Suggested Solution

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

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

The Best Class
Oracle Internet Academy

Solution:

```
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

Solution:

```
SELECT SUBSTR('Oracle Internet Academy', 13,3)AS "The Net"
FROM DUAL;
```

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

Solution:

23

```
SELECT LENGTH('Oracle Internet Academy')AS "Length"
FROM DUAL;
```

4. What's the position of "I" in "Oracle Internet Academy"?

   Solution:
   8
   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****

   Solution:
   SELECT LPAD('Oracle',10,'*')||LPAD('Internet',12,'*')||RPAD(LPAD('Academy',11,'*'),15,'*')AS "OIA"
   FROM DUAL;

6. 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.

   Solution:
   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

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.

   Solution:
   ```
   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.

   Solution:
   ```
   SELECT last_name, TRUNC(salary * 1.05333,2) AS "Salary with Raise"
   FROM employees
   WHERE department_id = 80;
   -Alternate answer -
   SELECT last_name, TRUNC(salary * .05333,2) AS "Raise Amount"
   FROM employees
   WHERE department_id = 80;
   ```

3. 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.

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

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

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.
   Solution:
   SELECT ROUND(MONTHS_BETWEEN (SYSDATE,event_date)) AS MONTHS
   FROM d_events where id=105;

2. Display the number of years between the Global Fast Foods employee Bob Miller's birthday and today. Round to the nearest year.
   Solution:
   SELECT last_name, ROUND(MONTHS_BETWEEN(SYSDATE, birthdate)/12)
   AS YEARS
   FROM f_staffs
   WHERE id = 9;
3. Your next appointment with the dentist is six months from today. On what day will you go to the dentist? Name the output, â€œAppointment.â€
   Solution:
   SELECT ADD_MONTHS(SYSDATE,6) AS Appointment FROM DUAL;

4. Calculate number of days between '05-Sep-2004', '15-Jun-2004'. (hint, you can try to calculate number of months

Solution:

```
SELECT ROUND(MONTHS_BETWEEN ('05-Sep-2004', '15-Jun-2004')*30.5) AS DAYS
FROM dual;
```

or

check default format, and use to_date function.

```
select *
from nls_session_parameters
where parameter = 'NLS_DATE_FORMAT';


select to_date( '05-Sep-2004')  - to_date ( '15-Jun-2004') from dual;
```

Database Programming with SQL5-1: Conversion Functions Practice Activities

Objectives

- Provide an example of an explicit data-type conversion and an implicit data-type conversion
- Explain why it is important, from a business perspective, for a language to have built-in data-conversion capabilities
- Construct a SQL query that correctly applies TO_CHAR, TO_NUMBER, and TO_DATE single-row functions to produce a desired result
- Apply the appropriate date and/or character format model to produce a desired output
- Explain and apply the use of YY and RR to return the correct year as stored in the database

1. List the last names and birthdays of Global Fast Food Employees. Convert the birth dates to character data in the Month DD, YYYY format. Suppress any leading zeros.

   Solution:
   ```
   SELECT last_name, TO_CHAR(birthdate, 'Month fmDD, RRRR') AS "Birthday"
   FROM f_staffs;
   ```

2. Format a query from the Global Fast Foods f_promotional_menus table to print out the start_date of promotional code 110 as: The promotion began on the tenth of February 2004.

   Solution:
   ```
   SELECT 'The promotion began on the '|| TO_CHAR(start_date, 'ddspth "of" Month YYYY') AS "Date"
   FROM f_promotional_menus
   WHERE code = 110;
   ```

3. List the ID, name, and salary for all Global Fast Foods employees. Display salary with a $ sign and two decimal places.

   Solution:
   ```
   SELECT id, first_name, TO_CHAR(salary, '$999.99')
   FROM f_staffs;
   ```

Database Programming with SQL
5-2: NULL Functions
Practice Activities
Objectives

- Demonstrate and explain the evaluation of a nested function
- List at least four general functions that work with any data type and relate to handling null values
- Explain the use of the COALESCE and the NVL functions
- Explain the use of general functions to deal with null values in data
- Construct and execute a SQL query that correctly applies NVL, NVL2, NULLIF, and COALESCE single-row functions

Use aliases to make the output more readable.

1. Create a report that shows the Global Fast Foods promotional name, start date, and end date from the f_promotional_menus table. If there is an end date, temporarily replace it with “end in two weeks.” If there is no end date, replace it with today’s date.

   Solution:
   ```
   SELECT name, NVL2(end_date, 'end in two weeks', SYSDATE) AS Promotion
   FROM f_promotional_menus;
   ```

2. The manager of Global Fast Foods has decided to give all staff who currently do not earn overtime an overtime rate of $5.00. Construct a query that displays the last names and the overtime rate for each staff member, substituting $5.00 for each null overtime value.

   Solution:
   ```
   SELECT last_name, TO_CHAR(NVL(overtime_rate,5.00), '$9999.99') AS "Overtime Rate"
   FROM f_staffs;
   ```

3. Not all Global Fast Foods staff members have a manager. Create a query that displays the employee last name and 9999 in the manager ID column for these employees.

   Solution:
   ```
   SELECT last_name, NVL(manager_id, 9999)
   FROM f_staffs;
   ```

4.
   a. Create a report listing the first and last names and month of hire for all employees in the EMPLOYEES table (use TO_CHAR to convert hire_date to display the month).

      Solution:
      ```
      SELECT first_name||' '||last_name "Name", to_char(hire_date,'Month') "Anniversary Month"
      FROM employees;
      ```

   b. Modify the report to display null if the month of hire is September. Use the NULLIF function.

      Solution:
      ```
      SELECT first_name||' '||last_name "Name",
      NULLIF(to_char(hire_date,'Month'),'September') "Anniversary Month"
      FROM employees;
      ```

5. Display the first name, last name, manager ID, and commission percentage of all employees in departments 80 and 90. In a 5th column called “Review”, again display the manager ID. If they don’t have a manager, display the commission percentage. If they don’t have a commission, display 99999.

   Solution:
   ```
   select first_name||' '||last_name as "Name", manager_id,commission_pct ,
    COALESCE(manager_id,commission_pct,9999) as "Review"
    from employees;
   ```

Database Programming with SQL5-3: Conditional Expressions Practice Activities

Objectives

- Compare and contrast the DECODE and CASE functions
- Construct and execute a SQL query that correctly uses the DECODE and CASE functions
- Construct and execute two methods for implementing IF-THEN-ELSE conditional logic

1. From the DJs on Demand d_songs table, create a query that replaces the 2-minute songs with “shortest” and the 10-minute songs with “longest”. Label the output column “Play Times”.

   Solution:

```
SELECT id, title, duration,
DECODE(duration, '2 min', 'shortest', '10 min', 'longest')
AS "Play Times"
FROM d_songs;
```

2. Use the Oracle database employees table and CASE expression to decode the department id. Display the department id, last name, salary, and a column called “New Salary” whose value is based on the following conditions:
   If the department id is 10 then 1.25 * salary
   If the department id is 90 then 1.5 * salary
   If the department id is 130 then 1.75 * salary
   Otherwise, display the old salary.

   Solution:

```
SELECT last_name, department_id, salary,
CASE department_id WHEN 10 THEN salary*1.25
WHEN 90 THEN salary*1.50
WHEN 130 THEN salary*1.75
ELSE salary END As "New Salary"
FROM employees;
```