



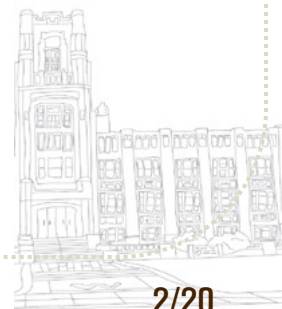
LINQ

이것이 C#이다



Contents

- ❖ 데이터! 데이터! 데이터!
- ❖ LINQ의 기본: from, where, orderby, select
- ❖ 여러 개의 데이터 원본에 질의하기
- ❖ group by로 데이터 분류하기
- ❖ LINQ의 비밀과 LINQ 표준 연산자



15.1 데이터! 데이터! 데이터!

❖ LINQ - C# 언어에 통합된 데이터 질의 기능

- 지루한 데이터 작업에서 프로그래머를 해방 시켜주는 기능

❖ 질의(질문)의 기본 요소

- From : 어떤 데이터 집합에서 찾을 것인가?
- Where : 어떤 값의 데이터를 찾을 것인가?
- Select : 어떤 항목을 추출할 것인가?

❖ LINQ 사용 전과 후

```
var profiles = from profile in arrProfile
               where profile.Height < 175
               orderby profile.Height
               select profile;
```

arrProfile 안에 있는 각 데이터로부터

Height가 175 미만인 객체만 골라

키순으로 정렬하여

profile 객체를 추출합니다.

```
foreach (var profile in profiles )
    Console.WriteLine("{0}, {1}", profile.Name, profile.Height);
```



15.2 LINQ의 기본: from

❖ 모든 LINQ 쿼리식은 반드시 from 절로 시작

❖ From 절의 지정 대상

- 형식: from <범위 변수> in <데이터 원본>
- 쿼리식의 대상이 될 데이터 원본
 - IEnumerable<T> 인터페이스를 상속 하는 형식
- 데이터 원본에 있는 각 요소 데이터를 나타내는 범위(쿼리) 변수
 - foreach 문의 반복 변수

```
int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

```
var result = from n in numbers
              where n % 2 == 0
              orderby n
              select n;
```

범위 변수 (n) 데이터 원본 (numbers)

❖ 데모 예제 - From



15.2 LINQ의 기본: where

- ❖ 필터Filter 역할을 하는 연산자, where
- ❖ From 절의 범위 변수가 가져야 하는 조건 입력
- ❖ 사용 사례

```
Profile[] arrProfile = {  
    new Profile(){Name="정우성", Height=186},  
    new Profile(){Name="김태희", Height=158},  
    new Profile(){Name="고현정", Height=172},  
    new Profile(){Name="이문세", Height=178},  
    new Profile(){Name="하동훈", Height=171}  
};
```

```
var profiles = from    profile in arrProfile  
               where   profile.Height < 175  
               select  profile;
```



15.2 LINQ의 기본: orderby

- ❖ 데이터의 정렬을 수행하는 연산자, orderby
- ❖ 정렬 기준을 매개변수로 입력
- ❖ 사용 사례

```
var profiles = from profile in arrProfile
               where profile.Height < 175
               orderby profile.Height ascending
               select profile;
```

```
var profiles = from profile in arrProfile
               where profile.Height < 175
               orderby profile.Height descending
               select profile;
```



15.2 LINQ의 기본: select

- ❖ 최종 결과를 추출하는 쿼리식의 마침표 같은 존재, **select**
- ❖ LINQ 질의 결과는 **IEnumerable<T>**로 반환
 - 형식 매개 변수 T는 바로 **select** 문에 의해 결정
- ❖ **사용 사례**

```
from    profile in arrProfile
where   profile.Height < 175
orderby profile.Height
select  new
{ Name = profile.Name, InchHeight = profile.Height * 0.393 };
```

정우성, 186
김태희, 158
고현정, 172
이문세, 178
하하, 171



김태희, 62.094
하하, 67.203
고현정, 67.596

- ❖ **데모 예제 - SimpleLinq**



15.3 여러 개의 데이터 원본에 질의하기

❖ from 문을 중첩해서 여러 데이터 원본에 접근

❖ 사용 사례

- 점수가 60점 미만인 학생이 소속되어 있는 학급과 그 학생의 점수 뽑기

```
Class[] arrClass =  
{  
    new Class(){Name="연두반", Score=new int[]{99, 80, 70, 24}},  
    new Class(){Name="분홍반", Score=new int[]{60, 45, 87, 72}},  
    new Class(){Name="파랑반", Score=new int[]{92, 30, 85, 94}},  
    new Class(){Name="노랑반", Score=new int[]{90, 88, 0, 17}}  
};
```

```
var classes = from c in arrClass  
              from s in c.Score  
              where s < 60  
              select new { c.Name, Lowest = s};
```

첫 번째 데이터 원본

두 번째 데이터 원본

❖ 데모 예제 - FromFrom



15.4 group by로 데이터 분류하기

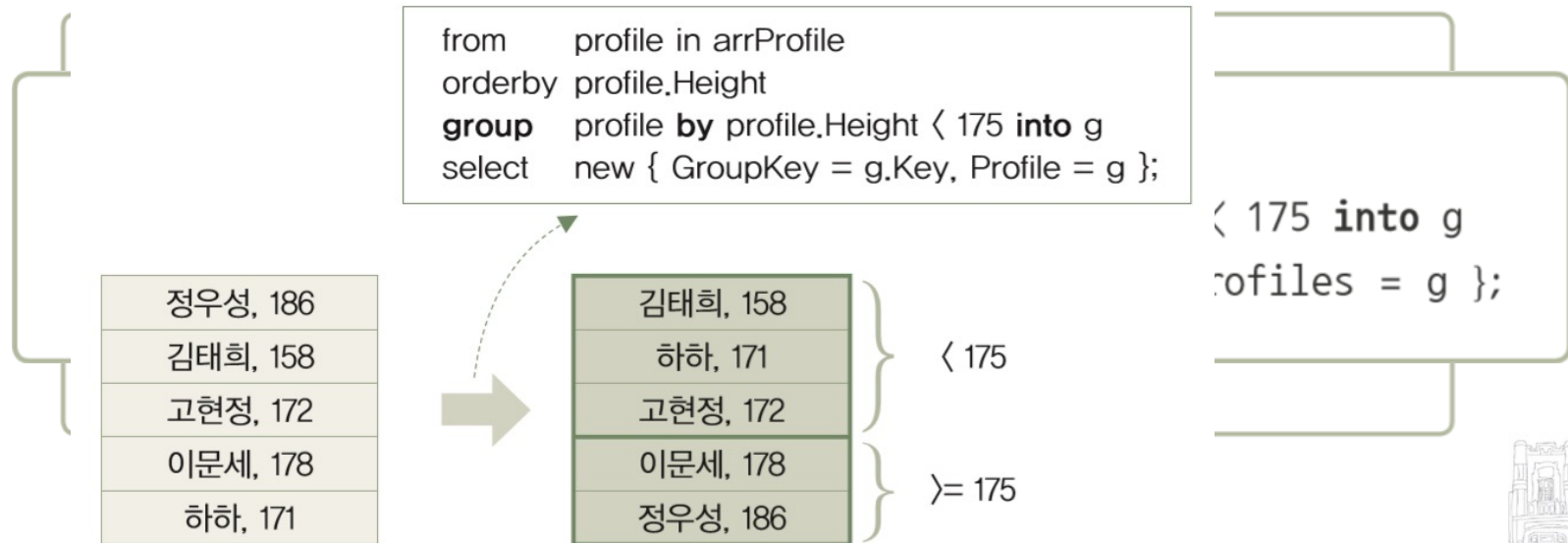
❖ 분류 기준에 따라 데이터 그룹화

❖ 사용 형식

- group A by B into C

❖ 사용 사례

- 키를 175 미만인지 175 이상인지 분류



❖ 데모 예제 – GroupBy



15.5 두 데이터 원본을 연결하는 join

❖ 각 데이터 원본 특정 필드 값을 비교해 일치하는 데이터끼리 연결

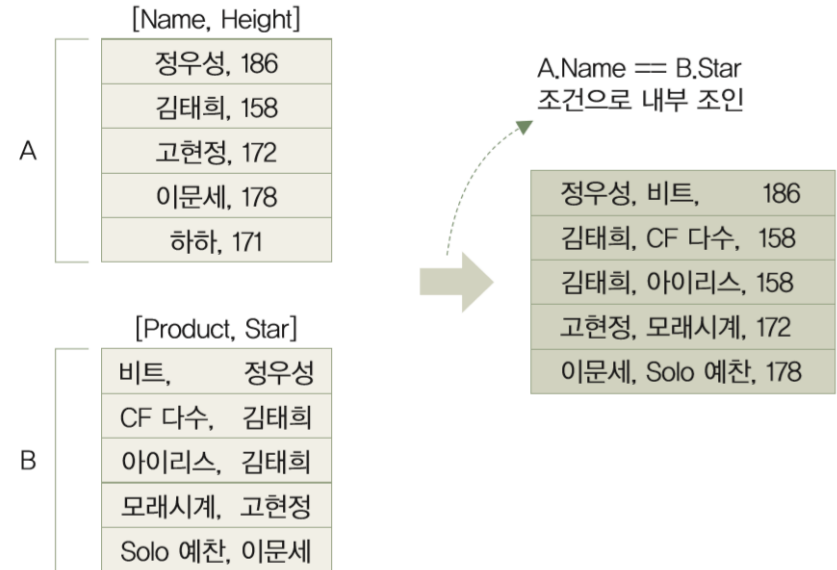
❖ 연결 방법

- 내부 조인 - 교집합
- 외부 조인 - 기준이 되는 데이터 원본 모두 포함



15.5.1 내부 조인

❖ 두 데이터 원본 사이에서 일치하는 데이터만 연결한 후 반환



❖ join 절을 통해 수행

```
var listProfile =  
    from profile in arrProfile  
    join product in arrProduct on profile.Name equals product.Star  
    select new  
    {  
        Name = profile.Name,  
        Work = product.Title,  
        Height = profile.Height  
    };
```

다음 형식의 배열

class Profile

```
{  
    public string Name { get; set; }  
    public int Height { get; set; }  
}
```

다음 형식의 배열

class Product

```
{  
    public string Title { get; set; }  
    public string Star { get; set; }  
}
```



15.5.2 외부 조인

❖ 기준 데이터 원본의 모든 데이터를 조인 결과에 반드시 포함

- 연결할 데이터 원본에 기준 데이터 원본과 일치하는 데이터가 없다면 빈 값으로 채움

A

[Name, Height]	
정우성, 186	
김태희, 158	
고현정, 172	
이문세, 178	
하하, 171	

B

[Product, Star]	
비트, 정우성	
CF 다수, 김태희	
아이리스, 김태희	
모래시계, 고현정	
Solo 예찬, 이문세	

A.Name == B.Star
조건으로 외부 조인

정우성, 비트,	186
김태희, CF 다수,	158
김태희, 아이리스,	158
고현정, 모래시계,	172
이문세, Solo 예찬,	178
하하, ,	171

❖ 사용 방법

다음 형식의 배열

```
class Profile
{
    public string Name { get; set; }
    public int Height { get; set; }
}
```

다음 형식의 배열

```
class Product
{
    public string Title { get; set; }
    public string Star { get; set; }
}
```

```
var listProfile =
    from profile in arrProfile
    join product in arrProduct on profile.Name equals product.Star into ps
    from product in ps.DefaultIfEmpty(new Product(){Title="그런거 없음"})
    select new
```

❖ 데모 예제 - Join



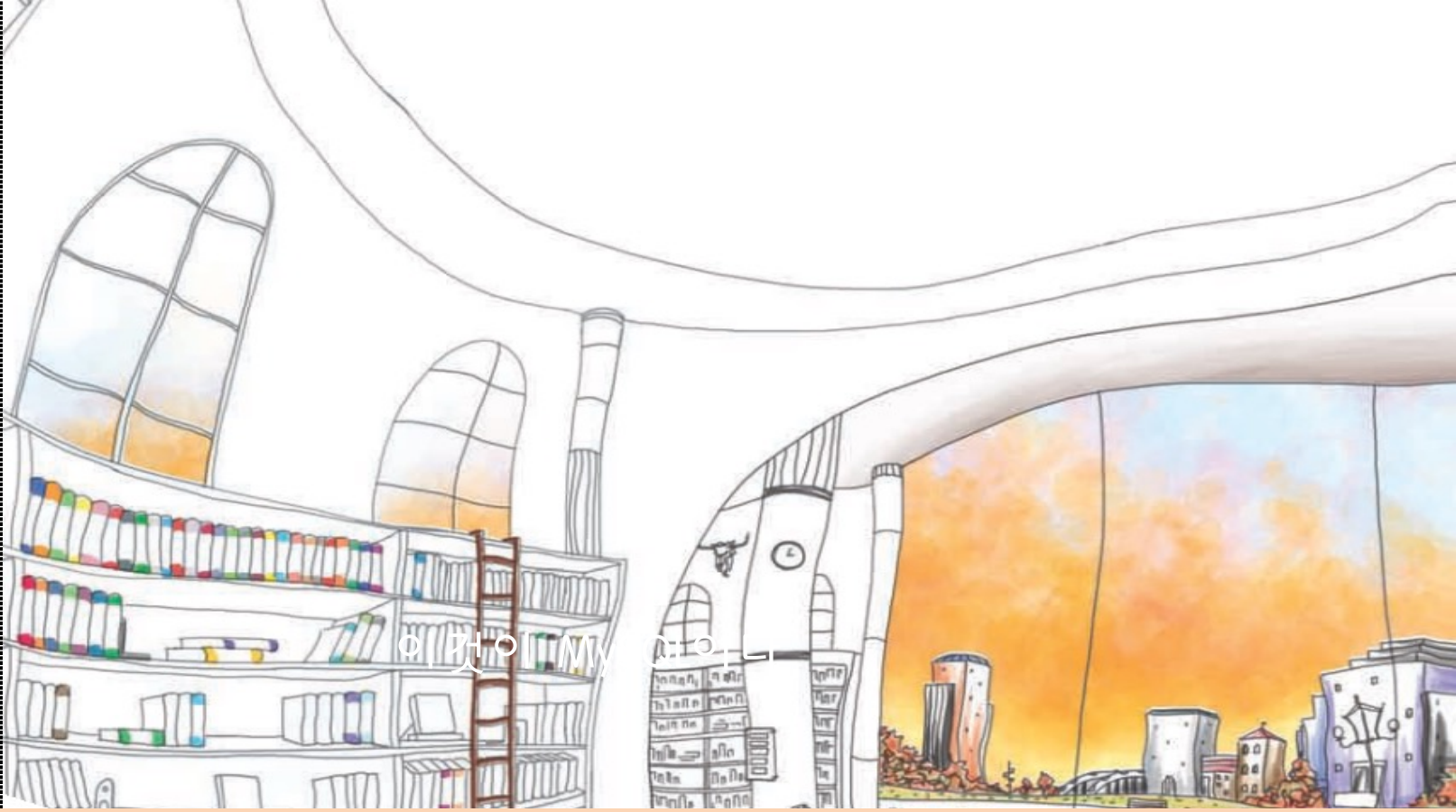
15.6 LINQ의 비밀과 LINQ 표준 연산자

❖ .NET 언어 중 C#과 VB에서만 사용 가능

- LINQ 쿼리식을 CLR이 이해하는 코드로 번역 하도록 컴파일러 업그레이드
- C# 지원 LINQ 쿼리 식(11) < LINQ 표준 연산자 수 (53)
- LINQ 쿼리식과 위 메소드를 함께 사용하는 방법 필요(p505 표 참조)
- → System.Linq 네임스페이스 필수

❖ 데모 예제 - MinMaxAvg





Thank You !

이것이 C#이다

