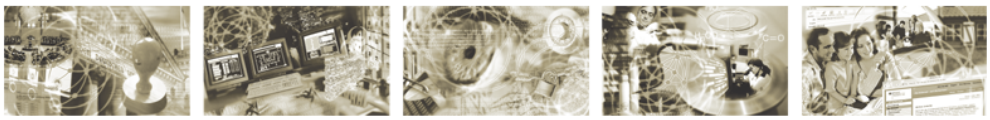
Bundesamt
für Sicherheit in der
Informationstechnik



Band M, Kapitel 8: Datenbanken

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: Hochverfuegbarkeit@bsi.bund.de
Internet: https://www.bsi.bund.de
© Bundesamt für Sicherheit in der Informationstechnik 2013

Inhaltsverzeichnis

1 Datenbanken....5
1.1 Datenbank-Design und Architektur....6
1.2 Datenhaltung....9
1.3 Datenintegrität....12
1.4 Virtualisierung....13
1.5 Überwachung....15

Tabellenverzeichnis

Tabelle 1-1: Maßnahmenkatalog Datenbanken: Datenbank-Design und Architektur....9
Tabelle 1-2: Maßnahmenkatalog Datenbanken: Datenhaltung....11
Tabelle 1-3: Maßnahmenkatalog Datenbanken: Datenintegrität....13
Tabelle 1-4: Maßnahmenkatalog Datenbanken: Virtualisierung....14
Tabelle 1-5: Maßnahmenkatalog Datenbanken: Überwachung....16

1 Datenbanken

Die nachfolgenden Maßnahmenkataloge beschreiben Maßnahmen im Sinne von Verfahren und Lösungen für die Realisierung von hoch verfügbaren Datenbanken und Datenbankmanagementsystemen. Die Darstellung der Maßnahmen erfolgt in Form von Maßnahmenclustern für die nachfolgend aufgeführten Subdomänen der HV-Domäne „Datenbanken“:

- Datenbank-Design und -Architektur
- Datenhaltung
- Datenintegrität
- Virtualisierung
- Überwachung

1.1 Datenbank-Design und Architektur

<table>
<tr><th>Nr.</th><th colspan="5">Maßnahmen</th></tr>
<tr><td rowspan="3">VM.8.1</td><td colspan="5">ANSI-SPARC 3-Schema-Konzept
Zur Verbesserung der Verfügbarkeit von Datenbanken sollte diese nach ANSI-SPARC 3-Schema konzeptioniert sein. So kann auf jeder Ebene Technik zum Einsatz kommen, die jede für sich die geforderte Verfügbarkeit garantieren und weiterhin über die komplette Kommunikations-strecke hinweg eine bedarfsgerechte Gesamtverfügbarkeit sicherstellen. Bei bestehenden Datenbanken ist zu prüfen, welche Techniken bereits zum Funktionsumfang gehören und welche zusätzlich konzipiert und umgesetzt werden müssen (Individualentwicklung).</td></tr>
<tr><td>Umsetzungsphase</td><td>Prinzip</td><td>Kriterien</td><td>wirkt gegen</td><td>Querverweis</td></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb
Notfallvorsorge</td><td>Skalierbarkeit
Redundanz
Automatismen
Separation</td><td>Fehlertolerant
Aktivierungszeit
SpoF</td><td>Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)</td></tr>
</table>

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.2</td><td>
Verteilte Datenbankmanagementsysteme (VDBMS)
Datenbankmanagementsysteme müssen auf verschiedenen und üblicherweise geografisch verteilten Rechnern laufen. Bei verteilten Systemen betrifft ein Rechnerausfall lediglich die von ihm verwaltete Datenmenge, die DB-Verarbeitung auf den übrigen Rechnern (an anderen Orten) bleibt davon unberührt. VDBMS gestatten eine Anpassung der Systemstruktur an die jeweilige Organisationsstruktur, ohne die Konsistenz der Datenbank zu beeinträchtigen. Eine weitere Steigerung der Verfügbarkeit wird erreicht, wenn Teile der Datenbank – unter Kontrolle der DBMS – repliziert gespeichert werden. Damit kann nach einem Rechnerausfall weiterhin auf die gesamte Datenbank zugegriffen werden, wenn die vom Ausfall betroffenen Daten auf einem weiteren Rechner vorliegen.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb
Notfallvorsorge</td><td>Skalierbarkeit
Redundanz
Automatismen
Separation</td><td>Fehlertolerant
Aktivierungszeit
SpoF</td><td>Naturkatastrophen
Menschliches Versagen
Sabotage, Manipulation
Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)</td></tr>
</table>
</td></tr>
</table>

Nr.	*Maßnahmen*
VM.8.3	**Mehrrechner-Datenbanksysteme (Rechner-Cluster)** Datenbanksysteme müssen als Mehrrechner-Datenbanksysteme konzipiert und betrieben werden. Mehrrechner-Datenbanksysteme zeichnen sich dadurch aus, dass mehrere Prozessoren oder DBMS-Instanzen an der Verarbeitung von DB-Operationen beteiligt sind. Der Speicher eines Mehrrechner-Datenbanksystems muss auf einer Shared-Disk-Architektur basieren. Die Shared-Disk-Architektur ermöglicht eine hohe Verfüg-barkeit bei gleichzeitiger Lastverteilung. Die Skalierbarkeit erfolgt in Mehrrechner-Datenbanksystemen auf zwei Ebenen, der Ausbau einzelner Rechner und Ausbau des Rechner-Clusters. Aus dieser Architektur ergeben sich unter Betrachtung von Verfügbarkeitsaspekten insbesondere Vorteile beim Ausfall einzelner Rechner-Cluster. Das Gesamtsystem kann auch in diesem Fall weiter betrieben werden, allerdings nur mit ein-geschränkter Performance. Diese Architektur ist nicht nur fehlertolerant gegenüber dem Ausfall einzelner Rechner, sondern erlaubt z. B. auch die Wartung einzelner Rechner bei gleichzeitiger Verfügbarkeit des Gesamtsystems. Neben diesen Verfügbarkeitsaspekten bietet diese Archi-tektur auch Vorteile hinsichtlich der Erweiterung des Gesamtsystems. Das Hinzufügen weiterer Rechner zum Cluster ist in der Regel im lau-fenden Betrieb möglich. Werden Änderungen an der Clusterverteilung durchgeführt, so können diese auch für Verbesserungen in der Lastver-teilung eingesetzt werden, um einen optimalen Betrieb zu gewährleisten.

Umsetzungsphase	**Prinzip**	**Kriterien**	**wirkt gegen**	**Querverweis**
Planung und Konzeption Beschaffung Implementierung Betrieb Notfallvorsorge	Skalierbarkeit Redundanz	Fehlertolerant Aktivierungszeit	Fehler in Hard- oder Software Geplante Ausfallzeiten	HVK (Datenbanken) VM.4.1

Nr.	*Maßnahmen*
VM.8.4	**Datenbank-Cluster** Aus mehreren aktiven Rechnerknoten (Rechner-Cluster) muss mittels einem DMBS ein Datenbank-Cluster errichtet werden. Das DMBS muss für den Cluster-Betrieb geeignete Unterstützung bieten. In dieser Betriebsart (Aktiv/Aktiv) müssen alle Knoten gleichzeitig aktiv sein. Im Normalbetrieb werden alle Knoten genutzt und die Last wird gleichmäßig über alle verteilt. Bei dem Ausfall eines Knoten wird die Last über die verbleibenden Knoten verteilt. Aufgrund dieser Architektur entstehen praktisch keine Ausfallzeiten. Die Überwachung des Clusters muss mittels spezieller Cluster-Agenten erfolgen, die in der Regel Bestandteil des DBMS sind.

Umsetzungsphase	**Prinzip**	**Kriterien**	**wirkt gegen**	**Querverweis**
Planung und Konzeption Beschaffung Implementierung Betrieb	Redundanz Automatismen	Fehlertolerant Aktivierungszeit SpoF	Fehler in Hard- oder Software Geplante Ausfallzeiten	HVK (Datenbanken) VM.4.1

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.5</td><td>
Cold-Standby Datenbank-Cluster (Failover-Cluster)
Zur Errichtung eines Cold-Standby Datenbank-Clusters muss für den Fehlerfall ein Ersatzsystem zur Verfügung gestellt werden. In dieser Betriebsart (Aktiv/Passiv, Failover-Cluster) ist nur ein Rechnerknoten im Cluster aktiv. Bei einem Ausfall des aktiven Knotens wird ein anderer (passiver) Knoten aktiv. Dabei wird der Status der Knoten überwacht und im Fehlerfall aktiviert eine geeignete Cluster-Software den zweiten Knoten. Hierzu müssen zwischen den Cluster-Knoten Statusinformationen ausgetauscht werden, die für den Betrieb des Clusters notwendig sind. Die Kommunikation muss über ein dediziertes Netzwerk verlaufen. Cold-Standby Datenbank-Cluster (Failover-Cluster) Zur Errichtung eines Cold-Standby Datenbank-Clusters muss für den Fehlerfall ein Ersatzsystem zur Verfügung gestellt werden. In dieser Betriebsart (Aktiv/Passiv, Failover-Cluster) ist nur ein Rechnerknoten im Cluster aktiv. Bei einem Ausfall des aktiven Knotens wird ein anderer (passiver) Knoten aktiv. Dabei wird der Status der Knoten überwacht und im Fehlerfall aktiviert eine geeignete Cluster-Software den zweiten Knoten. Hierzu müssen zwischen den Cluster-Knoten Statusinformationen ausgetauscht werden, die für den Betrieb des Clusters notwendig sind. Die Kommunikation muss über ein dediziertes Netzwerk verlaufen.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb</td><td>Redundanz
Automatismen</td><td>Fehlertolerant
Aktivierungszeit
SpoF</td><td>Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)
VM.4.1</td></tr>
</table>
</td></tr>
</table>

Nr.	*Maßnahmen*				
VM.8.6	**Geo-Cluster** Zur Gewährleistung der Verfügbarkeit eines Datenbank-Systems bei dem Ausfall eines Rechenzentrums (z. B. durch Naturkatastrophen oder Sabotage) müssen die Knoten eines Clusters auf verschiedene Rechenzentren geografisch verteilt werden. Dazu müssen Synchronisations- und Replikationsmechanismen eingesetzt werden, die eine geografische Verteilung von 100 km und mehr (über Kontinente hinweg) unterstützten.				
	Umsetzungsphase	**Prinzip**	**Kriterien**	**wirkt gegen**	**Querverweis**
	Planung und Konzeption Beschaffung Implementierung Betrieb Notfallvorsorge	Skalierbarkeit Redundanz Automatismen Separation	Skalierbarkeit Redundanz Automatismen Separation	Naturkatastrophen Sabotage, Manipulation Fehler in Hard- oder Software Geplante Ausfallzeiten	HVK (Datenbanken) VM.4.1

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.7</td><td>
Leseoptimierte Abfragearchitektur
Sollen Antworten eines DBMS innerhalb kürzester Zeit vorliegen, muss ein Baum-Cluster mit einem Loadbalancer und mehreren Datenbank-Instanzen realisiert werden. Änderungen finden im Vergleich zu den Leseoperationen nur selten statt und können im Zuge des Anfrage-Schedulings so mit niedrigerer, aber nicht unbestimmter Priorität bearbeitet werden. Der Schreibzugriff für Änderungen und Ergänzungen des Datenbestandes wird separat gehandhabt. Die Änderungen werden nur auf einer dedizierten Datenbank-Instanz vorgenommen und dort zwi-schengespeichert. Wenn die Anfragelast dies erlaubt, wird über einen Replikationsmechanismus der Datenstand zwischen den einzelnen Instan-zen abgeglichen. Um einen SpoF für Schreibzugriffe zu vermeiden, ist prinzipiell jede Instanz dazu in der Lage, wobei während des Betriebes nur eine Datenbankinstanz Schreibanfragen entgegen nimmt. Erst wenn alle Daten zwischen den Instanzen ausgetauscht und bestätigt sind, werden diese für die Lesezugriffe frei geschaltet.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb</td><td>Skalierbarkeit
Automatismen</td><td>Fehlertolerant
Aktivierungszeit</td><td>Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)</td></tr>
</table>
</td></tr>
</table>

Tabelle 1-1: Maßnahmenkatalog Datenbanken: Datenbank-Design und Architektur

1.2 Datenhaltung

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.8</td><td>
Datenbankspiegelung
Zur Vermeidung von Datenverlusten muss die Datenbank vollständig gespiegelt werden. In festgelegten Intervallen muss ein kompletter Abzug der Daten des DBMS auf ein (fernes) System überspielt werden. Dazu muss üblicherweise der normale Betrieb kurzzeitig unterbrochen wer-den, um die Datenintegrität zu wahren. Durch die Spiegelung wird im Wesentlichen ein normales Backup nachgebildet. Das ferne System wird in diesem Fall typischerweise als Cold-Standby betrieben und in einer Ausnahmesituation muss der Abzug zurückgespielt werden. In diesem Fall gehen maximal die geänderten Daten eines Sicherungsintervalls verloren. Daher sollte das Sicherungsintervall möglichst klein gewählt werden. Verkleinert man das Sicherungsintervall jedoch zu stark, so ist dieses Verfahren nicht mehr praktikabel, da die Datenverbindungen fast nur noch zur Übertragung der Daten verwendet werden.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb Notfallvorsorge</td><td>Redundanz
Automatismen</td><td>Fehlertolerant
Aktivierungszeit
SpoF
Reifegrad</td><td>Naturkatastrophen
Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)
CobiT, DS4</td></tr>
</table>
</td></tr>
</table>

Nr.	*Maßnahmen*
VM.8.9	**Synchrone Replikation** Die synchrone Replikation der Daten zu einem Sekundärsystem muss realisiert werden, wenn die Menge der übertragenen Daten zu minimieren ist und so zeitnah wie möglich erfolgen soll. Zu diesem Zweck muss das DBMS bei jeder atomaren Operation diese an das entfernte System weiterreichen und führt sie gleichzeitig im eigenen System aus. Das primäre (aktive) System blockiert solange, d. h. es hält eine positive Bestä-tigung so lange zurück, bis es eine positive Bestätigung vom Sekundärsystem (synchrones Commit) erhalten hat. Dieses Verfahren bietet die größtmögliche Sicherheit vor dem Verlust von Daten, da bei einem Ausfall des aktiven Systems maximal die zuletzt bearbeitete Transaktion verloren gehen kann. Andererseits kann das Verfahren der synchronen Replikation erhebliche Auswirkungen auf das Latenzzeitverhalten der Datenbankanwendung haben, insbesondere wenn zur Übertragung der Daten große Entfernungen, etwa in ein Backup-Rechenzentrum, zu überwinden sind. Wird die synchrone Replikation eingesetzt, so ist darauf zu achten, dass die Synchronität auch für nachgeordnete Redundanzhierarchien erfüllt ist. Daher sollen verschiedene Synchronisationsansätze wie ROWA, Voting oder das Primärkopien-Verfahren sicherstellen, dass eine durch-gängige Synchronisation gegeben ist.

Umsetzungsphase	**Prinzip**	**Kriterien**	**wirkt gegen**	**Querverweis**
Planung und Konzeption Beschaffung Implementierung Betrieb Notfallvorsorge	Redundanz Automatismen	Fehlertolerant Aktivierungszeit SpoF Reifegrad	Naturkatastrophen Fehler in Hard- oder Software Geplante Ausfallzeiten	HVK (Datenbanken) CobiT, DS4 VM.8.8

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.10</td><td>
Asynchrone Replikation
Soll die Replikation der Daten zu einem Sekundärsystem ohne eine Einschränkung der Performance oder der Latenzzeit erfolgen, müssen die Schreibvorgänge asynchron repliziert werden. Das primäre System arbeitet unabhängig von der Bestätigung des Sekundärsystems und kann daher ohne eine Einschränkung der Performance oder der Latenzzeit arbeiten. Da die Schreibvorgänge auf dem Sekundärsystem in der Regel zeitverzögert erfolgen, erhöht sich dadurch die Anzahl der bei Ausfall der Primär-Datenbank verlorenen Transaktionen. Neben synchroner und asynchroner Replikation existieren ebenfalls Mischrealisierungen, die beispielsweise Schreiboperationen in (schnellen) Zwischenspeichern (Caches) ablegen um diese periodisch an die Sekundärsysteme weiter zu leiten.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb
Notfallvorsorge</td><td>Redundanz
Automatismen</td><td>Fehlertolerant
Aktivierungszeit
SpoF
Reifegrad</td><td>Naturkatastrophen
Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)
CobiT, DS4 VM.8.8</td></tr>
</table>
</td></tr>
</table>

Tabelle 1-2: Maßnahmenkatalog Datenbanken: Datenhaltung

1.3 Datenintegrität

<table>
<tr><th>Nr.</th><th colspan="5">Maßnahmen</th></tr>
<tr><td>VM.8.11</td><td colspan="5">Transaktions-Recovery
Muss die die Atomarität der Transaktionen garantiert werden, muss ein Transaktions-Recovery realisiert werden. Um inkonsistente Zustände zu vermeiden, müssen alle abgebrochenen Transaktionen entfernt werden und ein Rollback muss durchgeführt werden. Ferner ist ein Zurücksetz-ten einzelner, noch nicht abgeschlossener Transaktionen im laufenden Betrieb erforderlich. Hierzu kann entweder ein vollständiges Zurücksetz-ten auf den Transaktionsbeginn oder ein partielles Zurücksetzten auf einen Rücksetzpunkt (Checkpoint) innerhalb einer Transaktion erfolgen.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung Betrieb</td><td>Redundanz</td><td>Fehlertolerant
Aktivierungszeit
Checkpoints Reifegrad</td><td>Fehler in Hard- oder Software</td><td>HVK (Datenbanken)
CobiT, DS4</td></tr>
</table></td></tr>
<tr><td>VM.8.12</td><td colspan="5">Crash-Recovery
Um zu verhindern, dass korrupte Transaktionen in die Datenbank gelangen oder dort verbleiben, muss ein Crash-Recovery unternommen wer-den. Durch ein Crash-Recovery kann versucht werden, die Atomarität und Persistenz von Transaktionen auch im Fehlerfall sicherzustellen. Dies kann zum Beispiel der Fall sein, wenn durch ein Betriebssystem- oder Anwendungsfehler eine/mehrere Transaktionen nicht vollständig ausgeführt werden konnten.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung Betrieb</td><td>Redundanz</td><td>Fehlertolerant
Aktivierungszeit
Atomarität Reifegrad</td><td>Fehler in Hard- oder Software</td><td>HVK (Datenbanken)
CobiT, DS4</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.13</td><td>
Medien-Recovery
Um die Daten nach dem Eintreten von Schadensereignissen, wie beispielsweise Hardwarefehler, Brand oder Manipulation wiederherzustellen, muss ein Medien-Recovery durchgeführt werden. Das Medien-Recovery, das auch als Langzeit-Recovery bezeichnet wird, baut auf Backupver-fahren wie „Online und Offline Backup“ auf. Wenn im Wiederherstellungsfall kein komplettes Backup zurückgespielt werden soll, sondern eine granulare Steuerung des Wiederherstellungszeitraumes möglich sein soll, müssen in regelmäßigen Abständen Checkpoints (Sicherungs-punkte) „gezogen“ werden. Dadurch gehen weniger Daten als bei einem vollständigen Rücksetzen verloren.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb</td><td>Redundanz</td><td>Fehlertolerant
Aktivierungszeit
Reifegrad</td><td>Fehler in Hard- oder Software</td><td>HVK (Datenbanken)
CobiT, DS4</td></tr>
</table>
</td></tr>
<tr><td>VM.8.14</td><td>
Katastrophen-Recovery
Um in einem Katastrophenfall die Daten wiederherstellen zukönnen, muss ein Katastrophen-Recovery durchgeführt werden. Bei der Durchfüh-rung eines Katastrophen-Recoverys wird auf eine Archivkopie zurückgegriffen, die an einem geografisch weit entfernten Ort vom Rechenzent-rum aufbewahrt wird. Ferner kann im Rahmen des Wiederanlaufs auf diese Archivkopie zurückgegriffen und die Verarbeitung in einem ande-ren Rechenzentrum fortgesetzt werden. Um ein Katastrophen-Recovery zu ermöglichen müssen die Speichermedien des Backups an einem geographisch getrennten Ort aufbewahrt werden (Auslagerung der Medien).
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb</td><td>Redundanz</td><td>Fehlertolerant
Aktivierungszeit
Distanz
Reifegrad</td><td>Naturkatastrophen
Sabotage, Manipulation
Fehler in Hard- oder Software</td><td>HVK (Datenbanken)
CobiT, DS4</td></tr>
</table>
</td></tr>
</table>

Tabelle 1-3: Maßnahmenkatalog Datenbanken: Datenintegrität

1.4 Virtualisierung

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.15</td><td>
Virtuelle DB
Eine virtuelle Datenbank auf der logischen Schicht besteht aus individuellen Tabellensätzen, Sichten und gespeicherten Prozeduren mehrerer Datenquellen. Für die aufrufende Anwendung erscheint eine virtuelle Datenbank wie eine einzelne Datenbank. Die aufrufende Anwendung kann jedoch nicht auf die zugrunde liegenden Datenquellen zugreifen, wodurch der Zugang auf andere Unternehmensdaten verhindert wird. Der Anwender erhält nur den Zugriff auf die Daten, die für die Erledigung seiner Aufgaben innerhalb des Gesamtprozesses erforderlich sind. Nach dem Prinzip der Separation werden die Daten aus dem gesamten Datenpool spezifisch für Anwender und Anwendung zugewiesen.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb
Notfallvorsorge</td><td>Skalierbarkeit
Redundanz
Automatismen
Virtualisierung</td><td>Fehlertolerant
Aktivierungszeit
SpoF</td><td>Naturkatastrophen
Sabotage, Manipulation
Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.16</td><td>
Virtuelle IP-Adresse (VIP)
Das Public Interface eines Datenbank-Clusters muss über eine virtuelle IP-Adresse angesprochen werden. Die virtuelle IP-Adresse muss mittels geeigneter Cluster-Software verwaltet werden und dient als logische Adresse für den gesamten Cluster.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung</td><td>Skalierbarkeit
Redundanz
Automatismen
Separation
Virtualisierung</td><td>Aktivierungszeit</td><td>Fehler in Hard- oder Software
Geplante Ausfallzeiten</td><td>HVK (Datenbanken)</td></tr>
</table>
</td></tr>
<tr><td>VM.8.17</td><td>
Virtuelle Verzeichnisse
Durch die Bereitstellung von Virtuellen Verzeichnissen erfolgt der Zugriff auf Ressourcen nicht über eine eindeutige Verzeichniszuordnung, sondern über einen Aliasnamen. Der Einsatz von virtuellen Verzeichnissen entkoppelt die direkte Zuordnung zwischen DBMS und Speicherort. Das DBMS greift mittels eines Aliasnamen und nicht mehr mit einer konkreten Adresse auf die Ressourcen zu. Dabei verhält sich die Darstel-lung des Aliasnamens des Verzeichnisses für das DBMS vollkommen transparent.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung</td><td>Skalierbarkeit
Automatismen
Virtualisierung</td><td>Fehlertolerant
Aktivierungszeit</td><td>Fehler in Hard- oder Software</td><td>HVK (Datenbanken)</td></tr>
</table>
</td></tr>
</table>

Tabelle 1-4: Maßnahmenkatalog Datenbanken: Virtualisierung

1.5 Überwachung

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.18</td><td>
Replikationsmonitors
Im Fall eines Datenbank-Clusters oder Standby-Systems muss die Replikation überwacht werden. Durch den Einsatz eines Replikationsmoni-tors kann die Replikation überwacht und deren Zustand ermittelt werden. Dieser Monitor überwacht permanent den Zustand des Clusters, mel-det Fehler und versucht Probleme sofort zu beseitigen, indem er, wenn möglich automatisch, Entscheidungen trifft.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb
Notfallvorsorge</td><td>Automatismen</td><td>Überwachung
Reaktionszeit</td><td>Sabotage, Manipulation
Fehler in Hard- oder Software</td><td>HVK (Datenbanken) VM.10.1</td></tr>
</table>
</td></tr>
</table>

Nr.	*Maßnahmen*
VM.8.19	**Verklemmungs-Erkennung** Zur Vermeidung von Verklemmungen und damit von Performanzeinbußen muss eine Verklemmungs-Erkennung durchgeführt werden. Auf der Basis der Sperrinformationen (global/lokal) können drei unterschiedliche Mechanismen zum Einsatz kommen, die der Erkennung von Verklemmungen dienen: • Timeout-Verfahren • Zentralisierte Verklemmungserkennung • Dezentralisierte Verklemmungserkennung

Umsetzungsphase	**Prinzip**	**Kriterien**	**wirkt gegen**	**Querverweis**
Planung und Konzeption Beschaffung Implementierung Betrieb	Automatismen	Überwachung Reaktionszeit	Fehler in Hard- oder Software	HVK (Datenbanken) VM.10.1

Nr.	*Maßnahmen*
VM.8.20	**Verklemmungs-Vermeidung** Der Ansatz der Verklemmungs-Vermeidung basiert im Wesentlichen auf Nutzung nichtsperrbasierter Synchronisationsverfahren. Hier bieten zeitstempelbasierte Verfahren oder das Verfahren der optimistischen Synchronisation Ansätze zur Verklemmungsvermeidung. Kennzeichnend für die optimistische Synchronisation ist die Aufteilung in drei Phasen: • Lesephase • Validierungsphase • Schreibphase

Umsetzungsphase	**Prinzip**	**Kriterien**	**wirkt gegen**	**Querverweis**
Planung und Konzeption Beschaffung Implementierung Betrieb	Automatismen	Überwachung Reaktionszeit	Fehler in Hard- oder Software	HVK (Datenbanken)

<table>
<tr><th>Nr.</th><th>Maßnahmen</th></tr>
<tr><td>VM.8.21</td><td>
Überwachung des DBMS
Zur Überwachung eines DBMS müssen die aktiven Mechanismen eines DBMS genutzt werden. Die ereignisbasierte Überwachung liefert de-taillierte Informationen über Datenbanken, Tabellen, Deadlocks, Table Spaces, Buffer Pools, Connections oder Transaktionen. Beispielsweise triggert jede Datenmanipulation eine Funktion, die geänderte Datensätze in eine Datei oder eine andere Datenstruktur schreibt und den Vorgang „loggt“. Ferner wird jedem Datensatz ein Zeitstempel zugeordnet, der im Fall einer Änderung auf den Zeitpunkt der Änderung gesetzt wird. An Hand der Zeitstempel kann später entschieden werden, welcher Datensatz sich nach dem Zeitpunkt der letzten Extraktion geändert hat.
<table>
<tr><th>Umsetzungsphase</th><th>Prinzip</th><th>Kriterien</th><th>wirkt gegen</th><th>Querverweis</th></tr>
<tr><td>Planung und Konzeption
Beschaffung
Implementierung
Betrieb</td><td>Automatismen</td><td>Überwachung
Reaktionszeit</td><td>Fehler in Hard- oder Software</td><td>HVK (Datenbanken)
VM.10.1
VM.10.14</td></tr>
</table>
</td></tr>
</table>

Tabelle 1-5: Maßnahmenkatalog Datenbanken: Überwachung