

2. 데이터베이스 활용과 클라우드 서버 환경 준비

Section 1 . 데이터베이스 시스템 / 데이터 모델 / SQL 기초 실습

SQL 고급



목차

01 **내장 함수**

02 **뷰**

03 **인덱스**

목표

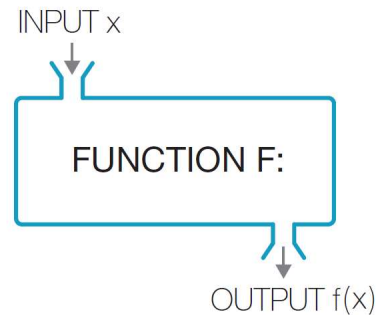
- 내장 함수의 의미를 알아본다.
- 자주 사용되는 내장 함수 몇 가지를 직접 실습해본다.
- 뷰의 의미를 알아보고, 뷰를 직접 생성, 수정, 삭제해본다.
- 데이터베이스의 저장 구조와 인덱스의 관계를 알아본다.
- 인덱스를 직접 생성, 수정, 삭제해본다.

01. 내장 함수

1. SQL 내장 함수
2. NULL 값 처리
3. 행 번호 출력

1. SQL 내장 함수

- SQL에서는 함수의 개념을 사용
- 수학의 함수와 마찬가지로 특정 값이나 열의 값을 입력 받아 그 값을 계산하여 결과 값을 돌려줌



함수의 원리

- SQL의 함수는 DBMS가 제공하는 내장 함수(built-in function), 사용자가 필요에 따라 직접 만드는 사용자 정의 함수(user-defined function)로 나뉨

1. SQL 내장 함수

- SQL 내장 함수는 상수나 속성 이름을 입력 값으로 받아 단일 값을 결과로 반환함
- 모든 내장 함수는 최초에 선언될 때 유효한 입력 값을 받아야 함

MySQL에서 제공하는 주요 내장 함수

| 구분 | | 함수 |
|------------------|--------------|--|
| 단일행 함수 | 숫자 함수 | ABS, CEIL, COS, EXP, FLOOR, LN, LOG, MOD, POWER, RAND, ROUND, SIGN, TRUNCATE |
| | 문자 함수(문자 반환) | CHAR, CONCAT, LEFT, RIGHT, LOWER, UPPER, LPAD, RPAD, LTRIM, RTRIM, REPLACE, REVERSE, RIGHT, SUBSTR, TRIM |
| | 문자 함수(숫자 반환) | ASCII, INSTR, LENGTH |
| | 날짜·시간 함수 | ADDDATE, CURRENT_DATE, DATE, DATEDIFF, DAYNAME, LAST_DAY, SYSDATE, TIME |
| | 변환 함수 | CAST, CONVERT, DATE_FORMAT, STR_TO_DATE |
| | 정보 함수 | DATABASE, SCHEMA, ROW_COUNT, USER, VERSION |
| | NULL 관련 함수 | COALESCE, ISNULL, IFNULL, NULLIF |
| 집계 함수 | | AVG, COUNT, MAX, MIN, STD, STDDEV, SUM |
| 윈도우 함수(혹은 분석 함수) | | CUME_DIST, DENSE_RANK, FIRST_VALUE, LAST_VALUE, LEAD, NTILE, RANK, ROW_NUMBER |

1. SQL 내장 함수

❖ 숫자 함수

숫자 함수의 종류

| 함수 | 설명 |
|---------------------|---|
| ABS(숫자) | 숫자의 절댓값을 계산 $ABS(-4.5) = > 4.5$ |
| CEIL(숫자) | 숫자보다 크거나 같은 최소의 정수 $CEIL(4.1) = > 5$ |
| FLOOR(숫자) | 숫자보다 작거나 같은 최소의 정수 $FLOOR(4.1) = > 4$ |
| ROUND(숫자, m) | 숫자의 반올림, m은 반올림 기준 자릿수 $ROUND(5.36, 1) = > 5.40$ |
| LOG(n, 숫자) | 숫자의 자연로그 값을 반환 $LOG(10) = > 2.30259$ |
| POWER(숫자, n) | 숫자의 n제곱 값을 계산 $POWER(2, 3) = > 8$ |
| SQRT(숫자) | 숫자의 제곱근 값을 계산(숫자는 양수) $SQRT(9.0) = > 3.0$ |
| SIGN(숫자) | 숫자가 음수면 -1, 0이면 0, 양수면 1 $SIGN(3.45) = > 1$ |

1. SQL 내장 함수

❖ 수학 함수

■ ABS 함수 : 절댓값을 구하는 함수

1 -78과 +78의 절댓값을 구하시오.

■ ROUND 함수 : 반올림한 값을 구하는 함수

2 4.875를 소수 첫째 자리까지 반올림한 값을 구하시오.

■ 숫자 함수의 연산

3 고객별 평균 주문 금액을 백 원 단위로 반올림한 값을 구하시오.

1. SQL 내장 함수

❖ 문자 함수

문자 함수의 종류

| 반환 구분 | 함수 | 설명 |
|---|--------------------------|---|
| 문자값 반환 함수 s : 문자열 c : 문자 n : 정수 k : 정수 | CONCAT(s1,s2) | 두 문자열을 연결, CONCAT(' 알라딘', ' 서점') => '알라딘 서점' |
| | LOWER(s) | 대상 문자열을 모두 소문자로 변환, LOWER('MR. SCOTT') => 'mr. scott' |
| | LPAD(s,n,c) | 대상 문자열의 왼쪽부터 지정한 자리수까지 지정한 문자로 채움 LPAD('Page 1', 10, '*') => '****Page 1' |
| | REPLACE(s1,s2,s3) | 대상 문자열의 지정한 문자를 원하는 문자로 변경 REPLACE('JACK & JUE', 'J', 'BL') => 'BLACK & BLUE' |
| | RPAD(s,n,c) | 대상 문자열의 오른쪽부터 지정한 자리수까지 지정한 문자로 채움 RPAD('AbC', 5, '*') => 'AbC**' |
| | SUBSTR(s,n,k) | 대상 문자열의 지정된 자리에서부터 지정된 길이만큼 잘라서 반환 SUBSTR('ABCDEFGH', 3, 4) => 'CDEF' |
| | TRIM(c FROM s) | 대상 문자열의 양쪽에서 지정된 문자를 삭제(문자열만 넣으면 기본값으로 공백 제거) TRIM('= ' FROM '=BROWNING=') => 'BROWNING' |
| | UPPER(s) | 대상 문자열을 모두 대문자로 변환 UPPER('mr. scott') => 'MR. SCOTT' |
| 숫자값 반환 함수 | ASCII(c) | 대상 알파벳 문자의 아스키 코드 값을 반환, ASCII('D') => 68 |
| | LENGTH(s) | 대상 문자열의 Byte 반환, 알파벳 1byte, 한글 3byte (UTF8) LENGTH('CANDIDE') => 7 |
| | CHAR_LENGTH(s) | 문자열의 문자 수를 반환, CHAR_LENGTH('데이터') => 3 |

1. SQL 내장 함수

❖ 문자 함수

■ REPLACE : 문자열을 치환하는 함수

4 도서제목에 야구가 포함된 도서를 농구로 변경한 후 도서 목록을 보이시오.

1. SQL 내장 함수

❖ 문자 함수

- **LENGTH** : 글자의 수를 세어주는 함수 (단위가 바이트(byte)가 아닌 문자 단위)

5 굿스포츠에서 출판한 도서의 제목과 제목의 글자 수를 확인하시오.
(한글은 2바이트 혹은 UNICODE 경우는 3바이트를 차지함)

- **SUBSTR** : 지정한 길이만큼의 문자열을 반환하는 함수

6 서점의 고객 중에서 같은 성(姓)을 가진 사람이 몇 명이나 되는지 성별 인원수를 구하시오.

1. SQL 내장 함수

❖ 날짜·시간 함수

날짜·시간 함수의 종류

| 함수 | 반환형 | 설명 |
|--------------------------------------|---------|---|
| STR_TO_DATE(string, format)) | DATE | 문자열(String) 데이터를 날짜형(Date)으로 반환 STR_TO_DATE('2019-02-14', '%Y-%m-%d') => 2019-02-14 |
| DATE_FORMAT(date, format) | STRING | 날짜형(Date) 데이터를 문자열(VARCHAR)로 반환 DATE_FORMAT('2019-02-14', '%Y-%m-%d') => '2019-02-14' |
| ADDDATE(date, interval) | DATE | DATE 형의 날짜에서 INTERVAL 지정한 시간만큼 더함 ADDDATE('2019-02-14', INTERVAL 10 DAY) => 2019-02-24 |
| DATE(date) | DATE | DATE 형의 날짜 부분을 반환 SELECT DATE('2003-12-31 01:02:03'); => 2003-12-31 |
| DATEDIFF(date1, date2) | INTEGER | DATE 형의 date1 – date2 날짜 차이를 반환 SELECT DATEDIFF('2019-02-14', '2019-02-04') => 10 |
| SYSDATE | DATE | DBMS 시스템상의 오늘 날짜를 반환하는 함수 SYSDATE() => 2018-06-30 21:47:01 |

1. SQL 내장 함수

❖ 날짜 함수

format의 주요 지정자

| 인자 | 설명 |
|----|-------------------------|
| %w | 요일 순서(0~6, Sunday=0) |
| %W | 요일(Sunday~Saturday) |
| %a | 요일의 약자(Sun~Sat) |
| %d | 1달 중 날짜(00~31) |
| %j | 1년 중 날짜(001~366) |
| %h | 12시간(01~12) |
| %H | 24시간(00~23) |
| %i | 분(0~59) |
| %m | 월 순서(01~12, January=01) |
| %b | 월 이름 약어(Jan~Dec) |
| %M | 월 이름(January~December) |
| %s | 초(0~59) |
| %Y | 4자리 연도 |
| %y | 4자리 연도의 마지막 2 자리 |

1. SQL 내장 함수

❖ 날짜 함수

7 서점은 주문일로부터 10일 후 매출을 확정한다. 각 주문의 확정일자를 구하시오.

1. SQL 내장 함수

❖ 날짜 함수

- STR_TO_DATE : 문자형으로 저장된 날짜를 날짜형으로 변환하는 함수
- DATE_FORMAT : 날짜형을 문자형으로 변환하는 함수

8 서점이 2014년 7월 7일에 주문 받은 도서의 주문번호, 주문일, 고객번호, 도서번호를 모두 보이시오. 단, 주문일은 '%Y-%m-%d' 형태로 표시한다.

1. SQL 내장 함수

❖ 날짜 함수

- SYSDATE : MySQL의 현재 날짜와 시간을 반환하는 함수

9 DBMS 서버에 설정된 현재 날짜와 시간, 요일을 확인 하시오.

2. NULL 값 처리

■ NULL 값이란?

- 아직 지정되지 않은 값
- NULL 값은 '0', '' (빈 문자), '' (공백) 등과 다른 특별한 값
- NULL 값은 비교 연산자로 비교가 불가능함
- NULL 값의 연산을 수행하면 결과 역시 NULL 값으로 반환됨

■ 집계 함수를 사용할 때 주의할 점

- 'NULL+숫자' 연산의 결과는 NULL
- 집계 함수 계산 시 NULL이 포함된 행은 집계에서 빠짐
- 해당되는 행이 하나도 없을 경우 SUM, AVG 함수의 결과는 NULL이 되며, COUNT 함수의 결과는 0.

2. NULL 값 처리

- NULL 값에 대한 연산과 집계 함수

Mybook

| bookid | price |
|--------|-------|
| 1 | 10000 |
| 2 | 20000 |
| 3 | NULL |

2. NULL 값 처리

■ NULL 값을 확인하는 방법 – IS NULL, IS NOT NULL

- NULL 값을 찾을 때는 '=' 연산자가 아닌 'IS NULL'을 사용,
- NULL이 아닌 값을 찾을 때는 '< >' 연산자가 아닌 'IS NOT NULL'을 사용함

Mybook

| bookid | price |
|--------|-------|
| 1 | 10000 |
| 2 | 20000 |
| 3 | NULL |

2. NULL 값 처리

- IFNULL : NULL 값을 다른 값으로 대체하여 연산하거나 다른 값으로 출력

IFNULL(속성, 값) /* 속성 값이 NULL이면 '값'으로 대체한다 */

10 이름, 전화번호가 포함된 고객목록을 보이시오. 단, 전화번호가 없는 고객은 '연락처없음'으로 표시한다.

3. 행번호 출력

- 내장 함수는 아니지만 자주 사용되는 문법
- MySQL에서 변수는 이름 앞에 @ 기호를 붙이며 치환문에는 SET과 := 기호를 사용함
- 자료를 일부분만 확인하여 처리할 때 유용함.

11 고객 목록에서 고객번호, 이름, 전화번호를 앞의 두 명만 보이시오.

02. 뷰

1. 뷰의 생성
2. 뷰의 수정
3. 뷰의 삭제

뷰

- 뷰(view) : 하나 이상의 테이블을 합하여 만든 가상의 테이블
- 뷰의 장점
 - 편리성 및 재사용성 : 자주 사용되는 복잡한 질의를 뷰로 미리 정의해 놓을 수 있음
→ 복잡한 질의를 간단히 작성
 - 보안성 : 사용자별로 필요한 데이터만 선별하여 보여줄 수 있고, 중요한 질의의 경우 질의 내용을 암호화할 수 있음
→ 개인정보(주민번호)나 급여, 건강 같은 민감한 정보를 제외한 테이블을 만들어 사용
 - 독립성 : 미리 정의된 뷰를 일반 테이블처럼 사용할 수 있기 때문에 편리하고, 사용자가 필요한 정보만 요구에 맞게 가공하여 뷰로 만들어 쓸 수 있음
→ 원본 테이블의 구조가 변해도 응용에 영향을 주지 않도록 하는 논리적 독립성 제공
- 뷰의 특징
 - 원본 데이터 값에 따라 같이 변함
 - 독립적인 인덱스 생성이 어려움
 - 삽입, 삭제, 갱신 연산에 많은 제약이 따름

1. 뷰의 생성

- 기본 문법

```
CREATE VIEW 뷰이름 [(열이름 [ ,...n ])]  
AS SELECT 문
```

- Book 테이블에서 '축구'라는 문구가 포함된 자료만 보여주는 뷰

- 위 SELECT 문을 이용해 작성한 뷰 정의문

1. 뷰의 생성

20 주소에 '대한민국'을 포함하는 고객들로 구성된 뷰를 만들고 조회하시오. 뷰의 이름은 vw_Customer로 설정하시오.

1. 뷰의 생성

21 Orders 테이블에 고객이름과 도서이름을 바로 확인할 수 있는 뷰를 생성한 후, '김연아' 고객이 구입한 도서의 주문번호, 도서이름, 주문액을 보이시오.

2. 뷰의 수정

❖ 기본 문법

```
CREATE OR REPLACE VIEW 뷰이름 [(열이름 [ ,...n ])]  
AS SELECT 문
```

22 [20]에서 생성한 뷰 vw_Customer는 주소가 대한민국인 고객을 보여준다. 이 뷰를 영국을 주소로 가진 고객으로 변경하시오. phone 속성은 필요 없으므로 포함시키지 마시오.

3. 뷰의 삭제

❖ 기본 문법

```
DROP VIEW 뷰이름 [ ,...n ];
```

23 앞서 생성한 뷰 vw_Customer를 삭제하시오.

연습문제

1. 다음에 해당하는 뷰를 작성하시오.

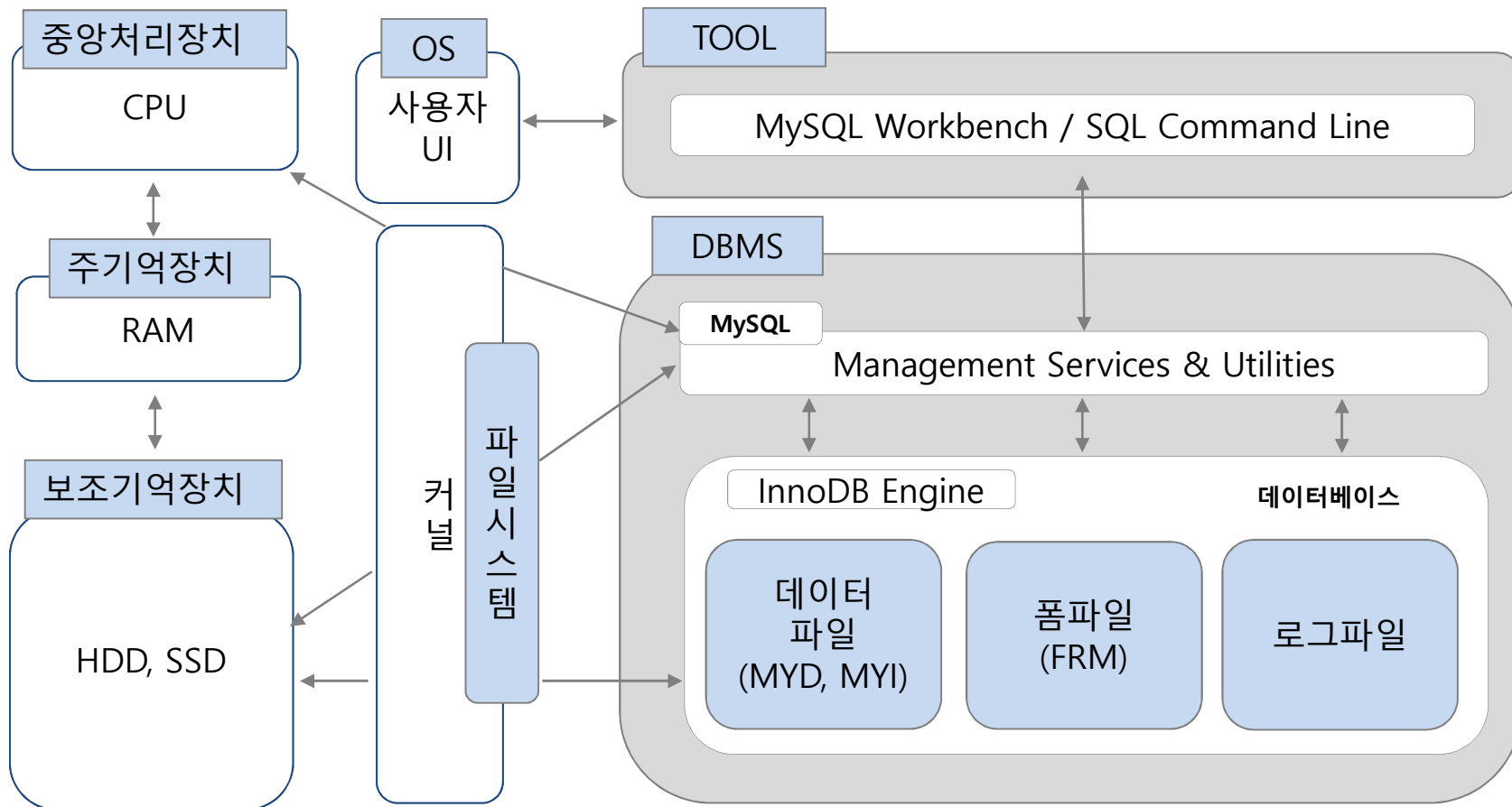
데이터베이스는 서점 데이터베이스를 이용한다.

- (1) 판매가격이 20,000원 이상인 도서의 도서번호, 도서이름, 고객이름, 출판사, 판매가격을 보여주는 highorders 뷰를 생성하시오.
- (2) 생성한 뷰를 이용하여 판매된 도서의 이름과 고객의 이름을 출력하는 SQL 문을 작성하시오.
- (3) highorders 뷰를 변경하고자 한다. 판매가격 속성을 삭제하는 명령을 수행하시오. 삭제 후 (2)번 SQL 문을 다시 수행하시오.

03. 인덱스

1. 데이터베이스의 물리적 저장
2. 인덱스와 B-tree
3. MySQL 인덱스
4. 인덱스의 생성
5. 인덱스의 재구성과 삭제

1. 데이터베이스의 물리적 저장



1. 데이터베이스의 물리적 저장

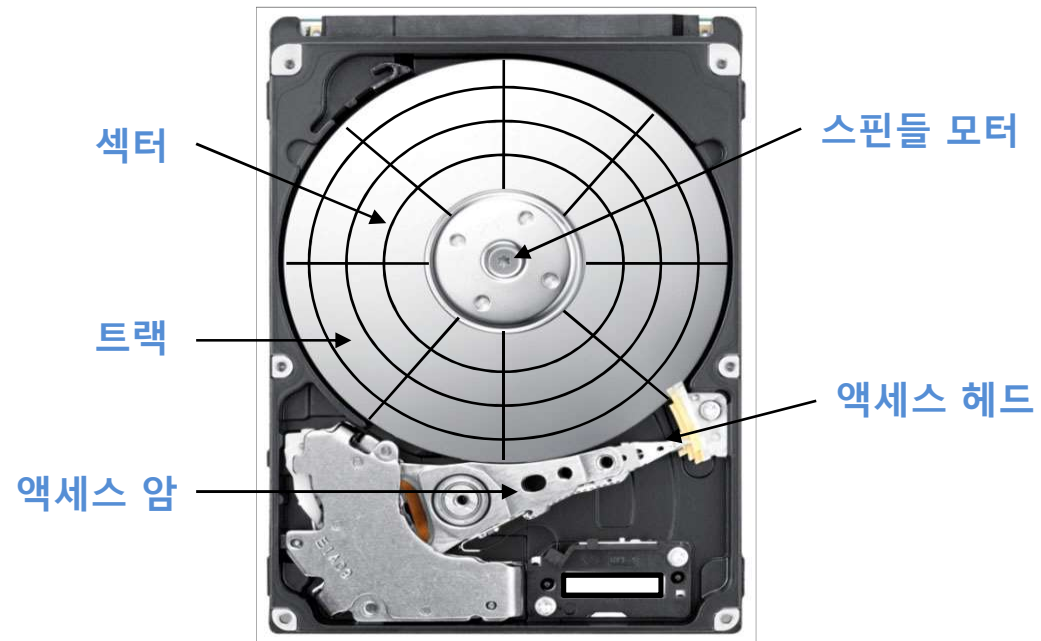
■ 실제 데이터가 저장되는 곳은 보조기억장치

- 하드디스크, SSD, USB 메모리 등

■ 가장 많이 사용되는 장치는 하드디스크

- 하드디스크는 원형의 플레이트(plate)로 구성되어 있고, 이 플레이트는 논리적으로 트랙으로 나뉘며 트랙은 다시 몇 개의 섹터로 나뉨
- 원형의 플레이트는 초당 빠른 속도로 회전하고, 회전하는 플레이트를 하드디스크의 액세스 암(access arm)과 헤더(header)가 접근하여 원하는 섹터에서 데이터를 가져옴
- 하드디스크에 저장된 데이터를 읽어 오는 데 걸리는 시간은 모터(motor)에 의해서 분당 회전하는 속도(RPM, Revolutions Per Minute), 데이터를 읽을 때 액세스 암이 이동하는 시간(latency time), 주기억장치로 읽어오는 시간(transfer time)에 영향을 받음

1. 데이터베이스의 물리적 저장



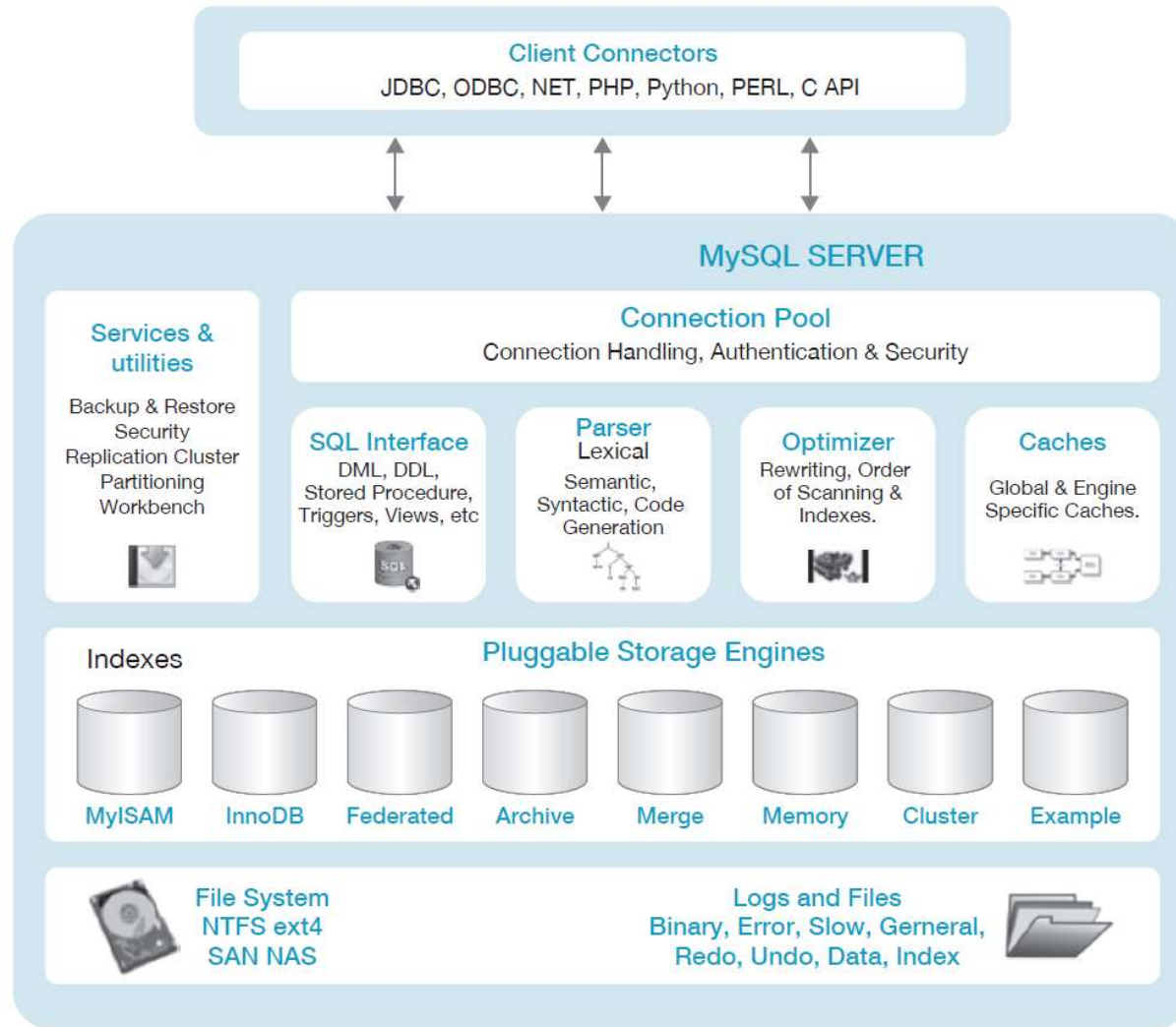
하드디스크의 구조

1. 데이터베이스의 물리적 저장

❖ 액세스 시간(access time)

액세스 시간 = 탐색시간(seek time, 액세스 헤드를 트랙에 이동시키는 시간)
+ 회전지연시간(rotational latency time, 섹터가 액세스 헤드에 접근하는 시간)
+ 데이터 전송시간(data transfer time, 데이터를 주기억장치로 읽어오는 시간)

1. 데이터베이스의 물리적 저장



MySQL의 DBMS 구조

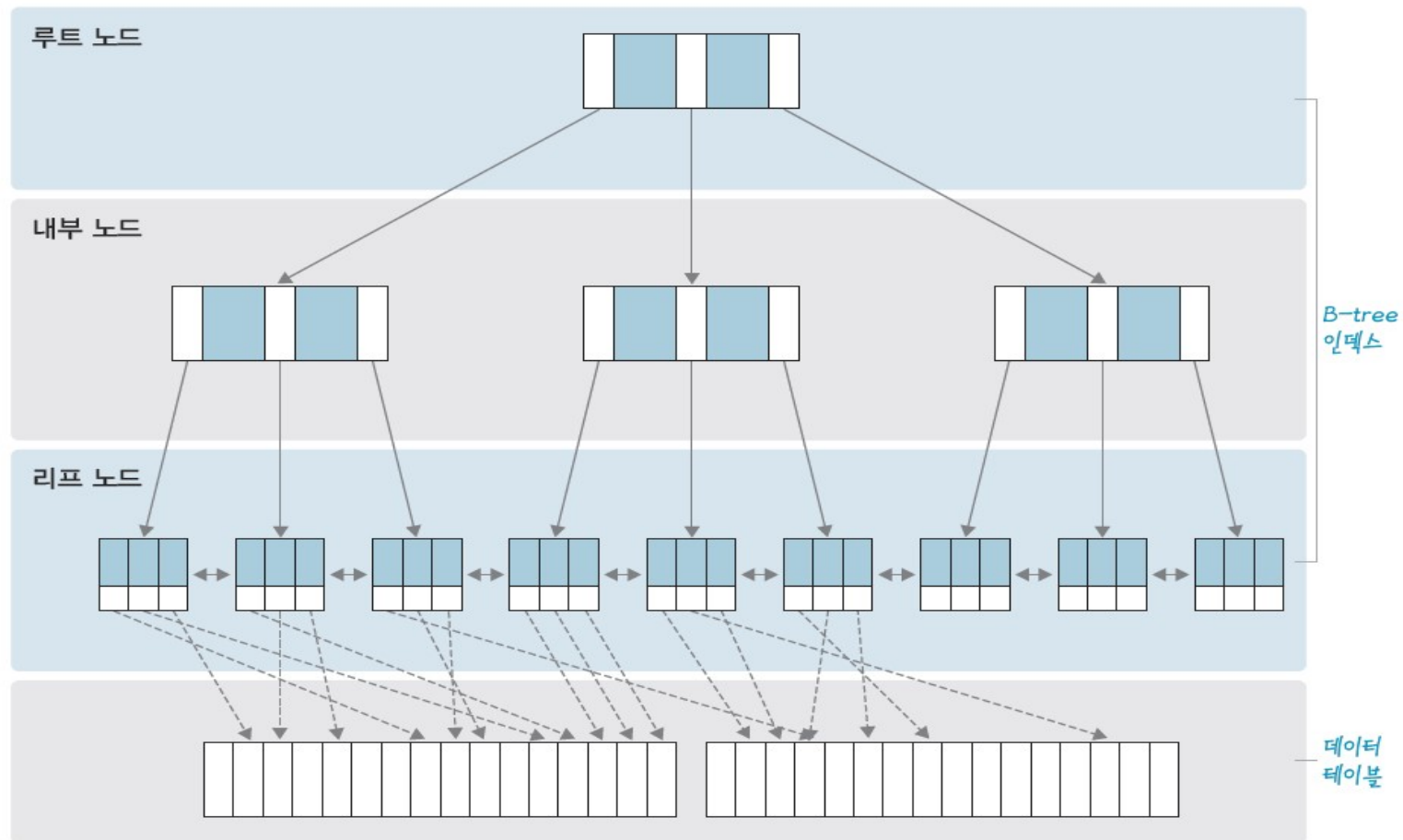
1. 데이터베이스의 물리적 저장

MySQL InnoDB 엔진 데이터베이스의 파일

| 파일 | 설명 |
|--------------------|--|
| 데이터 파일 (ibdata) | <ul style="list-style-type: none">• 사용자 데이터와 개체를 저장• 테이블과 인덱스로 구성• 확장자는 *.ibd |
| 폼파일 (frm File) | <ul style="list-style-type: none">• 테이블에 대한 각종 정보와 테이블을 구성하는 필드, 데이터 타입에 대한 정보 저장• 데이터베이스 구조 등의 변경사항이 있을 때 자동으로 업데이트됨 |

2. 인덱스와 B-tree

- 인덱스(index, 색인) : 도서의 색인이나 사전과 같이 데이터를 쉽고 빠르게 찾을 수 있도록 만든 데이터 구조



B-tree의 구조

2. 인덱스와 B-tree

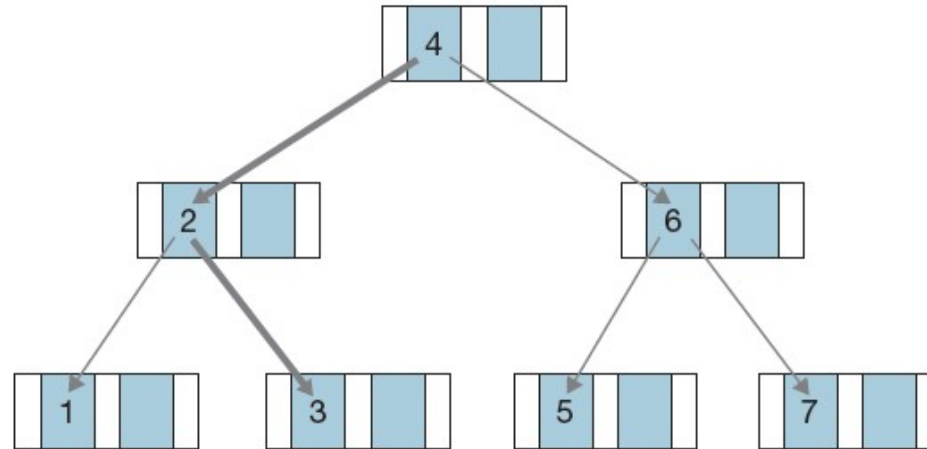


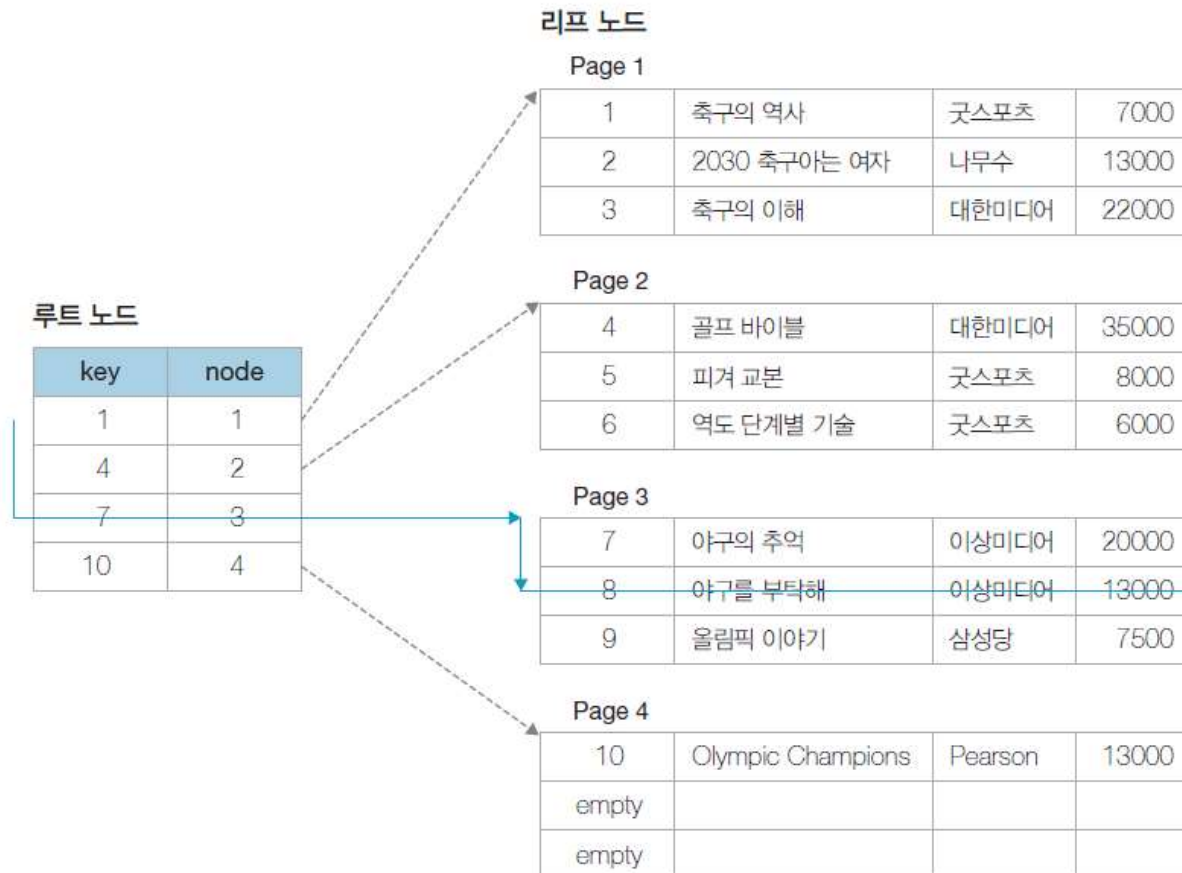
그림 4-11 B-tree에서 검색 예

■ 인덱스의 특징

- 인덱스는 테이블에서 한 개 이상의 속성을 이용하여 생성함
- 빠른 검색과 함께 효율적인 레코드 접근이 가능함
- 순서대로 정렬된 속성과 데이터의 위치만 보유하므로 테이블보다 작은 공간을 차지함
- 저장된 값들은 테이블의 부분집합이 됨
- 일반적으로 B-tree 형태의 구조를 가짐
- 데이터의 수정, 삭제 등의 변경이 발생하면 인덱스의 재구성이 필요함

3. MySQL 인덱스

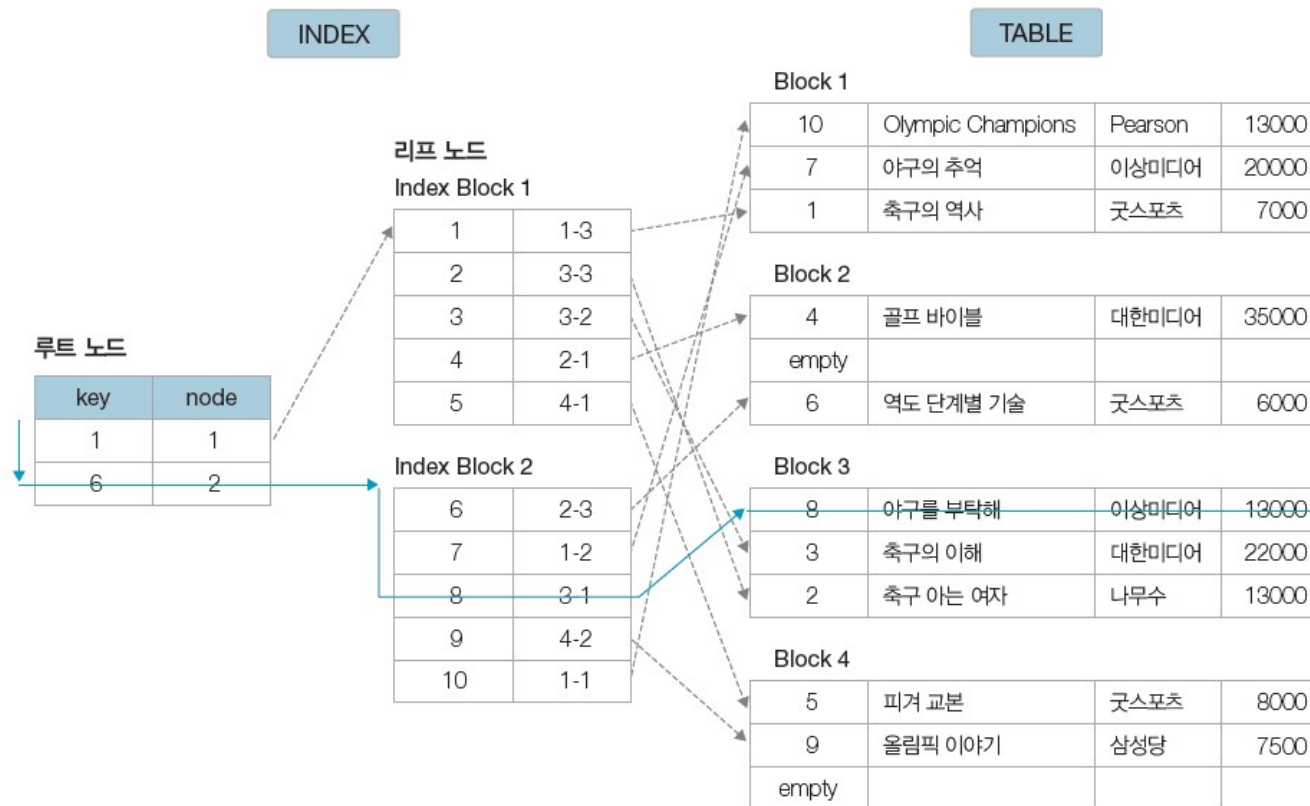
❖ 클러스터 인덱스



클러스터 인덱스 예

3. MySQL 인덱스

❖ MySQL 인덱스 B-tree



B-tree 인덱스의 예

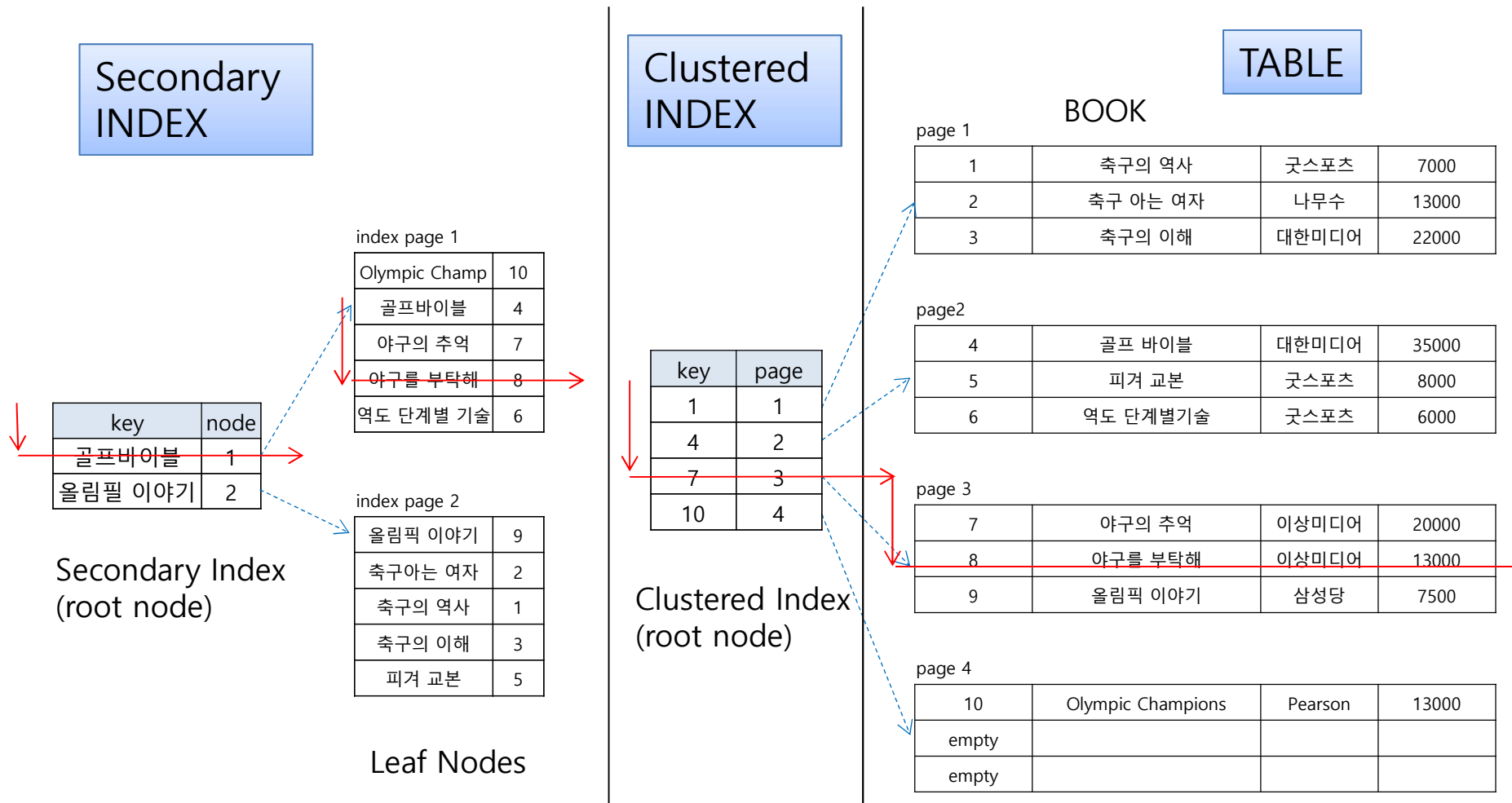
3. MySQL 인덱스

❖ MySQL 인덱스의 종류

MySQL 인덱스의 종류

| 인덱스 명칭 | 설명 / 생성 예 |
|----------|---|
| 클러스터 인덱스 | <ul style="list-style-type: none">• 기본적인 인덱스로 테이블 생성 시 기본키를 지정하면 기본키에 대하여 클러스터 인덱스를 생성한다.• 기본키를 지정하지 않으면 먼저 나오는 UNIQUE 속성에 대하여 클러스터 인덱스를 생성한다.• 기본키나 UNIQUE 속성이 없는 테이블은 MySQL 이 자체 생성한 행번호 (Row ID)를 이용하여 클러스터 인덱스를 생성한다. |
| 보조 인덱스 | <ul style="list-style-type: none">• 클러스터 인덱스가 아닌 모든 인덱스는 보조 인덱스이며 보조 인덱스의 각 레코드는 보조 인덱스 속성과 기본키 속성 값을 갖고 있다.• 보조 인덱스를 검색하여 기본키 속성 값을 찾은 다음 클러스터 인덱스로 가서 해당 레코드를 찾는다. |

3. MySQL 인덱스



[클러스터 인덱스와 보조 인덱스를 동시에 사용하는 검색]

4. 인덱스의 생성

❖ 인덱스 생성 시 고려사항

- 인덱스는 WHERE 절에 자주 사용되는 속성이어야 함
- 인덱스는 조인에 자주 사용되는 속성이어야 함
- 단일 테이블에 인덱스가 많으면 속도가 느려질 수 있음(테이블당 4~5개 정도 권장)
- 속성이 가공되는 경우 사용하지 않음
- 속성의 선택도가 낮을 때 유리함(속성의 모든 값이 다른 경우)

❖ 인덱스의 생성 문법

```
CREATE [UNIQUE] INDEX [인덱스이름]  
ON 테이블이름 (컬럼 [ASC | DESC] [{, 컬럼 [ASC | DESC]} ...])[;]
```

4. 인덱스의 생성

24 Book 테이블의 bookname 열을 대상으로 비 클러스터 인덱스 ix_Book을 생성하라.

25 Book 테이블의 publisher, price 열을 대상으로 인덱스 ix_Book2를 생성하시오.

5. 인덱스의 재구성과 삭제

- 인덱스의 재구성은 ANALYZE TABLE 명령을 사용함.
- 생성 문법

```
ANALYZE TABLE 테이블이름;
```

26 Book 테이블의 인덱스를 최적화하시오.

- 삭제 문법

```
DROP INDEX 인덱스이름
```

27 인덱스 ix_Book을 삭제하시오.

연습문제

[서점 데이터베이스 인덱스] 서점 데이터베이스에서 다음 SQL 문을 수행하고 데이터베이스가 인덱스를 사용하는 과정을 확인하세요.

(1) 다음 SQL 문을 수행해본다.

```
SELECT name, address FROM Customer WHERE name LIKE '박세리';
```

- (2) 실행 계획을 살펴본다. 실행 계획은 Workbench에서 [Query] → [Explain Current Statement]를 선택하면 표시된다.
- (3) Customer 테이블에 name으로 인덱스를 생성하시오. 생성 후 (1)번의 SQL 문을 다시 수행하고 실행 계획을 살펴보세요.
- (4) 같은 질의에 대한 두 가지 실행 계획을 비교해보세요.
- (5) (3)번에서 생성한 인덱스를 삭제하시오.