SavePoint(I)

Within a transaction, if we want to rollback a particular group of operations based on some condition then we need to use savepoint

a. getting savepoint using the following method

Savepoint sp = connection.setSavePoint();

b. To perform rollback operation for a particular group of operations w.r.t savepoint then we need to use rollback.

connection.rollback(sp);

c. we can release the savepoint or delete the savepoint as shown below

connection.releaseSavePoint(sp);

Case study

=====================================

connection.setAutocommit(false);

operation-1

operation-2

operation-3

SavePoint sp = connection.setSavePoint();

operation-4

operation-5

if(balance<1000)

connection.rollback(sp);

else

connection.releaseSavepoint(sp);

connection.commit();

if balance< 1000 then operation 4,5 will be rollbacked, otherwise all the operations will be committed.

eg:

connection.setAutocommit(false);

st.executeUpdate("insert into politicians values('BJP', 'Modi'));

st.executeUpdate("insert into politicians values('TRS','KCR')");

SavePoint sp = connection.setSavePoint();

st.executeUpdate("insert into politicians values('BJP','siddu')");

connection.rollback(sp);

..

...

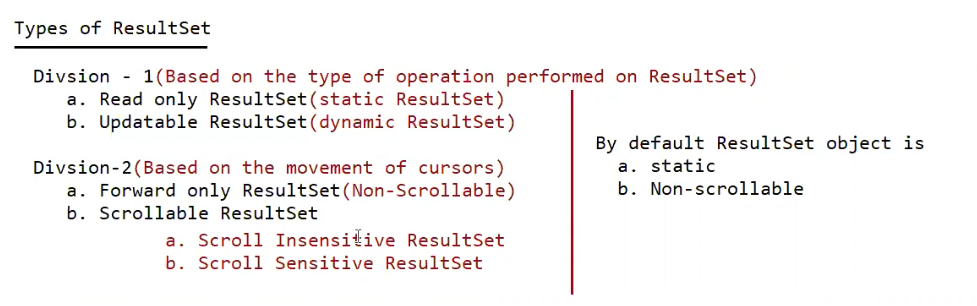
connection.releaseSavePoint(sp);

..

connection.commit();

Eg: Jdbc\_Transaction\_Savepoint\_Rollback

Types of ResultSet:



create table scrollableapp ( id int , name nvarchar2(15) , age int , address nvarchar2(15));

insert into scrollableapp values ( 1 , 'pavan' , 24 , 'vijayawada');

insert into scrollableapp values ( 2 , 'anand' , 23 , 'vijayawada');

insert into scrollableapp values ( 3 , 'pavankumar' , 24 , 'vijayawada');

insert into scrollableapp values ( 4 , 'charan' , 23 , 'ongole');

insert into scrollableapp values ( 5 , 'tarun' , 24 , 'vijayawada');

insert into scrollableapp values ( 6 , 'mpavan' , 24 , 'vijayawada');

insert into scrollableapp values ( 7 , 'ppavan' , 24 , 'vijayawada');

insert into scrollableapp values ( 8 , 'teja' , 24 , 'vijayawada');

absolute() -> it works from the BFR or from ALR.

relative() -> it works w.r.t current position.

In both the methods positive means move in forward direction, negative means move in backward direction.

Note:

rs.last() and rs.absolute(-1) both are equal

rs.first() and rs.absolute(1) both are equal

Eg: Scrollable\_App

Scroll InSensitive ResultSet

After getting resultSet if we are performing any operations in the database, and if those changes are not reflecting to the resultSet, such type of ResultSet are called as "Scroll Insensitive ResutlSet".

public static final int TYPE\_SCROLL\_INSENSITIVE

Scroll Sensitive ResultSet

After getting resultSet if we are performing any operations in the database, and if those changes are getting reflecting to the resultSet, such type of ResultSet are called as "Scroll Sensitive ResutlSet”.

public static final int TYPE\_SCROLL\_SENSITIVE

Note:

When we make the ResultSet as ScrollSensitive, then we need to use resultSet.refreshRow() to get the updated records from the database.

Eg: Scrollable\_Sensitive\_App