Database Programming with SQL  
**5-1: Conversion Functions**  
Vocabulary

* **CHAR** - Used for text and character data of fixed length, including  
  numbers, dashes, and special characters.
* **LTRIM** or **RTRIM** - 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 global\_fast\_food\_employees;**  
2. Convert January 3, 04, to the default date format 03-Jan-2004.

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

**FROM dual;**

3. 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, 'fmDay') || ' of ' || TO\_CHAR(start\_date, 'Month YYYY')**

**FROM f\_promotional\_menus**

**WHERE promo\_code = 110;**  
4. 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, 'fmDay') || ' of ' || TO\_CHAR(SYSDATE, 'Month YYYY')**

**FROM dual;**  
5. 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 full\_name, TO\_CHAR(salary, '$9999.99') AS salary**

**FROM global\_fast\_food\_employees;**  
6. 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 global\_fast\_food\_employees**

**WHERE first\_name = 'Ellen' AND last\_name = 'Abel';**  
7. 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'), TO\_CHAR(start\_date, 'DD-Mon-YYYY') AS valentine\_special\_start**

**FROM f\_promotional\_menus**

**WHERE promo\_code = 110;**

8. 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'), 'fmMonth DDth, YYYY') AS format2,**

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

**FROM dual;**

9. 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;**  
10. Convert JUNE192004 to a date using the fx format model.

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

**FROM dual;**  
11. What is the distinction between implicit and explicit datatype conversion? Give an example of each.

*Implicit:*

**SELECT '100' + 1 FROM dual;**

*Explicit:*

**SELECT TO\_NUMBER('100') + 1 FROM dual;**  
12. Why is it important from a business perspective to have datatype conversions?

Data-type conversion is crucial for ensuring that different types of data can be seamlessly integrated, validated, and processed in business applications. For example, it allows dates, numbers, and character data to be correctly interpreted in reports, financial calculations, and transaction records. This flexibility improves data quality and consistency across business systems.

**5-2: NULL Functions**  
Vocabulary

* **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 promo\_name, start\_date,**

**NVL(TO\_CHAR(end\_date, 'fmMonth DD, YYYY'), TO\_CHAR(SYSDATE, 'fmMonth DD, YYYY')) AS end\_date**

**FROM f\_promotional\_menus;**  
2. 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 global\_fast\_food\_employees;**3. 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 global\_fast\_food\_employees;**4. 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 global\_fast\_food\_employees;**  
5. Which statement(s) below will return null if the value of v\_sal is 50?

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

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

**-It will return the last name if it is not null**

**-If the last name is null, it will return the manager ID converted to a string**

**-If both are null, it returns null**  
7.  
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).

**SELECT first\_name, last\_name, TO\_CHAR(hire\_date, 'Month') AS month\_of\_hire**

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

**SELECT first\_name, last\_name, NULLIF(TO\_CHAR(hire\_date, 'Month'), 'September') AS month\_of\_hire**

**FROM employees;**  
8. 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;**

**5-3: Conditional Expressions**  
Vocabulary

* **DECODE** - Compares an expression to each of the search values
* **CASE** - 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 song\_name,**

**CASE**

**WHEN duration = 2 THEN 'shortest'**

**WHEN duration = 10 THEN 'longest'**

**ELSE TO\_CHAR(duration)**

**END 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.

**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;**  
3. 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);**