**Why use keys**

When you specify keys, there are underlining infrastructures (like index and constraint) in SQL automatically built to support the keys. These infrastructures help to organize data, ensure data quality, detect abnormalities, and make data search fast.

If you work in a SQL environment (not a big data or parallelized data warehouse environment), it’s generally encouraged to put keys on your tables.

**Types of Key (see diagram below):**

Primary key (PK)

It is a column or a combination of columns that can be used to identify a unique row in a table.

* It’s strongly suggested that you create a PK for every table you create. Please see the syntax below on how to create PK.
* PK can be a single column (an ID column) or combination of several columns (this is called composite key or compound key).
* Only 1 PK per table.
* PK will by default be clustered indexed (see index documentation) unless you instruct SQL to do otherwise.
* An “UNIQUE” constraint is automatically placed on PK. Reading for Unique constraint is supplied in the “Additional reading” section.
* If the PK is a surrogate key, then a UNIQUE constraint has to be placed on on the natural key(s).
* None of the columns as part of PK is NULLable.
* Size limitation < 900 bytes total (~900 English characters). So you cannot use a varchar(max) column, which is 2^31 bytes, as PK.

Foreign key (FK)

It often appears in a normalized data model. Dimension columns in a fact table often use FK reference to the PK of dimension tables.

* You can have multiple columns use FK references to other tables.
* FK column is not indexed by default (see index documentation)
* Size limit < 900 bytes total.

Candidate key

Sometimes in a table there are several potential value or combination of values that can identify a unique row. Each of these unique identifier is a candidate Key.

* You’ll often choose one of these candidate keys to be your PK.
* Choose the combination with lowest number of columns and can still identify a unique row.

Business key/ Natural key

It is a value or combination of values which identifies uniqueness of a row naturally in a table according to business rules. For example, in an “user” table, “SSN” is a business key that makes each user unique naturally.

* Business key/natural key is a type of candidate key that you may or may not choose as the PK.

**Format of Keys:**

Single column

the key is placed on a single column. This could be the case for PK or FK.

Composite (compound) key

The key is placed on a combination of columns. This could be a PK or UNIQUE key. If a composite key is used, the order of how the columns are referred is critical.

A block from the code sample below:

PRIMARY KEY CLUSTERED

(

[date\_key],

[entry\_bg\_site\_id],

[mkt\_country\_id]

)

This means SQL creates a sorting system (aka index) on [date\_key] first, then moves on to [entry\_bg\_site\_id], and finally [mkt\_country\_id]. This is the order that the columns should be used in joins.

So, this code will give you great join performance as the columns are used in the same order as they are placed when the key is created.



And this code will give you bad performance as the order of join doesn’t follow the order when the key is created**.**

****

Code sample for table creation with PK and FK:****

**Choose PK:**

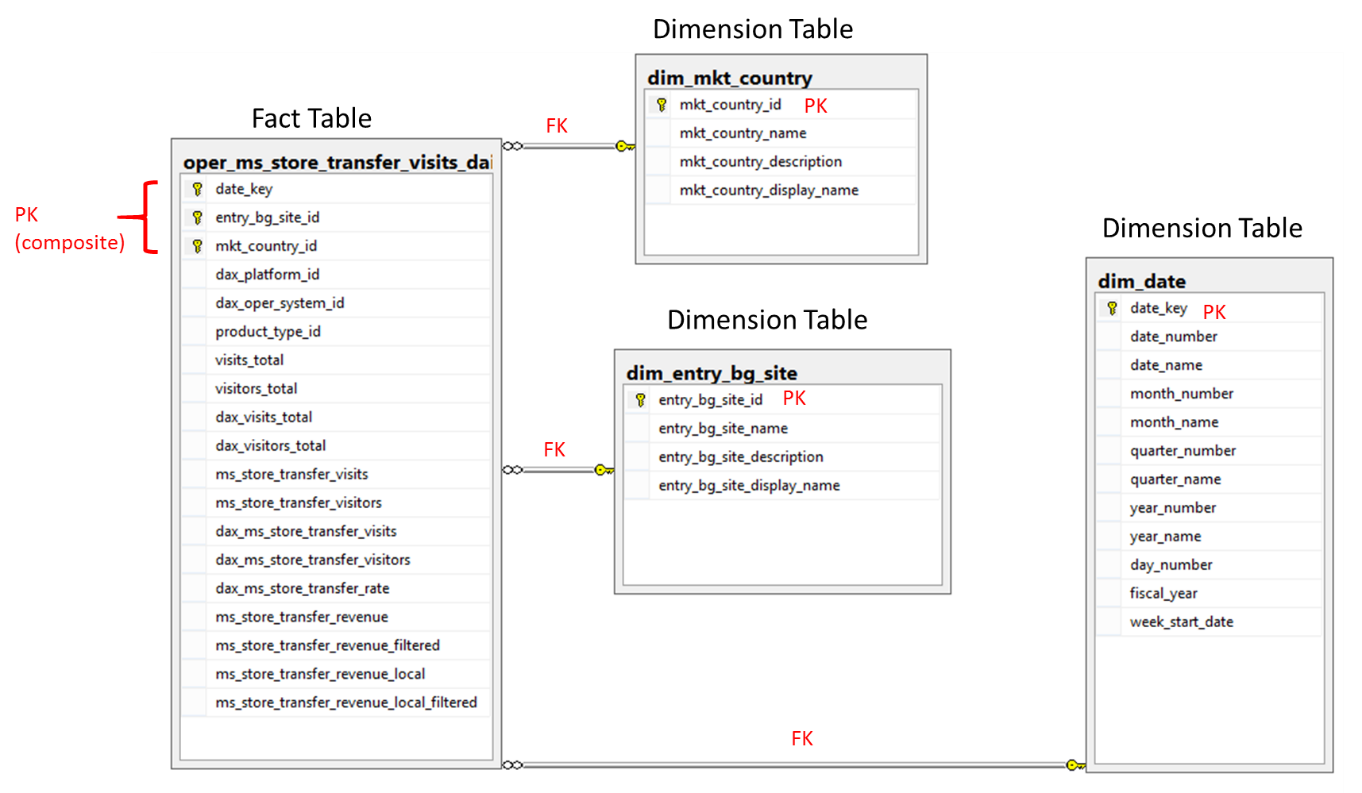
Rule-of-thumb: PK should be as small as possible but still be able to identify a unique row.

Show in the fact table below:

Option A: let’s say, combination of date\_key, entry\_bg\_site\_id, mkt\_country\_id, dax\_platform\_id, dax\_oper\_system\_id, product\_type\_id is unique to each row

Option B: let’s say, combination of date\_key, entry\_bg\_site\_id, mkt\_country\_id is also unique in each row

Option B is chosen as the PK because it’s smaller.

****

**Additional reading:**

Unique constraint: <https://technet.microsoft.com/en-us/library/ms191166(v=sql.105).aspx>

Create Primary Key: <https://msdn.microsoft.com/en-us/library/ms189039.aspx>

Choose PK: <http://www.kimballgroup.com/2006/07/design-tip-81-fact-table-surrogate-key/>

<http://sqlmag.com/database-administration/sql-design-how-choose-primary-key>