Ryan Li
Database Programming with SQL

5-1: Conversion Functions

Vocab

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
TO_NUMBER	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.
SYSDATE	Used for date and time values.
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. SELECT last_name,
   TRIM(TO_CHAR(birth_date, 'Month DD, YYYY')) AS formatted_birth_date
   FROM global_fast_food_employees;

2. SELECT TO_CHAR(TO_DATE('January 3, 04', 'Month DD, YY'), 'DD-Mon-YYYY') AS formatted_date

```
    FROM dual;
3.  SELECT 'The promotion began on the ' ||
    TO_CHAR(start_date, 'FMDDth') ||' of ' ||
    TO_CHAR(start_date, 'Month YYYY') AS promotion_message
    FROM f_promotional_menus
    WHERE promo_code = 110;

4.  SELECT 'Today is the ' || TO_CHAR(SYSDATE, 'FMDDth') || ' of ' ||
    TO_CHAR(SYSDATE, 'Month') || ', Two Thousand ' || TO_CHAR(SYSDATE, 'YYYY') AS
    formatted_today FROM dual;

5.  SELECT employee_id,
    first_name || ' ' || last_name AS employee_name,
    TO_CHAR(salary, '$999,999.99') AS formatted_salary
    FROM global_fast_food_employees;

6.  SELECT first_name,
    last_name,
    TO_CHAR(salary, '$999,999.99') AS current_salary,
    TO_CHAR(salary + 2000, '$999,999.99') AS new_salary
    FROM global_fast_food_employees
    WHERE first_name = 'Ellen' AND last_name = 'Abel';

7.  SELECT TO_CHAR(start_date, 'Day, DD-Mon-YYYY') AS promo_start_date
    FROM f_promotional_menus
    WHERE promo_code = 110;

8.  SELECT TO_CHAR(TO_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'Month DDth, YYYY')
    AS
    "December 25th, 2004",
    TO_CHAR(TO_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'FMMONTH DDth, YYYY') AS
    "DECEMBER 25TH, 2004",
    TO_CHAR(TO_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'DDth fmmonth, YYYY') AS
    "25th december, 2004"
    FROM dual;

9.  SELECT package_name,
    TO_CHAR(low_range, '$9999.99') AS formatted_low_range,
    TO_CHAR(high_range, '$9999.99') AS formatted_high_range
    FROM d_packages;

10. SELECT TO_DATE('JUNE192004', 'FXMONTHDDYYYY') AS formatted_date
    FROM dual;
```

11. SELECT '100' + 50 FROM dual;
12. From a business perspective, it is important to have data type conversions because it allows for more accurate and consistent data input within a business system. It allows for seamless and more effective queries, aggregation, and analysis of data.

<u>5-2: NULL Functions</u>

COALESCE	Converts nulls to an actual value
COALESCE	Returns the first non-null expression on the list
ISNULL	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.  SELECT
            promo_name AS "Promotional Name",
            start_date AS "Start Date",
            COALESCE(end_date, CURRENT_DATE) AS "End Date"
    FROM
            f_promotional_menus
    WHERE
            end_date IS NOT NULL
            OR CURRENT_DATE + INTERVAL '14' DAY = CURRENT_DATE;


2.  SELECT
            last_name AS "Last Name",
            NVL(overtime_rate, 0) AS "Overtime Status"
    FROM
            Employees;

3.  SELECT
            last_name AS "Last Name",
            NVL(overtime_rate, 5.00) AS "Overtime Rate"
    FROM
            Employees;
```

4.
   ```
   SELECT
           last_name AS "Last Name",
           NVL(manager_id, 9999) AS "Manager ID"
   FROM
           Employees;
   ```

5. c. `SELECT NULLIF(v_sal, 50) FROM emp;`

6.
   ```
   SELECT COALESCE(last_name, to_char(manager_id)) as NAME
   FROM f_staffs;
   ```

7. A.
```
   SELECT
           first_name AS "First Name",
           last_name AS "Last Name",
           TO_CHAR(hire_date, 'Month') AS "Month of Hire"
   FROM
           Employees;
   ```

8.
   ```
   SELECT
           first_name AS "First Name",
           NVL(specialty, 'No Specialty') AS "Specialty"
   FROM
           d_partners;
   ```

5-3: Conditional Expressions

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.
   ```
   SELECT CASE
           WHEN duration = 2 THEN 'shortest'
           WHEN duration = 10 THEN 'longest'
   ```

```
        ELSE TO_CHAR(duration)
        END AS "Play Times"
    FROM
        D_songs;

2.  SELECT
        department_id,
        last_name,
        salary,
        CASE
            WHEN department_id = 10 THEN salary * 1.25
            WHEN department_id = 90 THEN salary * 1.5
            WHEN department_id = 130 THEN salary * 1.75
```