

Rapport TP SQL

Création de la base de données

```
-- Création de la base de données
CREATE DATABASE transport_logistique;
```

Réponse :

```
✓ 139 12:51:33 CREATE DATABASE transport_logistique                                1 row(s) affected                                0.016 sec
```

```
-- Création de la table entrepots
CREATE TABLE entrepots (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nom_entrepot VARCHAR(140),
    adresse VARCHAR(200),
    ville VARCHAR(120),
    pays VARCHAR(120)
);
```

Réponse :

```
✓ 141 12:52:27 CREATE TABLE entrepots ( id INT PRIMARY KEY AUTO_INCREMENT, nom_entrepot VARCHAR(140), ad... 0 row(s) affected                                0.046 sec
```

```
-- Création de la table expéditions avec attribution des clés
étrangères
CREATE TABLE expeditions (
    id INT PRIMARY KEY AUTO_INCREMENT,
    date_expedition DATE,
    id_entrepot_source INT,
    id_entrepot_destination INT,
    poids DECIMAL,
    statut VARCHAR(40),
    FOREIGN KEY (id_entrepot_source) REFERENCES entrepots(id),
    FOREIGN KEY (id_entrepot_destination) REFERENCES entrepots(id)
);
```

Réponse :

```
✓ 142 12:57:16 CREATE TABLE expeditions ( id INT PRIMARY KEY AUTO_INCREMENT, date_expedition DATE, id_entre... 0 row(s) affected                                0.156 sec
```

Ajout de données

```
-- Insertion jeu de données entrepots
INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES
('Entrepot1', 'Adresse entrepot1', 'Lyon', 'France');
```

```

INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES
('Entrepot2', 'Adresse entrepot2', 'Pekin', 'Chine');

INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES
('Entrepot3', 'Adresse entrepot3', 'Barcelone', 'Espagne');

INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES
('Entrepot4', 'Adresse entrepot4', 'Lisbonne', 'Portugal');

INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES
('Entrepot5', 'Adresse entrepot5', 'Rome', 'Italie');

INSERT INTO entrepots(nom_entrepot, adresse, ville, pays)
VALUES('Entrepot6', 'Adresse entrepot6', 'Paris', 'France');

-- Insertion jeu de données entrepots

```

Réponse :

143	13:01:17	INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES ('Entrepot1', 'Adresse entrepot1', 'Lyon', 'France')	1 row(s) affected	0.000 sec
144	13:01:17	INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES ('Entrepot2', 'Adresse entrepot2', 'Paris', 'France')	1 row(s) affected	0.016 sec
145	13:01:17	INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES ('Entrepot3', 'Adresse entrepot3', 'Barcelone', 'Espagne')	1 row(s) affected	0.000 sec
146	13:01:17	INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES ('Entrepot4', 'Adresse entrepot4', 'Lisbonne', 'Portugal')	1 row(s) affected	0.015 sec
147	13:01:17	INSERT INTO entrepots(nom_entrepot, adresse, ville, pays) VALUES ('Entrepot5', 'Adresse entrepot5', 'Rome', 'Italie')	1 row(s) affected	0.000 sec

```

-- Insertion jeu de données expéditions

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-11-02', 1, 1, 64,
'En transit');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-10-16', 2, 2,
102, 'Livré');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-12-21', 3, 3,
118, 'En transit');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-09-12', 4, 4, 32,
'Livré');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-11-01', 5, 5, 74,
'En transit');

```

```

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-10-13', 1, 1, 91,
'Livré');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2024-01-10', 2, 2, 13,
'En transit');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2022-02-18', 3, 3,
273, 'Livré');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-10-08', 4, 4,
548, 'En transit');

INSERT INTO expeditions(date_expedition, id_entrepot_source,
id_entrepot_destination, poids, statut) VALUES ('2023-09-22', 5, 5,
195, 'Livré');

-- Insertion jeu de données expéditions

```

Réponse :

#	Time	Action	Message	Duration / Fetch
5	15:56:22	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUES...	1 row(s) affected	0.000 sec
6	15:56:22	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUES...	1 row(s) affected	0.015 sec
7	15:56:22	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUES...	1 row(s) affected	0.000 sec
8	15:56:22	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUES...	1 row(s) affected	0.016 sec
9	15:56:22	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUES...	1 row(s) affected	0.000 sec
10	15:56:23	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUES...	1 row(s) affected	0.016 sec

Requêtes de base :

```

-- Affichez tous les entrepôts.

SELECT * FROM entrepots;

```

Réponse :

1 • `SELECT * FROM entrepots`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	nom_entrepot	adresse	ville	pays
▶	1	Entrepot1	Adresse entrepot1	Lyon	France
2	Entrepot2	Adresse entrepot2	Paris	France	
3	Entrepot3	Adresse entrepot3	Barcelone	Espagne	
4	Entrepot4	Adresse entrepot4	Lisbonne	Portugal	
5	Entrepot5	Adresse entrepot5	Rome	Italie	
•	NULL	NULL	NULL	NULL	NULL

entreports 1 x | Apply | Revert

Output:

Action Output | # | Time | Action | Message

1 16:06:55 SELECT * FROM entrepots LIMIT 0, 1000 5 row(s) returned

-- Affichez toutes les expéditions

`SELECT * FROM expeditions;`

Réponse :

1 • `SELECT * FROM expeditions`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	date_expedition	id_entrepot_source	id_entrepot_destination	poids	statut
▶	1	2023-11-02	1	1	64	En transit
2	2023-10-16	2	2	102	Livré	
3	2023-12-21	3	3	118	En transit	
4	2023-09-12	4	4	32	Livré	
5	2023-11-01	5	5	74	En transit	
6	2023-10-13	1	1	91	Livré	
7	2024-01-10	2	2	13	En transit	
8	2022-02-18	3	3	273	Livré	
9	2023-10-08	4	4	548	En transit	
10	2023-09-22	5	5	195	Livré	
•	NULL	NULL	NULL	NULL	NULL	

expeditions 2 x | Apply | Revert

Output:

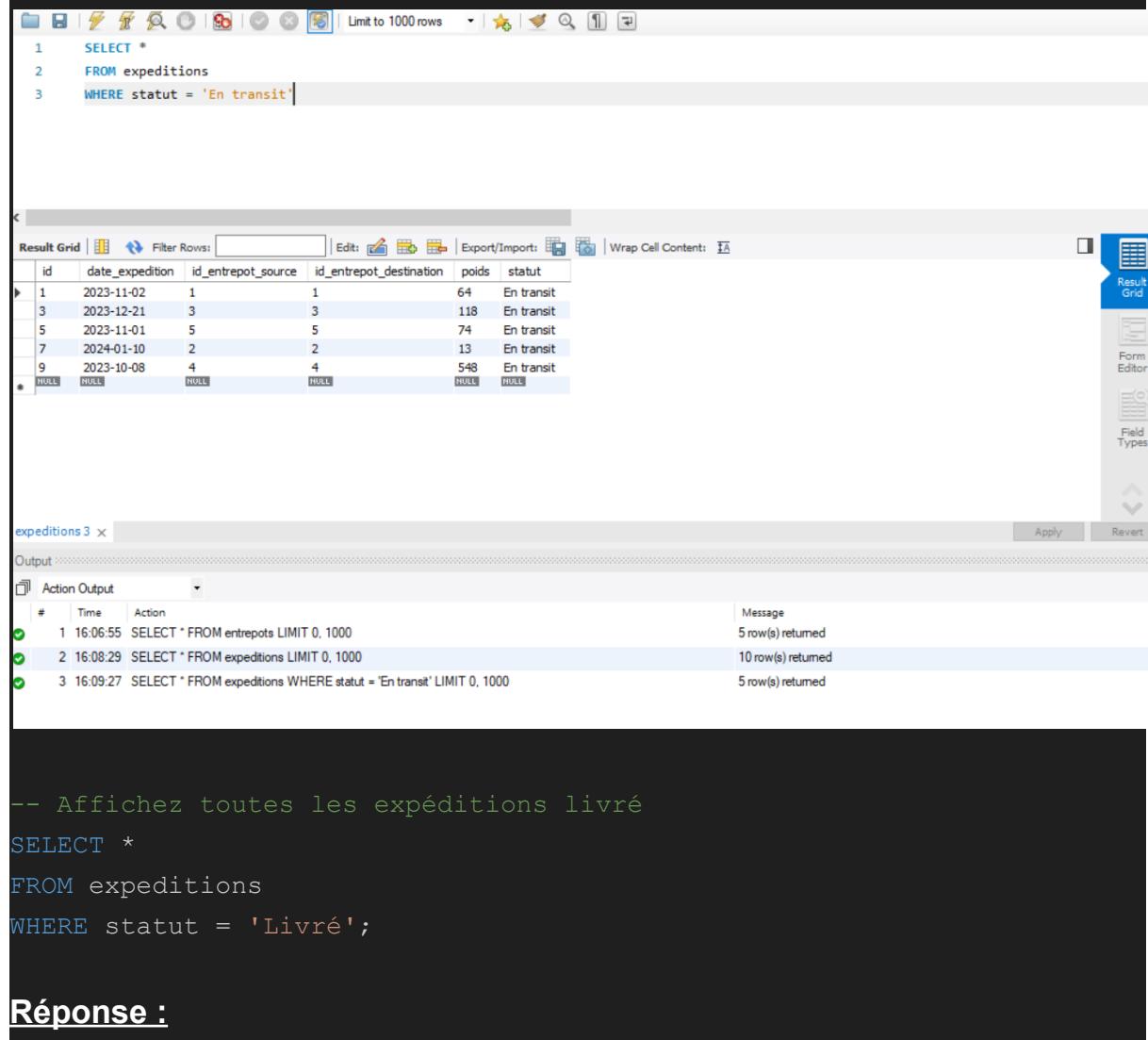
Action Output | # | Time | Action | Message

1 16:06:55 SELECT * FROM entrepots LIMIT 0, 1000 5 row(s) returned

2 16:08:29 SELECT * FROM expeditions LIMIT 0, 1000 10 row(s) returned

```
-- Affichez toutes les expéditions en transit
SELECT *
FROM expeditions
WHERE statut = 'En transit';
```

Réponse :



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** The query is displayed as:


```
1 SELECT *
2 FROM expeditions
3 WHERE statut = 'En transit';
```
- Result Grid:** The results of the query are shown in a table with the following data:

	id	date_expedition	id_entrepot_source	id_entrepot_destination	poids	statut
▶	1	2023-11-02	1	1	64	En transit
3	2023-12-21	3	3	118	En transit	
5	2023-11-01	5	5	74	En transit	
7	2024-01-10	2	2	13	En transit	
9	2023-10-08	4	4	548	En transit	
•	HULL	HULL	HULL	HULL	HULL	HULL
- Output:** The output section shows the following log entries:

#	Time	Action	Message
1	16:06:55	SELECT * FROM entrepots LIMIT 0, 1000	5 row(s) returned
2	16:08:29	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
3	16:09:27	SELECT * FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	5 row(s) returned

The screenshot shows the MySQL Workbench interface. At the top, a query is entered in the SQL editor:

```

1  SELECT *
2  FROM expeditions
3  WHERE statut = 'Livré'

```

The results are displayed in a grid:

	id	date_expedition	id_entrepot_source	id_entrepot_destination	poids	statut
▶	2	2023-10-16	2	2	102	Livré
▶	4	2023-09-12	4	4	32	Livré
▶	6	2023-10-13	1	1	91	Livré
▶	8	2022-02-18	3	3	273	Livré
▶	10	2023-09-22	5	5	195	Livré
*	NULL	NULL	NULL	NULL	NULL	NULL

Below the results, the 'Output' tab is selected, showing the execution history:

#	Time	Action	Message
1	16:06:55	SELECT * FROM entrepots LIMIT 0, 1000	5 row(s) returned
2	16:08:29	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
3	16:09:27	SELECT * FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	5 row(s) returned
4	16:10:14	SELECT * FROM expeditions WHERE statut = 'Livré' LIMIT 0, 1000	5 row(s) returned

Requêtes avancées :

```

--Affichez les entrepôts qui ont envoyé au moins une expédition en
transit
SELECT e.id, e.nom_entrepot
FROM entrepots e
WHERE EXISTS (
    SELECT 1
    FROM expeditions ex
    WHERE ex.id_entrepot_source = e.id
    AND ex.statut = 'En transit'
);

```

Réponse :

```

1 •  SELECT e.id, e.nom_entrepot
2   FROM entrepots e
3   WHERE EXISTS (
4       SELECT 1
5         FROM expeditions ex
6        WHERE ex.id_entrepot_source = e.id
7        AND ex.statut = 'en transit'
8   );

```

id	nom_entrepot
1	Entrepot1
2	Entrepot2
3	Entrepot3
4	Entrepot4
5	Entrepot5
*	HULL

entreports 6 x

Output

#	Time	Action	Message
1	20:29:11	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
2	20:29:56	SELECT COUNT(statut) FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	1 row(s) returned
3	20:30:53	SELECT COUNT(statut) FROM expeditions WHERE statut = 'Livré' LIMIT 0, 1000	1 row(s) returned
4	20:35:26	SELECT e.id AS id_entrepot, e.nom_entrepot, COUNT(ex.id) AS nombre_expeditions_employees FROM entrepots e JOIN expeditions ex ON e.id = ex.id_entrepot_source WHERE ex.statut = 'En transit' GROUP BY e.id, e.nom_entrepot	5 row(s) returned
5	20:37:42	SELECT DISTINCT e.id, e.nom_entrepot FROM entrepots e JOIN expeditions ex ON e.id = ex.id_entrepot_source WHERE ex.statut = 'En transit'	5 row(s) returned
6	20:39:04	SELECT e.id, e.nom_entrepot FROM entrepots e WHERE EXISTS (SELECT 1 FROM expeditions ex WHERE ex.id_entrepot_source = e.id AND ex.statut = 'En transit')	5 row(s) returned

```

--Affichez les entrepôts qui ont reçu au moins une expédition en
transit
SELECT e.id, e.nom_entrepot
FROM entrepots e
WHERE EXISTS (
    SELECT 1
    FROM expeditions ex
    WHERE ex.id_entrepot_destination = e.id
    AND ex.statut = 'En transit'
);

```

Réponse :

```

1 •  SELECT e.id, e.nom_entrepot
2   FROM entrepots e
3   WHERE EXISTS (
4     SELECT 1
5       FROM expeditions ex
6      WHERE ex.id_entrepot_destination = e.id
7      AND ex.statut = 'en transit'
8   )
  
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	nom_entrepot
1	1	Entrepot1
3	3	Entrepot3
4	4	Entrepot4
*	HULL	HULL

entreports 7 x

Output

Action Output

#	Time	Action	Message
2	20:29:56	SELECT COUNT(statut) FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	1 row(s) returned
3	20:30:53	SELECT COUNT(statut) FROM expeditions WHERE statut = 'Livré' LIMIT 0, 1000	1 row(s) returned
4	20:35:26	SELECT e.id AS id_entrepot, e.nom_entrepot, COUNT(ex.id) AS nombre_expeditions_employees FROM entrepots e JOIN expeditions ex ON e.id = ex.id_entrepot_sou... 5 row(s) returned	5 row(s) returned
5	20:37:42	SELECT DISTINCT e.id, e.nom_entrepot FROM entrepots e JOIN expeditions ex ON e.id = ex.id_entrepot_sou... 5 row(s) returned	5 row(s) returned
6	20:39:04	SELECT e.id, e.nom_entrepot FROM entrepots e WHERE EXISTS (SELECT 1 FROM expeditions ex ... 5 row(s) returned	5 row(s) returned
7	20:41:54	SELECT e.id, e.nom_entrepot FROM entrepots e WHERE EXISTS (SELECT 1 FROM expeditions ex ... 3 row(s) returned	3 row(s) returned

```

-- Affichez les expéditions qui ont un poids supérieur à 100 kg et qui
sont en transit.

SELECT *
FROM expeditions
WHERE poids > 100 AND statut = 'En transit';
  
```

Réponse :

```

1  SELECT *
2  FROM expeditions
3  WHERE poids > 100 AND statut = 'En transit'

```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar is a query editor window containing the following SQL code:

```

1  SELECT *
2  FROM expeditions
3  WHERE poids > 100 AND statut = 'En transit'

```

Below the query editor is a 'Result Grid' window displaying the results of the query. The results are as follows:

	id	date_expedition	id_entrepot_source	id_entrepot_destination	poids	statut
▶	3	2023-12-21	3	1	118	En transit
▶	9	2023-10-08	4	1	548	En transit
*	HULL	HULL	HULL	HULL	HULL	HULL

On the right side of the interface, there is a sidebar with three tabs: 'Result Grid' (which is selected), 'Form Editor', and 'Field Types'.

At the bottom of the interface, there is a 'Output' window showing the action history:

#	Time	Action	Message
23	16:23:09	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUE...	1 row(s) affected
24	16:23:09	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUE...	1 row(s) affected
25	16:23:09	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUE...	1 row(s) affected
26	16:23:09	INSERT INTO expeditions(date_expedition, id_entrepot_source, id_entrepot_destination, poids, statut) VALUE...	1 row(s) affected
27	16:23:24	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
28	16:24:52	SELECT * FROM expeditions WHERE poids > 100 AND statut = 'En transit' LIMIT 0, 1000	2 row(s) returned

```

-- Affichez le nombre d'expéditions envoyées par chaque entrepôt.
SELECT e.id AS id_entrepot, e.nom_entrepot, COUNT(ex.id) AS
nombre_expeditions_envoyees
FROM entrepots e
LEFT JOIN expeditions ex ON e.id = ex.id_entrepot_source
GROUP BY e.id, e.nom_entrepot;

```

Réponse :

MySQL Workbench interface showing a query and its results.

```

1 •  SELECT e.id AS id_entrepot, e.nom_entrepot, COUNT(ex.id) AS nombre_expeditions_envoyees
2   FROM entrepots e
3   LEFT JOIN expeditions ex ON e.id = ex.id_entrepot_source
4   GROUP BY e.id, e.nom_entrepot;

```

Result Grid

id_entrepot	nom_entrepot	nombre_expeditions_envoyees
1	Entrepot1	2
2	Entrepot2	2
3	Entrepot3	2
4	Entrepot4	2
5	Entrepot5	2

Output

#	Time	Action	Message
1	20:29:11	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
2	20:29:56	SELECT COUNT(statut) FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	1 row(s) returned
3	20:30:53	SELECT COUNT(statut) FROM expeditions WHERE statut = 'Livre' LIMIT 0, 1000	1 row(s) returned
4	20:35:26	SELECT e.id AS id_entrepot, e.nom_entrepot, COUNT(ex.id) AS nombre_expeditions_envoyees FROM entrepot...	5 row(s) returned

```
-- Affichez le nombre total d'expéditions en transit.
SELECT COUNT(statut)
FROM expeditions
WHERE statut = 'En transit';
```

Réponse :

MySQL Workbench interface showing a query and its results.

```

1 •  SELECT COUNT(statut)
2   FROM expeditions
3   WHERE statut = 'En transit';
4

```

Result Grid

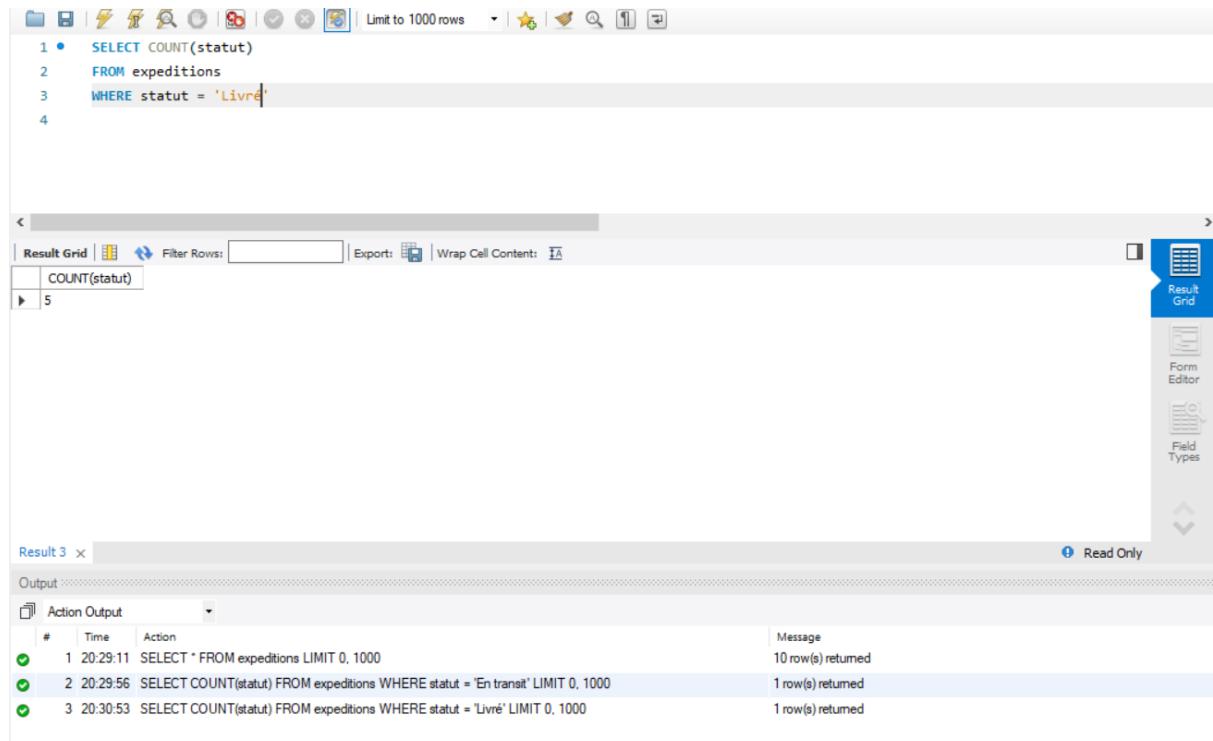
COUNT(statut)
5

Output

#	Time	Action	Message
1	20:29:11	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
2	20:29:56	SELECT COUNT(statut) FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	1 row(s) returned

```
-- Affichez le nombre total d'expéditions livrées.  
SELECT COUNT(statut)  
FROM expeditions  
WHERE statut = 'Livré';
```

Réponse :



The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the query editor window displays the following SQL code:

```
1 • 1. SELECT COUNT(statut)  
2. FROM expeditions  
3. WHERE statut = 'Livré'  
4.
```

Below the query editor is the results grid. The results are as follows:

COUNT(statut)
5

On the right side of the interface, there is a sidebar with three tabs: "Result Grid" (which is selected), "Form Editor", and "Field Types".

At the bottom, the "Result 3" tab is active, showing the following log entries:

Action	Time	Message
1. SELECT * FROM expeditions LIMIT 0, 1000	20:29:11	10 row(s) returned
2. SELECT COUNT(statut) FROM expeditions WHERE statut = 'En transit' LIMIT 0, 1000	20:29:56	1 row(s) returned
3. SELECT COUNT(statut) FROM expeditions WHERE statut = 'Livré' LIMIT 0, 1000	20:30:53	1 row(s) returned

```
-- Affichez le nombre total d'expéditions pour chaque mois de l'année  
en cours  
SELECT MONTH(date_expedition) AS mois, COUNT(*) AS nombre_expéditions  
FROM expeditions  
WHERE YEAR(date_expedition) = YEAR(CURDATE())  
GROUP BY mois  
ORDER BY mois;
```

Réponse :

Limit to 1000 rows

```

1 •  SELECT MONTH(date_expedition) AS mois, COUNT(*) AS nombre_expéditions
2   FROM expeditions
3   WHERE YEAR(date_expedition) = YEAR(CURDATE())
4   GROUP BY mois
5   ORDER BY mois;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only

mois	nombre_expéditions
9	2
10	3
11	2
12	1

Action Output

#	Time	Action	Message
3	20:30:53	SELECT COUNT(statut) FROM expeditions WHERE statut = 'Livré' LIMIT 0, 1000	1 row(s) returned
4	20:35:26	SELECT e.id AS id_entrepot, e.nom_entrepot, COUNT(ex.id) AS nombre_expeditions_employees FROM entrepots e JOIN expeditions ex ON e.id = ex.id_entrepot_source WHERE ex.statut = 'Livré' LIMIT 0, 1000	5 row(s) returned
5	20:37:42	SELECT DISTINCT e.id, e.nom_entrepot FROM entrepots e JOIN expeditions ex ON e.id = ex.id_entrepot_source WHERE ex.statut = 'Livré' LIMIT 0, 1000	5 row(s) returned
6	20:39:04	SELECT e.id, e.nom_entrepot FROM entrepots e WHERE EXISTS (SELECT 1 FROM expeditions ex WHERE ex.id_entrepot = e.id AND ex.statut = 'Livré' LIMIT 0, 1)	5 row(s) returned
7	20:41:54	SELECT e.id, e.nom_entrepot FROM entrepots e WHERE EXISTS (SELECT 1 FROM expeditions ex WHERE ex.id_entrepot = e.id AND ex.statut = 'Livré' LIMIT 0, 1)	3 row(s) returned
8	20:44:01	SELECT MONTH(date_expedition) AS mois, COUNT(*) AS nombre_expéditions FROM expeditions WHERE YEAR(date_expedition) = YEAR(CURDATE())	4 row(s) returned

```
-- Affichez les entrepôts qui ont envoyé des expéditions au cours des
30 derniers jours.

SELECT e.nom_entrepot, exp.date_expedition
FROM entrepots e
INNER JOIN expeditions exp
ON e.id = exp.id_entrepot_source
WHERE exp.date_expedition BETWEEN DATE_SUB(CURDATE(), INTERVAL 30 DAY)
AND CURDATE();
```

Réponse :

```

 1 •  SELECT e.nom_entrepot, exp.date_expedition
 2   FROM entrepots e
 3   INNER JOIN expeditions exp
 4   ON e.id = exp.id_entrepot_source
 5   WHERE exp.date_expedition BETWEEN DATE_SUB(CURDATE(), INTERVAL 30 DAY) AND CURDATE();

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

nom_entrepot	date_expedition
Entrepot1	2023-11-02
Entrepot2	2023-10-16
Entrepot5	2023-11-01
Entrepot1	2023-10-13
Entrepot4	2023-10-08

Result 11 x Read Only

Action Output

#	Time	Action	Message
7	20:41:54	SELECT e.id, e.nom_entrepot FROM entrepots e WHERE EXISTS (SELECT 1 FROM expeditions ex ...	3 row(s) returned
8	20:44:01	SELECT MONTH(date_expedition) AS mois, COUNT(*) AS nombre_expéditions FROM expeditions WHERE Y... 4 row(s) returned	
9	20:45:50	SELECT DISTINCT e.nom_entrepot FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.id_entrepot ... 5 row(s) returned	
10	20:46:37	SELECT DISTINCT e.nom_entrepot, e.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.i... Error Code: 1054. Unknown column 'e.date_expedition' in field list'	
11	20:46:53	SELECT e.nom_entrepot, exp.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.i... 7 row(s) returned	
12	20:48:43	SELECT e.nom_entrepot, exp.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.i... 5 row(s) returned	

```

-- Affichez les entrepôts qui ont reçu des expéditions au cours des 30 derniers jours.

SELECT e.nom_entrepot, exp.date_expedition
FROM entrepots e
INNER JOIN expeditions exp
ON e.id = exp.id_entrepot_destination
WHERE exp.date_expedition BETWEEN DATE_SUB(CURDATE(), INTERVAL 30 DAY)
AND CURDATE();

```

Réponse :

```

1 •  SELECT e.nom_entrepot, exp.date_expedition
2   FROM entrepots e
3   INNER JOIN expeditions exp
4   ON e.id = exp.id_entrepot_destination
5   WHERE exp.date_expedition BETWEEN DATE_SUB(CURDATE(), INTERVAL 30 DAY) AND CURDATE();

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

nom_entrepot	date_expedition
Entrepot3	2023-11-02
Entrepot5	2023-10-16
Entrepot3	2023-11-01
Entrepot2	2023-10-13
Entrepot1	2023-10-08

Result 12 x Read Only

Output:

Action Output

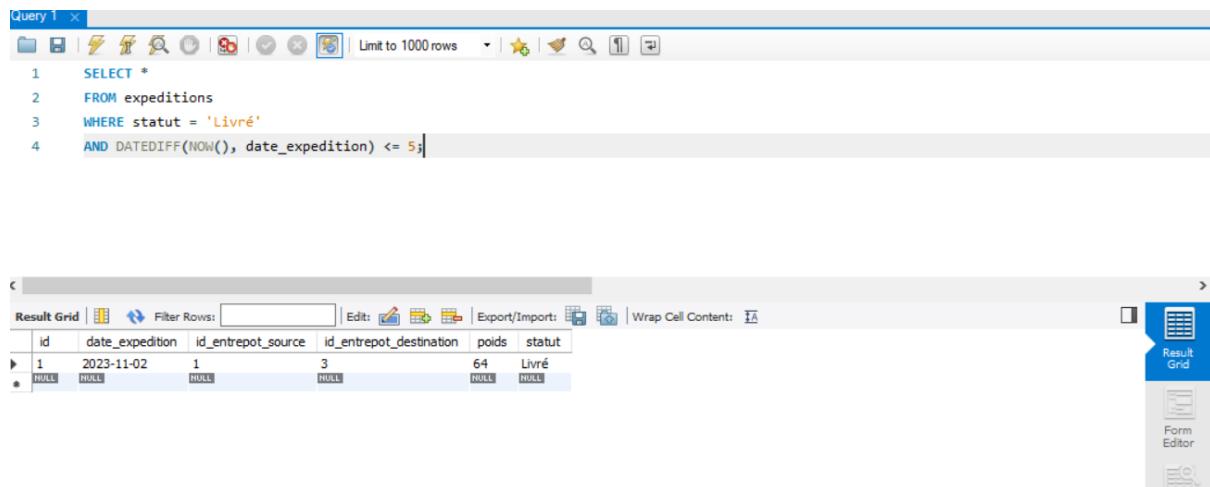
#	Time	Action	Message
8	20:44:01	SELECT MONTH(date_expedition) AS mois, COUNT(*) AS nombre_expéditions FROM expeditions WHERE Y...	4 row(s) returned
9	20:45:50	SELECT DISTINCT e.nom_entrepot FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.id_entrep...	5 row(s) returned
10	20:46:37	SELECT DISTINCT e.nom_entrepot, e.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e....	Error Code: 1054. Unknown column 'e.date_expedition' in 'field list'
11	20:46:53	SELECT e.nom_entrepot, exp.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.i...	7 row(s) returned
12	20:48:43	SELECT e.nom_entrepot, exp.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.i...	5 row(s) returned
13	20:50:11	SELECT e.nom_entrepot, exp.date_expedition FROM entrepots e INNER JOIN expeditions exp ON e.id = exp.i...	5 row(s) returned

```

-- Affichez les expéditions qui ont été livrées dans un délai de moins
de 5 jours ouvrables.
SELECT *
FROM expeditions
WHERE statut = 'Livré'
AND DATEDIFF(NOW(), date_expedition) <= 5;

```

Réponse :



Query 1

```

1  SELECT *
2  FROM expeditions
3  WHERE statut = 'Livré'
4  AND DATEDIFF(NOW(), date_expedition) <= 5;

```

Result Grid

	id	date_expedition	id_entrepot_source	id_entrepot_destination	poids	statut
▶	1	2023-11-02	1	3	64	Livré
◀	NULL	NULL	NULL	NULL	NULL	NULL

expeditions 18

Action Output

#	Time	Action	Message
15	20:54:38	SELECT * FROM Expeditions LIMIT 0, 1000	10 row(s) returned
16	20:55:24	SELECT * FROM expeditions WHERE statut = 'Livré' AND DATEDIFF(NOW(), date_expedition) <= 5 LIMIT 0, ...	0 row(s) returned
17	21:00:28	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
18	21:00:59	UPDATE expeditions SET statut = 'Livré' WHERE id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
19	21:01:10	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
20	21:01:17	SELECT * FROM expeditions WHERE statut = 'Livré' AND DATEDIFF(NOW(), date_expedition) <= 5 LIMIT 0, ...	1 row(s) returned

Requêtes complexes :

```

-- Affichez les expéditions en transit qui ont été initiées par un
entrepôt situé en Europe et à destination d'un entrepôt situé en Asie.
SELECT *
FROM expeditions AS e
INNER JOIN entrepots AS source_entrepot
    ON e.id_entrepot_source = source_entrepot.id
INNER JOIN entrepots AS destination_entrepot
    ON e.id_entrepot_destination = destination_entrepot.id
WHERE e.statut = 'En transit'
AND (source_entrepot.ville = 'Lyon' AND source_entrepot.pays =
'France')
AND (destination_entrepot.ville = 'Pekin' AND destination_entrepot.pays
= 'Chine');

```

Réponse :

Limit to 1000 rows

```

1 •  SELECT *
2   FROM expeditions AS e
3   JOIN entrepots AS source_entrepot
4     ON e.id_entrepot_source = source_entrepot.id
5   JOIN entrepots AS destination_entrepot
6     ON e.id_entrepot_destination = destination_entrepot.id
7   WHERE e.statut = 'en transit'
8   AND (source_entrepot.ville = 'Lyon' AND source_entrepot.pays = 'France')
9   AND (destination_entrepot.ville = 'Pekin' AND destination_entrepot.pays = 'Chine');
10

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Read Only

entrepot_source	id_entrepot_destination	poids	statut	id	nom_entrepot	adresse	ville	pays	id	nom_entrepot	adresse	ville	pays
2	91	En transit	1	Entrepot1	Adresse entrepot1	Lyon	France	2	Entrepot2	Adresse entrepot2	Pekin	Chine	

Result 26 | Action Output

#	Time	Action	Message
25	21:06:30	SELECT * FROM entrepots LIMIT 0, 1000	5 row(s) returned
26	21:06:44	SELECT * FROM expeditions LIMIT 0, 1000	10 row(s) returned
27	21:07:15	UPDATE expeditions SET statut = 'En transit' WHERE id = 6	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
28	21:07:18	SELECT * FROM expeditions AS e JOIN entrepots AS source_entrepot ON e.id_entrepot_source = source_entrepot.id	1 row(s) returned
29	21:08:50	SELECT source_entrepot.id AS source_entrepot_id, source_entrepot.nom_entrepot AS source_entrepot_nom, source_entrepot.ville AS source_entrepot_ville, source_entrepot.pays AS source_entrepot_pays FROM expeditions AS e JOIN entrepots AS source_entrepot ON e.id_entrepot_source = source_entrepot.id WHERE source_entrepot.ville = 'Lyon' AND source_entrepot.pays = 'France'	1 row(s) returned
30	21:09:04	SELECT * FROM expeditions AS e JOIN entrepots AS source_entrepot ON e.id_entrepot_source = source_entrepot.id WHERE source_entrepot.ville = 'Pekin' AND source_entrepot.pays = 'Chine'	1 row(s) returned

```

-- Affichez les entrepôts qui ont envoyé des expéditions à destination
d'un entrepôt situé dans le même pays.

SELECT DISTINCT source_entrepot.id,
               source_entrepot.nom_entrepot,
               source_entrepot.ville,
               source_entrepot.pays

FROM expeditions AS e
INNER JOIN entrepots AS source_entrepot ON e.id_entrepot_source =
source_entrepot.id
INNER JOIN entrepots AS destination_entrepot ON
e.id_entrepot_destination = destination_entrepot.id
WHERE source_entrepot.pays = destination_entrepot.pays;

```

Réponse :

```

1  SELECT DISTINCT source_entrepot.id,
2      source_entrepot.nom_entrepot,
3      source_entrepot.ville,
4      source_entrepot.pays
5  FROM expeditions AS e
6  INNER JOIN entrepots AS source_entrepot ON e.id_entrepot_source = source_entrepot.id
7  INNER JOIN entrepots AS destination_entrepot ON e.id_entrepot_destination = destination_entrepot.id
8  WHERE source_entrepot.pays = destination_entrepot.pays;
9

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | <img alt="refresh icon" data-bbox="838 14595 881 1

```

1 •  SELECT DISTINCT source_entrepot.id,
2           source_entrepot.nom_entrepot,
3           source_entrepot.ville,
4           source_entrepot.pays
5   FROM expeditions AS e
6   INNER JOIN entrepots AS source_entrepot ON e.id_entrepot_source = source_entrepot.id
7   INNER JOIN entrepots AS destination_entrepot ON e.id_entrepot_destination = destination_entrepot.id
8   WHERE source_entrepot.pays <> destination_entrepot.pays;

```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar is a query editor window containing the SQL code from the previous block. The results of the query are displayed in a 'Result Grid' table, which has columns: id, nom_entrepot, ville, and pays. The data shows five entries: Entrepot1 (Lyon, France), Entrepot2 (Pekin, Chine), Entrepot3 (Barcelone, Espagne), Entrepot4 (Lisbonne, Portugal), and Entrepot5 (Rome, Italie). Below the result grid is a 'Result 34' section, an 'Output' section, and an 'Action Output' section. The 'Action Output' section lists several database operations with their times, actions, and messages. For example, operation 35 shows a 'SELECT * from entrepots LIMIT 0, 1000' query returning 6 row(s) returned. Operation 36 shows an 'INSERT INTO expeditions(date_expedition,id_entrepot_source,id_entrepot_destination,poids,statut) VALUES...' query affected 1 row(s). Other operations include SELECT statements for source and destination warehouses.

```

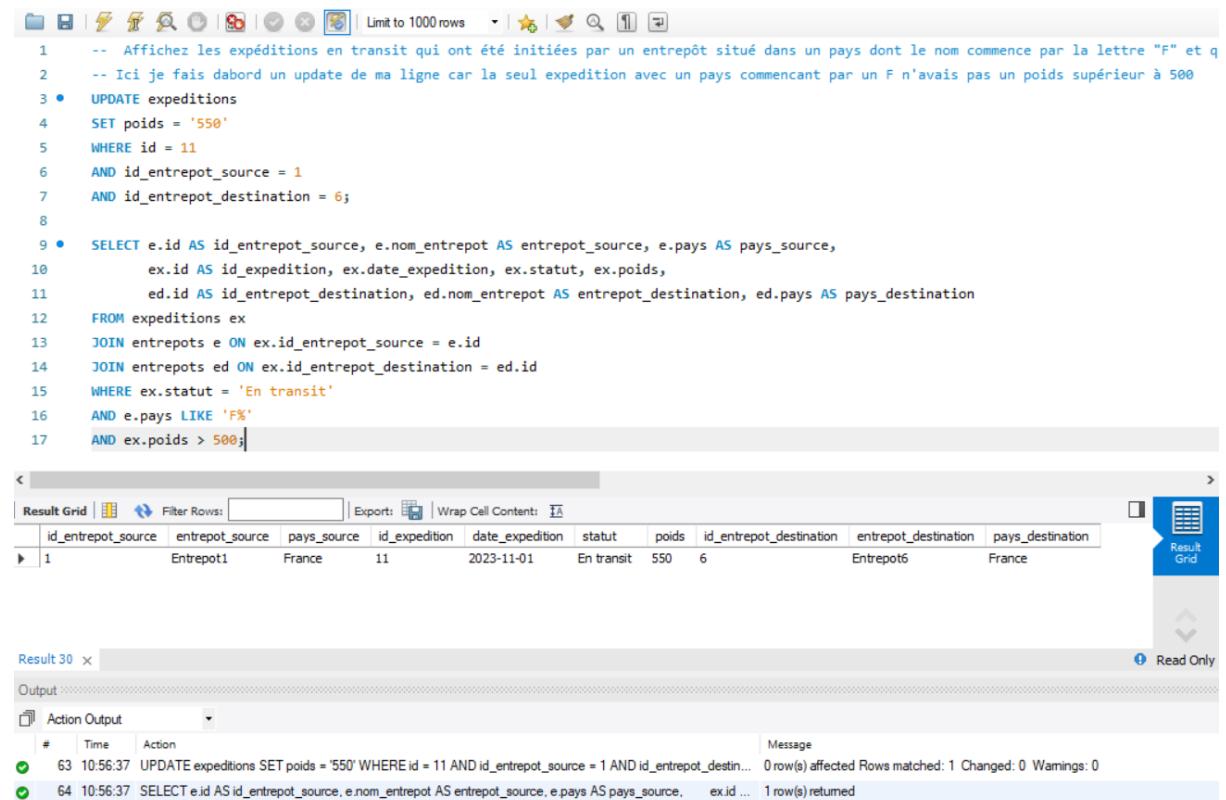
-- Affichez les expéditions en transit qui ont été initiées par un
entrepôt situé dans un pays dont le nom commence par la lettre "F" et
qui pèsent plus de 500 kg.

-- Ici je fais dabord un update de ma ligne car la seul expedition avec
un pays commençant par un F n'avais pas un poids supérieur à 500
UPDATE expeditions
SET poids = '550'
WHERE id = 11
AND id_entrepot_source = 1
AND id_entrepot_destination = 6;

SELECT e.id AS id_entrepot_source, e.nom_entrepot AS entrepot_source,
e.pays AS pays_source,
ex.id AS id_expedition, ex.date_expedition, ex.statut, ex.poids,
ed.id AS id_entrepot_destination, ed.nom_entrepot AS
entrepot_destination, ed.pays AS pays_destination
FROM expeditions ex
JOIN entrepots e ON ex.id_entrepot_source = e.id
JOIN entrepots ed ON ex.id_entrepot_destination = ed.id
WHERE ex.statut = 'En transit'
AND e.pays LIKE 'F%'
AND ex.poids > 500;

```

Réponse :



The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a text area contains the following SQL code:

```
1  -- Affichez les expéditions en transit qui ont été initiées par un entrepôt situé dans un pays dont le nom commence par la lettre "F" et q
2  -- Ici je fais dabord un update de ma ligne car la seul expedition avec un pays commençant par un F n'avais pas un poids supérieur à 500
3  UPDATE expeditions
4  SET poids = '550'
5  WHERE id = 11
6  AND id_entrepot_source = 1
7  AND id_entrepot_destination = 6;
8
9  • SELECT e.id AS id_entrepot_source, e.nom_entrepot AS entrepot_source, e.pays AS pays_source,
10    ex.id AS id_expedition, ex.date_expedition, ex.statut, ex.poids,
11    ed.id AS id_entrepot_destination, ed.nom_entrepot AS entrepot_destination, ed.pays AS pays_destination
12  FROM expeditions ex
13  JOIN entrepots e ON ex.id_entrepot_source = e.id
14  JOIN entrepots ed ON ex.id_entrepot_destination = ed.id
15  WHERE ex.statut = 'En transit'
16  AND e.pays LIKE 'F%'
17  AND ex.poids > 500;
```

Below the code, a result grid is displayed with the following data:

id_entrepot_source	entrepot_source	pays_source	id_expedition	date_expedition	statut	poids	id_entrepot_destination	entrepot_destination	pays_destination
1	Entrepot1	France	11	2023-11-01	En transit	550	6	Entrepot6	France

At the bottom, the output pane shows the following log entries:

- 63 10:56:37 UPDATE expeditions SET poids = '550' WHERE id = 11 AND id_entrepot_source = 1 AND id_entrepot_destination = 6; 0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
- 64 10:56:37 SELECT e.id AS id_entrepot_source, e.nom_entrepot AS entrepot_source, e.pays AS pays_source, ex.id AS id_expedition, ex.date_expedition, ex.statut, ex.poids, ed.id AS id_entrepot_destination, ed.nom_entrepot AS entrepot_destination, ed.pays AS pays_destination FROM expeditions ex JOIN entrepots e ON ex.id_entrepot_source = e.id JOIN entrepots ed ON ex.id_entrepot_destination = ed.id WHERE ex.statut = 'En transit' AND e.pays LIKE 'F%' AND ex.poids > 500; 1 row(s) returned

```
-- Affichez le nombre total d'expéditions pour chaque combinaison de
pays d'origine et de destination.

SELECT
    e1.pays AS pays_origine,
    e1.ville AS ville_origine,
    e2.pays AS pays_destination,
    e2.ville AS ville_destination,
    COUNT(*) AS nombre_expeditions
FROM expeditions AS exp
INNER JOIN entrepots AS e1
    ON exp.id_entrepot_source = e1.id
INNER JOIN entrepots AS e2
    ON exp.id_entrepot_destination = e2.id
WHERE e1.pays = e2.pays
GROUP BY e1.pays, e1.ville, e2.pays, e2.ville;
```

Réponse :

```

1 •  SELECT
2      e1.pays AS pays_origine,
3      e1.ville AS ville_origine,
4      e2.pays AS pays_destination,
5      e2.ville AS ville_destination,
6      COUNT(*) AS nombre_expeditions
7  FROM expeditions AS exp
8  INNER JOIN entrepots AS e1
9      ON exp.id_entrepot_source = e1.id
10 INNER JOIN entrepots AS e2
11      ON exp.id_entrepot_destination = e2.id
12 WHERE e1.pays = e2.pays
13 GROUP BY e1.pays, e1.ville, e2.pays, e2.ville;

```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

pays_origine	ville_origine	pays_destination	ville_destination	nombre_expeditions
France	Lyon	France	Paris	1

Result 31 x

Output ::::::::::::::::::::

Action Output

#	Time	Action	Message
64	10:56:37	SELECT e.id AS id_entrepot_source, e.nom_entrepot AS entrepot_source, e.pays AS pays_source, ex.id ...	1 row(s) returned
65	10:58:24	SELECT e1.pays AS pays_origine, e1.ville AS ville_origine, e2.pays AS pays_destination, e2.ville A... A...	1 row(s) returned

Read Only

```

-- Affichez les entrepôts qui ont envoyé des expéditions au cours des
30
-- derniers jours et dont le poids total des expéditions est supérieur
à 1000 kg.
-- Ici j'update en premier lieu ma ligne pour avoir un retour à ma
requête car aucune données n'allait être retourné
UPDATE expeditions
SET poids = '1300'
WHERE id = 1 AND date_expedition = '2023-11-02';

SELECT
    e.id AS id_entrepot,
    e.nom_entrepot,
    e.ville,
    e.pays,
    SUM(exp.poids) AS poids_total
FROM entrepots AS e
INNER JOIN expeditions AS exp
    ON exp.id_entrepot_source = e.id
WHERE exp.date_expedition >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)
GROUP BY e.id, e.nom_entrepot, e.ville, e.pays
HAVING poids_total > 1000;

```

Réponse :

```

 1 • SELECT
 2     e.id AS id_entrepot,
 3     e.nom_entrepot,
 4     e.ville,
 5     e.pays,
 6     SUM(exp.poids) AS poids_total
 7 FROM entrepots AS e
 8 INNER JOIN expeditions AS exp
 9     ON exp.id_entrepot_source = e.id
10 WHERE exp.date_expedition >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)
11 GROUP BY e.id, e.nom_entrepot, e.ville, e.pays
12 HAVING poids_total > 1000;

```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

	id_entrepot	nom_entrepot	ville	pays	poids_total
1	Entrepot1	Lyon	France		1941

Result 32 x

Output

Action Output

#	Time	Action	Message
65	10:58:24	SELECT e1.pays AS pays_origine, e1.ville AS ville_origine, e2.pays AS pays_destination, e2.ville AS ville_destination	1 row(s) returned
66	10:59:42	SELECT e.id AS id_entrepot, e.nom_entrepot, e.ville, e.pays, SUM(exp.poids) AS poids_total FROM entrepots e INNER JOIN expeditions exp ON exp.id_entrepot_source = e.id WHERE exp.date_expedition >= DATE_SUB(CURDATE(), INTERVAL 30 DAY) GROUP BY e.id, e.nom_entrepot, e.ville, e.pays HAVING poids_total > 1000;	1 row(s) returned

-- Affichez les expéditions qui ont été livrées avec un retard de plus de 2 jours ouvrables.

```

SELECT *
FROM expeditions
WHERE statut = 'Livré'
AND DATEDIFF(NOW(), date_expedition) > 2;

```

Réponse :

1 • `SELECT *
FROM expeditions
WHERE statut = 'Livré'
AND DATEDIFF(NOW(), date_expedition) > 2;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | **Result Grid** | Form Editor | Apply

id	date_expedition	id_entrepot_source	id_entrepot_destination	poids	statut
2	2023-10-16	2	5	102	Livré
4	2023-09-12	4	2	32	Livré
8	2022-02-18	3	1	273	Livré
10	2023-09-22	5	3	195	Livré
• HULL	HULL	HULL	HULL	NULL	NULL

expeditions 44 x

Output

Action Output

#	Time	Action	Message
46	21:19:36	UPDATE expeditions SET poids = 648 WHERE id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
47	21:19:42	SELECT * FROM expeditions LIMIT 0, 1000	11 row(s) returned
48	21:19:49	SELECT * FROM expeditions AS e JOIN entrepots AS source_entrepot ON e.id_entrepot_source = source_entrepot.id	0 row(s) returned
49	21:23:55	SELECT * FROM expeditions WHERE statut = 'livrée' AND DATEDIFF(NOW(), date_expedition) > 2 LIMIT 0, ...	0 row(s) returned
50	21:24:10	SELECT * FROM expeditions LIMIT 0, 1000	11 row(s) returned
51	21:24:38	SELECT * FROM expeditions WHERE statut = 'Livré' AND DATEDIFF(NOW(), date_expedition) > 2 LIMIT 0, 1...	4 row(s) returned

```
-- Affichez le nombre total d'expéditions pour chaque jour du mois en cours, trié par ordre décroissant
SELECT
    DAY(exp.date_expedition) AS jour,
    COUNT(*) AS nombre_expeditions
FROM expeditions AS exp
WHERE MONTH(exp.date_expedition) = MONTH(CURDATE())
AND YEAR(exp.date_expedition) = YEAR(CURDATE())
GROUP BY DAY(exp.date_expedition)
ORDER BY nombre_expeditions DESC;
```

Réponse :

```

1 • SELECT
2     DAY(exp.date_expedition) AS jour,
3     COUNT(*) AS nombre_expeditions
4 FROM expeditions AS exp
5 WHERE MONTH(exp.date_expedition) = MONTH(CURDATE())
6 AND YEAR(exp.date_expedition) = YEAR(CURDATE())
7 GROUP BY DAY(exp.date_expedition)
8 ORDER BY nombre_expeditions DESC;

```

jour	nombre_expeditions
1	2
2	1

Result 33 x

Output

#	Time	Action	Message
66	10:59:42	SELECT e.id AS id_entrepot, e.nom_entrepot, e.ville, e.pays, SUM(exp.poids) AS poids_total FR...	1 row(s) returned
67	11:01:36	SELECT DAY(exp.date_expedition) AS jour, COUNT(*) AS nombre_expeditions FROM expeditions AS ex...	2 row(s) returned

T-SQL :

```

-- Créez une vue qui affiche les informations suivantes pour chaque
entrepôt :
-- nom de l'entrepôt, adresse complète, nombre d'expéditions envoyées
au
-- cours des 30 derniers jours
CREATE VIEW Vue_Entrepot_Information AS
SELECT
    e.nom_entrepot,
    CONCAT(e.adresse, ' ', e.ville, ' ', e.pays) AS adresse_complete,
    COUNT(exp.id) AS nombre_expeditions
FROM entrepots AS e
LEFT JOIN expeditions AS exp ON exp.id_entrepot_source = e.id
WHERE exp.date_expedition >= CURDATE() - INTERVAL 30 DAY
GROUP BY e.id, e.nom_entrepot, e.adresse, e.ville, e.pays;

SELECT * FROM Vue_Entrepot_Information

```

Réponse :

Limit to 1000 rows

```

5     COUNT(exp.id) AS nombre_expeditions
6     FROM entrepots AS e
7     LEFT JOIN expeditions AS exp ON exp.id_entrepot_source = e.id
8     WHERE exp.date_expedition >= CURDATE() - INTERVAL 30 DAY
9     GROUP BY e.id, e.nom_entrepot, e.adresse, e.ville, e.pays;
10
11 •  SELECT * FROM Vue_Entrepot_Informations;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

nom_entrepot	adresse_complete	nombre_expeditions
Entrepot1	Adresse entrepot1, Lyon, France	3
Entrepot2	Adresse entrepot2, Pekin, Chine	2
Entrepot3	Adresse entrepot3, Barcelone, Espagne	1
Entrepot5	Adresse entrepot5, Rome, Italie	1
Entrepot4	Adresse entrepot4, Lisbonne, Portugal	1

Output

Action Output

#	Time	Action	Message
33	09:48:23	UPDATE expeditions SET poids = '550' WHERE id = 11 AND id_entrepot_source = 1 AND id_entrepot_destin...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
34	09:48:29	SELECT * FROM expeditions LIMIT 0, 1000	11 row(s) returned
35	09:48:36	SELECT e.id AS id_entrepot_source, e.nom_entrepot AS entrepot_source, e.pays AS pays_source, ex.id ...	1 row(s) returned
36	09:50:37	SELECT DAY(exp.date_expedition) AS jour, COUNT(*) AS nombre_expeditions FROM expeditions AS...	2 row(s) returned
37	09:53:38	CREATE VIEW Vue_Entrepot_Informations AS SELECT e.nom_entrepot, CONCAT(e.adresse, ',', e.ville, ...)	0 row(s) affected
38	09:54:33	SELECT * FROM Vue_Entrepot_Informations LIMIT 0, 1000	5 row(s) returned

```

-- Créez une procédure stockée qui prend en entrée l'ID d'un entrepôt
et
-- renvoie le nombre total d'expéditions envoyées par cet entrepôt au
cours du
-- dernier mois.

DELIMITER //

CREATE PROCEDURE GetTotalExpeditionsByEntrepot(
    IN entrepot_id INT
)
BEGIN
    DECLARE total_expeditions INT;
    DECLARE entrepot_nom VARCHAR(140);
    DECLARE entrepot_ville VARCHAR(120);
    DECLARE entrepot_pays VARCHAR(120);

    DECLARE last_month_start DATE;
    DECLARE last_month_end DATE;

    -- Calculer le début et la fin du mois dernier
    SET last_month_start = DATE_SUB(LAST_DAY(NOW() - INTERVAL 2 MONTH),
    INTERVAL DAY(LAST_DAY(NOW() - INTERVAL 2 MONTH)) - 1 DAY);
    SET last_month_end = LAST_DAY(NOW() - INTERVAL 1 MONTH);

```

```

-- Sélectionner le nom, la ville et le pays de l'entrepôt
SELECT nom_entrepot, ville, pays INTO entrepot_nom, entrepot_ville,
entrepot_pays
FROM entrepots
WHERE id = entrepot_id;

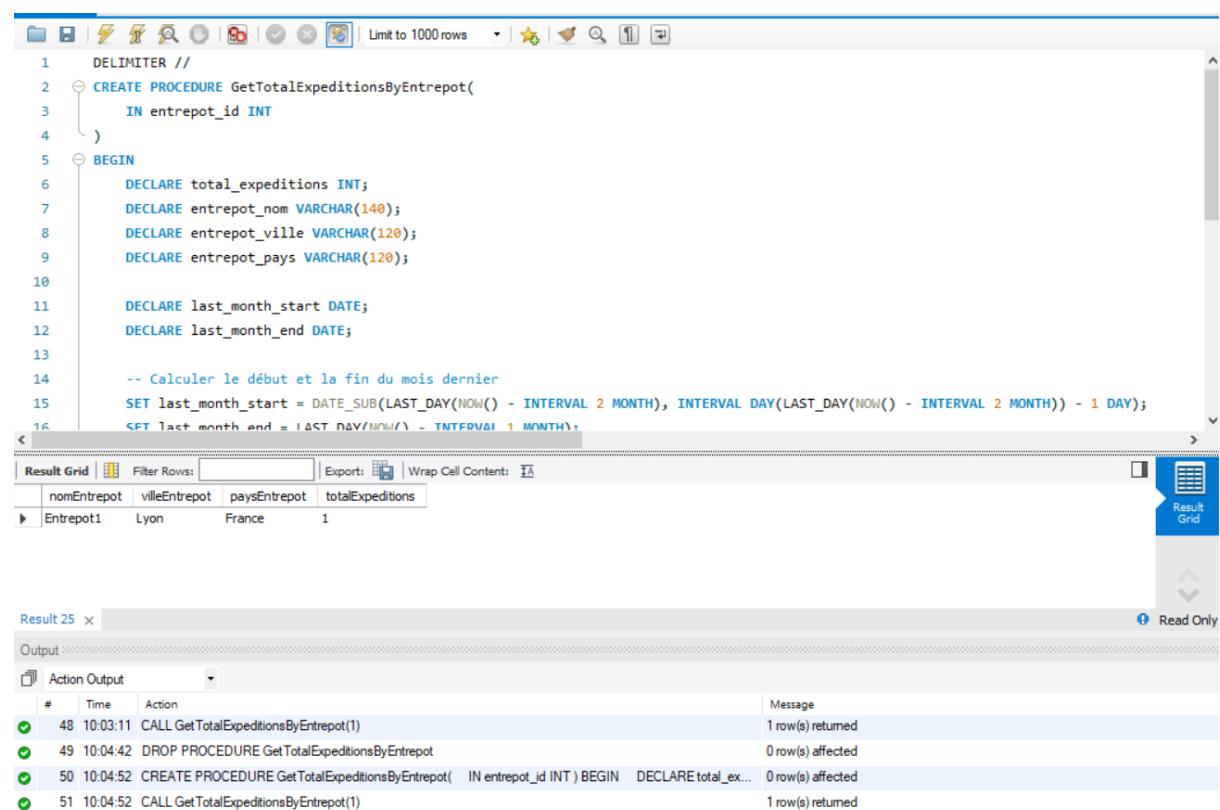
-- Sélectionner le nombre total d'expéditions pour l'entrepôt donné
et le mois dernier
SELECT COUNT(*) INTO total_expeditions
FROM expeditions
WHERE id_entrepot_source = entrepot_id
AND date_expedition BETWEEN last_month_start AND last_month_end;

-- Retourner le résultat
SELECT entrepot_nom AS nomEntrepot, entrepot_ville AS
villeEntrepot, entrepot_pays AS paysEntrepot, total_expeditions AS
totalExpeditions;
END //
DELIMITER ;

CALL GetTotalExpeditionsByEntrepot(1)

```

Réponse :



The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the MySQL code for creating a stored procedure named `GetTotalExpeditionsByEntrepot`. The procedure takes an integer parameter `entrepot_id` and returns the name, city, and country of the warehouse, along with the total number of shipments for the last month. The code includes declarations for variables and the calculation of the last month's start and end dates using MySQL functions.
- Result Grid:** Shows the output of the stored procedure for warehouse `Entrepot1` located in `Lyon`, `France`, with a total of `1` expedition.
- Output Tab:** Displays the history of actions taken in the session, including the creation of the procedure, its execution, and its subsequent drop.

```
-- Créez une fonction qui prend en entrée une date et renvoie le nombre
total
-- d'expéditions livrées ce jour-là.

DELIMITER //
CREATE FUNCTION GetTotalDeliveredShipments(dateToCheck DATE)
RETURNS INT
DETERMINISTIC
READS SQL DATA
BEGIN
    DECLARE totalDelivered INT;

    -- Sélectionner le nombre total d'expéditions livrées pour la date
    donnée
    SELECT COUNT(*) INTO totalDelivered
    FROM expeditions
    WHERE date_expedition = dateToCheck
    AND statut = 'Livré';

    -- Retourner le nombre total d'expéditions livrées
    RETURN totalDelivered;
END //
DELIMITER ;

SELECT GetTotalDeliveredShipments('2023-11-02')
```

Réponse :

The screenshot shows the MySQL Workbench interface. In the top pane, the SQL editor contains the following code:

```

1  DELIMITER //
2  •  CREATE FUNCTION GetTotalDeliveredShipments(dateToCheck DATE)
3  RETURNS INT
4  DETERMINISTIC
5  READS SQL DATA
6  BEGIN
7      DECLARE totalDelivered INT;
8
9      -- Sélectionner le nombre total d'expéditions livrées pour la date donnée
10     SELECT COUNT(*) INTO totalDelivered
11     FROM expeditions
12     WHERE date_expedition = dateToCheck
13     AND statut = 'Livré';
14
15     -- Retourner le nombre total d'expéditions livrées
16     RETURN totalDelivered;
17 END //
18 DELIMITER ;

```

The function is created successfully. In the bottom pane, the results of the query `GetTotalDeliveredShipments('2023-11-02')` are shown in a grid:

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
GetTotalDeliveredShipments('2023-11-02')			
1			

The output pane shows the following log entries:

Action Output	#	Time	Action	Message
	61	10:31:37	CREATE FUNCTION GetTotalDeliveredShipments(dateToCheck DATE) RETURNS INT DETERMINISTIC R...	0 row(s) affected
	62	10:31:37	SELECT GetTotalDeliveredShipments('2023-11-02') LIMIT 0, 1000	1 row(s) returned

BONUS :

Ajoutez une table "clients" contenant les colonnes suivantes :

```
-- Ajoutez une table "clients" contenant les colonnes suivantes :
CREATE TABLE clients(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nom VARCHAR(120),
    adresse VARCHAR(200),
    ville VARCHAR(160),
    pays VARCHAR(80)
);
```

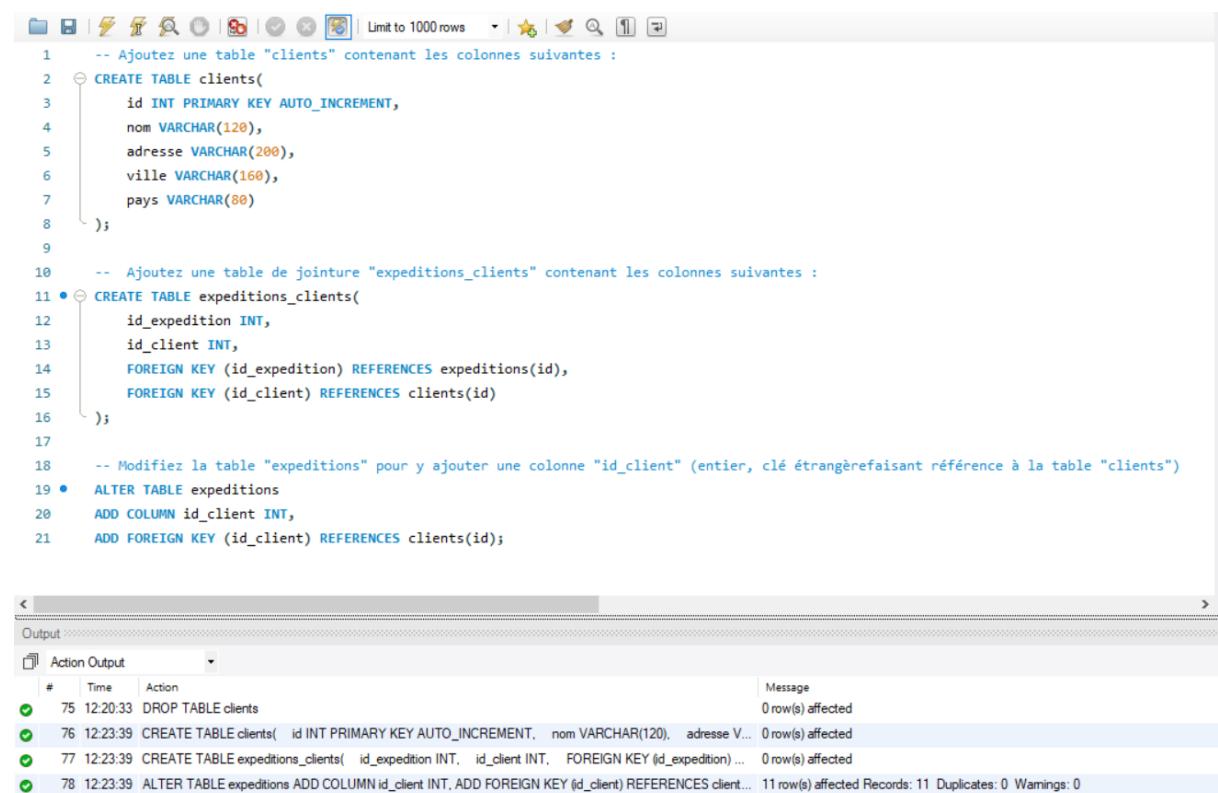
Ajoutez une table de jointure "expeditions_clients" contenant les colonnes suivantes :

```
-- Ajoutez une table de jointure "expeditions_clients" contenant les
colonnes suivantes :
CREATE TABLE expeditions_clients(
    id_expedition INT,
    id_client INT,
    FOREIGN KEY (id_expedition) REFERENCES expeditions(id),
    FOREIGN KEY (id_client) REFERENCES clients(id)
);
```

Modifiez la table "expeditions" pour y ajouter une colonne "id_client" (entier, clé étrangère faisant référence à la table "clients").

```
-- Modifiez la table "expeditions" pour y ajouter une colonne
"id_client" (entier, clé étrangère faisant référence à la table
"clients")
ALTER TABLE expeditions
ADD COLUMN id_client INT,
ADD FOREIGN KEY (id_client) REFERENCES clients(id);
```

Réponse :



The screenshot shows the MySQL Workbench interface with the 'clients' schema selected. The 'SQL' tab displays the following code:

```
-- Ajoutez une table "clients" contenant les colonnes suivantes :
CREATE TABLE clients(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nom VARCHAR(120),
    adresse VARCHAR(200),
    ville VARCHAR(160),
    pays VARCHAR(80)
);

-- Ajoutez une table de jointure "expeditions_clients" contenant les colonnes suivantes :
CREATE TABLE expeditions_clients(
    id_expedition INT,
    id_client INT,
    FOREIGN KEY (id_expedition) REFERENCES expeditions(id),
    FOREIGN KEY (id_client) REFERENCES clients(id)
);

-- Modifiez la table "expeditions" pour y ajouter une colonne "id_client" (entier, clé étrangère faisant référence à la table "clients")
ALTER TABLE expeditions
ADD COLUMN id_client INT,
ADD FOREIGN KEY (id_client) REFERENCES clients(id);
```

The 'Output' tab shows the execution results:

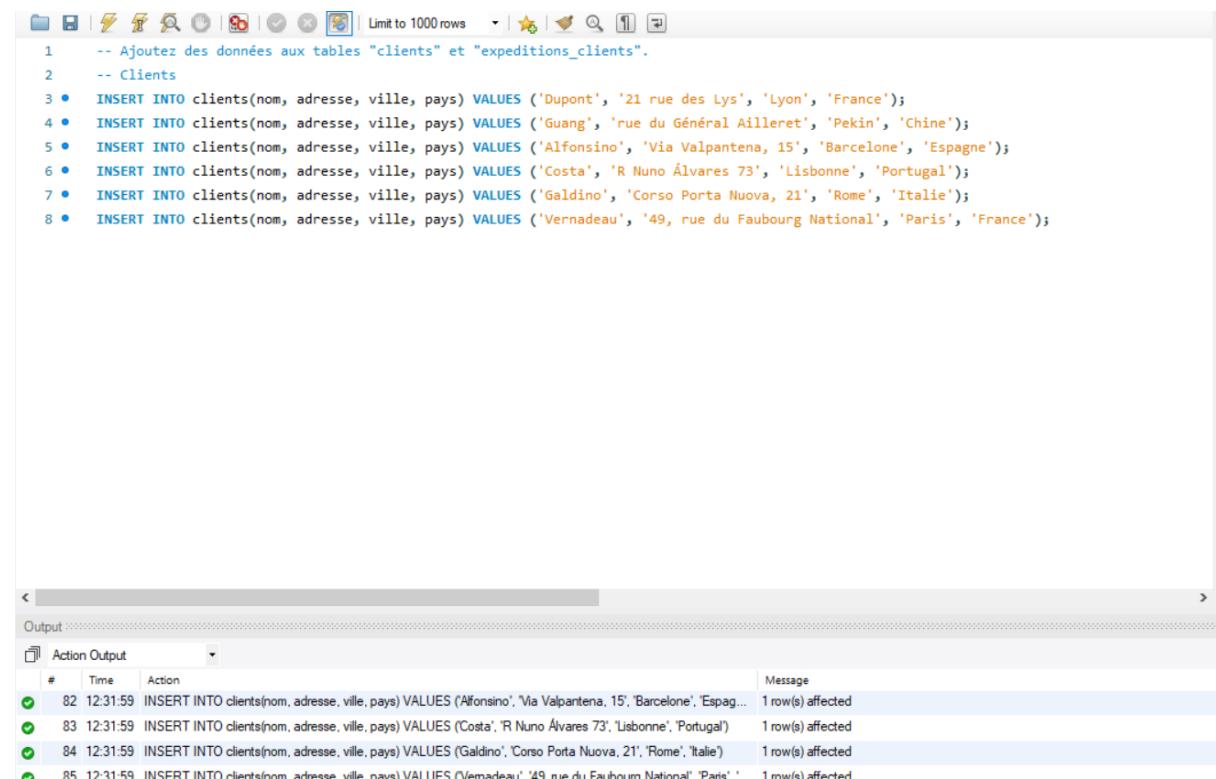
#	Time	Action	Message
75	12:20:33	DROP TABLE clients	0 row(s) affected
76	12:23:39	CREATE TABLE clients(id INT PRIMARY KEY AUTO_INCREMENT, nom VARCHAR(120), adresse V...)	0 row(s) affected
77	12:23:39	CREATE TABLE expeditions_clients(id_expedition INT, id_client INT, FOREIGN KEY (id_expedition) REFERENCES expeditions(id), FOREIGN KEY (id_client) REFERENCES clients(id))	0 row(s) affected
78	12:23:39	ALTER TABLE expeditions ADD COLUMN id_client INT, ADD FOREIGN KEY (id_client) REFERENCES clients(id)	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0

Ajoutez des données aux tables "clients" et "expeditions_clients".

Clients :

```
-- Ajoutez des données aux tables "clients" et "expeditions_clients".
-- Clients
INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Dupont', '21
rue des Lys', 'Lyon', 'France');
INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Guang', 'rue du
Général Ailleret', 'Pekin', 'Chine');
INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Alfonsino',
'Via Valpantena, 15', 'Barcelone', 'Espagne');
INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Costa', 'R Nuno
Álvares 73', 'Lisbonne', 'Portugal');
INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Galdino',
'Corso Porta Nuova, 21', 'Rome', 'Italie');
INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Vernadeau',
'49, rue du Faubourg National', 'Paris', 'France');
```

Réponse :



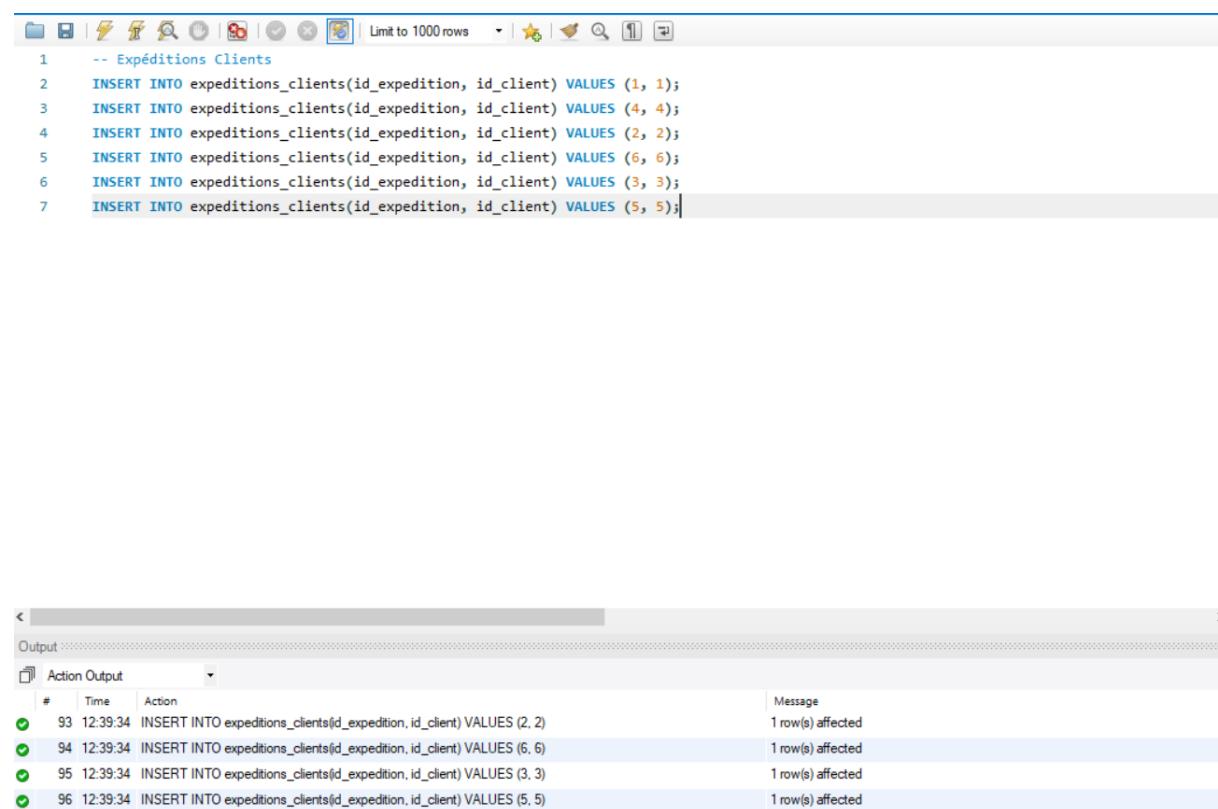
The screenshot shows the MySQL Workbench interface. The SQL editor pane contains the SQL code for inserting data into the 'clients' table. The output pane shows the results of the executed queries, indicating that 5 rows were affected for each of the 5 insert statements.

#	Time	Action	Message
82	12:31:59	INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Alfonsino', 'Via Valpantena, 15', 'Barcelone', 'Espagne')	1 row(s) affected
83	12:31:59	INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Costa', 'R Nuno Álvares 73', 'Lisbonne', 'Portugal')	1 row(s) affected
84	12:31:59	INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Galdino', 'Corso Porta Nuova, 21', 'Rome', 'Italie')	1 row(s) affected
85	12:31:59	INSERT INTO clients(nom, adresse, ville, pays) VALUES ('Vernadeau', '49, rue du Faubourg National', 'Paris', 'France')	1 row(s) affected

Expéditions Clients :

```
-- Expéditions Clients
INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (1, 1);
INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (4, 4);
INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (2, 2);
INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (6, 6);
INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (3, 3);
INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (5, 5);
```

Réponse :



The screenshot shows the MySQL Workbench interface. The SQL editor window contains the provided SQL script. The output window shows the results of the executed statements, which are all successful INSERT operations, each resulting in 1 row(s) affected.

#	Time	Action	Message
93	12:39:34	INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (2, 2)	1 row(s) affected
94	12:39:34	INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (6, 6)	1 row(s) affected
95	12:39:34	INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (3, 3)	1 row(s) affected
96	12:39:34	INSERT INTO expeditions_clients(id_expedition, id_client) VALUES (5, 5)	1 row(s) affected

Écrivez des requêtes pour extraire les informations suivantes :

Pour chaque client, affichez son nom, son adresse complète, le nombre total d'expéditions qu'il a envoyées et le nombre total d'expéditions qu'il a reçues :

```

-- Pour chaque client, affichez son nom, son adresse complète, le
-- nombre total d'expéditions qu'il a envoyées et le nombre total
-- d'expéditions qu'il a reçues
-- En premier lieu j'update tout mes id client pour les faire
correspondre avec les id expedition auquel je les ai attribué
UPDATE expeditions
SET id_client =
CASE
    WHEN id IN (1) THEN 1
    WHEN id IN (2) THEN 2
    WHEN id IN (3) THEN 3
    WHEN id IN (4) THEN 4
    WHEN id IN (5) THEN 5
    WHEN id IN (6) THEN 6
    ELSE id_client
END
WHERE id IN (1, 2, 3, 4, 5, 6);

SELECT
    c.nom AS Nom_Client,
    c.adresse AS Adresse_Client,
    COUNT(DISTINCT ec.id_expedition) AS Nombre_Expeditions_Envoyées,
    COUNT(DISTINCT e.id) AS Nombre_Expeditions_Reçues
FROM
    clients c
LEFT JOIN
    expeditions_clients ec ON c.id = ec.id_client
LEFT JOIN
    expeditions e ON c.id = e.id_client
GROUP BY
    c.id, c.nom, c.adresse;

```

Réponse :

```

 1  SELECT
 2      c.nom AS Nom_Client,
 3      c.adresse AS Adresse_Client,
 4      COUNT(DISTINCT ec.id_expedition) AS Nombre_Expeditions_Envoyées,
 5      COUNT(DISTINCT e.id) AS Nombre_Expeditions_Reçues
 6  FROM
 7      clients c
 8  LEFT JOIN
 9      expeditions_clients ec ON c.id = ec.id_client
10  LEFT JOIN
11      expeditions e ON c.id = e.id_client
12  GROUP BY
13      c.id, c.nom, c.adresse;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Read Only

Nom_Client	Adresse_Client	Nombre_Expeditions_Envoyées	Nombre_Expeditions_Reçues
Dupont	21 rue des Lys	1	1
Guang	rue du Général Alleret	2	1
Alfonsino	Via Valpantena, 15	2	1
Costa	R Nuno Álvares 73	1	1
Galdino	Corsó Porta Nuova, 21	1	1
Vernadeau	49, rue du Faubourg National	1	1

Action Output

#	Time	Action	Message
106	12:48:11	SELECT *from expeditions LIMIT 0, 1000	11 row(s) returned
107	12:49:03	SELECT c.nom AS Nom_Client, c.adresse AS Adresse_Client, COUNT(DISTINCT ec.id_expedition) AS...	6 row(s) returned