BIG DATA ANALYTICS - გამოსაცდელი კითხვები

პროფესორი: Alexander SHARMAZANASHVILI **აკადემიური წელი:** 2025-2026

კითხვები და პასუხები

1. რა არის Data და როგორ განსხვავდება Information-ისგან?

Data არის ინფორმაცია, რომელიც გადაყვანილია ციფრულ ფორმატში ეფექტური დამუშავებისთვის. Data-ს საწყისი წყარო არის სიგნალები, რომლებიც გენერირდება hardware ან software units-ის მიერ. პირველი წარმომადგენლები არის ციფრები და რიცხვები, რომელსაც ვუწოდებთ Raw Data (ნედლ მონაცემებს).

---

2. რა არის ADC და რა ფუნქცია აქვს?

ADC (Analog to Digital Converter) არის ელექტრული მოწყობილობა, რომელიც გარდაქმნის ანალოგურ სიგნალებს (ძაბვა, დენი, ხმა, ტემპერატურა) ციფრულ ფორმატში (0-ები და 1-ები). ADC მუშაობს 4 ძირითად ეტაპად: Input Signal Reading, Sampling, Quantization და Encoding. ეს არის Raw Data-ს ფორმირების ძირითადი ერთეული.

---

3. ჩამოთვალეთ Data-ს 3 ძირითადი ტიპი.

Structured Data - მონაცემები ფიქსირებულ ფორმატში (მაგ. ცხრილები, დატაბაზები). **Unstructured Data** - უცნობი ან განუსაზღვრელი სტრუქტურის მონაცემები (მაგ. ტექსტი, ფოტოები, ვიდეო). **Semi-structured Data** - შეიცავს ორივე ფორმის ელემენტებს და არ არის განსაზღვრული RDBMS table definition-ით (მაგ. XML, JSON ფაილები).

---

4. რა არის Big Data?

Big Data არის უზარმაზარი მოცულობის მონაცემთა კოლექცია, რომელიც ექსპონენციალურად იზრდება დროთა განმავლობაში. ტრადიციული data management ინსტრუმენტები ვერ ახერხებენ მის ეფექტურ შენახვას და დამუშავებას. Big Data ხასიათდება სამი ძირითადი მახასიათებლით, რომელსაც ეწოდება 3V: Volume (მოცულობა), Velocity (სიჩქარე) და Variety (მრავალფეროვნება).

5. ახსენით Big Data-ს 3V მახასიათებლები.

Volume ნიშნავს უზარმაზარ მოცულობას (zettabytes დიაპაზონი) და განსაზღვრავს არის თუ არა მონაცემი Big Data. **Velocity** არის მონაცემების გენერაციისა და დამუშავების სიჩქარე, მასიური და უწყვეტი შემოდინება. **Variety** მიუთითებს ჰეტეროგენულ წყაროებსა და მონაცემების სხვადასხვა ტიპებზე (სტრუქტურირებული და არასტრუქტურირებული).

---

6. მოიყვანეთ Big Data-ს რეალური მაგალითები.

NYSE (New York Stock Exchange) აგენერირებს დაახლოებით 1 TB ახალ trade data-ს დღეში. Facebook-ს დატაბაზებში ყოველდღიურად ემატება 500 TB ახალი მონაცემი. ერთი Jet Engine აგენერირებს 10 TB მონაცემს 30 წუთის ფრენის განმავლობაში. CERN-ის LHC-ში პროტონ-პროტონის შეჯახების დროს წარმოიქმნება 40 MB/წამში, რაც წელიწადში დაახლოებით 15 PB მონაცემს შეადგენს.

---

7. რა არის Big Data Analytics?

Big Data Analytics არის პროცესი, რომელიც მოიცავს დიდი მოცულობის მონაცემების კოლექციას, ორგანიზებას და ანალიზს, რათა აღმოაჩინოს შაბლონები და სხვა სასარგებლო ინფორმაცია. ეს პროცესი მოითხოვს სპეციალიზებულ software tools-ებს როგორიცაა predictive analytics, data mining, text mining და forecasting. მიზანია ორგანიზაციებმა გაიგონ მონაცემებში შეფარული ინფორმაცია და მიიღონ უკეთესი ბიზნეს გადაწყვეტილებები.

---

8. რა არის Data Silo და რა პრობლემებს ქმნის?

Data Silo არის data management სისტემა, რომელიც ვერ ურთიერთობს სხვა სისტემებთან და არის იზოლირებული. ის ქმნის რამდენიმე პრობლემას: მონაცემების მრავალჯერადი დუბლირება სხვადასხვა ადგილას, არასრული ხედვა მონაცემებზე და განსხვავებული პრიორიტეტები განყოფილებებს შორის. Silos-ის პრობლემის გადასაჭრელად გამოიყენება integration software, data lake ან data warehouse.

---

9. ჩამოთვალეთ Big Data Analytics-ის გამოყენების სფეროები.

Big Data Analytics გამოიყენება Healthcare-ში (პერსონალიზებული მედიცინა, smart wearables, EHR), E-commerce-ში (recommendation engines, 360° customer view), Media & Entertainment-ში (personalized content, targeted ads), Finance-ში (fraud detection, risk analysis, algorithmic trading), Travel Industry-ში (customized experience), Telecom-ში (network optimization) და Science-ში (CERN-ის particle physics experiments).

---

10. რა არის ATHENA Software და სად გამოიყენება?

ATHENA არის CERN-ის Big Data analytics პლატფორმა, რომელიც გამოიყენება particle physics experiments-ის მონაცემების დასამუშავებლად. მასში შედის 50,000+ ფაილი და 5 მილიონზე მეტი კოდის ხაზი. ATHENA-ს 6 ძირითადი კომპონენტია: Algorithms, Application Manager, Transient Data Stores, Services, Object Persistency და Data Access. დაახლოებით 50 პროგრამისტი მუშაობს მასზე ყოველდღიურად და ყოვლთვიურად 1000-ზე მეტი commit კეთდება.

---

11. რა არის Data Lake?

Data Lake არის ცენტრალური მონაცემთა რეპოზიტორია, სადაც ყველა ტიპის მონაცემი ინახება loosely defined schema-ში ნედლ (raw) ფორმატში მომავალი გამოყენებისთვის. მას აქვს schema-on-read მიდგომა, რაც ნიშნავს რომ schema განისაზღვრება მონაცემების წაკითხვის დროს. Data Lake უზრუნველყოფს metadata catalogue-ს, data governance-ს, scalability-ს და ნებისმიერი ტიპის მონაცემების ინტეგრაციას.

---

12. რა არის Data Warehouse?

Data Warehouse არის ცენტრალური რეპოზიტორია სტრუქტურირებული მონაცემებისთვის, რომელიც ოპტიმიზებულია querying და analysis-ისთვის. იგი იყენებს schema-on-write სტრატეგიას, რაც ნიშნავს რომ მონაცემები წინასწარ სტრუქტურირდება predefined schema-ს მიხედვით. Data Warehouse ეფუძნება ETL (Extract-Transform-Load) პრინციპს და უზრუნველყოფს სწრაფ queries-ს და complex analysis-ს.

---

13. შეადარეთ Data Lake და Data Warehouse.

Data Lake ინახავს ყველა ტიპის raw, unprocessed მონაცემებს (structured, semi-structured, unstructured) schema-on-read მიდგომით და დაბალი ღირებულებით. იგი გამოიყენება data scientists და engineers-ის მიერ. Data Warehouse ინახავს processed, structured მონაცემებს schema-on-write მიდგომით და მაღალი ღირებულებით. მას იყენებენ business analysts BI tools-ის საშუალებით. Data Lake უფრო მოქნილია, ხოლო Warehouse უფრო სწრაფი query performance-ით.

---

14. რა არის Data Lakehouse?

Data Lakehouse არის ჰიბრიდული არქიტექტურა, რომელიც აერთიანებს Data Lake-ისა და Data Warehouse-ის უპირატესობებს. ფორმულა: Lakehouse = Data Lake + Data Warehouse. იგი უზრუნველყოფს Data Lake-ის flexibility-სა და დაბალ ღირებულებას, ასევე Data Warehouse-ის performance-სა და data quality-ს. Lakehouse იყენებს cloud object storage-ს open file formats-ით და transactional/metadata layer-ს data integrity-სთვის.

---

15. რა არის Schema-on-read და Schema-on-write?

Schema-on-read ნიშნავს, რომ schema განისაზღვრება მონაცემების წაკითხვის დროს (Data Lake-ში). მონაცემები ინახება raw ფორმატში და schema გამოიყენება query-ის დროს, რაც უზრუნველყოფს მოქნილობას. Schema-on-write ნიშნავს, რომ schema განისაზღვრება მონაცემების ჩაწერამდე (Data Warehouse-ში). მონაცემები წინასწარ სტრუქტურირდება და შემდეგ ინახება, რაც უზრუნველყოფს სწრაფ querying-ს.

---

16. რა არის ETL და ELT?

ETL (Extract-Transform-Load) არის Data Warehouse-ის პროცესი: მონაცემები იღება წყაროდან, გარდაიქმნება სასურველ ფორმატში და შემდეგ იტვირთება warehouse-ში. ELT (Extract-Load-Transform) არის Data Lake-ის პროცესი: მონაცემები იღება, იტვირთება raw ფორმატში lake-ში და შემდეგ გარდაიქმნება საჭიროების მიხედვით. ETL უზრუნველყოფს data quality-ს, ხოლო ELT უფრო მოქნილია.

---

17. რა არის Scalability და რა ტიპები არსებობს?

Scalability არის სისტემის უნარი გაუმკლავდეს მონაცემების მზარდ რაოდენობას performance-ის დაკარგვის გარეშე. არსებობს ორი ძირითადი ტიპი: Vertical Scaling (მეტი CPU, RAM, HDD ერთ მანქანაში - იოლი, მაგრამ შეზღუდული) და Horizontal Scaling (მონაცემების/დავალებების განაწილება მრავალ მანქანაზე - პრაქტიკულად unlimited, მაგრამ კომპლექსური). Big Data-ში ძირითადად გამოიყენება Horizontal Scaling.

---

18. რა არის Metadata და რატომ არის მნიშვნელოვანი?

Metadata არის "მონაცემები მონაცემების შესახებ", რომელიც აღწერს Data Lake-ში შენახულ ინფორმაციას: წყარო, ფორმატი, schema, შექმნის თარიღი და ხარისხის მეტრიკები. Metadata საშუალებას აძლევს იპოვო საჭირო მონაცემები, უზრუნველყოფს data quality და governance-ს, აჩვენებს data lineage-ს და აჩქარებს query processing-ს. Metadata-ს გარეშე Data Lake იქცევა "data swamp"-ად (მონაცემთა ჭაობად).

---

19. რა არის Mono-Zone Architecture?

Mono-Zone არის უმარტივესი Data Lake არქიტექტურა ერთი ზონით, სადაც ყველა raw data ინახება native ფორმატში. უპირატესობებია: მარტივი იმპლემენტაცია, ეფექტური ღირებულება და გაფართოების მოქნილობა. ნაკლოვანებებია: არ არის data structure რაც იწვევს governance პრობლემებს, მონაცემები შეიძლება გახდეს არასამართავი და performance bottleneck. გამოიყენება proof of concept-ისთვის და initial data exploration-ისთვის.

---

20. რა არის Lambda Architecture?

Lambda Architecture არის სამშრიანი არქიტექტურა batch და real-time data processing-ისთვის. მასში შედის: Batch Layer (persistent memory, historical data), Speed Layer (transient/incremental data, real-time processing) და Serving Layer (მონაცემების მიწოდება end-users-ისთვის). უპირატესობებია high-throughput, fault tolerance და time series analysis. ნაკლოვანებებია კომპლექსურობა და resource-intensive. მაგალითი: CERN-ის ATHENA software.

---

21. რა არის Kappa Architecture?

Kappa Architecture არის Lambda-ს გამარტივებული ვერსია, რომელიც ამოიღებს Batch Layer-ს და ტოვებს მხოლოდ Speed Layer-ს real-time/streaming processing-ისთვის. უპირატესობებია: მარტივი (2 layer), ნაკლები complexity, resource efficient და მხოლოდ ერთი processing engine. ნაკლოვანებებია: recomputing linearly იზრდება data-სთან ერთად და არ არის იდეალური full historical data processing-ისთვის. გამოიყენება IoT analytics და electricity forecasting-ისთვის.

---

22. რა არის Data-Pond Architecture?

Data-Pond Architecture შედგება 5 ლოგიკურად გამოყოფილი pond-დან: Raw-data pond (ingested raw data, staging area), Analog-data pond (semi-structured IoT data), Application-data pond (Data Warehouse-ის მსგავსი, ETL), Textual-data pond (unstructured text) და Archival-data pond (inactive data). უპირატესობებია enhanced data management და governance, ხოლო ნაკლოვანება ის არის, რომ ორიგინალი raw data იკარგება და მაღალი management costs.

---

23. რა არის Zaloni Architecture?

Zaloni არის ერთ-ერთი ყველაზე გავრცელებული Data Lake არქიტექტურა 5 ზონით: Transient Landing Zone (დროებითი შენახვა, compliance checks), Raw Zone (მუდმივი შენახვა original ფორმატში), Trusted Zone (quality და compliance checked data), Refined Zone (end-user-ებისთვის მორგებული) და Sandbox (test area, unrestricted access). უპირატესობებია enhanced quality & security, better governance და hot/cold storage optimization. გამოიყენება banking, healthcare და fish farming-ში.

---

24. რა არის Multi-Zone Functional Architecture?

Multi-Zone Functional არის ჰიბრიდული არქიტექტურა 4 ზონით: Raw Ingestion Zone (batch და real-time data ingestion), Process Zone (data preparation, intermediate storage), Access Zone (visualization, ML, BI analytics) და Govern Zone (security, quality, lifecycle, metadata management). უპირატესობებია flexibility, scalability და continuous processing loop. გამოიყენება healthcare data, air traffic management და real-time IoT sensor data-სთვის.

---

25. რა არის Functional-Layered Architecture?

Functional-Layered არის საფეხურებრივი არქიტექტურა 4 layer-ით: Ingestion Layer (heterogeneous data collection), Storage Layer (metadata + raw data repositories), Transformation Layer (cleansing, transformation, integration) და Interaction Layer (user access, exploration, queries, visualization). უპირატესობებია ლოგიკური framework, independent maintenance და modularity. ნაკლოვანებაა governance enforcement-ის სირთულე და linear data flow oversimplification. გამოიყენება environmental monitoring და cyberattack detection-ისთვის.

---

26. რა არის Lakehouse Architecture?

Lakehouse Architecture აერთიანებს Data Lake და Data Warehouse-ის features-ს ერთ პლატფორმაში. იგი იყენებს cloud object storage-ს open file formats-ით და transactional/metadata layer-ს data integrity-სთვის. უპირატესობებია: simplifies enterprise analytics, single source of truth, unified data format, real-time analytics support. ნაკლოვანებებია transformation errors-ის შესაძლებლობა და performance loss დიდ data-ზე. გამოიყენება biomedical research და health data analytics-ისთვის.

---

27. როგორ მუშაობს CERN-ის LHC Trigger System?

LHC აგენერირებს დაახლოებით 600 მილიონ proton-proton collision-ს წამში, თითოეული ~1 MB data-ს წარმოქმნის, რაც 600 GB/s-ს შეადგენს და შეუძლებელია შენახვა. Trigger System უარყოფს “uninteresting” events-ს და ინახავს მხოლოდ “interesting” ones. ATLAS Trigger System ამცირებს 600M events/s-ს დაახლოებით 200 events/s-მდე, რაც 200 MB/s-ს (4 PB/წელიწადში) შეადგენს. ეს არის smart filtering-ის რეალური მაგალითი Big Data-ში.

---

28. რა არის Data Governance?

Data Governance არის პროცესი, რომელიც უზრუნველყოფს მონაცემების მართვის სტანდარტებს, პოლიტიკებსა და პროცედურებს. მასში შედის data quality (სიზუსტე, consistency), data security (access control, encryption), data lifecycle (შექმნა, შენახვა, არქივირება, წაშლა), compliance (რეგულაციებთან შესაბამისობა), metadata management და access management. Data Governance თავიდან აიცილებს “data swamp”-ს და უზრუნველყოფს trustworthy analysis.

---

29. რა არის IoT და როგორ უკავშირდება Big Data-ს?

IoT (Internet of Things) არის ფიზიკური ობიექტების ქსელი, რომლებიც აღჭურვილია sensors, software და ტექნოლოგიებით მონაცემების გაცვლისთვის. IoT devices აგენერირებენ high velocity და high volume მონაცემებს real-time streaming ფორმატში, რაც არის Big Data-ს კლასიკური მაგალითი. IoT data გამოიყენება Smart wearables-ში (healthcare), electricity sensors-ში (energy optimization) და university building sensors-ში (IoT management).

---

30. რა არის Hot და Cold Storage?

Hot Storage არის სწრაფი და ძვირი შენახვა ხშირად წვდომადი მონაცემებისთვის (SSD, RAM), გამოიყენება active databases და real-time analytics-ისთვის. Cold Storage არის ნელი და იაფი შენახვა იშვიათად წვდომადი მონაცემებისთვის (HDD, tape, cloud archive), გამოიყენება historical logs და archival data-სთვის. ეს სტრატეგია ბალანსირებს performance-სა და ღირებულებას და გამოიყენება Zaloni Architecture-ში.

---

31. რა არის Data Lineage?

Data Lineage არის მონაცემების სიცოცხლის ციკლის tracking - საიდან მოდის data, როგორ იცვლება და სად მიდის. იგი აჩვენებს source-ს, transformations-ს, movement-ს და usage-ს. Data Lineage აძლევს trust & quality-ს, უზრუნველყოფს compliance-ს (GDPR), საშუალებას აძლევს debugging-ს და impact analysis-ს. იგი იმპლემენტირებულია metadata management tools-ისა და Govern Zone-ის საშუალებით.

---

32. რა არის Sampling Frequency და Nyquist Theorem?

Sampling Frequency (Fs) არის სიხშირე, რომლითაც ADC იღებს samples უწყვეტი სიგნალიდან. Nyquist-Shannon Sampling Theorem ამბობს, რომ სიგნალის სწორი reconstruction-ისთვის Fs უნდა იყოს მინიმუმ ორჯერ მეტი ვიდრე სიგნალის უმაღლესი frequency ($Fs \geq 2 \times fmax$). თუ $Fs < 2 \times fmax$, ხდება aliasing (დამახინჯება). მაგალითად, audio signal-ისთვის (20 kHz) CD quality sampling არის 44.1 kHz.

---

33. რა არის Distributed Computing და მისი როლი Big Data-ში?

Distributed Computing არის პარადიგმა, სადაც დავალებები ნაწილდება მრავალ კომპიუტერზე (nodes), რომლებიც მუშაობენ ერთობლივად. ძირითადი პრინციპებია horizontal scaling, parallelization, data locality და fault tolerance. Big Data-ში Distributed Computing უზრუნველყოფს storage scalability (მონაცემები ნაწილდება clusters-ზე, მაგ. HDFS), process scalability (პარალელური processing, მაგ. MapReduce) და linear performance scaling. მაგალითებია CERN-ის Worldwide Grid და Hadoop Clusters.

---

34. რა პრობლემებს წყვეტს Data Lake?

Data Lake აერთიანებს dispersed data sources და წყვეტს Data Silos პრობლემას. Schema-on-read მიდგომა აძლევს flexibility-ს schema rigidity-ს ნაცვლად. იგი იღებს structured, unstructured და semi-structured მონაცემებს (variety). Data Lake უზრუნველყოფს horizontal scaling-ს და იაფ storage-ს commodity hardware ან cloud-ზე. ამასთან, Data Lake შეიძლება გახდეს "data swamp" თუ არ არის სწორი metadata და governance.

---

35. რა არის Data Swamp და როგორ ავიცილოთ თავიდან?

Data Swamp არის Data Lake, რომელიც გახდა არასამართავი, არახარისხიანი მონაცემების "ჭაობი", სადაც data ვერ იპოვება და არ არის გამოსადეგი. ეს ხდება poor metadata-ს, no governance-ს, lack of structure-ისა და no lifecycle management-ის გამო. თავიდან ასაცილებლად საჭიროა strong metadata management, data governance (Govern Zone), zone-based organization, data catalog, lifecycle policies და quality gates. Zaloni Architecture არის კარგი მაგალითი governance-ით.

---

36. რა არის Open File Formats და რატომ არის მნიშვნელოვანი?

Open File Formats არის ფაილის ფორმატები, რომელთა სპეციფიკაცია საჯაროდ ხელმისაწვდომია და არ არის vendor-specific. ძირითადი formats-ებია: Parquet (columnar, შესანიშნავი compression), ORC (columnar, built-in indexing), Avro (row-based, schema embedded), JSON (flexible) და CSV (universal). Open formats თავიდან აიცილებს vendor lock-in-ს, უზრუნველყოფს interoperability-ს, future-proofing-ს და cost optimization-ს. Lakehouse Architecture ეფუძნება open formats-ს.

---

37. რა არის ACID Properties?

ACID არის ოთხი თვისება database transaction-ების reliability-სთვის: **Atomicity** (all or nothing - transaction მთლიანად ასრულდება ან საერთოდ არა), **Consistency** (database რჩება valid state-ში constraints-ის მიხედვით), **Isolation** (concurrent transactions არ ერევა ერთმანეთში) და **Durability** (committed transaction-ის შედეგი permanent-ია crash-ის შემდეგაც). Traditional Data Lake-ში ACID არ არის, მაგრამ Lakehouse-ში transactional layer (Delta Lake, Iceberg) უზრუნველყოფს ACID properties-ს.

---

38. როგორ განსხვავდება Batch და Real-time Processing?

Batch Processing ამუშავებს მონაცემებს დაგროვების შემდეგ დიდ მოცულობაში (high latency, საათები/დღეები), გამოიყენება historical analysis და reports-ისთვის. Real-time/Stream Processing ამუშავებს მონაცემებს დაუყოვნებლივ პატარა chunks-ებად (low latency, წამები/milliseconds), გამოიყენება monitoring, alerts და fraud detection-ისთვის. Lambda Architecture აერთიანებს ორივეს: Batch Layer + Speed Layer, რაც იდეალურია hybrid workloads-ისთვის.

---

39. როგორ უნდა შევარჩიოთ Data Lake არქიტექტურა?

არჩევანი დამოკიდებულია use case-ზე: **Mono-Zone** proof of concept-ისთვის, **Lambda** batch+real-time და fault tolerance-ისთვის (CERN), **Kappa** streaming-only simple use cases-ისთვის, **Data-Pond** diverse data types-ისთვის, **Zaloni** enterprise governance და compliance-ისთვის (banking, healthcare), **Multi-Zone Functional** flexibility+governance-ისთვის, **Functional-Layered** modularity-სთვის, **Lakehouse** Lake+Warehouse unified platform-ისთვის. არ არსებობს “best” architecture - დამოკიდებულია requirements-ზე.

---

40. რა არის Quantization Error?

Quantization Error არის შეცდომა, რომელიც წარმოიქმნება ADC-ის quantization ეტაპზე, როცა უწყვეტი ანალოგური სიგნალი მიესადაგება დისკრეტულ დონეებს. მაგალითად, თუ ანალოგური მნიშვნელობაა 3.7V და ADC resolution არის 3-bit (8 დონე), უახლოესი დონე იქნება 4, და error იქნება 0.3V. მეტი bit resolution → მეტი precision → ნაკლები error. მაღალი precision sensors აგენერირებენ უფრო ზუსტ Big Data-ს.

---

41. რა არის Sandbox Zone?

Sandbox Zone არის იზოლირებული ტესტ გარემო Data Lake-ში, სადაც data scientists-ს აქვთ unrestricted access data exploration-ისთვის. იგი უზრუნველყოფს innovation-ს (თავისუფალი experimentation), risk-free testing-ს (არ ეხება production data), advanced analytics-ს (ML experiments) და faster time-to-insight-ს. Sandbox არის Zaloni Architecture-ის explicit zone და Multi-Zone Functional-ის Process zone-ის ნაწილი. Trade-off არის governance vs innovation.

---

42. რა არის Immutability Data Lake-ში?

Immutability (Append-Only Model) არის პრინციპი, რომ მონაცემები არ იშლება და არ იცვლება, მხოლოდ ემატება. ახალი data append-დება, შესწორება ახალი ვერსიის append-ით ხდება და ისტორია მთლიანად ინახება. უპირატესობებია time series analysis, audit trail, fault tolerance (recompute from history), reproducibility და simplicity. ნაკლოვანებებია storage overhead და GDPR challenges ("right to be forgotten"). Lambda Architecture explicitly immutable-ია.

---

43. რა როლი აქვს Metadata Layer-ს Lakehouse-ში?

Metadata Layer (transactional layer) Lakehouse-ში უზრუნველყოფს data integrity-ს ACID properties-ის საშუალებით. იგი tracks schema-ს, version-ებს, transactions-ს და file changes-ს transaction log-ში (მაგ. Delta Lake-ის _delta_log). ეს საშუალებას აძლევს safe updates/deletes-ს, time travel-ს (წინა versions-ის წაკითხვა), concurrent reads/writes-ს და schema evolution-ს. Metadata Layer არის ის, რაც Data Lake-ს აქცევს Lakehouse-ად.

---

44. როგორ ხდება Track Finding CERN-ის ATLAS-ში?

Track Finding არის პროცესი, რომელიც იდენტიფიცირებს ნაწილაკების ტრაექტორიებს მილიონობით data points-დან. გამოიყენება Kalman Filter algorithm: იწყებს რამდენიმე hit point-დან (seeds), extrapolate-ს შემდეგ position-ს ფიზიკის კანონების გამოყენებით, update-ს track estimate-ს actual hits-თან შედარებით და iterate-ს ყველა detector layer-ისთვის. Kalman Filter invented for missile control-ისთვის და ახლა გამოიყენება particle physics-ში real-time 200 MB/s data-ზე.

---

45. რა უპირატესობები აქვს Cloud Object Storage-ს?

Cloud Object Storage უზრუნველყოფს unlimited scalability-ს (pay-as-you-grow), cost-effectiveness-ს (იაფი, tiered pricing), durability-ს (99.999999999% AWS S3), accessibility-ს (HTTP/REST API, ხელმისაწვდომი ყველგან), open formats support-ს (Parquet, ORC, JSON) და metadata support-ს (versioning). Lakehouse Architecture ეფუძნება cloud object storage-ს (AWS S3, Azure Blob, Google Cloud Storage). ეს აძლევს Data Lake-ს unlimited growth potential-ს გარე hardware-ის გარეშე.

---

46. რა განსხვავებაა Vertical და Horizontal Scaling შორის?

Vertical Scaling ნიშნავს მეტი CPU, RAM, HDD-ს დამატებას ერთ მანქანაში - იოლია იმპლემენტაცია, მაგრამ შეზღუდულია hardware capacity-ით და ძვირია. **Horizontal Scaling** ნიშნავს მონაცემების/დავალებების განაწილებას მრავალ მანქანაზე - პრაქტიკულად unlimited, fault tolerant და cost-effective commodity hardware-ით, მაგრამ კომპლექსურია და საჭიროებს distributed computing expertise-ს. Big Data PB-scale-ზე horizontal scaling აუცილებელია.

---

47. რა თავისებურებები აქვს Healthcare Big Data-ს?

Healthcare Big Data ხასიათდება extreme variety-ით (structured EHR, semi-structured HL7, unstructured doctor notes და medical images, streaming ICU monitors, genomic DNA sequences - 3GB per person), high sensitivity-ით (HIPAA, GDPR compliance mandatory, patient privacy critical), life-critical ბუნებით (errors can kill, real-time alerts vital) და long-term storage requirements-ით (lifetime retention). გამოიყენება predictive analytics, personalized medicine, real-time alerting, EHR და medical imaging-ისთვის. რეკომენდებული არქიტექტურაა Zaloni ან Lakehouse.

---

48. როგორ იზომება Big Data Analytics-ის წარმატება?

წარმატება იზომება technical metrics-ით (query response time, throughput, latency, scalability, storage efficiency, uptime), business metrics-ით (ROI, time-to-insight, decision quality, accuracy), use-case specific metrics-ით (healthcare: lives saved; e-commerce: conversion rate; finance: fraud detection rate) და governance metrics-ით (data quality score, compliance rate, metadata coverage). წარმატების ფორმულა: Success = Technical Performance × Business Value × User Adoption. ყველა სამი კომპონენტი აუცილებელია.

---

49. რა გაკვეთილები შეიძლება ავიღოთ CERN-ის გამოცდილებიდან?

CERN-ის გამოცდილება გვასწავლის: filtering is critical (600M → 200 events/s), Lambda Architecture works at scale (ATHENA), distributed computing is mandatory (Worldwide Grid), software complexity is inevitable (50K+ files), algorithms matter (Kalman Filter), long-term preservation planning (decades retention), collaboration at scale (worldwide), incremental processing (multi-level trigger), trade-offs are necessary (perfect vs good) და Big Data enables breakthrough discoveries (Higgs Boson, Nobel Prize). Success formula: Smart filtering + Right architecture + Distributed computing + Strong engineering + Collaboration.

---

50. რა არის Data Lake-დან Data Lakehouse-მდე ევოლუცია?

Data Lake-მა გადაჭრა data silos და schema rigidity პრობლემები, მაგრამ შექმნა ახალი გამოწვევები: data swamp (poor metadata), security challenges, quality issues და slow performance. Data Warehouse აძლევდა quality და performance-ს, მაგრამ იყო ძვირი და rigid. Data Lakehouse გამოჩნდა როგორც hybrid გადაწყვეტა, რომელიც აერთიანებს Lake-ის flexibility და დაბალ cost-ს Warehouse-ის ACID properties, performance და quality-სთან. Lakehouse = Lake Storage + Warehouse Guarantees, იმპლემენტირებული transactional layer-ის (Delta Lake, Iceberg) საშუალებით.

დასკვნა

ეს 50 კითხვა მოიცავს ყველა ძირითად თემას Big Data Analytics კურსიდან: - ძირითადი კონცეფციები და დეფინიციები - Big Data მახასიათებლები და გამოყენება - Data Lake, Warehouse, Lakehouse - 8 Data Lake არქიტექტურა - Technical concepts (Scalability, ACID, Metadata) - CERN case study - Best practices

წარმატებები გამოცდაზე!