

Appendix B

Workshop Problems



Table of Contents

APPENDIX B WORKSHOP PROBLEMS	B-1
Workshop 8—Combining Result Sets	B-3

Workshop 8—Combining Result Sets

Create a TEMP Table

1. ___ **Locate an SQL script.** Open a DB2 Command Window (*Start | All Programs | IBM DB2 | Command Line Tools | Command Window*) and change to the drive and folder where your workshop files are located:

```
C:  
CD \CejCo
```

Find the **TempTabl.txt** file and edit it in Notepad or an equivalent text editor.

2. ___ **Modify the SQL script.** If necessary, uncomment the SET CURRENT SCHEMA statement, providing your schema name. Also, note that a DROP TABLE and COMMIT statement have been commented out. In case there already is a table called TEMP in your database (or in case you need to rerun this script), simply un-comment these two statements. Save your changes, if any.
3. ___ **Execute the SQL script.** Execute the script by running the following command:

```
DB2 -tvf TempTabl.txt
```

Verify that all statements execute successfully.

Note: In all of the following *SELECT* statements, be sure to provide meaningful column names for the result-set columns, as shown in the expected results.

Execute SQL through the Command Center

4. ___ Code and run a query to return all temporary (hourly) employees. Show employee number, last name, start date, hours worked, and hourly rate.

```
EMPNO  LASTNAME          STARTDATE  HRS_WORKED  HRLY_RATE
-----
TEMP01 JONES             12/08/1998      2080      30.00
TEMP02 DOE          08/01/1995       720     150.00
TEMP03 SMITH       02/29/1996       820      35.00

      3 record(s) selected.
```

5. ___ Code and run a query to return all full-time (salaried) employees who work in D-series departments. Show employee number, last name, hire date, salary, and bonus plus commission.

```
EMPNO  LASTNAME          HIREDATE    SALARY      ADD'L
-----
000060 STERN             09/14/1973    32250.00    3080.00
000070 PULASKI          09/30/1980    36170.00    3593.00
000150 ADAMSON         02/12/1972    25280.00    2522.00
000160 PIANKA          10/11/1977    22250.00    2180.00
000170 YOSHIMURA      09/15/1978    24680.00    2474.00
000180 SCOUTTEN        07/07/1973    21340.00    2207.00
000190 WALKER          07/26/1974    20450.00    2036.00
000200 BROWN           03/03/1966    27740.00    2817.00
000210 JONES           04/11/1979    18270.00    1862.00
000220 LUTZ            08/29/1968    29840.00    2987.00
000230 JEFFERSON       11/21/1966    22180.00    2174.00
000240 MARINO          12/05/1979    28760.00    2901.00
000250 SMITH           10/30/1969    19180.00    1934.00
000260 JOHNSON         09/11/1975    17250.00    1680.00
000270 PEREZ           09/30/1980    27380.00    2690.00

     15 record(s) selected.
```

6. ___ Combine the results of queries #4 and #5 into a single result set. Note that gross pay for temporary (hourly) employees is computed differently than gross pay for full-time (salaried) employees. Make sure that the combined result columns have names, and that gross pay is DECIMAL(9,2). Sequence the combined result rows by start date within descending gross pay.

EMPNO	LASTNAME	STARTDATE	GROSS PAY
TEMP02	DOE	08/01/1995	108000.00
TEMP01	JONES	12/08/1998	62400.00
000070	PULASKI	09/30/1980	39763.00
000060	STERN	09/14/1973	35330.00
000220	LUTZ	08/29/1968	32827.00
000240	MARINO	12/05/1979	31661.00
000200	BROWN	03/03/1966	30557.00
000270	PEREZ	09/30/1980	30070.00
TEMP03	SMITH	02/29/1996	28700.00
000150	ADAMSON	02/12/1972	27802.00
000170	YOSHIMURA	09/15/1978	27154.00
000160	PIANKA	10/11/1977	24430.00
000230	JEFFERSON	11/21/1966	24354.00
000180	SCOUTTEN	07/07/1973	23547.00
000190	WALKER	07/26/1974	22486.00
000250	SMITH	10/30/1969	21114.00
000210	JONES	04/11/1979	20132.00
000260	JOHNSON	09/11/1975	18930.00

18 record(s) selected.

Optional

7. ___ Find the total number of all employees, both temporary and full-time, as well as the number of different last names. (Hint: You'll need a nested table expression to do this.)

#EMPS	#NAMES
36	33

1 record(s) selected.

8. ____ Create a view called ALLEMPS defined as the union of all rows in TEMP and EMPLOYEE. For column compatibility, use the following guidelines:
- Columns that are only in the EMPLOYEE table should be selected as null (of the appropriate data type) from the TEMP table, with the exception of MIDINIT, which should be a space for all temporary employees.
 - Instead of salary, bonus, commission, hours worked, or hourly rate, define a single DECIMAL(9,2) column for gross pay, as in your solution to query #6.
 - Refer to Appendix A for the record layout of the EMPLOYEE table, and page 8-8 for the contents of the TEMP table.
 - The syntax of CREATE VIEW is:

```
CREATE VIEW ALLEMPS AS  
    fullselect goes here;
```
 - Comment out the CREATE VIEW part of the syntax and execute the SELECT statement (*fullselect*) as a stand-alone statement until you are satisfied that it is doing what you want. Then un-comment the CREATE VIEW syntax and run it through one last time.
 - Be sure to commit your changes to the database:

```
COMMIT;
```

9. ____ Select the entire contents of the ALLEMPS view to verify a 12-column, 36-row result. Make sure that all columns are named appropriately; that middle initial is space, while phone number, job, education level, and birth date are null for the three temporary employees; and that EDLEVEL in the view result is SMALLINT and not INTEGER. You may need to drop and recreate your view until it is the way you want it to be.

```

EMPNO  FIRSTNME          MIDINIT  LASTNAME          WORKDEPT ...
-----
TEMP01 JOE                JONES    D11
TEMP02 JANE                DOE       D21
TEMP03 BETTY         SMITH    D11
000010 CHRISTINE         I        HAAS       A00
...    ...                ...      ...            ...
000340 JASON            R        GOUNOT    E21
123456 first          m        last      -
...  PHONENO  HIREDATE    JOB          EDLEVEL  SEX  ...
...  -----
-    12/08/1998 -            - M
-    08/01/1995 -            - F
-    02/29/1996 -            - F
3978 01/01/1965 PRES          18 F
...  ...                ...      ...
5698 05/05/1947 FIELDREP    16 M
-    -                -            99 M
...  BIRTHDATE  PAY
...  -----
-    62400.00
-    108000.00
-    28700.00
08/24/1933 57970.00
...  ...                ...
05/17/1926 26247.00
-    -                -
36 record(s) selected.

```

10. ____ Recode your solution to query #7, but use ALLEMPS instead of a nested table expression.

```

#EMPS      #NAMES
-----
36         33
1 record(s) selected.

```

11. ____ The following three SELECT statements return three different results. The first returns detail-level information, the second returns subtotals by department, and the third returns a grand-total row. Combine these three results into a single result. Hint: The result sets must be made compatible—the same number of columns, with positionally compatible data types.

```
SELECT WORKDEPT, EMPNO, SALARY
   FROM EMPLOYEE
   WHERE WORKDEPT IS NOT NULL
   ORDER BY WORKDEPT, EMPNO;
```

```
SELECT WORKDEPT, SUM(SALARY)
   FROM EMPLOYEE
   WHERE WORKDEPT IS NOT NULL
   GROUP BY WORKDEPT
   ORDER BY WORKDEPT;
```

```
SELECT SUM(SALARY)
   FROM EMPLOYEE
   WHERE WORKDEPT IS NOT NULL;
```

Following are the expected combined results. Note the sequence of the result rows as well as the nulls in the subtotal and grand-total rows. Make sure that the SALARY column is defined as DECIMAL(11,2).

WORKDEPT	EMPNO	SALARY
A00	000010	52750.00
A00	000110	46500.00
A00	000120	29250.00
A00	-	128500.00
B01	000020	41250.00
B01	-	41250.00
C01	000030	38250.00
C01	000130	23800.00
C01	000140	28420.00
C01	-	90470.00
D11	000060	32250.00
D11	000150	25280.00
D11	000160	22250.00
D11	000170	24680.00
D11	000180	21340.00
D11	000190	20450.00
D11	000200	27740.00
D11	000210	18270.00
D11	000220	29840.00
D11	-	222100.00
D21	000070	36170.00
D21	000230	22180.00
D21	000240	28760.00

D21	000250	19180.00
D21	000260	17250.00
D21	000270	27380.00
D21	-	150920.00
E01	000050	40175.00
E01	-	40175.00
E11	000090	29750.00
E11	000280	26250.00
E11	000290	15340.00
E11	000300	17750.00
E11	000310	15900.00
E11	-	104990.00
E21	000100	26150.00
E21	000320	19950.00
E21	000330	25370.00
E21	000340	23840.00
E21	-	95310.00
-	-	873715.00

41 record(s) selected.

Appendix C

Workshop Solutions



Table of Contents

APPENDIX C WORKSHOP SOLUTIONS	C-1
Workshop 8—Combining Result Sets	C-3

Workshop 8—Combining Result Sets

Create a TEMP Table

Ask your instructor for help if you have difficulty with steps #1 through #3, which involve mostly following instructions.

Note: In all of the following *SELECT* statements, be sure to provide meaningful column names for the result-set columns, as shown in the expected results.

Execute SQL through the Command Center or SPUFI

4. ___ Code and run a query to return all temporary (hourly) employees. Show employee number, last name, start date, hours worked, and hourly rate.

```
SELECT EMPNO, LASTNAME, STARTDATE, HRS_WORKED, HRLY_RATE
FROM TEMP;
```

5. ___ Code and run a query to return all full-time (salaried) employees who work in D-series departments. Show employee number, last name, hire date, salary, and bonus plus commission.

```
SELECT EMPNO, LASTNAME, HIREDATE, SALARY
       , BONUS + COMM AS "ADD'L"
FROM EMPLOYEE
WHERE WORKDEPT LIKE 'D%';
```

6. ___ Combine the results of queries #4 and #5 into a single result set. Note that gross pay for temporary (hourly) employees is computed differently than gross pay for full-time (salaried) employees. Make sure that the combined result columns have names, and that gross pay is DECIMAL(9,2). Sequence the combined result rows by start date within descending gross pay.

```
SELECT EMPNO, LASTNAME, STARTDATE
       , DECIMAL(HRS_WORKED*HRLY_RATE,9,2) AS "GROSS PAY"
FROM TEMP
UNION ALL
SELECT EMPNO, LASTNAME, HIREDATE AS STARTDATE
       , DECIMAL(SALARY+BONUS+COMM,9,2) AS "GROSS PAY"
FROM EMPLOYEE
WHERE WORKDEPT LIKE 'D%'
ORDER BY "GROSS PAY" DESC, STARTDATE;
```

Optional

7. ____ Find the total number of all employees, both temporary and full-time, as well as the number of different last names. (Hint: You'll need a nested table expression to do this.)

```
SELECT COUNT(*) AS #EMPS
       , COUNT(DISTINCT LASTNAME) AS #NAMES
FROM (SELECT LASTNAME
      FROM TEMP
      UNION ALL
      SELECT LASTNAME
      FROM EMPLOYEE) AS X;
```

8. ____ Create a view called ALLEMPS defined as the union of all rows in TEMP and EMPLOYEE. For column compatibility, use the following guidelines:

- Columns that are only in the EMPLOYEE table should be selected as null (of the appropriate data type) from the TEMP table, with the exception of MIDINIT, which should be a space for all temporary employees.
- Instead of salary, bonus, commission, hours worked, or hourly rate, define a single DECIMAL(9,2) column for gross pay, as in your solution to query #6.

```
CREATE VIEW ALLEMPS AS
SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME
       , WORKDEPT, PHONENO
       , HIREDATE, JOB
       , EDLEVEL, SEX
       , BIRTHDATE
       , DECIMAL(SALARY+BONUS+COMM,9,2) AS PAY
FROM EMPLOYEE
UNION ALL
SELECT EMPNO, FIRSTNME, ' ' AS MIDINIT, LASTNAME
       , WORKDEPT, NULLIF(' ',' ') AS PHONENO
       , STARTDATE AS HIREDATE, NULLIF(' ',' ') AS JOB
       , NULLIF(SMALLINT(1),SMALLINT(1)) AS EDLEVEL, SEX
       , NULLIF(DATE('0001-1-1'),DATE('0001-1-1'))
       AS BIRTHDATE
       , DECIMAL(HRS_WORKED*HRLY_RATE,9,2) AS PAY
FROM TEMP;
```

Be sure to commit your changes to the database:

```
COMMIT;
```

9. ____ Select the entire contents of the ALLEMPS view to verify a 12-column, 36-row result. Make sure that all columns are named appropriately; that middle initial is space, while phone number, job, education level, and birth date are null for the three temporary employees; and that EDLEVEL in the view result is SMALLINT and not INTEGER. You may need to drop and recreate your view until it is the way you want it to be.

```
SELECT *
FROM ALLEMPS;
```

10. ____ Recode your solution to query #7, but use ALLEMPS instead of a nested table expression.

```
SELECT COUNT(*) AS #EMPS
       , COUNT(DISTINCT LASTNAME) AS #NAMES
FROM ALLEMPS;
```

11. ____ The following three SELECT statements return three different results. The first returns detail-level information, the second returns subtotals by department, and the third returns a grand-total row. Combine these three results into a single result. Hint: The result sets must be made compatible—the same number of columns, with positionally compatible data types.

```
SELECT WORKDEPT, EMPNO, SALARY
FROM EMPLOYEE
WHERE WORKDEPT IS NOT NULL
ORDER BY WORKDEPT, EMPNO;
```

```
SELECT WORKDEPT, SUM(SALARY)
FROM EMPLOYEE
WHERE WORKDEPT IS NOT NULL
GROUP BY WORKDEPT
ORDER BY WORKDEPT;
```

```
SELECT SUM(SALARY)
FROM EMPLOYEE
WHERE WORKDEPT IS NOT NULL;
```

Following are the expected combined results. Note the sequence of the result rows as well as the nulls in the subtotal and grand-total rows. Make sure that the SALARY column is defined as DECIMAL(11,2).

WORKDEPT	EMPNO	SALARY
A00	000010	52750.00
A00	000110	46500.00
A00	000120	29250.00
A00	-	128500.00
B01	000020	41250.00
B01	-	41250.00
C01	000030	38250.00
C01	000130	23800.00
C01	000140	28420.00
C01	-	90470.00

```

...          ...          ...
E21          000100          26150.00
E21          000320          19950.00
E21          000330          25370.00
E21          000340          23840.00
E21          -              95310.00
-            -              873715.00

```

41 record(s) selected.

```

SELECT WORKDEPT, EMPNO, SALARY
FROM EMPLOYEE
WHERE WORKDEPT IS NOT NULL
UNION ALL
SELECT WORKDEPT, NULLIF(' ', ' ') AS EMPNO
      , DECIMAL(SUM(SALARY), 11, 2) AS SALARY
FROM EMPLOYEE
WHERE WORKDEPT IS NOT NULL
GROUP BY WORKDEPT
UNION ALL
SELECT NULLIF(' ', ' ') AS WORKDEPT
      , NULLIF(' ', ' ') AS EMPNO
      , DECIMAL(SUM(SALARY), 11, 2) AS SALARY
FROM EMPLOYEE
WHERE WORKDEPT IS NOT NULL
ORDER BY WORKDEPT, EMPNO;

```