

SQL Exercises

Given the Ch06_Review database's structure and contents shown in Figure Q6.1 (available at the last page), use SQL commands to answer questions 1-25.

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 Table Q6.1. (Note that the JOB_CODE is the FK to JOB.)

Table Q6.1 The EMP_1 Table Structure

ATTRIBUTE (FIELD) NAME	DATA DECLARATION
EMP_NUM	CHAR(3)
EMP_LNAME	VARCHAR(15)
EMP_FNAME	VARCHAR(15)
EMP_INITIAL	CHAR(1)
EMP_HIREDATE	DATE
JOB_CODE	CHAR(3)

```
CREATE TABLE EMP_1 (  
EMP_NUM          CHAR(3)          PRIMARY KEY,  
EMP_LNAME        VARCHAR(15)       NOT NULL,  
EMP_FNAME        VARCHAR(15)       NOT NULL,  
EMP_INITIAL      CHAR(1),  
EMP_HIREDATE     DATE,  
JOB_CODE         CHAR(3),  
FOREIGN KEY (JOB_CODE) REFERENCES JOB);
```

- Having created the table structure in Question 1, write the SQL code to enter the first two rows for the table shown in Figure Q6.2.

FIGURE Q6.2 The Contents of the EMP_1 Table

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
▶	101	News	John	G	08-Nov-98	502
	102	Senior	David	H	12-Jul-87	501
	103	Arbough	June	E	01-Dec-94	500
	104	Ramoras	Anne	K	15-Nov-85	501
	105	Johnson	Alice	K	01-Feb-91	502
	106	Smithfield	William		22-Jun-02	500
	107	Alonzo	Maria	D	10-Oct-91	500
	108	Washington	Ralph	B	22-Aug-89	501
	109	Smith	Larry	W	18-Jul-95	501

```
INSERT INTO EMP_1 VALUES ('101', 'News', 'John', 'G', '08-Nov-98', '502');
INSERT INTO EMP_1 VALUES ('102', 'Senior', 'David', 'H', '12-Jul-87', '501');
```

- Assuming that the data shown in the EMP_1 table have been entered, write the SQL code that will list all attributes for a job code of 502.

```
SELECT *
FROM EMP_1
WHERE JOB_CODE = '502';
```

- Write the SQL code that will save the changes made to the EMP_1 table.

```
COMMIT;
```

- Write the SQL code to change the job code to 501 for the person whose personnel number is 106. After you have completed the task, examine the results, and then reset the job code to its original value.

```
UPDATE EMP_1
SET JOB_CODE = '501'
WHERE EMP_NUM = '106';
```

To see the changes:

```
SELECT *
FROM EMP_1
WHERE EMP_NUM = '106';
```

To reset, use
ROLLBACK;

6. Write the SQL code to delete the row for the person named William Smithfield, who was hired on June 22, 2002 and whose job code classification is 500. (*Hint: Use logical operators to include all the information given in this problem.*)

```
DELETE FROM EMP_1
WHERE EMP_LNAME = 'Smithfield'
AND EMP_FNAME = 'William'
AND EMP_HIREDATE = '22-June-02'
AND JOB_CODE = '500';
```

7. Write the SQL code that will restore the data to its original status; that is, the table should contain the data that existed before you made the changes in Questions 5 and 6.

```
ROLLBACK;
```

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 shown next:

```
EMP_PCT NUMBER(4,2)
PROJ_NUM CHAR(3)
```

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

There are two way to get this job done. The two possible solutions are shown next.

Solution A:

```
CREATE TABLE EMP_2 (
    EMP_NUM CHAR(3) NOT NULL UNIQUE,
    EMP_LNAME VARCHAR(15) NOT NULL,
    EMP_FNAME VARCHAR(15) NOT NULL,
    EMP_INITIAL CHAR(1),
    EMP_HIREDATE DATE NOT NULL,
    JOB_CODE CHAR(3) NOT NULL,
    PRIMARY KEY (EMP_NUM),
    FOREIGN KEY (JOB_CODE) REFERENCES JOB);
```

```
INSERT INTO EMP_2 SELECT * FROM EMP_1;
```

```
ALTER TABLE EMP_2
    ADD (EMP_PCT NUMBER (4,2)),
    ADD (PROJ_NUM CHAR(3));
```

Solution B:

```
CREATE TABLE EMP_2 AS SELECT * FROM EMP_1;
```

```
ALTER TABLE EMP_2  
  ADD (EMP_PCT NUMBER (4,2)),  
  ADD (PROJ_NUM CHAR(3));
```

9. Write the SQL code to enter an EMP_PCT value of 3.85 for the person whose employee number (EMP_NUM) is 103. Next, enter the remaining EMP_PCT values shown in Figure Q6.9:

FIGURE Q6.9 The Contents of the EMP_2 Table

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE	EMP_PCT	PROJ_NUM
▶ +	101	News	John	G	08-Nov-98	502	5.00	
+	102	Senior	David	H	12-Jul-87	501	8.00	
+	103	Arbough	June	E	01-Dec-94	500	3.85	
+	104	Ramoras	Anne	K	15-Nov-85	501	10.00	
+	105	Johnson	Alice	K	01-Feb-91	502	5.00	
+	106	Smithfield	William		22-Jun-02	500	6.20	
+	107	Alonzo	Maria	D	10-Oct-91	500	5.15	
+	108	Washington	Ralph	B	22-Aug-89	501	10.00	
+	109	Smith	Larry	wV	18-Jul-95	501	2.00	

```
UPDATE EMP_2  
SET EMP_PCT = 3.85  
WHERE EMP_NUM = '103';
```

To enter the remaining EMP_PCT values, use the following SQL statements:

```
UPDATE EMP_2  
SET EMP_PCT = 5.00  
WHERE EMP_NUM = '101';
```

```
UPDATE EMP_2  
SET EMP_PCT = 8.00  
WHERE EMP_NUM = '102';
```

Follow this format for the remaining rows.

10. Using a single command sequence, write the SQL code that will enter the project number (PROJ_NUM) = 18 for all employees whose job classification (JOB_CODE) is 500.

```
UPDATE EMP_2
SET     PROJ_NUM = '18'
WHERE  JOB_CODE = '500';
```

11. Using a single command sequence, write the SQL code that will enter the project number (PROJ_NUM) = 25 for all employees whose job classification (JOB_CODE) is 502 or higher. When you are done with questions 10 and 11, the EMP_2 table will contain the data shown in Figure Q6.11:

FIGURE Q6.11 The EMP_2 Table Contents After the Modifications

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE	EMP_PCT	PROJ_NUM
▶	101	News	John	G	08-Nov-98	502	5.00	25
	102	Senior	David	H	12-Jul-87	501	8.00	
	103	Arbough	June	E	01-Dec-94	500	3.85	18
	104	Ramoras	Anne	K	15-Nov-85	501	10.00	
	105	Johnson	Alice	K	01-Feb-91	502	5.00	25
	106	Smithfield	William		22-Jun-02	500	6.20	18
	107	Alonzo	Maria	D	10-Oct-91	500	5.15	18
	108	Washington	Ralph	B	22-Aug-89	501	10.00	
	109	Smith	Larry	W	18-Jul-95	501	2.00	

(You may assume that the table has been saved again at this point!)

```
UPDATE EMP_2
SET     PROJ_NUM = '25'
WHERE  JOB_CODE >= '502'
```

12. Write the SQL code that will enter a PROJ_NUM of 14 for those employees who were hired before January 1, 1992 and whose job code is at least 501. (You may assume that the table will be restored to the condition it was in following Question 11.)

```
UPDATE EMP_2
SET     PROJ_NUM = '14'
WHERE  EMP_HIREDATE <= '01-Jan-92'
AND    JOB_CODE >= '501';
```

13. Write the two SQL command sequences required to:

There are many ways to accomplish both tasks. We are illustrating the shortest way to do the job next.

- a. Create a temporary table named TEMP_1, whose structure is composed of the EMP_2 attributes EMP_NUM and EMP_PCT.**

The SQL code shown in problem 13b contains the solution for problem 13a.

- b. Copy the matching EMP_2 values into the TEMP_1 table.**

```
CREATE TABLE TEMP_1 AS SELECT EMP_NUM, EMP_PCT FROM EMP_2;
```

An alternate way would be to create the table and then, use an INSERT with a sub-select to populate the rows.

```
CRATE TABLE TEMP_1 AS (  
EMP_NUM CHAR(3),  
EMP_PCT NUMBER(4,2));
```

```
INSERT INTO TEMP_1  
SELECT EMP_NUM, EMP_PCT FROM EMP_2;
```

14. Write the SQL command that will delete the newly created TEMP_1 table from the database.

```
DROP TABLE TEMP_1;
```

15. Write the SQL code required to list all employees whose last names start with 'Smith'. In other words, the rows for both Smith and Smithfield should be included in the listing.

```
SELECT *  
FROM EMP_2  
WHERE EMP_LNAME LIKE 'Smith%';
```

16. Using the EMPLOYEE, JOB, and PROJECT tables in the Ch06_Review database (see Figure Q6.1), write the SQL code that will produce the results shown in Figure Q6.16.

FIGURE Q6.16 The Query Results for Question 16

	PROJ_NAME	PROJ_VALUE	PROJ_BALANCE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
▶	Rolling Tide	\$805,000.00	\$500,345.20	Senior	David	H	501	Systems Analyst	\$96.75
	Evergreen	\$1,453,500.00	\$1,002,349.75	Arbough	June	E	503	Electrical Engineer	\$84.50
	Starflight	\$2,650,500.00	\$2,309,879.76	Alonzo	Maria	D	500	Programmer	\$35.75
	Amber Wave	\$3,500,500.00	\$2,110,345.91	vWashington	Ralph	B	501	Systems Analyst	\$96.75

```

SELECT PROJ_NAME, PROJ_VALUE, PROJ_BALANCE,
EMPLOYEE.EMP_LNAME,
      EMP_FNAME, EMP_INITIAL, EMPLOYEE.JOB_CODE,
      JOB.JOB_DESCRIPTION,
      JOB.JOB_CHG_HOUR
FROM    PROJECT, EMPLOYEE, JOB
WHERE   EMPLOYEE.EMP_NUM = PROJECT.EMP_NUM
AND     JOB.JOB_CODE = EMPLOYEE.JOB_CODE;

```

17. Write the SQL code that will produce a virtual table named REP_1, containing the same information that was shown in Question 16.

```

CREATE VIEW REP_1 AS
SELECT PROJ_NAME, PROJ_VALUE, PROJ_BALANCE,
EMPLOYEE.EMP_LNAME,
      EMP_FNAME, EMP_INITIAL, EMPLOYEE.JOB_CODE,
      JOB.JOB_DESCRIPTION,
      JOB.JOB_CHG_HOUR
FROM    PROJECT, EMPLOYEE, JOB
WHERE   EMPLOYEE.EMP_NUM = PROJECT.EMP_NUM
AND     JOB.JOB_CODE = EMPLOYEE.JOB_CODE;

```

18. Write the SQL code to find the average bonus percentage in the EMP_2 table you created in question 8.

```

SELECT AVG(EMP_PCT)
FROM    EMP_2;

```

19. Write the SQL code that will produce a listing for the data in the EMP_2 table in ascending order by the bonus percentage.

```

SELECT      *
FROM        EMP_2
ORDER BY    EMP_PCT;

```

20. Write the SQL code that will list only the different project numbers found in the EMP_2 table.

```
SELECT DISTINCT PROJ_NUM
FROM EMP_2;
```

21. Write the SQL code to calculate the ASSIGN_CHARGE values in the ASSIGNMENT table in the Ch06_Review database. (See Figure Q6.1.) Note that ASSIGN_CHARGE is a derived attribute that is calculated by multiplying the ASSIGN_CHG_HR and the ASSIGN_HOURS.

```
UPDATE ASSIGNMENT
SET ASSIGN_CHARGE = ASSIGN_CHG_HR * ASSIGN_HOURS;
```

22. Using the data in the ASSIGNMENT table, write the SQL code that will, for each employee, yield the total number of hours worked and the total charges stemming from those hours worked. The results of running this query are shown in Figure Q6.22.

FIGURE Q6.22 Total Hours and Charges by Employee

	EMP_NUM	EMP_LNAME	SumOfASSIGN_HOURS	SumOfASSIGN_CHARGE
▶	01	News	3.1	\$387.50
	103	Arbough	19.7	\$1,664.65
	104	Ramoras	11.9	\$1,218.70
	105	Johnson	12.5	\$1,382.50
	108	Washington	8.3	\$840.15
	113	Joebrood	3.8	\$192.85
	115	Bawangji	12.5	\$1,276.75
	117	Williamson	18.8	\$649.54

```
SELECT          ASSIGNMENT.EMP_NUM, EMPLOYEE.EMP_LNAME,
                Sum(ASSIGNMENT.ASSIGN_HOURS) AS SumOfASSIGN_HOURS,
                Sum(ASSIGNMENT.ASSIGN_CHARGE) AS SumOfASSIGN_CHARGE
FROM            EMPLOYEE, ASSIGNMENT
WHERE           EMPLOYEE.EMP_NUM = ASSIGNMENT.EMP_NUM
GROUP BY       ASSIGNMENT.EMP_NUM, EMPLOYEE.EMP_LNAME;
```


23. Write a query to produce the total number of hours and charges for each of the projects represented in the ASSIGNMENT table. The output is shown in Figure Q6.23.

FIGURE Q6.23 Total Hours and Charges by Project

	PROJ_NUM	SumOfASSIGN_HOURS	SumOfASSIGN_CHARGE
▶	15	20.5	\$1,806.52
	18	23.7	\$1,544.80
	22	27.0	\$2,593.16
	25	19.4	\$1,668.16

```

SELECT      ASSIGNMENT.PROJ_NUM,
            Sum(ASSIGNMENT.ASSIGN_HOURS) AS SumOfASSIGN_HOURS,
            Sum(ASSIGNMENT.ASSIGN_CHARGE) AS
SumOfASSIGN_CHARGE
FROM        ASSIGNMENT
GROUP BY   ASSIGNMENT.PROJ_NUM

```

24. Write the SQL code to generate the total hours worked and the total. The results are shown in Figure Q6.24.

FIGURE Q6.24 Total Hours and Charges, All Employees

	SumOfSumOfASSIGN_HOURS	SumOfSumOfASSIGN_CHARGE
▶	90.6	\$7,612.64

```

SELECT      Sum(ASSIGNMENT.ASSIGN_HOURS) AS SumOfASSIGN_HOURS,
            Sum(ASSIGNMENT.ASSIGN_CHARGE) AS SumOfASSIGN_CHARGE
FROM        ASSIGNMENT;

```

FIGURE Q6.1 STRUCTURE AND CONTENTS OF THE CH06_REVIEW DATABASE

