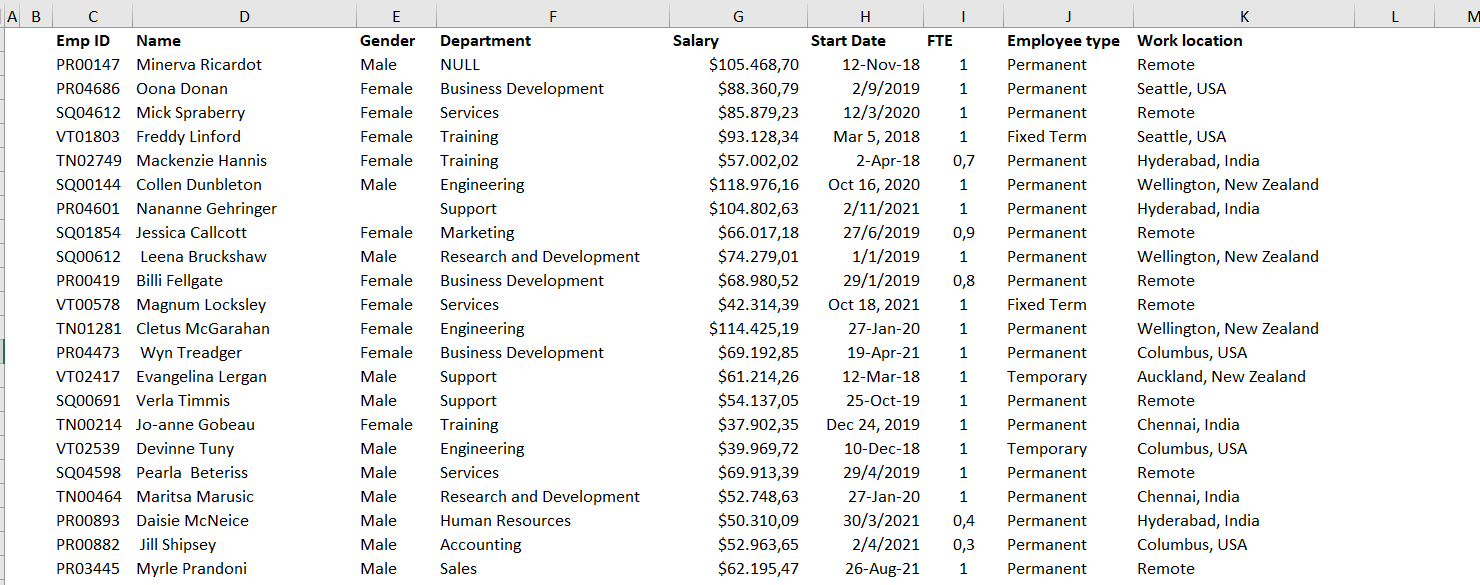
|  |
| --- |
| Dataset Schema |

****

|  |  |  |
| --- | --- | --- |
|  |  |  |



|  |
| --- |
| 1. Splitting Name column to Firstname and Lastname |

# split Name column to Firstname and Lastname (Splitting when blank is found)

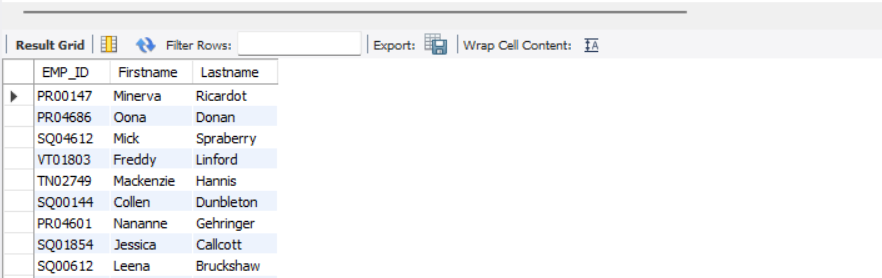
SELECT

EMP\_ID,

SUBSTR(Name, 1,INSTR(Name, " ")-1) as Firstname,

SUBSTR(Name, INSTR(Name," ")+1, LENGTH(Name)) as Lastname

FROM employees



# update table - split Name column to Firstname and Lastname

alter table employees

add column Firstname varchar(35) after Name;

update employees

set Firstname = SUBSTR(Name, 1,INSTR(Name, " ")-1)

where true;

alter table employees

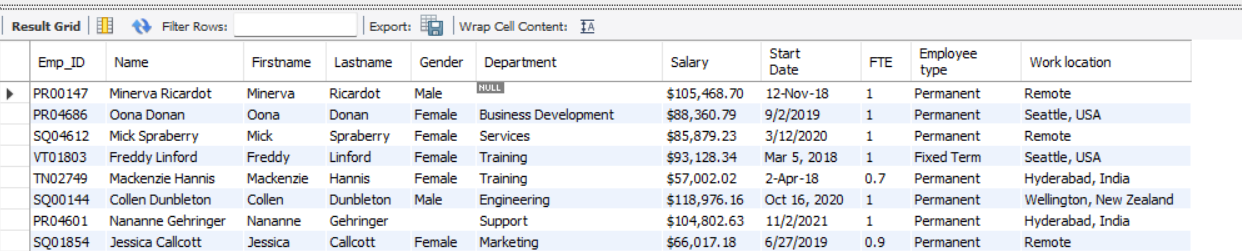
add column Lastname varchar(35) after Firstname;

update employees

set Lastname = SUBSTR(Name, INSTR(Name," ")+1, LENGTH(Name))

where true;

select \* from employees



|  |
| --- |
| 1. Splitting Work location to city and country columns |

# similarly split work location to city and country. If employee is working remotely both columns will indicate Remote (Splitting when “,” is found)

SELECT

SUBSTR(`Work location`, 1,INSTR(`Work location`, ",")-1) as city,

SUBSTR(`Work location`, INSTR(`Work location`,",")+1, LENGTH(`Work location`)) as country

FROM employees

# add a column named city after ‘Work location’

alter table employees

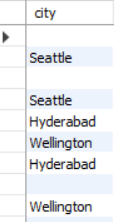
add column city varchar(35) after `Work location`;

update employees

set city = SUBSTR(`Work location`, 1,INSTR(`Work location`, ",")-1)

where true;

# this creates some blanks , so we are going to replace them with the “Remote” word.

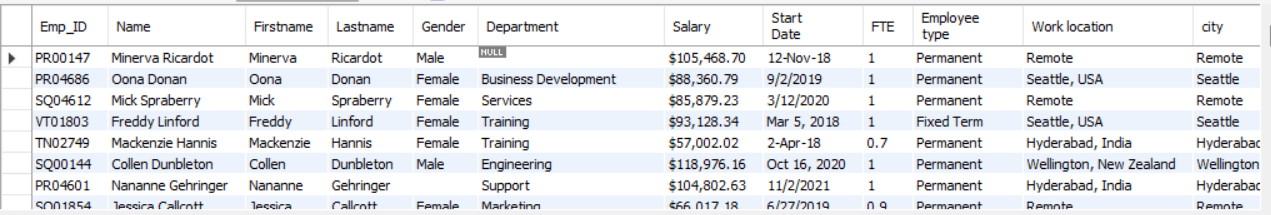


update employees

set city = "Remote"

where city IS NULL OR city = '';

select \* from employees



# add a column named country after column city

alter table employees

add column country varchar(35) after `city`;

update employees

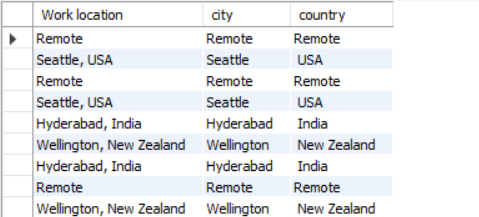
set country = SUBSTR(`Work location`, INSTR(`Work location`,",")+1, LENGTH(`Work location`))

where true;

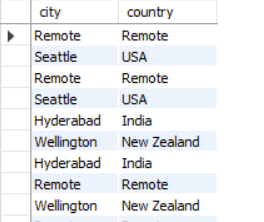
select

`Work location`, city, country

from employees



|  |
| --- |
| 1. Trimming leading and trailing whitespaces in city and country and more |

#As we see in the above image some whitespaces are produces in country column. We remove them using the TRIM function

update employees

set city = TRIM(city);

update employees

set country = TRIM(country);

select city,country

from employees

# doing the same for all the other columns

update employees

set Name = TRIM(Name);

update employees

set Firstname = TRIM(Firstname);

update employees

set Lastname = TRIM(Lastname);

update employees

set Gender = TRIM(Gender);

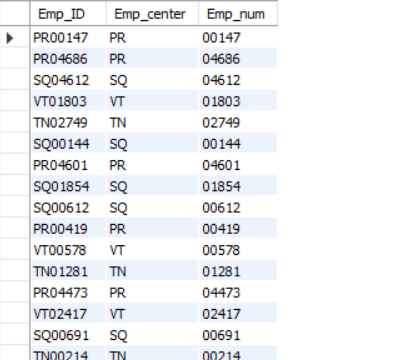
update employees

set Department = TRIM(Department);

update employees

set `Employee type` = TRIM(`Employee type`);

|  |
| --- |
| 1. Split Emp\_ID to Emp\_center and Emp\_num |

select

Emp\_ID,

left(Emp\_ID,2) as Emp\_center,

SUBSTR(Emp\_ID, 3, LENGTH(Emp\_ID)) as Emp\_num

from employees

# now update the table with new rows Emp\_center

and Emp\_num

alter table employees

add column Emp\_center varchar(35) after Emp\_ID;

update employees

set Emp\_center = left(Emp\_ID,2)

where true;

alter table employees

add column Emp\_num varchar(35) after Emp\_center;

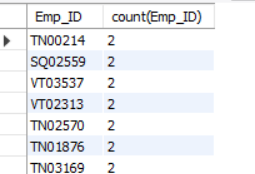
update employees

set Emp\_num = SUBSTR(Emp\_ID, 3, LENGTH(Emp\_ID))

where true;

|  |
| --- |
| 1. Remove Duplicates |

# checking for duplicates in employees (35 rows affected)

SELECT

Emp\_ID, count(Emp\_ID)

FROM

employees

GROUP BY

Emp\_ID

HAVING

COUNT(Emp\_ID) > 1;

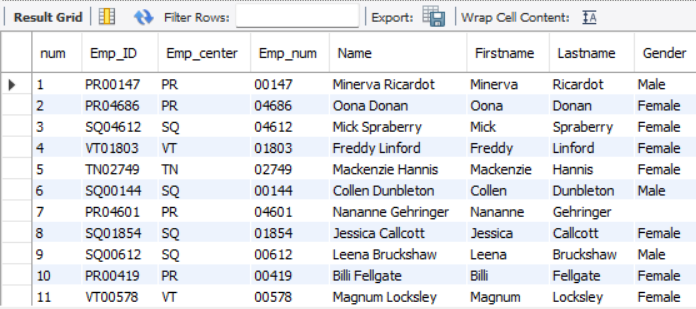
# adding a column named num and updating it with ascending numbers

alter table employees

add column num INT first

SET @pos := 0;

UPDATE employees SET num = ( SELECT @pos := @pos + 1 )



# now delete duplicates from employees (35 rows affected)

DELETE t1 FROM employees t1

INNER JOIN employees t2

WHERE

t1.num < t2.num AND

t1.Emp\_ID = t2.Emp\_ID;

# check again for duplicates (none)

SELECT

Emp\_ID, count(Emp\_ID)

FROM

employees

GROUP BY

Emp\_ID

HAVING

COUNT(Emp\_ID) > 1;

#new table has 241 rows now

|  |
| --- |
| 1. Adding column full\_part |

# add column full\_part , fte = 1 fulltime, fte < 1 partime

alter table employees

add column full\_part varchar(35) after FTE

update employees

set full\_part = "fulltime"

where FTE = 1

update employees

set full\_part = "partime"

where FTE < 1

select \*

from employees

where full\_part = NULL; (0 rows affected)

|  |
| --- |
| 1. Changing Date format |

We have 93 rows in the ’12-Nov-18’ format

We have 121 rows in the ‘9/2/2020’ format

We have 27 rows in the ‘Mar 5, 2018’ format

|  |  |  |
| --- | --- | --- |
| 93 rows | 121 rows | 27 rows |
|  |  |  |

So our 93+121+27 = 241 rows must be converted to the same date Format

**First update the 121 rows**

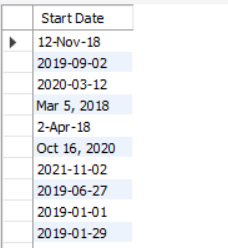
update employees

set `Start Date` = str\_to\_date(`Start Date` , '%m/%d/%Y')

where `Start Date` LIKE '%/%/%'

select `Start Date`

from employees



**Next update the 27 rows**

SELECT STR\_TO\_DATE(

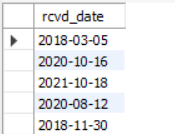
CONCAT( SUBSTR(`Start Date`, INSTR(`Start Date`,",")-2, 2) , '/', month(str\_to\_date(substr(`Start Date`,1,3) , '%b') ), '/', SUBSTR(`Start Date`, INSTR(`Start Date`,",")+2, LENGTH(`Start Date`)) ),

'%d/%m/%Y'

) as rcvd\_date

FROM employees

where `Start Date` LIKE '%,%'



**And now update (27 rows affected)**

update employees

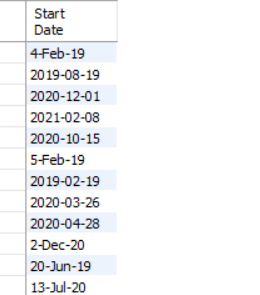
set `Start Date` = STR\_TO\_DATE(

CONCAT( SUBSTR(`Start Date`, INSTR(`Start Date`,",")-2, 2) , '/', month(str\_to\_date(substr(`Start Date`,1,3) , '%b') ), '/', SUBSTR(`Start Date`, INSTR(`Start Date`,",")+2, LENGTH(`Start Date`)) ),

'%d/%m/%Y'

)

where `Start Date` LIKE '%,%'



**Finally update the 4-Feb-19 Format**

update employees

set `Start Date` =

STR\_TO\_DATE(

CONCAT( SUBSTR(`Start Date`, 1, 1) , '/',

month(str\_to\_date(substr(`Start Date`,3,3) , '%b') ), '/',

year(str\_to\_date(SUBSTR(`Start Date`, 7, 2), '%y') ) ) ,

'%d/%m/%Y'

)

where `Start Date` LIKE '\_-%'

**and update the 14-Feb-19 Format**

update employees

set `Start Date` =

STR\_TO\_DATE(

CONCAT( SUBSTR(`Start Date`, 1, 2) , '/',

month(str\_to\_date(substr(`Start Date`,4,3) , '%b') ), '/',

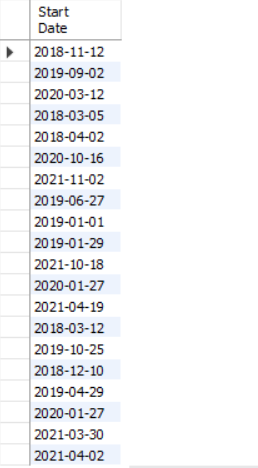
year(str\_to\_date(SUBSTR(`Start Date`, 8, 2), '%y') ) ) ,

'%d/%m/%Y'

)

where `Start Date` LIKE '\_\_-%'

**now all 241 rows are in date format**

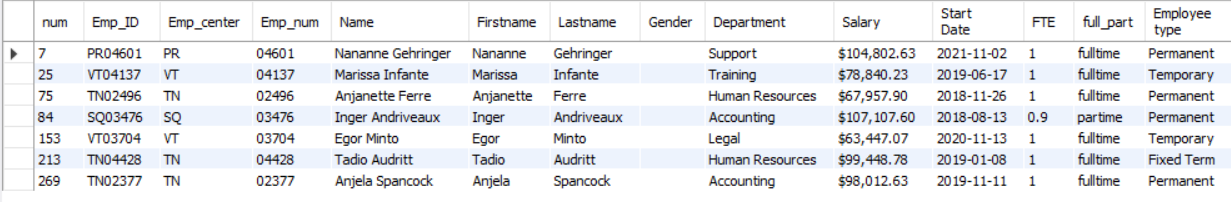


|  |
| --- |
| 1. Dealing with NULLs |

**In the Gender column there are some NULLs**

select \* from employees

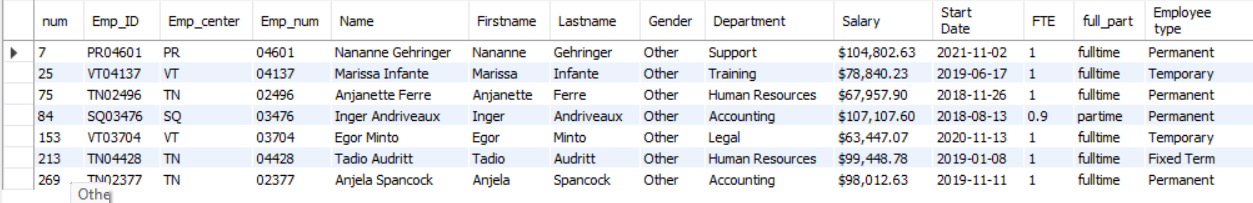
where Gender <> 'Male' and Gender <> 'Female'



update employees

set Gender = 'Other'

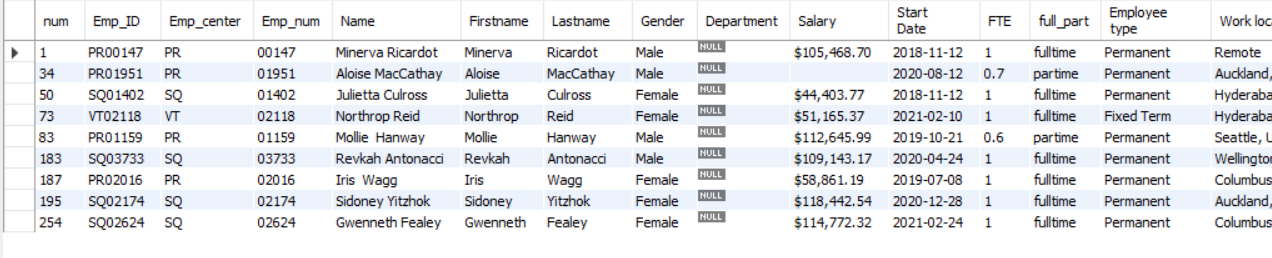
where Gender <> 'Male' and Gender <> 'Female'



**Some Nulls exist in the Department column, so we delete these rows.**

select \* from employees

where Department is NULL



delete from employees where Department is NULL (now 232 rows remain in database)

