Article
Robert Cemper · Apr 25, 2020 2m read

# Static WHERE Conditions

The typical WHERE condition in SQL relates mostly to some content of the rows you work on.
So it needs to be calculated and checked for each row you access.
Differently (and that's why I named it STATIC) is a WHERE condition that is independent of the rows you access.

The nice thing about it:
It is accessed and evaluated only once before anything else happens.
You could interpret it as a kind of main switch to your table.
In the past, this was often used as a workaround for simple SQL access rules.
A more tricky use is the combination with a SQLprocedure like this very simple example:

```
 Class User.Hack
{ ClassMethod chk(any As %String) As %Boolean [ SqlName = mychk, SqlProc ]
{   set if=$get(any)#2 ;; just a placeholder for a complex condition
  break              ;; for interactive examination
  quit if
  } }
```

and to get all lines and check only once (!!) you run:

```
SELECT id,name,Home_State from sample.person where home_State %startswith 'A'
  and MYCHK('rcc')=1
```

differently, dynamic use (with ID)  it will be evaluated for each line
- line with odd ID

```
SELECT id,name,Home_State from sample.person where home_State %startswith 'A'
  and MYCHK(id)=1
```

- lines with even IDs

```
SELECT id,name,Home_State from sample.person where home_State %startswith 'A'
  and MYCHK(id)=0
```

The important point is you have a backdoor to your SQLstatement to pass your specific parameters into a query **before** any access to your data.
I have used this already inside notes for various replies.
A further article will demonstrate a practical use case.

#Other

**Source URL:**https://community.intersystems.com/post/static-where-conditions