

# Mybatis

## 페이징 처리

# ▶ Mybatis 페이징 처리

## ✓ Mybatis RowBounds

Mybatis에서는 게시판 페이지 처리를 위한 객체인 RowBounds를 제공한다.

## ✓ RowBounds 객체 생성 방법

```
int offset = 10; // 뛰어 넘을 row 갯수, 0부터 시작
int limit = 25;  // 읽어 올 row 갯수

RowBounds rowBounds = new RowBounds(offset, limit);

session.selectList("boardMapper.selectAll", null, rowBounds);
// 넘겨줄 parameter 객체가 없으므로 null을 기입하여 session을 작성한다.
```

## ▶ Mybatis 페이징 처리

✓ 게시판 페이지 계산 식 → Controller

```
int listCount = 200;    // 게시글 총 갯수          --> 200 이라 가정
int currentPage = 1;    // 현재 페이지            --> 1이라 가정
int limit = 10;         // 한 화면에 보여질 페이지 개수 --> 10이라 가정
int maxPage;            // 마지막 페이지 (총 페이지 수)
int startPage;          // 현재 페이지에서 보여질 시작 페이지
int endPage;            // 현재 페이지에서 보여질 마지막 페이지

maxPage = (int)(Math.ceil(((double)listCount / limit)));

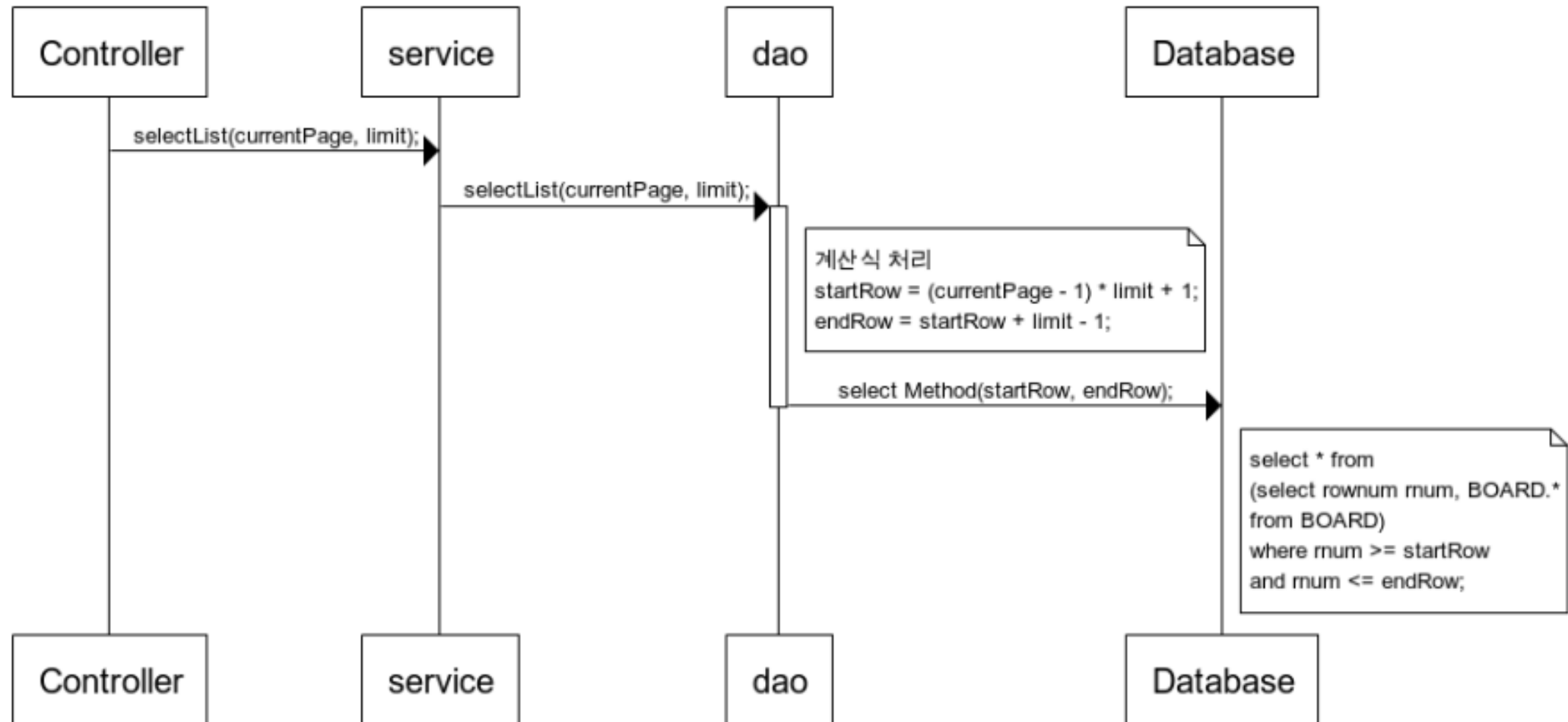
startPage = ((currentPage-1)/limit)*limit +1;

endPage = startPage + limit;

if(endPage > maxPage){
    endPage = maxPage;
}
```

## ▶ Mybatis 페이징 처리

### ✓ Mybatis 적용 전 일반적인 페이징 시퀀스 다이어그램



# ▶ Mybatis 페이징 처리

## ✓ Mybatis 적용 전 일반적인 페이징 처리 → DAO

### \* SQL Query 구문 생성

```
SELECT *  
FROM (SELECT ROWNUM RNUM, BOARD.*  
      FROM BOARD  
      ORDER BY BOARD_DATE DESC)  
WHERE RNUM >= ? AND RNUM <= ?
```

### \* 표시할 게시글 ROWNUM 순서 계산

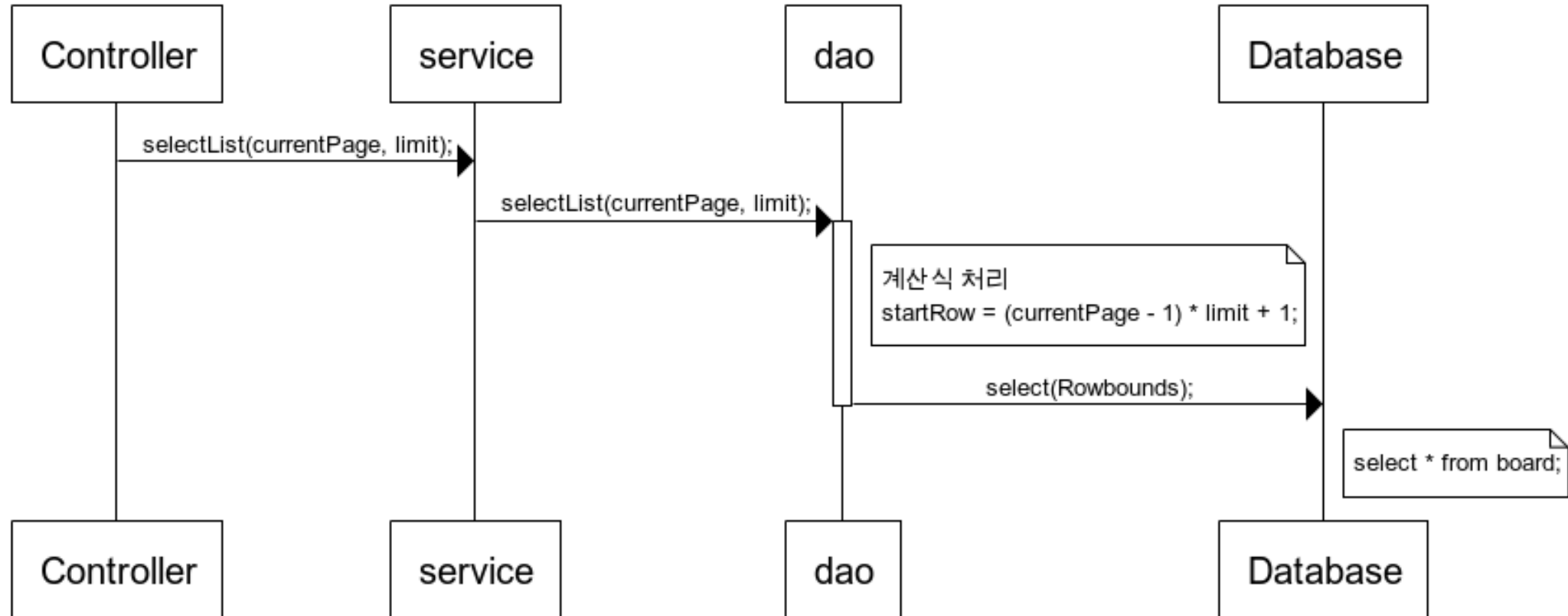
```
int startRow = (currentPage - 1) * limit + 1;  
int endRow   = startRow + limit - 1;
```

### \* 구문 실행

```
pstmt = conn.prepareStatement(sql);  
pstmt.setInt(1, startRow);  
pstmt.setInt(2, endRow);  
rset = pstmt.executeQuery();
```

## ▶ Mybatis 페이징 처리

### ✓ Mybatis 적용 후 페이징 시퀀스 다이어그램



# ▶ Mybatis 페이징 처리

## ✓ Mybatis 적용 후 페이징 처리 → DAO

### \* mapper에 SQL Query 구문 생성

```
<select>  
    SELECT * FROM BOARD  
</select>
```

### \* 구문 실행

```
ArrayList<Board> list = new ArrayList<Board>  
(session.selectList("boardMapper.selectAll", null, rowBounds));
```

### \* RowBounds 객체 생성

```
int offset = (currentPage - 1) * limit;  
RowBounds rowBounds = new RowBounds(offset, limit);
```

## ▶ Mybatis 페이징 처리

### ✓ ROWNUM VS RowBounds

	ROWNUM / ROW_NUMBER	RowBounds
장점	대량의 데이터도 빠르게 페이징 처리를 하여 가져올 수 있다.	구현이 쉽고, 코드의 유지보수가 간편하다.
단점	페이징 처리를 구현하기 위한 코드가 복잡하다.	대량의 데이터를 사용할 경우 수행속도가 느다.