

Topic: SQL

STEP1: -- Create database and then Create table ;

```
CREATE TABLE class1 (  
    Enrollment_No INTEGER PRIMARY KEY,  
    student_name varchar(20) NOT NULL,  
    section varchar(5) NOT NULL,  
    subject_id int(5),  
    marks int(5)  
);
```

NOTE : Enrollment_No is PRIMARY KEY

The screenshot shows an Online SQL Editor with the following components:

- Left Panel:** A tree view showing database schema elements:
 - Class [-]**: Enrollment_No [integer], student_name [varchar(20)], section [varchar(5)], subject_id [int(5)], marks [int(5)].
 - Customers [-]**: customer_id [int], first_name [varchar(100)], last_name [varchar(100)], age [int], country [varchar(100)].
 - Orders [-]**: order_id [integer], item [varchar(100)], amount [integer], customer_id [integer].
- Input Panel:** Contains the SQL statement:

```
CREATE TABLE class (  
    Enrollment_No INTEGER PRIMARY KEY,  
    student_name varchar(20) NOT NULL,  
    section varchar(5) NOT NULL,  
    subject_id int(5),  
    marks int(5)  
);
```
- Output Panel:** Displays the message: "Error: table class already exists".
- Available Tables Panel:** Shows a list of tables:
 - Class**: Enrollment_No, student_name, section, subject_ (empty).
 - Customers**: customer_id, first_name, last_name, age, cc (5 rows of data).
 - Orders**: order_id, item, amount, customer_id (empty).

STEP 2: -- insert into the value of created table class1;

```
INSERT INTO Class1 VALUES (1, 'TIM', 'A',1,70);
```

```
INSERT INTO Class1 VALUES (2, 'JIM', 'A',2,75);
```

```
INSERT INTO Class1 VALUES (3, 'KIM', 'B',3,65),
```

```
(4, 'TOM', 'B',4,77),
```

```
(5, 'JOHN', 'C',5,60)
```

```
,(6, 'JOE', 'C',1,82)
```

```
,(7, 'JAMES', 'B',2,76),
```

```
(8, 'HENRY', 'C', 5, 68),
(9, 'MATT', 'B', 3, 71),
(10, 'PAUL', 'A', 4, 79);
```

Programiz
Online SQL Editor

Class [-]

- Enrollment_No [integer]
- student_name [varchar(20)]
- section [varchar(5)]
- subject_id [int(5)]
- marks [int(5)]

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Input

```
INSERT INTO CLASS VALUES (1, 'TIM', 'A', 1, 70);
INSERT INTO CLASS VALUES (2, 'JIM', 'A', 2, 75);
INSERT INTO CLASS VALUES (3, 'KIM', 'B', 3, 65),
(4, 'TOM', 'B', 4, 77),
(5, 'JOHN', 'C', 5, 60),
(6, 'JOE', 'C', 1, 82),
(7, 'JAMES', 'B', 2, 76),
(8, 'HENRY', 'C', 5, 68),
(9, 'MATT', 'B', 3, 71),
(10, 'PAUL', 'A', 4, 79);
```

Output

SQL query successfully executed. However, the result set is empty.

Available Tables

Class

Enrollment_No	student_name	section	subject_
1	TIM	A	1
2	JIM	A	2
3	KIM	B	3
4	TOM	B	4
5	JOHN	C	5
6	JOE	C	1
7	JAMES	B	2
8	HENRY	C	5
9	MATT	B	3
10	PAUL	A	4

Customers

customer_id	first_name	last_name	age	cc
-------------	------------	-----------	-----	----

STEP 3:

-- fetch Query

SELECT section, COUNT(Enrollment_No) as NO_of_Candidate

FROM class1

where marks >= 75

group by section;

And Then Display The Output

Programiz
Online SQL Editor

Class [-]

- Enrollment_No [integer]
- student_name [varchar(20)]
- section [varchar(5)]
- subject_id [int(5)]
- marks [int(5)]

Class1 [-]

- Enrollment_No [integer]
- student_name [varchar(20)]
- section [varchar(5)]
- subject_id [int(5)]
- marks [int(5)]

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Input

```
marks >= 75
INSERT INTO class1 VALUES (1, 'TIM', 'A', 1, 70);
INSERT INTO class1 VALUES (2, 'JIM', 'A', 2, 75);
INSERT INTO class1 VALUES (3, 'KIM', 'B', 3, 65),
(4, 'TOM', 'B', 4, 77),
(5, 'JOHN', 'C', 5, 60),
(6, 'JOE', 'C', 1, 82),
(7, 'JAMES', 'B', 2, 76),
(8, 'HENRY', 'C', 5, 68),
(9, 'MATT', 'B', 3, 71),
(10, 'PAUL', 'A', 4, 79);

SELECT section, COUNT( Enrollment_No) as NO_of_Candidate
FROM class1
where marks >= 75
group by section;
```

Output

section	NO_of_Candidate
A	2
B	2
C	1

Available Tables

Enrollment_No	student_name	section	subject_id	marks
1	TIM	A	1	70
2	JIM	A	2	75
3	KIM	B	3	65
4	TOM	B	4	77
5	JOHN	C	5	60
6	JOE	C	1	82
7	JAMES	B	2	76
8	HENRY	C	5	68
9	MATT	B	3	71
10	PAUL	A	4	79

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

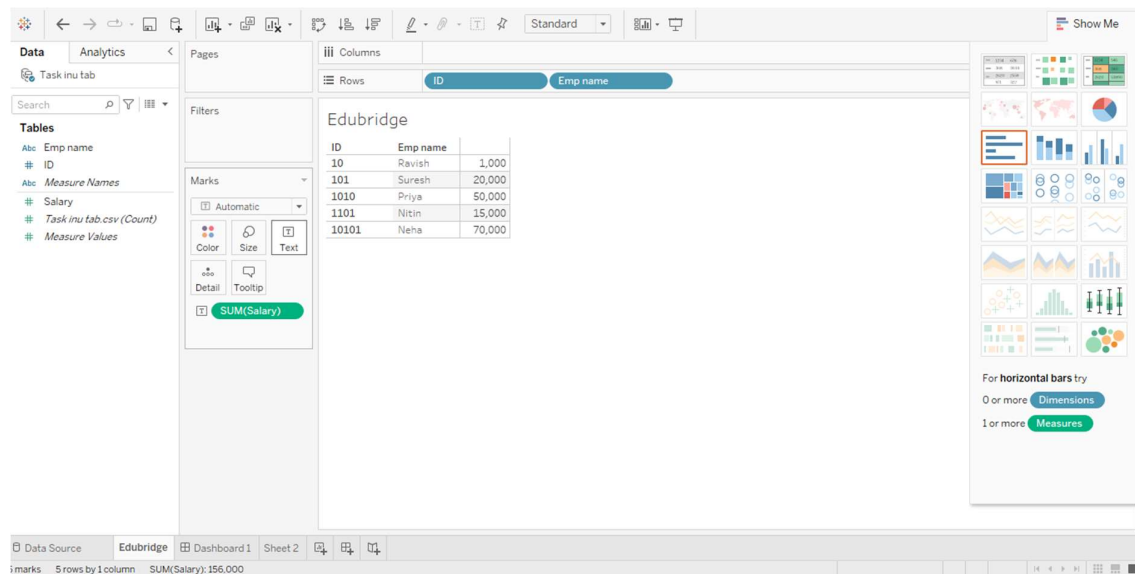
Topic: Tableau

First Create A Csv File On Excel And Save Local Disk

Import The Data From Local Disk To Tableau Worksheet And Use Worksheet

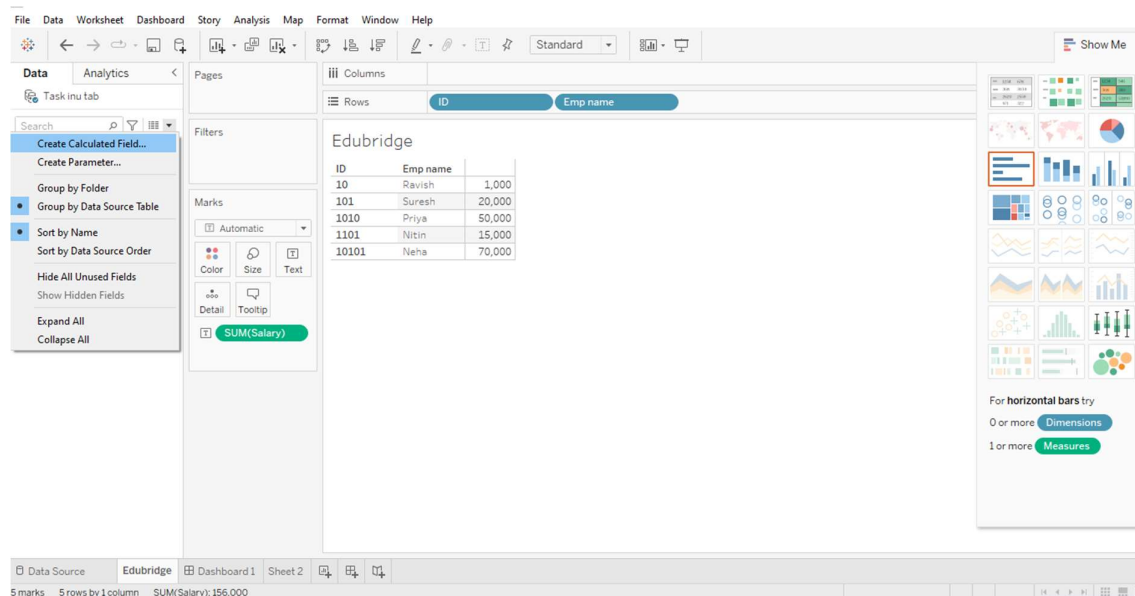
STEP1:

Need Column Drag & Drop (ID & Empname) Into The Rows And Same Time Salary Drag To Marks Label.



STEP 2:

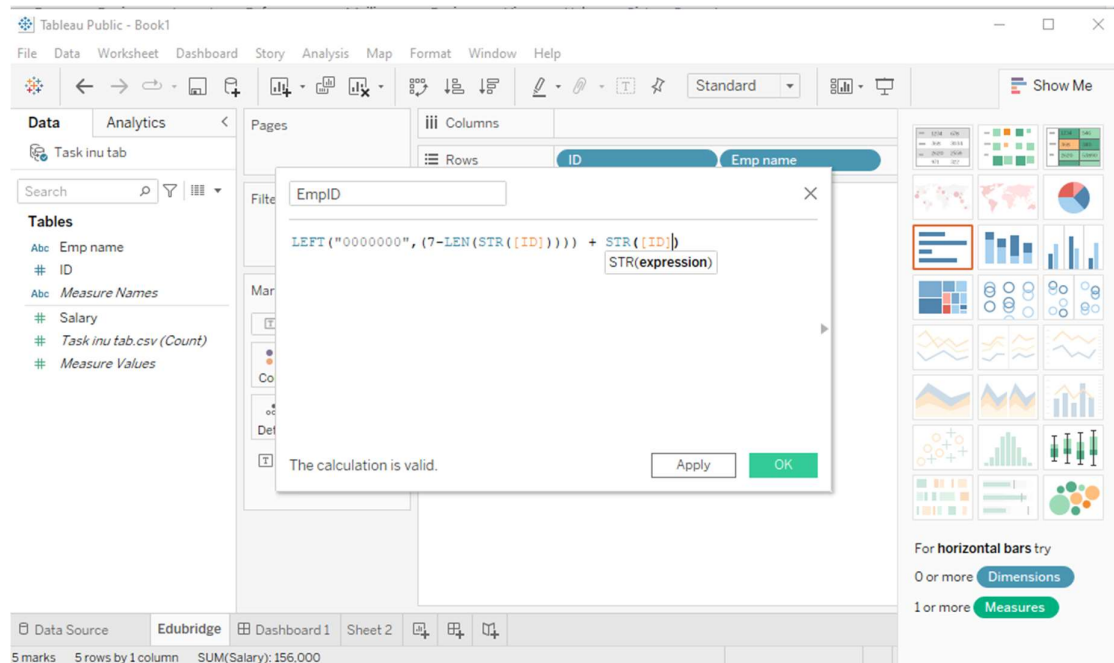
Then Near Filter By Click Icon Show The Options Here And Create Calculated Field.. Click



STEP 3:

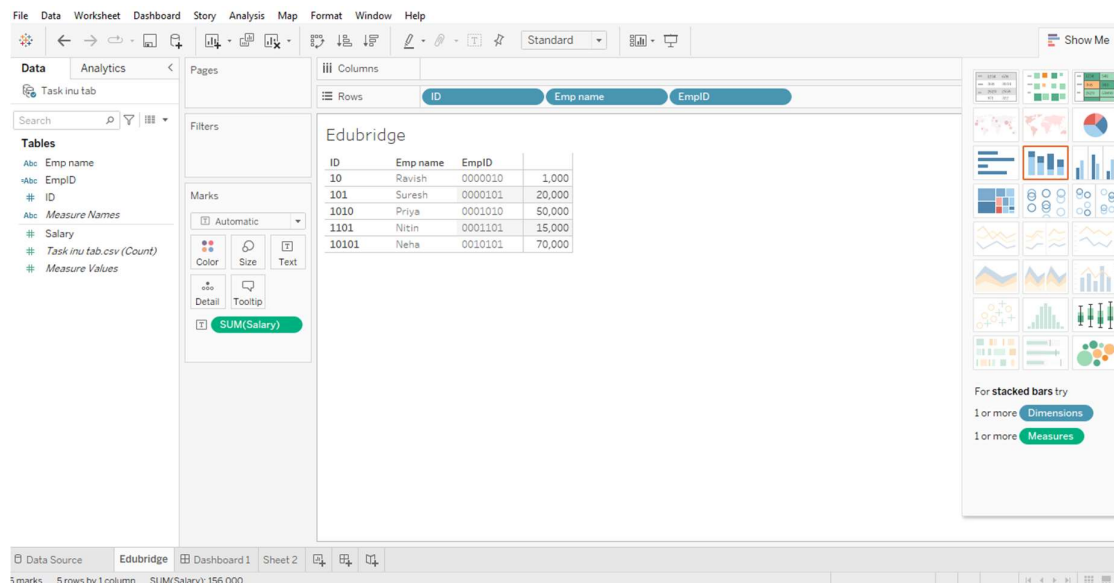
The Dialogue Box Will Be Open Put Column Name Empid New And, Then Text Box Write The Formulas

`LEFT("0000000",(7-LEN(STR([ID])))) + STR([ID])`



STEP 4:

New Measure Name Will Be Created And Just Drag And Drop The Rows Show The Table Create A Seven Digit EMP ID { .Format (0000000)}

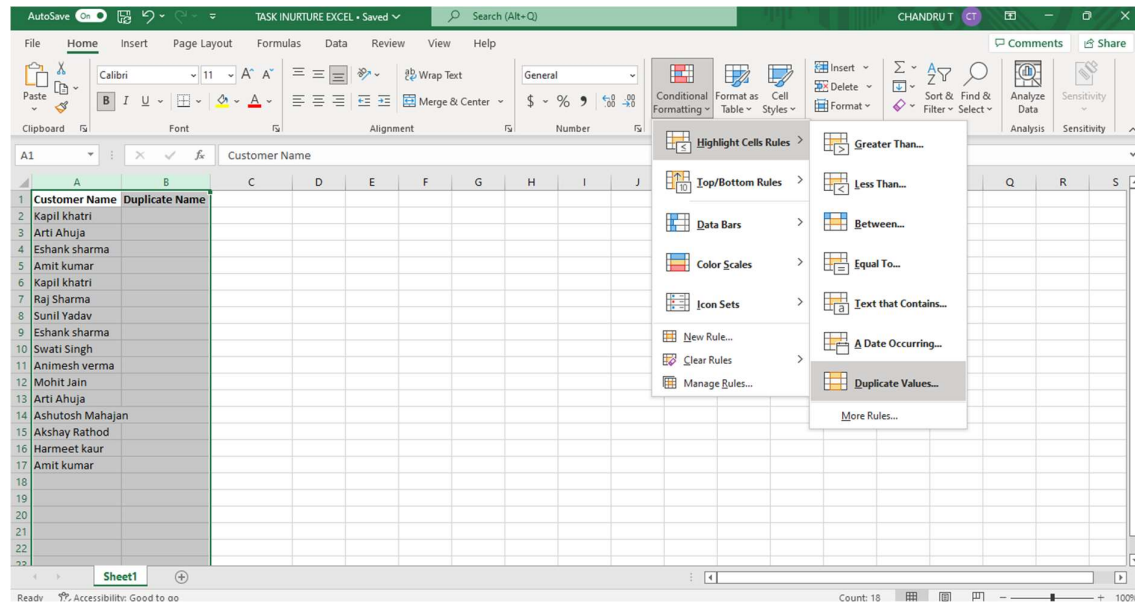


Topic: Excel

STEP 1:

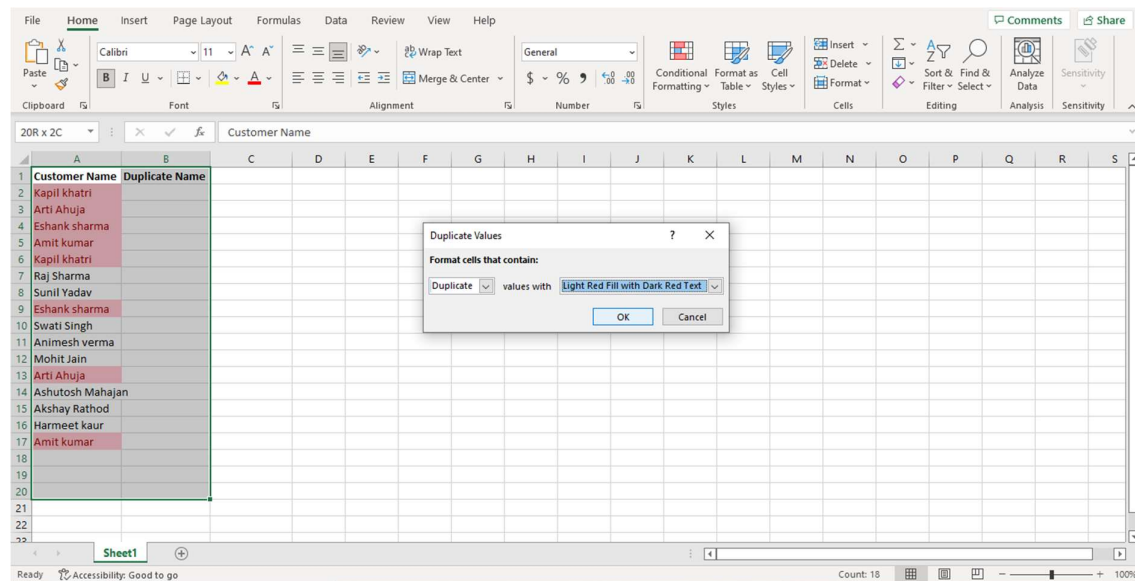
Create the Excel file include data Customer name.

And then select the entries cells, open home Go to conditional formatting and then choose Highlight cells rules click and choose duplicate values (go with click the option or use shortcut key your options).



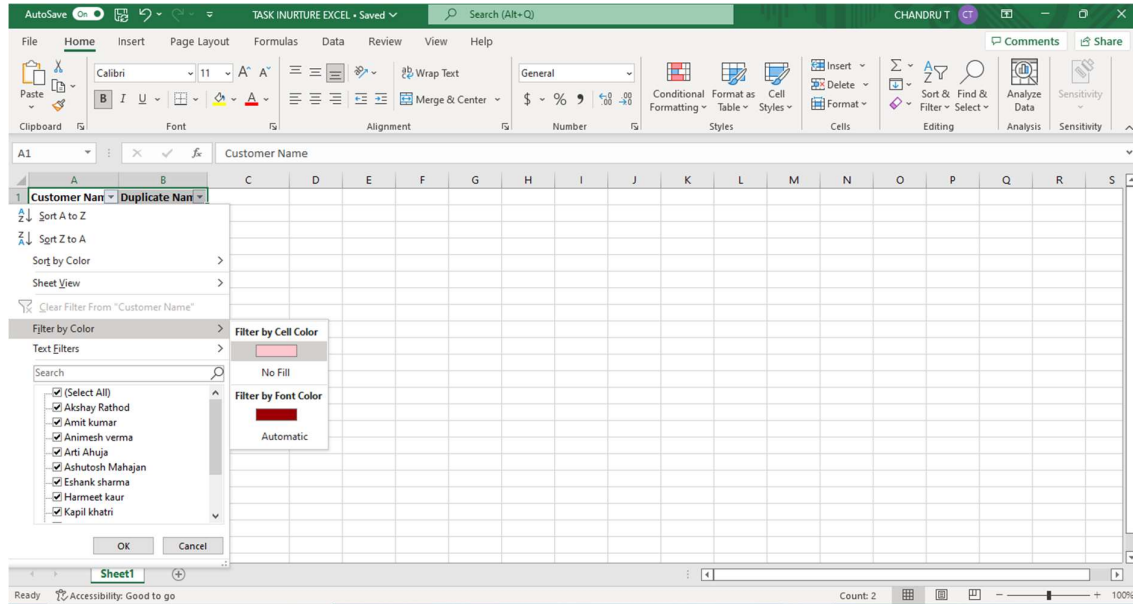
STEP 2:

choose duplicate value with highlights the red colour (choose duplicate values or unique values options)



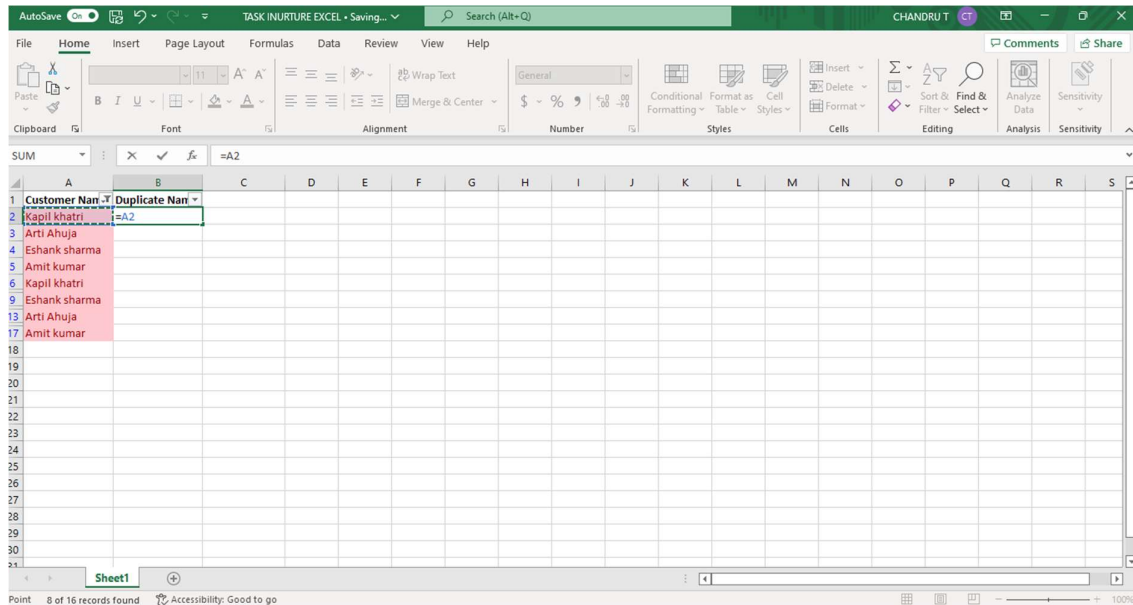
STEP 3:

Go to filter option or use shortcut key (CTRL + SHIFT+L) and then filter by fill colour or no fill choose one



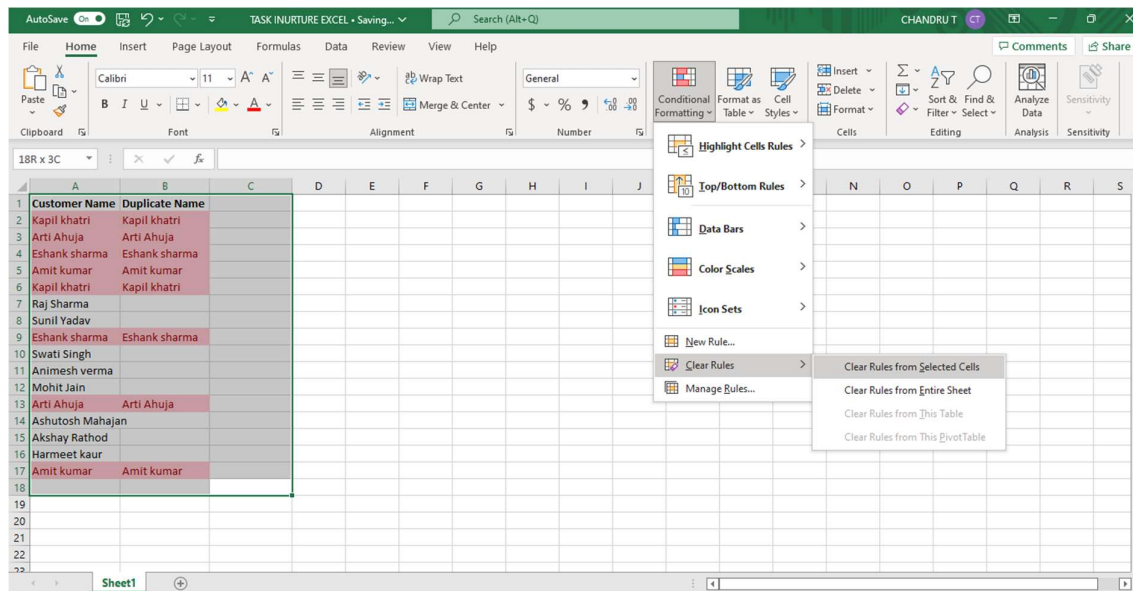
STEP 4:

And then use copy paste formulae (= cell name (A2))and drag the down cell paste after remove the filter or use shortcut key (CTRL + SHIFT+L) again ,



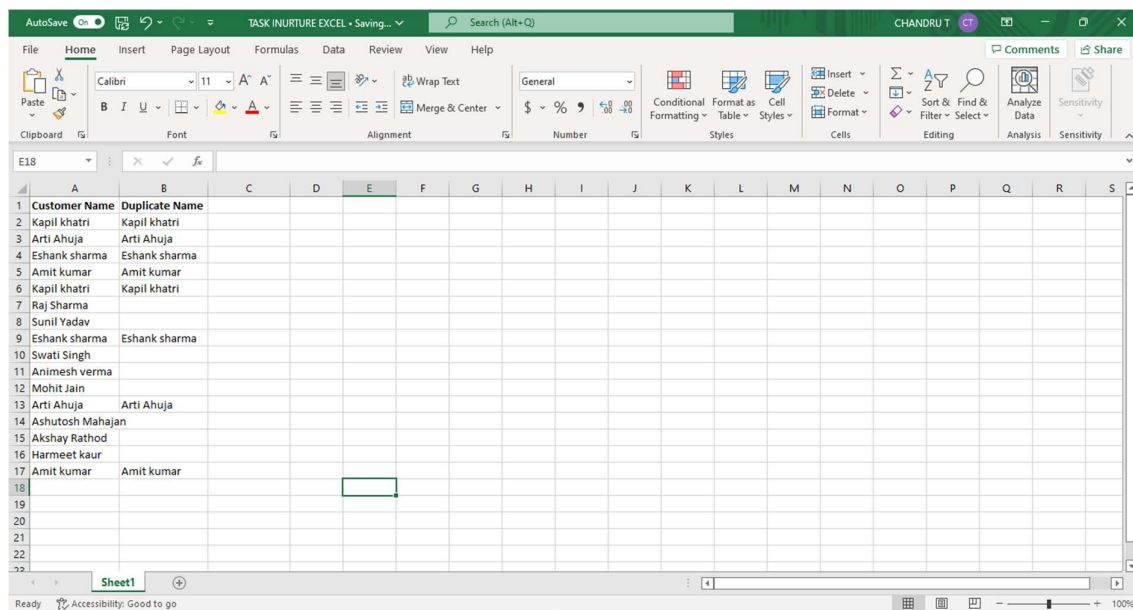
STEP 5:

Again go to conditional formatting and then choose clear rule the remove highlights colour and then clear rules from selected cells



STEP 6:

Final outputs see the below images .



Topic: Machine Learning

Description You need to classify fetal health to avoid the abnormalities for the child and mother while giving birth. You will be provided with a CSV consisting of 22 columns out of which the first 21 columns are the factors based on which you need to classify the fetal health which is the last column.

You need to classify the health status into the following:

1.0 - Normal

2.0 - Suspect

3.0 – Pathological