

29강. 비연결형 데이터베이스 (2)

MS-SQL, SqlDataAdapter, DataSet

목차

1. DataSet
2. SqlDataAdapter
3. DataSet.Fill() 와 DataSet.Update()
4. SqlCommand 와 SqlCommandBuilder
5. 관계형데이터베이스 처리

학습 과정

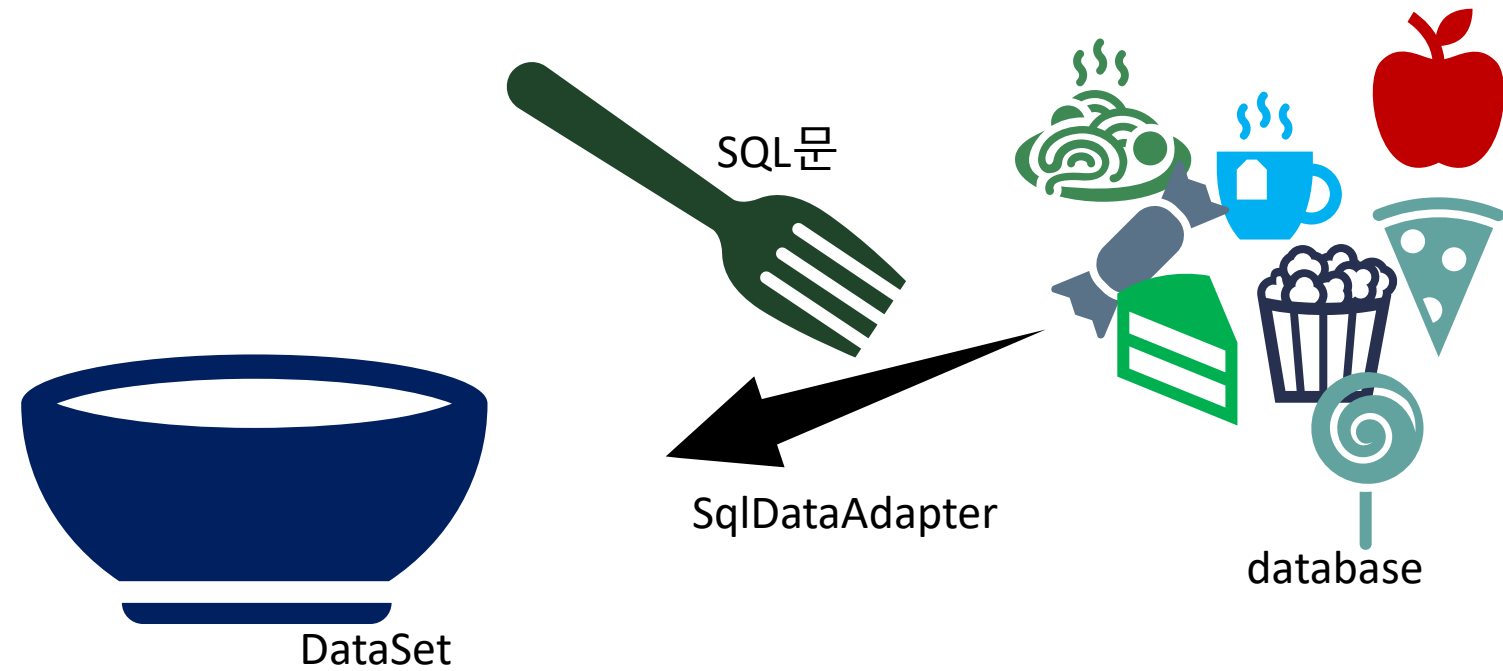
DataTable



DataSet



SqlDataAdapter



DataSet

DataSet

역할

- 다수의 테이블 관리 ➔ DataSet.DataTables 컬렉션

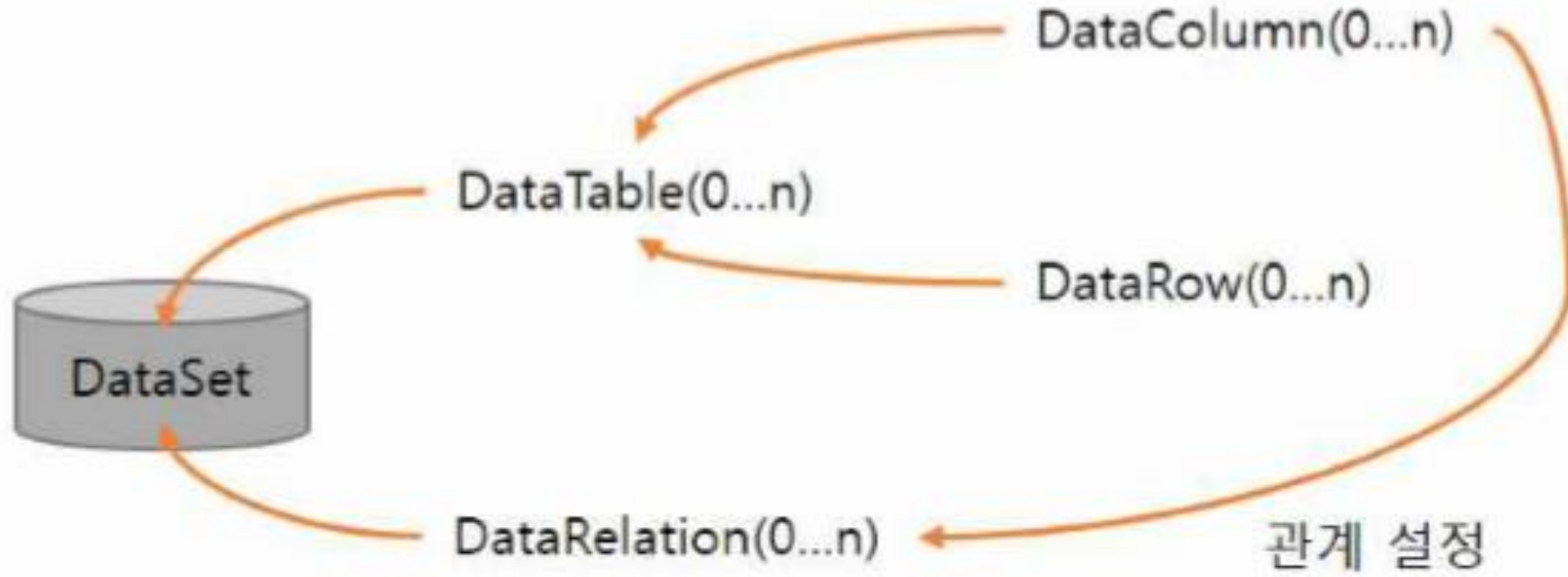
특징

- 테이블명 대소문자 구별
- XML 형식으로 데이터 읽기/출력 가능
- 데이터 관계 설정 가능(DataRelation)

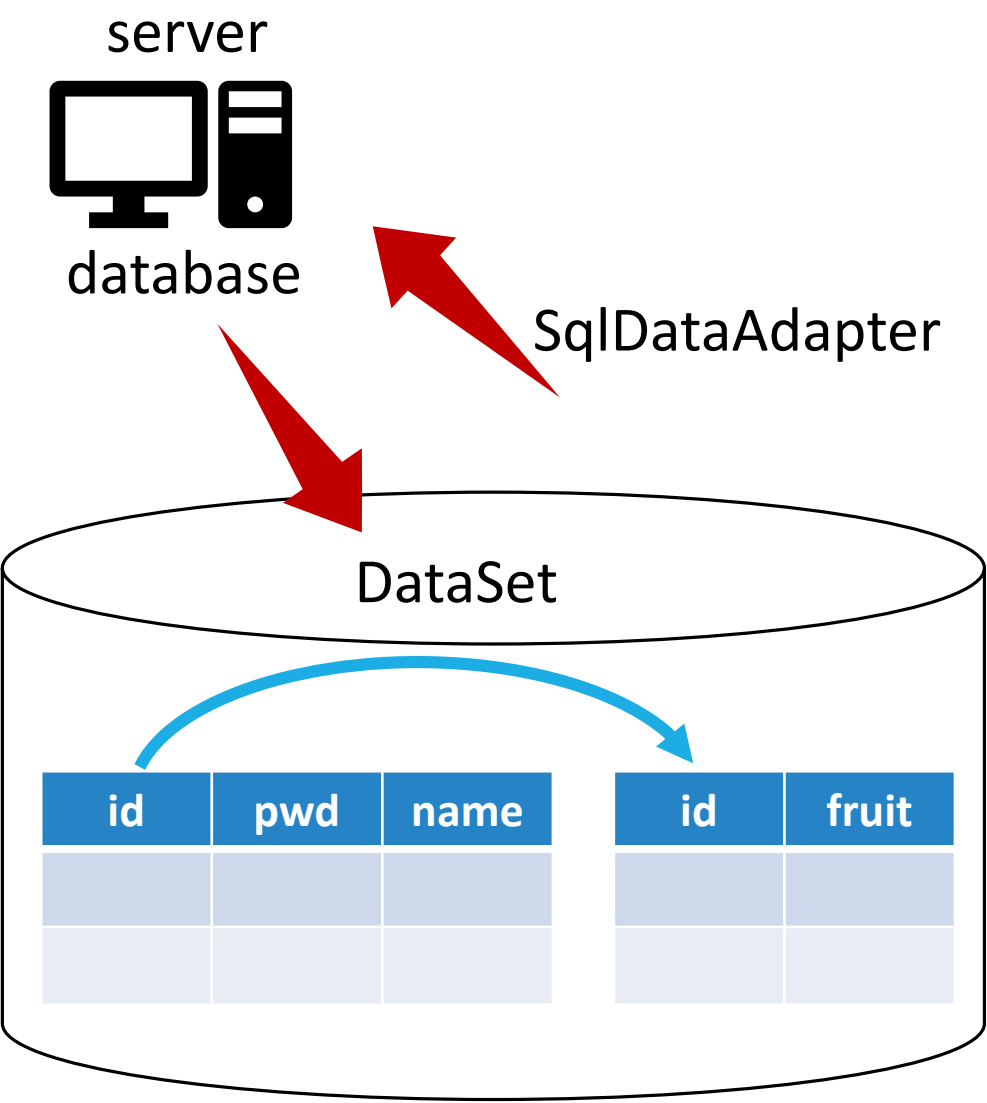
DataSet 테이블 관계 설정 방법

- 코드
- 마법사(Wizard) ➔ “결국 코드이다”

DataSet 구조



DataSet 구조



DataSet

네임스페이스

- System.Data

주요 속성

- DataSetName : 데이터셋 이름 설정
- Tables
 - public **DataTableCollection** Tables { get; }

DataSet

DataSet 생성

- `public DataSet(string dataSetName)`

Table 추가

- `public DataTableCollection Tables { get; }`
- `DataTableCollection.Add()`
: `public void Add(DataTable table)`

DataSet

■ DataSet에 Table, 컬럼 설정

id	pwd	name	
string, max :10	string, max:8	string, max:15	

```
DataSet1 = new DataSet("Info");
DataSet1.Tables.Add(new DataTable("Table1"));
DataSet1.Tables.Add(new DataTable("Table2"));

// 테이블1
DataColumn Col;
Col = new DataColumn("id", typeof(string));
Col.MaxLength = 10;
Col.Unique = true;
DataSet1.Tables[0].Columns.Add(Col);
DataSet1.Tables[0].PrimaryKey = new DataColumn[] { Col };

Col = new DataColumn("pwd", typeof(string));
Col.MaxLength = 8;
DataSet1.Tables[0].Columns.Add(Col);
Col = new DataColumn("name", typeof(string));
Col.MaxLength = 15;
DataSet1.Tables[0].Columns.Add(Col);
```

관계형 데이터베이스

❖ 개요

테이블 사이에 중복된 데이터를 줄이기 위해
관계를 설정하여 테이블을 다루는 데이터베이스

❖ 기본키(Primary key)와 외래 키(Foreign Key)

기본키 : 중복되지 않는 테이블의 대표 컬럼

외래키 : 기본키와 관계를 맺을 수 있는 컬럼

	id	pwd	name
▶	12345	12345	대한
	22222	12345	민국
*			

Parent Table

	id	fruit_name
▶	22222	수박
	12345	딸기
	12345	메론
*		

Child Table

DataSet

❖ 기본키 설정

Table.PrimaryKey 속성에 설정

```
DataColumn Col;  
Col = new DataColumn("id", typeof(string));  
Col.MaxLength = 10;  
Col.Unique = true;  
DataSet1.Tables[0].Columns.Add(Col);  
DataSet1.Tables[0].PrimaryKey = new DataColumn[] { Col };
```

DataSet

- ❖ 외래 키 설정
ForeignKeyConstraint 클래스 사용

```
public ForeignKeyConstraint( string constraintName,  
                             DataColumn parentColumn,  
                             DataColumn childColumn )
```

- ❖ 외래 키 설정소스
생성 후에 DataSet.Table.Constraints.Add()에 컬럼
설정

```
ForeignKeyConstraint FK  
    = new ForeignKeyConstraint("FK",  
                               Parent Column, Child Column);  
DataSet.Tables[1].Constraints.Add(FK);
```

// 테이블2

```
Col = new DataColumn("id", typeof(string));
```

```
Col.MaxLength = 10;
```

```
DataSet1.Tables[1].Columns.Add(Col);
```

```
ForeignKeyConstraint FK = new ForeignKeyConstraint("FK", DataSet1.Tables[0].Columns["id"], Col);
```

```
DataSet1.Tables[1].Constraints.Add(FK);
```

```
Col = new DataColumn("fruit_name", typeof(string));
```

```
Col.MaxLength = 20;
```

```
DataSet1.Tables[1].Columns.Add(Col);
```

DataSet

❖ 관계형 데이터베이스 실습 Table 구조

○ parent table

id	pwd	name	
string, max :10	string, max:8	string, max:15	

○ child table

id	fruit
string, max :10	string, max:20

Form1

	id	pwd	name
▶	12345	12345	대한
	22222	12345	민국
*			

과일보기

	id	fruit_name
▶	22222	수박
	12345	딸기
	12345	메론
*		

나미보기

DataSet

관계형 데이터베이스 실습 단계

- 1단계

DataSet, DataTable, DataColumn, Primary Key,
Foreign Key 설정

[소스 보기](#)

- 2단계

버튼에 따른 기능 코드 작성

[소스 보기](#)

MS SQL 키 설정

Primary Key와 Foreign Key 만들기

MSI01-PC#SQLEXPRESS (SQL Server 12.0.5203 - sa)

- 데이터베이스
 - 시스템 데이터베이스
 - school
 - 데이터베이스 다이어그램
 - 테이블
 - 시스템 테이블
 - FileTables
 - dbo.tblFruit
 - 열
 - 키
 - 제약 조건
 - 트리거
 - 인덱스
 - 통계
 - dbo.tblInfo
 - 열
 - 키
 - 제약 조건

테이블(T)...
디자인(G)

MSI01-PC#SQLEXPRESS - dbo.tblFruit

열 이름	데이터 형식
id	nchar(10)
이름	nchar(10)

기본 키 설정(Y)
열 삽입(M)
열 삭제(N)
관계(H)...
인덱스/키(I)...
전체 텍스트 인덱스(F)...
XML 인덱스(X)...
CHECK 제약 조건(O)...
공간 인덱스(P)...
변경 스크립트 생성(S)...
속성(R) Alt+Enter

	필드 이름	데이터 형식	Null 허용
	id	nchar(10)	<input type="checkbox"/>
	pwd	nchar(8)	<input checked="" type="checkbox"/>
	name	nchar(15)	<input checked="" type="checkbox"/>
	age	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Foreign Key 설정

MSI01-PC\SQLEXPRESS...ol - dbo.tblFruit* X MSI01-PC\

열 이름	데이터 형식
id	

외래 키 관계

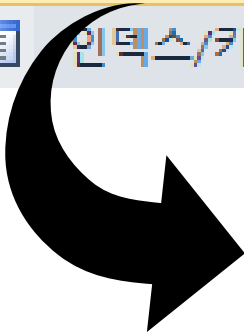
선택한 관계(S):

[추가] 단추를 사용하여 새 관계를(를) 만드십시오.

추가(A)

삭제(D)

닫기(C)



외래 키 관계



선택한 관계(S):

FK_tblFruit_tblFruit*

새 관계의 속성을 편집하고 있습니다. 새 관계들(를) 사용하려면 우선 '테이블 및 열 사양' 속성을 채워야 합니다.

▼ (일반)

만들거나 다시 설정할 때 기존 예

> 테이블 및 열 사양



▼ ID

(이름)

FK_tblFruit_tblFruit

외래 키 관

테이블 및 열

선택한 관

FK_tblF

관계 이름(N):

FK_tblFruit_tblFruit

기본 키 테이블(P):

tblFruit

tblFruit

tblInfo

tblStudent

Table 1

외래 키 테이블:

tblFruit

테이블 및 열

?

×

관계 이름(N):

FK_tblFruit_tblInfo

기본 키 테이블(P):

tblInfo

외래 키 테이블:

tblFruit

id

id

확인

취소

SqlDataAdapter

MS-SQL 연결 및 데이터 다루기

Part 1. 연결하여 DataSet에 저장하기

SqlDataAdapter

❖ 역할 1

- ① 데이터베이스 연결
- ② DataSet를 채우기
- ③ SQL Server 데이터베이스 업데이트

※ SqlDataAdapter에 SqlConnection 연결시
Open(), Close() 불필요

SqlDataAdapter

❖ 연결과 생성

◦ 연결

◦ SqlConnection 사용

```
SqlConnection Conn = new SqlConnection();
```

```
Conn.ConnectionString =
```

```
“server=.\SQLEXPRESS:database=school;uid=sa;pwd=12345”;
```

◦ SqlDataAdapter 생성

```
SqlDataAdapter dataAdapter = new SqlDataAdapter(  
    “SELECT * FROM tblInfo”, Conn);
```

```
public SqlDataAdapter(  
    string selectCommandText,  
    SqlConnection selectConnection  
)
```

SqlDataAdapter

❖ DataSet 채우기

- Fill() 메소드

- 저장한 레코드의 수를 리턴, 생성되지 않으면 0을 리턴

```
public int Fill(  
    DataSet dataSet,  
    string srcTable  
)  
  
public override int Fill(  
    DataSet dataSet
```

```
public int Fill(  
    DataTable dataTable  
)
```

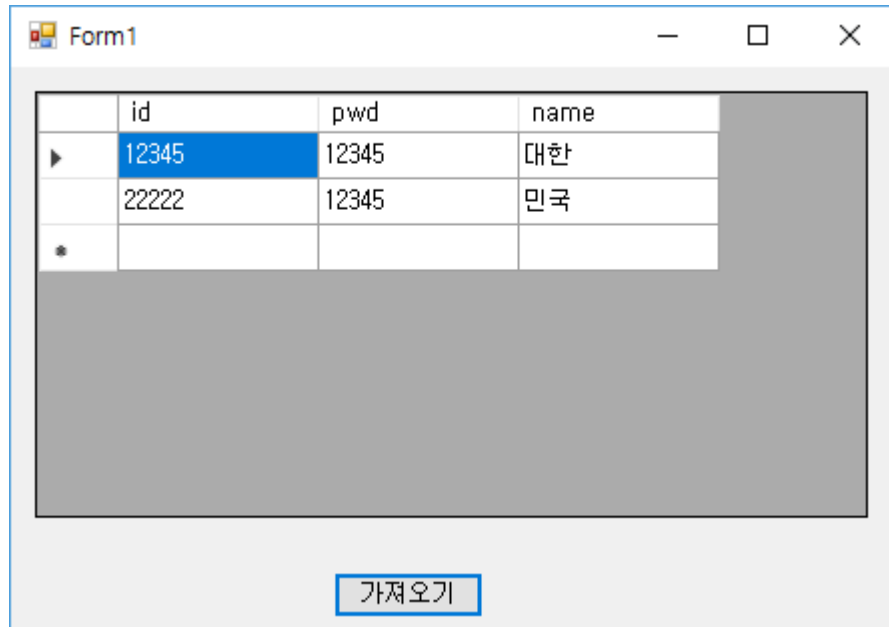
SqlDataAdapter

❖ DataSet 채우기

- SqlDataAdapter.Fill() 메소드 역할
 - SelectCommand의 SQL문 실행
 - 데이터베이스 서버의 레코드를 DataSet에 채우는 역할
(읽기 또는 조회)

SqlDataAdapter

실습1. 데이터 베이스의 테이블의 내용을 읽어
데이터그리드뷰에 출력하기



데이터베이스 : school, 테이블: tblInfo

소스보기

SqlDataAdapter 속성 다루기

SQL문 설정과 메서드

SqlDataAdapter에 SQL문 설정과 실행

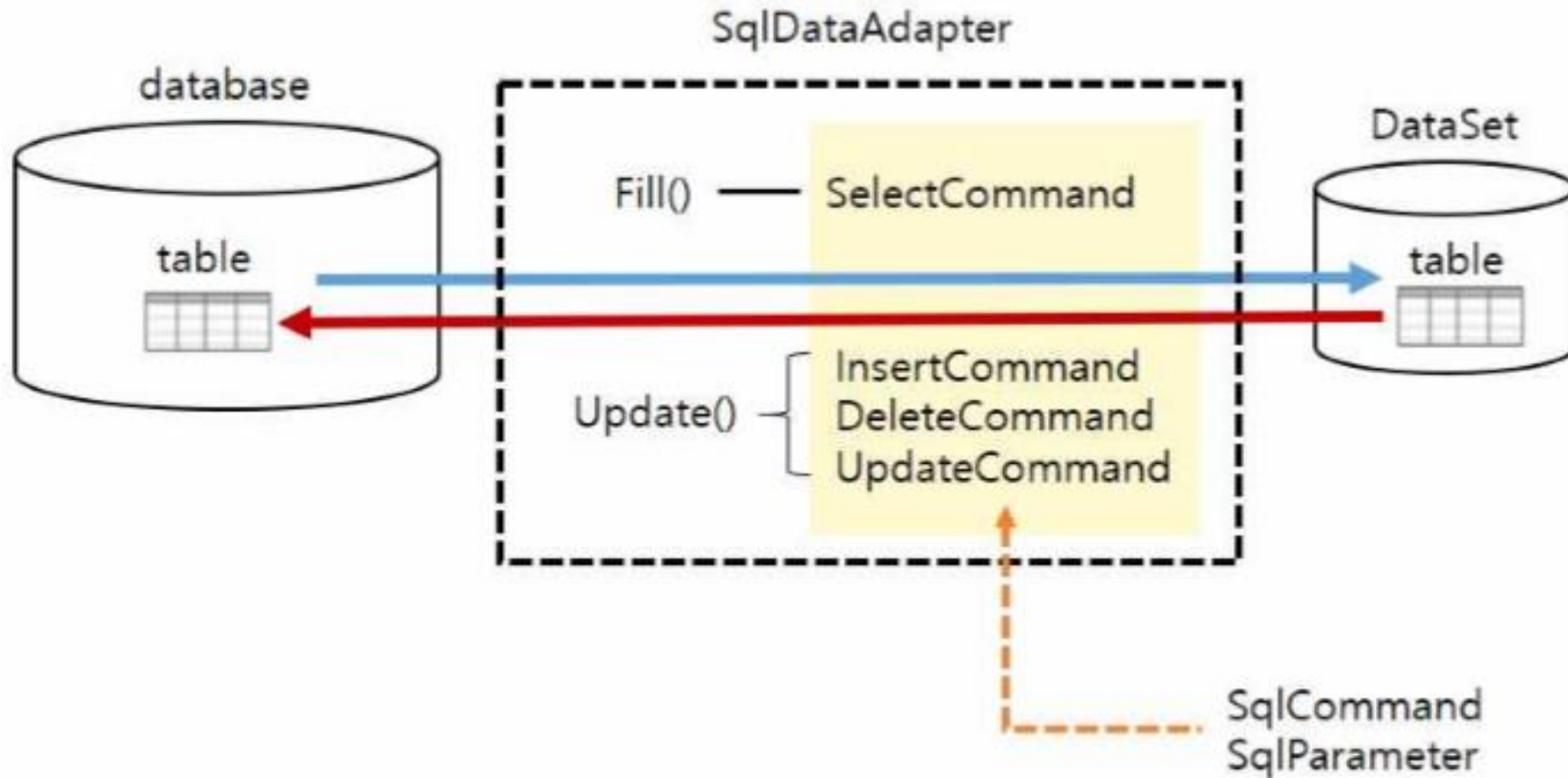
❖ SQL 설정

- SelectCommand(조회), InsertCommand(삽입), DeleteCommand(삭제), UpdateCommand(갱신) 속성에 SQL문 할당

❖ SQL 실행

- Fill() 로 DataSet 채우기(database → DataSet)
 - SelectCommand 실행
- Update()로 SQL DataBase 업데이트
(DataSet → database)
 - InsertCommand, DeleteCommand, UpdateCommand 실행

SqlDataAdapter 사용 구조도



SqlDataAdapter

속성으로 SQL 설정

- SelectCommand : 선택된 데이터베이스 테이블의 레코드를 DataSet에 **저장**
- DeleteCommand : DataSet 테이블의 레코드와 일치하도록 데이터베이스 테이블의 레코드 **삭제**
- InsertCommand : DataSet 테이블의 레코드와 일치하도록 데이터베이스 테이블의 레코드 **입력**
- UpdateCommand : DataSet 테이블의 수정된 내용과 일치하도록 데이터베이스 테이블의 레코드 **수정**



SqlCommand : SQL문 설정

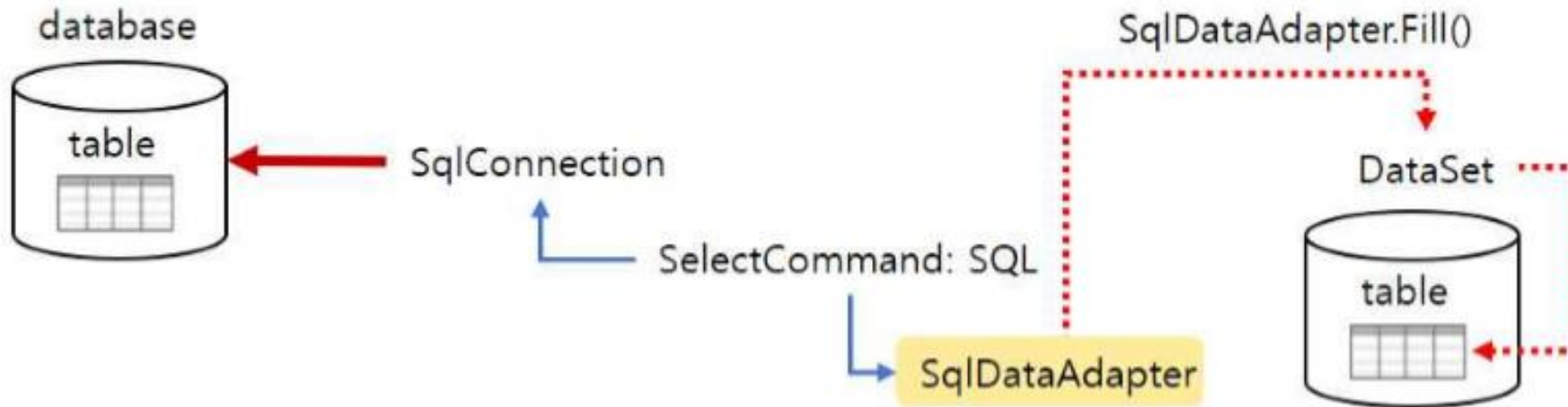


SqlDataAdapter에 적용

SelectCommand

SelectCommand

SelectCommand SQL 설정 구조



SelectCommand

속성

- `public SqlCommand SelectCommand { get; set; }`

실습 1

- SelectCommand를 사용하여 tblInfo에 있는 레코드 출력하기 ← 콘솔형태

[소스보기](#)

동적 SQL 문을 위한 Paramater

SqlCommand.Parameter

- SQL문 안에 사용할 매개변수를 정의

SqlCommand.Parameters.Add()

```
public SqlParameter Add(  
    string parameterName,  
    SqlDbType sqlDbType,  
    int size,  
    string sourceColumn  
)
```

SqlDbType.NVarChar

SqlDbType.Int

동적 SQL 문을 위한 Paramater

❖ SQL 문에 매개변수 사용

SqlDataAdapter.SelectCommand.CommandText = "SQL문"

Or

SqlDataAdapter.SelectCommand = new SqlCommand(
"SQL문", SqlConnection);

❖ SQL문 구성

"SELECT * FROM table명 where name = @name"

❖ @name 에 대한 정의

SqlDataAdapter.SelectCommand.Parameters.Add(
"@name", SqlDbType.NVarChar, 20, "name")

동적 SQL 문을 위한 Paramater

❖ SQL Parameter 매개변수에 값 할당

- SqlDataAdapter.SelectCommand.Parameters["@name"].Value =
"홍길동"
- SqlDataAdapter.SelectCommand.Parameters.AddWithValue(
"@name", "홍길동");
➔ 정의와 할당을 동시에

동적 SQL 문을 위한 Paramater

❖ 동적 SQL문 실습

database: example table : tblEx01

id	name	age
int, primarykey	string, max:15	int

Id	name	age
1	이태성	20
7	이태성	30
8	이태성	47

```
"SELECT * FROM tblEx01 WHERE name = @name"
```

소스보기

InsertCommand

입 력

InsertCommand

속성

- `public SqlCommand InsertCommand { get; set; }`

SQL 문

- `INSERT INTO table명 (컬럼1, 컬럼2 ..) VALUES (값1, 값2)`

SQL Insert 실행 방법

- 동적 SQL문을 활용한 방법
- DataSet 과 Update() 호출

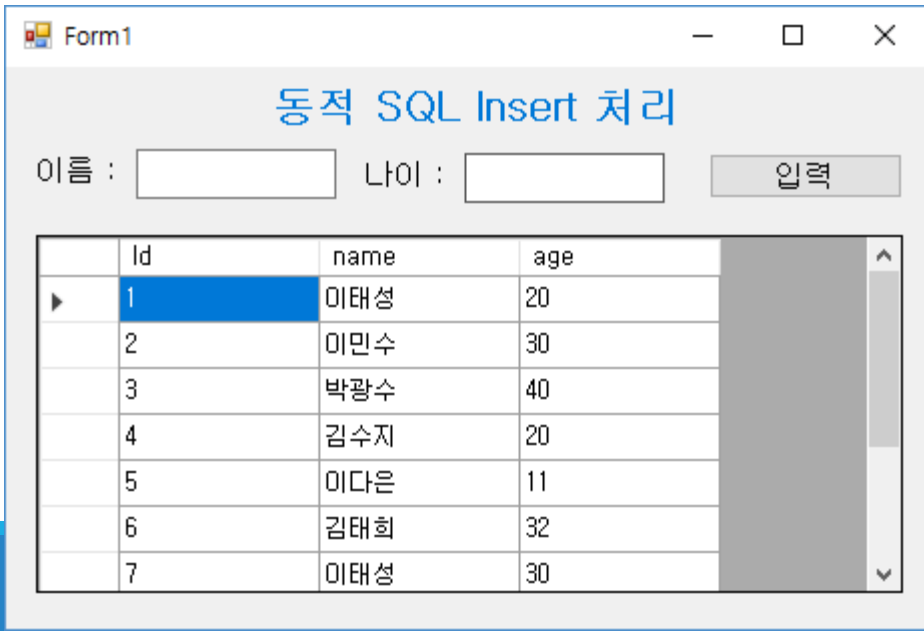
SQL Insert 실행 - 동적 SQL문 활용

동적 SQL문을 활용한 방법

- `SqlDataAdapter.InsertCommand.Parameters.Add()`
- `SqlDataAdapter.InsertCommand.Parameters["@매개변수"].Value = "값";`

동적 SQL문 실습

- database Example table : tblEx01



	Id	name	age
▶	1	이태성	20
	2	이민수	30
	3	박광수	40
	4	김수지	20
	5	이다은	11
	6	김태희	32
	7	이태성	30

소스보기

SQL Insert 실행 – Update() 활용

Update() 를 실행하기 위한 과정

1. SQL 문 정의

① `SqlDataAdapter.InsertCommand.Parameters.Add("@변수명",);`

2. DataSet 다루기

① `Fill()` : 채우기

② `DataSet.Rows.NewRow()` 로 레코드 생성

③ `DataSet.Tables["table명"].Rows.Add()` 레코드 입력

④ `SqlDataAdapter.Update()` 호출

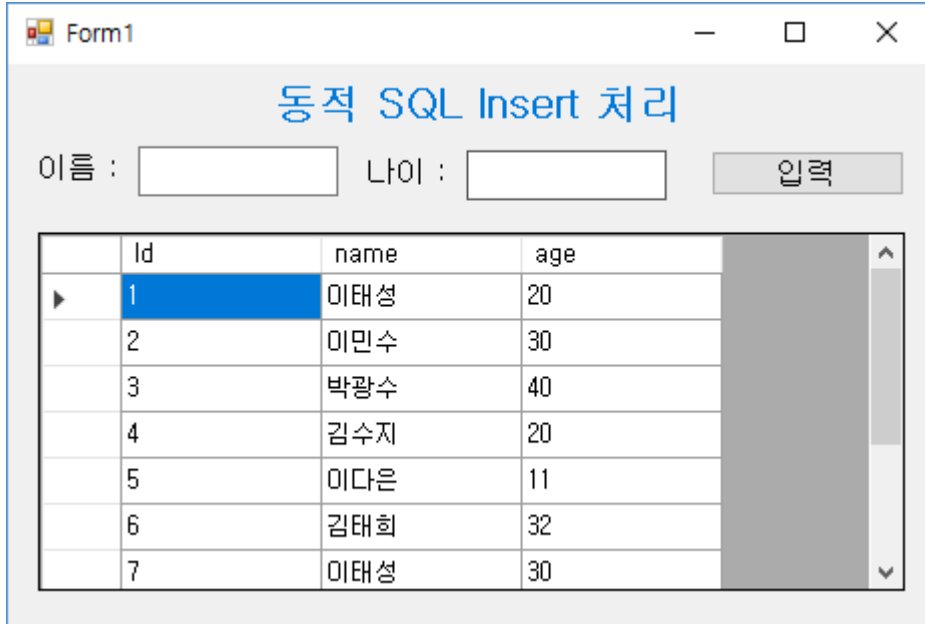
“DataSet의 변화를 SQL문을 통해 실행하므로
직접적인 데이터 입력할 필요가 없다”

“@매개변수와 외부 입력을 직접할 필요가 없다”

SQL Insert 실행 – Update() 실습

Update()를 사용하여 데이터를 입력

- database Example table: tblEx01



동적 SQL Insert 처리

이름 : 나이 :

	Id	name	age
▶	1	이태성	20
	2	이민수	30
	3	박광수	40
	4	김수지	20
	5	이다운	11
	6	김태희	32
	7	이태성	30

소스보기

DeleteCommand

삭제

DeleteCommand

속성

- `public SqlCommand DeleteCommand { get; set; }`

삭제 SQL문

- `DELETE table명 WHERE 조건`

삭제 단계

- ① 기본 키(primary key) 검색 ➔ where
- ② 삭제 SQL 실행
- ③ 결과 출력

DeleteCommand

SQL Delete 실행 방법

- 동적 SQL 문을 사용
- DataSet과 Update() 사용

DeleteCommand – 동적 SQL문 사용

동적 SQL문 작성

- “DELETE FROM table명 WHERE id = @id”

매개변수 정의

- SqlDataAdapter.DeleteCommand.Parameters.Add()

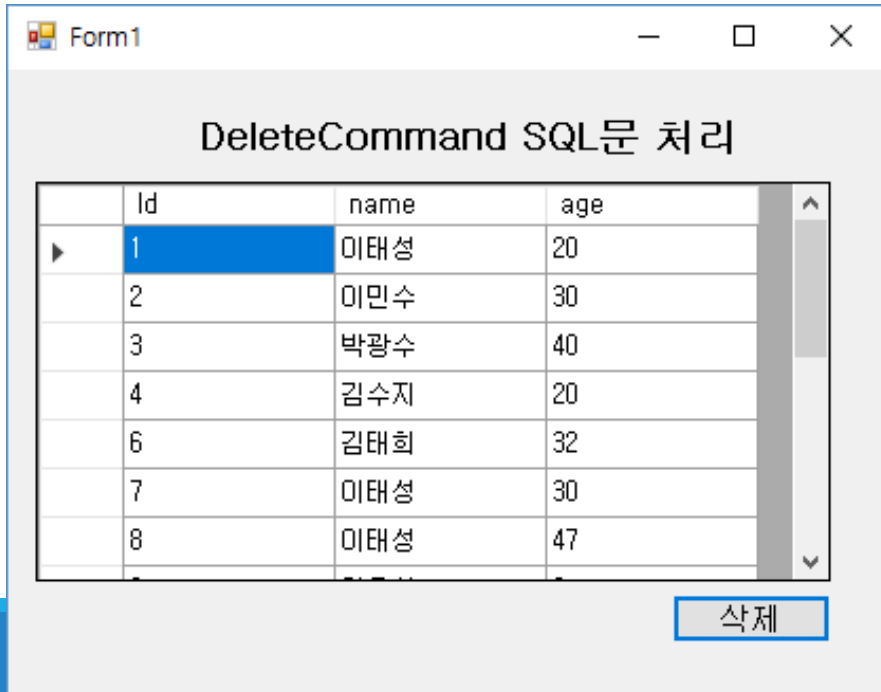
SQL문 실행

- SqlDataAdapter.DeleteCommand.ExecuteNonQuery()

DeleteCommand – 동적 SQL문 사용

동적 SQL문 실습

- database : example table : tblEx01
- 선택한 행 삭제
- DataGridView 선택한 셀 id 값 가져오기
 - (int)dataGridView1.CurrentRow.Cells["id"].Value;



Form1

DeleteCommand SQL문 처리

	Id	name	age
▶	1	이태성	20
	2	이민수	30
	3	박광수	40
	4	김수지	20
	6	김태희	32
	7	이태성	30
	8	이태성	47

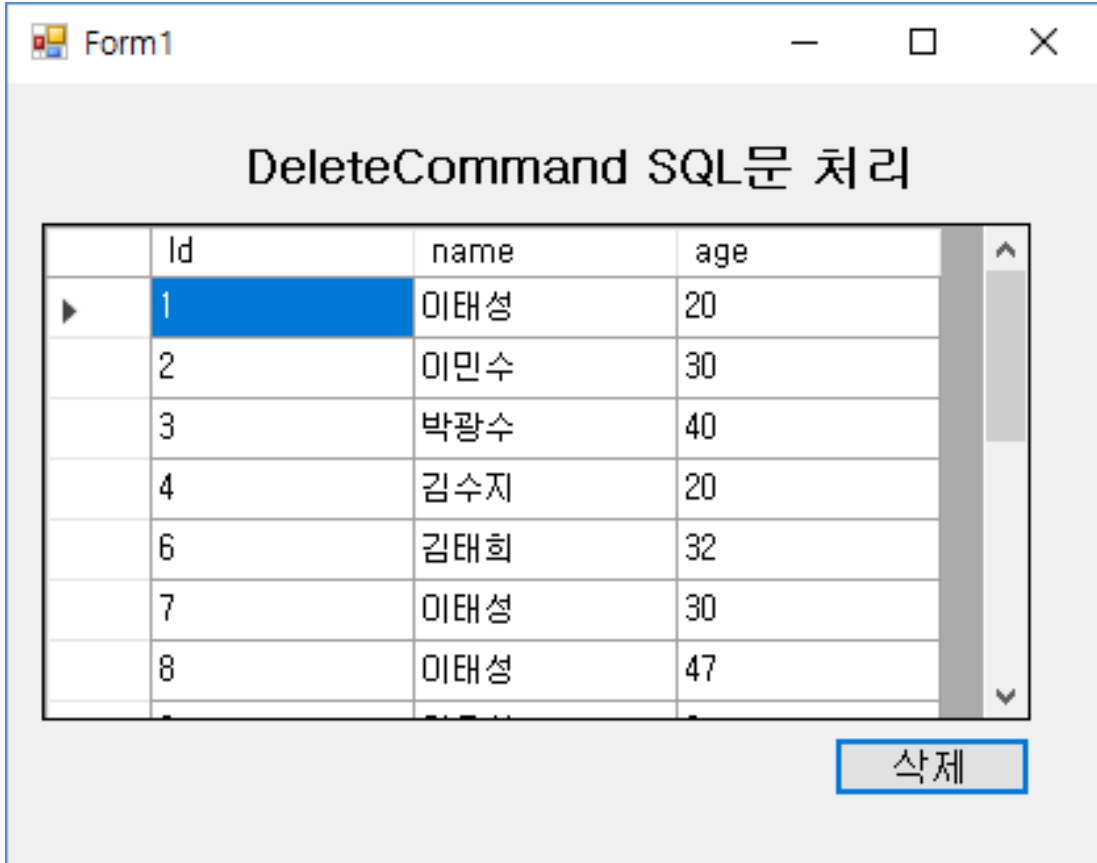
삭제

소스보기

SQL Delete 실행 – Update() 활용

Update()를 사용하여 데이터를 입력

- database Example table: tblEx01



The screenshot shows a Windows application window titled 'Form1'. Inside the window, there is a label 'DeleteCommand SQL문 처리'. Below the label is a table with 4 columns: 'Id', 'name', and 'age'. The table contains 8 rows of data. The first row (Id: 1, name: 이태성, age: 20) is highlighted in blue. To the right of the table is a vertical scrollbar. Below the table is a button labeled '삭제' (Delete).

	Id	name	age
▶	1	이태성	20
	2	이민수	30
	3	박광수	40
	4	김수지	20
	6	김태희	32
	7	이태성	30
	8	이태성	47

삭제

UpdateCommand

수정

UpdateCommand

속성

- `public SqlCommand UpdateCommand { get; set; }`

갱신 SQL문

- `UPDATE table명 SET 컬럼=@매개변수, WHERE 조건`

갱신 과정

- ① 기본 키(primary key) 검색 ➔ where
- ② 갱신 SQL 실행
- ③ 결과 출력

UpdateCommand – 동적 SQL문 사용

동적 SQL문 작성

- “UPDATE table명 SET 컬럼1 =@매개변수1 WHERE 조건

매개변수 정의

- SqlDataAdapter.UpdateCommand.Parameters.Add()

SQL문 실행

- SqlDataAdapter.UpdateCommand.ExecuteNonQuery()

UpdateCommand – 동적 SQL문 사용

동적 SQL문 실습

- database : example table : tblEx01
- 이름과 나이 데이터 수정

Form1

SQL UpdateCommand 처리

	Id	name	age
▶	24	하미	31
	26	김민수	12
	27	이태성	27
*			

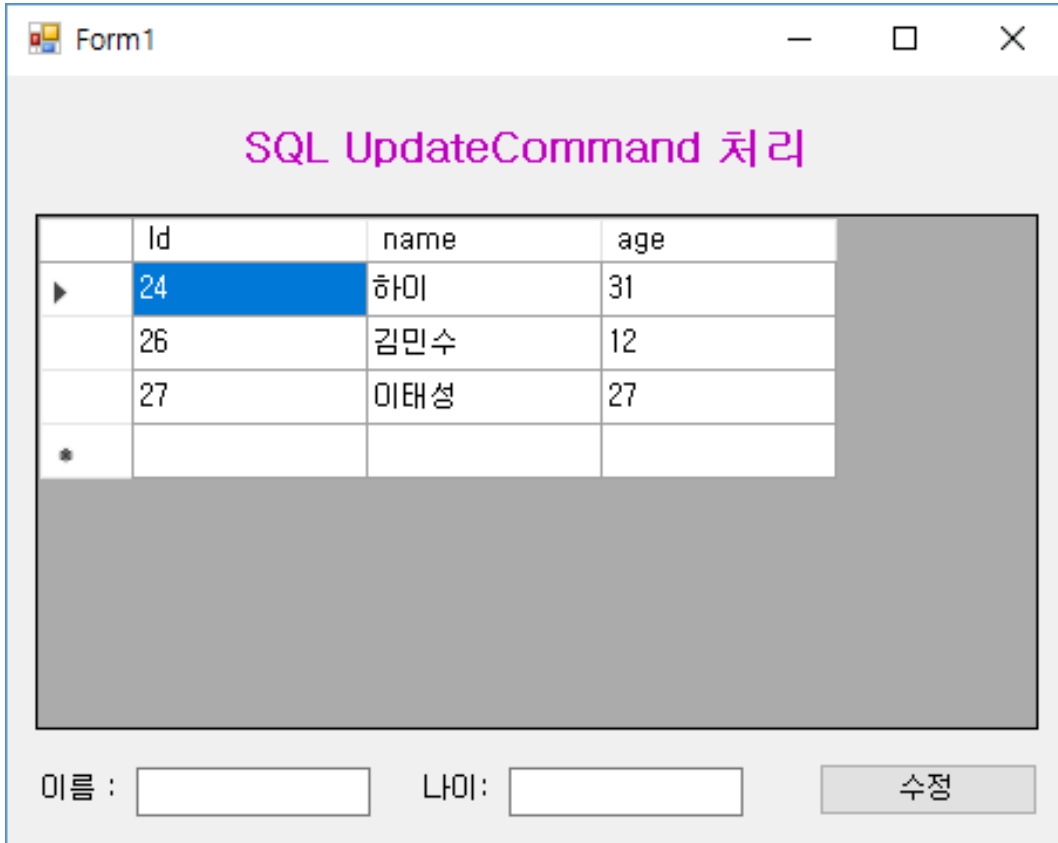
이름 : 나이 :

소스보기

SQL Update 실행 – Update() 활용

Update()를 사용하여 데이터 수정

- database Example table: tblEx01



Form1

SQL UpdateCommand 처리

	Id	name	age	
▶	24	하미	31	
	26	김민수	12	
	27	이태성	27	
*				

이름 : 나이:

소스보기

SQL Command 한꺼번에 다루기

SelectCommand, InsertCommand,
UpdateCommand, DeleteCommand

SqlCommand 설정

SqlCommand 설정 방법

- SqlDataAdapter에 SqlCommand 설정을 모아서 처리
- SqlCommandBuilder 를 통해 자동 생성

✓ 제약사항

- ① 하나의 테이블만 읽어서 처리
- ② Join된 쿼리문은 처리 불가
- ③ 기본 키가 반드시 있어야 한다.
- ④ SqlDataAdapter를 생성할 때 Select 문 반드시 명시

SqlCommand 모아서 처리

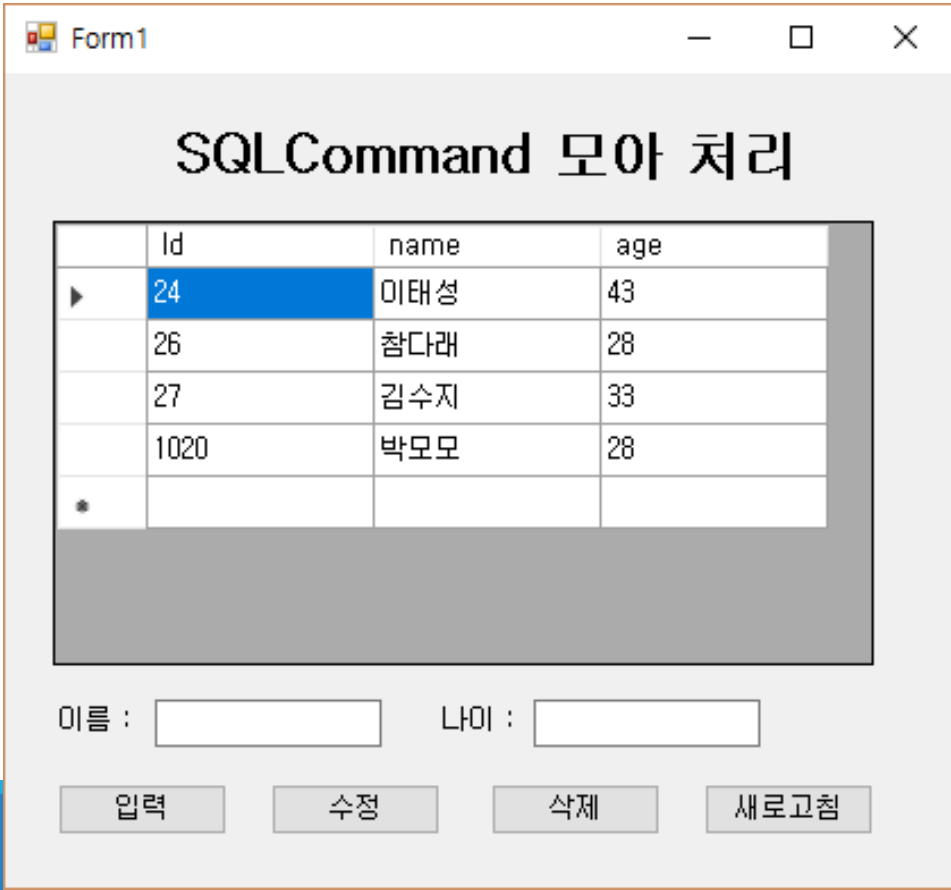
❖ 과정

- 하나의 함수 안에 SqlCommand 를 모두 설정
- Fill()과 Update()를 사용하여 처리 ➡ DataSet 사용

SqlCommand 모아서 처리

❖ 실습

- database : example , table : tblEx01
- 입력, 수정, 삭제, 새로 고침이 될 수 있도록 이전 소스 수정



Form1

SqlCommand 모아 처리

	Id	name	age
▶	24	이태성	43
	26	참다래	28
	27	김수지	33
	1020	박모모	28
*			

이름 : 나이 :

소스보기

SqlCommandBuilder 의 SqlCommand 처리

SqlCommandBuilder

- JOIN이 없는 일반적인 SQL문을 유틸리티로 해결
- 연결된 SQL Server 데이터베이스와 DataSet의 변경 내용을 조정하는 데 사용되는 단일 테이블 명령을 자동으로 생성합니다 (MSDN 해설)

생성자

- `public SqlCommandBuilder(SqlDataAdapter adapter)`

SqlCommandBuilder를 사용한 SqlCommand처리

SqlCommandBuilder를 생성하기 전에 Select 문 설정

- dataAdapter = new SqlDataAdapter("SELECT * FROM tblEx01", Conn);
- cmdBuilder = new SqlCommandBuilder(dataAdapter);

The screenshot shows a Windows Form titled 'Form1' with a title bar containing standard Windows window controls. The form has a light gray background and a title 'SqlCommandBuilder 처리' centered at the top. Below the title is a table with four columns: 'Id', 'name', and 'age'. The table contains five rows of data. The first row is highlighted in blue. Below the table is a gray rectangular area. At the bottom of the form, there are two input fields labeled '이름 :' and '나이 :'. Below these fields are four buttons: '입력', '수정', '삭제', and '새로고침'.

	Id	name	age
▶	24	이태성	27
	26	라면	12
	27	김수지	33
	1021	이철수	24
*			

이름 : 나이 :

소스보기

관계형 데이터베이스 처리

DataSet

❖ 관계형 데이터베이스 example02 구조

- parent table : tblPeople

no	name	age
int, primary key	string, max:15	int

- child table : tblFruit

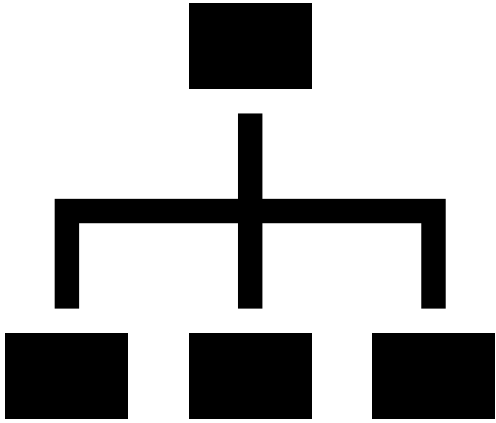
ex	no	fruit
int, primary key	int, foreign key	string, max:20

- 관계 설정

- MS SQL
primary key, foreign key

DataSet

- ❖ 레코드 삭제할 때 주의할 점
 - 부모 테이블의 내용을 삭제할 때는 자식 테이블의 내용부터 먼저 삭제할 것



DataSet

❖ 관계형 데이터베이스 실습

- database : example02 / tblPeople, tblFruit

Form1

관계형 데이터베이스 실습

people table

	no	name	age
▶	5	이태성	20
	6	홍길동	32
*			

fruit table

	id	no	fruit
▶	6	6	사과
	12	5	수박
	13	6	복숭아
	14	6	참외
*			

이름 : 나이 :

입력 삭제 수정 과일 보기

이름 : 과일 :

입력 삭제 수정