데이터베이스 6번 과제

14010995 김창학

목차

1. 최종 요구사항 분석

2. 최종 ER-Diagram

3. 최종 DB Schema(ER-relation Mapping)

4. 최종 SQL 캡쳐 화면

5. 프로그램의 구현 단계(Implementation Level)

1. 최종 요구사항 분석.

- LANGUAGE DATA Entity는 자연어와 자연어를 처리한 데이터를 가지는 attribute이다. 모든 LANGUAGE DATA는 DATA MINER를 무조건 한 명만 가질 수 있고, DATA MANAGER를 한 명까지 가질 수 있다. DATA MINER와 DATA MANAGER는 한 개 이상의 LANGUAGE DATA를 가질 수 있다. LANGUAGE DATA는 한 개 이상의 프로젝트에 들어갈 수 있다. 한 개의 LANGUAGE DATA는 여러 개의 SUBJECT를 가질 수 있다.

- DATASET SIMILIARITY Relation은 두 LANGUAGE DATA가 얼마나 관련이 있는지 저장한다. 각 LANGUAGE DATA는 자기보다 큰 LANGUAGE DATA의 관계를 모두 가지고 있어야 한다.(X개의 데이터 셋이 있다면 X!개의 관계를 유지하고 있어야 함) -> 본인과 본인은 무조건 100%

- DATA MINER Entity는 데이터를 수집해 LANGUAGE DATA에 저장한다. 모든 DATA MINER는 DATA MANAGER를 한 명까지 가진다. DATA MINER의 KEY는 MANAGER의 KEY와 결합해야 KEY의 역할을 수행할 수 있다. SUBJECT FIELD는 한 개 이상 설정할 수 있다.

- DATA MANAGER Entity는 DATA MINER를 관리하면서 동시에 LANGUAGE DATA의 오류를 확인하고 데이터의 일관성이 유지되도록 관리한다. DATA MANAGER는 최대 10명의 DATA MINER를 관리할 수 있다.

- PROJECT Entity는 ANALIST가 진행하는 프로젝트를 저장한 곳이다. 모든 PROJECT는 DATA ANALIST를 무조건 한 명만 가진다. 한 프로젝트는 여러 개의 LANGUAGE DATA를 가질 수 있다.

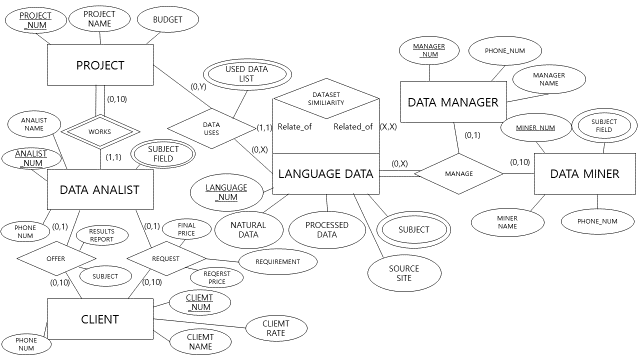
- DATA USES Relation은 프로젝트가 사용하는 LANGUAGE DATA의 관계이다.

- DATA ANALIST Entity는 PROJECT를 통해서 데이터를 이용한 결과를 도출해 CLIENT에게 제공한다. DATA ANALIST는 최대 10개의 프로젝트와 고객을 동시에 가질 수 있다. ANALIST는 여러 개의 SUBJECT FIELD(전문 분야)를 가질 수 있다.

- CLIENT Entity는 작업을 의뢰하는 고객이다. 모든 CLIENT는 DATA ANALIST를 여러 명 지정할 수 있다.

- OFFER과 REQUEST Relation는 CLIENT와 ANALIST의 관계와 행위를 설명한다.

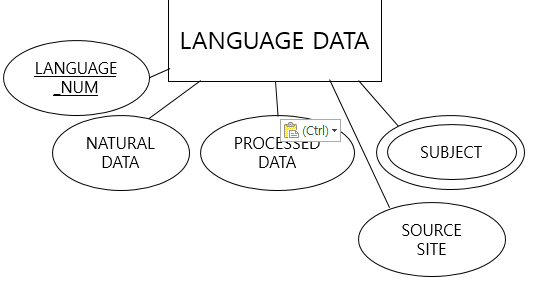
2. 최종 ER - Diagram

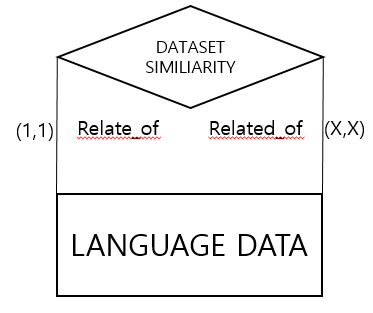


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| LANGUAGE\_NUM | NATURAL DATA | PROCEED DATA | SUBJECT | SOURCE SITE | DATA\_MINER  NO | DATA\_MANAGERNO |
| 1 | “뭐 이 자식아” | [0.5,1,1,1,0.8] | 일상 데이터 | http://... | 1 | 1 |
| 2 | “의왕” | [1,1,0,0,0.4] | 지역 데이터 | http://... | 2 | 2 |
| 3 | “안녕하세요” | [4,5,2,1,0] | 대화 데이터 | http://... | 3 | 1 |
| 4 | “반갑습니다” | [5,5,5,0,0] | 대화 데이터 | http://... | 4 | 2 |

3. 최종 DB Schema

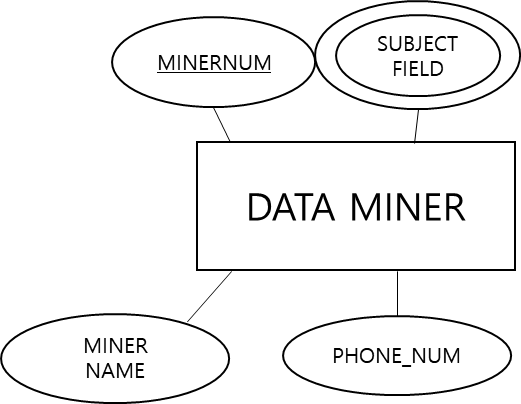
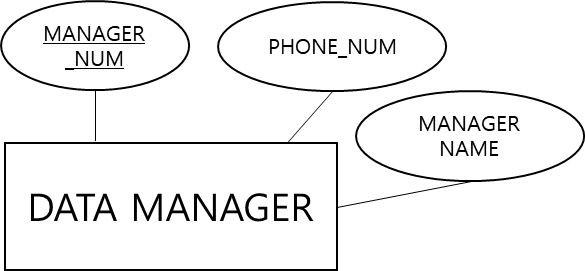
- LANGUAGE DATA(Entity)



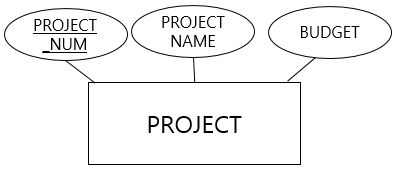
* DATASET SIMILARITY(Relation)

|  |  |  |
| --- | --- | --- |
| LANGUAGE\_NUM1 | LANGUAGE\_NUM2 | RELATED SCORE |
| 1 | 1 | 100.0% |
| 1 | 2 | 56.3% |
| 1 | 3 | 33.2% |
| 1 | 3 | 33.2% |
| 2 | 1 | 56.3% |
| 2 | 2 | 100.0% |
| 2 | 3 | 67.7% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MINER\_NUM | MINER NAME | PHONE\_NUM | SUBJECT\_FIELD | DATA\_MANAGER NO |
| 1 | KIM | 010-0000-0000 | 일상, 대화 | 1 |
| 2 | LEE | 010-1234-1234 | 강연 | 1 |
| 3 | PART | 010-1234-5678 | 과학, 공학 | 2 |
| 4 | CHOI | 010-2222-3333 | 사학, 문학 | 2 |

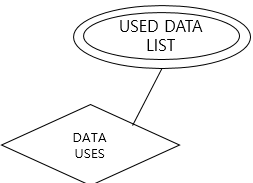
* DATA MINER(Entity)
* DATA MANAGER(Entity)

|  |  |  |
| --- | --- | --- |
| MANAGER\_NUM | MANAGER NAME | PHONE\_NUM |
| 1 | KIM | 010-0000-0000 |
| 2 | LEE | 010-1234-1234 |
| 3 | CHOI | 010-1234-5678 |
| 4 | PARK | 010-2222-3333 |

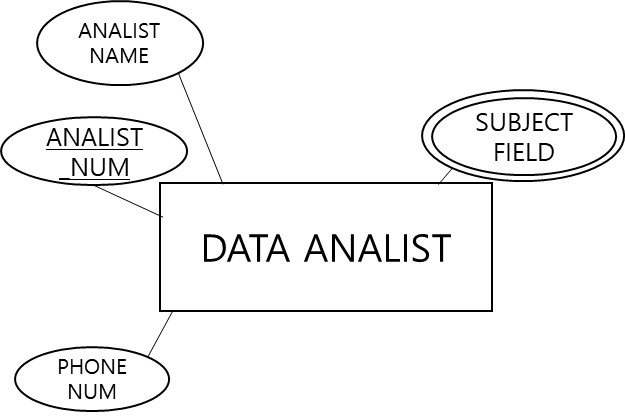
* MANAGE (Relation)
* PROJECT(Entity)

|  |  |  |  |
| --- | --- | --- | --- |
| ANALIST\_NO | PROJECT\_NUM | PROJECT\_NAME | BUDGET |
| 1 | 1 | 연구1 | 1,000,000 |
| `1 | 2 | 연구2 | 2,000,000 |
| 2 | 1 | 활용1 | 100,000 |
| 2 | 2 | 활용2 | 150,000 |

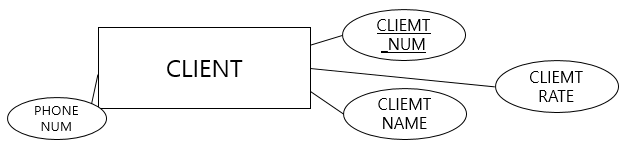
|  |  |
| --- | --- |
| PROJECT\_NO | LANGUAGEDATA\_NO |
| 1 | 1 |
| 1 | 2 |
| 2 | 3 |
| 2 | 4 |

* DATA USES(Relation)
* WORKS(Relation)

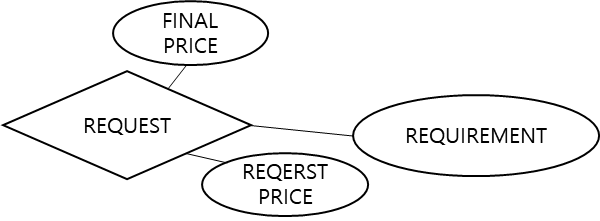
WORKS

* DATA ANALIST(Entity)

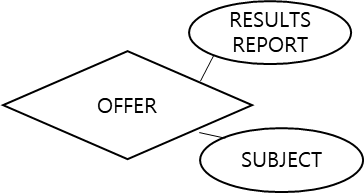
|  |  |  |  |
| --- | --- | --- | --- |
| ANALIST\_NUM | ANALIST NAME | PHONE\_NUM | SUBJECT\_FIELD |
| 1 | KIM | 010-0000-0000 | 일상, 대화 |
| 2 | LEE | 010-1234-1234 | 강연 |
| 3 | CHOI | 010-1234-5678 | 과학, 공학 |
| 4 | PARK | 010-2222-3333 | 사학, 문학 |

* CLIENT(Entity)

|  |  |  |  |
| --- | --- | --- | --- |
| CLIENT\_NUM | CLIENT\_NAME | PHONE\_NUM | CLIENT\_RATE |
| 1 | KIM | 010-0000-0000 | A |
| 2 | LEE | 010-1234-1234 | A |
| 3 | CHOI | 010-1234-5678 | B |
| 4 | PARK | 010-2222-3333 | C |

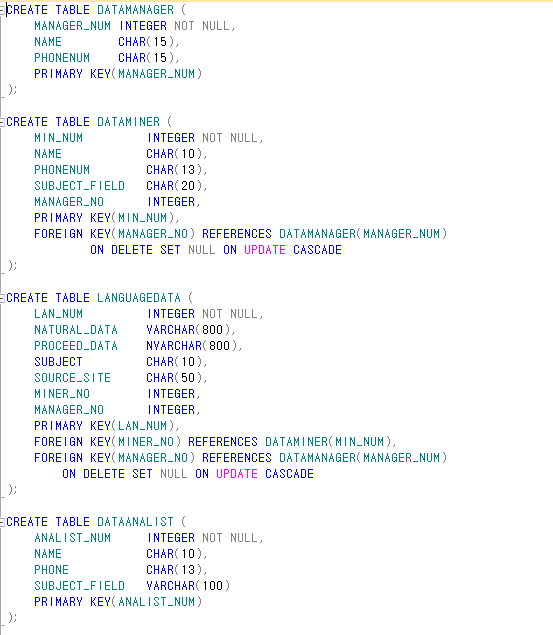
* REQUEST(Relation)

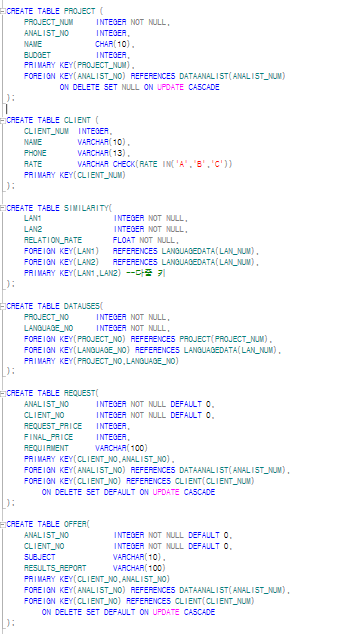
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLIENT\_NUM | ANALIST\_NUM | REQUEST PRICE | FINAL PRICE | REQUIRMENT |
| 1 | 1 | 700,000 | 1,000,000 |  |
| 2 | 1 | 1,000,000 | 2,000,000 |  |
| 3 | 2 | 50,000 | 100,000 |  |
| 4 | 2 | 80,000 | 150,000 |  |

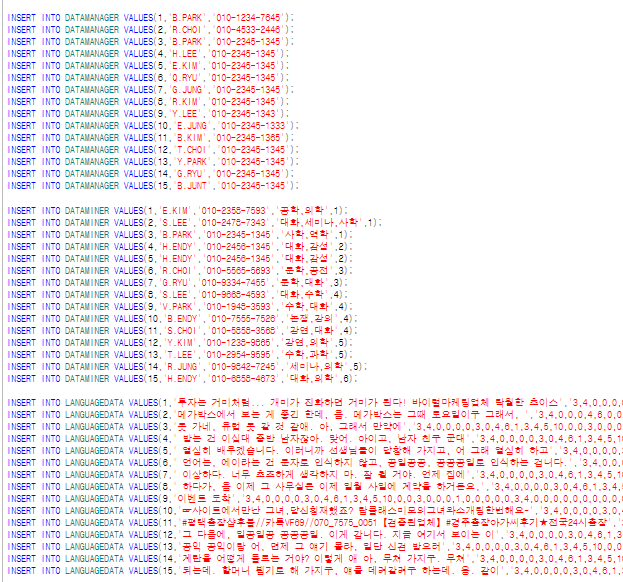
* OFFER(Relation)

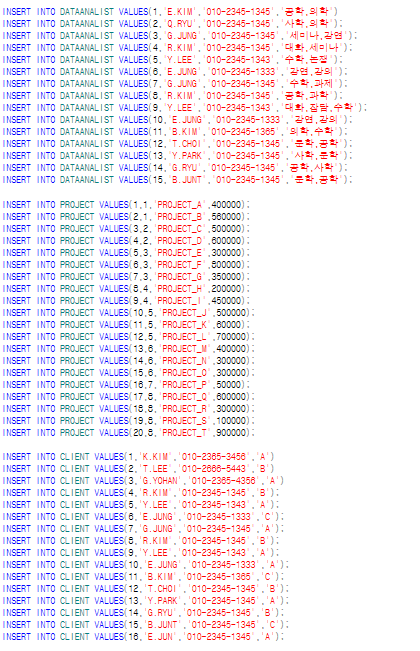
|  |  |  |  |
| --- | --- | --- | --- |
| CLIENT\_NUM | ANALIST\_NUM | SUBJECT | RESULTS REPORT |
| 1 | 1 | 회화 | A.docx |
| 2 | 1 | 강연 | B.docx |
| 3 | 2 | 교습 | C.docx |
| 4 | 2 | 과학 | D.docx |

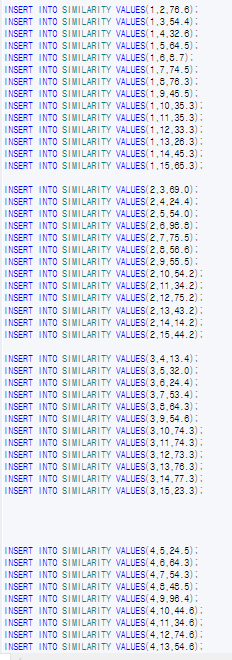
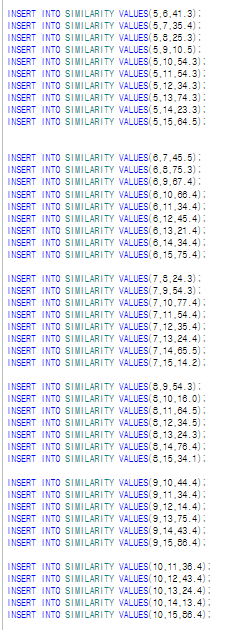
4. 최종 SQL 캡쳐 화면

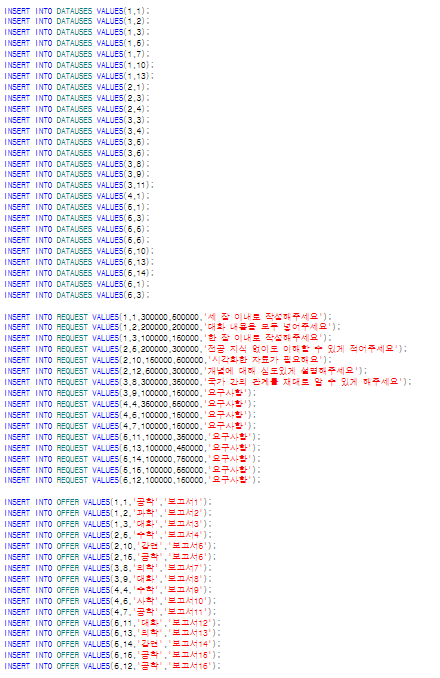








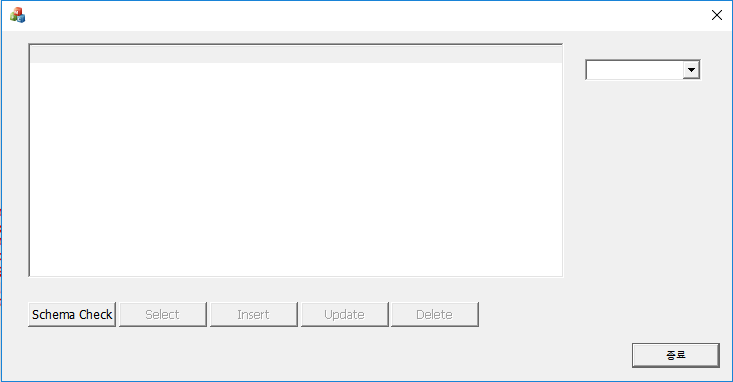
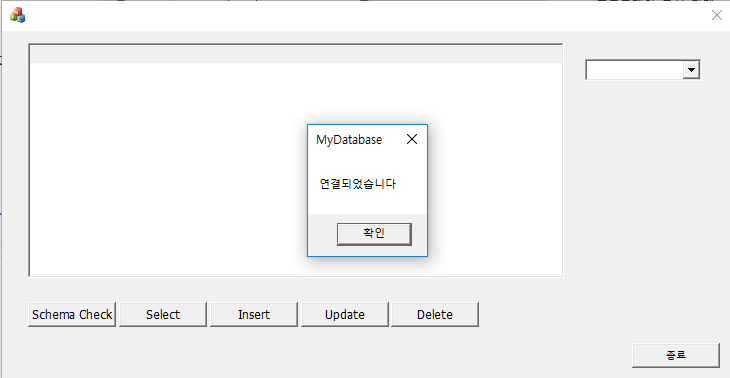




5. 프로그램의 구현 단계

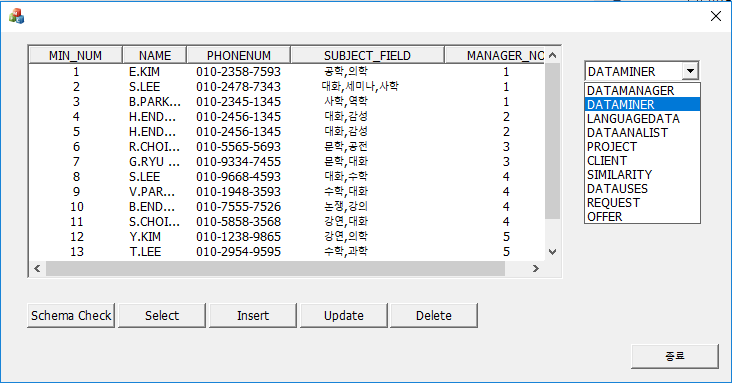
|  |  |
| --- | --- |
| 기능 | 구현 단계 |
| Schema Check | 구현됨 |
| Select Query | 6개 구현됨(2,3,4중 Join Table 구현됨) |
| Insert Query | 4개 구현됨 |
| Delete Query | 4개 구현됨. |
| Update Query | 4개 구현됨 |

7. 프로그램 실행 캡쳐 화면



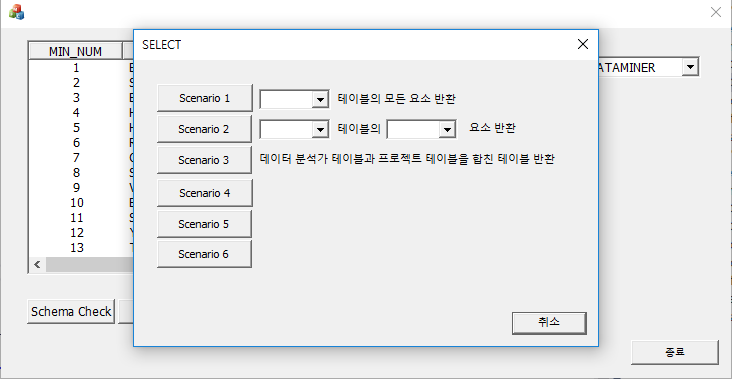
사용한 쿼리 - SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES

Schema Check를 눌렀을 때



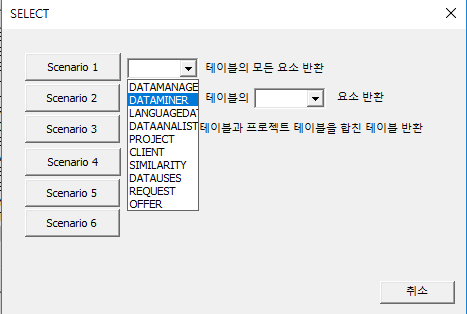
(SELECT \* FROM %s, table)

오른쪽 콤보박스에 테이블명이 나오고. 누르면 테이블의 정보를 보여준다.

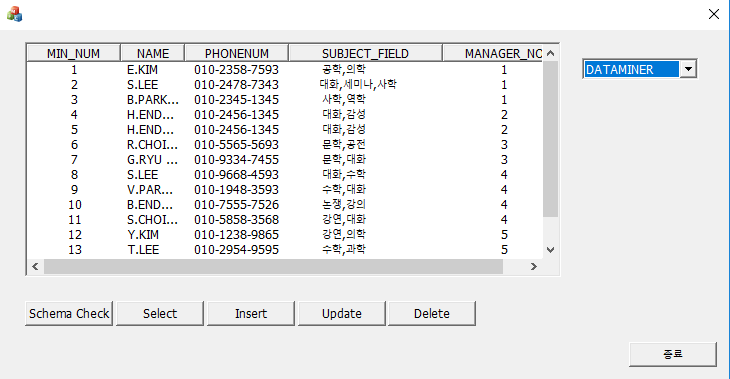


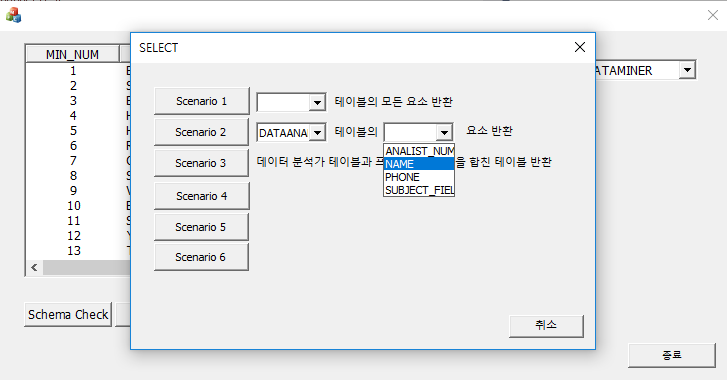
SELECT QUERY

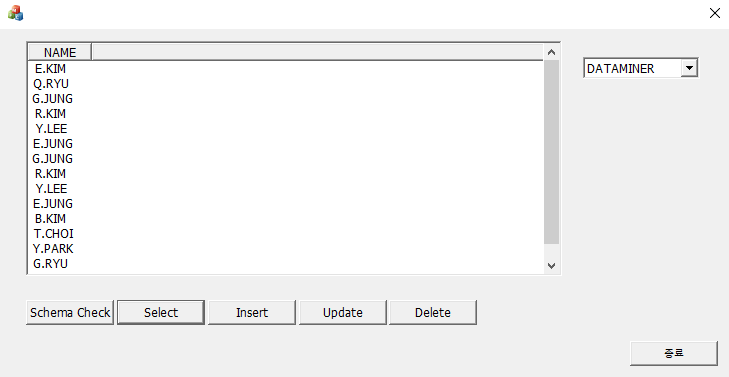
Senario1 – SELECT \* FROM %s, Table



메인 메뉴에 나온다.

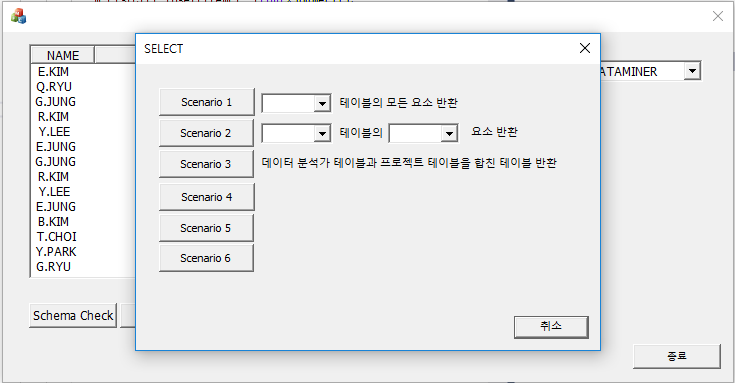


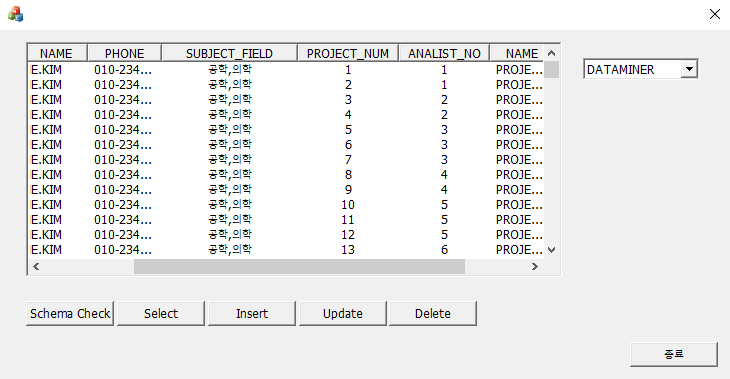
Senario2 – 각 테이블의 Column중 하나를 선택한다. - SELECT %s FROM %s, Column, Table



Senario 3 – DATAANALIST와 PROJECT를 Join한 테이블 반환.

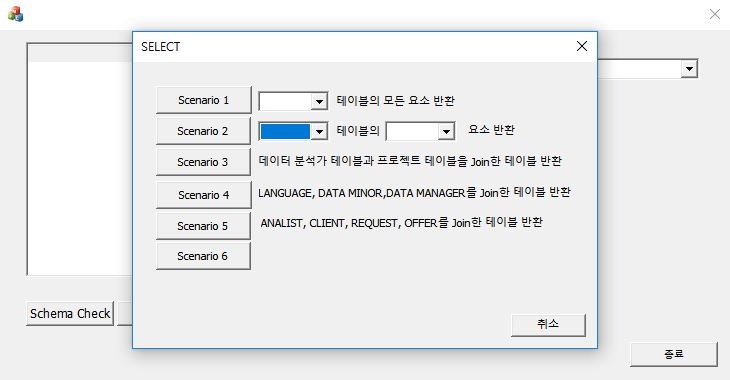
SELECT \* FROM DATAANALIST AS D, PROJECT AS P

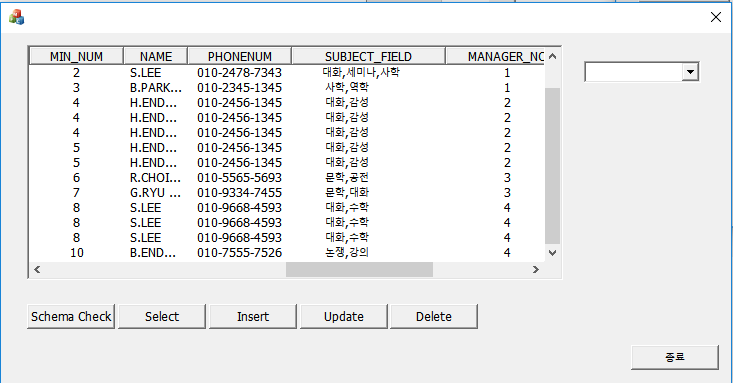




Senario 4 – LANGUAGE,DATA MINOR, DATA MANAGER를 JOIN한 테이블 반환

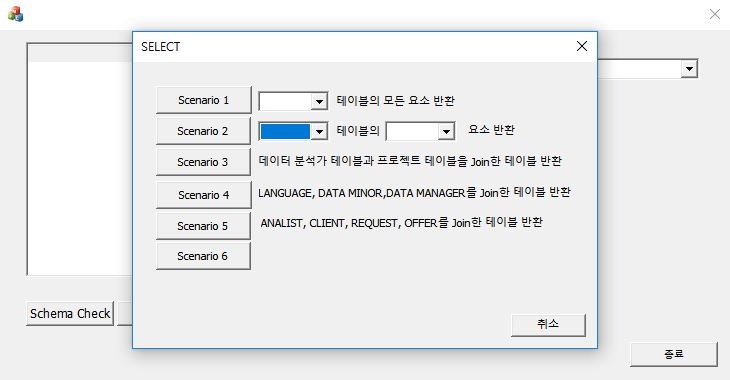
SELECT \* FROM DATAANALIST AS D, PROJECT AS P WHERE D.ANALIST\_NUM = P.ANALIST\_NO

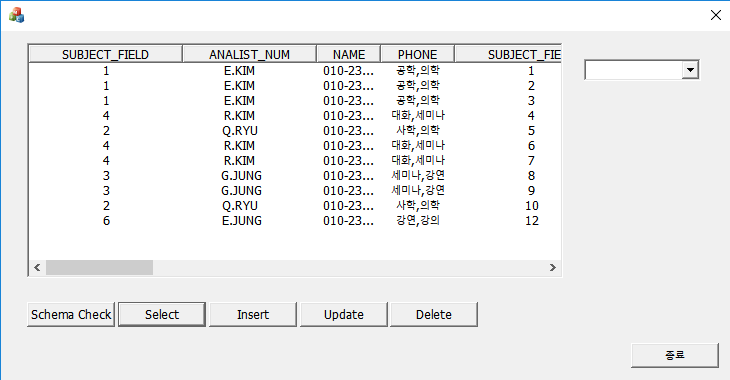




Senario 5 – DATA ANALIST, CLIENT, REQUEST, OFFER를 JOIN한 TABLE 반환

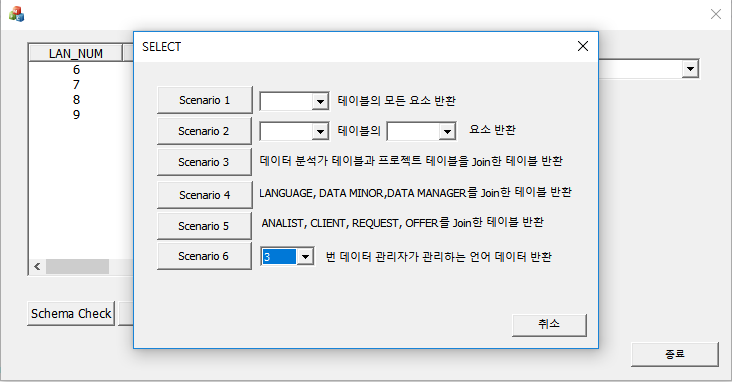
SELECT \* FROM LANGUAGEDATA AS L, DATAMINER AS M, DATAMANAGER AS R WHERE L.MANAGER\_NO = R.MANAGER\_NUM AND L.MINER\_NO = M.MIN\_NUM

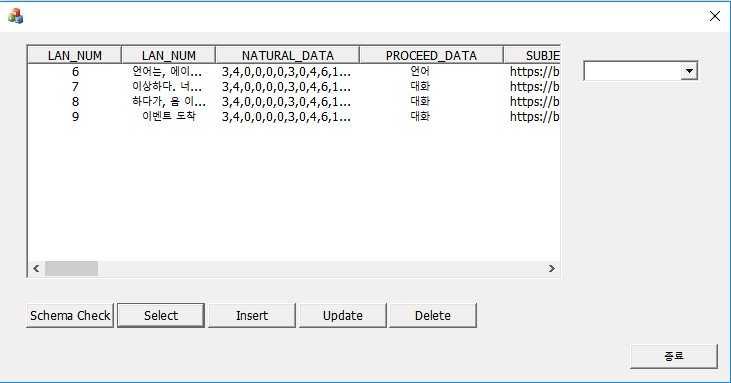




Senario 6 – 특정 테이블 관리자가 관리하는 언어 데이터 모두 반환

SELECT \* FROM LANGUAGEDATA WHERE MANAGER\_NO = %s

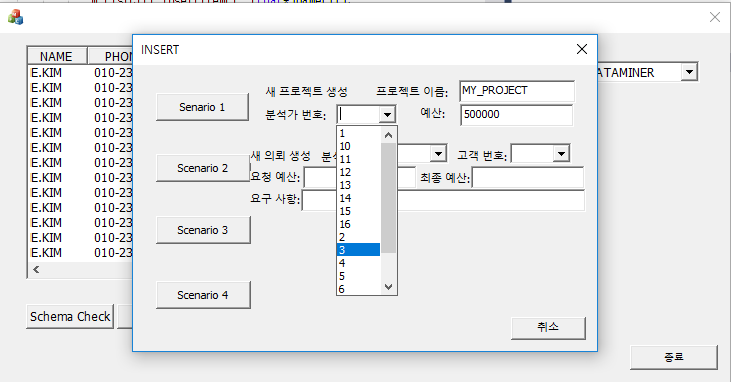


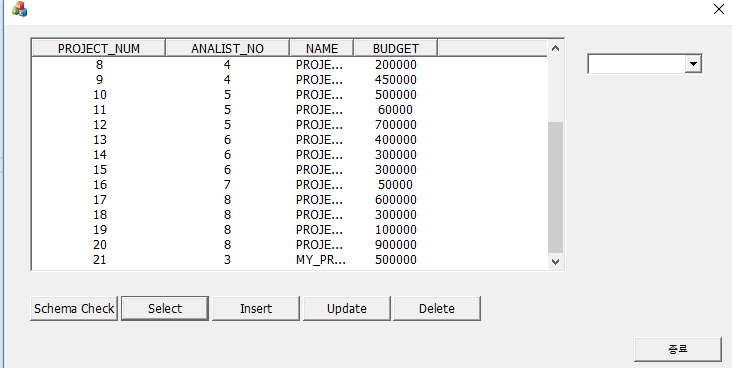


INSERT QUERY

Senario 1 – 입력값에 따라 새 프로젝트가 생성됨.

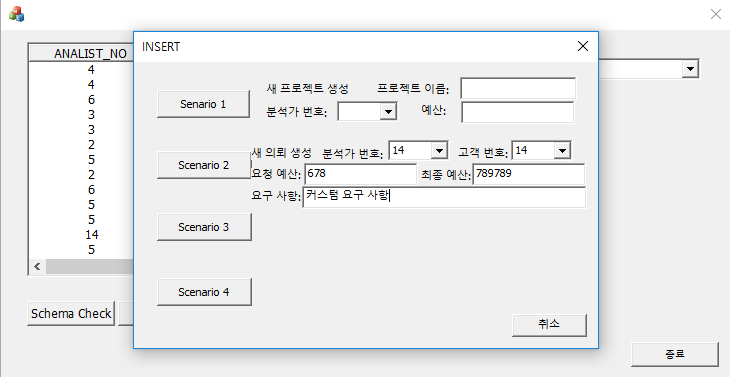
"INSERT INTO PROJECT VALUES(%d,%s,'%s',%s)",projectnum,Analist,str1,str2

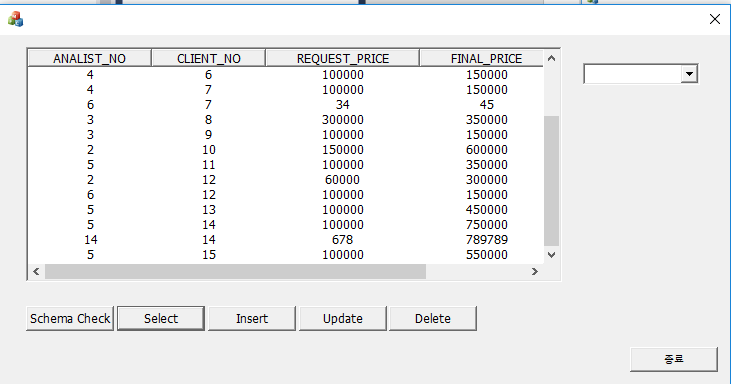




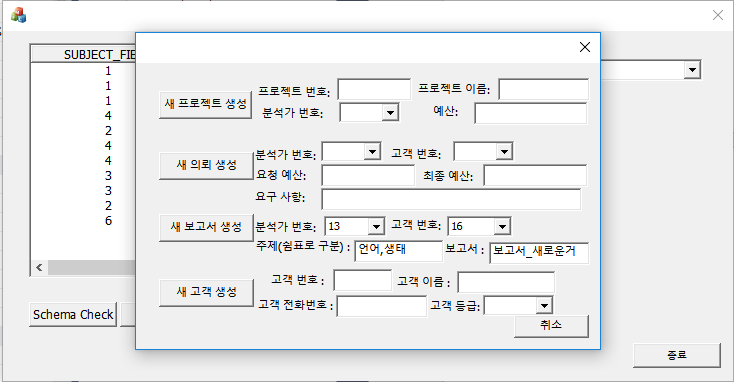
Senario 2 – 입력값에 따라 새 의뢰 생성.

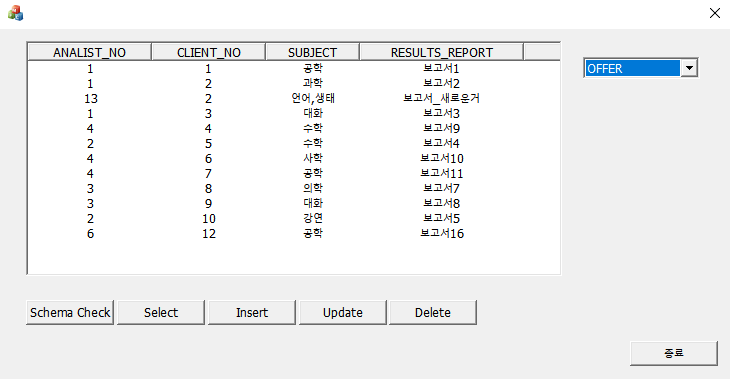
"INSERT INTO REQUEST VALUES(%s,%s,%s,%s,'%s')", Analist, Client, str1, str2,str3





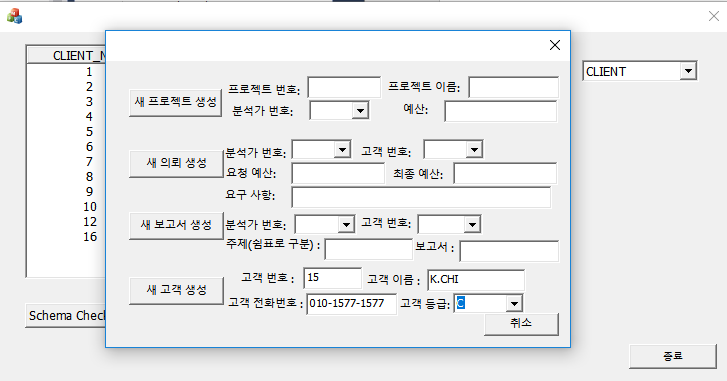
Senario 3 – 새 OFFER TABLE 생성

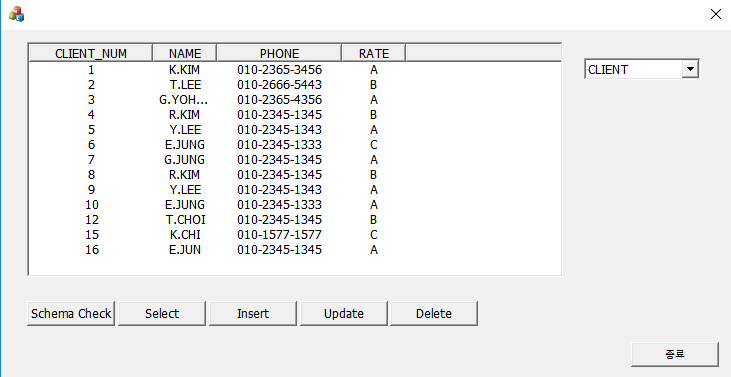




Senario 4 – 새 CLIENT TABLE 생성

INSERT INTO OFFER VALUES(%s,%s,'%s','%s')



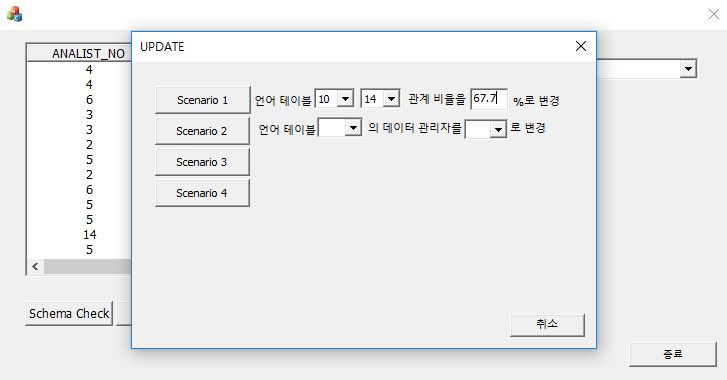


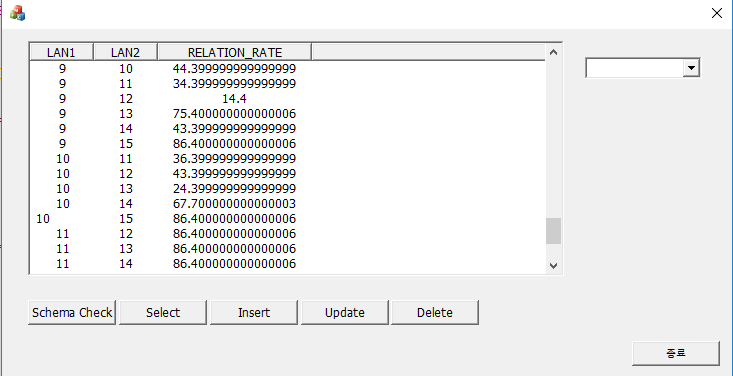
UPDATE QUERY

Senario 1 - 언어 테이블 간의 SIMILARITY 변경

INSERT INTO CLIENT VALUES(%s,'%s','%s','%s')

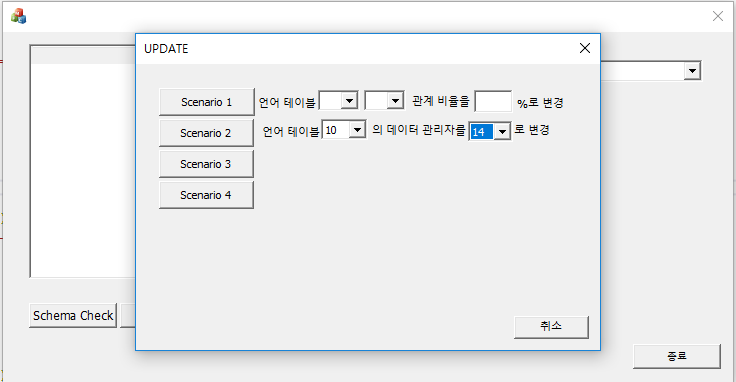
UPDATE SIMILARITY SET LAN1 = %s, LAN2 = %s, RELATION\_RATE = %s WHERE LAN1 = %s AND LAN2 = %s", str1, str2, str3, str1, str2

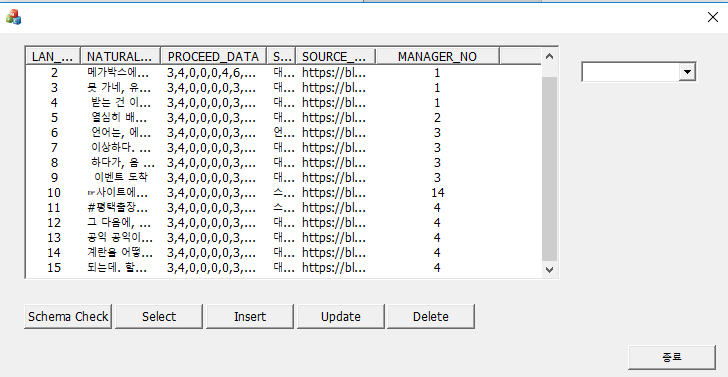




Senario 2 – 언어 테이블의 데이터 관리자 변경

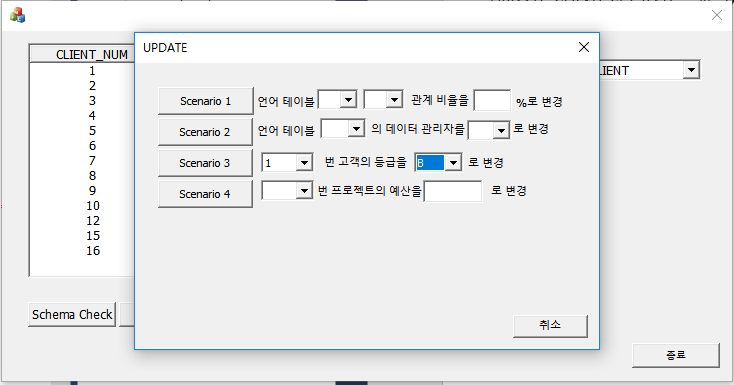
UPDATE LANGUAGEDATA SET MANAGER\_NO = %s WHERE LAN\_NUM = %s", str2, str1

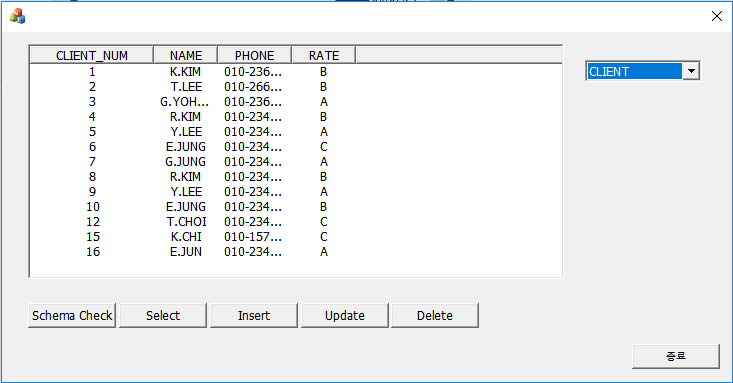




Senario 3 – 고객 테이블의 등급 변경

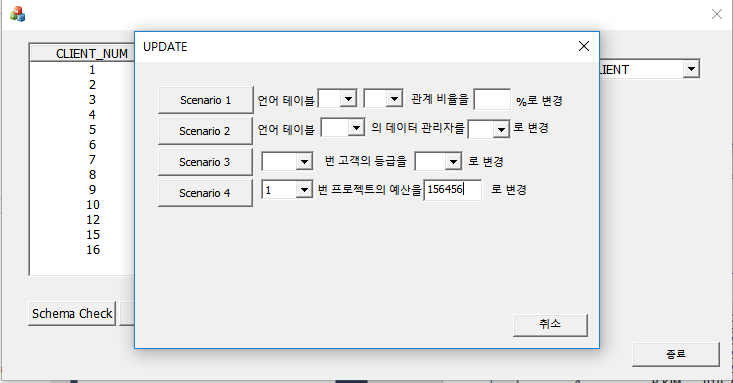
UPDATE CLIENT SET RATE = %s WHERE CLIENT\_NUM = %s

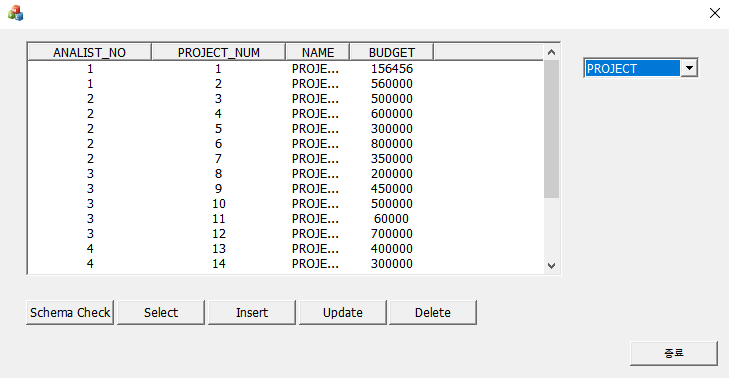




Senario 4 – 프로젝트 테이블의 예산 변경

UPDATE PROJECT SET BUDGET = %s WHERE PROJECT\_NUM = %s



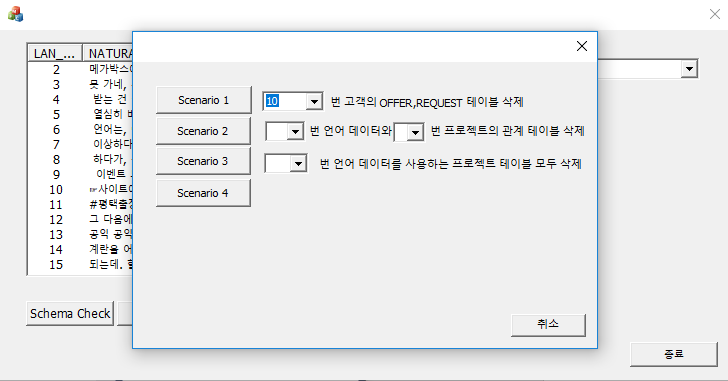


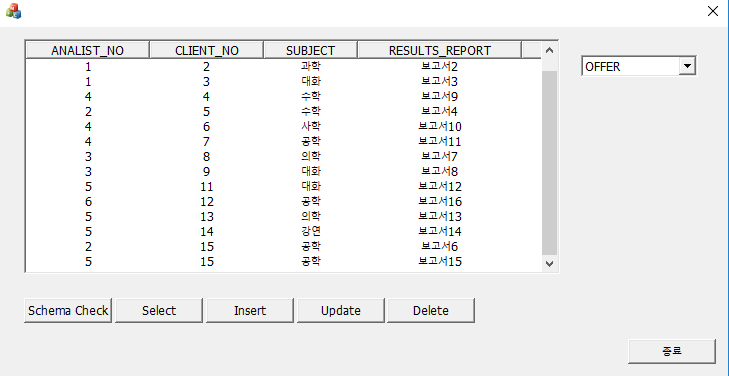
DELETE QUERY

Senario 1. 고객의 REQUEST,OFFER 모두 삭제.

DELETE FROM OFFER WHERE CLIENT\_NO = %s", Table

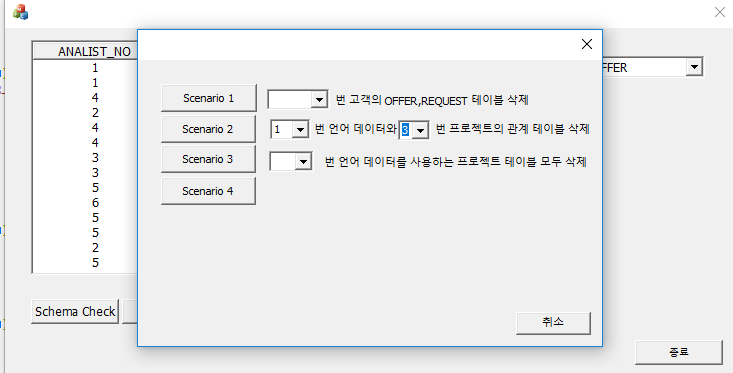
DELETE FROM REQUEST WHERE CLIENT\_NO = %s", Table

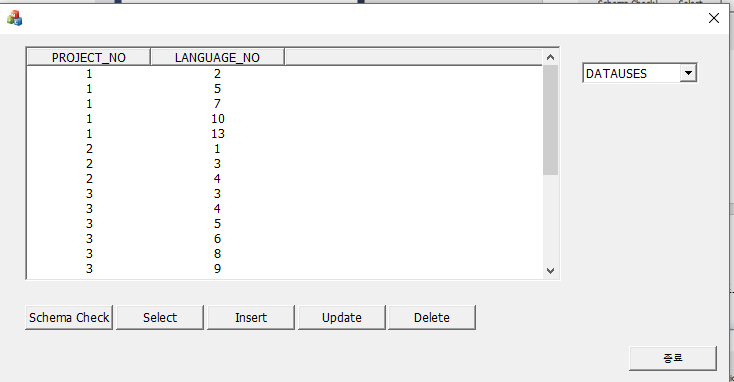




Senario 2. 특정 언어 데이터와 특정 프로젝트와의 관계 삭제.

DELETE FROM DATAUSES WHERE PROJECT\_NO = %s AND LANGUAGE\_NO = %s", str1,str2

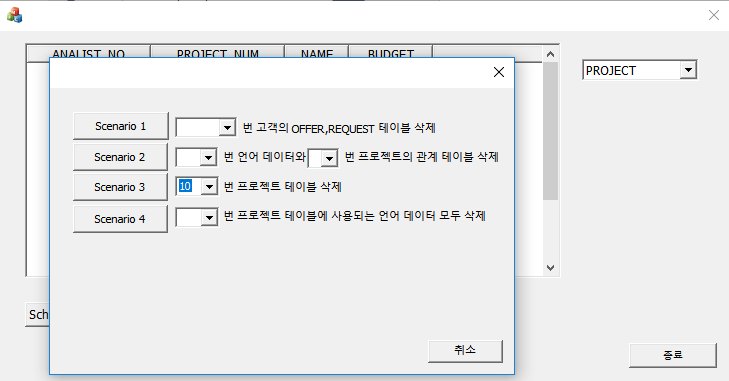


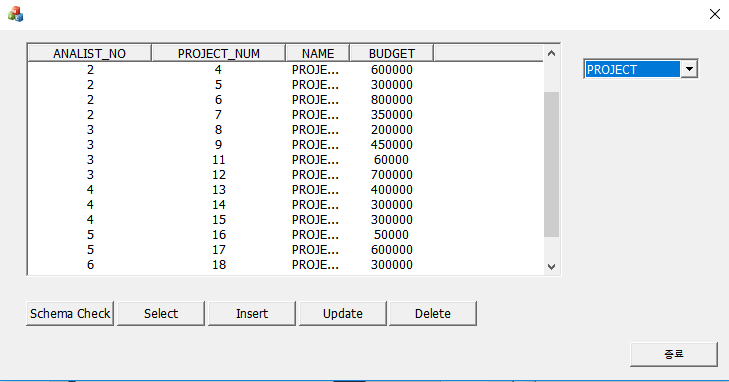


Senario 3. 프로젝트 테이블 삭제.

DELETE FROM DATAUSES WHERE PROJECT\_NO = %s

DELETE FROM PROJECT WHERE PROJECT\_NUM = %s





Senario 4. 특정 프로젝트 테이블이 사용하고 있는 언어 데이터 모두 삭제.

CREATE TABLE TMP(NUM INTEGER NOT NULL);

INSERT TMP SELECT CLIENT\_NO FROM REQUEST WHERE ANALIST\_NO = 5;

DELETE FROM REQUEST WHERE ANALIST\_NO = 5;

DELETE FROM OFFER WHERE ANALIST\_NO = 5;

DELETE FROM CLIENT WHERE CLIENT\_NUM IN (SELECT NUM FROM TMP);

DROP TABLE TMP;

