

채용 정보 관리 프로그램 DB 구축


SQL을 이용한 프로그램 개발

A조 강예영 서주형 윤철희 이상헌 이은수





TABLE OF CONTENTS

- 
- 01 개발목표 및 개발과정
 - 02 전체구조
 - 03 상세기능
 - 04 시연

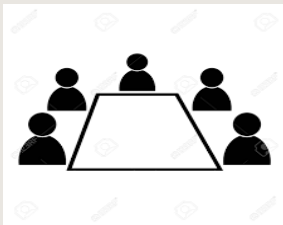


개발목표

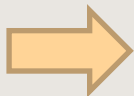
“구직자는 원하는 기업에 대한 채용정보를 얻고
기업은 원하는 구직자의 정보를 얻을 수 있다.”



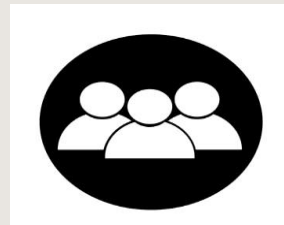
개발과정



아이디어 회의



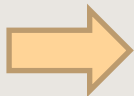
개발환경 설정



작업 분배



DB 구축 후 C#과 연동



코딩 및 오류 수정



결과 종합



개발환경

- Visual Studio 2019(.Net Framework 4.5.2)
- Microsoft SQL Server Management Studio



개발과정

관리 항목	담당자	일정구분	09월 05일	09월 06일	09월 07일	09월 08일	09월 09일
DATA-TREE 구현	이상현, 이은수	예정일					
		완료일					
SQL 명령 TEXTBOX 기능 구현	강예영, 윤철희	예정일					
		완료일					
Table 제작	서주형	예정일					
		완료일					
포트폴리오 제작	강예영	예정일					
		완료일					

예정 : → 진행률
 완료 : ●



전체구조

클래스 구조

DataSource



```
class DataSource
{
    private static SqlConnection con = null;
    private static string strcon = "Data Source=192.168.0.39;Initial Catalog=RecruitDataBase;User ID=GroupA;Password=1234";

    참조 4개
    public static SqlConnection GetConnection() // Sql 연결...
    참조 4개
    public static void Close(SqlConnection con) // Sql 연결 끊기
}
```

MainDataSource



```
class MainDataSource
{
    참조 6개
    public void DoSql(RichTextBox textBox) // select 이외 구문 들어올 때 ...
    참조 2개
    public DataTable DoSqlSelect(RichTextBox textBox) // select 구문 들어올 때...
}
```

SqlStringCheck



```
class SqlStringCheck
{
    MainDataSource ddl = new MainDataSource();

    //List<string> c3 = new List<string> { "USE" };
    List<string> c4 = new List<string> { "drop", "exec" };
    List<string> c5 = new List<string> { "alter" };
    List<string> c6 = new List<string> { "create", "insert", "delete", "select", "update" };
    List<string> c14 = new List<string> { "DROP PROCEDURE" };
    List<string> c15 = new List<string> { "ALTER PROCEDURE" };
    List<string> c16 = new List<string> { "CREATE PROCEDURE" };

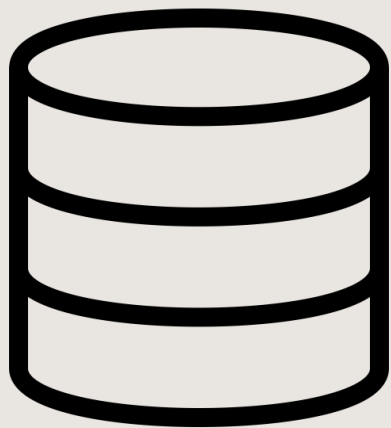
    참조 1개
    private void IsCheck(RichTextBox textBox, DataGridView dataGridView) // 입력받은 쿼리구문을 검열 후 해당 함수로 보내줌...
    참조 1개
    public void Stringcheck(RichTextBox textBox, DataGridView dataGridView, TabPage tabPage) // 쿼리구문 실행 후 textBox, dataGridView에 담기...
}
```

전체구조

User_info

Company_info

Recruit_info



DB

	열 이름	데이터 형식	Null 허용
▶ 🔑	IDX	numeric(18, 0)	<input type="checkbox"/>
	name	varchar(50)	<input checked="" type="checkbox"/>
	duty	varchar(50)	<input checked="" type="checkbox"/>
	career	numeric(18, 0)	<input checked="" type="checkbox"/>
	Gender	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

	열 이름	데이터 형식	Null 허용
▶ 🔑	IDX	numeric(18, 0)	<input type="checkbox"/>
	employeeName	varchar(50)	<input checked="" type="checkbox"/>
	industry	varchar(50)	<input checked="" type="checkbox"/>
	scale	numeric(18, 0)	<input checked="" type="checkbox"/>
	address	varchar(50)	<input checked="" type="checkbox"/>

	열 이름	데이터 형식	Null 허용
▶ 🔑	idx	numeric(18, 0)	<input type="checkbox"/>
	employeeName	varchar(50)	<input checked="" type="checkbox"/>
	hireNumber	numeric(18, 0)	<input checked="" type="checkbox"/>
	duty	nchar(10)	<input checked="" type="checkbox"/>
	salary	numeric(18, 0)	<input checked="" type="checkbox"/>
	career	numeric(18, 0)	<input checked="" type="checkbox"/>
	Major	varchar(50)	<input checked="" type="checkbox"/>
	enterpriseForm	varchar(50)	<input checked="" type="checkbox"/>



상세기능

- TreeView 구현
- 사용자 작성 쿼리(실행 버튼) 구현



TreeView(테이블 조회)

The screenshot shows the AGroupManagement application interface. On the left, a TreeView displays the database structure with 'User' selected under the 'Table' category. A red box highlights the 'User' entry, and a red label '해당 테이블 조회' (Query this table) points to it. The main area displays a table of user data. A red box highlights the table, and a red label '조회 결과 출력' (Output query results) points to the bottom of the table. On the right, the 'SQL 쿼리문 작성' (Write SQL query) panel is empty, with buttons for '실행' (Execute), '초기화' (Reset), and a third button.

IDX	name	duty	career	Gender
1	서주형	요리사	3	남성
2	이은수	사원	3	남성
3	강예영	회사원	5	여성
4	윤철희	프로게이머	4	남성
5	송진영	프로그래머	6	남성
6	구예진	사무직	4	여성
7	황윤정	간호직	9	여성
8	박진희	사무직	1	여성
9	이상현	사장	10	남성
12	영구	바보	5	남성
13	영구	바보	5	남성
14	맹칠미	거지	5	여성
15	맹칠미	거지	5	여성
16	맹칠미	거지	5	여성
17	맹칠미	거지	5	여성
18	주호민	작가	15	중성
19	주호민	작가	15	중성
20	철수	연구원	3	남성

SQL 쿼리문 작성

실행 초기화

조회 결과 출력

● TreeView(프로시저/DDL)

AGroupManagement

파일(F) 편집(E) 보기(V) 프로젝트(P) 도구(T) 창(W) 도움말(H)

Table
 User
 Company
 Recruit
 Procedure
 DDL
 create
 alter
 drop
 List
 User_Select
 Company_Select
 Company_Insert
 Company_Update
 Company_Delete
 Recruit_Select
 Recruit_Insert
 Recruit_Update
 Recruit_Delete
Sequence

IDX	employeeName	hireNumber	duty	salary	career	Major	enter
1	현대자동차	1	자동차제조	5500	5	기계공학	대기업
2	메인트런스	202	철도 운송지원...	3200	2	자동차학과	중견기업
3	케이지이티에스	154	지정외폐기물...	3800	4	전기안전관리업	중견기업
4	제일제당	7279	설탕제조업	4200	5	식품학과	대기업
5	영진제약	634	완제의약품제조...	3700	4	약학과	대기업
6	유니드	450	기타기초무기...	5200	3	화학학과	대기업
	떡잎상사	4	조선무역업	3500	4	경제학과	대기업

SQL 쿼리문 작성

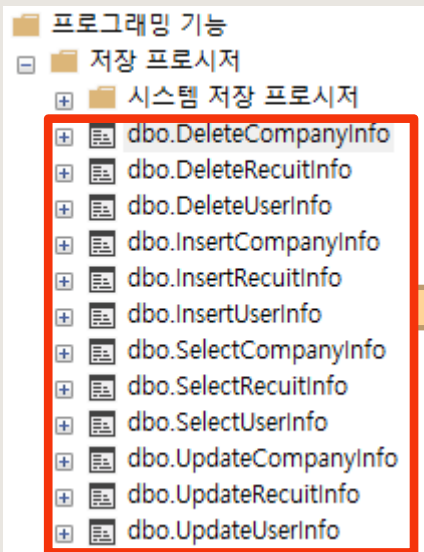
```
CREATE PROCEDURE --(프로시저명 입력)
AS
BEGIN
SET NOCOUNT ON;
END
```

해당 프로시저 조회

해당 프로시저 자동 생성

결과 메세지 hi 실행 초기화

● TreeView(프로시저/List)



프로시저 자동 생성

DB에 구축된
프로시저



실행버튼

SQL 쿼리문 작성

```
select * from User_info;
```

사용자가 쿼리 입력

실행

초기화



AGroupManagement

파일(F) 편집(E) 보기(V) 프로젝트(P) 도구(T) 창(W) 도움말(H)

Table
Procedure
Sequence

IDX	name	duty	career	Gender
5	영구	연구원	5	중성
6	서주현	요리사	5	남성
7	송진영	프로그래머	7	남성
8	윤철희	프로게이머	2	남성

SQL 쿼리문 작성

```
select * from User_info;
```

결과 메시지 hi

실행 초기화

서버로부터 결과 출력



시연





THANKS!

Any question?