```
------ Inspection Report (https://redmine.scubeenterprise.com/issues/19508)
----------------------------------------
---- Description: guidesheet_data
---- Version:1.0
----------------------------------------
---- Created:  2024-08-09 - js
---- Modified: 2024-08-20 - js - A trick here is that the client expects "comments" to be "violations",
and guide_item_comment to be "comment".
--------------------------------------
---- Development Code
--DECLARE @inspection_id int;
--
--set @inspection_id = 18002;
--------------------------------------------

--Main SQL below this

/*
comments
observations
isCos_isRpt
preResult
stuffed_asitabValues
__idx_val
_idx_val
idx_val
_resultt
resultt
final_query
*/

declare @comments table (
        serv_prov_code                  varchar(100)
        ,GUIDE_ITEM_SEQ_NBR             int
        ,ASITAB_NAME            varchar(2000)
        ,ATTRIBUTE_VALUE        varchar(2000)
        ,asitab_row_index       int
)
insert into @comments
select distinct
                a.SERV_PROV_CODE
                ,gi.GUIDE_ITEM_SEQ_NBR
                ,asitv.ASITAB_NAME
                ,asitv.ATTRIBUTE_VALUE
        ,asitv.asitab_row_index
    from G6ACTION a with(nolock)
    join b1permit b with(nolock) on b.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(b.b1_per_id1,b.b1_per_id2,b.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3)
    join GGUIDESHEET g with(nolock) on g.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(g.b1_per_id1,g.b1_per_id2,g.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3) and g.G6_ACT_NUM = a.g6_act_num
    join gguidesheet_item gi with(nolock) on gi.SERV_PROV_CODE = a.SERV_PROV_CODE and
gi.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR
    join GGDSHEET_ITEM_ASITAB_VALUE asitv with(nolock) on asitv.SERV_PROV_CODE =
a.SERV_PROV_CODE and asitv.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR and
asitv.GUIDEITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR
    where
        1=1
        and a.SERV_PROV_CODE = 'DAVISCO'
        and a.g6_act_num = @inspection_id
        and ASITAB_NAME in ('Comment')
;

declare @observations table (
    serv_prov_code                  varchar(100)
    ,GUIDE_ITEM_SEQ_NBR             int
    ,ASITAB_NAME            varchar(2000)
    ,ATTRIBUTE_VALUE        varchar(2000)
    ,asitab_row_index       int
)
insert into @observations
select distinct
    a.SERV_PROV_CODE
    ,gi.GUIDE_ITEM_SEQ_NBR
    ,asitv.ASITAB_NAME
    ,asitv.ATTRIBUTE_VALUE
    ,asitv.asitab_row_index
from G6ACTION a
join b1permit b with(nolock) on b.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(b.b1_per_id1,b.b1_per_id2,b.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3)
join GGUIDESHEET g with(nolock) on g.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(g.b1_per_id1,g.b1_per_id2,g.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3) and g.G6_ACT_NUM = a.g6_act_num
join gguidesheet_item gi with(nolock) on gi.SERV_PROV_CODE = a.SERV_PROV_CODE and
gi.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR
```

```
join GGDSHEET_ITEM_ASITAB_VALUE asitv with(nolock) on asitv.SERV_PROV_CODE =
a.SERV_PROV_CODE and asitv.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR and
asitv.GUIDEITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR
where
        1=1
        and a.SERV_PROV_CODE = 'DAVISCO'
        and a.g6_act_num = @inspection_id
        and ASITAB_NAME in ('Observation')
;
--), isCos_isRpt as (
declare @isCos_isRpt table (
        serv_prov_code                  varchar(100)
        ,GUIDE_ITEM_SEQ_NBR int
        ,asitab_row_index       int
        ,isCos  int
        ,isRpt  int
        ,isMajor int
)
insert into @isCos_isRpt
        select distinct
                 a.serv_prov_code
                ,gi.GUIDE_ITEM_SEQ_NBR
                ,asitv.ASITAB_ROW_INDEX
                ,max(case when asitv.asitab_name = 'COS' and asitv.ATTRIBUTE_VALUE =
'CHECKED' then 1 else 0 end) as isCos
                ,max(case when asitv.asitab_name = 'RPT' and asitv.ATTRIBUTE_VALUE =
'CHECKED' then 1 else 0 end) as isRpt
                ,max(case when asitv.asitab_name = 'Major Violation' and asitv.ATTRIBUTE_VALUE =
'CHECKED' then 1 else 0 end) as isMajor

        from G6ACTION a with(nolock)
        join b1permit b with(nolock) on b.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(b.b1_per_id1,b.b1_per_id2,b.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3)
        join GGUIDESHEET g with(nolock) on g.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(g.b1_per_id1,g.b1_per_id2,g.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3) and g.G6_ACT_NUM = a.g6_act_num
        join gguidesheet_item gi with(nolock) on gi.SERV_PROV_CODE = a.SERV_PROV_CODE and
gi.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR
        join GGDSHEET_ITEM_ASITAB_VALUE asitv with(nolock) on asitv.SERV_PROV_CODE =
a.SERV_PROV_CODE and asitv.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR and
asitv.GUIDEITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR
        where
                1=1
        and a.SERV_PROV_CODE = 'DAVISCO'
        and a.g6_act_num = @inspection_id
        and ASITAB_NAME in ('COS','RPT','Major Violation')
    group by a.serv_prov_code,gi.GUIDE_ITEM_SEQ_NBR,asitv.ASITAB_ROW_INDEX
;

declare @stuffed_asitabValues table (
    serv_prov_code varchar(200)
    ,guide_item_seq_nbr int
    ,string varchar(2000)
)
--;with _stuffed_asitabValues as (
insert into @stuffed_asitabValues
    select distinct
         a.SERV_PROV_CODE
        ,gi.GUIDE_ITEM_SEQ_NBR
        --,c.ASITAB_ROW_INDEX
        ,coalesce(c.ATTRIBUTE_VALUE,'') + coalesce( ' ('+o.ATTRIBUTE_VALUE+')','')
        + coalesce(case when isCos_isRpt.isRpt=1 then ' REPEAT.' else '' end,'')
        + coalesce(case when isCos_isRpt.isCos=1 then ' CORRECTED ON SITE.' else ''
end,'')
        + coalesce(case when isCos_isRpt.isMajor=1 then ' MAJOR.' else '' end,'')
        as string
    from G6ACTION a
    join b1permit b with(nolock) on b.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(b.b1_per_id1,b.b1_per_id2,b.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3)
    join GGUIDESHEET g with(nolock) on g.SERV_PROV_CODE = a.SERV_PROV_CODE and
concat(g.b1_per_id1,g.b1_per_id2,g.b1_per_id3) =
concat(a.b1_per_id1,a.b1_per_id2,a.b1_per_id3) and g.G6_ACT_NUM = a.g6_act_num
    join gguidesheet_item gi with(nolock) on gi.SERV_PROV_CODE = a.SERV_PROV_CODE and
gi.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR
    join GGDSHEET_ITEM_ASITAB_VALUE asitv with(nolock) on asitv.SERV_PROV_CODE =
a.SERV_PROV_CODE and asitv.GUIDESHEET_SEQ_NBR = g.GUIDESHEET_SEQ_NBR and
asitv.GUIDEITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR
    left join @comments c on c.serv_prov_code = a.serv_prov_code and
c.GUIDE_ITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR and c.ASITAB_ROW_INDEX =
asitv.ASITAB_ROW_INDEX
    left join @observations o on o.serv_prov_code = a.serv_prov_code and
o.GUIDE_ITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR  and o.ASITAB_ROW_INDEX =
asitv.ASITAB_ROW_INDEX
    left join @isCos_isRpt isCos_isRpt on isCos_isRpt.serv_prov_code = a.serv_prov_code and
isCos_isRpt.GUIDE_ITEM_SEQ_NBR = gi.GUIDE_ITEM_SEQ_NBR and
isCos_isRpt.ASITAB_ROW_INDEX = asitv.ASITAB_ROW_INDEX
```

```
	where
		1=1
		and a.SERV_PROV_CODE = 'DAVISCO'
		and a.g6_act_num = @inspection_id
;



declare @idx_val table (
	guide_type varchar(2000)
	,idx int
	,val varchar(2000)
	,sub_cat varchar(2000)
	,Header_Start int
	,Header_End int
	,isBroaderCat int
	,_sort_order int
	,is_sub_cat int
	,broader_cat varchar(2000)
)
;with ___idx_val as (
	select distinct
		 a.GUIDE_TYPE
		,a.GUIDE_ITEM_TEXT as Header
		,a.GUIDE_ITEM_DISPLAY_ORDER as Header_Start
		,a.GUIDE_ITEM_TEXT as [value] --c.value
	from rguidesheet_item a with (nolock)
   --filters:
	join gguidesheet_item gi with(nolock) on gi.SERV_PROV_CODE = a.SERV_PROV_CODE and gi.GUIDE_TYPE = a.GUIDE_TYPE
	join GGUIDESHEET g with(nolock) on g.SERV_PROV_CODE = a.SERV_PROV_CODE and g.GUIDESHEET_SEQ_NBR = gi.GUIDESHEET_SEQ_NBR
	--CROSS APPLY STRING_SPLIT(trim(a.GUIDE_ITEM_TEXT), CHAR(10)) c
	where
		1=1
		and a.SERV_PROV_CODE = 'DAVISCO'
		and a.GUIDE_ITEM_STATUS_VISIBLE is null
		and a.REC_STATUS = 'A'
		and g.G6_ACT_NUM = @inspection_id
		--and c.value <> ''
)
, __idx_val as (
	select
		h.GUIDE_TYPE
```

```
        , h.Header
        , h.Header_Start
        , min(ISNULL(h2.Header_Start,10000)) - 1 as Header_End
    from ___idx_val h
    left join ___idx_val h2 on h.GUIDE_TYPE = h2.GUIDE_TYPE and h.Header_Start <
h2.Header_Start
    group by h.GUIDE_TYPE, h.Header, h.Header_Start
    --order by GUIDE_TYPE, Header_Start
)
, _idx_val as (
----insert into @_idx_val
    select distinct
        r.guide_type
        ,r.GUIDE_ITEM_DISPLAY_ORDER as idx
        ,r.GUIDE_ITEM_TEXT as [value] --c.[value]
        ,r.GUIDE_ITEM_TEXT as val
        ,a.header as sub_cat
        ,a.Header_Start
        ,a.Header_End
        --,(case when r.guide_item_Text like c.value +'%' then 1 else 2 end) as
_order_based_on_splitting
        ,(case
            when a.header_start = a.header_end then 1
            --when c.value = r.GUIDE_ITEM_TEXT then 0
            --when r.guide_item_text like c.value +'%' and r.guide_item_text <> c.value
and r.guide_item_text not like '[0-9]%)%' then 1
            --when r.guide_item_text not like '[0-9]%)%' then 1
            else 0
            end
        ) as isBroaderCat
        ,(case
            when --when is_broader_cat, then is_sub_cat=0
                (case
                    when a.header_start = a.header_end then 1
                    --when c.value = r.GUIDE_ITEM_TEXT then 0
                    --when r.guide_item_text like c.value +'%' and
r.guide_item_text <> c.value and r.guide_item_text not like '[0-9]%)%' then 1
                    else 0
                    end
                ) = 1 then 0
            when r.GUIDE_ITEM_TEXT = a.header then 1
            when r.GUIDE_ITEM_TEXT like '%'+a.header then 1
            else 0
            end
        ) as is_sub_cat
    from __idx_val a
    left join rguidesheet_item r with(nolock) on r.GUIDE_TYPE = a.GUIDE_TYPE and
r.GUIDE_ITEM_DISPLAY_ORDER between a.Header_Start and a.Header_End
    --CROSS APPLY STRING_SPLIT(r.GUIDE_ITEM_TEXT, CHAR(10)) c
    where
        1=1
        --and r.GUIDE_ITEM_TEXT like '%'+char(10)+'%'
        --and isnull(c.value,'') <> ''
    --order by guide_type,r.GUIDE_ITEM_DISPLAY_ORDER, case when r.guide_item_Text like
c.value +'%' then 1 else 2 end
)
--select * from ___idx_val  order by header_start
--select * from __idx_val           order by header_start
--select * from _idx_val

insert into @idx_val
select
    guide_type
    ,idx
    ,a.[value] as val
    ,sub_cat
    ,Header_Start
    ,Header_End
    ,isBroaderCat
    ,ROW_NUMBER() over (order by
            guide_type
            ,idx
            --,_order_based_on_splitting
        ) as _sort_order
    ,is_sub_cat
    ,isnull(c.[value],'') as broader_cat
from _idx_val a
outer apply (select top 1 [value] from _idx_val where header_start <= a.header_start and
isBroaderCat = 1 order by Header_Start asc) c
order by
    guide_type
    , ROW_NUMBER() over (order by
            guide_type
            ,idx
            --,_order_based_on_splitting
        )
;
```

```
declare @almost_there table (
    rownum int
    ,GUIDE_TYPE varchar(2000)
    ,toPrint varchar(2000)
    ,guide_item_status varchar(200)
    ,isBroaderCat int
    ,isSubcatHeader int
)
;with _resultt as (
    select
        a.serv_prov_code
        ,sub_cat.broader_cat
        ,sub_cat.sub_cat
        ,gi.guide_item_text
        ,(case gi.GUIDE_ITEM_STATUS
            when 'IN' then 'IN COMPLIANCE'
            when 'OUT' THEN 'OUT OF COMPLIANCE'
            WHEN 'NA' THEN 'NOT APPLICABLE'
            WHEN 'NO' THEN 'NOT OBSERVED'
            else ''
            end
         ) as guide_item_status
        ,gi.guide_item_comment
        ,asitv.string
        ,gi.GUIDE_TYPE
        ,gi.GUIDE_ITEM_DISPLAY_ORDER
        ,sub_cat._sort_order
    from GGUIDESHEET a
    join GGUIDESHEET_ITEM gi with(nolock) on gi.SERV_PROV_CODE = a.SERV_PROV_CODE
and gi.GUIDESHEET_SEQ_NBR = a.GUIDESHEET_SEQ_NBR
    left join @idx_val sub_cat on sub_cat.guide_type = gi.guide_type and sub_cat.idx =
gi.guide_item_display_order
    join G6ACTION i with(nolock) on i.SERV_PROV_CODE = a.SERV_PROV_CODE and
i.g6_act_num = a.G6_ACT_NUM
    left join @stuffed_asitabValues asitv
        on asitv.SERV_PROV_CODE = a.SERV_PROV_CODE
        and asitv.GUIDE_ITEM_SEQ_NBR = gi.guide_item_seq_nbr
    where
        1=1
        and a.serv_prov_code = 'DAVISCO'
        and a.g6_act_num = @inspection_id
        --and asitv.string is not null
        and gi.guide_item_status in ('IN','OUT','NA','NO')

--order by gi.GUIDE_ITEM_DISPLAY_ORDER
)
----select * from _resultt
,resultt as (

    ----this ensures sub_cat values are present! rem: the union will omit recurrences of these header rows.
    select
        a.idx
        ,a.val as toPrint
        ,'' as guide_item_status
        ,1 as isSubcatHeader
        ,r.GUIDE_TYPE
        ,a.isBroaderCat
        ,a._sort_order
    from @idx_val a
    join _resultt r on a.GUIDE_TYPE = r.guide_type and a.sub_cat = r.sub_cat
    where
        1=1
        --and a.sub_cat = a.val
        and a.is_sub_cat = 1 and a.isBroaderCat = 0

    union
    --This gets broader categories, which get a different shading in the report
    select
        a.idx
        ,replace(replace(a.val,char(13),''),char(10),' ') as toPrint
        ,'' as guide_item_status
        ,0 as isSubcatHeader
        ,a.GUIDE_TYPE
        ,a.isBroaderCat
        ,a._sort_order
    from @idx_val a
    join _resultt r on a.GUIDE_TYPE = r.guide_type and a.broader_cat = r.broader_cat
    where
        1=1
        --and a.sub_cat = a.val
        and a.is_sub_cat = 0 and a.isBroaderCat = 1



    union
```

```
        --This gets the actual guide item and guide item status values
        select distinct
    a.idx
    ,coalesce(r.guide_item_text,a.val) as toPrint
    ,coalesce(r.GUIDE_ITEM_STATUS,'') as guide_item_status
    ,(case when a.sub_Cat = a.val then 1 else 0 end) as isSubcatHeader
                ,r.GUIDE_TYPE
                ,a.isBroaderCat
                ,a._sort_order
  from @idx_val a
  join _resultt r
                on a.GUIDE_TYPE = r.guide_type
                and case
                        when replace(r.guide_item_text,'        ','') like '[0-9]) %' then
substring(replace(r.guide_item_text,'   ',''),4,len(replace(r.guide_item_text,'     ','')))
                        when replace(r.guide_item_text,'        ','') like '[0-9][0-9]) %' then
substring(replace(r.guide_item_text,'   ',''),5,len(replace(r.guide_item_text,'     ','')))
                        else replace(r.guide_item_text,'        ','')
                        end
                        =
                        case
                        when replace(a.val,'    ','') like '[0-9]) %' then substring(replace(a.val,'
        ',''),4,len(replace(a.val,' ','')))
                        when replace(a.val,'    ','') like '[0-9][0-9]) %' then substring(replace(a.val,'
        ',''),5,len(replace(a.val,' ','')))
                        else replace(a.val,'    ','')
                        end
                        --replaces tabs and ensures old guidesheet_text values without preceding
digits still match

  where
    1=1
    and r.serv_prov_code = 'DAVISCO'
    and exists (
      select 1
      from _resultt
      where
        1=1
        and sub_cat = a.sub_cat
    )
    and (
      a.val = a.sub_cat
      or r.sub_cat is not null
    )
  --order by a.idx

  union

  select
    a.idx
    ,'Comment: '+r.GUIDE_ITEM_COMMENT as toPrint
    ,'' as guide_item_status
    ,isSubcatHeader=0 --(case when a.sub_Cat = a.val then 1 else 0 end) as isSubcatHeader
                ,r.GUIDE_TYPE
                ,isBroaderCat = 0
                ,a._sort_order
  from @idx_val a
  left join _resultt r
                on a.GUIDE_TYPE = r.guide_type
                and case
                        when replace(r.guide_item_text,'	',''') like '[0-9]) %' then
substring(replace(r.guide_item_text,'	',''),4,len(replace(r.guide_item_text,'	','')))
                        when replace(r.guide_item_text,'	','') like '[0-9][0-9]) %' then
substring(replace(r.guide_item_text,'	',''),5,len(replace(r.guide_item_text,'	','')))
                        else replace(r.guide_item_text,'	','')
                        end
                        =
                        case
                        when replace(a.val,'	','') like '[0-9]) %' then substring(replace(a.val,'
        ',''),4,len(replace(a.val,' ','')))
                        when replace(a.val,'	','') like '[0-9][0-9]) %' then substring(replace(a.val,'
        ',''),5,len(replace(a.val,' ','')))
                        else replace(a.val,'	','')
                        end
                        --replaces tabs and ensures old guidesheet_text values without preceding
digits still match

  where
    1=1
    and r.serv_prov_code = 'DAVISCO'
    and r.GUIDE_ITEM_COMMENT is not null
    and exists (
      select 1
      from _resultt
      where
        1=1
        and sub_cat = a.sub_cat
    )
    and (
      a.val = a.sub_cat
      or r.sub_cat is not null
    )

  union

  select
    a.idx
    ,'Violation: '+r.string as toPrint
    ,'' as guide_item_status
    ,isSubcatHeader = 0 --(case when a.sub_Cat = a.val then 1 else 0 end) as isSubcatHeader
                ,r.GUIDE_TYPE
                ,isBroaderCat = 0
                ,a._sort_order
                --,'aaaa'[r]
                --,a.*
                --,'rrrr'[r]
                --,r.*
  from @idx_val a
  left join _resultt r
                on a.GUIDE_TYPE = r.guide_type
                and case
                        when replace(r.guide_item_text,'        ','') like '[0-9]) %' then
substring(replace(r.guide_item_text,'   ',''),4,len(replace(r.guide_item_text,'     ','')))
                        when replace(r.guide_item_text,'        ','') like '[0-9][0-9]) %' then
substring(replace(r.guide_item_text,'   ',''),5,len(replace(r.guide_item_text,'     ','')))
                        else replace(r.guide_item_text,'        ','')
                        end
                        =
                        case
                        when replace(a.val,'     ','') like '[0-9]) %' then substring(replace(a.val,'
        ',''),4,len(replace(a.val,' ','')))
                        when replace(a.val,'     ','') like '[0-9][0-9]) %' then substring(replace(a.val,'
        ',''),5,len(replace(a.val,' ','')))
                        else replace(a.val,'     ','')
                        end
                        --replaces tabs and ensures old guidesheet_text values without preceding
digits still match
  where
    1=1
                --and r.string in (
                --      'Food handlers do not have valid food handler permits. (no permits)
CORRECTED ON SITE.'
                --      ,'Food handlers without a valid food handler permit or certificate of
completion of an approved food handler course are being allowed to handle food that is served to
the public. (some dude without a permit was serving food) REPEAT.'
                --)
    and r.serv_prov_code = 'DAVISCO'
    and r.string is not null
    and exists (
      select 1
      from _resultt
      where
        1=1
        and sub_cat = a.sub_cat
    )
    and (
      a.val = a.sub_cat
      or r.sub_cat is not null
    )
)
--select * from _resultt
--select * from resultt

--, almost_there as (
insert into @almost_there
select
        ROW_NUMBER() over (order by
                r.guide_type
                ,_sort_order
                , idx
                , case
                        when toPrint like 'Violation: %' then 1
                        when toPrint like 'Comment: %' then 2
                        else 0
        end) rownum
        ,GUIDE_TYPE
        ,toPrint
        ,guide_item_status
        ,isBroaderCat
        ,isSubcatHeader
        --,'rrrr' [r]
        --,r.*
from resultt r
order by r._sort_order
```

```
;

select
         curRow.rownum idx
        ,curRow.GUIDE_TYPE
        ,curRow.toPrint as toPrint
        ,curRow.guide_item_status
        ,curRow.isBroaderCat
        ,curRow.isSubcatHeader
        ,(case when curRow.toPrint like 'Violation:%' and nextRow.toPrint like 'Violation:%' then 1
else 0 end) as isViolation_hideBottomBorder
        ,(case when curRow.toPrint like 'Violation:%' and prevRow.toPrint like 'Violation:%' then 1
else 0 end) as isViolation_hideTopBorder
from @almost_there curRow
left join @almost_there prevRow on prevRow.rownum = curRow.rownum-1
left join @almost_there nextRow on nextRow.rownum = curRow.rownum+1
order by curRow.rownum
```