
# Towards the .NET Junior Developer

The extremely solid course

# Lesson 6

Databases and SQL basics

# Agenda

- Database development
  - Database structure
  - Common database operations
  - Tables
  - Keys
  - Views
  - Stored procedures, functions
  - Transactions
  - Indexes
- T-SQL basics
  - SELECT
  - WHERE
  - GROUP BY
  - JOIN
  - UNION
- Books of the day
- Links of the day
- Hometask

# Database development

# Database development – database structure

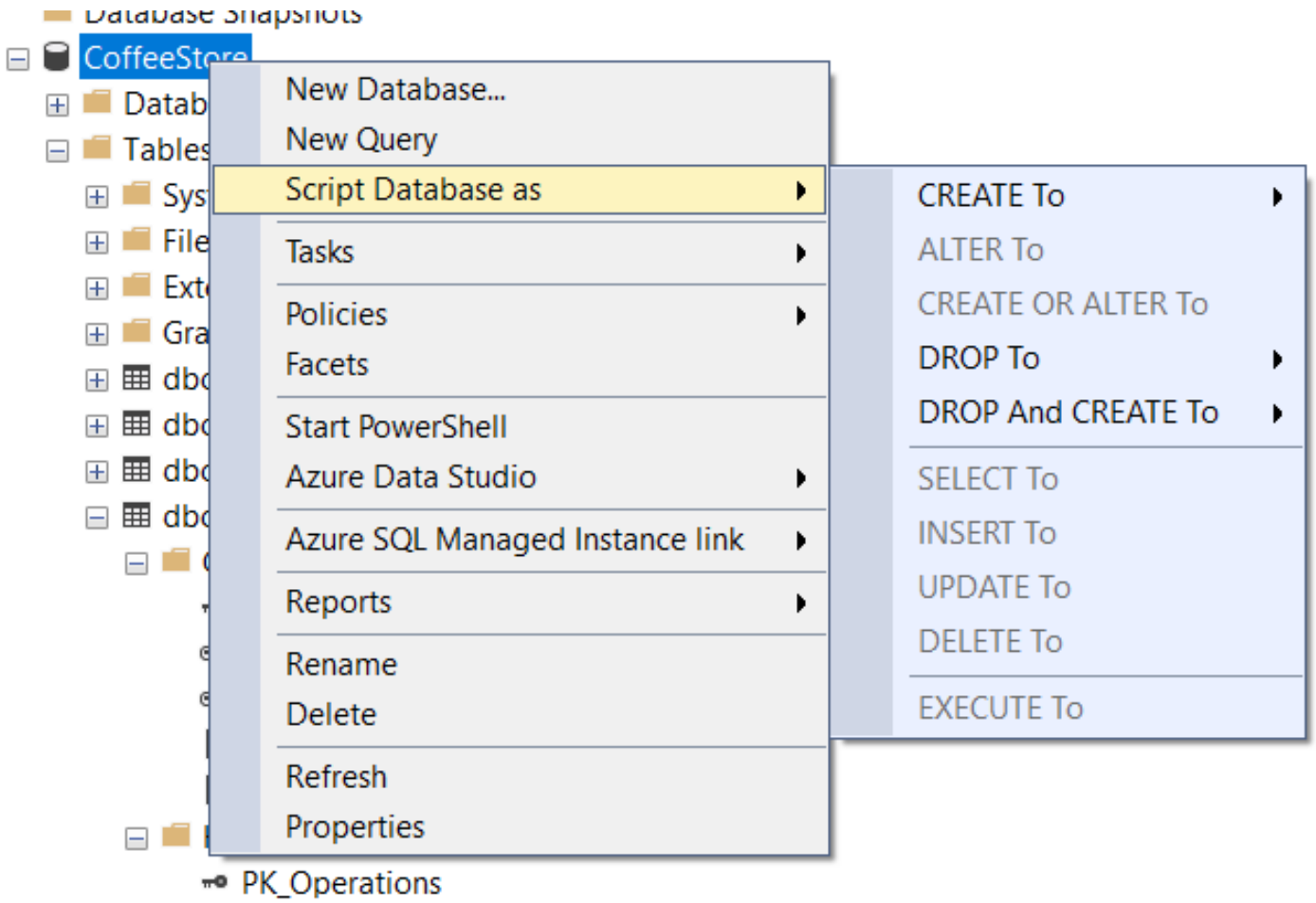
- CoffeeStore
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
    - dbo.Clients
    - dbo.PropertyEnums
    - dbo.PropertyEnumValues
    - dbo.Operations
    - dbo.OperationType
    - dbo.OrderDetails
    - dbo.Orders
    - dbo.OrderStatuses
    - dbo.Origins
    - dbo.ProductProperties
    - dbo.ProductPropertyValues
    - dbo.Products
    - dbo.ProductTypes
  - Views
  - External Resources
  - Synonyms
  - Programmability
  - Service Broker
  - Storage
  - Security

system tables

user-defined tables

additional stuff

# Database development – common database operations

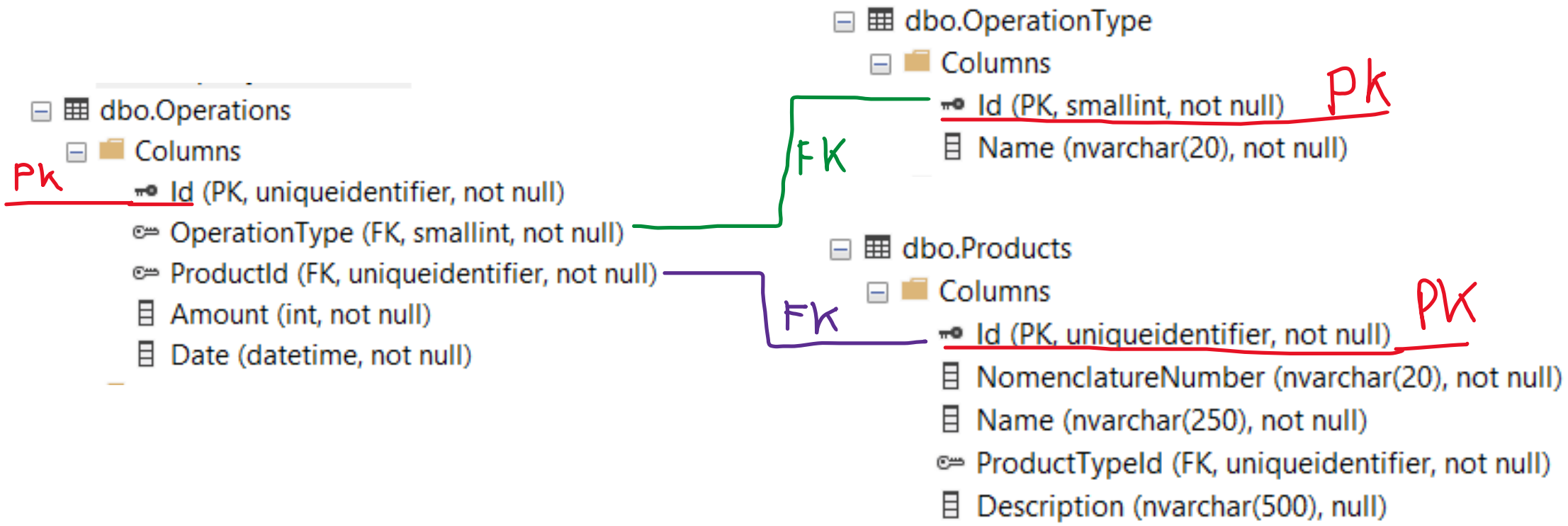


# Database development – tables

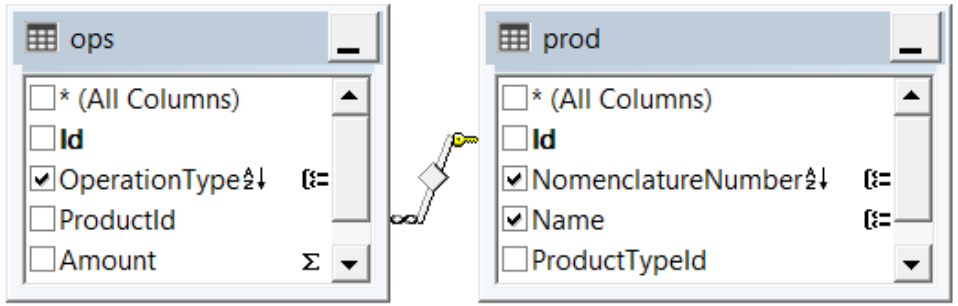
- dbo.Operations
  - Columns
    - Id (PK, uniqueidentifier, not null)
    - OperationType (FK, smallint, not null)
    - ProductId (FK, uniqueidentifier, not null)
    - Amount (int, not null)
    - Date (datetime, not null)
  - Keys
    - PK_Operations
    - FK_Operations_Operations
    - FK_Operations_Products
  - Constraints
  - Triggers
  - Indexes
    - idx__operations__operation_type (Non-Unique, Non-Clustered)
    - idx__operations_product_id (Non-Unique, Non-Clustered)
    - PK_Operations (Clustered)
  - Statistics

```sql
CREATE TABLE [dbo].[Operations](
    [Id] [uniqueidentifier] NOT NULL,
    [OperationType] [smallint] NOT NULL,
    [ProductId] [uniqueidentifier] NOT NULL,
    [Amount] [int] NOT NULL,
    [Date] [datetime] NOT NULL,
 CONSTRAINT [PK_Operations] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Operations]
WITH CHECK ADD  CONSTRAINT [FK_Operations_Operations]
FOREIGN KEY([OperationType])
REFERENCES [dbo].[OperationType] ([Id])
GO

ALTER TABLE [dbo].[Operations]
CHECK CONSTRAINT [FK_Operations_Operations]
GO

ALTER TABLE [dbo].[Operations]
WITH CHECK ADD  CONSTRAINT [FK_Operations_Products]
FOREIGN KEY([ProductId])
REFERENCES [dbo].[Products] ([Id])
GO

ALTER TABLE [dbo].[Operations]
CHECK CONSTRAINT [FK_Operations_Products]
GO
```

# Database development – keys



# Database development – views




| | Column | Alias | Table | Output | Sort Type | Sort Order | Group By | Filter |
|---|---|---|---|---|---|---|---|---|
| ▸ | Name | | prod | ☑ | | | Group By | |
| | Nomenclature... | | prod | ☑ | Ascending | 1 | Group By | |
| | OperationType | | ops | ☑ | Ascending | 2 | Group By | |
| | Amount | Amount | ops | ☑ | | | Sum | |
| | | | | ☐ | | | | |

```
SELECT TOP (100) PERCENT prod.Name, prod.NomenclatureNumber, ops.OperationType, SUM(ops.Amount) AS Amount
FROM     dbo.Operations AS ops INNER JOIN
             dbo.Products AS prod ON ops.ProductId = prod.Id
GROUP BY prod.NomenclatureNumber, prod.Name, ops.OperationType
ORDER BY prod.NomenclatureNumber, ops.OperationType
```

# Database development – stored procedures, functions

```
USE [CoffeeStore]
GO
/****** Object:  StoredProcedure [dbo].[sp_setOrderStatus]    Script Date: 8/19/2022 10:42:22 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[sp_setOrderStatus] @orderId uniqueidentifier, @status smallint
AS
BEGIN
    UPDATE dbo.Orders
    SET OrderStatus = @status
    WHERE Id = @orderId
END
```

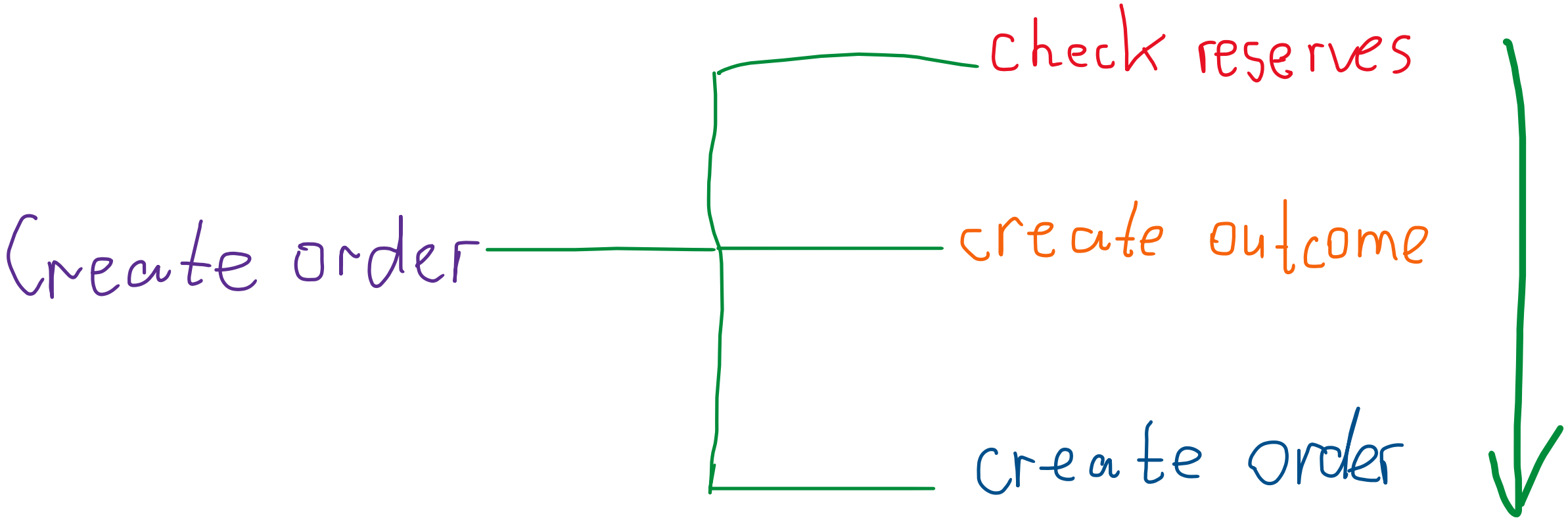
# Database development – stored procedures, functions

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION fn_getClientUnfinishedOrders
(
    @clientId uniqueidentifier
)
RETURNS TABLE
AS
RETURN
(
    SELECT Id AS OrderId FROM dbo.Orders
    WHERE ClientId = @clientId AND OrderStatus < 2
)
GO
```

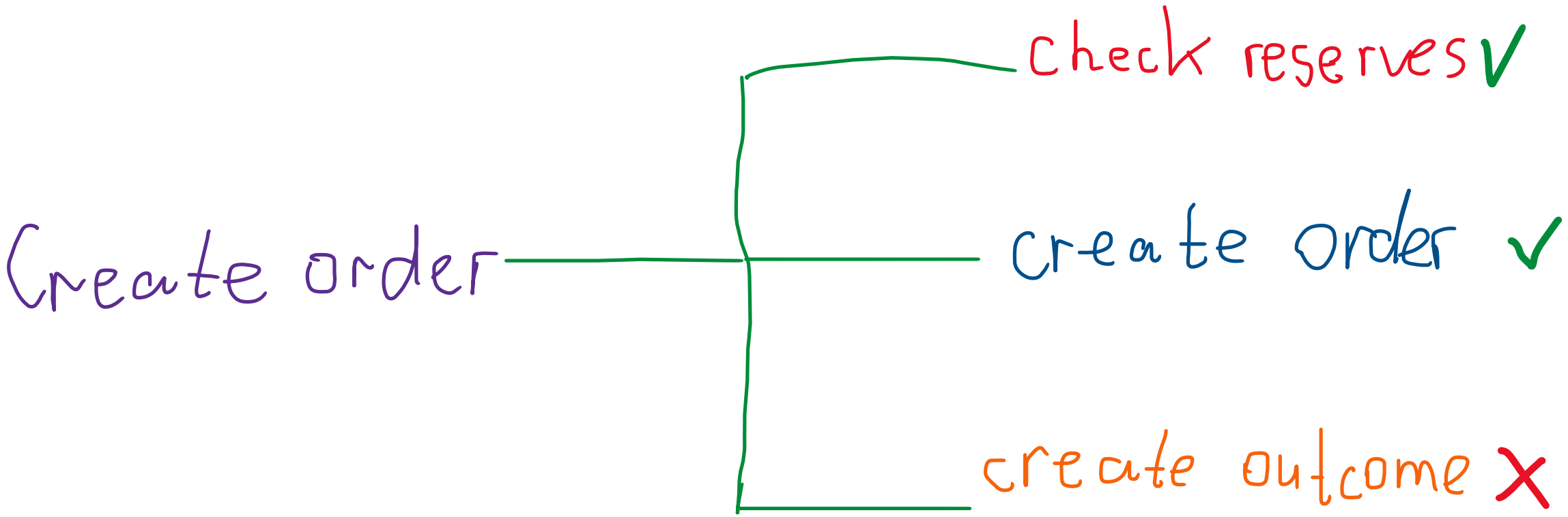
# Database development – stored procedures, functions

| Stored Procedure (SP) | Function (UDF - User Defined) |
|---|---|
| SP can return zero, single or multiple values. | Function must return a single value (which may be a scalar or a table). |
| We can use transaction in SP. | We can't use transaction in UDF. |
| SP can have input/output parameter. | Only input parameter. |
| We can call function from SP. | We can't call SP from function. |
| We can't use SP in SELECT/ WHERE/ HAVING statement. | We can use UDF in SELECT/ WHERE/ HAVING statement. |
| We can use exception handling using Try-Catch block in SP. | We can't use Try-Catch block in UDF. |

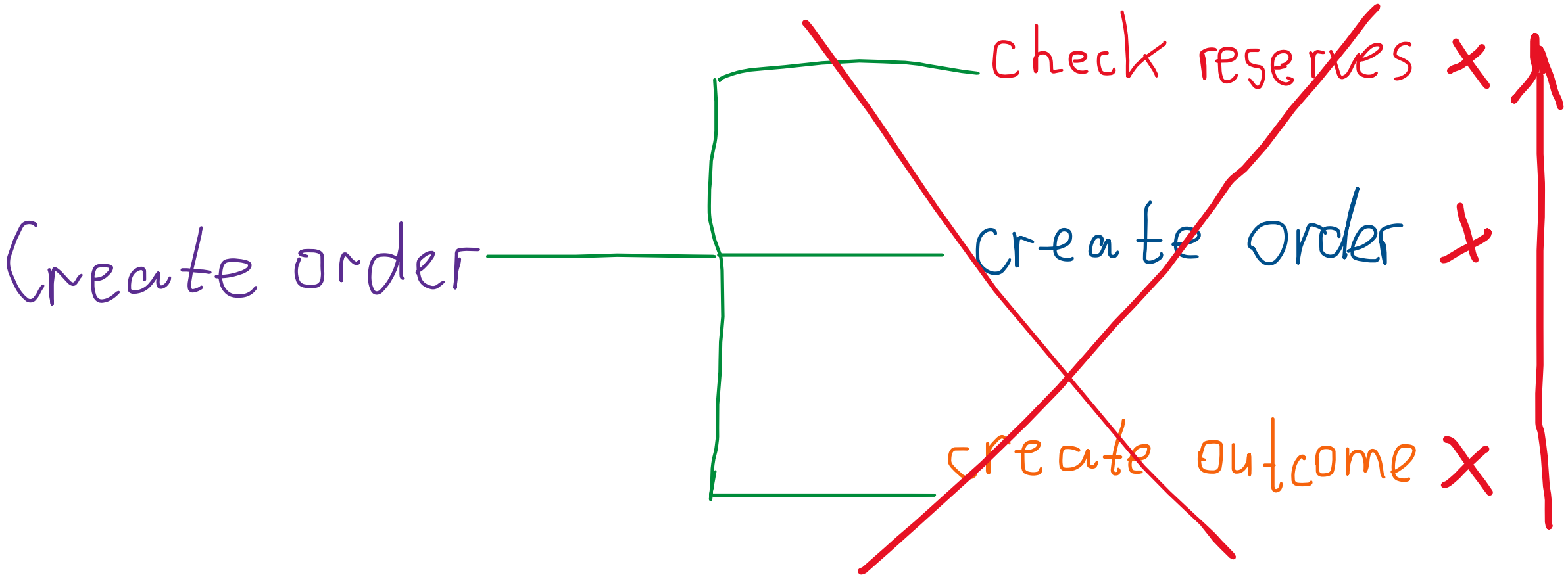
# Database development – transactions



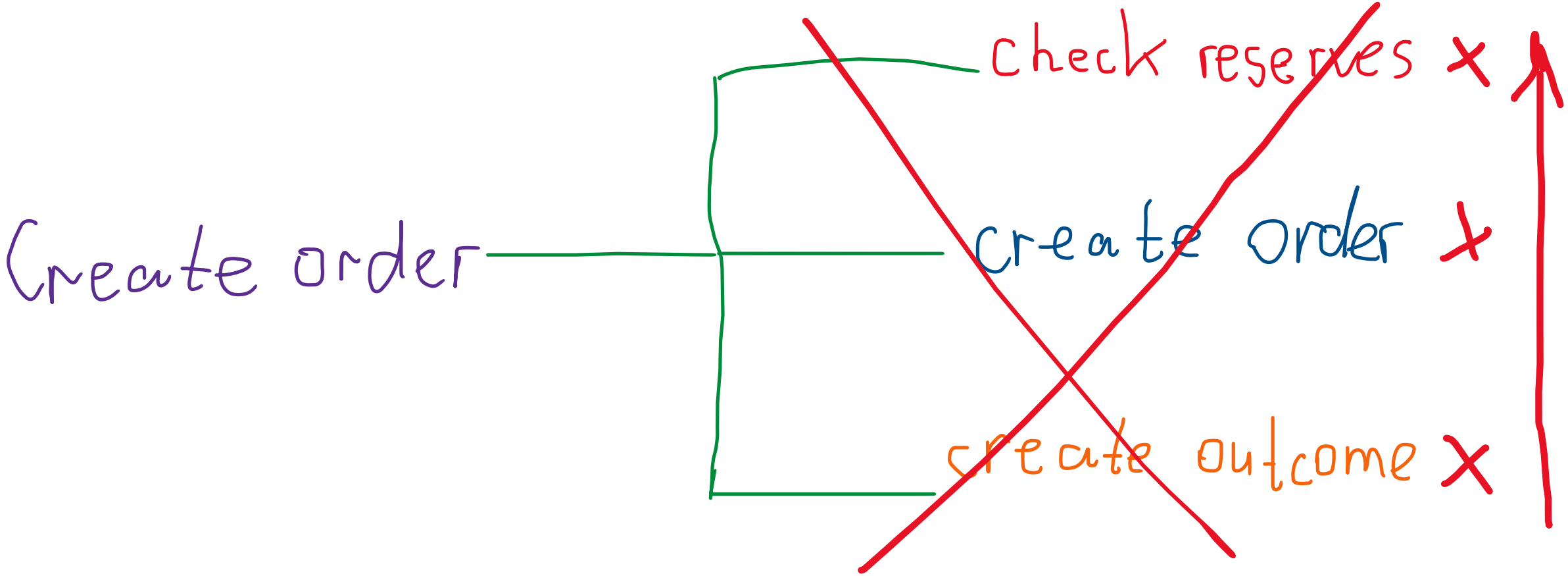
# Database development – transactions



# Database development – transactions



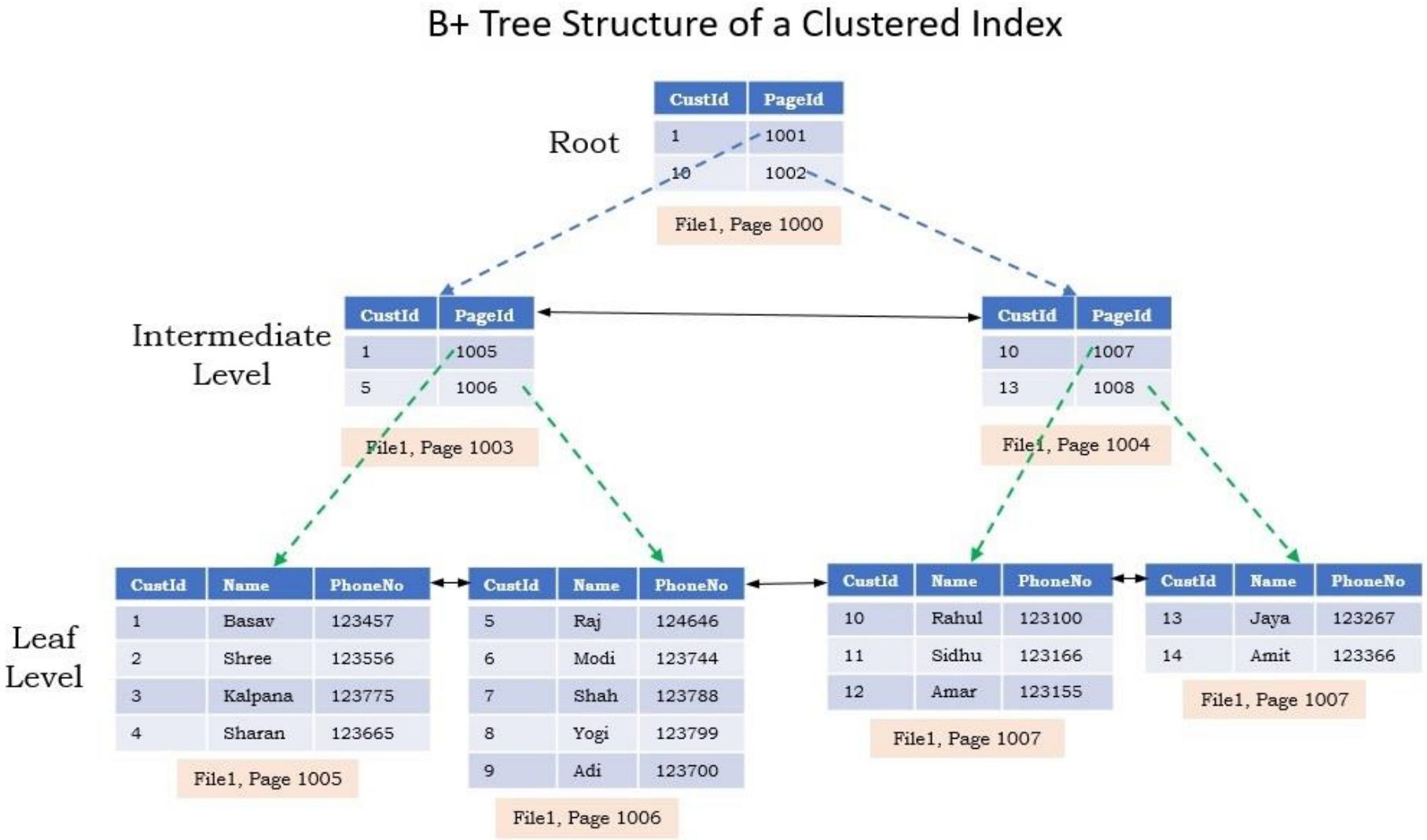
# Database development – transactions



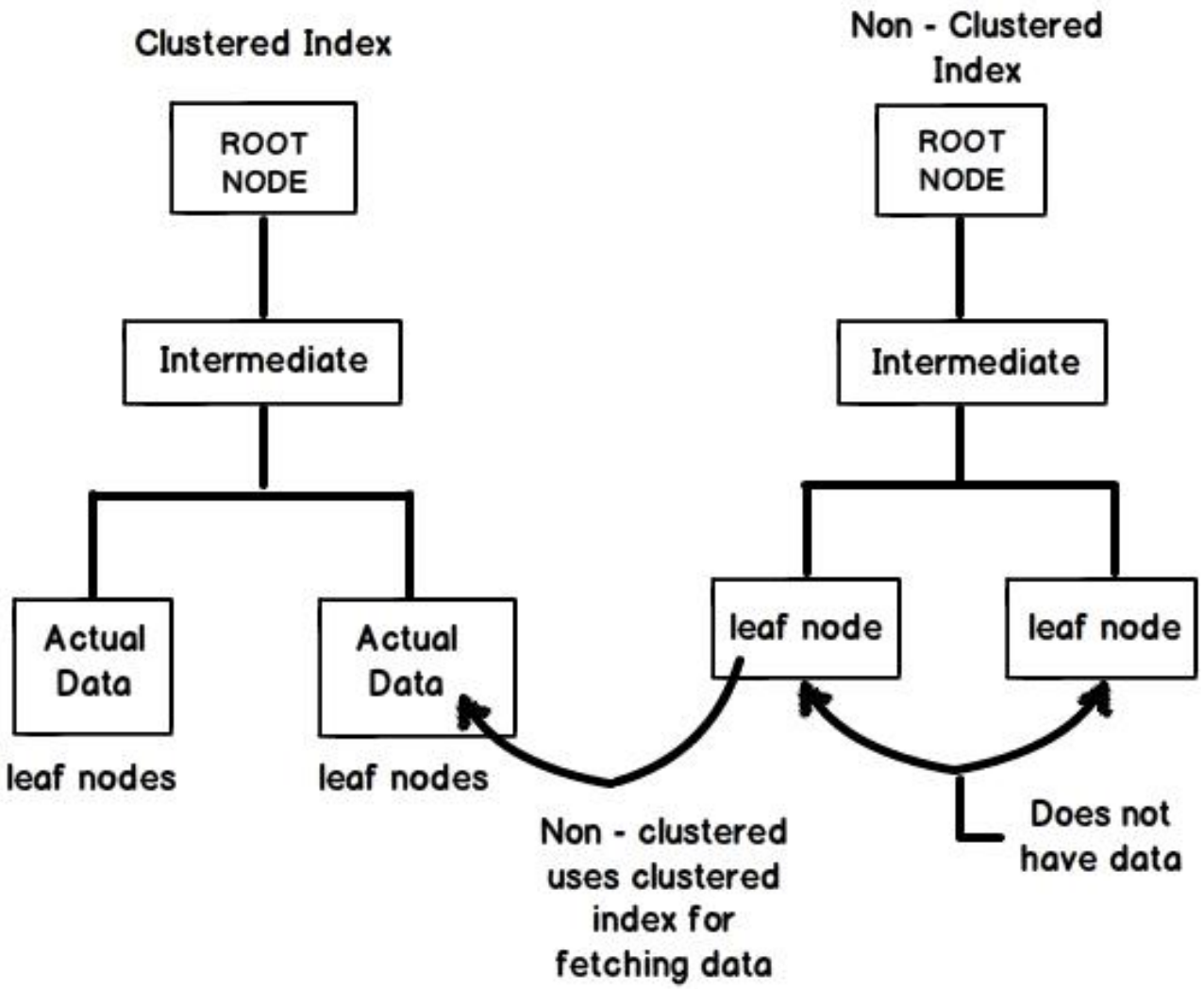
# Database development – transactions

```
BEGIN TRY
    DECLARE @availableAmount INT = dbo.fn_getReservesForProduct(@productId)

    IF @availableAmount <= @amount
        RETURN

    BEGIN TRANSACTION
        SET @orderId = NEWID()
        DECLARE @now DATETIME2(7) = GETDATE()

        INSERT INTO dbo.Orders(Id, ClientId, OrderStatus, CreatedDate, ModifiedDate)
        VALUES(@orderId, @clientId, 0, @now, @now)

        INSERT INTO dbo.OrderDetails(Id, OrderId, ProductId, Amount, Sum)
        VALUES(NEWID(), @orderId, @productId, @amount, @sum)
    COMMIT
END TRY

BEGIN CATCH
    SET @orderId = NULL
    ROLLBACK
END CATCH
```

# Database development – indexes



# Database development – indexes



# Database development – indexes

```
ALTER TABLE [dbo].[Operations] ADD  CONSTRAINT [PK_Operations] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF
GO
```

```
CREATE NONCLUSTERED INDEX [idx__operations__operation_type] ON [dbo].[Operations]
(
    [OperationType] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF
GO
```

# Database development demo

# T-SQL basics

# T-SQL basics - SELECT

```sql
SELECT
    [Id] AS ProductId
    ,[NomenclatureNumber] AS NN
    ,[Name] AS ProductName
    ,[ProductTypeId]
    ,[Description]
FROM [CoffeeStore].[dbo].[Products]
```

5 %

Results | Messages

| ProductId | NN | ProductName | ProductTypeId | Description |
|---|---|---|---|---|
| 88A11753-DC9D-4899-A7E9-BD0EA580F25A | 0001-1200 | Sunshine Mexico | 0C57276D-022F-4983-A175-550E19B17318 | Mexican Arabica with the fruit notes |
| 73D83D32-3BAF-4C78-BCBE-C628151524A3 | 0020-1445 | Barista FN-150 | 3C910B0F-E313-4111-BB4F-D997332DFFF7 | Smart coffee machine for the home usage |

# T-SQL basics - WHERE

```sql
SELECT
    [Id] AS ProductId
    ,[NomenclatureNumber] AS NN
    ,[Name] AS ProductName
    ,[ProductTypeId]
    ,[Description]
FROM [CoffeeStore].[dbo].[Products]
WHERE ProductTypeId = '0c57276d-022f-4983-a175-550e19b17318'
```

46 %

Results | Messages

| ProductId | NN | ProductName | ProductTypeId | Description |
|---|---|---|---|---|
| 88A11753-DC9D-4899-A7E9-BD0EA580F25A | 0001-1200 | Sunshine Mexico | 0C57276D-022F-4983-A175-550E19B17318 | Mexican Arabica with the fruit notes |

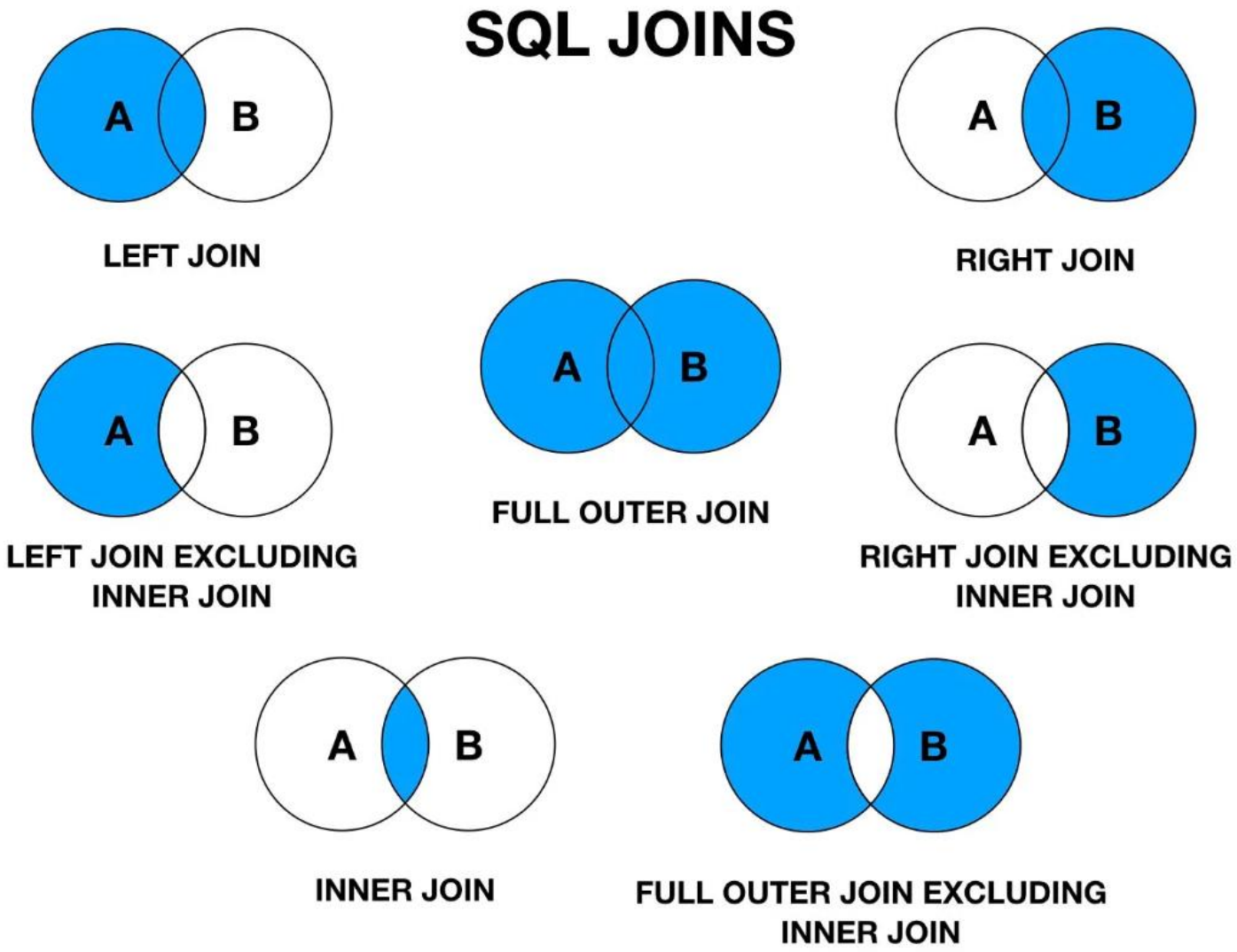
# T-SQL basics – GROUP BY

```sql
SELECT OrderStatus AS Status, COUNT(1) AS OrdersNumber
FROM [CoffeeStore].[dbo].[Orders]
GROUP BY OrderStatus
```

Results | Messages

| Status | OrdersNumber |
|---|---|
| 2 | 2 |
| 3 | 1 |

# T-SQL basics - JOIN



# T-SQL basics - JOIN

```sql
SELECT
    [NomenclatureNumber]
    ,[Name]
    ,[Description]
FROM [CoffeeStore].[dbo].[Products]
```

%

Results | Messages

| NomenclatureNumber | Name | Description |
|---|---|---|
| 0001-1200 | Sunshine Mexico | Mexican Arabica with the fruit notes |
| 0020-1445 | Barista FN-150 | Smart coffee machine for the home usage |
| 0001-1320 | Sunshine Columbia | Columbian magic with the chocolate notes |

```sql
SELECT
    [ProductId]
    ,[Amount]
    ,[Sum]
FROM [CoffeeStore].[dbo].[OrderDetails]
```

5 %

Results | Messages

| ProductId | Amount | Sum |
|---|---|---|
| 73D83D32-3BAF-4C78-BCBE-C628151524A3 | 1 | 17500.00 |
| 88A11753-DC9D-4899-A7E9-BD0EA580F25A | 2 | 2400.00 |
| 88A11753-DC9D-4899-A7E9-BD0EA580F25A | 4 | 4800.00 |

# T-SQL basics – INNER JOIN

```sql
SELECT
    Products.[NomenclatureNumber]
    ,Products.[Name]
    ,Products.[Description]
    ,Details.[Amount]
    ,Details.[Sum]
FROM [CoffeeStore].[dbo].[Products] AS Products
INNER JOIN [CoffeeStore].[dbo].[OrderDetails] AS Details
ON Products.Id = Details.ProductId
```

6 %

Results | Messages

| NomenclatureNumber | Name | Description | Amount | Sum |
|---|---|---|---|---|
| 0020-1445 | Barista FN-150 | Smart coffee machine for the home usage | 1 | 17500.00 |
| 0001-1200 | Sunshine Mexico | Mexican Arabica with the fruit notes | 2 | 2400.00 |
| 0001-1200 | Sunshine Mexico | Mexican Arabica with the fruit notes | 4 | 4800.00 |

# T-SQL basics – LEFT JOIN

```sql
SELECT
    Products.[NomenclatureNumber]
    ,Products.[Name]
    ,Products.[Description]
    ,Details.[Amount]
    ,Details.[Sum]
FROM [CoffeeStore].[dbo].[Products] AS Products
LEFT JOIN [CoffeeStore].[dbo].[OrderDetails] AS Details
ON Products.Id = Details.ProductId
```

6 %

Results | Messages

| NomenclatureNumber | Name | Description | Amount | Sum |
|---|---|---|---|---|
| 0001-1200 | Sunshine Mexico | Mexican Arabica with the fruit notes | 2 | 2400.00 |
| 0001-1200 | Sunshine Mexico | Mexican Arabica with the fruit notes | 4 | 4800.00 |
| 0020-1445 | Barista FN-150 | Smart coffee machine for the home usage | 1 | 17500.00 |
| 0001-1320 | Sunshine Columbia | Columbian magic with the chocolate notes | NULL | NULL |

# T-SQL basics – UNION

```sql
SELECT
    [Id]
    ,[Name]
FROM [CoffeeStore].[dbo].[Origins]

UNION

SELECT
    [Id]
    ,[Name]
FROM [CoffeeStore].[dbo].[Clients]
```

%

Results | Messages

| Id | Name |
|---|---|
| 646C032B-0AA8-4E1D-9C9B-4C8EAF3F10D4 | Mexico |
| 94BBF610-CF34-4BF2-972E-562CBD84B6EE | Columbia |
| F4610161-8779-4E99-A26F-8EC30EEAB712 | Ekaterina |
| 5B91CA37-6ED0-4972-8737-9A45E6636E04 | Aleksander |

# Books of the day

Groff J., Oppel A., Weinberg P. – SQL: The Complete Reference

Redmond E., Wilson J. – Seven databases in seven weeks

Gorman B.L. - Practical Entity Framework: Database Access for Enterprise Applications

# Links of the day

SQL запросы быстро. Часть 1 / Хабр (habr.com)

SQL Tutorial - Full Database Course for Beginners (Codecamp on Youtube)

Entity Framework Core Tutorials (entityframeworktutorial.net)

# Hometask

Write the script to create database for web shop. Create the tables below:

- dbo.Stocks (stocks information – Id, Name, Address)
- dbo.StockKeepingUnits (stock items in your warehouses – Id, StockId, Name, Description, Amount)
- dbo.Shipments (shipment info with date/time, stock and skus – Id, StockId, SkuId, Amount)
- dbo.ShopItemCategories (goods categorization – Id, Name, Description)
- dbo.ShopItems (information about the goods of your shop – Id, SkuId, CategoryId, Name, Description)
- dbo.Prices (price-on-date for your shop items – ShopItemId, DateTime, Price)
- dbo.Clients (registered clients info – Id, Name, Phone, Email, Delivery Address)
- dbo.OrderStatuses (list of order statuses – Value, Description)
- dbo.Orders (information about created orders – Id, ClientId, OrderStatus)
- dbo.OrderDetails (order rows information – OrderId, ShopItemId, Price, Amount, Sum)

That’s all for this time!