endava

Public

Osnovni Data koncepti

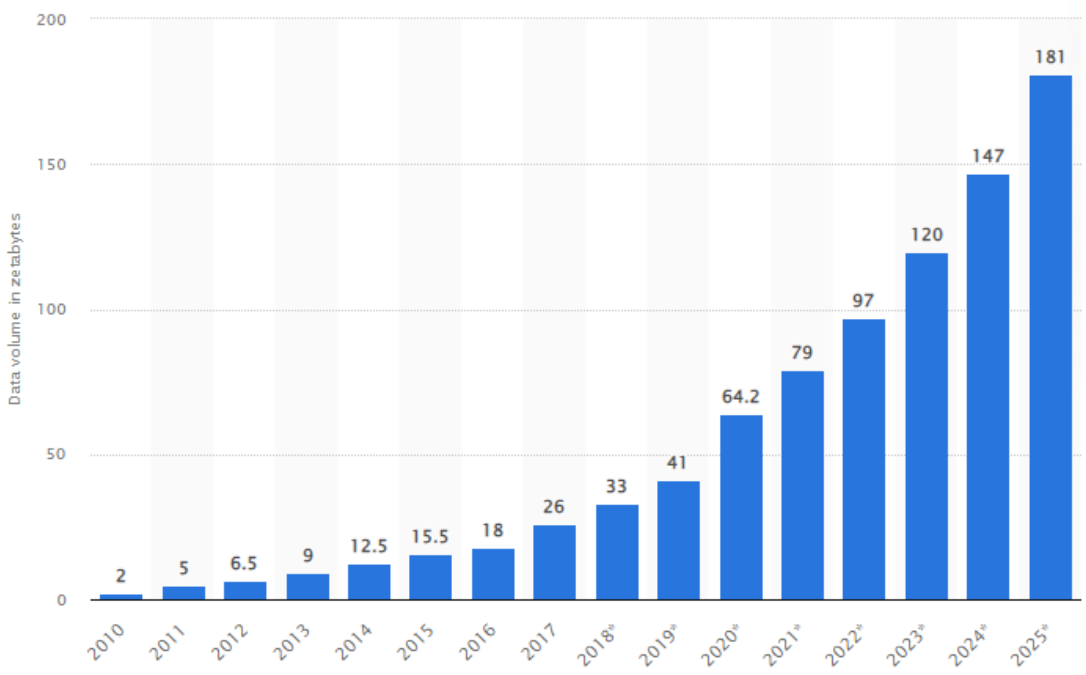
1. ŠTA JE DATA? DEFINICIJA I POJAŠNJENJE.
2. DATA FORMATI
3. CUVANJE PODATAKA (FILE STORES, DATABASES)
4. DATOTEKE (CSV, JSON, XML, BLOB)
5. BAZE PODATAKA (RELACIONE, NE-RELACIONE)
6. TRANSAKCIONO PROCESIRANJE PODATAKA (OLTP)
7. ANALITICKO PROCESIRANJE PODATAKA (OLAP)
8. JEZERA PODATAKA (DATA LAKE)
9. OBRADA PODATAKA (BATCHING VS. STREAMING)

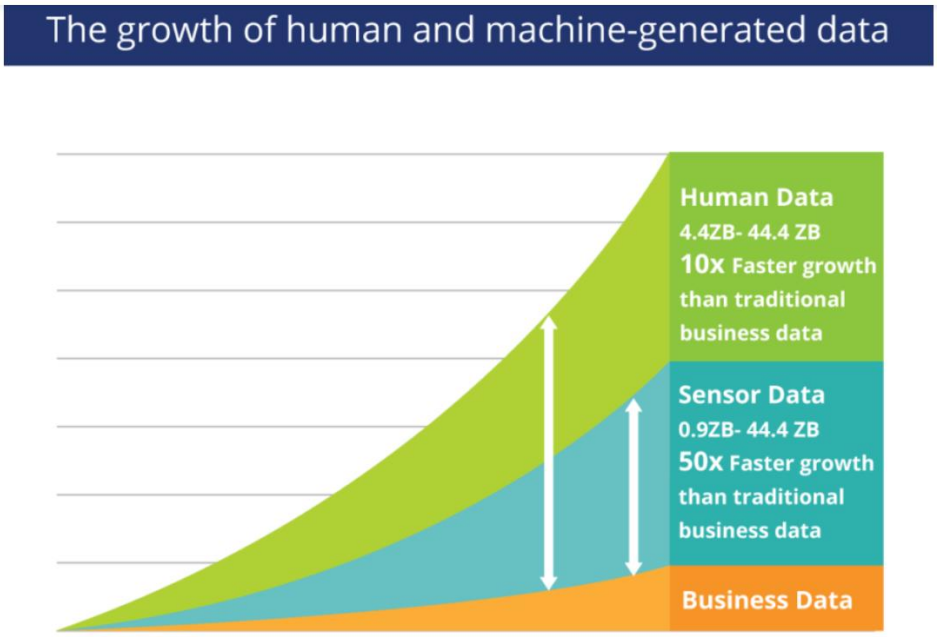
Predavači: Miroslav Kusmuk – Data Consultant

Oktobar 2023.

1) Sta je Data? Definicija I pojasnjenje.

U poslednjih nekoliko decenija, količina podataka koje su generirasali sistemi, aplikacije, različiti uređaji značajno se povećala. Podaci su svuda i dostupni su u različitim strukturama i formatima. Razumevanje podataka i njihovo istraživanje otkriva zanimljive činjenice i pomaže da se donesu tačne, ispravne i pravovremene odluke.





Podatak je kolekcija činjenica kao što su brojevi, opisi i zapažanja koji se koriste u donošenju odluka. Obrađeni i analizirani podaci koji obezbeđuju neko novo znanje predstavljaju informacije.

2) *Data formati*

Podaci se mogu klasifikovati kao strukturirani, polustrukturirani i nestrukturirani.

Struktuirani podaci

Strukturirani podaci koji se pridrzavaju fiksne scheme, tako da svi podaci imaju ista polja ili svojstva. Svaki red u tabeli ima isti broj kolona.

Najcesce, schema za entitete strukturiranih podataka je tabela, drugim recima, podaci su predstavljeni u jednoj ili vise tabela, koje se sastoje od redova koji predstavljaju svaku instancu entiteta, i kolona koje predstavljaju atribute entiteta. Baze podataka koje sadrže tabele u ovom obliku nazivaju se relacionim bazama podataka. Strukturirani podaci se cesto cuvaju u bazi podataka u kojoj se vise tabela mogu referencirati jedna na drugu, koriscenjem vrednosti kljuca u relacionom modelu.
Naredna slika ilustruje primer dve tabele u etrgovini. Prva tabela sadrzi detalje klijenata organizacije, a druga tabela sadrzi podatke o proizvodima koje organizacija prodaje.

Customer

ID	FirstName	LastName	Email	Address
1	Joe	Jones	joe@litware.com	1 Main St.
2	Samir	Nadoy	samir@northwind.com	123 Elm Pl.

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

ata-concepts/2-identify-need-data-solutions

CustomerID	Title	FirstName	MiddleName	LastName	Suffix	CompanyName	Phone
1	Mr.	Orlando	N.	Gee	NULL	A Bike Store	245-555-0173
2	Mr.	Keith	NULL	Harris	NULL	Progressive Sports	170-555-0127
3	Ms.	Donna	F.	Carreras	NULL	Advanced Bike Components	279-555-0130
4	Ms.	Janet	M.	Gates	NULL	Modular Cycle Systems	710-555-0173
5	Mr.	Lucy	NULL	Harrington	NULL	Metropolitan Sports Supply	828-555-0186
6	Ms.	Rosmarie	J.	Carroll	NULL	Aerobic Exercise Company	244-555-0112
7	Mr.	Dominic	P.	Gash	NULL	Associated Bikes	192-555-0173
10	Ms.	Kathleen	M.	Garza	NULL	Rural Cycle Emporium	150-555-0127
11	Ms.	Katherine	NULL	Harding	NULL	Sharp Bikes	926-555-0159
12	Mr.	Johnny	A.	Caprio	Jr.	Bikes and Motorbikes	112-555-0191
16	Mr.	Christopher	R.	Beck	Jr.	Bulk Discount Store	1 (11) 500 555-0132
18	Mr.	David	J.	Liu	NULL	Catalog Store	440-555-0132
19	Mr.	John	A.	Beaver	NULL	Center Cycle Shop	521-555-0195
20	Ms.	Jean	P.	Handley	NULL	Central Discount Store	582-555-0113
21	N...	Jinghao	NULL	Liu	NULL	Chic Department Stores	928-555-0116
22	Ms.	Linda	E.	Burnett	NULL	Travel Systems	121-555-0121
23	Mr.	Kerim	NULL	Hanif	NULL	Bike World	216-555-0122
24	Mr.	Kevin	NULL	Liu	NULL	Eastside Department Store	926-555-0164
25	Mr.	Donald	L.	Blanton	NULL	Coalition Bike Company	357-555-0161
28	Ms.	Jackie	E.	Blackwell	NULL	Commuter Bicycle Store	972-555-0163
29	Mr.	Bryan	NULL	Hamilton	NULL	Cross-Country Riding Supp...	344-555-0144
30	Mr.	Todd	R.	Logan	NULL	Cycle Merchants	783-555-0110
34	Ms.	Barbara	J.	German	NULL	Cycles Wholesaler & Mfg.	1 (11) 500 555-0181
37	Mr.	Jim	NULL	Geist	NULL	Two Bike Shops	724-555-0161

ProductID	Name	ProductNumber	Color	StandardCost	ListPrice	Size	Weight
680	HL Road Frame - Black, 58	FR-R92B-58	Black	1059.31	1431.50	58	1016.04
706	HL Road Frame - Red, 58	FR-R92R-58	Red	1059.31	1431.50	58	1016.04
707	Sport-100 Helmet, Red	HL-U509-R	Red	13.0863	34.99	NULL	NULL
708	Sport-100 Helmet, Black	HL-U509	Black	13.0863	34.99	NULL	NULL
709	Mountain Bike Socks, M	SO-B909-M	White	3.3963	9.50	M	NULL
710	Mountain Bike Socks, L	SO-B909-L	White	3.3963	9.50	L	NULL
711	Sport-100 Helmet, Blue	HL-U509-B	Blue	13.0863	34.99	NULL	NULL
712	AWC Logo Cap	CA-1098	Multi	6.9223	8.99	NULL	NULL
713	Long-Sleeve Logo Jersey, S	LJ-0192-S	Multi	38.4923	49.99	S	NULL
714	Long-Sleeve Logo Jersey, M	LJ-0192-M	Multi	38.4923	49.99	M	NULL
715	Long-Sleeve Logo Jersey, L	LJ-0192-L	Multi	38.4923	49.99	L	NULL
716	Long-Sleeve Logo Jersey, XL	LJ-0192-X	Multi	38.4923	49.99	XL	NULL
717	HL Road Frame - Red, 62	FR-R92R-62	Red	868.6342	1431.50	62	1043.26
718	HL Road Frame - Red, 44	FR-R92R-44	Red	868.6342	1431.50	44	961.61
719	HL Road Frame - Red, 48	FR-R92R-48	Red	868.6342	1431.50	48	979.75
720	HL Road Frame - Red, 52	FR-R92R-52	Red	868.6342	1431.50	52	997.90
721	HL Road Frame - Red, 56	FR-R92R-56	Red	868.6342	1431.50	56	1016.04
722	LL Road Frame - Black, 58	FR-R38B-58	Black	204.6251	337.22	58	1115.83
723	LL Road Frame - Black, 60	FR-R38B-60	Black	204.6251	337.22	60	1124.90
724	LL Road Frame - Black, 62	FR-R38B-62	Black	204.6251	337.22	62	1133.98
725	LL Road Frame - Red, 44	FR-R38R-44	Red	187.1571	337.22	44	1052.33
726	LL Road Frame - Red, 48	FR-R38R-48	Red	187.1571	337.22	48	1070.47
727	LL Road Frame - Red, 52	FR-R38R-52	Red	187.1571	337.22	52	1088.62

Polu-struktuirani podaci

Polustrukturirani podaci su informacije koje se ne nalaze u relacionoj bazi podataka, ali ipak imaju neku strukturu.
Cesti primeri su dokumenti u JSON (JavaScript Object Notation) formatu. Naredni primer prikazuje par dokumenata koji predstavljaju podatke o klijentu.
U oba slucaja, svaki dokument sadrzi child dokument adresu, ali polja u ova dva dokumenta se mogu razlikovati po klijentu. Neki klijenti mogu imati email adresu, neki mogu imati vise, a neki ne moraju imati.

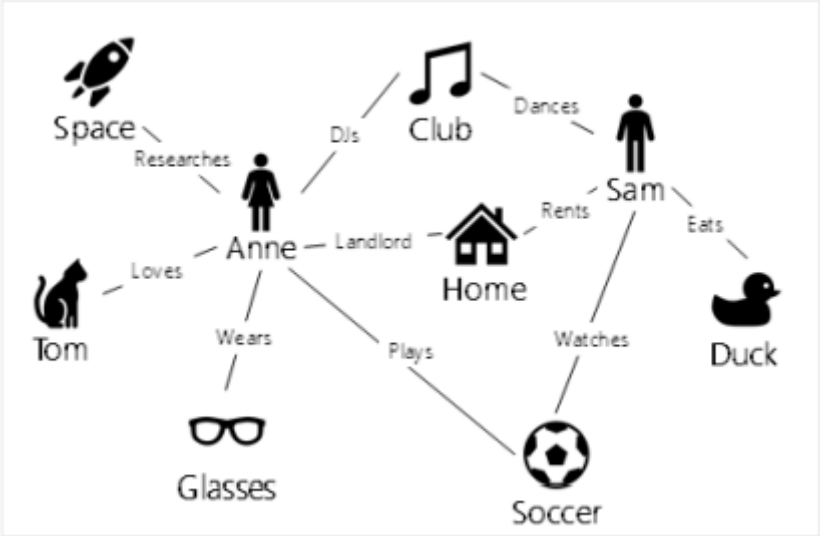
JSON Copy

```
// Customer 1
{
  "firstName": "Joe",
  "lastName": "Jones",
  "address":
  {
    "streetAddress": "1 Main St.",
    "city": "New York",
    "state": "NY",
    "postalCode": "10099"
  },
  "contact":
  [
    {
      "type": "home",
      "number": "555 123-1234"
    },
    {
      "type": "email",
      "address": "joe@litware.com"
    }
  ]
}

// Customer 2
{
  "firstName": "Samir",
  "lastName": "Nadoy",
  "address":
  {
    "streetAddress": "123 Elm Pl.",
    "unit": "500",
    "city": "Seattle",
    "state": "WA",
    "postalCode": "98999"
  },
  "contact":
  [
    {
      "type": "email",
      "address": "samir@northwind.com"
    }
  ]
}
```

Postoje i drugi tipovi polustrukturiranih podataka (key-value, graph baze podataka).
Key-value baze podataka, koriste kljuceve kao jedinstveni identifikator da dodju do odredjene vrednosti. Te vrednosti mogu biti bilo sta od broja, stringa, ili slozenog objekta npr. JSON file-a. Key-value baze se razlikuju od relacionih baza podataka, gde su tabele sacinjene od redova i kolona, sa predefinisanim tipovima podataka.

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Grpah baze podataka se mogu koristiti za cuvanje informacija o slozenim relacijama. Graph sadrzi cvor (node) koji predstavlja informaciju o objektima i ivicu (edge) koja predstavllja relaciju izmedju objekata.



Nestruktuirani podaci

Nestruktuirani podaci su uglavnom podaci koje je teško ubaciti u relacione tabele baza podataka radi analize ili izvršavanja upita nad njima. Ne mogu se tražiti određeni elementi unutar tih podataka. Podaci ovakvog tipa predstavljaju slike, audio i video fajlove.

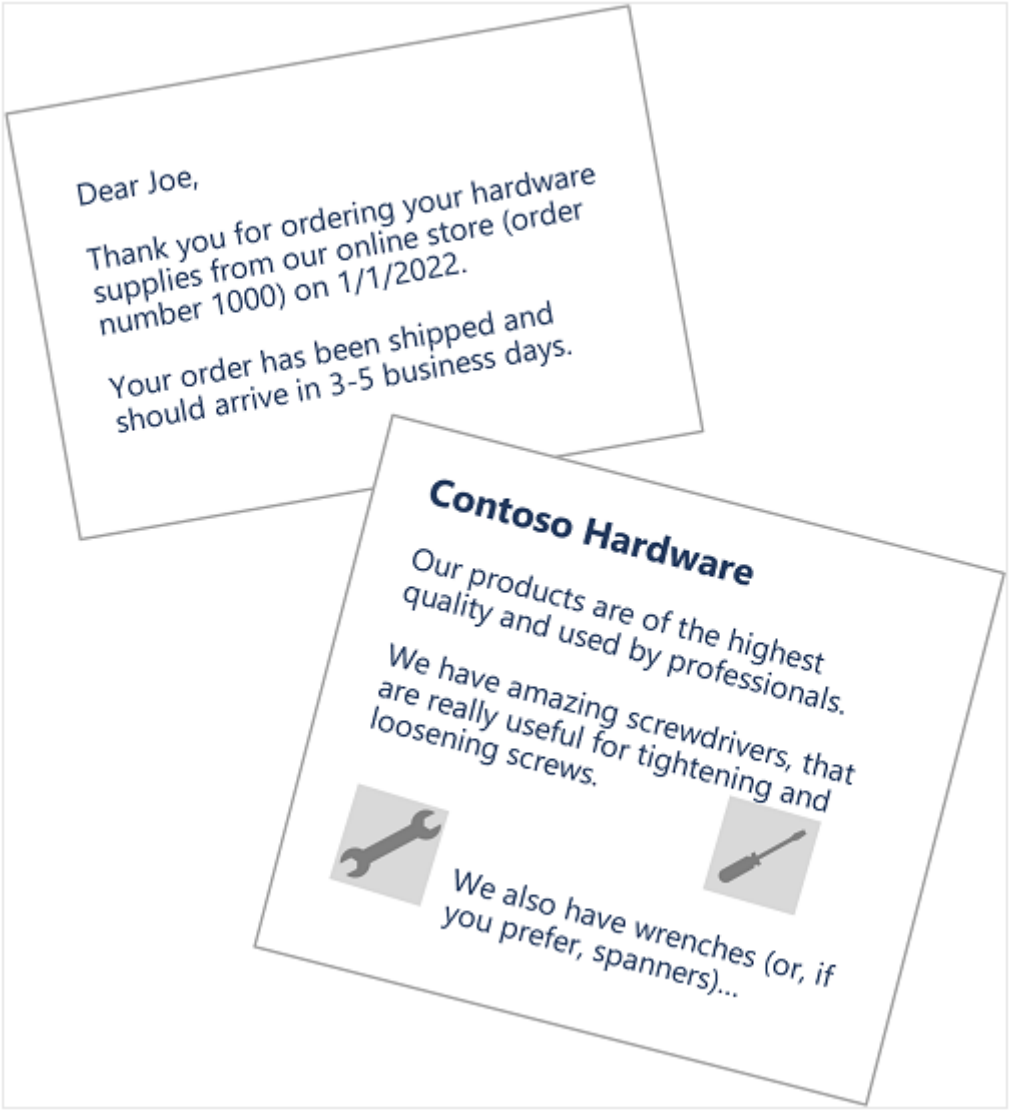
Nestruktuirani podaci su podaci koji ne prate neki definisani format. Kao i u slučaju struktuiranih podataka i nestrukturirani podaci mogu biti mašinski ili ljudski generisani.

Neki primeri mašinski generisanih nestruktuiranih podataka su:

- Satelitske slike: Ovo uključuje podatke o vremenskim prilikama ili podatke koje vlade prikupljaju prilikom satelitskog nadgledanja.
- Naučni podaci: ovo uključuje seizmičke slike, atmosferske podatke itd

Datoteke se takođe mogu smatrati oblikom nestrukturiranih podataka, iako u nekim slučajevima može sadržati metapodatke koji ukazuju na vrstu datoteke (fotografija, Word

dokument, Excel tabela itd.), vlasnika i druge elemente koji bi mogli biti pohranjeni kao polja. Međutim, glavni sadržaj datoteke je nestrukturiran.



3) Cuvanje podataka (File stores, Databases)

Dva su osnovna nacina cuvanja podataka:
- File store (u datotekama)
- Baze podataka

4) Datoteke

Sposobnost cuvanja podataka u datotekama je kljucni element svakog racunarskog sistema. Datoteke se mogu cuvati u lokalnim file sistemima na hard-disc-u, ili na prenosivim medijima, ali u vecini organizacija vazne datoteke sa podacima se cuvaju centralno u nekoj vrsti shared file sistema. Sve vise se ta centralna lokacija za skladistenje nalazi u cloud-u, sto omogucava isplativo, bezbedno I pouzdano skladistenje velkih kolicina podataka.

Konkrento, format file-a koji se koristi za skladistenje podataka zavisi od brojnih faktora:
- Tip podataka koji se cuvaju (strukturirani, polustrukturirani, nestrukturirani)
- Aplikacije i servisi koji ce morati da citaju, pisu i obradjuju podatke
- Potreba da file-ovi budu citljivi od strane ljudi, ili optimizovani za efikasno skladistenje I obradu.

Najcesci formati:
CSV (Comma-separated values) – podaci se cuvaju u plain text-u, sa specificnim delimiterima, gde su pojedinacna polja odvojena zarezom. Opciono, prvi red moze ukljucivati nazive kolona. Dobra su izbor za strukturirane podatke, koji zahtevaju pristup aplikacijama i servisima, citljivi su od strane ljudi.
JSON (JavaScript Object Notation) – koristi hijerarhijsku schemu dokumenata za definisanje objekata koji imaju vise atributa. Svaki atribut moze biti objekat ili niz, cineci JSON fleksibilnim formatom koji je dobar I za strukturirane I polustrukturirane podatke.
XML (Extensible Markup Language) – citljiv od strane ljudi, slican JSON-u. Koriste oznake zatvorene ugaonim zgradama (<../>) da definise elemente I atribute.

XML Copy

```xml
<Customers>
  <Customer name="Joe" lastName="Jones">
    <ContactDetails>
      <Contact type="home" number="555 123-1234"/>
      <Contact type="email" address="joe@litware.com"/>
    </ContactDetails>
  </Customer>
  <Customer name="Samir" lastName="Nadoy">
    <ContactDetails>
      <Contact type="email" address="samir@northwind.com"/>
    </ContactDetails>
  </Customer>
</Customers>
```

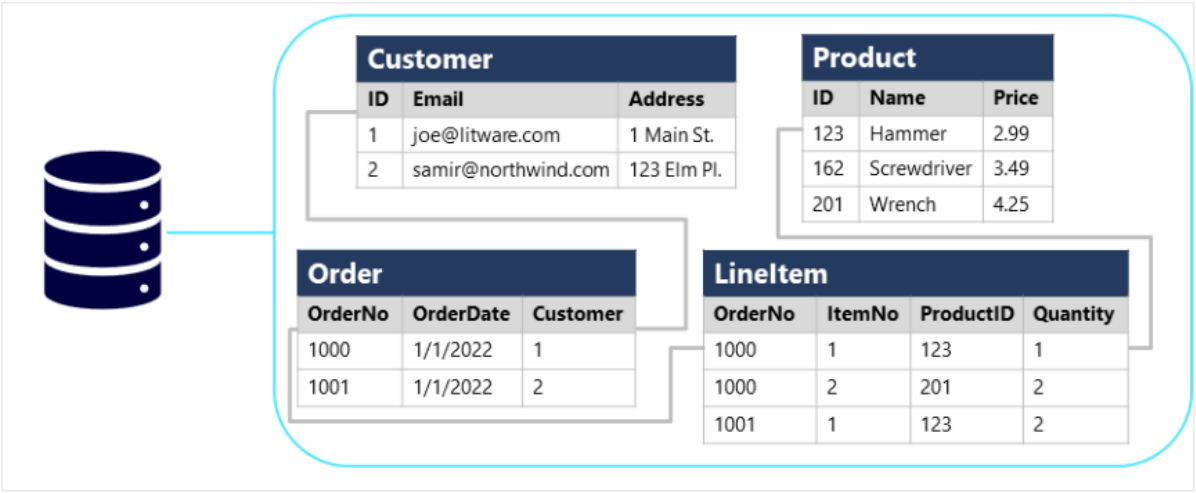
BLOB (Binary Large Object) – sve datoteke se cuvaju kao binarni podaci (1 i 0), ali u formatima citljivim ljudima, byte-ovi se mapiraju u znakove (obicno ASCII ili Unicode).

Medjutim, neki formati datoteka, posebno za nestrukturirane podatke, cuvaju podatke kao sirove binarne podatke koje aplikacije moraju interpretirati I prikazati. Uobicajeni tipovi podataka, koji se cuvaju kao binarni ukljucuju slike, video, audio I dokumente specificne za aplikaciju.
Optimizovani file formati – file formati koji omogucavaju kompresovanje, indeksiranje, efikasno cuvanje I obradu (Avro, ORC and Parquet).

5) Baze podataka (Relacione, Ne-Relacione)

Baza podataka se koristi kao centralni sistem u kome se podaci mogu cuvati i pretrazivati. U pojednostavljenom smislu, file sistem na kome se cuvaju datoteke je neka vrsta baze podataka, ali kada koristimo termin u profesionalnom smislu, misli se na sistem za upravljanje zapisima (recordima) podataka a ne datotekama.

Relacione baze podataka se koriste za skladistenje i pretrazivanje strukturiranih podataka. Podaci se cuvaju u tabelama koje predstavljaju entitete, npr, klijente, proizvode, narudzbenice. Tabela sadrži redove, a svaki red predstavlja jednu instancu entiteta. Svaka instanca entiteta ima primarni kljuc, koji ga jedinstveno identifikuje, i ovi kljucevi se koriste kao referenca na instancu entiteta u drugim tabelama. Npr. primarni kljuc klijenta, moze se referencirati u narudzbenici, da bi se naznacilo koji klijent je poslao porudzbinu. Upotreba kluceva za referenciranje entiteta, omogucava normalizaciju relacione baze podataka, sto znaci eliminaciju duplih vrednosti (redundasi), tako da se detalji klijenta cuvaju samo jednom, a ne za svaki porudzbinu koju klijent postavlja. Tabelama I podacima se upravlja, pomocu strukturiranog upitnog jezika (SQL).



U scenariju e-trgovine, svaki red u tabeli kupaca sadrži podatke za jednog kupca, svaki red u tabeli proizvoda definiše jedan proizvod, a svaki red u tabeli narudžbi predstavlja narudžbu koju je napravio kupac.

Customers

Customer ID	Customer Name	Customer Address
C1	Fred	...
C2	Bert	...
C3	Jane	...

Products

Product ID	Product Name	Description
P1	Shirt	...
P2	Tie	...
P3	Collar	...

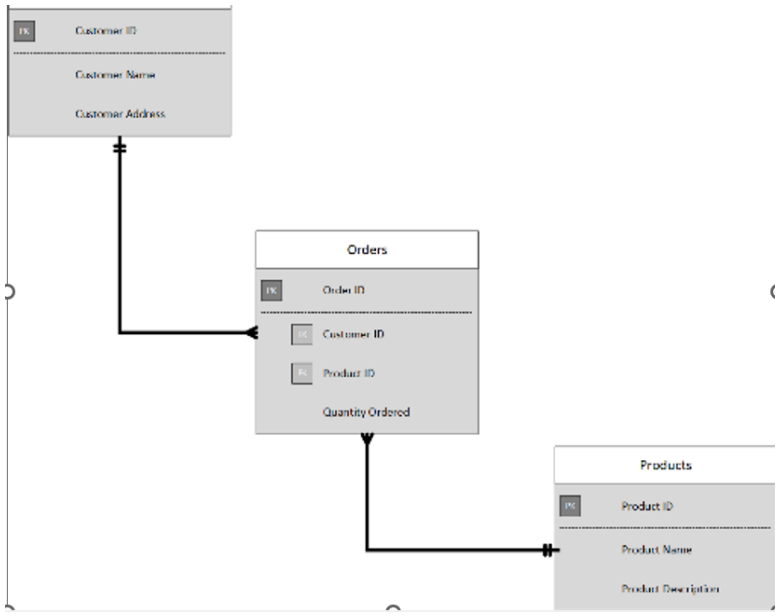
Orders

Order ID	Customer ID	Product ID	Quantity
1000	C1	P1	1
1001	C2	P1	3
1002	C1	P3	1
1003	C1	P3	2
1004	C2	P2	4
1005	C1	P2	2
1006	C3	P3	1

Relaciona baya podataka dizajnira se kreiranjem modela podataka. Primarni ključ označava kolonu (ili kombinaciju kolona) koja jedinstveno identifikuje svaki red. Svaka tabela mora da ima primarni ključ.

Linije koje povezuju tabele označavaju tip odnosa. U navedenom primeru odnos između kupaca i narudžbi je 1-prema-više (jedan kupac može imati više narudžbi, ali svaka narudžba odnosi se na tačno jednog kupca). Slično tome, odnos između narudžbi i proizvoda je više prema 1 (nekoliko narudžbi može biti za isti proizvod).

Kolone označene FK su kolone stranog ključa. Oni upućuju na primarni ključ druge tabele i koriste se za održavanje odnosa između tabela. Strani ključ takođe pomaže da se identifikuju i spreče anomalije, kao što su narudžbe za kupce koje ne postoje u tabeli Kupci. U modelu ispod, stupci ID kupca i ID proizvoda u tabeli Narudžbe povezuju se na kupca koji je naručio i proizvod koji je naručen:



Glavne karakteristike relacione baze podataka su:
• Svi podaci su tabelarni. Entiteti su modelovani kao tabele, svaka instanca entiteta je red u tabeli, a svako svojstvo je definisano kao kolona.
• Svi redovi u istoj tabeli imaju isti broj kolona.
• Tabela može imati neograničen broj redova.
• Primarni ključ jedinstveno identifikuje svaki red u tabeli.
• Strani ključ upućuje na redove u drugoj, povezanoj tabeli. Za svaku vrednost u koloni stranog ključa, trebao bi postojati red s istom vrednošću u odgovarajućoj koloni primarnog ključa u drugoj tabeli.

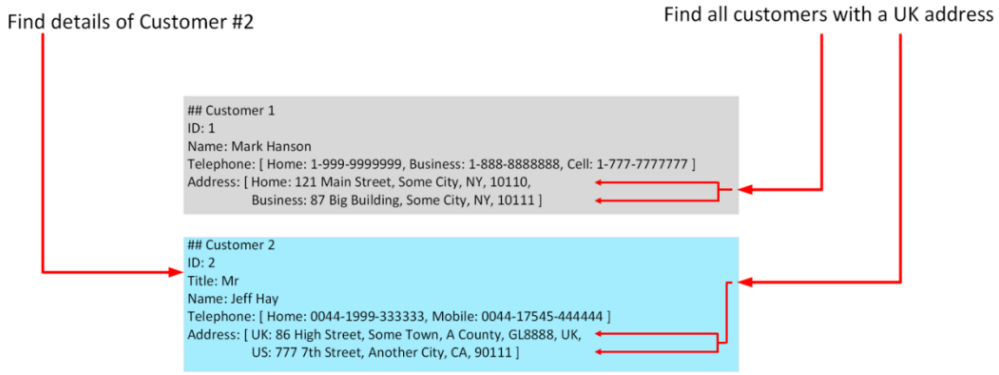
Podaci dolaze u svim oblicima i veličinama i mogu se koristiti u mnoge svrhe. Mnoge organizacije koriste relacione baze podataka za pohranjivanje ovih podataka. Međutim, relacioni model nekada nije najprikladnija struktura. Format podataka može biti previše raznolik da bi se lako modelovao kao skup relacionih tabela. Npr, podaci mogu sadržati stavke kao što su video, audio, slike, vremenske informacije itd... Takođe, zahtevi za obradom podataka možda neće najbolje odgovarati pokušaju konverzije ovih podataka u relacioni format. U ovim situacijama, možda bi bilo bolje koristiti nerelacione modele koji mogu čuvati podatke u njihovom originalnom formatu i omogućiti brz pristup.

Ključni aspekt **nerelacionih baza podataka** je taj što vam omogućavaju pohranjivanje podataka na vrlo fleksibilan način. Nerelacione baze podataka ne nameću šemu podacima. Umesto toga, fokusiraju se na same podatke, a ne na to kako ih strukturirati. Ovaj pristup znači da možete pohraniti informacije u prirodnom formatu, koji odražava način na koji biste ih analizirali i koristili.
U nerelacionom sistemu, informacije za entitete pohranjujete u zbirke ili kontejnere, a ne u relacione tabele. Dva entiteta u istoj kolekciji mogu imati različit skup polja umesto regularnog skupa kolona koji se nalaze u relacioonoj tabeli.
Nedostatak fiksne šeme znači da svaki entitet mora biti samoopisujući. Često se to postiže označavanjem svakog polja imenom podataka koje ono predstavlja. Npr nerelacijski skup entiteta kupaca može izgledati ovako:

```
## Customer 1
ID: 1
Name: Mark Hanson
Telephone: [ Home: 1-999-9999999, Business: 1-888-8888888, Cell: 1-777-7777777 ]
Address: [ Home: 121 Main Street, Some City, NY, 10110,
           Business: 87 Big Building, Some City, NY, 10111 ]

## Customer 2
ID: 2
Title: Mr
Name: Jeff Hay
Telephone: [ Home: 0044-1999-333333, Mobile: 0044-17545-444444 ]
Address: [ UK: 86 High Street, Some Town, A County, GL8888, UK,
           US: 777 7th Street, Another City, CA, 90111 ]
```

Mogućnosti pronalaženja podataka nerelacione baze podataka mogu varirati. Svaki entitet bi trebao imati jedinstvenu vrednost ključa. Entiteti u kolekciji se obično pohranjuju u redosledu ključ/vrednost. U gornjem primjeru, jedinstveni ključ je ID polje. Najjednostavniji tip nerelacijske baze podataka omogućava aplikaciji da specificira jedinstveni ključ ili raspon

ključeva kao kriterijum upita. U primeru kupaca, baza podataka bi omogućila aplikaciji da pita pretražuje samo po ID-u. Filtriranje podataka na drugim poljima zahtevalo bi skeniranje cele kolekcije entiteta, raščlanjivanje svakog entiteta naizmenično, a zatim primenu bilo kojeg kriterijuma upita na svaki entitet kako bi se pronašla sva podudaranja. U donjem primjeru, upit koji obuhvata detalje o klijentu po ID-u može brzo identifikovati koji entitet treba obuhvati. Upit koji pokušava pronaći sve kupce sa adresom u UK-u morao bi iterirati kroz svaki entitet i za svaki entitet redom pregledati svako polje. Ako baza podataka sadrži mnogo miliona entiteta, ovaj upit bi mogao potrajati dosta vremena.



Nerelacione baze podataka su vrlo pogodne za: IoT, maloprodaja i marketing, igre, web i mobilne aplikacije.

Relaciona baza podataka restrukturira podatke u fiksni format koji je dizajniran da odgovori na specifične upite. Kada se podaci moraju vrlo brzo uneti, ili je upit nepoznat i neograničen pogodnije je da se koristi nerelaciona baza podataka.

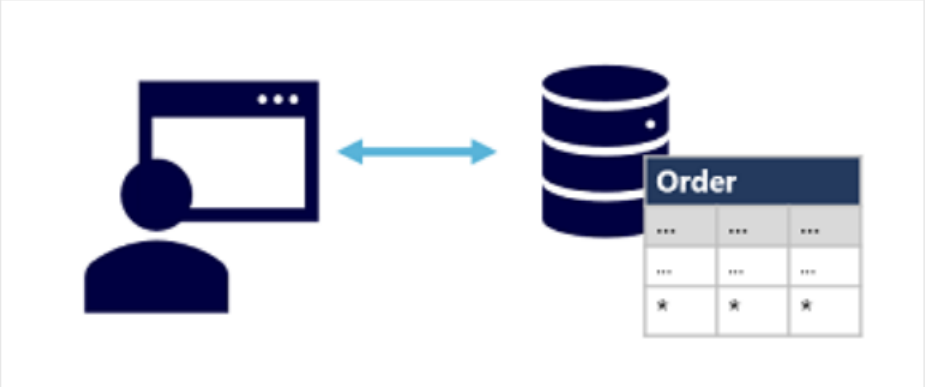
6) *Transakciono procesiranje podataka (OLTP)*

Primarna upotreba relacionih baza podataka je rukovanje i obrada transakcija.

Transakcioni sistem belezi transakcije koje obuhvataju specificne dogadjaje koje organizacija zeli da prati. Transakcija moze biti finansijska, kao kretanje novca izmedju racuna u bankarskom sistemu, moze biti I deo prodaje, kao pracenje placanja za robu I usluge od klijenata. Shvatite transakciju kao malu jedinicu rada.

Sve operacije određene transakcije moraju biti izvršene, a ako dođe do „pucanja“ transakcije sve prethodno izvršene operacije se poništavaju. Ne postoji delimično izvršenje transakcije. Svaka transakcija baze podataka ima definisanu početnu tačku, nakon čega slede koraci za modifikovanje podataka unutar baze podataka.

Transakcioni sistemi su cesto (high-volume) velikog obima, ponekad obradjuju milione transakcija u jednom danu. Podaci koji se obradjuju moraju da budu dostupni veoma brzo. Posao koji obavljaju transakcioni sistemi naziva se transakciono procesiranje podataka (OnLine Transactional Processing OLTP).



Transakcioni sistemi se oslanjaju na sisteme baza podataka, koje su optimizovane za operacije citanja i pisanja, kako bi se podrzale operacije, u kojima se zapisi kreiraju, preuzimaju, azuriraju I uklanjaju (CRUD operacije). Sve ove operacije se primenjuju transakcijski, na nacin koji obezbedjuje integritet podataka uskladistenih u bazi podataka. Da bi se postigao integritet OLTP sistemi primenjuju transakcije koje podrzavaju tzv. ACID semantiku (osobine).

Glavne karakteristike transakcije su:

A. **A (Atomicity)** - Atomičnost garantuje da se svaka transakcija tretira kao jedna jedinica, koja ili u potpunosti uspe ili potpuno ne uspe. Ako bilo koji deo transakcija ne bude uspešan i baza podataka ostaje nepromijenjena. Atomski sistem mora garantovati atomičnost u svakoj situaciji, uključujući nestanke struje, greške i padove.
B. **C (Consistency)** - Izvršenje transakcije nad bazom podataka koja je konzistentna, mora prevesti bazu u takođe konzistentno stanje, odnosno očuvati konzistentnost baze. Očuvanje konzistentnosti baze podrazumeva poštovanje svih ograničenja integriteta.

C. **I (Isolation)** - Kaže se da je skup transakcija nezavistan, ako je rezultat njihovog konkurentnog izvršavanja isti kao da su se izvršavale jedna po jedna, bez preklapanja izvršavanja. Saglasno tome, osobina nezavisnosti obezbeđuje konkurentno izvršavanje transakcija.
D. **D (Durability**) - Trajnost garantuje da će jednom kada je transakcija izvršena, ona ostati izvršena čak i ako dođe do kvara sistema kao što je nestanak struje ili pad.

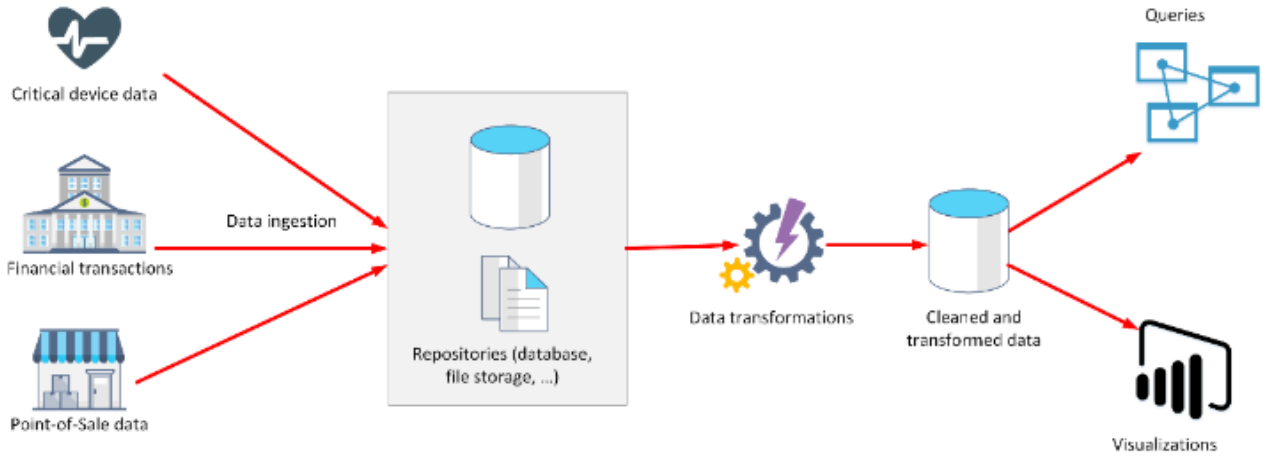
Sistemi baza podataka koji rade sa transakcijama veoma su složeni Oni moraju upravljati istovremenim korisnicima koji eventualno pokušavaju pristupiti i modifikovati iste podatke u isto vreme, obrađujući transakcije u izolaciji dok održavaju bazu podataka konzistentnom i povratnom. Mnogi sistemi implementiraju konzistentnost i izolaciju odnosa primenom zaključavanja podataka kada se ažuriraju. Zaključavanje sprečava drugi proces da čita podatke dok se zaključavanje ne otpusti. Zaključavanje se otpušta samo kada se transakcija izvrši ili vrati. Ekstenzivno zaključavanje može dovesti do loših performansi, dok aplikacije čekaju da se zaključavanje otpusti.

```
Begin

  delete from partija_tmp;

  insert into partija_tmp (partija, klijent)
    select kd.partija, kd.klijent
     from kredit_dospece kd
    where kd.stanje = 0;

  update partija_kredita pk set pk.status_kredita = ('OTPLACEN') where
  pk.partija in (select p1.patija from partija_tmp p1);
  update partija p set p.status = 'N' where p.partija in (select p1.patija
  from partija_tmp p1);

  commit;

end;
```

Distribuirane baze podataka se široko koriste u mnogim organizacijama. Distribuirana baza podataka je baza podataka u kojoj se podaci pohranjuju na različitim fizičkim lokacijama. Može se držati na više računara koji se nalaze na istoj fizičkoj lokaciji (npr, centar podataka), ili može biti raspršen preko mreže međusobno povezanih računara. U poređenju sa nedistribuiranim sistemima baza podataka, svakom ažuriranju podataka u distribuiranoj bazi podataka treba vremena da se primeni na više lokacija. Ako vam je potrebna transakciona konzistentnost u ovom scenariju, zaključavanja se mogu zadržati jako dugo, posebno ako postoji mrežni kvar između baza podataka u kritičnom trenutku. Da bi se suprotstavili ovom problemu, mnogi sistemi za upravljanje distribuiranim bazama podataka ublažavaju stroge zahtjeve za izolacijom transakcija i implementiraju "eventualnu konzistentnost". U ovom obliku konzistentnosti, dok aplikacija piše podatke, svaku promjenu snima jedan server, a zatim asinhrono propagira na druge servere u sistemu distribuirane baze podataka. Iako ova strategija pomaže da se minimizira kašnjenje, može dovesti do privremenih nedosljednosti u podacima.

7) Analiticko procesiranje podataka (OLAP)

Za razliku od sistema dizajniranih da podržavaju OLTP, analitički sistem je dizajniran da podrži poslovne korisnike koji trebaju upite nad podacima i steći širu sliku informacija koje se nalaze u bazi podataka.
Analitički sistemi se bave prikupljanjem sirovih podataka, analizranjem i njihovim korištenjem. Organizacija može koristiti ove uvide za donošenje poslovnih odluka. Npr, detaljni uvidi za proizvodnu kompaniju mogu ukazivati na trendove koji im omogućavaju da odrede na koje linije proizvoda da se fokusiraju radi profitabilnosti.
Većina analitičkih sistema za obradu podataka treba da obavlja slične zadatke: unos podataka, transformaciju podataka, upite podataka i vizualizaciju podataka.

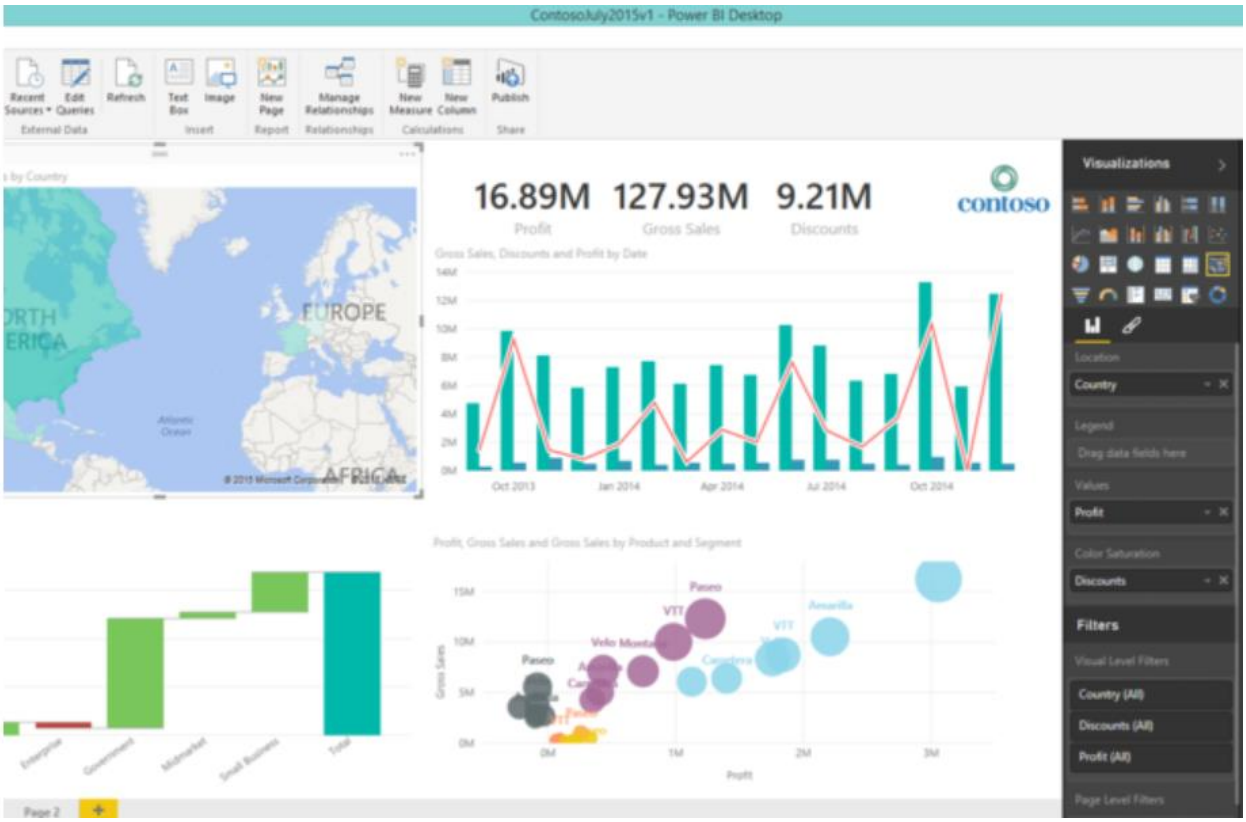


1) **Unošenje podataka (Data ingestion):** Unošenje podataka je proces hvatanja neobrađenih podataka. Ovi podaci se mogu uzeti iz kontrolnih uređaja koji mere informacije okoline, kao što su temperatura, uređaji na prodajnom mestu koji evidentiraju artikle koje je kupac kupio u supermarketu, finansijskih podataka koji evidentiraju kretanje novca između bankovnih računa i vremenskih podataka iz vremena stanice. Neki od ovih podataka mogu doći iz zasebnog OLTP sistema. Da biste obradili i analizirali ove podatke, prvo morate pohraniti podatke u neku vrstu skladišta. Repozitorijum može biti skladište datoteka, baza podataka dokumenata, ili čak relaciona baza podataka.

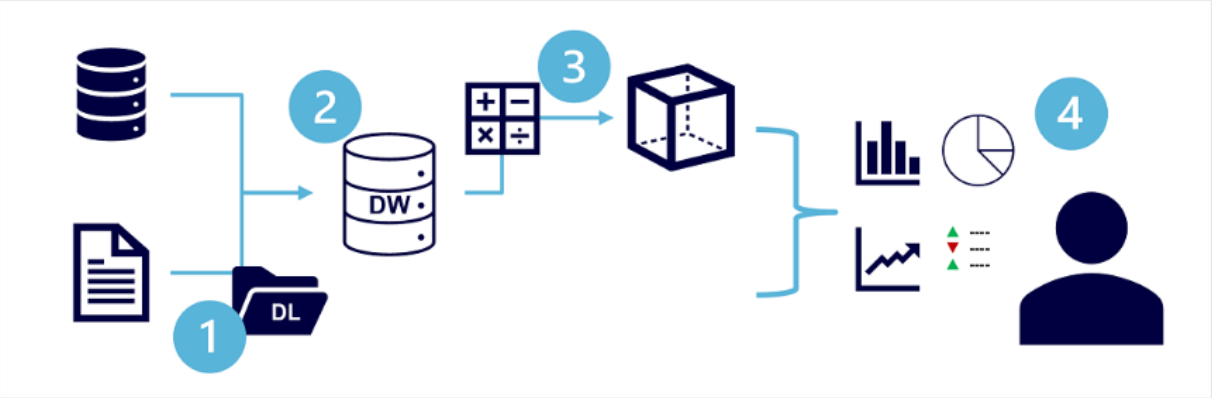
2) **Transformacija podataka/obrada podataka (Data Transformation/Data Processing):** neobrađeni podaci možda nisu u formatu koji je prikladan za postavljanje upita. Podaci mogu sadržati anomalije koje bi trebalo filtrirati ili ih je potrebno na neki način transformisati. Na primer, datumi ili adrese će možda morati da se konvertuju u standardni format. Nakon što se podaci unesu u skladište podataka, možda će bizi potrebno izvršiti neke operacije čišćenja i ukloniti sve upitne ili nevažeće podatke, ili izvršiti neke agregacije kao što je izračunavanje profita, marže i drugih ključnih pokazatelja učinka (KPI Key Performance Indicators). KPI-ji su način na koji se mjere poslovanja za rast i učinak.

3) **Upit za podatke (Data Querying):** Nakon što se podaci unesu i transformišu, možete postaviti upit za podatke da biste ih analizirali. Možda tražite trendove ili pokušavate da utvrdite uzrok problema u vašim sistemima. Mnogi sistemi za upravljanje bazama podataka pružaju alate koji vam omogućavaju da izvršite ad-hoc upite prema vašim podacima i generišete redovne izveštaje.

4) **Vizualizacija podataka (Data Visualization):** Podaci predstavljeni u tabelama kao što su redovi i kolone, ili kao dokumenti, nisu uvek intuitivni. Vizualizacija podataka često može biti korisna kao alat za ispitivanje podataka. Možete generisati grafikone kao što su trakasti grafikoni, linijski grafikoni, rezultati grafikona na geografskim kartama, tortni grafikoni ili ilustrirati kako se podaci mijenjaju tokom vremena. Microsoft nudi alate za vizualizaciju kao što je Power BI za pružanje bogatog grafičkog prikaza vaših podataka.



Arhitektura OLAP sistema



1. Podaci iz jednog ili vise transakcionih izvora, datoteka, real time izvora podataka mogu da se load-uju u data lake ili dwh. Load operacija ukljucuje extract, transform and load (ETL), ili extract, load and transform (ELT) procese, u kojima se podaci ciste, filtriraju I restrukturiraju za analizu. U ETL procesu, podaci se transformisu pre ucitavanja u DWH, dok se kod ELT procesa, podaci loaduju u DWH a zatim transformisu.
2. ETL process kopira podatke iz datoteka I OLTP Sistema u DWH koji je optimizovan za citanje podataka. Schema skladista podataka se zasniva na tabelama cinjenica koje sadrze numericke vrednosti koji se analiziraju (iznosi prodaje), sa povezanim tabelama dimenzija koje predstavljaju entitete prema kojima zelimo da ih merimo (kupac ili proizvod).
3. Podaci u DWH, mogu biti agregirani I ucitani u model za analiticku obradu OLAP. Nazivaju se I Cubes, agregirane numericke vrednosti iz tabele cinjenica se izracunavaju za dimenzije iz tabele dimenzija (prihod prodaje moze biti sumiran prema datumu, kupcu ili proizvodu.)
4. Podaci u jezeru podataka, DWH I analitickom modelu, se mogu pitati (queried), da bi se proizveli izvestaji, vizuelizacije I dashboard-i.

OLAP model je agregirani tip skladistenja podataka koji je optimizovan za analitiku. Posto su OLAP podaci unapred agregirani, upiti za vracanje analitika mogu se brzo pokrenuti. Agregacije podataka su po dimenzijama na razlicitim nivoima, sto omogucava drill up/down da bi se videle agregacije na vise hijerarhijskih nivoa. Npr. da bismo pronasli ukupnu prodaju po regionu, gradu ili za pojedinacnu adresu.
Razliciti tipovi korisnika mogu obavljati analiticke poslove u razlicitim fazama arhitekture. Npr.
Data scientisti mogu raditi direktno sa datotekama u data lake-u
Data analiticari mogu raditi upite direktno u dwh u svrhu izrade izvestaja I vizuelizacija.
Biznis korisnici mogu koristiti agregirane podatke u analitickom modelu u formi izvestaja ili dashboarda.

8) Jezera podataka (Data Lake)

Data lake je repozitorijum za velike kolicine neobradjenih podataka.
Posto su podaci sirovi I neobradjeni, veoma brzo se ucitavaju I azuriraju, ali podaci nisu u strukturi koja je pogodna za analizu. Mozemo zamisliti data lake kao stage-ing za unesene podatke, pre nego sto se pretvore u format koji je pogodan za analizu.

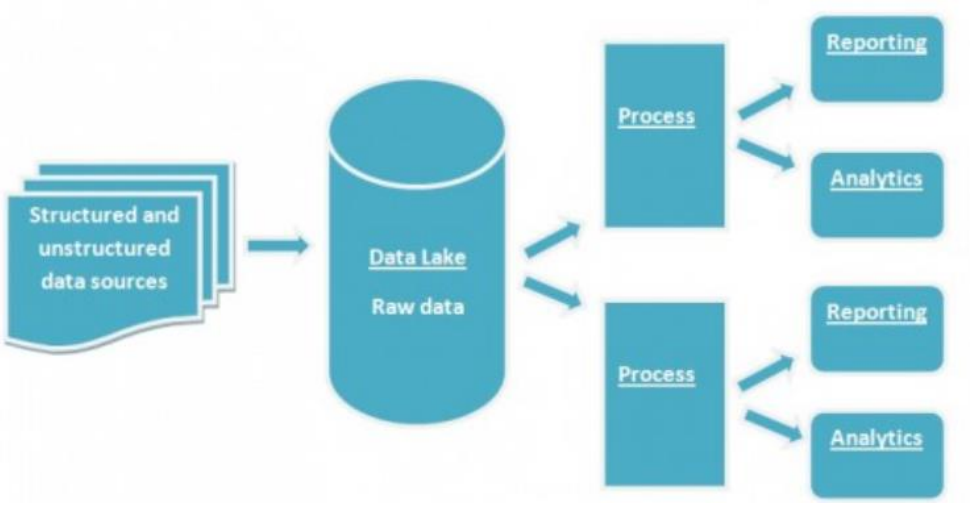
Skladista podataka takodje cuvaju velike kolicine podataka, ali su podaci u skladistu obradjeni u format za efikasnu analizu. Data lake sadrzi sirove podatke, a skladiste podataka sadrzi strukturirane podatke.

Skladišta podataka obično skladište i obrađuju određene skupove podataka. Takvo skladište podataka svakako ima svojih prednosti, ali sa druge strane skupovi podataka ponekad mogu da daju nedovoljno jasnu sliku koja se tiče celokupne poslovne aktivnosti preduzeća. Tu se kao izuzetno korisna javljaju Jezera podataka.

Jezero podataka je spremište za skladištenje koje može da uskladišti veliku količinu strukturiranih, polustrukturiranih i nestrukturiranih podataka. To je mesto za čuvanje svake vrste podataka u izvornom formatu bez fiksnih ograničenja veličine naloga ili datoteke.

U svom osnovnom principu Jezera podataka su vrlo slična Skladištima podataka. Osnovna razlika je u tome što jezera putem API servisa uvlače i obrađuju podatke iz unutrašnjih i eksternih izvora u bilo kom formatu (strukturirani i nestrukturirani) i u njihovom izvornom obliku.

Parametri	Data Lakes	Skladište podataka
Podaci	Jezera podataka čuvaju sve.	Skladište podataka fokusira se samo na poslovne procese.
Obrada	Podaci se uglavnom ne obrađuju	Visoko obrađeni podaci.
Tip podataka	Može biti nestrukturirano, polustrukturirano i strukturirano.	Uglavnom je u tabelarnom obliku i strukturi.
Korisnici	Data Lake uglavnom koristi Data Scientist.	Poslovni profesionalci široko koriste skladište podataka.
Skladište	Dizajn jezera podataka za jeftino skladištenje.	Koristi se skupo skladište koje omogućava brzo vreme odziva.
Sigurnost	Nudi manju kontrolu.	Omogućava bolju kontrolu podataka.
Šema	Šema čitanja (nema predefinisanih šema).	Šema za pisanje (predefinisane šeme).
Obrada podataka	Pomaže u brzom unošenju novih podataka.	Uvođenje novog sadržaja zahteva puno vremena.
Granularnost podataka	Podaci na niskom nivou detalja ili granularnosti.	Podaci na rezimeu ili zbirnom nivou detalja.
Alati	Može da koristi otvoreni izvor / alate poput Hadoop / Map Reduce	Uglavnom komercijalni alati.



Tako, na primer, **Jezera podataka** omogućavaju velikoj kompaniji da ima **jedno centralno mesto za skladištenje i obradu** podataka iz unutrašnjih izvora.
Podaci u Jezeru podataka se ne transformišu sve dok ne budu potrebni za analizu.
Jezera podataka omogućavaju korisnicima da im pristupe i istražuju na svoj način, bez potrebe da ih premeštaju u drugi sistem. Izveštaji dobijeni iz jezera podataka obično se izvršavaju u hodu, umesto da se redovno izvlače analitički izveštaji sa druge platforme ili recimo iz Skladišta podataka.

Trenutno ne postoji univerzalno prihvaćeni skup pravila, standarda ili smernica o upravljanju podacima u Jezerima podataka. S obzirom da je standardizacija preduslov masovnije primene neke tehnologije, ovo za sada može predstavljati neku vrstu prepreke za širu implementaciju ovog modela.

Manje kompanije mogu videti veću prednost u Skladištima podataka jer se mogu lakše fokusirati na određene grupe njima bitnih podataka i tako obezbediti potrebne informacije za svoje poslovanje.
Sa druge strane, **velike kompanije mogu više težiti ka Jezerima podataka** jer zbog svoje kompleksne organizacije i mnoštva paralelnih poslovnih aktivnosti, mogućnost rada sa velikim grupama izvornih podataka može biti veoma korisno za njihov poslovni uspeh.

9) Obrada podatak (Batching vs. Streaming)

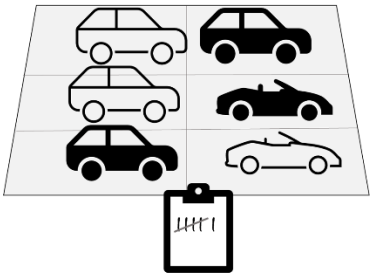
Obrada podataka je jednostavno pretvaranje sirovih podataka u smislene informacije kroz proces. Postoje dva opšta načina za obradu podataka:

1) **Paketna obrada (Batching)**, u kojoj se višestruki zapisi podataka prikupljaju i pohranjuju pre nego što se obrađuju zajedno u jednoj operaciji.
2) **Stream obrada (Streaming)**, u kojoj se izvor podataka stalno prati i obrađuje u realnom vremenu kako se događaju novi podaci.

1) Paketna obrada (Batching),

U grupnoj obradi, novopristigli elementi podataka se prikupljaju u grupu. Cela grupa se tada obrađuje u budućnosti kao serija. Vreme kada se svaka grupa obrađuje može se odrediti na više načina. Npr. možete obraditi podatke na osnovu planiranog vremenskog intervala (svaki sat), ili se može pokrenuti kada stigne određena količina podataka ili kao rezultat nekog drugog događaja.

Na primer, pretpostavimo da želite da analizirate saobraćaj na putu prebrojavanjem broja automobila na deonici puta. Pristup grupne obrade za ovo bi zahtevalo da sakupite automobile na parkingu, a zatim ih prebrojite u jednoj iteraciji dok miruju.



Ako je put, sa puno automobila koji se voze u čestim intervalima, ovaj pristup može biti nepraktičan; i neće se dobiti nikakvi rezultati dok se ne parkiraju i ne prebroje svi automobili.

Bolji primer grupne obrade je način na koji kompanije za kreditne kartice rukuju naplatom. Kupac ne prima račun za svaku posebnu kupovinu kreditnom karticom, već jedan mesečni račun za sve kupovine u tom mesecu.

Prednosti serijske obrade:
• Velike količine podataka mogu se obraditi u pogodnom trenutku.
• Može se zakazati da se pokreće u vreme kada bi računari ili sistemi inače mogli biti neaktivni, kao što je preko noći ili tokom sati van špica.

Nedostaci serijske obrade uključuju:

• Vremensko kašnjenje između unosa podataka i dobijanja rezultata.
• Svi ulazni podaci paketnog posla moraju biti spremni pre nego što se paket može obraditi. To znači da se podaci moraju pažljivo proveriti. Problemi sa podacima, greškama i padom programa do kojih dolazi tokom paketnih poslova dovode do zaustavljanja čitavog procesa. Ulazni podaci moraju se pažljivo proveriti pre nego što se posao može ponovo pokrenuti. Čak i manje greške u podacima mogu sprečiti pokretanje paketnog posla.

2) Stream obrada (Streaming)

U obradi toka, svaki novi deo podataka se obrađuje kada stigne. Za razliku od grupne obrade, nema čekanja do sledećeg intervala grupne obrade - podaci se obrađuju kao pojedinačne jedinice u realnom vremenu umesto da se obrađuje jedna po jedna serija. Streaming obrada podataka je korisna u većini scenarija gde se novi, dinamički podaci generišu na kontinuiranoj osnovi.

Npr, bolji pristup hipotetičkom problemu brojanja automobila mogao bi biti brojanjem automobila u stvarnom vremenu dok prolaze.

U ovom pristupu, ne morate čekati da se svi automobili parkiraju da biste počeli da ih obrađujete, a možete agregirati podatke u vremenskim intervalima; npr., prebrojavanjem broja automobila koji prođu svake minute.

Primeri prenosa podataka iz stvarnog sveta uključuju:

• Finansijska institucija prati promene na tržištu dionica u realnom vremenu, izračunava vrednost pod rizikom i automatski rebalansira portfelje na osnovu kretanja cena deonica
• Kompanija za online igre prikuplja podatke u realnom vremenu o interakcijama između igrača i igre i unosi podatke u svoju platformu za igre. Zatim analizira podatke u realnom vremenu, nudi podsticaje i dinamična iskustva za angažovanje svojih igrača.
• Web stranica za nekretnine koja prati podskup podataka s mobilnih uređaja i daje preporuke nekretnina koje treba posetiti u realnom vremenu na osnovu njihove geo-lokacije.

Stream obrada je idealna za operacije koje su kritične po vremenu koje zahtevaju trenutni odgovor u realnom vremenu. Npr., sistem koji nadzire zgradu zbog dima i toplote treba da aktivira alarme i otključa vrata kako bi omogućio stanovnicima da odmah pobegnu u slučaju požara.

Batching vs Streaming

Osim načina na koji grupna obrada i streaming obrada rukuju podacima, postoje i druge razlike:

Opseg podataka: *Grupna obrada* može obraditi sve podatke u skupu podataka.
Obrada streama obično ima pristup samo najnovijim primljenim podacima ili unutar vremenskog okvira (na primjr, posednjih 30 sekundi).

Veličina podataka: *Grupna obrada* je pogodna za efikasno rukovanje velikim skupovima podataka.
Stream obrada je namijenjena za pojedinačne zapise ili mikro serije koje se sastoje od nekoliko zapisa.

Performanse: Kašnjenje za *grupnu obradu* je obično nekoliko sati.
Obrada streama se obično događa odmah, s kašnjenjem od nekoliko sekundi ili milisekundi. Latencija je vrijeme potrebno za prijem i obradu podataka.

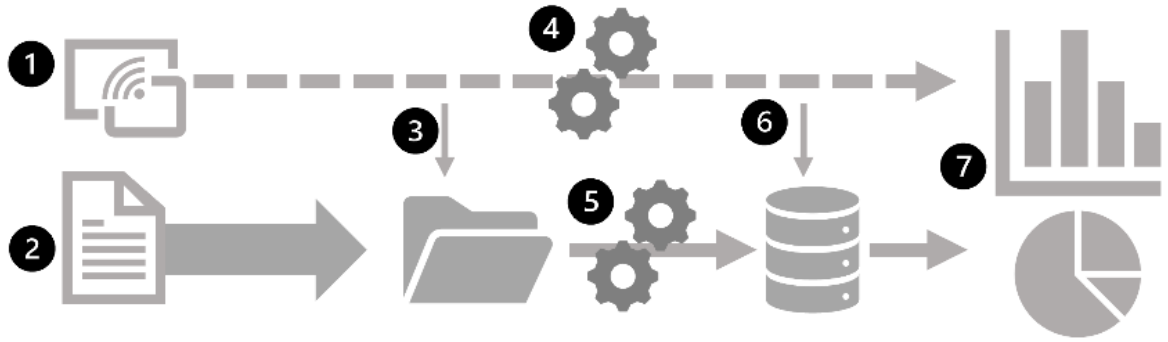
Analiza: Obično koristite *grupnu obradu* za izvođenje složene analitike.
Streaming se koristi za jednostavne funkcije odgovora, agregate ili proračune kao što su pokretni proseeci.

Kombinacija Batch i stream obrade.

Mnoga analitička rešenja velikih razmera uključuju mešavinu batch i stream obrade, omogućavajući istorijsku analizu podataka i analizu podataka u stvarnom vremenu. Uobičajeno je da rešenja za obradu toka hvataju podatke u stvarnom vremenu, obrađuju ih filtriranjem ili agregacijom i predstavljaju ih kroz kontrolne ploče i vizualizacije u stvarnom vremenu (npr., prikazujući ukupni broj automobila koji su prošli putem unutar trenutnog sat), dok se obrađeni rezultati zadržavaju u skladištu podataka za historijsku analizu uz grupno obrađene podatke (na primjer, da bi se omogućila analiza obima prometa u protekloj godini).

Čak i kada nije potrebna analiza ili vizualizacija podataka u realnom vremenu, tehnologije za striming se često koriste za hvatanje podataka u stvarnom vremenu i njihovo pohranjivanje u skladište podataka za naknadnu grupnu obradu .

Sledeći dijagram pokazuje neke načine na koje se batching i sreaming obrada mogu kombinovati u arhitekturi analize podataka velikih razmera.



1. Događaji podataka iz izvora podataka za striming se snimaju u realnom vremenu.

2. Podaci iz drugih izvora se unose u skladište podataka (često jezero podataka) za grupnu obradu.

3. Ako analitika u realnom vremenu nije potrebna, snimljeni striming podaci se upisuju u skladište podataka za naknadnu grupnu obradu.

4. Kada je potrebna analitika u realnom vremenu, tehnologija obrade toka se koristi za pripremu striming podataka za analizu ili vizualizaciju u realnom vremenu; često filtriranjem ili agregacijom podataka preko vremenskih prozora.

5. Podaci koji se ne emituju se periodično obrađuju grupno kako bi se pripremili za analizu, a rezultati se čuvaju u analitičkom skladištu podataka (koji se često naziva i skladište podataka) za historijsku analizu.

6. Rezultati obrade toka mogu se takođe zadržati u analitičkom skladištu podataka kako bi se podržala istorijska analiza.

7. Analitički i vizualizacioni alati se koriste za predstavljanje i istraživanje podataka u realnom vremenu i istorijskih podataka.