| Experiment No.6 |
| --- |
| Implement SET operators and Datetime functions. |
| Date of Performance:21/02/25 |
| Date of Submission:07/03/25 |

**Aim**: **Implement SET operators and Datetime functions**

**Objective:** SET operators in SQL are used to combine results from two queries, such as UNION, INTERSECT, and MINUS, while Datetime functions are used to manipulate and extract parts of date and time values, like NOW(), DATEADD(), and DATEDIFF()

**Theory:**

**SET OPERATORS:**

**1. UNION / UNION ALL:**

- UNION: Returns result from both queries after eliminating duplications.

e.g.:
```
SELECT employee_id, job_id
FROM employees
UNION
SELECT employee_id, job_id
FROM job_history;
```

- UNION ALL: returns results from both queries, including all duplications.

e.g.:
```
SELECT employee_id, job_id, department_id
FROM employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

**2. INTERSECT:**

e.g.:
```
SELECT employee_id, job_id
FROM employees
INTERSECT
SELECT employee_id, job_id
FROM job_history;
```

**3. MINUS:**

e.g.:
```
SELECT employee_id, job_id
FROM employees
MINUS
SELECT employee_id, job_id
FROM job_history;
```

**Datetime functions:**

**1. CURDATE()**
Returns the current date (without time).
Example:
SELECT CURDATE();

**2. CURTIME()**
Returns the current time (without the date).
SELECT CURTIME();

**3. NOW()**
Returns the current date and time.
SELECT NOW();

**4. DATE()**
Extracts the date part of a datetime value (removes the time).
Example:
SELECT DATE(NOW());

**5. TIME()**
Extracts the time part of a datetime value (removes the date).
Example:
SELECT TIME(NOW());

**6. YEAR()**
Extracts the year from a date or datetime value.
Example:
SELECT YEAR(NOW());

**7. MONTH()**
Extracts the month from a date or datetime value.
Example:
SELECT MONTH(NOW());

**8. DAY()**
Extracts the day of the month from a date or datetime value.
Example:
SELECT DAY(NOW());

**9. DATE_ADD()**
Adds a specified time interval to a date or datetime.
Example:
SELECT DATE_ADD(NOW(), INTERVAL 5 DAY);

**10. DATE_SUB()**
Subtracts a specified time interval from a date or datetime.
Example:
SELECT DATE_SUB(NOW(), INTERVAL 7 DAY);

**11. DATEDIFF()**
Returns the difference in days between two dates.
Example:
SELECT DATEDIFF('2025-02-01', '2025-01-27');

**12. TIMEDIFF()**
Returns the difference in time between two time values.
Example:
SELECT TIMEDIFF('15:00:00', '14:30:00');

**13. STR_TO_DATE()**
Converts a string into a date, based on a specified format.
Example:
SELECT STR_TO_DATE('2025-01-27', '%Y-%m-%d');

**14. DATE_FORMAT()**
Formats a date or datetime value according to a specified format.
Example:
SELECT DATE_FORMAT(NOW(), '%Y-%m-%d %H:%i:%s');

**15. UNIX_TIMESTAMP()**
Returns the current date and time as a Unix timestamp (seconds since 1970-01-01).
Example:
SELECT UNIX_TIMESTAMP();

**16. FROM_UNIXTIME()**
Converts a Unix timestamp to a datetime value.
Example:
SELECT FROM_UNIXTIME(1706359800);

**Implementation:**

```
CREATE DATABASE set_ops;
USE set_ops;

CREATE TABLE Student (
    student_id INT PRIMARY KEY,
    student_name VARCHAR(50),
    department_id INT,
    admission_date DATE
);

CREATE TABLE Department (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);

INSERT INTO Student (student_id, student_name, department_id, admission_date) VALUES
(1, 'Alice', 101, '2023-09-10'),
(2, 'Bob', 102, '2022-08-15'),
(3, 'Charlie', 103, '2021-07-20'),
(4, 'David', 101, '2023-06-05'),
(5, 'Eve', 104, '2024-01-15'),
(6, 'Frank', 105, CURDATE());

INSERT INTO Department (department_id, department_name) VALUES
(101, 'Computer Science'),
(102, 'Mechanical Engineering'),
(104, 'Electrical Engineering'),
(105, 'Civil Engineering');
```

```
SELECT department_id FROM Student
UNION
SELECT department_id FROM Department;

SELECT department_id FROM Student
UNION ALL
SELECT department_id FROM Department;

SELECT s.department_id
FROM Student AS s
WHERE s.department_id IN (
   SELECT d.department_id
   FROM Department AS d
);

SELECT department_id
FROM Student
WHERE department_id NOT IN (
   SELECT department_id
   FROM Department
);

SELECT CURRENT_DATE;

SELECT * FROM Student
WHERE admission_date >= NOW() - INTERVAL 1 YEAR;

SELECT student_name, DATE_FORMAT(admission_date, '%Y-%m-%d')
FROM Student;

SELECT student_name, EXTRACT(YEAR FROM admission_date) AS admission_year,
```

EXTRACT(MONTH FROM admission_date) AS admission_month
FROM Student;

**Output:**

1. SELECT department_id FROM Student
   UNION
   SELECT department_id FROM Department;

| department_id |
|---|
| 101 |
| 102 |
| 103 |
| 104 |
| 105 |

2. SELECT department_id FROM Student
UNION ALL
SELECT department_id FROM Department;

| department_id |
|---|
| 101 |
| 102 |
| 103 |
| 101 |
| 104 |
| 105 |
| 101 |
| 102 |
| 104 |
| 105 |

3. SELECT s.department_id

```
FROM Student AS s
WHERE s.department_id IN (
    SELECT d.department_id
    FROM Department AS d
);
```

| department_id |
|---|
| 101 |
| 102 |
| 101 |
| 104 |
| 105 |

4. SELECT department_id

```
FROM Student
WHERE department_id NOT IN (
    SELECT department_id
    FROM Department
);
```

| department_id |
|---|
| 103 |

5. SELECT CURRENT_DATE;

| CURRENT_DATE |
|---|
| 2025-03-27 |

6. SELECT * FROM Student

WHERE admission_date >= NOW() - INTERVAL 1 YEAR;.

| | student_id | student_name | department_id | admission_date |
|---|---|---|---|---|
| ▶ | 6 | Frank | 105 | 2025-03-27 |
| * | NULL | NULL | NULL | NULL |

7. SELECT student_name, DATE_FORMAT(admission_date, '%Y-%m-%d') FROM Student;

| | student_name | DATE_FORMAT(admission_date, '%Y-%m-%d') |
|---|---|---|
| ▶ | Alice | 2023-09-10 |
| | Bob | 2022-08-15 |
| | Charlie | 2021-07-20 |
| | David | 2023-06-05 |
| | Eve | 2024-01-15 |
| | Frank | 2025-03-27 |

8. SELECT student_name, EXTRACT(YEAR FROM admission_date) AS admission_year,

EXTRACT(MONTH FROM admission_date) AS admission_month

FROM Student;

| | student_name | admission_year | admission_month |
|---|---|---|---|
| ▶ | Alice | 2023 | 9 |
| | Bob | 2022 | 8 |
| | Charlie | 2021 | 7 |
| | David | 2023 | 6 |
| | Eve | 2024 | 1 |
| | Frank | 2025 | 3 |

**Conclusion:** The experiment successfully demonstrated the implementation of SQL SET operators and Datetime functions, showcasing their ability to manipulate and combine query results as well as extract and format date and time values. These functionalities enhance data retrieval and analysis capabilities, ensuring efficient handling of relational databases.