Candidate Key column must be UNIQUE and no nulls.   
  
Candidate Key is conceptual term when designing tables.  For example, SSIS Data Profiler identifies potential Primary Key as Candidate Keys.  
  
**Primary Key, Unique Key and Foreign Key are real constraints - database objects.  
  
PK & UK require UNIQUE index. FK is not unique unless it doubles as PK also.  
  
Surrogate Key, Candidate Key and Natural Key are logical data modelling terms.**If you remove the PK constraint in AdventureWorks.Production.Product table, it has 4 Candidate Keys:  
  
ProductID  
Name  
ProductNumber  
rowguid  
  
I know the above because ProductID is PK and the other 3 has unique index on them and no nulls.  
  
The designer picked ProductID (int identity(1,1)) to become the Primary Key.  In this capacity as a meaningless number, the ProductID Primary Key is called a Surrogate Key (surrogate to the Natural Keys: Name, ProductNumber). Rowguid also meaningless number, used for row versioning (replication).  Name & ProductNumber (more or less) are meaningful keys.

SELECT TOP (5) ProductID, Name, ProductNumber, rowguid

FROM AdventureWorks2008.Production.Product

ORDER BY NEWID()

/\*

ProductID Name ProductNumber rowguid

742 HL Mountain Frame - Silver, 46 FR-M94S-46 A189D86E-D923-4336-B13D-A5DB6F426540

789 Road-250 Red, 44 BK-R89R-44 0AA71AD6-AFAF-43C6-9745-35D815B50A5B

492 Paint - Black PA-187B DF20E514-3D47-491B-9454-0911EC3F7D29

413 Internal Lock Washer 4 LI-1200 7F7413BB-BAD2-47E4-9BC4-D98B194BE35D

709 Mountain Bike Socks, M SO-B909-M 18F95F47-1540-4E02-8F1F-CC1BCB6828D0

\*/

In the following example the Candidate Keys are defined as UNIQUE keys, that means, they can be referenced by Foreign Keys. **Normally though, you only place a UNIQUE INDEX on Candidate Keys (or Natural Keys) since you use the Primary Key for referencing.**  
CLUSTERED UNIQUE INDEX is created for the PK and non-clustered unique indexes for the UK-s. All PK & UK-s are constraints as database objects. Clustered is not a requirement for PK only UNIQUE INDEX is, SQL Server implements it that way as default unless you define it differently.

USE tempdb;

-- Table with PRODUCTID PRIMARY KEY and 3 UNIQUE KEYs

CREATE TABLE Product (

ProductId INT IDENTITY ( 1 , 1 ) PRIMARY KEY,

ProductName VARCHAR(32) UNIQUE,

ProductNo VARCHAR(12) UNIQUE,

Decription VARCHAR(256) NOT NULL,

rowguid UNIQUEIDENTIFIER DEFAULT (newid()) UNIQUE,

ModifiedDate SMALLDATETIME DEFAULT (getdate()));

-- ProductID is FOREIGN KEY

CREATE TABLE ProductPhoto (

ProductPhotoID INT IDENTITY ( 1 , 1 ) PRIMARY KEY,

ProductID INT references Product(ProductID),

Size CHAR(2),

Photo VARBINARY(MAX),

ModifiedDate SMALLDATETIME DEFAULT (getdate()));