

[Database]트랜잭션(COMMIT,ROLLBACK)과 Lock 이란?

노트북: [TIL-MY]

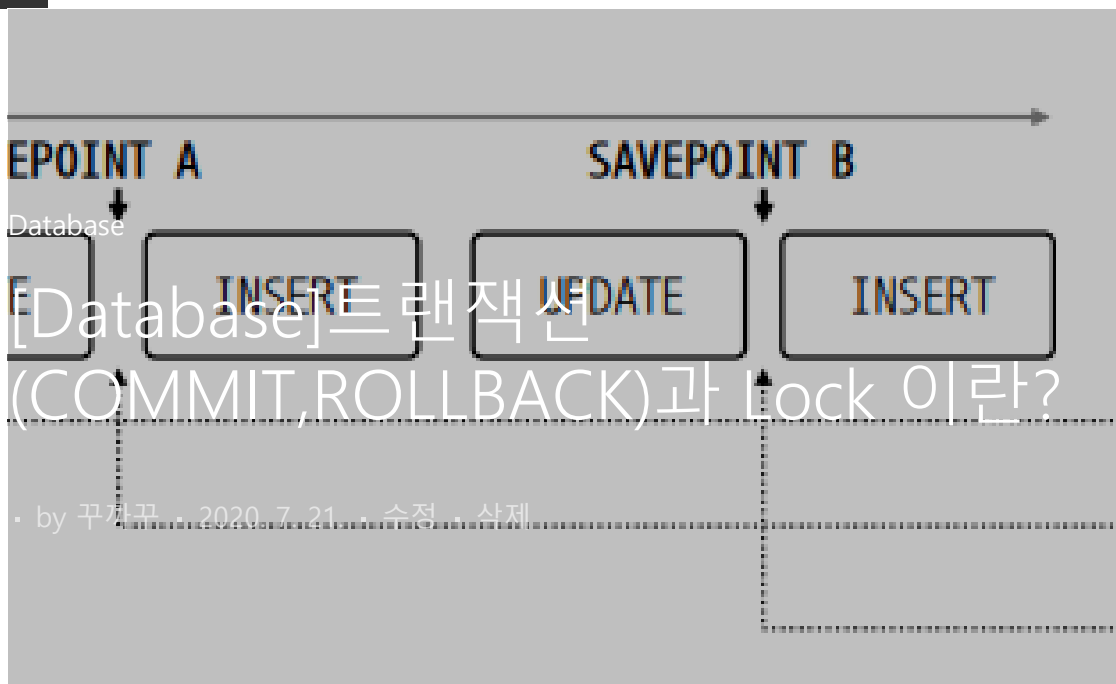
만든 날짜: 2020-07-21 오전 9:55

URL: <https://continuous-development.tistory.com/31>

나무늘보의 개발 블로그

홈

태그



트랜잭션

데이터 일관성을 유지시키려는 목적으로 사용하는
논리적으로 연관된 작업들의 집합 / 일련의 일의 작

분류 전체보기

Python

Database

ASP.NET

Algorithm

Deep learning

업 단위이다.

이런 트랜잭션에는 하나 이상의 연관된 **DML 구문**과 하나 이상의 **DDL 구문**이 들어간다.

트랜잭션 시작

첫 번째 DML 구문이 실행될 때 시작됨

트랜잭션 종료

COMMIT / ROLLBACK 명령이 실행될 때 종료(COMMIT)

DDL 구문이 실행될 때 종료 -> (AUTO COMMIT)

비정상적으로 DBMS 종료되는 경우 -> AUTO ROLLBACK

DDL을 중간에 쓰게되면 AUTO COMMIT이 돼서 저장되어버린다.

AWS

ETC..

공지사항

글 보실 때 주의사항

: 최근글 : 인기글



2020.07.21

[Data] INSERT

2020.07.20

[Database] DDL(데이터 정..)

2020.07.19

[Data] 테이블...

2020.07.18

[Data] Sub...

2020.07.17

최근댓글

태그

데이터 정의어,
INSERT함수,
trunc 사용,

그래서 보통 DDL을 중간에 안쓴다. DDL은 초반과 마지막에 들어간다.

--commit을 통한 트랜잭션 종료
INSERT INTO ~
UPDATE ~
COMMIT

--DDL을 통한 트랜잭션이 종료
UPDATE ~
DELETE FROM ~
CREATE OR REPLACE VIEW ~

트랜잭션 제어

COMMIT - 변경된 데이터를 저장하고 트랜잭션을 종료하는 명령

ROLLBACK - 변경 작업을 취소하고 트랜잭션을 종료하는 명령 / 기본적으로 데이터 상태를 트랜잭션의 시작 시점으로 돌림

SAVEPOINT savepoint_name : 트랜잭션의 특정 시점을 기록하는 명령

INSERT UPDATE
DELETE,
sql rollback,
사용법,
DELETE 사용법,
서브쿼리 사용법,
CRATE TABLE,
sql commit,
INSERT 사용법,
trunc 함수,
rollback 사용법,
Subquery 함수,
commit 사용법,
DDL, 인스턴스,
UPDATE함수,
AWS, 날짜함수,
SQL 테이블 생성,
select trunc,
테이블 생성,
SQL, 설정,
DELETE함수,
Oracle,
SUBQUERY함수,
Oracle SQL,
UPDATE 사용법

—

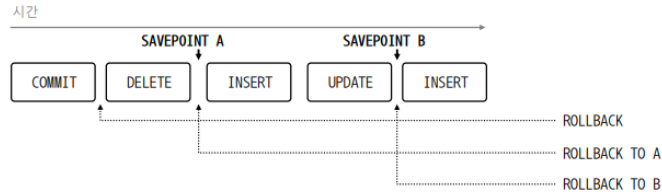
전체 방문자

93

Today : 0

Yesterday : 2

ROLLBACK TO savepoint_name : 지정한 특정 시
점으로 데이터 상태를 되돌릴 수 있음



```
ALTER TABLE EMPLOYEE
```

```
DISABLE CONSTRAINTS FK_MGRID;
```

```
SAVEPOINT S0; --savepoint
```

```
INSERT INTO DEPARTMENT
```

```
VALUES ('40', '기획전략팀', 'A1');
```

```
SAVEPOINT S1; --savepoint
```

```
UPDATE EMPLOYEE
```

```
SET DEPT_ID = '40'
```

```
WHERE DEPT_ID IS NULL;
```

```
SAVEPOINT S2; --savepoint
```

```
DELETE FROM EMPLOYEE
```

--이런식으로 SAVEPOINT 지점을 잡아 준후
--ROLLBACK TO S1/S2/S3 이런식으로 원하는 지
점으로 돌아갈수 있게끔 한다.

Lock

동시성 제어 개념 : 다수 사용자들이 동시에 동일한 데이터에 접근하여 변경하려 하는 것을 제어한다.

EX) 영화관 좌석

특징

서로 다른 사용자가 동시에 동일한 행을 변경할 수 없도록 방지한다.

다른 사용자가 commit 되지 않은 변경 내용을 Over write 할 수 없도록 방지한다.

트랜잭션이 실행되는 동안 자동으로 수행/유지/관리 한다.

```
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL> SELECT EMP_NAME
2 SET MARriage='N'
3 FROM EMPLOYEE
4 WHERE EMP_ID = '143';
EMP_NAME MA
-----
나승원 Y
SQL> UPDATE EMPLOYEE
2 SET MARriage='N'
3 WHERE EMP_ID='143';
1 row updated.
SQL>
SQL>
SQL> UPDATE EMPLOYEE
2 SET MARriage='N'
3 WHERE EMP_ID='143';
UPDATE EMPLOYEE
ERROR at line 1:
ORA-00942: table or view does not exist
SQL>
SQL>
SQL> SELECT EMPLOYEE
2
SQL> E;
SP2-0042: unknown command "E" - rest of line ignored.
SQL> UPDATE EMPLOYEE
2 SET MARriage='N'
3 WHERE EMP_ID='143';
```

이 두 개의 세션 중에 한쪽에서 커밋을 마치지 않은 상태에서 다른 SQL에 창에서 같은 데이터를 수정하려면 LOCK이 걸려서 되지 않는다.



'Database' 카테고리의 다른 글

- [\[Database\]트랜잭션\(COMMIT,ROLLBACK\)과 Lock 이란?](#) (0) 09:55:21
- [\[Database\] INSERT,UPDATE,DELETE 문 정리와 다양한 예제](#) (0) 2020.07.20
- [\[Database\] DDL\(데이터 정의어\), DML\(데이터 조작어\), DCL\(데이터 제어어\) 이란?](#) (0) 2020.07.19
- [\[Database\] 테이블 생성\(CREATE TABLE\) 및 제약조건\(CONSTRAINT\)](#) (0) 2020.07.18
- [\[Database\] Subquery\(단일,다중\) 사용법](#) (0) 2020.07.17
- [\[Database\] SET Operator\(UNION,UNION ALL, INTERSECT,MINUS\) & IN Operator사용법](#) (0) 2020.07.17

태그

[COMMIT](#)[commit 사용법](#)[Rollback](#)[rollback 사용법](#)[savepoint](#)[sql commit](#)[sql rollback](#)[트랜잭션](#)

관련글



[Database]... [Database]... [Database]... [Database]...

댓글 0

