DP-900 AZURE DATA FUNDAMENTALS

MASTER THE BASICS OF AZURE CLOUD AND DATA

SKILLS/TOPICS

Describe core data concepts (25–30%)

Identify considerations for relational data on Azure (20–25%)

Describe considerations for working with non-relational data on Azure (15–20%)

Describe an analytics workload on Azure (25–30%)

MICROSOFT AZURE DATA FUNDAMENTALS: EXPLORE CORE DATA CONCEPTS

- **Core Data Concepts**
  - Identify data formats
  - Explore file storage
  - Explore databases
  - Explore transactional data processing
  - Explore analytical data processing
- **Data Roles and Services**
  - Explore job roles in the world of data
  - Identify data services

DATA IS EVERYWHERE, IN A MULTITUDE OF STRUCTURES AND FORMATS.

Easier to collect

Cheaper to store

Grow revenue and make profits

DATA FORMATS

Data can be classified as structured, semi-structured and unstructured

Structured data – fixed schema, tabular, (rows/columns)

Semi-structured data – some structure (allows variation)

Unstructured data – without any structure

STRUCTURED DATA

Customer				
ID	**FirstName**	**LastName**	**Email**	**Address**
1	Joe	Jones	joe@litware.com	1 Main St.
2	Samir	Nadoy	samir@northwind.com	123 Elm Pl.

Product		
ID	**Name**	**Price**
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Rows represent instance of a data entity

Columns represent attributes of the entity

Relational Model

SEMI-STRUCTURED DATA

JavaScript Object Notation (JSON)

Extensible Markup Language (XML)

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<message>
    <warning>
        Hello World
    </warning>
</message>
```

JSON

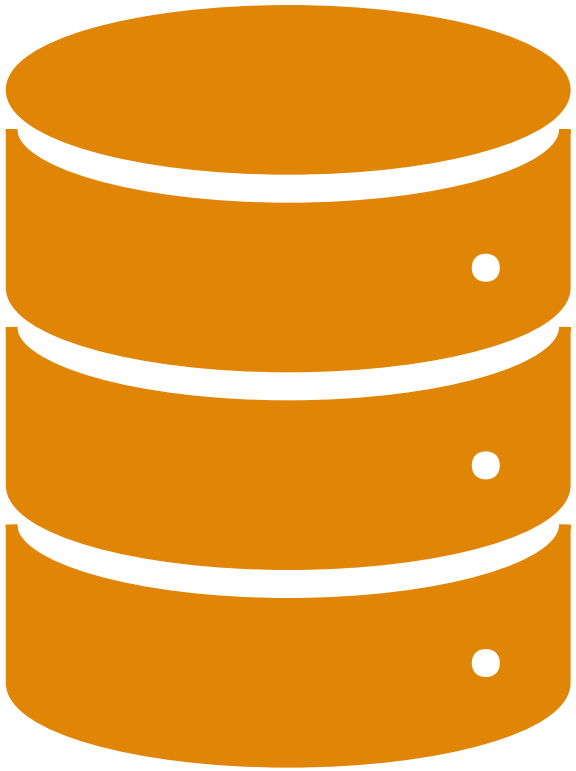
```json
// Customer 1
{
  "firstName": "Joe",
  "lastName": "Jones",
  "address":
  {
    "streetAddress": "1 Main St.",
    "city": "New York",
    "state": "NY",
    "postalCode": "10099"
  },
  "contact":
  [
    {
      "type": "home",
      "number": "555 123-1234"
    },
    {
      "type": "email",
      "address": "joe@litware.com"
    }
  ]
}

// Customer 2
{
  "firstName": "Samir",
  "lastName": "Nadoy",
  "address":
  {
    "streetAddress": "123 Elm Pl.",
    "unit": "500",
    "city": "Seattle",
    "state": "WA",
    "postalCode": "98999"
  },
  "contact":
  [
    {
      "type": "email",
      "address": "samir@northwind.com"
    }
  ]
}
```

UNSTRUCTURED DATA

No specific structure

- Documents
- Audio files
- Videos
- Images

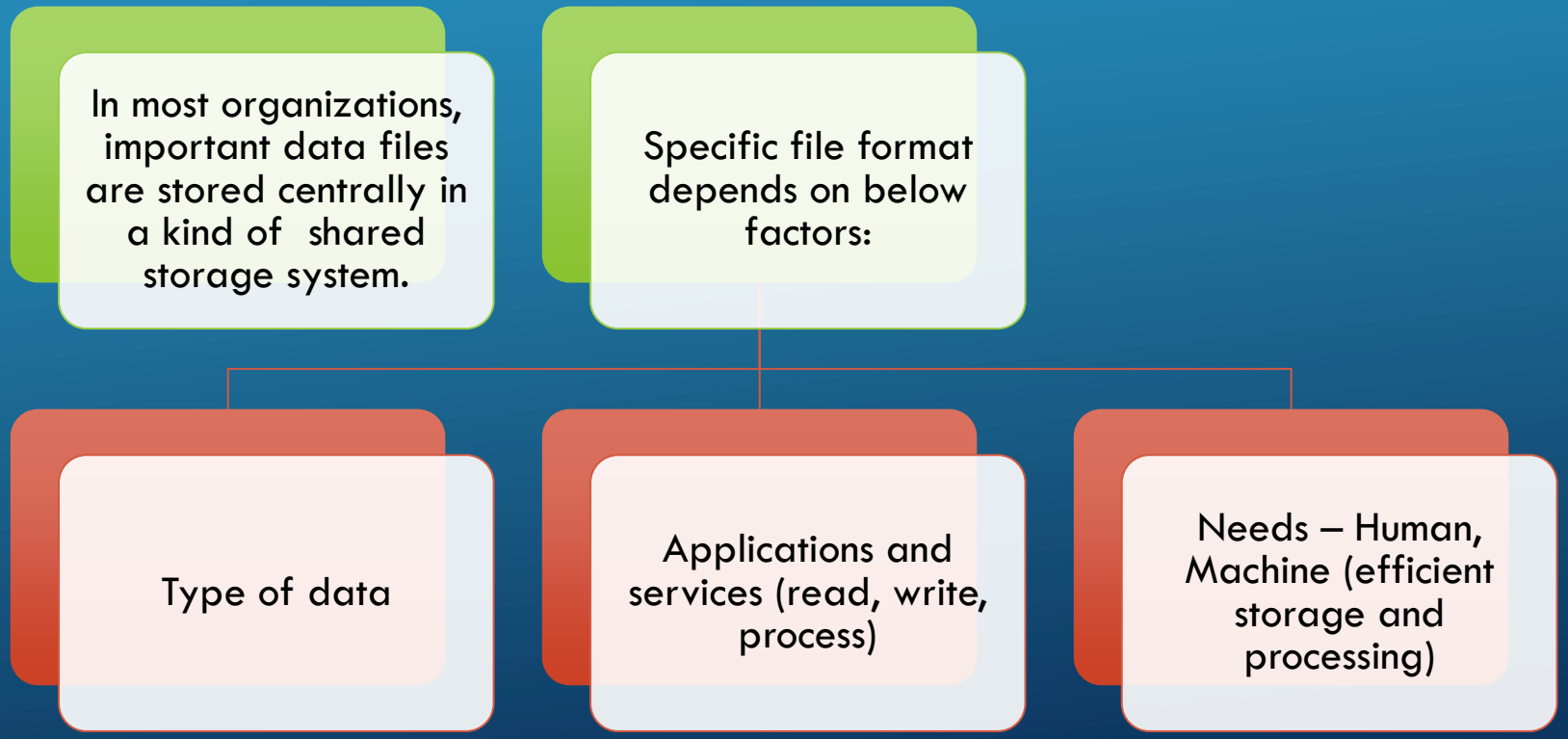
DATA STORES



There are two broad categories of data store in common use:

- File stores
- Databases

FILE STORAGE



COMMON FILE FORMATS

- Delimited text files
- JavaScript Object Notation (JSON)
- Extensible Markup Language (XML)
- Binary Large Object (BLOB)
- Optimized file formats

OPTIMIZED FILE FORMATS

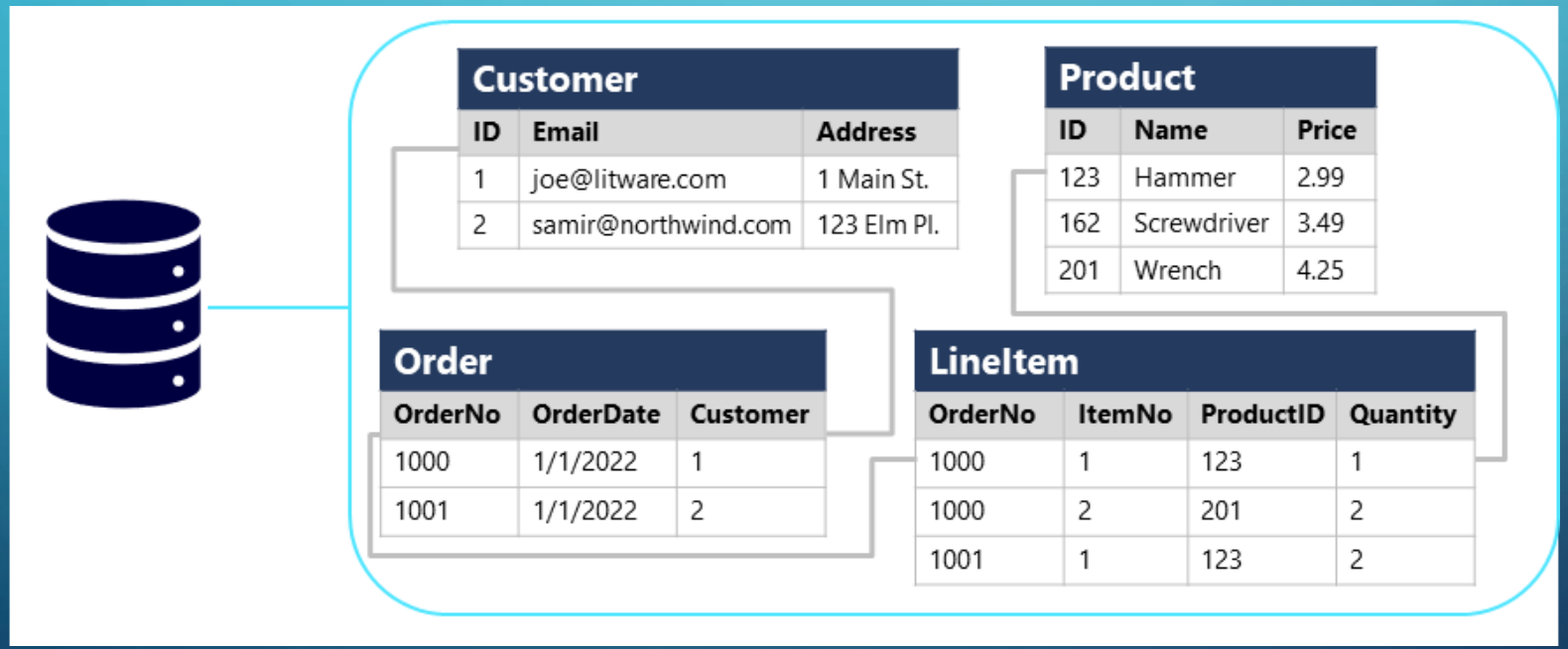
Avro – row-based format, created by Apache

ORC – Optimized row columnar format, developed by HortonWorks for optimizing read and write operations in Apache Hive

Parquet – Columnar data format, created by Cloudera and Twitter

DATABASES

- Relational Databases – commonly used to store and query structured data.



Customer

ID	Email	Address
1	joe@litware.com	1 Main St.
2	samir@northwind.com	123 Elm Pl.

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Order

OrderNo	OrderDate	Customer
1000	1/1/2022	1
1001	1/1/2022	2

LineItem

OrderNo	ItemNo	ProductID	Quantity
1000	1	123	1
1000	2	201	2
1001	1	123	2

NON-RELATIONAL DATABASES

There are four common types of Non-relational database commonly in use

Key-value databases in which each record consists of a unique key and an associated value, which can be in any format.

Document databases, which are a specific form of key-value database in which the value is a JSON document (which the system is optimized to parse and query)

Column family databases, which store tabular data comprising rows and columns, but you can divide the columns into groups known as column-families. Each column family holds a set of columns that are logically related together.

Graph databases, which store entities as nodes with links to define relationships between them.

NON-RELATIONAL DATABASES

Products

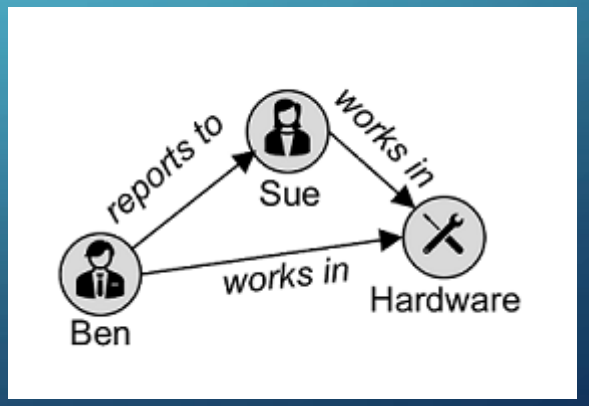
Key	Value
123	"Hammer ($2.99)"
162	"Screwdriver ($3.49)"
201	"Wrench ($4.25)"

Customers

Key	Document
1	{ "name": "Joe Jones", "email": "joe@litware.com" }
2	{ "name": "Samir Nadoy", "email": "Samir@northwind.com" }

Orders

Key	Customer		Product	
	Name	Address	Name	Price
1000	Joe Jones	1 Main St.	Hammer	2.99
1001	Samir Nadoy	123 Elm Pl.	Wrench	4.25



TRANSACTIONAL DATA PROCESSING (OLTP)

- A transactional system records *transactions* that encapsulate specific events that the organization wants to track
  - Banking
  - Retail
  - Others
- Transactional systems are often high-volume, sometimes handling many millions of transactions in a single day. The data being processed must be accessible very quickly.
- Data storage is optimized for both read and write operations
- Ensures integrity of the data
- Support ACID semantics

ACID PROPERTIES

Atomicity - Each transaction is treated as a single unit, which succeeds completely or fails completely.

Consistency - Transactions can only take the data in the database from one valid state to another.

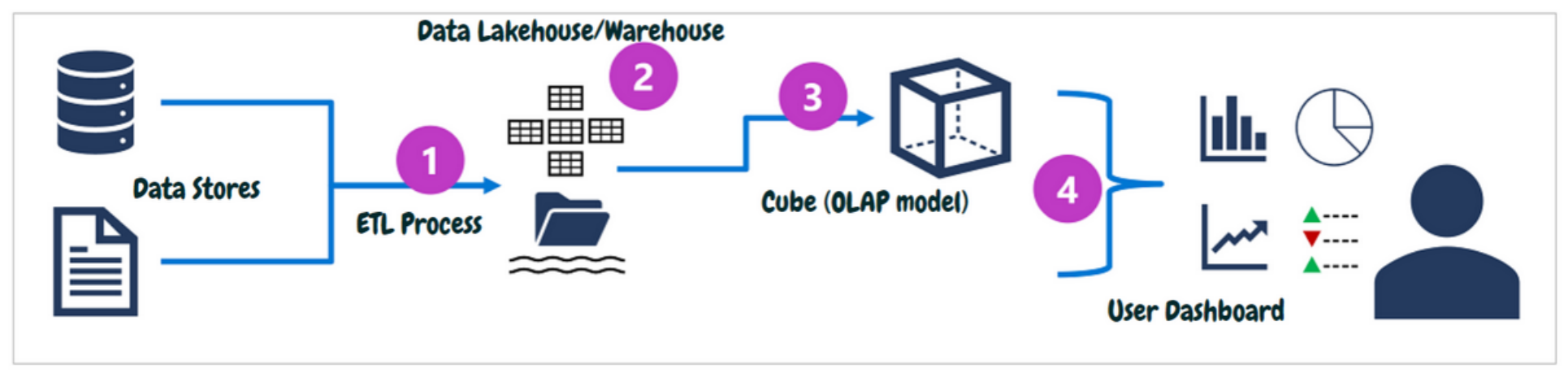
Isolation - Concurrent transactions cannot interfere with one another and must result in a consistent database state.

Durability - When a transaction has been committed, it will remain committed.

OLTP systems are typically used to support live applications that process business data - often referred to as *line of business* (LOB) applications.

ANALYTICAL DATA PROCESSING (OLAP)

- Analytical data processing typically uses read-only (or read-mostly) systems that store vast volumes of historical data or business metrics.
- An OLAP model is an aggregated type of data storage that is optimized for analytical workloads.
- Common architecture for enterprise-scale analytics:



Data lakes are common in large-scale data analytical processing scenarios, where a large volume of file-based data must be collected and analyzed.

Data warehouses are an established way to store data in a relational schema that is optimized for read operations – primarily queries to support reporting and data visualization.

Data Lakehouses are a more recent innovation that combine the flexible and scalable storage of a data lake with the relational querying semantics of a data warehouse.

JOB ROLES IN THE WORLD OF DATA

Database administrators manage databases, assigning permissions to users, storing backup copies of data and restore data in the event of a failure.

Data engineers manage infrastructure and processes for data integration across the organization, applying data cleaning routines, identifying data governance rules, and implementing pipelines to transfer and transform data between systems.

Data analysts explore and analyze data to create visualizations and charts that enable organizations to make informed decisions.

AZURE DATA SERVICES

Azure SQL

Azure databases
For open-source
relational databases

Azure Cosmos DB

Azure Storage

Azure Data Factory

Azure Synapse
Analytics

Azure Databricks

Azure HDInsight

Azure Stream Analytics

Azure Data Explorer

Microsoft Purview

Microsoft Fabrics

AZURE SQL

- *Azure SQL* is the collective name for a family of relational database solutions based on the Microsoft SQL Server database engine.
- Azure SQL services include:
  - **Azure SQL Database** – a fully managed platform-as-a-service (PaaS) database hosted in Azure
  - **Azure SQL Managed Instance** – a hosted instance of SQL Server with automated maintenance, which allows more flexible configuration than Azure SQL DB but with more administrative responsibility for the owner.
  - **Azure SQL VM** – a virtual machine with an installation of SQL Server, allowing maximum configurability with full management responsibility.

AZURE DATABASE FOR OPEN-SOURCE RELATIONAL DATABASES

- **Azure Database for MySQL** - a simple-to-use open-source database management system that is commonly used in *Linux*, *Apache*, *MySQL*, and *PHP* (LAMP) stack apps.
- **Azure Database for MariaDB** - a newer database management system, created by the original developers of MySQL. The database engine has since been rewritten and optimized to improve performance. MariaDB offers compatibility with Oracle Database (another popular commercial database management system).
- **Azure Database for PostgreSQL** - a hybrid relational-object database. You can store data in relational tables, but a PostgreSQL database also enables you to store custom data types, with their own non-relational properties.

AZURE COSMOS DB

- Azure Cosmos DB is a global-scale non-relational (*NoSQL*) database system that supports multiple application programming interfaces (APIs), enabling you to store and manage data as JSON documents, key-value pairs, column-families, and graphs.

AZURE STORAGE

Azure Storage is a core Azure service that enables you to store data in:

- **Blob containers** - scalable, cost-effective storage for binary files.
- **File shares** - network file shares such as you typically find in corporate networks.
- **Tables** - key-value storage for applications that need to read and write data values quickly.

AZURE DATA FACTORY

Azure Data Factory is an Azure service that enables you to define and schedule data pipelines to transfer and transform data. You can integrate your pipelines with other Azure services, enabling you to ingest data from cloud data stores, process the data using cloud-based compute, and persist the results in another data store.

AZURE SYNAPSE ANALYTICS

Azure Synapse Analytics is a comprehensive, unified Platform-as-a-Service (PaaS) solution for data analytics that provides a single service interface for multiple analytical capabilities, including:

- **Pipelines** - based on the same technology as Azure Data Factory.
- **SQL** - a highly scalable SQL database engine, optimized for data warehouse workloads.
- **Apache Spark** - an open-source distributed data processing system that supports multiple programming languages and APIs, including Java, Scala, Python, and SQL.
- **Azure Synapse Data Explorer** - a high-performance data analytics solution that is optimized for real-time querying of log and telemetry data using Kusto Query Language (KQL).

AZURE DATABRICKS

Azure Databricks is an Azure-integrated version of the popular Databricks platform, which combines the Apache Spark data processing platform with SQL database semantics and an integrated management interface to enable large-scale data analytics.

AZURE HDINSIGHT

- Azure HDInsight is an Azure service that provides Azure-hosted clusters for popular Apache open-source big data processing technologies, including:
- **Apache Spark** - a distributed data processing system that supports multiple programming languages and APIs, including Java, Scala, Python, and SQL.
- **Apache Hadoop** - a distributed system that uses *MapReduce* jobs to process large volumes of data efficiently across multiple cluster nodes. MapReduce jobs can be written in Java or abstracted by interfaces such as Apache Hive - a SQL-based API that runs on Hadoop.
- **Apache HBase** - an open-source system for large-scale NoSQL data storage and querying.
- **Apache Kafka** - a message broker for data stream processing.

AZURE STREAM ANALYTICS

Azure Stream Analytics is a real-time stream processing engine that captures a stream of data from an input, applies a query to extract and manipulate data from the input stream, and writes the results to an output for analysis or further processing.

AZURE DATA EXPLORER

Azure Data Explorer is a standalone service that offers the same high-performance querying of log and telemetry data as the Azure Synapse Data Explorer runtime in Azure Synapse Analytics.

AZURE PURVIEW

Microsoft Purview provides a solution for enterprise-wide data governance and discoverability. You can use Microsoft Purview to create a map of your data and track data lineage across multiple data sources and systems, enabling you to find trustworthy data for analysis and reporting.

AZURE FABRIC

- Microsoft Fabric is a unified Software-as-a-Service (SaaS) analytics platform based on open and governed lakehouse that includes functionality to support:
- Data ingestion and ETL
- Data lakehouse analytics
- Data warehouse analytics
- Data Science and machine learning
- Realtime analytics
- Data visualization
- Data governance and management


END OF PART-1
WHAT DID WE COVER TILL NOW?

MICROSOFT AZURE DATA FUNDAMENTALS: EXPLORE RELATIONAL DATA IN AZURE

- **Explore fundamental relational data concepts**
  - Understand relational data
  - Understand normalization
  - Explore SQL
  - Describe database objects
- **Explore relational database services**
  - Describe Azure SQL services and capabilities
  - Describe Azure services for open-source databases
  - Exercise: Explore Azure relational database services

MICROSOFT AZURE DATA FUNDAMENTALS: EXPLORE RELATIONAL DATA IN AZURE

In the early years of computing systems, every application stored data in its own unique structure.

The *relational* database model was designed to solve the problem of multiple arbitrary data structures.

One of the key advantages of the relational database model is its use of *tables*, which are an intuitive, efficient, and flexible way to store and access structured information.

Simple yet powerful

RELATIONAL DATA

- Format for structured data
- *tables* represent entities from the real world
- Retail system example:
- Each row has same columns
- Each column has specific data type
- Not all columns need to have a value (can be empty or null)
- Available data types depend on Database system

Customer

ID	FirstName	MiddleName	LastName	Email	Address	City
1	Joe	David	Jones	joe@litware.com	1 Main St.	Seattle
2	Samir		Nadoy	samir@northwind.com	123 Elm Pl.	New York

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Order

OrderNo	OrderDate	Customer
1000	1/1/2022	1
1001	1/1/2022	2

LineItem

OrderNo	ItemNo	ProductID	Quantity
1000	1	123	1
1000	2	201	2
1001	1	123	2

UNDERSTAND NORMALIZATION

Normalization is a term used by database professionals for a schema design process that minimizes data duplication and enforces data integrity.

Process of normalization:

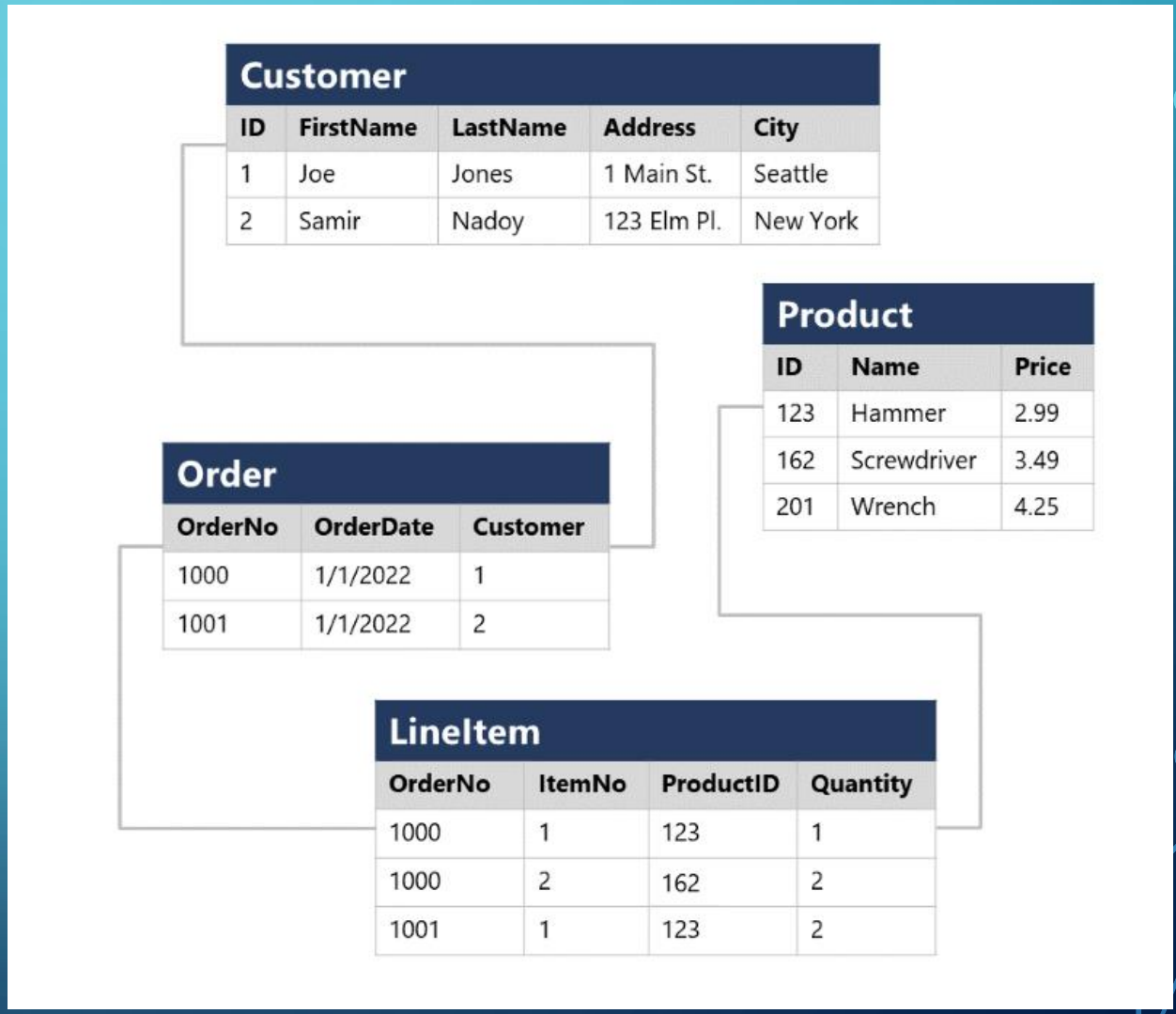
1. Separate each *entity* into its own table.
2. Separate each discrete *attribute* into its own column.
3. Uniquely identify each entity instance (row) using a *primary key*.
4. Use *foreign key* columns to link related entities.

WHAT'S THE PROBLEM WITH THIS TABLE?

Sales Data				
OrderNo	**OrderDate**	**Customer**	**Product**	**Quantity**
1000	1/1/2022	Joe Jones, 1 Main St, Seattle	Hammer ($2.99)	1
1000	1/1/2022	Joe Jones- 1 Main St, Seattle	Screwdriver ($3.49)	2
1001	1/1/2022	Samir Nadoy, 123 Elm Pl, New York	Hammer ($2.99)	2
...	...	...	...	...

NORMALIZED TABLE

- Each entity (customer, product, sales order, and line item) is stored in its own table
- Eliminates data duplication
- Each value is constrained to its own data type
- Instances of each entity are uniquely identified by an ID or other key value
- Primary, foreign and composite key



Customer

ID	FirstName	LastName	Address	City
1	Joe	Jones	1 Main St.	Seattle
2	Samir	Nadoy	123 Elm Pl.	New York

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Order

OrderNo	OrderDate	Customer
1000	1/1/2022	1
1001	1/1/2022	2

LineItem

OrderNo	ItemNo	ProductID	Quantity
1000	1	123	1
1000	2	162	2
1001	1	123	2

EXPLORE SQL

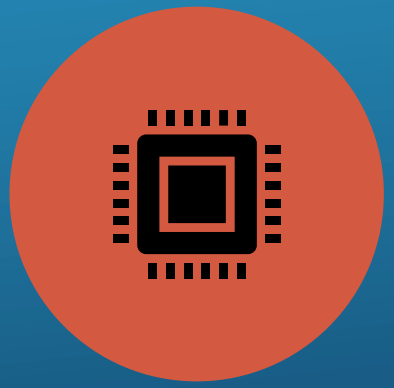
Structured Query Language (SQL) is used to communicate with a relational database.

Dialects of SQL include:

- *Transact-SQL (T-SQL) - This version of SQL is used by Microsoft SQL Server and Azure SQL services.*
- *pgSQL - This is the dialect, with extensions implemented in PostgreSQL.*
- *PL/SQL - This is the dialect used by Oracle. PL/SQL stands for Procedural Language/SQL.*

SQL STATEMENT TYPES

DATA DEFINITION LANGUAGE (DDL)



DATA CONTROL LANGUAGE (DCL)

DATA MANIPULATION LANGUAGE (DML)

DDL STATEMENTS

Statement	Description
CREATE	Create a new object in the database, such as a table or a view.
ALTER	Modify the structure of an object. For instance, altering a table to add a new column.
DROP	Remove an object from the database.
RENAME	Rename an existing object.

```
CREATE TABLE Product
(
    ID INT PRIMARY KEY,
    Name VARCHAR(20) NOT NULL,
    Price DECIMAL NULL
);
```

```
CREATE TABLE Product
(
    ID INT PRIMARY KEY,
    Name VARCHAR(20) NOT NULL,
    Price DECIMAL NULL
);
```

DCL STATEMENTS

Statement	Description
GRANT	Grant permission to perform specific actions
DENY	Deny permission to perform specific actions
REVOKE	Remove a previously granted permission

```
GRANT SELECT, INSERT, UPDATE
ON Product
TO user1;
```

DML STATEMENTS

Statement	Description
SELECT	Read rows from a table
INSERT	Insert new rows into a table
UPDATE	Modify data in existing rows
DELETE	Delete existing rows

```
SELECT FirstName, LastName,
Address, City
FROM Customer
WHERE City = 'Seattle'
ORDER BY LastName;
```

```
SELECT
o.OrderNo,
o.OrderDate,
c.Address,
c.City
FROM Order AS o
JOIN Customer
AS c
ON o.Customer =
c.ID
```

```
UPDATE Customer
SET Address = '123 High St.'
WHERE ID = 1;
```

```
DELETE FROM Product WHERE ID = 162;
```

```
INSERT INTO Product(ID, Name, Price)
VALUES (99, 'Drill', 4.99);
```

DATABASE OBJECTS

A view is a virtual table based on the results of a **SELECT** query.

```
CREATE VIEW Deliveries
AS
SELECT o.OrderNo, o.OrderDate,
       c.FirstName, c.LastName, c.Address,
c.City
FROM Order AS o JOIN Customer AS c
ON o.Customer = c.ID;
```

DATABASE OBJECTS …

- A *stored procedure* is a group of SQL commands that can be stored and reused multiple times.

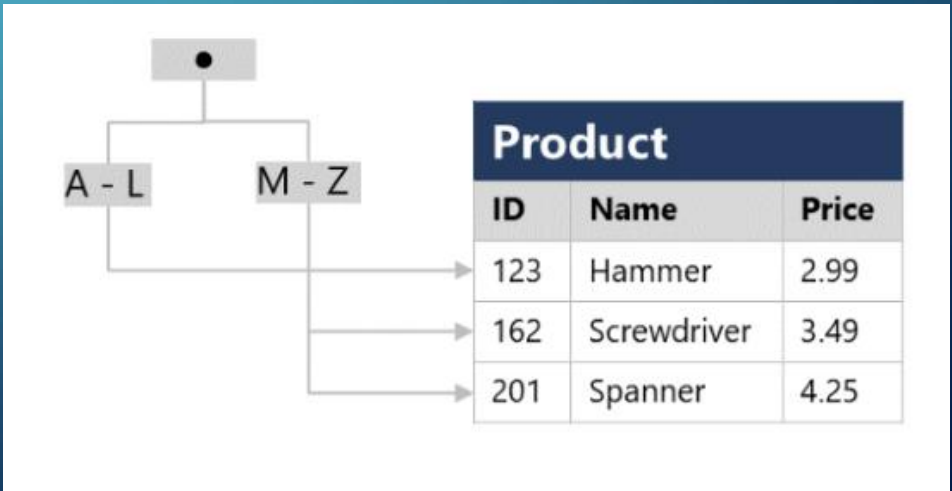
```
CREATE PROCEDURE RenameProduct
    @ProductID INT,
    @NewName VARCHAR(20)
AS
UPDATE Product
SET Name = @NewName
WHERE ID = @ProductID;
```

```
EXEC RenameProduct 201, 'Spanner';
```

DATABASE OBJECTS …

- An *index* is used to quickly locate the data without the need to go through each row.
- Index contains copy of data in sorted order.
- DBMS use index to fetch the data quickly (via WHERE clause)
- User can create multiple indexes

```
CREATE INDEX idx_ProductName ON Product(Name);
```



AZURE DATABASE SERVICES

- Most Azure database services are fully managed
- Enterprise-grade performance with built-in high availability means you can scale quickly and reach global distribution without worrying about costly downtime.
- Built-in security with automatic monitoring and threat detection, automatic tuning for improved performance.
- Guaranteed availability.

AZURE SQL SERVICES AND CAPABILITIES

--	SQL Server on Azure VMs	Azure SQL Managed Instance	Azure SQL Database
Type of cloud service	IaaS 	PaaS 	PaaS 
SQL Server compatibility	Fully compatible with on-premises physical and virtualized installations. Applications and databases can easily be "lift and shift" migrated without change.	Near-100% compatibility with SQL Server. Most on-premises databases can be migrated with minimal code changes by using the Azure Database Migration service	Supports most core database-level capabilities of SQL Server. Some features depended on by an on-premises application may not be available.
Architecture	SQL Server instances are installed in a virtual machine. Each instance can support multiple databases.	Each managed instance can support multiple databases. Additionally, *instance pools* can be used to share resources efficiently across smaller instances.	You can provision a *single database* in a dedicated, managed (logical) server; or you can use an *elastic pool* to share resources across multiple databases and take advantage of on-demand scalability.
Availability	99.99%	99.99%	99.995%
Management	You must manage all aspects of the server, including operating system and SQL Server updates, configuration, backups, and other maintenance tasks.	Fully automated updates, backups, and recovery.	Fully automated updates, backups, and recovery.
Use cases	Use this option when you need to migrate or extend an on-premises SQL Server solution and retain full control over all aspects of server and database configuration.	Use this option for most cloud migration scenarios, particularly when you need minimal changes to existing applications.	Use this option for new cloud solutions, or to migrate applications that have minimal instance-level dependencies.

AZURE SERVICES FOR OPEN-SOURCE DATABASES

Azure provides services for other popular relational database systems – MySQL, MariaDB, PostgreSQL

Motivation – Move on-premises apps to Azure quickly

MySQL – Leading open-source relational database for Linux, Apache, MySQL, and PHP (LAMP) stack apps
Editions : Community, Standard, Enterprise

MariaDB – Created by developers of MySQL
Rewritten and optimized to improve performance * Support for temporal data * Can hold several versions of data

PostgreSQL – Hybrid relational-object database
Stores custom datatypes with non-relational properties
Ability to store and manipulate geometric data

BENEFITS OF AZURE DATABASE FOR MYSQL/MARIADB/POSTGRESQL

- High availability features built-in.
- Predictable performance.
- Easy scaling that responds quickly to demand.
- Secure data, both at rest and in motion.
- Automatic backups and point-in-time restore for the last 35 days.
- Enterprise-level security and compliance with legislation.

EXERCISE: EXPLORE AZURE RELATIONAL DATABASE SERVICES

Explore Azure SQL Database


END OF PART-2
WHAT DID WE COVER TILL NOW?