

1. Напишіть SQL-запит, який для таблиці `orders` з атрибута `date` витягує рік, місяць і число. Виведіть на екран їх у три окремі атрибути поряд з атрибутом `id` та оригінальним атрибутом `date` (всього вийде 5 атрибутів).

```
USE testbase;
SELECT
    id,
    date,
    YEAR(date) AS order_year,
    MONTH(date) AS order_month,
    DAY(date) AS order_day
FROM
    Orders;
```

The screenshot shows a database interface with two panes. The top pane contains the SQL code:

```
USE testbase;
SELECT
    id,
    date,
    YEAR(date) AS order_year,
    MONTH(date) AS order_month,
    DAY(date) AS order_day
FROM
    Orders;
```

The bottom pane displays the results of the query in a table titled "orders 1". The table has columns: id, date, order_year, order_month, and order_day. The data shows 9 rows of orders from July 1996, with the day column explicitly named.

	id	date	order_year	order_month	order_day
1	10,248	1996-07-04	1996	7	4
2	10,249	1996-07-05	1996	7	5
3	10,250	1996-07-08	1996	7	8
4	10,251	1996-07-08	1996	7	8
5	10,252	1996-07-09	1996	7	9
6	10,253	1996-07-10	1996	7	10
7	10,254	1996-07-11	1996	7	11
8	10,255	1996-07-12	1996	7	12
9	10,256	1996-07-15	1996	7	15

2. Напишіть SQL-запит, який для таблиці `orders` до атрибута `date` додає один день. На екран виведіть атрибут `id`, оригінальний атрибут `date` та результат додавання.

```
USE testbase;
SELECT
```

```

    id,
    date AS original_date,
    DATE_ADD(date, INTERVAL 1 DAY) AS date_plus_one_day
FROM
    Orders;

```

The screenshot shows the MySQL Workbench interface with two tabs: 'Script-3' and 'Script-10'. The 'Script-3' tab contains the provided SQL code. The 'Script-10' tab shows the results of the query:

```

USE testbase;

SELECT
    id,
    date AS original_date,
    DATE_ADD(date, INTERVAL 1 DAY) AS date_plus_one_day
FROM
    Orders;

```

The results table has three columns: 'id', 'original_date', and 'date_plus_one_day'. The data is as follows:

	id	original_date	date_plus_one_day
1	10,248	1996-07-04	1996-07-05
2	10,249	1996-07-05	1996-07-06
3	10,250	1996-07-08	1996-07-09
4	10,251	1996-07-08	1996-07-09
5	10,252	1996-07-09	1996-07-10
6	10,253	1996-07-10	1996-07-11
7	10,254	1996-07-11	1996-07-12
8	10,255	1996-07-12	1996-07-13
9	10,256	1996-07-15	1996-07-16

3. Напишіть SQL-запит, який для таблиці `orders` для атрибута `date` відображає кількість секунд з початку відліку (показує його значення `timestamp`). Для цього потрібно знайти та застосувати необхідну функцію. На екран виведіть атрибут `id`, оригінальний атрибут `date` та результат роботи функції.

```

USE testbase;
SELECT
    id,
    date AS original_date,

```

```
UNIX_TIMESTAMP(date) AS timestamp_seconds  
FROM  
Orders;
```

The screenshot shows a MySQL Workbench environment. In the top-left pane, there is a code editor with the following SQL query:

```
USE testbase;  
SELECT  
    id,  
    date AS original_date,  
    UNIX_TIMESTAMP(date) AS timestamp_seconds  
FROM  
    Orders;
```

In the bottom-right pane, there is a results grid titled "orders 1" with the following data:

	id	original_date	timestamp_seconds
1	10,248	1996-07-04	836,427,600
2	10,249	1996-07-05	836,514,000
3	10,250	1996-07-08	836,773,200
4	10,251	1996-07-08	836,773,200
5	10,252	1996-07-09	836,859,600
6	10,253	1996-07-10	836,946,000
7	10,254	1996-07-11	837,032,400
8	10,255	1996-07-12	837,118,800
9	10,256	1996-07-15	837,378,000

4. Напишіть SQL-запит, який рахує, скільки таблиця `orders` містить рядків з атрибутом `date` у межах між `1996-07-10 00:00:00` та `1996-10-08 00:00:00`.

```
USE testbase;  
SELECT  
    COUNT(*) AS orders_in_range  
FROM  
    Orders  
WHERE  
    date BETWEEN '1996-07-10' AND '1996-10-08';
```

The screenshot shows the MySQL Workbench interface. At the top, there are two tabs: "Script-3" and "Script-10". The "Script-3" tab contains the following SQL query:

```
USE testbase;  
SELECT  
    COUNT(*) AS orders_in_range  
FROM  
    Orders  
WHERE  
    date BETWEEN '1996-07-10' AND '1996-10-08';
```

The "Script-10" tab is partially visible on the right. Below the tabs, the "Results" tab is active, showing the output of the query:

	orders_in_range
1	72

5. Напишіть SQL-запит, який для таблиці `orders` виводить на екран атрибут `id`, атрибут `date` та JSON-об'єкт `{"id": <атрибут id рядка>, "date": <атрибут date рядка>}`. Для створення JSON-об'єкта використайте функцію.

```
USE testbase;  
SELECT  
    id,  
    date,  
    JSON_OBJECT('id', id, 'date', date) AS order_json  
FROM  
    Orders;
```

```
USE testbase;

SELECT
    id,
    date,
    JSON_OBJECT('id', id, 'date', date) AS order_json
FROM
    Orders;
```

Orders 1 × Statistics 1

Enter a SQL expression to filter rows

	123 ↗ id	⌚ date	{ } order_json
1	10,248	1996-07-04	{"id": 10248, "date": "1996-07-04"}
2	10,249	1996-07-05	{"id": 10249, "date": "1996-07-05"}
3	10,250	1996-07-08	{"id": 10250, "date": "1996-07-08"}
4	10,251	1996-07-08	{"id": 10251, "date": "1996-07-08"}
5	10,252	1996-07-09	{"id": 10252, "date": "1996-07-09"}
6	10,253	1996-07-10	{"id": 10253, "date": "1996-07-10"}
7	10,254	1996-07-11	{"id": 10254, "date": "1996-07-11"}
8	10,255	1996-07-12	{"id": 10255, "date": "1996-07-12"}
9	10,256	1996-07-15	{"id": 10256, "date": "1996-07-15"}