



IBM Information Management software

UNION




Unit Objectives

After completing this unit, you should be able to:

- Produce a single result table containing data from more than one query
- State the UNION rules
- State the difference between UNION and UNION ALL

Multiple Queries - Multiple Reports



QUERY 1

```
SELECT some_columns  
FROM some_table(s)  
WHERE some_conditions
```

DATA FROM QUERY 1

QUERY 2

```
SELECT some_columns  
FROM some_table(s)  
WHERE some_conditions
```

DATA FROM QUERY 2

UNIONing Queries Together - Single Report

-- QUERY 1

```
SELECT some_columns  
FROM some_table(s)  
WHERE some_conditions
```

UNION ALL

-- QUERY 2

```
SELECT some_columns  
FROM some_table(s)  
WHERE some_conditions
```

DATA FROM QUERY 1

DATA FROM QUERY 2

Union Rule Number 1



```
SELECT COL_6, COL_3, COL_8, COL_14  
FROM TEST_TAB_A  
WHERE COL_7 = 'Y'
```

UNION ALL

```
SELECT COL_D, COL_Y, COL_A, COL_P  
FROM TEST_TAB_B  
WHERE COL_R < 7
```

Every query in the stack must return
the same number of columns

Union Rule Number 2

```
SELECT COL_6, COL_3, COL_8, COL_14
FROM TEST_TAB_A
WHERE COL_7 = 'Y'

UNION ALL

SELECT COL_D, COL_Y, COL_A, COL_P
FROM TEST_TAB_B
WHERE COL_R < 7
```

The diagram illustrates the column mapping for a UNION ALL operation. It shows two SQL queries. The first query selects columns COL_6, COL_3, COL_8, and COL_14 from TEST_TAB_A where COL_7 = 'Y'. The second query selects columns COL_D, COL_Y, COL_A, and COL_P from TEST_TAB_B where COL_R < 7. Colored arrows connect the columns of the same index between the two queries: a red arrow from COL_6 to COL_D, a green arrow from COL_3 to COL_Y, another red arrow from COL_8 to COL_A, and an orange arrow from COL_14 to COL_P.

The data types of the nth column of each query in the stack must be compatible

Union Rule Number 3

```
SELECT COL_6, COL_3, COL_8 AS C3  
FROM TEST_TAB_A  
WHERE COL_7 = 'Y'
```

UNION ALL

```
SELECT COL_D, COL_Y, COL_A AS C3  
FROM TEST_TAB_B  
WHERE COL_R < 7  
ORDER BY C3 DESC, 2
```

The ORDER BY clause must be the last clause on the last query in the stack

UNION ALL - Example 1

For workdepts C01 and A00 display workdept, last name concatenated to first name and in a GENDER column print MALE or FEMALE where appropriate



```
SELECT WORKDEPT, LASTNAME || ', ' ||  
       FIRSTNAME AS NAME,  
       'MALE' AS GENDER  
FROM EMPLOYEE  
WHERE SEX = 'M'  
       AND WORKDEPT IN ('A00', 'C01')
```

UNION ALL

```
SELECT WORKDEPT, LASTNAME || ', ' ||  
       FIRSTNAME AS NAME,  
       'FEMALE' AS GENDER  
FROM EMPLOYEE  
WHERE SEX = 'F'  
       AND WORKDEPT IN ('A00', 'C01')  
ORDER BY WORKDEPT
```

<u>WORKDEPT</u>	<u>NAME</u>	<u>GENDER</u>
A00	HAAS, CHRISTINE	FEMALE
A00	LUCCHESSI, VINCENZO	MALE
A00	O'CONNELL, SEAN	MALE
C01	KWAN, SALLY	FEMALE
C01	QUINTANA, DOLORES	FEMALE
C01	NICHOLLS, HEATHER	FEMALE

UNION ALL - Example 2

As sketched out below, use two lines per department. On line one print manager's information, on line two print department information.



```
SELECT  MGRNO , 'Dept.:',
        DEPTNAME AS NAME
FROM    DEPARTMENT
WHERE   MGRNO IS NOT NULL
UNION ALL
SELECT  MGRNO, 'Mgr.:',
        LASTNAME AS NAME
FROM    DEPARTMENT D, EMPLOYEE E
WHERE   D.MGRNO = E.EMPNO
ORDER BY 1,2 DESC
```



MGRNO		NAME
000010	Mgr.:	HAAS
000010	Dept.:	SPIFFY COMPUTER SERVICE DIV.
000020	Mgr.:	THOMPSON
000020	Dept.:	PLANNING
000030	Mgr.:	KWAN
000030	Dept.:	INFORMATION CENTER
000050	Mgr.:	GEYER
000050	Dept.:	SUPPORT SERVICES

UNION ALL - Example 3

```
SELECT  LASTNAME, EDLEVEL
FROM    EMPLOYEE
WHERE   JOB = 'ANALYST'
```



LASTNAME	EDLEVEL
QUINTANA	16
NICHOLLS	18

```
SELECT  LASTNAME, EDLEVEL
FROM    EMPLOYEE
WHERE   EDLEVEL = 18
```



LASTNAME	EDLEVEL
HAAS	18
THOMPSON	18
NICHOLLS	18
LUTZ	18

```
SELECT  LASTNAME, EDLEVEL
FROM    EMPLOYEE
WHERE   JOB = 'ANALYST'
```

UNION ALL

```
SELECT  LASTNAME, EDLEVEL
FROM    EMPLOYEE
WHERE   EDLEVEL = 18
```



LASTNAME	EDLEVEL
QUINTANA	16
NICHOLLS	18
HAAS	18
THOMPSON	18
NICHOLLS	18
LUTZ	18

UNION

SELECT	LASTNAME, EDLEVEL	➔	LASTNAME	EDLEVEL
FROM	EMPLOYEE		NICHOLLS	18
WHERE	JOB = 'ANALYST'		QUINTANA	16

SELECT	LASTNAME, EDLEVEL	➔	LASTNAME	EDLEVEL
FROM	EMPLOYEE		HAAS	18
WHERE	EDLEVEL = 18		LUTZ	18
			NICHOLLS	18
			THOMPSON	18

SELECT	LASTNAME, EDLEVEL	➔	LASTNAME	EDLEVEL
FROM	EMPLOYEE		HAAS	18
WHERE	JOB = 'ANALYST'		LUTZ	18
UNION			NICHOLLS	18
SELECT	LASTNAME, EDLEVEL		QUINTANA	16
FROM	EMPLOYEE	THOMPSON	18	
WHERE	EDLEVEL = 18			


UNION - Generation of Fitting Result Rows


```
SELECT EMPNO, SUBSTR(FIRSTNME, 1, 1) || '.' || MIDINIT,  
LASTNAME, SALARY AS INCOME, 1 AS SORT  
FROM EMPLOYEE  
UNION ALL  
SELECT EMPNO, '',  
'', BONUS, 2 AS SORT  
FROM EMPLOYEE  
UNION ALL  
SELECT EMPNO, '',  
'', COMM, 3 AS SORT  
FROM EMPLOYEE  
UNION ALL  
SELECT EMPNO, '',  
'SUM: ', SALARY + BONUS + COMM,  
4 AS SORT  
FROM EMPLOYEE  
ORDER BY EMPNO, SORT
```



EMPNO	LASTNAME	INCOME	SORT
000010	C.I HAAS	52750.00	1
000010		1000.00	2
000010		4220.00	3
000010	SUM:	57970.00	4
000020	M.L THOMPSON	41250.00	1
000020		800.00	2
000020		3300.00	3
000020	SUM:	45350.00	4

EXCEPT and INTERSECT

SELECT	LASTNAME, EDLEVEL		LASTNAME	EDLEVEL
FROM	EMPLOYEE			
WHERE	JOB = 'ANALYST'			
EXCEPT			QUINTANA	16
SELECT	LASTNAME, EDLEVEL			
FROM	EMPLOYEE			
WHERE	EDLEVEL = 18			

SELECT	LASTNAME, EDLEVEL		LASTNAME	EDLEVEL
FROM	EMPLOYEE			
WHERE	JOB = 'ANALYST'			
INTERSECT			NICHOLLS	18
SELECT	LASTNAME, EDLEVEL			
FROM	EMPLOYEE			
WHERE	EDLEVEL = 18			

Checkpoint



1. True or False? The results of arbitrary SELECTs can be combined by means of UNION and ordered to get a new result.
2. Why or when should you use UNION?
 - a. When all duplicate rows have to be eliminated.
 - b. When SELECTs with an arbitrary number of columns should be combined.
3. Which rules do you have to consider when using UNION?

Checkpoint Solutions



1. False. The SELECTs must follow very strict UNION rules.
2. a
3. Equal number of columns.
Compatible data types.
ORDER BY must be the last clause.
In the ORDER BY clause, numbers must be used when the column being sorted does not have the same name in all SELECTs.

Unit Summary

Having completed this unit, you should be able to:

- Produce a single result table containing data from more than one query
- State the UNION rules
- State the difference between UNION and UNION ALL