Module 5: Implementing Data Integrity
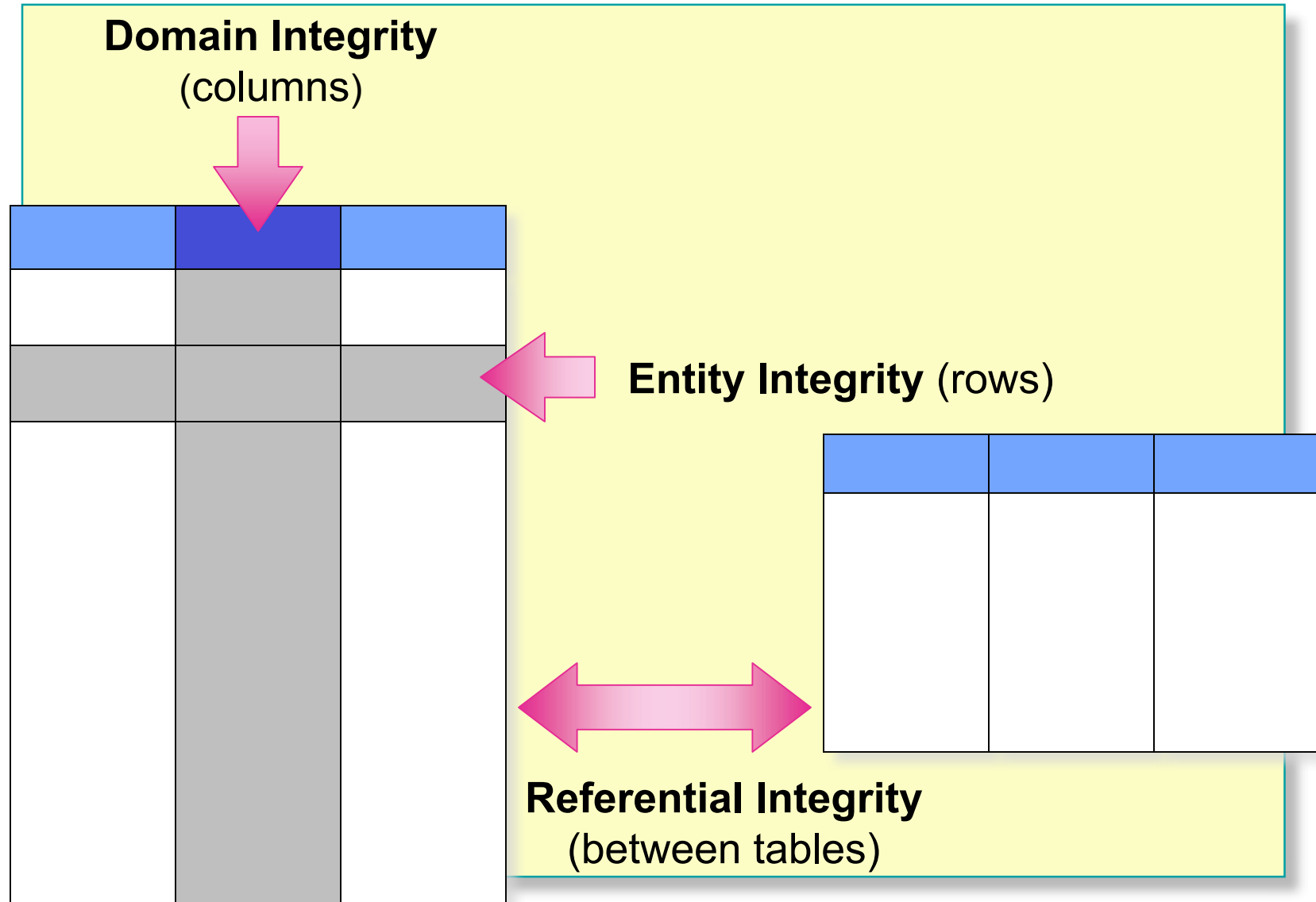
Overview

- **Types of Data Integrity**

- **Enforcing Data Integrity**

- **Defining Constraints**

- **Types of Constraints**

- **Disabling Constraints**

- **Using Defaults and Rules**

- **Deciding Which Enforcement Method to Use**

Types of Data Integrity

Domain Integrity (columns)

Entity Integrity (rows)

Referential Integrity (between tables)

Enforcing Data Integrity

- **Declarative Data Integrity**

  - Criteria defined in object definitions

  - SQL Server enforces automatically

  - Implement by using constraints, defaults, and rules

- **Procedural Data Integrity**

  - Criteria defined in script

  - Script enforces

  - Implement by using triggers and stored procedures

◆ Defining Constraints

- **Determining Which Type of Constraint to Use**

- **Creating Constraints**

- **Considerations for Using Constraints**

Determining Which Type of Constraint to Use

Type of integrity	Constraint type
Domain	DEFAULT
	CHECK
	REFERENTIAL
Entity	PRIMARY KEY
	UNIQUE
Referential	FOREIGN KEY
	CHECK

Creating Constraints

- **Use CREATE TABLE or ALTER TABLE**

- **Can Add Constraints to a Table with Existing Data**

- **Can Place Constraints on Single or Multiple Columns**

  - Single column, called column-level constraint

  - Multiple columns, called table-level constraint

Considerations for Using Constraints

- **Can Be Changed Without Recreating a Table**

- **Require Error-Checking in Applications and Transactions**

- **Verify Existing Data**

◆ Types of Constraints

- DEFAULT Constraints

- CHECK Constraints

- PRIMARY KEY Constraints

- UNIQUE Constraints

- FOREIGN KEY Constraints

- Cascading Referential Integrity

DEFAULT Constraints

- **Apply Only to INSERT Statements**

- **Only One DEFAULT Constraint Per Column**

- **Cannot Be Used with IDENTITY Property or rowversion Data Type**

- **Allow Some System-supplied Values**

```
USE Northwind
ALTER TABLE dbo.Customers
ADD
CONSTRAINT DF_contactname DEFAULT 'UNKNOWN'
FOR ContactName
```

CHECK Constraints

- **Are Used with INSERT and UPDATE Statements**

- **Can Reference Other Columns in the Same Table**

- **Cannot:**

  - Be used with the **rowversion** data type

  - Contain subqueries

```
USE Northwind
ALTER TABLE dbo.Employees
ADD
CONSTRAINT CK_birthdate
CHECK (BirthDate > '01-01-1900' AND BirthDate <
getdate())
```

PRIMARY KEY Constraints

- **Only One PRIMARY KEY Constraint Per Table**

- **Values Must Be Unique**

- **Null Values Are Not Allowed**

- **Creates a Unique Index on Specified Columns**

```
USE Northwind
ALTER TABLE dbo.Customers
ADD
CONSTRAINT PK_Customers
   PRIMARY KEY NONCLUSTERED (CustomerID)
```

UNIQUE Constraints

- **Allow One Null Value**

- **Allow Multiple UNIQUE Constraints on a Table**

- **Defined with One or More Columns**

- **Enforced with a Unique Index**

```
USE Northwind
ALTER TABLE dbo.Suppliers
ADD
CONSTRAINT U_CompanyName
   UNIQUE NONCLUSTERED (CompanyName)
```

FOREIGN KEY Constraints

- **Must Reference a PRIMARY KEY or UNIQUE Constraint**

- **Provide Single or Multicolumn Referential Integrity**

- **Do Not Automatically Create Indexes**

- **Users Must Have SELECT or REFERENCES Permissions on Referenced Tables**

- **Use Only REFERENCES Clause Within Same Table**

```
USE Northwind
ALTER TABLE dbo.Orders
ADD CONSTRAINT FK_Orders_Customers
   FOREIGN KEY (CustomerID)
   REFERENCES dbo.Customers(CustomerID)
```

Cascading Referential Integrity

NO ACTION

Customers

CustomerID (PK)	

1 INSERT new CustomerID

Orders

CustomerID (FK)	

2 UPDATE old CustomerID to new CustomerID

Customers

CustomerID (PK)	

3 DELETE old CustomerID

CASCADE

Customers

CustomerID (PK)	

1 UPDATE CustomerID

CASCADE

Orders

CustomerID (FK)	

◆ Disabling Constraints

- Disabling Constraint Checking on Existing Data

- Disabling Constraint Checking When Loading New Data

Disabling Constraint Checking on Existing Data

- **Applies to CHECK and FOREIGN KEY Constraints**

- **Use WITH NOCHECK Option When Adding a New Constraint**

- **Use if Existing Data Will Not Change**

- **Can Change Existing Data Before Adding Constraints**

```
USE Northwind
ALTER TABLE dbo.Employees
WITH NOCHECK
   ADD CONSTRAINT FK_Employees_Employees
   FOREIGN KEY (ReportsTo)
   REFERENCES dbo.Employees(EmployeeID)
```

Disabling Constraint Checking When Loading New Data

- **Applies to CHECK and FOREIGN KEY Constraints**

- **Use When:**

  - Data conforms to constraints

  - You load new data that does not conform to constraints

```
USE Northwind
ALTER TABLE dbo.Employees
NOCHECK
   CONSTRAINT FK_Employees_Employees
```

Using Defaults and Rules

- **As Independent Objects They:**
  - Are defined once
  - Can be bound to one or more columns or user-defined data types

```
CREATE DEFAULT phone_no_default
  AS '(000)000-0000'
GO
EXEC sp_bindefault phone_no_default,
  'Customers.Phone'
```

```
CREATE RULE regioncode_rule
  AS @regioncode IN ('IA', 'IL', 'KS', 'MO')
GO
EXEC sp_bindrule regioncode_rule,
  'Customers.Region'
```

Deciding Which Enforcement Method to Use

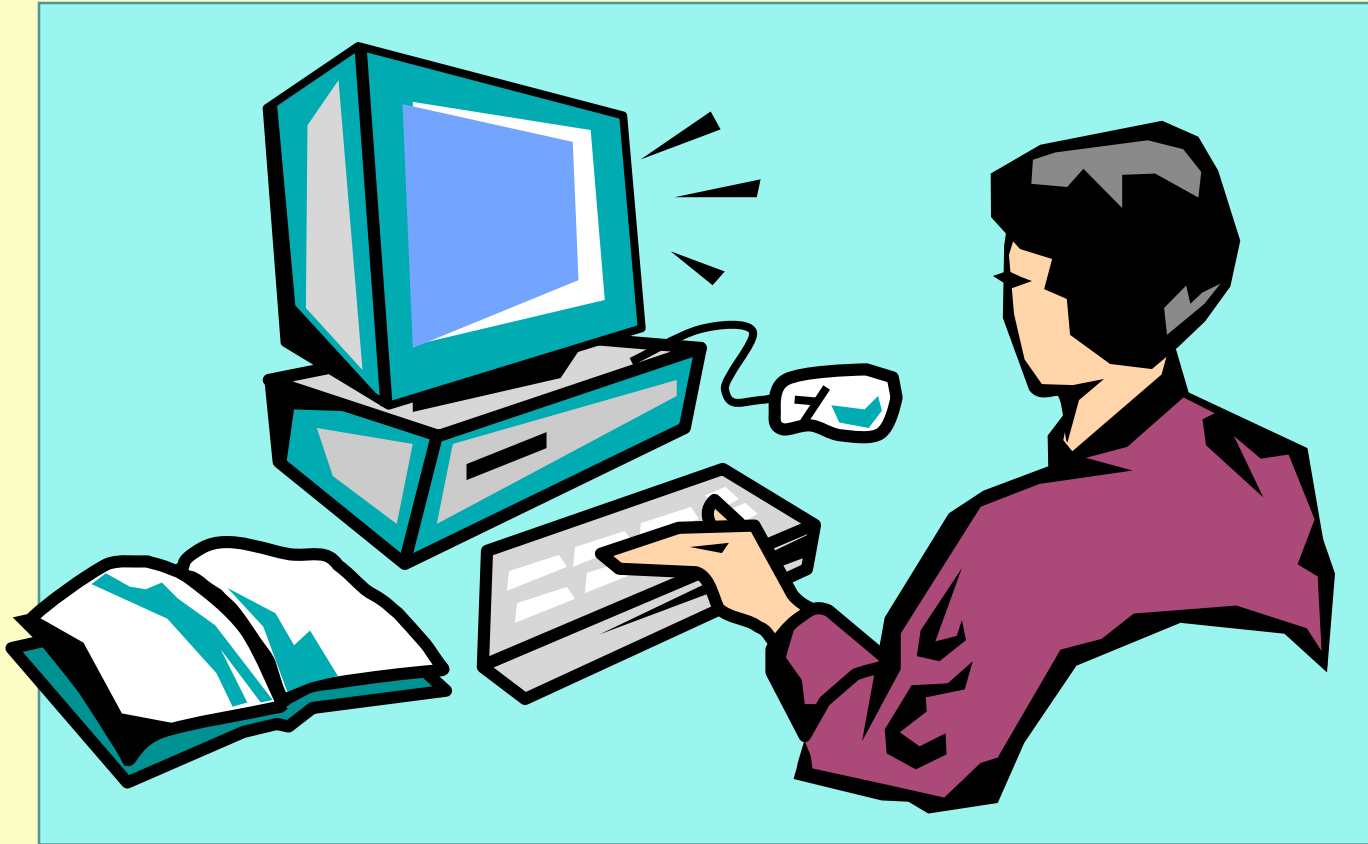| Data integrity components	Functionality	Performance costs	Before or after modification
Constraints	Medium	Low	Before
Defaults and rules	Low	Low	Before
Triggers	High	Medium-High	After
Data types, Null/Not Null	Low	Low	Before

Recommended Practices

☑ **Use Constraints Because They Are ANSI-compliant**

☑ **Use Cascading Referential Integrity Instead of Triggers**

Lab A: Implementing Data Integrity

Review

- **Types of Data Integrity**

- **Enforcing Data Integrity**

- **Defining Constraints**

- **Types of Constraints**

- **Disabling Constraints**

- **Using Defaults and Rules**

- **Deciding Which Enforcement Method to Use**