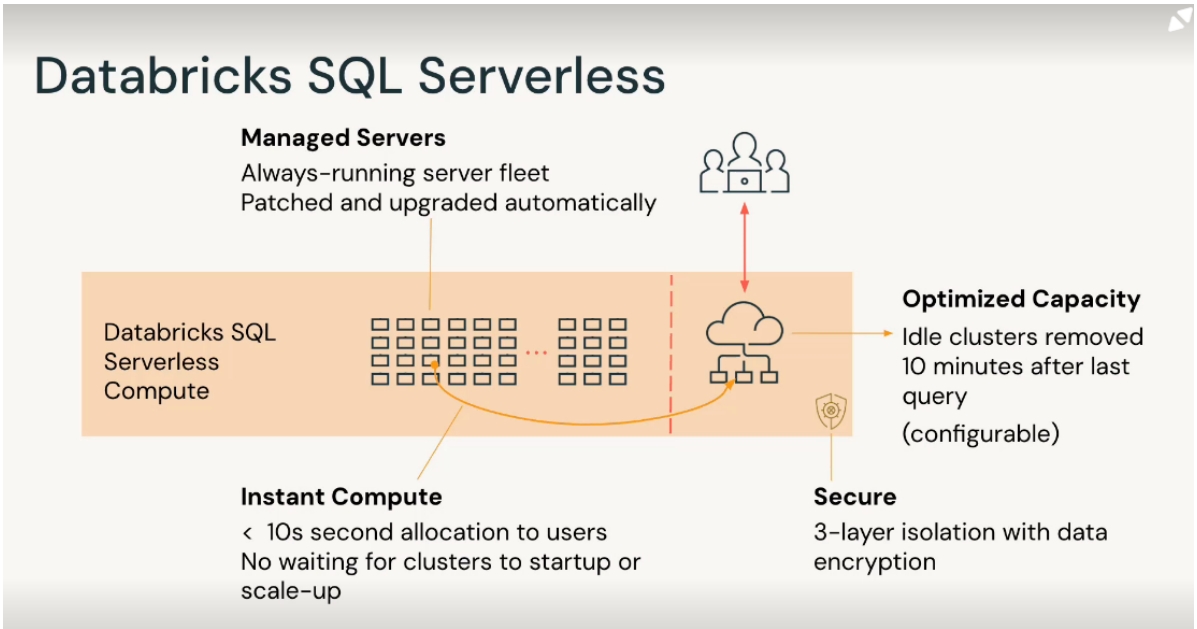
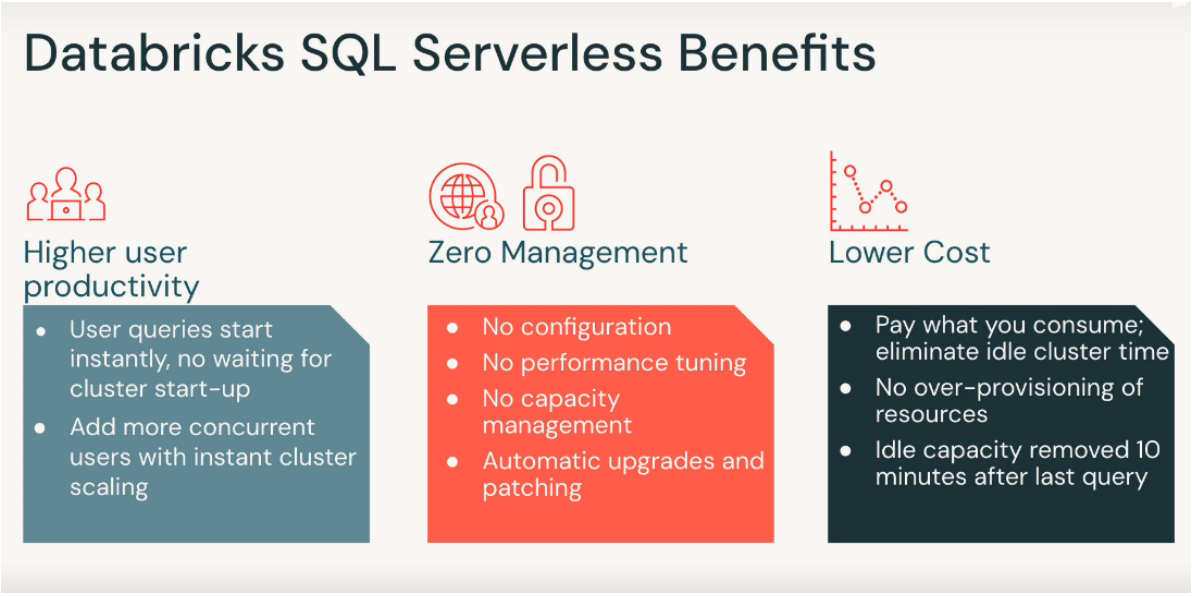
💙

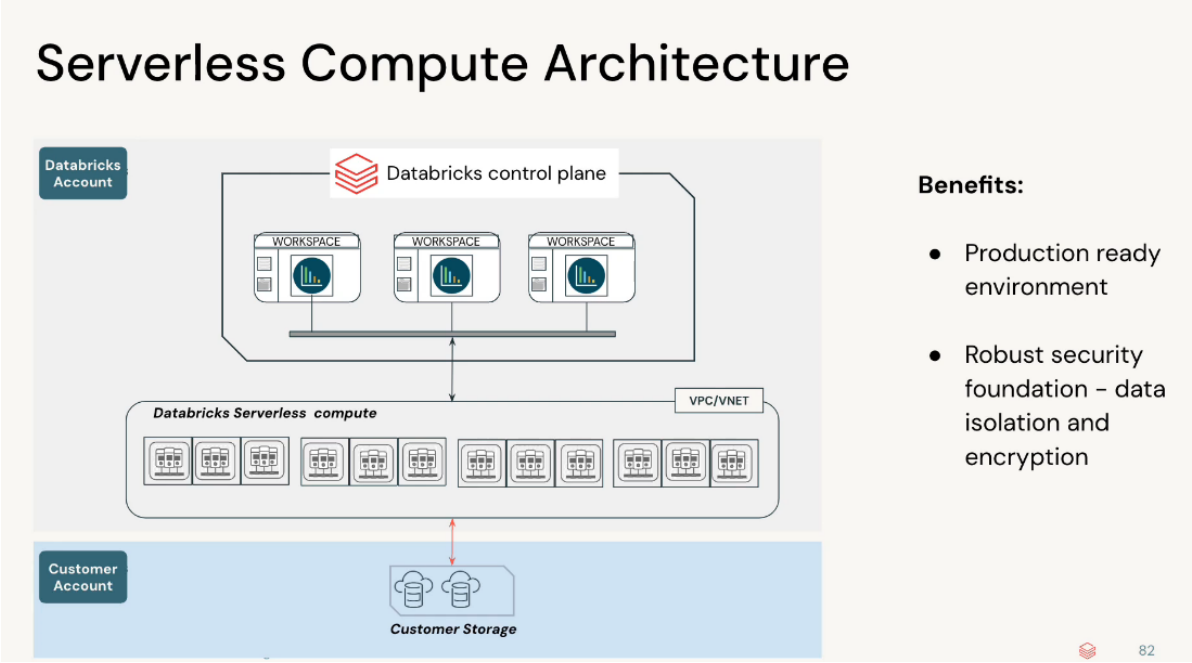
Revisão/Conceitos

- **Data Explorer** It can be used to view metadata and data, as well as view/change permissions.
- **Databricks SQL** é um data warehouse sem servidor construído na arquitetura lakehouse que permite executar todas as suas cargas de trabalho de BI e ETL em escala com preço/desempenho até 12x melhor, um modelo de governança unificado, formatos abertos e APIs e suas ferramentas preferidas. O **Databricks SQL** utiliza nosso mecanismo de consulta vetorial de última geração, **Photon** , e vem com milhares de otimizações para fornecer o melhor desempenho para todas as suas ferramentas, tipos de consulta e aplicativos do mundo real. Isso inclui E/S preditiva alimentada por IA que elimina o ajuste de desempenho, como indexação, por meio da pré-busca inteligente de dados usando redes neurais.
- **Databricks SQL** is a part of databricks data intelligence platform, built on top lakehouse architecture, the databricks DI (Data Intelligence) platform is more than integration beetween the Data Warehouse and the Delta Lake. É uma arquitetura que suporta o trabalho de um analista de dados, engenheiro de dados e cientista de dados no mesmo local. Com o Databricks é possível trabalhar com batch data ou live stream data.
- **Databricks SQL provides:**
  - Better price / performance than other cloud data warehouses.
  - Simplify discovery and sharing of new insights.
  - Connect to familiar BI tools, like Tableau or Power BI.
  - Simplified administration and unified governance (with Unit Catalog)
- **Databricks SQL Serverless** elimina a necessidade de gerenciar, configurar ou dimensionar a infraestrutura em nuvem no lakehouse, liberando sua equipe de dados para o que eles fazem de melhor. Os armazéns SQL do Databricks fornecem computação SQL instantânea e elástica — desacoplada do armazenamento — e serão dimensionados automaticamente para fornecer

simultaneidade ilimitada sem interrupção, para casos de uso de alta simultaneidade.







- **Serveless Cluster** Você pode criar um serveless cluster (databricks gerencia esse cluster) .O beneficio de utilizar um serveless cluster é que o databricks tem vários cluster que você pode pegar emprestado quando você precisar deles. Com o serverless , os clientes do Databricks podem acessar computação quase instantânea, com gerenciamento mínimo e menor TCO. Esta computação e os seus recursos associados são geridos pela Databricks num plano de computação sem servidor dentro da conta Databricks do cliente. Para proteger os dados dos clientes, as cargas de trabalho sem servidor são executadas em múltiplas camadas de isolamento. Toda a computação é efêmera, dedicada exclusivamente a essa carga de trabalho e apagada com segurança assim que a carga de trabalho for concluída. Essas camadas de isolamento são exaustivamente testadas tanto por nossa equipe interna de segurança ofensiva quanto por empresas externas de testes de penetração, para proteger seus dados em todos os momentos.
- **Cluster na sua conta:** Você pode criar um cluster em sua própria conta.
- **Unit Catalog** para gerenciamento de dados e governança.
  - Implements access control on data.
  - Access control is always enabled.
  - Work across multiple workspaces.
  - Grant permissions to users at account level

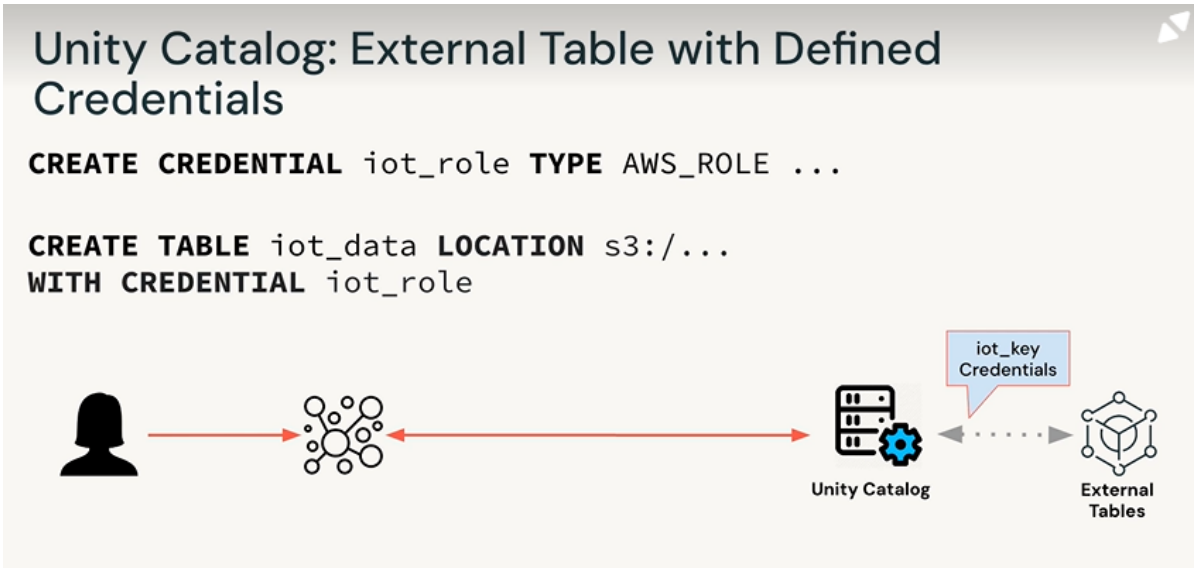
- **Unity Catalog Object Model**

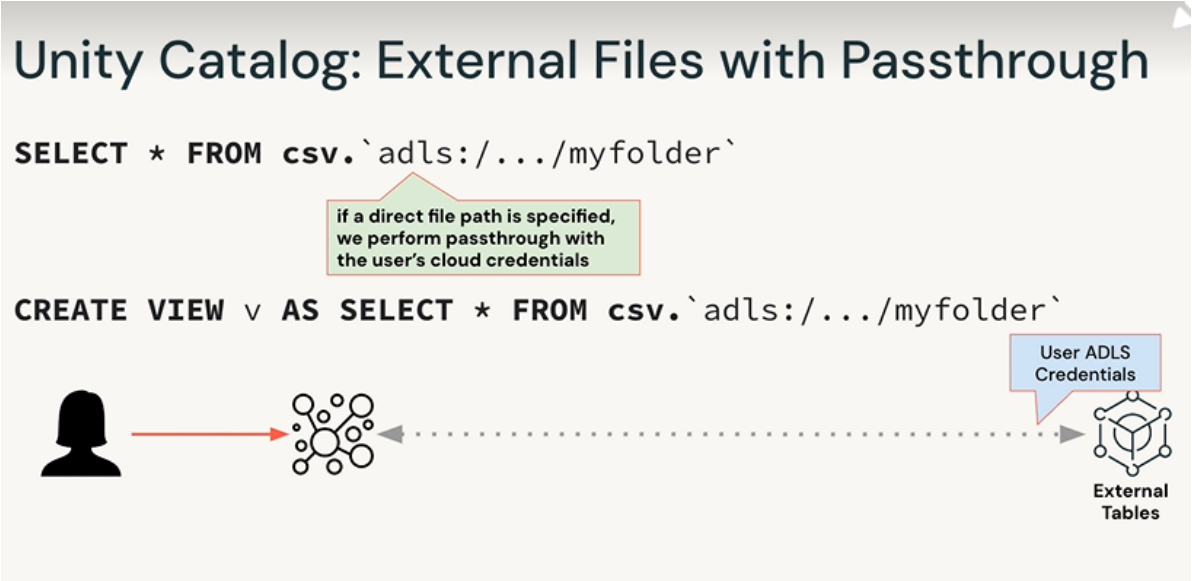
  - Metastore
  - Catalogs
  - Schemas
    - (External tables, Managed tables, Views)

- **Metastore:** store data assets; permissions; created with default storage location (external object store); metastore admin. **armazenar ativos de dados; permissões; criado com local de armazenamento padrão (armazenamento de objetos externo); administrador do metastore.** O Metastore no Databricks é um componente essencial que armazena metadados sobre os dados, incluindo informações sobre a estrutura e localização física dos dados. Ele permite acesso eficiente e integração com outros sistemas de armazenamento de dados, como o Data Lake Storage, facilitando a consulta e análise de dados no ambiente do Databricks.
- **Catalog:** First level of organization. Users can see all catalogs where USAGE is granted. O Catálogo no Databricks é um sistema centralizado para armazenar metadados relacionados aos dados e recursos disponíveis na plataforma. Ele organiza informações sobre tabelas, visualizações, funções e fluxos de trabalho, facilitando a busca e acesso a esses recursos. O Catálogo simplifica o gerenciamento de dados e recursos, promovendo o desenvolvimento, colaboração e análise de dados na plataforma.
- **Schema:** aka, database. Second level of organization. No Databricks, o schema se refere à estrutura dos dados armazenados em tabelas ou conjuntos de dados. Ele define os tipos de dados de cada coluna, as restrições de integridade e outras propriedades relacionadas à organização e interpretação dos dados. O schema é fundamental para garantir a consistência e a interpretação correta dos dados ao realizar operações de consulta, análise e transformação no ambiente do Databricks. * Users can see all schemas where USAGE is granted on both, the schema and the catalog.
- **Managed Table:** Third level of organization. Supported format: Delta. Uma "managed table" (tabela gerenciada) no Databricks é uma tabela que é gerenciada diretamente pelo serviço Databricks. Isso significa que o

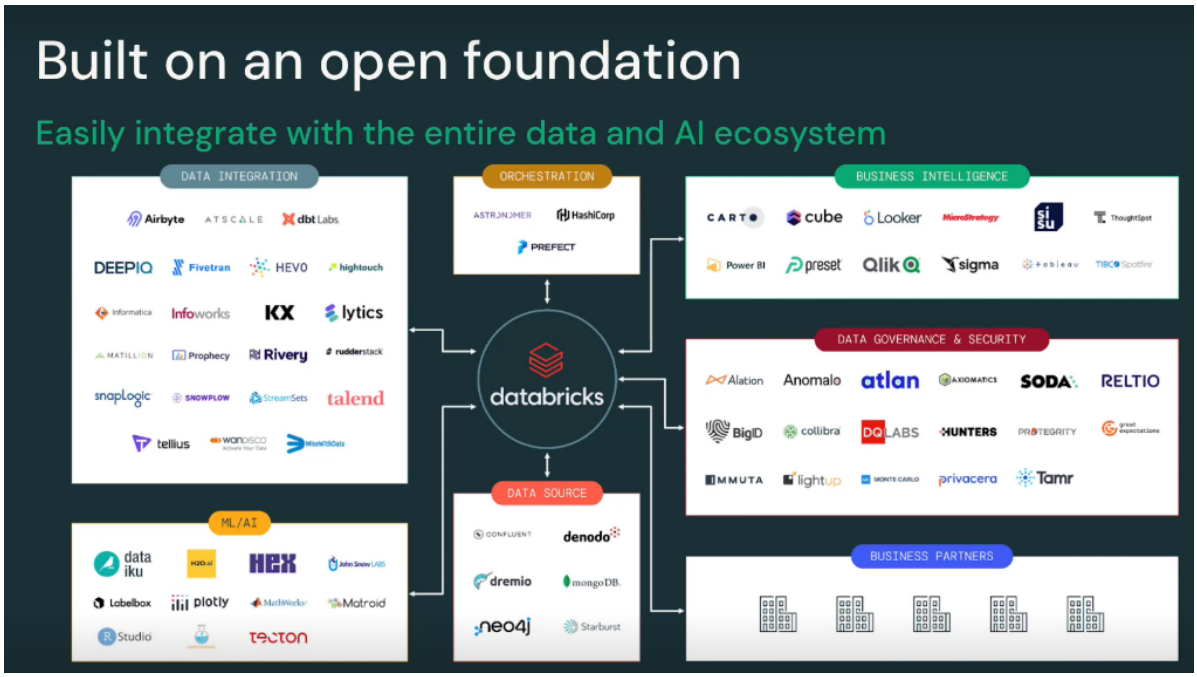
Databricks assume a responsabilidade pelo armazenamento e manutenção dos dados subjacentes à tabela.

- **External Table:** Uma tabela externa no Databricks é uma tabela cujos dados residem em um local externo, como armazenamento de objetos na nuvem ou sistemas de arquivos distribuídos. Ao contrário das tabelas gerenciadas, onde o Databricks controla o armazenamento, em tabelas externas, os dados permanecem no local externo e o Databricks apenas mapeia uma estrutura de tabela sobre eles. Isso permite compartilhar dados entre ambientes e acessar dados existentes, mas não oferece controle direto sobre o armazenamento físico dos dados. **Two credential types: Storage Credential or External Location.**





- **Delta Table:** Is the default data table format in Databricks and is a feature of the Delta Lake open source data framework. Delta tables are typically used for data lakes, where data is ingested via streaming or in large batches.
- **View:** Third level of organization: can be composes from tables and views in multiple schemas or catalogs. creating using “CREATE VIEW”. Uma "view" em bancos de dados é uma representação virtual de uma tabela ou conjunto de tabelas. Ela é como uma consulta SQL armazenada que pode ser tratada como uma tabela, mas não armazena dados próprios. Em vez disso, ela consiste em uma consulta SQL que é executada dinamicamente sempre que a view é acessada.
- **Databricks Integration:**



- **Compute:** A parte de "compute" no Databricks refere-se aos recursos de processamento disponíveis na plataforma para executar análises de dados em grande escala. Isso inclui **clusters** sob demanda, o uso do **Apache Spark** como **motor de processamento** distribuído e runtimes otimizados para machine learning. Esses recursos garantem que as operações de análise, como consultas SQL e treinamento de modelos, sejam executadas de maneira eficiente e escalável.
- **Workflows:** A parte de "workflows" no Databricks refere-se à capacidade da plataforma de organizar e gerenciar fluxos de trabalho de dados de maneira eficiente. Isso inclui a automação de tarefas, agendamento de processos ETL (Extração, Transformação e Carregamento), execução de pipelines de dados e coordenação de atividades de análise de dados. Em resumo, os workflows no Databricks facilitam a criação, execução e monitoramento de fluxos de trabalho de dados de ponta a ponta.
- **Cluster:** Um "cluster" é um conjunto de recursos de computação, como máquinas virtuais ou servidores, que trabalham juntos para executar tarefas de processamento de dados em paralelo. No contexto do Databricks e de ambientes de big data em geral, um cluster é usado para executar operações de análise de dados, como consultas SQL, manipulação de dados e treinamento de modelos de machine learning. Um cluster é uma coleção de

recursos de computação utilizados para executar tarefas de análise de dados em ambientes de big data, oferecendo escalabilidade e paralelismo para lidar com grandes volumes de dados de maneira eficiente. Cluster maiores oferecem um desempenho melhor porém um custo mais elevado, e cluster menores apesar de econômicos oferecem desempenho pior.

- **Single node:** Um "single node" (nó único) refere-se a um ambiente de computação que consiste em apenas uma única máquina física ou virtual. Ao contrário de um ambiente distribuído, onde o processamento de dados é dividido entre várias máquinas em um cluster, um single node executa todas as operações em uma única máquina.
- **Lakehouse:**

  O termo "Lakehouse" no contexto do Databricks se refere a uma abordagem unificada para o gerenciamento e análise de dados que combina características de data lakes e data warehouses. A ideia por trás do conceito de Lakehouse é proporcionar as vantagens de ambos os modelos, combinando a capacidade de armazenamento escalável e flexível dos data lakes com a capacidade de processamento rápido e análise estruturada dos data warehouses.

  No ambiente do Databricks, o Lakehouse é realizado através do uso do Delta Lake, uma tecnologia desenvolvida pela Databricks que permite armazenar dados em um formato altamente eficiente e confiável, proporcionando transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), controle de versão e outras funcionalidades avançadas.

  As principais características do Lakehouse no Databricks incluem:

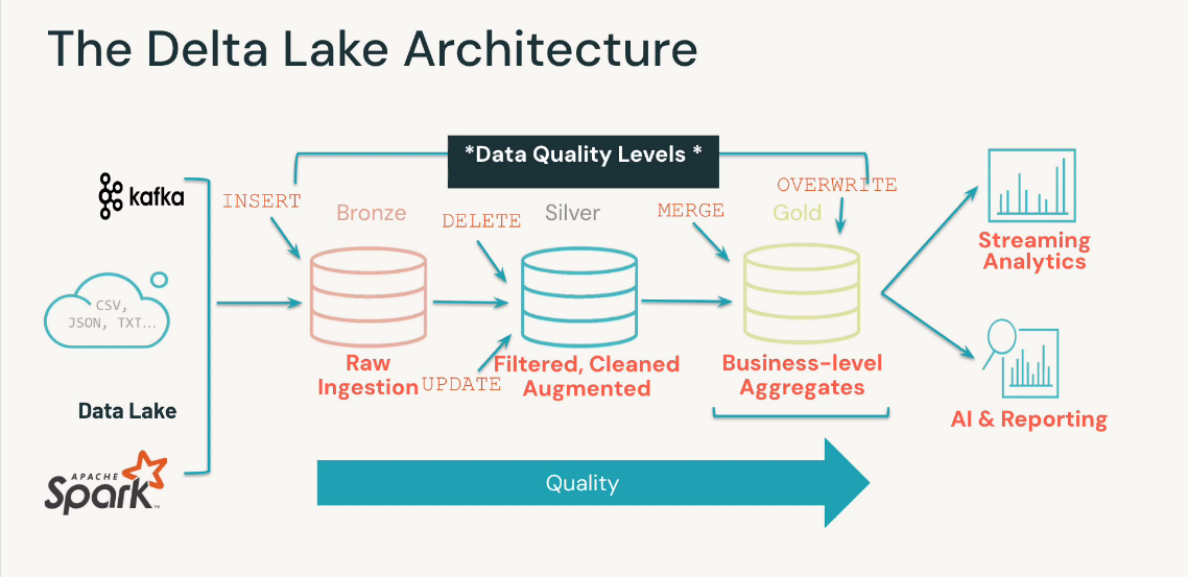
  1. **Armazenamento de Dados em Formato Delta**: O Delta Lake é um formato de armazenamento otimizado para armazenar dados de forma eficiente, fornecendo recursos como transações ACID e controle de versão.
  2. **Processamento Analítico em Tempo Real e Batch**: O Databricks suporta a execução de consultas analíticas em tempo real e em lotes nos dados armazenados no formato Delta Lake, permitindo análises rápidas e flexíveis.
  3. **Integração com Ferramentas de BI e ML**: O Lakehouse no Databricks é compatível com uma variedade de ferramentas de Business Intelligence

(BI) e Machine Learning (ML), facilitando a análise e exploração de dados.

4. **Segurança e Conformidade**: O Databricks oferece recursos avançados de segurança e conformidade para garantir a proteção dos dados armazenados no Lakehouse.

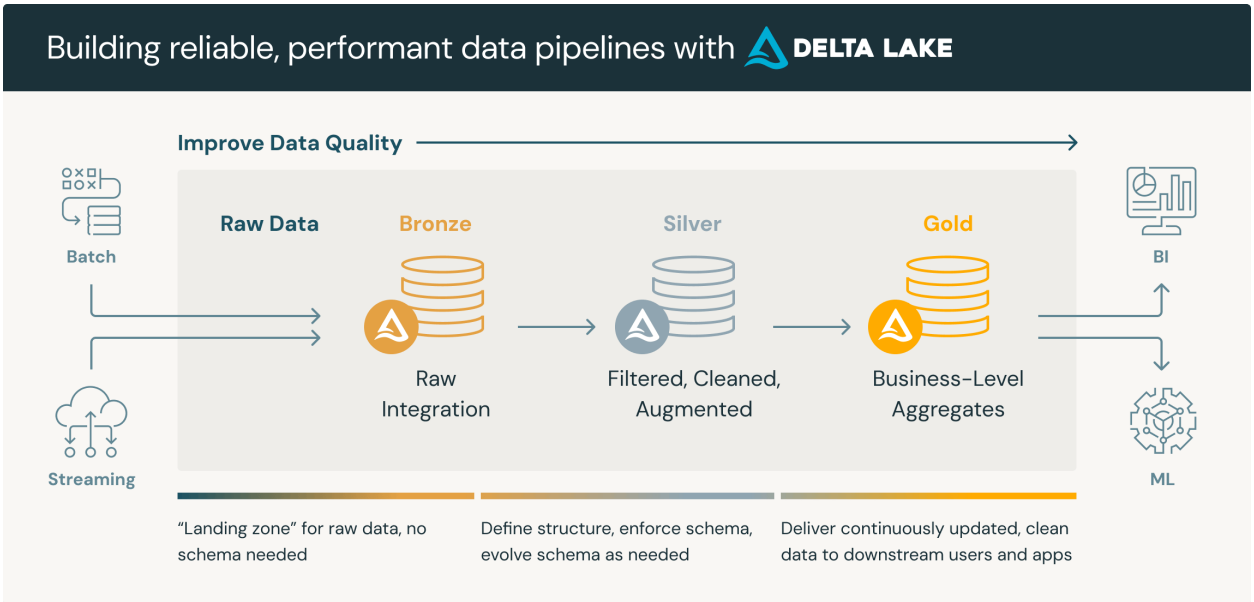
Em resumo, o conceito de Lakehouse no Databricks combina as vantagens dos data lakes e data warehouses em uma abordagem unificada para o gerenciamento e análise de dados, permitindo escalabilidade, flexibilidade e desempenho para uma ampla gama de casos de uso de dados.

- **Delta Lake Achitecture:** A arquitetura Delta Lake é uma estrutura de armazenamento de dados desenvolvida pela Databricks, oferecendo confiabilidade, escalabilidade e desempenho para ambientes de análise de dados em grande escala. Ela garante transações ACID, permite esquema evolutivo, oferece funcionalidades como Time Travel e Compaction, e é amplamente utilizada em ambientes de big data para armazenamento e processamento confiáveis de dados.

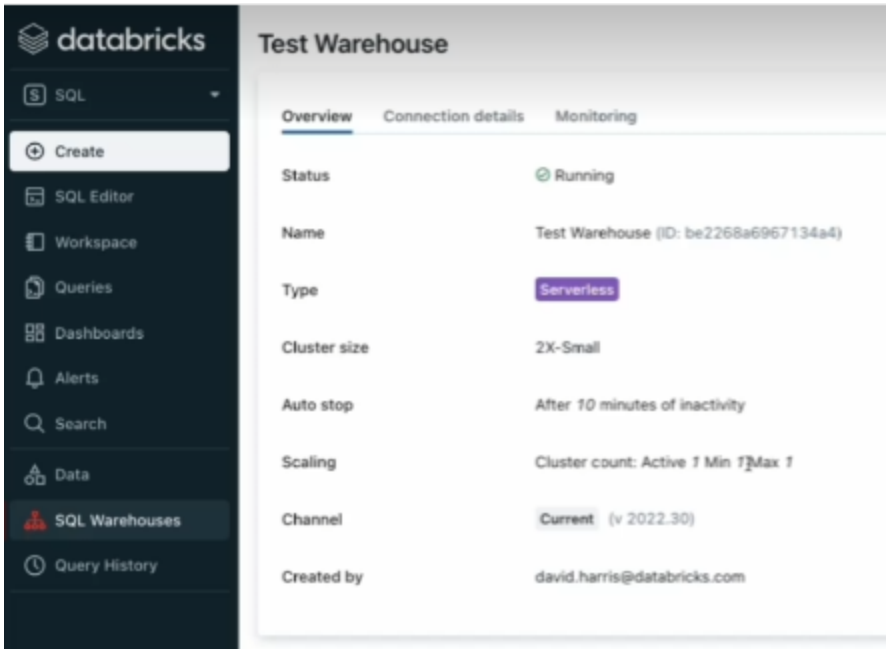


- **Lakehouse Architectures**
  - É um Data Source
  - Sistema em Nuvem
  - Bronze - ingestão dos dados

    - Silver – Limpeza dos dados e faz os joins
    - Gold – dado limpo, para a área de negócio
- **Camadas:**
    - Bronze:
        - Replica os dados do arquivos no formato delta
    - Prata:
        - 1:1 com a bronze, limpeza de estrutura
        - Nova nomenclatura
        - Filtros desejados
        - Também no formato delta
    - Gold:
        - Camada planejada pelo negócio
        - Olha para essa camada para fazer o BI e Data Science
        - Cruza informações e faz relacionamentos entre os a dados
        - Entrega o dado agregado
        - Camada que alimenta os times de negocio
- **Mandellion Architecture:** Is a data design pattern used to logically organize data in a lakehouse, with the goal of incrementally and progressively improving the structure and quality of data as it flows through each layer of the architecture (from Bronze ⇒ Silver ⇒ Gold layer tables). Medallion architectures are sometimes also referred to as "multi-hop" architectures.



- **SQL Warehouse:** é uma instância de banco de dados SQL que permite aos usuários executar consultas SQL em dados armazenados no ambiente da plataforma, oferecendo escalabilidade, desempenho e recursos avançados de segurança e conformidade.



- **As consultas SQL** do Databricks fornecem um ambiente versátil dentro da plataforma Databricks para escrever, testar e executar código SQL para a analise de dados e procesamento
- **ALTER TABLE RENAME TO** Para renomear uma tabela.
- **INSERT INTO** normalmente é usado para adicionar novos registros a uma tabela e não atualiza registros existentes.
- **MERGE INTO** é adequado tanto para atualizar registros existentes quanto para inserir novos registros, dependendo se uma correspondência for encontrada na tabela de destino.
- **COPY INTO** é um comando especializado usado no Databricks para carregar dados em uma tabela de fontes externas, como arquivos em um sistema de arquivos. COPY INTO é ideal para importar dados incrementais em massa no Databricks. Para fazer um COPY INTO com cabeçalho no final do código tem que ter o header = true

  COPY INTO allows SQL users to idempotently and incrementally ingest data from cloud object storage into Delta tables. It can be used in Databricks SQL, notebooks, and Databricks Jobs.
- **ROLLUP** é usado para agregação hierárquica de dados. Começa com o nível mais detalhado e avança para níveis mais amplos, terminando com um total geral (começa na coluna mais a esquerda do GROUP BY)
- **CUBE** produz todos os subtotais possíveis e o ROLLUP somente da subtottais de hierarquia
- **ANALYZE TABLE** é usado para coletar estatísticas importantes sobre uma tabela. Seu foco principal é a coleta de métricas de dados para fins de otimização interna
- **DESCRIBE EXTENDED:** You can view a metadata table using DESCRIBE EXTENDED. You can see the column's name, data type, comment and table detailed information.

  ```
  DESCRIBE EXTENDED temporary_schema.simple_table
  ```

- **DESCRIBE HISTORY** é utilizado para recuperar informações sobre cada operação de gravação em uma tabela Delta, mostrando a versão, carimbo de data/hora, usuário e outros detalhes da operação.

```
DESCRIBE HISTORY simple_table;
```

- **DROP TABLE IF EXISTS** deleta a tabela caso ela exista.

```
DROP TABLE IF EXISTS temporary_schema.simple_table;
```

- **CREATE OR REPLACE VIEW** para criar uma view.
- **UPDATE**

  ```
  UPDATE customers SET loyalty_segment = 10 WHERE loyalty_segme
  UPDATE customers SET loyalty_segment = 0 WHERE loyalty_segmer
  DESCRIBE HISTORY customers;
  ```

  O comando acima faz dois updates para fazer mudanças na tabela.

  Rodando o comando DESCRIBE HISTORY podemos ver os updates no log e seus timestamps.

- **VERSION AS OF**
  We can select a specific delta table's version. This feature of Delta tables is called "Time travel" and its very powerful.

  ```
  SELECT loyalty_segment FROM customers VERSION AS OF 1;
  ```

  We can also use TIMESTAMP AS OF to SELECT based on a table's state on a specific date and time, and you can find more information on this in the documentation.

- **RESTORE**

  If we wish to restore a table to a previous version or timestamp, we can use the RESTORE command.

```
-- Restore the employee table to a specific timestamp
> RESTORE TABLE employee TO TIMESTAMP AS OF '2022-08-02 00:0
0:00';

-- Restore the employee table to a specific version number re
trieved from DESCRIBE HISTORY employee
RESTORE TABLE employee TO VERSION AS OF 1;
```

- **EXPLODE()**

```
SELECT transaction_id, explode(products) AS product FROM tran
sactions;
```

explode(products)

This function will transform each element of the array into a separate row, duplicating the values of other columns as necessary.

- **MEDIA X VARIANCIA**
- A média é uma medida de tendência central que indica o valor médio de um conjunto de dados, enquanto a variância é uma medida de dispersão que indica o quão distantes os valores individuais estão da média.
- **Data security:**
  - Describe the different levels of data object access available with Unity Catalog.
  - Identify that catalogs, schemas and tables can all have unique owners.
  - Describe how to organize owned data objects for the purposes of security.
  - Identify that the creator of a data object becomes the owner of that data object.
  - Identify the responsabilities of data ownership.
  - Update data object permission to address user access needs in a variety of common scenarios.

    - Identify PII (Personal Identification) data objects as needing additional, organization-specific considerations.

- **PII data: Personally Identifiable Information** Em resumo, PII data se refere a informações pessoais que podem ser usadas para identificar individualmente uma pessoa e são consideradas sensíveis, sujeitas a regulamentos de privacidade de dados (LGPD). O PII no Databrciks geralmente é tratado através do Delta Lake para dar controles de acessos finos.
- **SQL Automation:** Using Databricks SQL, you can automate many tasks that make working in the Lakehouse much easier. The automations that are available to you in Databricks SQL are:
    - Query refresh schedules
    - Dashboard refresh schedules
    - Alerts

    These automations are configured within Databricks SQL and are independent of any automations in the rest of the Lakehouse, meaning, they can affect data anywhere in the Lakehouse, but they do not use Workflows or Jobs. Furthermore, they use SQL Warehouses, as opposed to clusters.
- **Query Refresh Schedule:** You can use scheduled query executions to keep your dashboards updated or to enable routine alerts. Let's make a query and put it on a refresh schedule.
- **Alerts:** similar to queries and dashboards, can be organized within a folder structure to help you keep track of their location. Alerts allow you to configure notifications when a field returned by a scheduled query meets a specific threshold. Databricks SQL alerta execução query periodicamente, avalia condições definidas e envia notificações se uma condição for atendida. Você pode configurar o alerta para monitorar seu negócio e enviar notificações quando os dados relatados estiverem fora dos limites esperados. programar um alerta executa sua query subjacente e verifica os critérios de alerta. Isto é independente de qualquer programar que possa existir na query subjacente.
- **Sharing queries:** We can share queries with other members of the team. You can share the query with users and groups who are configured in your workspace. These users and groups can have “Can Manage”, “Can Edit,” “Can

Run," or "Can View" permissions. Those with "Can Edit" permissions can also run the query. In order to allow for "Can Edit" permissions, the "Credentials" dropdown must be changed to "Run as Viewer."

Note that any "Can edit" permissions that were granted must be revoked before the credential type for the query can be changed back to "Run as owner".

- **Sharing Dashboards:** Sharing a dashboard is accomplished in the same fashion as sharing a query. Click 'Share' from the upper right corner of any dashboard to open the Sharing dialogue.
- **Refreshing Dashboards and Sharing Results:** Adding a refresh schedule to a dashboard is similar to adding a refresh schedule to a query. However, when you add a refresh schedule to a dashboard, you have the option of including subscribers which will notify users when the dashboard is updated.
- **Partner Connect:** permite que você crie contas de teste com parceiros de tecnologia selecionados da Databricks e conecte seu workspace do Databricks às soluções dos parceiros diretamente pela interface da Databricks. Isso permite que você experimente soluções de parceiros usando seus dados no Databricks Lakehouse e adote as soluções que melhor atendam às suas necessidades comerciais.

  O Partner Connect fornece uma alternativa mais simples às conexões manuais de parceiros, provisionando os recursos necessários do Databricks em seu nome e passando as informações dos recursos para o parceiro. Os recursos necessários podem incluir um Databricks SQL warehouse (antigo endpoint do Databricks SQL), um principal de serviço e um token de acesso pessoal.

Databricks Partner Connect

Databricks Partner Connect is a **dedicated ecosystem of integrations** that allows users to **easily connect** with popular **data ingestion, transformation and BI partner products.**

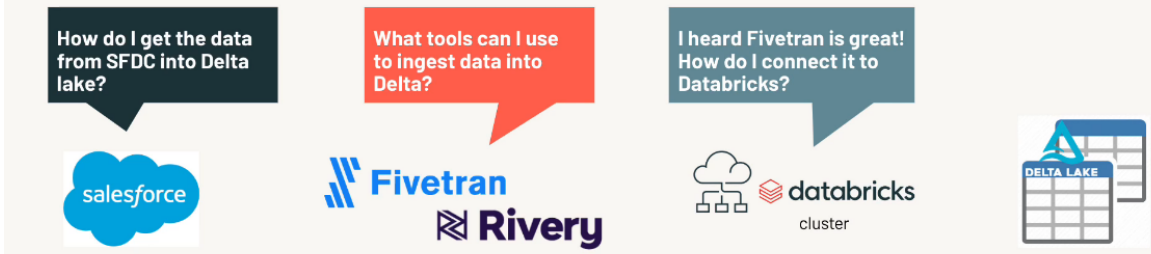
This helps data analysts get useful data into their lakehouse **faster** without the need to manually configure each product so they can get data-driven insights

Databricks Partner Connect

Helps Data Analysts who:

- Struggle to connect to their choice of BI tools
- Struggle to bring data from SaaS apps (Google Analytics, Facebook, etc.) to run SQL queries
- Have to wait on eng/ops to generate data for analysis

Partner Connect Makes it Easy



DATABRICKS PARTNER CONNECT

- Many partner integrations take as few as **6 clicks**
- **No** context or page **switches** required
- **Automatically** launches a cluster, calls Partner API to pass on PAT token and the cluster configuration details
- Sets up all the **necessary configs** for an optimized user experience
- Creates **trial account** in the partner product if an account doesn't exist