Database Programming with SQL  
4-1: Case and Character Manipulation  
  
Vocabulary

* DUAL Table - Dummy table used to view results from functions and calculations
* Format- The arrangement of data for storage or display.
* INITCAP - Converts alpha character values to uppercase for the first letter of  
  each word, all other letters in lowercase.
* Single-row functions - Functions that accept character data as input and can return both character and numeric values.
* TRIM - Removes all specified characters from either the beginning or the ending of a string.
* Operator - A symbol that represents a quantity or a relationship between  
  quantities
* Single-row Functions - Functions that operate on single rows only and return one result per row
* UPPER - Converts alpha characters to upper case
* Input - Raw data entered into the computer
* CONCAT - Concatenates the first character value to the second character  
  value; equivalent to concatenation operator (||).
* Output - Data that is processed into information
* LOWER - Converts alpha character values to lowercase.
* LPAD - Pads the left side of a character, resulting in a right-justified value
* SUBSTR - Returns specific characters from character value starting at a  
  specific character position and going specified character positions  
  long
* REPLACE - Replaces a sequence of characters in a string with another set of  
  characters.
* INSTR - Returns the numeric position of a named string.
* LENGTH - Returns the number of characters in the expression
* RPAD - Pads the right-hand side of a character, resulting in a left- justified  
  value.

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 'The Best Class ' || 'Oracle ' || 'Internet ' || 'Academy' FROM DUAL;**

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

**SELECT 'The Net' AS "The Net", LOWER(SUBSTR('Oracle Internet Academy', 9, 3)) FROM DUAL;**

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

**23**  
4. What’s the position of “I” in “Oracle Internet Academy”?

**8**  
5. Starting with the string “Oracle Internet Academy”, pad the string to  
create \*\*\*\*Oracle\*\*\*\*Internet\*\*\*\*Academy\*\*\*\*

**SELECT RPAD(LPAD('Oracle', 10, '\*'), 14, '\*') || RPAD(LPAD('Internet', 12, '\*'), 16, '\*') || RPAD(LPAD('Academy', 11, '\*'), 15, '\*') FROM DUAL;**

6. Starting with the string “Oracle Internet Academy”, pad the string to produce:  
Oracle$$$Internet$$$Academy

**SELECT REPLACE('Oracle Internet Academy', ' ', '$$$') 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 'The Best Class' AS "The Best Class", REPLACE('Oracle Internet Academy', 'Internet', '2013-2014') 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, LENGTH(order\_total)**  
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 first\_name || ' ' || last\_name || ' ' || address || ' ' || city || ', ' || state || ' ' || postal\_code AS ADDRESS**

**FROM F\_CUSTOMERS**

**WHERE first\_name = 'ZOE' AND last\_name = 'TWEE';**  
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 = &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 hire date which is then compared to an abbreviated month (JAN, FEB, MAR) passed into the query via a substitution variable

**SELECT \***

**FROM EMPLOYEES**

**WHERE TO\_CHAR(hire\_date, 'MON') = &hire\_month;**

4-2: Number Functions  
  
Vocabulary

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

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, salary, ROUND(salary / 1.55, 2) AS rounded\_salary**

**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 new\_salary**

**FROM EMPLOYEES**

**WHERE department\_id = 80;**  
3. Use a MOD number function to determine whether 38873 is an even number or an odd number.

**SELECT MOD(38873, 2) AS result**

**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) AS round1,**

**ROUND(30695.348, 2) AS round2,**

**ROUND(30695.348, -2) AS round3,**

**TRUNC(2.3454, 2) AS trunc1**

**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?

**For a single person, it is not much of a loss to lose $0.004 per paycheck. For 1000 people, the loss becomes $4, and for 1,000,000 people, it becomes $4000. From a business perspective, if this practice continues, it could either save or lose a substantial amount of money over time for a company.**

4-3: Date Functions  
  
Vocabulary

* SYSDATE - A function that returns the current date and time of the database  
  server.
* ADD\_MONTHS - Add calendar months to date
* LAST\_DAY - Last day of the month
* NEXT\_DAY - Next day of the date specified
* MONTHS\_BETWEEN - Number of months between due dates

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\_between**

**FROM EVENTS**

**WHERE event\_name = 'Vigil Wedding';**  
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.”

**SELECT ROUND((school\_start - vacation\_start) \* 30.5) AS Days**

**FROM DUAL;**  
3. Display the days between January 1 and December 31.

**SELECT ROUND((TO\_DATE('31-DEC', 'DD-MON') - TO\_DATE('01-JAN', 'DD-MON'))) AS Days**

**FROM DUAL;**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 nearest\_month,**

**ROUND(SYSDATE, 'YEAR') AS nearest\_year,**

**TRUNC(SYSDATE, 'MONTH') AS trunc\_month,**

**TRUNC(SYSDATE, 'YEAR') AS trunc\_year**

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

**SELECT LAST\_DAY(TO\_DATE('01-JUN-2005', 'DD-MON-YYYY')) 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 ROUND(MONTHS\_BETWEEN(SYSDATE, birth\_date) / 12, 0) AS years**

**FROM EMPLOYEES**

**WHERE first\_name = 'Bob' AND last\_name = 'Miller';**  
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 MONTHS\_BETWEEN(TO\_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO\_DATE('29-JULY-2024', 'DD-MON-YYYY')) AS months\_between**

**FROM DUAL;**

***Output: 5.09677419354838709677419354838709677419***

10. What’s the date of the next Friday after your birthday this year? Name the output, “First Friday.”

**SELECT NEXT\_DAY(TO\_DATE('29-JULY-2024', 'DD-MON-YYYY'), 'FRIDAY') AS First\_Friday**

**FROM DUAL;**

***Output: 02-Aug-2024***

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

**MONTHS\_BETWEEN**  
12. Name a date function that will return a date.

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

**Businesses need to manipulate date data for scheduling and planning purposes like creating accurate project timelines, payroll processing, and tracking deadlines for reports.**

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

**SELECT ROUND(86.678, 2) AS rounded\_value**

**FROM DUAL;**

***86.68***  
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(cd\_title) AS "DJs on Demand Collections"**

**FROM DJ\_CD**

**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 LOWER(last\_name) || UPPER(SUBSTR(first\_name, 1, 1)) AS "User Passwords"**

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

**SELECT REPLACE(UPPER('It’s a small world'), 'IT’S A SMALL WORLD', 'HELLO WORLD') AS greeting**

**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 REPLACE(REPLACE('fiddledeedum', 'fiddle', ''), 'dum', '') AS Nonsense**

**FROM DUAL;**  
6. Replace every “i” in Mississippi with “$.”

**SELECT REPLACE('Mississippi', 'i', '$') AS new\_string**

**FROM DUAL;**  
7. Using DUAL, convert 5332.342 to 5300.

**SELECT ROUND(5332.342, -2) AS rounded\_value**

**FROM DUAL;**

***5300***

8. Using DUAL, convert 3.14159 to 3.14.

**SELECT ROUND(3.14159, 2) AS rounded\_value**

**FROM DUAL;**

***3.14***  
9. Using DUAL, convert 73.892 to 73.8.

**SELECT ROUND(73.892, 1) AS rounded\_value**

**FROM DUAL;**

***73.9***  
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;**

***04-Apr-2025***  
11. What is the date 10 years from now? Label the column “Future.”

**SELECT ADD\_MONTHS(SYSDATE, 120) AS Future**

**FROM DUAL;**

***02-Oct-2034***  
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 TO\_DATE('29-FEB-2028', 'DD-MON-YYYY') AS Future**

**FROM DUAL;**

***29-Feb-2028***  
13. Write a statement that will find any of the DJs on Demand CD themes that have an “ie” in their names.

**SELECT cd\_title**

**FROM DJ\_CD**

**WHERE cd\_title 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 cd\_title, year**

**FROM DJ\_CD**

**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, hire\_date**

**FROM EMPLOYEES**

**WHERE hire\_date BETWEEN TO\_DATE('01-JAN-1997', 'DD-MON-YYYY') AND SYSDATE**

**ORDER BY hire\_date DESC;**