**SQL SERVER – Difference Between Candidate Keys and Primary Key**

**Introduction**

Not long ago, I had an interesting and extended debate with one of my friends regarding which column should be primary key in a table. The debate instigated an in-depth discussion about candidate keys and primary keys. My present article revolves around the two types of keys.

Let us first try to grasp the definition of the two keys.

**Candidate Key** – A Candidate Key can be any column or a combination of columns that can qualify as unique key in database. There can be multiple Candidate Keys in one table. Each Candidate Key can qualify as Primary Key.

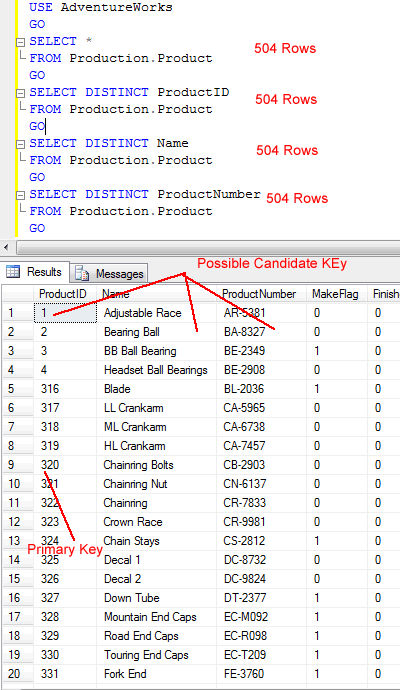
**Primary Key** – A Primary Key is a column or a combination of columns that uniquely identify a record. Only one Candidate Key can be Primary Key.

One needs to be very careful in selecting the Primary Key as an incorrect selection can adversely impact the database architect and future normalization. For a Candidate Key to qualify as a Primary Key, it should be Non-NULL and unique in any domain. I have observed quite often that Primary Keys are seldom changed. I would like to have your feedback on not changing a Primary Key.

**An Example to Understand Keys**

Let us look at an example where we have multiple Candidate Keys, from which we will select an appropriate Primary Key.

Given below is an example of a table having three columns that can qualify as single column Candidate Key, and on combining more than one column the number of possible Candidate Keys touches seven. A point to remember here is that only one column can be selected as Primary Key. The decision of Primary Key selection from possible combinations of Candidate Key is often very perplexing but very imperative!



On running the following script it will always give 504 rows in all the options. This proves that they are all unique in database and meet the criteria of a Primary Key.

Run the following script to verify if all the tables have unique values or not.

**USE AdventureWorks  
GO  
SELECT \*  
FROM Production.Product  
GO  
SELECT DISTINCT ProductID  
FROM Production.Product  
GO  
SELECT DISTINCT Name  
FROM Production.Product  
GO  
SELECT DISTINCT ProductNumber  
FROM Production.Product  
GO**

All of the above queries will return the same number of records; hence, they all qualify as Candidate Keys. In other words, they are the candidates for Primary Key. There are few points to consider while turning any Candidate Key into a Primary Key.

**Select a key that does not contain NULL**

It may be possible that there are Candidate Keys that presently do not contain value (not null) but technically they can contain null. In this case, they will not qualify for Primary Key. In the following table structure, we can see that even though column **[name]** does not have any NULL value it does not qualify as it has the potential to contain NULL value in future.

**CREATE TABLE [Production].[Product](  
[ProductID] [int] IDENTITY(1,1) NOT NULL,  
[Name] [dbo].[Name] NULL,  
[ProductNumber] [nvarchar](25) NOT NULL,  
[Manufacturer] [nvarchar](25) NOT NULL  
)**

**Select a key that is unique and does not repeat**

It may be possible that Candidate Keys that are unique at this moment may contain duplicate value. These kinds of Candidate Keys do not qualify for Primary Key. Let us understand this scenario by looking into the example given above. It is absolutely possible that two Manufacturers can create products with the same name; the resulting name will be a duplicate and only the name of the Manufacturer will differ in the table. This disqualifies Name in the table to be a Primary Key.

**Make sure that Primary Key does not keep changing**

This is not a hard and fast rule but rather a general recommendation: Primary Key values should not keep changing. It is quite convenient for a database if Primary Key is static. Primary Keys are referenced in numerous places in the database, from Index to Foreign Keys.  If they keep changing then they can adversely affect database integrity, data statistics as well as internal of Indexes.

**Selection of Primary Key**

Let us examine our case by applying the above three rules to the table and decide on the appropriate candidate for Primary Key. Name can contain NULL so it disqualifies as per Rule 1 and Rule 2. Product Number can be duplicated for different Manufacturers so it disqualifies as per Rule 2. ProductID is Identity and Identity column cannot be modified. So, in this case ProductID qualifies as Primary Key.

Please note that many database experts suggest that it is not a good practice to make Identity Column as Primary Key. The reason behind this suggestion is that many times Identity Column that has been assigned as Primary Key does not play any role in database. There is no use of this Primary Key in both application and in T-SQL. Besides, this Primary Key may not be used in Joins. It is a known fact that when there is JOIN on Primary Key or when Primary Key is used in the WHERE condition it usually gives better performance than non primary key columns. This argument is absolutely valid and one must make sure not to use such Identity Column. However, our example presents a different case. Here, although ProductID is Identity Column it uniquely defines the row and the same column will be used as foreign key in other tables. If a key is used in any other table as foreign key it is likely that it will be used in joins.

**Quick Note on Other Kinds of Keys**

The above paragraph evokes another question – what is a foreign key? A foreign key in a database table is a key from another table that refers to the primary key in the table being used. A primary key can be referred by multiple foreign keys from other tables. It is not required for a primary key to be the reference of any foreign keys. The interesting part is that a foreign key can refer back to the same table but to a different column. This kind of foreign key is known as “self-referencing foreign key”.

**Summary**

A table can have multiple Candidate Keys that are unique as single column or combined multiple columns to the table. They are all candidates for Primary Key. Candidate keys that follow all the three rules – 1) Not Null, 2) Unique Value in Table and 3) Static – are the best candidates for Primary Key. If there are multiple candidate keys that are satisfying the criteria for Primary Key, the decision should be made by experienced DBAs who should keep performance in mind.