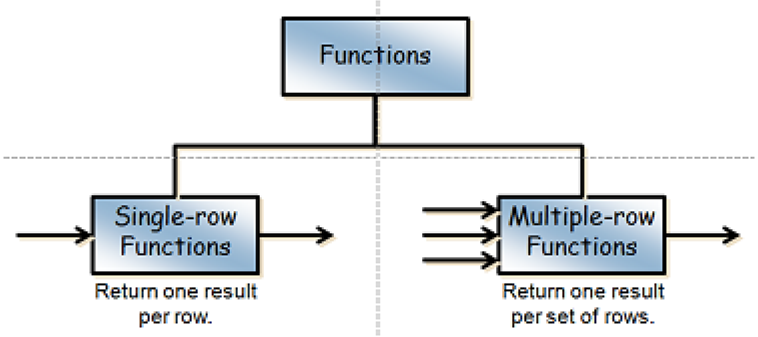
Chapter 4
Functions

Objectives

- Describe the various types of functions available in SQL
- Use character, number, and date functions in SELECT statements
- Use conversion functions

Functions

- Are programs that take zero or more arguments and return a single value.
- Oracle has a number of built-in functions that can be called from SQL statements
- Five significant classes of functions
  - Single row
  - Aggregate, multiple-row or group functions
  - Analytical functions
  - Object-reference functions
  - Programmer-defined functions


Functions
Single-row Functions
Return one result per row.
Multiple-row Functions
Return one result per set of rows.

- Will concentrate on single-row functions now.

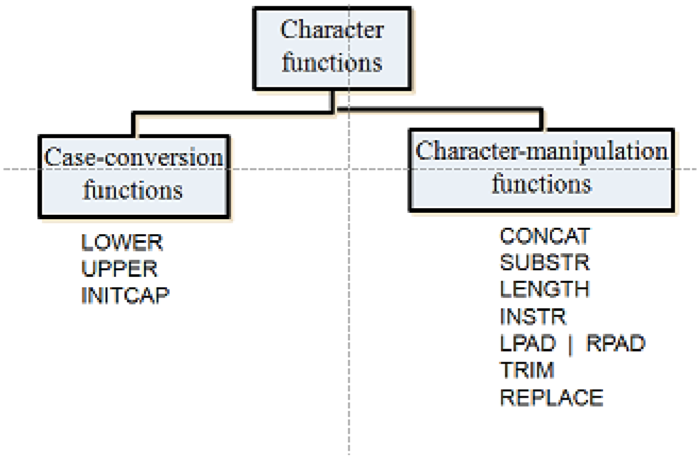
Single-Row Functions

- Can be incorporated into SQL (and PL/SQL).
- Can be used in the SELECT, WHERE and ORDER BY clauses of the SELECT statements.
- Manipulate data items
- Accept arguments and return one value
- Act on each row that is returned
- Return one result per row
- May modify the data type
- Can be nested
- Accept arguments that can be a column or an expression

Five main types:

1. Character functions
2. Numeric functions
3. Conversion functions
4. Date functions (Chapter 4)
5. Regular expression functions (New to 10g and expanded in 11g)

Character functions



- Accept character input
- Example:

```
SELECT ename, job
FROM emp
WHERE UPPER(ename) LIKE 'S%' ;
```

- Example 2:

```
SELECT INITCAP(ename), INITCAP(job)
FROM emp
WHERE UPPER(ename) LIKE 'S%' ;
```

Case-conversion

- Can combine functions

```
SELECT INITCAP(ename), INITCAP(SUBSTR(job,1,5))
FROM emp
WHERE UPPER(ename) LIKE 'S%' ;
```

Character-manipulation

Numeric functions

- Use these functions to perform calculations.
- Accept an input number
- Some examples:

  - ROUND: Rounds value to a specified decimal
  - TRUNC: Truncates value to a specified decimal
  - MOD: Returns remainder of division

- Example:
  - Want to get an average hourly salary for SCOTT worker King.
  - Want the results rounded to the nearest whole dollar amount.
  - Have a monthly salary.
  - Assume 37.5 hrs/week and 4.5 weeks/month.

```
SELECT INITCAP(ename), ROUND((sal / (37.5 * 4.5)),2)
FROM emp
WHERE UPPER(ename) LIKE 'KING' ;

SELECT INITCAP(ename), ROUND((sal / (37.5 * 4.5)))
FROM emp
WHERE UPPER(ename) LIKE 'KING' ;

SELECT INITCAP(ename), TRUNC(sal / (37.5 * 4.5))
FROM emp
WHERE UPPER(ename) LIKE 'KING' ;
```

- Working with NULLs
  - NULL values represent unknown or a lack of data.
  - Any arithmetic operation on a NULL value results in a NULL.

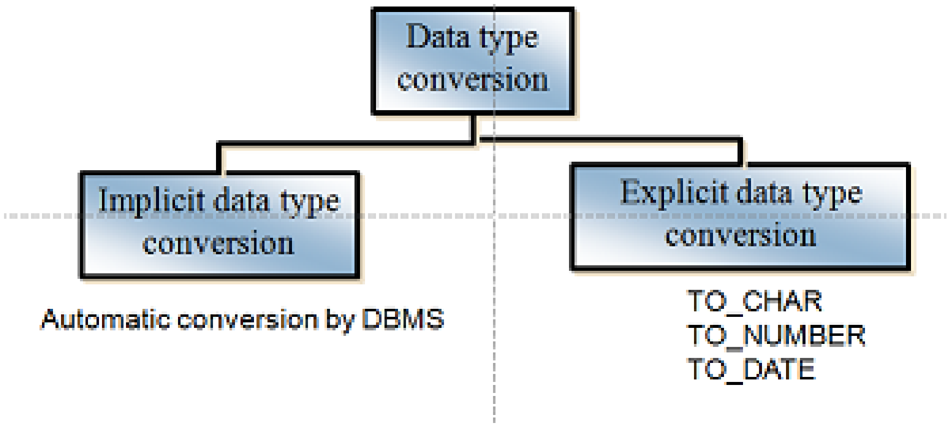
- NVL Function
  - Converts a null value to an actual value
    - Data types that can be used are date, character, and number
    - Data types must match
      - NVL(commission_pct,0)
      - NVL(hire_date, ‘01-JAN-97’)
      - NVL(job_id, ‘No job yet’)

- NULL Value (NVL) Example
  - A company rule states that consultants are required to bill out at least as much as their salary. Sales people need to sell as least as much as their salary times their commission. You need to write a query that will return those amounts. (Assume sales people are those with commissions.)

```
SELECT last_name,
   salary,
   commission_pct,
   salary + (salary * commission_pct) "Needs to earn"
FROM employees
WHERE UPPER(last_name) LIKE 'T%';

SELECT last_name,
   salary,
   commission_pct,
   salary + (salary * NVL(commission_pct,0)) "Needs to earn"
FROM employees
WHERE UPPER(last_name) LIKE 'T%';
```

Conversion functions


Data type conversion
Implicit data type conversion
Explicit data type conversion
Automatic conversion by DBMS
TO_CHAR
TO_NUMBER
TO_DATE

- Can apply special formatting to Dates and to Numbers

- TO_CHAR with dates
  - The format model:
  - Must be enclosed with single quotes
  - Is case-sensitive
  - Can include any valid data format element
  - Has an fm element to remove padded blanks or suppress leading zeros
  - Is separated from the date value by a comma
  - TO_CHAR (date, 'format model')

- Date example:

```
SELECT TO_CHAR(SYSDATE,'"Today is the "Ddspth" day of "Month" of
  "Year')
FROM dual;

SELECT LAST_NAME,
  HIRE_DATE,
  TO_CHAR(HIRE_DATE, 'FMMonth DD YYYY')  "Date Started"
FROM EMPLOYEES
WHERE DEPARTMENT_ID = 100
ORDER BY LAST_NAME;
```

Number example:

- Converting numbers to characters

  ```
  SELECT 'Guess what... ' || INITCAP(ename) || ' earns ' ||
    TO_CHAR(sal, 'fm$99,999.00') || ' a year!'
  FROM emp
  WHERE UPPER(first_name) LIKE 'S%';
  ```

- Converting characters to numbers

  ```
  SELECT CITY,
    POSTAL_CODE "Old Code",
    TO_NUMBER(POSTAL_CODE) + 1 "New Code"
  FROM LOCATIONS
  WHERE COUNTRY_ID = 'US'
  ORDER BY POSTAL_CODE;
  ```

Regular expressions

- Is a pattern you use to match against a string.
- These functions are new to Oracle 10g
- Use with extreme caution

- For example, find all employees hired in the years 1980, 1981 and 1982.

  ```
  SELECT INITCAP(ename), hiredate
  FROM emp
  WHERE REGEXP_LIKE(TO_CHAR(hiredate, 'YYYY'), '^198[0-2]$');
  ```

SOME ADVANCED QUERIES
Conditional Expressions
(Chapter 7)

DECODE Expression

- The DECODE function substitutes values based on a condition using IF..THEN..ELSE logic.
- If a value is equal to another value, then the substitution value is returned.
- A default value is also permitted.

```
SELECT losal,hisal,
DECODE (grade, 1, 'Bottom of the heap',
   2, 'A bit better',
   3, 'Middle of the pack',
   4, 'On the way up',
   5, 'Top of the heap') Description
FROM salgrade;
```

CASE Expressions

- A CASE expression also allows you to use IF..THEN..ELSE logic in SQL statements without have to invoke procedures.
- Are similar to the Oracle DECODE function, but they offer more flexibility and logical power.
- Also, tend to be easier to read than the DECODE function.
- New to Oracle 9i.

CASE Expression Example

- You need to display the last name and the commission percentage of all employees, and whether their commission is low, average, or high.
  - Low = 0.1
  - Average = 0.15
  - High = 0.2
  - Anything else will be “N/A”

```
SELECT last_name, commission_pct,
    (CASE commission_pct
          WHEN 0.1 THEN 'Low'
          WHEN 0.15 THEN 'Average'
          WHEN 0.2 THEN 'High'
          ELSE 'N/A'
    END) Commission
FROM employees
ORDER BY last_name;
```

```
SELECT last_name, commission_pct,
   (CASE      WHEN commission_pct<=0.1 THEN 'Low'
        WHEN commission_pct<=0.15 THEN 'Average'
        WHEN commission_pct<=0.2 THEN 'High'
        WHEN commission_pct>0.2 THEN 'Way High'
        ELSE 'N/A'
   END) Commission
FROM employees
ORDER BY last_name;
```

Summary

- Examined single-row functions
- A single-row function returns a value for each row as it is retrieved from the table.
- Can be used in the SELECT, WHERE and ORDER BY clauses of SELECT statements.