

✓ Tests SQL commentés — Base Tifosi

Ce document regroupe les 10 requêtes demandées, avec pour chacune :

- le but de la requête,
- le code SQL exécuté,
- le résultat attendu (théorique),
- le résultat obtenu (preuves exportées),
- un commentaire sur la cohérence ou les éventuels écarts.

❖ Test 01 — Afficher la liste des noms des focaccias par ordre alphabétique croissant

- But : Vérifier que la table `focaccia` est correctement peuplée et que les données peuvent être triées alphabétiquement.
- Code SQL :

```
SELECT nom AS focaccia
FROM focaccia
ORDER BY nom ASC;
```

- Résultat attendu : Liste des 8 focaccias triées alphabétiquement
- Résultat obtenu (preuve) :
 - [Export PDF](#) : tests/exports/test01/test01.pdf
 - [Export CSV](#) : tests/exports/test01/test01.csv
- Commentaire: Le résultat doit contenir exactement 8 lignes, cohérentes avec le seed. Aucun écart n'est constaté.

❖ Test 02 — Afficher le nombre total d'ingrédients

- But : Vérifier le peuplement de la table `ingredient`.
- Code SQL :

```
SELECT COUNT(*) AS nb_ingredients
FROM ingredient;
```

- Résultat attendu : Il doit y avoir 25 ingrédients, comme dans le fichier Excel fourni.
- Résultat obtenu (preuve) :
 - [Export PDF](#) : tests/exports/test02/test02.pdf
 - [Export CSV](#) : tests/exports/test02/test02.csv
- Commentaire: Le résultat est bien égal à 25. Si le résultat avait été différent, cela aurait indiqué qu'un ingrédient a été oublié ou mal orthographié dans le seed.

Test 03 — Afficher le prix moyen des focaccias

- But : Contrôler que les valeurs numériques sont bien interprétées et que l'agrégat AVG fonctionne.
- Code SQL :

```
SELECT ROUND(AVG(prix), 2) AS prix_moyen_focaccia
FROM focaccia;
```

- Résultat attendu : Moyenne calculée sur les 8 prix insérés = 10.375 (arrondi selon MySQL => 2 chiffres après la virgule)
- Résultat obtenu (preuve) :
 - [Export PDF](#) : tests/exports/test03/test03.pdf
 - [Export CSV](#) : tests/exports/test03/test03.csv
- Commentaire: Le résultat obtenu est 10,38 ce qui correspond bien à la moyenne des prix arrondie au centième. Cela confirme que les prix sont correctement stockés et que l'agrégation fonctionne.

Test 04 — Afficher la liste des boissons avec leur marque, triée par nom de boisson

- But : Vérifier la jointure boisson → marque (relation N–1) et l'ordre alphabétique sur le nom de la boisson.
- Code SQL :

```
SELECT b.nom AS boisson, m.nom AS marque
FROM boisson b
JOIN marque m ON m.id = b.marque_id
ORDER BY b.nom ASC;
```

- Résultat attendu : 12 lignes (toutes les boissons), chacune associée à la bonne marque, ordonnées par nom de boisson.
- Résultat obtenu (preuve) :
 - [Export PDF](#) : tests/exports/test04/test04.pdf
 - [Export CSV](#) : tests/exports/test04/test04.csv
- Commentaire: Le résultat obtenu correspond à toutes les boissons, avec la bonne marque pour chacune. Aucune incohérence n'a été détectée.

Test 05 — Afficher la liste des ingrédients pour une Raclaccia

- But : Tester une jointure N–N avec filtrage sur une focaccia particulière.
- Code SQL :

```
SELECT i.nom AS ingredient, fi.quantite
FROM focaccia f
```

```

JOIN focaccia_ingredient fi ON fi.focaccia_id = f.id
JOIN ingredient i ON i.id = fi.ingredient_id
WHERE f.nom = 'Raclaccia'
ORDER BY i.nom ASC;

```

- Résultat attendu : Les ingrédients de la Raclaccia :Base Tomate, Raclette, Cresson, Ail, Champignon, Parmesan, Poivre et leur quantité.
- Résultat obtenu (preuve) :
 - [Export PDF](#) : tests/exports/test05/test05.pdf
 - [Export CSV](#) : tests/exports/test05/test05.csv
- Commentaire: Les 7 ingrédients attendus de la Raclaccia, avec leurs quantités, ont bien été obtenus. Un écart indiquerait que le seed de la Raclaccia a été mal rempli.

Test 06 — Afficher le nom et le nombre d'ingrédients pour chaque focaccia

- But : Vérifier les relations N–N entre focaccia et ingredient, et s'assurer que chaque focaccia a bien un nombre cohérent d'ingrédients.
- Code SQL :

```

SELECT f.nom AS focaccia, COUNT(fi.ingredient_id) AS nb_ingredients
FROM focaccia f
LEFT JOIN focaccia_ingredient fi ON fi.focaccia_id = f.id
GROUP BY f.id, f.nom
ORDER BY nb_ingredients DESC, f.nom ASC;

```

- Résultat attendu : Chaque focaccia doit apparaître avec son nombre d'ingrédients (aucun NULL).
- Résultat obtenu (preuve) :
 - [Export PDF](#) : tests/exports/test06/test06.pdf
 - [Export CSV](#) : tests/exports/test06/test06.csv
- Commentaire: Toutes les focaccias apparaissent avec un nombre d'ingrédients non nul. Une focaccia avec zéro ingrédient aurait révélé un seed incomplet.

Test 07 — Afficher le nom de la focaccia qui a le plus d'ingrédients

- But : Identifier la focaccia la plus garnie en nombre d'ingrédients et vérifier la capacité à combiner GROUP BY + ORDER BY + LIMIT.
- Code SQL :

```

SELECT f.nom AS focaccia, COUNT(fi.ingredient_id) AS nb_ingredients
FROM focaccia f
LEFT JOIN focaccia_ingredient fi ON fi.focaccia_id = f.id
GROUP BY f.id, f.nom
ORDER BY nb_ingredients DESC, f.nom ASC
LIMIT 1;

```

- Résultat attendu : donner la focaccia la plus garnie et son nombre d'ingrédients (paysanne avec 12 ingrédients)
- Résultat obtenu (preuve) :
 - Export PDF : [tests/exports/test07/test07.pdf](#)
 - Export CSV : [tests/exports/test07/test07.csv](#)
 - .txt : [tests/exports/test07/test07.txt](#)
 - screen .png : [tests/exports/test07/test07.png](#)
- Commentaire: À l'écran, phpMyAdmin affiche bien une seule ligne : « Paysanne / 12 ». Cependant, lors de l'export PDF/CSV, phpMyAdmin réexécute la requête sans prendre correctement en compte le LIMIT 1, ce qui produit une liste complète des focaccias triées par nombre d'ingrédients. Le fichier .txt (copie du presse-papiers) et la capture .png confirment que la requête renvoie bien « Paysanne / 12 » comme focaccia la plus garnie.

Test 08 — Afficher la liste des focaccia qui contiennent de l'ail

- But : Tester un filtre sur un ingrédient précis dans la table d'association.
- Code SQL :

```
SELECT DISTINCT f.nom AS focaccia
FROM focaccia f
JOIN focaccia_ingredient fi ON fi.focaccia_id = f.id
JOIN ingredient i ON i.id = fi.ingredient_id
WHERE i.nom = 'Ail'
ORDER BY f.nom ASC;
```

- Résultat attendu : les focaccias suivantes : Mozaccia, Gorgonzollaccia, Raclaccia et Paysanne.
- Résultat obtenu (preuve) :
 - Export PDF : [tests/exports/test08/test08.pdf](#)
 - Export CSV : [tests/exports/test08/test08.csv](#)
- Commentaire: On obtient bien les 4 focaccias contenant de l'ail : Mozaccia, Gorgonzollaccia, Raclaccia et Paysanne. Une focaccia manquante ou en trop indiquerait un problème dans le seed ou une orthographe incohérente de l'ingrédient.

Test 09 — Afficher la liste des ingrédients inutilisés

- But : Vérifier la complétude du seed et repérer les ingrédients présents dans ingredient mais jamais liés à une focaccia.
- Code SQL :

```
SELECT i.nom AS ingredient_non_utilise
FROM ingredient i
LEFT JOIN focaccia_ingredient fi ON fi.ingredient_id = i.id
WHERE fi.ingredient_id IS NULL
ORDER BY i.nom ASC;
```

- Résultat attendu : Les 2 ingrédients non utilisés : Salami et Tomate cerise.
- Résultat obtenu (preuve) :
 - [Export PDF : tests/exports/test09/test09.pdf](#)
 - [Export CSV : tests/exports/test09/test09.csv](#)
- Commentaire: Les ingrédients Salami et Tomate cerise n'apparaissent dans aucune focaccia. Ce test confirme que ces ingrédients sont bien présents dans la table ingredient mais jamais utilisés dans focaccia_ingredient.

❖ Test 10 — Afficher la liste des focaccia qui n'ont pas de champignons

- But : Récupérer les focaccias qui n'utilisent pas l'ingrédient "Champignon".
- Code SQL :

```
SELECT f.nom AS focaccia_sans_champignon
FROM focaccia f
WHERE NOT EXISTS (
    SELECT 1
    FROM focaccia_ingredient fi
    JOIN ingredient i ON i.id = fi.ingredient_id
    WHERE fi.focaccia_id = f.id
        AND i.nom = 'Champignon'
)
ORDER BY f.nom ASC;
```

- Résultat attendu : les focaccias Américaine et Hawaïenne.
- Résultat obtenu (preuve) :
 - [Export PDF : tests/exports/test10/test10.pdf](#)
 - [Export CSV : tests/exports/test10/test10.csv](#)
- Commentaire: Le résultat contient bien les focaccias Américaine et Hawaïenne. Une focaccia manquante ou en trop aurait indiqué un problème dans le seed ou dans la requête. L'utilisation de NOT EXISTS permet d'éviter les pièges possibles de NOT IN en présence de valeurs NULL.