DS-2002: Data Systems

An Overview of SQL Databases

Prof. Jon Tupitza

UNIVERSITY of VIRGINIA

SQL Database Design

Understanding the Principles that Govern Database Structure

UNIVERSITY *of* VIRGINIA

Fundamental Structures: Enforcing Data Integrity

Essential Design Concepts & Database Objects Required for Enforcing Data Integrity

Entity Integrity

- Enforced by the **Table**
  - Entities (nouns):
    - People, Places and Things
  - Concrete: Employees, Customers, Products
  - Conceptual: Sales, Scenarios, etc.

Domain Integrity

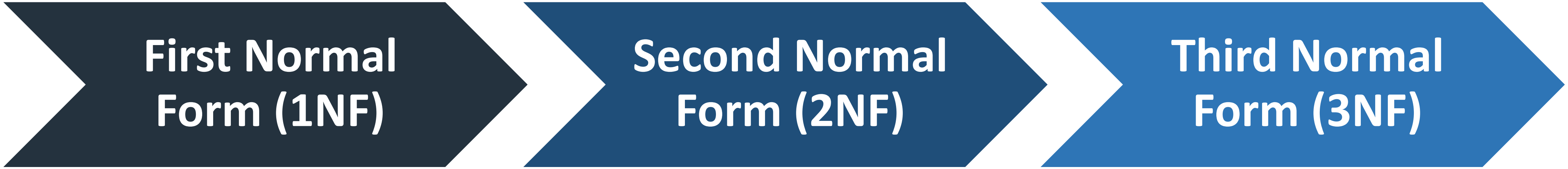
- Enforced by the **Column**
  - Data Type definition:
    - Int, Decimal, Float, Char, Nchar, Varchar, Nvarchar, DateTime
  - Constraints:
    - Primary Key, Check, Unique, & Default

Relational Integrity

- Enforced by the **Foreign Key Relationship**
  - One-to-Many: Foreign key relates to Primary key
  - Many-to-Many: Primary keys relate to Foreign keys via a *Juncture table*

Database Normalization: The Normal Forms

There are other Normal Forms, but Resolving to 3rd NF is Considered Appropriate



First Normal Form (1NF)

Second Normal Form (2NF)

Third Normal Form (3NF)

- A tables columns must contain only atomic values; they may not contain multiple values
- **Ex:** a column named **telephone_number** may contain only one phone number.

- The table must first satisfy the first normal form.
- The table must be free of partial dependencies; i.e., all columns that are not the Primary Key must depend on the Primary Key

- The table must first satisfy both the first and second normal forms
- The table must be free of transitive dependencies; i.e., no column may depend on any column that is not the Primary Key.

Workload Characteristics: Form Follows Function

Two Essentially Incompatible Workloads… They Have a Contentious Relationship

Online Transaction Processing (OLTP)

- Characterized by a large volume of transactions each of which affect a small number of rows
- Online Sales, Bank Deposits & Transfers
- Highly Normalized Database Schema

Online Analytical Processing (OLAP)

- Characterized by a small volume of read transactions each of which affect a large number of rows
- Periodic Post-hoc Analysis *(What Happened?)*
- De-Normalized Multi-Dimensional Schema

! These two **don't** play well together: They contend for the same hardware resources!

Database Paradigms: Design Approaches

The Design Approach Accommodates the Workload Characteristic

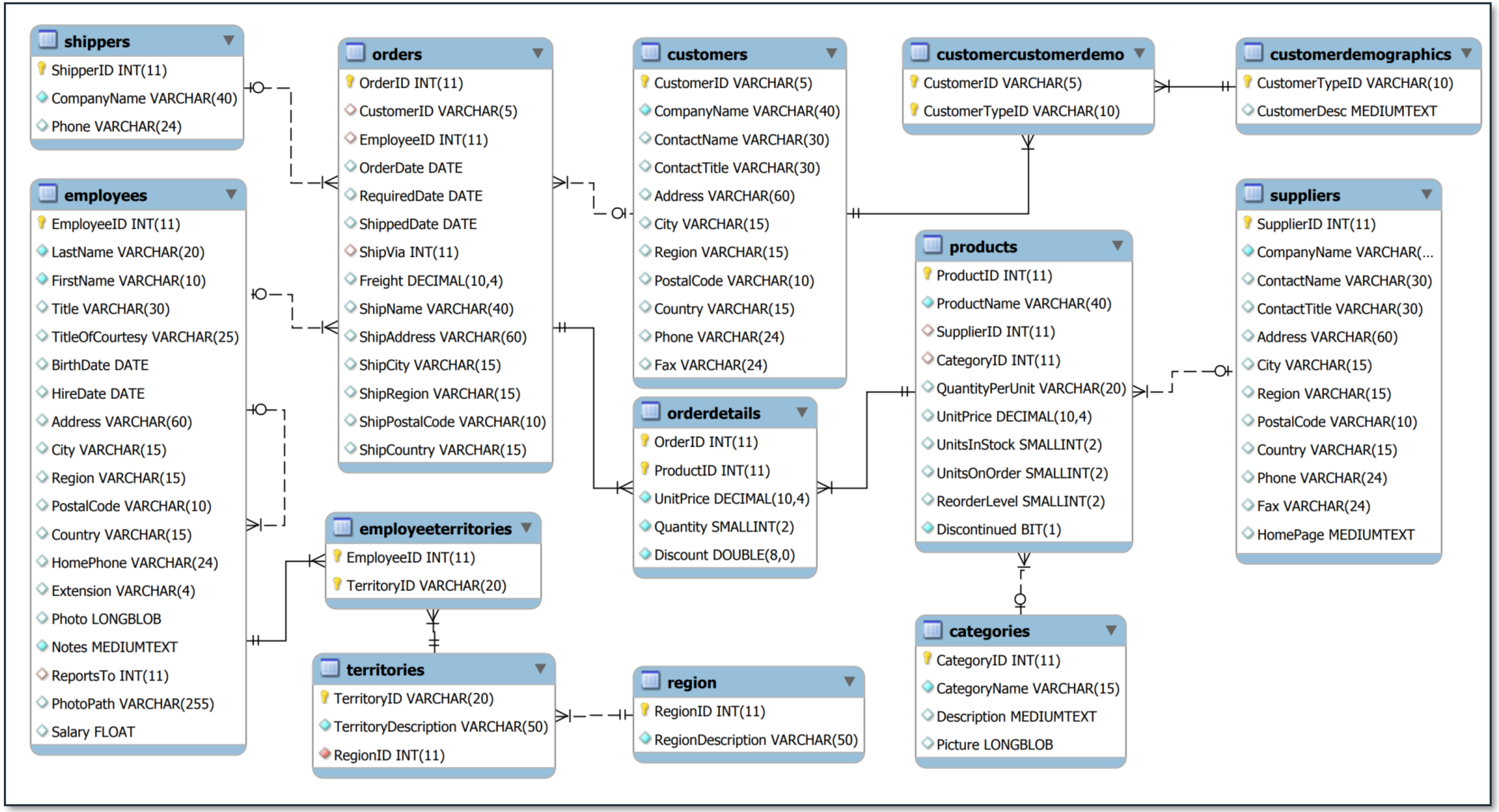
Normalized Relational Database:

- Optimized for Online Transaction Processing (OLTP) workloads
- Aims to Eliminate Data Redundancy and Minimize Storage Requirements
- **Complex:** Sacrifices User-Friendliness in Favor of Transactional Performance

Multi-Dimensional Relational Database:

- Optimized for Online Analytical Processing (OLAP) workloads
- Aims to Optimize Query Performance and Provide an Intuitive User Experience
- **Simple:** Accepts Data Repetition in Favor of User-Friendliness and Improved Query Performance

OLTP Database: Normalized Schema



shippers
ShipperID INT(11)
CompanyName VARCHAR(40)
Phone VARCHAR(24)

employees
EmployeeID INT(11)
LastName VARCHAR(20)
FirstName VARCHAR(10)
Title VARCHAR(30)
TitleOfCourtesy VARCHAR(25)
BirthDate DATE
HireDate DATE
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
HomePhone VARCHAR(24)
Extension VARCHAR(4)
Photo LONGBLOB
Notes MEDIUMTEXT
ReportsTo INT(11)
PhotoPath VARCHAR(255)
Salary FLOAT

orders
OrderID INT(11)
CustomerID VARCHAR(5)
EmployeeID INT(11)
OrderDate DATE
RequiredDate DATE
ShippedDate DATE
ShipVia INT(11)
Freight DECIMAL(10,4)
ShipName VARCHAR(40)
ShipAddress VARCHAR(60)
ShipCity VARCHAR(15)
ShipRegion VARCHAR(15)
ShipPostalCode VARCHAR(10)
ShipCountry VARCHAR(15)

employeeterritories
EmployeeID INT(11)
TerritoryID VARCHAR(20)

territories
TerritoryID VARCHAR(20)
TerritoryDescription VARCHAR(50)
RegionID INT(11)

customers
CustomerID VARCHAR(5)
CompanyName VARCHAR(40)
ContactName VARCHAR(30)
ContactTitle VARCHAR(30)
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
Phone VARCHAR(24)
Fax VARCHAR(24)

orderdetails
OrderID INT(11)
ProductID INT(11)
UnitPrice DECIMAL(10,4)
Quantity SMALLINT(2)
Discount DOUBLE(8,0)

region
RegionID INT(11)
RegionDescription VARCHAR(50)

customercustomerdemo
CustomerID VARCHAR(5)
CustomerTypeID VARCHAR(10)

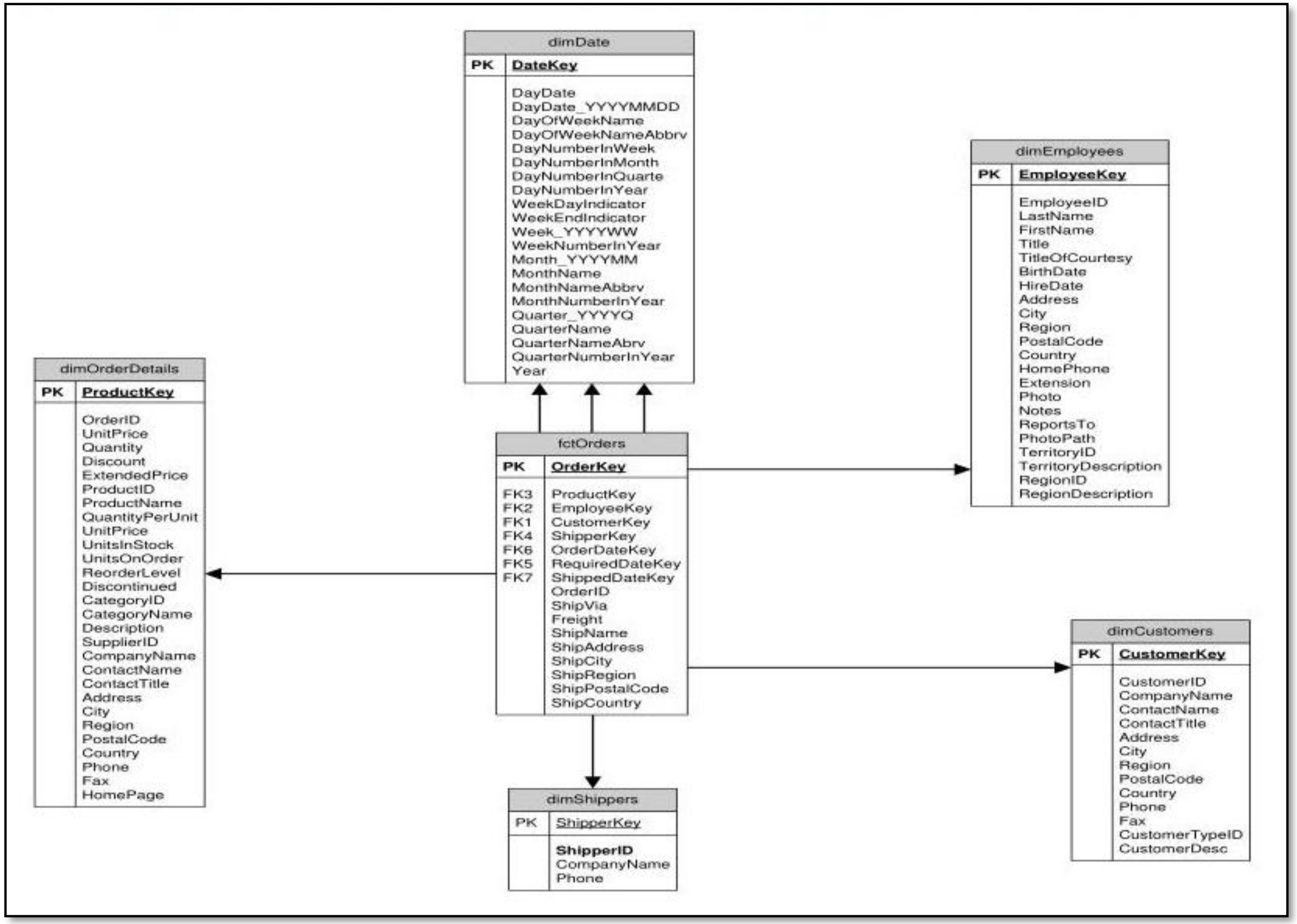
products
ProductID INT(11)
ProductName VARCHAR(40)
SupplierID INT(11)
CategoryID INT(11)
QuantityPerUnit VARCHAR(20)
UnitPrice DECIMAL(10,4)
UnitsInStock SMALLINT(2)
UnitsOnOrder SMALLINT(2)
ReorderLevel SMALLINT(2)
Discontinued BIT(1)

categories
CategoryID INT(11)
CategoryName VARCHAR(15)
Description MEDIUMTEXT
Picture LONGBLOB

customerdemographics
CustomerTypeID VARCHAR(10)
CustomerDesc MEDIUMTEXT

suppliers
SupplierID INT(11)
CompanyName VARCHAR(...
ContactName VARCHAR(30)
ContactTitle VARCHAR(30)
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
Phone VARCHAR(24)
Fax VARCHAR(24)
HomePage MEDIUMTEXT

OLAP Database: Multi-Dimensional (Star) Schema



	dimDate
PK	DateKey
	DayDate DayDate_YYYYMMDD DayOfWeekName DayOfWeekNameAbbrv DayNumberInWeek DayNumberInMonth DayNumberInQuarte DayNumberInYear WeekDayIndicator WeekEndIndicator Week_YYYYWW WeekNumberInYear Month_YYYYMM MonthName MonthNameAbbrv MonthNumberInYear Quarter_YYYYQ QuarterName QuarterNameAbrv QuarterNumberInYear Year

	dimEmployees
PK	EmployeeKey
	EmployeeID LastName FirstName Title TitleOfCourtesy BirthDate HireDate Address City Region PostalCode Country HomePhone Extension Photo Notes ReportsTo PhotoPath TerritoryID TerritoryDescription RegionID RegionDescription

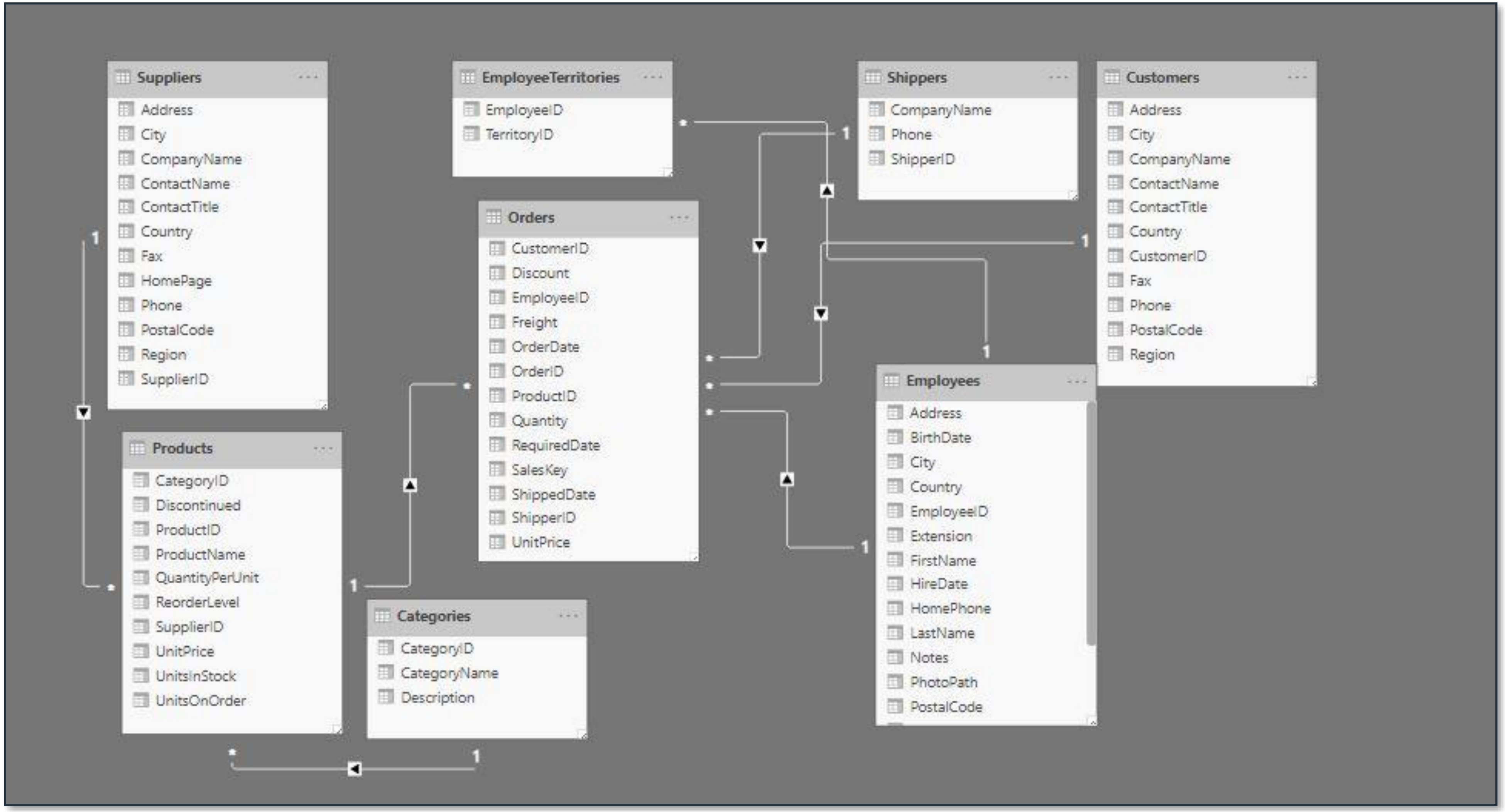
	dimOrderDetails
PK	ProductKey
	OrderID UnitPrice Quantity Discount ExtendedPrice ProductID ProductName QuantityPerUnit UnitPrice UnitsInStock UnitsOnOrder ReorderLevel Discontinued CategoryID CategoryName Description SupplierID CompanyName ContactName ContactTitle Address City Region PostalCode Country Phone Fax HomePage

	fctOrders
PK	OrderKey
FK3	ProductKey
FK2	EmployeeKey
FK1	CustomerKey
FK4	ShipperKey
FK6	OrderDateKey
FK5	RequiredDateKey
FK7	ShippedDateKey
	OrderID ShipVia Freight ShipName ShipAddress ShipCity ShipRegion ShipPostalCode ShipCountry

	dimCustomers
PK	CustomerKey
	CustomerID CompanyName ContactName ContactTitle Address City Region PostalCode Country Phone Fax CustomerTypeID CustomerDesc

	dimShippers
PK	ShipperKey
	ShipperID CompanyName Phone

OLAP Database: Snowflake Schema



Suppliers
- Address
- City
- CompanyName
- ContactName
- ContactTitle
- Country
- Fax
- HomePage
- Phone
- PostalCode
- Region
- SupplierID

EmployeeTerritories
- EmployeeID
- TerritoryID

Shippers
- CompanyName
- Phone
- ShipperID

Customers
- Address
- City
- CompanyName
- ContactName
- ContactTitle
- Country
- CustomerID
- Fax
- Phone
- PostalCode
- Region

Orders
- CustomerID
- Discount
- EmployeeID
- Freight
- OrderDate
- OrderID
- ProductID
- Quantity
- RequiredDate
- SalesKey
- ShippedDate
- ShipperID
- UnitPrice

Products
- CategoryID
- Discontinued
- ProductID
- ProductName
- QuantityPerUnit
- ReorderLevel
- SupplierID
- UnitPrice
- UnitsInStock
- UnitsOnOrder

Categories
- CategoryID
- CategoryName
- Description

Employees
- Address
- BirthDate
- City
- Country
- EmployeeID
- Extension
- FirstName
- HireDate
- HomePhone
- LastName
- Notes
- PhotoPath
- PostalCode

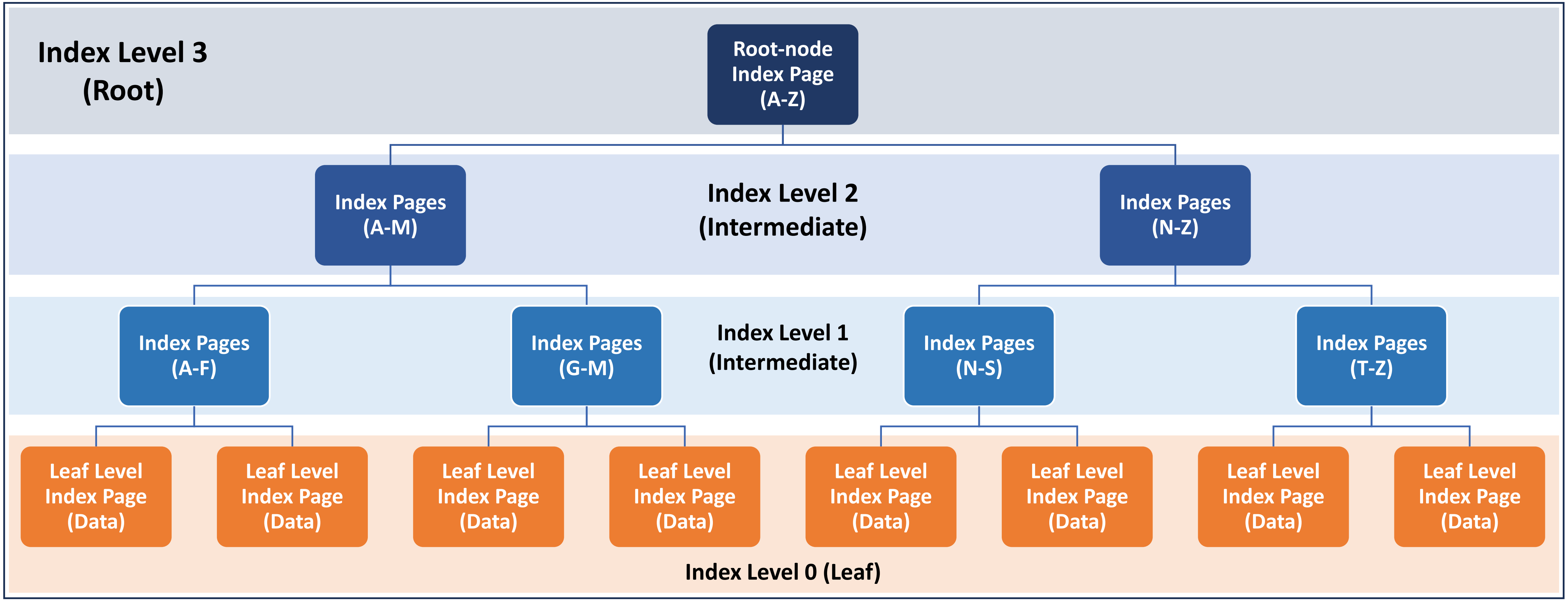
Database Management Server (DBMS) Internal Structures

Understanding the System Constructs that Optimize Data Storage and Retrieval

UNIVERSITY of VIRGINIA

BTree (Balanced Tree): Physical Storage Structure

Organizes Data to Enable Applying Binary Logic for Quickly Locating Data on Disk


Index Level 3
(Root)
Root-node Index Page (A-Z)
Index Level 2 (Intermediate)
Index Pages (A-M)
Index Pages (N-Z)
Index Level 1 (Intermediate)
Index Pages (A-F)
Index Pages (G-M)
Index Pages (N-S)
Index Pages (T-Z)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Leaf Level Index Page (Data)
Index Level 0 (Leaf)

** Conceptual Representation of A BTree Index*

Indexes: Quickly Finding a Needle in a Haystack

Physical Data Constructs that Optimize Locating Data on Storage Subsystems

Clustered Indexes

- BTree (Balanced Tree) Structure that determines the physical order of data in a table according to the Primary Key.
- Row-wise Structure; i.e., identifies the row(s) in which search terms exists.
- Analogous to a Dictionary, where each word is stored in alphabetical order with words categorized into sections according to their first letter (e.g., A, B).

Non-Clustered Indexes

- References the Clustered Index to optimize data retrieval without altering the physical order of the data.
- Row-wise Structure; i.e., identifies the row(s) in which search terms exists.
- Analogous to an Index located near the end of a textbook, where references to pages containing specific keywords are identified; organized alphabetically.

Transactions: Maintaining Database Integrity

The **ACID** Properties of a Transaction Must be Enforced to Ensure the Integrity of a Database

Atomicity

- All operations in a transaction must be treated as a single [atomic] unit, and must either be fully committed (i.e., succeed) or be rolled-back (i.e., have no effect whatsoever).

Consistency

- Each transaction must guarantee the valid state of the database; both before and after the transaction is either committed or rolled-back.
- All defined rules, constraints, and relationships must be enforced and maintained.

Isolation

- Concurrent transactions must be completely independent; having no effect on each other.
- Transactions must remain invisible to each other until they are either committed or rolled-back.

Durability

- Once committed, the effects of each transaction must be permanently saved (written to disk); even if the system fails amidst their execution.
- A transaction log (aka, Journal file) serves as a write-ahead mechanism to ensure each committed transaction can be replayed to successfully recover the database to a consistent state in the event of a system failure.

The SQL Language

Understanding the Structured Query Language

UNIVERSITY of VIRGINIA

The SQL Language: Principal Components

Three Primary Aspects of the ANSI-Compliant SQL Language

Data Definition Language (DDL)

- CREATE, ALTER, DROP, TRUNCATE TABLE, ENABLE & DISABLE TRIGGER
- *Used to manage database structures*

Data Control Language (DCL)

- GRANT, REVOKE, DENY, EXECUTE AS
- *Used to control access to server & database objects (permissions)*

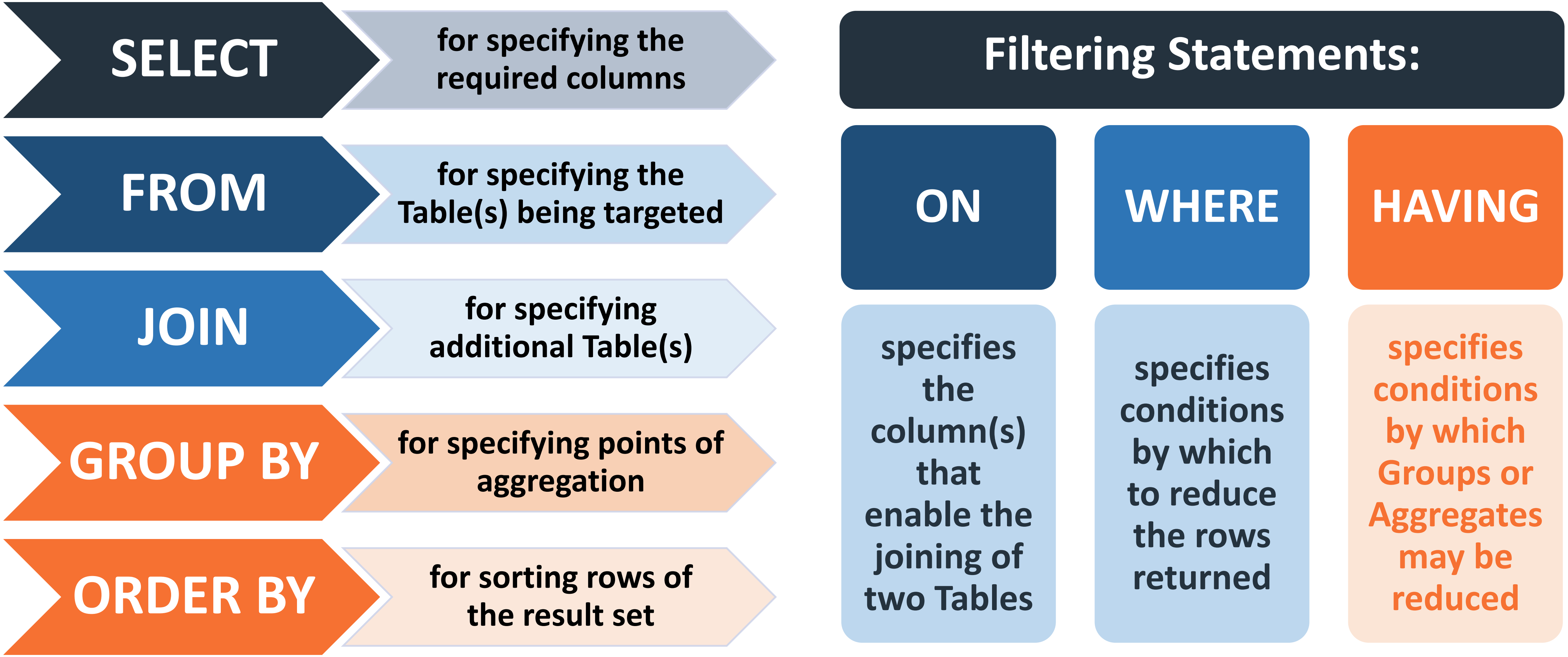
Data Manipulation Language (DML)

- SELECT, INSERT, UPDATE, DELETE, MERGE, and BULK INSERT
- *Used to manipulate database content (data)*

[Microsoft Docs | Transact-SQL Reference (Database Engine)]

Query a SQL Database: The SELECT Statement

Essential Components of Data Retrieval



SELECT for specifying the required columns

FROM for specifying the Table(s) being targeted

JOIN for specifying additional Table(s)

GROUP BY for specifying points of aggregation

ORDER BY for sorting rows of the result set

Filtering Statements:

ON specifies the column(s) that enable the joining of two Tables

WHERE specifies conditions by which to reduce the rows returned

HAVING specifies conditions by which Groups or Aggregates may be reduced

Q & A

An Overview of SQL Databases