```sql
-- ROW_NUMBER(), RANK(), and DENSE_RANK()
SELECT emp_name, salary,
ROW_NUMBER() OVER (ORDER BY salary DESC ) AS ROW_NUM,
RANK() OVER(ORDER BY salary DESC) AS RNK,
DENSE_RANK() OVER(ORDER BY salary DESC) AS DENSE_RNK
FROM employees;
```

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

emp_name	salary	ROW_NUM	RNK	DENSE_RNK
Deepak	95000	1	1	1
Bhavna	85000	2	2	2
Ekta	80000	3	3	3
Ankit	70000	4	4	4
Chetan	60000	5	5	5

```sql
select * from employees;

-- ROW_NUMBER(), RANK(), and DENSE_RANK()
SELECT emp_name, salary,
ROW_NUMBER() OVER (ORDER BY salary DESC ) AS ROW_NUM,
RANK() OVER(ORDER BY salary DESC) AS RNK,
DENSE_RANK() OVER(ORDER BY salary DESC) AS DENSE_RNK
FROM employees;
```

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

emp_name	salary	ROW_NUM	RNK	DENSE_RNK
Deepak	95000	1	1	1
Bhavna	85000	2	2	2
Ekta	80000	3	3	3
Ankit	70000	4	4	4
Chetan	60000	5	5	5

```sql
SELECT * FROM products_with_duplicates;

SELECT
    product_name,
    price,
    ROW_NUMBER() OVER (ORDER BY price DESC) AS row_num,
    RANK() OVER (ORDER BY price DESC) AS rank_num,
    DENSE_RANK() OVER (ORDER BY price DESC) AS dense_rank_num
FROM products_with_duplicates;
```

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

product_name	price	row_num	rank_num	dense_rank_num
Smart Watch	5000	1	1	1
Smart Watch	5000	2	1	1
Bluetooth Speaker	3000	3	3	2
Wireless Earbuds	3000	4	3	2
Bluetooth Speaker	3000	5	3	2
Wireless Earbuds	3000	6	3	2
Power Bank	2000	7	7	3
Fitness Band	2000	8	7	3
Power Bank	2000	9	7	3
Fitness Band	2000	10	7	3
Tablet Case	1500	11	11	4

```sql
SELECT * FROM sales;


-- Running Total:
SELECT sale_date, amount,
SUM(amount) OVER(
ORDER BY sale_date
ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS RUNNING_TOTAL
FROM sales;
```

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

sale_date	amount	RUNNING_TOTAL
2024-01-01	100	100
2024-01-02	150	250
2024-01-03	120	370
2024-01-04	200	570
2024-01-05	130	700

```sql
-- Moving Average (1 before, current, 1 after):
SELECT
  sale_date,
  amount,
  AVG(amount) OVER (
    ORDER BY sale_date
    ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
  ) AS moving_avg
FROM sales;
```

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

sale_date	amount	moving_avg
2024-01-01	100	125.0000
2024-01-02	150	123.3333
2024-01-03	120	156.6667
2024-01-04	200	150.0000
2024-01-05	130	165.0000