Database Programming with SQL

4-1: Case and Character Manipulation

Practice Activities

Objectives

• Select and apply single-row functions that perform case conversion and/or character manipulation
• Select and apply character case-manipulation functions LOWER, UPPER, and INITCAP in a
SQL query
• Select and apply character-manipulation functions CONCAT, SUBSTR, LENGTH, INSTR,
LPAD, RPAD, TRIM, and REPLACE in a SQL query
• Write flexible queries using substitution variables

Vocabulary

Identify the vocabulary word for each definition below.

- Dummy table used to view results from functions and calculations
    - DUAL
- The arrangement of data for storage or display.
    - FORMAT
- Converts alpha character values to uppercase for the first letter of each word, all other letters in lowercase.
    - INITCAP
- Functions that accept character data as input and can return both character and numeric values.
    - CHARACTER MANIPULATION FUNCTIONS
- Removes all specified characters from either the beginning or the ending of a string.
    - TRIM
- A symbol that represents a quantity or a relationship between quantities
    - EXPRESSION
- Functions that operate on single rows only and return one result per row
    - SINGLE ROW FUNCTIONS
- Converts alpha characters to upper case
    - UPPER
- Raw data entered into the computer
    - INPUT

- Concatenates the first character value to the second character value; equivalent to concatenation operator (||).
    - CONCAT
- Data that is processed into information
    - OUTPUT
- Converts alpha character values to lowercase.
    - LOWER
- Pads the left side of a character, resulting in a right-justified value
    - LPAD
- Returns specific characters from character value starting at a specific character position and going specified character positions long
    - SUBSTR
- Replaces a sequence of characters in a string with another set of characters.
    - REPLACE
- Returns the numeric position of a named string.
    - INSTR
- Returns the number of characters in the expression
    - LENGTH
- Pads the right-hand side of a character, resulting in a left- justified value.
    - RPAD

Try It / Solve It

1. Using the three separate words “Oracle,” “Internet,” and “Academy,” use one command to produce the following output:

The Best Class
Oracle Internet Academy

SELECT CONCAT ‘Oracle’|| ‘Internet’|| ‘Academy’ AS “The Best Class”
FROM DUAL;

2. Use the string “Oracle Internet Academy” to produce the following output:

The Net
net

SELECT SUBSTR(‘Oracle Internet Academy’, 13, 3) AS “The Net”
FROM DUAL;

3. What is the length of the string “Oracle Internet Academy”?

SELECT LENGTH(‘Oracle Internet Academy’) AS “LENGTH”
FROM DUAL;

4. What’s the position of “I” in “Oracle Internet Academy”?

SELECT INSTR(‘Oracle Internet Academy’, ‘I’) AS “POSITION”

FROM DUAL;
5. Starting with the string “Oracle Internet Academy”, pad the string to create ****Oracle****Internet****Academy****
SELECT LPAD(‘Oracle’, 10, ‘*’)||LPAD(Internet, 12, ‘*’)||RPAD(LPAD(Academy, 11, ‘*’), 15, ‘*’) AS “OAI”
FROM DUAL;
6. Starting with the string “Oracle Internet Academy”, pad the string to produce: Oracle$$$Internet$$$Academy
SELECT RPAD(‘Oracle’, 9, ‘$’)||RPAD(‘Internet’, 11, ‘$’)|| ‘Academy’ AS “OIA”
FROM DUAL;
7. Using the string ‘Oracle Internet Academy’, produce the output shown using the REPLACE
function.

The Best Class
Oracle 2013-2014 Academy

SELECT REPLACE(‘Oracle Internet Academy’, ‘Interet’, ‘2013-2014’) AS “The Best Class”
FROM DUAL;
8. List the order date and the order total from the Global Fast Foods F_ORDERS table. Name the order total as TOTAL, and fill in the empty spaces to the left of the order total with $.
SELECT order_date, LPAD(‘order_total’, 10, ‘$’) AS “TOTAL”
FROM F_ORDERS;
SELECT order_date, order_total AS “TOTAL”, LPAD(‘TOTAL’, 12, ‘$’)
FROM F_ORDERS;
9. Write a query that will output a column called “ADDRESS” which has the following information:
ZOE TWEE 1009 OLIVER AVENUE BOSTON, MA 12889. Use the Global Fast Foods F_CUSTOMERS table.

SELECT UPPER(first_name) ||' '|| UPPER(last_name) ||' '|| UPPER(address)||' ' ||UPPER(city) ||',
'||UPPER(state)||' '||zip AS "ADDRESS"
FROM f_customers
WHERE id = 456;

10.Write a query to return the first character of the first name concatenated to the last_name, the salary, and the department id for employees working in department 20. Give the first expression an alias of Name. Use the EMPLOYEES table. Change the query to use a substitution variable instead of the hard coded value 20 for department

id. Run the query for department 30 and 50 without changing the original where-clause in your statement.

```
SELECT SUBSTR(first_name, 1,1) || last_name AS "Name", salary, department_id
FROM employees
WHERE department_id = 20;
SELECT SUBSTR(first_name, 1,1) || last_name AS "Name", salary, department_id
FROM employees
WHERE department_id = :dept_id;
```

11.Using a substitution variable for the department name, write a query listing department id,
department name, and location id for departments located in the_department_of_your_choice.
Use the DEPARTMENTS table. Note: All substitution variables in OAE are treated as character
strings, so no quotes (‘ ‘) are needed.

```
SELECT department_id, department_name, location_id
FROM departments
WHERE department_name = :dept_name;
```

12.Write a query that returns all the employee data depending on the month of their hire date. Use the EMPLOYEES table. The statement should return the month part of the hiredate which is then compared to an abbreviated month (JAN, FEB, MAR) passed into the query via a substitution Variable.

```
SELECT *
FROM employees
WHERE SUBSTR(hire_date, 4, 3) = :entered_month;
```

Database Programming with SQL

4-2: Number Functions

Practice Activities

Objectives
• Select and apply the single-row number functions ROUND, TRUNC, and MOD in a SQL query

• Distinguish between the results obtained when TRUNC is applied to a numeric value and
ROUND is applied to a numeric value
• State the implications for business when applying TRUNC and ROUND to numeric values

Vocabulary
Identify the vocabulary word for each definition below.

- Used to terminate the column, expression, or value to a specified number of decimal places
    - TRUNC
- These functions accept numeric input and return numeric values.
    - NUMBER FUNCTIONS
- Returns the remainder of a division.
    - MOD
- Rounds the column, expression, or value to a set number of decimal places.
    - ROUND

Try It / Solve It
1. Display Oracle database employee last_name and salary for employee_ids between 100 and 102. Include a third column that divides each salary by 1.55 and rounds the result to two decimal
Places.

```
SELECT last_name, ROUND(salary/1.55,2) AS "Salary Calculation"
FROM employees
WHERE employee_id BETWEEN 100 AND 102;
```

2. Display employee last_name and salary for those employees who work in department 80. Give each of them a raise of 5.333% and truncate the result to two decimal places.

```
SELECT last_name, TRUNC(salary * 1.05333,2) AS "Salary with Raise"
FROM employees
WHERE department_id = 80;
```

3. Use a MOD number function to determine whether 38873 is an even number or an odd number.

Odd.

```
SELECT MOD(38873,2)
FROM DUAL;
```

4. Use the DUAL table to process the following numbers:
845.553 - round to one decimal place
30695.348 - round to two decimal places
30695.348 - round to -2 decimal places
2.3454 - truncate the 454 from the decimal place

```
SELECT round(845.553,1)
FROM DUAL;
```
30695.348 - round to two decimal places
```
SELECT round(30695.348,2)
FROM DUAL;
```
30695.348 - round to -2 decimal places
```
SELECT ROUND(30695.348,-2)
FROM DUAL;
```
2.3454 - truncate the 454 from the decimal place
```
SELECT TRUNC(2.3454,1)
FROM DUAL;
```

5. Divide each employee's salary by 3. Display only those employees' last names and salaries who earn a salary that is a multiple of 3.

```
SELECT last_name, salary
FROM employees
WHERE MOD(salary,3) = 0;
```

6. Divide 34 by 8. Show only the remainder of the division. Name the output as EXAMPLE.

```
SELECT MOD(34,8) AS Example
FROM DUAL;
```

7. How would you like your paycheck – rounded or truncated? What if your paycheck was calculated to be $565.784 for the week, but you noticed that it was issued for $565.78. The loss of .004 cent would probably make very little difference to you. However, what if this was done to one thousand people, one hundred thousand people, or one million people! Would it make a difference then? How much of a difference?

.004 * 1 = .004; .004 *1000 = $4.00;
.004*100,000 = $ 400.00;
.004 * 1,000,000 = $4000.00

Database Programming with SQL

4-3: Date Functions

Practice Activities

Objectives
• Select and apply the single-row functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY,
LAST_DAY, ROUND, and TRUNC that operate on date data
• Explain how date functions transform Oracle dates into date data or numeric values
• Demonstrate proper use of the arithmetic operators with dates
• Demonstrate the use of SYSDATE and date functions
• State the implications for world businesses to be able to easily manipulate data stored in date
format

Vocabulary
Identify the vocabulary word for each definition below.

- A function that returns the current date and time of the database server.
    - SYSDATE
- Add calendar months to date
    - ADD_MONTHS
- Last day of the month
    - LAST_DAY
- Next day of the date specified
    - NEXT_DAY
- Number of months between due dates
    - MONTHS_BETWEEN

Try It / Solve It
1. For DJs on Demand, display the number of months between the event_date of the Vigil wedding and today's date. Round to the nearest month.

SELECT ROUND(MONTHS_BETWEEN (SYSDATE,event_date)) AS MONTHS

FROM d_events where id=105;

2. Display the days between the start of last summer’s school vacation break and the day school started this year. Assume 30.5 days per month. Name the output “Days.”

3. Display the days between January 1 and December 31.

SELECT ROUND(MONTHS_BETWEEN ('31-Dec-2004','01-Jan-2004')*30.5) AS DAYS
FROM DUALI;

4. Using one statement, round today's date to the nearest month and nearest year, and truncate it to the nearest month and nearest year. Use an alias for each column.

SELECT ROUND(SYSDATE,'MONTH') AS Month,ROUND(SYSDATE,'YEAR') AS YEAR,TRUNC(SYSDATE,'MONTH')AS Month, TRUNC(SYSDATE,'YEAR') AS Year
FROM DUAL;

5. What is the last day of the month for June 2005? Use an alias for the output.

SELECT LAST_DAY('01-Jun-2005') AS "LAST DAY"
FROM DUAL;

6. Display the number of years between the Global Fast Foods employee Bob Miller’s birthday and today. Round to the nearest year.

SELECT last_name, ROUND(MONTHS_BETWEEN(SYSDATE, birthdate)/12) AS YEARS
FROM f_staffs
WHERE id = 9;

7. Your next appointment with the dentist is six months from today. On what day will you go to the dentist? Name the output, “Appointment.”

SELECT ADD_MONTHS(SYSDATE,6) AS Appointment
FROM DUAL;

8. The teacher said you have until the last day of this month to turn in your research paper. What day will this be? Name the output, “Deadline.”

SELECT LAST_DAY(SYSDATE) AS Deadline

FROM DUAL;

9. How many months between your birthday this year and January 1 next year?

```
SELECT ROUND(MONTHS_BETWEEN('24-Mar-2024', '05-Jan-2025'))
FROM DUAL;
```

10.What's the date of the next Friday after your birthday this year? Name the output, "First
Friday."

```
SELECT ROUND(NEXT_DAY('29-Mar-2025','Friday'))AS "First Friday"
FROM DUAL;
```

11.Name a date function that will return a number.

MONTHS_BETWEEN

12.Name a date function that will return a date.
ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, TRUNC

13.Give one example of why it is important for businesses to be able to manipulate date data?

To schedule payrolls and payments

<u>Extension Exercises</u>

1. Using DUAL, write a statement that will convert 86.678 to 86.68.

```
SELECT ROUND(86.678,2)
FROM DUAL;
```

2. Write a statement that will display the DJs on Demand CD titles for cd_numbers 90 and 91 in
uppercase in a column headed "DJs on Demand Collections."

```
SELECT UPPER(title) AS "DJs on Demand Collections"
FROM d_cds
WHERE cd_number IN (90,91);
```

3. Write a statement that will create computer usernames for the DJs on Demand partners. The
usernames will be the lowercase letters of the last name + the uppercase first letter in the first
name. Title the column “User Passwords.” For example, Mary Smythers would be smythersM.

```
SELECT CONCAT(LOWER(last_name),UPPER(SUBSTR(first_name,1,1))) AS "User Passwords"
FROM d_partners;
```

4. Write a statement that will convert “It’s a small world” to “HELLO WORLD.”

```
SELECT UPPER(CONCAT('hello ',(SUBSTR('Its a small world',13, 18))))
FROM DUAL;
```

5. Write a statement that will remove the “fiddle” from “fiddledeedee” and the “dum” from “fiddledeedum.” Display the result “fiddledeedeedee” in a column with the heading “Nonsense.”

```
SELECT SUBSTR('fiddledeedum',1,9)||SUBSTR('fiddledeedee',7, 12)AS "Nonsense"
FROM DUAL;
```

6. Replace every “i” in Mississippi with “$.”

```
SELECT REPLACE('Mississippi','i','$')
FROM DUAL;
```

7. Using DUAL, convert 5332.342 to 5300.

```
SELECT ROUND(5332.342,-2)
FROM DUAL;
```

8. Using DUAL, convert 3.14159 to 3.14.

```
SELECT ROUND(3.14159 ,2)
FROM DUAL;
```

9. Using DUAL, convert 73.892 to 73.8.

```
SELECT TRUNC(73.892,1)
FROM DUAL;
```

10.What is the next Friday six months from now? Label the column “Future.”

```
SELECT NEXT_DAY(ADD_MONTHS(SYSDATE,6),'Friday') AS "Future"
FROM DUAL;
```

11.What is the date 10 years from now? Label the column “Future.”

```
SELECT ADD_MONTHS(SYSDATE,120) AS "Future"
FROM DUAL;
```

12.Leap years occur every four years. Remember, 2004 was a leap year. Now create a function that will show the date of the next leap year as 29-Feb-2008. Label the column “Future.”

```
SELECT ADD_MONTHS(LAST_DAY('01-Feb-2004'),48) AS "Future"
FROM DUAL;
```

13.Write a statement that will find any of the DJs on Demand CD themes that have an “ie” in their names.

```
SELECT description
FROM d_themes
WHERE description LIKE '%ie%';
```

14.Write a statement that will return only the DJs on Demand CDs with years greater than 2000 but less than 2003. Display both the title and year.

```
SELECT title, year
FROM d_cds
WHERE year > 2000 AND year < 2003;
```

15.Write a statement that will return the Oracle database employee’s employee ID and his starting hire dates between January 1, 1997 and today. Display the result ordered from most recently hired to the oldest.

```
SELECT employee_id, start_date
FROM job_history
WHERE start_date BETWEEN '01-Jan-1997'AND SYSDATE
```

```
ORDER BY start_date DESC, employee_id;
```