

```
### Database Creation ###
CREATE DATABASE tp_sql;
USE tp_sql;

### Tables Creation ###
CREATE TABLE prestation (
    prestation_id INT NOT NULL AUTO_INCREMENT,
    prestation_type VARCHAR(255) NOT NULL,
    PRIMARY KEY(prestation_id)
);

CREATE TABLE categorie (
    categorie_id INT NOT NULL AUTO_INCREMENT,
    categorie_libelle VARCHAR(255) NOT NULL,
    PRIMARY KEY(categorie_id)
);

CREATE TABLE carte (
    carte_id VARCHAR(10) NOT NULL,
    carte_nom VARCHAR(255) NOT NULL,
    categorie_id INT NOT NULL,
    carte_caution INT NOT NULL,
    carte_date DATE NOT NULL,
    PRIMARY KEY(carte_id),
    FOREIGN KEY(categorie_id) REFERENCES categorie(categorie_id)
);

CREATE TABLE ticket (
    ticket_id VARCHAR(10) NOT NULL,
    carte_id VARCHAR(10) NOT NULL,
    ticket_date DATE NOT NULL,
    PRIMARY KEY(ticket_id),
    FOREIGN KEY(carte_id) REFERENCES carte(carte_id)
);

CREATE TABLE tarif (
    prestation_id INT NOT NULL,
    categorie_id INT NOT NULL,
    tarif_prix DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY(prestation_id, categorie_id),
    FOREIGN KEY(prestation_id) REFERENCES prestation(prestation_id),
    FOREIGN KEY(categorie_id) REFERENCES categorie(categorie_id)
);

CREATE TABLE depot (
    carte_id VARCHAR(10) NOT NULL,
    depot_date DATE NOT NULL,
    depot_montant INT NOT NULL,
    PRIMARY KEY(carte_id, depot_date),
```

```

    FOREIGN KEY(carte_id) REFERENCES carte(carte_id)
);

CREATE TABLE achat (
    ticket_id VARCHAR(10) NOT NULL,
    prestation_id INT NOT NULL,
    achat_nombre INT NOT NULL,
    PRIMARY KEY(ticket_id, prestation_id),
    FOREIGN KEY(ticket_id) REFERENCES ticket(ticket_id),
    FOREIGN KEY(prestation_id) REFERENCES prestation(prestation_id)
);

### Insert Values ###

INSERT INTO prestation (prestation.prestation_type)
VALUES
("Plat principal"),
("Plat + Dessert"),
("Entrée + Plat"),
("Supplément Entrée"),
("Supplément dessert"),
("Repas complet"),
("Alcool"),
("Apéritif"),
("Soda");

INSERT INTO categorie (categorie.categorie_libelle)
VALUES
("Petits revenus"),
("Revenus moyens"),
("Gros revenus"),
("Visiteurs");

INSERT INTO carte (carte.carte_id, carte.carte_nom, carte.categorie_id,
carte.carte_caution, carte.carte_date)
VALUES
("C1","Nina Mcdowell", 3, 6, "2022-08-27"),
("C2","Murphy Lane", 2, 9, "2022-08-15"),
("C3","Hillary Garza", 2, 7, "2022-09-07"),
("C4","Ciaran Ball", 3, 6, "2022-09-11"),
("C5","Kirsten Norton", 3, 7, "2022-09-28"),
("C6","Abdul Weiss", 3, 9, "2022-09-19"),
("C7","Rinah Reilly", 1, 8, "2022-09-11"),
("C8","Kimberly Bryan", 3, 9, "2022-09-17"),
("C9","Galena Keller", 4, 7, "2022-08-16"),
("C10","Francesca Briggs", 2, 9, "2022-08-16"),
("C11","Abdul Weiss", 2, 9, "2022-09-19"),
("C12","Rinah Reilly", 1, 8, "2022-09-11"),
("C13","Kimberly Bryan", 1, 9, "2022-09-17"),

```

```
("C14","Galena Keller", 1, 7, "2022-08-16"),  
("C15","Francesca Briggs", 1, 9, "2022-08-16");
```

```
INSERT INTO ticket (ticket.ticket_id, ticket.carte_id, ticket.ticket_date)  
VALUES
```

```
("T01", "C1", "2022-10-03"),  
("T02", "C1", "2022-10-05"),  
("T03", "C1", "2022-10-10"),  
("T04", "C1", "2022-10-11"),  
("T05", "C1", "2022-11-02"),  
("T06", "C2", "2022-10-11"),  
("T07", "C2", "2022-10-13"),  
("T08", "C2", "2022-11-02"),  
("T09", "C2", "2022-11-10"),  
("T10", "C2", "2022-11-14"),  
("T11", "C3", "2022-10-11"),  
("T12", "C3", "2022-11-20"),  
("T13", "C3", "2022-10-24"),  
("T14", "C4", "2022-11-02"),  
("T15", "C4", "2022-11-14"),  
("T16", "C4", "2022-11-24"),  
("T17", "C4", "2022-12-12"),  
("T18", "C6", "2022-12-27"),  
("T19", "C6", "2022-11-14"),  
("T20", "C7", "2022-12-19"),  
("T21", "C7", "2022-11-22"),  
("T22", "C7", "2022-11-10"),  
("T23", "C7", "2022-11-08"),  
("T24", "C7", "2022-10-12"),  
("T25", "C8", "2022-11-22"),  
("T26", "C8", "2022-10-18"),  
("T27", "C9", "2022-12-19"),  
("T28", "C9", "2022-11-22"),  
("T29", "C10", "2022-10-20"),  
("T30", "C10", "2022-10-17"),  
("T31", "C10", "2022-10-14"),  
("T32", "C12", "2022-12-08"),  
("T33", "C12", "2022-11-23"),  
("T34", "C12", "2022-11-21"),  
("T35", "C12", "2022-10-19"),  
("T36", "C13", "2022-11-23"),  
("T37", "C13", "2022-11-15"),  
("T38", "C14", "2022-10-12"),  
("T39", "C14", "2022-09-21"),  
("T40", "C15", "2022-09-01");
```

```
INSERT INTO tarif (tarif.prestation_id, tarif.categorie_id, tarif.tarif_prix)  
VALUES  
(1, 1, 2.50),
```

```
(1, 2, 2.70),
(1, 3, 3.10),
(1, 4, 5.00),
(2, 1, 3.10),
(2, 2, 3.80),
(2, 3, 4.50),
(2, 4, 7.00),
(3, 1, 2.90),
(3, 2, 3.10),
(3, 3, 3.70),
(3, 4, 6.50),
(4, 1, 0.20),
(4, 2, 0.40),
(4, 3, 0.90),
(4, 4, 1.20),
(5, 1, 0.10),
(5, 2, 0.30),
(5, 3, 0.70),
(5, 4, 1.10),
(6, 1, 3.80),
(6, 2, 4.00),
(6, 3, 4.30),
(6, 4, 9.00),
(7, 1, 1.20),
(7, 2, 1.30),
(7, 3, 1.50),
(7, 4, 2.00),
(8, 1, 1.30),
(8, 2, 1.40),
(8, 3, 1.60),
(8, 4, 2.10),
(9, 1, 0.80),
(9, 2, 0.90),
(9, 3, 1.00),
(9, 4, 2.30);
```

```
INSERT INTO depot (depot.carte_id, depot.depot_date, depot.depot_montant)
VALUES
```

```
("C1", "2022-09-01", 5),
("C1", "2022-10-13", 30),
("C10", "2022-08-18", 5),
("C10", "2022-09-01", 25),
("C10", "2022-09-14", 30),
("C10", "2022-10-14", 25),
("C12", "2022-09-14", 80),
("C12", "2022-09-16", 30),
("C13", "2022-09-23", 40),
("C13", "2022-10-01", 20),
("C14", "2022-08-17", 20),
```

```
("C14", "2022-09-14", 30),
("C15", "2022-08-18", 30),
("C15", "2022-09-23", 30),
("C15", "2022-12-01", 45),
("C2", "2022-09-05", 20),
("C2", "2022-09-07", 30),
("C2", "2022-11-03", 15),
("C3", "2022-09-10", 40),
("C4", "2022-09-12", 20),
("C4", "2022-10-03", 30),
("C6", "2022-09-23", 60),
("C7", "2022-09-14", 5),
("C7", "2022-10-05", 20),
("C7", "2022-11-14", 30),
("C8", "2022-09-22", 20),
("C8", "2022-09-30", 30),
("C9", "2022-08-20", 5),
("C9", "2022-09-02", 20);
```

```
INSERT INTO achat (achat.ticket_id, achat.prestation_id, achat.achat_nombre)
VALUES
```

```
("T01", 1, 1),
("T01", 7, 2),
("T02", 1, 1),
("T02", 4, 2),
("T02", 9, 3),
("T03", 6, 1),
("T03", 8, 1),
("T04", 1, 1),
("T04", 4, 2),
("T04", 7, 1),
("T05", 6, 3),
("T06", 3, 1),
("T07", 2, 3),
("T08", 1, 2),
("T08", 7, 3),
("T09", 1, 1),
("T09", 5, 2),
("T09", 8, 3),
("T10", 1, 1),
("T10", 9, 4),
("T11", 3, 1),
("T11", 7, 2),
("T12", 1, 2),
("T12", 4, 3),
("T13", 9, 4),
("T14", 6, 4),
("T15", 2, 1),
("T15", 7, 3),
```

```
("T15", 8, 2),
("T16", 2, 1),
("T16", 9, 2),
("T17", 6, 2),
("T18", 6, 1),
("T19", 1, 2),
("T19", 9, 4),
("T20", 6, 2),
("T21", 1, 3),
("T21", 8, 3),
("T22", 2, 1),
("T22", 7, 1),
("T22", 8, 1),
("T23", 6, 2),
("T24", 1, 3),
("T25", 3, 2),
("T25", 9, 2),
("T26", 6, 1),
("T27", 2, 1),
("T27", 7, 2),
("T28", 6, 4),
("T29", 3, 3),
("T29", 9, 3),
("T30", 1, 1),
("T30", 7, 1),
("T30", 9, 1),
("T31", 6, 1),
("T32", 2, 1),
("T33", 3, 2),
("T34", 6, 1),
("T34", 7, 1),
("T35", 3, 1),
("T35", 7, 2),
("T36", 6, 1),
("T37", 3, 1),
("T37", 8, 2),
("T38", 2, 1),
("T38", 7, 1),
("T39", 1, 1),
("T39", 5, 2),
("T40", 1, 3),
("T40", 4, 4);
```

## 1- Contenu de la table prestation

```
SELECT * FROM prestation;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

	prestation_id	prestation_type
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	Plat principal
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	Plat + Dessert
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	3	Entrée + Plat
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	4	Supplément Entrée
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	5	Supplément dessert
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	6	Repas complet
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	7	Alcool
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	8	Apéritif
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9	Soda

## 2- Contenu de la table usager avec les catégories en toutes lettres

```
SELECT carte.carte_nom, carte.carte_caution, carte.carte_date, categorie.categorie_libelle FROM carte NATURAL JOIN categorie;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	carte_caution	carte_date	categorie_libelle
Rinah Reilly	8	2022-09-11	Petits revenus
Kimberly Bryan	9	2022-09-17	Petits revenus
Galena Keller	7	2022-08-16	Petits revenus
Francesca Briggs	9	2022-08-16	Petits revenus
Rinah Reilly	8	2022-09-11	Petits revenus
Francesca Briggs	9	2022-08-16	Revenus moyens
Abdul Weiss	9	2022-09-19	Revenus moyens
Murphy Lane	9	2022-08-15	Revenus moyens
Hillary Garza	7	2022-09-07	Revenus moyens
Nina McDowell	6	2022-08-27	Gros revenus
Ciaran Ball	6	2022-09-11	Gros revenus
Kirsten Norton	7	2022-09-28	Gros revenus
Abdul Weiss	9	2022-09-19	Gros revenus
Kimberly Bryan	9	2022-09-17	Gros revenus
Galena Keller	7	2022-08-16	Visiteurs

### 3- Liste des numéros et noms des usagers

```
SELECT carte.carte_id, carte.carte_nom FROM carte;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Cancel](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

					carte_id	carte_nom		
<input type="checkbox"/>		Edit		Copy		Delete	C1	Nina Mcdowell
<input type="checkbox"/>		Edit		Copy		Delete	C10	Francesca Briggs
<input type="checkbox"/>		Edit		Copy		Delete	C11	Abdul Weiss
<input type="checkbox"/>		Edit		Copy		Delete	C12	Rinah Reilly
<input type="checkbox"/>		Edit		Copy		Delete	C13	Kimberly Bryan
<input type="checkbox"/>		Edit		Copy		Delete	C14	Galena Keller
<input type="checkbox"/>		Edit		Copy		Delete	C15	Francesca Briggs
<input type="checkbox"/>		Edit		Copy		Delete	C2	Murphy Lane
<input type="checkbox"/>		Edit		Copy		Delete	C3	Hillary Garza
<input type="checkbox"/>		Edit		Copy		Delete	C4	Ciaran Ball
<input type="checkbox"/>		Edit		Copy		Delete	C5	Kirsten Norton
<input type="checkbox"/>		Edit		Copy		Delete	C6	Abdul Weiss
<input type="checkbox"/>		Edit		Copy		Delete	C7	Rinah Reilly
<input type="checkbox"/>		Edit		Copy		Delete	C8	Kimberly Bryan
<input type="checkbox"/>		Edit		Copy		Delete	C9	Galena Keller

### 4- Moyenne du montant des dépôts (création d'une vue)

✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
1 CREATE VIEW depot_average AS
2 SELECT AVG(depot.depot_montant) AS depot_montant_average
3 FROM depot;
4 |
5 SELECT * FROM depot_average;
```

☒ Enable foreign key checks

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Cancel](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

**depot\_montant\_average**

27.2414



## 5- Montant maximum des dépôts

```
SELECT MAX(depot.depot_montant) AS depot_montant_max FROM depot;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [E](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

depot_montant_max
80

## 6- Montant des dépôts par usagers (création d'une vue)

```
1 CREATE VIEW depot_par_usager AS
2 SELECT DISTINCT carte.carte_nom, SUM(depot.depot_montant)
3 FROM carte
4 NATURAL JOIN depot
5 GROUP BY carte_nom;
6
7 SELECT * FROM depot_par_usager
```

☒ Enable foreign key checks

Go Cancel

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	SUM(depot.depot_montant)
Abdul Weiss	60
Ciaran Ball	50
Francesca Briggs	190
Galena Keller	75
Hillary Garza	40
Kimberly Bryan	110
Murphy Lane	65
Nina McDowell	35
Rinah Reilly	165

## 7- Liste des usagers ayant payé plus de 10€ de caution

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SELECT carte.carte_id, carte.carte_nom, carte.carte_caution FROM carte WHERE carte.carte_caution > 10;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

carte_id	carte_nom	carte_caution
----------	-----------	---------------

Query results operations

## 8- Liste des usagers par catégories

✓ Showing rows 0 - 14 (15 total, Query took 0.0004 seconds.) [categorie\_libelle: **GROS REVENUS... - VISITEURS...**]

```
SELECT carte.carte_nom, categorie.categorie_libelle FROM carte NATURAL JOIN categorie ORDER BY categorie.categorie_libelle;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	categorie_libelle ▲ 1
Nina Mcdowell	Gros revenus
Ciaran Ball	Gros revenus
Kirsten Norton	Gros revenus
Abdul Weiss	Gros revenus
Kimberly Bryan	Gros revenus
Rinah Reilly	Petits revenus
Kimberly Bryan	Petits revenus
Galena Keller	Petits revenus
Francesca Briggs	Petits revenus
Rinah Reilly	Petits revenus
Francesca Briggs	Revenus moyens
Abdul Weiss	Revenus moyens
Murphy Lane	Revenus moyens
Hillary Garza	Revenus moyens
Galena Keller	Visiteurs

### 9- Liste des usagers ayant déjà effectué au moins un dépôt supérieur à 15€

```
SELECT DISTINCT carte.carte_nom FROM carte NATURAL JOIN depot WHERE depot.depot_montant > 15
ORDER BY carte.carte_nom;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

**carte\_nom** ▲ 1

- Abdul Weiss
- Ciaran Ball
- Francesca Briggs
- Galena Keller
- Hillary Garza
- Kimberly Bryan
- Murphy Lane
- Nina Mcdowell
- Rinah Reilly

### 10- Liste des usagers n'ayant jamais effectué de dépôt







```
SELECT DISTINCT carte.carte_nom FROM carte WHERE NOT EXISTS( SELECT depot.carte_id FROM depot WHERE
depot.carte_id = carte.carte_id );
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼

Extra options

← T → **carte\_nom**

<input type="checkbox"/>	 Edit	 Copy	 Delete	Abdul Weiss
<input type="checkbox"/>	 Edit	 Copy	 Delete	Kirsten Norton

## 11- Nombre de dépôt effectué par chaque usager

```
SELECT carte.carte_nom, COUNT(depot.carte_id) as depot_number FROM carte NATURAL JOIN depot GROUP BY carte.carte_nom;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	depot_number
Abdul Weiss	1
Ciaran Ball	2
Francesca Briggs	7
Galena Keller	4
Hillary Garza	1
Kimberly Bryan	4
Murphy Lane	3
Nina Mcdowell	2
Rinah Reilly	5

## 12- Liste des personnes ayant déjà déjà achetés « supplément entrée » (création d'une vue)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0046 seconds.)

```
CREATE VIEW supplemetenet_entree_achetee AS SELECT DISTINCT carte.carte_nom FROM carte NATURAL JOIN achat NATURAL JOIN ticket NATURAL JOIN prestation WHERE prestation.prestation_type = "Supplément Entrée";
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available

✓ Showing rows 0 - 2 (3 total, Query took 0.0020 seconds.)

```
SELECT * FROM supplemetenet_entree_achetee;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼ | Filter rows: Search this table

Extra options

carte_nom
Nina Mcdowell
Francesca Briggs
Hillary Garza

Snipping Tool

### 13- Liste de personnes par prestation

```
SELECT DISTINCT prestation.prestation_type, carte.carte_nom FROM carte NATURAL JOIN ticket NATURAL JOIN achat NATURAL JOIN prestation ORDER BY prestation.prestation_type;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

1 ▾ > >> | ☐ Show all | Number of rows: 25 ▾

Extra options

prestation_type	carte_nom
Alcool	Ciaran Ball
Alcool	Francesca Briggs
Alcool	Rinah Reilly
Alcool	Galena Keller
Alcool	Murphy Lane
Alcool	Hillary Garza
Alcool	Nina Mcdowell
Apéritif	Nina Mcdowell
Apéritif	Ciaran Ball
Apéritif	Rinah Reilly
Apéritif	Kimberly Bryan
Apéritif	Murphy Lane
Entrée + Plat	Francesca Briggs
Entrée + Plat	Rinah Reilly
Entrée + Plat	Kimberly Bryan
Entrée + Plat	Murphy Lane
Entrée + Plat	Hillary Garza
Plat + Dessert	Ciaran Ball
Plat + Dessert	Rinah Reilly
Plat + Dessert	Galena Keller

### 14- Nombre de personnes venues au restaurant et nombre de tickets pour le 22/11/2022

```
SELECT ticket.ticket_date, COUNT(DISTINCT carte.carte_nom) AS nombre_personnes, COUNT(*) AS nombre_tickets FROM ticket NATURAL JOIN carte WHERE ticket.ticket_date = "2022-11-22";
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

Extra options

ticket_date	nombre_personnes	nombre_tickets
2022-11-22	3	3

15- Nom de personnes par jour (considérer que les jours où il y a eu un achat)

```
SELECT ticket.ticket_date, carte.carte_nom FROM ticket NATURAL JOIN carte ORDER BY ticket.ticket_date;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ]

1 > >> | ☐ Show all | Number of rows: 25

Extra options

ticket_date	1	carte_nom
2022-09-01		Francesca Briggs
2022-09-21		Galena Keller
2022-10-03		Nina Mcdowell
2022-10-05		Nina Mcdowell
2022-10-10		Nina Mcdowell
2022-10-11		Nina Mcdowell
2022-10-11		Murphy Lane
2022-10-11		Hillary Garza
2022-10-12		Rinah Reilly
2022-10-12		Galena Keller
2022-10-13		Murphy Lane
2022-10-14		Francesca Briggs
2022-10-17		Francesca Briggs
2022-10-18		Kimberly Bryan
2022-10-19		Rinah Reilly
2022-10-20		Francesca Briggs
2022-10-24		Hillary Garza

16- Nombre de personnes ayant déjà acheté un apéritif

```
SELECT COUNT(*) AS nombre_personnes, prestation.prestation_type FROM achat NATURAL JOIN ticket NATURAL JOIN carte NATURAL JOIN prestation WHERE prestation.prestation_type = "Apéritif";
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Run ]

Extra options

nombre_personnes	prestation_type
6	Apéritif

## 17- Liste des achats (date, nombre) effectués par « Nina Mcdowell »

```
SELECT ticket.ticket_date, achat.achat_nombre, carte.carte_nom FROM achat NATURAL JOIN ticket NATURAL JOIN carte WHERE carte.carte_nom = "Nina Mcdowell";
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows:


Extra options


ticket_date	achat_nombre	carte_nom
2022-10-03	1	Nina Mcdowell
2022-10-03	2	Nina Mcdowell
2022-10-05	1	Nina Mcdowell
2022-10-05	2	Nina Mcdowell
2022-10-05	3	Nina Mcdowell
2022-10-10	1	Nina Mcdowell
2022-10-10	1	Nina Mcdowell
2022-10-11	1	Nina Mcdowell
2022-10-11	2	Nina Mcdowell
2022-10-11	1	Nina Mcdowell
2022-11-02	3	Nina Mcdowell

## 18- Recette du supplément entrée (création d'une vue)

```
CREATE VIEW recette_supplement_entree AS SELECT SUM(achat.achat_nombre * tarif.tarif_prix) AS recette FROM achat NATURAL JOIN ticket NATURAL JOIN carte NATURAL JOIN prestation NATURAL JOIN tarif WHERE prestation.prestation_type = "Supplément Entrée";
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

 Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

 Showing rows 0 - 0 (1 total, Query took 0.0034 seconds.)

```
SELECT * FROM recette_supplement_entree;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows:  Filter rows:

Extra options

**recette**

5.60

## 19- Recette de la catégorie « petits revenus »

```
SELECT SUM(achat.achat_nombre * tarif.tarif_prix) AS recette FROM achat NATURAL JOIN ticket NATURAL JOIN carte NATURAL JOIN tarif NATURAL JOIN categorie WHERE categorie.categorie_libelle = "Petits Revenus";
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 ▼

Extra options

recette
83.50

## 20- Recette des achats effectués le 22/11/2022

```
SELECT SUM(achat.achat_nombre * tarif.tarif_prix) AS recette FROM achat NATURAL JOIN ticket NATURAL JOIN carte NATURAL JOIN tarif WHERE ticket.ticket_date = "2022-11-22";
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 ▼

Extra options

recette
56.80

## 21- Quels usagers ont acheté au moins 2 fois la prestation « repas complet » (2 fois prestation is so not clear enough)

```
SELECT carte.carte_nom, COUNT(*) AS nombre_fois FROM achat NATURAL JOIN carte NATURAL JOIN prestation WHERE prestation.prestation_type = "Repas Complet" GROUP BY carte.carte_nom HAVING COUNT(*) >= 2;
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	nombre_fois
Abdul Weiss	24
Ciaran Ball	12
Francesca Briggs	24
Galena Keller	24
Hillary Garza	12
Kimberly Bryan	24
Kirsten Norton	12
Murphy Lane	12
Nina McDowell	12
Rinah Reilly	24



22- A quelle date « Nina Mcdowell » est venu au restaurant la dernière fois

```
SELECT carte.carte_nom, MAX(ticket.ticket_date) AS last_visit FROM ticket NATURAL JOIN carte WHERE
carte.carte_nom = "Nina Mcdowell" ORDER BY ticket.ticket_date;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	last_visit
Nina Mcdowell	2022-11-02

23- Qui sont la ou les utilisateurs qui sont venus pour la dernière fois au restaurant

```
SELECT carte.carte_nom, ticket.ticket_date FROM ticket NATURAL JOIN carte WHERE ticket.ticket_date
= (SELECT MAX(ticket.ticket_date) FROM ticket);
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

carte_nom	ticket_date
Abdul Weiss	2022-12-27

24- Quel pourcentage des usagers ont acheté au moins une fois la prestation « plat principal »

```
SELECT ( SELECT COUNT(DISTINCT carte.carte_nom) FROM achat NATURAL JOIN carte NATURAL JOIN ticket
NATURAL JOIN prestation WHERE prestation.prestation_type = "Plat Principal" ) / ( SELECT
COUNT(DISTINCT carte.carte_nom) FROM carte ) * 100 AS perentage;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Extra options

perentage
70.0000

25- Liste des prestations jamais achetées en septembre 2019

```
SELECT DISTINCT prestation.prestation_type FROM prestation WHERE prestation.prestation_type NOT IN ( SELECT DISTINCT prestation.prestation_type FROM achat NATURAL JOIN ticket NATURAL JOIN prestation WHERE MONTH(ticket.ticket_date) = 09 AND YEAR(ticket.ticket_date) = 2019);
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

	prestation_type
<input type="checkbox"/> Edit Copy Delete	Plat principal
<input type="checkbox"/> Edit Copy Delete	Plat + Dessert
<input type="checkbox"/> Edit Copy Delete	Entrée + Plat
<input type="checkbox"/> Edit Copy Delete	Supplément Entrée
<input type="checkbox"/> Edit Copy Delete	Supplément dessert
<input type="checkbox"/> Edit Copy Delete	Repas complet
<input type="checkbox"/> Edit Copy Delete	Alcool
<input type="checkbox"/> Edit Copy Delete	Apéritif
<input type="checkbox"/> Edit Copy Delete	Soda

## 26- Liste des prestations jamais achetés par « Nina Mcdowell »

```
SELECT DISTINCT prestation.prestation_type FROM prestation WHERE prestation.prestation_type NOT IN ( SELECT DISTINCT prestation.prestation_type FROM achat NATURAL JOIN ticket NATURAL JOIN prestation NATURAL JOIN carte WHERE carte.carte_nom = "Nina Mcdowell" );
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

	prestation_type
<input type="checkbox"/> Edit Copy Delete	Plat + Dessert
<input type="checkbox"/> Edit Copy Delete	Entrée + Plat
<input type="checkbox"/> Edit Copy Delete	Supplément dessert

## 27- Liste des prestations achetés par « Abdul Weiss »

```
SELECT DISTINCT prestation.prestation_type FROM achat NATURAL JOIN ticket NATURAL JOIN prestation NATURAL JOIN carte WHERE carte.carte_nom = "Abdul Weiss";
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

**prestation\_type**

Repas complet

Plat principal

Soda

## 28- Recette des boissons (soda, apéritif et alcool)

```
SELECT SUM(achat.achat_nombre * tarif.tarif_prix) AS recette FROM achat NATURAL JOIN ticket NATURAL JOIN prestation NATURAL JOIN tarif NATURAL JOIN carte WHERE prestation.prestation_type = "Soda" OR prestation.prestation_type = "Apéritif" OR prestation.prestation_type = "alcool";
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

Extra options

**recette**

65.40

29- Pourcentage des gens ayant déjà acheté au moins une fois la prestation « alcool » + « apéritif »

```
SELECT (SELECT COUNT(*) FROM ( SELECT carte.carte_nom FROM achat NATURAL JOIN ticket NATURAL JOIN prestation NATURAL JOIN carte WHERE prestation.prestation_type = "Apéritif" OR prestation.prestation_type = "Alcool" GROUP BY carte.carte_nom HAVING COUNT(DISTINCT prestation.prestation_type) = 2 ) AS apéritif_plus_alcool ) * 100.0 / (SELECT COUNT(DISTINCT carte.carte_nom) FROM carte) AS percentage;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Extra options

**percentage**

40.00000

30- Faire la requête permettant d'insérer un nouvel utilisateur dans la base

```
CREATE USER aqua;
```

31- Faire la requête permettant de modifier le nom d'un utilisateur dans la base

✓ MySQL returned an empty result set

```
RENAME USER aqua TO useless;
```

32- Faire la requête permettant la suppression de l'utilisateur qu'on avait ajouté

```
DROP USER useless;
```