# Oracle null column index

If index was created against a nullable column, then oracle will index the rows only have not null values. So, if we try to query for null value then oracle will skip the index and go for full table scan since it knows null values are not available in index.

This is the original cursor with parameters,

**CURSOR** auto\_stopped\_clockings **IS**

**SELECT** \*

**FROM** shop\_oper\_clocking\_tab

**WHERE** clocking\_type = 'LABOR'

**AND** company = company\_

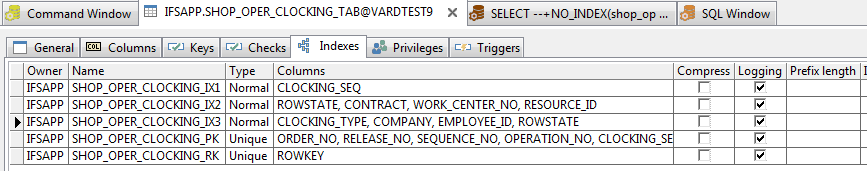
**AND** (employee\_id = employee\_id\_ **OR** employee\_id\_ **IS** **NOT** **NULL**)

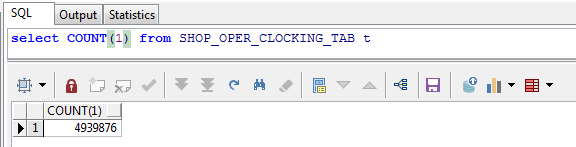
**AND** (team\_id\_ **IS** **NULL** **OR** team\_id = team\_id\_)

**AND** auto\_stopped = 'TRUE'

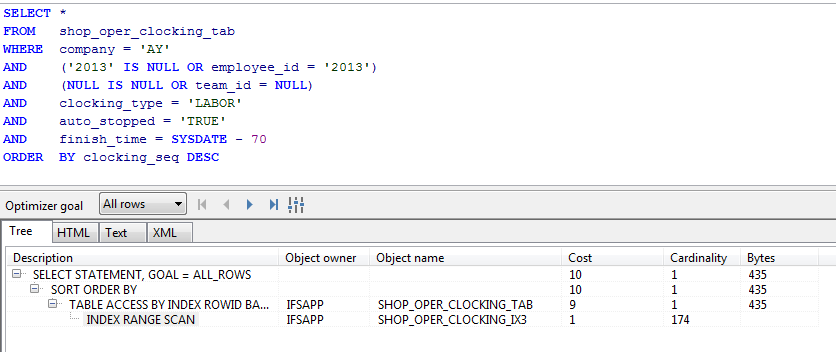
**AND** finish\_time = finish\_time\_

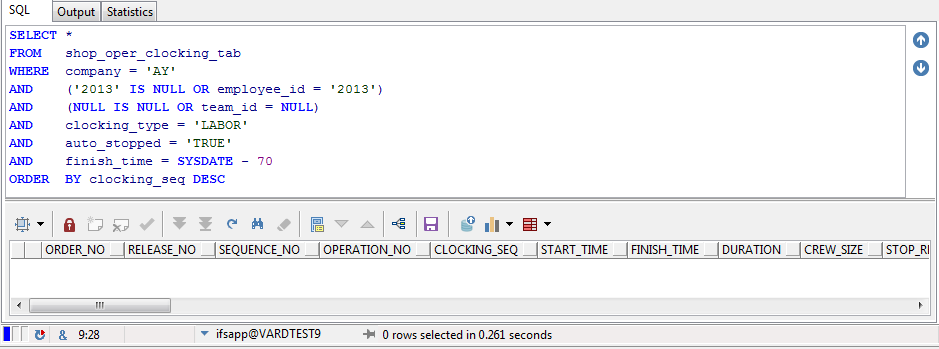
**ORDER** **BY** clocking\_seq **DESC**;



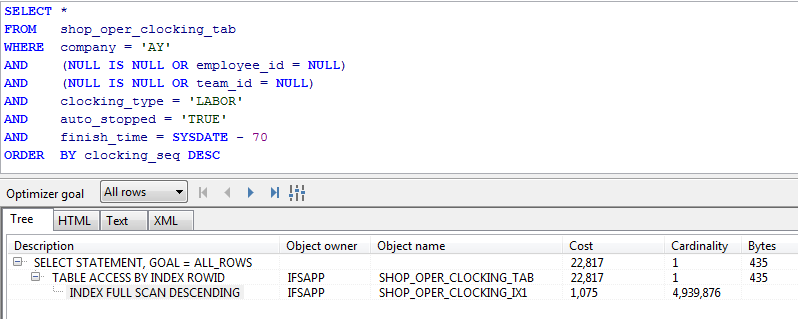


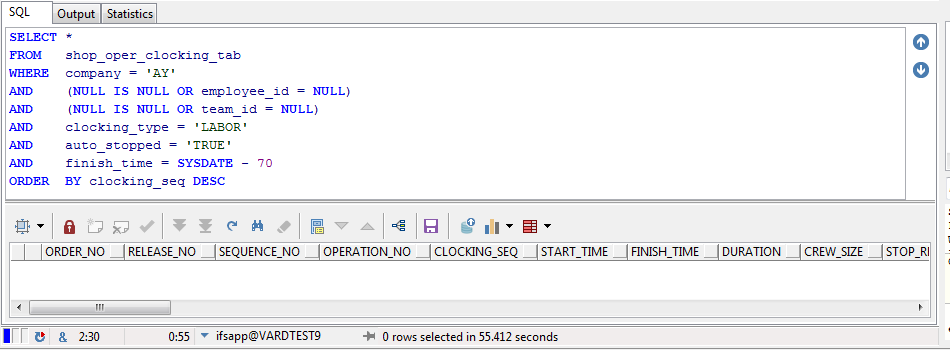
In sql we replace employee id with value and run the query as sql statement then it is executing as follow,





But if we try to use null in query, then output will look like follow,





In this case oracle go for full table scan since it knows that there are no null values in index.

We can test the real-world scenario using following pl/sql block

**DECLARE**

company\_ **VARCHAR2**(200) := 'AY';

employee\_id\_ **VARCHAR2**(200) := '2013';

team\_id\_ **VARCHAR2**(200) := **NULL**;

finish\_time\_ **DATE** := **SYSDATE**;

**CURSOR** auto\_stopped\_clockings **IS**

**SELECT** \*

**FROM** shop\_oper\_clocking\_tab

**WHERE** company = company\_

**AND** (employee\_id = employee\_id\_ **OR** employee\_id\_ **IS** **NULL**)

**AND** (team\_id = team\_id\_ **OR** team\_id\_ **IS** **NULL**)

**AND** clocking\_type = 'LABOR'

**AND** auto\_stopped = 'TRUE'

**AND** finish\_time = finish\_time\_

**ORDER** **BY** clocking\_seq **DESC**;

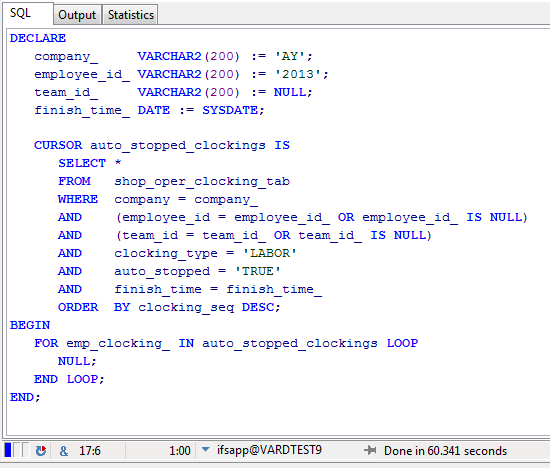
**BEGIN**

**FOR** emp\_clocking\_ **IN** auto\_stopped\_clockings **LOOP**

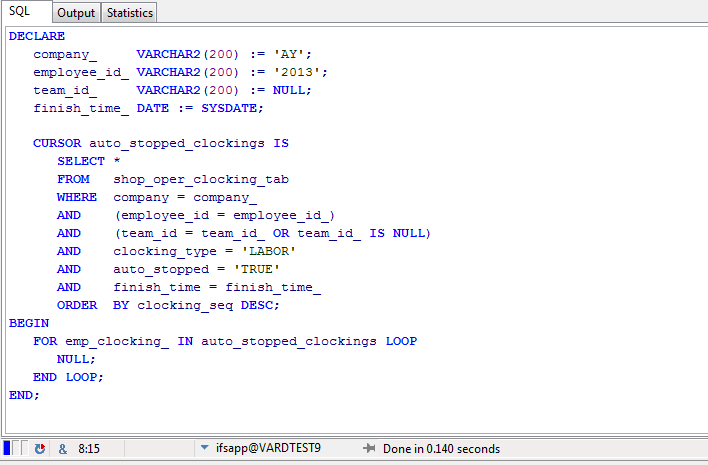
**NULL**;

**END** **LOOP**;

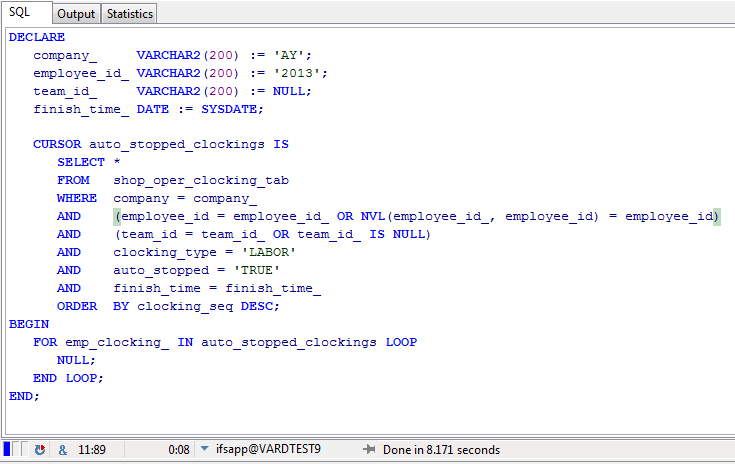
**END**;



At this point oracle cannot predict the value of employee id null or not. So, it tries full table scan. But if we remove “employee\_id\_ is null” from where clause, then it will go for index scan since oracle think that employee id is always available.



So, the solution was let oracle to think that employee id is not null.



Enforcing index scan using index hint is also not working since oracle does concern on null values