SQL Functions

SQL Functions

- Take input (arguments), define logic (executable statements), and produce output
- Readability and modularity -> Maintainability
- Function can appear anywhere in SQL statement
- Two types: User defined and System defined
- Example
  - System defined -> ABS(n) Returns the absolute value of a number

  ```
  SELECT ABS(10) FROM DUAL; Answer: 10
  ```

  - System defined -> ROUND(n, precision) Rounds a value to a specified precision

  ```
  SELECT P_CODE, P_Price, ROUND(P_Price, 1) AS PRICE1,
                          ROUND(P_Price, 0) AS PRICE0
  FROM Product;
  ```

  - CEIL, FLOOR, etc

System Defined

- String Functions
  - String Manipulation: SUBSTR(), STRCMP(), …
  - Concatentaion: CONCAT()
  - Length of String: LENGTH()
  - LOWER/UPPER: LOWER() and UPPER()
- Example
  - SELECT CONCAT(‘Dan’, ‘ Morgan’) FROM DUAL;

  Answer: Dan Morgan
  - SELECT LENGTH(‘Dan’) FROM DUAL;

  Answer: 3

System Defined

- Conversion Functions -> Take a value of a given datatype and convert it to the equivalent value in another datatype
- Example:
  - TO_CHAR

  Takes a date value and converts it to character string
  - TO_DATE

  Takes character string and converts it to a date format

User Defined

- Created/Implemented by programmer
- Can manipulate data values
  - Reverse a string: Mary Jones -> Senoj yram

```
SELECT name, reverse_name(name)
From Professors;
```

- Can extend SQL where activities are too complex
  - Calculate how long an employee has been working for a business, rounded to a whole number of months

```
SELECT eid, how_many_months(hire_date)
FROM Employee;
```

User Defined

- Example -> Calculate tax for an employee

```
CREATE FUNCTION tax(P_value IN Number)
RETURN Number IS
BEGIN
   RETURN(P_value*0.08);
END;

SELECT eid, name, salary, tax(salary)
From Employee
WHERE dept_id=50;
```

User Defined

- Can be used in:
  - SELECT target_list
  - Conditional expression in WHERE/HAVING clauses
  - ORDER BY or GROUP BY
  - VALUE of the INSERT statement
  - SET clause of UPDATE statement
  - Can be used anywhere we have value/expression

```
SELECT eid, tax(salary)
FROM Employee
WHERE tax(salary) >
                    (SELECT MAX(tax(salary)) FROM Employee
                     WHERE dept_id=20)
ORDER BY tax(salary) DESC;
```

User Defined

- Syntax

```
CREATE FUNCTION <func_name>
 (<param_name1> IN <data type>, <param_name2> IN <data type>,
   ……………..)
RETURN <function return value data type> IS
<variable declaration>
BEGIN
   Executable commands;
   RETURN (return value)
   ….
   [Exception Exception handlers]
END
```

User Defined

- Example

```
CREATE FUNCTION query_max_sal(P_dept_id IN Number)
RETURN Number IS
v_num NUMBER;
BEGIN
        SELECT MAX(Salary) INTO v_num
        FROM Employee
        WHERE dept_id=P_dept_id;
        RETURN(v_num);
END;
UPDATE employee SET salary=query_max_sal(dept_id)
WHERE eid=174);
```