

FUNCTION

Function

❖ Function

- ✓ 입력 데이터를 이용해서 연산을 수행한 후 출력 값을 만들어 내는 개체
- ✓ 함수에 입력하는 데이터를 Argument(Parameter, 매개변수, 인수)라 하고 출력을 Return Data 라고도 함



Function

❖ Function

✓ 함수의 종류

● 매개변수의 데이터에 따른 분류

- ◆ Scala 함수: 값 하나를 계산하는 함수
- ◆ Group 함수: 여러 개의 값으로부터 통계 값을 계산하는 함수

● 기능에 따른 분류

- ◆ 문자 함수(String Functions): 문자열을 다루어 연산하고 문자열을 반환
- ◆ 날짜와 시간 함수(Date & Time Functions): 날짜와 시간을 연산하고 결과를 문자열 형식으로 반환
- ◆ 수치 함수(Numeric Functions): 수치 데이터를 다루어 연산하고 결과를 수치 데이터로 반환
- ◆ 제어 함수(Control Flow Functions): 조건에 따라 데이터를 연산하거나 처리 순서를 제어하는데 특별히 반환하는 값은 없음
- ◆ 집계 함수(Aggregate Functions): GROUP BY 절과 함께 사용하여 그룹 별 집계 결과를 반환
- ◆ 정보 함수(Information Functions): 데이터베이스의 각종 정보를 반환
- ◆ 암호 함수(Encryption Functions): 데이터를 암호화하거나 복호화하여 결과를 반환
- ◆ 기타 함수: NULL 처리나 데이터 타입을 변경하기 위한 함수

Scala Function

❖ 문자열 함수

- ✓ ASCII(문자), CHAR(숫자), CHR(숫자): 문자의 ASCII 값을 돌려주거나 숫자의 ASCII 코드 값에 해당하는 문자를 반환하는 함수
- ✓ HEX(숫자 | 문자열 | 변수 | 열), UNHEX(16진수 문자열): 16진수 문자열 변환
- ✓ BIT_LENGTH(문자열), CHAR_LENGTH(문자열), CHARACTER_LENGTH(문자열), LENGTH(문자열), LENGTHB(문자열): 할당된 Bit 또는 Byte 또는 문자 개수
- ✓ CONCAT(문자열1, 문자열2 ...), CONCAT_WS(구분자, 문자열1, 문자열2,...): 문자열 결합
- ✓ ELT(위치, 문자열1, 문자열2), FIELD(찾을 문자열, 문자열1, 문자열2, ..), FIND_IN_SET(찾을 문자열, 문자열 리스트), INSTR(기준 문자열, 부분 문자열), LOCATE(부분 문자열, 기준 문자열)
 - ELT()는 문자열 중에서 위치 번째에 해당하는 문자열을 반환
 - FIELD()는 문자열 중에서 찾을 문자열의 위치를 찾아서 반환하는데 매치되는 문자열이 없으면 0을 반환
 - FIND_IN_SET()은 찾을 문자열을 문자열 리스트에서 찾아서 위치를 반환하는데 문자열 리스트는 콤마(,)로 구분되어 있어야 하며 공백이 없어야 함
 - INSTR()은 기준 문자열에서 부분 문자열을 찾아서 그 시작 위치를 반환
 - LOCATE ()는 INSTR()과 동일하지만 입력하는 데이터의 순서가 반대로 되어 있음

Scala Function

❖ 문자열 함수

- ✓ FORMAT(숫자, 소수점 자릿수): 숫자를 소수점 아래 자릿수까지 표현하는데 1000 단위마다 , 를 표시
- ✓ BIN(숫자), HEX(숫자), OCT(숫자): 2진수, 16진수, 8진수를 리턴
- ✓ INSERT(기준 문자열, 위치, 길이, 삽입할 문자열): 기준 문자열의 위치부터 길이 만큼을 지우고 삽입할 문자열을 끼워 넣음
- ✓ LEFT(문자열, 길이), RIGHT(문자열, 길이): 문자열의 길이만큼 잘라서 반환
- ✓ UPPER(문자열), UCASE(문자열), LOWER(문자열), LCASE(문자열): 대소문자 변환
- ✓ LPAD(문자열, 길이, 채울 문자열), RPAD(문자열, 길이, 채울 문자열): 문자열에서 길이 만큼 채울 문자열로 채우는 함수
- ✓ LTRIM(문자열), RTRIM(문자열): 좌우 공백을 제거하는 함수
- ✓ TRIM(문자열), TRIM(방향 자를 문자열 FROM 문자열): 좌우 공백을 모두 제거하거나 방향 자를 문자열은 LEADING, BOTH, TRAILING
- ✓ REPEAT(문자열, 횟수): 문자열을 횟수만큼 반복
- ✓ REPLACE(문자열, 원래 문자열, 바꿀 문자열): 문자열을 치환
- ✓ REVERSE(문자열): 문자열을 반대로 반환
- ✓ SPACE(길이): 길이 만큼의 공백 반환

Scala Function

❖ 문자열 함수

- ✓ SUBSTRING(문자열, 시작 위치, 길이), SUBSTRING(문자열 FROM 시작 위치 FOR 길이): 시작 위치부터 길이만큼 문자를 반환
- ✓ SUBSTRING_INDEX(문자열, 구분자, 횟수): 문자열에서 구분자가 왼쪽부터 횟수 번째 나오면 그 이후의 오른쪽은 버리는데 횟수가 음수면 오른쪽부터 세고 왼쪽을 버림



Scala Function

❖ 문자열 함수

- ✓ usertbl 테이블에서 name 과 birthyear를 연결해서 조회

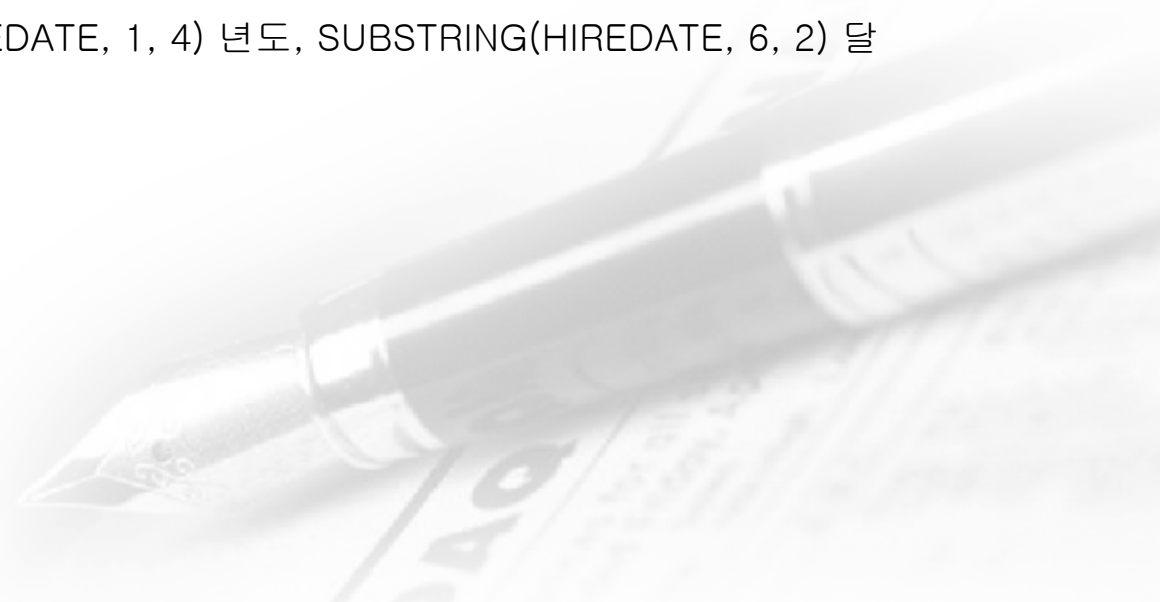
```
SELECT concat(name, birthyear)
FROM usertbl;
```

- ✓ usertbl 테이블에서 name 과 name의 길이를 조회

```
SELECT name, length(name)
FROM usertbl;
```

- ✓ 직원들의 입사 년도만 출력하려면 SUBSTRING 함수를 이용해서 입사일을 저장하고 있는 HIREDATE 컬럼에서 첫 글자부터 4개를 추출하고 입사한 달만 출력하려면 HIREDATE 컬럼에서 여섯번째 글자부터 2개를 추출

```
SELECT SUBSTRING(HIREDATE, 1, 4) 년도, SUBSTRING(HIREDATE, 6, 2) 달
FROM EMP;
```



Scala Function

❖ 문자열 함수

✓ 연습문제

- EMP Table에서 HIREDATE의 년도가 1981인 사원의 모든 정보를 조회(단 SUBSTRING 함수를 이용해서 WHERE절을 작성)
- EMP Table에서 ENAME이 E로 끝나는 사원의 모든 정보를 조회(단 SUBSTRING 함수를 이용해서 WHERE절을 작성)



Scala Function

❖ 수치 함수

- ✓ ABS(숫자): 절대 값을 반환
- ✓ ACOS(숫자), ASIN(숫자), ATAN(숫자), ATAN2(숫자1, 숫자2), SIN(숫자), COS(숫자), TAN(숫자): 삼각 함수 관련된 함수
- ✓ CEILING(숫자), FLOOR(숫자), ROUND(컬럼이름이나 표현식, 반올림할 자릿수)
 - 올림, 내림, 반올림을 수행
 - 소수 자릿수는 양수로 설정하면 되고 음수는 1의 자리부터 반올림
- ✓ CONV(숫자, 원래 진수, 변환할 진수): 숫자를 원래 진수에서 변환할 진수로 계산
- ✓ DEGREES(숫자), RADIANS(숫자), PI(): 라디안 값을 각도 값으로 각도 값을 라디안으로 그리고 파이 값을 반환
- ✓ EXP(X), LN(숫자), LOG(숫자), LOG(밑수, 숫자), LOG2(숫자), LOG10(숫자): 지수와 로그와 관련된 함수
- ✓ MOD(숫자1, 숫자2) 또는 숫자1 % 숫자2 또는 숫자1 MOD 숫자2: 나머지를 구해주는 함수
- ✓ POW 또는 POWER(숫자1, 숫자2), SQRT(숫자): 거듭 제곱 값과 제곱근 리턴
- ✓ SIGN(숫자): 부호를 반환해주는 함수
- ✓ TRUNCATE(숫자, 정수): 숫자를 버리는 함수
- ✓ LEAST(값 나열), GREATEST(값 나열): 가장 작거나 큰 값을 리턴
- ✓ RAND(): 0이상 1미만의 실수를 리턴

Scala Function

❖ 수치 함수

- ✓ EMP Table에서 EMPNO가 홀수인 데이터의 모든 컬럼을 검색

```
SELECT *  
FROM EMP  
WHERE MOD(EMPNO, 2) = 1;
```



Scala Function

❖ 날짜 함수

✓ 현재 날짜 및 시간

- CURRENT_DATE(), CURDATE() -> 현재 날짜를 년-월-일 형태로 리턴
- CURRENT_TIME(), CURTIME(), LOCALTIME() -> 현재 시간을 시:분:초 형태로 리턴
- NOW(), LOCALTIME(), LOCALTIMESTAMP(), SYSDATE(), CURRENT_TIMESTAMP() -> 현재 날짜 및 시간

✓ ADDTIME(날짜/시간, 시간), SUBTIME(날짜/시간, 시간): 날짜/시간을 기준으로 시간을 더하거나 뺀 결과를 구함

- SELECT ADDTIME('2025-01-01 23:59:59', '1:1:1'), ADDTIME('15:00:00', '2:10:10');
- SELECT SUBTIME('2025-01-01 23:59:59', '1:1:1'), SUBTIME('15:00:00', '2:10:10');



Scala Function

❖ 날짜 함수

- ✓ ADDDATE(날짜, 차이), SUBDATE(날짜, 차이): 날짜를 기준으로 차이를 더하거나 뺀 날짜를 구함
 - SELECT ADDDATE('2025-01-01', INTERVAL 31 DAY), ADDDATE('2025-01-01', INTERVAL 1 MONTH);
 - SELECT SUBDATE('2025-01-01', INTERVAL 31 DAY), SUBDATE('2025-01-01', INTERVAL 1 MONTH);
 - 단위
 - ◆ YEAR 년
 - ◆ MONTH 월
 - ◆ DAY 일
 - ◆ HOUR 시
 - ◆ MINUTE 분
 - ◆ SECOND 초
 - ◆ WEEK 주
 - ◆ DAY_HOUR '일 시간' 조합 (예: INTERVAL 3 12 DAY_HOUR은 3일 12시간)

Scala Function

❖ 날짜 함수

- ✓ YEAR(날짜), MONTH(날짜), DAY(날짜), HOUR(시간), MINUTE(시간), SECOND(시간), MICROSECOND(시간), DATE(날짜), TIME(날짜), DAYOFWEEK(날짜), MONTHNAME(날짜), DAYOFTYEAR(날짜): 날짜 및 시간에서 필요한 부분을 추출
- ✓ LAST_DAY(날짜): 주어진 날짜의 그 달의 마지막 날을 리턴
- ✓ MAKEDATE(연도, 정수): 연도에서 정수만큼 지난 날짜를 리턴
- ✓ MAKETIME(시, 분, 초): 시간 생성
- ✓ PERIOD_ADD(연월, 개월 수): 연월에서 개월만큼 지난 연월을 리턴
- ✓ PERIOD_DIFF(연월1, 연월2): 연월1 - 연월2 의 개월 수를 리턴
- ✓ QUARTER(날짜): 분기 리턴
- ✓ TIME_TO_SEC(시간): 시간을 초 단위로 리턴



Scala Function

❖ 날짜 함수

✓ 특정 날짜 생성

- STR_TO_DATE(날짜 문자열, 서식 문자열)

```
SELECT STR_TO_DATE('1986-05-05 11:00:00', '%Y-%m-%d %H:%i:%S');
```

✓ 날짜 간의 차이

- DATEDIFF(날짜 데이터, 날짜 데이터), TIMEDIFF(날짜 또는 시간, 날짜 또는 시간),
TIMESTAMPDIFF(단위, datetime_expr1, datetime_expr2)

```
SELECT DATEDIFF(CURRENT_DATE(), STR_TO_DATE('1986-05-05', '%Y-%m-%d'));
```

```
SELECT TIMESTAMPDIFF(MONTH, '1985-05-05', '2025-01-01');
```

✓ 날짜 데이터 연산

- 날짜 데이터 와 INTERVAL 정수 단위 형태의 덧셈과 뺄셈 가능

```
CURRENT_DATE() + INTERVAL 1 DAY : 현재 날짜에 하루를 추가
```

Scala Function

❖ 시스템 정보 함수

- ✓ ROW_COUNT(): 직전에 실행할 DML 구문에 영향받은 행의 개수 리턴
- ✓ DEFAULT(컬럼 이름): 컬럼에 설정된 기본값 리턴하는데 AUTO_INCREMENT가 설정된 경우는 0
- ✓ LAST_INSERT_ID(): AUTO_INCREMENT가 설정된 열이 있는 테이블에서 마지막으로 삽입된 데이터의 일련번호 값 리턴
- ✓ FOUND_ROWS(): 같이 수행된 SELECT 구문에서의 행의 개수 리턴
- ✓ DATABASE(): 사용 중인 데이터베이스 이름 리턴
- ✓ USER(): 사용 중인 유저 이름 리턴
- ✓ VERSION(): 데이터베이스 버전 리턴
- ✓ JSON_OBJECT(컬럼 이름 나열): JSON 문자열로 리턴

```
SELECT JSON_OBJECT('name', '김철수', 'age', 30, 'city', '서울');
```

- ✓ SLEEP(초)
 - ✓ ROWNUM(): 행 번호 리턴
- ```
SELECT ROWNUM(), ENAME
FROM EMP;
```

# Scala Function

## ❖ 제어 흐름 함수

### ✓ IF 함수

- 수식이 참 또는 거짓인지 결과에 따라서 2중 분기

```
SELECT IF (100>200, '참이다', '거짓이다');
```

### ✓ CASE ~ WHEN ~ ELSE ~ END

- CASE는 내장 함수는 아니며 Operator로 분류하기도 함
- DECODE\_ORACLE 이라는 함수를 이용해서도 작성 가능(MySQL에는 존재하지 않음)
- 다중 분기에 사용될 수 있음

```
SELECT
 CASE 10
 WHEN 1 THEN '일'
 WHEN 5 THEN '오'
 WHEN 10 THEN '십'
 ELSE '모름'
END;
```





# Scala Function

## ❖ NULL 관련 함수

- ✓ IFNULL(1번째 인자, 2번째 인자): 첫번째 인자 값이 NULL이 아니면 첫번째 인자값을 리턴하고 아닐 경우 2번째 인자값을 리턴 - NVL
  - SELECT IFNULL(10,20);
  - SELECT IFNULL(NULL,20);
  - SELECT IFNULL(10/'S',20);
- ✓ NULLIF(1번째 인자, 2번째 인자): 첫번째 인자 값과 두번째 인자 값이 같으면 NULL을 리턴하고 아닐 경우 첫번째 인자 값을 리턴
  - SELECT NULLIF(10,10);
  - SELECT NULLIF(10,NULL);
- ✓ NVL2(1번째 인자, 2번째 인자, 3번째 인자): 첫번째 인자 값이 NULL 이면 세번째 인자인 값3을 리턴하고 NULL이 아니면 두번째 인자 값을 리턴
  - SELECT NVL2(NULL,1, 2);
  - SELECT NVL2('X', 1, 2);
- ✓ SELECT COALESCE(인자 1, 2, 3...): 인자 값들 중 첫번째 NULL이 아닌 값을 리턴하고 NULL이 아닌 값이 없을 경우 NULL 리턴
  - SELECT COALESCE(NULL,1);

# Scala Function

## ❖ 타입 변환

✓ 묵시적 형 변환: 자동적으로 형 변환이 수행되는 것

- 숫자로 구성된 문자열을 산술 연산에 사용하는 경우 자동으로 숫자로 변환  
`SELECT '100' + '200';`
- 숫자로 시작하는 문자열을 산술 연산에 사용하는 경우 자동으로 숫자로 변환  
`SELECT '100' + '200GIGA';`
- 숫자로 시작하지 않는 문자열을 산술 연산에 사용하는 경우 0 으로 변환  
`SELECT '100' + 'GI200GA';`



# Scala Function

## ❖ 타입 변환

### ✓ 명시적 형 변환

- TO\_CHAR(날짜 또는 시간, 형식 문자열): 날짜 나 시간을 문자열로 변환
- CAST(데이터 AS 데이터 유형): 데이터를 데이터 유형으로 변환
  - ◆ 숫자를 문자로 변환  

```
SELECT CAST(1 AS CHAR(10)) AS testChar
```
  - ◆ 문자를 숫자로 변환  

```
SELECT CAST('1' AS SIGNED) AS testInt
```
- CONVERT(데이터, 데이터 유형): 데이터를 데이터 유형으로 변환
  - ◆ 숫자를 문자로 변환  

```
SELECT CONVERT(1, CHAR(10)) AS testChar;
```
  - ◆ 문자를 숫자로 변환  

```
SELECT CONVERT('1', SIGNED) AS testInt;
```
- MYSQL에서는 VARCHAR로는 형 변환이 불가