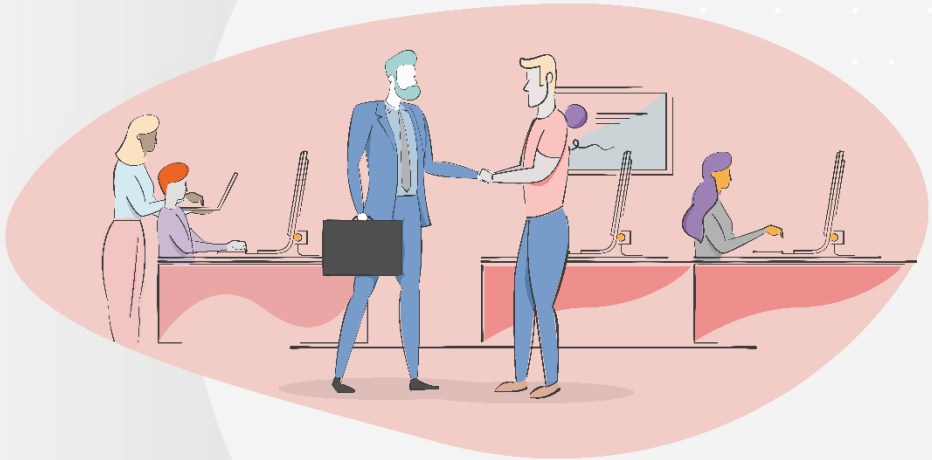
Storing and querying data



L5 Data Engineer Higher Apprenticeship
Module 2 / 12 (“**Databases and Data Lakes**”)
Topic 1 / 5

rev. 1 (2024)

Webinar agenda

This webinar will cover the following:

- Data storage options
- Files systems and RDBS data
- Implementing cloud storage

Webinar length: 3 hours

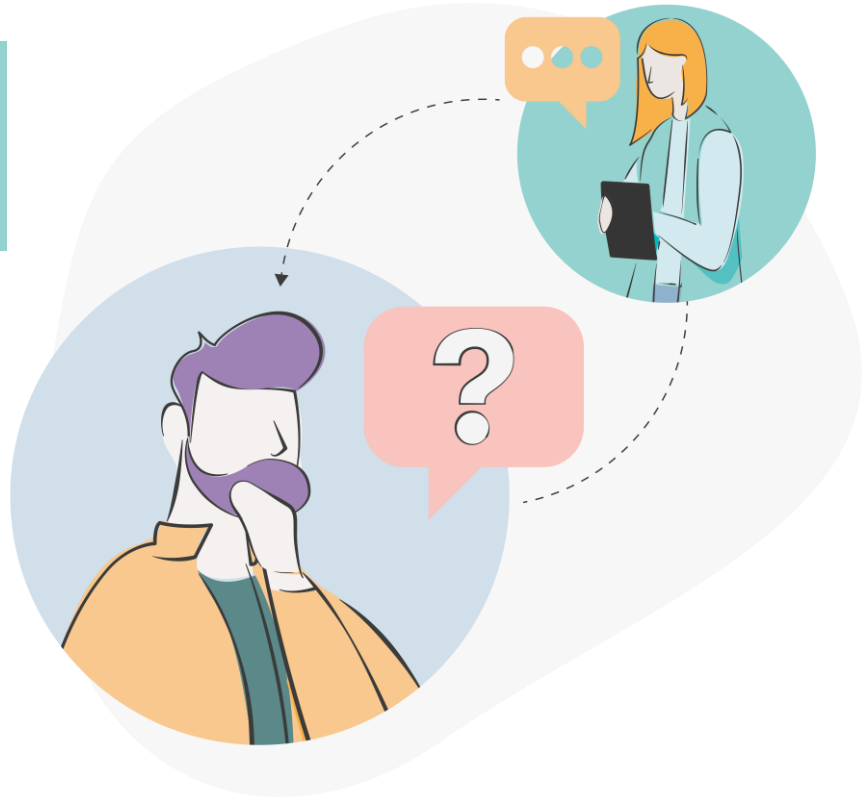
Poll

Building Careers Through Education

Everybody Matters. Stronger Together. Student, Learner and Client Centric. Trust and Respect. Embrace Change.

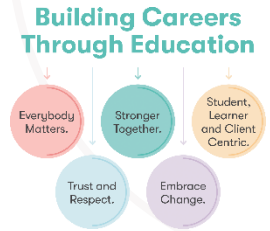
Which technologies do you use for data storage at work?

Submit your responses to the chat!



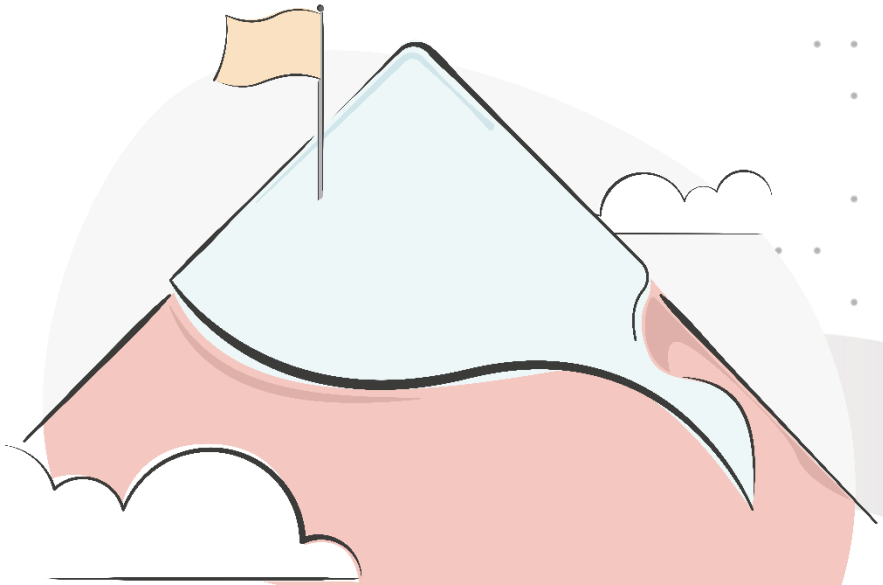
BPP

Learning objectives



By the end of today's webinar, you will be able to:

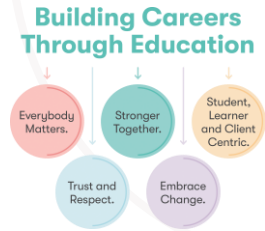
- **Demonstrate Knowledge of Data Storage Options**: You will be able to compare and contrast various data storage options such as local files and cloud storage
- **Preview Filesystem and RDBS data**: You will demonstrate their ability to list and query data located in two different sources, the filesystem, and the relational database system
- **Implement Cloud-Based Storage**: You will use AWS to store S3 data and perform data queries using SQL



Sounds like a lot? Don't worry! We will provide real-world examples for each of the key concepts that you learn about today.

BPP

e-Learning recap



Use the chat to answer the following themes that you explored in your e-learning before the session

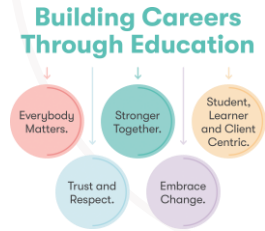
Why are businesses not throwing away data?

Storing files

On-premise pros and cons

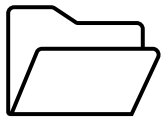
BPP

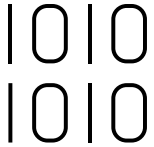
Data storage options

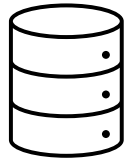


These are the most common options for storing data in an organisation:

Filesystems	RDBMS (SQL)	Non-relational databases (NoSQL)
• Text data • Binary data	• Local • Remote	• Cloud

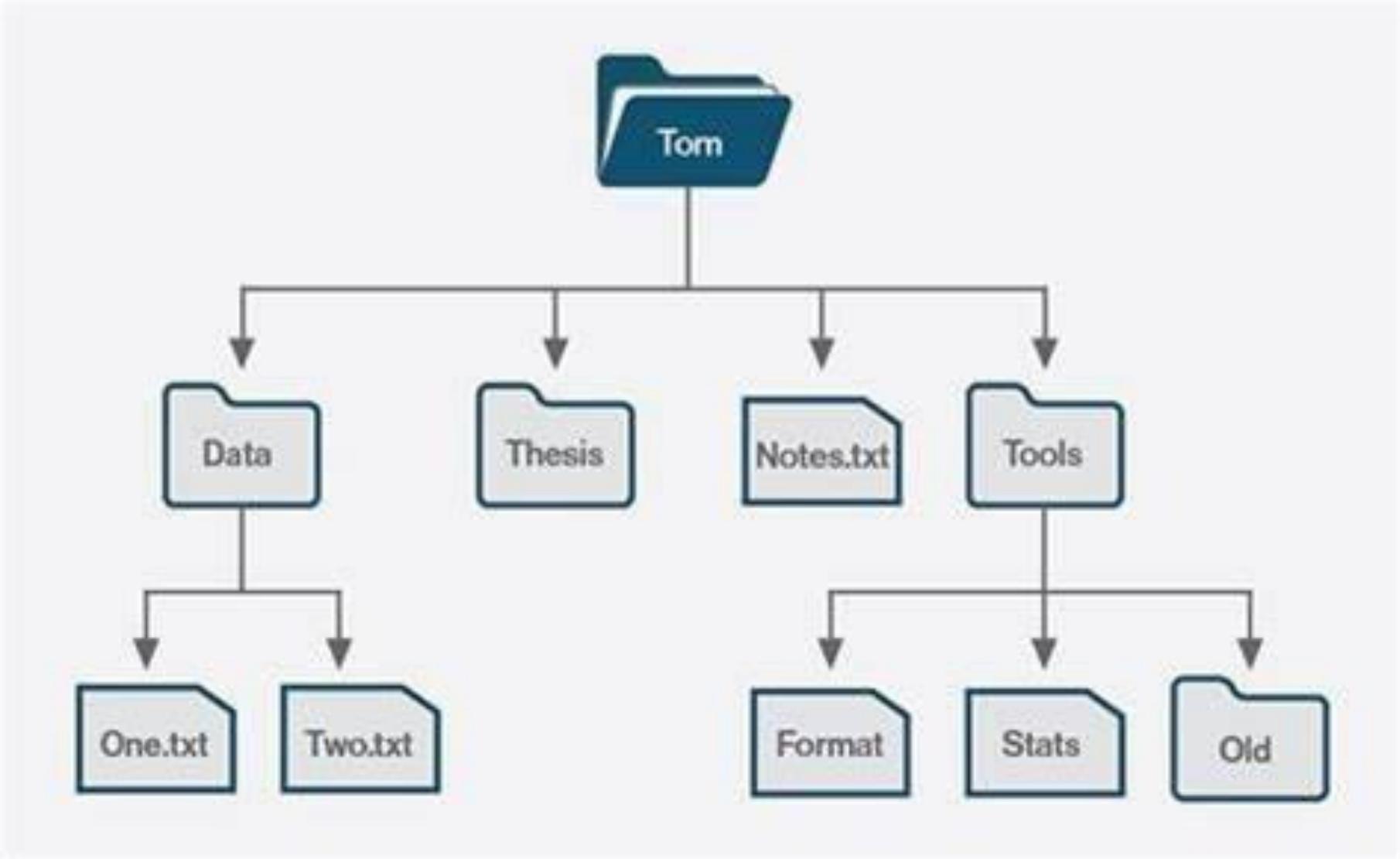






BPP

What is a filesystem?



Exercise - Exploring your filesystem

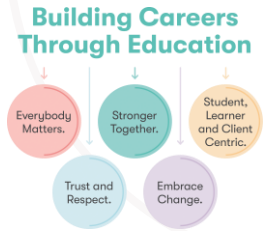
Demonstration of how to get into the PowerShell console

```
PowerShell>ls
```

The home directory is set as the current location when you launch the PowerShell console, thus running the preceding command without any parameters will display the contents of your home folder.

BPP

Exercise - Exploring your filesystem



Cd ..	Change your current working directory
Mkdir	Create new directories
Rm	Deleting files
Tree	Display the directory structure
Ctrl+C	Cancelling or stopping a script/command
Explorer	Windows/files explorer
Ni	Create new items
Cp	Coping files and directories
Mv	Moving and renaming files

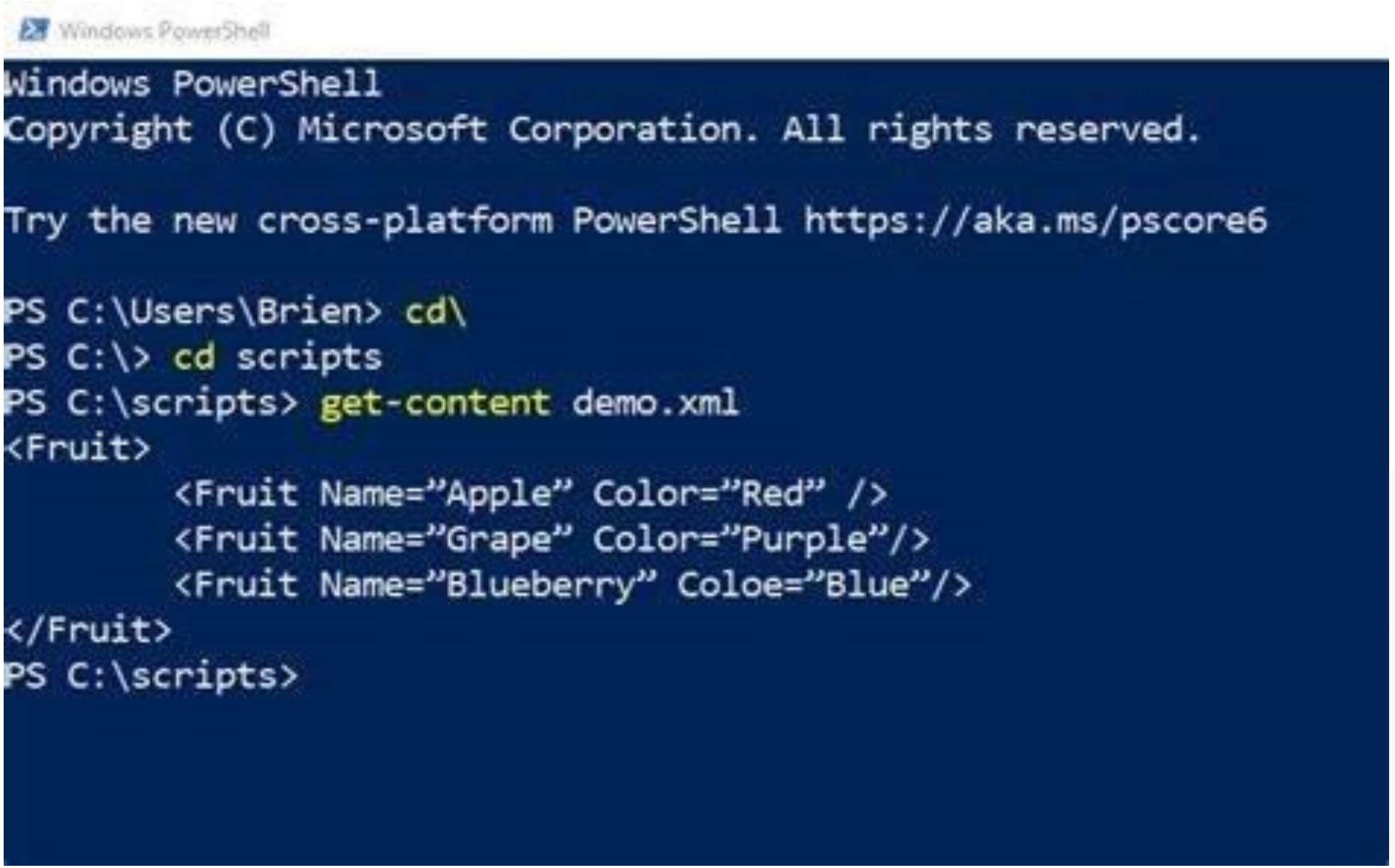
BPP

Show subfolders and calculate their sizes

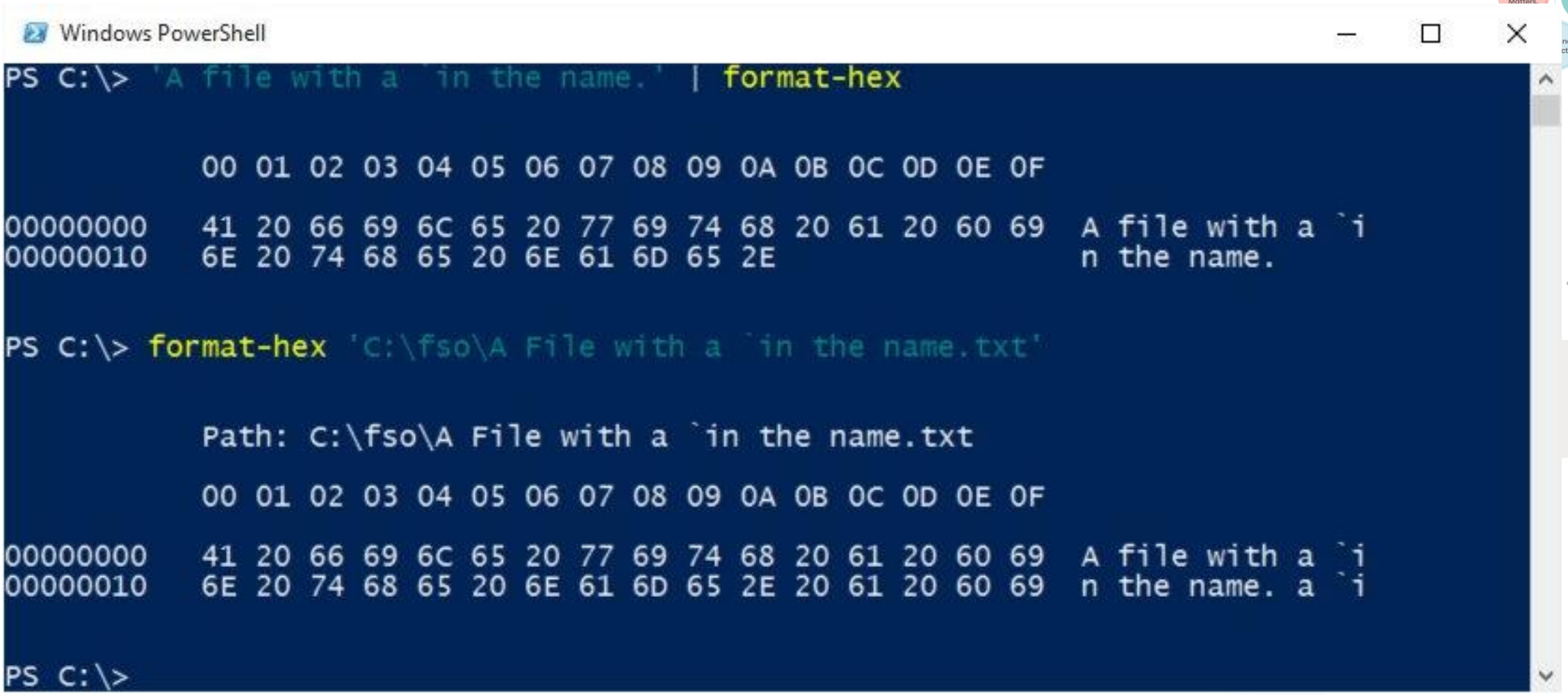

```powershell
$directory = "C:\"


Get-ChildItem -Path $directory -Directory | ForEach-Object {
    $folder = $_
    $totalSize = (Get-ChildItem -Path $_.FullName -File | Measure-Object -Property Length -Sum).Sum /
1MB
    "{0} has a total size of {1:N2} MB" -f $folder.Name, $totalSize
}
```

BPP

Reading a file

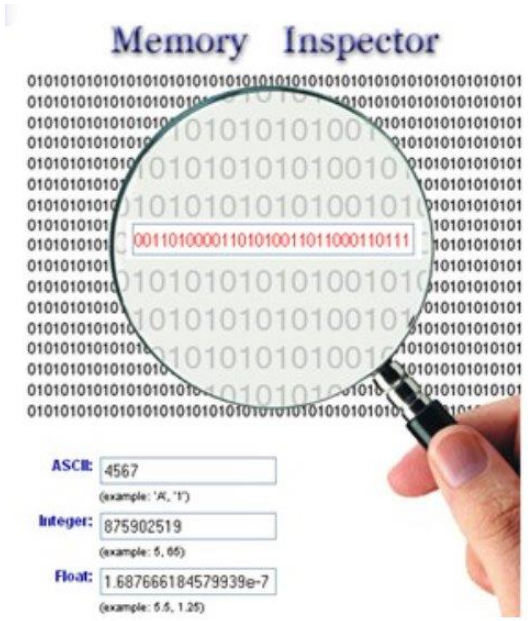


Hexadecimal display



BPP

Text vs Binary files



ASCII Table

00101110	.
00101111	/
00110000	0
00110001	1
00110010	2
00110011	3
00110100	4
00110101	5
00110110	6
00110111	7
00111000	8
00111001	9
00111010	:
00111011	;
00111100	<
00111101	=
00111110	>
00111111	?
01000000	@
01000001	A
01000010	B
01000011	C

Text files

Contain human-readable text (e.g., .txt, .csv).

Binary

Store non-human-readable data (e.g., images, executables).

Unicode

If a file contains **only** the decimal bytes 9–13, 32–126, it's probably a pure ASCII text file.

Otherwise, it's not. However, it may still be text in another encoding.

If, in **addition** to the above bytes, the file contains **only** the decimal bytes 128–255, it's probably a text file in an 8-bit or variable-length ASCII-based encoding such as ISO-8859-1, UTF-8 or ASCII+Big5.

If not, for some purposes you may be able to stop here and consider the file to be binary. However, it may still be text in a 16- or 32-bit encoding.
If a file doesn't meet the above constraints, examine the first 2–4 bytes of the file for a byte-order mark:

- If the first two bytes are hex FE FF, the file is **tentatively** UTF-16 BE.
- If the first two bytes are hex FF FE, and the following two bytes are **not** hex 00 00 , the file is **tentatively** UTF-16 LE.
- If the first four bytes are hex 00 00 FE FF, the file is **tentatively** UTF-32 BE.
- If the first four bytes are hex FF FE 00 00, the file is **tentatively** UTF-32 LE.

Unicode

AΩB

UTF-8

41 CE A9 42

UTF-16

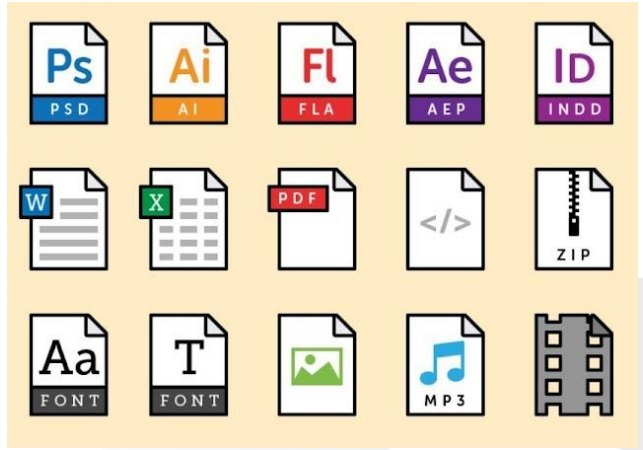
00 41 03 A9 00 42

UTF-32

00 00 00 41 00 00 03 A9 00 00 00 42

MIME types

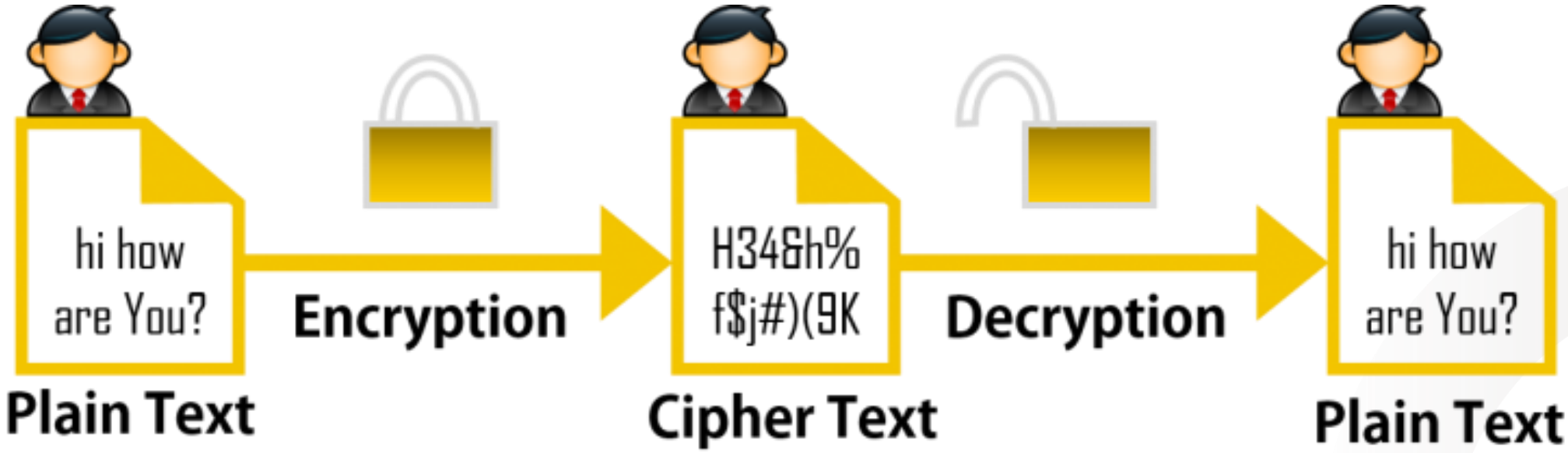
A MIME type is a label used to identify a type of data. It is used so software can know how to handle the data. It serves the same purpose on the Internet that file extensions do on Microsoft Windows.



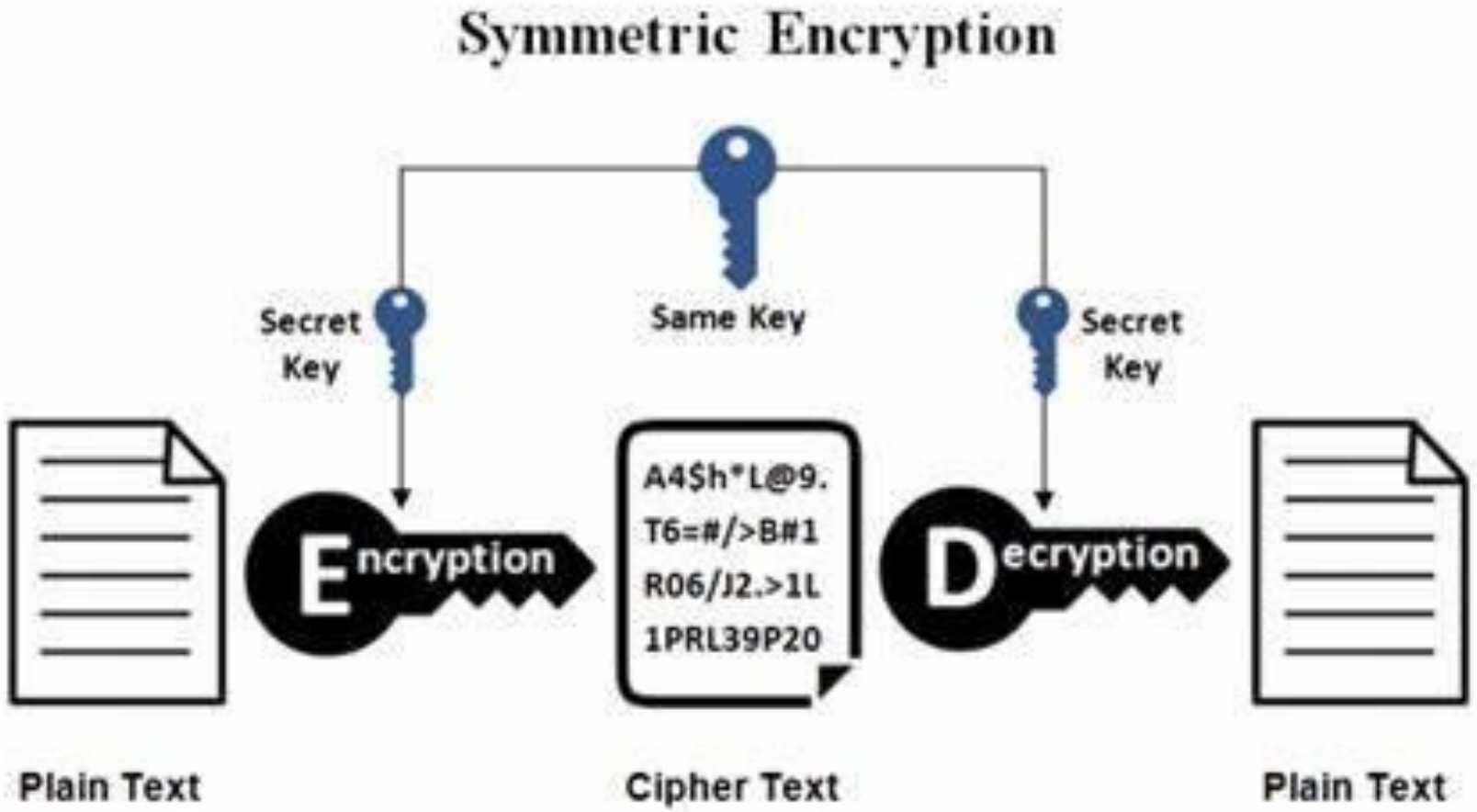
So if a server says "This is text/html" the client can go "Ah, this is an HTML document, I can render that internally", while if the server says "This is application/pdf" the client can go "Ah, I need to launch the FoxIt PDF Reader plugin that the user has installed and that has registered itself as the application/pdf handler."

File encryption

Files can be encrypted



Encryption uses keys



From files to databases

Why should be move from files to databases?

- Files get messy
- A filesystem is not the most efficient structure for querying data
- Databases are a structured way of querying data

From remote databases to the cloud

A local database is a database running on your own laptop/computer

Remote databases are more common as they provide access to multiple users

These can live on your employer's server(s), not necessarily in the cloud

Cloud database solutions, however, tend to use the same concepts as older on-premise remote database solutions

So, for example, you could provide database access via the cloud without your customers realising it's in the cloud and not on-premise

Cloud also offers other types of storage, such as plain file storage.

BPP

Tyes of cloud storage

Block storage

- Offers dedicated, low latency storage
- Is callable and offers high performance
- Is similar to local direct attached storage or a storage area network (SAN)
- Example: Amazon Elastice Block Sotrage (Amazon EBS)

File storage

- Stores data as files
- Is highly scalable
- Is ideal for storage such as content repositories and media stores
- Example: Amazon Elastic File Systems (Amazon EFS)

Object storage

- Stores unstructured, semi structured, or structured data
- Is highly scalable
- Uses a unique identifier for each object
- Has a lower cost than traditional storage
- Example: Amazon Simple Storage Service (Amazon S3)

BPP

Cloud storage has evolved

- Before the cloud era (pre-2010), we often talked about Data Warehouses which are massive databases stored on premise, that are very highly organised

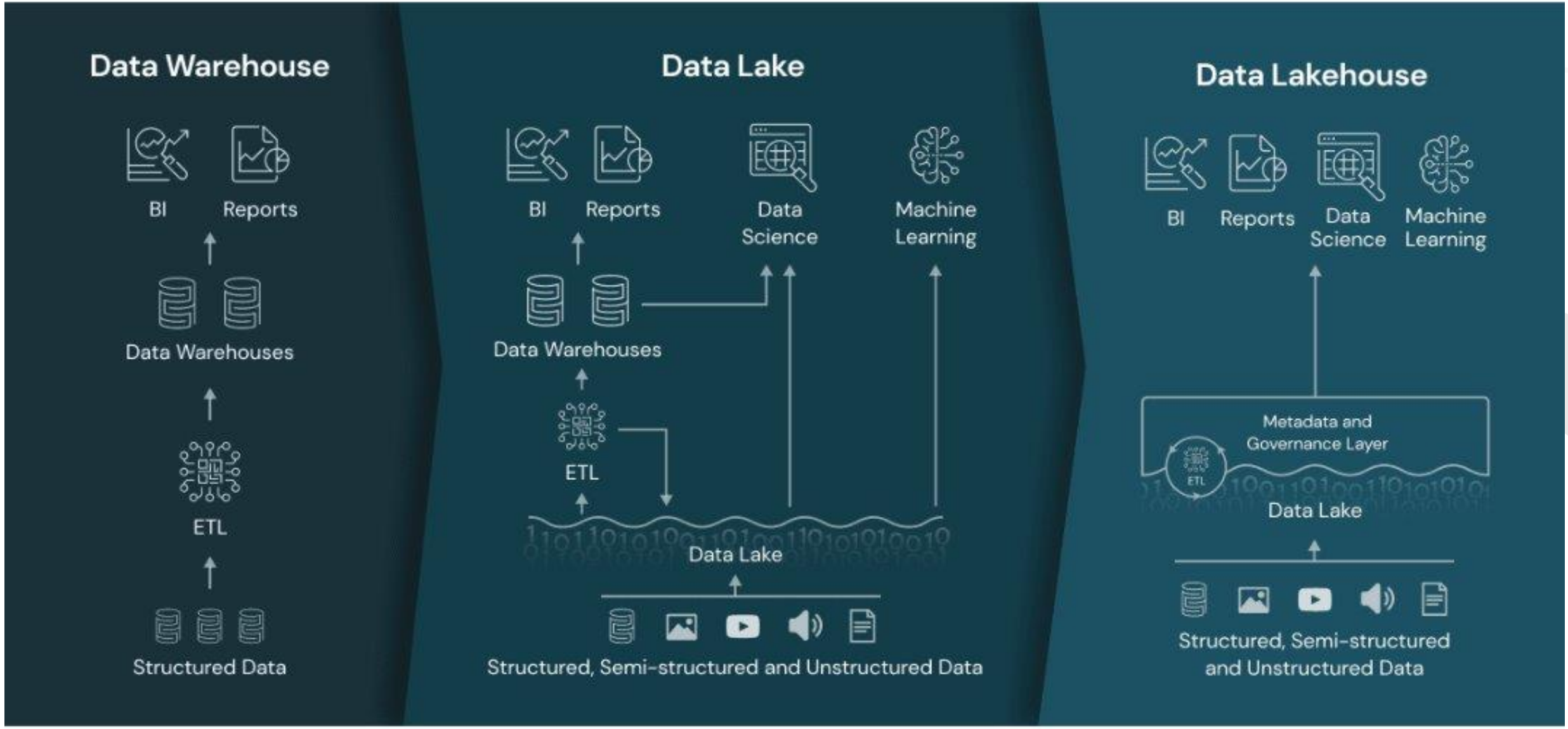
- During the cloud transition era (2010-2015) it was common to talk about Data Lakes which are unstructured areas in data centres for storing all sorts of Big Data

- Post-2015 this has evolved to Data Lakehouses which combine the benefits of both Lakes and Warehouses

BPP

Cloud storage has evolved

Building Careers Through Education



Data model

Data modeling skills are essential to keep data tidy and useful

- Data models are conceptual ways for describing the data
- They can use SQL, or other languages
- They can also be visual (remember diagrams)

The data model description generally consists of three parts:

- Structure of the data – what does it look like
- Operations on the data – what's available
- Constraints on the data – what's allowed

BPP

Structure of the data

Schema

- (e.g., table names, attribute names)
- Describe the conceptual structure of the data

Different from data structure

- (e.g., list, array)
- Data structure can be seen as a physical data model

BPP

Operations on the data

Query language (e.g., SQL)

- Describe what operations that can be performed on data

Two kinds of operations

- Operations that retrieve information
- Operations that change the database

Different from programming languages (e.g., C, Java)

- Support a set of limited operations
- Allow for query optimizations

BPP

Constraints on the data

Constraints	Different kinds of constraints	Different from programming languages (e.g., C, Java)
• (e.g., age > 0, student# is unique) • describe limitations on what the data can be.	• Domain constraints • Integrity constraints	• Ensure the correctness of data

BPP

Commonly used data models

Relational Data Model

Key-Value Data Model

Semi-structured Data Model (e.g., Json, XML)

BPP

The Relational Model in brief

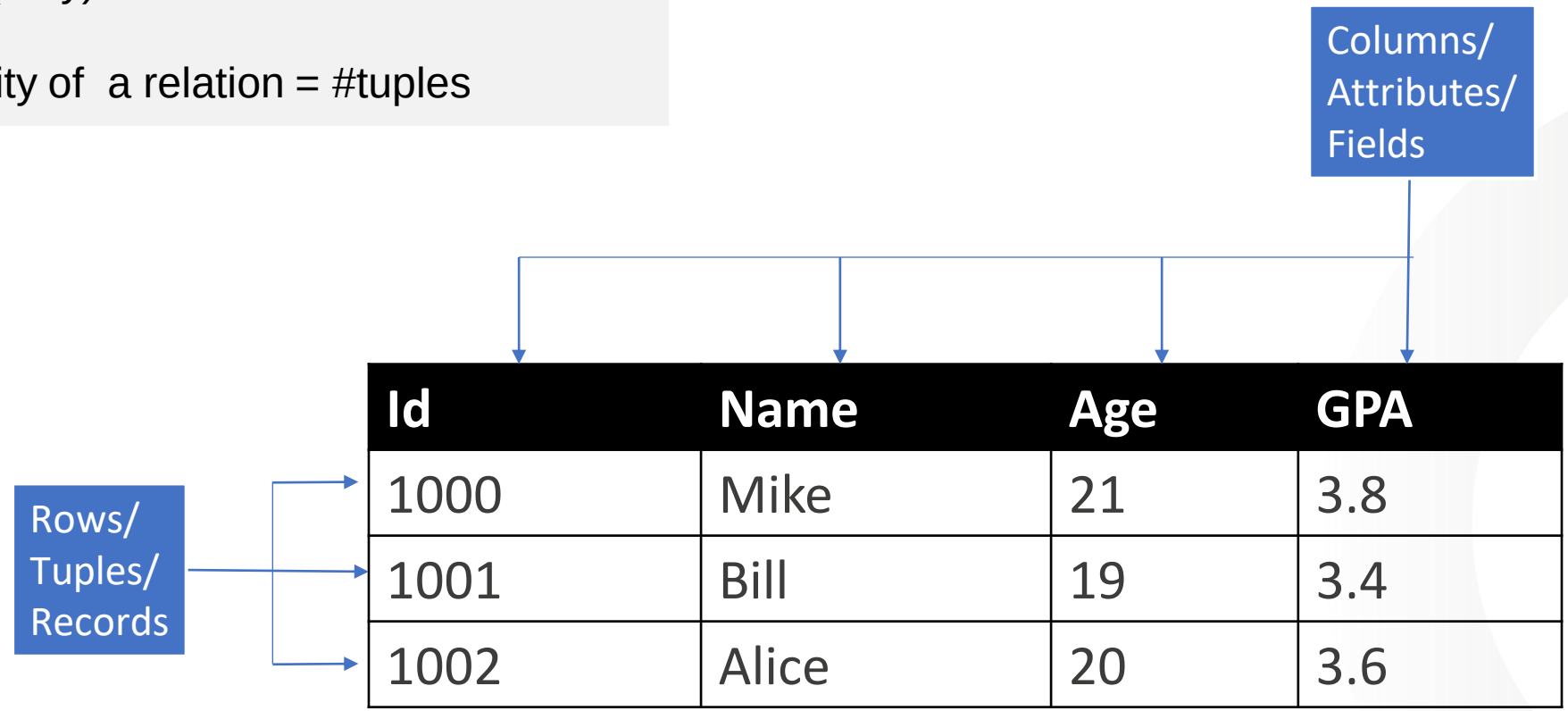
Id	Name	Age	GPA
1000	Mike	21	3.8
1001	Bill	19	3.4
1002	Alice	20	3.6

Structure of the Data	Query language	Constraints on the data
• Table structure.	• SQL	• E.g., id is unique, age > 10, name is not NULL

BPP

Terminology

- Relations/Tables
- Columns/Attributes/Fields
- Rows/Tuples/Records
- Degree (arity) of a relation = #attributes
- Cardinality of a relation = #tuples



Schema

Relation schema

- The name of a relation + The set of attributes for a relation

Student(id, sname, age, gpa)

Database schema

- The set of schemas for the relations of a database

Student (sid, sname, age, gpa)

Take (sid, cid)

Course (cid, cname, credit)

BPP

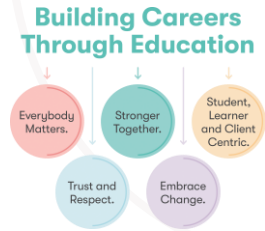
Domains

- Each attribute has a domain (data type)
- Examples
  - Text: CHAR(20), VARCHAR(50), TEXT
  - Integer: INT, SMALLINT
  - Real: DOUBLE, FLOAT
  - Few more that are vendor specific

```
Student(id:INT, sname:VARCHAR(50), age:INT, gpa:FLOAT)
```

BPP

Equivalent Representations of a Relation



Order does not matter!

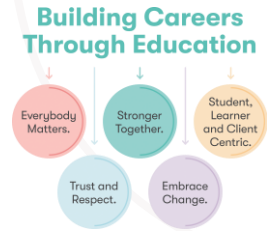
Id	Name	Age	GPA
1000	Mike	21	3.8
1001	Bill	19	3.4
1002	Alice	20	3.6

Id	Name	GPA	Age
1000	Mike	3.8	21
1001	Bill	3.4	19
1002	Alice	3.6	20

Id	Name	Age	GPA
1000	Mike	21	3.8
1002	Alice	20	3.6
1001	Bill	19	3.4

BPP

Exercise 1 - Terminology



Accounts

AcctNo	Type	Balance
12345	savings	12000
23456	checking	1000
34567	savings	25

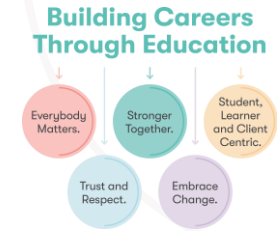
Customers

fname	lname	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

1. List two other ways of saying “rows”
2. List two other sayings of “columns”
3. List another saying of “table”

BPP

Exercise 2 - Terminology



Accounts

AcctNo	Type	Balance
12345	savings	12000
23456	checking	1000
34567	savings	25

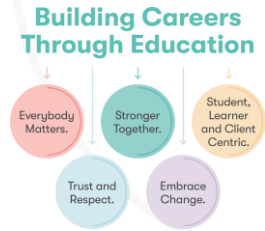
Customers

fname	lname	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

4. Indicate the attributes of each relation
5. Indicate the tuples of each relation
6. Indicate the degree of each relation
7. Indicate the cardinality of each relation

BPP

Exercise 3 - Terminology



Accounts

AcctNo	Type	Balance
12345	savings	12000
23456	checking	1000
34567	savings	25

Customers

fname	lname	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

8. Describe the schema for each relation
9. Describe the database schema
10. Specify a suitable domain for each attribute

BPP

Exercise 4 - Terminology

Customers

fname	lname	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

11. Indicate another equivalent way to represent this relation
12. Are there different ways to represent this relation?

BPP

Keys

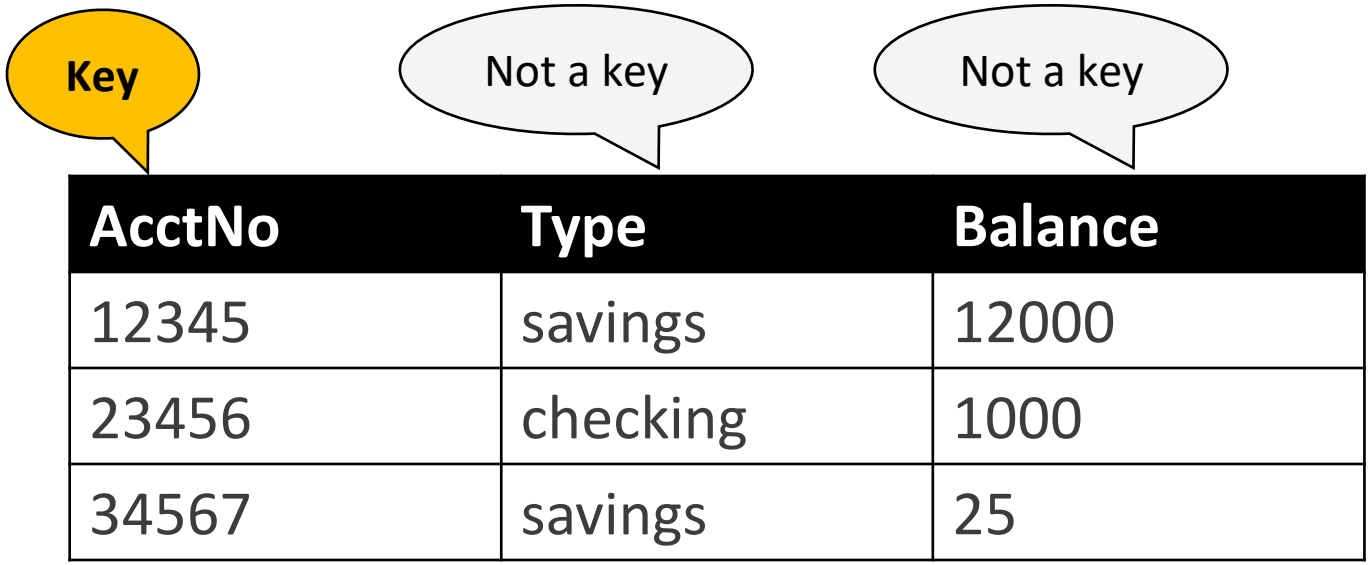
Key = one (or multiple) attributes that uniquely identify a record

AcctNo	Type	Balance
12345	savings	12000
23456	checking	1000
34567	savings	25

BPP

Keys

Key = one (or multiple) attributes that uniquely identify a record



AcctNo	Type	Balance
12345	savings	12000
23456	checking	1000
34567	savings	25

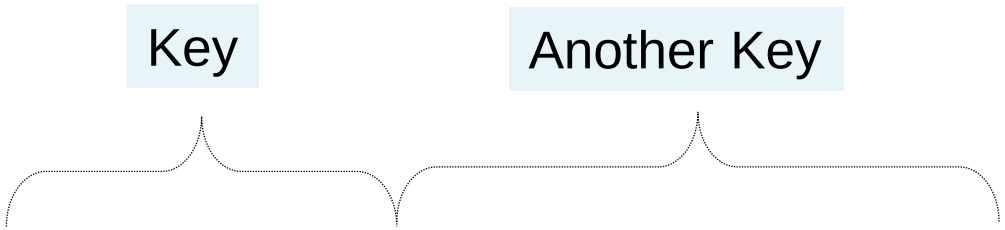
Multiple-attribute key

Multiple-attribute Key = multiple attributes that uniquely identify a record

Key = fname, lname

fname	lname	age	salary
Robbie	Banks	20	10k
Alice	Banks	30	8k
Alice	Smith	25	12k

Multiple keys



SSN	fname	lname	age	salary
123-456-789	Robbie	Banks	20	10k
222-111-709	Alice	Banks	30	8k
345-498-712	Alice	Smith	25	12k

We can choose one key as primary key (e.g., SSN)

Foreign keys

Attribute(s) whose value is a key of a record in some other relation

Accounts

acctNo	type	balance
12345	savings	12000
23456	checking	1000
34567	savings	25

Customers

fname	lname	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

Foreign key to Accounts.acctNo

BPP

Foreign keys

Attribute(s) whose value is a key of a record in some other relation

Customers

fname	lname	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

Accounts

acctNo	type	balance
12345	savings	12000
23456	checking	1000
34567	savings	25

Foreign key to Accounts.acctNo

Discussion

Tables are NOT ordered

- They are sets or multisets (bags)

Tables DO NOT prescribe how they are implemented/stored on disk

- This is called physical data independence

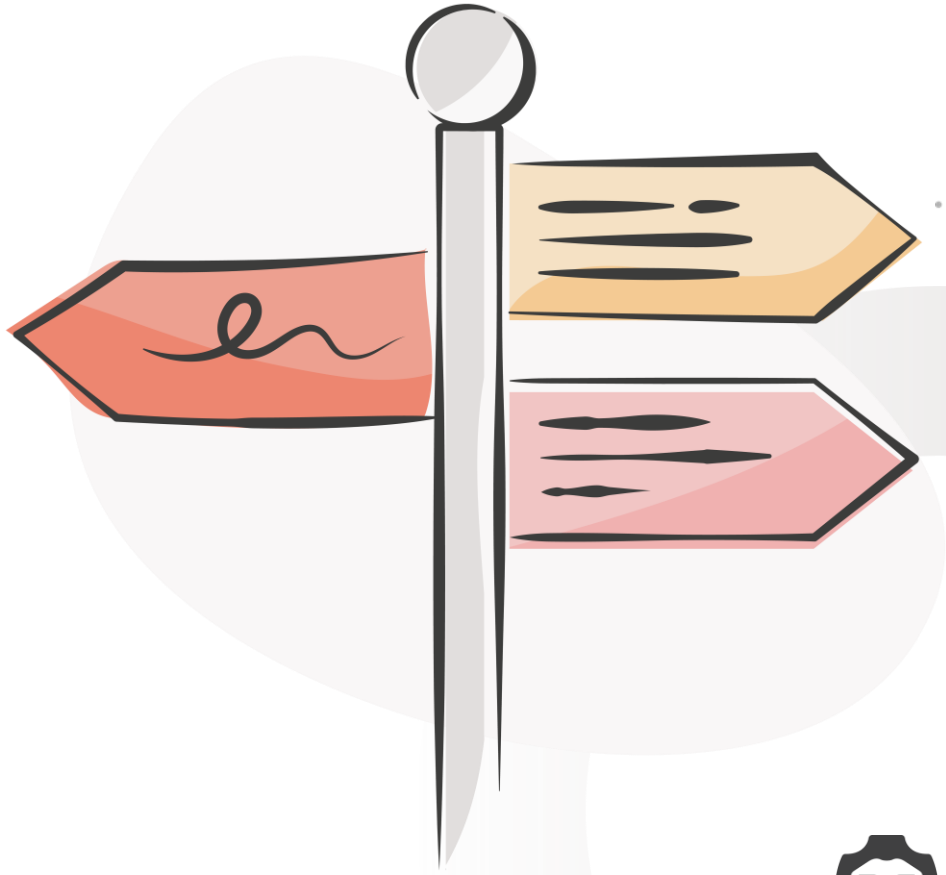
BPP

Outline

An overview of data models

Basics of the Relational Model

Define a relational schema in SQL



SQL DDL

SQL

- SQL stands for Structured Query Language

SQL is divided into two parts

- Data Manipulation Language (DML) which allows users to create, modify and query data
- Data Definition Language (DDL) which is used to define external and conceptual schemas

The DDL supports the creation, deletion and modification of tables

- Including the specification of domain constraints and other constraints

BPP

Creating Tables

- To create a table use the **CREATE TABLE** statement
- Specify the table name, field names and domains

```
CREATE TABLE Customer (
    sin         CHAR(11),
    firstName   CHAR(20),
    lastName    CHAR(20),
    age         INTEGER,
    income      REAL)
```

Question – is SQL case sensitive?

Answer – SQL keywords (create and table for example) are not case sensitive.

Named objects (tables, columns etc.) *may* be.

BPP

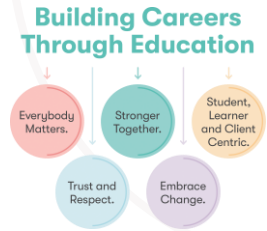
Deleting Tables

- To delete a table use the **DROP TABLE** statement
- This not only deletes all of the records but also deletes the table schema

```
DROP TABLE Customer
```

BPP

Modifying Tables



- Columns can be added or removed to tables using the **ALTER TABLE** statement
  - **ADD** to add a column and
  - **DROP** to remove a column

```
ALTER TABLE Customer
ADD height INTEGER
```

```
ALTER TABLE Customer
DROP height
```

BPP

Inserting Records

- To insert a record into an existing table use the **INSERT** statement
- The list of column names is optional
- If omitted the values must be in the same order as the columns

```
INSERT INTO Customer(SSN, firstName, lastName, age,
    income)
VALUES ('111', 'Sam', 'Spade', 23, 65234)
```

BPP

Deleting Records

- To delete a record use the **DELETE** statement
  - The **WHERE** clause specifies the record(s) to be deleted

```
DELETE
FROM Customer
WHERE SSN= '111'
```

- Be careful, the following SQL query deletes *all* the records in a table

```
DELETE
FROM Customer
```

BPP

Modifying Records

- Use the **UPDATE** statement to modify a record, or records, in a table
  - Note that the **WHERE** statement is evaluated *before* the **SET** statement
- Like **DELETE** the **WHERE** clause specifies which records are to be updated

```
UPDATE Customer
SET age = 37
WHERE SSN= '111'
```

BPP

Choosing your purpose-built database

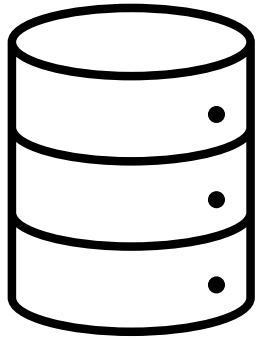
- Use the **UPDATE** statement to modify a record, or records, in a table
  - Note that the **WHERE** statement is evaluated *before* the **SET** statement
- Like **DELETE** the **WHERE** clause specifies which records are to be updated

```
UPDATE Customer
SET age = 37
WHERE SSN= '111'
```

BPP

Choosing your purpose-built database

Choosing the right database is key to supporting your application architecture.



Your database will affect what your application can handle, how it will perform, and the operation that you are responsible for.

Consider several factors:

- Application workload
- Data shape
- Performance requirements
- Operations burden

BPP

AWS Redshift

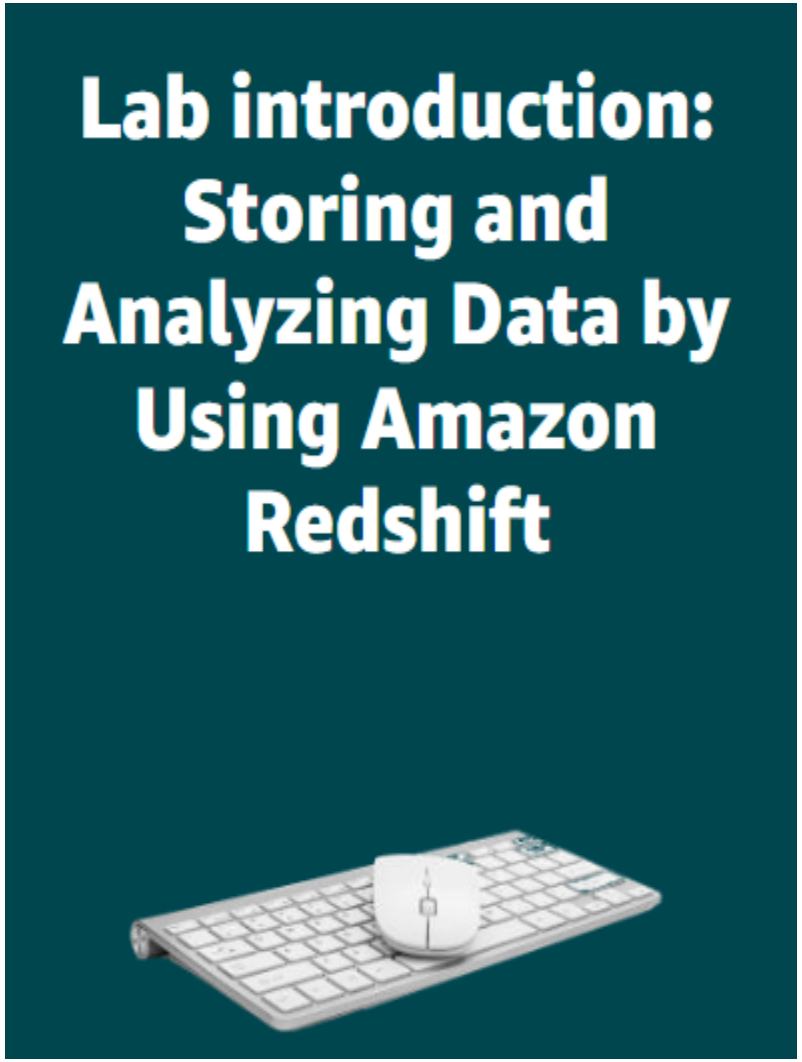
AWS Redshift is a data warehousing solution from Amazon Web Services. Redshift shines in its ability to handle huge volumes of data — capable of processing structured and unstructured data in the range of exabytes (10^{18} bytes). However, the service can also be used for large-scale data migrations.

Similar to many other AWS services, it can be deployed with just a few clicks and provides a plethora of options to import data. Additionally, the data in Redshift is always encrypted for added security.

BPP

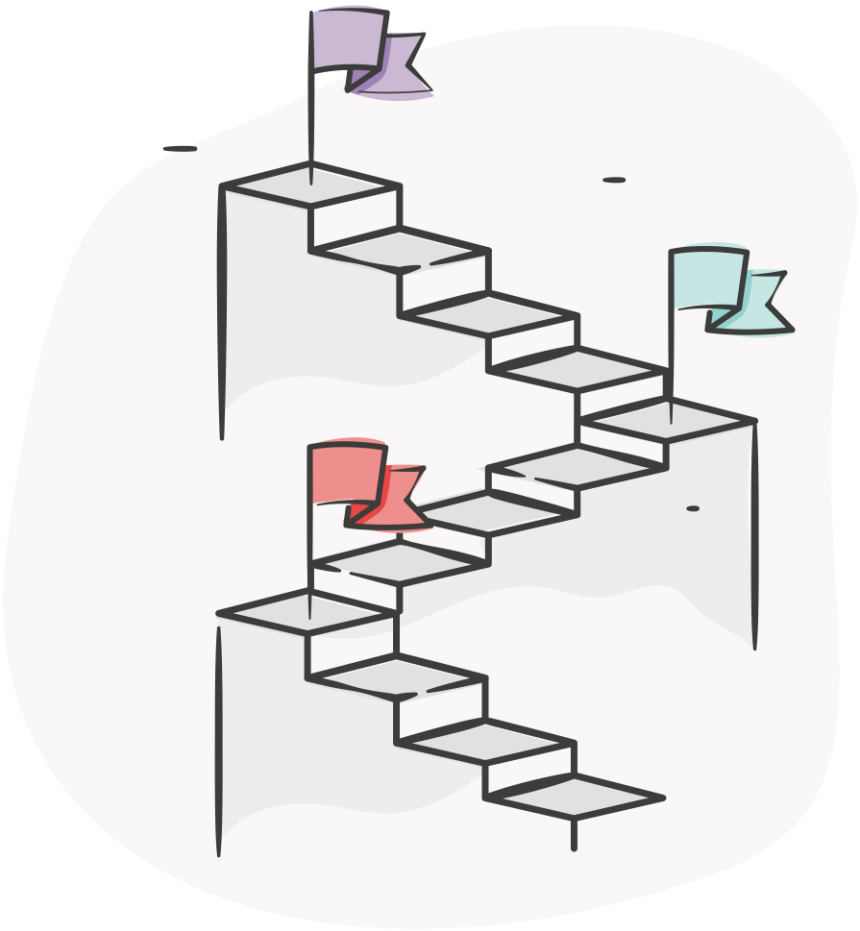
Lab Introduction

Storing and Analyzing Data by Using Amazon Redshift



- In this lab, you will build a proof of concept to use Amazon Redshift to address the need to query large datasets.
- You will use SQL queries for a music ticket sales dataset. These data files are already stored in Amazon S3 in another account, and you will load them into a Redshift database.
- Open your lab environment to start the lab and find additional details about the tasks that you will perform during this lab.

Learning Summary



BPP

Learning Journal Activity

In your learning journal, write the following report on cloud storage in your company:

- ❑ Which cloud storage solutions are used by your org?
- ❑ Which teams use them?
- ❑ What are the use cases and business requirements?
- ❑ How is the benefit to the business evidenced?
- ❑ Are there any challenges?

Additional research for extend/stretch:

- ❑ *Who monitors the cloud solutions internally?*
- ❑ *What is the customer's satisfaction with your org's cloud storage offering?*

Thank you

Do you have any questions, comments, or feedback?