KNOWLEDGE & DEVELOPMENT SOLUTIONS

Data Bases and Data Types

DL Day 3 class

CL | Corp | IT | PL

Learning Objectives

By the end of this lesson, you will be able to:

- **Define the concept of a database and how it is used to store data**:
  - Understand the fundamental concept of a database as an organized data storage system.
- **Identify the core components of a relational database table**:
  - Recognize the key elements that make up a relational database table.
- **Explain basic database terminology like rows, columns, tables, and relationships**:
  - Familiarize yourself with essential database terminology, including rows, columns, tables, and relationships.
- **Define data types**:
  - Understand the concept of data types and their role in defining the kind of data that can be stored in a database.

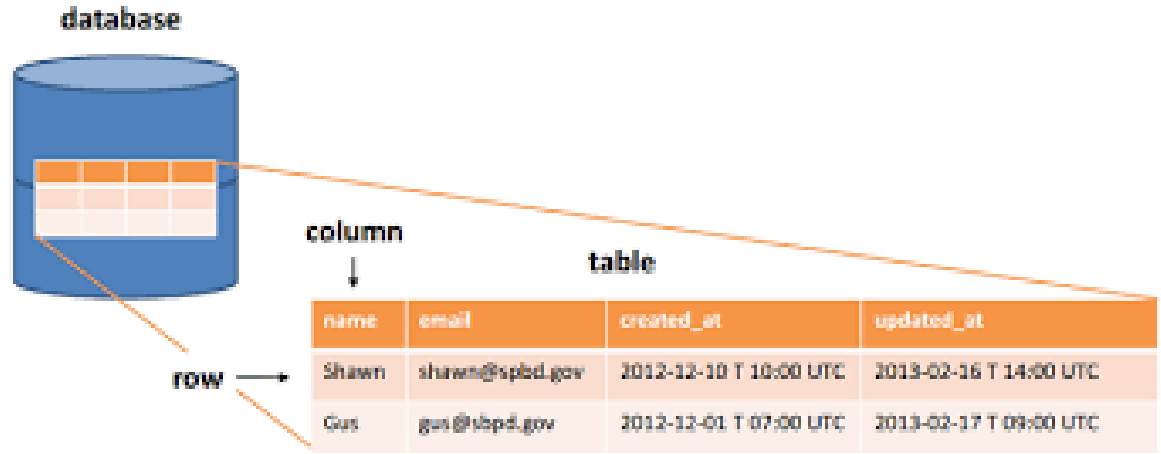
Learning Objectives

- **Database Skills for Analysts**:
  - Enable analysts to:
    - Retrieve, combine, and prepare data for analytical projects, extracting valuable insights.
    - Identify connections between datasets to enrich analysis.
    - Empower self-service analytics.

- **Strong Database Familiarity**:
  - Allows analysts to gather, investigate, and prepare data with flexibility and precision.

Databases

- **Database Definition**:
  - A database is an organized collection of data stored and accessed electronically from a computer system.
- **Database Benefits**:
  - Databases allow users to easily store, update, retrieve, and manage large amounts of information.
  - Data is structured and organized for efficient use.



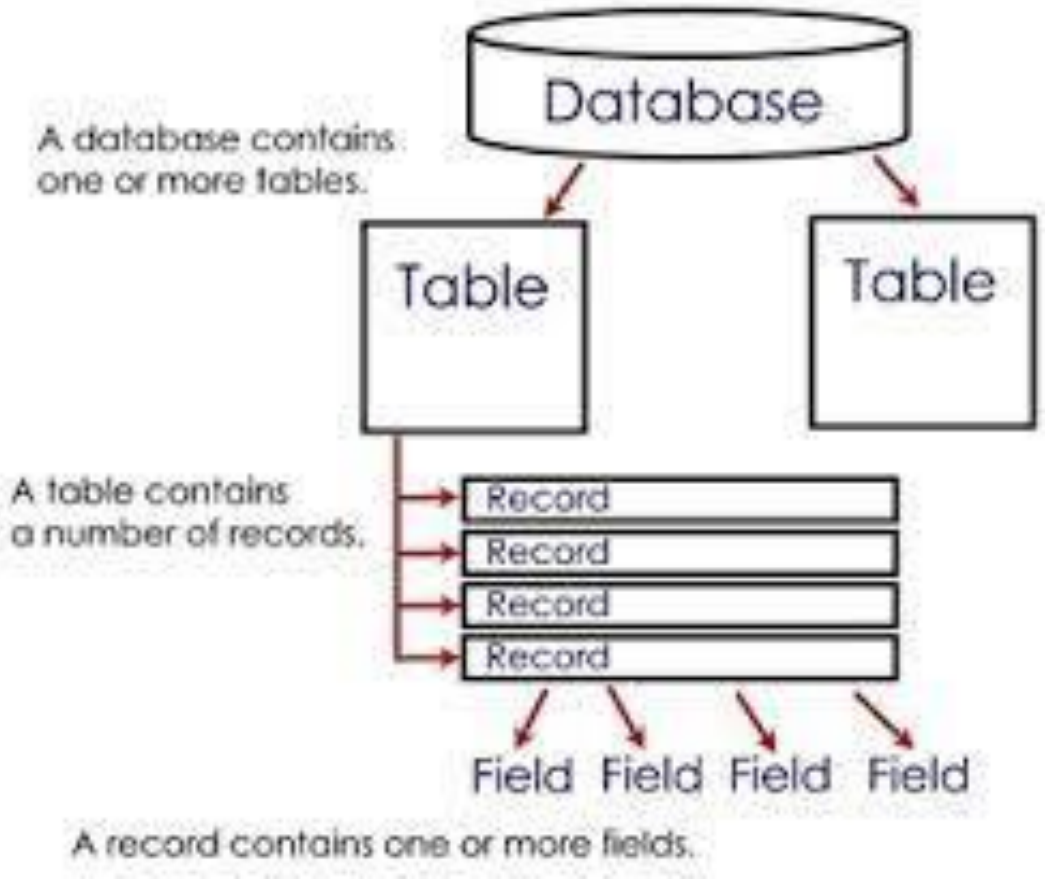
Databases

- **Database Management Systems (DBMS):** DBMS is software used for various database operations.
  - It includes creating, accessing, managing, searching, and analyzing data.
  - Popular DBMS options include MySQL, Oracle, Microsoft SQL Server, and MongoDB.

Databases

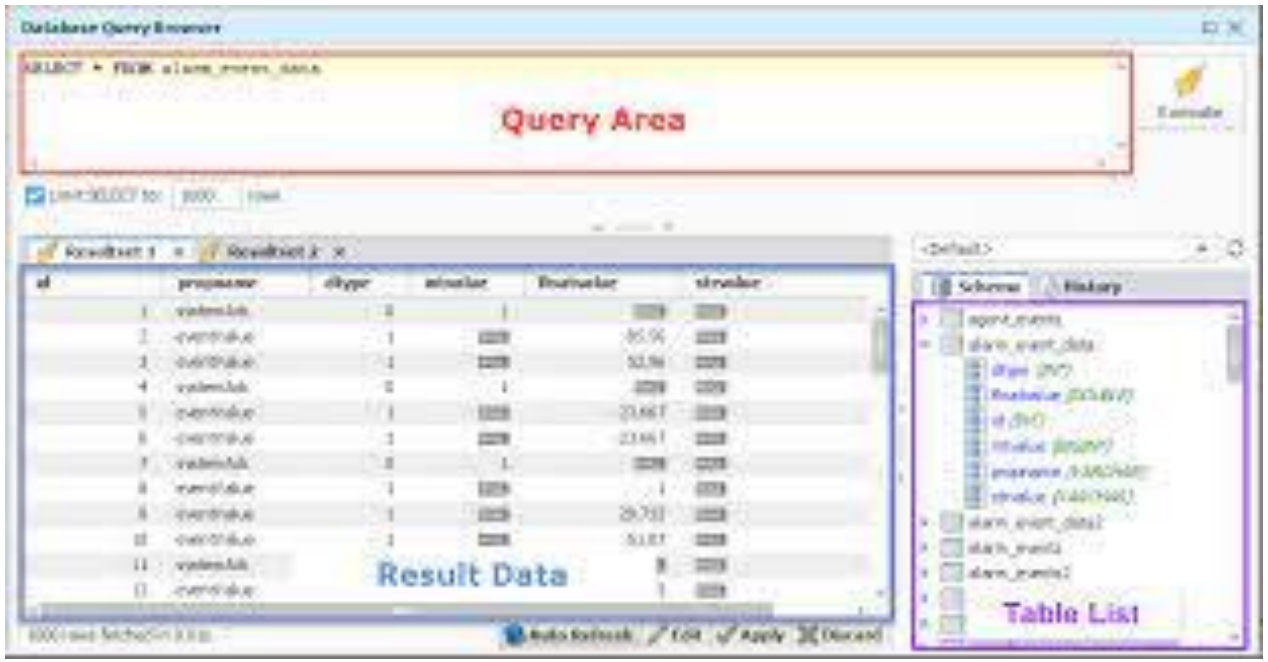
- **Tables**:
  - Databases organize information into tables.
  - Tables store data in rows and columns.
    - For instance, a "Customers" table might include columns like "Name," "Address," and "Phone Number," with each row representing an individual customer.
- **Records**:
  - Also known as rows.
    - Each record is an individual entry stored in a table.
    - It collects related data across different fields.
- **Fields**:
  - The columns in a database table are called fields or attributes.
    - Fields represent specific characteristics or attributes of the larger entries, such as a name or date.

Databases


Database
A database contains one or more tables.
Table
Table
A table contains a number of records.
Record
Record
Record
Record
Field
Field
Field
Field
A record contains one or more fields.

Databases

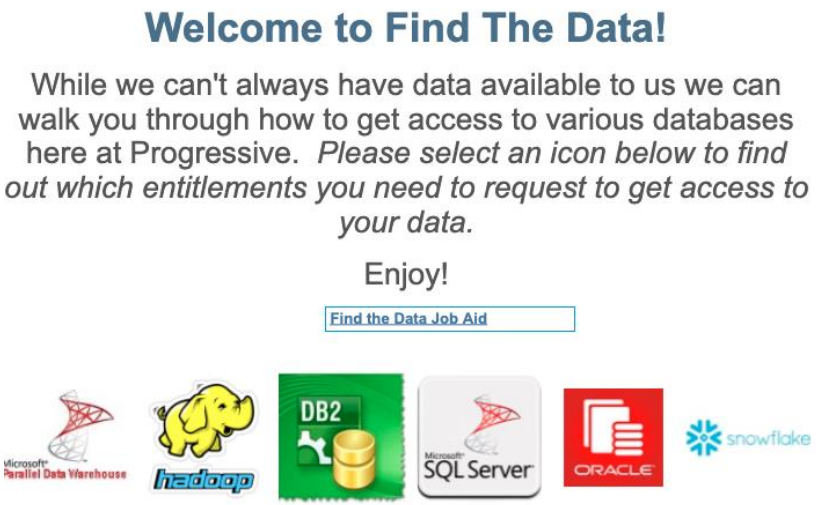
- **Queries:** A database query is a request for specific information from a database. Allows for powerful analysis and filtering.



Databases

- Databases At Progressive: Findthedata/

https://tableauserver/t/PersonalLines/views/FindTheData/Home?%3Aembed=y&%3AshowShareOptions=true&%3Adisplay_count=no&%3AshowVizHome=no#1



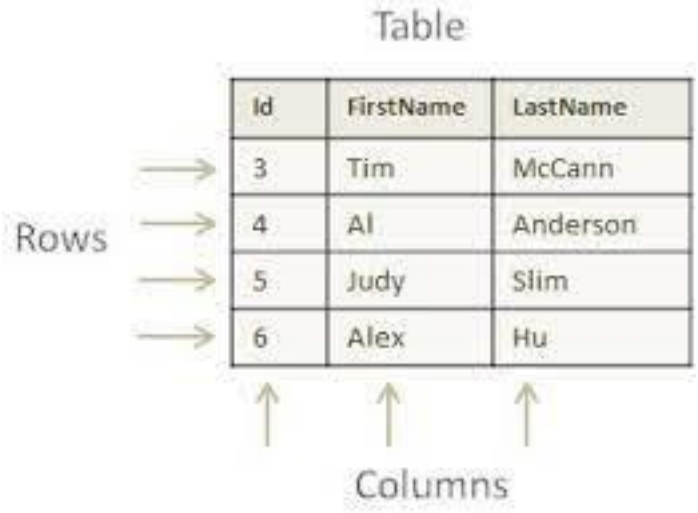
Relational Databases

- **Relational Databases**: Organize structured data into tables for easy management, access, manipulation, and SQL-based linking.

- **Advantages**:
  - **Flexibility**: Easily add new data categories, fields, and tables without altering existing structures.
  - **Simplicity**: Streamlined data organization for improved efficiency.
  - **Performance**: Efficient data retrieval and processing.

Relational Databases

- **Data Structure**:
  - Structured, tabular data storage.
  - Utilizes two-dimensional database tables.
  - Tables mimic spreadsheets with cell-based layouts.
- **Data Representation**:
  - Each **row** represents a unique record.
  - **Columns** define various fields or attributes for record types.

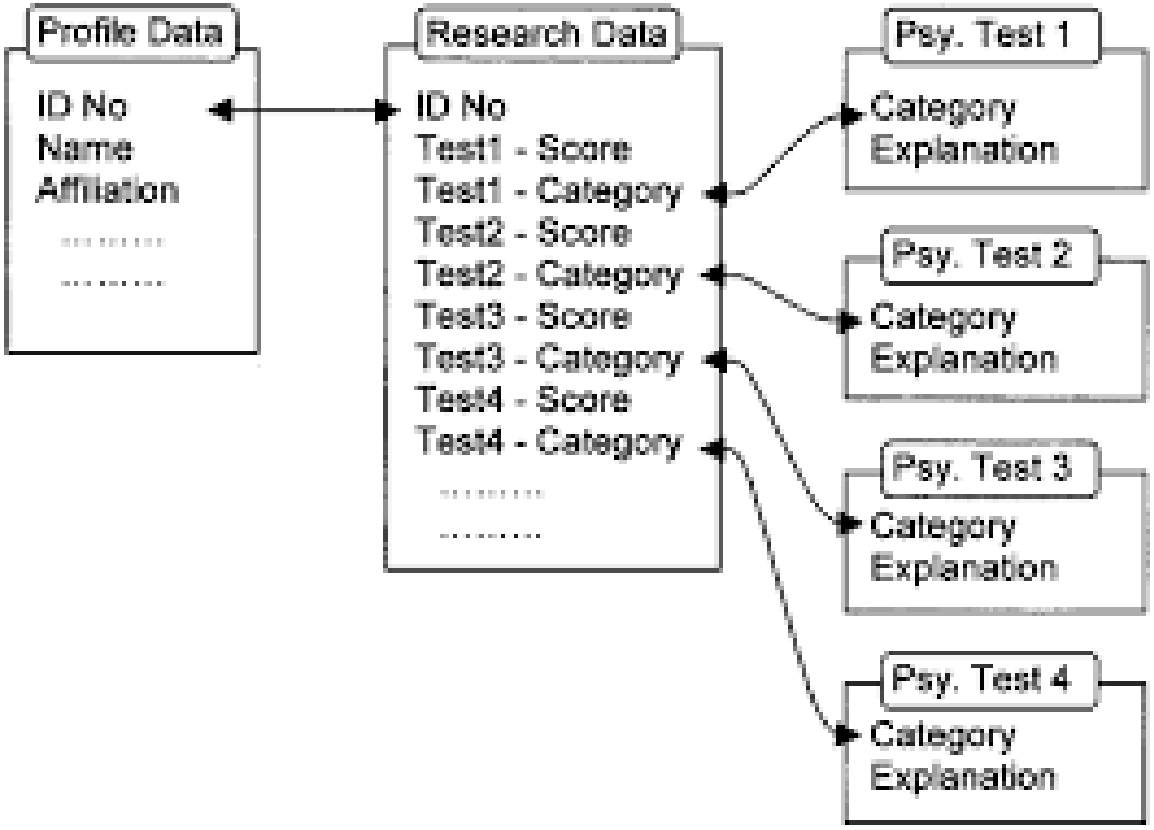


Relational Databases

- **Data Organization**:
  - Tables organize data into fields and records.
  - Fields (columns) contain attributes (e.g., name, age) of a record.
  - Records (rows) represent individual data entries (e.g., customer, product).

- **Structured Relationships**:
  - Tables are interconnected.
  - Structured relationships connect records across tables.

Relational Databases



Relational Databases

- **Key Constraints**:
  - Define essential table relationships.
  - Ensure data accuracy and consistency.
- **Primary Key**:
  - Uniquely identifies each record in a table.
  - Guarantees record uniqueness within the table.
- **Foreign Keys**:
  - Establish links between records in different tables.
  - Create relationships between related data.

Relational Databases

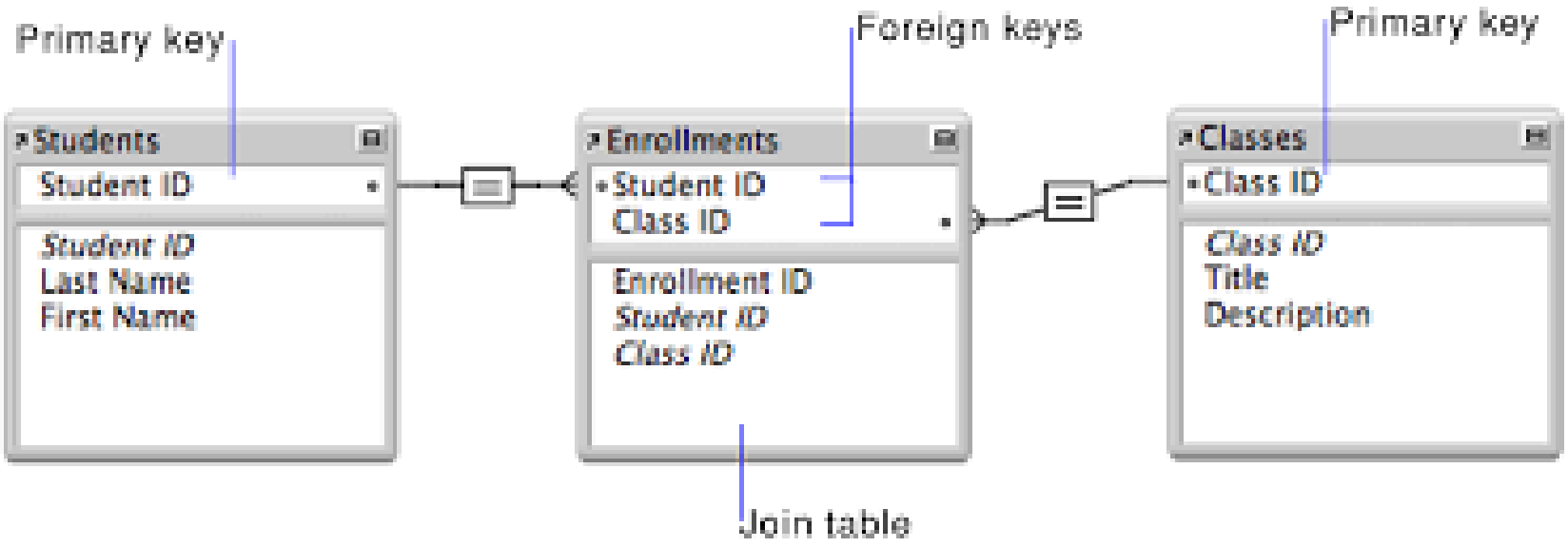


Table Data Structures

- **Structured Format**:
  - Database tables offer an organized, tabular structure.
  - Ensures efficient and consistent data storage.

- **Data Types**:
  - Crucial aspect of table structure.
  - Define allowable data for each column.
  - Constraints ensure data adheres to predefined formats.

Table Data Structures

- **Data Types Overview**:
  - Data types define the kind of data that can be stored in a database.
  - Examples of common data types:
    - **TEXT**: Allows letters, numbers, and special characters.
    - **NUMBER or INTEGER**: Requires numerical values.
    - **DATE**: Ensures properly formatted date values.

- **Data Type Enforcement**:
  - Designating an AGE column to accept only integers enforces data consistency.
  - Each data type serves a specific purpose in database design and integrity.

Table Data Structures

- To understand SQL Server data types, let's look at the following page to create a new Google account (for reference purpose only):

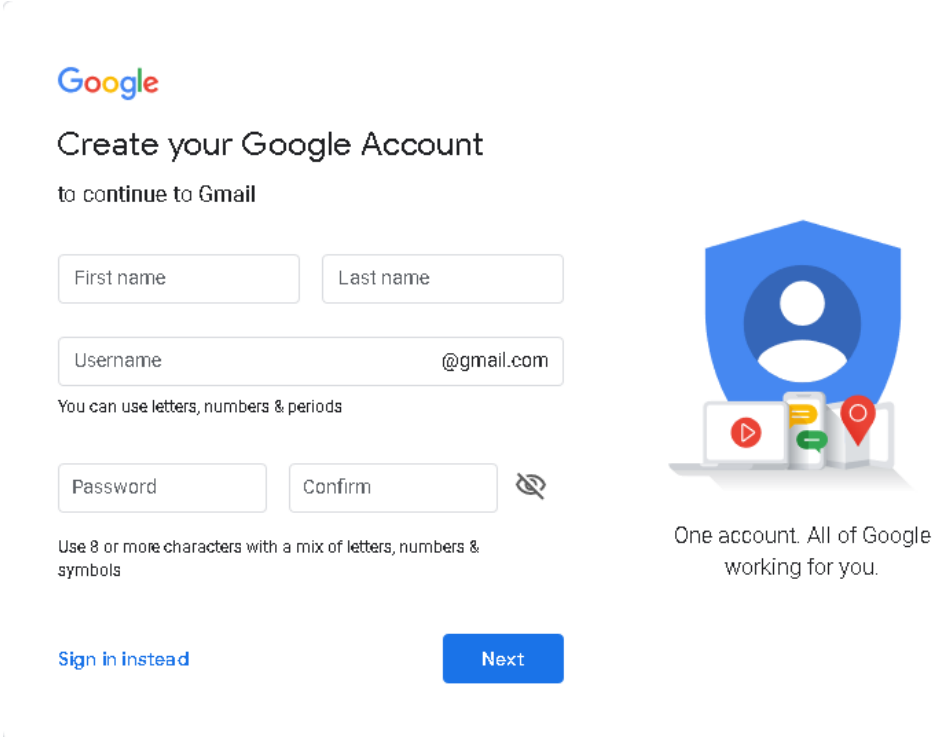


Table Data Structures

- **Input Requirements**:
  - First Name and Last Name: Alphabets only.
  - Username: Letters, numbers, and periods.
  - Password: Alphabets, numbers, and special characters.

- **Data Storage**:
  - Store user data in a Database Server.
  - Utilize various data types to match specific requirements.

Table Data Structures

- Categories of Databases Data Types

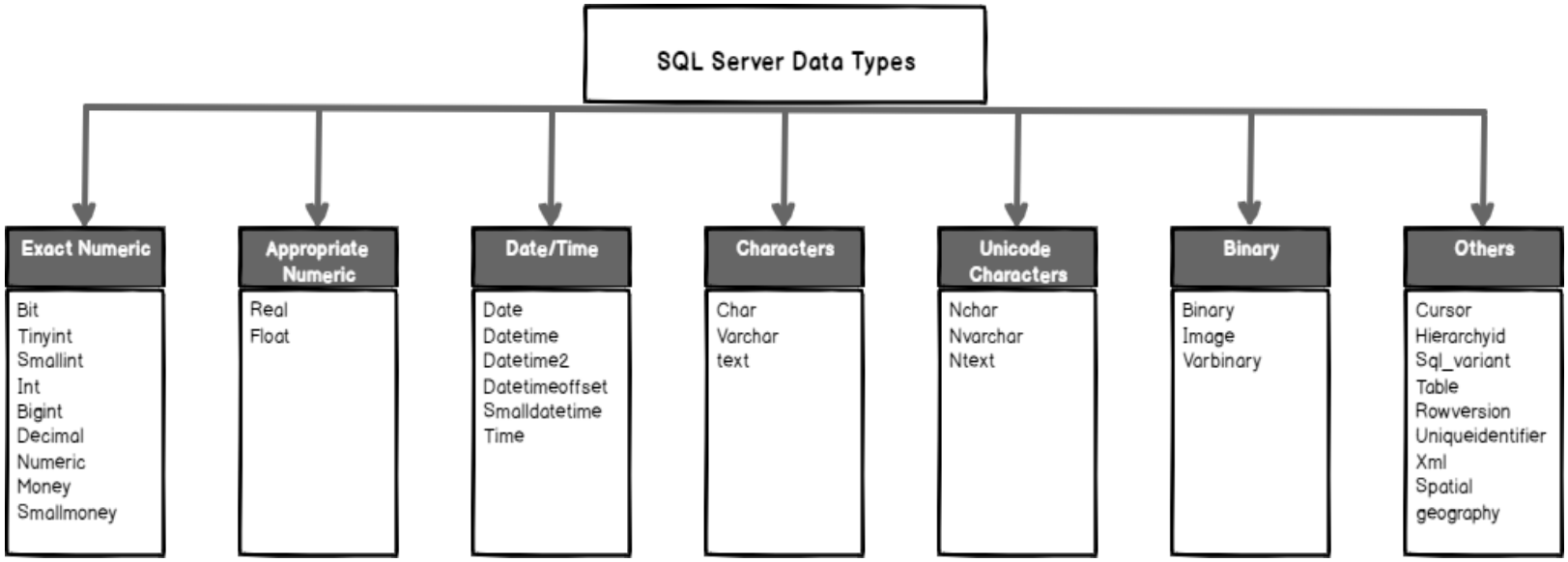
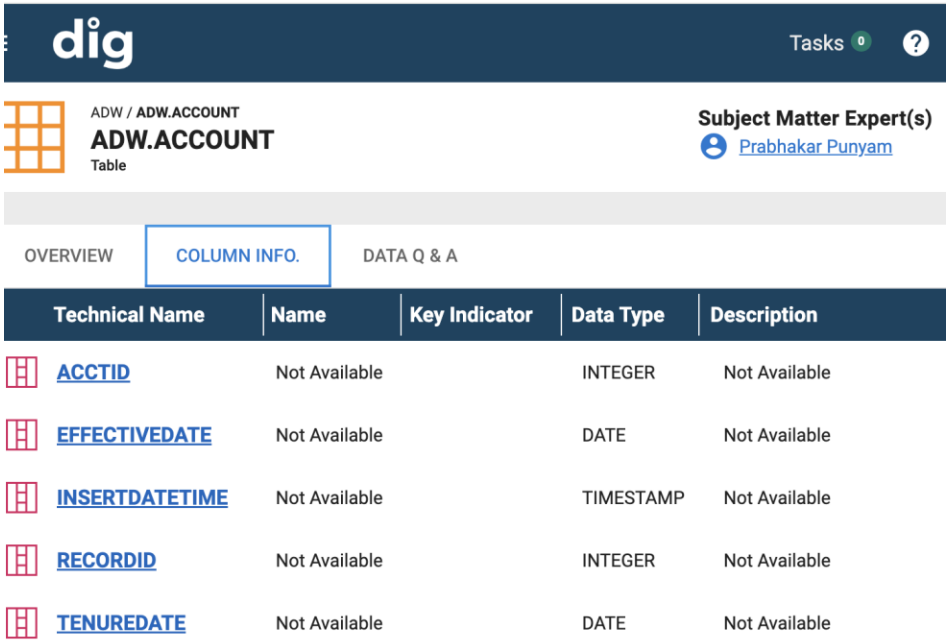


Table Data Structures

- **Exact Numeric**:
  - Used for precise numeric values without rounding errors.
- **Approximate Numeric**:
  - Suitable for numeric values where precision is not critical.
- **Date and Time**:
  - Designed for storing date and time information.
- **Character Strings**:
  - Used for storing text data, with varying lengths.
- **Unicode Character Strings**:
  - Ideal for storing multilingual or special character text.

Table Data Structures

- Using DIG to begin your data journey
  - Type DIG/ into your browser
  - take a look at some of the databases
    - Review the tables and columns
    - Identify the types



Query

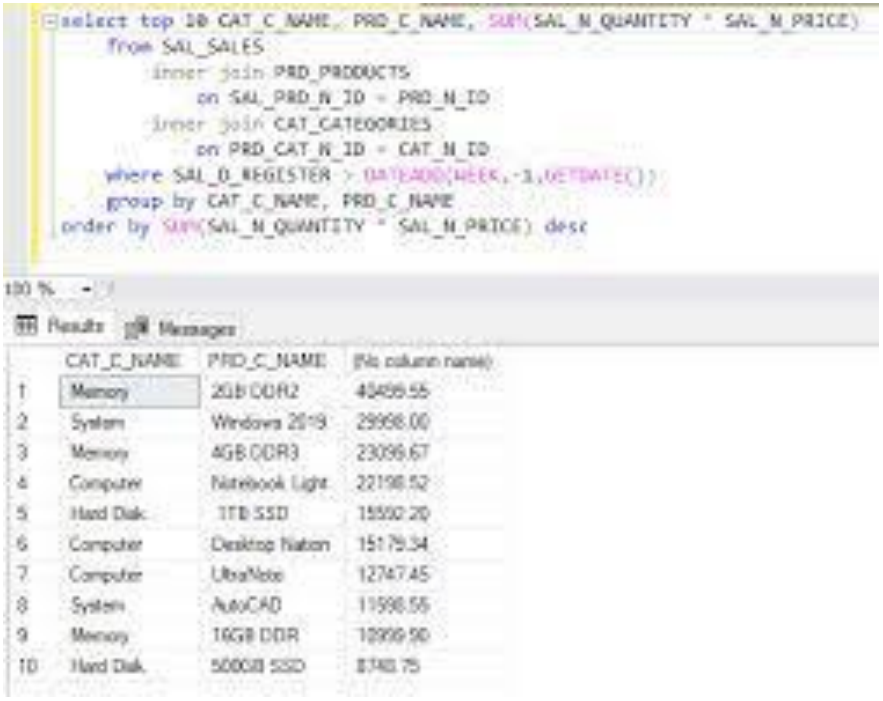
- A query is a request made to a database to retrieve specific data that meets certain criteria.
- Some key points about database queries:
  - A query allows selective retrieval of data from a database rather than retrieving all records at once.
  - Queries are constructed using query languages like SQL with syntax specifying what data to retrieve and filter on.
  - The most common SQL queries are SELECT statements which define the table columns to show and the conditions rows must meet using WHERE clauses.

Query

- Query criteria can filter on text matches, numeric & date ranges, associations between tables to derive relevant information.
- Queries help answer specific business questions like "What products had the most sales last month?" without needing manual analysis.
- Results can be sorted, aggregated (summaries), and segmented as needed for reporting or analysis.
- Parameters can be used to make dynamic queries that change based on user inputs.

Query

- In summary, database queries allow users to selectively retrieve the most relevant data for their needs from among large datasets in an efficient manner.
  - They underlie analytics, business intelligence and many database applications.



```
select top 10 CAT_C_NAME, PRD_C_NAME, SUM(SAL_N_QUANTITY * SAL_N_PRICE)
    from SAL_SALES
        inner join PRD_PRODUCTS
            on SAL_PRD_N_ID = PRD_N_ID
        inner join CAT_CATEGORIES
            on PRD_CAT_N_ID = CAT_N_ID
    where SAL_D_REGISTER > DATEADD(WEEK,-1,GETDATE())
    group by CAT_C_NAME, PRD_C_NAME
order by SUM(SAL_N_QUANTITY * SAL_N_PRICE) desc
```

100 %

Results | Messages

	CAT_C_NAME	PRD_C_NAME	(No column name)
1	Memory	2GB DDR2	49499.55
2	System	Windows 2019	29968.00
3	Memory	4GB DDR3	23096.67
4	Computer	Notebook Light	22198.52
5	Hard Disk	1TB SSD	15500.20
6	Computer	Desktop Nation	15179.34
7	Computer	UltraNote	12747.45
8	System	AutoCAD	11998.55
9	Memory	16GB DDR	10999.50
10	Hard Disk	500GB SSD	8749.75

Summary

- Key Points
  1. Databases provide electronic storage and data management via optimized table structures and sophisticated database management systems for reliably scaling data.
  2. Relational databases organize data into tables with formal connections between them to model real-world entities and relationships. This provides simplicity and performance.
  3. Tables consisting of rows, columns and keys give a standardized structure combined with the flexibility to adapt for future needs through constraints, normalizations etc.
  4. SQL queries allow selectively filtering and aggregating data to derive insights efficiently without manual effort even from billions of rows due to the deliberate underlying database designs.

Summary

From storing to accessing data, databases emphasize formalized tables and queries as specialized structures engineered for the era of data-driven decisions. This model balances optimal speed, scale and integrity demands allowing modern applications to leverage data systematically

Exercises

- Entity Relationship Diagram Design
  - Break into small teams. Give each a business scenario and have them draw out entities, attributes, relationships to model database schema through an ERD.
- Normalization Practice
  - Provide sample non-normalized data set to teams. Have them normalize it to appropriate consistency form.
- Query Formulation
  - Offer some sample problem statements/business questions. Teams write SQL queries to retrieve requested data. Compare different query strategies.
- Database Optimization
  - Set up simple database environment. Have teams identify bottlenecks like long read/writes, redundant/inconsistent data based on usage. Perform corrections.

Exercises

- Use Case Conceptualization
  - Teams come up with ideas to leverage databases for various scenarios: ecommerce, banking, supply-chain etc. Present possibilities and limitations.
- Database Administration Exercise
  - Give sandbox access to practice tasks like user provisioning, backup, restoration, high availability.

- The core themes would be strengthening both conceptual (ERD, use cases) and technical (queries, optimization) hands-on skills while encouraging collaborative thinking for analyzing needs, problem resolution and evaluation of tradeoffs. Tie learning to real-world applicability.