

Day26

🕒 작성일시	@2022년 8월 3일 오전 10:01
▼ 강의 번호	
▼ 유형	
🔗 자료	
☑ 복습	<input type="checkbox"/>

I . 알고리즘.

Sieve of Eratosthenes (에라토스테네스의 체)

- 소수 prime number를 찾아내는 알고리즘
- 소수 : 2이상의 정수에서 1과 그 수 자체로만 나눌수 있는 수
- 소수는 나열되어있는 구간이 불규칙하므로 임의로 찾기가 힘들다

소수 (prime number)

- 소수는 2 이상의 정수 중에서 1과 그 수 자신 외에는 나눌수 없는 숫자.
- 10 이하에서는 2, 3, 5, 7이 소수에 해당된다.

4는 2와 2로, 6은 2와 3으로 나눌수 있다.

⇒ 따라서 소수가 아니다. 이러한 숫자들은 합성수. 즉 숫자와 숫자가 합성된 것이라고 할 수 있다. 반대로 말하면 합성수가 아닌 숫자들이 바로 소수인 것이다.

⇒ 소수(prime)에서의 '소'란 합성되지 않은 소박한 숫자 의미의 소를 뜻한다. 혹은 모든 소의 소(근본)을 의미하기도 한다.

소수인지 아닌지 구분 하는 것은 의외로 어렵다.

소수의 내용만 보면 어렵지 않아보이지만, 사실은 아주 어렵다. 무엇이 어려운지도 바로 찾아내기 힘들다. 예시로 3의 배수는 3, 6, 9, 12, 15 처럼 3개의 간격으로 나열된다. 따라서 1에서 100까지의 사이에 있는 3의 배수를 찾는 일은 간단하다.

⇒ 하지만 소수는 규칙성이 없기 때문에 각각이 불규칙하다. 즉, 소수는 한번에 열거하기가 어렵다.

가장 먼저 떠오르는 방법은 하나하나 그 수보다 작은 숫자로 나누어 보는 것이다. 2부터 100까지의 소수를 찾으려면 먼저 2로 나눌수 있는 수를 지우고, 그 다음은 3으로 나눌수 있는 수를 지우는 식으로 99까지 진행한다. 하지만 이 방법은 비효율적이고, 수의 범위가 커지는 경우에는 많은 시간이 소모될 것이다.

	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

고대 그리스의 과학자인 에라토스테네스는 위의 방법을 개선하여 소수를 효율적으로 발견하는 방법을 알아냈다. 그의 이름을 따서 에라토스테네스의 체라고 부른다.

에라토스테네스의 체

! 어떤 수 이하의 범위에 존재하는 소수를 찾고 싶은 경우
그 수의 제곱근보다 작은 소수의 배수만 없애면 남은 수가 소수다.

라는 생각을 바탕으로 소수를 찾는 방법이다.

ex) 100이하의 소수를 모두 찾아내려면 먼저 100의 제곱근 이하 소수를 선택한다.

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

10 이하의 소수는 2, 3, 5, 7 총 4개이다.

우선 2에서 100까지의 표에서 2로 나눌 수 있는 수, 즉 2의 배수중 2를 제외한 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 7의 배수들도 모두 삭제한 결과는 아래와 같다.

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

이 방법을 사용하면 1에서 100까지의 모든 숫자를 해당숫자보다 작은 수로 나눌 수 있는지 하나하나 전부 순서대로 확인하는 것 보다는 훨씬 더 빨리 소수를 찾을 수 있다.

에라토스테네스의 체 알고리즘.

- 에라토스테네스의 체는 크게 3개의 처리로 구성한다.



- (1) 어떤 수 이하의 모든 정수 데이터를 준비.
- (2) 어떤 수의 제곱근보다 작은 소수의 배수들을 차례로 제거
- (3) 마지막까지 남은 수들을 출력.

ex) 10이하의 소수를 구하는 경우

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

- ① 10 이하의 정수 데이터들을 준비한다.

먼저 체의 대상이 되는 10까지의 정수를 데이터로 준비한다 여기에서는 11개의 요소를 가지는 정수형 배열을 준비한다. 배열의 이름은 arr로 진행한다. 첨자는 0부터 시작하기 때문에 첨자를 10까지로 동일하게 사용하려면 요소를 11개 준비해야 한다.

0	1	2	3	4	5	6	7	8	9	10

요소는 그 첨자가 소수인지 여부를 판정하는 데이터를 넣는 것으로 사용하자.

즉, 소수가 아닌 것으로 판정된 첨자의 요소에 소수가 아니다 라는 것을 나타내는 데이터를 넣다.

⇒ 즉 소수의 가능성이 있는 경우에는 1을 대입하고, 소수가 아닌 경우에는 0을 대입한다.

위에서의 개념 소개에서는 소수가 아닌 수를 제거하는 방식을 0을 대입하는 방식으로 처리한다. 따라서 초기값을 1로 모두 대입해둔다. 그리고 소수가 아닌 것으로 판

정된 수(첨자)의 요소에는 0을 대입한다.

이러한 방식을 통해 소수의 배수를 모두 제거 (0을 대입) 한 후, 마지막에 1이 남아있는 요소들의 첨자는 모두 소수이고, 0이 대입되어 있는 요소의 첨자는 소수가 아니라고 구별할 수 있다.

②10의 제곱근 약 3.16보다 작은 소수의 배수를 차례대로 제거한다.

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

10의 제곱근은 약 3.16이므로 제곱근 이하의 소수는 2와 3이다. 따라서 2의 배수 모두와 3의 배수 모두를 제거하면 된다.

먼저 2의 배수들을 모두 지운다. 이때, 지운다는 의미를 0을 대입한다는 표현으로 적용할 수 있다.

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

다음은 3의 배수들을 모두 지운다. 이때, 지운다는 의미를 0을 대입한다는 표현으로 적용할 수 있다.

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

③마지막까지 남은 수들을 출력한다.

제곱근 이하의 소수들의 모든 배수를 제거(0으로 대입) 했다면 나머지는 모두 소수 뿐이다. (여전히 요소의 값이 1인) 그럼 이제 첨자가 2 이상이고, 1이 들어가는 요

소의 첨자들만 뽑아내면 그것들이 바로 10이하의 소수가 된다.

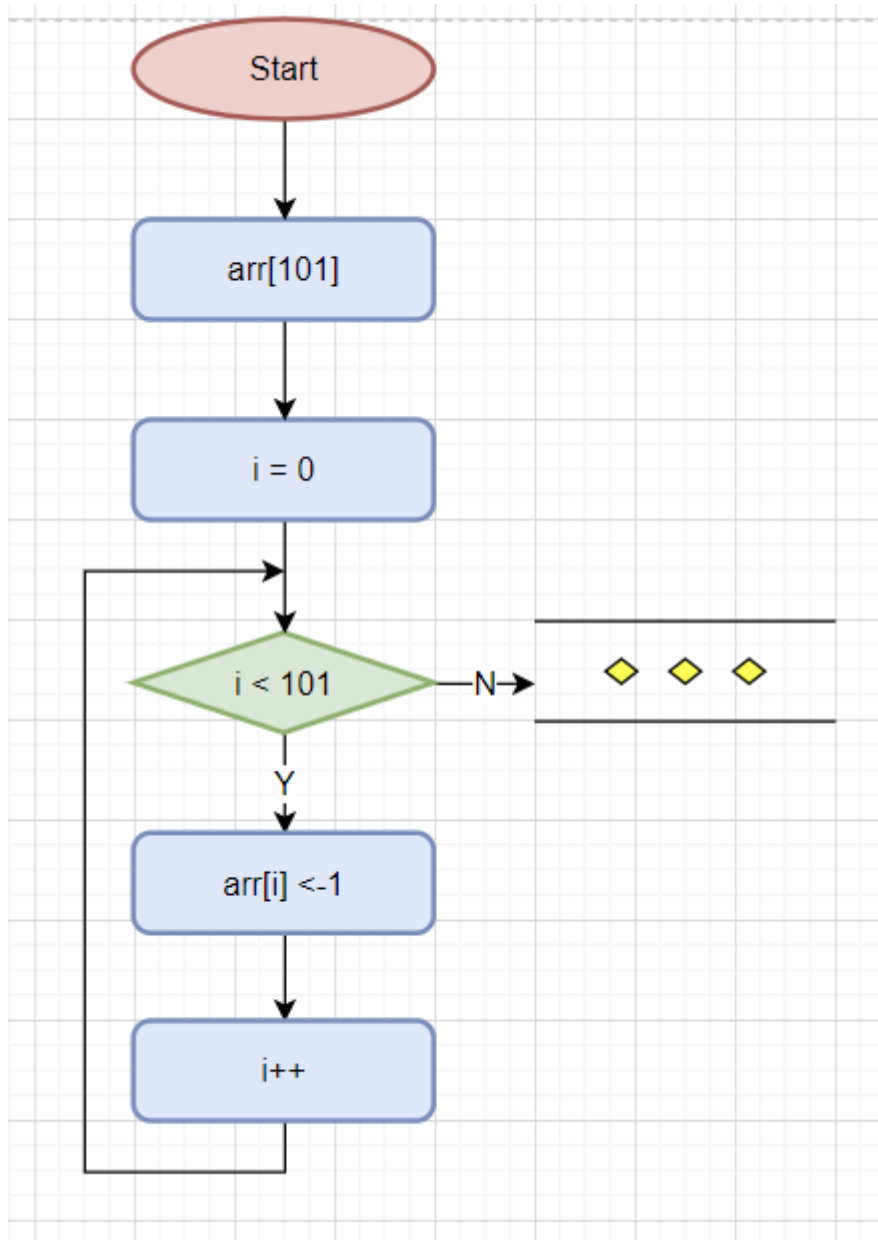
1	1	1	1	0	1	0	1	0	0	0
0	1	2	3	4	5	6	7	8	9	10

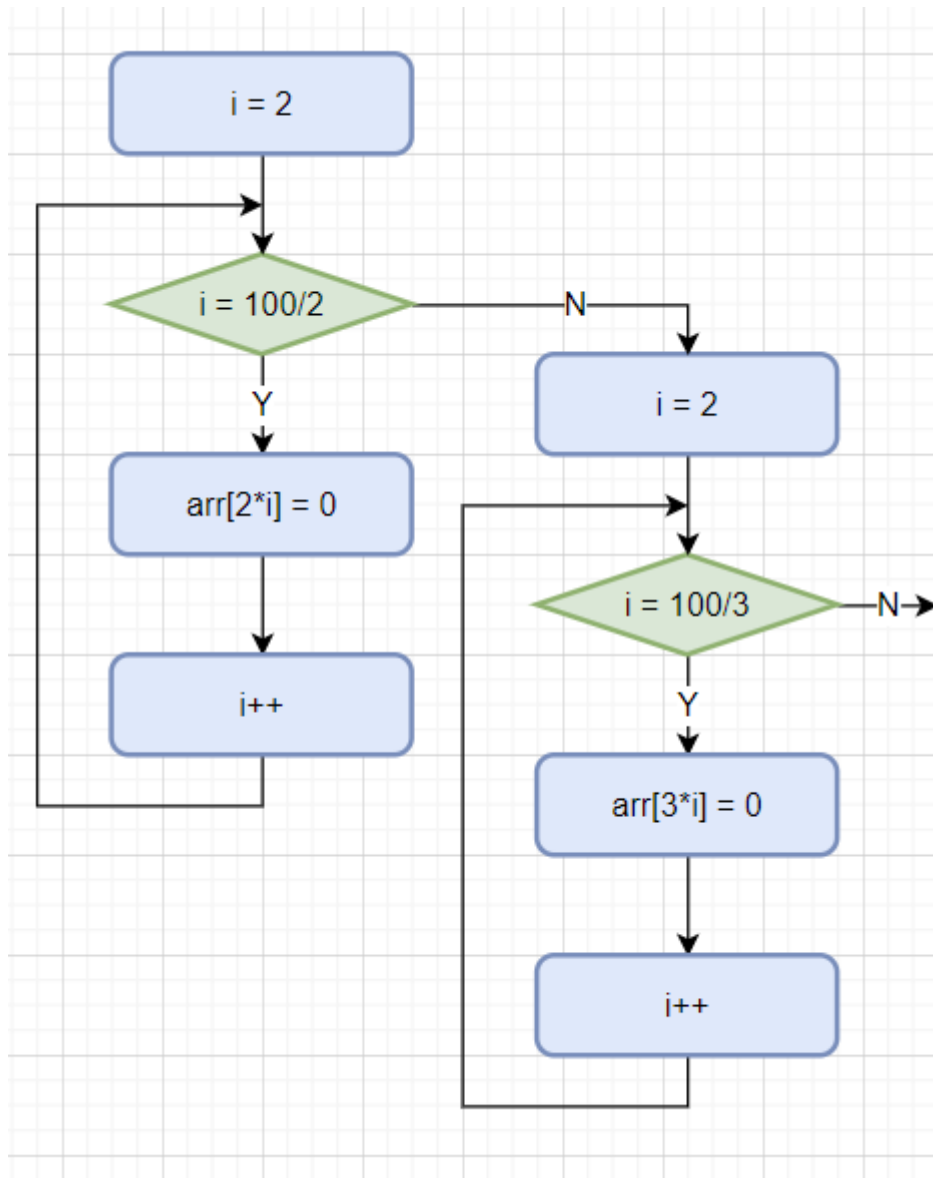
에라토스테네스의 체 순서도

이번에는 수를 늘려 100이하의 소수를 구하는 경우로 생각해본다.

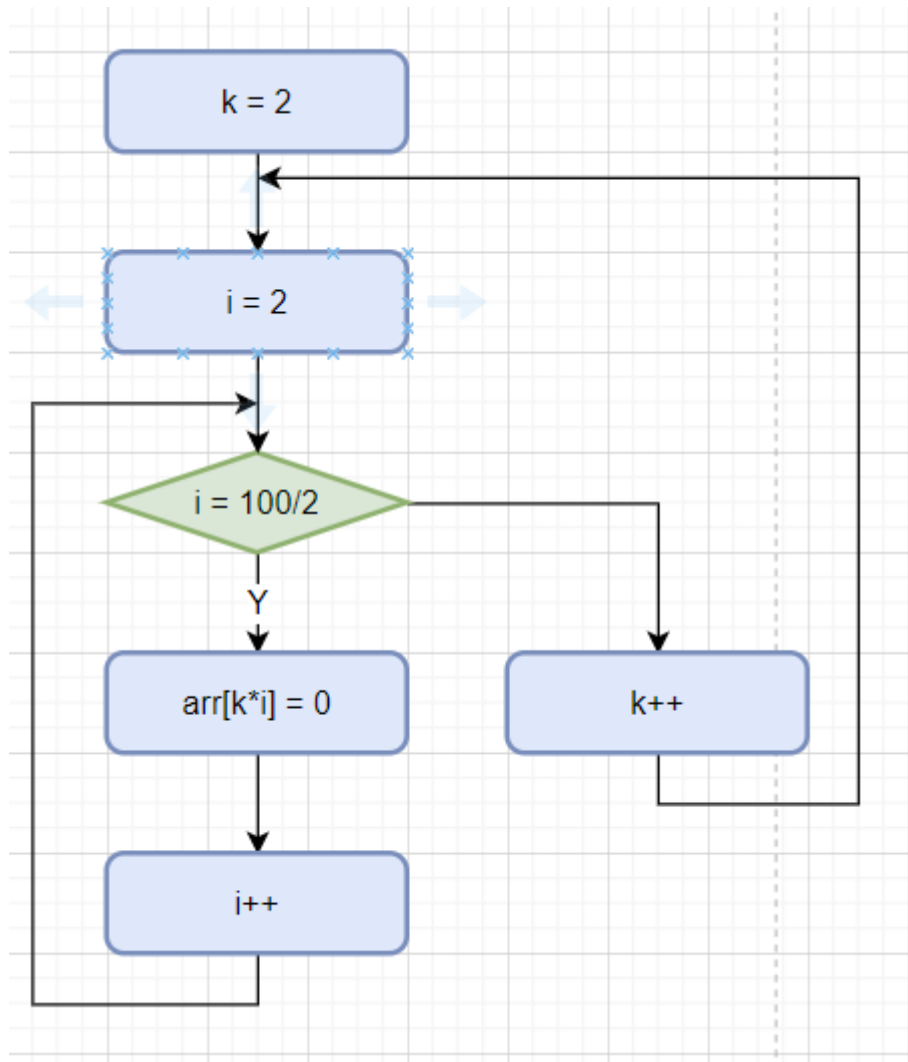
따라서 배열의 요소 수는 101개가 된다. 앞에서 생각한 것과 마찬가지로 첨자들을 수로 사용하기 위해서 이다. 첨자는 0부터 시작하므로 101개가 필요하다.

모든 요소에는 초기값으로 1을 대입하여 초기화 한다.





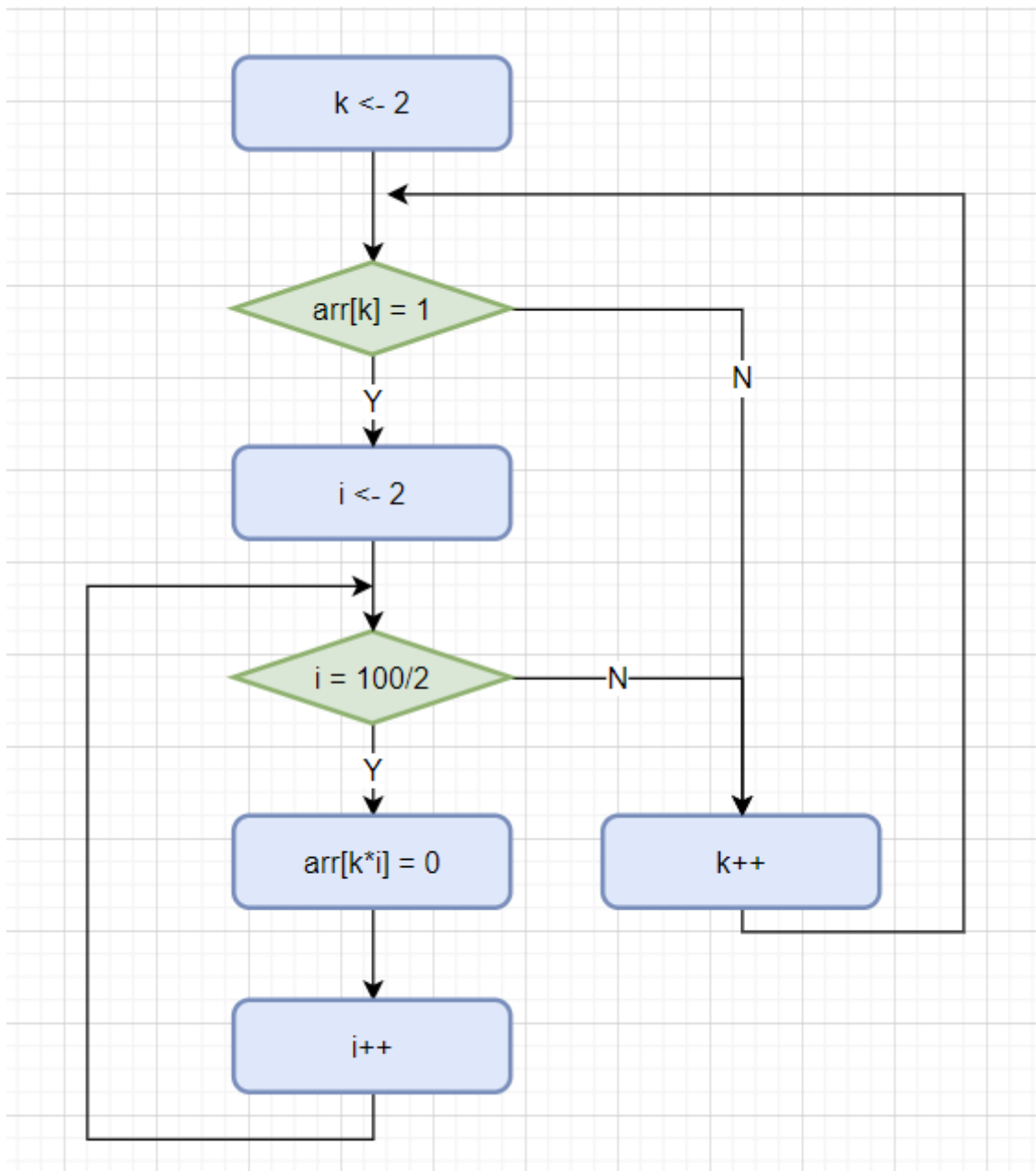
위 두 그림을 하나로 뭉치면



변수 k는 하나씩 증가한다. 2인 경우에는 2의 배수들을 전부 0으로 바꾸고, 3인 경우에는 3의 배수들을 전부 0으로 바꾸고 계속 반복 시킨다. 그러나 4는 소수가 아니다.

3 다음의 소수는 5이므로, 5의 배수를 제거해야 한다. k값이 소수인지를 판단하는 처리를 추가해야한다. ⇒ 2와 3의 배수 처리 제거가 끝난 상태에서 배열을 생각해보면 2의 배수와 3의 배수는 모두 0인 상태가 되었다. 이 시점에서 첨자가 소수가 아닌 요소에는 0이 되어있다. 즉, 4에는 0이 대입되어 있다.

이때 k에 대입된 값이 소수인지의 여부에 따라 반복문으로의 진입을 해야 한다. 따라서 arr[k]가 1이면 소수이고, arr[k] = 0이면 소수가 아닌 것이다.



arr[k] = 1 이 참인 경우, k는 소수이므로 k의 배수를 제거하는 반복처리가 실행된다. 반대로 arr[k] = 1이 거짓인 경우, k는 소수가 아니므로 k를 하나 증가시켜 다음 k가 소수인지를 다시 판명하게 된다.

k값은 이제 2, 3으로 증가하고, k=4인 경우, arr[4] = 0이므로 배수를 제거하는 반복처리로 돌아가지 않고 k만 하나 증가하게 된다.

그 다음 k = 5인 경우 arr[5] = 1이므로 5의 배수들을 모두 제거하고 계속 반복 된다. ⇒ 탈출 지점이 없어 무한루프에 빠지게 된다.

에라토스테네스의 체는 어떤 숫자의 제곱근보다 작은 소수의 배수를 제거하면 남은 수가 소수이다 라는 이론을 가지고 있다. 따라서 이번 예제에서는 그 어떤 수가 100이므로 k는 100의 제곱근 즉 10 이하가 된다. 따라서 k가 100의 제곱근 이하인가? 라는 판별식을 추가하여 반복을 마치게 된다.

DB

I . Chapter 5 다중테이블 질의 (쿼리)

- 하나의 테이블에만 쿼리를 질의를 던지는 것이 적지 않다. 두개 세개 더 많은 테이블의 질의를 던질 수 있어야 한다.

```
1 DESC employee;
```

COLUMNS (9r × 6c)

Field	Type	Null	Key	Default	Extra
emp_id	smallint(5) unsigned	NO	PRI	(NULL)	auto_increment
fname	varchar(20)	NO		(NULL)	
lname	varchar(20)	NO		(NULL)	
start_date	date	NO		(NULL)	
end_date	date	YES		(NULL)	
superior_emp_id	smallint(5) unsigned	YES	MUL	(NULL)	
dept_id	smallint(5) unsigned	YES	MUL	(NULL)	
title	varchar(20)	YES		(NULL)	
assigned_branch_id	smallint(5) unsigned	YES	MUL	(NULL)	

```
1 DESC department;
```

COLUMNS (2r × 6c)

Field	Type	Null	Key	Default	Extra
dept_id	smallint(5) unsigned	NO	PRI	(NULL)	auto_increment
name	varchar(20)	NO		(NULL)	

⇒ 이 두개 테이블을 동시에 조회하는 방법.

(1) From (테이블) join (테이블)

```
mysql> SELECT e.fname, e.lname, d.name
-> FROM employee e JOIN department d;
```

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

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

결과 #1 (54r x 3c)		
fname	lname	name
Michael	Smith	Operations
Michael	Smith	Loans
Michael	Smith	Administration

(2) Inner joins

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

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

결과 #1 (18r x 3c)		
fname	lname	name
Susan	Hawthorne	Operations
Helen	Fleming	Operations
Chris	Tucker	Operations

- 같은 테이블 두번 사용 (테이블 재사용)

호스트: 127.0.0.1 데이터베이스: bank 테이블: em		
1	SELECT e.fname, a.lname	
2	FROM employee AS e, employee AS a;	

※ 별명은 as 붙이고 붙일거 쓰기.

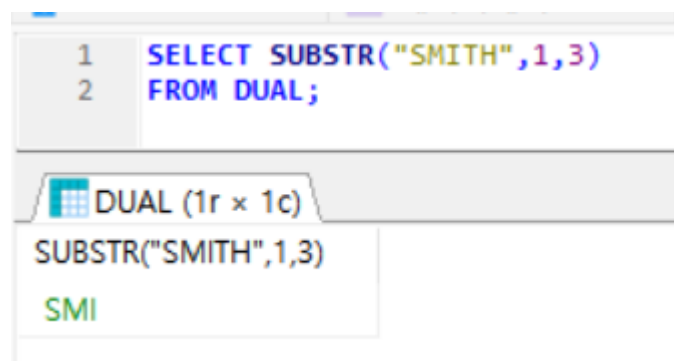
- SQL의 함수는 크게 두가지로 나뉜다.

(1) 단일행 함수 - 하나의 행을 입력받아 하나의 행으로 출력하는 함수

(2) 다중행 함수 - 여러행의 값을 입력받아 하나의 행으로 출력하는 함수

Quiz 17

영어 단어 SMITH에서 SMI만 잘라내서 출력해 보겠습니다.



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

- dual테이블 - 더미테이블. 특정 테이블에 종속되지 않는 내용을 확인할때 사용되는 일종의 연습용 테이블이다.
- SQL은 다른 대부분의 컴퓨터 언어와는 다르게 1부터 시작한다.

Quiz 18

이름을 출력하고 그 옆에 이름의 철자 개수를 출력해 보겠습니다.

ENAME	LENGTH(ENAME)
KING	4
BLAKE	5

1	SELECT	ename,	LENGTH(ename)
2	FROM	emp;	

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

Quiz 19

사원 이름 SMITH에서 알파벳 철자 M이 몇 번째 자리에 있는지 출력해

INSTR('SMITH', 'M')

2

1	SELECT	INSTR('SMITH', 'M')
2	FROM	dual;

dual (1r × 1c)	
INSTR('SMITH', 'M')	2

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

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

(쉬운버전)

```
1 SELECT SUBSTR('abcdefgh@naver.com', 10, 18)
2   from DUAL;
3
```

결과 #1 (1r x 1c)

SUBSTR('abcdefgh@naver.com', 10, 18)
naver.com

(어려운 버전)

호스트: 127.0.0.1 데이터베이스: bank 쿼리*

```
1 SELECT substr('abcdefgh@naver.com', instr('abcdefgh@naver.com', '@')+1)
2   from DUAL;
3
```

결과 #1 (1r x 1c)

substr('abcdefgh@naver.com', instr('abcdefgh@naver.com', '@')+1)
naver.com

Quiz 16

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.

UPPER(ENAME)	LOWER(ENAME)	INITCAP(ENAME)
KING	king	King
BLAKE	blake	Blake
:	:	:
ADAMS	adams	Adams
MILLER	milller	Miller

```

1 SELECT UPPER(ename), LOWER(ename), CONCAT(SUBSTR(UPPER(ename),1,1),SUBSTR(LOWER(ename), 2))
2     from emp;
3

```

결과 #1 (14r x 3c)		
UPPER(ename)	LOWER(ename)	CONCAT(SUBSTR(UPPER(ename),1,1),SUBSTR(LOW...
KING	king	King
BLAKE	blake	Blake
CLARK	clark	Clark
JONES	jones	Jones
MARTIN	martin	Martin
ALLEN	allen	Allen
TURNER	turner	Turner

Quiz 20

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

KING	5***
BLAKE	285*
:	:
ADAMS	11**
MILLER	13**

```

1 SELECT ENAME, REPLACE(sal, 0, '*')
2     FROM emp;

```

emp (14r x 2c)	
ENAME	REPLACE(sal, 0, '*')
KING	5***
BLAKE	285*
CLARK	245*

⇒ 찾아서 바꾸는 함수 REPLACE(찾을 곳, 바꿀것, 최종 바뀌어야 하는 것)

Quiz 21

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

ENAME	SALARY1	
KING	*****5000	5000*****
BLAKE	*****2850	2850*****
CLARK	*****2450	2450*****
:	:	:
SCOTT	*****3000	3000*****
ADAMS	*****1100	1100*****
MILLER	*****1300	1300*****

호스트: 127.0.0.1 | 데이터베이스: bank | 쿼리*

```

1 SELECT ENAME, LPAD(sal,10,'*') as SALARY1, RPAD(sal,10,'*') as SALARY2
2 FROM emp;

```

emp (14r x 3c)

ENAME	SALARY1	SALARY2
KING	*****5000	5000*****
BLAKE	*****2850	2850*****
CLARK	*****2450	2450*****
JONES	*****2975	2975*****
MARTIN	*****1250	1250*****

⇒ 일정 칸수를 채우는데 이제 왼쪽 오른쪽 나머지 칸에 맞춰서 채우는 함수 : LPAD(왼쪽), RPAD(오른쪽)

Quiz 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.

smith	mith	smit	mith
-------	------	------	------

(배운거로)

호스트: 127.0.0.1 데이터베이스: bank 쿼리*

```

1 SELECT "smith", SUBSTR("smith", 2,5),
2     SUBSTR("smith", 1,LENGTH("smith")-1),
3     SUBSTR("smiths", 2, LENGTH("smiths")-2)
4 from DUAL;
5

```

DUAL (1r x 4c)

smith	SUBSTR("smith", 2,5)	SUBSTR("smith", 1,LENGTH("smith")-1)	SUBSTR("smiths", 2, LENGTH("smiths")-2)
smith	mith	smit	mith

(함수 사용)

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;	0
TRIM(원본문자열, 삭제할문자(선택)) LTRIM(원본문자열, 삭제할문자(선택)) RTRIM(원본문자열, 삭제할문자(선택))	왼쪽, 오른쪽 지정문자를 삭제	SELECT Ltrim(ENAME), Rtrim(ENAME) FROM EMP;	선택문자열 없음 공백만 삭제

```

1 SELECT LTRIM(" smith "), RTRIM(" smith "), TRIM(" " FROM " smith ")
2 FROM DUAL;

```

smith (1r x 3c)

LTRIM(" smith ")	RTRIM(" smith ")	TRIM(" " FROM " smith ")
smith	smith	smith

⇒ LTRIM, RTRIM, TRIM 은 주로 여백 날리는 용도로 사용한다.

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), 몇번째인지를지정(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(ENAME) FROM EMP;	선택문자열 없음 공백만 삭제
ROUND(숫자, 반올림위치(선택))	특정위치에서반올림	select ROUND(SAL, 1), ROUND(SAL) FROM EMP;	O
TRUNC(숫자, 버림위치(선택))	특정위치에서 버림		TRUNCATE(숫자, 버림위치(필수))
CEIL(숫자)	지정한 숫자와 가까운 큰 정수를 찾음	SELECT CEIL(3.141592), FLOOR(3.141592);	O
FOOR(숫자)	지정한 숫자와 가까운 작은 정수		
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_MONTHS((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	select date_add(now(), interval 1 day), date_add(now(), interval 1 hour), date_add('20211018', interval 1 month);	*MYSQL
*mysql 함수 DATE_SUB(date, INTERVAL expr unit)	시간빼기	select date_sub(now(), interval 1 day), date_sub(now(), interval 1 hour), date_sub('20211018', interval 1 month);	*MYSQL

MONTHS_BETWEEN(날짜 데이터1, 날짜 데이터 2)	두 날짜 간의 개월수 차이를 구함.	select months_between(hiredate,sysdate) from emp;	없음
*mysql 함수 DATEDIFF(날짜데이터 1, 날짜데이터2)	두날짜 사이의 차이를 구함		*MYSQL SELECT datediff(NOW(), '20220620')
NEXT_DAY(날짜데이터, 요일문자)	둘아오날 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');
TURUNC(날짜, 포맷)	날짜 버림		
TO_CHAR		SELECT TO_CHAR(SYSDATE(), 'YYYY/MM/DD HH24:MI:SS')	없음 DATE_FORMAT 함수 써야함
*MY SQL 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 함수 써야함

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, 'O','X') FROM EMP	X
*MYSQL IFNULL(비교군, '리턴내용')	NULL 일 경우 사용자 지정내용 리턴	*MYSQL SELL 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], [조건N],[데이터가 조건N와 일치할때 반환할 결과]), [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 EMPNO, ENAME, JOB, SAL, DECODE(JOB , 'MANAGER', SAL*1.1, 'SALESMAN', SAL*1.1, 'ANALYST', SAL , SAL * 1.03)</p> <p>FROM EMP;</p>	<p>없음 IF나 CASE문 쓸것</p> <p>ENCODE(문자열,키) 암호화 함수 DECODE(문자열, 키) 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 바로 반환할 결과] END</p>	<pre>SELECT EMPNO, CASE JOB WHEN 'MANAGER' THEN SAL*1.1 WHEN 'SALESMAN' THEN SAL*1.05 ELSE SAL*1.03 END FROM EMP;</pre>	O
* MYSQL IF문 다중행함수	<p>오라클 함수의 decode함수는 조건과 같을 경우만 비교하지만, 조건문을 적어 줄수있다.</p> <p>IF (조건, 조건이 일치할경우 VALUE, 조건이 일치하지 않을 경우 VALUE)</p>	<pre>*MY SQL SELECT deptno, IF(deptno <= 20,'RESEARCH','OPERATIONS') deptno as " Dept Name" FROM dept;</pre>	

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

• 반올림

호스트: 127.0.0.1
데이터베이스: bank
쿼리*

```

1 SELECT '1234.5678', ROUND(1234.5678,1), ROUND (1234.5678, -1)
2 FROM DUAL;
3
4 /* 반 올 림   ROUND
5    ROUND(대 상 , 반 올 림 위 치 )
6    반 올 림 위 치 가 소 수 점 아 래 에 서 는 양 수로 그 자 리 까 지로 반 올 림 .
7    반 올 림 위 치 가 소 수 점 위 에 서 는 (정 수 자 리 에 서 는 ) 음 수로 그 자 리 에 서 반 올 림 .
8
9 */
```

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 '1234.5678', ROUND(1234.5678,1), ROUND (1234.5678, -1)
FROM DUAL;
```

```
/* 반올림 ROUND
ROUND(대상, 반올림위치)
반올림 위치가 소수점 아래에서는 양수로 그 자리까지로 반올림.
반올림 위치가 소수점 위에서는 (정수자리에서는) 음수로 그 자리 에서 반올림.
*/
```

• Trunc : 버리기

```
1  /* Truncate : 버리기
2  반올림과 똑같은 방식
3  SQL만 truncate, 나머지는 TRUNC
4  */
5
6  SELECT '1234.5678' AS 숫자 , TRUNCATE(1234.5678,1)
7  FROM DUAL;
```

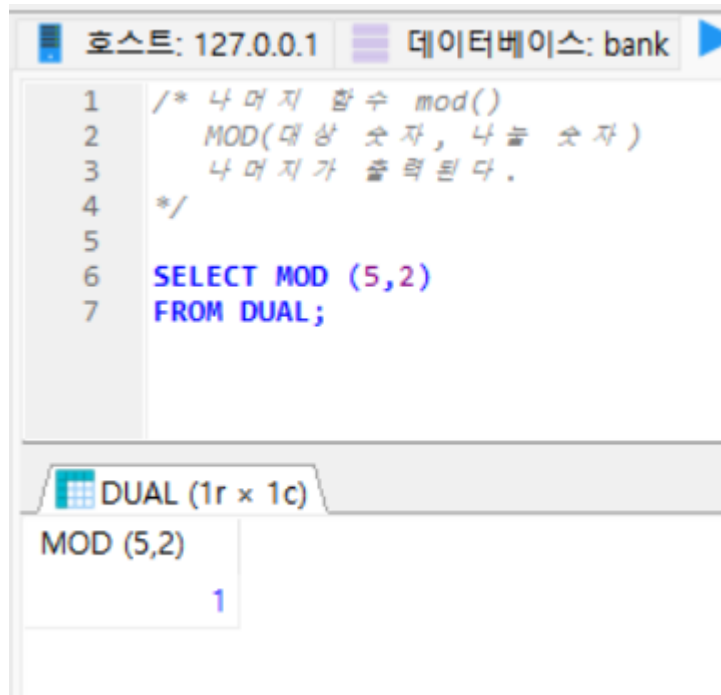
DUAL (1r x 2c)

숫자	TRUNCATE(1234.5678,1)
1234.5678	1,234.5

```
/* Truncate : 버리기
반올림과 똑같은 방식
SQL만 truncate, 나머지는 TRUNC
*/
```

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

• 나머지 연산자



Quiz 25


사원 번호와 사원 번호가 홀수이면 1, 짝수이면 0을 출력

7839	1
7698	0
7782	0
:	:

```

1  /* 나머지 함수 mod()
2     MOD(대상 숫자, 나눌 숫자)
3     나머지가 출력된다.
4  */
5
6  SELECT EMPNO, MOD(empno,2)
7  FROM emp;
8

```

 emp (14r x 2c)

EMPNO	MOD(empno,2)
7,839	1
7,698	0
7,782	0
7,566	0
7,654	0
7,499	1
7,844	0

```

/* 나머지 함수 mod()
   MOD(대상 숫자, 나눌 숫자)
   나머지가 출력된다.
*/


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

```

사원 번호가 짝수인 사원들의 사원 번호와 이름을 출력하는 쿼리

EMPNO	ENAME
7698	BLAKE
7782	CLARK
7566	JONES
7654	MARTIN
:	:

```
1 SELECT empno, ENAME
2 FROM emp
3 WHERE MOD(empno,2)=0;
```

 emp (10r x 2c)

empno	ENAME
7,698	BLAKE
7,782	CLARK
7,566	JONES
7,654	MARTIN
7,844	TURNER
7,900	JAMES
7,902	FORD
7,788	SCOTT
7,876	ADAMS
7,934	MILLER

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

※ 조건절을 달때는 where 사용

- 몫을 구하는 함수

다음의 쿼리는 10을 3으로 나눈 몫을 출력하는 쿼리입니다.

FLOOR(10/3)
3

```

1 SELECT FLOOR (10/3)
2 FROM DUAL;
3
4 /* 나눗셈의 몫을 구하는 방법
5 FLOOR (나눠야 하는 수 / 나눌 수 )
6 */

```

DUAL (1r x 1c)
FLOOR (10/3)
3

- CASE 구문

1	/* CASE 구문
2	
3	SQL에서 케이스구문은
4	CASE WHEN (조건) THEN (결과)
5	반복
6	나머지 ELSE
7	END as
8	*/
9	
10	
11	SELECT ename, job, sal,
12	CASE
13	when sal >= 3000 then 500
14	When sal >= 2000 then 300
15	when sal >= 1000 then 200
16	ELSE 0
17	END AS Bonus
18	FROM emp;
19	

emp (14r x 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	200

/* CASE 구문

SQL에서 케이스구문은

CASE WHEN (조건) THEN (결과)

반복

나머지 ELSE

END as

*/

SELECT ename, job, sal,

CASE

when sal >= 3000 then 500

When sal >= 2000 then 300

when sal >= 1000 then 200

ELSE 0

END AS Bonus

FROM emp;

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

ENAME	JOB	보너스
KING	PRESIDENT	0
BLAKE	MANAGER	400
CLARK	MANAGER	400
JONES	MANAGER	400
MARTIN	SALESMAN	500
ALLEN	SALESMAN	500
TURNER	SALESMAN	500
JAMES	CLERK	400
WARD	SALESMAN	500
FORD	ANALYST	500
SMITH	CLERK	400
SCOTT	ANALYST	500
ADAMS	CLERK	400
MILLER	CLERK	400


```

1  SELECT ename, job,
2         CASE
3             when job IN ("SALESMAN","ANALYST") then 500
4             when job IN ("CLERK","MANAGER") then 400
5             ELSE 0
6         END AS Bonus
7  FROM emp;

```

 emp (14r x 3c)

ename	job	Bonus
KING	PRESIDENT	0
BLAKE	MANAGER	400
CLARK	MANAGER	400
JONES	MANAGER	400
MARTIN	SALESMAN	500
ALLEN	SALESMAN	500
TURNER	SALESMAN	500

```

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;

```