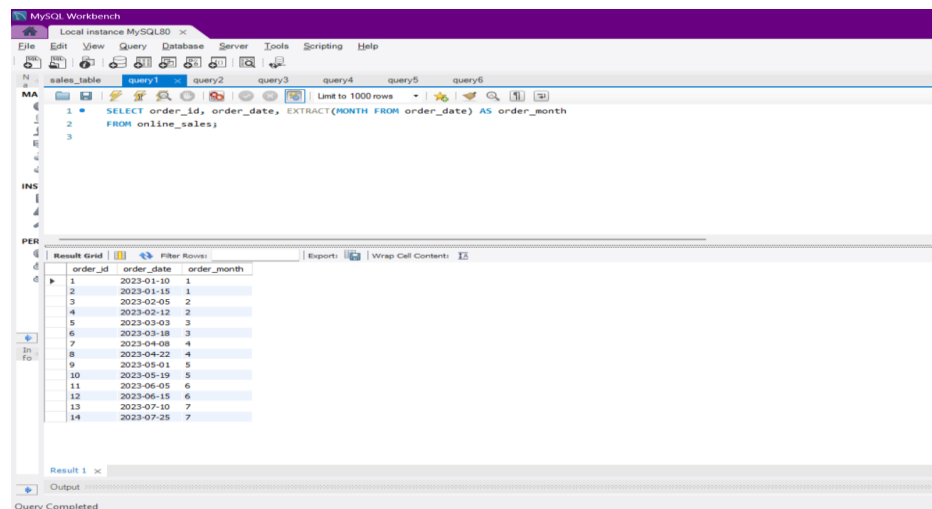


QUERIES AND OUTPUT:

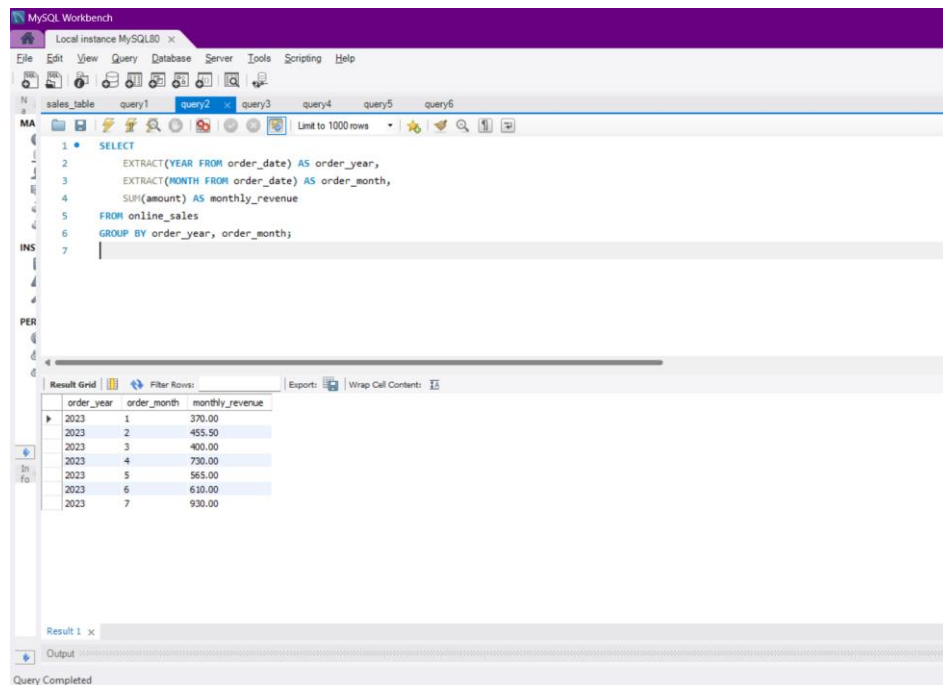


MySQL Workbench interface showing a query executed on the 'online_sales' table. The query selects 'order_id', 'order_date', and 'order_month' (extracted from 'order_date'). The result grid displays 14 rows of data.

```
1 SELECT order_id, order_date, EXTRACT(MONTH FROM order_date) AS order_month
2 FROM online_sales;
```

order_id	order_date	order_month
1	2023-01-10	1
2	2023-01-15	1
3	2023-02-05	2
4	2023-02-12	2
5	2023-03-03	3
6	2023-03-18	3
7	2023-04-08	4
8	2023-04-22	4
9	2023-05-01	5
10	2023-05-19	5
11	2023-06-05	6
12	2023-06-15	6
13	2023-07-10	7
14	2023-07-25	7

[A]

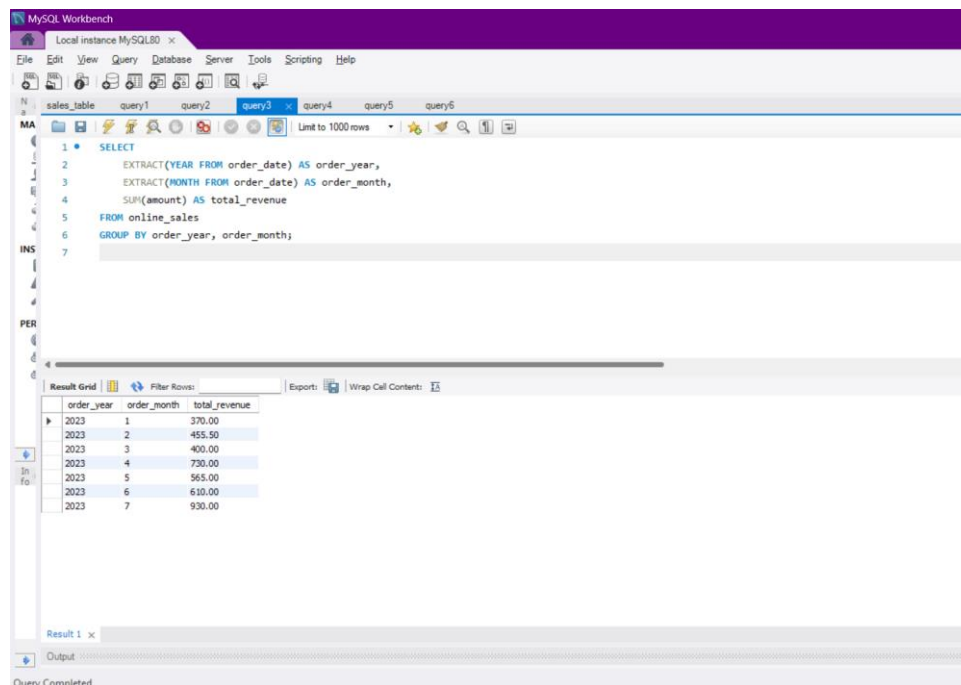


MySQL Workbench interface showing a query executed on the 'online_sales' table. The query selects 'order_year' (extracted from 'order_date'), 'order_month' (extracted from 'order_date'), and 'monthly_revenue' (sum of 'amount'). The result grid displays 7 rows of data.

```
1 SELECT
2   EXTRACT(YEAR FROM order_date) AS order_year,
3   EXTRACT(MONTH FROM order_date) AS order_month,
4   SUM(amount) AS monthly_revenue
5 FROM online_sales
6 GROUP BY order_year, order_month;
```

order_year	order_month	monthly_revenue
2023	1	370.00
2023	2	455.50
2023	3	400.00
2023	4	730.00
2023	5	565.00
2023	6	610.00
2023	7	930.00

[B]



MySQL Workbench interface showing a query executed on the 'online_sales' table. The query selects 'order_year' (extracted from 'order_date'), 'order_month' (extracted from 'order_date'), and 'total_revenue' (sum of 'amount'). The result grid displays 7 rows of data.

```
1 SELECT
2   EXTRACT(YEAR FROM order_date) AS order_year,
3   EXTRACT(MONTH FROM order_date) AS order_month,
4   SUM(amount) AS total_revenue
5 FROM online_sales
6 GROUP BY order_year, order_month;
```

order_year	order_month	total_revenue
2023	1	370.00
2023	2	455.50
2023	3	400.00
2023	4	730.00
2023	5	565.00
2023	6	610.00
2023	7	930.00

[C]

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

query1 query2 query3 query4 query5 query6

Limit to 1000 rows

```

1 SELECT
2   EXTRACT(YEAR FROM order_date) AS order_year,
3   EXTRACT(MONTH FROM order_date) AS order_month,
4   COUNT(DISTINCT order_id) AS total_orders
5 FROM online_sales
6 GROUP BY order_year, order_month;
7

```

Result Grid

order_year	order_month	total_orders
2023	1	2
2023	2	2
2023	3	2
2023	4	2
2023	5	2
2023	6	2
2023	7	2

Query Completed

[D]

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

query1 query2 query3 query4 query5 query6

Limit to 1000 rows

```

1 SELECT
2   EXTRACT(YEAR FROM order_date) AS order_year,
3   EXTRACT(MONTH FROM order_date) AS order_month,
4   SUM(amount) AS total_revenue,
5   COUNT(DISTINCT order_id) AS total_orders
6 FROM online_sales
7 GROUP BY order_year, order_month
8 ORDER BY order_year, order_month ASC;
9

```

Result Grid

order_year	order_month	total_revenue	total_orders
2023	1	370.00	2
2023	2	455.50	2
2023	3	400.00	2
2023	4	730.00	2
2023	5	565.00	2
2023	6	610.00	2
2023	7	930.00	2

Query Completed

[E]

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

query1 query2 query3 query4 query5 query6

Limit to 1000 rows

```

1 SELECT
2   EXTRACT(YEAR FROM order_date) AS order_year,
3   EXTRACT(MONTH FROM order_date) AS order_month,
4   SUM(amount) AS total_revenue,
5   COUNT(DISTINCT order_id) AS total_orders
6 FROM online_sales
7 WHERE order_date BETWEEN '2023-01-01' AND '2023-04-30'
8 GROUP BY order_year, order_month
9 ORDER BY order_year, order_month ASC;
10

```

Result Grid

order_year	order_month	total_revenue	total_orders
2023	1	370.00	2
2023	2	455.50	2
2023	3	400.00	2
2023	4	730.00	2

Query Completed

[F]