**Programming 4,5**

**Database Programming with SQL  
4-1: Case and Character Manipulation  
Practice Activities**

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **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 |
| **Character 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 function** | 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 |

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`)||` AS “The Best Class” FROM DUAL;

1. 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;

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

23

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

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

8

SELECT INSTR(`Oracle Internet Academy` , `I`) AS “Position of I” From DUAL;

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

SELECT RPAD(LPAD('Oracle', 10, '\*'), 15, '\*') ||

RPAD(LPAD('Internet', 10, '\*'), 15, '\*') ||

RPAD(LPAD('Academy', 10, '\*'), 15, '\*') AS PaddedString

FROM DUAL;

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

SELECT RPAD('Oracle', 9, '$') ||

RPAD('Internet', 11, '$') ||

('Academy') AS PaddedString

FROM DUAL;

1. 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', 'Internet', '2013-2014') AS "The Best Class" FROM DUAL;

1. 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) + 5, '$') AS TOTAL FROM F\_ORDERS;

1. 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 CONCAT(CONCAT(FIRST\_NAME, ' '), LAST\_NAME) || ' ' || ADDRESS AS ADDRESS

FROM F\_CUSTOMERS

WHERE FIRST\_NAME = 'ZOE' AND LAST\_NAME = 'TWEE';

1. 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;

1. 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;

1. 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 employess

WHERE SUBSTR(hire\_date, 4, 3) = :month\_abbrev;

**Database Programming with SQL  
4-2: Number Functions  
Practice Activities**

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **Truncate (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 adjusted\_salary

FROM employees

WHERE employee\_id BETWEEN 100 AND 102;

1. 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, salary, TRUNC(salary \* 1.05333, 2) AS salary\_with\_raise FROM employees WHERE department\_id = 80;

1. 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;

To determine if a number is odd or even, we use the MOD function. If the result is 0, it’s even, otherwise, it’s odd.

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

SELECT ROUND(845.553, 1) AS rounded\_value FROM DUAL;

1. 30695.348 - round to two decimal places

SELECT ROUND(30695.348, 2) AS rounded\_value FROM DUAL;

1. 30695.348 - round to -2 decimal places

SELECT ROUND(30695.348, -2) AS rounded\_value FROM DUAL;

1. 2.3454 - truncate the 454 from the decimal place

SELECT TRUNC(2.3454, 2) AS truncated\_value FROM DUAL;

1. 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;

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

SELECT MOD(34, 8) AS EXAMPLE FROM DUAL;

1. 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?

* Rounding would give you $565.78, removing the small fraction of 0.004 cents.
* Truncating the number keeps you with the same amount but if done on a larger scale (like for 1,000, 100,000, or 1 million people), the total loss would add up.
* For 1,000 people: $0.004 \* 1,000 = $4.00 lost
* For 100,000 people: $0.004 \* 100,000 = $400 lost
* For 1,000,000 people: $0.004 \* 1,000,000 = $4,000 lost

**Database Programming with SQL  
4-3: Date Functions  
Practice Activities**

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **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';

1. 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 (end\_date - start\_date) \* 30.5 AS "Days" FROM vacation\_period WHERE period\_name = 'Summer Break';

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

SELECT TO\_DATE('31-DEC-2024', 'DD-MON-YYYY') - TO\_DATE('01-JAN-2024', 'DD-MON-YYYY') AS days\_between FROM DUAL;

1. 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 "Rounded Month",

ROUND(SYSDATE, 'YEAR') AS "Rounded Year",

TRUNC(SYSDATE, 'MONTH') AS "Truncated Month",

TRUNC(SYSDATE, 'YEAR') AS "Truncated Year"

FROM DUAL;

1. 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;

1. 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, TO\_DATE('15-FEB-1980', 'DD-MON-YYYY')) / 12, 0) AS "Years" FROM employees WHERE first\_name = 'Bob' AND last\_name = 'Miller';

1. 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;

1. 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;

1. 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('11-FEB-2024', 'DD-MON-YYYY')) AS months\_between FROM DUAL;

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

SELECT NEXT\_DAY(TO\_DATE('11-FEB-2024', 'DD-MON-YYYY'), 'FRIDAY') AS "First Friday" FROM DUAL;

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

MONTHS\_BETWEEN()

1. Name a date function that will return a date.

ADD\_MONTHS()

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

* Being able to manipulate date data allows businesses to schedule events, manage deadlines, and track employee records efficiently, which is crucial for planning and operations.

**Extension Exercises**

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

SELECT ROUND(86.678, 2) FROM DUAL;

1. 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 cds

WHERE cd\_number IN (90, 91);

1. 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;

1. 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;

1. 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('fiddledeedee', 'fiddle', '') || REPLACE('fiddledeedum', 'dum', '') AS "Nonsense"

FROM DUAL;

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

SELECT REPLACE('Mississippi', 'i', '$') AS replaced\_text

FROM DUAL;

1. Using DUAL, convert 5332.342 to 5300.

SELECT TRUNC(5332.342, -2) FROM DUAL;

1. Using DUAL, convert 3.14159 to 3.14.

SELECT ROUND(3.14159, 2) FROM DUAL;

1. Using DUAL, convert 73.892 to 73.8.

SELECT ROUND(73.892, 1) FROM DUAL;

1. 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;

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

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

FROM DUAL;

1. 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;

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

SELECT cd\_theme

FROM cds

WHERE cd\_theme LIKE '%ie%';

1. 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, release\_year

FROM cds

WHERE release\_year > 2000 AND release\_year < 2003;

1. 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;

**Database Programming with SQL  
5-1: Conversion Functions  
Practice Activities**

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **CHAR** | Used for text and character data of fixed length, including numbers, dashes, and special characters. |
| **TRIM** | Used to remove padded blanks or to suppress leading zeros |
| **Conversion functions** | Functions that convert a value from one datatype to another. |
| **NUMBER** | Used to store variable-length numeric data. |
| **VARCHAR2** | Used for character data of variable length, including numbers, special characters, and dashes. |
| **DATE** | Used for date and time values. |
| **TO\_CHAR** | Converts dates or numbers to character strings with optional formatting |
| **RR** | Century value depends on the specified year and the last two digits of the current year |
| **TO\_NUMBER** | Converts a character string containing digits to a number with optional formatting |
| **DD** | Numeric day of the month |
| **TO\_DATE** | Converts a character string representing a date to a date value with optional formatting |

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.

SELECT last\_name, TO\_CHAR(birth\_date, 'Month DD, YYYY') AS birthday FROM employees;

1. Convert January 3, 04, to the default date format 03-Jan-2004.

SELECT TO\_DATE('January 3, 04', 'Month DD, RR') AS converted\_date

FROM dual;

1. 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.

SELECT 'The promotion began on the ' || TO\_CHAR(start\_date, 'fmDDth "of" Month YYYY') AS promo\_date

FROM f\_promotional\_menus

WHERE promo\_code = 110;

1. Convert today’s date to a format such as: “Today is the Twentieth of March, Two Thousand Four”

SELECT 'Today is the ' || TO\_CHAR(SYSDATE, 'fmDDth "of" Month, "Two Thousand" YYYY') AS formatted\_date

FROM dual;

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

SELECT employee\_id, first\_name || ' ' || last\_name AS name, TO\_CHAR(salary, '$9999.99') AS salary

FROM employees;

1. Ellen Abel is an employee who has received a $2,000 raise. Display her first name and last name, her current salary, and her new salary. Display both salaries with a $ and two decimal places. Label her new salary column AS New Salary.

SELECT first\_name, last\_name,

TO\_CHAR(salary, '$9999.99') AS current\_salary,

TO\_CHAR(salary + 2000, '$9999.99') AS new\_salary

FROM employees

WHERE first\_name = 'Ellen' AND last\_name = 'Abel';

1. On what day of the week and date did Global Fast Foods’ promotional code 110 Valentine’s Special begin?

SELECT TO\_CHAR(start\_date, 'Day, DD-Mon-YYYY') AS start\_day

FROM f\_promotional\_menus

WHERE promo\_code = 110;

1. Create one query that will convert 25-Dec-2004 into each of the following (you will have to convert  
   25-Dec-2004 to a date and then to character data):  
   December 25th, 2004  
   DECEMBER 25TH, 2004  
   25th december, 2004

SELECT TO\_CHAR(TO\_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'Month DDth, YYYY') AS format1,

TO\_CHAR(TO\_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'MONTH DDth, YYYY') AS format2,

TO\_CHAR(TO\_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'DDth month, YYYY') AS format3

FROM dual;

1. Create a query that will format the DJs on Demand d\_packages columns, low-range and high- range package costs, in the format $2500.00.

SELECT TO\_CHAR(low\_range, '$9999.99') AS low\_range\_cost,

TO\_CHAR(high\_range, '$9999.99') AS high\_range\_cost

FROM d\_packages;

1. Convert JUNE192004 to a date using the fx format model.

SELECT TO\_DATE('JUNE192004', 'fxMonthDDYYYY') AS converted\_date

FROM dual;

1. What is the distinction between implicit and explicit datatype conversion? Give an example of each.

**Implicit conversion** happens automatically when needed by Oracle (e.g., when comparing a number to a string that can be interpreted as a number).

SELECT \* FROM employees WHERE salary = '5000';

**Explicit conversion** requires functions like TO\_CHAR, TO\_NUMBER, or TO\_DATE.

SELECT TO\_NUMBER('5000') + 100 FROM dual;

1. Why is it important from a business perspective to have datatype conversions?

Datatype conversion is important because it allows the integration of data from different sources, ensures accurate data manipulation, and helps in formatting data for reports, making the information easier to understand and use in decision-making processes.

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

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **NVL** | Converts nulls to an actual value |
| **COALESCE** | Returns the first non-null expression in the list |
| **NVL2** | Examines the first expression; if the first expression is not null, it returns the second expression; if the first expression is null, it returns the third expression |
| **NULLIF** | Compares two expressions; if they are equal, the function returns null; if they are not equal, the function returns the first expression |

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.

SELECT promotional\_name,

start\_date,

NVL(end\_date, TO\_CHAR(SYSDATE, 'YYYY-MM-DD')) AS end\_date

FROM f\_promotional\_menus;

1. Not all Global Fast Foods staff members receive overtime pay. Instead of displaying a null value for these employees, replace null with zero. Include the employee’s last name and overtime rate in the output. Label the overtime rate as “Overtime Status”.

SELECT last\_name,

NVL(overtime\_rate, 0) AS "Overtime Status"

FROM employees;

1. 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.

SELECT last\_name,

NVL(overtime\_rate, 5.00) AS "Overtime Rate"

FROM employees;

1. 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.

SELECT last\_name,

NVL(manager\_id, 9999) AS "Manager ID"

FROM employees;

1. Which statement(s) below will return null if the value of v\_sal is 50?  
   a. SELECT nvl(v\_sal, 50) FROM emp;  
   b. SELECT nvl2(v\_sal, 50) FROM emp;  
   c. SELECT nullif(v\_sal, 50) FROM emp;  
   d. SELECT coalesce (v\_sal, Null, 50) FROM emp;

C. SELECT nullif(v\_sal, 50) FROM emp;

1. What does this query on the Global Fast Foods table return?  
   SELECT COALESCE(last\_name, to\_char(manager\_id)) as NAME  
   FROM f\_staffs;

This query returns a list of names from the f\_staffs table. If a staff member has a last\_name, it returns that. If the last\_name is missing (NULL), it returns the manager\_id as a string in its place.

1. 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).  
   b. Modify the report to display null if the month of hire is September. Use the NULLIF function.
2. SELECT first\_name,

last\_name,

TO\_CHAR(hire\_date, 'Month') AS month\_of\_hire

FROM employees;

1. SELECT first\_name,

last\_name,

NULLIF(TO\_CHAR(hire\_date, 'Month'), 'September') AS month\_of\_hire

FROM employees;

1. For all null values in the specialty column in the DJs on Demand d\_partners table, substitute “No Specialty.” Show the first name and s

SELECT first\_name,

NVL(specialty, 'No Specialty') AS specialty

FROM d\_partners;

**Database Programming with SQL  
5-3: Conditional Expressions  
Practice Activities**

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **DECODE** | Compares an expression to each of the search values |
| **conditional statement** | An if-then-else expression whose value depends on the truth-value of a Boolean expression. |
| **CASE** | Implements conditional processing within a SQL statement; it meets the ANSI standard. |

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”.

SELECT

CASE

WHEN duration = 2 THEN 'shortest'

WHEN duration = 10 THEN 'longest'

ELSE TO\_CHAR(duration)

END AS "Play Times"

FROM d\_songs;

1. 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.

SELECT

department\_id,

last\_name,

salary,

CASE

WHEN department\_id = 10 THEN 1.25 \* salary

WHEN department\_id = 90 THEN 1.5 \* salary

WHEN department\_id = 130 THEN 1.75 \* salary

ELSE salary

END AS "New Salary"

FROM employees;

1. 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.

SELECT

first\_name,

last\_name,

manager\_id,

commission\_pct,

CASE

WHEN manager\_id IS NULL THEN NVL(commission\_pct, 99999)

ELSE manager\_id

END AS "Review"

FROM employees

WHERE department\_id IN (80, 90);