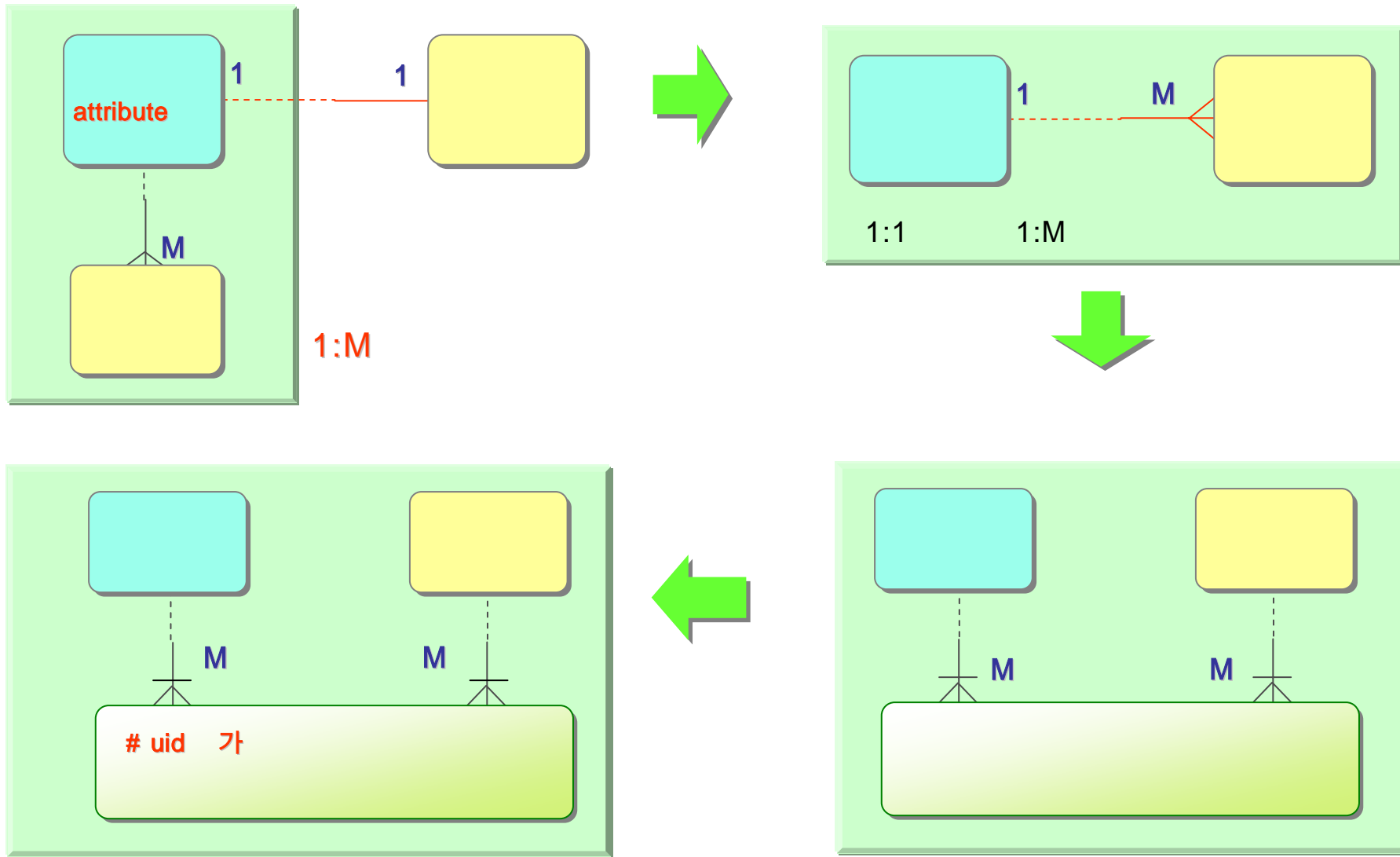
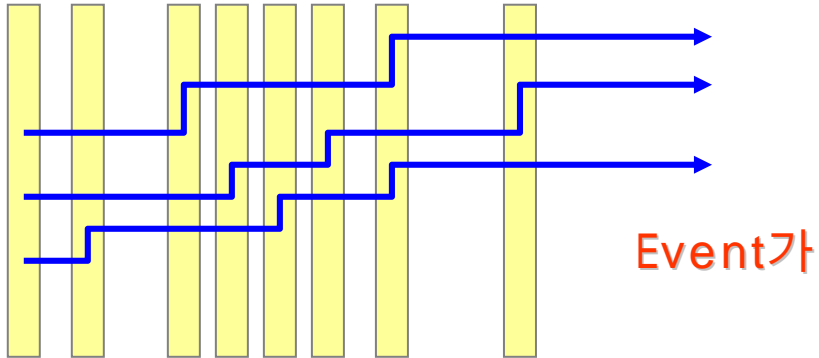


이력관리 시의 관계 변화



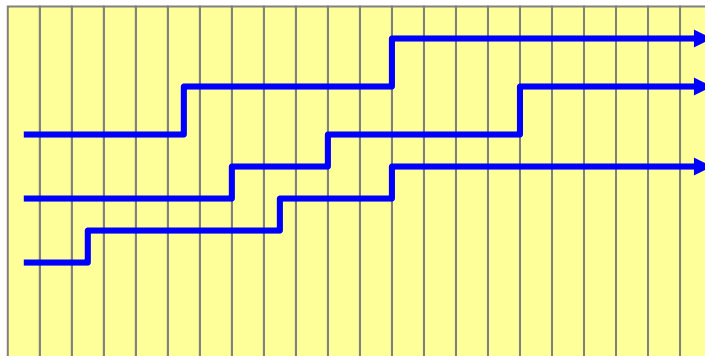
전통적인 이력관리 방법

EVENT



가?

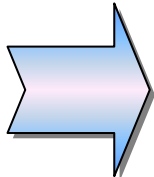
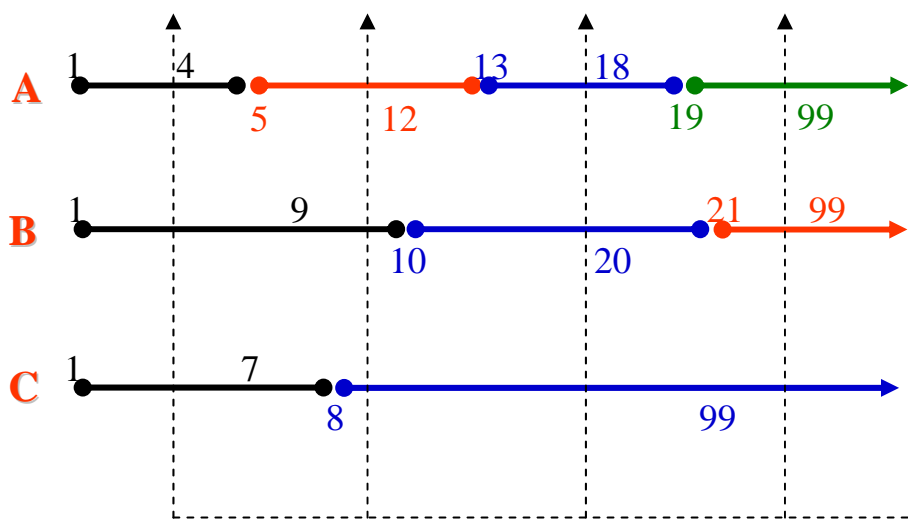
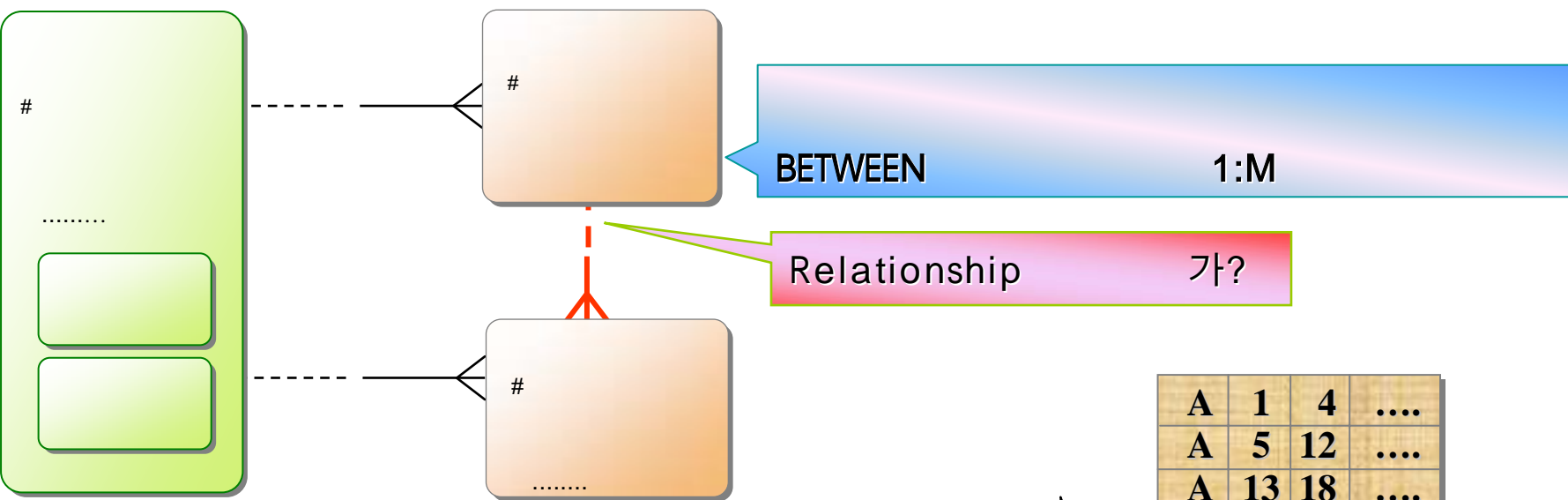
DAILY



가

가

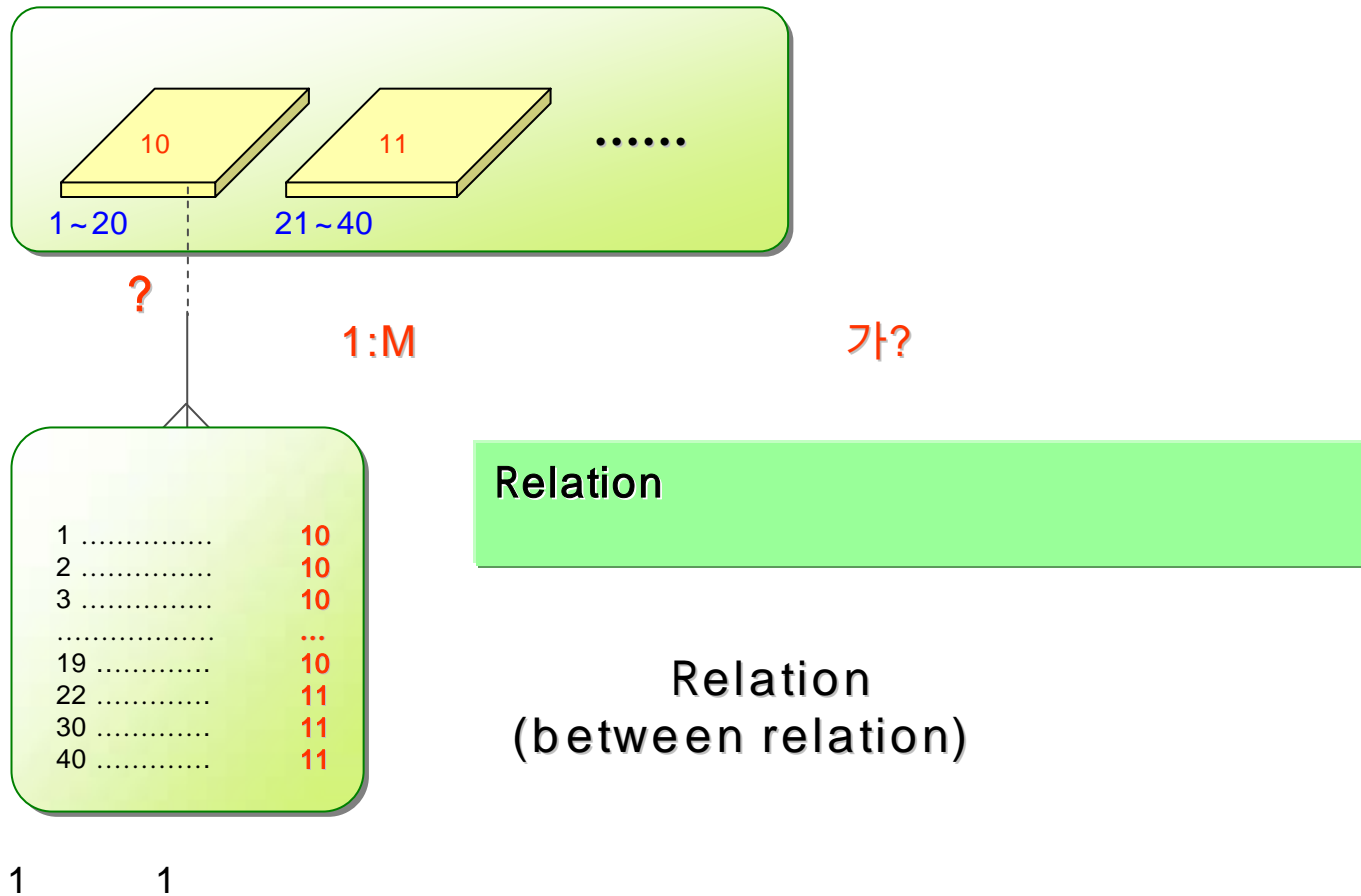
BETWEEN Relationship



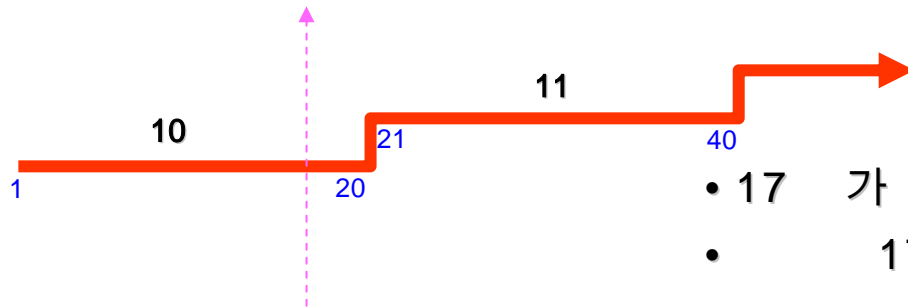
A	1	4
A	5	12
A	13	18
A	19	99
B	1	9
B	10	20
B	21	99
C	1	7
C	8	99

가

BETWEEN RELATION의 개념



이력선분의 기하학적 고찰



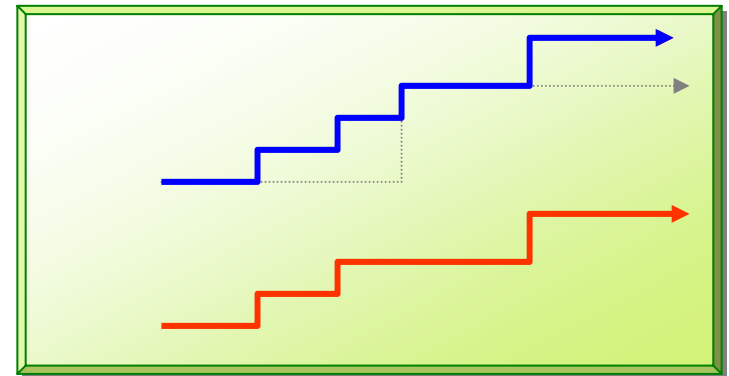
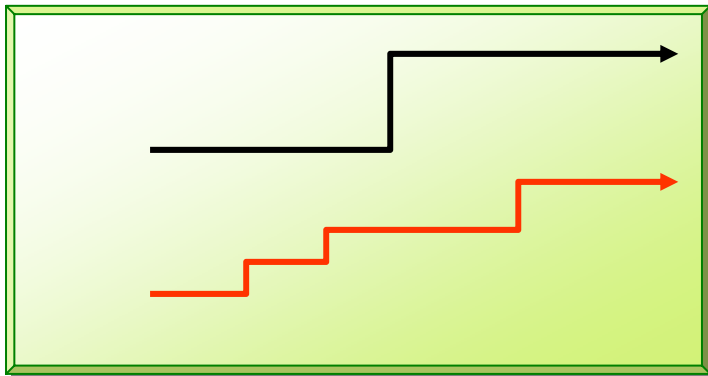
• 17 가 = !
 • 17 !!!
 •
 •



- : $\leq 17 \leq$
- : 17 Between and

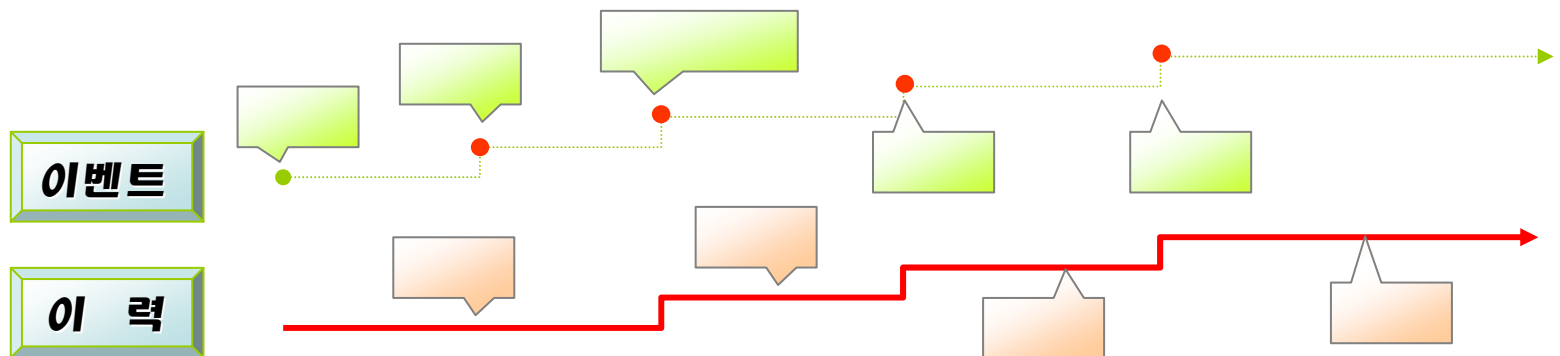
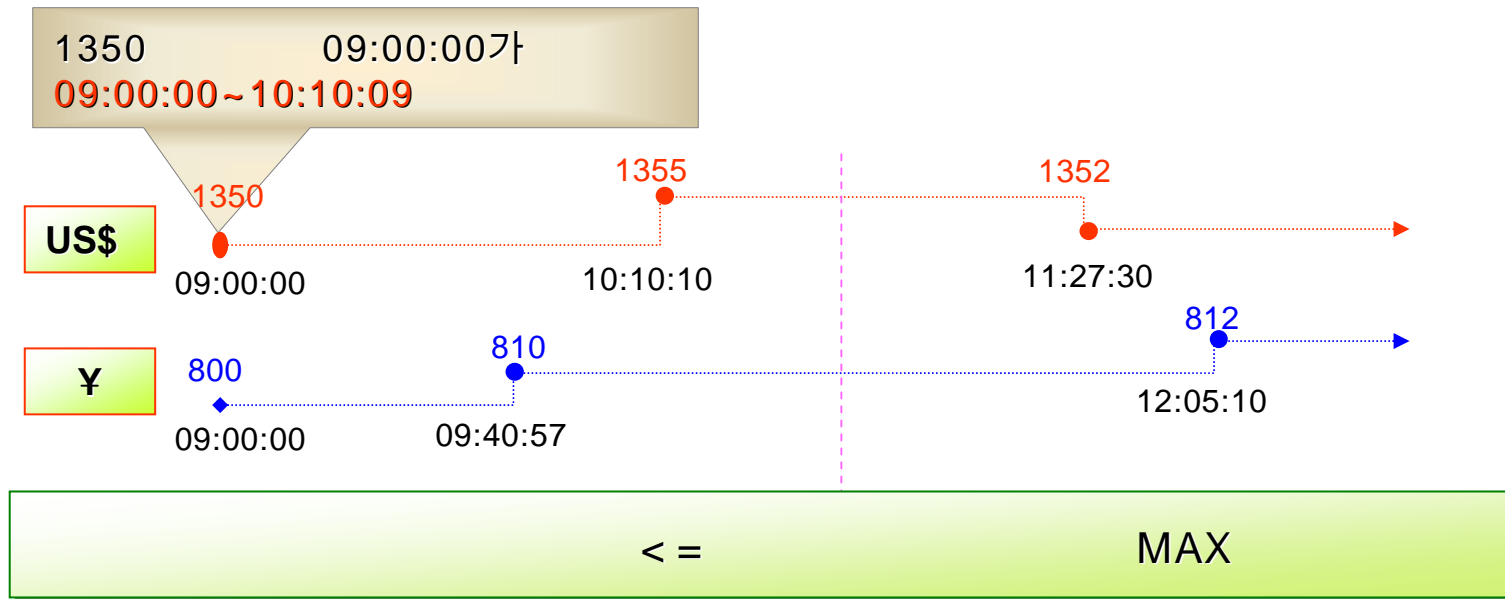


Package



=

이력(선분)과 EVENT(점)의 차이

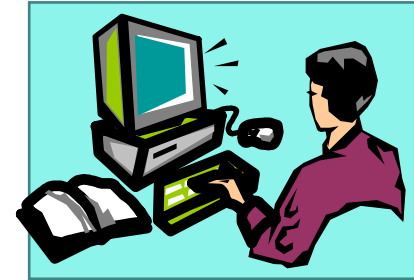


점 이력 모델

			...
7788	10	1999/ 02/ 04	...
7788	30	2000/ 07/ 21	NULL
7788	20	2002/ 05/ 15	...
8123	40	2000/ 03/ 23	
8123	30	2001/ 11/ 05	
8123	10	2002/ 09/ 17	
...	



' (: 2002 3 1)



1

SELECT
FROM
WHERE = (select MAX() from where < '2002/03/01');

Subquery (1)

2

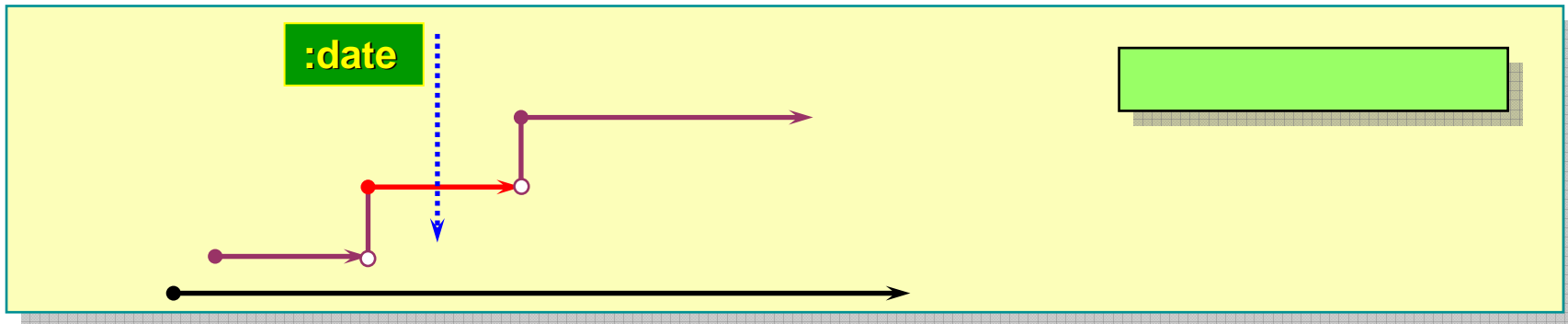
SELECT
FROM
WHERE ROWID = (select /* + index_desc(idx_) */ ROWID RID
from where < '2002/03/01 and rownum <= 1);

Subquery (2)

선분이력(한편 넣기)

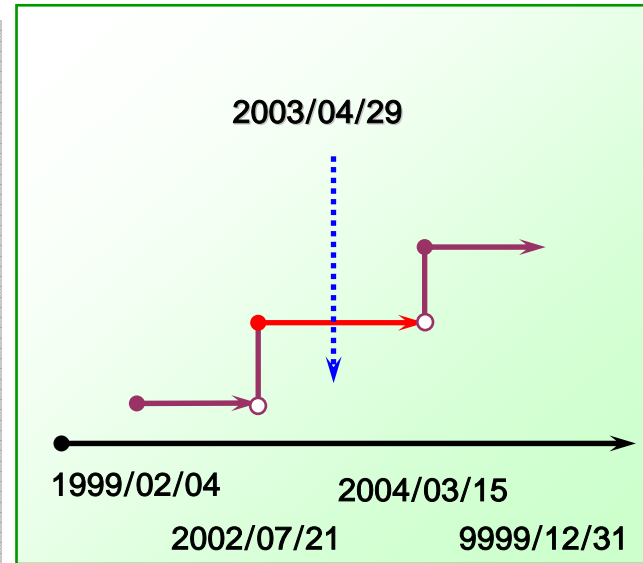
- ' (Day)' 가 ,
- (time)

- where :date >= and :date <=
- where :date between and



선분이력(한편 넣기 예제)

				~ ~ ~
7788	10	1999/ 02/ 04	2002/ 07/ 20	~ ~ ~
7788	30	2002/ 07/ 21	2004/ 05/ 14	NULL
7788	20	2004/ 05/ 15	9999/ 12/ 31	~ ~ ~
8123	40	2000/ 03/ 23	2001/ 11/ 04	
8123	30	2001/ 11/ 05	2002/ 09/ 16	
8123	10	2002/ 09/ 17	9999/ 12/ 31	
~ ~ ~	~ ~ ~	~ ~ ~	~ ~ ~	



'7788'

, 2001 4 29

가?

```
SELECT
FROM
WHERE
AND
AND
```

,
 <= '2003/04/29'
 >= '2003/04/29'
 = '7788'

```
SELECT
FROM
WHERE '2003/04/29' BETWEEN
AND
AND
```

,
 AND
 = '7788'

선분이력(양편 넣기)



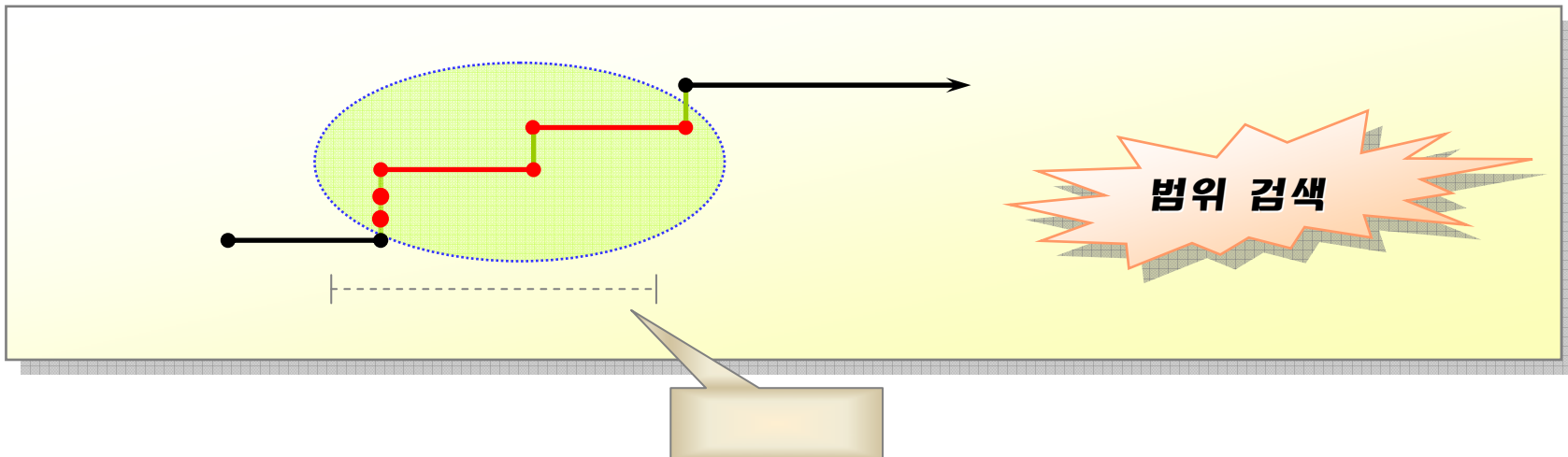
:

Query



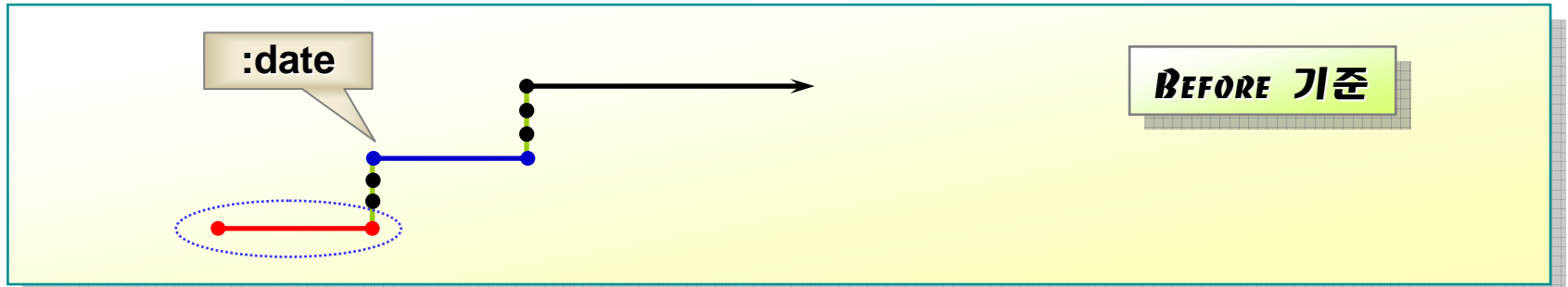
where

between : 1 and : 2

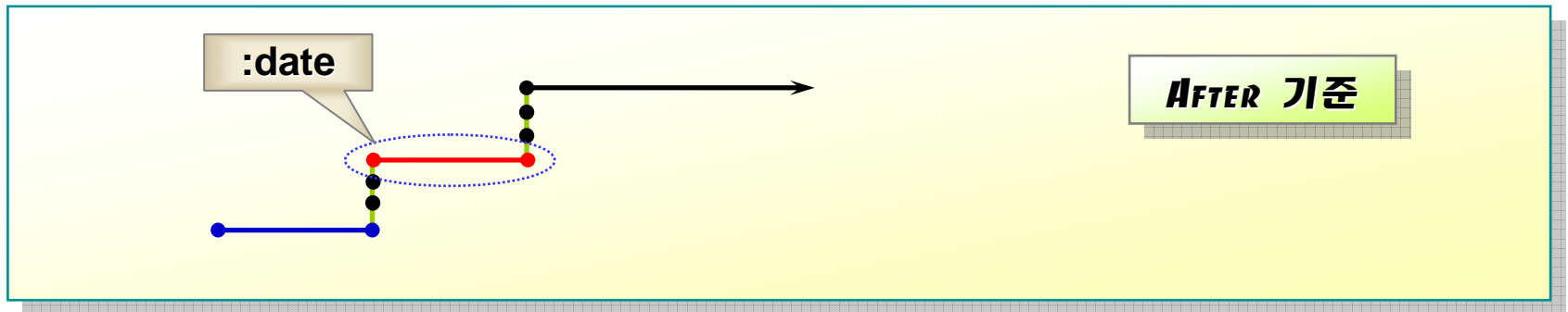


선분이력(양편 넣기)

- where :date $\cdots >$ and :date \leq



- where :date $=>$ and :date $\cdots <$

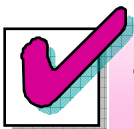
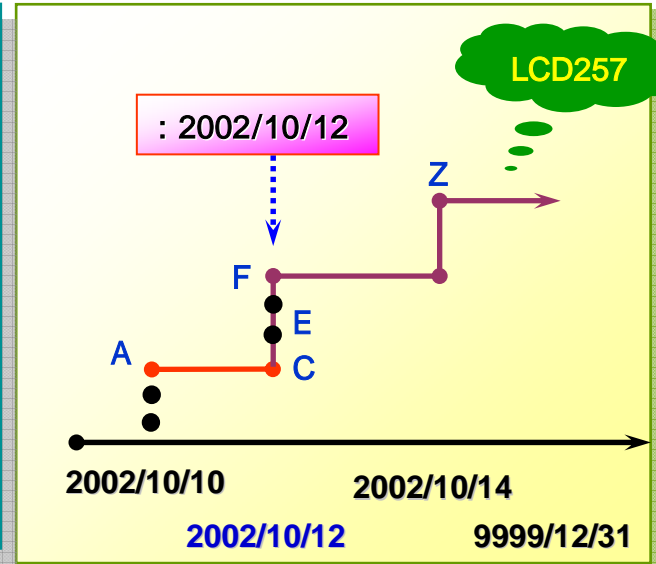


=

선분이력(양편 넣기 예제)

부품상태변경이력

CDR788	E	2002/10/12	9999/12/31	5
LCD257	A	2002/10/10	2002/10/12	1
LCD257	C	2002/10/12	2002/10/12	1
LCD257	E	2002/10/12	2002/10/12	2
LCD257	F	2002/10/12	2002/10/14	3
LCD257	Z	2002/10/14	9999/12/31	1
~~~	~~~	~~~	~~~	~~~



1.

'LCD257'

? (2002/10/12 )

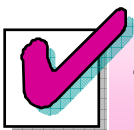
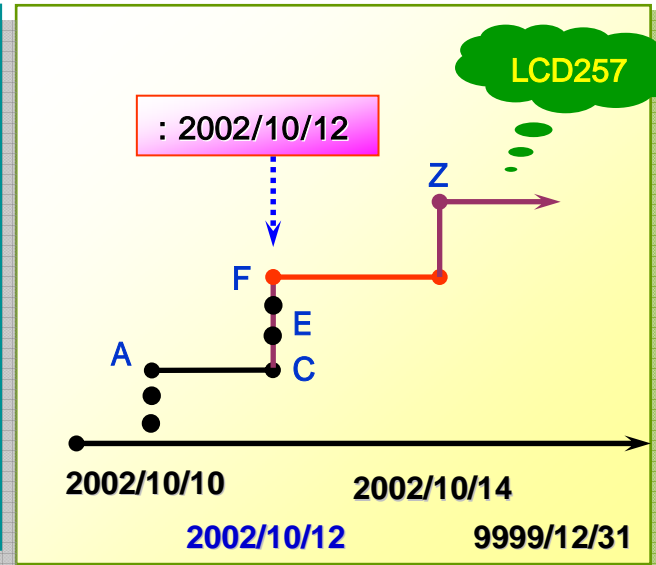
```

SELECT      ,      ,      FROM
WHERE      < '2002/10/12'
AND      => '2002/10/12'
AND      = 'LCD257'
    
```

# 선분이력(양편 넣기 예제)

## 부품상태변경이력

CDR788	E	2002/10/12	9999/12/31	5
LCD257	A	2002/10/10	2002/10/12	1
LCD257	C	2002/10/12	2002/10/12	1
LCD257	E	2002/10/12	2002/10/12	2
LCD257	F	2002/10/12	2002/10/14	3
LCD257	Z	2002/10/14	9999/12/31	1
~~~	~~~	~~~	~~~	~~~



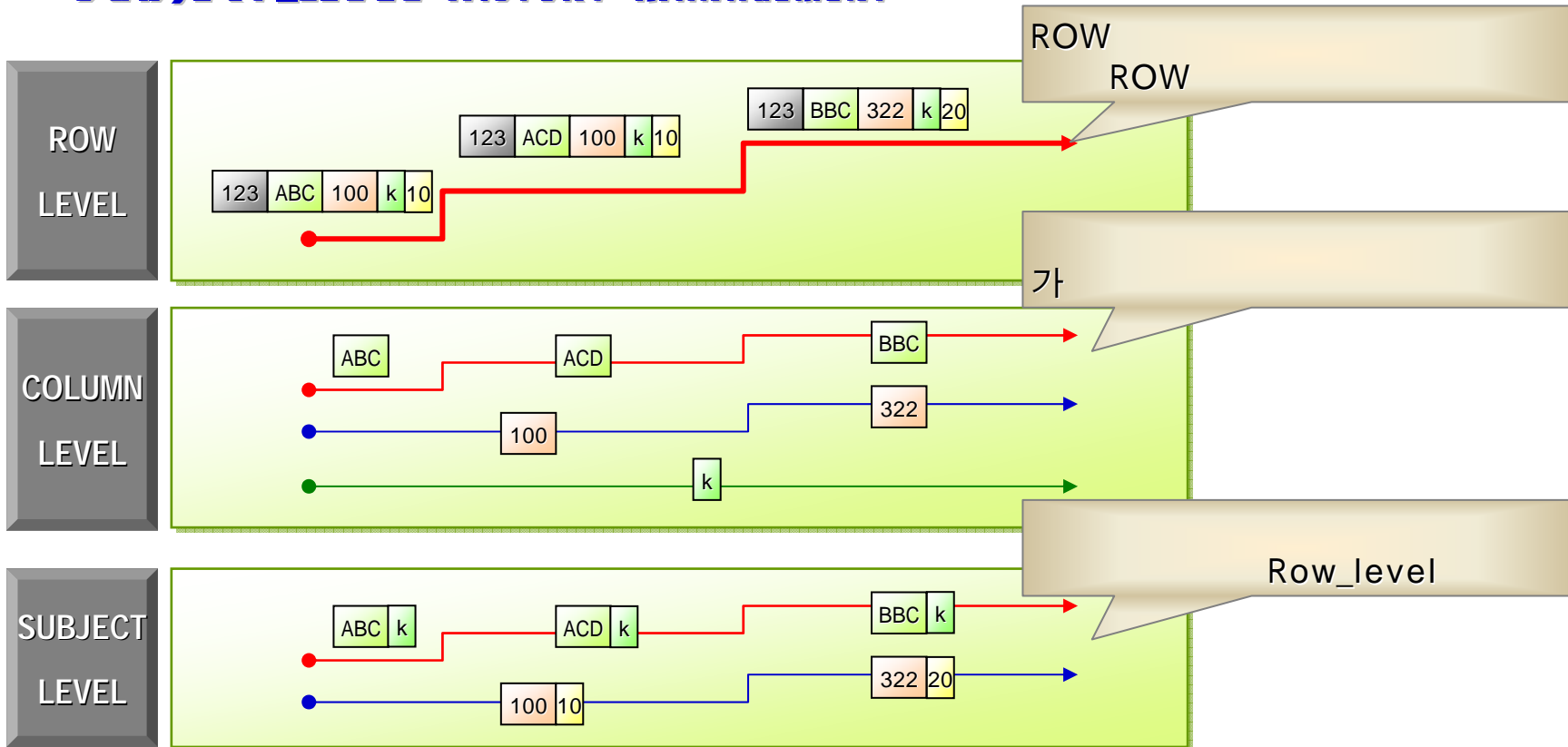
1. 'LCD257' ? (2002/10/12)

```

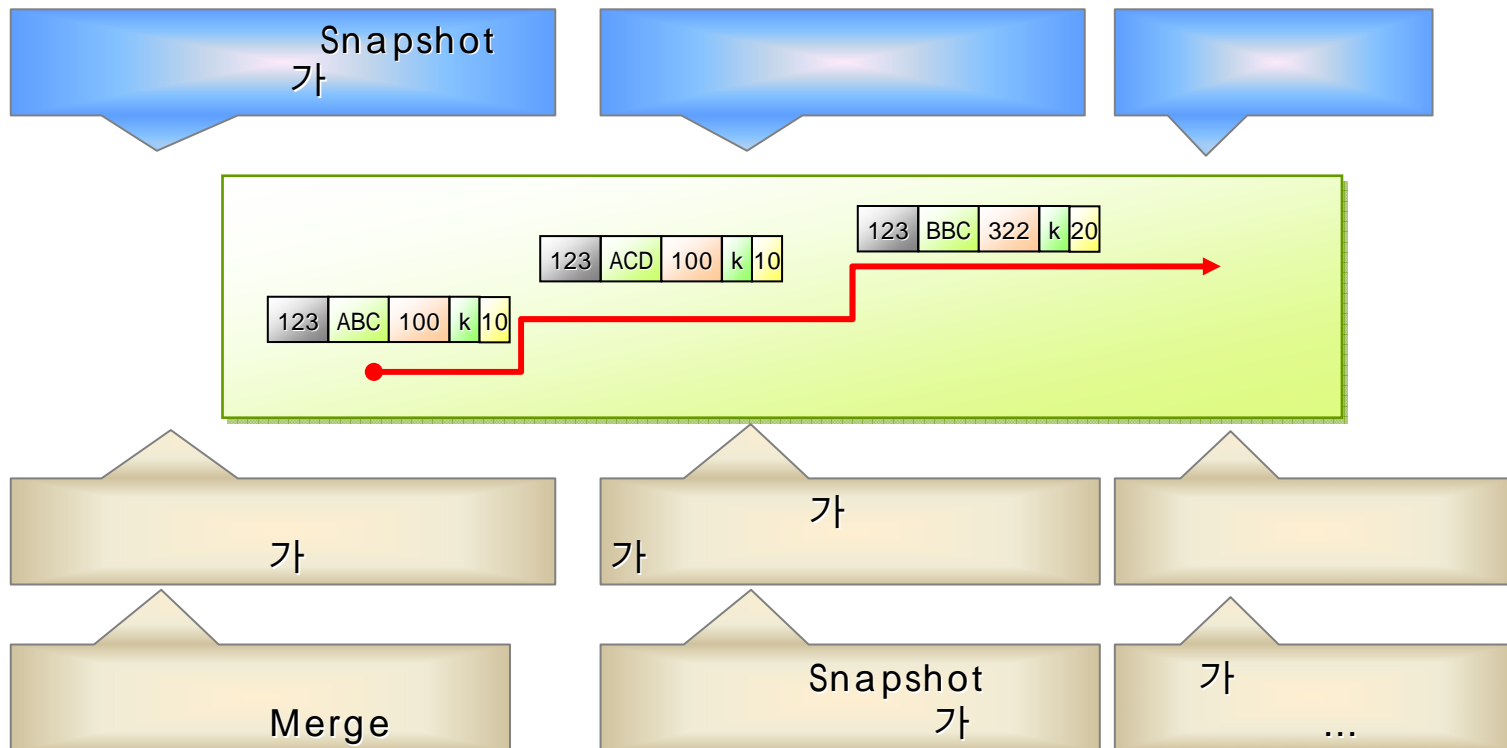
SELECT      ,      ,      FROM
WHERE      =< '2002/10/12'
AND        > '2002/10/12'
AND        = 'LCD257'
    
```

이력관리 유형

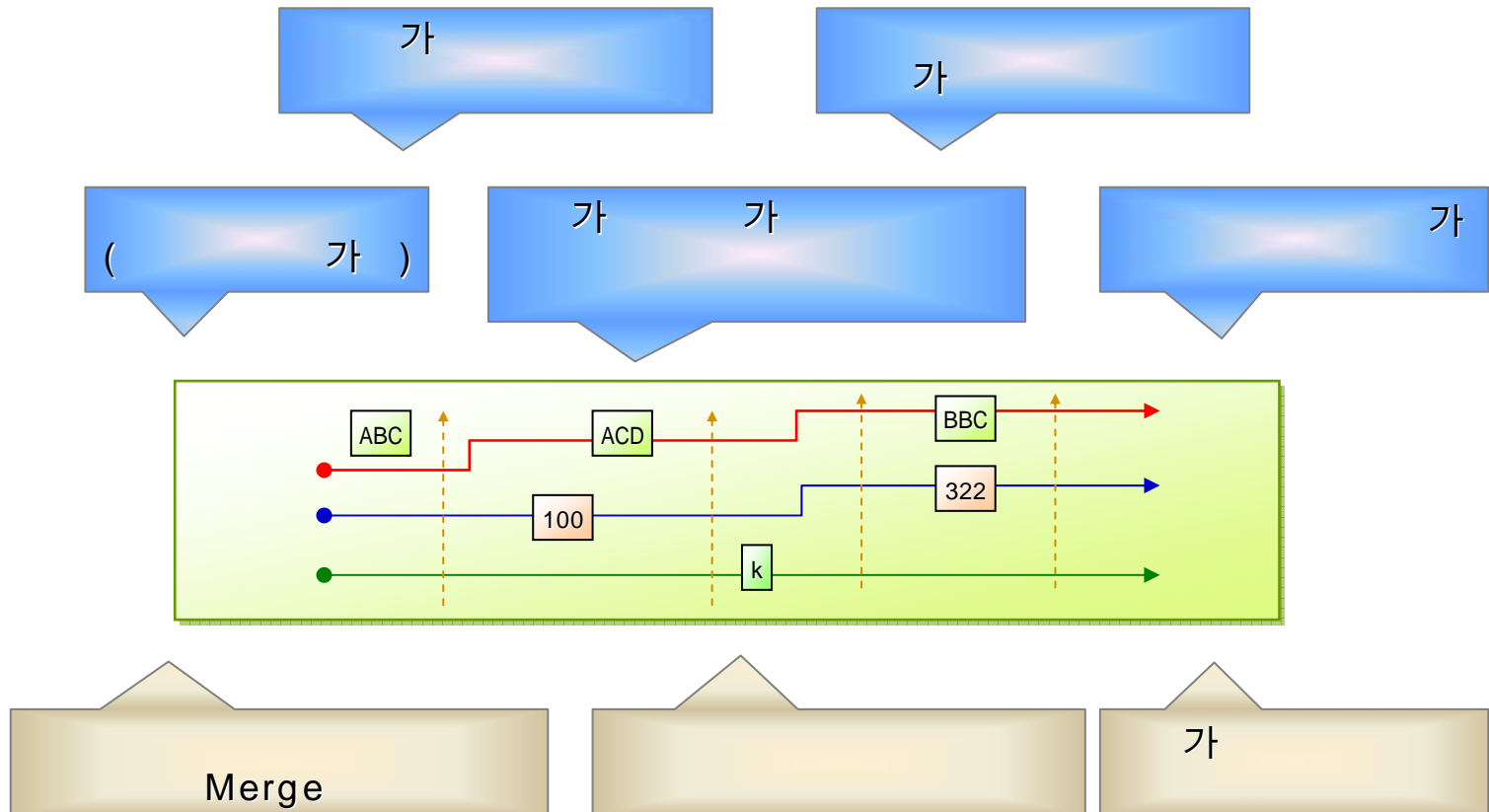
- **ROW_LEVEL HISTORY MANAGEMENT**
- **COLUMN_LEVEL HISTORY MANAGEMENT**
- **SUBJECT_LEVEL HISTORY MANAGEMENT**



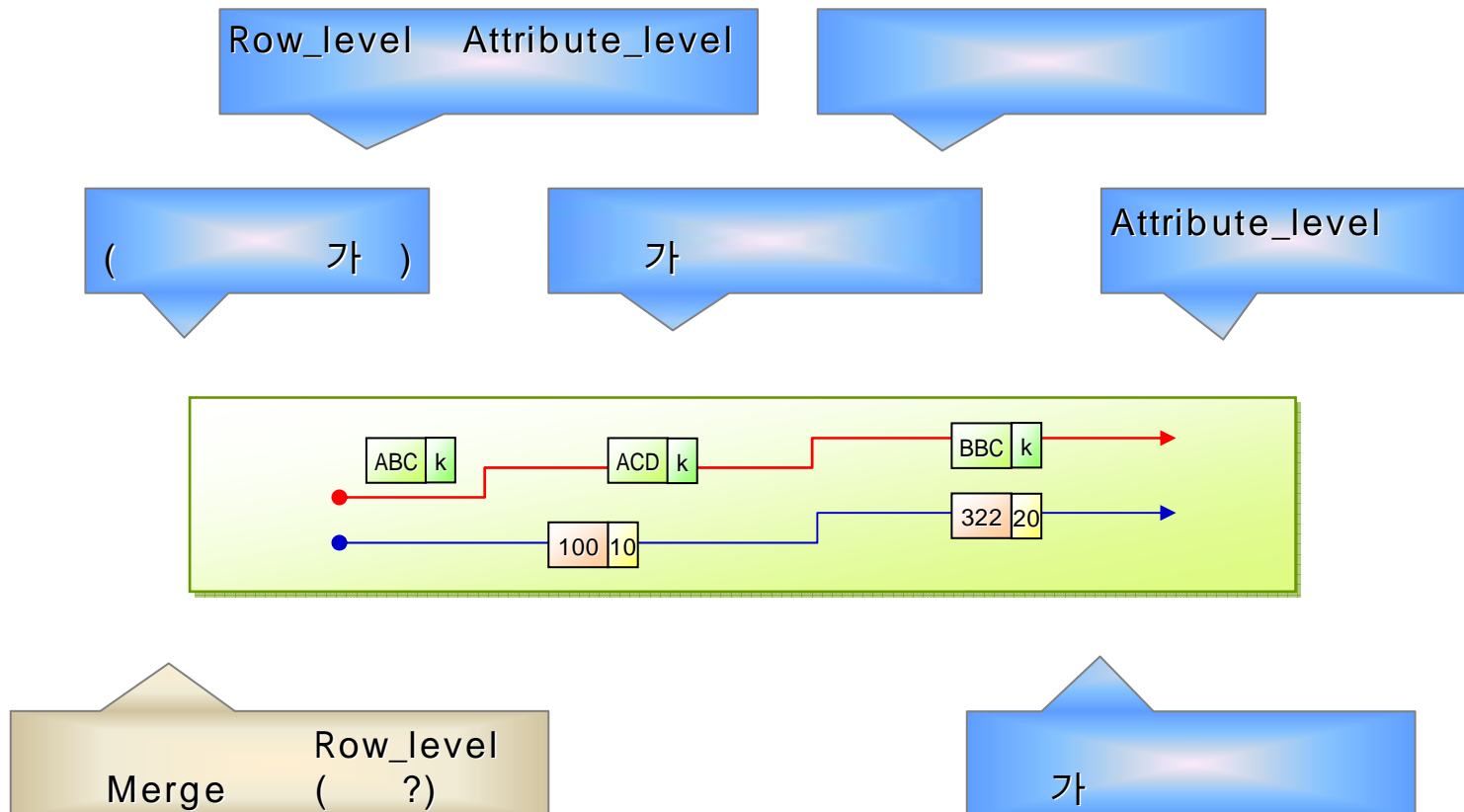
ROW_LEVEL HISTORY



COLUMN_LEVEL HISTORY



SUBJECT_LEVEL HISTORY



물리 설계 고려사항 - 결합인덱스

Column_name BETWEEN : AND :

:Value BETWEEN AND

=

<= :value AND
>= :value

:value가 520

:value가 10

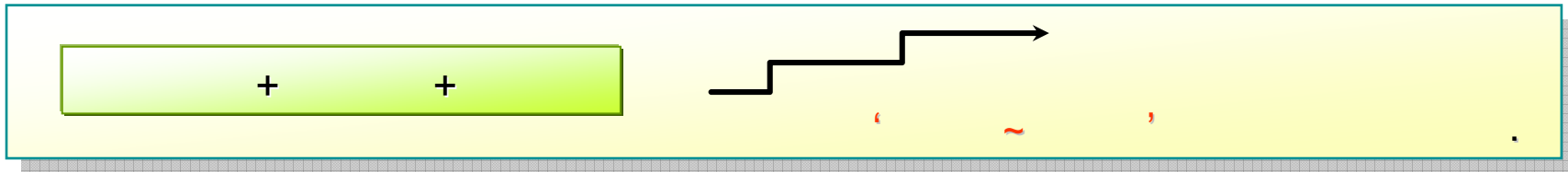
1	4
5	12
13	18
19	99
100	110
111	120
121	199
200	207
208	289
290	320
321	400
401	500
501	600
601	700
701	9999

4	1
12	5
18	13
99	19
110	100
120	111
199	121
207	200
289	208
320	290
400	321
500	401
600	501
700	601
9999	701

◆ +
◆ +
◆ rownum = 1 가
◆ Default
(; '99991231')

물리 설계 고려사항 - PRIMARY KEY

가?



다양한 형태 존재



Unique ?

Unique ?

Unique ?

{

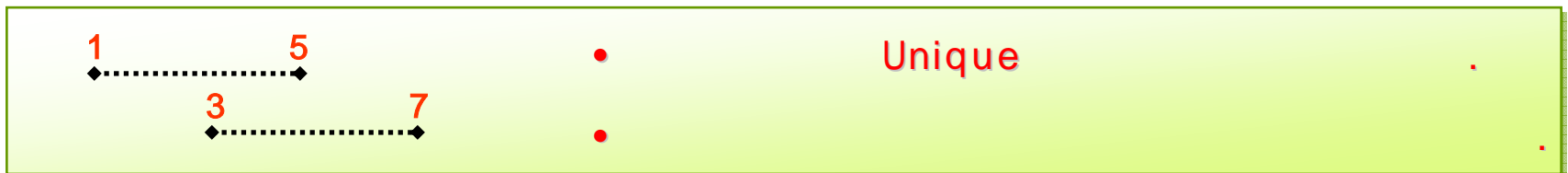
가

Unique

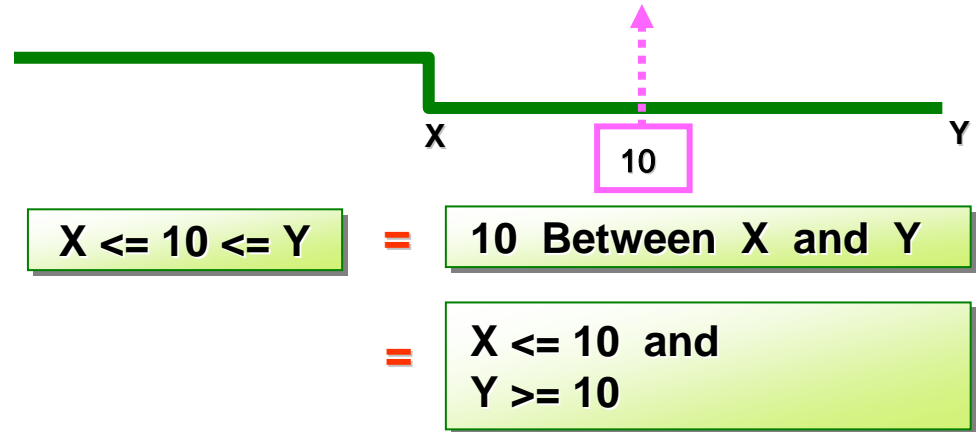
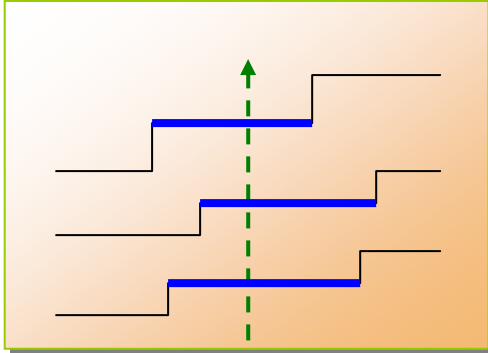
, DBMS가 check

Constraints

.



물리 설계 고려사항 - 종료점 처리



가

위 두 문장은 동일하다. (속도도 동일)

☐
☐
☐
☐
☐
☐
☐

NULL

가 가

(; 9999/12/31)

가 TABLE creation Default constraints