FAITH KANE

2021

MYSQL

Faith Kane
2021

THE BIG 6 ELEMENTS OF A SQL SELECT STATEMENT

Necessary Optional

1	SELECT	COLUMN_NAME(S)
2	FROM	TABLE_NAME
3	WHERE	LOGICAL_CONDITION(S)
4	GROUP BY	COLUMN_NAME(S)
5	HAVING	LOGICAL_CONDITION(S)
6	ORDER BY	COLUMN_NAME(S)

Faith Kane
2021

LET'S FOCUS ON THE GROUP BY CLAUSE

Necessary Optional

1 SELECT COLUMN_NAME(S)

2 FROM TABLE_NAME

3 WHERE LOGICAL_CONDITION(S)

4 GROUP BY COLUMN_NAME(S)

5 HAVING LOGICAL_CONDITION(S)

6 ORDER BY COLUMN_NAME(S)

Faith Kane
2021

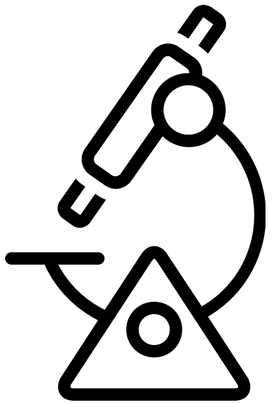
LET'S FOCUS ON THE GROUP BY CLAUSE

Necessary Optional

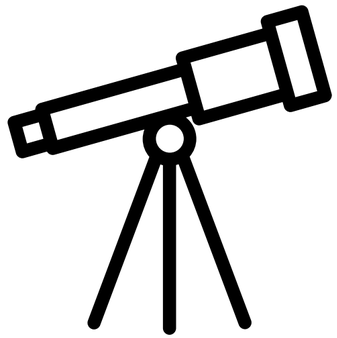
1	SELECT	IDENTIFIES COLUMN(S)
2	FROM	IDENTIFIES TABLE
3	WHERE	RECORD-FILTERING CRITERIA
4	GROUP BY	SPECIFIES HOW TO GROUP DATA
5	HAVING	GROUP-FILTERING CRITERIA
6	ORDER BY	SPECIFIES ORDER OF RESULTS

Faith Kane
2021

SO WHAT'S SO GREAT ABOUT GROUP BY?



Zoom In



Zoom Out

Faith Kane
2021

HOW DOES GROUP BY WORK?

What if I want to group the observations in my dataset by item_name...

	item_name	item_price
0	Chips and Fresh Tomato Salsa	2.39
1	Izze	3.39
2	Nantucket Nectar	3.39
3	Chips and Tomatillo-Green Chili Salsa	2.39
4	Chicken Bowl	8.49
5	Chicken Bowl	10.98
6	Side of Chips	1.69
7	Steak Burrito	11.75
8	Steak Soft Tacos	9.25
9	Steak Burrito	9.25

Faith Kane
2021

HOW DOES GROUP BY WORK?

...such that each row represents one unique item_name?

	item_name
0	6 Pack Soft Drink
1	Barbacoa Bowl
2	Barbacoa Burrito
3	Barbacoa Crispy Tacos
4	Barbacoa Salad Bowl
5	Barbacoa Soft Tacos
6	Bottled Water
7	Bowl
8	Burrito
9	Canned Soda
10	Canned Soft Drink
11	Carnitas Bowl
12	Carnitas Burrito
13	Carnitas Crispy Tacos
14	Carnitas Salad
15	Carnitas Salad Bowl
16	Carnitas Soft Tacos
17	Chicken Bowl

Faith Kane
2021

HOW DOES GROUP BY WORK?

I have 2 observations with the item_name Crispy Tacos.

	item_name	item_price
520	Crispy Tacos	7.4
521	Crispy Tacos	7.4

Faith Kane
2021

HOW DOES GROUP BY WORK?

I have 7 observations with the item_name Veggie Soft Tacos.

	item_name	item_price
738	Veggie Soft Tacos	11.25
781	Veggie Soft Tacos	8.75
1395	Veggie Soft Tacos	8.49
1699	Veggie Soft Tacos	11.25
2384	Veggie Soft Tacos	8.75
2851	Veggie Soft Tacos	8.49
3889	Veggie Soft Tacos	8.49

Faith Kane
2021

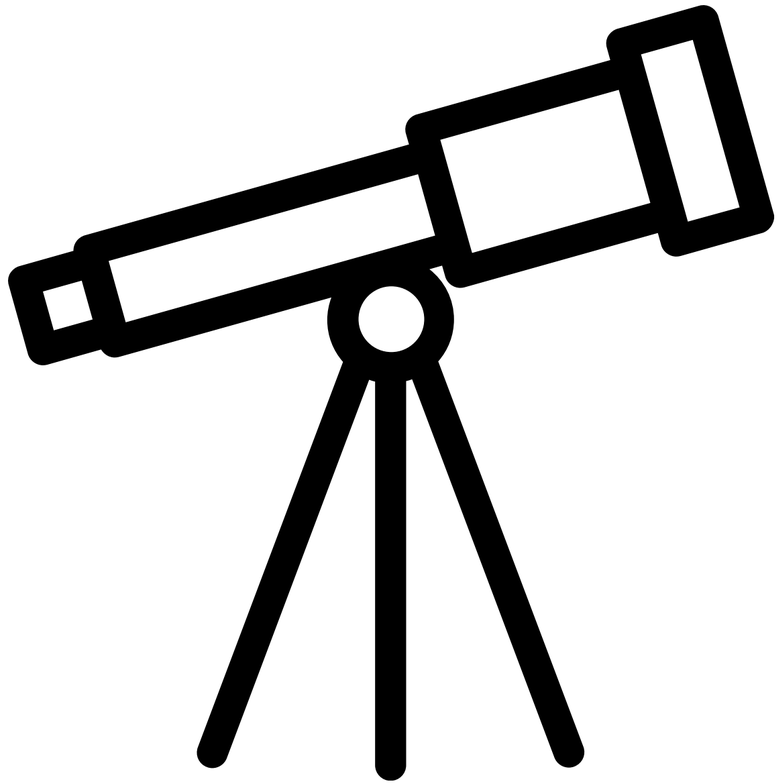
HOW DOES GROUP BY WORK?

And I have hundreds of observations with the item_name Chicken Bowl.

	item_name	item_price
4	Chicken Bowl	8.49
5	Chicken Bowl	10.98
13	Chicken Bowl	11.25
19	Chicken Bowl	8.75
26	Chicken Bowl	8.49
...	...	...
4590	Chicken Bowl	11.25
4591	Chicken Bowl	8.75
4595	Chicken Bowl	8.75
4599	Chicken Bowl	8.75
4604	Chicken Bowl	8.75

Faith Kane
2021

ZOOM OUT



Faith Kane
2021

THESE VALUES CAN'T ALL FIT IN ONE CELL!

item_name	item_price
Chicken Bowl	8.75 8.49 11.25 8.49 8.49 10.98 8.49 8.75
Crispy Tacos	7.40 7.40
Veggie Soft Tacos	11.25 8.75 11.25 8.75 8.49 11.25 8.49

Faith Kane
2021

USE AN AGGREGATE FUNCTION TO SUMMARIZE

item_name	item_price_mean
Chicken Bowl	9.66
Crispy Tacos	7.40
Veggie Soft Tacos	9.35

Faith Kane
2021

USE AN AGGREGATE FUNCTION TO SUMMARIZE

item_name	item_count
Chicken Bowl	726
Crispy Tacos	2
Veggie Soft Tacos	7

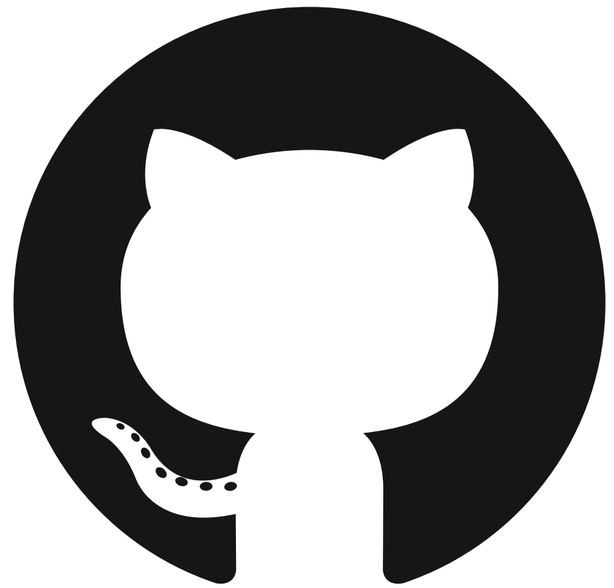
Faith Kane
2021

USE AN AGGREGATE FUNCTION TO SUMMARIZE

item_name	item_price_sum
Chicken Bowl	7011.51
Crispy Tacos	14.80
Veggie Soft Tacos	65.47

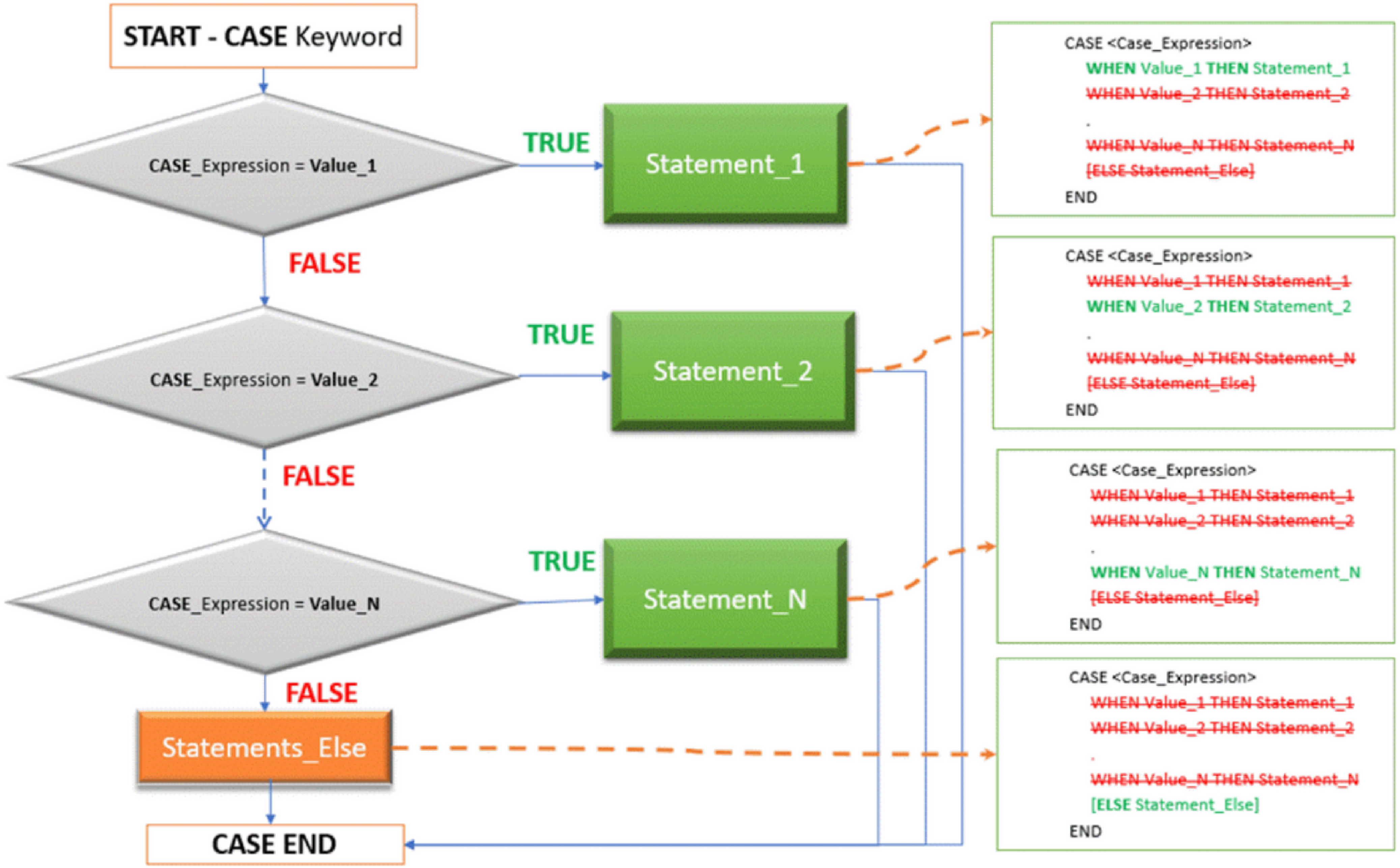
Faith Kane
2021

CHECK OUT MY NOTEBOOK HERE



Faith Kane
2021

LET'S LOOK AT CASE STATEMENTS



Faith Kane
2021

WHERE DO CASE STATEMENTS FIT INTO OUR BIG 6 ELEMENTS?

Necessary Optional

1
```
SELECT    COLUMN_NAME(S),
  CASE
      WHEN COLUMN_NAME CONDITION_A THEN VALUE_1
      WHEN COLUMN_NAME CONDITION_B THEN VALUE_2
      ELSE VALUE_3
      END AS NEW_COLUMN_NAME
```

2
```
FROM      TABLE_NAME
```

3
```
WHERE     LOGICAL_CONDITION(S)
```

4
```
GROUP BY  COLUMN_NAME(S)
```

5
```
HAVING    LOGICAL_CONDITION(S)
```

6
```
ORDER BY  COLUMN_NAME(S)
```

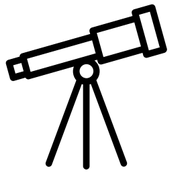
Faith Kane
2021

USE THE SAME CONDITIONAL OPERATORS AS IN WHERE CLAUSES

Operator	What it Means
=	Equals
<>	Does NOT Equal
>	Greater Than
<	Less Than
>=	Greater Than Or Equal To
<=	Less Than Or Equal To
BETWEEN	A Range Between Two Values
LIKE	Matching a Pattern Like This
IN()	Equals One of These Values

Faith Kane
2021

WHY DO WE USE CASE STATEMENTS?



WHEN I WANT TO ZOOM OUT OR

REDUCE THE NOISE IN MY DATA

WHEN I WANT TO BUCKET OR BIN

MY VALUES

Faith Kane
2021

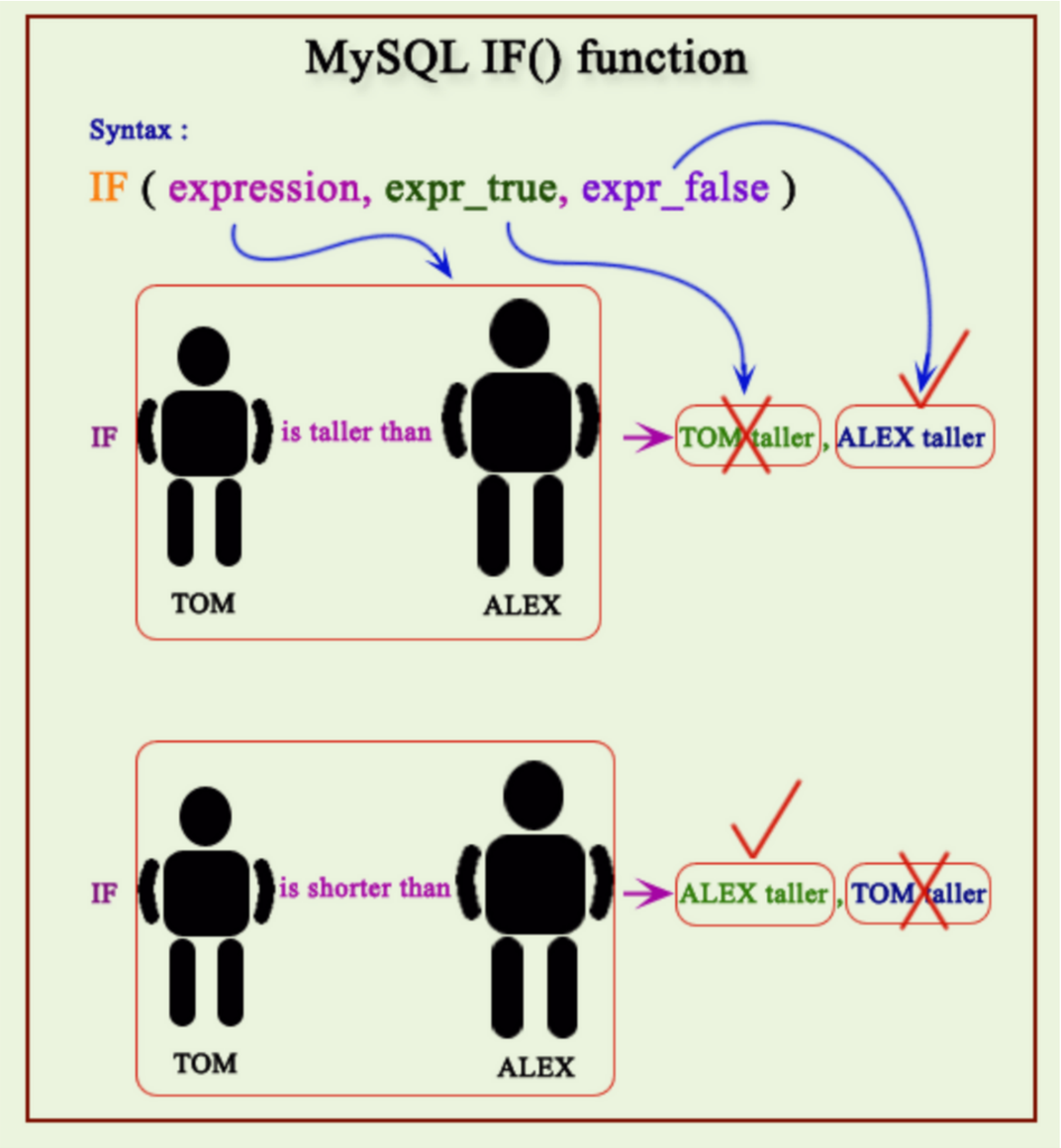
WHEN DO WE USE CASE STATEMENTS?

1. WHEN I HAVE MORE THAN TWO OPTIONAL VALUES

2. WHEN I NEED MORE FLEXIBILITY IN MY CONDITIONAL TESTS

Faith Kane
2021

LET'S LOOK AT THE IF FUNCTION



Faith Kane
2021

Necessary Optional

WHERE DOES THE IF FUNCTION FIT INTO OUR BIG 6 ELEMENTS?

```
1 SELECT
    COLUMN_NAME(S),
    IF (CONDITION, VAL_IF_TRUE, VAL_IF_FALSE) AS NEW_COLUMN
2 FROM      TABLE_NAME
3 WHERE     LOGICAL_CONDITION(S)
4 GROUP BY  COLUMN_NAME(S)
5 HAVING    LOGICAL_CONDITION(S)
6 ORDER BY  COLUMN_NAME(S)
```

Faith Kane
2021

WHY DO WE USE THE IF FUNCTION?

1. WHEN I AM EVALUATING A SINGLE CONDITION TO TRUE OR FALSE

2. IF I WANT A COLUMN OF BOOLEAN VALUES (DUMMY VARIABLE)

Faith Kane
2021

CHECK OUT MY NOTEBOOK HERE