# **CODE OF THE TABLES CREATION**

```
CREATE TABLE if not exists Users(
 ID_U INT NOT NULL AUTO_INCREMENT,
 user_uuid int not null,
 subscription_date DATE,
 language varchar(40),
 country varchar(40),
 platform varchar(40),
 cooking_date DATE,
 chef_id int not null,
 challenge_id int not null,
 primary key (ID_U)
);
CREATE TABLE if not exists Recipes(
 ID_R INT NOT NULL AUTO_INCREMENT,
 recipe_id int not null,
 level varchar(40),
 category varchar(40),
 stars int not null,
 chef_id int not null,
 challenge_id int not null,
 primary key (ID_R)
);
```

# CODE TO IMPORT THE DATA OF THE DATASET

```
LOAD DATA INFILE 'C:\food_data_clean.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY ""
IGNORE 1 LINES
(@col1, @col2, @dummy1, @col3, @col4, @dummy2, @col5, @col6, @col7, @col8)
SET cooking_date = @col1, user_uuid = @col2, chef_id = @col3,
language = @col4, challenge_id = @col5, country = @col6, subscription_date = @col7, platform = @col8;
LOAD DATA INFILE 'C:\food_data_clean.csv'
INTO TABLE recipes
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY ""
IGNORE 1 LINES
(@dummy1, @dummy2, @col1, @col2, @dummy3, @col3, @col4, @dummy4, @dummy5, @dummy6,
@col5, @col6)
SET category = @col1, chef_id = @col2, level = @col3,
challenge_id = @col4, recipe_id = @col5, stars = @col6;
```

# **CODE OF ALL THE QUERIES**

### Min e max subscription year in the dataset

**SELECT** 

MIN(subscription\_date) AS first\_year,
MAX(subscription\_date) AS last\_year
FROM users;

The years in the dataset are 2020, 2021, and 2022.

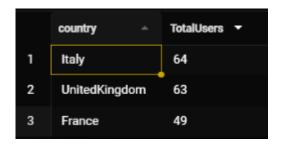
Let's calculate the amount of users who enrolled each year.

SELECT country, COUNT(DISTINCT user\_uuid) AS TotalUsers

FROM users

WHERE EXTRACT(YEAR from subscription\_date) = 2020

GROUP BY country;



SELECT country, COUNT(DISTINCT user\_uuid) AS TotalUsers

FROM users

WHERE EXTRACT(YEAR from subscription\_date) = 2021

**GROUP BY country;** 



SELECT country, COUNT(DISTINCT user\_uuid) AS TotalUsers

FROM users

WHERE EXTRACT(YEAR from subscription\_date) = 2022

GROUP BY country;



There has been a drop of new subscriptions in 2022.

The majority of the users comes from Italy.

## Let's move on the recipes, checking how much they grew during the years.

#### **SELECT**

MIN(cooking\_date) AS first\_year,

MAX(cooking\_date) AS last\_year

FROM users;

The years when the recipes have been posted are 2021, 2022, 2023.

#### **SELECT**

EXTRACT(YEAR FROM u.cooking\_date) AS year,

COUNT(DISTINCT r.recipe\_id) AS TotalRecipes,

COUNT(DISTINCT u.challenge\_id) as total\_challenges,

count(distinct u.user\_uuid) as total\_users

#### **FROM**

users u

#### **INNER JOIN**

recipes r ON u.challenge\_id = r.challenge\_id

#### **GROUP BY**

EXTRACT(YEAR FROM u.cooking\_date)

#### **ORDER BY**

year DESC;



Let's now calculate how many recipes received 4 and 5 stars.

SELECT COUNT(DISTINCT recipe\_id) AS TotalRecipes

**FROM** recipes

WHERE stars >=4;

The recipes which received a rating of 4 and 5 stars are 327, on a total of 593 recipes (according to the unique values calculated with Pandas).

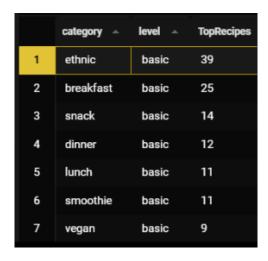
## Let's analyze more in detail which recipes are per level and category.

SELECT category, level, COUNT(DISTINCT recipe\_id) AS TopRecipes

**FROM recipes** 

WHERE stars >= 4 AND level = 'basic'

GROUP BY category, level;

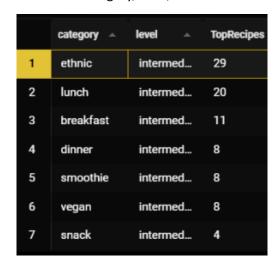


SELECT category, level, COUNT(DISTINCT recipe\_id) AS TopRecipes

FROM recipes

WHERE stars >= 4 AND level = 'intermediate'

GROUP BY category, level;

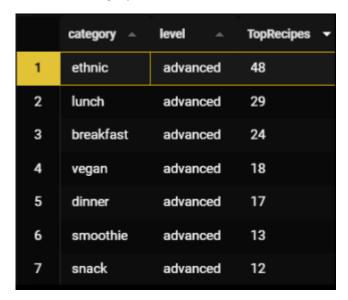


SELECT category, level, COUNT(DISTINCT recipe\_id) AS TopRecipes

FROM recipes

WHERE stars >= 4 AND level = 'advanced'

GROUP BY category, level;

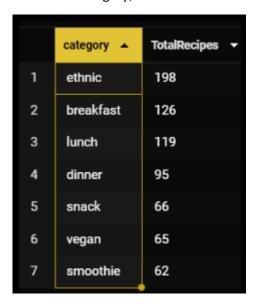


# Quick check of the total recipes published per category.

SELECT category, COUNT(DISTINCT recipe\_id) AS TotalRecipes

**FROM** recipes

GROUP BY category;



The top categories are then ethnic, breakfast, lunch. To find out who are the most popular chefs, I group only the most successful categories by chef.

SELECT category, chef\_id, COUNT(DISTINCT recipe\_id) AS TopRecipes

**FROM** recipes

WHERE stars >= 4 AND category IN ('ethnic', 'breakfast', 'lunch')

GROUP BY category, chef\_id

**ORDER BY TopRecipes DESC** 

LIMIT 10;



Now it's time to analyze the users. Let's sort them by country of origin per each category and level of recipes.

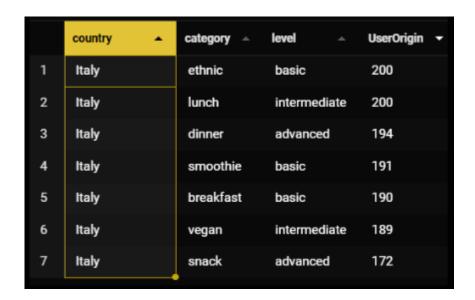
SELECT country, category, level, COUNT(DISTINCT user\_uuid) AS UserOrigin

**FROM** recipes

INNER JOIN users ON recipes.chef\_id = users.chef\_id

WHERE stars >= 4 AND country = 'Italy'

GROUP BY category;

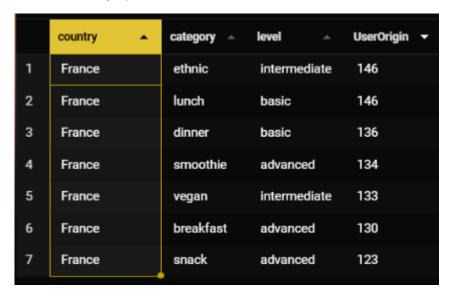


SELECT country, category, level, COUNT(DISTINCT user\_uuid) AS UserOrigin FROM recipes

INNER JOIN users ON recipes.chef\_id = users.chef\_id

WHERE stars >= 4 AND country = 'France'

GROUP BY category;

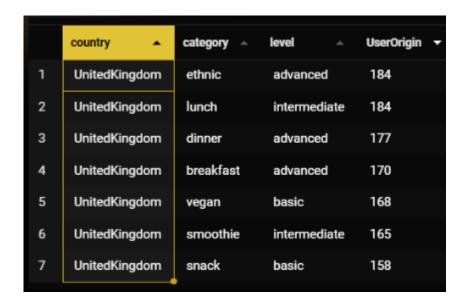


SELECT country, category, level, COUNT(DISTINCT user\_uuid) AS UserOrigin FROM recipes

INNER JOIN users ON recipes.chef\_id = users.chef\_id

WHERE stars >= 4 AND country = 'UnitedKingdom'

GROUP BY category;



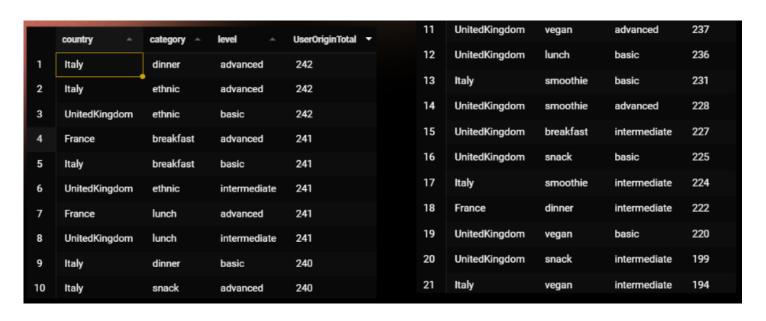
## Let's move on in analyzing the users regardless of the successful recipes.

SELECT country, category, level, COUNT(DISTINCT user\_uuid) AS UserOriginTotal

**FROM** recipes

INNER JOIN users ON recipes.chef\_id = users.chef\_id

GROUP BY category, level;



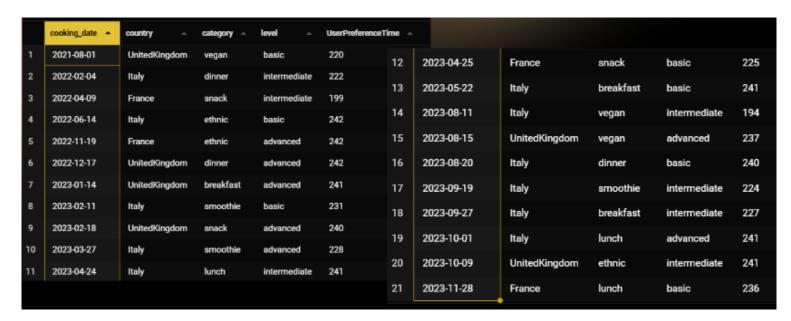
Also from the overall statistics we deduct that the most active users come from Italy, followed by United Kingdom and France. But here the 3 favourite categories are dinner, ethnic, breakfast. Even though in United Kingdom the favourite recipes were the ethnic advanced, without taking account the rating are the ethnic basic.

# For a in-depth study, let's analyze also the preferences of the users during the years of subscription.

SELECT cooking\_date, country, category, level, COUNT(DISTINCT user\_uuid) AS UserPreferenceTime FROM recipes

INNER JOIN users ON recipes.chef\_id = users.chef\_id

GROUP BY category, level;



During the years the same preferences per nationality changed especially in United Kingdom. But the users from Italy and United Kingdom have always been the most active ones.

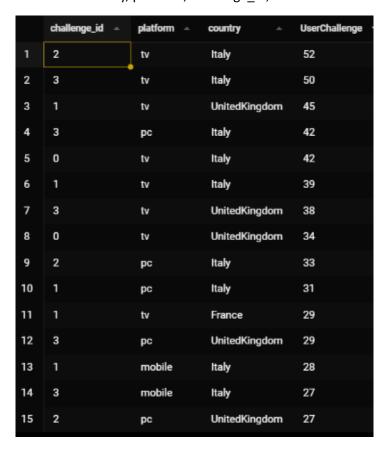
## Eventually, let's analyze the challenges per country of origin and platform used.

SELECT u.challenge id, u.platform, u.country, COUNT(DISTINCT u.user uuid) AS UserChallenge

FROM recipes r

INNER JOIN users u ON r.challenge id = u.challenge id

GROUP BY country, platform, challenge id;



Also here, we deduct that the most active users come from Italy and United Kingdom.

The most chosen platform is the tv.

# **CONCLUSIONS**

The most active users in the interaction with recipes come from Italy and United Kingdom for the recipes with any rating (but are also the subscribers with the highest number).

The preferences per category of recipes changed during the years especially in United Kingdom.

The most published categories of recipes are ethnic and breakfast, which are also the most successful together with the lunch category.

The rating per difficulty of recipe changes according to the country of origin.

The only stable data throughout the whole analysis is the one of the tv as the most chosen channel for the challenges.