메일_보내기

- description :
- author : 도봉산핵주먹
- email : hylee@repia.com
- lastupdate : 2020-05-07

적용 이유

> PAN*** (수집기)내부 프로젝트 진행 중 수집 데이터 건마다 DB 커넥트를 해서 부하가 생김.
> 커넥트를 최대한 줄이기 위해 구현함.

기존 소스

```
public class RpCubridinsert
{
    --- 생략 ---
    List<RpCllctVO> cllctVOs = new ArrayList<>();
    // 수집한 데이터 조회
    cllctVOs = dbUtil.getCllctSelectList(mode, sDate, eDate, runType);

    // 수집데이터 갯수 만큼 반복
    for(RpCllctVO cllctVO: cllctVOs)
    {
        dbUtilCu.insertCllctCubridData(cllctVO);
    }
    --- 생략 ---
}
public int insertCllctCubridData(RpCllctVO cllctVO) throws SQLException,
UnsupportedEncodingException
{
    DBManagerCu dbm = DBManagerCu.getInstanceCu();
    Connection conn = dbm.getConnection();

    PreparedStatement pstmt = null;
    int ret = 0;
    String sqlText;

    sqlText = " INSERT INTO TB_INSEQCMS_BBS_DATA "
            + " (           "
            + "     BBS_ID          "
            + "     ,SEQ            "
            + "   -- 등등 생략 --    "
            + " )                    "
            + "     VALUES          "
            + " (             "
```

```
            + "      'news-media'     "
            + "      ,?               "
            + "   -- 등등 생략 --     "
            + ")                          ";

    try
    {
        pstmt = conn.prepareStatement(sqlText);

        pstmt.setInt(1,     cllctVO.getBitnaraSeq());
                    -- 등등 생략 --

        ret = pstmt.executeUpdate();
    }
    finally
    {
        dbm.freeConnection(conn, pstmt);
    }
    return ret;
}
```

batch() 적용 소스

```
public class RpCubridinsert
{
    --- 생략 ---
    List<RpCllctVO> cllctVOs = new ArrayList<>();
    // 수집한 데이터 조회
    cllctVOs = dbUtil.getCllctSelectList(mode, sDate, eDate, runType);

    // cllctVOs 수집 List 데이터를 가지고
    // insert 모듈 호출
    dbUtilCu.insertCllctCubridData(cllctVOs);
    --- 생략 ---
}

// 수집한 데이터 조회
cllctVOs = dbUtil.getCllctSelectList(mode, sDate, eDate, runType);

dbUtilCu.cllctBatchData(cllctVOs);

public int insertCllctCubridData(RpCllctVO cllctVO) throws SQLException,
UnsupportedEncodingException
{
    DBManagerCu dbm = DBManagerCu.getInstanceCu();
    Connection conn = dbm.getConnection();

    PreparedStatement pstmt = null;
    int ret = 0;
```

```
    String sqlText;

    sqlText = " INSERT INTO TB_INSEQCMS_BBS_DATA "
            + " (            "
            + "     BBS_ID          "
            + "     ,SEQ            "
            + "   -- 등등 생략 --    "
            + " )                    "
            + "     VALUES          "
            + " (             "
            + "     'news-media'    "
            + "     ,?              "
            + "   -- 등등 생략 --    "
            + ")                         ";

    pstmt = conn.prepareStatement(sqlText);


    for(RpCllctVO cllctVO: cllctVOs)
    {

        try
        {

            pstmt.setInt(1,    cllctVO.getBitnaraSeq());
                        -- 등등 생략 --

            // Batch Job으로 모은다.
            pstmt.addBatch();

        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
    // Batch Job으로 모은걸 한번에 실행한다.
    pstmt.executeBatch();

    try {
        pstmt.close();
        psFileTmt.close();
        psMariaTmt.close();
    } catch (Exception e2) {e2.printStackTrace();}

    try {
        connCu.close();
        conn.close();
    } catch (Exception e2) {e2.printStackTrace();}
```

```
}
```

중요 내용

> addBatch()를 사용하면 SQL문장을 LIST에 올려 놓고 executeBatch()를 사용 할 때 비로서 실제 DataBase에 저장 또는 변경 되는 것입니다.

소스 출처

> 위 소스는 내부 프로젝트 P***(기업)에서 검증이 되었습니다.

Tip

> 10만건 이상 List에 올렸다가 한번에 실행하면 오류가 날수도있다.

```
    if( (count % 100000) == 0){
        System.out.println(count + "건 처리중");
        pstmt.executeBatch();
    }
```

> for문안에 addBatch() 를 하고 카운터를 늘린다음 10만건이 되면 바로 Batch()를 실행해야한다.

Troubleshooting

Ref

- 한번에 여러 레코드 입력시 addBatch() 쓰세요

주레피, 도봉산핵주먹, .Batch(), javaDB, 데이터대용량처리

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:java:batch&rev=1588842876

Last update: **2022/03/10 19:52**