***What does the sub-clause “OR1=1” & “AND 1=1” Means?***

***And wy we use AND 1=1 for query fixing not OR 1=1?***

Let’s look at the example below;

<https://www.finvent.com/details.php?id=8>  
this link will show me a result. That result will be come from a database by using a query in backend. Suppose that query is  
SELECT \* FROM tbl\_details WHERE id=’8’

Now in traditional attack we will do the following things sequentially.  
<https://www.finvent.com/details.php?id=8>’  
//This will inject the database query and the database will through an error. Cause the database query will be then  
SELECT \* FROM tbl\_details WHERE id=’8’’

<https://www.finvent.com/details.php?id=8>’ -- -  
//This will fix the error by commenting out the rest of the query. Specifically here in our case   
-- -‘  
now if I put a negative sign before the id then the link will be  
<https://www.finvent.com/details.php?id=-8>’ -- -

and the database query will be  
SELECT \* FROM tbl\_details WHERE id=’-8’-- -’  
But in database there is no such id which value is -1.  
So this will return nothing.  
  
But here if I put the “OR 1=1” clause then the link will be  
<https://www.finvent.com/details.php?id=-8>’ OR 1=1 -- -  
and the database query will be  
SELECT \* FROM tbl\_details WHERE id=’-8’ OR 1=1 -- -’  
this will return all the data from the tbl\_details table.

But WHY?

--  
SELECT \* FROM tbl\_details WHERE id=’-8’ OR 1=1.  
This query means SELECT all the data (\*) FROM the table tbl\_details WHERE id=-8 *OR* WHERE 1=1. Look here, “WHERE” clause always return Boolean type value (True or False).   
Here in our query  
WHERE id=-8 (Return FALSE)  
But  
WHERE 1=1 (Return **TRUE**) //Cause 1=1 is always true.  
And both of them are concatenated with **OR** that means,   
**IF ANYONE OF THE BOOLEAN PARAMETER RETURN TRUE THEN THE DATABASE QUERY WILL RETURN VALUE.**

Now why this will return all data?  
Because first WHERE parameter is FALSE and second WHERE parameter is TRUE. Look, First WHERE parameter is filled with a field and a defined value (WHERE id=-8) and the second WHERE parameter is not filled with any field and thus any value. So whenever that second WHERE parameter will be true this will return all the data from the table tbl\_detals.  
--

Now “AND 1=1”  
If I put AND 1=1 clause in the link  
<https://www.finvent.com/details.php?id=-8>’ AND ‘1’=’1’ -- -  
and the database query will be  
SELECT \* FROM tbl\_details WHERE id=’-8’ AND 1=1 -- -’  
this will not return any value. Thus this will not fix the query.

But WHY?

--  
SELECT \* FROM tbl\_details WHERE id=’-8’ AND 1=1.  
This query means SELECT all the data(\*) FROM the table tbl\_details WHERE id=-8 *AND* WHERE 1=1. Look here, “WHERE” clause always return Boolean type value (True or False).   
Here in our query  
WHERE id=-8 (Return FALSE)  
But  
WHERE 1=1 (Return **TRUE**) //Cause 1=1 is always true.  
And both of them are concatenated with **AND** that means,   
**IF BOTH OF THE BOOLEAN PARAMETER RETURN TRUE THEN THE DATABASE QUERY WILL RETURN VALUE.**  
--

Now why this will return no data or not fix the query?  
Because first WHERE parameter is FALSE and second WHERE parameter is TRUE. But the query needs both the parameter value as TRUE. That’s why this will not return a value.  
If I just change the query a bit

<https://www.finvent.com/details.php?id=8>’ AND ‘1’=’1’ -- -  
and the database query will be  
SELECT \* FROM tbl\_details WHERE id=’8’ AND 1=1 -- -’  
And THIS WILL RETURN THE EXACT VALUE THAT THE ORIGINAL LINK (THE ORIGINAL QUERY) WAS INTENDED TO RETURN. THUS THIS WILL BE USED TO DO **QUERY FIXING** NOT THE ‘OR 1=1’.

Cause AND 1=1 returns the exact value of original URL or query;   
whereas OR 1=1 returns all the value form the database table from which the value is coming from.