

NE > < ACRO

PLATFORM
Education for SeedIT

Concept & 5 Demo

Nexacro

1. Nexacro N 제품에서는 iOS 운영체제 앱 배포 시 Update Type을 "Server"로 설정할 경우 LiteDBConnection 오브젝트에서 사용할 DB파일을 직접 배포할 수 없습니다. iOS 운영체제의 권한 처리에 대한 제약이며 LiteDBConnection 오브젝트를 사용해야 하는 경우 아래와 같이 스크립트로 DB 파일을 직접 생성해서 사용할 수 있습니다.
2. 체험판 버전 NexacroN에서는 LiteDBConnection, LiteDBStatement를 이용하여 자체DB인 sqlite(LiteDB)연동시 LiteDBConnection.set_datasource(path)에서 error로 인하여 DB open이 안되므로 Nexacro17 개발용 버전으로 진행하겠습니다.
3. SQLite 커맨드 라인 툴은 사용자가 모든 명령을 직접 입력해야 하고 결과물을 확인할 수 있는 UI 인터페이스를 제공하지 않습니다. 만일 사용이 불편하다면 DB Browser for SQLite와 같이 UI를 제공하는 애플리케이션을 사용해도 같은 결과를 얻을 수 있습니다.
4. LiteDBConnection, LiteDBStatement는 데이터베이스를 연결하고 쿼리를 수행하여 데이터를 조작할 수 있게 해주는 오브젝트입니다. 오브젝트 이름에서 알 수 있듯이 경량 데이터베이스 엔진인 SQLite를 연결해 사용할 수 있으며 사용자에게 SQLite를 조작할 수 있는 API를 제공합니다.

Nexacro

LiteDBConnection 오브젝트를 스크립트로 DB 파일을 직접 생성하기

```
/**
 * open 메소드 실행 시 LiteDBConnection.openReadWriteCreate 옵션을 설정합니다.
 */
this.Button00_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
    this.LiteDBConnection00.open("%USERAPP%.s3db", LiteDBConnection.openReadWriteCreate);
}

/**
 * DB 파일을 생성하고 연결된 상태에서 데이터 테이블을 생성하고 필요한 작업을 처리합니다.
 */
var CREATE_QUERY = "CREATE TABLE Persons ()";
this.LiteDBConnection00_onsuccess = function(obj:nexacro.LiteDBConnection,e:nexacro.LiteDBEventInfo)
{
    switch(e.reason)
    {
        case 5: // LiteDBConnection open()
            this.LiteDBStatement00.set_ldbconnection(this.LiteDBConnection00);
            this.LiteDBStatement00.executeUpdate(CREATE_QUERY);
            break;
    }
}
```

RP	작성일	설명
90369	2020-10-08	앱 배포 방식에 따른 제약사항입니다. DB 파일을 직접 배포하지 않고 특정 시점에 DB 파일을 생성해 접근하는 방식으로 제약사항을 회피할 수 있습니다.


1.SQLite 커맨드 라인 툴 설치하기

파일명	설명
sqlite3.exe	데이터베이스를 생성, SQL문을 수행하여 데이터 베이스 관리
sqlite3_analyzer.exe	데이터베이스 파일을 분석
sqldiff.exe	데이터베이스 파일 비교

2.SQLite 커맨드 라인 툴 설치하기

[SQLite Download](#)**Click!!**

sqlite-tools-win32-x86-3410000.zip파일을
click하여 다운로드한다.



Home About Documentation Download License Support Purchase

SQLite Download Page

Source Code

[sqlite-amalgamation-3410000.zip](#) (2.50 MiB) C source code as an [amalgamation](#), version 3.41.0.
(SHA3-256: 052c721508b208403ab5ba939ee3cead56a8bd03225ffc778948e3c59c83b7ee)

[sqlite-autoconf-3410000.tar.gz](#) (2.98 MiB) C source code as an [amalgamation](#). Also includes a "configure" script and [TEA](#) makefiles for the [TCL Interpreter](#).
(SHA3-256: d783ab44a2b44394331d392b8b8d4d2ea4964cbb2bafc7c6c649bcbdb3c9ffe)

Documentation

[sqlite-doc-3410000.zip](#) (10.47 MiB) Documentation as a bundle of static HTML files.
(SHA3-256: 61be0faa6599bc8ac2596b3be3db3edc0d6b5fc5d0a8f0b6b2f897ea22dedae4)

Precompiled Binaries for Android

[sqlite-android-3410000.aar](#) (3.26 MiB) A precompiled Android library containing the core SQLite together with appropriate Java bindings, ready to be used in an Android application.
(SHA3-256: 20a7da773fdb2560f90cfe4f8c088b930df905039b73c2db3a893cb972700884)

Precompiled Binaries for Linux

[sqlite-tools-linux-x86-3410000.zip](#) (2.16 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#).
(SHA3-256: 7f601bfcd9ae63ff12d0136d71f41468618342e944c677f0b3c5ec35646e37)

Precompiled Binaries for Mac OS X (x86)

[sqlite-tools-osx-x86-3410000.zip](#) (1.54 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#).
(SHA3-256: 11a765584ee7819b5b734f95c89f788bcac68912c7ca13082758181034f3ce5)

Precompiled Binaries for Windows

[sqlite-dll-win32-x86-3410000.zip](#) (562.90 KiB) 32-bit DLL (x86) for SQLite version 3.41.0.
(SHA3-256: 4494f81f53e713fe17823088ee3574db21a7fde6d477916b9c8bb7d2e293c93e)

[sqlite-dll-win64-x64-3410000.zip](#) (903.85 KiB) 64-bit DLL (x64) for SQLite version 3.41.0.
(SHA3-256: e58b92f93d54348cea1351479719f9685fc86a043841b569bc3528b8b2ba1536)

[sqlite-tools-win32-x86-3410000.zip](#) (1.91 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#).
(SHA3-256: 94c8e42e1cc9cb92a3781dbbcd36d3a3227e94e6a6e0ff7aa12fce78a745210)

Nexacro-DB연동

SQLite database 생성 및 오픈하기

1. 데이터베이스 파일 오픈하기(기존에 생성한 데이터베이스가 있을 경우)

이름	수정한 날짜	유형	크기
sqldiff.exe	2023-02-22 오전 4:39	응용 프로그램	574KB
sqlite3.exe	2023-02-22 오전 4:39	응용 프로그램	1,106KB
sqlite3_analyzer.exe	2023-02-22 오전 4:39	응용 프로그램	2,053KB

1. 압축을 풀면 좌측과 같이 exe파일 3개중에서 sqlite3.exe 더블클릭하여 실행한다.

```
C:\JAVA\sqlite-tools-win32-x86-3410000\sqlite3.exe
SQLite version 3.41.0 2023-02-21 18:09:37
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open nexacro_sample_db.sqlite
sqlite> .databases
main: C:\JAVA\sqlite-tools-win32-x86-3410000\nexacro_sample_db.sqlite r/w
sqlite>
```

1. .open nexacro_sample_db.sqlite 입력하여 database를 open한다.
2. .databases 명령어를 입력하여 nexacro_sample_db.sqlite파일을 확인할 수 있다.

2.데이터베이스 파일 생성 및 오픈하기(기존에 생성한 데이터베이스가 없을경우)

```
C:\JAVAM\sqlite-tools-win32-x86-3410000>sqlite3.exe nexacro_sample_db.sqlite ①
SQLite version 3.41.0 2023-02-21 18:09:37
Enter ".help" for usage hints.
sqlite> .databases ②
main: C:\JAVAM\sqlite-tools-win32-x86-3410000\nexacro_sample_db.sqlite r/w
sqlite>
```

1. sqlite3.exe 파일이 위치한 경로로 이동 후 커맨드 프롬프트에서 'sqlite3.exe nexacro_sample_db.sqlite' 라고 입력합니다.
(만일 해당 경로에 같은 파일이 존재하면 파일을 새로 생성하지 않고 기존 파일을 오픈합니다.)
2. '.databases' 명령어로 현재 오픈된 데이터베이스 파일을 확인할 수 있습니다.

1. TB_POS(직책 정보), TB_EMP(직원 정보), TB_DEPT(부서 정보) 테이블을 생성한다.

```
/* TB_POS Table */
CREATE TABLE IF NOT EXISTS `TB_POS` (
  `POS_CD`      VARCHAR2 ( 2 ),
  `POS_NAME`    VARCHAR2 ( 50 )
);
```

```
/* TB_EMP Table */
CREATE TABLE IF NOT EXISTS `TB_EMP` (
  `EMPL_ID`     VARCHAR2 ( 5 ),
  `FULL_NAME`   VARCHAR2 ( 50 ) DEFAULT (null),
  `DEPT_CD`     VARCHAR2 ( 2 ) DEFAULT (null),
  `POS_CD`      VARCHAR2 ( 2 ),
  `HIRE_DATE`   VARCHAR2 ( 8 ),
  `GENDER`      VARCHAR2 ( 1 ),
  `MARRIED`     BOOLEAN ( 1 ),
  `SALARY`      INTEGER ( 12 ),
  `MEMO`        TEXT
);
```

```
/* TB_DEPT Table */
CREATE TABLE IF NOT EXISTS `TB_DEPT` (
  `DEPT_CD`     VARCHAR2 ( 2 ) DEFAULT (NULL),
  `DEPT_NAME`   VARCHAR2 ( 50 ) DEFAULT (NULL)
);
```

```
sqlite> create table if not exists 'TB_POS'(
(x1...> 'POS_CD' VARCHAR2(2),
(x1...> 'POS_NAME' VARCHAR2(50));
sqlite> create table if not exists 'TB_EMP'(
(x1...> 'EMPL_ID' VARCHAR2(5),
(x1...> 'FULL_NAME' VARCHAR2(50) DEFAULT(null),
(x1...> 'DEPT_CD' VARCHAR2(2) DEFAULT(null),
(x1...> 'POS_CD' VARCHAR2(2),
(x1...> 'HIRE_DATE' VARCHAR2(8),
(x1...> 'GENDER' VARCHAR2(1),
(x1...> 'MARRIED' BOOLEAN(1),
(x1...> 'SALARY' INTEGER(12),
(x1...> 'MEMO' TEXT);
sqlite> create table if not exists 'TB_DEPT'(
(x1...> 'DEPT_CD' VARCHAR2(2) DEFAULT (null),
(x1...> 'DEPT_NAME' VARCHAR2(50) DEFAULT (null));
sqlite> .tables
TB_DEPT  TB_EMP  TB_POS
sqlite>
```

SQL문 수행이 완료된 후 테이블이 정상적으로 생성되었는지 '.tables' 명령어로 확인합니다.

1. 테이블을 생성했으면 이제 실제 데이터를 입력합니다.

TB_POS TABLE

```
/* TB_POS Table */
INSERT INTO `TB_POS` VALUES ('07','Officer');
INSERT INTO `TB_POS` VALUES ('06','Supervisor');
INSERT INTO `TB_POS` VALUES ('05','Assistant Manager');
INSERT INTO `TB_POS` VALUES ('04','Department Manager');
INSERT INTO `TB_POS` VALUES ('03','Division Manager');
INSERT INTO `TB_POS` VALUES ('02','President');
INSERT INTO `TB_POS` VALUES ('01','Chairman');
```

TB_DEPT TABLE

```
/* TB_DEPT Table */
INSERT INTO `TB_DEPT` VALUES ('01','Accounting Team');
INSERT INTO `TB_DEPT` VALUES ('02','Finances Team');
INSERT INTO `TB_DEPT` VALUES ('03','Human Resource Team');
INSERT INTO `TB_DEPT` VALUES ('04','Marketing Team');
INSERT INTO `TB_DEPT` VALUES ('05','Sales Team');
INSERT INTO `TB_DEPT` VALUES ('06','Consulting Team');
INSERT INTO `TB_DEPT` VALUES ('07','Design Team');
INSERT INTO `TB_DEPT` VALUES ('08','Mobile Team');
INSERT INTO `TB_DEPT` VALUES ('09','Education Team');
INSERT INTO `TB_DEPT` VALUES ('10','Technical Support Team');
INSERT INTO `TB_DEPT` VALUES ('11','Customer Support Team');
```

TB_EMP TABLE

```
/* TB_EMP Table */
INSERT INTO `TB_EMP` VALUES ('AA001','Oleg','08','03','20072504','W',1,70000,'green');
INSERT INTO `TB_EMP` VALUES ('AA010','Aladdin','07','06','20041909','M',0,120000,'green');
INSERT INTO `TB_EMP` VALUES ('BB010','Curran','02','03','20070111','M',1,210000,'orange');
INSERT INTO `TB_EMP` VALUES ('BB020','Adam','07','07','20011602','W',1,90000,'indigo');
INSERT INTO `TB_EMP` VALUES ('BB030','Heather','03','03','20061406','M',0,50000,'blue');
INSERT INTO `TB_EMP` VALUES ('CC110','Lester','07','05','20171604','M',0,100000,'green');
INSERT INTO `TB_EMP` VALUES ('CD120','Elijah','10','01','20151304','W',0,110000,'gold');
INSERT INTO `TB_EMP` VALUES ('AB100','Angela','08','04','20021110','M',1,670000,'blue');
INSERT INTO `TB_EMP` VALUES ('AC310','Rae','02','05','20072603','W',1,90000,'green');
INSERT INTO `TB_EMP` VALUES ('AC210','Denton','02','04','20172503','M',1,90000,'skyblue');
INSERT INTO `TB_EMP` VALUES ('DD002','Nora','05','01','20160202','W',0,420000,'blue');
INSERT INTO `TB_EMP` VALUES ('DD200','Adrian','05','04','20160212','W',0,80000,'violet');
INSERT INTO `TB_EMP` VALUES ('AD020','Ulric','04','04','20042107','W',0,420000,'yellow');
INSERT INTO `TB_EMP` VALUES ('BD030','Veronica','10','06','20130701','M',1,70000,'green');
INSERT INTO `TB_EMP` VALUES ('AA200','Phyllis','02','06','20120812','W',1,350000,'orange');
```

1. 테이블을 생성했으면 이제 실제 데이터를 입력합니다.

TB_POS TABLE

아래 'TB_POS' 테이블의 INSERT 문을 복사해서 sqlite 프롬프트에다 붙여넣기 하여 직책정보를 입력한다.

```
INSERT INTO `TB_POS` VALUES ('07','Officer');  
INSERT INTO `TB_POS` VALUES ('06','Supervisor');  
INSERT INTO `TB_POS` VALUES ('05','Assistant Manager');  
INSERT INTO `TB_POS` VALUES ('04','Department Manager');  
INSERT INTO `TB_POS` VALUES ('03','Division Manager');  
INSERT INTO `TB_POS` VALUES ('02','President');  
INSERT INTO `TB_POS` VALUES ('01','Chairman');
```

```
sqlite> INSERT INTO `TB_POS` VALUES ('07','Officer');  
sqlite> INSERT INTO `TB_POS` VALUES ('06','Supervisor');  
sqlite> INSERT INTO `TB_POS` VALUES ('05','Assistant Manager');  
sqlite> INSERT INTO `TB_POS` VALUES ('04','Department Manager');  
sqlite> INSERT INTO `TB_POS` VALUES ('03','Division Manager');  
sqlite> INSERT INTO `TB_POS` VALUES ('02','President');  
sqlite> INSERT INTO `TB_POS` VALUES ('01','Chairman');
```

1. 테이블을 생성했으면 이제 실제 데이터를 입력합니다.

TB_EMP TABLE

아래 'TB_EMP' 테이블의 INSERT 문을 복사해서 sqlite 프롬프트에다 붙여넣기 하여 직원정보를 입력한다.

```
INSERT INTO `TB_EMP` VALUES ('AA001','Oleg','08','03','20072504','W',1,70000,'green');
INSERT INTO `TB_EMP` VALUES ('AA010','Aladdin','07','06','20041909','M',0,120000,'green');
INSERT INTO `TB_EMP` VALUES ('BB010','Curran','02','03','20070111','M',1,210000,'orange');
INSERT INTO `TB_EMP` VALUES ('BB020','Adam','07','07','20011602','W',1,90000,'indigo');
INSERT INTO `TB_EMP` VALUES ('BB030','Heather','03','03','20061406','M',0,50000,'blue');
INSERT INTO `TB_EMP` VALUES ('CC110','Lester','07','05','20171604','M',0,100000,'green');
INSERT INTO `TB_EMP` VALUES ('CD120','Elijah','10','01','20151304','W',0,110000,'gold');
INSERT INTO `TB_EMP` VALUES ('AB100','Angela','08','04','20021110','M',1,670000,'blue');
INSERT INTO `TB_EMP` VALUES ('AC310','Rae','02','05','20072603','W',1,90000,'green');
INSERT INTO `TB_EMP` VALUES ('AC210','Denton','02','04','20172503','M',1,90000,'skyblue');
INSERT INTO `TB_EMP` VALUES ('DD002','Nora','05','01','20160202','W',0,420000,'blue');
INSERT INTO `TB_EMP` VALUES ('DD200','Adrian','05','04','20160212','W',0,80000,'violet');
INSERT INTO `TB_EMP` VALUES ('AD020','Ulric','04','04','20042107','W',0,420000,'yellow');
INSERT INTO `TB_EMP` VALUES ('BD030','Veronica','10','06','20130701','M',1,70000,'green');
INSERT INTO `TB_EMP` VALUES ('AA200','Phyllis','02','06','20120812','W',1,350000,'orange');
```

```
sqlite> INSERT INTO `TB_EMP` VALUES ('AA001','Oleg','08','03','20072504','W',1,70000,'green');
sqlite> INSERT INTO `TB_EMP` VALUES ('AA010','Aladdin','07','06','20041909','M',0,120000,'green');
sqlite> INSERT INTO `TB_EMP` VALUES ('BB010','Curran','02','03','20070111','M',1,210000,'orange');
sqlite> INSERT INTO `TB_EMP` VALUES ('BB020','Adam','07','07','20011602','W',1,90000,'indigo');
sqlite> INSERT INTO `TB_EMP` VALUES ('BB030','Heather','03','03','20061406','M',0,50000,'blue');
sqlite> INSERT INTO `TB_EMP` VALUES ('CC110','Lester','07','05','20171604','M',0,100000,'green');
sqlite> INSERT INTO `TB_EMP` VALUES ('CD120','Elijah','10','01','20151304','W',0,110000,'gold');
sqlite> INSERT INTO `TB_EMP` VALUES ('AB100','Angela','08','04','20021110','M',1,670000,'blue');
sqlite> INSERT INTO `TB_EMP` VALUES ('AC310','Rae','02','05','20072603','W',1,90000,'green');
sqlite> INSERT INTO `TB_EMP` VALUES ('AC210','Denton','02','04','20172503','M',1,90000,'skyblue');
sqlite> INSERT INTO `TB_EMP` VALUES ('DD002','Nora','05','01','20160202','W',0,420000,'blue');
sqlite> INSERT INTO `TB_EMP` VALUES ('DD200','Adrian','05','04','20160212','W',0,80000,'violet');
sqlite> INSERT INTO `TB_EMP` VALUES ('AD020','Ulric','04','04','20042107','W',0,420000,'yellow');
sqlite> INSERT INTO `TB_EMP` VALUES ('BD030','Veronica','10','06','20130701','M',1,70000,'green');
sqlite> INSERT INTO `TB_EMP` VALUES ('AA200','Phyllis','02','06','20120812','W',1,350000,'orange');
```

1. 테이블을 생성했으면 이제 실제 데이터를 입력합니다.

TB_DEPT TABLE

아래 'TB_DEPT' 테이블의 INSERT 문을 복사해서 sqlite 프롬프트에다 붙여넣기 하여 부서정보를 입력한다.

```
INSERT INTO `TB_DEPT` VALUES ('01','Accounting Team');  
INSERT INTO `TB_DEPT` VALUES ('02','Finances Team');  
INSERT INTO `TB_DEPT` VALUES ('03','Human Resource Team');  
INSERT INTO `TB_DEPT` VALUES ('04','Marketing Team');  
INSERT INTO `TB_DEPT` VALUES ('05','Sales Team');  
INSERT INTO `TB_DEPT` VALUES ('06','Consulting Team');  
INSERT INTO `TB_DEPT` VALUES ('07','Design Team');  
INSERT INTO `TB_DEPT` VALUES ('08','Mobile Team');  
INSERT INTO `TB_DEPT` VALUES ('09','Education Team');  
INSERT INTO `TB_DEPT` VALUES ('10','Technical Support Team');  
INSERT INTO `TB_DEPT` VALUES ('11','Customer Support Team');
```

```
sqlite> INSERT INTO `TB_DEPT` VALUES ('01','Accounting Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('02','Finances Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('03','Human Resource Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('04','Marketing Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('05','Sales Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('06','Consulting Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('07','Design Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('08','Mobile Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('09','Education Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('10','Technical Support Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('11','Customer Support Team');
```

데이터 입력 및 확인을 완료하면 '.exit' 명령으로 프로그램을 종료합니다. 생성된 nexacro_sample_db.sqlite 파일은 별다른 처리없이 LiteDB 오브젝트에 연결해 사용이 가능하다.

1. 테이블을 생성했으면 이제 실제 데이터를 입력합니다.

TB_DEPT TABLE

아래 'TB_DEPT' 테이블의 INSERT 문을 복사해서 sqlite 프롬프트에다 붙여넣기 하여 부서정보를 입력한다.

```
INSERT INTO `TB_DEPT` VALUES ('01','Accounting Team');  
INSERT INTO `TB_DEPT` VALUES ('02','Finances Team');  
INSERT INTO `TB_DEPT` VALUES ('03','Human Resource Team');  
INSERT INTO `TB_DEPT` VALUES ('04','Marketing Team');  
INSERT INTO `TB_DEPT` VALUES ('05','Sales Team');  
INSERT INTO `TB_DEPT` VALUES ('06','Consulting Team');  
INSERT INTO `TB_DEPT` VALUES ('07','Design Team');  
INSERT INTO `TB_DEPT` VALUES ('08','Mobile Team');  
INSERT INTO `TB_DEPT` VALUES ('09','Education Team');  
INSERT INTO `TB_DEPT` VALUES ('10','Technical Support Team');  
INSERT INTO `TB_DEPT` VALUES ('11','Customer Support Team');
```

```
sqlite> INSERT INTO `TB_DEPT` VALUES ('01','Accounting Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('02','Finances Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('03','Human Resource Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('04','Marketing Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('05','Sales Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('06','Consulting Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('07','Design Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('08','Mobile Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('09','Education Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('10','Technical Support Team');  
sqlite> INSERT INTO `TB_DEPT` VALUES ('11','Customer Support Team');
```

데이터 입력 및 확인을 완료하면 '.exit' 명령으로 프로그램을 종료합니다. 생성된 nexacro_sample_db.sqlite 파일은 별다른 처리없이 LiteDB 오브젝트에 연결해 사용이 가능하다.

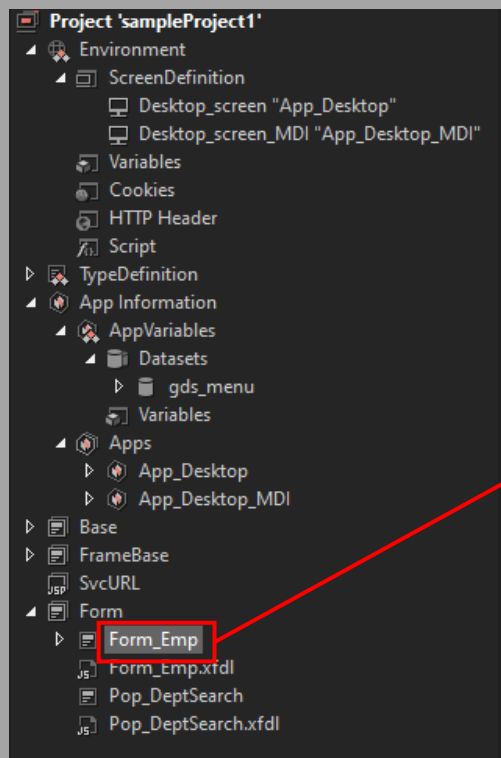
Nexacro-DB연동

데이터베이스 연결 하기

1. 애플리케이션에서 데이터베이스를 연결, 종료할 때는 LiteDBConnection 오브젝트를 사용합니다. async, busytimeout, datasource, openflag, preconnect등의 속성을 설정하면 데이터베이스 연결시 옵션을 설정할 수 있습니다.

Nexacro17과 로컬DB(LiteDB)연동하기

-화면 개발 실습에서 공부했던 SampleProject1 Project로 LiteDB연동하는 작업을 공부해 보겠습니다.



Employees

Retrive add del save

Department Find Gender ☐ All ☐ Male ☐ Female Change reset

No	FULL	DEPT	EMPL	POS	HIRE	SALA	GEND	MARR	MEM
1	Olivia	01	AA001	02	2010-1	83,000	Female	true	ivory
2	John	02	AA002	02	2005-1	76,000	Male	false	greeny
3	Jacksc	02	BB001	03	2007-0	95,000	Male	true	aliceb
4	Maia	04	BB002	01	2009-0	60,000	Female	false	ivory
5	Adam	04	CC001	04	2001-0	88,000	Male	true	greeny
6	Tyler	04	DD001	03	2006-0	60,000	Male	false	lightgr

Name

Emp ID

Department

Position

Gender ☐ Male ☐ Female

Married ☐

Hire Date

Annual Salary

Officer
Assistant Manager
Division Manager

Project Explorer에서 Form->Form_Emp를 Click하여 Form_Emp를 연다.

Nexacro-DB연동

데이터베이스 연결 하기

1. Form_Emp 폼 수정하기

컴포넌트	설명	속성	속성값
 Button	LiteDB open Button	id	btn_dbopen
		text	open
 Edit	LiteDB File 경로	id	edt_dbfile
		value	C:\JAVA\sqlite_LiteDB\nexacro_sample_db.sqlite

Form_Emp 폼에서 아래 그림처럼 button을 추가한다.

Employees

Retrive add del save

Department Find Gender ☐ All ☐ Male ☐ Female Change reset

No	FULL	DEPT	EMPL	POS_	HIRE_	SALA	GEND	MARR	MEM
1	Olivia	01	AA001	02	2010-1	83,000	Female	true	ivory
2	John	02	AA002	02	2005-1	76,000	Male	false	greeny
3	Jacksc	02	BB001	03	2007-0	95,000	Male	true	aliceb
4	Maia	04	BB002	01	2009-0	60,000	Female	false	ivory
5	Adam	04	CC001	04	2001-0	88,000	Male	true	greeny
6	Tyler	04	DD001	03	2006-0	60,000	Male	false	lightgr

Name

Emp ID

Department

Position

Gender ☐ Male ☐ Female

Married ☐

Hire Date

Annual Salary



Employees

C:\JAVA\sqlite_LiteDB\nexacro_sample_db.sqlite Retrive open add del save

Department Find Gender ☐ All ☐ Male ☐ Female Change reset

No	FULL	DEPT	EMPL	POS_	HIRE_	SALA	GEND	MARR	MEM
1	Olivia	01	AA001	02	2010-1	83,000	Female	true	ivory
2	John	02	AA002	02	2005-1	76,000	Male	false	greeny
3	Jacksc	02	BB001	03	2007-0	95,000	Male	true	aliceb
4	Maia	04	BB002	01	2009-0	60,000	Female	false	ivory
5	Adam	04	CC001	04	2001-0	88,000	Male	true	greeny
6	Tyler	04	DD001	03	2006-0	60,000	Male	false	lightgr

Name

Emp ID

Department

Position

Gender ☐ Male ☐ Female

Married ☐

Hire Date

Annual Salary

2. LiteDBConnection, LiteDBStatement 오브젝트 추가하기

컴포넌트/오브젝트	설명	속성	속성값
LiteDBConnection	LiteDB 연결 컴포넌트	id	LiteDBConnection00
LiteDBStatement	LiteDB에 query문을 전송	id	LiteDBStatement00

TypeDefinition > Objects 더블클릭

TypeDefinition - Objects > DeviceAPI.json 선택

nexacro.LiteDBConnection, nexacro.LiteDBStatement 체크

2. LiteDBConnection, LiteDBStatement 오브젝트 추가하기

The screenshot shows the Nexacro IDE interface. At the top, a red box highlights the 'Invisible Object' icon in the toolbar. A red arrow points from this icon to the 'Invisible Object' panel at the bottom left. In this panel, 'LiteDBConnection00' and 'LiteDBStatement00' are listed and highlighted with a red box. The main workspace displays a form titled 'Form_Emp.xfdl' with a table of employee data and various input fields for details like Name, Emp ID, Department, Position, Gender, Married, Hire Date, and Annual Salary. The 'Binding Components List - ds_emp' panel on the right shows the data source for the form.

No	FULL	DEPT	EMPL	POS	HIRE	SALA	GEND	MARR	MEM
1	Olivia	01	AA001	02	2010-1	83,000	Female	true	ivory
2	John	02	AA002	02	2005-1	75,000	Male	false	greeny
3	Jacksc	02	BB001	03	2007-0	65,000	Male	true	aliceb
4	Maia	04	BB002	01	2009-0	60,000	Female	false	ivory
5	Adam	04	CC001	04	2001-0	88,000	Male	true	greeny
6	Tyler	04	DD001	03	2006-0	60,000	Male	false	lightgr

LiteDBConnection, LiteDBStatement 컴포넌트를 Invisible Object 항목에 추가한다.

3. LiteDBConnection 컴포넌트의 메소드

- 애플리케이션에서 데이터베이스를 연결, 종료할 때는 LiteDBConnection 오브젝트를 사용합니다.
- 지금부터는 데이터베이스에 연결하고 데이터베이스의 tb_emp 테이블을 연결하여 grd_list로 출력하는 작업을 진행하겠습니다.

LiteDBConnection > open 메소드

- 데이터베이스를 연결하는 메소드입니다. 첫 번째 인자로 SQLite 파일의 위치와 파일명을 입력하고, 두 번째 인자로 데이터베이스를 어떻게 오픈할지 설정합니다.

LiteDBConnection > datasource 메소드

- 연결할 데이터베이스의 위치 정보를 설정하는 속성입니다. 데이터베이스 경로와 파일명을 같이 입력합니다.

LiteDBConnection > close 메소드

- 데이터베이스 연결을 닫는 메소드입니다.

LiteDBConnection > isConnected 메소드

- 데이터베이스가 연결되어 있는지 확인하는 메소드입니다.

LiteDBErrorEventInfo 오브젝트

- LiteDBConnection, LiteDBStatement 오브젝트의 수행에 오류가 생겼을 때 onerror 이벤트 함수로 전달되는 EventInfo 오브젝트입니다.

LiteDBEventInfo 오브젝트

- LiteDBConnection, LiteDBStatement 오브젝트의 작업 수행이 성공했을 때 onsuccess 이벤트 함수로 전달되는 EventInfo 오브젝트 입니다.

4. 이벤트 작성하기

Open 버튼 이벤트 함수 작성하기

Employees: C:\JAVAW\sqlite_LiteDBW\nexacro_sample_db.sqlite Retrive **open** add del save

Department: Find Gender: ☐ All ☐ Male ☐ Female Change reset

No	FULL	DEPT	EMPL	POS_	HIRE_	SALA	GEND	MARR	MEM
1	Olivia	01	AA001	02	2010-1	83,000	Female	true	ivory
2	John	02	AA002	02	2005-1	76,000	Male	false	greeny
3	Jackso	02	BB001	03	2007-0	95,000	Male	true	aliceb
4	Maia	04	BB002	01	2009-0	60,000	Female	false	ivory
5	Adam	04	CC001	04	2001-0	88,000	Male	true	greeny
6	Tyler	04	DD001	03	2006-0	60,000	Male	false	lightgr

Name:

Emp ID:

Department:

Position:
Officer
Assistant Manager
Division Manager
Chaiman

Gender: ☐ Male ☐ Female

Married: ☐

Hire Date:

Annual Salary:

Properties

btn_dbopen (Button)

Input filter text

Event

onclick	btn_dbopen onclick
oncontextmenu	
ondblclick	
ondrag	
ondragenter	
ondragleave	
ondragmove	
ondrop	
onkeydown	
onkeyup	

- Open 버튼을 선택한 후 Properties 창에서 btn_dbopen 컴포넌트의 onclick 이벤트의 btn_dbopen onclick 이벤트를 선택한다.

4. 이벤트 작성하기

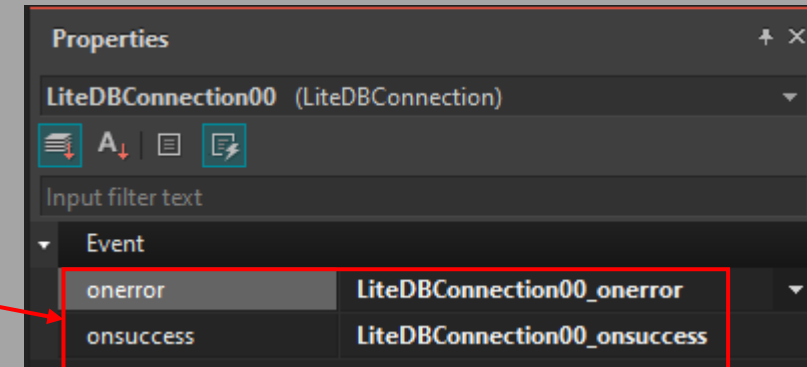
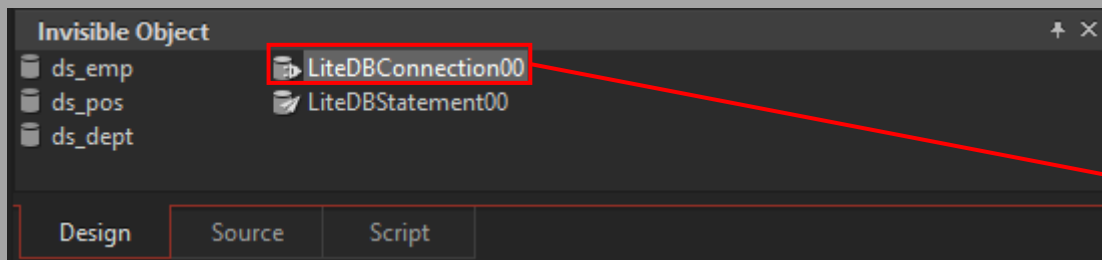
Open 버튼 이벤트 함수 작성하기

```
this.btn_dbopen_onclick = function(obj:nexacro.Button,e:nexacro.ClickEventInfo)
{
    this.LiteDBConnection00.isConnected();
};
```

btn_dbopen_onclick 이벤트 함수를 동작시켜서 LiteDBConnection00 컴포넌트에 isConnected() 메서드를 실행시켜 데이터베이스와 Connection 되었는지 확인하다.

4. 이벤트 작성하기

LiteDBConnection00 onerror 이벤트와 onsuccess 이벤트 작성하기



Form_Emp 폼 하단의 Invisible Object 창에서 LiteDBConnection00 컴포넌트를 선택하여 Properties 창에서 onerror 이벤트와 onsuccess 이벤트를 발생시킨다.
- onerror 이벤트는 LiteDBConnection00_onerror 이벤트를 onsuccess 이벤트는 LiteDBConnection00_onsuccess 이벤트를 선택하여 코드를 작성한다.

4. 이벤트 작성하기

LiteDBConnection00_onerror 이벤트의 코드 작성하기

```
this.LiteDBConnection00_onerror = function(obj:nexacro.LiteDBConnection,e:nexacro.LiteDBErrorEventInfo)
{
    //e.statuscode.toString() 메서드를 확인하여 어떤코드가 발생되는지 확인한다.
    this.alert("e.statuscode.toString() : " + e.statuscode.toString());
    switch(e.statuscode.toString()){
        case "1203":
            alert("DB close failed.");
            break;
        case "1205":
            this.LiteDBConnection00.set_datasource(this.edt_dbfile.value);
            this.LiteDBConnection00.open();
            break;
        case "1206":
            alert("DB open failed.");
        case "00001":
        case "1201":
        case "1202":
        case "1204":
        case "1207":
        default:
            var strResult = "[LiteDBConnection00_onerror]";
            strResult += "\n[" + e.errortype + " " + e.statuscode + "] " + e.errormsg;
            strResult += "\ne.ldberrorcode: " + e.ldberrorcode;
            strResult += "\ne.ldberrormsg: " + e.ldberrormsg;
            alert(strResult);
    }
};
```

4. 이벤트 작성하기

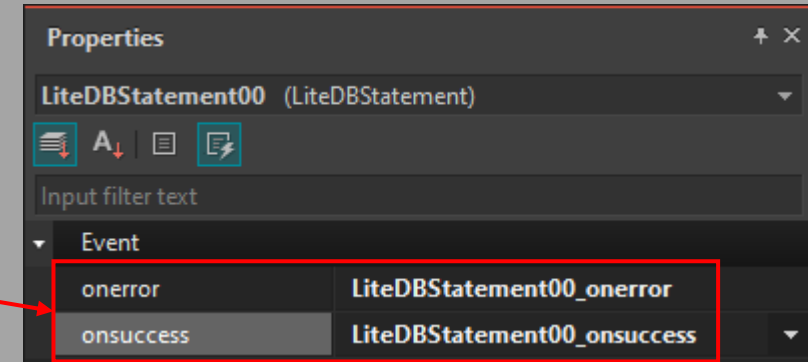
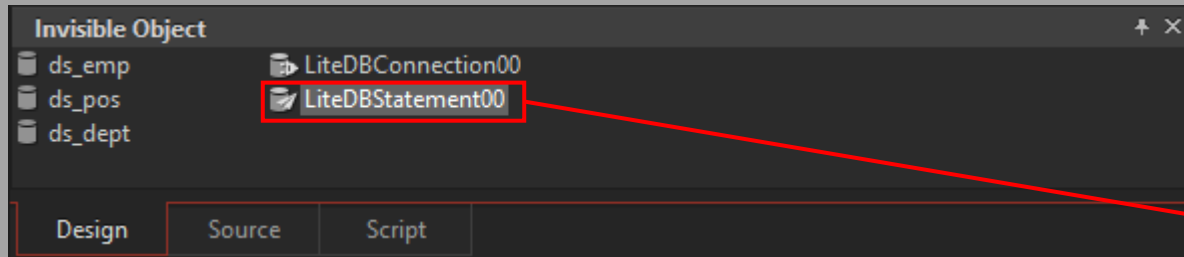
LiteDBConnection00_onsuccess 이벤트의 코드 작성하기

```
this.LiteDBConnection00_onsuccess = function(obj:nexacro.LiteDBConnection,e:nexacro.LiteDBEventInfo)
{
    alert("e.reason : " + e.reason);
    switch(e.reason){
        case 2: //LiteDBConnection close()
            this.grd_list.set_visible(false);
            alert("DB close succeed.");
            break;
        case 4: //LiteDBConnection isConnected()
            alert("Already connected to DB.");
            break;
        case 5: //LiteDBConnection open()
            this.grd_list.set_visible(true);

            this.LiteDBStatement00.set_ldbconnection(this.LiteDBConnection00);
            this.LiteDBStatement00.executeQuery("select * from tb_emp");
            break;
        case 1:
        case 3:
        case 6:
        default:
            break;
    }
};
```

4. 이벤트 작성하기

LiteDBStatement00 onerror 이벤트와 onsuccess 이벤트 작성하기



Form_Emp 폼 하단의 Invisible Object 창에서 LiteDBStatement00 컴포넌트를 선택하여 Properties 창에서 onerror 이벤트와 onsuccess 이벤트를 발생시킨다.
- onerror 이벤트는 LiteDBStatement00_onerror 이벤트를 onsuccess 이벤트는 LiteDBStatement00_onsuccess 이벤트를 선택하여 코드를 작성한다.

4. 이벤트 작성하기

LiteDBStatement00_onerror 이벤트의 코드 작성하기

```
this.LiteDBStatement00_onerror = function(obj:nexacro.LiteDBStatement,e:nexacro.LiteDBErrorEventInfo)
{
    switch(e.statuscode.toString())
    {
        case "1210":    //LiteDBStatement executeQuery() error
            alert("LiteDBStatement executeQuery() error");
            break;
        case "1211":    //LiteDBStatement executeUpdate() error
            alert("LiteDBStatement executeUpdate() error");
            break;
        case "00001":    //Parameter setting error
        case "1201":    //DB response timeout error
        default:        //Unknown error
            var strResult = "[LiteDBStatement00_onerror]";
            strResult += "\n[" + e.errortype + " " + e.statuscode + "] " + e.errormsg;
            strResult += "\ne.ldberrorcode: " + e.ldberrorcode;
            strResult += "\ne.ldberrormsg: " + e.ldberrormsg;
            alert(strResult);
    }
};
```

4. 이벤트 작성하기

LiteDBStatement00_onsuccess 이벤트의 코드 작성하기

-연결된 데이터베이스에 쿼리 수행이 성공하면 발생할 onsucces 이벤트 함수를 작성합니다. 쿼리문의 결과 값으로 받은 데이터를 Dataset으로 받아 Grid로 출력합니다.

```
this.LiteDBStatement00_onsuccess = function(obj:nexacro.LiteDBStatement,e:nexacro.LiteDBEventInfo)
{
    this.ds_emp.copyData(e.returnValue);
    this.grd_list.createFormat();
};
```

Nexacro-DB연동

데이터베이스 연결 하기

5. open 버튼으로 DB 연동하기

화면상단에 open 버튼을 클릭하면 grd_list grid가 LiteDB의 tb_emp 테이블을 연동하여 값이 변경되는 것을 확인할 수 있습니다.

TopLeftFrame - New Form

Employees: C:\JAVAW\sqlite_LiteDB\nexacro_sample_db.sqlite

Retrive open add del save

Department: Find Gender: ☐ All ☐ Male ☐ Female Change reset

No	FULL	DEPT	EMPL	POS	HIRE	SALA	GEND	MARR	MEMO
1	Olivia	01	AA001	02	2010-1	83,000	Female	true	ivory
2	John	02	AA002	02	2005-1	76,000	Male	false	green
3	Jacksc	02	BB001	03	2007-0	95,000	Male	true	aliceb
4	Maia	04	BB002	01	2009-0	60,000	Female	false	ivory
5	Adam	04	CC001	04	2001-0	88,000	Male	true	green
6	Tyler	04	DD001	03	2006-0	60,000	Male	false	lightgr

Name: Olivia

Emp ID: AA-001

Department: Accounting Team

Position: Assistant Manager **Division Manager** Chairman

Gender: ☐ Male ☒ Female

Married: ☒

Hire Date: 2010-10-03 일

Annual Salary: 83,000

Memo: ivory

Total: 462000

TopLeftFrame - New Form

Employees: C:\JAVAW\sqlite_LiteDB\nexacro_sample_db.sqlite

Retrive open add del save

Department: Find Gender: ☐ All ☐ Male ☐ Female Change reset

EMPL	FULL	DEPT	POS	HIRE	GEND	MARR	SALA	MEMO
AA001	Oleg	08	03	2007250	W	1	70,000	green
AA010	Aladdir	07	06	2004190	M	0	120,000	green
BB010	Curran	02	03	2007011	M	1	210,000	orange
BB020	Adam	07	07	2001160	W	1	90,000	indigo
BB030	Heathe	03	03	2006140	M	0	50,000	blue
CC110	Lester	07	05	2017160	M	0	100,000	green
CD120	Elijah	10	01	2015130	W	0	110,000	gold
AB100	Angela	08	04	2002111	M	1	670,000	blue
AC310	Rae	02	05	2007260	W	1	90,000	green
AC210	Denton	02	04	2017250	M	1	90,000	skyblu
DD002	Nora	05	01	2016020	W	0	420,000	blue
DD200	Adrian	05	04	2016021	W	0	80,000	violet
AD020	Ulric	04	04	2004210	W	0	420,000	yellow
BD030	Veronic	10	06	2013070	M	1	70,000	green
AA200	Phyllis	02	06	2012081	W	1	350,000	orange

Name: Oleg

Emp ID: AA-001

Department: Assistant Manager **Division Manager** Chairman

Position: Assistant Manager **Division Manager** Chairman

Gender: ☐ Male ☒ Female

Married: ☒

Hire Date: 2007-25-04 일

Annual Salary: 70,000

Memo: green

5. add 버튼으로 데이터 tb_emp 테이블에 입력하기

화면 상단의 add버튼을 click 하여 값입력한 후 save버튼을 click 하여 tb_emp 테이블에 저장합니다.

Main_Menu1
Main_Menu2

Employees
C:\JAVASqlite_LiteDB\nexacro_sample_db.sqlite
Retrive
ope
1
add
del
4
save

Department
Find
Gender
All
Male
Female
Change
reset

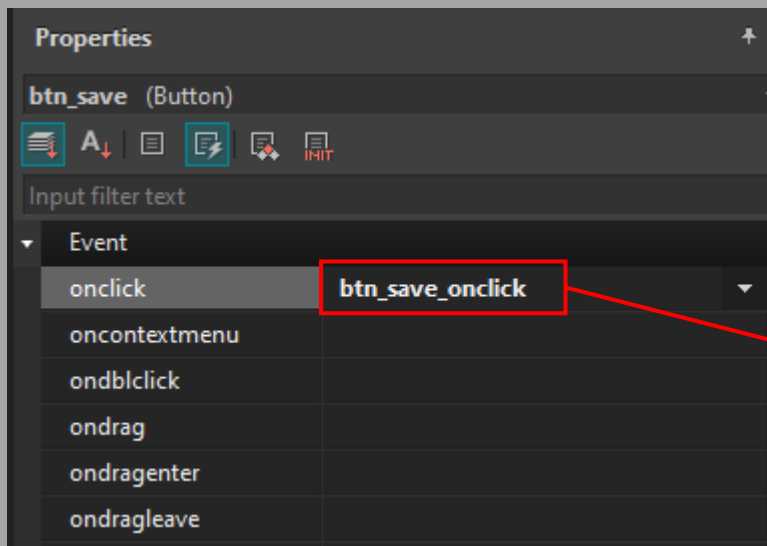
EMPL	FULL	DEPT	POS	HIRES	GEND	MARR	SALAF	MEMO
AA010	Aladdir	07	06	200419C	M	0	120,000	green
BB010	Curran	02	03	2007011	M	1	210,000	orange
BB020	Adam	07	07	200116C	W	1	90,000	indigo
BB030	Heathe	03	03	200614C	M	0	50,000	blue
CC110	Lester	07	05	201716C	M	0	100,000	green
CD120	Elijah	10	01	201513C	W	0	110,000	gold
AB100	Angela	08	04	2002111	M	1	670,000	blue
AC310	Rae	02	05	200726C	W	1	90,000	green
AC210	Denton	02	04	201725C	M	1	90,000	skyblue
DD002	Nora	05	01	201602C	W	0	420,000	blue
DD200	Adrian	05	04	2016021	W	0	80,000	violet
AD020	Ulric	04	04	200421C	W	0	420,000	yellow
BD030	Veronic	10	06	201307C	M	1	70,000	green
AA200	Phyllis	02	06	2012081	W	1	350,000	orange
AA300	phh	03	02	202303C	M	1	85,000	green
AA400	pjh	03	02	202303C	M	0	80,000	yello
AA500	phj	04	03	202303C	M	1	60,000	black
BB040	lee	04	02	202303C	M	0	50,000	green

3
Name
lee
Emp ID
BB-040
Department
Design Team
Position
Officer
Assistant Manager
Division Manager
Chairman
Gender
Male
Female
Married
Hire Date
2023-03-30
Annual Salary
50,000
Memo
green

1. add 버튼 click하면 grd_list하단에 1행이 추가됩니다.
2. Name 부터 Memo까지 순서대로 입력한 후 save 버튼을 click하여 데이터베이스에 값을 저장합니다.

6. save 버튼 btn_save_onclick 이벤트 작성

화면 상단의 add버튼을 click 하여 값입력한 후 save버튼을 click 하여 tb_emp 테이블에 저장합니다.



```
this.btn_save_onclick = function(obj:nexacro.Button,e:nexacro.ClickEventInfo)
{
    var marriey;
    //tb_emp 테이블의 MARRIED 필드에 입력시 1 또는 0으로 입력해야함.
    if (this.div_detail.form.chk_married.value == true){
        marriey = 1;
    }else{
        marriey = 0;
    }
    var strQuery = "insert into tb_emp values('" + this.div_detail.form.msk_id.value + "','";
    strQuery += this.div_detail.form.edt_name.value + "','";
    strQuery += this.div_detail.form.cbo_dept.value + "','";
    strQuery += this.div_detail.form.lst_pos.value + "','";
    strQuery += this.div_detail.form.cal_date.value + "','";
    strQuery += this.div_detail.form.rdo_gender.value + "','";
    strQuery += marriey + "','";
    strQuery += this.div_detail.form.msk_salary.value + "','";
    strQuery += this.div_detail.form.txt_memo.value + "')";
    this.fn_LiteDBStatement_query(strQuery);
};
```

btn_save 버튼의 onclick 이벤트에 btn_save_onclick 이벤트를 작성해 준다.

6. save 버튼 btn_save_onclick 이벤트 작성

fn_LiteDBStatement_query 메서드를 작성한다.

```
this.fn_LiteDBStatement_query = function(strQuery){
    this.LiteDBStatement00.set_query(strQuery);
    var arrString = this.LiteDBStatement00.query.split(" ");
    switch(arrString[0]){
        case "select":
            this.LiteDBStatement00.executeQuery();
            break;
        case "insert":
        case "update":
        case "delete":
            this.LiteDBStatement00.executeUpdate();
            this.alert("DB " + arrString[0] + " success!!");
            break;
        default:
            alert(arrString[0] + "is illegal query statement.");
    }
};
```

fn_LiteDBStatement_query 메서드는 insert, update, delete 문 실행시 공동으로 사용되는 메서드 이다.

6. btn_del 버튼 click시 tb_emp 테이블에 해당 데이터 삭제하기

화면 아래의 grd_list에서 삭제할 데이터를 선택하고 btn_del 버튼을 click한다.

Main_Menu1Main_Menu2

EmployeesC:\JAVAW\sqlite_LiteDB\W\nexacro_sample_db.sqliteRetriveopenadddelsave

DepartmentFindGenderAllMaleFemaleChangereset

EMPL	FULL	DEPT	POS	HIRE	GEND	MARR	SALAF	MEMO
AA010	Aladdin	07	06	2004190	M	0	120,000	green
BB010	Curran	02	03	2007011	M	1	210,000	orange
BB020	Adam	07	07	2001160	W	1	90,000	indigo
BB030	Heathe	03	03	2006140	M	0	50,000	blue
CC110	Lester	07	05	2017160	M	0	100,000	green
CD120	Elijah	10	01	2015130	W	0	110,000	gold
AB100	Angela	08	04	2002111	M	1	670,000	blue
AC310	Rae	02	05	2007260	W	1	90,000	green
AC210	Denton	02	04	2017250	M	1	90,000	skyblue
DD002	Nora	05	01	2016020	W	0	420,000	blue
DD200	Adrian	05	04	2016021	W	0	80,000	violet
AD020	Ulric	04	04	2004210	W	0	420,000	yellow
BD030	Veronic	10	06	2013070	M	1	70,000	green
AA200	Phyllis	02	06	2012081	W	1	350,000	orange
AA500	phj	04	03	2023033	M	1	60,000	black
DD300	lee	05	02	2023033	M	1	80,000	orange

NameDenton

Emp IDAC-210

DepartmentHR Team▼

OfficerAssistant ManagerDivision ManagerChairman

GenderMaleFemale

Married☒

Hire Date2017-25-03

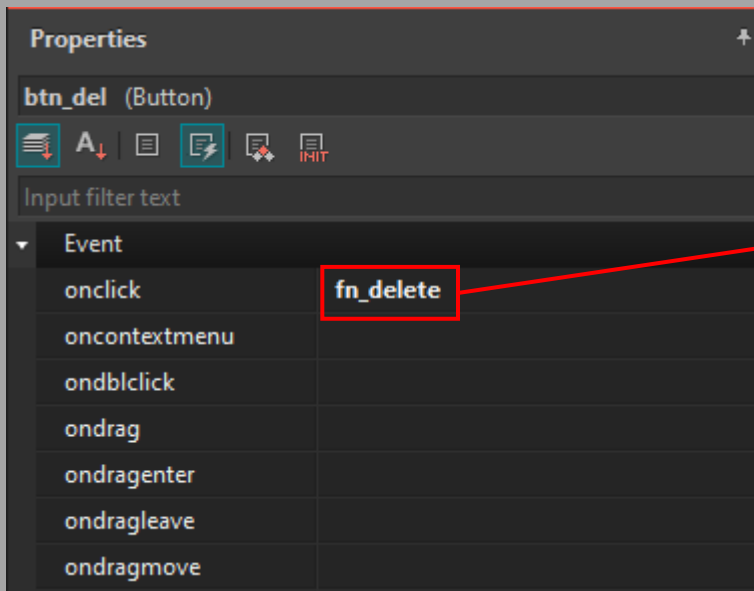
Annual Salary90,000

skyblue

Memo

6. btn_del 버튼 click시 tb_emp 테이블에 해당 데이터 삭제하기

btn_del 버튼의 onclick 이벤트에 fn_delete 메서드를 작성한다.



```
this.fn_delete = function(obj:nexacro.Button,e:nexacro.ClickEventInfo)
{
    var empId = this.div_detail.form.msk_id.value;
    var strQuery = "delete from tb_emp where empl_id = '" + empId + "'";

    // 선택되어 있는 row를 삭제
    this.ds_emp.deleteRow(this.ds_emp.rowposition);
    this.fn_LiteDBStatement_query(strQuery);
};
```

fn_delete 이벤트 메서드의 하단의 this.fn_LiteDBStatement_query(strQuery); 는 insert, update, delete 문 실행시 공동으로 사용되는 메서드이다.

감사합니다.

- Fin -