Lecture 21

Database Processing
SQL and noSQL

Several programming languages and query languages are commonly used to interact with and process queries in databases and data warehouses. The choice of language often depends on the specific database management system (DBMS) or data warehouse being used. Here are some common languages for querying and processing data:

SQL (Structured Query Language):
Description: SQL is the standard language for managing and querying relational databases. It allows users to define, manipulate, and query data in a relational database.
Common Use: Used with relational database management systems (RDBMS) like MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and others.

PL/pgSQL (Procedural Language/PostgreSQL):
Description: An extension of SQL that adds procedural programming features. It is specific to the PostgreSQL database system.
Common Use: Used for writing stored procedures and functions in PostgreSQL.

T-SQL (Transact-SQL):
Description: An extension of SQL developed by Microsoft. It includes additional procedural programming constructs and is used with Microsoft SQL Server.
Common Use: Used for writing stored procedures, triggers, and other procedural code in SQL Server.

PL/SQL (Procedural Language/SQL):
Description: Oracle's extension of SQL with procedural language features. It allows for the creation of stored procedures, functions, and triggers.
Common Use: Used with Oracle Database for writing procedural code.

MDX (Multidimensional Expressions):
Description: A query language for querying multidimensional databases, often associated with Online Analytical Processing (OLAP) systems.
Common Use: Used in conjunction with databases that support OLAP, such as Microsoft SQL Server Analysis Services and others.

DAX (Data Analysis Expressions):
Description: A formula language used in Power BI, Excel Power Pivot, and SQL Server Analysis Services. It is used for creating custom formulas and expressions.
Common Use: Commonly used in business intelligence and analytics scenarios.

CQL (Cassandra Query Language):
Description: A query language for Apache Cassandra, a NoSQL database. It is similar to SQL but tailored for the Cassandra data model.
Common Use: Used for querying and interacting with Apache Cassandra databases.

HiveQL:
Description: The query language for Apache Hive, a data warehouse infrastructure built on top of Hadoop. It provides SQL-like queries for data stored in Hadoop Distributed File System (HDFS).
Common Use: Used in the context of big data processing with Apache Hive.

Spark SQL:
Description: Part of the Apache Spark ecosystem, Spark SQL provides a programming interface for data manipulation using SQL-like queries. It allows users to query structured and semi-structured data.
Common Use: Used for querying data within Apache Spark applications.

KQL (Kusto Query Language):
Description: The query language used in Azure Data Explorer (Kusto), a service for exploring and analyzing large volumes of data quickly.
Common Use: Used for querying and analyzing data in Azure Data Explorer.

GraphQL:
Description: A query language for APIs that enables clients to request only the data they need. It provides a more flexible and efficient alternative to traditional REST APIs.
Common Use: Widely used in web development for client-server communication.

These languages serve different purposes, and the choice depends on the type of database or data warehouse, as well as the specific requirements of the application or analysis being performed.

SQL (Structured Query Language) and **NoSQL (Not Only SQL)** are two different types of database management systems, each with its own set of characteristics and use cases. Here are the key differences between SQL and NoSQL databases:

1. Data Model:
SQL: SQL databases are relational and use a structured schema, where data is organized into tables with predefined columns and relationships between tables.
NoSQL: NoSQL databases can be non-relational and offer various data models, including document-oriented, key-value pairs, wide-column stores, and graph databases. The schema is more flexible, allowing for dynamic and hierarchical data.

2. Schema:
SQL: SQL databases have a fixed schema, and any changes to the structure require altering the entire database.
NoSQL: NoSQL databases have a dynamic or schema-less structure, allowing for easy modification and adaptation to evolving data requirements.

3. Scalability:
SQL: Traditional SQL databases are typically scaled vertically by adding more resources (CPU, RAM) to a single server.
NoSQL: NoSQL databases are often designed for horizontal scalability, allowing for distributed and scalable architectures by adding more servers to a database cluster.

4. ACID Properties:
SQL: SQL databases adhere to ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data integrity and transactional consistency.
NoSQL: NoSQL databases may relax ACID properties for improved performance and scalability. They often prioritize eventual consistency and partition tolerance over immediate consistency.

5. Query Language:
SQL: SQL databases use SQL as the standard query language for defining and manipulating relational data.
NoSQL: NoSQL databases use various query languages or APIs, depending on the specific type of database. Examples include MongoDB's query language for document stores or Cassandra Query Language (CQL) for wide-column stores.

6. Use Cases:
SQL: Well-suited for applications with complex relationships and structured data, such as financial systems, ERP (Enterprise Resource Planning) systems, and traditional business applications.
NoSQL: Well-suited for scenarios with large volumes of unstructured or semi-structured data, such as content management systems, real-time big data applications, and IoT (Internet of Things) platforms.

7. Schema Evolution:
SQL: Changes to the schema can be complex and may require significant planning and downtime for migration.
NoSQL: NoSQL databases allow for dynamic and flexible schema evolution, making it easier to adapt to changing data requirements without downtime.

8. Examples:
SQL: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server.
NoSQL: MongoDB (document store), Cassandra (wide-column store), Redis (key-value store), Neo4j (graph database).

9. Consistency Model:
SQL: Emphasizes strong consistency, ensuring that transactions are executed in a manner that preserves the integrity of the data.
NoSQL: Offers various consistency models, including eventual consistency, where data consistency is guaranteed at some point in time but not necessarily immediately.

It's important to note that the choice between SQL and NoSQL depends on specific project requirements, scalability needs, data structure, and the nature of the application. Some projects may even use a combination of both types of databases to meet different needs within the same system (polyglot persistence).

Since most query languages are extensions of SQL, it’s useful to see how SQL works in its basic structure.

Let’s look at some specific examples of SQL code and what it does.

Database Creation:
SQL starts with creating a database using the CREATE DATABASE statement.

```
CREATE DATABASE MyDatabase;
```

Table Creation:
Tables hold the data in SQL databases. They are created using the CREATE TABLE statement.

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Age INT,
    Department VARCHAR(50)
);
```

Data Insertion:
Data is inserted into tables using the INSERT INTO statement.

```
INSERT INTO Employees (EmployeeID, FirstName, LastName, Age,
Department)
VALUES (1, 'John', 'Doe', 30, 'HR');
```

Data Retrieval (Simple Query):
Basic data retrieval is done using the SELECT statement.

```
SELECT FirstName, LastName FROM Employees;
```

Filtering Data (WHERE Clause):
Data can be filtered using the WHERE clause.

```
SELECT * FROM Employees WHERE Age > 25;
```

Sorting Data (ORDER BY Clause):
Sorting is achieved using the ORDER BY clause.

```
SELECT * FROM Employees ORDER BY LastName ASC;
```

Updating Data (UPDATE Statement):
Data can be updated using the UPDATE statement.

```
UPDATE Employees SET Department = 'Finance' WHERE EmployeeID = 1;
```

Deleting Data (DELETE Statement):
Data can be deleted using the DELETE statement.

```
DELETE FROM Employees WHERE Age < 30;
```

Joining Tables (INNER JOIN):
Tables can be joined to combine data using the JOIN clause.

```
SELECT Employees.EmployeeID, Employees.FirstName, Employees.LastName,
Departments.DepartmentName
FROM Employees
INNER JOIN Departments ON Employees.DepartmentID =
Departments.DepartmentID;
```

Other kinds of joins are also possible, such as left joins, right joins and outer joins.

Subqueries:
Subqueries are queries nested inside other queries.

```
SELECT FirstName, LastName
FROM Employees
WHERE DepartmentID IN (SELECT DepartmentID FROM Departments WHERE
DepartmentName = 'IT');
```

Examples:

Simple SELECT:

```
SELECT FirstName, LastName FROM Employees;
```

Filtering with WHERE:

```
SELECT * FROM Employees WHERE Age > 25;
```

Sorting with ORDER BY:

```
SELECT * FROM Employees ORDER BY LastName ASC;
```

Joining Tables:

```
SELECT Employees.EmployeeID, Employees.FirstName, Employees.LastName,
Departments.DepartmentName
FROM Employees
INNER JOIN Departments ON Employees.DepartmentID =
Departments.DepartmentID;
```

Grouping and Aggregation:

```
SELECT Department, AVG(Age) AS AverageAge, COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY Department;
```

Subqueries in WHERE Clause:

```
SELECT FirstName, LastName
FROM Employees
WHERE DepartmentID IN (SELECT DepartmentID FROM Departments WHERE
DepartmentName = 'IT');
```

Using HAVING Clause:

```
SELECT Department, AVG(Age) AS AverageAge
FROM Employees
GROUP BY Department
HAVING AVG(Age) > 30;
```

Window Functions (ROW_NUMBER):

```
SELECT FirstName, LastName, Age, ROW_NUMBER() OVER (PARTITION BY
Department ORDER BY Age DESC) AS Rank
FROM Employees;
```

Common Table Expressions (CTE):

```
WITH HighSalaryEmployees AS (
    SELECT FirstName, LastName, Salary
    FROM Employees
    WHERE Salary > 80000
)
SELECT * FROM HighSalaryEmployees;
```

Dynamic SQL with Stored Procedures:

```
CREATE PROCEDURE GetEmployeesByDepartment(IN departmentName
VARCHAR(50))
BEGIN
    SET @sql = 'SELECT * FROM Employees WHERE Department = ?';
    PREPARE stmt FROM @sql;
    EXECUTE stmt USING departmentName;
    DEALLOCATE PREPARE stmt;
END;
```

These examples showcase the progression from basic SQL operations to more advanced features, including joins, subqueries, aggregations, window functions, and stored procedures. As the complexity increases, SQL becomes a powerful tool for managing and analyzing relational databases.

Resources:

1. https://www.talend.com/resources/what-is-data-processing/
2. https://www.simplilearn.com/what-is-data-processing-article
3. https://www.referenceforbusiness.com/management/Comp-De/Data-Processing-and-Data-Management.html
4. https://support.microsoft.com/en-gb/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5
5. https://brewminate.com/a-brief-history-of-database-processing-since-the-1960s/
6. https://www.geeksforgeeks.org/types-of-databases/
7. https://www.matillion.com/blog/the-types-of-databases-with-examples
8. https://www.javatpoint.com/dbms-language
9. https://www.integrate.io/blog/the-sql-vs-nosql-difference/
10. https://www.mongodb.com/nosql-explained/nosql-vs-sql
11. https://www.ibm.com/blog/sql-vs-nosql/
12. https://www.geeksforgeeks.org/difference-between-sql-and-nosql/

Lecture 22

Data Generalization
Census as a use case

Data generalization in data mining is a process of summarizing or aggregating detailed data to produce more generalized or abstracted information while retaining the essential characteristics and trends of the original data. It's a common technique used for data anonymization, knowledge discovery, and reducing data complexity. Data generalization is often applied when dealing with sensitive or confidential information, such as personal data, to protect privacy. Here are some key aspects of data generalization:

Hierarchy of Data Abstraction: Data generalization typically involves the creation of a hierarchy of data abstractions. At the top of the hierarchy are highly abstracted and generalized data, while at the bottom are detailed, fine-grained data. Analysts can choose the level of abstraction that suits their analysis needs.

Examples of Data Generalization Techniques:
Attribute Generalization: This involves generalizing attributes or features of the data. For example, specific age values might be generalized into age groups or ranges (e.g., 20-30, 31-40, etc.).

Suppression: Some data points may be entirely suppressed or removed to protect sensitive information. For instance, if a dataset includes names and addresses, you might suppress the addresses to maintain privacy.

Rounding and Bucketing: Numeric values can be rounded to a certain precision or grouped into buckets or intervals. For example, precise income values could be rounded to the nearest thousand or grouped into income brackets.

Data Masking: Sensitive parts of data, such as parts of an email address or a phone number, can be replaced with symbols or random characters to conceal identity.

Categorization: Data can be categorized or labeled to reduce complexity. For example, a list of job titles can be grouped into categories like "Management," "Technical," "Administrative," etc.

Privacy Preservation: One of the primary purposes of data generalization is to preserve individual privacy while still making data available for analysis. By aggregating and abstracting data, it becomes more challenging to identify individuals or extract sensitive information.

Trade-off Between Privacy and Utility: Data generalization involves a trade-off between privacy and data utility. As data is more generalized, it provides better privacy protection but may lose some granularity that could be valuable for analysis. Striking the right balance is essential.

Data Mining and Knowledge Discovery: Generalized data can be used in data mining and knowledge discovery tasks. By working with abstracted data, patterns, trends, and insights can be extracted without revealing sensitive details. This is particularly important in fields like healthcare, finance, and customer analytics.

Regulatory Compliance: Many data privacy regulations, such as GDPR in Europe, require organizations to implement data generalization techniques to protect individuals' data privacy and ensure compliance with data protection laws.

Data generalization is a crucial technique for managing and protecting sensitive data while enabling data analytics. It plays a significant role in ensuring that data can be used for research, analysis, and business insights without compromising individuals' privacy or confidentiality.

The hierarchy of data abstraction is a structured way of organizing data at various levels of detail, starting from the most detailed or fine-grained data at the bottom of the hierarchy and moving to more generalized or abstracted data at the top. This hierarchy allows data to be presented at different levels of granularity and provides users or analysts with the flexibility to choose the level of abstraction that best suits their needs. Here's more information about the hierarchy of data abstraction:

Fine-Grained Data (Bottom Level): At the lowest level of the hierarchy, you have fine-grained or detailed data. This level includes raw and unprocessed data with all attributes and values intact. Fine-grained data is the most specific and retains the highest level of detail, making it suitable for granular analysis and individual-level insights.

Attribute Generalization: Moving up the hierarchy, attribute generalization involves abstracting or summarizing specific attributes within the data. For example, numeric values might be aggregated into ranges, and text data might be categorized into broader groups. Attribute generalization is a way of simplifying data while preserving its essential characteristics. It can help reduce complexity and make data more manageable.

Data Reduction and Aggregation: The next level involves data reduction and aggregation, where data is summarized or aggregated based on certain criteria or dimensions. This can include summing, averaging, or counting data based on specific attributes or time periods. Aggregated data is more abstract than attribute-generalized data and is often used for higher-level trend analysis and reporting.

Categorization and Labeling: Further up the hierarchy, data may be categorized and labeled to group similar data points together. This is often done to create higher-level labels that represent broader concepts. Categorization allows for the simplification of data, making it easier to identify trends and patterns in the data.

High-Level Summaries (Top Level): At the top of the hierarchy are high-level summaries. These are the most abstract representations of data and are typically used for executive-level reporting and decision-making. High-level summaries provide a broad overview of the data and are particularly useful when the audience is not interested in detailed data but seeks key insights.

The choice of where to position data within the hierarchy of data abstraction depends on the specific goals of the analysis and the level of detail required. As data is moved up the hierarchy, it becomes more generalized, which can help protect privacy and simplify data analysis. However, there is a trade-off between privacy and data utility; highly abstracted data may lose some granularity and specific details that might be valuable for certain analyses.

The hierarchy of data abstraction is a valuable concept in fields like data anonymization, data mining, and business intelligence, where balancing the need for privacy and the need for valuable insights is crucial. It allows data analysts to tailor data to the needs of different audiences and use cases, all while maintaining the integrity of the underlying data.

The United States Census Bureau conducts the decennial census, a population count that occurs every ten years. The Census Bureau also conducts various other surveys and programs to collect demographic, economic, and social data. Here is an overview of how the Census collects and uses data:

1. Decennial Census: The primary function of the decennial census is to count every person living in the United States. The process involves several key steps:

Preparation: The Census Bureau spends years planning and preparing for the decennial census, including designing questionnaires, establishing enumeration methods, and conducting field tests.

Questionnaire Distribution: The Census Bureau mails or delivers questionnaires to households across the country. In recent years, the Census has promoted online responses as an option.

Self-Response: Residents are encouraged to complete and return their census forms on their own. This can be done by mail, phone, or online.

Non-Response Follow-Up: For households that do not respond, the Census Bureau conducts non-response follow-up (NRFU) by sending enumerators to visit households in person to collect the necessary information.

Data Processing: The collected data undergoes extensive processing to ensure accuracy. This includes data cleaning, coding, and imputation for missing information.

Data Release: The final aggregated and anonymized data is released to the public. This data is used for apportionment (determining the number of seats each state gets in the House of Representatives) and redistricting.

2. American Community Survey (ACS): The American Community Survey is an ongoing survey conducted by the Census Bureau, providing more detailed and frequent demographic, economic, social, and housing information. Unlike the decennial census, the ACS is conducted annually, collecting data from a sample of households. Key steps include:

Sampling: The ACS surveys a sample of households throughout the year, collecting information on various topics.

Data Release: ACS data is released annually, providing timely and detailed information about communities.

Long-Form Replacement: The ACS replaced the traditional long-form census questionnaire that used to be part of the decennial census. It provides more up-to-date and comprehensive data.

3. Other Census Bureau Programs:
Economic Census: The Census Bureau conducts economic censuses to collect data on businesses, including information on employment, revenue, and industry characteristics.

Housing Surveys: Various housing surveys provide information on housing characteristics, homeownership rates, and rental housing.

Population Estimates: The Census Bureau produces annual population estimates between decennial censuses, helping to track population changes.

Use of Census Data: The data collected by the Census Bureau is used for a variety of purposes:

Representation: Census data is used to determine the distribution of seats in the U.S. House of Representatives among the states through a process called apportionment.

Redistricting: Census data is used in the redistricting process, helping to redraw legislative district boundaries at various levels of government.

Funding Allocation: Census data is used to allocate federal funds to states and localities for various programs and services, including education, transportation, and healthcare.

Policy and Planning: Government agencies, researchers, businesses, and nonprofits use census data for policy formulation, planning, and decision-making.

Demographic Analysis: Census data provides valuable insights into the demographic composition of the population, helping to understand trends and changes over time.

It's important to note that the Census Bureau takes privacy seriously and has strict confidentiality protections in place. Individual responses are confidential by law, and the Census Bureau works to ensure that data is aggregated and anonymized before release. The goal is to collect accurate and comprehensive data to inform public policy and ensure fair representation and distribution of resources.

To protect privacy while ensuring the usefulness of the collected data, the U.S. Census Bureau employs various techniques, including data generalization, to safeguard individual responses. Here are some of the methods used to protect privacy in census data:

1. Data Aggregation and Generalization:
Aggregated Reporting: Census data is often reported in aggregated form, presenting statistics at higher levels, such as state, county, or census tract. This helps in preventing the identification of individuals.

Generalization of Geographic Data: Geographic data, such as exact addresses, is often generalized to a higher level of granularity, like census blocks, to protect the identity of specific households.

Rounding and Perturbation: Numerical values in the data are often rounded or perturbed (adding small random noise) to prevent the exact identification of individuals or households.

2. Disclosure Avoidance:
Statistical Disclosure Limitation Techniques: The Census Bureau employs statistical disclosure limitation (SDL) techniques to ensure that individual responses cannot be easily deduced from the released data. This involves introducing noise, suppression, or other modifications to the data.

Differential Privacy: In recent years, the Census Bureau has adopted differential privacy, a rigorous mathematical framework, to protect individual privacy. Differential privacy adds controlled noise to the data to prevent the re-identification of respondents.

Topcoding: In some cases, extreme values (outliers) may be replaced with a maximum threshold, a process known as topcoding, to protect the identity of individuals with unusual characteristics.

3. Aggregation of Demographic Characteristics:
Grouping Demographic Characteristics: Demographic characteristics, such as age, may be grouped into broader categories to reduce the risk of identification.

Combining Data: Data on specific characteristics may be combined to present a more generalized picture, reducing the granularity of information.

4. Temporal Methods:
Data Suppression: In some cases, the Census Bureau may suppress or limit the release of data for certain small populations or geographic areas to prevent potential identification.

Changing Data Over Time: By changing or perturbing data over time, the Census Bureau adds an additional layer of protection against re-identification.

5. Legal and Ethical Safeguards:
Confidentiality Laws: The Census Bureau is bound by strict confidentiality laws, such as Title 13 of the United States Code, which protects the privacy of individual responses. Census Bureau employees face severe penalties for unauthorized disclosure of information.
Data Stewardship and Ethics: The Census Bureau follows ethical guidelines and practices for data stewardship, ensuring that privacy is a top priority in all data-handling processes.

6. Public Education and Communication:
Promoting Privacy Awareness: The Census Bureau engages in public education campaigns to raise awareness about the importance of responding to the census while assuring individuals that their data is protected.
Transparency: The Census Bureau is transparent about its data protection methods and communicates the steps taken to safeguard individual privacy.

It's important to note that the Census Bureau continuously reviews and updates its privacy protection methods to stay ahead of evolving privacy challenges. The goal is to strike a balance between data utility and individual privacy, ensuring that census data remains accurate, relevant, and confidential.

Access to decennial census data and American Community Survey (ACS) data is provided by the U.S. Census Bureau through various platforms and tools. Here are common ways to access census data:

1. U.S. Census Bureau Website: The official website of the U.S. Census Bureau is a primary source for accessing census data. The website provides access to a wide range of demographic, economic, and social data. Key sections include:

Data Tools and Apps: The Census Bureau offers interactive tools and applications that allow users to explore and visualize data. Examples include the American FactFinder, Data.census.gov, and the Census Bureau API.

Data Releases and Publications: Data releases, reports, and publications are available on the website, providing in-depth analyses and insights into various demographic and economic topics.

Guidance and Documentation: The website includes guidance documents, methodology information, and technical documentation that help users understand the data and its limitations.

Specialized Surveys: In addition to the decennial census and ACS, the Census Bureau conducts various specialized surveys, and data from these surveys are often available on the website.

2. Data.census.gov: Data.census.gov is a platform designed for accessing and exploring U.S. Census Bureau data. It provides a user-friendly interface for finding, downloading, and visualizing data from the decennial census, ACS, and other surveys. Users can search for specific data sets, create custom tables, and download data in various formats.

3. Census Bureau API: The Census Bureau provides an Application Programming Interface (API) that allows developers to programmatically access census data. The API provides a way to retrieve data in a structured format, making it suitable for integration into custom applications or analysis tools.

4. Public Use Microdata Sample (PUMS): The Census Bureau releases Public Use Microdata Sample (PUMS) files, which provide individual-level or household-level data while protecting individual privacy. PUMS files are available for the decennial census and the ACS.
Researchers and analysts can use PUMS files to conduct more detailed analyses and customized tabulations.

5. Local Libraries and Universities: Many local libraries, universities, and research institutions provide access to census data. Libraries may offer assistance in accessing and navigating census databases, and universities often have data services or research centers that can provide support.

6. Federal Statistical Research Data Centers (FSRDCs): The Census Bureau collaborates with the Federal Statistical Research Data Centers, which are secure research facilities that provide approved researchers with access to restricted-use microdata. Researchers can apply for access to these centers to conduct more detailed analyses.

7. Subscription Services: Some third-party organizations or subscription services provide advanced tools and interfaces for accessing and analyzing census data. These services may offer additional features and support for users with specific data needs.

It's important to note that the availability of data may vary depending on the specific survey, dataset, and geographic level. Users should familiarize themselves with the documentation and methodologies provided by the Census Bureau to ensure proper use and interpretation of the data.

Resources:

1. https://www.census.gov/data.html
2. https://www.immuta.com/blog/what-is-data-generalization/
3. https://satoricyber.com/data-masking/data-generalization/
4. https://www.geeksforgeeks.org/basic-approaches-for-data-generalization-dwdm/
5. https://www.privitar.com/blog/data-generalization-advanced-de-identification/
6. https://www.tutorialspoint.com/basic-approaches-for-data-generalization-dwdm
7. https://www.datamining365.com/2020/02/data-generalization-summarization.html