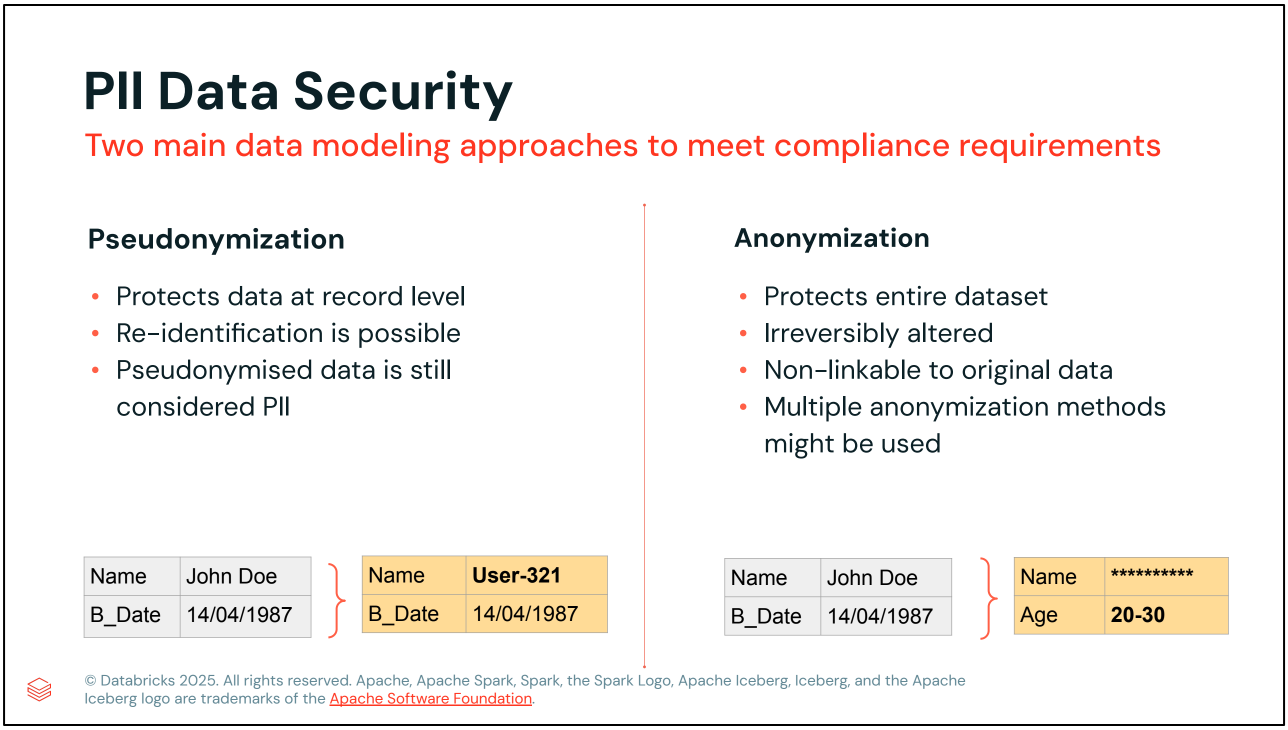
databricks

PII Data Security

LECTURE

Pseudonymization & Anonymization



PII Data Security

Two main data modeling approaches to meet compliance requirements

Pseudonymization

- Protects data at record level
- Re-identification is possible
- Pseudonymised data is still considered PII

Name	John Doe
B_Date	14/04/1987

Name	**User-321**
B_Date	14/04/1987

Anonymization

- Protects entire dataset
- Irreversibly altered
- Non-linkable to original data
- Multiple anonymization methods might be used

Name	John Doe
B_Date	14/04/1987

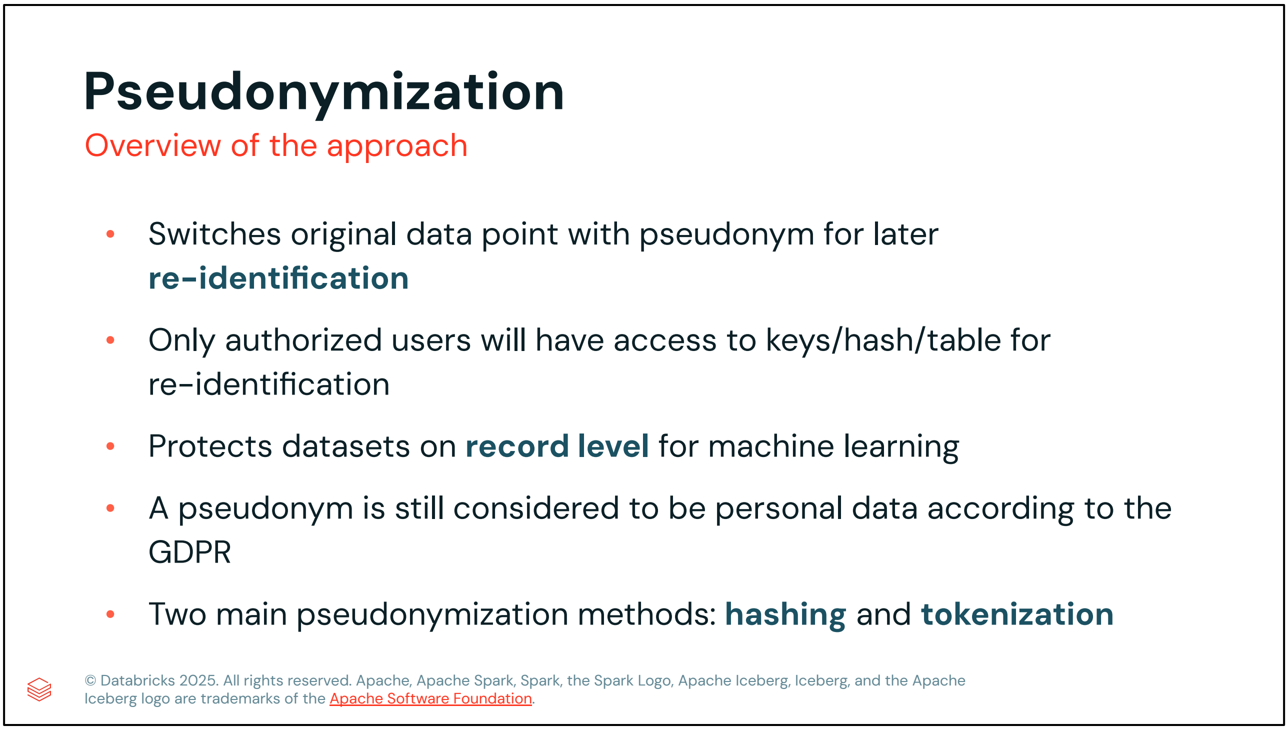
Name	***********
Age	**20-30**

Let's dive a bit deeper into approaches for pseudonymization and anonymization.

However, before we begin, please note that with sufficient time and access to additional data, it is often possible to re-identify most data regardless of the approach used. Applying pseudonymization and anonymization techniques to datasets will reduce the risk of data exfiltration and reduce visibility to most users of the dataset, but will not eliminate the risk altogether.

On the left you can see an example of pseudonymization, where “John Doe” in the Name column was changed to “User-321”. This key can also be used to join data later on. If you do have to delete it, it's easy to make sure that the data isn't compromised in other areas.

The image on the right is an example of anonymization. This can be implemented using dynamic views and based on the permissions that you configure for a user group or member. Specifically, you can hide or change the way you present certain information.



Pseudonymization

Overview of the approach

- Switches original data point with pseudonym for later **re-identification**
- Only authorized users will have access to keys/hash/table for re-identification
- Protects datasets on **record level** for machine learning
- A pseudonym is still considered to be personal data according to the GDPR
- Two main pseudonymization methods: **hashing** and **tokenization**

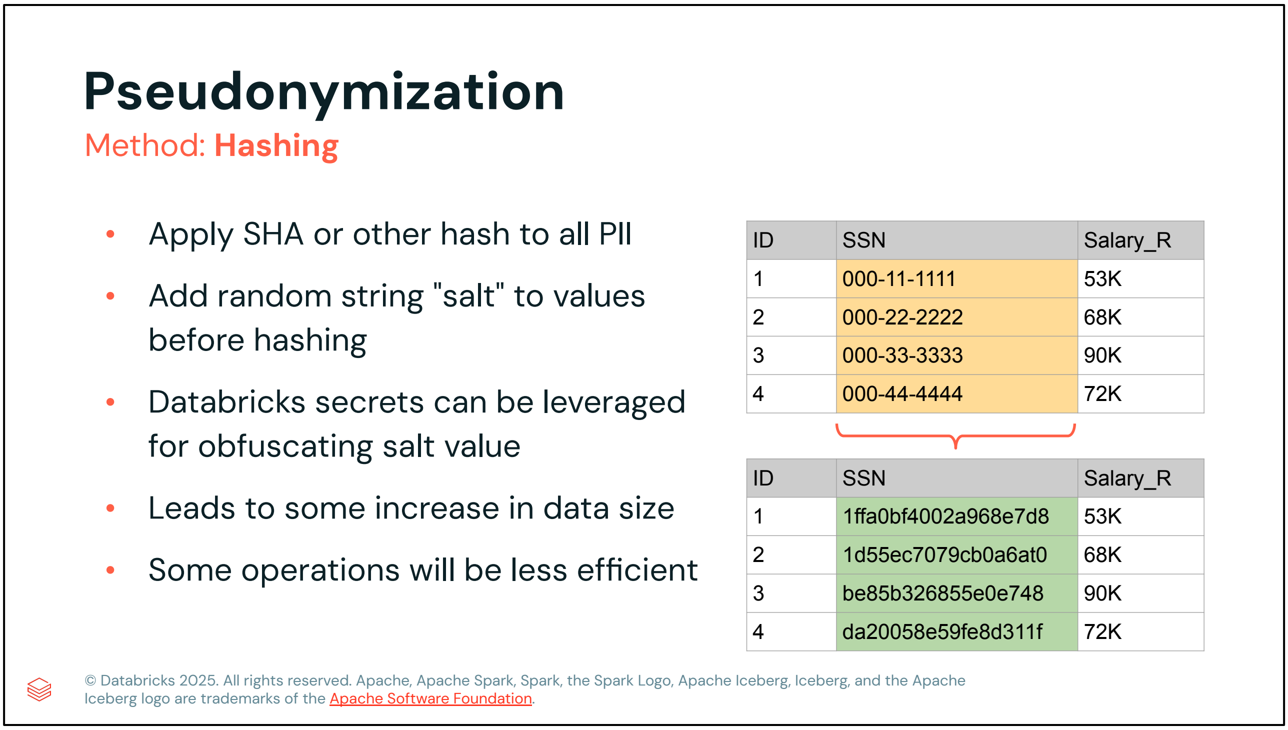
Pseudonymization involves replacing PII or personally identifiable information with artificial identifiers or pseudonyms. Be aware that even pseudonymized data is still considered personal information.

It offers data protection at a record level, replacing meaningful values with generated but equally unique values such as tokens, hashes, or encrypted data.

Pseudonymization allows the reverse of the process and re-identification when necessary.

Applying pseudonymization to personal data can reduce the risks to the data subjects concerned and help controllers and processors meet their data protection requirements. Data scientists can still work on complete records but cannot easily access the literal values represented by the random strings they are analyzing.

Here we have two options: hashing and tokenization



Pseudonymization

Method: **Hashing**

- Apply SHA or other hash to all PII
- Add random string "salt" to values before hashing
- Databricks secrets can be leveraged for obfuscating salt value
- Leads to some increase in data size
- Some operations will be less efficient

ID	SSN	Salary_R
1	000-11-1111	53K
2	000-22-2222	68K
3	000-33-3333	90K
4	000-44-4444	72K

ID	SSN	Salary_R
1	1ffa0bf4002a968e7d8	53K
2	1d55ec7079cb0a6at0	68K
3	be85b326855e0e748	90K
4	da20058e59fe8d311f	72K

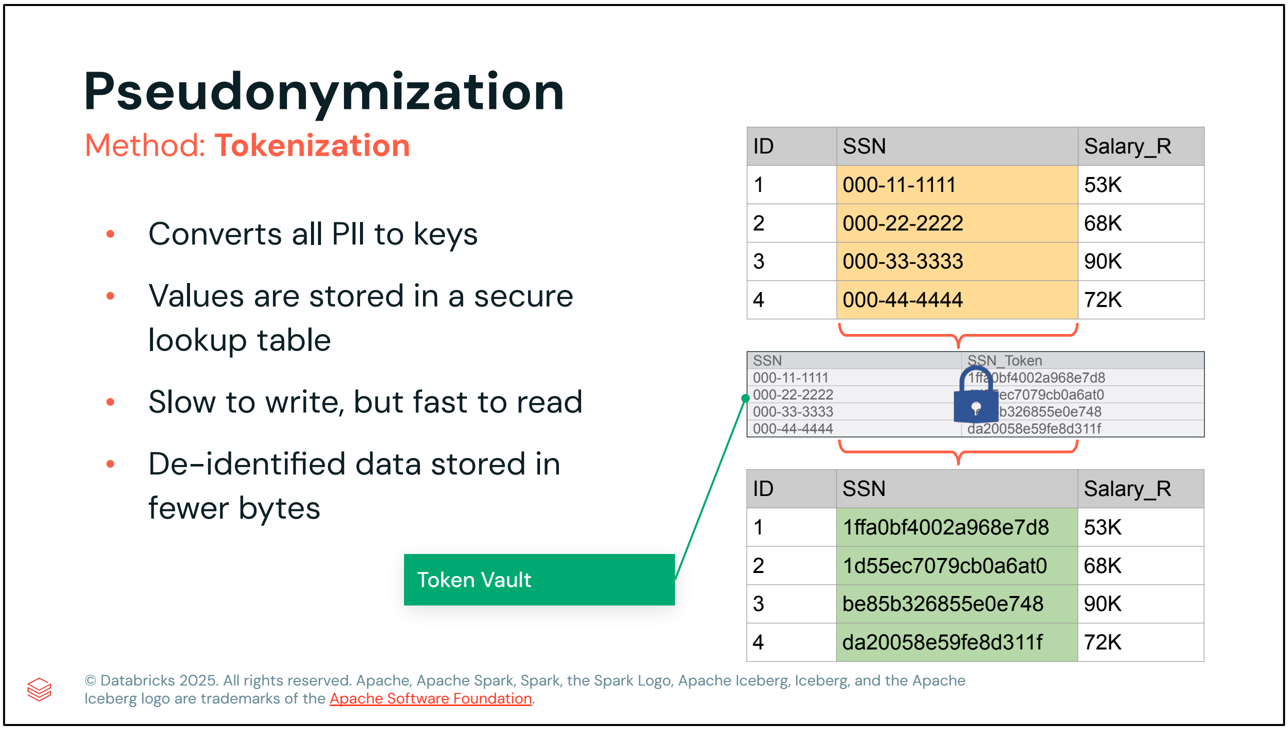
Applying a hashing function to personally identifiable information will result in a random string of characters obscuring data values from end users.

Because hashes are deterministic, adding a random string to the beginning or end of a value can help to reduce the risk of reversing a hash. This is called salting. For example, if you're passing in a specific social security number, instead of just inputting the value to the hashing function, you can append it to a randomized string.

We can also use the Databricks Secrets API in order to store these salt values. Permissions to these secrets can be granted only to production jobs and authorized users, and will ensure that salt values are never displayed in plain text within the notebooks.

While hashing does not require a full re-architecture of a data system, it will lead to some increase in data size as hash values take up more bytes than the data they replace. Note that different hashes will use different numbers of bytes. Some operations will be less efficient, and certain data may need to be extracted prior to hashing to allow downstream processing.

For example, if an ML pre-processing step extracts the domain from an email address prior to prediction, the domain should be stored and hashed in a separate column from the full email address hash so that the hashed domain can still be used.



Pseudonymization

Method: **Tokenization**

- Converts all PII to keys
- Values are stored in a secure lookup table
- Slow to write, but fast to read
- De-identified data stored in fewer bytes

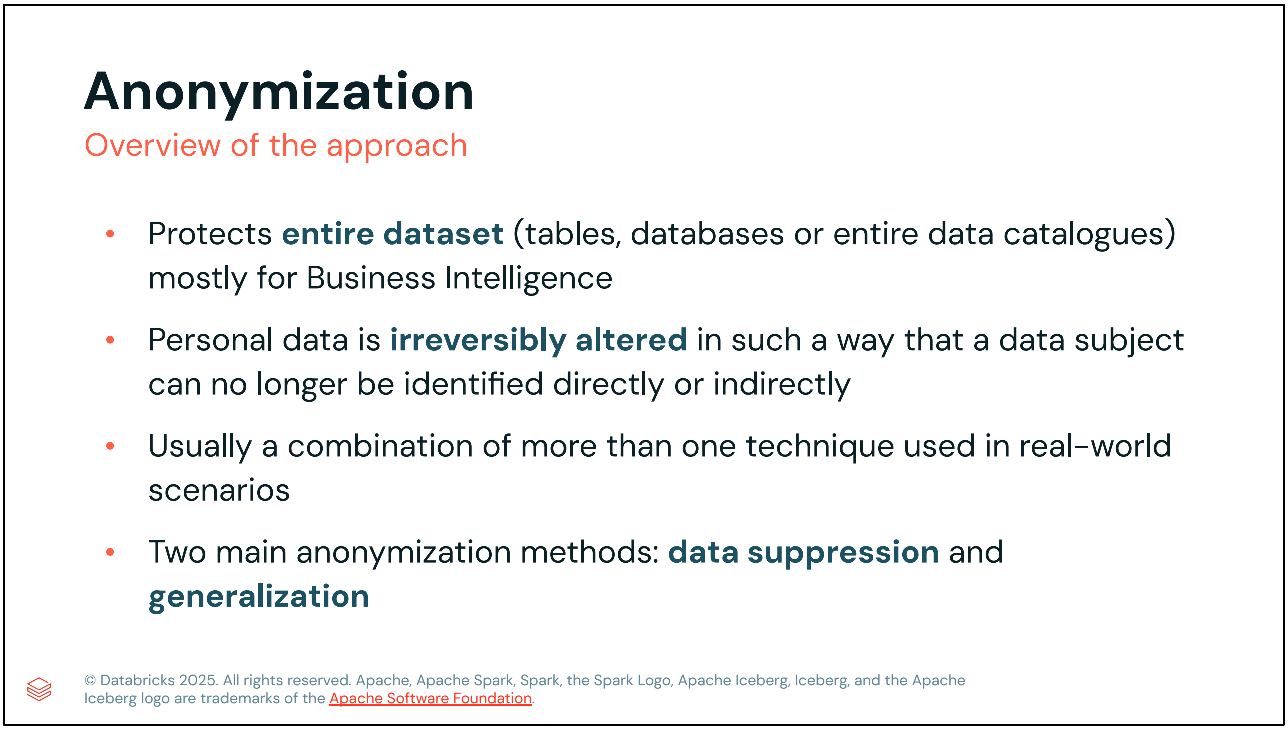
ID	SSN	Salary_R
1	000-11-1111	53K
2	000-22-2222	68K
3	000-33-3333	90K
4	000-44-4444	72K

SSN	SSN_Token
000-11-1111	1ffa0bf4002a968e7d8
000-22-2222	[illegible]ec7079cb0a6at0
000-33-3333	[illegible]b326855e0e748
000-44-4444	da20058e59fe8d311f

ID	SSN	Salary_R
1	1ffa0bf4002a968e7d8	53K
2	1d55ec7079cb0a6at0	68K
3	be85b326855e0e748	90K
4	da20058e59fe8d311f	72K

With tokenization, we'll convert all of our PII to keys and then end up storing our values in a secure lookup table. Know that this is slow to write, but fast to read, in part because our de-identified data will be stored in fewer bytes, typically just encoding the key that we'll use to look up the data in a long value.

To tokenize a dataset, we're going to start by getting all of the columns and turning them into an array of structs. And once all the values have been identified, each unique value is assigned a token. These tokens are used as keys for unique values in the token vault. The table exposed to end users just contains these keys, usually stored as long values.



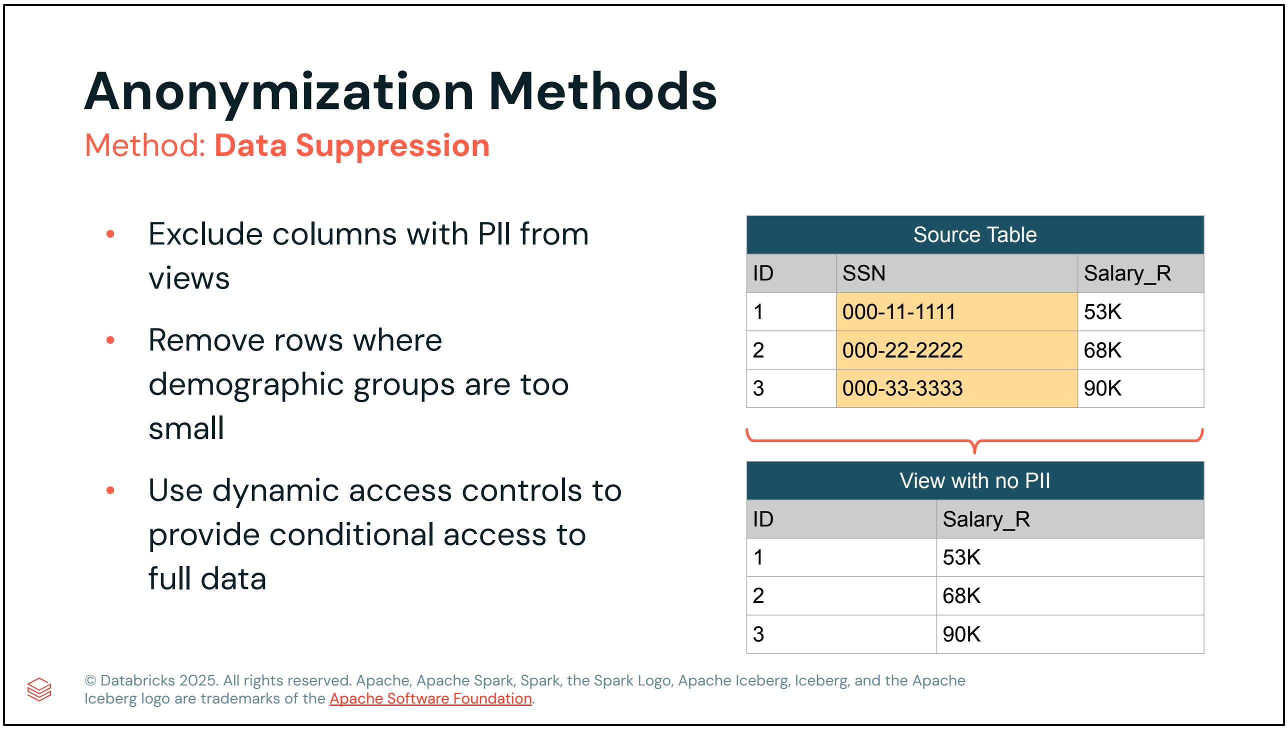
Anonymization

Overview of the approach

- Protects **entire dataset** (tables, databases or entire data catalogues) mostly for Business Intelligence
- Personal data is **irreversibly altered** in such a way that a data subject can no longer be identified directly or indirectly
- Usually a combination of more than one technique used in real-world scenarios
- Two main anonymization methods: **data suppression** and **generalization**

About Anonymization:

- It protects entire datasets, including tables, databases, and whole data catalogs, and it irreversibly alters personal data to prevent direct or indirect identification of data subjects. This is not a problem, as Business analysts are often more interested in aggregations and trends that can still be tracked without seeing each record.
- Anonymization typically employs a combination of multiple techniques in real-world scenarios, such as data suppression and generalization



Anonymization Methods

Method: **Data Suppression**

- Exclude columns with PII from views
- Remove rows where demographic groups are too small
- Use dynamic access controls to provide conditional access to full data

Source Table

ID	SSN	Salary_R
1	000-11-1111	53K
2	000-22-2222	68K
3	000-33-3333	90K

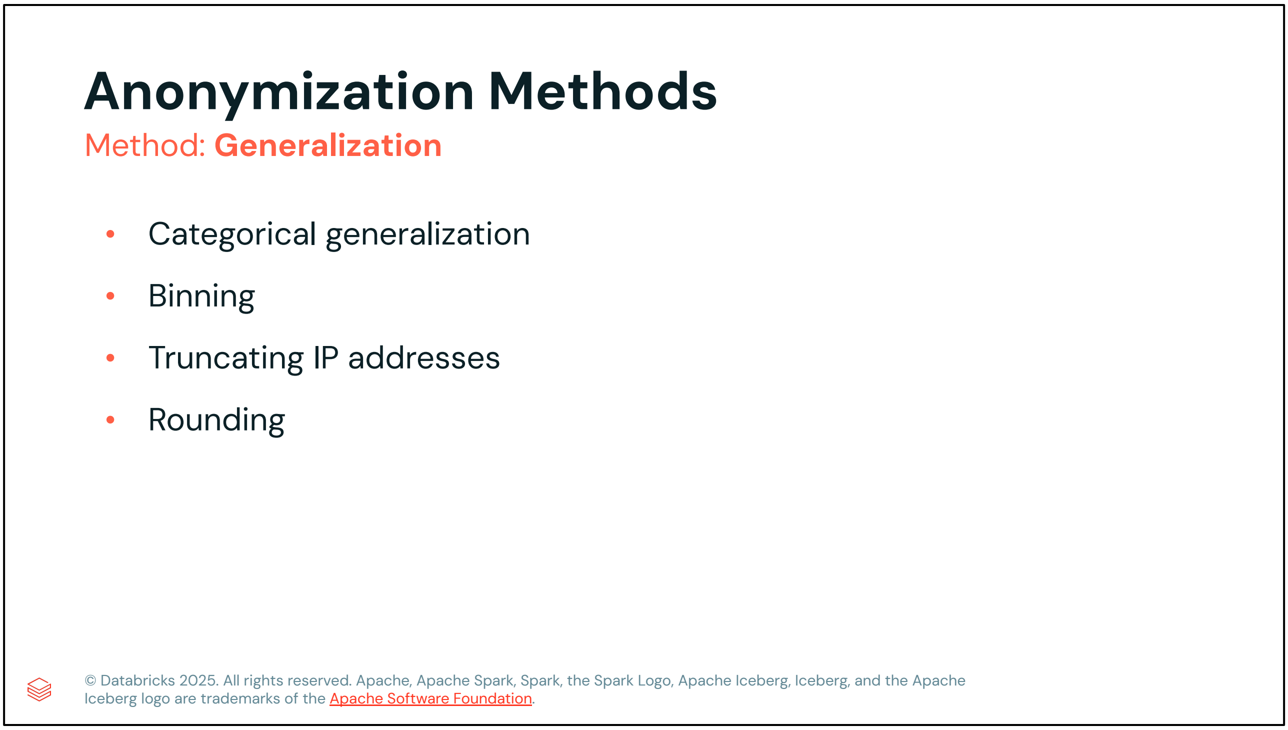
View with no PII

ID	Salary_R
1	53K
2	68K
3	90K

Conditional filters and dynamic access controls can be used to remove access to columns or rows of data without reducing the ability of analysts to do reporting.

For example, aggregate reporting for a region can still be completed without access to full customer names or addresses. In some cases, demographic data can be easily used to re-identify someone. For example, even large companies may have a limited number of customers living in a small city, especially when a given geography has only one corresponding record.

Aggregation will not provide any obfuscation of underlying data and can potentially expose sensitive data through reports and dashboards. Setting a filter to remove rows with low counts for grouping columns can help to provide protection for individual identities. Databricks has dynamic access controls that allow data to be redacted or filtered based on group memberships, which we'll discuss later in the course.

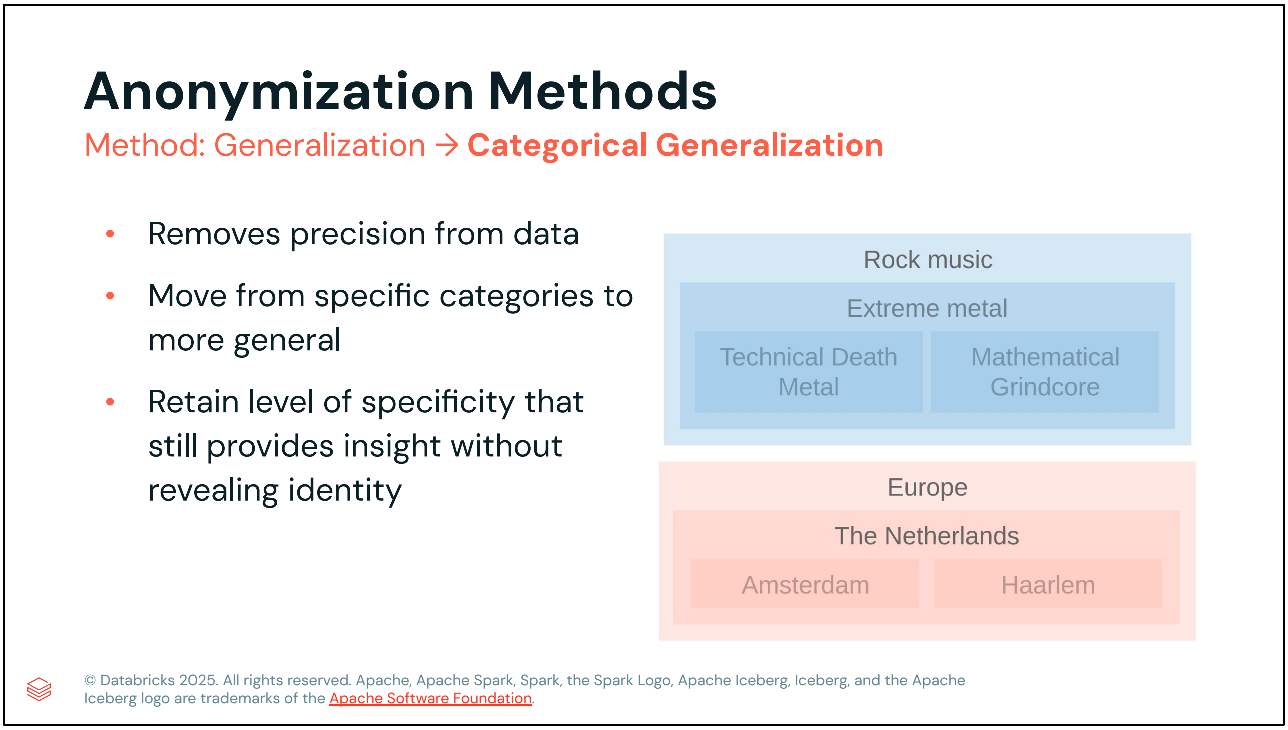


Anonymization Methods

Method: **Generalization**

- Categorical generalization
- Binning
- Truncating IP addresses
- Rounding

Generalization can be thought of as a way of anonymizing data by removing specificity. Different types of data support different types of generalization, such as categorical generalization, binning, truncating IP addresses, and rounding.



Anonymization Methods

Method: Generalization → **Categorical Generalization**

- Removes precision from data
- Move from specific categories to more general
- Retain level of specificity that still provides insight without revealing identity

Rock music

Extreme metal

Technical Death Metal

Mathematical Grindcore

Europe

The Netherlands

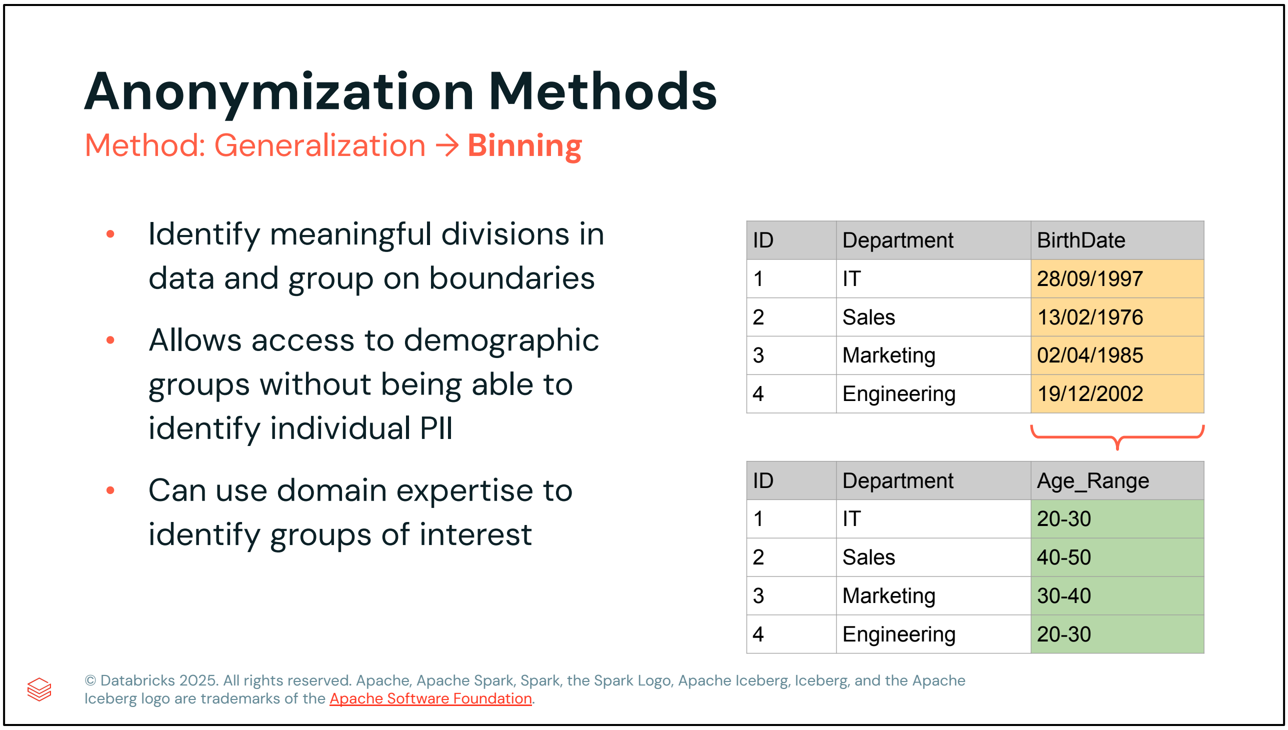
Amsterdam

Haarlem

With categorical generalization, the goal is to remove precision from data. In this example, we're grouping smaller cities into larger regional groups like state or country to ensure that smaller geographies are not revealed.

So when you submit an anonymous feedback survey, for example, but they end up analyzing it and they group it by team, and you were the only person who responded on your team, all your information is exposed.

So instead of that, you would be anonymizing that level of specificity, and then grouping by department or organization. And because so much data from social media has been collected and leaked, even seemingly innocuous preferences can be used to easily identify and target individuals.



Anonymization Methods

Method: Generalization → **Binning**

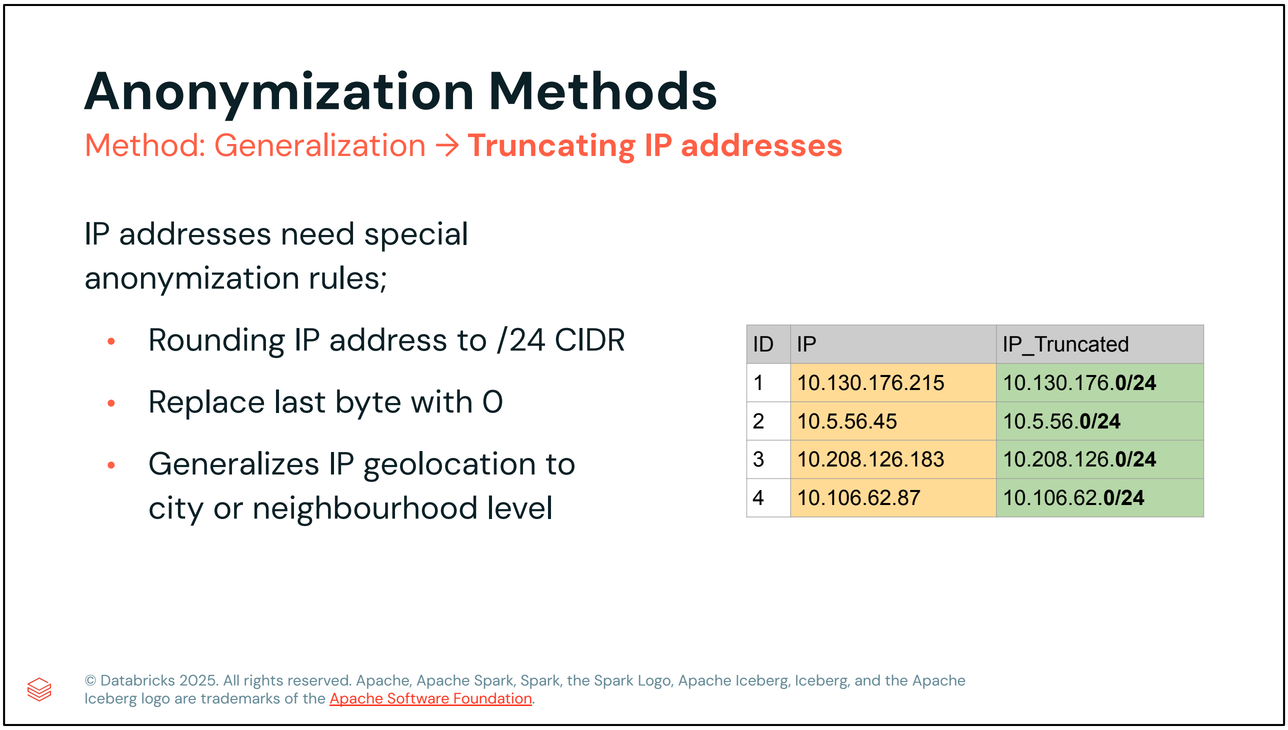
- Identify meaningful divisions in data and group on boundaries
- Allows access to demographic groups without being able to identify individual PII
- Can use domain expertise to identify groups of interest

ID	Department	BirthDate
1	IT	28/09/1997
2	Sales	13/02/1976
3	Marketing	02/04/1985
4	Engineering	19/12/2002

ID	Department	Age_Range
1	IT	20-30
2	Sales	40-50
3	Marketing	30-40
4	Engineering	20-30

Some examples of binning would be creating a 10-year age range to report on age-based trends or grouping salaries into bands based on published standards. Reports and dashboards can still provide meaningful insights, but analysts will be unable to identify the exact salary of a given individual.

Domain expertise can come in handy when going to define meaningful groups. Analysts can help to define how bins will be calculated based on reporting needs. So based on the use cases of whoever is analyzing your data, you would want to create these groups that make sense. In some cases you might not actually need that information. In others, you have to figure out a different way because you're just getting rid of that information. So it is very specific to the audience and the use case.



Anonymization Methods

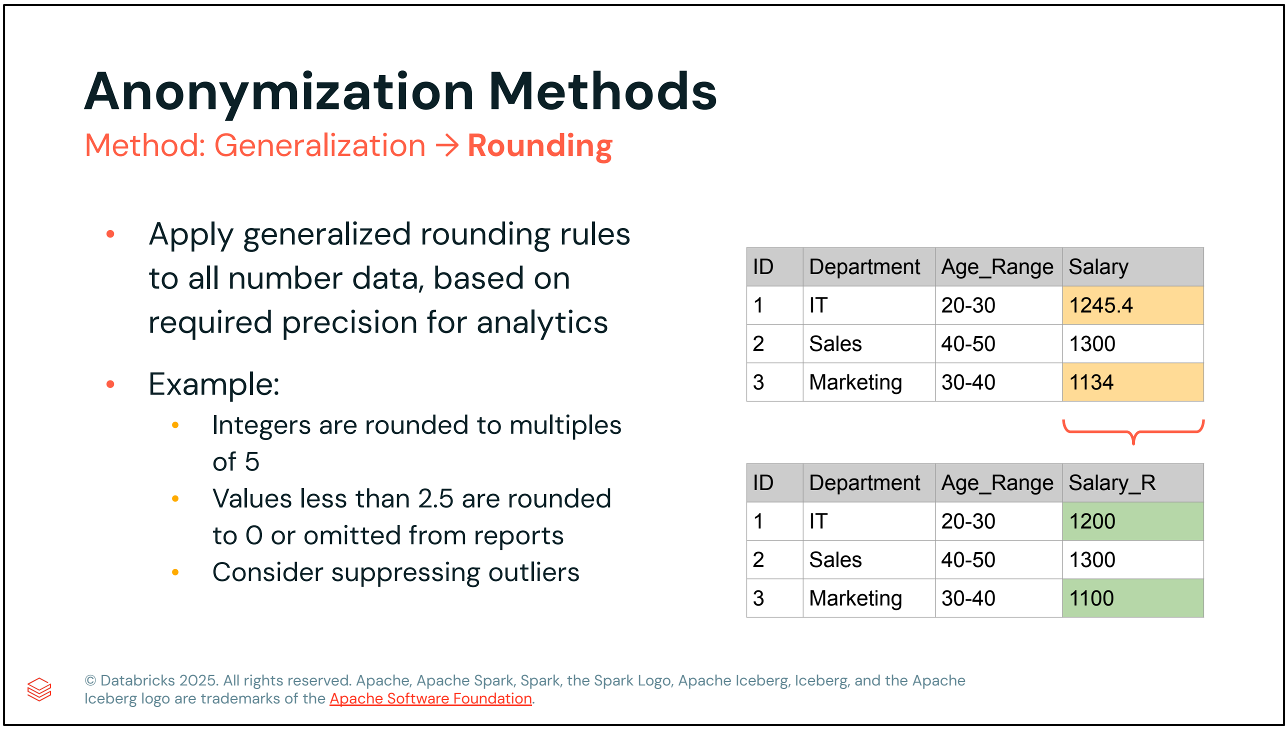
Method: Generalization → **Truncating IP addresses**

IP addresses need special anonymization rules;

- Rounding IP address to /24 CIDR
- Replace last byte with 0
- Generalizes IP geolocation to city or neighbourhood level

ID	IP	IP_Truncated
1	10.130.176.215	10.130.176.**0/24**
2	10.5.56.45	10.5.56.**0/24**
3	10.208.126.183	10.208.126.**0/24**
4	10.106.62.87	10.106.62.**0/24**

Another type of generalized anonymization method is truncating, which is a common use case for IP addresses. To truncate an IP address, we can take the last byte from it and replace it with a zero so that it is in the /24 CIDR range.



Anonymization Methods

Method: Generalization → **Rounding**

- Apply generalized rounding rules to all number data, based on required precision for analytics
- Example:
  - Integers are rounded to multiples of 5
  - Values less than 2.5 are rounded to 0 or omitted from reports
  - Consider suppressing outliers

ID	Department	Age_Range	Salary
1	IT	20-30	1245.4
2	Sales	40-50	1300
3	Marketing	30-40	1134

ID	Department	Age_Range	Salary_R
1	IT	20-30	1200
2	Sales	40-50	1300
3	Marketing	30-40	1100

Many reports can be safely completed with rounded data. General trends will be the same as in unrounded analytics because data will be equally rounded up and down. So consider the precision necessary to provide insights. If trends appear in the thousands, there's no need to store or expose precision to the tens place.

A simple example would be rounding everything to the nearest 5. Note that even with rounding, the lowest and highest groups may still reveal outliers that need to be suppressed.