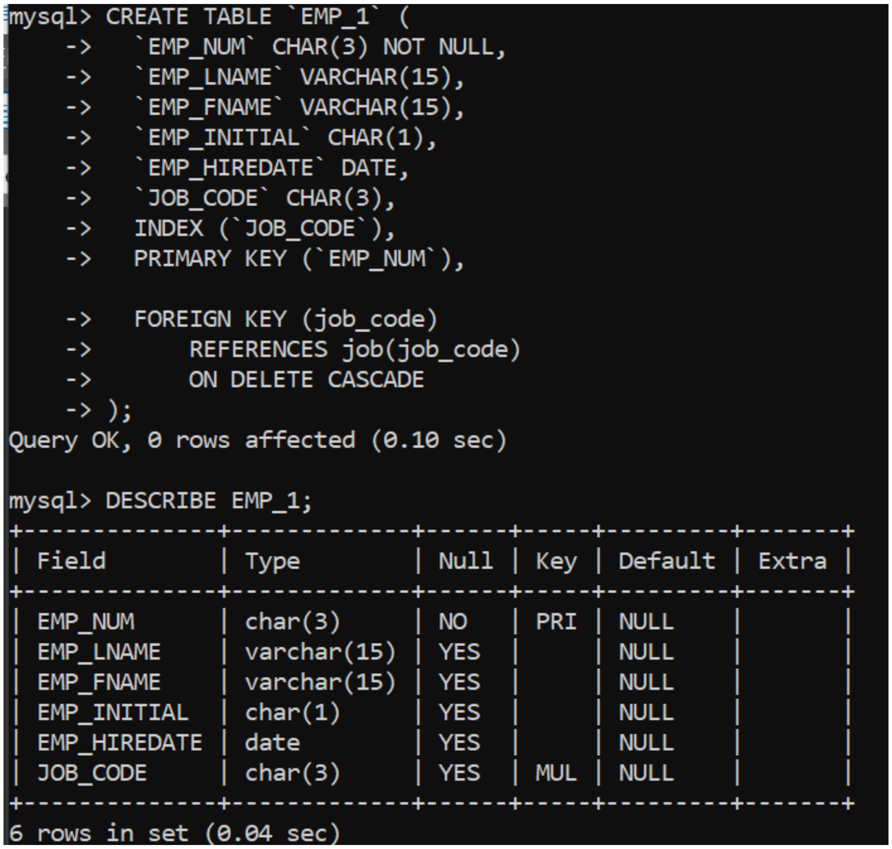
Ch7_Ungraded_Practice_Lee_Brunovsky

1. Write the SQL code that will create the table structure for a table named EMP_1. This table is a subset of the EMPLOYEE table. The basic EMP_1 table structure is summarized in the table below. (Note that the JOB_CODE is the FK to JOB.)

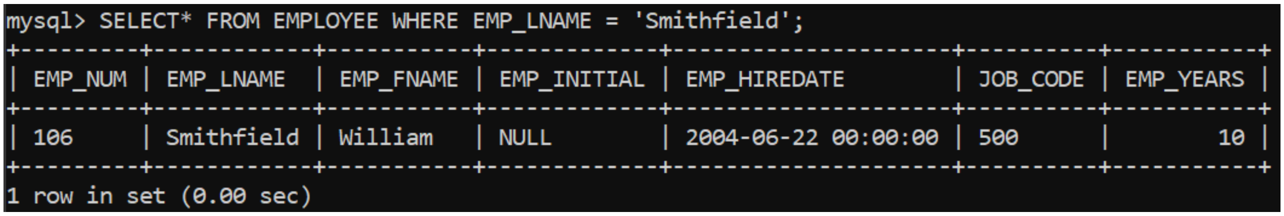
```
CREATE TABLE `EMP_1` (
  `EMP_NUM` CHAR(3) NOT NULL,
  `EMP_LNAME` VARCHAR(15),
  `EMP_FNAME` VARCHAR(15),
  `EMP_INITIAL` CHAR(1),
  `EMP_HIREDATE` DATE,
  `JOB_CODE` CHAR(3),
  INDEX (`JOB_CODE`),
  PRIMARY KEY (`EMP_NUM`),
  FOREIGN KEY (job_code)
      REFERENCES job(job_code)
      ON UPDATE CASCADE
);
```



```
mysql> CREATE TABLE `EMP_1` (
    ->   `EMP_NUM` CHAR(3) NOT NULL,
    ->   `EMP_LNAME` VARCHAR(15),
    ->   `EMP_FNAME` VARCHAR(15),
    ->   `EMP_INITIAL` CHAR(1),
    ->   `EMP_HIREDATE` DATE,
    ->   `JOB_CODE` CHAR(3),
    ->   INDEX (`JOB_CODE`),
    ->   PRIMARY KEY (`EMP_NUM`),

    ->   FOREIGN KEY (job_code)
    ->       REFERENCES job(job_code)
    ->       ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> DESCRIBE EMP_1;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| EMP_NUM      | char(3)     | NO   | PRI | NULL    |       |
| EMP_LNAME    | varchar(15) | YES  |     | NULL    |       |
| EMP_FNAME    | varchar(15) | YES  |     | NULL    |       |
| EMP_INITIAL  | char(1)     | YES  |     | NULL    |       |
| EMP_HIREDATE | date        | YES  |     | NULL    |       |
| JOB_CODE     | char(3)     | YES  | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
6 rows in set (0.04 sec)
```

6. Write the SQL code to delete the row for the person named William Smithfield, who was hired on June 22, 2004, and whose job code classification is 500. (Hint: Use logical operators to include all of the information given in this problem.)

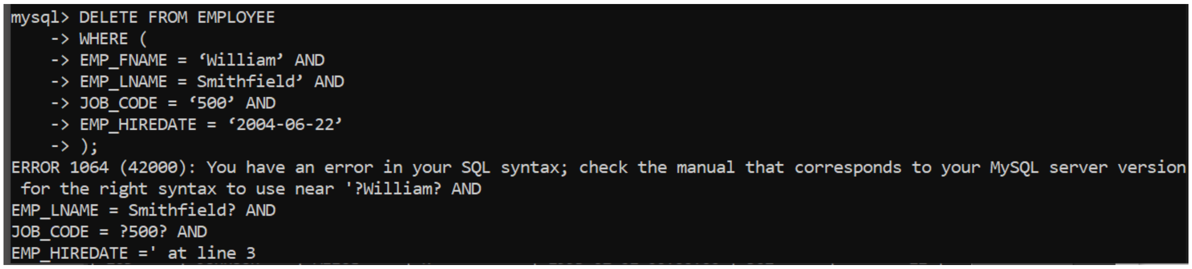
Before



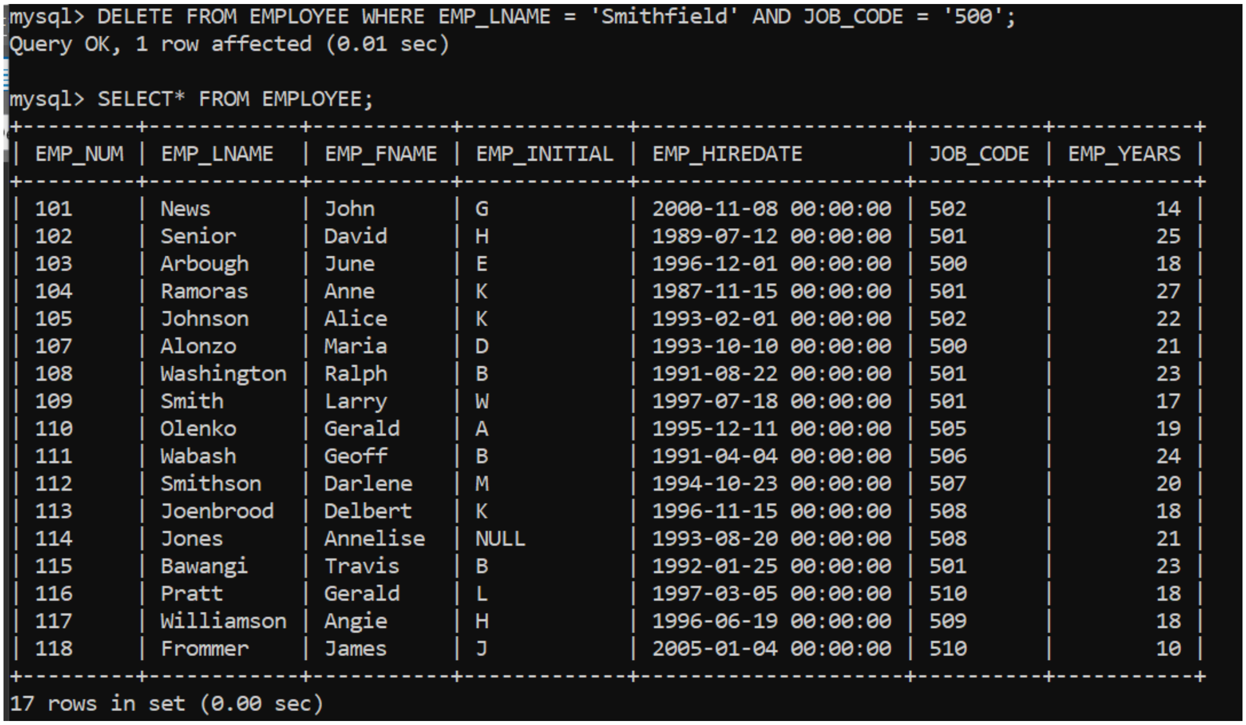
```
mysql> SELECT* FROM EMPLOYEE WHERE EMP_LNAME = 'Smithfield';
+---------+------------+----------+-------------+---------------------+----------+-----------+
| EMP_NUM | EMP_LNAME  | EMP_FNAME | EMP_INITIAL | EMP_HIREDATE        | JOB_CODE | EMP_YEARS |
+---------+------------+----------+-------------+---------------------+----------+-----------+
| 106     | Smithfield | William  | NULL        | 2004-06-22 00:00:00 | 500      |        10 |
+---------+------------+----------+-------------+---------------------+----------+-----------+
1 row in set (0.00 sec)
```

After ***I cannot get the correct results with more than one AND statement. PLEASE EXPLAIN***

DELETE FROM EMPLOYEE
WHERE (
EMP_FNAME = ‘William’ AND
EMP_LNAME = Smithfield’ AND
JOB_CODE = ‘500’ AND
EMP_HIREDATE = ‘2004-06-22’
);



```
mysql> DELETE FROM EMPLOYEE
    -> WHERE (
    -> EMP_FNAME = ‘William’ AND
    -> EMP_LNAME = Smithfield’ AND
    -> JOB_CODE = ‘500’ AND
    -> EMP_HIREDATE = ‘2004-06-22’
    -> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
 for the right syntax to use near '?William? AND
EMP_LNAME = Smithfield? AND
JOB_CODE = ?500? AND
EMP_HIREDATE =' at line 3
```



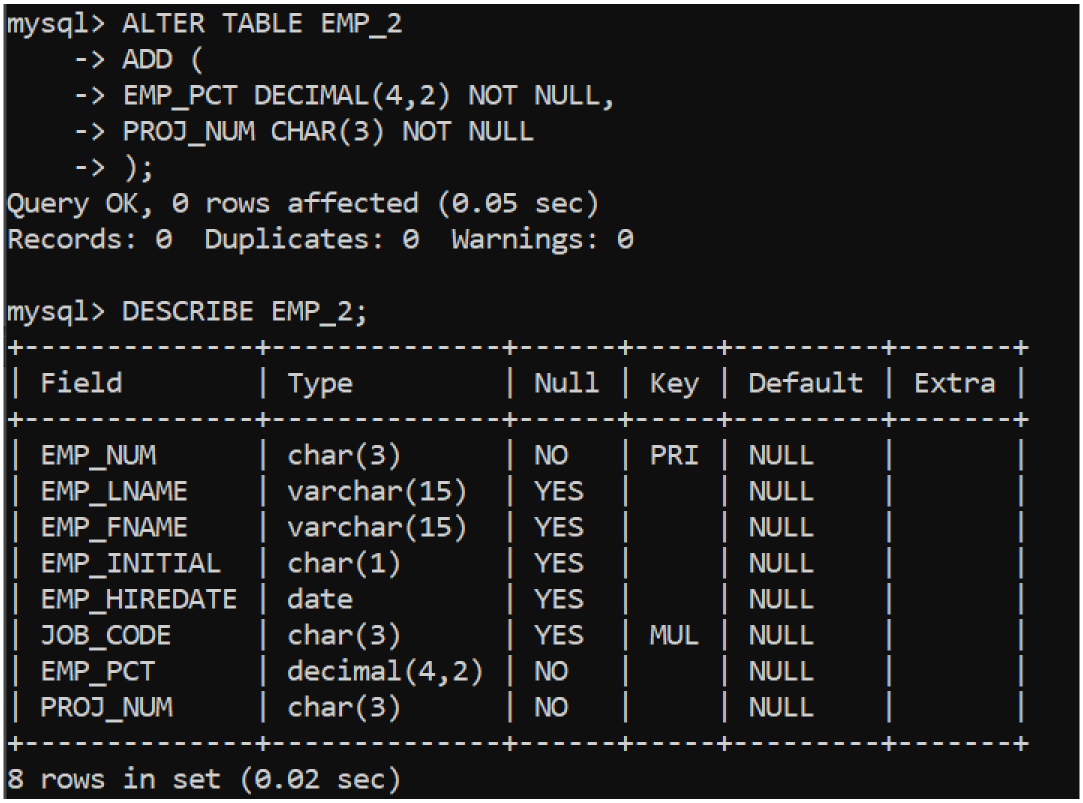
```
mysql> DELETE FROM EMPLOYEE WHERE EMP_LNAME = 'Smithfield' AND JOB_CODE = '500';
Query OK, 1 row affected (0.01 sec)

mysql> SELECT* FROM EMPLOYEE;
+---------+------------+-----------+-------------+---------------------+----------+-----------+
| EMP_NUM | EMP_LNAME  | EMP_FNAME | EMP_INITIAL | EMP_HIREDATE        | JOB_CODE | EMP_YEARS |
+---------+------------+-----------+-------------+---------------------+----------+-----------+
| 101     | News       | John      | G           | 2000-11-08 00:00:00 | 502      |        14 |
| 102     | Senior     | David     | H           | 1989-07-12 00:00:00 | 501      |        25 |
| 103     | Arbough    | June      | E           | 1996-12-01 00:00:00 | 500      |        18 |
| 104     | Ramoras    | Anne      | K           | 1987-11-15 00:00:00 | 501      |        27 |
| 105     | Johnson    | Alice     | K           | 1993-02-01 00:00:00 | 502      |        22 |
| 107     | Alonzo     | Maria     | D           | 1993-10-10 00:00:00 | 500      |        21 |
| 108     | Washington | Ralph     | B           | 1991-08-22 00:00:00 | 501      |        23 |
| 109     | Smith      | Larry     | W           | 1997-07-18 00:00:00 | 501      |        17 |
| 110     | Olenko     | Gerald    | A           | 1995-12-11 00:00:00 | 505      |        19 |
| 111     | Wabash     | Geoff     | B           | 1991-04-04 00:00:00 | 506      |        24 |
| 112     | Smithson   | Darlene   | M           | 1994-10-23 00:00:00 | 507      |        20 |
| 113     | Joenbrood  | Delbert   | K           | 1996-11-15 00:00:00 | 508      |        18 |
| 114     | Jones      | Annelise  | NULL        | 1993-08-20 00:00:00 | 508      |        21 |
| 115     | Bawangi    | Travis    | B           | 1992-01-25 00:00:00 | 501      |        23 |
| 116     | Pratt      | Gerald    | L           | 1997-03-05 00:00:00 | 510      |        18 |
| 117     | Williamson | Angie     | H           | 1996-06-19 00:00:00 | 509      |        18 |
| 118     | Frommer    | James     | J           | 2005-01-04 00:00:00 | 510      |        10 |
+---------+------------+-----------+-------------+---------------------+----------+-----------+
17 rows in set (0.00 sec)
```

8. Write the SQL code to create a copy of EMP_1, naming the copy EMP_2. Then write the SQL code that will add the attributes EMP_PCT and PROJ_NUM to its structure. The EMP_PCT is the bonus percentage to be paid to each employee. The new attribute characteristics are:

EMP_PCTNUMBER(4,2) PROJ_NUMCHAR(3) (Note: If your SQL implementation allows it, you may use DECIMAL(4,2) rather than NUMBER(4,2).)

```
CREATE TABLE EMP_2 LIKE EMP_1;
INSERT INTO EMP_2 SELECT * FROM EMP_1;
```

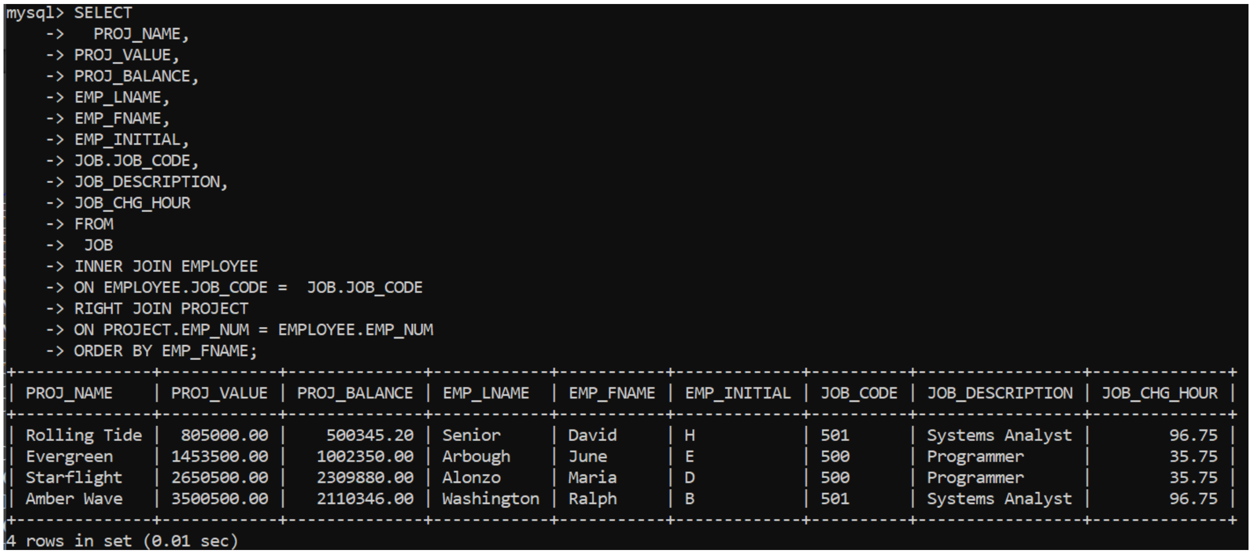
```
ALTER TABLE EMP_2
ADD (
EMP_PCT DECIMAL(4,2) NOT NULL,
PROJ_NUM CHAR(3) NOT NULL
);
```



```
mysql> ALTER TABLE EMP_2
    -> ADD (
    -> EMP_PCT DECIMAL(4,2) NOT NULL,
    -> PROJ_NUM CHAR(3) NOT NULL
    -> );
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE EMP_2;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| EMP_NUM      | char(3)      | NO   | PRI | NULL    |       |
| EMP_LNAME    | varchar(15)  | YES  |     | NULL    |       |
| EMP_FNAME    | varchar(15)  | YES  |     | NULL    |       |
| EMP_INITIAL  | char(1)      | YES  |     | NULL    |       |
| EMP_HIREDATE | date         | YES  |     | NULL    |       |
| JOB_CODE     | char(3)      | YES  | MUL | NULL    |       |
| EMP_PCT      | decimal(4,2) | NO   |     | NULL    |       |
| PROJ_NUM     | char(3)      | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
8 rows in set (0.02 sec)
```

16. Using the EMPLOYEE, JOB, and PROJECT tables in the Ch07_ConstructCo database (see Figure P7.1), write the SQL code that will produce the results shown in Figure P7.16.

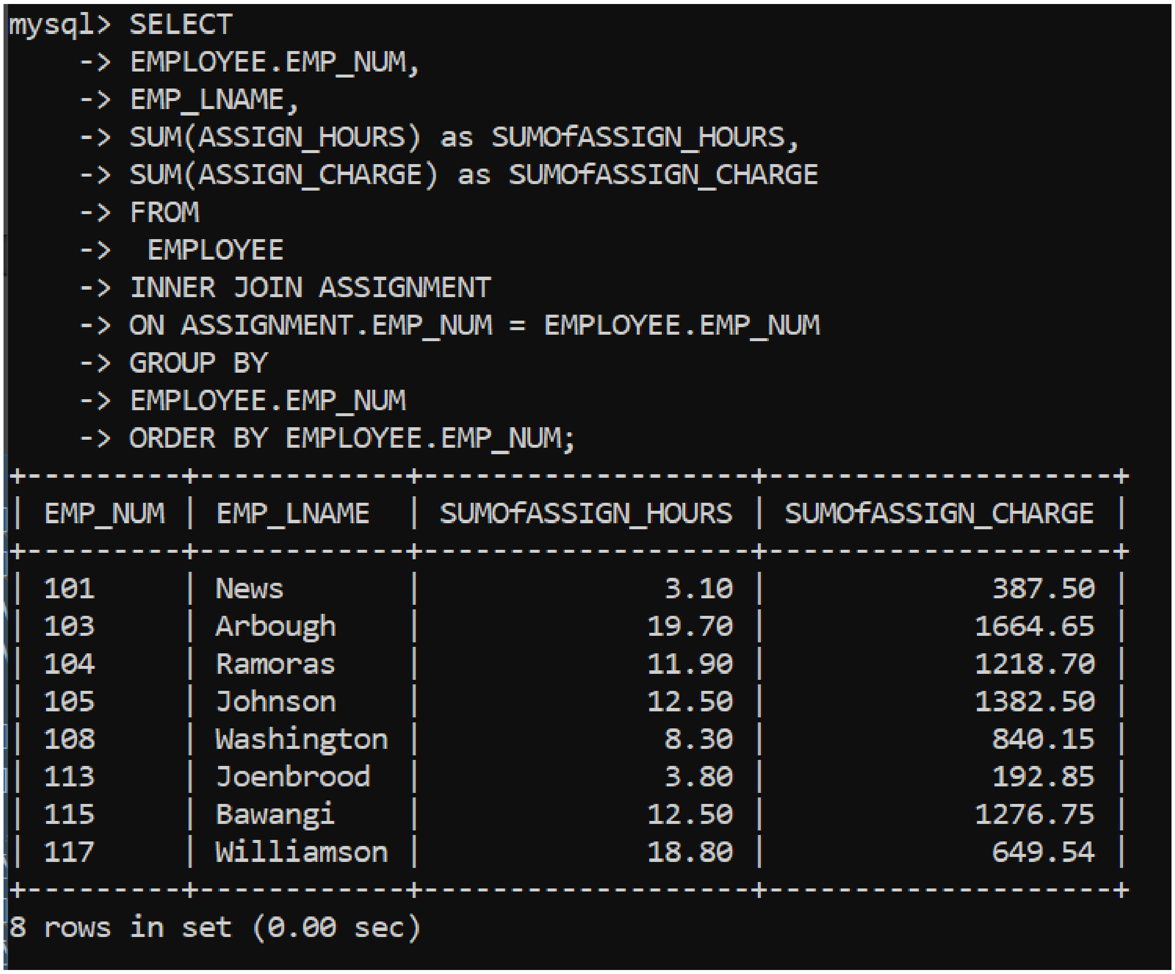
```
SELECT
        PROJ_NAME,
        PROJ_VALUE,
        PROJ_BALANCE,
        EMP_LNAME,
        EMP_FNAME,
        EMP_INITIAL,
        JOB.JOB_CODE,
        JOB_DESCRIPTION,
        JOB_CHG_HOUR
FROM
        JOB
INNER JOIN EMPLOYEE
        ON EMPLOYEE.JOB_CODE =  JOB.JOB_CODE
RIGHT JOIN PROJECT
        ON PROJECT.EMP_NUM = EMPLOYEE.EMP_NUM
ORDER BY EMP_FNAME;
```



```
mysql> SELECT
    ->   PROJ_NAME,
    -> PROJ_VALUE,
    -> PROJ_BALANCE,
    -> EMP_LNAME,
    -> EMP_FNAME,
    -> EMP_INITIAL,
    -> JOB.JOB_CODE,
    -> JOB_DESCRIPTION,
    -> JOB_CHG_HOUR
    -> FROM
    ->  JOB
    -> INNER JOIN EMPLOYEE
    -> ON EMPLOYEE.JOB_CODE =  JOB.JOB_CODE
    -> RIGHT JOIN PROJECT
    -> ON PROJECT.EMP_NUM = EMPLOYEE.EMP_NUM
    -> ORDER BY EMP_FNAME;
+--------------+------------+--------------+------------+-----------+-------------+----------+-----------------+--------------+
| PROJ_NAME    | PROJ_VALUE | PROJ_BALANCE | EMP_LNAME  | EMP_FNAME | EMP_INITIAL | JOB_CODE | JOB_DESCRIPTION | JOB_CHG_HOUR |
+--------------+------------+--------------+------------+-----------+-------------+----------+-----------------+--------------+
| Rolling Tide |  805000.00 |    500345.20 | Senior     | David     | H           | 501      | Systems Analyst |        96.75 |
| Evergreen    | 1453500.00 |   1002350.00 | Arbough    | June      | E           | 500      | Programmer      |        35.75 |
| Starflight   | 2650500.00 |   2309880.00 | Alonzo     | Maria     | D           | 500      | Programmer      |        35.75 |
| Amber Wave   | 3500500.00 |   2110346.00 | Washington | Ralph     | B           | 501      | Systems Analyst |        96.75 |
+--------------+------------+--------------+------------+-----------+-------------+----------+-----------------+--------------+
4 rows in set (0.01 sec)
```

22. Using the data in the ASSIGNMENT table, write the SQL code that will yield the total number of hours worked for each employee and the total charges stemming from those hours worked. The results of running that query are shown in Figure P7.22.

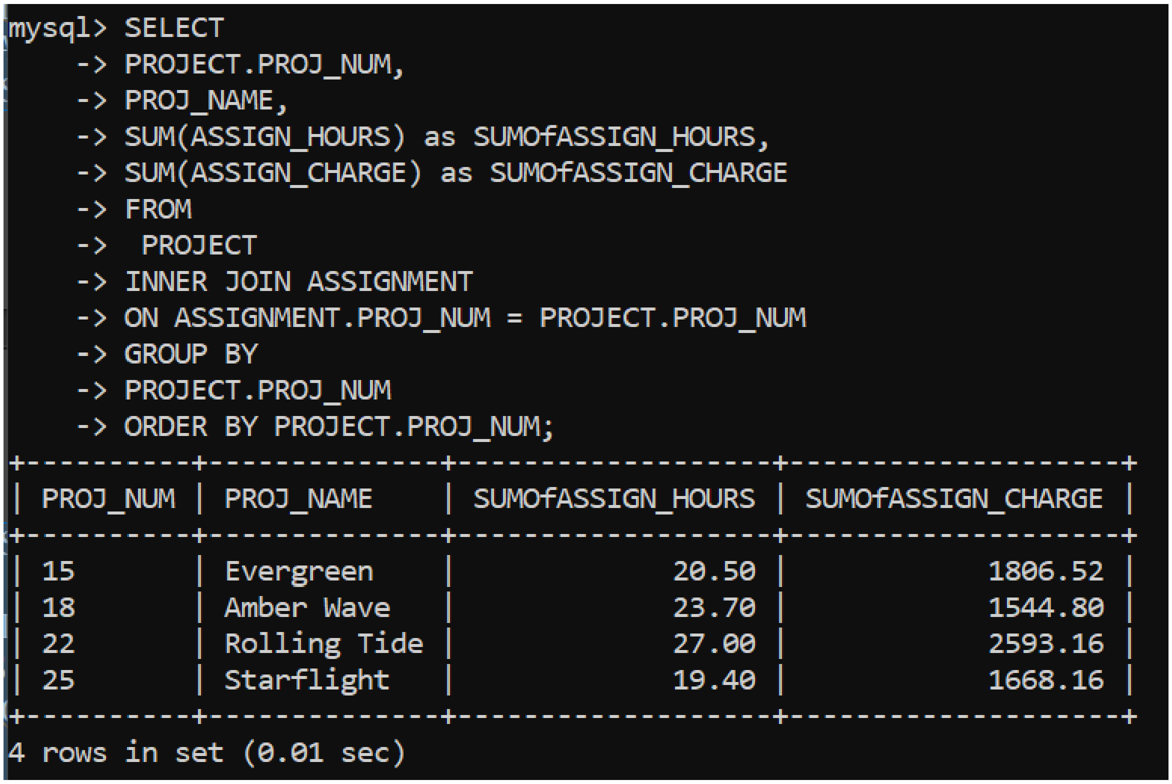
```
SELECT
	EMPLOYEE.EMP_NUM,
	EMP_LNAME,
	SUM(ASSIGN_HOURS) as SUMOfASSIGN_HOURS,
	SUM(ASSIGN_CHARGE) as SUMOfASSIGN_CHARGE
FROM
	EMPLOYEE
INNER JOIN ASSIGNMENT
	ON ASSIGNMENT.EMP_NUM = EMPLOYEE.EMP_NUM
GROUP BY
	EMPLOYEE.EMP_NUM
ORDER BY EMPLOYEE.EMP_NUM;
```



```
mysql> SELECT
    -> EMPLOYEE.EMP_NUM,
    -> EMP_LNAME,
    -> SUM(ASSIGN_HOURS) as SUMOfASSIGN_HOURS,
    -> SUM(ASSIGN_CHARGE) as SUMOfASSIGN_CHARGE
    -> FROM
    ->  EMPLOYEE
    -> INNER JOIN ASSIGNMENT
    -> ON ASSIGNMENT.EMP_NUM = EMPLOYEE.EMP_NUM
    -> GROUP BY
    -> EMPLOYEE.EMP_NUM
    -> ORDER BY EMPLOYEE.EMP_NUM;
+---------+------------+-------------------+--------------------+
| EMP_NUM | EMP_LNAME  | SUMOfASSIGN_HOURS | SUMOfASSIGN_CHARGE |
+---------+------------+-------------------+--------------------+
| 101     | News       |              3.10 |             387.50 |
| 103     | Arbough    |             19.70 |            1664.65 |
| 104     | Ramoras    |             11.90 |            1218.70 |
| 105     | Johnson    |             12.50 |            1382.50 |
| 108     | Washington |              8.30 |             840.15 |
| 113     | Joenbrood  |              3.80 |             192.85 |
| 115     | Bawangi    |             12.50 |            1276.75 |
| 117     | Williamson |             18.80 |             649.54 |
+---------+------------+-------------------+--------------------+
8 rows in set (0.00 sec)
```

24. Write the SQL code to generate the total hours worked and the total charges made to all projects. The results should be the same as those shown in Figure P7.24. (Hint: This is a nested query. If you use Microsoft Access, you can generate the result by using the query output shown in Figure P7.23 as the basis for this query.)

```
SELECT
        PROJECT.PROJ_NUM,
        PROJ_NAME,
        SUM(ASSIGN_HOURS) as SUMOfASSIGN_HOURS,
        SUM(ASSIGN_CHARGE) as SUMOfASSIGN_CHARGE
FROM
        PROJECT
INNER JOIN ASSIGNMENT
        ON ASSIGNMENT.PROJ_NUM = PROJECT.PROJ_NUM
GROUP BY
        PROJECT.PROJ_NUM
ORDER BY PROJECT.PROJ_NUM;
```



```
mysql> SELECT
    -> PROJECT.PROJ_NUM,
    -> PROJ_NAME,
    -> SUM(ASSIGN_HOURS) as SUMOfASSIGN_HOURS,
    -> SUM(ASSIGN_CHARGE) as SUMOfASSIGN_CHARGE
    -> FROM
    ->  PROJECT
    -> INNER JOIN ASSIGNMENT
    -> ON ASSIGNMENT.PROJ_NUM = PROJECT.PROJ_NUM
    -> GROUP BY
    -> PROJECT.PROJ_NUM
    -> ORDER BY PROJECT.PROJ_NUM;
+----------+--------------+-------------------+--------------------+
| PROJ_NUM | PROJ_NAME    | SUMOfASSIGN_HOURS | SUMOfASSIGN_CHARGE |
+----------+--------------+-------------------+--------------------+
| 15       | Evergreen    |             20.50 |            1806.52 |
| 18       | Amber Wave   |             23.70 |            1544.80 |
| 22       | Rolling Tide |             27.00 |            2593.16 |
| 25       | Starflight   |             19.40 |            1668.16 |
+----------+--------------+-------------------+--------------------+
4 rows in set (0.01 sec)
```