیار بیشتر می شود.

چه زمانی استفاده از حالت **high-performance** مناسب می باشد؟

این حالت در شرایط یا سناریوی بازیابی از حادثه (**disaster-recovery scenario**) که در آن سرور های اصلی و جانشین از یکدیگر فاصله ی بسیار زیادی دارند و نیز در مواردی که نمی خواهید خطاهای جزئی سرور اصلی را تحت تاثیر خودشان قرار دهند، بسیار کارآمد تلقی می گردد. (**disaster-recovery**: طرح یک واکنش به هنگامی که خطری نرم افزار و سخت افزار را تهدید می کند).

توجه: log shipping (ارسال گزارشات به سرور) می تواند مکمل بسیار مناسبی برای قرینه سازی (**mirroring**) و همچنین جایگزین درخوری برای قرینه سازی ناهمزمان (**asynchronous mirroring**) باشد.

WITNESS و تاثیر آن بر حالت کارایی بهینه (**high-performance**)

در صورت استفاده از **Transact-SQL** برای پیکربندی حالت کارایی بهینه، هرگاه که خاصیت **SAFETY** بر روی **OFF** تنظیم شده بود، اکیدا توصیه می کنیم که خاصیت **WITNESS** را نیز بر روی **OFF** تنظیم کنید. اگرچه **Witness** می تواند همراه با حالت کارایی بهینه بکار گرفته شود، اما استفاده از آن نه تنها هیچ مزیتی ندارد بلکه ریسک را نیز بالا می برد.

چنانچه اتصال **witness** از **session** به دلیل از کار افتادگی یکی از سرورهای همراه (**partner**) قطع شود، در آن صورت پایگاه داده ی مورد نظر از دسترس خارج می گردد. دلیل این امر این است که اگرچه حالت کارایی بهینه لزوماً به یک **witness** نیاز ندارد، اما در صورت استفاده از آن، **session** به یک حد نصاب متشکل از حداقل دو نمونه ی سرور دارد که در صورت برقرار نبودن این شرط (اگر حد نصاب برای **session** برپا نباشد)، **session** دیگر نمی تواند به پایگاه داده سرویس دهد.

هنگامی که **witness** در **session** بر روی حالت **high-performance** تنظیم شده، اجرای حد نصاب بدین معنا است که:

اگر سرور جانشین از دست برود، سرور اصلی باید به **witness** متصل باشد، در غیر این صورت سرور اصلی پایگاه داده ی مستقر بر روی خود را از حالت آنلاین خارج ساخته تا اینکه یا سرور **witness** و یا سرور جانشین (**mirror**) دوباره به **session** ملحق شوند.

در صورت از دست رفت سرور اصلی، اجرای **forced service** بر روی سرور جانشین (تحمیل سرویس به آن) لازمه آن است که سرور جانشین به **witness** متصل باشد.

واکنش به از کار افتادگی سرور اصلی (**principal server**)

هنگامی که سرور اصلی خراب می شود، مالک پایگاه داده چندین گزینه پیش روی خود دارد:

پایگاه داده را (در حالت غیر قابل دسترس) رها کند تا سرور اصلی بار دیگر دسترس پذیر شود.

در صورت سالم بودن پایگاه داده ی اصلی و گزارشات تراکنش (**transaction log**) آن، این گزینه

تمامی تراکنش های تایید و ثبت شده را البته با قربانی کردن دسترس پذیری، حفظ می کند.

Session را به طور کامل متوقف ساخته و پایگاه داده را به صورت دستی بروز رسانی کند، سپس یک **session** قرینه سازی جدید را آغاز کند.

اگر پایگاه داده ی اصلی از دست رفته ولی سرور اصلی همچنان پابرجا است، باید بلافاصله از انتهای (**tail**) فرایند ثبت گزارشات (**log**) یک نسخه ی پشتیبان بر روی سرور اصلی تهیه کند. (**tail backup**) به فرایند ضبط هر سطر گزارشی که هنوز از آن نسخه ی پشتیبان گرفته نشده گفته می شود تا از این طریق از دست رفتن اطلاعات جلوگیری شود و حفظ تمامیت زنجیره ی گزارشات تضمین گردد). اگر **tail-log backup** با موفقیت اجرا شد، حذف **mirroring** بهترین گزینه خواهد بود. پس از حذف **mirroring**، می توان گزارشات (**log**) را بر روی پایگاه داده ی جانشین قبلی بالا آورد (احیا کرد) این امر باعث می شود کلیه اطلاعات و داده ها حفظ شوند.

توجه: چنانچه **tail-log backup** با شکست مواجه شود، دیگر نمی توان منتظر سرور اصلی شد تا خود را بازیابی کند. در چنین موردی باید از **forced service** بهره گرفت، زیرا که وضعیت جاری **session** را حفظ می کند.

گزینه ی آخر این است که از **forced service** برای سرور جانشین استفاده کند (که در آن احتمال از دست رفت اطلاعات وجود دارد).

سرویس اجباری (**forced service**) صرفاً یک روش بازیابی از حادثه (**disaster recovery**) می باشد و استفاده از آن باید به ندرت صورت گیرد. سرویس اجباری تنها در شرایطی امکان پذیر می باشد که سرور اصلی از کار افتاده باشد، **session** ناهمزمان باشد (امنیت تراکنش یا **transaction safety** بر روی **OFF** تنظیم شده باشد) و نیز در مواردی که **session** دارای **witness** نباشد (خاصیت **WITNESS** بر روی **OFF** تنظیم شده باشد) و یا **witness** به سرور جانشین متصل باشد (البته در صورت وجود حد نصاب).

سرویس اجباری باعث می شود سرور جانشین نقش سرور اصلی را بر عهده گرفته و نسخه ی عینی از پایگاه داده را که در دست دارد به سرویس گیرنده ارائه دهد. هنگامی که سرویس به سرور جانشین تحمیل می شود، هر گزارش تراکنشی که در اختیار **principal** است ولی به سرور جانشین ارسال نشده، از دست می رود. به همین خاطر است که توصیه می کنیم از سرویس اجباری تنها در شرایطی استفاده کنید که از دست

رفت احتمالی اطلاعات و داده ها از اهمیت چندانی برخوردار نیست ولی دسترس پذیری بالای پایگاه داده امری ضروری می باشد.

قرینه سازی همزمان – حالت high-safety (synchronous mirroring)

در این بخش شما با نحوه ی عملکرد قرینه سازی آشنا شده، سپس دو حالت مختلف امنیت بهینه (با **automatic failover** و بدون آن) را خواهید آموخت.

در این قسمت همچنین مطالبی را در رابطه با **witness** و نقش آن در **automatic failover** را مطرح خواهیم کرد.

هنگامی که **transaction safety** بر روی **FULL** تنظیم می شود، قرینه سازی (**mirroring**) در حالت **high-safety** اجرا شده و پس از یک مرحله ی اولیه ی هماهنگ سازی به طور همزمان عمل می کند. این بخش به شرح جزئیات **session** قرینه سازی می پردازد که ویژه ی عملیات همزمان (**asynch operation**) تنظیم و پیگیربندی شده است.

به منظور ایجاد بستر مناسب و زمینه سازی برای عملیات همزمان ویژه ی **session**، سرور جانشین (**mirror**) باید پایگاه داده ی جانشین را با پایگاه داده ی اصلی هماهنگ (**synch**) کند. هنگامی که **session** آغاز می گردد، سرور اصلی شروع به ارسال گزارشات فعال خود به سرور جانشین می کند. سرور جانشین تمامی سطرهای گزارش های دریافتی را بلافاصله بر روی دیسک می نویسد (**write** می کند). به مجرد اینکه تمامی سطرهای گزارش بر روی دیسک نوشته شدند، پایگاه های داده هماهنگ سازی می شوند. مادام اینکه ارتباط بین **partner** ها برقرار است، پایگاه های داده هماهنگ باقی می مانند. **توجه:** جهت نظارت بر تغییراتی که در وضعیت یک **session** قرینه سازی پایگاه داده رخ می دهد، کافی است از کلاس **event Database Mirroring State Change** " استفاده کنید.

پس از اینکه هماهنگ سازی پایان می یابد، تمامی تراکنش هایی که بر روی پایگاه داده ی اصلی تایید ثبت (**commit**) شده اند بر روی سرور قرینه (**mirror**) نیز تایید ثبت می شوند که حفظ تمامیت اطلاعات را به ارمغان می آورد. برای انجام این کار باید ابتدا منتظر شوید تا سرور اصلی از سرور جانشین یک پیغام مبنی بر اینکه سرور جانشین گزارش تراکنش را بر روی دیسک ثبت قطعی کرده، دریافت کند، سپس تراکنش را بر روی پایگاه داده ی اصلی تایید ثبت کنید.

دقت داشته باشید که این انتظار میزان نهفتگی (**latency**) تراکنش را افزایش می دهد.

مدت زمان لازم برای هماهنگ سازی به میزان عقب افتادگی پایگاه داده ی قرینه از پایگاه داده ی اصلی به هنگام آغاز شدن **session** (که بر مبنای تعداد سطرهای گزارشی که در ابتدا از سرور اصلی دریافت شده سنجیده می شود) و بار کاری پایگاه داده ی اصلی و نیز سرعت سیستم سرور جانشین بستگی دارد. پس از اینکه **session** هماهنگ می شود، گزارش ثبت قطعی شده که هنوز بر روی پایگاه داده ی جانشین مجدداً انجام (**redo**) نشده، در **redo queue** باقی می ماند.

به مجرد اینکه پایگاه داده ی جانشین (با پایگاه داده ی اصلی) هماهنگ می شود، وضعیت هر دو نسخه ی پایگاه داده به **SYNCHRONIZED** تغییر می یابد.

نگهداشت عملیات همزمان به صورت زیر انجام می پذیرد:

با دریافت تراکنش از یک سرویس گیرنده، سرور اصلی گزارشات تراکنش دریافتی را در **transaction log** ثبت (**write**) می کند.

سرور اصلی تراکنش را در پایگاه داده ی مربوطه ثبت کرده و به موازات آن (همزمان با آن)، سطرهای گزارش (**log record**) را به سرور جانشین ارسال می کند. سرور اصلی پیش از تایید کردن هر یک از موارد زیر برای سرویس گیرنده، منتظر دریافت یک تأییدیه از سرور جانشین می ماند: یک تراکنش که باید تایید ثبت شود و یا یک بازگشت به حالت اولیه (**rollback**).

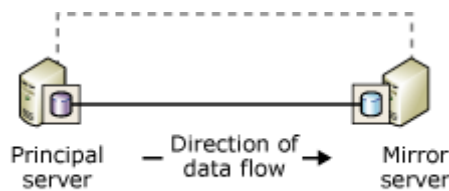
سرور جانشین گزارشات را بر روی دیسک ثبت قطعی (**harden**) کرده، در پی آن یک تأییدیه به سرور اصلی بازمی گرداند.

به محض دریافت تصدیق (**acknowledgement**) از سرور جانشین، سرور اصلی یک تأییدیه به سرویس گیرنده ارسال می نماید.

حالت **high-safety** با هماهنگ سازی اطلاعات بین هر دو مکان، از داده های شما محافظت کرده و تمامیت آن ها را تضمین می کند. کلیه ی تراکنش های تایید ثبت شده، به طور قطع بر روی دیسک سرویس دهنده ی قرینه (**mirror server**) ثبت می شوند.

حالت امنیه بهینه (**high-safety**) بدون **automatic failover**

نگاره ی زیر پیکربندی حالت **high-safety** را بدون **automatic failover** به نمایش می گذارد. همان طور که مشاهده می کنید پیکربندی متشکل از تنها دو همراه (**partner**) است.



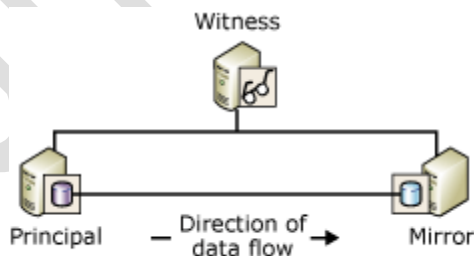
Transaction safety is FULL.

زمانی که **partner** ها به یکدیگر متصل هستند و پایگاه داده نیز از پیش هماهنگ سازی شده، در این حالت **manual failover** پشتیبانی می شود. در صورت از کار افتادگی نمونه ی سرور جانشین، نمونه ی سرور اصلی از این اتفاق هیچ تاثیری نپذیرفته و البته بدون قرینه سازی داده ها (**mirroring data**) همچنان پابرجا می ماند. حال چنانچه سرور اصلی از دست برود، سرور جانشین به حالت تعلیق (**suspend**) درآمده، ولی امکان تحمیل سرویس (**forced service**) به سرور جانشین (با احتمال از دست رفت اطلاعات) همواره وجود دارد.

حالت امنیه بهینه (high-safety) با **automatic failover**

automatic failover با فراهم آوردن امکان سرویس دادن به پایگاه داده حتی پس از از کار افتادگی یکی از سرورها، قابلیت دسترس پذیری بالا را به ارمغان می آورد. **automatic failover** لازمه ی آن است که **session** دارای یک نمونه ی سرور سوم باشد، منظور همان سرور **witness** است که در حالت ایده آل بر روی یک رایانه ی مجزا (سومی) مستقر می شود.

نگاره ی زیر پیکربندی یک **session** حالت امنیت بهینه را نمایش می دهد که از **automatic failover** پشتیبانی می کند.



Transaction safety is FULL.

بر خلاف دو سرور همراه (**partner**)، **witness** به پایگاه داده سرویس نمی دهد. تنها وظیفه ای که سرور نام برده بر عهده دارد، کسب اطمینان از عملکرد صحیح و فعال بودن سرور اصلی می باشد. سرور جانشین

تنها در صورتی که هر دو سرور **witness** و **mirror** حتی پس از قطع اتصال آن ها از سرور اصلی (**principal**)، ارتباط خود را با یکدیگر حفظ کرده اند، **automatic failover** را راه اندازی می کند.

در صورت انتخاب و تنظیم **witness**، **session** به یک **quorum** (حد نصاب) نیاز دارد. یک ارتباط که بین حداقل دو نمونه ی سرور برقرار می شود و امکان دسترسی پذیری پایگاه داده را فراهم می آورد.

برای اینکه زمینه ی راه اندازی **automatic failover** مهیا شود، لازم است شرایط زیر برقرار باشند:

پایگاه داده باید از پیش هماهنگ سازی (**synch**) شده باشد.

خرابی زمانی رخ دهد که هر سه نمونه ی سرور متصل بوده، و سرور **witness** و **mirror** همواره ارتباط خود را با یکدیگر حفظ کنند.

از دست رفت یکی از سرورهای همراه (**partner**)، نتیجه ی زیر را به دنبال می آورد:

اگر سرور اصلی تحت شرایط ذکر شده در بخش بالا از کار بیافتد، **automatic failover** رخ می دهد. سرور جانشین نقش خود را به نقش سرور اصلی تغییر داده (نقش **principal** را ایفا می کند) و پایگاه داده ی خود را به عنوان پایگاه داده ی اصلی ارائه می دهد.

اگر سرور اصلی هنگامی از دسترس خارج شود که هیچ یک از شرایط نام برده برآورده نشده است، استفاده از سرویس اجباری (**forced service**) امکان پذیر می باشد.

چنانچه تنها سرور جانشین (**mirror**) از دسترس خارج شده یا با خرابی مواجه شود، سرور اصلی (**principal**) و **witness** همچنان به کار خود ادامه می دهند.

اگر **session** ارتباط خود با **witness** را از دست بدهد، **quorum** (حد نصاب) مجاب می کند حداقل دو سرور همراه اتصال خود با **session** را حفظ کنند. چنانچه هر یک از دو سرور همراه از دسترس خارج شده و حد نصاب را نقض کنند، پایگاه داده نیز در پی آن از دسترس خارج گشته و تا زمانی که حد نصاب برقرار نشده قابل استفاده نمی باشد. این حد نصاب باعث می شود که در صورت غیاب **witness**، پایگاه داده هیچگاه به صورت **exposed** اجرا نشود (یعنی هیچگاه بدون اینکه قرینه سازی شده/از آن رونوشت و پشتیبان گرفته شود اجرا گردد).

توجه: اگر شما از قبل اطلاع دارید که امکان دارد **witness** تا طولانی مدت قطع باقی بماند، در این صورت به شما توصیه می کنیم **witness** را به طور کلی از **session** حذف کنید تا اینکه دوباره در دسترس قرار گیرد.

تنظیمات Transact-SQL و حالت های اجرایی قرینه سازی پایگاه داده

این بخش یک **mirroring session** را از نظر تنظیمات ALTER DATABASE و حالت های (state)

مختلف پایگاه داده ی قرینه سازی شده (mirrored database) و همچنین witness (در صورت وجود

آن) توصیف می کند. مطالب این بخش ویژه ی کاربرانی تنظیم شده که بجای SQL Server

Management Studio، منحصر یا عمدتا از Transact-SQL برای مدیریت قرینه سازی پایگاه داده ی

خود بهره می گیرند.

نکته: بجای استفاده از Transact-SQL، می توانید با استفاده از صفحه ی Mirroring پنجره ی محاوره ی

Database Properties، حالت اجرایی session را در Object Explorer تنظیم و مدیریت کنید.

چگونه امنیت تراکنش و حالت سرور witness بر روی حالت اجرایی تاثیر می گذارد

حالت اجرایی یک session توسط ترکیبی متشکل از تنظیمات امنیت تراکنش (transaction safety

settings) و وضعیت سرور witness (witness state) تعیین می شود. مالک پایگاه می تواند هر زمانی

که لازم دانست سطح امنیت تراکنش را تغییر دهد، و در صورت لزوم سرور witness را به session اضافه

کرده یا از آن حذف کند.

امنیت تراکنش

امنیت تراکنش، یک خاصیت پایگاه داده ی مختص به قرینه سازی (mirroring-specific property) است

که تعیین می کند آیا session به صورت همزمان اجرا شود یا غیرهمزمان. در کل دو سطح امنیتی وجود

دارد که عبارتند از:

FULL

OFF

SAFETY FULL

حالت full باعث می شود session به صورت غیرهمزمان و در حالت عملیاتی امنیت بهینه (high-safety)

اجرا شود. در صورت وجود سرور witness، session از automatic failover نیز پشتیبانی می کند.

هنگامی که شما با استفاده از دستور **session ALTER DATABASE** را برقرار می کنید، **session** در حالی آغاز می گردد که خاصیت **SAFETY** بر روی مقدار **FULL** تنظیم شده است. به عبارت دیگر **session** در حالت **high-safety** بالا می آید. پس از شروع **session**، می توانید **witness** را به آن اضافه کنید.

SAFETY OFF

غیر فعال کردن **transaction safety** (تنظیم آن بر روی مقدار **OFF**)، باعث می شود **session** به صورت غیرهمزمان و در حالت کارایی بهینه (**high-performance**) اجرا شود. چنانچه خاصیت **SAFETY** بر روی **OFF** تنظیم شود، در آن صورت خاصیت **WITNESS** نیز باید بر روی **OFF** تنظیم شود (حالت پیش فرض).

تنظیمات مربوطه به امنیت تراکنش پایگاه داده بر روی هر سرور همراه (**partner**)، در **catalog view**

sys.database_mirroring و داخل ستون های **mirroring_safety_level** و **mirroring_safety_level_desc** ضبط می شود.

حالت های مختلف witness

در صورت استفاده از **witness**، حد نصاب (**quorum**) باید رعایت شود. از این رو، حالت **witness** از اهمیت بالایی برخوردار است.

Witness دارای دو حالت زیر می باشد:

زمانی که **witness** به یکی از سرورهای همراه متصل است، می گوئیم **witness** نسبت به آن سرور در حالت **CONNECTED** قرار داشته و حدنصاب را با وجود آن سرور تکمیل می کند. در چنین شرایطی، پایگاه داده ی مورد نظر با وجود عدم دسترسی به یکی از دو سرور همراه (**partner**)، باز هم در دسترس خواهد بود.

در شرایطی که **witness** وجود دارد اما در حال حاضر به یک سرور همراه متصل نمی باشد، می گوئیم **witness** در رابطه با آن سرور (**partner**) در وضعیت **DISCONNECTED** یا **UNKNOWN** به سر می برد. در چنین مواردی، **witness** حد نصاب را با آن سرور رعایت نکرده و اگر دو سرور همراه نیز به یکدیگر متصل نباشند، پایگاه داده کاملاً از دسترس خارج می گردد.

وضعیت هر **witness** بر روی نمونه سرور در **catalog view**، **sys.database_mirroring** و داخل ستون های **mirroring_witness_state** و **mirroring_witness_state_desc** ضبط می شود. جدول زیر به طور خلاصه شرح می دهد چگونه حالت اجرایی یک **session** به تنظیمات امنیت تراکنش و وضعیت **witness** بستگی دارد:

وضعیت witness	امنیت تراکنش	حالت اجرایی
NULL (witness بدون) ²	OFF	حالت کارایی بهینه (High-performance)
NULL (witness بدون)	FULL	حالت امنیت بهینه (High-safety) بدون automatic failover
CONNECTED	FULL	حالت امنیت بهینه با automatic failover ¹

در صورت قطع اتصال **witness** از **session**، توصیه می کنیم **WITNESS** را بر روی **OFF** تنظیم کنید تا زمانی که نمونه ی سرور مورد نظر دوباره در دسترس قرار بگیرد. ²در صورتی که **witness** حاضر بوده و در حالت اجرایی **high-performance** فعال می باشد، می توان گفت که **witness** در **session** هیچ مشارکتی ندارد. با این حال، جهت در دسترس گذاشتن پایگاه داده، حداقل دو نمونه ی سرور باید به **session** مربوطه متصل باشند. از این رو پیشنهاد می کنیم خاصیت **WITNESS** را در **session** های **high-performance** بر روی **OFF** تنظیم کرده و در همین حالت باقی بگذارید.

مشاهده ی تنظیمات مربوطه به **safety** و حالت سرور **witness**

به منظور مشاهده ی تنظیمات مربوط به امنیت و حالت **witness**، کافی است **sys.database_mirroring** را مورد استفاده قرار دهید. ستون های مربوطه به شرح زیر می باشند:

شرح	ستون	عامل
تنظیمات transaction safety ویژه ی بروز رسانی ها بر روی سرور	mirroring_safety_level or mirroring_safety_level_desc	امنیت تراکنش

		<p>جانشین، بر روی یکی حالات زیر قرار می گیرد:</p> <p>UNKNOWN OFF FULL</p> <p>NULL = پایگاه داده در دسترس نمی باشد.</p>
<p>آیا سرور در witness کار هست؟</p>	<p>mirroring_witness_name</p>	<p>اسم سرور پایگاه داده، mirroring witness یا NULL. NULL بیانگر این است که هیچ سرور witness ایی موجود نمی باشد.</p>
<p>وضعیت witness</p>	<p>mirroring_witness_state or mirroring_witness_state_desc</p>	<p>وضعیت witness در پایگاه داده ی مستقر بر روی یک سرور همراه (partner) در یکی از حالت های زیر قرار دارد:</p> <p>UNKNOWN CONNECTED DISCONNECTED</p> <p>NULL = هیچ witness ای وجود ندارد یا اینکه پایگاه داده در دسترس نمی باشد.</p>

برای مثال، می توانید کد زیر را در سرور اصلی یا جانشین وارد کنید:

```
SELECT mirroring_safety_level_desc, mirroring_witness_name, mirroring_witness_stat
```

عوامل موثر بر عملکرد و رفتار session به هنگام از کار افتادگی سرور اصلی

جدولی که در زیر مشاهده می کنید، به طور خلاصه اثر ترکیبی تنظیمات امنیت تراکنش (**transaction**

safety setting)، وضعیت پایگاه داده (**database state**) و نیز وضعیت **witness** بر **session** به مجرد

از دسترس خارج شدن پایگاه داده ی اصلی را شرح می دهد:

<p>امنیت تراکنش</p>	<p>وضعیت قرینه سازی (Mirroring state) از پایگاه داده ی جانشین</p>	<p>وضعیت witness</p>	<p>عملکرد یا رفتاری که به هنگام از کار افتادگی سرور اصلی، session از خود نشان می دهد.</p>
---------------------	---	----------------------	---

FULL	SYNCHRONIZED	CONNECTED	Automatic failover رخ می دهد.
FULL	SYNCHRONIZED	DISCONNECTED	Mirror server stops; failover is not possible, and the database cannot be made available. سرور قرینه (mirror) از کار افتاده، failover امکان پذیر نمی باشد و همچنین پایگاه داده را نمی توان در دسترس قرار داد.
OFF	SUSPENDED یا DISCONNECTED	NULL (witness وجود ندارد	می توان سرویس را به سرور قرینه به زور اعمال کرد (با احتمال از دست اطلاعات).
FULL	SYNCHRONIZING یا SUSPENDED	NULL (witness وجود ندارد	می توان سرویس را به سرور قرینه تحمیل کرد (با احتمال از دست رفت اطلاعات).