

Database Management System

EXPERIMENT 10 USING THE SET OPERATIONS

NAME: Madhan RV

ROLL: 241801142

1. Display department IDs that don't contain job ID ST_CLERK

```
SELECT department_id  
FROM departments  
MINUS  
SELECT department_id  
FROM employees  
WHERE job_id = 'ST_CLERK';
```

Expected Output:

DEPARTMENT_ID

10
20
40
60
70
80
90
100

110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270

2. Display countries with no departments

```
SELECT country_id, country_name
FROM countries
MINUS
SELECT c.country_id, c.country_name
FROM countries c
JOIN locations l ON c.country_id = l.country_id
JOIN departments d ON l.location_id = d.location_id;
```

Expected Output:

CO COUNTRY_NAME

-- -----

AR Argentina

AU Australia
BE Belgium
BR Brazil
CH Switzerland
CN China
DK Denmark
EG Egypt
FR France
IL Israel
IN India
IT Italy
JP Japan
KW Kuwait
ML Malaysia
MX Mexico
NG Nigeria
NL Netherlands
SG Singapore
UK United Kingdom
US United States of America
ZM Zambia
ZW Zimbabwe

3. Display jobs for departments 10, 50, and 20 in that order

```
SELECT job_id, department_id
FROM employees
WHERE department_id = 10
UNION ALL
SELECT job_id, department_id
FROM employees
WHERE department_id = 50
UNION ALL
```

```
SELECT job_id, department_id  
FROM employees  
WHERE department_id = 20;
```

Expected Output:

JOB_ID	DEPARTMENT_ID
AD_ASST	10
ST_CLERK	50
ST_MAN	50
MK_MAN	20
MK_REP	20

4. Display employees with same job as when hired

```
SELECT employee_id, job_id  
FROM employees  
INTERSECT  
SELECT employee_id, job_id  
FROM job_history;
```

Expected Output:

EMPLOYEE_ID	JOB_ID
101	AC_ACCOUNT
176	SA REP

5. Display all departments and employees (full outer join simulation)

```
SELECT e.last_name, e.department_id
FROM employees e
UNION
SELECT '----No Employee----', d.department_id
FROM departments d
WHERE d.department_id NOT IN (
    SELECT DISTINCT department_id
    FROM employees
    WHERE department_id IS NOT NULL
)
ORDER BY department_id;
```

Alternative approach:

```
SELECT e.last_name AS "Employee/Department",
       e.department_id AS "Dept ID"
  FROM employees e
UNION
SELECT d.department_name,
       d.department_id
  FROM departments d
 ORDER BY "Dept ID";
```

Expected Output:

Employee/Department	Dept ID
-----	-----
Whalen	10
-----	-----

Marketing	20
Hartstein	20
Fay	20

Purchasing	30
Raphaely	30
Khoo	30
Baida	30
Tobias	30
Himuro	30
Colmenares	30

Human Resources	40

Shipping	50
Weiss	50
Fripp	50
Kaufling	50
Vollman	50
Mourgos	50
... (continues for all departments)	

Additional Set Operator Examples:

UNION - Remove duplicates

```

SELECT job_id FROM employees
WHERE department_id IN (10, 20)
UNION
SELECT job_id FROM employees
WHERE department_id IN (20, 50);

```

Expected Output:

JOB_ID

AD_ASST
MK_MAN
MK_REP
ST_CLERK
ST_MAN

UNION ALL - Keep duplicates

```
SELECT job_id FROM employees
WHERE department_id IN (10, 20)
UNION ALL
SELECT job_id FROM employees
WHERE department_id IN (20, 50);
```

Expected Output:

JOB_ID

AD_ASST
MK_MAN
MK_REP
MK_MAN
MK_REP
ST_CLERK
ST_CLERK
ST_CLERK
ST_CLERK
ST_CLERK
ST_MAN

INTERSECT - Common elements

```
SELECT department_id FROM employees  
WHERE job_id = 'SA_REP'  
INTERSECT  
SELECT department_id FROM employees  
WHERE salary > 10000;
```

Expected Output:

DEPARTMENT_ID

80

Key Set Operator Concepts:

1. **UNION** - Combines results, removes duplicates
2. **UNION ALL** - Combines results, keeps duplicates
3. **INTERSECT** - Returns common rows from both queries
4. **MINUS** - Returns rows from first query not in second query
5. **All queries must have same number and type of columns**
6. **ORDER BY can only be used at the end**

.