# 47_ YASH SARANG_ EXPERIMENT 11_ DBMS

## Transition Control:

A transition can defined as a group of tasks. A single task is the min. processing unit which cannot be divided further.

Let's take an eg. of a bank employee transfering 500 Rs from A to B.

| A's acc | B's acc. |
|---|---|
| Open_Acc(A) = A.balance | Open_Acc(B) = B.balance |
| New_Bal = Old_Bal − 500 | New_Bal = Old_Bal + 500. |
| A.balance = New_Bal | B.balance = New_Bal. |
| Close_Acc(A) | Close Acc(B). |

## Concurrent Control.

1. Concurrency control manages the transactions simultaneously without letting them interfere with each other.
2. The main objective of concurrency control is to allow many users perform different operations at the same time.
3. Using more than one transaction concurrently improves the performance of system.
4. If we are not able to perform the operations concurrently, then there can be serious problems such as loss of data integrity & consistency.
5. It reduces waiting time of transaction.

```
SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
------------------- ---------- ---------- ---------- ----------


subrato                  55000          1          3          1
manas                    51750          2          2          2
jayesh                   65000          3          1          3
sarthak                46287.5          4          4          4
arnav                    50000          5          5          5
```

- CHANGING SALARY OF SSN=1
- COMMIT,SAVING IT
- CHANGING SALARY OF SSN=2
- ROLLING BACK THE CHANGE

```
SQL> set serveroutput on;
SQL>  DECLARE
  2   salary emp.salary%type;
  3  BEGIN
  4  dbms_output.put_line('Updating salary of employee,ssn=1');
  5  update emp set salary=salary+10000 where ssn=1;
  6   dbms_output.put_line('Committing transaction');
  7  COMMIT;
  8   dbms_output.put_line('Adding a savepoint');
  9  SAVEPOINT sal1;
 10  dbms_output.put_line('Updating salary of employee,ssn=2');
 11  update emp set salary=salary+10000 where ssn=2;
 12  dbms_output.put_line('Rolling back to savepoint');
 13  ROLLBACK TO sal1;
 14  END;
 15  /
Updating salary of employee,ssn=1
Committing transaction
Adding a savepoint
Updating salary of employee,ssn=2
Rolling back to savepoint

PL/SQL procedure successfully completed.
```

- WE CAN SEE THAT ONLY SALARY OF EMP SSN=1 GETS UPDATED

```
SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
------------------- ---------- ---------- ---------- ----------


subrato                  65000          1          3          1
manas                    51750          2          2          2
jayesh                   65000          3          1          3
sarthak                46287.5          4          4          4
arnav                    50000          5          5          5
```

```
SQL>  set serveroutput on;
SQL> DECLARE
  2   rollno student.sno%type;
  3  s_age student.age%type;
  4  snm student.sname%type;
  5  s_cr student.course%type;
  6  BEGIN
  7  rollno:=&sno;
  8  s_age:=&age;
  9  snm:='&name';
 10  s_cr:='&course':
 11  insert into student values (rollno,s_age,snm,s_cr);
 12  dbms_output.put_line('one record inserted');
 13  COMMIT;
 14  SAVEPOINT save1;
 15   rollno:=&sno;
 16   s_age:=&age;
 17  snm:='&name';
 18  s_cr:='&course':
 19  INSERT into student values (rollno,s_age,snm,s_cr);
 20   dbms_output.put_line('one record inserted');
 21  ROLLBACK to SAVEPOINT save1;
 22  END;
 23  /
```

```
Enter value for sno: 60
old   7: rollno:=&sno;
new   7: rollno:=60;
Enter value for age: 19
old   8: s_age:=&age;
new   8: s_age:=19;
Enter value for name: subrato
old   9: snm:='&name';
new   9: snm:='subrato';
Enter value for course: ai
old  10: s_cr:='&course':
new  10: s_cr:='ai':
Enter value for sno: 61
old  15:  rollno:=&sno;
new  15:  rollno:=61;
Enter value for age: 19
old  16:  s_age:=&age;
new  16:  s_age:=19;
Enter value for name: vivek
old  17: snm:='&name';
new  17: snm:='vivek';
Enter value for course: it
old  18: s_cr:='&course':
new  18: s_cr:='it':
```

```
SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEE_ID                               NOT NULL NUMBER(38)
 ENAME                                     NOT NULL VARCHAR2(20)
 SALARY                                             NUMBER
 FK_EMPLOYEE                                        NUMBER(38)
 D_NO                                               NUMBER(38)

SQL> lock table employee in share mode;

Table(s) Locked.

SQL> commit;

Commit complete.
```

```
SQL> lock table employee in exclusive mode;

Table(s) Locked.

SQL> rollback;

Rollback complete.
```

```
SQL> lock table system.Donor in exclusive mode;

Table(s) Locked.

SQL> lock table system.Receiver in exclusive mode;
lock table system.Receiver in exclusive mode
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource


SQL> rollback;

Rollback complete.
```

```
SQL> insert into emp values('mahi',30000,6,6,6);

1 row created.
```

```
SQL> savepoint s1;

Savepoint created.

SQL> update emp
  2  set salary=salary+1000;

13 rows updated.

SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------


subrato                  56000          1          3          1
manas                    52750          2          2          2
jayesh                   66000          3          1          3
sarthak                47287.5          4          4          4
arnav                    51000          5          5          5




NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------
Ramesh                   11000          1          2          1
mahi                     31000          6          6          6

13 rows selected.

SQL> rollback to s1;

Rollback complete.
```

```
SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------


subrato                  55000          1          3          1
manas                    51750          2          2          2
jayesh                   65000          3          1          3
sarthak                46287.5          4          4          4
arnav                    50000          5          5          5




NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------
Ramesh                   10000          1          2          1
mahi                     30000          6          6          6
```