

Databases

H1 Databanken gekaderd

H1. Databanken gekaderd

1. Wie gebruikt databanktechnologie
2. Basisbegrippen
3. Gegevensbeheer
4. Delen van een databanksysteem
5. Kenmerken van databanken en databankbeheer



1. Wie gebruikt databanktechnologie

Wie gebruikt databank(technologie)?

- Traditionele bedrijfsapplicaties (loonberekening, tijdsregistratie, ...)
- Biometrische applicaties (vingerafdrukken, resultaten scans)
- Sensor-applicaties (in kerncentrales, ...)
- GIS applicaties (geografische informatie systemen (Google Maps, ...))
- Big Data applicaties (Walmart, ...)
- 'Internet of Things (IoT)' applicaties (Telematics, ...)
- → opslag en terug ophalen van informatie (data)



2. Basisbegrippen

Basisdefinities: databank.

Een **gedeelde** verzameling van **logisch met elkaar verbonden** gegevens en hun **beschrijving**, ontworpen om aan de **informatienoden** van een organisatie te voldoen.

(T. Connolly).

- digitaal opgeslagen
- specifiek bedrijfsproces
- specifieke groep (gebruikers en applicaties)

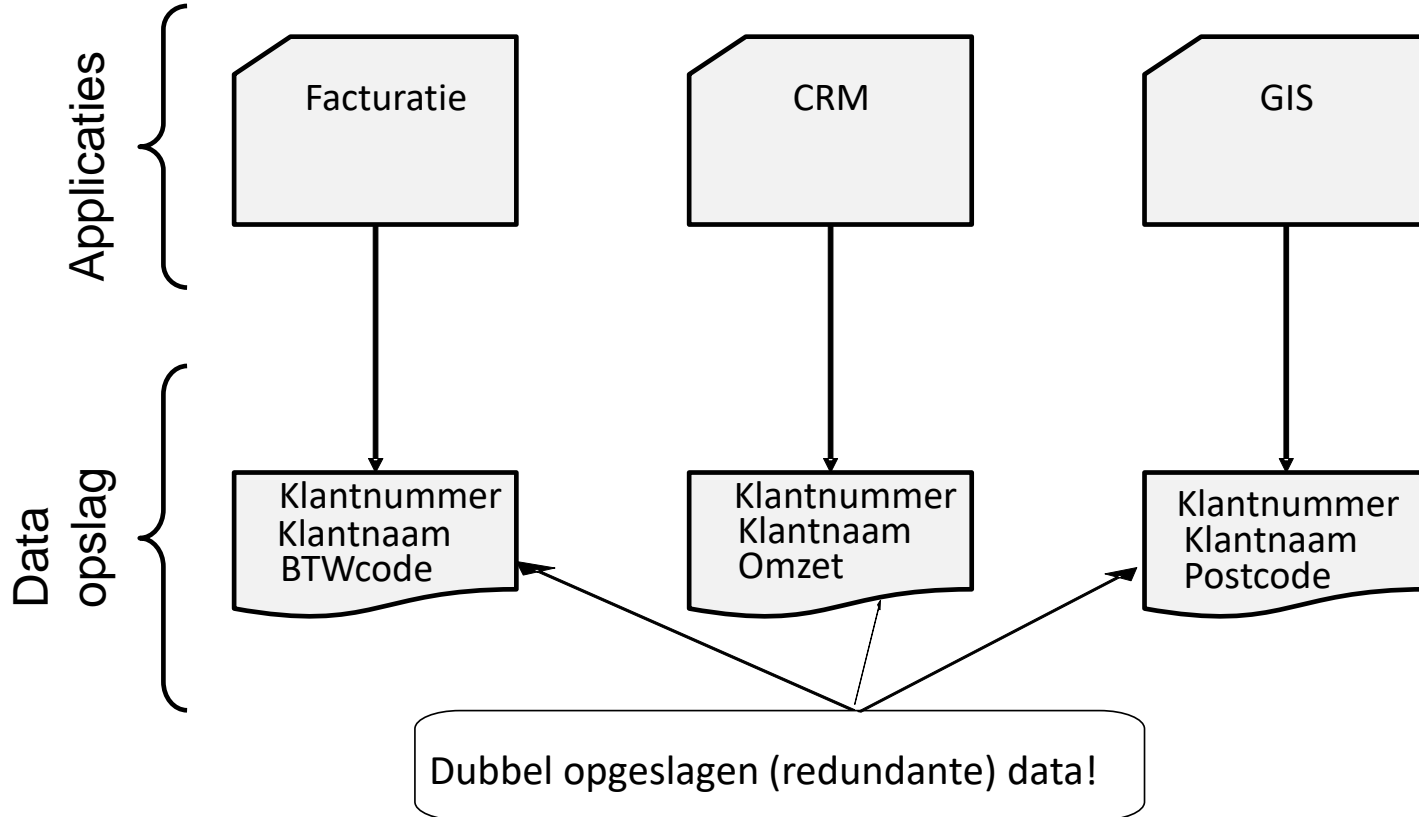
Basisdefinities: DBMS.

- Een Database Management System (DBMS):
 - een verzameling computerprogramma's (softwaremodules)
 - Nodig om een databank te
 - definiëren
 - creëren
 - wijzigen
 - beheren
 - gebruiken (gegevens invoeren en 'lezen').
- Databanksysteem: databank + DBMS



3. Gegevensbeheer

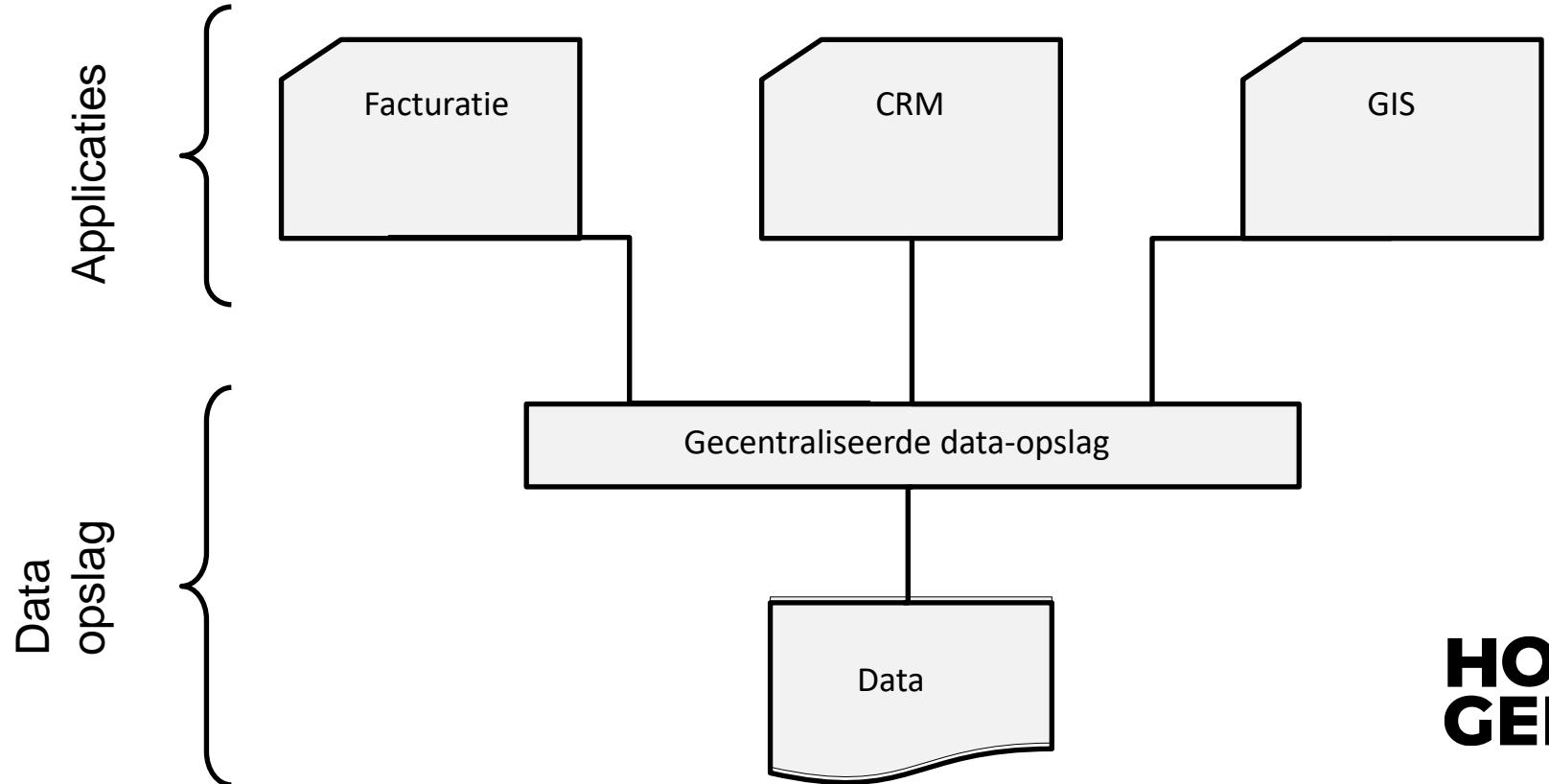
Gedecentraliseerd <-> gecentraliseerd.



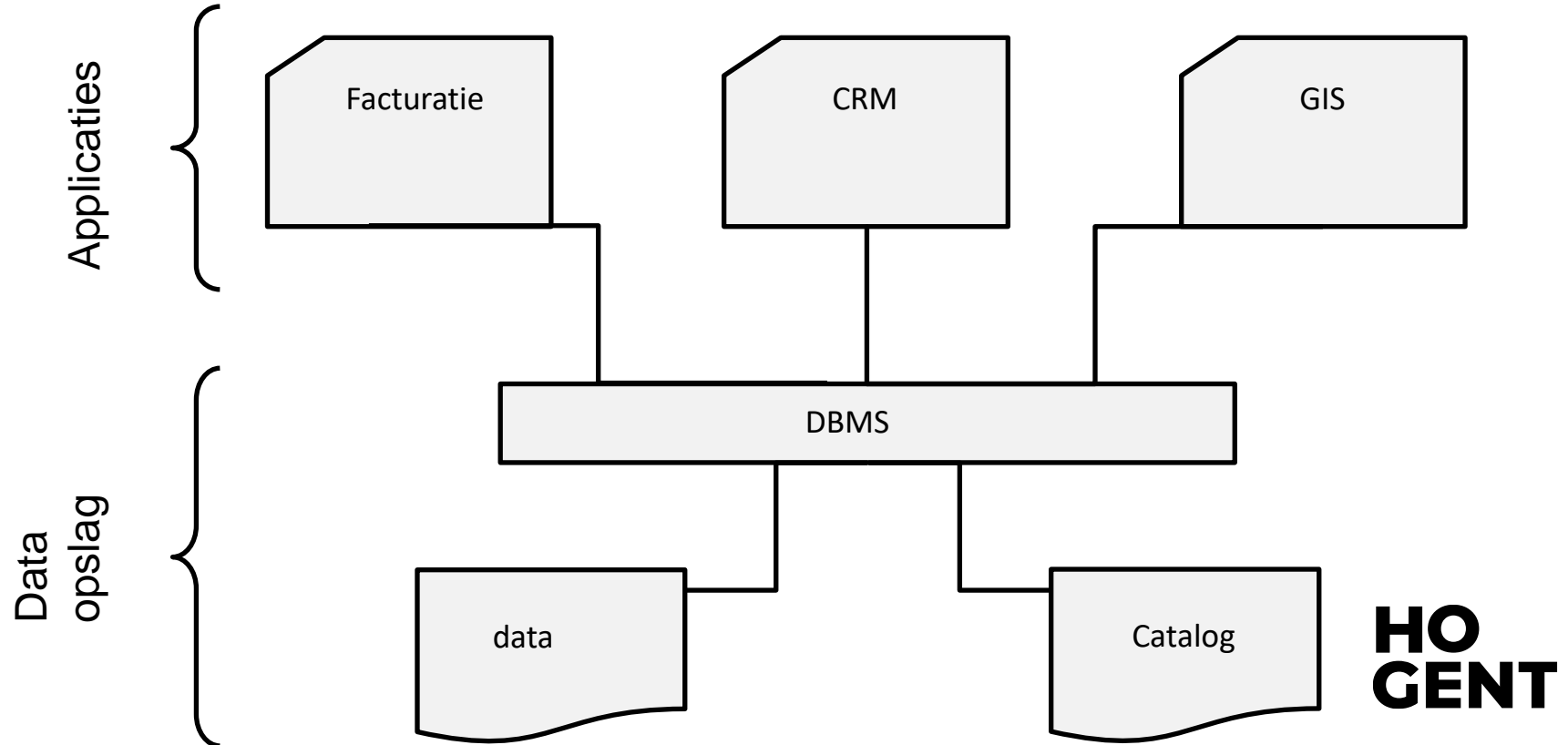
Data management.

- Gedecentraliseerde aanpak:
 - Dubbele of redundante opslag
 - Risico op inconsistente data
 - Sterke koppeling tussen applicaties en data
 - Concurrente toegang moeilijk te realiseren
 - Applicaties voor meerdere diensten/bedrijven moeilijk te realiseren

Data management.



Data management.



Data management.

- **Gecentraliseerde** aanpak:
 - Efficiënter
 - Consistente data
 - Eenvoudiger te beheren
 - Losse koppeling tussen applicaties en data
 - DBMS biedt mogelijkheden bij het concurrent 'beheren' van data

Data management.

- (Txt) Bestand
- (Relationele) Databank (SQL)

```
Procedure FindCustomer;  
begin  
  open file Customer.txt;  
  Read(Customer)  
  While not EOF(Customer)  
  If Customer.name='Bart' then  
    display(Customer);  
  EndIf  
  Read(Customer);  
EndWhile;  
End;
```

```
SELECT *  
FROM Customer  
WHERE name = 'Bart'
```



4. Delen van een databanksysteem

Elementen van een databanksysteem.

- Databankmodel versus instances
- Data model
- 3-lagen architectuur
- Catalog
- Databankgebruikers

Databankmodel versus instances.

- Databankmodel = databankschema
 - bevat
 - beschrijving van de databankstructuur
 - specificaties v/d elementen, hun eigenschappen, relaties, beperkingen, ...
 - Opgesteld tijdens databankontwerp
 - Wijzigt niet om de haverklap
 - Opgeslagen in de catalog
- Toestand van een databank
 - Op dat ogenblik aanwezige data
 - Wijzigt voortdurend

Databankmodel versus instances.

- Voorbeeld databankmodel :

Kunstenaar (naam, geboorteplaats, geboortedatum)

Kunstwerk (naam, museum, jaar)

Museum (naam, stad)

Databankmodel versus instances.

- Toestand van een databank :

MUSEUM

naam	stad
Sint-Pietersbasiliek	Rome
Museo Reina Sofia	Madrid
Gemäldealerie van de Staatliche Museen	Berlin

KUNSTENAAR

naam	geboorteplaats	geboortedatum
Michelangelo	Caprese	06/03/1475
Rembrandt	Leiden	15/07/1606
Picasso	Malaga	25/08/1881

KUNSTWERK

naam	museum	jaar
Guernica	Museo Reina Sofia	1937
Christuskop	Gemäldealerie van de Staatliche Museen	1648
Pieta	Sint-Pietersbasiliek	1499

Datamodel.

- **Gegevensmodel**: weergave van de gegevens met hun kenmerken en hun relaties
- **Conceptueel gegevensmodel**: perfecte weergave van de gegevensvereisten van de 'business'
opgesteld met de business
omgezet naar logisch en intern model

Gegevensmodel.

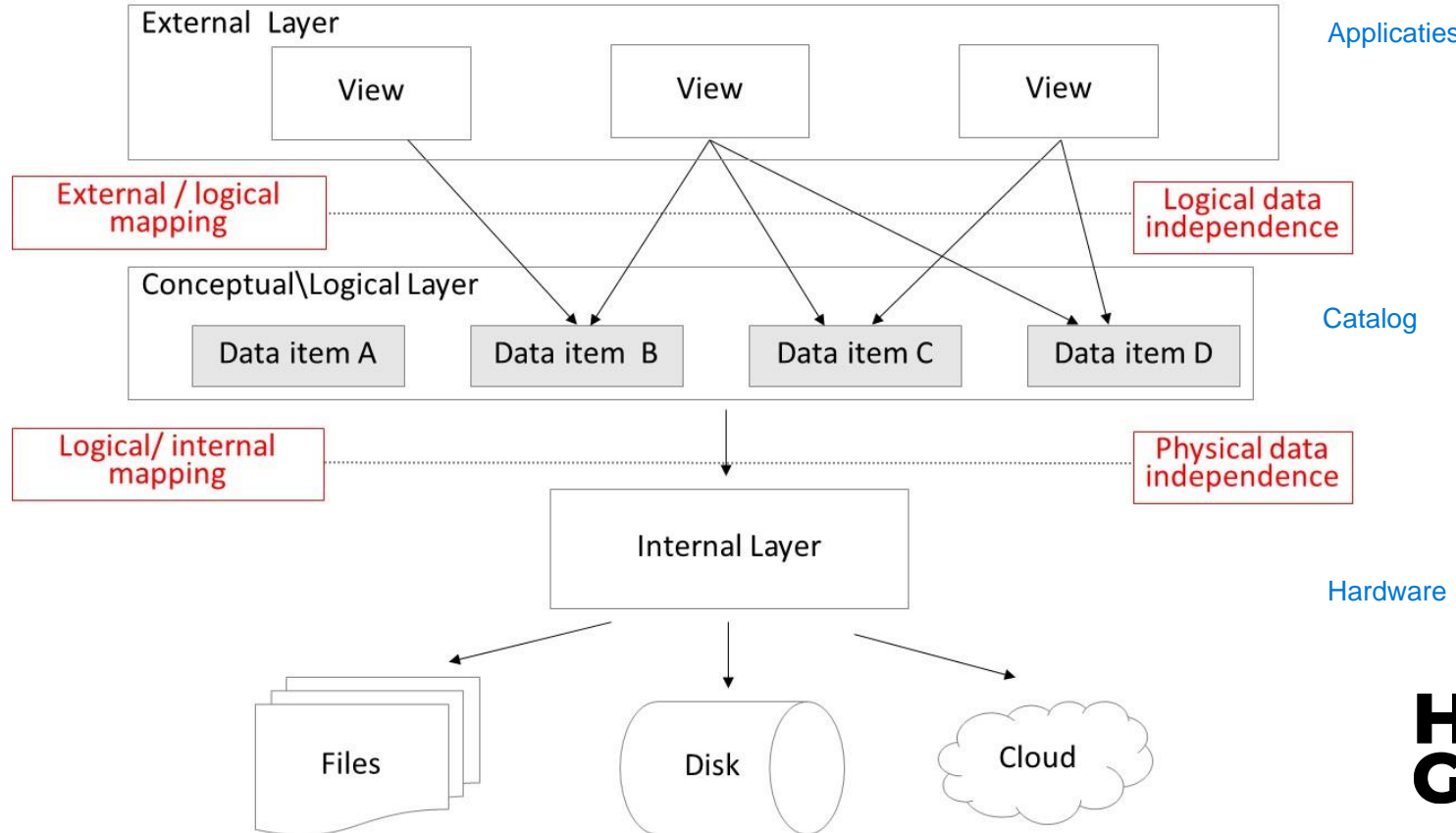
om te babbelen met de klant

- Conceptueel gegevensmodel :
 - algemene beschrijving gegevenselementen, kenmerken en relaties
 - Gebruikt door 'IT' en 'business'
 - Weergave hoe 'de business' de gegevens ziet
 - Voorstelling: (E)ERD diagram
- !!Veronderstellingen en ontbrekende informatie duidelijk vermelden!!

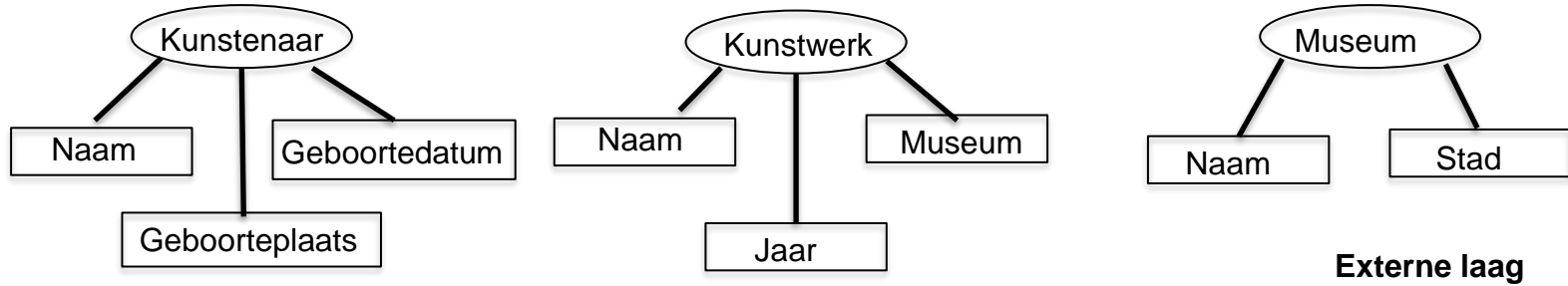
Gegevensmodel.

- Logisch gegevensmodel
 - vertaling conceptueel gegevensmodel naar het type databankmodel
 - Relationeel, hiërarchisch, OO, XML, NoSQL
 - omzetten naar intern (fysiek) gegevensmodel
 - Geeft informatie over fysieke opslag:
 - waar worden welke gegevens opgeslagen
 - Onder welke vorm
 - Indexen die het ophalen versnellen
 - Zeer DBMS afhankelijk
 - Externe gegevensmodellen
 - Deelverzameling van het logisch model
 - Voor een specifieke doelgroep.

3-Lagen model.



3-Lagen model.



Kunstenaar (naam, geboorteplaats, geboortedatum)
Kunstwerk (naam, museum, jaar)
Museum (naam, stad)

Logische laag

Verschillende server: - redundantie --> disaster recovery (backup)
- op verschillende locaties voor dichtbij proxy server



Brussel

Parijs

New York

synchronisatie: via Catalog

Interne laag

**HO
GENT**

Catalog.

Regels DB staan hier

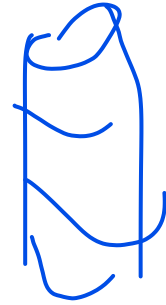
- Schatkist van DBMS
- Definities en beschrijving van de elementen in de DB (= metadata)
- Definities logisch gegevensmodel en intern gegevensmodel
- Zorgt voor synchronisatie en consistentie van de gegevensmodellen

Databank gebruikers.

wij meer

- IT Architect ontwerpt conceptueel gegevensmodel in samenspraak met de 'business'
- Dbontwerper vertaalt conceptueel model naar logisch en intern model
- DBA (databankbeheerder of database administrator) implementeert en monitort DB
- Applicatieontwikkelaar schrijft views databankprogramma's/databankapplicaties
- 'Business' gebruikt databankapplicaties en voert op die manier databankacties uit

- smv: 1. info
2. conceptueel model
----- (mapping)
3. logisch model (relationeel)
3.1. normalisatie
4. fysiek (SQL)



5. Kenmerken van databanken en databankbeheer

Nog enkele belangrijke begrippen.

- Gegevensonafhankelijkheid [losgekoppelde apps; view <-> DBMS](#)
- Modellen
- Gestructureerde en ongestructureerde gegevens
- Integriteitsregels
- Redundante gegevens

Gegevensonafhankelijkheid.

- = wijzigingen aan de gegevensbeschrijving hebben weinig tot geen impact op de applicaties
- Fysieke gegevensonafhankelijkheid: wijzigingen van de opslagspecificaties hebben geen invloed op het logisch model noch op de applicatie DBMS <-> SSD/catalog
→ wordt opgevangen door het DBMS
- Logische gegevensonafhankelijkheid: minimale aanpassingen aan de applicaties bij wijzigingen aan het logisch model

(On)gestructureerde gegevens.

- **Gestructureerde gegevens** rel DB
 - Kunnen in een logisch datamodel voorgesteld worden
 - Integriteitsregels kunnen opgesteld en afgedwongen worden
 - Vereenvoudigen, opzoeken, verwerken en analyseren
 - Voorbeelden: naam, geboortjaar, geboorteplaats van een kunstenaar
- **Ongestructureerde gegevens**
 - kunnen niet op een zinvolle manier worden geïnterpreteerd door een applicatie
 - Voorbeelden: gesprekken op social media, e-mails
 - Let op: er bestaat veel meer ongestructureerde data dan gestructureerde data

(On)gestructureerde gegevens.

- Semi-gestructureerde gegevens

vb FB (eigen profiel)

- De structuur van de gegevens is zeer onregelmatig of zeer wisselend.
- Voorbeelden: webpagina's van individuele gebruikers op een social media platform, samenvattende human resources documenten

Integriteitsregels.

betrouwbaar / niet corrupt maken

- **Integriteitsregels** worden gedefinieerd op basis van het conceptueel model en opgeslagen in de catalog
 - Worden afgedwongen door het DBMS
- Vastleggen hoe gegevens worden opgeslagen (syntactische regel)
 - Voorbeeld: geboortedatum, eenheidsprijs
- Vastleggen wanneer gegevens correct zijn (semantische regel)
 - Voorbeeld: Eenheidsprijs > 0 ;
geboortedatum niet $>$ vandaag

Redundante gegevens.

kosten maar backup

- Databank = centrale en unieke opslag gegevens
- Soms worden databanken gedupliceerd uit veiligheidsoverwegingen of omwille van performantie
→redundantie
- DBMS is verantwoordelijk voor de synchronisatie en garandeert de juistheid van de gegevens

Indeling van DBMS-systemen.

- Op basis van het gegevensmodel
- Op basis van het gebruik

Indeling op basis van het gegevensmodel.

- Hiërarchisch DBMS

- Het gegevensmodel is een omgekeerde boom
- DML is procedureel en gebaseerd op 'recordverwerking'
- Geen query processor (logisch en fysisch datamodel lopen door elkaar)
- Voorbeeld: IMS (IBM)

- Netwerk DBMS

- Gebruiken een netwerk gegevensmodel
- CODASYL DBMS [COnference on DAta SYstem Languages](#)
- DML is procedureel en gebaseerd op 'recordverwerking'
- geen query processor (logisch en fysisch datamodel lopen door elkaar)
- Voorbeeld: CA-IDMS (Computer Associates → Broadcom)

Indeling op basis van het gegevensmodel. ^{wij}

- Relationale DBMS

- Maakt gebruik van het relationeel gegevensmodel
- Momenteel het meest frequent gebruikt in de bedrijfswereld
- SQL (beschrijvend en gebaseerd op resultset)
- Query processor
- Strikte scheiding tussen het logisch en het fysisch gegevensmodel
- Voorbeelden: MySQL (open source, Oracle), Db2 (IBM), ^{IBM bekend voor `mainframes`} Oracle DBMS (Oracle), SQLServer (Microsoft)

Indeling op basis van het gegevensmodel.

- Object-Oriented DBMS (OODBMS)
 - Gebaseerd op het OO gegevensmodel
 - Geen probleem om het gegevensmodel van de databank te verenigen met de OO programmeertaal
 - Voorbeelden: db4o (open source, Versant), Caché (Intersystems), GemStone/S (GemTalk Systems)
 - Alleen doorgebroken in niche markten omwille van de complexiteit van het OODBMS

Indeling op basis van het gegevensmodel.

- Object-Relationeel DBMS (ORDBMS)
 - Wordt ook extended relationeel DBMS (ERDBMS) genoemd
 - Gebruikt een relationeel gegevensmodel uitgebreid met OO concepten
 - DML is SQL (beschrijvend en gebaseerd op resultset)
 - Voorbeelden: Oracle DBMS (Oracle), DB2 (IBM), SQLServer (Microsoft)

Indeling op basis van het gegevensmodel.

- **XML DBMS** lichte boom structuur
html achtig
 - Maakt gebruik van het XML gegevensmodel om gegevens op te slaan
 - Native XML DBMS (voorbeelden: BaseX, eXist) zal de boomstructuur van een XML document projecteren op een fysische opslagstructuur
 - XML-enabled DBMS (voorbeelden: Oracle, IBM Db2) zijn bestaande DBMS-systemen die uitgebreid zijn met faciliteiten om gegevens uit een XML gegevensmodel op te slaan

Indeling op basis van het gegevensmodel.

- NoSQL DBMS

- Bedoeld om massaal veel en ongestructureerde data op te slaan
- Kan ingedeeld worden in
 - key-value stores (sleutelwaardedatabanken)
 - column-oriented databases (gegevens worden per kolom gestockeerd ipv per rij)
 - graph databases
- Focust op schaalbaarheid van de databank en op werken met onregelmatige en snel veranderende gegevensstructuren
- Voorbeelden: Apache Hadoop, MongoDB, Neo4j

Indeling op basis van gebruik.

- On-line transaction processing (OLTP)
 - Focust op het beheren en verwerken van operationele en transactionele gegevens
 - De databankserver moet heel veel eenvoudige transacties per tijdseenheid kunnen verwerken
 - Het DBMS moet goed ontworpen zijn om heel veel korte eenvoudige queries uit te voeren
- On-line analytical processing (OLAP)
 - Focust op het gebruiken van operationele gegevens om strategische en tactische beslissingen te nemen
 - Een beperkt aantal gebruikers zal complexe queries uitvoeren
 - Het DBMS moet complexe queries efficiënt kunnen verwerken.

Indeling op basis van gebruik.

blobstore - fotos

- **Big Data & Analytics**

- NoSQL databanken
- Focust op meer flexibele databankstructuren en databanken zonder schema
- Stockeert ongestructureerde data zoals e-mails, Twitter tweets, Facebook posts, ...

- **Multimedia**

- Multimedia DBMS-systemen zorgen voor opslagruimte voor multimedia data zoals tekst, foto's, audio, video, 3D games, ...
- Moet ook de bijhorende query mogelijkheden (op basis van inhoud) aanbieden

Indeling op basis van gebruik.

- Geometrische toepassingen

ggl maps

- Een geometrisch DBMS laat toe om geometrische gegevens op te slaan en op te vragen (zowel 2D als 3D)

Voorbeeld: Geografische Informatie Systemen (GIS)

- Sensoren

time series DB

- Een sensor DBMS beheert de gegevens van sensoren

Voorbeelden: biomedische gegevens van kledij of medische apparatuur of gegevens uit telematica toepassingen

Indeling op basis van gebruik.

- **Mobiel** roomDB
 - Een mobiele DBMS werkt op een mobiel toestel (smartphone, tablet , ...)
 - Moet altijd online zijn en kunnen werken in een omgeving met een beperkte verwerkingskracht, weinig opslagruimte en beperkte energievoorziening (batterij)
- **Open source**
 - De code van een open source DBMS is voor iedereen toegankelijk en iedereen kan code toevoegen
 - Zie ook: www.sourceforge.net
 - Voorbeeld: MySQL (Oracle)