



Day25


📅 Date	@03/08/2022
🕒 작성일시	@2022년 8월 3일 오전 10:01
🔍 강의 번호	BD101
☰ 유형	Algorithm Data
👤 강사명	AustinYoon
☑ 강의자료	<input type="checkbox"/>
☑ 노션 복습	<input type="checkbox"/>
☑ 코딩 복습	<input type="checkbox"/>
☑ 주말숙제(교제)	<input type="checkbox"/>
☑ 정리	<input type="checkbox"/>

Day25

Aug

AlgorithmLab/Day25 at main · Kray273/AlgorithmLab
Contribute to Kray273/AlgorithmLab development by creating an account on GitHub.
<https://github.com/Kray273/AlgorithmLab/tree/main/Day25>

Kray273/
AlgorithmLab



1 Contributor
0 Issues
0 Stars
0 Forks

▼ 에라토스테네스의 체_Sieve of Eratosthenes

- 소수 prime number를 찾아내는 알고리즘

- 소수는 2 이상의 정수에서 1과 그 수 자체로만 나눌 수 있는 수
- 소수는 나열되어 있는 구간이 불규칙하므로 임의로 찾기가 힘들다.



소수는 2 이상의 정수 중에서 1과 그 수 자신 외에는 나눌수 없는 숫자.
10이하에서는 2,3,5,7이 소수에 해당한다.
소수 prime에서의 '소'는 합성되지 않은 소박한 숫자라는 뜻을 가진다.
모든 수의 소(근본)을 의미하기도 한다.

소수인지 아닌지를 구분하는 것은 의외로 어렵다.



소수의 내용만으로 보면 어렵지 않아보이지만 사실은 의외로 아주 어렵다. 무엇이 어려운지도 바로 찾아내기 어렵다.

예로 3의 배수는 3,6,9,12,15 처럼 3개의 간격으로 나열된다. 따라서 1에서 100까지의 사이에 있는 3의 배수를 찾는 일은 간단하다.

하지만 소수는 규칙성이 없기에(규칙이 있다고 생각하는게 리만가설) 간격이 불규칙하고 랜덤하다. 즉 소수를 한번에 열거하기가 어렵다.

• 소수를 찾아내는 방법

- 고대 그리스의 과학자인 에라토스테네스는 모든 숫자로 나눠서 소수를 찾는 방법을 개선하여 소수를 효율적으로 발견하는 방법을 알아냈다. 그의 이름을 따서 '에라토스테네스의 체'라고 부른다.

▼ Sieve of Eratosthenes

- 어떤 수 이하의 범위에 존재하는 모든 소수를 찾고 싶은 경우

‘그 수의 제곱근보다 작은 소수의 배수만 없애면 남은 소가 소수다’

라는 생각을 바탕으로 소수를 찾는 방법이다.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100



예를 들어

1. 100 이하의 소수를 모두 찾아내려면 먼저 100의 제

곱근 이하 소수를 선택한다.

2. 루트 100의 제곱근은 10이다. 즉 제곱하면 100이 되는 수는 10이다.

3. 10 이하에서 소수는 2,3,5,7, 네개이다.

☞ 우선 2에서 100까지의 표에서 2로 나눌 수 있는 수를 2를 제외하고 나머지를 모두 삭제한다.

맨 처음에 소수인 2를 발견한 후 2의 배수를 모두 지운다.

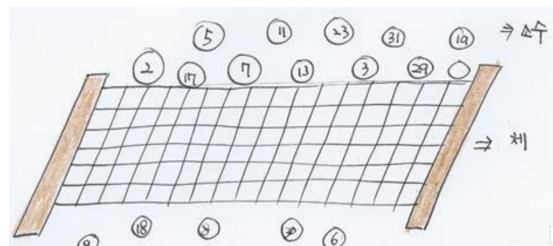
	2	3		5		7		9	
11		13		15		17		19	
21		23		25		27		29	
31		33		35		37		39	
41		43		45		47		49	
51		53		55		57		59	
61		63		65		67		69	
71		73		75		77		79	
81		83		85		87		89	
91		93		95		97		99	

이런 방식으로 3보다 큰 수에 대해서도 $n \times n$ 부터 지워주면 된다.

	2	3		5		7			
11		13				17		19	
		23		25				29	
31				35		37			
41		43				47		49	
		53		55				59	
61				65		67			
71		73				77		79	
		83		85				89	
91				95		97			

☞ 같은 방식으로 5와 7을 모두 제거하면 다음과 같이 된다.

☞ 이 방법을 사용하면 1에서 100까지의 모든 숫자를 해당 숫자보다 작은 수로 나눌 수 있는지 하나하나 전부 순서대로 확인하는 것보다는 훨씬 더 빨리 소수를 찾을 수 있다.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

▼ Algorithm

- 에라토스테네스의 체는 크게 3개의 처리로 구성한다.
 1. 어떤 수 이하의 모든 정수 데이터 준비
 2. 어떤 수의 제곱근 보다 작은 소수의 배수들을 차례로 제거한다.
 3. 마지막까지 남은 수들을 출력한다.



우선 10이하의 소수를 구하는 경우로 생각해보자.

1. 10이하의 정수 데이터를 준비한다.
2. 10의 제곱근 약 3.16보다 작은 소수의 배수를 차례대로 제거한다.
3. 마지막까지 남은 수들을 출력한다.

▼ (1) 10 이하의 정수 데이터들을 준비한다.

- 먼저 체의 대상이 되는 10까지의 정수를 데이터로 준비하자. 여기에서는 11개의 요소를 가지는 정수형 배열을 준비한다. 배열의 이름은 arr로 정하고, 첨자는 0부터 시작하기 때문에 첨자를 10까지 동일하게 하기 위해 요소를 11개를 준비한다.

0	1	2	3	4	5	6	7	8	9	10

- 요소는 그 첨자가 소수인지의 여부를 판정하는 데이터를 넣는 것으로 사용하자. 즉, 소수가 아닌 것으로 판정된 첨자의 요소에 '소수가 아니다'라는 것을 나타내는 데이터를 넣자. 다시말해 **소수의 가능성이 있는 경우에는 1을 대입하고, 소수가 아닌 경우에는 0을 대입하자.**
 - 위의 개념 소개에서 사용한 소수가 아닌 수를 제거하는 방식을 0을 대입하는 방식으로 처리한다.
 - 따라서, 초기값을 1로 전부 대입해둔다. 그리고 소수가 아닌 것으로 판정된 수(첨자)의 요소에는 0을 대입한다.
 - 이러한 방식을 통해 소수의 배수를 모두 제거(0을 대입한 요소들)한 후 '1'이 남아 있는 요소들의 첨자는 모두 소수이고 0이 대입되어 있는 요소의 첨자는 '소수가 아니다'라고 구별할 수 있다.

1	1	1	1	1	1	1	1	1	1	1
0	1	2	3	4	5	6	7	8	9	10

▼ (2) 10의 제곱근은 약 3.16이므로 제곱근의 이하의 소수 2와 3의 배수들을 모두 제거한다.

- 먼저 2의 배수들을 모두 지운다. 이때 지운다는 의미를 0을 대입한다는 표현으로 적용한다.

1	1	1	1	0	1	0	1	0	1	0
0	1	2	3	4	5	6	7	8	9	10

- 다음으로 3의 배수에 0을 대입한다.

1	1	1	1	0	1	0	1	0	0	0
0	1	2	3	4	5	6	7	8	9	10

▼ (3) 마지막까지 남은 수들을 출력한다.

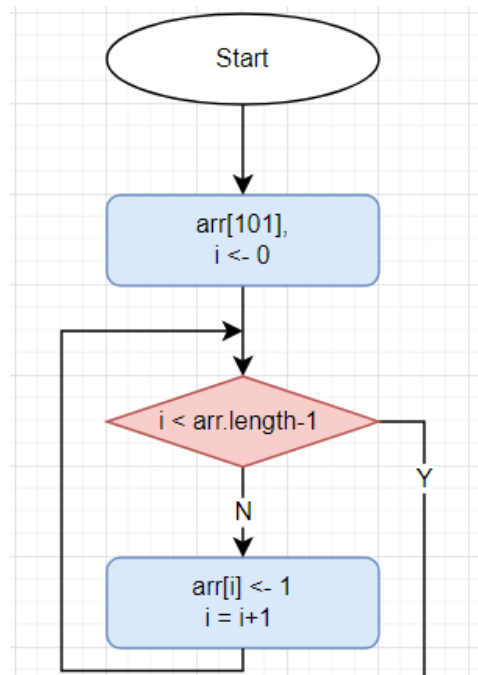
- 제곱근 이하의 소수들의 모든 배수를 제거(0을 대입)했다면 나머지는 모두 소수임으로 요소의 값이 '1'인 첨자는 소수이다.
- 첨자가 2이상이고 1이 들어 있는 요소들의 첨자들만 뽑아낸다.

1	1	1	1	0	1	0	1	0	0	0
0	1	2	3	4	5	6	7	8	9	10

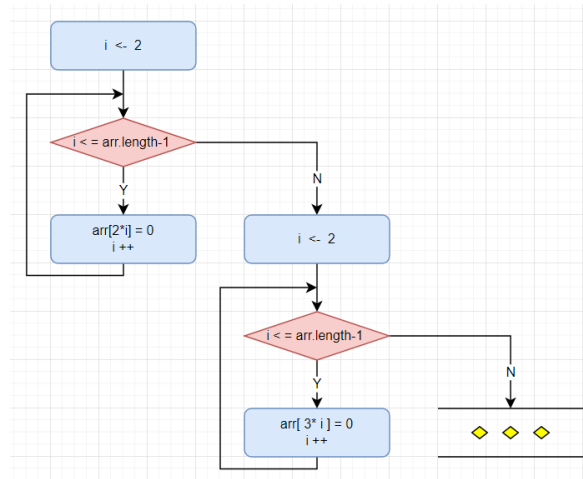
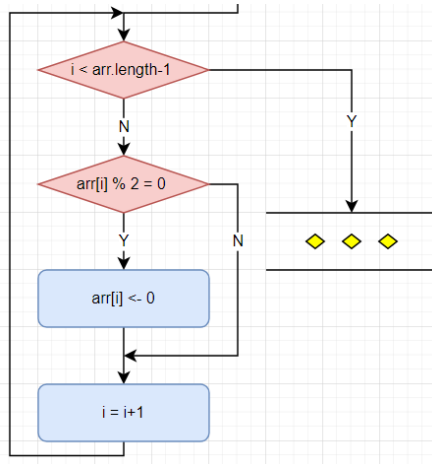
▼ Flow chart

- 수를 늘려서 100 이하의 소수를 모두 구하는 경우로 생각해보자.

▼ (1) 100 이하의 정수 데이터들을 준비하고 1을 대입한다.



▼ (2) 100 제곱근인 10, 10 이하의 소수의 배수들을 모두 제거한다.



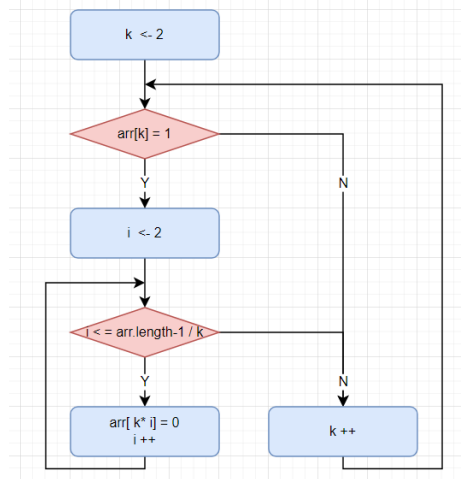
- K에 대입된 값이 소수인지의 여부에 따라 반복문으로 진입을 해야한다.
 - 따라서 $arr[k] = 1$ 이면 소수, $arr[k]$ 가 0이면 소수가 아니라는 것이다.



변수 k는 하나씩 증가한다. 2인 경우에는 2의 배수들이 전부 0이 되고, 3인 경우에는 3의 배수들을 전부 0이 되는 것을 반복한다. 그러나 4는 소수가 아니다. 3 다음의 소수는 5이므로 5의 배수를 제거해야한다. 따라서 k값이 소수인지를 판단하는 처리를 추가해야한다. 허나. 2와 3의 배수를 제거하는 처리를 한 뒤에는 이미 2와 3의 배수에는 모두 요소가 0이 되어 있다. 즉. 이 시점에는 첨자가 소수가 아닌 경우 0이 대입되어 있다.



$arr[k] = 1$ 이 Yes인 경우 k는 소수이므로 k의 배수를 제거하는 반복 처리가 실행된다. 이와 반대로 No의 경우는 k는 하나 늘려 k가 소수인지 판단하게 된



다.

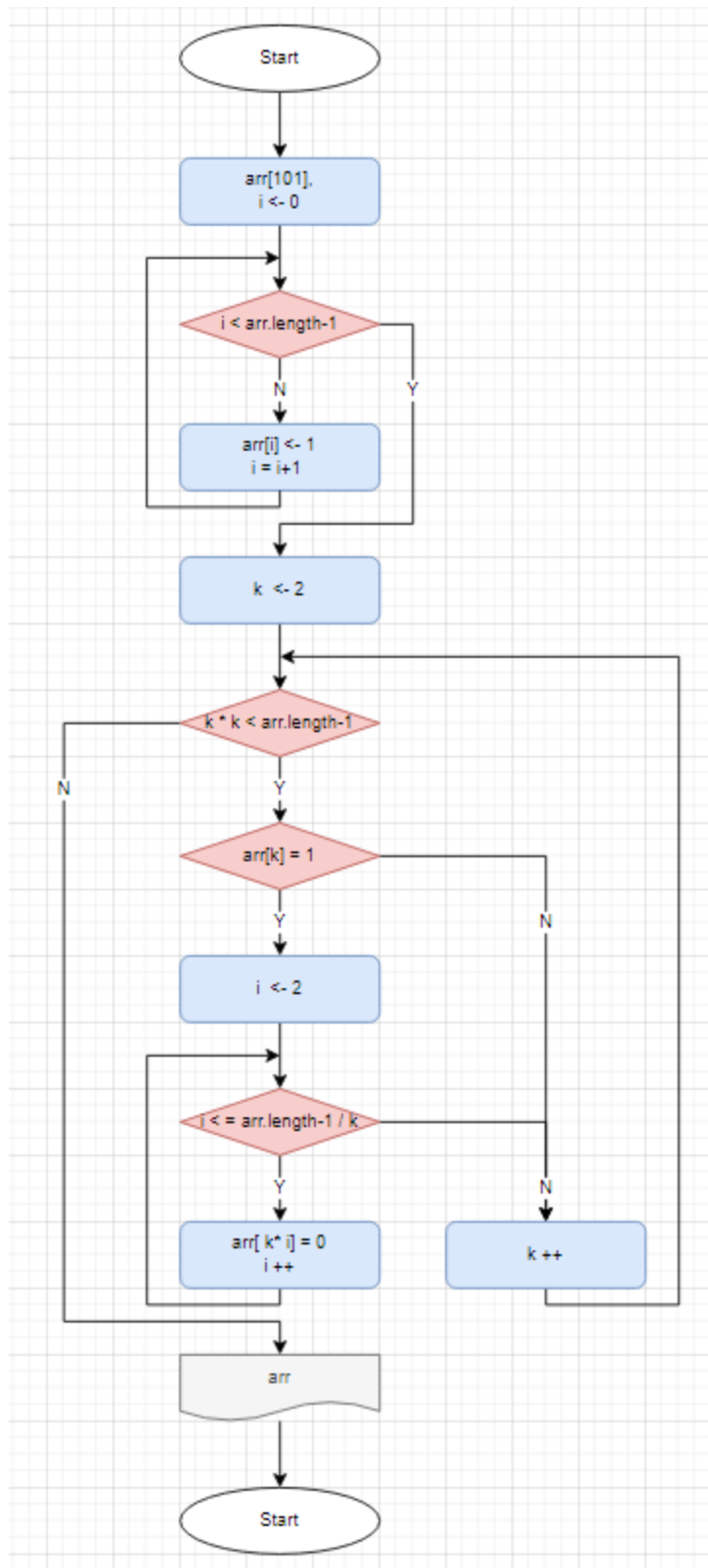
이를 과정을 통해 $k=4$ 인 경우 $arr[4]$ 는 이미 0이므로 배수를 제거하는 반복처리로 들어가지 않고 하나 증가하게 된다.

허나... k 는 무한이 증가하며 반복에 빠지게 된다.

- k 의 무한루프 해결



에라토스테네스의 체는 '어떤 숫자의 제곱근보다 작은 소수의 배수를 제거하면 남은 소가 소수이다.' 라는 이론이다. 따라서 이번 예제에서는 그 어떤 수가 100이므로 k 는 100의 제곱근 즉 10 이하가 된다. 따라서 'k가 100의 제곱근 이하인가?' 라는 판별식을 추가하여 반복을 마치고 소수를 출력하자.



DB

Chapter5 다중 테이블 질의(쿼리)

- 하나의 테이블에만 쿼리를 (질의) 던지는 것이 적지 않아. 두개 세개 더 많은 테이블의 질의를 던질 수 있어야 한다.

```
1 DESC employee;
```

Field	Type	Null	Key	Default	Extra
emp_id	NUMBER(4)	NO	PK		emp_id
fname	VARCHAR2(10)	NO			
lname	VARCHAR2(10)	NO			
dept_id	NUMBER(4)	NO			emp_deptment

```
1 DESC department;
```

Field	Type	Null	Key	Default	Extra
dept_id	NUMBER(4)	NO	PK		emp_deptment
name	VARCHAR2(30)	NO			

```
1 SELECT e.fname, e.lname, d.name
2 FROM employee e JOIN department d;
3
```

Result #1 (54r x 3c)

fname	lname	name
Michael	Smith	Operations
Michael	Smith	Loans
Michael	Smith	Administration

- 내부 조인

```
SELECT e.fname, e.lname, d.name
FROM employee e JOIN department d
ON e.dept_id = d.dept_id;
```

Result #1 (18r x 3c)

fname	lname	name
Susan	Hawthorne	Operations
Helen	Fleming	Operations
Chris	Tucker	Operations
Sarah	Parker	Operations
Jane	Grossman	Operations
Paula	Roberts	Operations

- 테이블 재사용

```
SELECT a.account_id, e.emp_id,
b_a.name open_branch, b_e.name emp_branch
FROM account a INNER JOIN branch b_a
ON a.open_branch_id = b_a.branch_id
INNER JOIN employee e
ON a.open_emp_id = e.emp_id
INNER JOIN branch b_e
ON e.assigned_branch_id = b_e.branch_id
WHERE a.product_cd = 'CHK';
```

Result #1 (10r x 4c)

account_id	emp_id	open_branch	emp_branch
10	1	Headquarters	Headquarters
14	1	Headquarters	Headquarters
21	1	Headquarters	Headquarters
1	10	Woburn Branch	Woburn Branch

```
SELECT e.fname, a.lname
FROM employee AS e, employee AS a;
```

employee (324r × 2c)	
fname	lname
Michael	Smith
Susan	Smith
Robert	Smith
Susan	Smith

Q16

사원 테이블의 이름을 출력하는데 첫 번째 컬럼은 이름을 대문자로 출력하고 두 번째 컬럼은 이름을 소문자로 출력하고 세 번째 컬럼은 이름의 첫 번째 철자는 대문자로 하고 나머지는 소문자로 출력해 보겠습니다.

```
SELECT UPPER(ename), LOWER(ename),
       CONCAT(UPPER(SUBSTR(ename,1,1)), LOWER(SUBSTR(ename,2,4)))
FROM emp;
```

emp (14r × 3c)		
UPPER(ename)	LOWER(ename)	INITCAP(ename)
KING	king	King
BLAKE	blake	Blake
CLARK	clark	Clark

- SQL에서의 함수는 크게 2가지로 나눌수 있다.
 1. 단일행 함수 - 하나의 행을 입력받아 하나의 행으로 출력하는 함수
 2. 다중행 함수 - 여러행의 값을 입력받아 하나의 행으로 출력하는 함수

Q17.

영어 단어 SMITH에서 SMI만 잘라내서 출력해 보겠습니다.

```
SELECT SUBSTR('SMITH',1,3)
FROM DUAL;
```

DUAL (1r × 1c)	
SUBSTR("SMITH",1,3)	
SMI	

- dual 테이블은 더미 테이블로 특정 테이블에 종속되지 않는 내용을 확인할 때 사용되는 일종의 가상 연습용 테이블이다.
- SQL은 다른 대부분의 컴퓨터 언어와는 달리 1에서 시작한다.

Q.18. 이름을 출력하고 그 옆에 이름의 철자 개수를 출력해 보겠습니다.

```
SELECT ename, LENGTH(ename)
FROM emp;
```

emp (14r × 2c)	
ename	LENGTH(ename)
KING	4
BLAKE	5
CLARK	5
...	...

Q.19. 사원 이름 SMITH에서 알파벳 철자 M이 몇 번째 자리에 있는지 출력해 보겠습니다.

```
SELECT INSTR('SMITH', 'M')
FROM DUAL;
```

DUAL (1r × 1c)	
INSTR('SMITH', 'M')	2

abcdefgh@naver.com 이메일에서 naver.com만 추출하고 싶다'

```
-- SELECT SUBSTR('abcdefgh@naver.com', 10, 18)
-- FROM DUAL;

SELECT SUBSTR('abcdefgh@naver.com', INSTR('abcdefgh@naver.com', '@'))
FROM DUAL
```

DUAL (1r × 1c)	
SUBSTR('abcdefgh@naver.com', INSTR('abcdefgh@naver.com', '@'))	naver.com

Q.20. 이름과 월급을 출력하는데, 월급을 출력할 때 숫자 0을 *(별표)로 출력해 보겠습니다.

```
SELECT ename, REPLACE(sal, 0, '*')
FROM emp;
```

emp (14r × 2c)	
ename	REPLACE(sal, 0, '*')
KING	5***
BLAKE	285*

Q.21. 이름과 월급을 출력하는데 월급 컬럼의 자릿수를 10자리로 하고, 월급을 출력하고 남은 나머지 자리에 별표(*)를 채워서 출력해 보겠습니다.

```
SELECT ename, LPAD(sal,10,'*') AS salary1, RPAD(sal,10,'*') AS salary2
FROM emp;
```

emp (14r x 3c)		
ename	salary1	salary2
KING	*****5000	5000*****
BLAKE	*****2850	2850*****
CLARK	*****2450	2450*****

Q.22. 첫 번째 컬럼은 영어 단어 smith 철자를 출력하고, 두 번째 컬럼은 영어 단어 smith에서 s를 잘라서 출력하고, 세 번째 컬럼은 영어 단어 smith에서 h를 잘라서 출력하고, 네 번째 컬럼은 영어 단어 smiths의 양쪽에 s를 잘라서 출력해 보겠습니다.

```
SELECT ' SMITH ', LTRIM(' SMITH '),
       RTRIM(' SMITH '), TRIM(' ' FROM " SMITHS ")
FROM dual;
```

SMITHS (1r x 4c)			
SMITH	LTRIM(' SMITH ')	RTRIM(' SMITH ')	TRIM(' ' FROM " SMITHS ")
SMITH	SMITH	SMITH	SMITHS

- 여백 날리는 용도!!

TRIM(삭제용어(선택), 삭제할문자(선택, 기본문자열 데이터 (필수)))	특정 문자를 지움	SELECT trim(ENAME), trim('A' FROM ENAME), trim(LEADING 'A' FROM ENAME), trim(TRAILING 'S' FROM ENAME) FROM EMP;	0
LTRIM(원본문자열, 삭제할문자(선택)) RTRIM(원본문자열, 삭제할문자(선택))	왼쪽, 오른쪽 지정문자를 삭제	SELECT Ltrim(ENAME), Rtrim(ENAME) FROM EMP;	선택문자열 있음 공백만 삭제

함수명	설명	예제	MYSQL 확인
UPPER(문자열)	대문자로 변환하여 반환	select upper(ENAME), lower(ENAME), INITCAP(ENAME) FROM EMP;	O
LOWER(문자열)	소문자로 변환하여 반환		O
INITCAP(문자열)	괄호안 문자데이터중 첫글자는 대문자로, 나머지는 소문자로 변환후 반환		함수 없음!
SUBSTR(문자열데이터, 시작위치, 추출길이(선택))	문자열 데이터를 시작 위치부터 추출길이만큼 추출합니다. 추출 길이 없을경우 마지막까지 추출	select JOB, SUBSTR(JOB,2), SUBSTR(JOB,1,2) FROM EMP;	O
INSTR(문자열, 위치를 찾으려는 부분문자, 시작위치(선택), 몇번째인지지정(선택))	문자열이 어디에 포함되어있는지 확인시 사용	SELECT INSTR('HELLO WORLD','L'), INSTR('HELLO WORLD','L',5), INSTR('HELLO WORLD','L',2,2) FROM DUAL;	SELECT INSTR('FOOBAR','BAR');
REPLACE(문자열, 찾는 문자열, 대체할문자(선택))	다른문자로 바꾸는 함수!	select JOB , replace(JOB, 'A' , 'Q') FROM EMP;	대체할 문자 생략 불가.
LPAD(문자열,자리수,채울문자(선택)) RPAD(문자열,자리수,채울문자(선택))	데이터의 빈 공간을 설정한 자리수 만큼 설정한 문자로 채운다.	SELECT lpad(ENAME,10,'@'), RPAD(ENAME,10,'@') FROM EMP;	채울문자 필수입력
CONCAT(문자열,문자열)	두 문자열 데이터를 합친다.	SELECT concat(ENAME, JOB) FROM EMP;	O
TRIM(삭제옵션(선택), 삭제할문자(선택), 원본문자열 데이터 (필수))	특정 문자를 지움	SELECT trim(ENAME), trim('A' FROM ENAME), trim(LEADING 'A' FROM ENAME), trim(TRAILING 'S' FROM ENAME) FROM EMP;	O

LTRIM(원본문자열, 삭제할문자(선택)) RTRIM(원본문자열, 삭제할문자(선택))	왼쪽, 오른쪽 지정문자를 삭제	SELECT Ltrim(ENAME), Rtrim(E NAME) FROM EMP;	선택문자열 없음 공백만 삭제
ROUND(숫자, 반올림위치(선택))	특정위치에서반올림	select ROUND(SAL, 1), ROUND(S AL) FROM EMP;	O
TRUNC(숫자, 버림위치(선택))	특정위치에서 버림		TRUNCATE(숫자, 버림위치(필수))
CEIL(숫자)	지정한 숫자와 가까운 큰 정수를 찾을	SELECT CEIL(3.141592), FLOOR (3.141592);	O
FOOR(숫자)	지정한 숫자와 가까운 작은 정수		O
MOD(나눗셈될숫자, 나눌숫자)	숫자를 나눈 나머지 값을 구함	SELECT MOD(3.141592,3);	O
날짜관련			
SYSDATE	날짜 데이터	SELECT SYSDATE AS NOW, SYSDATE+1 AS TOMORROW,SYSDATE -1 AS YESTERDAY FROM DUAL;	SELECT SYSDATE(), NOW(, NOW()+1, NOW()-1;
ADD_MONTH((날짜데이터), 더할 개월수)	몇개월 이후의 날짜를 구함	SELECT sysdate, ADD_MONTHS(SYSDATE,3) FROM DUAL;	없음 DATE_ADD사용 DATEADD(NOW(), INTERVAL 1 MONTH)
*mysql 함수 DATE_ADD(date, INTERVAL expr unit)	시간 더하기	*MYSQL select date_add(now(), interval 1 day), date_add(now(), interval 1 hour), date_add('20211018', interval 1 month);	
*mysql 함수 DATE_SUB(date, INTERVAL expr unit)	시간빼기	*MYSQL select date_sub(now(), interval 1 day), date_sub(now(), interval 1 hour), date_sub('20211018', interval 1 month);	

MONTHS_BETWEEN(날짜 데이터1, 날짜 데이터 2)	두 날짜 간의 개월수 차이를 구함.	select months_between(hiredate,sysdate) from emp;	없음
*mysql 함수 DATEDIFF(날짜데이터1, 날짜데이터2)	두 날짜 사이의 차이를 구함	*MYSQL SELECT datediff(NOW(), '20220620')	
NEXT_DAY(날짜데이터, 요일문자)	돌아오는 요일,달 마지막 날짜를 구함	SELECT SYSDATE, NEXT_DAY(SYSDATE, '월요일'), LAST_DAY(SYSDATE) FROM DUAL	없음
LAST_DAY(날짜데이터)	속한 달의 마지막 날짜를 출력		
ROUND(날짜, 포맷)	날짜 반올림	SELECT round(NOW(), 'CC'), round(NOW(), 'YYYY'), round(NOW(), 'Q'), round(NOW(), 'DDD'), round(NOW(), 'HH'), TRUNC(NOW(), 'CC'), TRUNC(NOW(), 'YYYY'), TRUNC(NOW(), 'Q'), TRUNC(NOW(), 'DDD'), TRUNC(NOW(), 'HH');	SELECT round(NOW(), 'CC'), round(NOW(), 'YYYY'), round(NOW(), 'Q'), round(NOW(), 'DDD'), round(NOW(), 'HH'), TRUNCATE(NOW(), 'CC'), TRUNCATE(NOW(), 'YYYY'), TRUNCATE(NOW(), 'Q'), TRUNCATE(NOW(), 'DDD'), TRUNCATE(NOW(), 'HH');
TRUNC(날짜, 포맷)	날짜 버림		
TO_CHAR		SELECT TO_CHAR(SYSDATE(), 'YYYY/MM/DD HH24:MI:SS')	없음
*MYSQL DATE_FORMAT(날짜, 포맷)	날짜, 숫자데이터를 문자 데이터로 변환	*MYSQL SELECT DATE_FORMAT(SYSDATE(), '%Y-%M-%D');	
TO_NUMBER		SELECT TO_NUMBER('1') FROM DUAL;	CAST 함수써야함

TO_NUMBER		SELECT TO_NUMBER('1') FROM DUAL;	CAST 함수써야함
*MYSQL CAST(문자열 AS INTEGER)	문자데이터를 숫자데이터로 변환	*MYSQL SELECT CAST('1' AS INTEGER) AS NUM;	
TO_DATE	문자데이터를 날짜 데이터로 변환	TO_DATE	X DATE_FORMAT 함수 쓸것
NVL()	열 또는 데이터를 입력하여 해당데이터가 NULL이 아닐 경우데이터를 그대로 반환하고, NULL인 경우지정한 데이터를 반환.	SELECT NVL(COMM,0) FROM EMP;	X IFNULL 함수 쓸것
NVL2()	데이터가 NULL이 아닐때 반환데이터를 추가로 지정가능	SELECT NVL2(COMM, '0', 'X') FROM EMP	X
*MYSQL IFNULL(비교군, '리턴내용')	NULL 일경우 사용자 지정내용 리턴	*MYSQL SELECT IFNULL(NAME, '값이없습니다') FROM TEST	

DECODE	<p>DECODE([검사대상이 될 열 또는 데이터, 연산이나 함수의 결과],[조건1],[데이터가 조건1과 일치할때 반환결과], [조건2],[데이터가 조건2와 일치할때 반환할 결과],</p> <p>.....</p> <p>[조건N],[데이터가 조건N와 일치할때 반환할 결과]</p> <p>],</p> <p>[위조건과 일치한 경우가 없을때 반환할 결과])</p> <p>IF나 /SWITCH-CASE 조건문과 같은 이 데이터를 조건에부합하는지 비교하고 부합하는 경우 설정한 결과를 반환한다.</p>	<p>SELECT</p> <p>EMPNO, ENAME, JOB, SAL,</p> <p>DECODE(</p> <p>JOB</p> <p>, 'MANAGER', SAL*1.1,</p> <p>'SALESMAN', SAL*1.1,</p> <p>'ANALYST', SAL</p> <p>,SAL* 1.03)</p> <p>FROM EMP;</p>	<p>없음 IF나 CASE문 쓸것</p> <p>ENCODE(문자열,키)</p> <p>암호화 함수</p> <p>DECODE(문자열, 키)</p> <p>복호화 함수</p> <p>사용 용도가 다름</p>
--------	--	---	--

CASE	<p>조건을 비교하되 각 조건에 사용하는 데이터가 서로 상관없어도 비교가능, 기준데이터 값이 같은 데이터외에 다양한 조건을 사용할 수있다.</p> <p>CASE [비교할 데이터] WHEN [조건1] THEN [조건1 True 반환할 결과] WHEN [조건2] THEN [조건2 True 반환할 결과] WHEN [조건3] THEN [조건3 True 반환할 결과] ... WHEN [조건n] THEN [조건n True 반환할 결과] ELSE [위조건에 일치하는경우가 없을때 바나할 결과] END</p>	<p>SELECT EMPNO, CASE JOB WHEN 'MANAGER' THEN SAL*1.1 WHEN 'SALESMAN' THEN SAL*1.05 ELSE SAL*1.03 END FROM EMP;</p>	O
* MYSQL IF문	<p>오라클 함수의 decode함수는 조건과 같을 경우만 비교하지만 , 조건문을 적어 줄수있다.</p> <p>IF (조건, 조건이 일치할경우 VALUE, 조건이 일치하지 않을 경우 VALUE)</p>	<p>*MY SQL SELECT deptno, IF(deptno <= 20,'RESEARCH','OPERATIONS') deptno as "Dept Name" FROM dept;</p>	
다중행함수			

SUM(해당열)	합계	SELECT COUNT(SAL), COUNT(ALL SAL),COUNT(distinct SAL), SUM(SAL), MAX(SAL), MIN(SAL), AVG(SAL) FROM EMP;
COUNT(해당열)	데이터의 갯수를 구해줌	
MAX(),MIN()	최대값 , 최소값	
AVG()	평균값	

반올림

```
/*반올림ROUND
```

```
ROUND(대상, 반올림 위치)
```

반올림 위치가 소수점 아래면 양수로 그 자리까지로 반올림

반올림 위치가 소수점 위에는(즉 정수) 음수로 그 자리에서 반올림.

```
*/
```

```
SELECT '1234.5678', ROUND(1234.5678,1), ROUND(1234.5678, -1)
FROM dual;
```

dual (1r x 3c)		
1234.5678	ROUND(1234.5678,1)	ROUND(1234.5678,-1)
1234.5678	1,234.6	1,230

Quiz

23

876.567 숫자를 출력하는데 소수점 두 번째 자리인 6에서 반올림해서 출력해 보겠습니다.

```
SELECT '876.567', ROUND(876.567,1)
FROM dual;
```

dual (1r x 2c)	
876.567	ROUND(876.567,1)
876.567	876.6

버리기

```
/* TRUNCATE 버리기
```

```
TRUNCATE(대상, 버릴 위치)
```

버릴 위치가 소수점 아래면 양수로 그 자리까지 남기고 버림.

버릴 위치가 소수점 위에는(즉 정수) 음수로 그 자리에서 버림.

```
*/
```

```
SELECT '1234.5678' AS 숫자, TRUNCATE(1234.5678,1), TRUNCATE(1234.5678, -2)
FROM dual;
```

dual (1r x 3c)		
숫자	TRUNCATE(1234.5678,1)	TRUNCATE(1234.5678,-2)
1234.5678	1,234.5	1,200

나머지 연산자

```
/* 나머지 연산자 함수 mod()
```

```
SELECT MOD(대상 숫자, 나눌 숫자)
FROM dual;
```

```
*/
```

```
SELECT MOD(5, 2)
FROM dual;
```

dual (1r × 1c)	
MOD(5, 2)	1

Quiz

25

사원 번호와 사원 번호가 홀수이면 1, 짝수이면 0을 출력

```
SELECT empno, (MOD(empno, 2))
FROM emp;
```

emp (14r × 2c)	
empno	(MOD(empno, 2))
7,839	1
7,698	0
7,782	0

Quiz

25-

2

사원 번호가 짝수인 사원들의 사원 번호와 이름을 출력하는 쿼리

```
SELECT empno, ename
FROM emp
WHERE MOD(empno, 2) = 0;
```

emp (10r × 2c)	
empno	ename
7,698	BLAKE
7,782	CLARK
7,566	JONES
7,654	MARTIN

Quiz

다음의 쿼리는 10을 3으로 나눈 몫을 출력하는 쿼리입니다.

25-
3

```
SELECT FLOOR(10/3)
FROM dual;
```

dual (1r × 1c)	
FLOOR(10/3)	3

CASE 구문

/* CASE 구문 (경우에 따라)

```
CASE
  WHEN 조건 THEN 결과
  WHEN 조건 THEN 결과
  ELSE 결과
END AS 별명

*/
SELECT ename, job, sal,
       CASE
         WHEN sal >= 3000 THEN 500
         WHEN sal >= 2000 THEN 300
         ELSE 0
       END AS Bonus
FROM emp;
```

emp (14r × 4c)			
ename	job	sal	Bonus
KING	PRESIDENT	5,000	500
BLAKE	MANAGER	2,850	300
CLARK	MANAGER	2,450	300
JONES	MANAGER	2,975	300
MARTIN	SALESMAN	1,250	300
ALLEN	SALESMAN	1,600	300
TURNER	SALESMAN	1,500	300
JAMES	CLERK	950	0
WARD	SALESMAN	1,250	300
FORD	ANALYST	3,000	500
SMITH	CLERK	800	0
SCOTT	ANALYST	3,000	500
ADAMS	CLERK	1,100	300
MILLER	CLERK	1,300	300

Quiz

- 26 다음의 쿼리는 보너스를 출력할 때 직업이 SALESMAN, ANALYST이면 500을 출력하고, 직업이 CLERK, MANAGER이면 400을 출력하고, 나머지 직업은 0을 출력하는 쿼리입니다.

```
SELECT ename, job,
       CASE
         WHEN job IN('SALESMAN','ANALYST') THEN 500
         WHEN job IN('CLERK','MANAGER') THEN 400
         ELSE 0
       END AS Bonus
FROM emp;
```

emp (14r × 3c)		
ename	job	Bonus
KING	PRESIDENT	0
BLAKE	MANAGER	400
CLARK	MANAGER	400
JONES	MANAGER	400
MARTIN	SALESMAN	500
ALLEN	SALESMAN	500