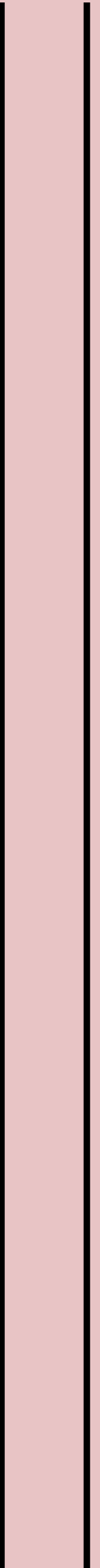




Wine Shops Project

CFG SQL Kickstarter May 2024

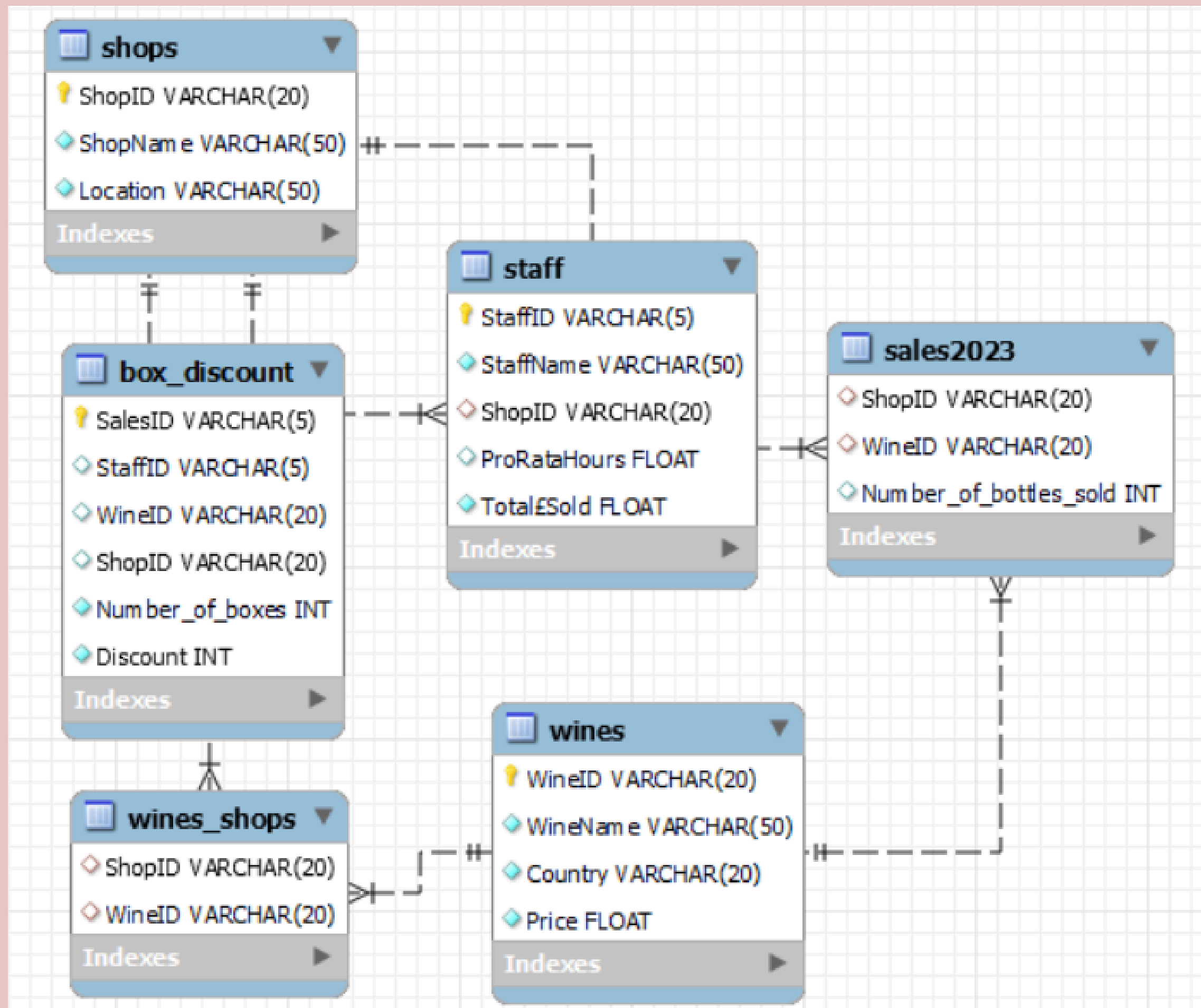


For this project I created made up data from 3 fictional wine shops in London.

I chose this project because I used to manage a wine bar/shop in London and I felt it would be easier for me to learn SQL if I used data that seemed both relevant and realistic to me.

It is also a love letter to English wine.





Tables

1. Shops

2. Wines

3. Wines_Shops

4. Sales2023

5. Staff

6. Box_discount

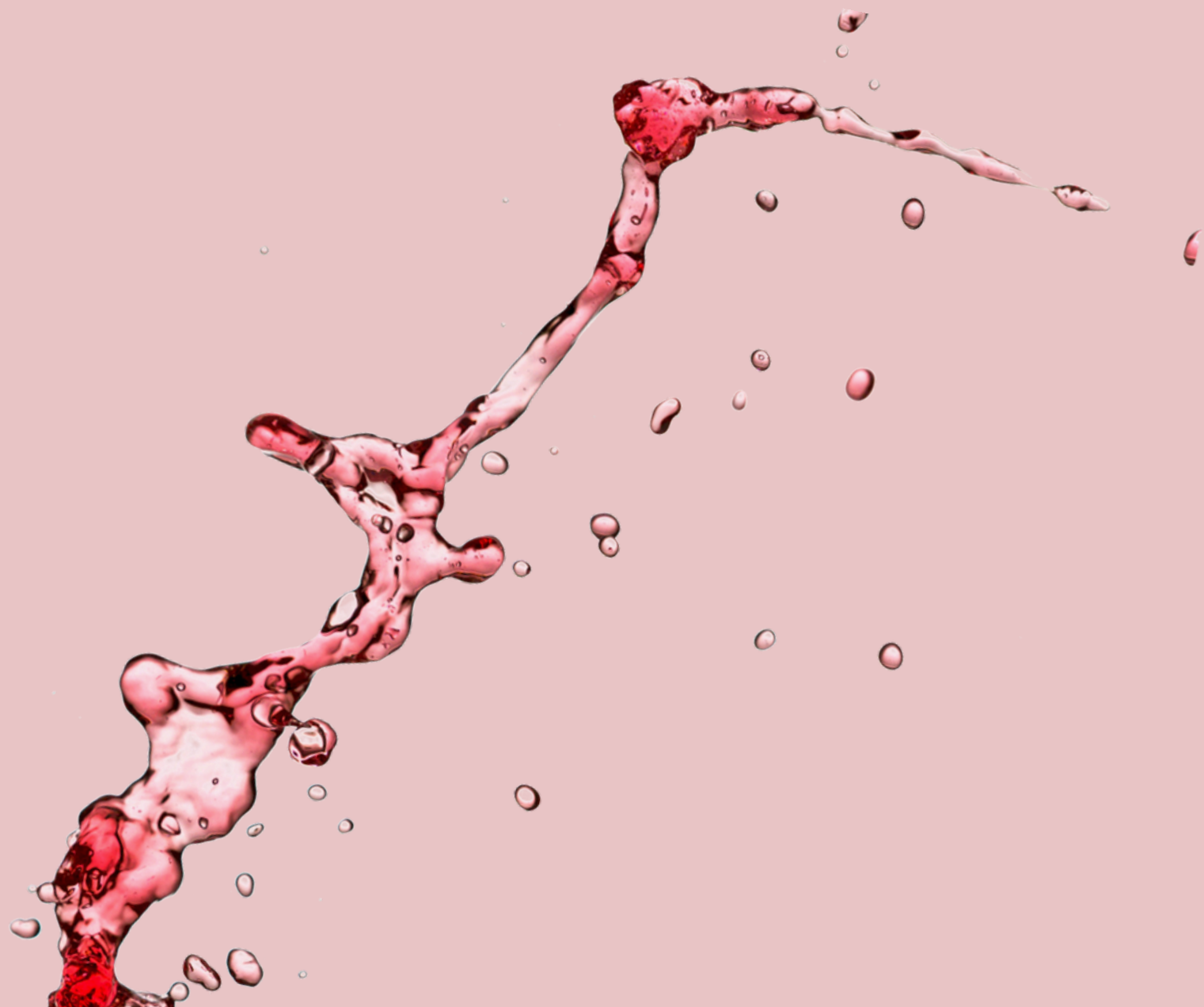


Table 1 – Shops

The first table tells us about each shop and has ShopID as the Primary Key, the name and location of each shop

ShopID	ShopName	Location
HW002	Hansen Wines	Chingford
WG001	Wines on the Green	Angel
WW003	Weird Wines	Walthamstow

Table 2 – Wines

The second table is a list of each wine sold by the company. It has WineId as the Primary Key, the name of the wine, the wine's country of origin and the price per bottle.

WineID	WineName	Country	Price
red1	Gamay	France	18.5
red2	Pinot Noir	England	22
red3	Tempranillo	Spain	15
red4	Touriga Nacional	Portugal	16
sparkling1	Chalklands	England	31
sparkling2	Canterbury Rose	England	29
sparkling3	Pet Nat	Portugal	17
sparkling4	Champagne	France	45
sparkling5	Cava	Spain	16
sparkling6	Pink Cava	Spain	15
white1	Bacchus	England	15
white2	Furmint	Hungary	17.5
white3	Gruner Veltliner	Austria	19

Table 3 – Wines_Shops

The third table tells us which wines are available in each shop.
It has ShopID and WineID, both are Foreign Keys.

ShopID	WineID
WG001	red1
WG001	red2
WG001	red4
WG001	white1
WG001	white2
WG001	sparkling1
WG001	sparkling2
WG001	sparkling3
WG001	sparkling4

HW002	red1
HW002	red2
HW002	red3
HW002	red4
HW002	white1
HW002	white2
HW002	white3
HW002	sparkling1
HW002	sparkling2
HW002	sparkling3
HW002	sparkling4
HW002	sparkling5
HW002	sparkling6

WW003	red2
WW003	red3
WW003	red4
WW003	white1
WW003	white2
WW003	white3
WW003	sparkling1
WW003	sparkling2
WW003	sparkling5
WW003	sparkling6

Table 4 – Sales2023

The fourth table has ShopID (FK), WineID (also FK) and the number of bottles sold of each wine in 2023

ShopID	WineID	Numb
WG001	red1	168
WG001	red2	454
WG001	red4	237
WG001	white1	713
WG001	white2	497
WG001	sparkling1	154
WG001	sparkling2	132
WG001	sparkling3	174
WG001	sparkling4	45

HW002	red1	371
HW002	red2	239
HW002	red3	187
HW002	red4	98
HW002	white1	235
HW002	white2	189
HW002	white3	321
HW002	sparkling1	78
HW002	sparkling2	43
HW002	sparkling3	110
HW002	sparkling4	67
HW002	sparkling5	104
HW002	sparkling6	88

WW003	red2	477
WW003	red3	330
WW003	red4	189
WW003	white1	401
WW003	white2	372
WW003	white3	245
WW003	sparkling1	110
WW003	sparkling2	56
WW003	sparkling5	134
WW003	sparkling6	162

Table 5 – Staff

The fifth table shows us details about the staff members. It has their StaffID, their name, ProRataHours tell us if they work full or part time. It also shows how much they have sold for during an arbitrary period.

StaffID	StaffName	ShopID	ProRataHours	Total£Sold
S01	Tilly	HW002	1	2145
S03	Megan	WG001	1	2315
S04	Sabina	WW003	0.6	717.5
S05	Chineye	HW002	0.6	695.5
S06	Neil	WG001	1	2417.5
S08	Sam	WW003	1	1890
S09	Yonas	HW002	0.6	717.5
S10	Siva	WW003	0.6	957.5

Table 6 – Box_discount

If customers buy a box of wine of the same kind they get 10% discount, if they buy 2+ boxes they get 20% discount.

Staff members who sell a box also get a £2 bonus per box.

SalesID	StaffID	WineID	ShopID	Number_of_boxes	Discount
1106	S03	red1	WG001	1	10
1156	S01	sparkling3	HW002	2	20
1177	S06	sparkling4	WG001	1	10
1210	S08	red4	WW003	1	10
1231	S08	sparkling6	WW003	1	10
1275	S010	red2	WW003	1	10
1283	S01	sparkling1	HW002	1	10
1326	S05	red2	HW002	1	10

View

This view using an inner join shows us which staff members work where

```
CREATE VIEW StaffLocation
AS
SELECT
s.Location,
st.StaffName
FROM Shops AS s
INNER JOIN
Staff AS st
ON
s.ShopID =
st.ShopID;
```

Location	StaffName
Chingford	Tilly
Chingford	Chineye
Chingford	Yonas
Chingford	Charles
Angel	Megan
Angel	Neil
Walthamstow	Sabina
Walthamstow	Sam
Walthamstow	Siva

Stored function

Wines with a price of more than £20 are much harder to sell.

This stored function tells us which wines are going to be harder to sell.

This is good to know so each shop can balance the number of harder to sell wines and you can also set up a staff incentive with a prize to the person who sells the most 'hard to sell' wines.

```
DELIMITER //
```

```
CREATE FUNCTION hard_to_sell_wines(price float)
```

```
RETURNS varchar(3)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE hard_to_sell varchar(10);
```

```
    IF price > 20 THEN
```

```
        SET hard_to_sell = 'Yes';
```

```
    ELSEIF price <= 20 THEN
```

```
        SET hard_to_sell = 'No';
```

```
    END IF;
```

```
    RETURN (hard_to_sell);
```

```
END //
```

```
DELIMITER ;
```

```
SELECT
```

```
    WineName,
```

```
    Price,
```

```
    hard_to_sell_wines(price) AS HardToSell
```

```
FROM
```

```
    Wines;
```

WineName	HardToSell
Gamay	No
Pinot Noir	Yes
Tempranillo	No
Touriga Nacional	No
Chalklands	Yes
Canterbury Rose	Yes
Pet Nat	No
Champagne	Yes
Cava	No
Pink Cava	No
Bacchus	No
Furmint	No
Gruner Veltliner	No

Query with a subquery

How many English wines were sold across all 3 shops?

Is there a market for English wine?

```
SELECT WineID, SUM(number_of_bottles_sold) AS Bottles_sold
FROM Sales2023
WHERE WineID IN (SELECT WineID
FROM Wines
WHERE country = 'England')
GROUP BY WineID;
```



WineID	Bottles_sold
red2	1170
sparkling1	342
sparkling2	231
white1	1349

Procedure

This procedure will save time when adding new wines.

```
CREATE PROCEDURE InsertValues
(
  IN WineID varchar(20),
  IN WineName varchar(50),
  IN Country varchar(20),
  IN Price float
)
BEGIN
  INSERT INTO Wines(WineID, WineName, Country, Price)
  VALUES (WineID, WineName, Country, Price);
END//
DELIMITER ;
```

Calling the procedure

```
CALL InsertValues ('white4', 'Chardonnay', 'England', 19.00);
```

WineID	WineName	Country	Price
red1	Gamay	France	18.5
red2	Pinot Noir	England	22
red3	Tempranillo	Spain	15
red4	Touriga Nacional	Portugal	16
sparkling1	Chalklands	England	31
sparkling2	Canterbury Rose	England	29
sparkling3	Pet Nat	Portugal	17
sparkling4	Champagne	France	45
sparkling5	Cava	Spain	16
sparkling6	Pink Cava	Spain	15
white1	Bacchus	England	15
white2	Furmint	Hungary	17.5
white3	Gruner Veltliner	Austria	19
white4	Chardonnay	England	19

Trigger

The company requires all staff to work a minimum of 0.6 pro rata hours.

If a new staff member is added and they request less hours, the TRIGGER will update the hours to be 0.6 according to company policy.

```
DELIMITER //  
CREATE TRIGGER minimum_hours  
BEFORE INSERT ON Staff  
FOR EACH ROW  
IF NEW.ProRataHours < 0.6 THEN SET NEW.ProRataHours = 0.6;  
END IF; //
```

A new staff member has requested to work less than 0.6 pro rata hours per week

```
INSERT INTO Staff (StaffID, StaffName, ShopID, ProRataHours, Total£Sold)
VALUES ('S11', 'Charles', 'HW002', 0.4, 0.00);
```

StaffID	StaffName	ShopID	ProRataHours	Total£Sold
S01	Tilly	HW002	1	2145
S03	Megan	WG001	1	2315
S04	Sabina	WW003	0.6	717.5
S05	Chineye	HW002	0.6	695.5
S06	Neil	WG001	1	2417.5
S08	Sam	WW003	1	1890
S09	Yonas	HW002	0.6	717.5
S10	Siva	WW003	0.6	957.5
S11	Charles	HW002	0.4	0

A view using multiple tables

```
CREATE VIEW Box_bonus
AS
SELECT bd.SalesID,
bd.StaffID,
st.StaffName,
bd.Number_of_boxes,
w.WineName,
s.ShopName
FROM Box_discount AS bd
JOIN Wines AS w ON w.WineID = bd.WineID
JOIN Shops AS s ON s.ShopID = bd.ShopID
JOIN Staff AS st ON bd.StaffID = st.StaffID;
```

This view shows us a lot of useful information about the boxes of wine sold.

SalesID	StaffID	StaffName	Number_of_boxes	WineName	ShopName
1106	S03	Megan	1	Gamay	Wines on the Green
1156	S01	Tilly	2	Pet Nat	Hansen Wines
1177	S06	Neil	1	Champagne	Wines on the Green
1210	S08	Sam	1	Touriga Nacional	Weird Wines
1231	S08	Sam	1	Pink Cava	Weird Wines
1283	S01	Tilly	1	Chalklands	Hansen Wines
1326	S05	Chineye	1	Pinot Noir	Hansen Wines

But in order to pay staff their bonuses you really only need their name and how many boxes they have sold

```
SELECT bb.StaffName, SUM(bb.Number_of_boxes) AS boxes_sold  
FROM Box_bonus AS bb  
GROUP BY bb.StaffName;
```

StaffName	boxes_sold
Megan	1
Tilly	3
Neil	1
Sam	2
Chineye	1

Query with Group By and Having

Which wines have sold less than 300 bottles in total in 2023 across all three shops?

```
SELECT sa.WineID, SUM(sa.Number_of_bottles_sold) AS Total_sold, w.WineName, w.Price
FROM Sales2023 AS sa
JOIN Wines AS w ON sa.WineID = w.WineID
GROUP BY WineID
HAVING SUM(sa.Number_of_bottles_sold) < 300;
```


Which wines have sold less than 300 bottles in total in 2023 across all three shops?

WineID	Total_sold	WineName	Price
sparkling2	231	Canterbury Rose	29
sparkling3	284	Pet Nat	17
sparkling4	112	Champagne	45
sparkling5	238	Cava	16
sparkling6	250	Pink Cava	15

Cheers!

