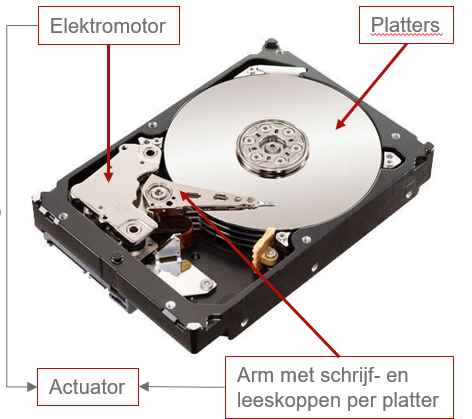
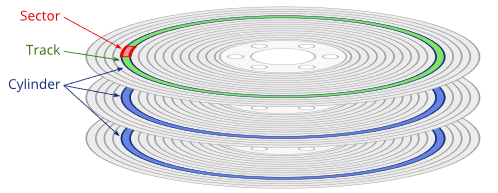
**Deel 2 - Opslag fundamentals**

**I/O Patterns**

* Werken met een database houdt in
  + Lees- en/of schrijfopdrachten van gegevens
  + Van en naar database bestanden op schijf
  + Van en naar transaction logfile op schijf
  + Van en naar data in het RAM geheugen
  + We spreken van I/O (Input/Output)
* Bedoeling is dit zo snel en efficiënt mogelijk te laten verlopen => Fundamentele kennis van opslag gerelateerde hardware noodzakelijk

**Harde schijf**

* Magnetische opslag in een luchtdichte behuizing
* Twee of meer platters (schijven) met magnetische coating
* Actuator
  + Set van schrijf- en leeskoppen op een beweegbare arm
  + Elektromotor voor aandrijving arm
* Schijven draaien rond aan bepaalde snelheid
  + Eenheid = Revolutions Per Minute (RPM)
  + Desktop/server => van 7,2K tot 15K RPM
  + Notebooks => 5,4K RPM
* Coating van platter ingedeeld in concentrische cirkels = **sporen**
* Elk spoor onderverdeeld in stukken = **sectoren**
* Sporen die loodrecht op elkaar liggen = **cilinder**
* Sporen onderverdeeld in sectoren van 512 bytes



**I/O opdracht**

* 2 manieren
  + Sequentieel
  + Random

**Sequentieel**

* Er wordt netjes van de ene sector naar de aangrenzende sector gesprongen
* Wanneer alle sectoren “verwerkt” zijn wordt naar het volgende spoor gesprongen
* Verwerking gebeurt **snel** omdat actuator weinig moet bewegen

**Random**

* Er wordt van de ene locatie naar de andere gesprongen
* Verwerking gebeurt **trager** omdat actuator regelmatig moet bewegen

**Database workloads**

* I/O opdrachten bij een database gebeuren volgens een bepaald patroon
* Zo’n patroon noemen we een database workload
* Belangrijkste database workloads
  + OLTP (Online Transaction Processing)
  + DW (Data Warehouse)
  + OLAP (OnLine Analytical Processing)
* Overige workloads
  + Log Workload, Tempdb workload,...

**OLTP (Online Transaction Processing)**

* Acties
  + Regelmatig schrijven naar database bestanden
  + Regelmatig schrijven naar transaction log bestand
  + Leesoperaties om data in RAM te brengen indien nog niet aanwezig
* Gevolg naar I/O Patterns toe
  + Leesoperaties indien van harde schijf
    - Random omwille van schrijfoperaties die zorgen voor verplaatsing actuator
  + Schrijfoperaties naar database
    - Random omwille van leesoperaties die zorgen voor verplaatsing actuator
  + Schrijfoperaties naar transaction log bestand
    - Sequentieel of random afhankelijk van aantal factoren (verderop meer)

**DW (Data Warehouse)**

* Acties
  + Grote sequentiële schrijfoperaties naar database bestanden (Dataload)
  + Schrijven naar transaction log bestand bij dataload
* Gevolg naar I/O Patterns toe
  + Schrijfoperaties naar database
    - Sequentieel want tijdens dataload meestal geen leesoperaties
  + Schrijfoperaties naar transaction log bestand
    - Sequentieel of random afhankelijk van factor opslag (verderop meer)

**OLAP (OnLine Analytical Processing)**

* Acties
  + Voornamelijk leesoperaties van cube bestanden in data warehouse
* Gevolg naar I/O Patterns toe
  + Leesoperaties van database
    - Random omwille van vele gelijktijdige leesoperaties

**Nadenken welke workload van toepassing zal zijn vooraleer SQL Server te installeren!**

**Overige scenario’s en impact op I/O Patterns**

* Meerdere databases op dezelfde SQL Server instantie

=> Zorgt voor meer Random I/O i.p.v. Sequentiële I/O

=> (Nog meer) Verspringen door lezen en/of schrijven voor meerdere databases

* Transaction log bestanden van meerdere databases op dezelfde LUN (Logical Unit Number)

=> Schrijven naar transaction log bestand is in principe Sequentiële I/O

=> Naar meerdere gelijktijdig schrijven zorgt voor Random I/O

* In geval je gebruik maakt van High Availability/Disaster Recovery features die lezen uit de transaction log (bv. Back-up restore)

=> I.p.v. schrijven (Sequentiële I/O) moet worden gelezen van de LUN

* Index creaties of onderhoud (Rebuild of Reorganize)

=> Veroorzaakt lezen en schrijven naar data bestanden

=> Zal Sequentiële I/O druk veroorzaken voor schrijven van index

* Database back-uppen

=> Veroorzaakt lezen van database- en transaction log bestanden  
 en schrijven naar back-up bestanden

=> Zal Sequentiële I/O druk veroorzaken

* Database back-up restoren

=> Veroorzaakt lezen van back-up bestanden en schrijven naar database- en  
 transaction log bestanden

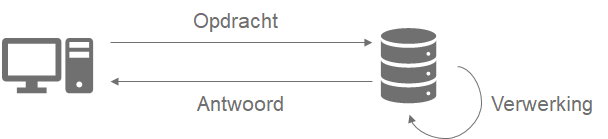
=> Zal Sequentiële I/O druk veroorzaken

**Opslag eigenschappen en performantie**

* Hardware eigenschappen hebben invloed op performantie
* Belangrijkste eigenschappen in functie van opslag
  + Latency (vertraging)
  + Throughput
  + I/O Block Size
  + IOPS

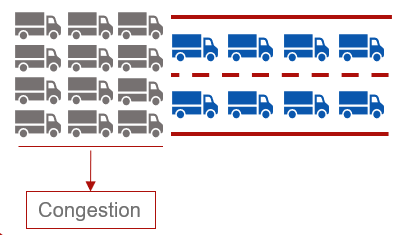
**Latency (vertraging)**

* Wachttijd nodig om I/O opdracht te volbrengen



* Hoe lager hoe beter
* Uitgedrukt in tijdseenheid of clock cycles
* Latency voor de harddisk
  + **Laag** voor Sequentiële I/O opdrachten
  + **Hoger** voor Random I/O opdrachten

**Throughput**

* Hoeveel data kan simultaan getransfereerd worden
* Hoe hoger de bandbreedte hoe meer data
* Opstopping (congestion) bij meer data dan bandbreedte

**I/O Block Size**

* **Blocking** = bundelen van gegevens (makkelijker en sneller met data werken)
* **Deblocking** = data uit een blok halen
* Data in SQL Server wordt gelezen/geschreven in blokken
* Block Size verwijst naar de hoeveelheid data vervat in 1 blok
* Kan variëren in functie van I/O operatie
* Hoge blocksize
  + Minder I/O operaties, kans op meer overtollige data
* Kleine blocksize
  + Meer I/O operaties maar minder overtollige data

**IOPS**

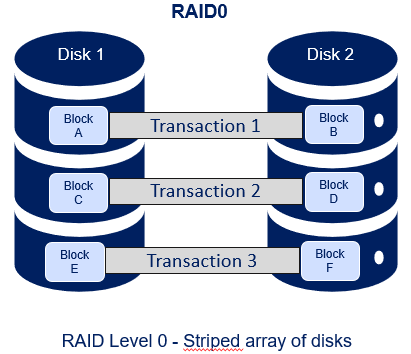
* I/O operations per second
* Alleenstaand is het een weinig zeggend gegeven
* Altijd in combinatie bekijken met de overige 3
* Wordt meestal bekeken op 4 meetpunten
  + Random read IOPS
  + Random write IOPS
  + Sequential read IOPS
  + Sequential write IOPS

**Belangrijkste karakteristieken m.b.t. database workload**

* **Online Transaction Processing (OLTP)**
  + Aantal IOPS
  + Latency
  + Throughput
* **Datawarehouse en OLAP (=Business Intelligence)**
  + Throughput
  + Latency
* Kiezen welk soort opslag te gebruiken
  + Hangt af van soort workload, performantie vereisten en beschikbaar budget
  + Bestaande infrastructuur
  + Capaciteiten/kennis van werknemers
  + Politiek

**Hardware - RAID configuraties**

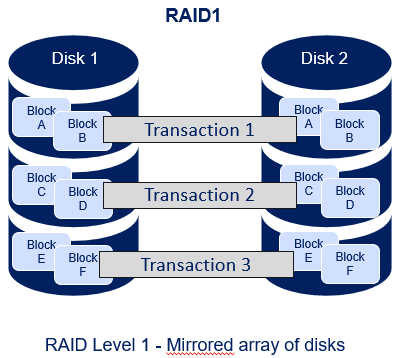
* Harde schijven kunnen af en toe crashen!
* Verhogen van snelheid/betrouwbaarheid door redundante disks toe te voegen
* Niet zomaar schijven toevoegen zonder meer
* RAID = Redundant Array of Inexpensive/Independent Disks
* Redundantie door data te spreiden over meerdere disks
* Grotere logische drives door combineren van meerdere fysische disks
* Doorstaan van faling van een disk
* Heeft minstens 2 disks nodig
* Betere I/O performantie door parallelle toegang
* Meerdere varianten
  + RAID0
  + RAID1
  + RAID5
  + (Ook RAID2,3,4,6)
* Combinaties
  + RAID01 (0+1)
  + RAID10 (1+0)
  + RAID50 (5+0)

**RAID0**

* **Striping**
* Striped array zonder fout tolerantie
* Bestaat uit n disks
* **Data wordt gelijkmatig verspreid over het aantal beschikbare disks**
* **Capaciteit = n x smin** waarbij
  + **n** = aantal disks
  + **smin** = capaciteit kleinste disk
* **Voordelen**
  + Hoge performantie
  + Simpel ontwerp
* **Nadelen**
  + Geen fout tolerantie
  + Alles weg bij crash van 1 disk
  + Betrouwbaarheid lager dan bij 1 disk
* **Gebruik**
  + Tijdelijk bewaren van data met  
    nood aan hoge bandbreedte
  + Compilatie van grote software projecten

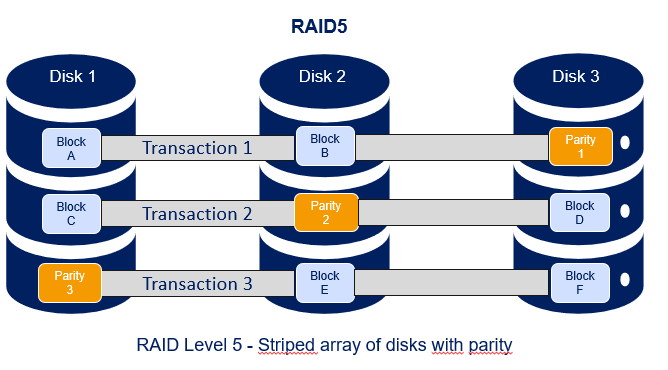
*Aantal IOPS en writes stijgt wanneer er meer disks worden toegevoegd*

**RAID1**

* **Mirroring without parity or striping**
* Bestaat uit minstens 2 disks
* **Data wordt gedupliceerd op alle disks in de array**  
  
* **Capaciteit**
  + 1x kleinste disk van de RAID array
* **Voordelen**
  + Kan overweg met falen van een disk
  + Hogere lees performantie door simultaan lezen van meerdere disks
  + Simpel ontwerp
  + Geen data reconstructie nodig
* **Nadelen**
  + Grote overhead
* **Gebruik**
  + Applicaties die hoge  
    betrouwbaarheid nodig hebben
  + Meestal gebruikt voor kleinere  
    data hoeveelheden

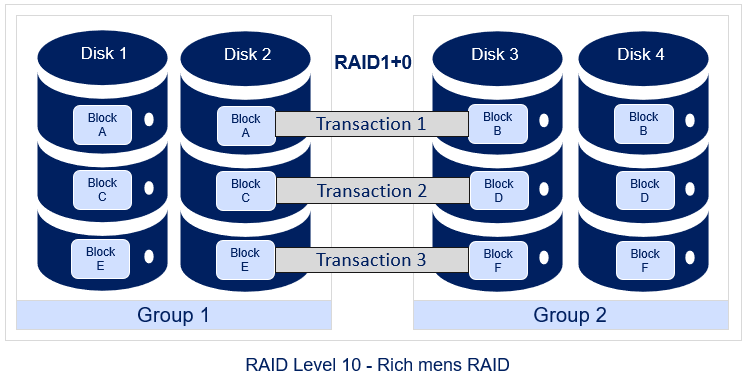
**RAID5**

* **Striping with distributed/interleaving parity**
* Bestaat uit n+1 (minimum 3) disks

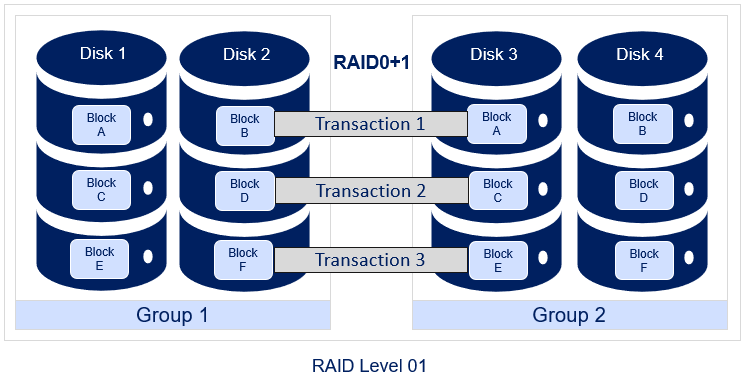


* **Data wordt verspreid over n disks (varieert van transactie tot transactie)**
* **Pariteitsinformatie wordt berekend en naar overige disk geschreven**
* **Capaciteit = (n -1) x smin**
* **Voordelen**
  + Hoge data rates
  + Weinig overhead (enkel pariteit bepalen)
  + Gelijkmatig gespreide toegang
* **Nadelen**
  + Schrijf operaties trager dan bij andere RAID configuraties
  + Disk falen heeft middelmatige impact op performantie Data terug opbouwen
  + Complex ontwerp
  + Complexer om terug op te bouwen
* **Gebruik**
  + Applicaties die hoge betrouwbaarheid nodig hebben
  + Meest veelzijdige RAID niveau
  + Meest gebruikte RAID niveau

**RAID10 (1+0)**

* **Stripeset of mirrors**
* Bestaat uit 2n (minimum 4) disks
* Voorkeursconfiguratie voor I/O intensieve applicaties zoals databases
* **Capaciteit = p x smin** (waarbij **p** aantal RAID1 pairs is)
* **Voordelen**
  + Combinatie van voordelen RAID0 (performantie) en RAID1 (redundantie)
  + Simpel ontwerp
  + Hoge data rates
  + Betere throughput (Behalve t.o.v. RAID0) en latency dan alle andere RAID levels
  + Zeer geschikt voor write-intensieve workloads
* **Nadelen**
  + Grote overhead
  + Duurder dan andere RAID configuraties
* **Gebruik**
  + Applicaties die hoge betrouwbaarheid en performantie nodig hebben

**RAID01 (0+1)**

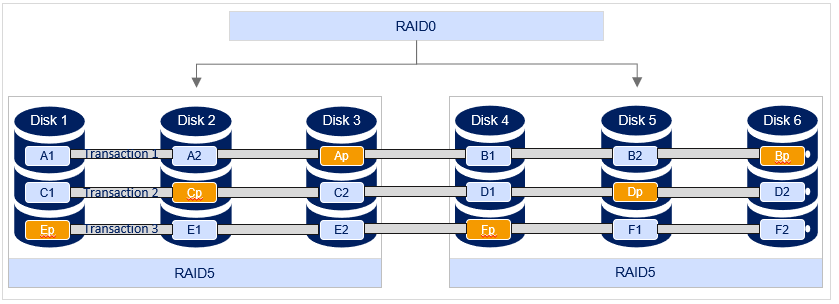
* **Mirrored stripesets**
* Bestaat uit 2n (minimum 4) disks
* **Voordelen**
  + Hoge performantie
  + Kan overweg met single disk failure (wordt dan RAID0)
  + Simpel ontwerp
* **Nadelen**
  + Grote overhead
* **Gebruik**
  + Applicaties die matige betrouwbaarheid en hoge performantie nodig hebben

**RAID10 vs RAID01**

* **Gelijk(w)aardig** op vlak van
  + Performantie
  + Opslagcapaciteit
* **Verschil** zit in fouttolerantie bij gebruik van >= 6 disks
  + Hoger in RAID10 dan in RAID01

**RAID50 (5+0)**

* **Parity and disk striping across multiple drives**
* Bestaat uit minstens twee, 3-disk RAID5 arrays
* **Splitst data op in kleinere blokken die gestriped worden over de RAID5 arrays**



* **Voordelen**
  + Betere redundantie dan RAID5
  + Betere performantie dan RAID5, vooral bij writes
  + Kan overweg met single disk failure in elke RAID5 array  
      
     Indien nog een disk uitvalt in die array valt de ganse RAID
  + Betere performantie bij verlies van 1 disk dan RAID5
  + Goed compromis tussen RAID5 en RAID10
  + Minder duur dan RAID10
* **Nadelen**
  + Duurder dan RAID5
* **Gebruik**
  + Data warehousing
  + Applicaties met hoge vereisten naar capaciteit en fout-tolerantie toe

**Aandachtspunten**

* Hou rekening met volgende punten bij bepalen RAID level
  + Het aantal spindles in een array is heel belangrijk
    - Groot aantal kleinere disks   
      => betere performantie dan klein aantal grote disks
  + RAID5 heeft een write performantie handicap
    - Kan geen verlies overleven van meer dan 1 disk in de array
    - Hoe meer disks in de array, hoe meer kans op verliezen van een disk
    - In geval budget een rol speelt, gebruik RAID5 dan om weinig gebruikte data op te zetten  
      (bv weinig veranderende data bestanden, back-ups)
  + RAID10 en 1 hebben zeer goede write performantie
    - RAID10 meer redundantie dan RAID5
    - Probeer altijd RAID1 of RAID10 te gebruiken voor log-bestanden
  + Onderhandel niet met jezelf omtrent opslag
    - Vraag altijd RAID10 aan en onderhandel dan naar beneden indien het moet
    - Gebruik RAID5 voor weinig gebruikte data bestanden en back-ups indien je geen RAID10 kan krijgen

**Overige opslag gerelateerde hardware**

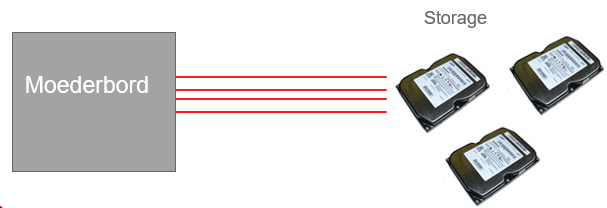
**SSD (Solid State Drive)**

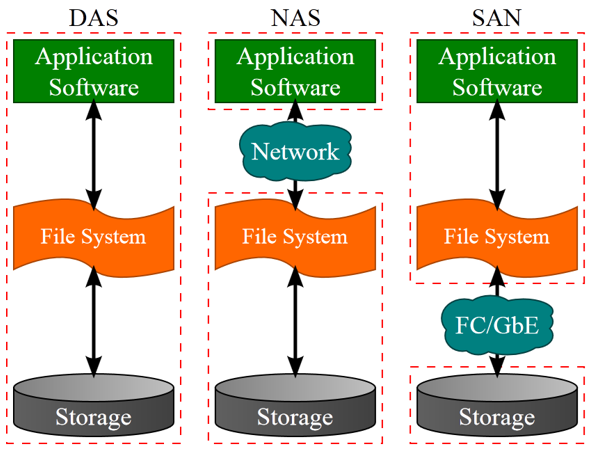
* Te gebruiken als een gewone standaard disk
* Volledig elektronisch (geen bewegende delen)
* Energiezuiniger
* Sneller dan een gewone standaard disk, vooral voor random I/O operaties
* Minder gevoelig voor schade
* Vluchtige (DRAM) en niet-vluchtige (NAND-Flash) uitvoering
* Geen fragmentatie zoals bij een gewone standaard disk
* Nadelen
  + Limiet op aantal schrijf-operaties bij niet-vluchtige uitvoering
  + Niet-vluchtige versie kan alleen maar schrijven naar lege geheugencellen
  + Geheugencellen waarvan “vermoed” wordt dat ze data bevatten moeten eerst leeggemaakt worden
  + Schrijven naar lege geheugenblokken zeer snel, maar pak trager indien een overwrite moet gebeuren

**Tips aanvragen opslagruimte**

* Vraag niet enkel opslagruimte aan op basis van opslagcapaciteit vereisten
  + Kans op minder goede performantie omdat men niet weet waarvoor het gebruikt zal worden
* Specifieer je performantie vereisten
  + Sequentiële performantie in MB of GB per seconde
  + Random performantie in IOPS
* Overweeg gebruik van “short-stroking” om I/O performantie te verbeteren
  + Bewust kleiner percentage van beschikbare opslagruimte gebruiken
  + Actuator moet minder ver bewegen om data te vinden
* Meer opslagruimte aanvragen dan wat je nodig zal hebben
  + Handig bij onderhandelingen

**Host Bus Adapter (HBA)**

* Component die voorziet in fysische verbinding tussen een computer en opslag media of een netwerk
* Voorziet ook in het verwerken van I/O i.p.v. dat CPU dit doet
* Stuurt de beweging van data aan tussen het moederbord en de disks volgens een bepaalde interface
* Belangrijk in opslag en I/O discussies vanwege potentieel verzadigd geraken
* IDE (Integrated Drive Electronics Verouderd!)
* SATA (Serial Advanced Technology Attachments Thuisgebruik)
* SCSI (Small Computer System Interface Servers)
* Firewire
* Over het netwerk via netwerk protocols
  + AoE (ATA over Ethernet)
  + iSCSI (internet Small Computer System Interface)
  + Fibre Channel (over ethernet, over IP, internet Fibre Channel)

**DAS - NAS - SAN**

**DAS (Direct Attached Storage)**

* Rechtstreeks gekoppeld aan de server

**NAS (Network Attached Storage)**

* Verbonden via het netwerk
* Technologie voor file-and-print servers
* Maakt gebruik van bestandsgeoriënteerde protocollen waarbij een deel van een bestand wordt opgevraagd i.p.v. een blok op een opslagmedium
* Niet geschikt voor SQL Server data-, transaction log en back-up bestanden

**SAN (Storage Area Network)**

* Architectuur die dient als koppeling tussen servers en opslagapparaten zodanig dat het OS denkt dat het opslagapparaat rechtstreeks is aangekoppeld
* Meestal iets trager dan DAS (intensiever gebruik door meerdere toepassingen)
* Duurder, in aanschaf en onderhoud maar zeer flexibel en onderhoudbaar  
   Beheer opslagsysteem mogelijk zonder server uit bedrijf te nemen
* Maakt meestal gebruik van het iSCSI protocol (niet de fysieke aansluiting) of Fibre Channel

**Logische drives**

* Drive arrays moeten aangemaakt en aangeboden worden aan de host (server)
* Gebruik Windows Disk Manager om drives te initialiseren en formatteren
  + Wijs drive letters en naam toe aan elke LUN
  + Gebruik hierbij je naamgeving schema
* Nodige directories op deze drives aanmaken voor de installatie
* Test de performantie van je logische drives
  + Eerst met CrystalDiskMark Snelle test, enkel als “eerste ronde” voor SQLIO
  + Daarna met SQLIO

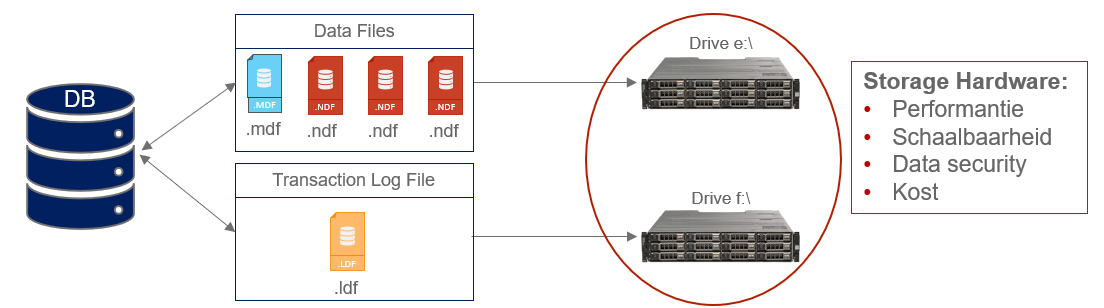
=> Voor meer gedetailleerde testen (duurt langer)

=> SQL Server moet hiervoor niet geïnstalleerd zijn

=> Gebruikt geen database specifieke workload!

**Database bestanden**

* Database in SQL Server bestaat uit >= 2 bestanden
  + MDF = Primary Data File (1x) Bevat database metadata en eigenlijke data
  + NDF = Secondary Data File Bevat eigenlijke data, meerdere bestanden kunnen bestaan
  + LDF = Transaction Log File Bevat historiek van alle data gerelateerde bewerkingen
* **Belangrijke instellingen van bestanden:**
* File type
  + ROWS Databestand (.mdf standaard aanwezig, extra bestanden .ndf)
  + LOG Transaction log bestand (Best slechts 1 per database!)
* Size
  + Initiële grootte van bestand op schijf
  + Deze ruimte wordt gealloceerd en kan dan gevuld worden
* Autogrowth
  + Geeft aan met hoeveel ruimte het bestand mag vergroot worden
  + Kan in MB of percentages ingesteld worden, inclusief limiet
* Path en filename

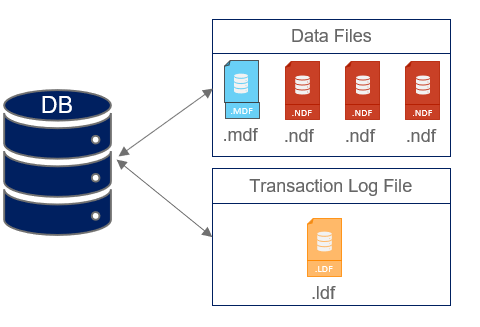


**Performance tweaks**

**I/O opsplitsen**

* I/O opsplitsen voor betere performantie en betrouwbaarheid
* Databases, transaction log, OS op aparte storage hardware plaatsen

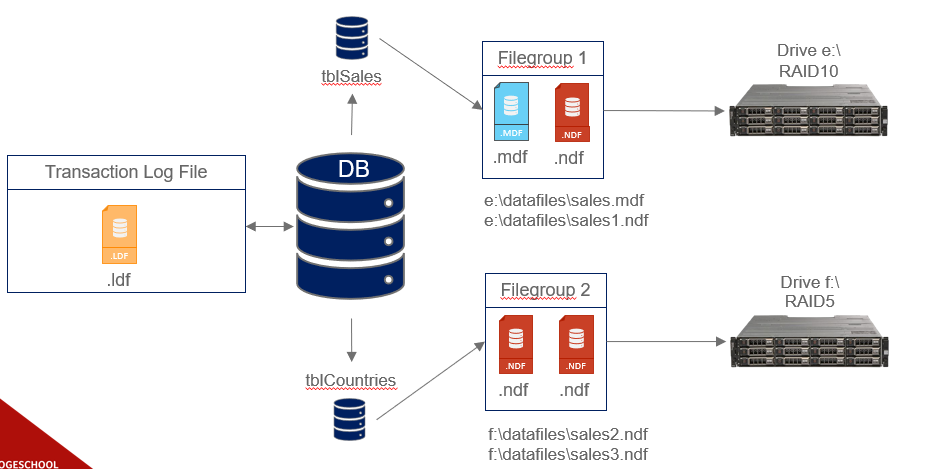
**Database bestanden**

* Hoeveel Data Files gebruiken in een database?
* Hanteer verhouding 1 datafile per 2-4 processorkernen
* SQL Server Storage Engine regelt wegschrijven van data bij gebruik meerdere files
* Schrijft data proportioneel weg over files (Proportional Fill Algorithm)
  + Niet eerst een bestand volledig vullen en dan het volgende
  + Elk bestand krijgt data (groter bestand = meer data)
* Datafile groter maken zorgt voor vertraging van database operaties
* Groei zorgt ook voor meer fragmentatie
* Settings voor filegrowth voldoende hoog instellen om frequente automatische groei te vermijden

**=> Shrinking** toepassen

* + Verplaatst volledige pagina’s aan het einde van data file(s) naar de eerste vrije ruimte.
* Beter niet toepassen indien niet noodzakelijk
  + Intensief proces naar I/O toe
  + Vergroot het transaction log bestand
  + Kan resulteren in ernstige fragmentatie van indexen waarna deze best opnieuw opgebouwd worden

**Filegroups**

* Elk databasebestand heeft locatie op schijf
* Databasebestand(en) worden toegewezen aan een filegroup
* Filegroup is logische container
* Databasebestanden met eenzelfde locatie kunnen gebundeld worden in 1 filegroup
* Tabel kan niet worden toegewezen aan databasebestand
  + Wel aan filegroup
  + Via filegroup kunnen we op die manier instellen op welke hardware tabellen worden weggeschreven
  + Te doen bij creatie tabel!
* Tabel toewijzen aan filegroup
  + Bij creatie van tabel
  + Via SQL Server Management Studio
  + Via SQL instructie

**USE MyDB**

**CREATE TABLE** MyTable

(

cola int PRIMARY KEY, colb char(8)

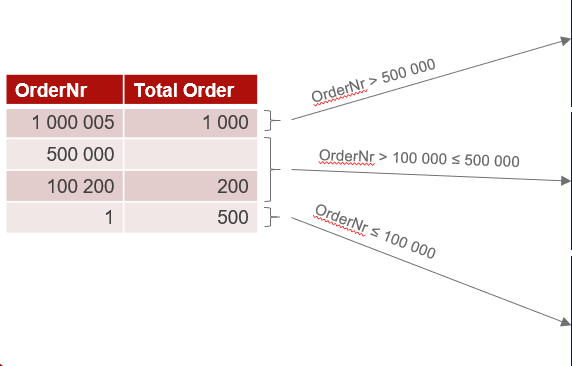
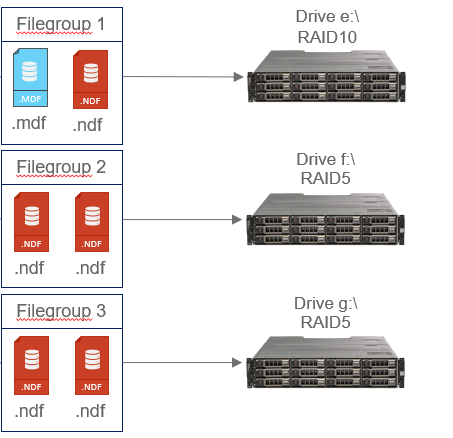
)

**ON** Filegroup1

GO

**Data partitionering**

* Inhoud van een tabel kan opgedeeld worden in stukken
* Elk stuk op zijn beurt toewijsbaar aan een filegroup
* Op basis van in te stellen regel(s), bv:
  + Datum > 01-01-2021
  + …
* SQL Server past regels toe op data om partitionering door te voeren
* Kan bv. gebruikt worden om meest recente data op performante hardware te plaatsen

**Best practises**

* Configuratie van filegroups en databasebestanden
  + Default configuratie is ok voor kleine databases
  + Tuning en aanpassing voor high-performance databases
* Databasebestanden
  + Pre-configureer de grootte van files
  + Stel waarde van autogrowth voldoende groot in
  + Niet automatisch verkleinen
  + Databasebestanden, transaction log, OS bestanden op fysisch gescheiden volumes
  + Databasebestanden en transaction log op (gescheiden) RAID volumes
  + Creëer bijkomende databasebestanden met identieke grootte
* Creëer partities op de zwaarst gebruikte tabellen/indexen
* SAN en DAS Vergelijk I/O performantie.
* Tempdb database
  + Monitor I/O
  + Tune storage
  + Creëer bijkomende databasebestanden met dezelfde bestandsgrootte
    - # Processor cores
    - < 8 # files = # cores
    - > 8 Begin met 8 tempdb data files  
       Voeg extra files in groepen van 4 toe wanneer nodig