SQL – TCL / DCL

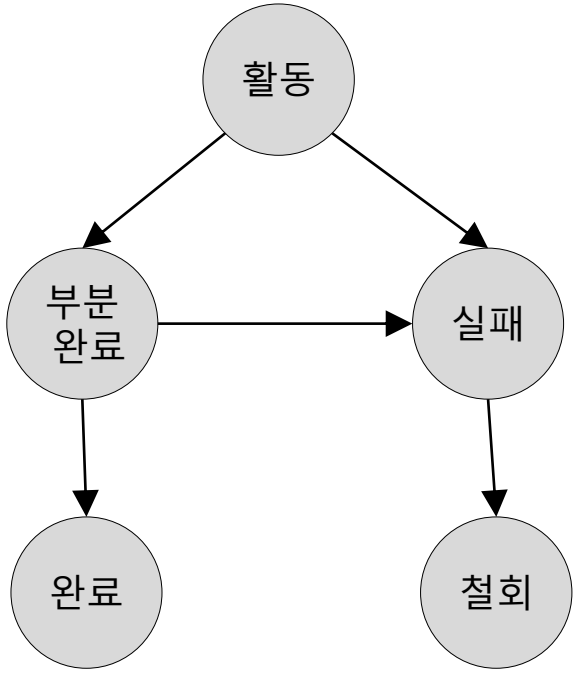
문혜영

TCL(Transaction Control Language)

- **트랜잭션 개념**
  - 데이터베이스의 논리적 연산 단위
    - 의미적으로 분할할 수 없는 최소의 단위
    - 일반적으로 하나의 트랜잭션은 여러 SQL 문장을 포함함
    - 성공시 모든 연산을 반영, 취소시 모든 연산을 취소함 → All or Nothing
  - 트랜잭션의 예
    - 도서 주문 : 재고 수량 감소, 주문 내역 생성, 결제, 포인트 적립
    - 계좌 이체 : 원 계좌의 잔액 감소, 다른 계좌의 잔액 증가
    - 교통카드 충전 : 잔액 증가, 결제

- 트랜잭션의 상태

1. **활동(Active)**
    - 트랜잭션이 시작되고 실행 중인 상태
    - 트랜잭션은 데이터베이스에 대한 연산(읽기, 쓰기 등)을 수행
2. **부분 완료(Partially Committed)**
    - 트랜잭션이 마지막 연산까지 실행
      -> 아직 데이터베이스에 영구적으로 반영되지는 않은 상태
3. **완료(Committed)**
    - 트랜잭션의 모든 연산이 성공적으로 데이터베이스에 반영되고 완료된 상태
4. **실패(Failed)**
    - 트랜잭션 수행 중 오류가 발생-> 정상적으로 완료될 수 없는 상태
5. **철회(Aborted)**
    - 실패한 트랜잭션이 시스템에 의해 취소
      -> 트랜잭션 시작 전 상태로 롤백된 상태



- **예시 :** 은행 계좌 A에서 다른 계좌 B로 1000달러를 이체하려고 함
- **트랜잭션 과정**:
    1. **트랜잭션 시작 (활동 상태)**
        - 이체 프로세스가 시작되면 트랜잭션은 '활동(active)' 상태가 됨
    2. **돈을 계좌 A에서 차감**
        - 시스템은 계좌 A의 잔액에서 1000달러를 빼고
        - 이 단계가 성공적으로 완료되면 트랜잭션은 '부분 완료(partially committed)' 상태에 들어감.
    3. **돈을 계좌 B에 추가**
        - 1000달러를 계좌 B에 추가
        - 이 단계가 성공적으로 완료되면, 이체 프로세스는 완료
        - 트랜잭션은 '완료(committed)' 상태
    4. **트랜잭션 완료 (완료 상태)**
        - 모든 단계가 성공적으로 완료되면, 이체는 확정되고, 더 이상 취소할 수 없음
        - 트랜잭션은 데이터베이스에 영구적으로 기록

- **트랜잭션의 특성 (ACID 특성)**
  - 원자성 (Atomicity)
    - 트랜잭션에서 정의된 연산들은 모두 성공적으로 실행되던지 아니면 전혀 실행되지 않은 상태로 남아 있어야 한다.
    - (all or nothing)
  - 일관성 (Consistency)
    - 트랜잭션이 실행 되기 전의 데이터베이스 내용이 잘못 되어 있지 않다면,
      트랜잭션이 실행된 이후에도 데이터베이스의 내용에 잘못이 있으면 안된다.
  - 고립성 (Isolation)
    - 트랜잭션이 실행되는 도중에 다른 트랜잭션의 영향을 받아서는 안된다.
  - 지속성 (Durability)
    - 트랜잭션이 성공적으로 수행되면 그 트랜잭션이 갱신한 데이터베이스의 내용은 영구 적으로 저장된다.

- **CRUD 분석**
  - ‘생성(Create), 읽기(Read), 갱신(Update), 삭제(Delete)’
  - 트랜잭션의 CRUD 연산에 대해 CRUD 매트릭스를 작성하여 분석.
  - 테이블에 발생되는 트랜잭션의 주기별 발생 횟수를 파악
    -> 연관된 테이블들을 분석.
  - 많은 트랜잭션이 몰리는 테이블을 파악
    -> 디스크 구성 시 유용한 자료로 활용
  - 외부 프로세스 트랜잭션의 부하가 집중되는 데이터베이스 채널을 파악
    -> 분산시킴으로써 연결 지연이나 타임아웃 오류를 방지

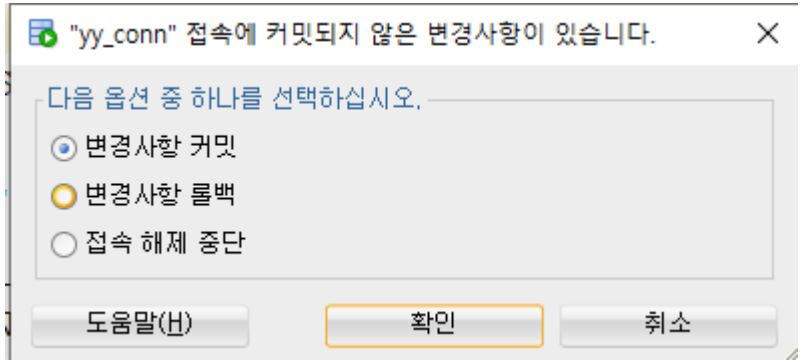
- **CRUD 매트릭스**
  - 행 : 프로세스, 열 : 테이블, 행과 열이 만나는 위치에는 프로세스가 테이블에 발생시키는 변화를 표시

업무 / 테이블	계좌(Accounts)	고객(Customers)	거래(Transactions)
계좌 관리	C, R, U, D	R	R
고객 관리	R	C, R, U, D	R
거래 처리	C, R	R	C, R, U

    - **계좌 관리** 업무
      - '계좌(Accounts)' 테이블에서 모든 CRUD 작업을 수행,
      - '고객(Customers)'과 '거래(Transactions)' 테이블에서는 조회(Read) 작업을 수행.
    - **고객 관리** 업무
      - '고객(Customers)' 테이블에서 모든 CRUD 작업을 수행하며,
      - '계좌(Accounts)'와 '거래(Transactions)' 테이블에서는 조회(Read) 작업을 수행.
    - **거래 처리** 업무
      - '거래(Transactions)' 테이블에서 거래의 생성(Create), 조회(Read), 업데이트(Update)를 수행,
      - '계좌(Accounts)' 테이블에서는 계좌 생성(Create)과 조회(Read),
      - '고객(Customers)' 테이블에서는 조회(Read) 작업을 수행

- **트랜잭션 분석**
  - 목적
    - CRUD 매트릭스를 기반으로 테이블에 발생하는 트랜잭션 양을 분석
      -> 테이블에 저장되는 데이터의 양을 유추
      -> 이를 근거로 DB 용량을 산정하고 DB 구조를 최적화
    - 프로세스가 과도하게 접근하는 테이블을 확인
      -> 여러 디스크에 배치
      -> 디스크 입 · 출력 분산을 통한 성능 향상을 기대

TCL



- **TCL**
  - 트랜잭션을 제어하기 위한 명령어
    - COMMIT : 변경된 내용을 DB에 영구적으로 반영
    - ROLLBACK:
      - 기본 – 변경된 내용을 버리고 변경 전 상태(마지막 COMMIT)로 복귀
      - SAVEPOINT를 지정한 경우, 지정한 저장점까지만 복귀
        - SAVEPOINT: 부분 복귀를 위해 지정한 저장점
  - 트랜잭션은 SQL문 실행시 자동 시작, COMMIT / ROLLBACK 실행시 종료
  - 자동 커밋 / 자동 롤백
    - DDL문장 수행시 DDL 수행 전에 자동으로 커밋 (auto commit)
    - DB를 정상적으로 접속 종료하면 자동 커밋
    - 애플리케이션의 이상 종료로 DB와의 접속이 단절되었을 때 자동 롤백

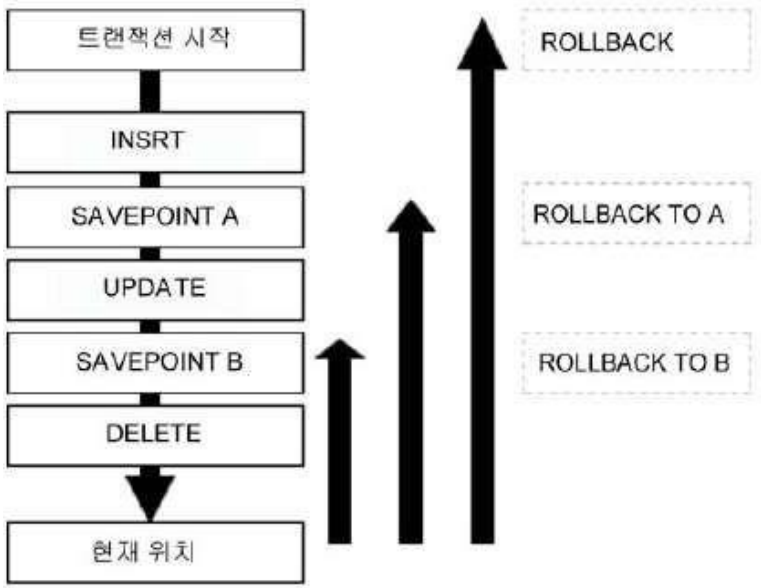
- **ROLLBACK**
  - ROLLBACK 이후의 데이터 상태

예)
STUDENT.SQL

```
SELECT * FROM STUDENT;
COMMIT;
UPDATE STUDENT SET S_SCORE=S_SCORE*2;
SELECT * FROM STUDENT;
ROLLBACK;
SELECT * FROM STUDENT;
```

  - 변경한 내용이 모두 취소됨
  - 이전 데이터가 다시 재저장 됨
  - 관련된 행에 대한 잠금이 해제되어 모든 사용자가 변경할 수 있음

- **SAVEPOINT**
  - 미리 지정한 SAVEPOINT까지만 ROLLBACK
    - 특정 저장점까지 롤백하면 그 이후의 명령과 저장점은 모두 무효가 됨
  - 일부 tool에서는 지원되지 않음
  - 동일 이름으로 여러 저장점 정의시 나중에 정의한 저장점이 유효

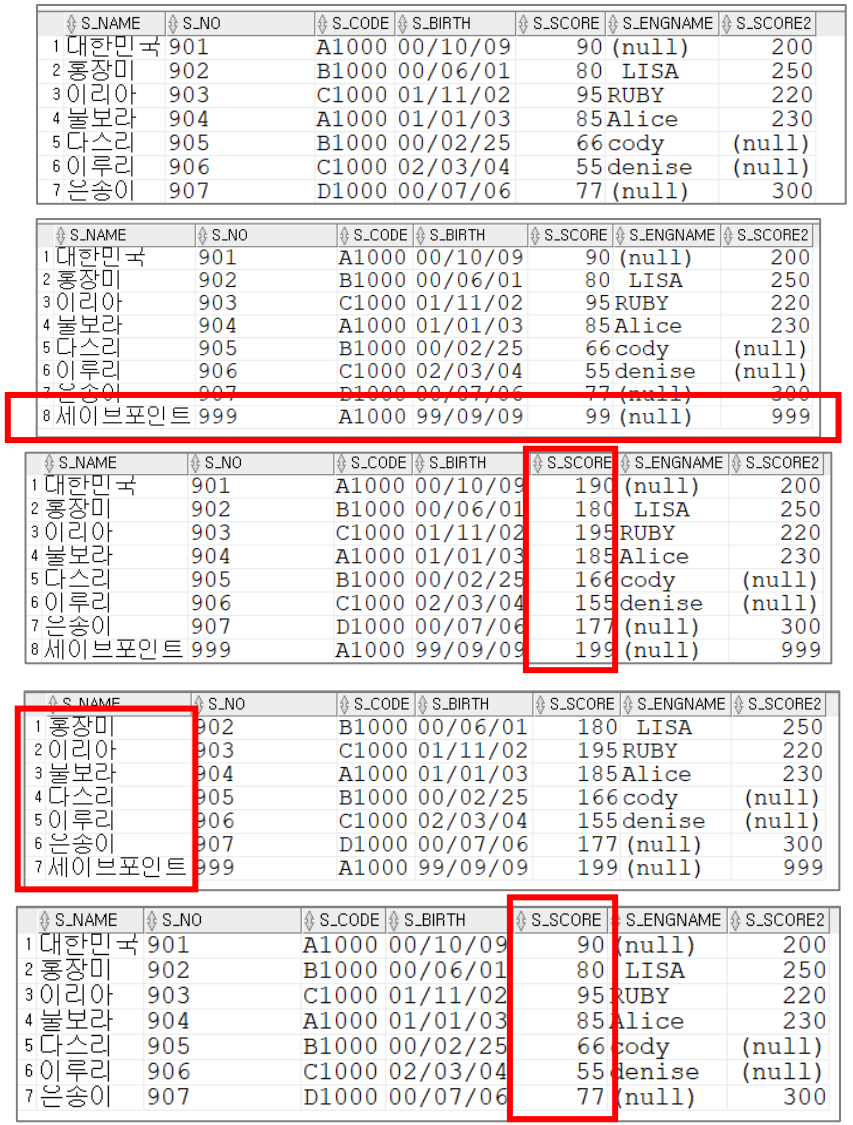



```
...
1  SAVEPOINT PT1;
...
2  SAVEPOINT PT1;
...
3  SAVEPOINT PT2;
...
   ROLLBACK TO PT1;
```

(2)번으로 ROLLBACK → (2)번 이후는 무효가 됨

- 예) STUDENT.SQL

```
SELECT * FROM STUDENT;
INSERT INTO STUDENT VALUES
('세이브포인트','999','A1000', TO_DATE('2999-09-09', 'YYYY-MM-DD'),99,'',999);
SELECT * FROM STUDENT;
SAVEPOINT A;
UPDATE STUDENT SET S_SCORE=S_SCORE+100;
SELECT * FROM STUDENT;
SAVEPOINT B;
DELETE FROM STUDENT WHERE S_NAME='대한민국';
SELECT * FROM STUDENT;
ROLLBACK A;
SELECT * FROM STUDENT;
```



	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300
8	세이브포인트	999	A1000	99/09/09	99	(null)	999

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	190	(null)	200
2	홍장미	902	B1000	00/06/01	180	LISA	250
3	이리아	903	C1000	01/11/02	195	RUBY	220
4	불보라	904	A1000	01/01/03	185	Alice	230
5	다스리	905	B1000	00/02/25	166	cody	(null)
6	이루리	906	C1000	02/03/04	155	denise	(null)
7	은송이	907	D1000	00/07/06	177	(null)	300
8	세이브포인트	999	A1000	99/09/09	199	(null)	999

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	홍장미	902	B1000	00/06/01	180	LISA	250
2	이리아	903	C1000	01/11/02	195	RUBY	220
3	불보라	904	A1000	01/01/03	185	Alice	230
4	다스리	905	B1000	00/02/25	166	cody	(null)
5	이루리	906	C1000	02/03/04	155	denise	(null)
6	은송이	907	D1000	00/07/06	177	(null)	300
7	세이브포인트	999	A1000	99/09/09	199	(null)	999

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300

- 예) STUDENT.SQL

```
SELECT * FROM STUDENT;
INSERT INTO STUDENT VALUES
('세이브포인트','999','A1000', TO_DATE('2999-09-09', 'YYYY-MM-DD'),99,'',999);
SELECT * FROM STUDENT;
SAVEPOINT A;
UPDATE STUDENT SET S_SCORE=S_SCORE+100;
SELECT * FROM STUDENT;
SAVEPOINT B;
DELETE FROM STUDENT WHERE S_NAME='대한민국';
SELECT * FROM STUDENT;
ROLLBACK B;
SELECT * FROM STUDENT;
```

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300
8	세이브포인트	999	A1000	99/09/09	99	(null)	999

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	190	(null)	200
2	홍장미	902	B1000	00/06/01	180	LISA	250
3	이리아	903	C1000	01/11/02	195	RUBY	220
4	불보라	904	A1000	01/01/03	185	Alice	230
5	다스리	905	B1000	00/02/25	166	cody	(null)
6	이루리	906	C1000	02/03/04	155	denise	(null)
7	은송이	907	D1000	00/07/06	177	(null)	300
8	세이브포인트	999	A1000	99/09/09	199	(null)	999

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	홍장미	902	B1000	00/06/01	180	LISA	250
2	이리아	903	C1000	01/11/02	195	RUBY	220
3	불보라	904	A1000	01/01/03	185	Alice	230
4	다스리	905	B1000	00/02/25	166	cody	(null)
5	이루리	906	C1000	02/03/04	155	denise	(null)
6	은송이	907	D1000	00/07/06	177	(null)	300
7	세이브포인트	999	A1000	99/09/09	199	(null)	999

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300

DCL

- DCL(Data Control Language, 데이터 제어어)
  - DCL은 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는데 사용됨.
  - 데이터베이스 관리자가 데이터 관리를 목적으로 사용.
  - **종류**
    - GRANT : 권한을 부여, REVOKE : 권한을 취소.
    - **1) 사용자등급 지정 및 해제**

- GRANT 사용자등급 TO 사용자_ID_리스트 [IDENTIFIED BY 암호];
- REVOKE 사용자등급 FROM 사용자_ID_리스트;

      - 사용자 등급
        - DBA : 데이터베이스 관리자
        - RESOURCE : 데이터베이스 및 테이블 생성 가능자
        - CONNECT : 단순 사용자
      - 예1) 사용자 ID가 "NANA"인 사람에게 데이터베이스 및 테이블을 생성할 수 있는 권한을 부여

```
GRANT RESOURCE TO NANA;
```

      - 예2) 사용자 ID가 "RUBY"인 사람에게 단순히 데이터베이스에 있는 정보를 검색할 수 있는 권한을 부여

```
GRANT CONNECT TO RUBY;
```

      - 예3) 새로운 사용자 yyid 생성하고, 비번은 yypw로 함, DBA, RESOURCE, CONNECT 권한을 부여함

```
CREATE USER yyid IDENTIFIED BY yypw ;
GRANT CONNECT, DBA, RESOURCE TO yyid ;
```

- 예.

- 새로운 사용자 test1을 생성하고 패스워드를 pw1로 설정

```
CREATE USER test1 IDENTIFIED BY pw1 ;
```

- 사용자 ' test1'의 비번을 'pw2'로 변경

```
ALTER USER test1 IDENTIFIED BY pw2 ;
```

- 사용자 ' test1 ' 에게
  새로운 사용자 계정을 생성할 수 있는 권한을 부여

```
GRANT CREATE USER TO test1;
```

- 사용자 ' test1 ' 로부터
  새로운 사용자 계정을 생성할 수 있는 권한을 회수

```
REVOKE CREATE USER FROM test1;
```

- 사용자 test1 계정을 삭제

```
DROP USER test1 CASCADE;
```

DROP 연산 수행 시…

- CASCADE 옵션 사용시 사용자가 생성한 객체도 함께 삭제됨
- CASCADE 옵션 미사용시 사용자가 객체를 갖고 있지 않은 경우에 만 삭제 실행

- **2) 테이블 및 속성에 대한 권한 부여 및 취소**

  - GRANT 권한_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];
  - REVOKE [GRANT OPTION FOR] 권한_리스트 ON 개체 FROM 사용자 [CASCADE];

  - 권한 종류 : ALL, SELECT, INSERT, DELETE, UPDATE, ALTER 등
  - WITH GRANT OPTION : 부여 받은 권한을 다른 사용자에게 다시 부여할 수 있는 권한을 부여함
  - REVOKE GRANT OPTION FOR : 다른 사용자에게 권한을 부여할 수 있는 권한을 취소함
  - CASCADE : 권한 취소 시 권한을 부여 받았던 사용자가 다른 사용자에게 부여한 권한도 연쇄적으로 취소함

예) 사용자 ID가 "NANA"인 사람에게 <STUDENT> 테이블에 대한 모든 권한과
다른 사람에게 권한을 부여할 수 있는 권한까지 부여.

```
GRANT ALL ON STUDENT TO NANA WITH GRANT OPTION;
```

예) 사용자 ID가 "RUBY"인 사람에게 부여한 <STUDENT> 테이블에 대한 권한 중
UPDATE 권한을 다른 사람에게 부여할 수 있는 권한만 취소..

```
REVOKE GRANT OPTION FOR UPDATE ON STUDENT FROM RUBY;
```

- <STUDENT>테이블에서 'S_NO'가 '907'인 학생의 정보를 삭제한 후 COMMIT을 수행하시오.

- <STUDENT>테이블에서 'S_NO'가 '906'인 학생의 정보를 삭제하시오.

- SAVEPOINT 'S1'을 설정하고 'S_NO'가 '905'인 학생의 정보를 삭제하시오.

DEPT.SQL

	D_CODE	D_NAME
1	A1000	경영정보
2	B1000	데이터사이언스
3	C1000	AI
4	D1000	성악

STUDENT.SQL

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)
7	은송이	907	D1000	00/07/06	77	(null)	300

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230

- SAVEPOINT ‘S2’을 설정하고 ‘S_NO’가 ‘904’인 학생의 정보를 삭제하시오.

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220

- SAVEPOINT ‘S2’까지 ROLLBACK을 수행하시오.

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230

- SAVEPOINT ‘S1’까지 ROLLBACK을 수행하시오.

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)

- SAVEPOINT 없이 ROLLBACK을 수행하시오.

	S_NAME	S_NO	S_CODE	S_BIRTH	S_SCORE	S_ENGNAME	S_SCORE2
1	대한민국	901	A1000	00/10/09	90	(null)	200
2	홍장미	902	B1000	00/06/01	80	LISA	250
3	이리아	903	C1000	01/11/02	95	RUBY	220
4	불보라	904	A1000	01/01/03	85	Alice	230
5	다스리	905	B1000	00/02/25	66	cody	(null)
6	이루리	906	C1000	02/03/04	55	denise	(null)

수고하셨습니다