

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

Database mirroring

مدرس : مهندس افشین رفوآ

دوره آموزش MVC

Database mirroring (قرینه سازی/پشتیبان گیری و ایجاد یک رونوشت عینی از پایگاه

داده)

عبارت است از ایجاد رونوشت یا کپی هایی اضافه بر سازمان از یک پایگاه داده و نگهداشت آن ها. مقصود از انجام این کار، کسب اطمینان از قابل دسترس بودن مداوم داده ها و کاهش یا به حداقل رساندن زمان از کار افتادگی (downtime) است که ممکن است از خرابی داده ها یا از دست رفتگی آن ها و نیز از شرایطی که در آن عملیات یک شبکه به صورت جزئی دچار اختلال شده، نشات گرفته شود. **Mirroring** (قرینه سازی) به ازای هر پایگاه داده یک رونوشت عینی ایجاد می کند (مختص هر پایگاه داده جداگانه پیاده سازی می شود) و تنها با آن دسته از پایگاه داده هایی کار می کند که از مدل **full recovery** (بازیابی کامل) بهره می گیرند.

مزایای Database mirroring

میزان دسترس پذیری یک بانک اطلاعاتی را افزایش می دهد.

در صورت بروز یک اشتباه غیر قابل پوشش (disaster)، در **high-safety mode** و با بهره گیری از مکانیزم failover خودکار (یک روش نگهداشت و محافظت از داده از خرابی و شکست ها که در آن سیستم ذخیره به مجرد از کار افتادن سیستم اصلی، جایگزین شده و عملیات را بدست می گیرد)، کپی یا نسخه ی ذخیره از پایگاه داده را بالا آورده و از این طریق مانع از دست رفت داده

می گردد. در دیگر **operating mode** ها (حالت عملیاتی)، مدیر پایگاه داده (**database administrator**) این گزینه را پیش رو دارد که البته با کمی ریسک از دست رفت اطلاعات، کاری کند که نسخه ی ذخیره ی به زور بالا آید.

حفاظت اطلاعات را افزایش می دهد.

Database mirroring بسته به حالت عملیاتی، **high-safety** (حالت امنیت بهینه) یا **high-performance** (کارایی بهینه)، **redundancy** یا افزونگی از اطلاعات (نسخه های تکراری و اضافه بر سازمان از داده ها) فراهم می نماید.

سرور همراه **Database mirroring (partner)** که بر روی **SQL Server 2008** یا ویرایش های جدیدتر **run** می شود اتوماتیک وار سعی می کند خطاهایی را که مانع از خوانده شدن صفحه ی دربردارنده ی اطلاعات (**data page**) می شود، برطرف سازد. سرور همراه ای (**partner**) که قادر به خواندن صفحه ی دلخواه نیست، درخواست نسخه ی جدید را از **partner** دیگر می کند. چنانچه این درخواست (**request**) با موفقیت اجرا شود، کپی درخواست شده جایگزین صفحه ای که قابلیت خواندن آن وجود ندارد می شود که در نهایت باعث رفع مشکل می گردد.

قابلیت دسترس پذیری **database production** (پایگاه داده ای که به منظور پردازش روزانه ی تراکنش ها بکار می رود. یک **database production** دربردارنده ی داده هایی است که برای انجام **task** های تولیدی مانند ایجاد و بروز رسانی قابلیت ها بکار می رود.) را به خصوص در زمان بروز رسانی و ارتقا بالا می برد.

به منظور کاهش مدت زمان از کار افتادگی یک پایگاه داده که از آن رونوشت عینی گرفته شده (قرینه سازی شده)، می توان به ترتیب و پی در پی نمونه های (**instance**) **SQL Server** که میزبانی **failover partner** ها (سرورهایی که در صورت از کار افتادگی یکی برای بالا آوردن پایگاه داده به دیگری روی می آورد) را بر عهده دارند، بروز رسانی کرد. در این صورت مدت زمان از کار افتادگی تنها به مدت زمانی که برای اجرای یک **failover** لازم است محدود می شود. این شکل از بروز رسانی تحت عنوان **rolling upgrade** (بروز رسانی تدریجی) نیز شناخته می شود.

واژه ها و تعاریف تخصصی مربوط به مفهوم Database mirroring

automatic failover: فرایندی است که طی آن به مجرد از کار افتادگی سرویس دهنده ی اصلی (**principal server**)، **mirror server** (سرویس دهنده ی جانشین یا قرینه) نقش سرور اصلی را بر عهده گرفته و نسخه ی پشتیبان از پایگاه داده ی مورد نظر را (به عنوان پایگاه داده ی اصلی) بالا می آورد.

failover partners: دو نمونه ی سرور (سرور اصلی یا جانشین) که به عنوان **role-switching partner** (همراه هایی که نقش هایشان را بایکدیگر تعویض می کنند و یا به عبارتی مکمل یکدیگر هستند) برای سرور قرینه سازی شده (**mirrored server**) نقش ایفا می کنند.

forced service: یک مکانیزم **failover** است که توسط مالک پایگاه داده راه اندازی می شود. این مکانیزم به محض از کار افتادگی سرور اصلی (**principal server**) به پایگاه داده ی جانشین **mirror database** (با وجود اینکه در حالت نامشخص به سر می برد) ، **service** اعمال می کند (انتقال می دهد).

High-performance mode: در این حالت **session** قرینه سازی (**session**) ای که در آن از پایگاه داده رونوشت عینی گرفته شده یا به عبارتی دیگر قرینه سازی صورت می گیرد) به طور ناهمزمان (**asynchronous**) اجرا می شود و تنها از سرور اصلی و سرور جانشین استفاده می کند. همچنین در این حالت تنها تعویض نقشی (**role-switching**) که صورت می گیرد از نوع **forced service** (با از دست رفت احتمالی اطلاعات) می باشد.

High-safety mode: **session** قرینه سازی به صورت همزمان اجرا می شود و در صورت نیاز علاوه بر سرور اصلی و جانشین یک **witness** نیز بکار می برد.

manual failover: یک مکانیزم **failover** است که در حالی که سرور اصلی هنوز فعال (پابرجا) است توسط مالک پایگاه داده راه اندازی می شود. وظیفه ی آن در انتقال **service** از پایگاه داده ی اصلی به پایگاه داده ی جانشین در زمانی که هر دو در حالت هماهنگ به سر می برند (با یکدیگر هماهنگ سازی شده اند)، خلاصه می شود.

mirror database: کپی یا نسخه ی عینی از پایگاه داده که به طور معمول کاملا با پایگاه داده ی اصلی هماهنگ سازی (**synchronize**) می شود.

mirror server: در یک پیکربندی **database mirroring**، به نمونه‌ی سرور که پایگاه داده‌ی جانشین (**mirror database**) بر روی آن مستقر می‌باشد گفته می‌شود.

principal database: در **database mirroring**، به یک پایگاه داده‌ی خواندنی-نوشتنی (**read-write**) اطلاق می‌گردد که رکورد های گزارش تراکنش (**transaction log record**) آن به نسخه‌ی عینی (**copy**) پایگاه داده‌ی مورد نظر (**mirror database**) اعمال می‌شود.

principal server: در **database mirroring**، سرور همراه ای (**partner**) که پایگاه داده‌ی آن در حال حاضر پایگاه داده‌ی اصلی (**principal database**) می‌باشد.

redo queue: سطرهای گزارش تراکنش دریافتی که در حافظه‌ی سرور جانشین در حالت انتظار (تعلیق) قرار دارند.

Role: سرور اصلی و سرور جانشین با ایفای نقش **principal** و **mirror**، همدیگر را تکمیل می‌کنند. در صورت نیاز، نقش **witness** را سرور دیگر (سوم) ایفا می‌کند.

role switching (تعویض نقش): به بر عهده گرفتن و ایفای نقش اصلی (**principal role**) توسط **mirror** گفته می‌شود.

Session: ارتباط ای که در طی قرینه سازی پایگاه داده (**mirroring**) مابین سرور اصلی، سرور جانشین و سرور **witness** (در صورت وجود آن) برقرار می‌شود.

پس از اینکه **mirroring session** شروع شده یا از سر گرفته می‌شود، به فرایندی گفته می‌شود که طی آن سطرهای گزارش (**log record**) پایگاه داده‌ی اصلی که با گذشت زمان بر روی سرور اصلی (**principal server**) جمع شده، به سرور جانشین ارسال می‌گردد. سرور جانشین نیز این سطرهای گزارش را بلافاصله بر روی حافظه‌ی خود ذخیره می‌کند تا با سرور اصلی هماهنگ گردد.

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

Witness: یک نمونه یا **instance** اختیاری که به سرور جانشین این امکان را می دهد که زمان مناسب و دقیق 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 (**high-safety**) طراحی شده است.

مروری بر Database mirroring

Database mirroring دو نسخه ی عینی (کپی) از یک پایگاه داده نگه می دارد که باید بر روی نمونه های مختلف **SQL Server Database Engine** (موتور پایگاه داده ی **SQL Server**) مستقر شوند. به طور معمول، این نمونه های سرور خود بر روی رایانه هایی که در مکان های مختلف قرار دارند، مستقر می شوند. راه اندازی و پیاده کردن **database mirroring** بر روی یک پایگاه داده، باعث ایجاد یک رابطه (**relationship**) بین نمونه های مختلف سرور می شود. این ارتباط تحت عنوان **database mirroring session** (قرینه سازی از پایگاه داده) شناخته می شود.

یک سرویس دهنده یا سرور نمونه است که پایگاه داده را به سرویس گیرنده ها یا کلاینت ها ارائه می دهد، این سرویس دهنده **principal server** (سرور اصلی) اطلاق می گردد. نمونه ی دیگری نیز وجود دارد که بسته به پیکربندی و حالت (**state**) پایگاه داده به عنوان یک سرور جایگزین ذخیره ایفای نقش می کند. هنگامی که **mirroring session** پایگاه داده هماهنگ سازی می شود، **database mirroring** یک سرور جانشین آماده (**hot standby server**) ارائه می دهد که امکان اجرای مکانیزم **failover** سریع را بدون از دست رفت داده از تراکنش های اجرا و ثبت شده (**committed transaction**) فراهم می آورد. حال زمانی که **session** هماهنگ سازی (**synchronize**) نشود، **mirror server** به عنوان یک سرویس دهنده ی جایگزین آماده (اما با احتمال از دست رفت داده) قابل دسترس و استفاده می باشد.

سرورهای اصلی (**principal**) و جانشین (**mirror**) با برقراری ارتباط و همکاری با یکدیگر نقش **partner** را در یک **mirroring session** پایگاه داده ایفا می کنند. همان طور که پیش تر شرح داده شد، دو سرور نام برده یکدیگر را تکمیل می کنند: **principal role** و **mirror role**. در هر زمان معینی، یک همراه یا **partner** نقش اصلی (**principal role**) و دیگری نقش سرور جایگزین را ایفا می کند. هر همراه نیز مالک

نقش جاری خود خوانده می شود. همراهی که مالک نقش اصلی می باشد تحت عنوان **principal server** شناخته می شود و نسخه ی عینی از پایگاه داده که هم اکنون در اختیار دارد نیز پایگاه داده ی اصلی یا **principal database** تلقی می گردد. حال آن همراهی که مالک نقش جانشین یا **mirror role** می باشد، به عنوان **mirror server** (سرور جانشین) شناخته می شود و نسخه ی عینی که از پایگاه داده در دست دارد، پایگاه داده ی جانشین (**mirror database**) جاری محسوب می شود. هنگامی که **database mirroring** بر روی محیط تولیدی یا **production environment** پیاده (**deploy**) می شود، همان پایگاه داده ی اصلی (**principal database**)، **production database** (پایگاه داده ی تولیدی) محسوب می شود.

Database mirroring شامل/اجرای مجدد تمامی عملیات درج (**insert**)، بروز رسانی (**update**) و حذف (**delete**) می شود، عملیات نام برده به محض رخ دادن یا پیاده شدن بر روی پایگاه داده ی اصلی (**principal database**) در پایگاه داده ی جانشین (**mirror database**) منعکس (اعمال) می شوند. اجرای مجدد تمامی عملیات مذکور از طریق ارسال جریانی (**stream**) از سطرهای گزارش تراکنش فعال به سرور جانشین صورت می گیرد که سطرهای گزارش (**log record**) را به طور پی در پی و به ترتیب به پایگاه داده ی جانشین اعمال می کند. درست برخلاف تکثیر (**replication**) که در سطح منطقی کار می کند، **mirroring** در سطح رکوردهای گزارش فیزیکی عمل می کند. سرور اصلی جریان رکوردها یا سطرهای گزارش تراکنش (**transaction log record**) را پیش از ارسال آن ها به سرور جانشین، فشرده می سازد. این قابلیت برای نخستین بار در ویرایش 2008 پایگاه داده ی **SQL Server** معرفی شد. این فشرده سازی گزارش در تمامی **mirroring session** ها رخ می دهد.

نکته: یک نمونه ی سرور (**server instance**) قادر است در چندین **mirroring session** که به طور همزمان در حال اجرا هستند با یک یا چند همراه مختلف شرکت داشته باشد. نمونه ی سرور همچنین می تواند در برخی از **session** ها نقش همراه را بازی کرده و در برخی دیگر نقش یک **witness** را ایفا کند. نمونه ی سرور جانشین باید بر روی همان نسخه ی **SQL Server** اجرا شود که نمونه ی سرور اصلی از آن استفاده می کند.

حالات عملیاتی مختلف (operating mode)

یک **mirroring session** معمولاً به صورت همزمان یا ناهمزمان اجرا می‌شود. در عملیات (اجرایی) ناهمزمان (**asynch**) تراکنش‌ها بدون اینکه منتظر سرور جانشین بمانند تا گزارشات (**log**) را بر روی حافظه‌ی خود ثبت کند، تایید اجرا (**commit**) می‌شود که در نتیجه باعث افزایش سرعت می‌گردد. در عملیات (اجرایی) همزمان، تراکنش برای هر دو همراه (**partner**) تایید اجرا می‌شوند ولی در عین حال باعث می‌شود میزان نپفتگی یا تاخیر (**latency**) تراکنش افزایش یابد که یک نکته‌ی منفی محسوب می‌شود.

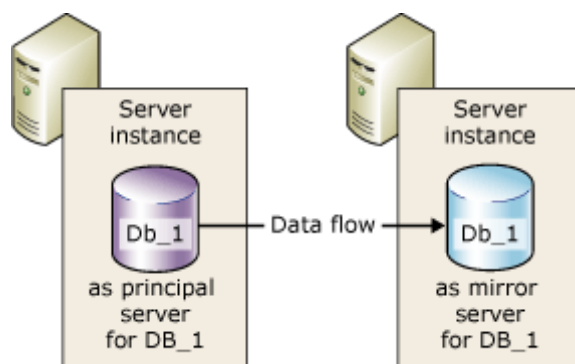
به طور کلی دو حالت عملیاتی **mirroring** وجود دارد که عبارت‌اند از: **high-safety mode** (حالت امنیت بهینه) و **high-performance mode** (حالت کارایی بهینه). **high-safety mode** از عملیات همزمان پشتیبانی می‌کند. در حالت امنیت بهینه، به محض اینکه **session** آغاز می‌شود، سرور جانشین (**mirror**) پایگاه داده‌ی جانشین را با پایگاه داده اصلی (**principal**) همزمان می‌سازد. به مجرد اینکه دو پایگاه داده هماهنگ می‌شوند، یک تراکنش برای هر دو همراه ثبت اجرا (**commit**) می‌شود. این امر یک کاستی محسوب می‌شود زیرا که اجرای همزمان تراکنش برای هر دو همراه منجر به افزایش نپفتگی یا تاخیر تراکنش می‌گردد.

دومین حالت عملیاتی، حالت کارایی بهینه (**high-performance mode**) است که به صورت ناهمزمان اجرا می‌شود. سرور جانشین سعی می‌کند با سطرهای گزارش (**log record**) ارسال شده توسط سرور اصلی هماهنگ باقی بماند. پایگاه داده‌ی جانشین ممکن است کمی از پایگاه داده‌ی اصلی عقب بیفتد. لازم به ذکر است، اختلاف زمانی بین دو سرور معمولاً بسیار محدود است. با این وجود، چنانچه سرور اصلی زیر بار بسیار سنگین کاری است و یا سیستم سرور جانشین اضافه بارگذاری (**overload**) شده، در آن صورت اختلاف بین دو سرور فاحش می‌شود.

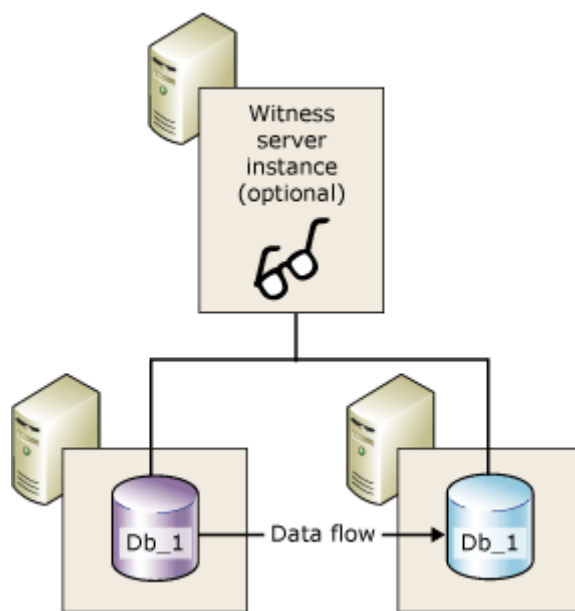
در حالت کارایی بهینه، به محض اینکه سرور اصلی یک سطر گزارش (**log record**) به سرور جانشین می‌فرستد، در پی آن سرور اصلی نیز یک تاییدیه (**confirmation**) به سرویس گیرنده (**client**) ارسال می‌کند. سرور اصلی دیگر منتظر دریافت تصدیق (**acknowledgement**) از سرور جانشین نمی‌ماند، بدین معنا که تراکنش‌ها بدون اینکه منتظر سرور جانشین شوند تا گزارشات را بر روی حافظه ذخیره کند، ثبت

اجرا می گردند (**commit** می شوند). عملیات ناهمزمان به سرور اصلی این امکان را می دهد که با حداقل یا کمترین میزان نهفتگی تراکنش اجرا شود (اما با احتمال از دست رفت جزئی اطلاعات).

تمامی **mirroring session** پایگاه داده تنها از یک سرور اصلی و یک سرور جانشین پشتیبانی می کنند. این پیکربندی در نگاره ی زیر به تصویر کشیده شده است:



حالت امنیت بهینه همراه با مکانیزم **automatic failover**، به یک سرور سومی نیاز دارد که از آن به عنوان **witness server** یاد می شود. بر خلاف دو **(principal&mirror) partner**، سرور **witness** به پایگاه داده سرویس نمی دهد. تنها کاری که این سرور می کند پشتیبانی از **automatic failover** است. **Witness** این کار را با بررسی این که آیا سرور اصلی سالم (**up**) بوده و در حال فعالیت می باشد، انجام می دهد. سرور جانشین تنها زمانی مکانیزم **automatic failover** را راه اندازی می کند که **witness** و **mirror** هر دو پس از اینکه از سرور اصلی قطع شدند، همچنان به یکدیگر متصل باقی بمانند. نگاره ی زیر یک نوع پیکربندی را نشان می دهد که در آن یک سرویس دهنده ی **witness** بکار رفته:



نکته: ایجاد یک **mirroring session** جدید یا افزودن یک سرور **witness** به پیکربندی جاری (**mirroring**) لازمه ی آن است که تمامی نمونه های سرور (**server instance**) از یک ویرایش یکسان **SQL Server** بهره بگیرند. با این حال، هنگام ارتقا به **SQL Server 2008** یا ویرایش جدیدتر، نسخه های نمونه های سرور می توانند متفاوت باشند.

Transaction safety (امنیت تراکنش) و حالت های عملیاتی

اینکه حالت عملیاتی ناهمزمان (**asynch**) است یا همزمان، همگی به تنظیمات امنیت تراکنش بستگی دارد. اگر شما منحصرا از **SQL Server Management Studio** برای پیکربندی **database mirroring** بهره بگیرید، به دنبال آن تنظیمات امنیت تراکنش نیز به هنگام انتخاب حالت عملیاتی، به صورت خودکار پیکربندی می شوند.

چنانچه از **Transact-SQL** برای پیکربندی **database mirroring** استفاده کنید، در آن صورت لازم است با نحوه ی تنظیم امنیت تراکنش نیز آشنا باشید. امنیت تراکنش (**Transaction safety**) را می توان از طریق خاصیت (**property**) **SAFETY** دستور **ALTER DATABASE** مدیریت کرد. در یک پایگاه داده که عملیات **mirroring** (گرفت رونوشت عینی) در حال پیاده شدن بر روی آن است، خاصیت **SAFETY** یا بر روی **FULL** و یا بر روی **OFF** قرار دارد (تنظیم می شود).

چنانچه گزینه ی **SAFETY** بر روی **FULL** تنظیم شده باشد، در آن صورت عملیات **database mirroring** به طور همزمان اجرا خواهد شد، البته پس از مرحله ی هماهنگ سازی اولیه. اگر یک سرور **witness** در حالت عملیاتی امنیت بهینه اجرا شود (بر روی حالت **high-safety** تنظیم شود)، **session** مربوطه نیز از **automatic failover** پشتیبانی می کند.

حال چنانچه گزینه ی **SAFETY** بر روی **OFF** تنظیم شده باشد، در آن صورت عملیات **database mirroring** به صورت ناهمزمان اجرا خواهد شد. **Session** در حالت کارایی بهینه بالا می آید (در حالت **high-safety** اجرا خواهد شد) و گزینه ی **WITNESS** نیز باید بر روی **OFF** تنظیم شده باشد.

Role-switching (تغییر نقش)

در چارچوب **database mirroring session** (گرفت یک رونوشت عینی از پایگاه داده)، نقش های **principal** و **mirror** به طور معمول قابل تعویض هستند. از این پروسه به عنوان **role-switching** یاد می شود. **Role-switching** عبارت اند از واگذار کردن یا انتقال نقش **principal** (ایفا کردن نقش سرور اصلی) به سرور **mirror** یا جانشین. در پروسه ی **role-switching**، سرور جانشین (**mirror server**) به عنوان یک **failover partner** برای سرور اصلی عمل می کند. هنگامی که یک تعویض نقش رخ می دهد، سرور جانشین ایفای نقش **principal** (سرور اصلی) را بر عهده گرفته و در پی آن نسخه ی عینی (کپی) که از پایگاه داده ذخیره دارد را بالا می آورد. سرویس دهنده ای که پیش تر نقش سرور اصلی را بر عهده داشت، در صورت موجود بودن آن، نقش سرور جانشین یا **mirror** را متقبل می شود. نقش ها می توانند بارها و بارها جای خود را بایکدیگر عوض کنند.

در کل سه نوع مختلف **role-switching** وجود دارد:

Automatic failover: نیازمند حضور سرور جانشین و یک سرور **witness** و نیز بالا آمدن سرور در حالت **high-safety** می باشد. همچنین لازمه ی آن است که پایگاه داده از پیش هماهنگ سازی (**synch**) شده و سرور **witness** نیز به سرور جانشین متصل باشد.

همان طور که پیش تر شرح داده شد، وظیفه ی سرور **witness** این است که از عملکرد صحیح (و **up** بودن) سرور همراه (**partner**) اطمینان حاصل کند. چنانچه سرور جانشین اتصال خود به سرور اصلی را از دست

بدهد، ولی در عین حال **witness** همچنان با سرور اصلی ارتباط خود را حفظ کند، در این صورت سرور جانشین دیگر مکانیزم **failover** را راه اندازی نمی کند.

Manual failover : مستلزم بالا آمدن سرور در حالت **high-safety** می باشد. سرورهای همراه (**partner**) باید همواره به یکدیگر متصل بوده، پایگاه داده نیز باید از قبل هماهنگ سازی (**synch**) شده باشد.

Forced service : احتمال از دست رفت داده در آن وجود دارد.

در حالت عملیاتی **high-performance** و نیز در حالت **high-safety** بدون **automatic failover**، اگر سرور اصلی از کار افتاده باشد ولی سرور جانشین همچنان در دسترس باشد، سرویس اجباری (**forced service**) امکان پذیر می باشد.

*** بسیار مهم ***

حالت **high-performance** گونه ای تعبیه شده که بدون **witness** اجرا شود. اما در صورت وجود یک سرور **witness**، سرویس اجباری زمانی امکان پذیر می باشد که سرور مزبور (**witness**) به سرور جانشین (**mirror**) متصل باشد.

هر سناریویی که در آن تعویض نقش صورت می گیرد، به مجرد اینکه پایگاه داده ی اصلی بالا می آید (آنلاین می شود)، برنامه های سمت سرویس گیرنده (**client application**) می توانند با اتصال مجدد به پایگاه داده خود را بازیابی (**recover**) کنند.

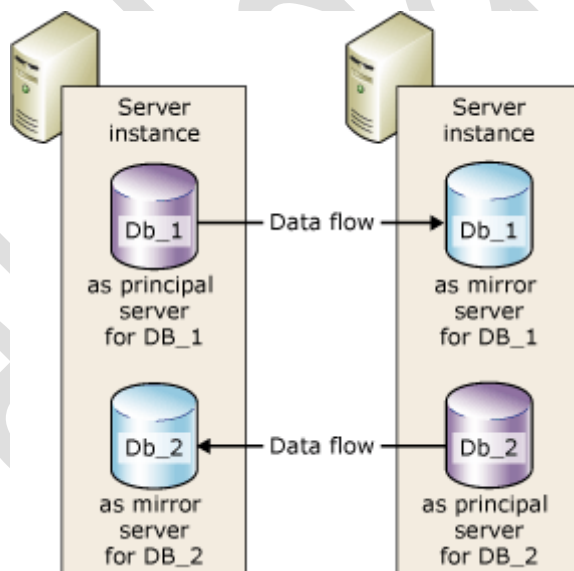
Session های همزمان

یک نمونه ی سرور قادر است در چندین **mirroring session** که به طور همزمان اجرا می شوند (یکبار به ازای هر پایگاه داده که قرینه سازی شده) با همان نمونه های سرور یا نمونه های متفاوت همکاری داشته باشد. اغلب یک نمونه ی سرور منحصر به عنوان یک **witness** یا **partner** در تمامی **mirroring session** های خود نقش ایفا می کند. اما به این خاطر که هر **session** از دیگر **session** ها کاملاً مستقل می باشد، یک نمونه ی سرور می تواند در برخی **session** ها نقش سرور **witness** و در دیگر **session** ها نقش **partner** را ایفا کند. به عنوان مثال، چهار **session** زیر را بین سه نمونه ی سرور **SSInstance_1**

SSInstance_2، SSInstance_3 and در نظر بگیرید. هر نمونه ی سرور در برخی session ها به عنوان یک partner و در برخی دیگر به عنوان witness ایفای نقش می کند:

Server instance	Session for database A	Session for database B	Session for database C	Session for database D
SSInstance_1	Witness	Partner	Partner	Partner
SSInstance_2	Partner	Witness	Partner	Partner
SSInstance_3	Partner	Partner	Witness	Witness

تصویر زیر دو نمونه ی سرور را نمایش می دهد که به عنوان partner در دو mirroring session با یکدیگر مشارکت دارند. یک session ویژه ی پایگاه داده ای به نام Db_1 می باشد و دیگری مخصوص پایگاه داده ای با نام Db_2.



هر پایگاه داده از دیگر پایگاه های داده کاملا مستقل می باشد. برای مثال، یک نمونه ی سرور ممکن است در ابتدای امر نقش mirror را در دو پایگاه داده ایفا بکند. در صورت رخداد failover برای هر یک از پایگاه های داده، نمونه ی سرور مورد نظر ممکن است برای پایگاه داده ای که failover برای آن روی داده نقش سرور اصلی را بازی کند و در همین حین نقش سرور جانشین را برای پایگاه داده ی دیگر ایفا کند.

برای ذکر مثال دیگر، یک نمونه ی سرور را در نظر بگیرید که نقش سرور اصلی را برای دو یا چندین پایگاه داده در حالت **high-safety** به همراه **automatic failover** ایفا می کند. در چنین موردی اگر که نمونه ی سرور از کار بیافتد (با خرابی مواجه شود)، کلیه ی پایگاه های داده خودکار به پایگاه های داده های جانشین مربوطه ی خود **failover** می شوند (روی می آورند).

به هنگام راه اندازی سروری که قرار است هم به عنوان **partner** و هم به عنوان **witness** انجام وظیفه کند، لازم است اطمینان حاصل کنید که گره ی آخر **mirroring session (endpoint)** از هر دو نقش پشتیبانی می کند. همچنین باید از وجود منابع کافی مطمئن شوید.

نکته: از آنجایی که پایگاه های داده ی قرینه سازی (**mirrored**) شده کاملاً از یکدیگر مجزا و مستقل می باشند، نمی توانند به صورت گروهی به جانشین مربوطه ی خود **failover** شوند (روی بیاورند).

اتصال سرویس گیرنده ها

Microsoft .NET Data Provider از قابلیت **Client-connection** (اتصال-سرویس گیرنده) برای **mirroring session** در **SQL Server** پشتیبانی می کند.

تاثیر ایجاد وقفه در یک session

مالک پایگاه داده می تواند هرگاه که مایل بود، یک **session** را متوقف سازد. ایجاد وقفه (**pause**) باعث می شود که وضعیت **session** حفظ شده و در عین حال (**mirroring**) حذف گردد. هنگامی که یک **session** متوقف می شود، سرور اصلی هیچ سطر گزارش (**log record**) جدیدی به سرور جانشین ارسال نمی کند. تمامی این سطرهای گزارش فعال باقی مانده و در گزارش تراکنش (**transaction log**) پایگاه داده ی اصلی جمع می شوند. مادام اینکه **mirroring session** در حالت توقف باقی بماند، گزارش تراکنش کوتاه (**truncate**) نمی شود. بنابراین، اگر **mirroring session** به طور طولانی مدت در حالت توقف باقی بماند، همچنان بر سطرهای گزارش (**log**) افزوده می شود.

راه اندازی session قرینه سازی (mirroring) از پایگاه داده

پیش از اینکه **mirroring session** آغاز شود، مالک پایگاه داده یا مدیر سیستم باید پایگاه داده ی جانشین (**mirror**) را ایجاد کرده، گره های پایانی (**endpoint**) و **login** آن را راه اندازی کند (به آن وارد شده) و در برخی موارد همچنین لازم است گواهینامه (**certificate**) ایجاد و پیکربندی کند.

قابلیت همکاری و همزیستی گزینه سازی با دیگر امکانات موتور پایگاه داده

Database mirroring می تواند همراه با ویژگی ها یا مولفه های **SQL Server** که زیر نام برده شده، بکار رود.

Log shipping (انتقال گزارشات به سرور دیگر)

Full-text catalogs (کاتالوگ های متن کامل)

Database snapshot (نسخه ی فوری از پایگاه داده)

Replication (تکثیر)