IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
SHERMAN DIVISION

R2 Solutions LLC, Plaintiff, v. Databricks, Inc., Defendant.	Civil Action No. 4:23-cv-01147 Jury Trial Demanded

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff R2 Solutions LLC files this Complaint against Databricks, Inc. for infringement of U.S. Patent No. 8,190,610 ("the '610 patent"). The '610 patent is sometimes referred to as the "patent-in-suit."

THE PARTIES

1. Plaintiff R2 Solutions LLC ("R2") is a Texas limited liability company located in Frisco, Texas.

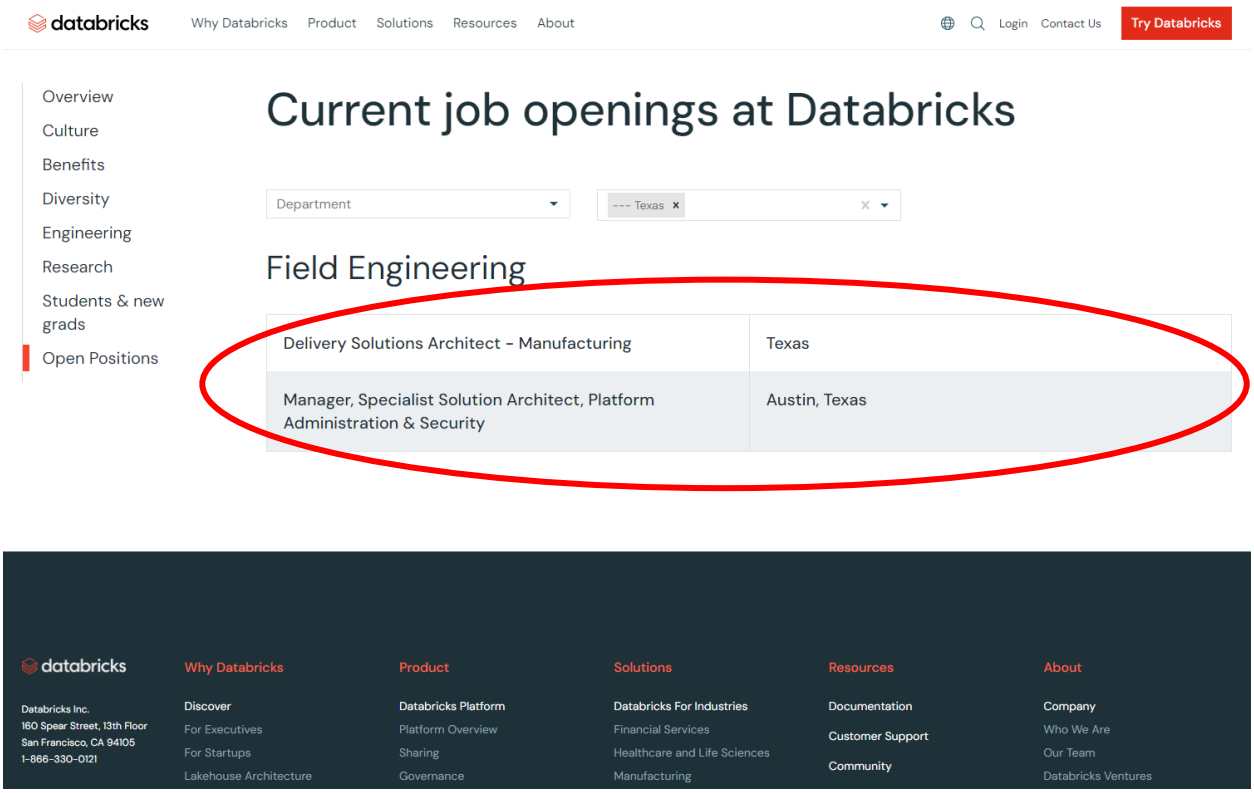
2. Defendant Databricks, Inc. ("Databricks") is a Delaware corporation headquartered at 160 Spear St., Suite 1300, San Francisco, CA 94105 and has a regular and established place of business in this District at 6900 Dallas Pkwy, Suite 02-106, Plano, TX 75024. Databricks may be served with process through its registered agent, United Agent Group Inc., at 5444 Westheimer, #1000, Houston, TX 77056.

JURISDICTION AND VENUE

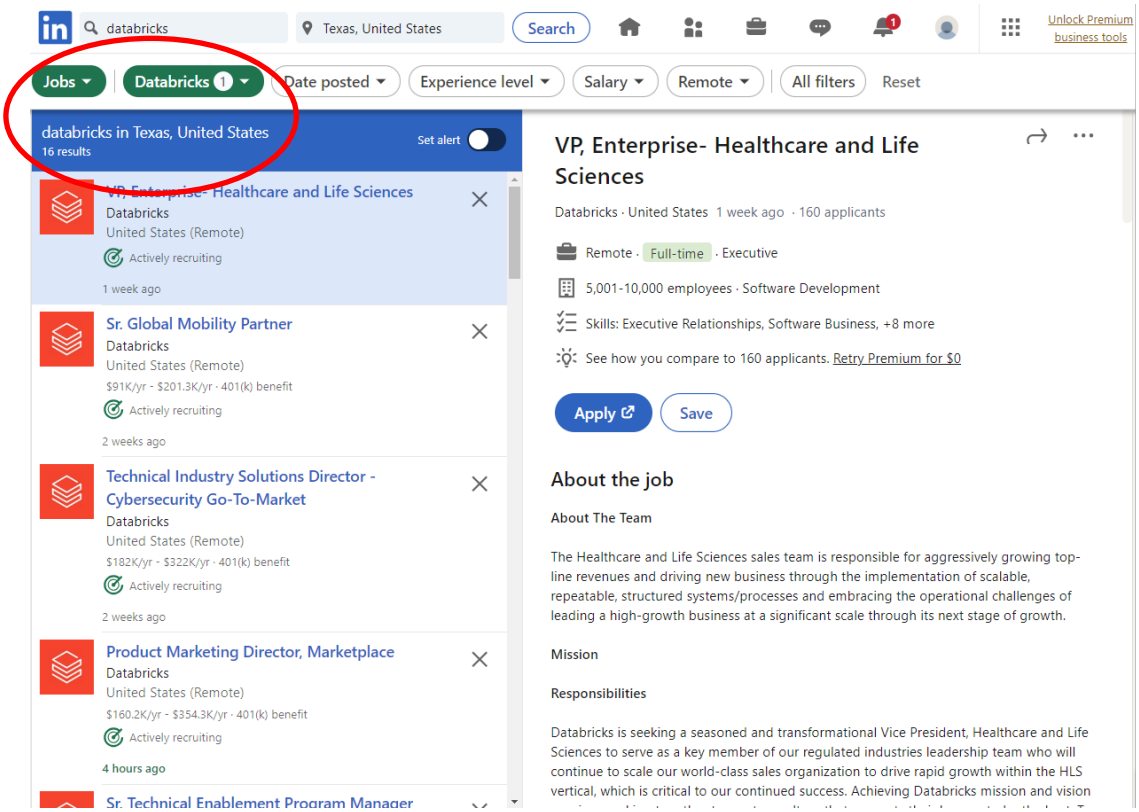
3. This action arises under the patent laws of the United States, 35 U.S.C. § 101, *et seq.* This Court's jurisdiction over this action is proper under the above statutes, including 35

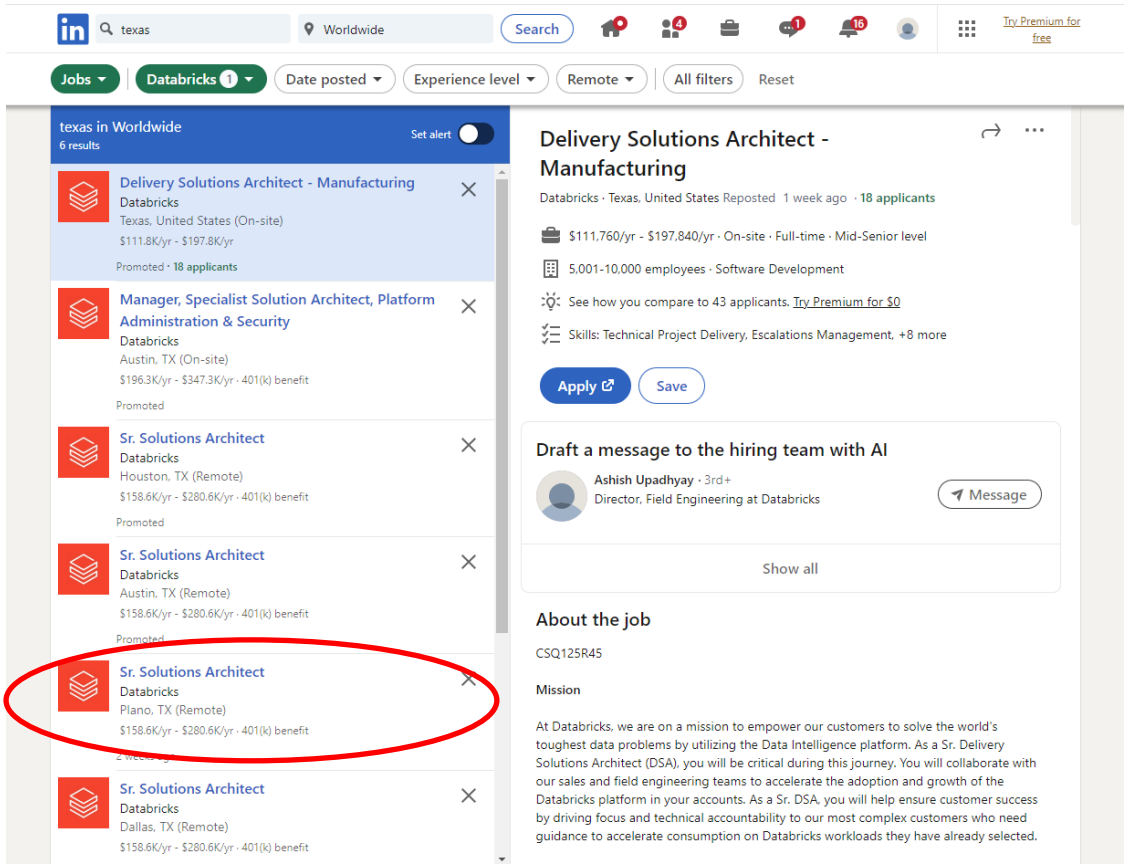
U.S.C. § 271, *et seq*., 28 U.S.C. § 1331 (federal question jurisdiction), and 28 U.S.C. § 1338 (jurisdiction over patent actions).

4. This Court has personal jurisdiction over Databricks because, among other things, Databricks does business in this State by, among other things, "recruit[ing] Texas residents, directly or through an intermediary located in this State, for employment inside or outside this State." Tex. Civ. Prac. & Rem. Code § 17.042(3). For instance, Databricks has multiple job openings in Texas as of December 18, 2023:[1]

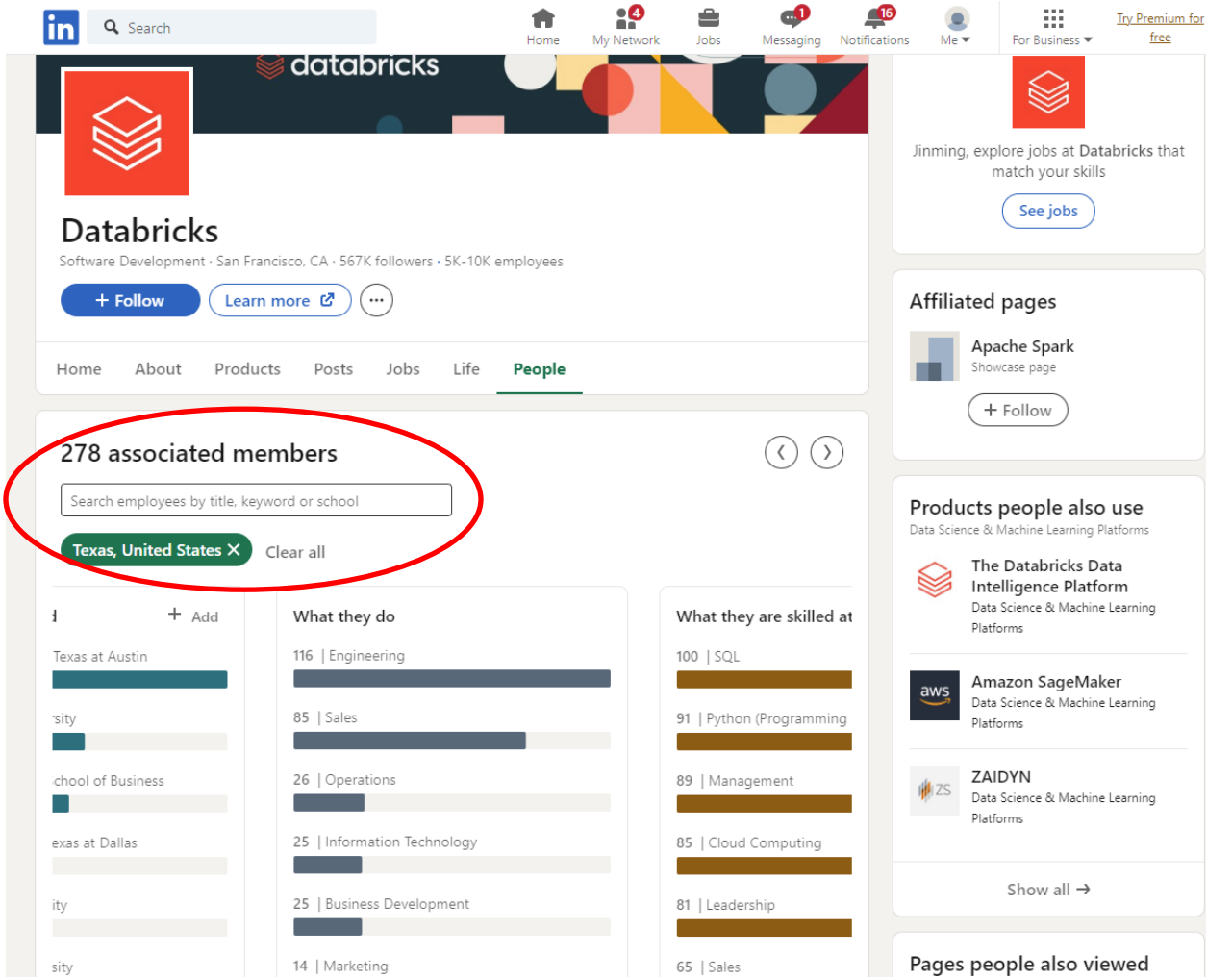


[1] https://www.databricks.com/company/careers/open-positions?department=all&location=Texas; https://www.linkedin.com/jobs/search/?currentJobId=3782993305&f_C=3477522&geoId=102748797&keywords=databricks&location=Texas%2C%20United%20States&origin=JOB_SEARCH_PAGE_SEARCH_BUTTON&refresh=true; https://www.linkedin.com/jobs/search/?currentJobId=3765311929&f_C=3477522&geoId=92000000&keywords=texas&origin=COMPANY_PAGE_JOBS_KEYWORD.


databricks
Texas, United States
Search
Unlock Premium business tools
Jobs
Databricks 1
Date posted
Experience level
Salary
Remote
All filters
Reset
databricks in Texas, United States
16 results
Set alert
VP, Enterprise- Healthcare and Life Sciences
Databricks
United States (Remote)
Actively recruiting
1 week ago
Sr. Global Mobility Partner
Databricks
United States (Remote)
$91K/yr - $201.3K/yr · 401(k) benefit
Actively recruiting
2 weeks ago
Technical Industry Solutions Director - Cybersecurity Go-To-Market
Databricks
United States (Remote)
$182K/yr - $322K/yr · 401(k) benefit
Actively recruiting
2 weeks ago
Product Marketing Director, Marketplace
Databricks
United States (Remote)
$160.2K/yr - $354.3K/yr · 401(k) benefit
Actively recruiting
4 hours ago
Sr. Technical Enablement Program Manager
VP, Enterprise- Healthcare and Life Sciences
Databricks · United States 1 week ago · 160 applicants
Remote · Full-time · Executive
5,001-10,000 employees · Software Development
Skills: Executive Relationships, Software Business, +8 more
See how you compare to 160 applicants. Retry Premium for $0
Apply
Save
About the job
About The Team
The Healthcare and Life Sciences sales team is responsible for aggressively growing top-line revenues and driving new business through the implementation of scalable, repeatable, structured systems/processes and embracing the operational challenges of leading a high-growth business at a significant scale through its next stage of growth.
Mission
Responsibilities
Databricks is seeking a seasoned and transformational Vice President, Healthcare and Life Sciences to serve as a key member of our regulated industries leadership team who will continue to scale our world-class sales organization to drive rapid growth within the HLS vertical, which is critical to our continued success. Achieving Databricks mission and vision


texas
Worldwide
Search
Try Premium for free
Jobs
Databricks 1
Date posted
Experience level
Remote
All filters
Reset
texas in Worldwide
6 results
Set alert
Delivery Solutions Architect - Manufacturing
Databricks
Texas, United States (On-site)
$111.8K/yr - $197.8K/yr
Promoted · 18 applicants
Manager, Specialist Solution Architect, Platform Administration & Security
Databricks
Austin, TX (On-site)
$196.3K/yr - $347.3K/yr · 401(k) benefit
Promoted
Sr. Solutions Architect
Databricks
Houston, TX (Remote)
$158.6K/yr - $280.6K/yr · 401(k) benefit
Promoted
Sr. Solutions Architect
Databricks
Austin, TX (Remote)
$158.6K/yr - $280.6K/yr · 401(k) benefit
Promoted
Sr. Solutions Architect
Databricks
Plano, TX (Remote)
$158.6K/yr - $280.6K/yr · 401(k) benefit
Sr. Solutions Architect
Databricks
Dallas, TX (Remote)
$158.6K/yr - $280.6K/yr · 401(k) benefit
Delivery Solutions Architect - Manufacturing
Databricks · Texas, United States Reposted 1 week ago · 18 applicants
$111,760/yr - $197,840/yr · On-site · Full-time · Mid-Senior level
5,001-10,000 employees · Software Development
See how you compare to 43 applicants. Try Premium for $0
Skills: Technical Project Delivery, Escalations Management, +8 more
Apply
Save
Draft a message to the hiring team with AI
Ashish Upadhyay · 3rd+
Director, Field Engineering at Databricks
Message
Show all
About the job
CSQ125R45
Mission
At Databricks, we are on a mission to empower our customers to solve the world's toughest data problems by utilizing the Data Intelligence platform. As a Sr. Delivery Solutions Architect (DSA), you will be critical during this journey. You will collaborate with our sales and field engineering teams to accelerate the adoption and growth of the Databricks platform in your accounts. As a Sr. DSA, you will help ensure customer success by driving focus and technical accountability to our most complex customers who need guidance to accelerate consumption on Databricks workloads they have already selected.

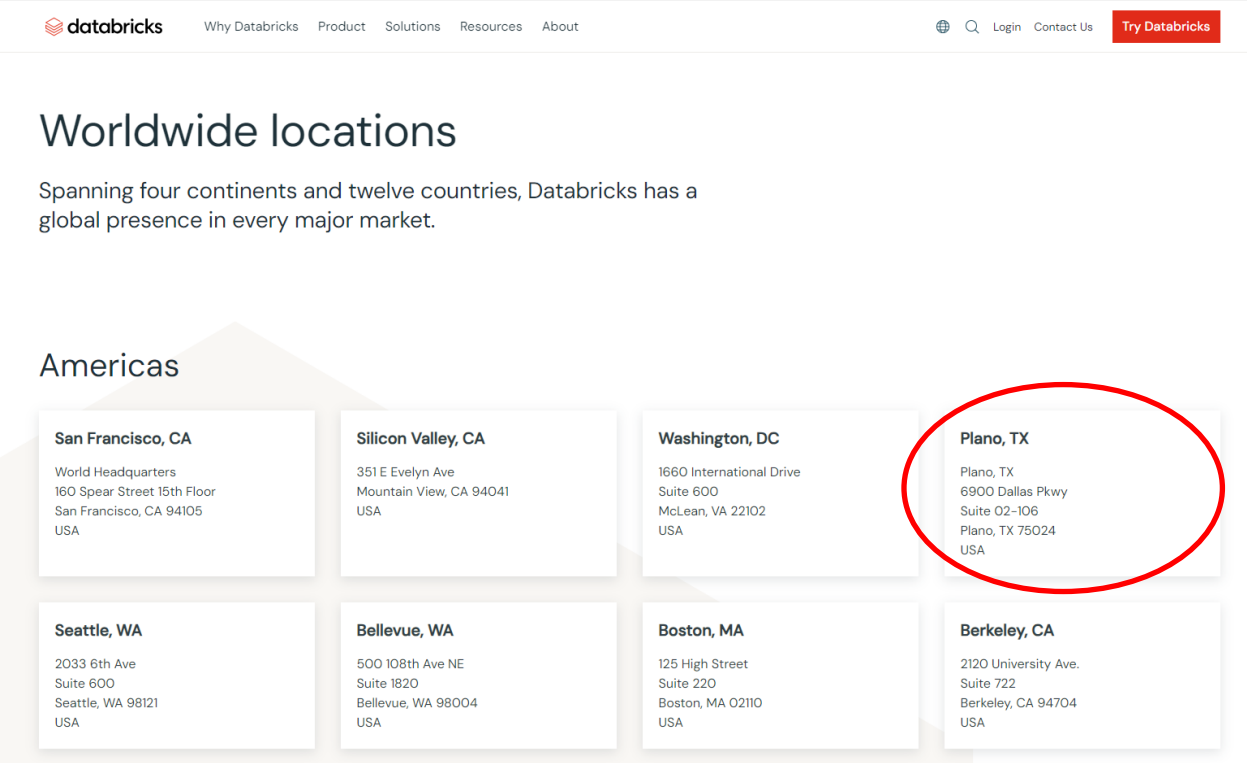
5. And according to its LinkedIn page, Databricks has 278 employees in its Texas office (as of December 18, 2023):[2]



6. Further, this Court has personal jurisdiction over Databricks because it has engaged, and continues to engage, in continuous, systematic, and substantial activities within this State, including the substantial marketing and sale of products and services within this State and this District. Indeed, this Court has personal jurisdiction over Databricks because it has committed acts giving rise to R2's claims for patent infringement within and directed to this

[2] https://www.linkedin.com/company/databricks/people/?facetGeoRegion=102748797

District, has derived substantial revenue from its goods and services provided to individuals and entities in this State and this District, and maintains regular and established places of business in this District, including at least its brick-and-mortar location in Plano, Texas.[3]



7. Relative to patent infringement, Databricks has committed and continues to commit acts in violation of 35 U.S.C. § 271, and has made, used, offered for sale, and/or sold infringing products, systems, and/or services in this State, including this District, and has otherwise engaged in infringing conduct within and directed at, or from, this District. Such infringing products, systems, and/or services (collectively, the "Accused Instrumentalities") include the Databricks Data Intelligence Platform/Databricks Lakehouse Platform, and any other platform(s) offered or provided by Databricks that utilize Apache Spark or any other similar functionality.

[3] https://www.databricks.com/company/contact/office-locations.

8. Databricks' infringing activities have caused harm to R2 in this District. Databricks and/or its partners offer to sell and sell the Accused Instrumentalities within this District, and on information and belief, Databricks, its partners, and/or their customers make the Accused Instrumentalities in this District and use the Accused Instrumentalities in this District in an infringing manner. For example, Databricks, its partners, and/or their customers (induced by Databricks) implement and exert control over the Accused Instrumentalities via cloud-based and on-premises solutions that utilize computers and/or servers located in this District. Outputs from such methods and systems are generated by and/or delivered to devices implementing the Accused Instrumentalities in this District. Databricks and/or its partners provide the Accused Instrumentalities (and services therewith) to customers in this District, and Databricks' customers in this District obtain data analytics facilitated by the Accused Instrumentalities, whether via Databricks' implementation of the Accused Instrumentalities on their behalf, or via their use of the Accused Instrumentalities provided to them by Databricks. These are purposeful acts and transactions in this State and this District such that Databricks reasonably should know and expect that it could be haled into this Court.

9. Venue is proper in this District under 28 U.S.C. §§ 1391 and 1400(b) because Databricks has a regular and established place of business in Plano, which is in this District. Venue is further proper in this District because Databricks has directly infringed and/or induced the infringement of others, including its customers, in this District. As set out above, Databricks has at least offered for sale and sold the Accused Instrumentalities in this District and has used the Accused Instrumentalities in an infringing manner in this District. In addition, Databricks' customers have made and continue to make the Accused Instrumentalities in this District, and have used and continue to use the Accused Instrumentalities in an infringing manner in this

District. These infringements were, and continue to be, induced by Databricks (as set out further below).

BACKGROUND

10. The patent-in-suit was filed by Yahoo! Inc. ("Yahoo!") in 2006. At the time, Yahoo! was a leading Internet communications, commerce, and media company. Yahoo! invested billions of dollars in research and development over this period, filing hundreds of patent applications each year to cover the innovative computing technologies emerging from its expansive research and development efforts.

11. Yahoo! began as a directory of websites that two Stanford graduate students developed as a hobby. The name "Yahoo" stands for "Yet Another Hierarchical Officious Oracle," a nod to how the original Yahoo! database was arranged hierarchically in layers of subcategories. From this initial database, Yahoo! would develop and promulgate numerous advancements in the field of data storage and recall.

12. For example, in 1995, Yahoo! introduced Yahoo! Search. This software allowed users to search the Yahoo! directory, making it the first popular online directory search engine. This positioned Yahoo! as the launching point for most users of the World Wide Web. By 1998, Yahoo! had the largest audience of any website or online service. In the early 2000s, Yahoo! continued to develop its suite of technologies in the web search and database industries. The patent-in-suit relates to innovations during this period associated with data analytics.

THE PATENT-IN-SUIT

13. The '610 patent is entitled, "MapReduce for Distributed Database Processing." The '610 patent lawfully issued on May 29, 2012, and stems from U.S. Patent Application No.

11/539,090, which was filed on October 5, 2006. A copy of the '610 patent is attached hereto as Ex. 1.

14. R2 Solutions is the owner of the patent-in-suit with all substantial rights, including the exclusive right to enforce, sue, and recover damages for past and future infringements.

15. The claims of the patent-in-suit are directed to patent-eligible subject matter under 35 U.S.C. § 101. They are not directed to abstract ideas, and the technologies covered by the claims consist of ordered combinations of features and functions that, at the time of invention, were not, alone or in combination, well-understood, routine, or conventional.

16. Indeed, the specification of the patent-in-suit discloses shortcomings in the prior art and then explains, in detail, the technical way the claimed inventions resolve or overcome those shortcomings. For example, the specification explains that "conventional MapReduce implementations do not have facility to efficiently process data from heterogeneous sources" and that "it is impractical to perform joins over two relational tables that have different schemas." '610 patent at 3:9-20. To solve these problems, the '610 patent provides a clear technological improvement to existing MapReduce systems by describing and implementing a novel MapReduce architecture where mapping and reducing functions can be applied to data from heterogeneous data sources (i.e., data sources having different schema) to accomplish the merger of heterogeneous data based on a key in common between or among the heterogeneous data. For example, the '610 patent explains how implementation of, e.g., "data groups" realizes these improvements:

> In general, partitioning the data sets into data groups enables a mechanism to associate (group) identifiers with data sets, map functions and iterators (useable within reduce functions to access intermediate data) and, also, to produce output

> data sets with (group) identifiers. It is noted that the output group identifiers may differ from the input/intermediate group identifiers.

'610 patent at 3:58-64.

17. The technological advantages of a "data group"-centric system are shown to "enhance[] the utility of the MapReduce programming methodology." '610 patent at 1:32-33. As the specification explains:

> [T]he MapReduce concept may be utilized to carry out map processing independently on two or more related datasets (e.g., related by being characterized by a common key) even when the related data sets are heterogeneous with respect to each other, such as data tables organized according to different schema. The intermediate results of the map processing (key/value pairs) for a particular key can be processed together in a single reduce function by applying a different iterator to intermediate values for each group. In this way, operations on the two or more related datasets may be carried out more efficiently or in a way not even possible with the conventional MapReduce architecture.

Id. at 8:47-58.

18. Such a solution is embodied, for example, in Claim 1 of the '610 patent:

> A method of processing data of a data set over a distributed system, wherein the data set comprises a ***plurality of data groups***, the method comprising:
>
> partitioning the data of each one of the data groups into a plurality of data partitions that each have a plurality of key-value pairs and ***providing each data partition to a selected one of a plurality of mapping functions*** that are each user-configurable to independently output a plurality of lists of values for each of a set of keys found in such map function's corresponding data partition to form corresponding ***intermediate data for that data group and identifiable to that data group***, wherein ***the data of a first data group has a different schema than the data of a second data group*** and ***the data of the first data group is mapped differently than the data of the second data group*** so that different lists of values are output for the corresponding different

intermediate data, ***wherein the different schema and corresponding different intermediate data have a key in common***; and

reducing the intermediate data for the data groups to at least one output data group, including ***processing the intermediate data for each data group in a manner that is defined to correspond to that data group***, so as to result in a ***merging of the corresponding different intermediate data based on the key in common***,

wherein the mapping and reducing operations are performed by a distributed system.

(emphasis added).

19. The concept of "data groups" as found in Claim 1 of the '610 patent in the context of MapReduce attains a novel and technological improvement in computer capabilities. For example, employing "data groups" allows a diverse data set to be fed to collections of mapping and reducing functions within the same MapReduce architecture to ultimately be joined and/or merged in spite of the diversity. Per Claim 1, the improved MapReduce architecture in the reducing phase is able to selectively employ specialized processing based on the "data group" from which the data being reduced originated, and this specialized processing enables the MapReduce architecture in the reducing phase to accomplish the merger of intermediate data hailing from different data groups.

20. The inventions described and claimed in the '610 patent improve the speed, efficiency, effectiveness, and functionality of computer systems. Moreover, the inventions provide an improvement in computer functionality rather than improvement in performance of an economic task or other tasks for which a computer is used merely as a tool. The '610 patent itself states that the claimed inventions "enhance[] the utility of the MapReduce programming methodology." '610 patent at Abstract, 1:31-33, 1:66-2:2. The '610 patent specification goes on to explain that "[t]he intermediate results of the map processing (key/value pairs) for a particular

key can be processed together in a single reduce function by applying a different iterator to intermediate values for each group." *Id.* at Abstract, 1:37-39, 2:4-8. And the specification discusses the use of multiple processors to perform processing functions in parallel. *See id.* As a result, computer functionality is improved. *Id.* at 1:42-44.

21. Additionally, the claimed inventions provide for more dynamic, customizable, and efficient processing of large sets of data. S*ee, e.g.*, '610 patent at 2:58-61, 4:18-22. The inventions provide optimization of such processing, which increases efficiency and reduces processor execution time. For example, the specification describes a combiner function that "helps reduce the network traffic and speed up the total execution time." '610 patent at 3:1-8. The specification also discusses the use of configurable settings to reduce processing overhead. *See, e.g.*, *id.* at 4:60-62, 5:33-39.

22. In essence, the patent-in-suit relates to novel and non-obvious inventions in the fields of data analytics and database structures.

DEFENDANT'S PRE-SUIT KNOWLEDGE OF ITS INFRINGEMENT

23. Prior to the filing of this Complaint, Databricks was notified on numerous occasions of the '610 patent and the R2 portfolio to which the '610 patent belongs.

24. On April 28, 2022, R2 filed suit against American Airlines, Inc., styled *R2 Solutions LLC v. American Airlines, Inc.*, Case No. 4:22-cv-00353 (E.D. Tex. Apr. 28, 2022) (the "AA litigation"), alleging infringement of the '610 patent.

25. On January 10, 2023, R2 served Databricks with a subpoena in connection with the AA litigation. The subpoena specifically identified the '610 patent and sought materials and testimony regarding Databricks' systems and products that are now accused in this lawsuit.

26. On information and belief, Databricks has had knowledge of the ’610 patent and its infringements since shortly after April 28, 2022, when R2 filed the AA litigation. At the very least, Databricks has had knowledge of the ’610 patent since being served with a subpoena in connection with the AA litigation on January 10, 2023.

COUNT I
INFRINGEMENT OF U.S. PATENT NO. 8,190,610

27. This cause of action arises under the patent laws of the United States, and in particular, 35 U.S.C. §§ 271, *et seq*.

28. R2 Solutions is the owner of the ’610 patent with all substantial rights to the ’610 patent, including the exclusive right to enforce, sue, and recover damages for past and future infringements.

29. The ’610 patent is valid and enforceable and was duly issued in full compliance with Title 35 of the United States Code.

Direct Infringement (35 U.S.C. § 271(a))

30. Databricks has directly infringed, and continues to directly infringe, one or more claims of the ’610 patent in this District and elsewhere in Texas and the United States.

31. To this end, Databricks has infringed and continues to infringe, either by itself or via an agent, at least claims 1-32 of the ’610 patent by, among other things, making, offering to sell, selling, and/or using the Accused Instrumentalities.

32. For example, Databricks uses the Accused Instrumentalities in an infringing manner as detailed in Exhibit 2. Databricks both uses the Accused Instrumentalities for itself and implements the Accused Instrumentalities to provide analytics services to its customers. Databricks offers these services on a per-“Databricks Unit” (“DBU”) basis, and a “DBU” “is a normalized unit of processing power on the Databricks Lakehouse Platform used for

measurement and pricing purposes. The number of DBUs a workload consumes is driven by processing metrics, which may include the compute resources used and the amount of data processed."[4]

33. In addition, on information and belief, Databricks makes and uses the Accused Instrumentalities for itself and for its customers. Databricks also offers to sell, and sells, the Accused Instrumentalities to its customers for implementation directly by the customers. Such making, offering to sell, and selling directly infringes the '610 patent as detailed in Exhibit 3.

34. Databricks is liable for its direct infringements of the '610 patent pursuant to 35 U.S.C. § 271.

Indirect Infringement (Inducement – 35 U.S.C. § 271(b))

35. In addition and/or in the alternative to its direct infringements, Databricks has indirectly infringed and continues to indirectly infringe one or more claims of the '610 patent by inducing direct infringement by its customers, partners, and end users.

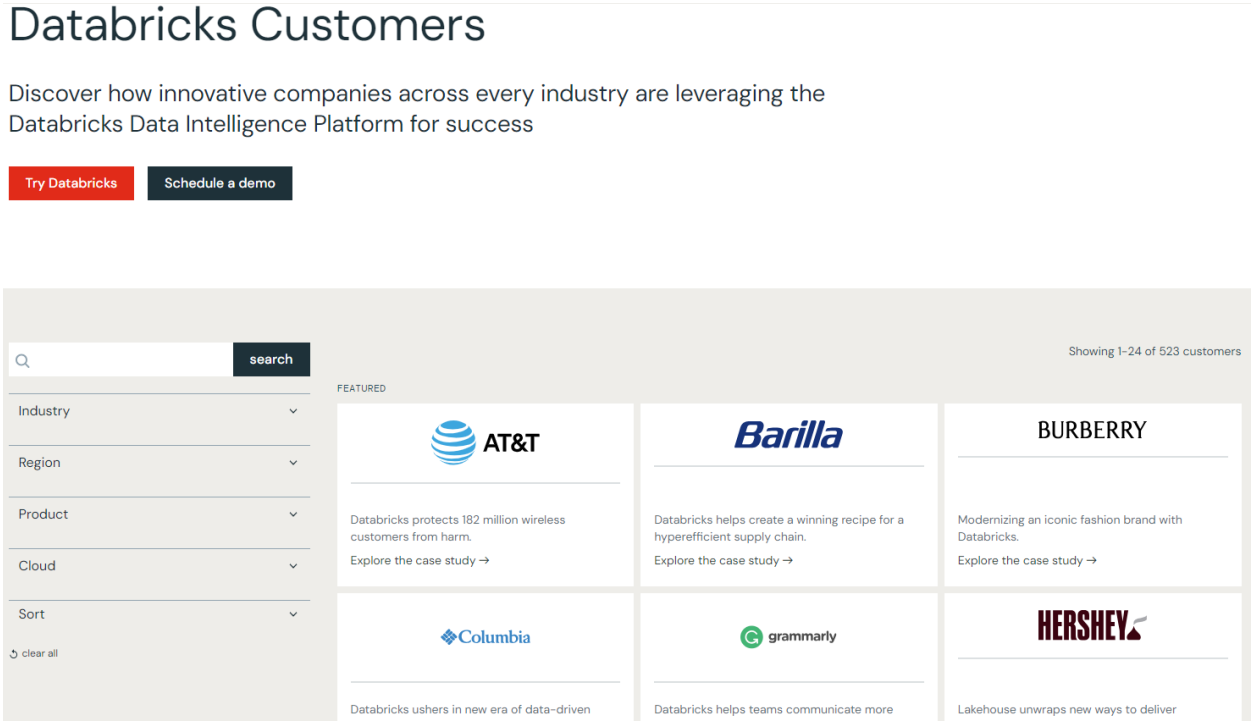
36. On information and belief, Databricks has had knowledge of the '610 patent and its infringements since shortly after April 28, 2022, when R2 filed the AA litigation. At the very least, Databricks has had knowledge of the '610 patent and its infringements since being served with a subpoena in connection with the AA litigation on January 10, 2023.

37. Despite having knowledge of the '610 patent and knowledge of its scope, Databricks has specifically intended, and continues to specifically intend, for persons (such as Databricks' customers, partners, and end users) to make the Accused Instrumentalities and use the Accused Instrumentalities in ways that infringe the '610 patent, including at least claims 1-

[4] https://www.databricks.com/product/pricing.

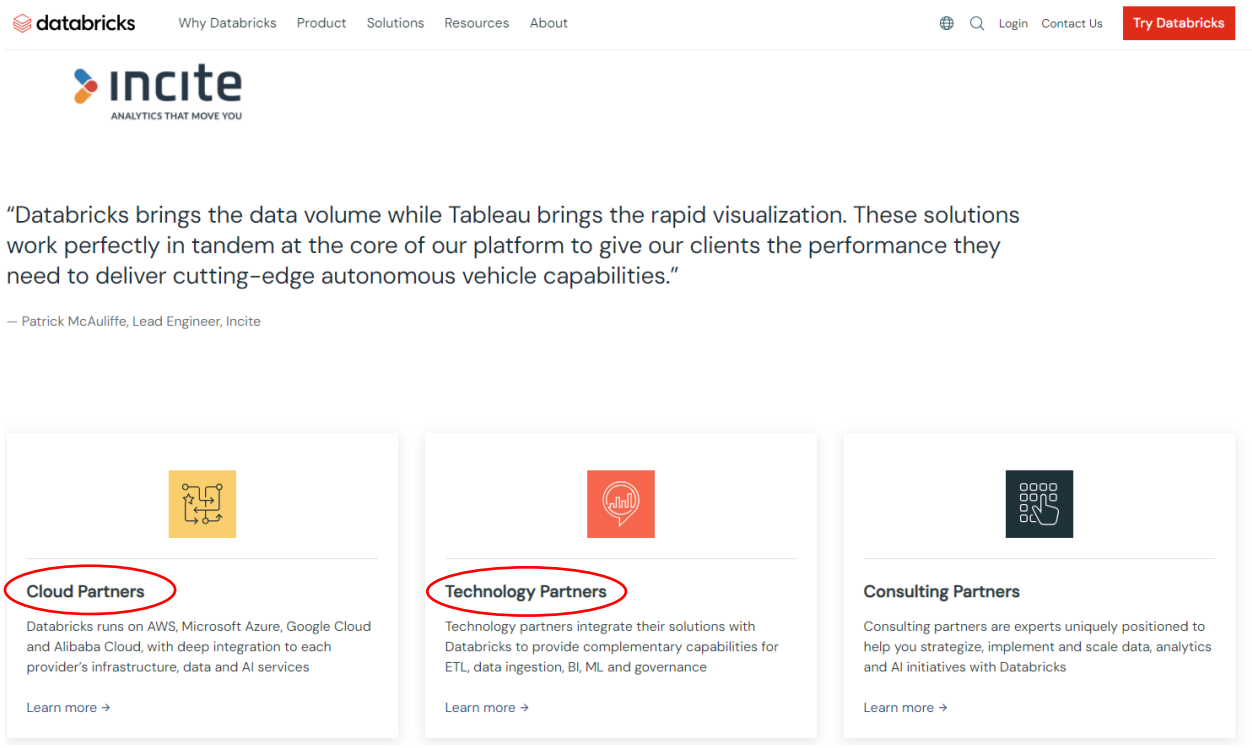
32. Databricks has also specifically intended, and continues to specifically intend, for its partners to offer for sale and sell the Accused Instrumentalities. Databricks knew or should have known that its actions have induced, and continue to induce, such infringements.

38. Databricks provides the Accused Instrumentalities to its customers:[5]



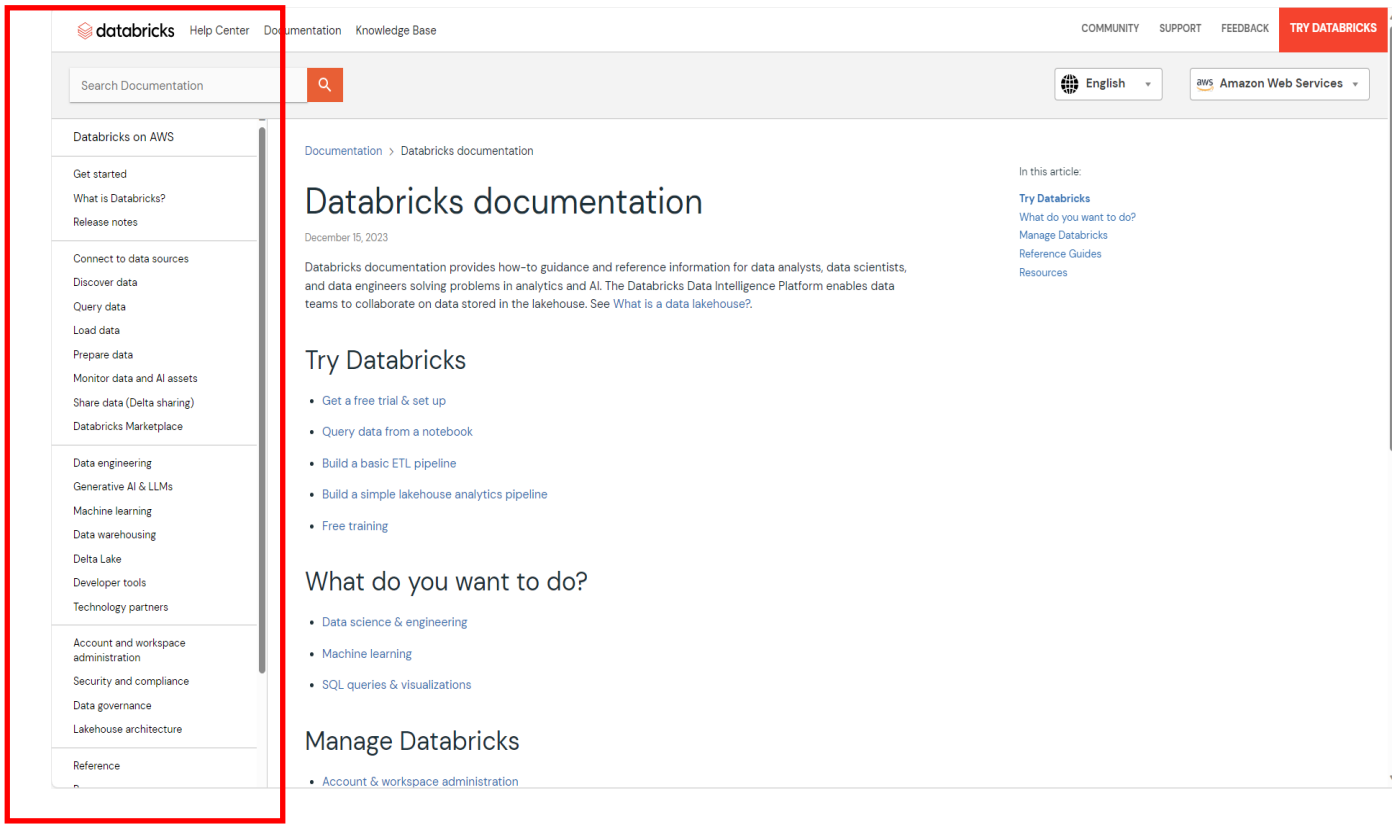
39. Databricks also provides its partners with the Accused Instrumentalities for distribution, resale, and/or to enable its partners to provide data analytics services to end users:

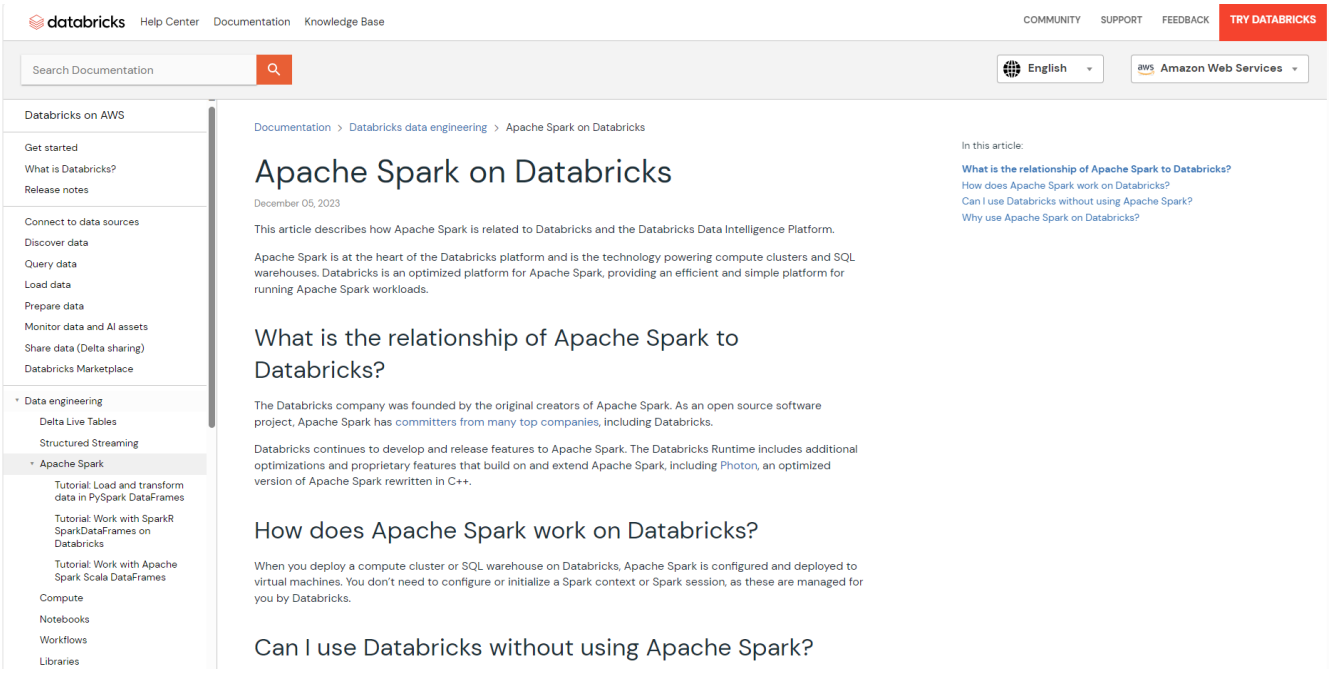
[5] https://www.databricks.com/customers.

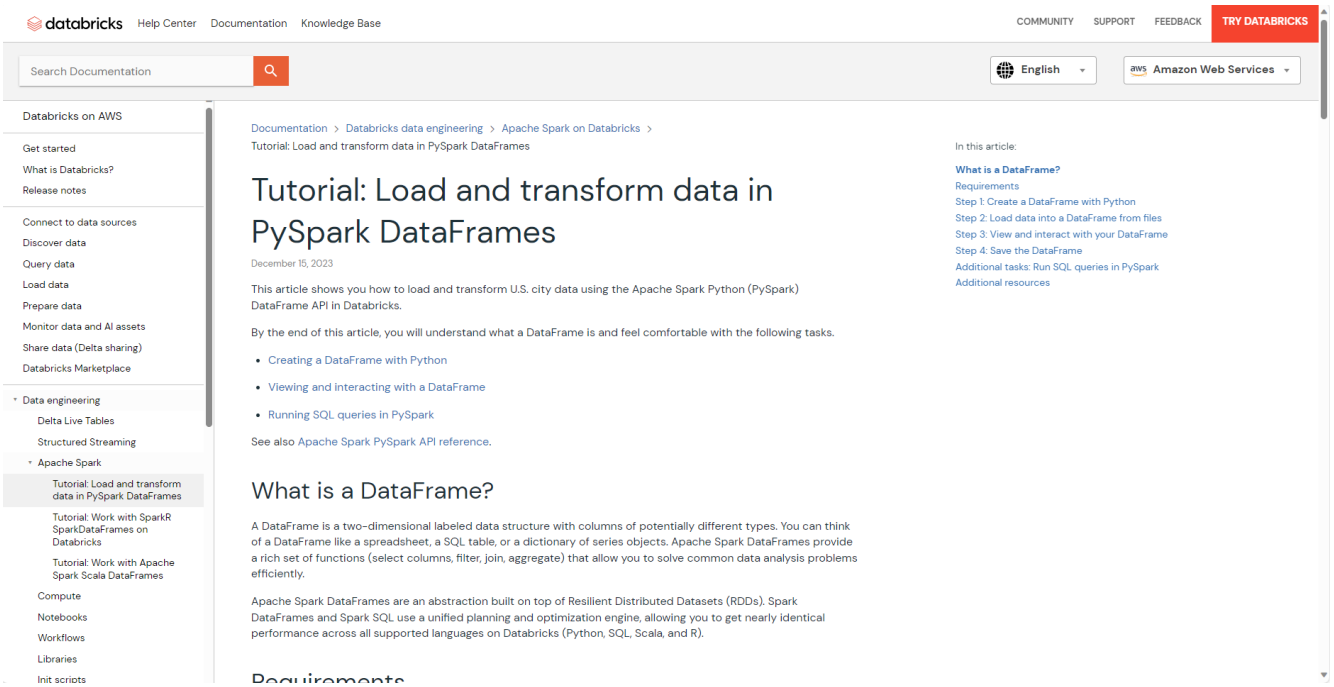


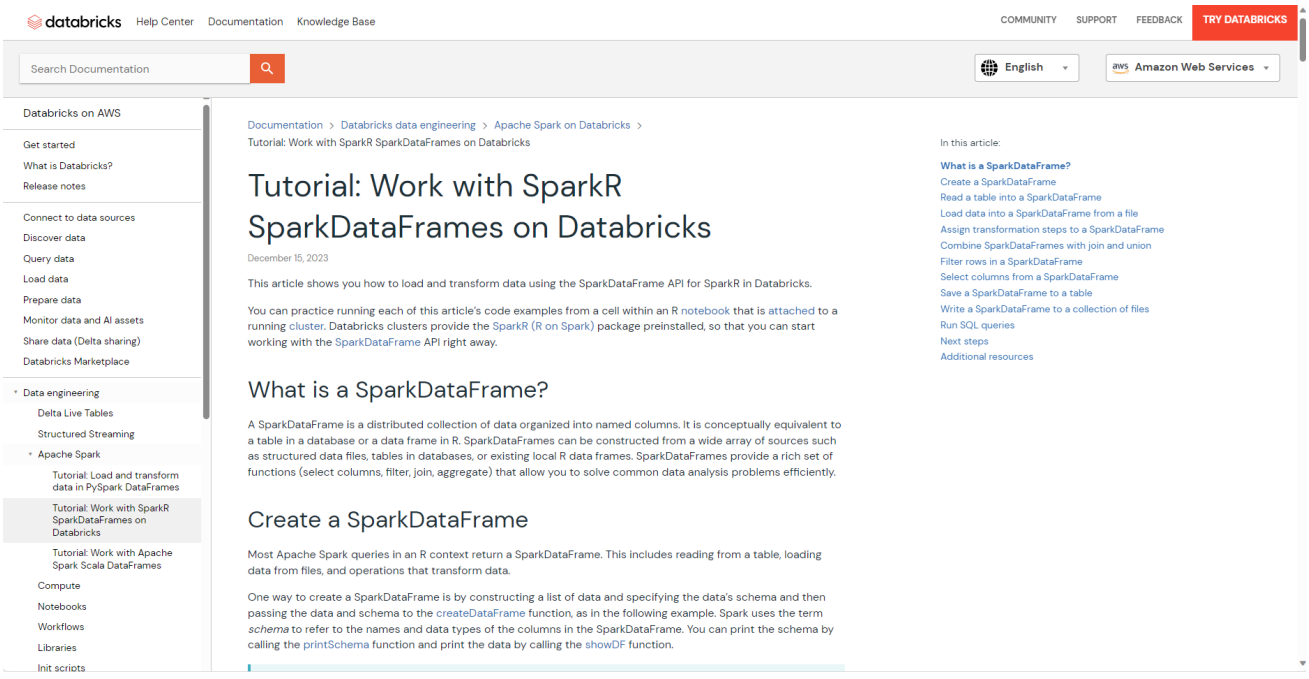
40. Databricks instructs and encourages partners, customers, and end users to make the Accused Instrumentalities and use the Accused Instrumentalities in ways that infringe the '610 patent. For example, the Databricks' website includes a "Documents" page with explicit instructions on how to implement and operate each Accused Instrumentality in an infringing manner:[6]

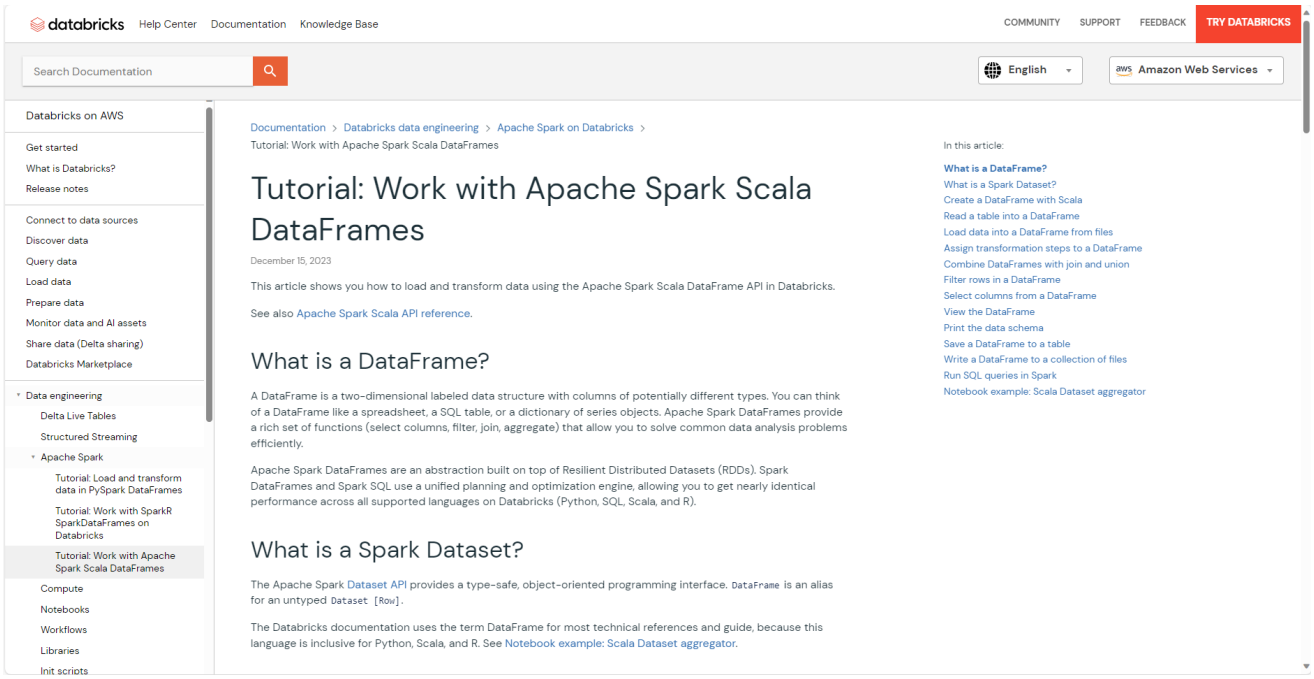
[6] https://docs.databricks.com/en/;
https://docs.databricks.com/en/spark/index.html;
https://docs.databricks.com/en/getting-started/dataframes-python.html;
https://docs.databricks.com/en/getting-started/dataframes-r.html;
https://docs.databricks.com/en/getting-started/dataframes-scala.html;
https://docs.databricks.com/en/delta-live-tables/transform.html.

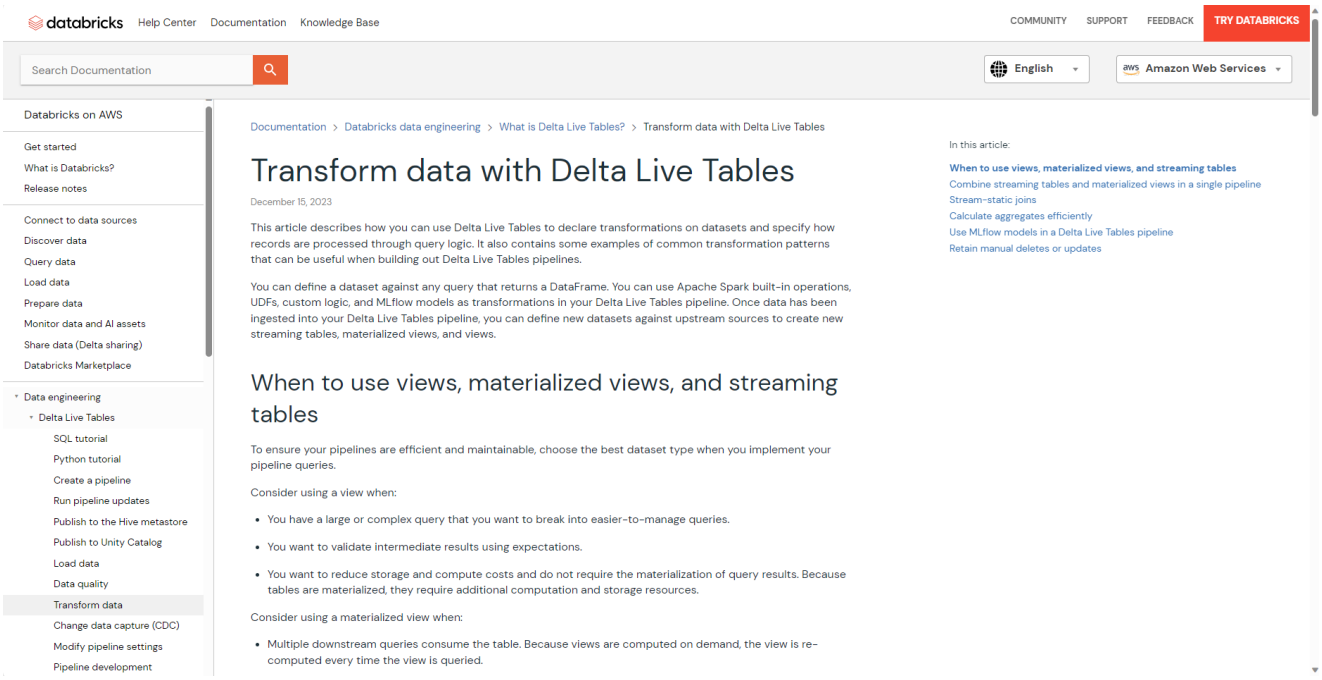

databricks
Help Center
Documentation
Knowledge Base
COMMUNITY
SUPPORT
FEEDBACK
TRY DATABRICKS
Search Documentation
English
Amazon Web Services
Databricks on AWS
Get started
What is Databricks?
Release notes
Connect to data sources
Discover data
Query data
Load data
Prepare data
Monitor data and AI assets
Share data (Delta sharing)
Databricks Marketplace
Data engineering
Generative AI & LLMs
Machine learning
Data warehousing
Delta Lake
Developer tools
Technology partners
Account and workspace administration
Security and compliance
Data governance
Lakehouse architecture
Reference
Documentation › Databricks documentation
Databricks documentation
December 15, 2023
Databricks documentation provides how-to guidance and reference information for data analysts, data scientists, and data engineers solving problems in analytics and AI. The Databricks Data Intelligence Platform enables data teams to collaborate on data stored in the lakehouse. See What is a data lakehouse?.
In this article:
Try Databricks
What do you want to do?
Manage Databricks
Reference Guides
Resources
Try Databricks
Get a free trial & set up
Query data from a notebook
Build a basic ETL pipeline
Build a simple lakehouse analytics pipeline
Free training
What do you want to do?
Data science & engineering
Machine learning
SQL queries & visualizations
Manage Databricks
Account & workspace administration


databricks
Help Center
Documentation
Knowledge Base
COMMUNITY
SUPPORT
FEEDBACK
TRY DATABRICKS
Search Documentation
English
Amazon Web Services
Databricks on AWS
Get started
What is Databricks?
Release notes
Connect to data sources
Discover data
Query data
Load data
Prepare data
Monitor data and AI assets
Share data (Delta sharing)
Databricks Marketplace
Data engineering
Delta Live Tables
Structured Streaming
Apache Spark
Tutorial: Load and transform data in PySpark DataFrames
Tutorial: Work with SparkR SparkDataFrames on Databricks
Tutorial: Work with Apache Spark Scala DataFrames
Compute
Notebooks
Workflows
Libraries
Documentation › Databricks data engineering › Apache Spark on Databricks
Apache Spark on Databricks
December 05, 2023
This article describes how Apache Spark is related to Databricks and the Databricks Data Intelligence Platform.
Apache Spark is at the heart of the Databricks platform and is the technology powering compute clusters and SQL warehouses. Databricks is an optimized platform for Apache Spark, providing an efficient and simple platform for running Apache Spark workloads.
In this article:
What is the relationship of Apache Spark to Databricks?
How does Apache Spark work on Databricks?
Can I use Databricks without using Apache Spark?
Why use Apache Spark on Databricks?
What is the relationship of Apache Spark to Databricks?
The Databricks company was founded by the original creators of Apache Spark. As an open source software project, Apache Spark has committers from many top companies, including Databricks.
Databricks continues to develop and release features to Apache Spark. The Databricks Runtime includes additional optimizations and proprietary features that build on and extend Apache Spark, including Photon, an optimized version of Apache Spark rewritten in C++.
How does Apache Spark work on Databricks?
When you deploy a compute cluster or SQL warehouse on Databricks, Apache Spark is configured and deployed to virtual machines. You don't need to configure or initialize a Spark context or Spark session, as these are managed for you by Databricks.
Can I use Databricks without using Apache Spark?


databricks
Help Center
Documentation
Knowledge Base
COMMUNITY
SUPPORT
FEEDBACK
TRY DATABRICKS
Search Documentation
English
Amazon Web Services
Databricks on AWS
Get started
What is Databricks?
Release notes
Connect to data sources
Discover data
Query data
Load data
Prepare data
Monitor data and AI assets
Share data (Delta sharing)
Databricks Marketplace
Data engineering
Delta Live Tables
Structured Streaming
Apache Spark
Tutorial: Load and transform data in PySpark DataFrames
Tutorial: Work with SparkR SparkDataFrames on Databricks
Tutorial: Work with Apache Spark Scala DataFrames
Compute
Notebooks
Workflows
Libraries
Init scripts
Documentation > Databricks data engineering > Apache Spark on Databricks >
Tutorial: Load and transform data in PySpark DataFrames
Tutorial: Load and transform data in PySpark DataFrames
December 15, 2023
This article shows you how to load and transform U.S. city data using the Apache Spark Python (PySpark) DataFrame API in Databricks.
By the end of this article, you will understand what a DataFrame is and feel comfortable with the following tasks.
Creating a DataFrame with Python
Viewing and interacting with a DataFrame
Running SQL queries in PySpark
See also Apache Spark PySpark API reference.
What is a DataFrame?
A DataFrame is a two-dimensional labeled data structure with columns of potentially different types. You can think of a DataFrame like a spreadsheet, a SQL table, or a dictionary of series objects. Apache Spark DataFrames provide a rich set of functions (select columns, filter, join, aggregate) that allow you to solve common data analysis problems efficiently.
Apache Spark DataFrames are an abstraction built on top of Resilient Distributed Datasets (RDDs). Spark DataFrames and Spark SQL use a unified planning and optimization engine, allowing you to get nearly identical performance across all supported languages on Databricks (Python, SQL, Scala, and R).
Requirements
In this article:
What is a DataFrame?
Requirements
Step 1: Create a DataFrame with Python
Step 2: Load data into a DataFrame from files
Step 3: View and interact with your DataFrame
Step 4: Save the DataFrame
Additional tasks: Run SQL queries in PySpark
Additional resources


databricks
Help Center
Documentation
Knowledge Base
COMMUNITY
SUPPORT
FEEDBACK
TRY DATABRICKS
Search Documentation
English
Amazon Web Services
Databricks on AWS
Get started
What is Databricks?
Release notes
Connect to data sources
Discover data
Query data
Load data
Prepare data
Monitor data and AI assets
Share data (Delta sharing)
Databricks Marketplace
Data engineering
Delta Live Tables
Structured Streaming
Apache Spark
Tutorial: Load and transform data in PySpark DataFrames
Tutorial: Work with SparkR SparkDataFrames on Databricks
Tutorial: Work with Apache Spark Scala DataFrames
Compute
Notebooks
Workflows
Libraries
Init scripts
Documentation > Databricks data engineering > Apache Spark on Databricks >
Tutorial: Work with SparkR SparkDataFrames on Databricks
Tutorial: Work with SparkR SparkDataFrames on Databricks
December 15, 2023
This article shows you how to load and transform data using the SparkDataFrame API for SparkR in Databricks.
You can practice running each of this article's code examples from a cell within an R notebook that is attached to a running cluster. Databricks clusters provide the SparkR (R on Spark) package preinstalled, so that you can start working with the SparkDataFrame API right away.
What is a SparkDataFrame?
A SparkDataFrame is a distributed collection of data organized into named columns. It is conceptually equivalent to a table in a database or a data frame in R. SparkDataFrames can be constructed from a wide array of sources such as structured data files, tables in databases, or existing local R data frames. SparkDataFrames provide a rich set of functions (select columns, filter, join, aggregate) that allow you to solve common data analysis problems efficiently.
Create a SparkDataFrame
Most Apache Spark queries in an R context return a SparkDataFrame. This includes reading from a table, loading data from files, and operations that transform data.
One way to create a SparkDataFrame is by constructing a list of data and specifying the data's schema and then passing the data and schema to the createDataFrame function, as in the following example. Spark uses the term *schema* to refer to the names and data types of the columns in the SparkDataFrame. You can print the schema by calling the printSchema function and print the data by calling the showDF function.
In this article:
What is a SparkDataFrame?
Create a SparkDataFrame
Read a table into a SparkDataFrame
Load data into a SparkDataFrame from a file
Assign transformation steps to a SparkDataFrame
Combine SparkDataFrames with join and union
Filter rows in a SparkDataFrame
Select columns from a SparkDataFrame
Save a SparkDataFrame to a table
Write a SparkDataFrame to a collection of files
Run SQL queries
Next steps
Additional resources





41. Other exemplary instructions and documentation that explain how to make and use the Accused Instrumentalities in an infringing manner are set out in Exhibits 2 and 3.

Damages

42. R2 has been damaged as a result of Databricks' infringing conduct described in this Count. Databricks is, thus, liable to R2 in an amount that adequately compensates it for Databricks' infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

43. Despite having knowledge of the '610 patent, and knowledge that it is potentially directly and/or indirectly infringing claims of the '610 patent, Databricks has nevertheless continued its infringing conduct in an egregious manner. On information and belief, Databricks knew of the '610 patent and its scope, yet continued to manufacture, use, and sell infringing products. At the very least, Databricks was willfully blind to the '610 patent and its application to the Accused Instrumentalities. For at least these reasons, Databricks' infringing activities have been, and continue to be, willful, wanton, and deliberate in disregard of R2's rights with respect to the '610 patent, justifying enhanced damages under 35 U.S.C. § 284.

DEMAND FOR A JURY TRIAL

R2 demands a trial by jury on all issues triable of right by jury pursuant to Rule 38 of the Federal Rules of Civil Procedure.

PRAYER FOR RELIEF

R2 respectfully requests that this Court enter judgment in its favor and grant the following relief:

(i) Judgment and Order that Databricks has directly and/or indirectly infringed one or more claims of the patent-in-suit;

(ii) Judgment and Order that Databricks must pay R2 past and future damages under 35 U.S.C. § 284, including supplemental damages arising from any continuing,

post-verdict infringement for the time between trial and entry of the final judgment, together with an accounting, as needed, as provided under 35 U.S.C. § 284;

(iii) Judgment and Order that Databricks must pay R2 reasonable ongoing royalties on a go-forward basis after Final Judgment;

(iv) Judgment and Order that Databricks' infringement of the '610 patent has been willful from the time that Databricks became aware of the infringing nature of its products, and that the Court award treble damages pursuant to 35 U.S.C. § 284;

(v) Judgment and Order that Databricks must pay R2 pre-judgment and post-judgment interest on the damages award;

(vi) Judgment and Order that Databricks must pay R2's costs;

(vii) Judgment and Order that the Court find this case exceptional under the provisions of 35 U.S.C. § 285 and, accordingly, order Databricks to pay R2's attorneys' fees; and

(viii) Such other and further relief as the Court may deem just and proper.

Dated: December 28, 2023 Respectfully submitted,

/s/ Edward R. Nelson III
EDWARD R. NELSON III
State Bar No. 00797142
ed@nelbum.com
BRENT N. BUMGARDNER
State Bar No. 00795272
brent@nelbum.com
CHRISTOPHER G. GRANAGHAN
State Bar No. 24078585
chris@nelbum.com
JOHN P. MURPHY
State Bar No. 24056024
murphy@nelbum.com

CARDER W. BROOKS
State Bar No. 24105536
carder@nelbum.com
NELSON BUMGARDNER CONROY PC
3131 West 7th Street, Suite 300
Fort Worth, Texas 76107
817.377.9111

COUNSEL FOR PLAINTIFF
R2 SOLUTIONS LLC