# Raj Kumar Rai – Bergfreunde Technical Interview Preparation (Detailed Version)

**Date:** 29 October
**Interviewer:** Christopher Barth (Head of Data)
**Duration:** 45 minutes
**Focus:** Databricks Lakehouse · MLOps · Generative AI · Cost & Architecture

## 1. Elevator Pitch (2 Minutes)

I'm Raj Kumar Rai — currently Lead Data Engineer at Freeletics in Munich. Over my 19+ years in data, I've specialized in building scalable, cloud-based data architectures. My recent focus has been designing and optimizing Databricks-based Lakehouse platforms on AWS.

At Freeletics, I architected an enterprise Lakehouse that unifies batch and streaming data pipelines using PySpark, Autoloader, and Delta Live Tables. I implemented Unity Catalog for governance, MLflow for MLOps lifecycle management, and Databricks Jobs with CI/CD automation for reproducibility. This platform powers analytics, personalization, and AI workflows across the company.

I collaborate with data scientists to design reusable Feature Stores and automate training-to-serving workflows. Recently, I've extended this to Generative AI — using Mosaic AI and AgentBricks to integrate LLMs directly with structured Delta data. These agents generate metadata summaries, automate quality reports, and accelerate data discovery.

Bergfreunde's goal of a modern, scalable, and AI-ready Lakehouse strongly resonates with my background in bridging data engineering, MLOps, and AI innovation using Databricks technologies.

## 2. Key Technical Topics

• Databricks Lakehouse architecture (Bronze–Silver–Gold layers)
• Delta Lake, Unity Catalog, MLflow, Feature Store
• LakeFlow / Delta Live Tables / PySpark
• Batch & Streaming integration (Autoloader, Kinesis)
• Cost optimization: spot clusters, caching, Z-ordering, tiered S3
• MLOps lifecycle (training, registry, serving, retraining)
• Generative AI via Mosaic AI & AgentBricks
• Governance, lineage, CI/CD automation

## 3. Core Questions & Answers (Detailed)

**Q1: Describe your Lakehouse architecture.**

I designed a multi-layer Delta Lakehouse architecture with clear separation between Bronze (raw), Silver (cleaned), and Gold (consumption) layers. Data from multiple sources like app events, payments, and APIs are ingested via Autoloader and Kinesis into Bronze. Transformations and deduplication happen in Silver, while Gold provides analytics and ML-ready aggregates. Delta Lake ensures ACID transactions and schema enforcement. Delta Live Tables handle data quality and monitoring, while Unity Catalog manages lineage, auditing, and access control. This structure scales cost-effectively and supports both BI and ML use cases.

**Q2: How do you build ML pipelines?**

I build end-to-end ML pipelines using Databricks MLflow and Feature Store. The process begins with feature engineering on curated Silver data, registering reusable features in Feature Store for model consistency. Models are trained and tracked via MLflow Tracking, then promoted through the MLflow Model Registry for version control. Deployment occurs through Databricks Model Serving or batch inference jobs. Retraining is automated via Airflow or Databricks Workflows triggered by data freshness SLAs. This ensures reproducibility, governance, and scalability across ML workflows.

**Q3: How do you integrate Generative AI?**

Using Mosaic AI and AgentBricks within Databricks, I've integrated LLM capabilities directly into the data ecosystem. Mosaic AI handles fine-tuning, evaluation, and inference orchestration. AgentBricks enables multi-step agents that access Delta data, execute SQL, and use Vector Search for contextual recall. For instance, I built an internal data assistant that answers catalog questions and generates metadata summaries automatically. Another agent drafts analytical summaries by reading Delta tables and event logs. This brings intelligence and automation directly into data workflows.

**Q4: How do you ensure cost efficiency and performance?**

Cost optimization is an ongoing practice. I replaced static clusters with job clusters using spot instances and autoscaling, achieving 30% compute savings. Partition pruning, Z-ordering, and caching were applied to accelerate frequent queries. Data lifecycle policies automatically move older data from S3 Standard to Glacier. I also monitor query metrics via Databricks REST APIs and created S3 cost dashboards for proactive insights. The goal is a self-tuning Lakehouse that minimizes waste and maximizes ROI.

**Q5: How do you collaborate with Data Science and BI teams?**

I act as the bridge between engineering, BI, and DS teams. BI uses curated Gold tables in Power BI/Looker, while DS accesses Feature Store data for training. We maintain strict data contracts with schema versioning to avoid production breakages. ML models use the same underlying curated data, ensuring consistency between experimentation and

production. Frequent syncs ensure alignment on business KPIs and new features. This collaboration model promotes data trust and faster delivery cycles.

**Q6: How do you handle governance and lineage?**

Governance is built into every layer via Unity Catalog. It controls permissions at schema, table, and column level, with lineage tracking that maps data from raw sources to models. I also use MLflow for experiment lineage — linking trained models to datasets and parameters. Metadata and documentation are auto-generated through Databricks Jobs and stored in a shared workspace. This ensures auditability and compliance without slowing innovation.

**Q7: What trends in Data & AI excite you?**

I’m excited about two trends: (1) Lakehouse-native LLM integration — using tools like Mosaic AI and AgentBricks to bring intelligent agents into data operations; (2) the convergence of analytics, ML, and GenAI under one governed platform. These trends enable true self-service AI capabilities where users can query, analyze, and generate insights conversationally on enterprise data.

## 4. Generative AI & AgentBricks Use Cases (From Raj's Work)

**Use Case 1 – Automated Metadata Enrichment**
Developed an internal agent using AgentBricks that connects Databricks Unity Catalog and LLMs. The agent scans Delta tables and generates structured metadata descriptions, column summaries, and freshness insights. This automated documentation reduced manual metadata work by 80%.

**Use Case 2 – Data Quality Summarization Agent**
Using Mosaic AI orchestration, built an LLM-powered summarizer that reviews daily pipeline logs and produces English-language quality reports (e.g., anomalies, null spikes, schema drifts). These reports are shared via Slack for proactive monitoring.

**Use Case 3 – Vector Search–Enabled Knowledge Assistant**
Integrated Databricks Vector Search with AgentBricks to power a question-answer assistant. It allows business teams to ask natural-language questions like "Show top-performing campaigns for the last quarter" and fetch results from Delta tables. This connected structured analytics to conversational AI, driving adoption across departments.

## 5. Smart Closing Questions

• How far along is Bergfreunde in its Lakehouse journey — are you already on Databricks or evaluating migration?
• What are the current priorities: scalability, ML integration, or analytics enablement?
• Are there any personalization or GenAI initiatives planned where Mosaic AI and AgentBricks could accelerate value creation?

## 6. Final Tips

• Always link your answers to business impact (e.g., cost reduction, faster delivery, AI enablement).
• Emphasize Databricks as an enabler — not just a tool.
• Highlight Mosaic AI and AgentBricks when discussing future trends.
• Maintain a leadership tone and show you think end-to-end.
• End by expressing curiosity about their roadmap and challenges.