

#Section10

[Tips] PL/SQL 활용

데이터 생성

테스트에 사용할 데이터를 생성해 봅니다.

사용 PL/SQL

- 커서 For Loop
- Collection - Nested Table
- BULK COLLECT INTO

```
-- 주문서생성
declare
  r_cst_info cst_info%rowtype;
  r_menu menu%rowtype;
  v_num number(10); -- 주문수량
begin

  -- for 구문
  FOR indx IN 1 .. 100000
  Loop
    -- 고객정보
    select *
    into r_cst_info
    from (
      select *
      from cst_info
      order by DBMS_RANDOM.RANDOM
    )
    where rownum=1
    ;

    -- 메뉴정보
    select *
    into r_menu
    from (
      select *
      from menu
      order by DBMS_RANDOM.RANDOM
    )
    where rownum=1
    ;

    v_num := ROUND(DBMS_RANDOM.VALUE(1, 10));
    -- 주문서를생성한다.
    insert into real_ord( ord_seq, cst_id, mnu_id, mnu_size, mnu_ice,
      qty, price, total_price, point_use, point_add)
    values( 1, r_cst_info.cst_id, r_menu.mnu_id, 'N', 'N',
      v_num , r_menu.mnu_price, v_num * r_menu.mnu_price , 0, 0
    );
  End Loop;

  commit;
end;
```

```
-- 주문서생성
declare
  -- 고객정보 Nested Table - Collection
  Type NT_cst_info is table of cst_info%rowtype;
  r_cst_info NT_cst_info := NT_cst_info();

  -- 메뉴정보 Nested Table - Collection
  Type NT_menu is table of menu%rowtype;
  r_menu NT_menu := NT_menu();

  -- 랜덤숫자
  cst_num number(10);
  menu_num number(10);
begin

  -- 고객정보 입력
  select *
  BULK COLLECT INTO r_cst_info
  from cst_info order by cst_id;

  -- 메뉴정보 입력
  select *
  BULK COLLECT INTO r_menu
  from menu order by mnu_id;

  -- 주문서 생성
  FOR indx IN 1 .. 100000
  Loop
    -- 랜덤 함수 실행
    cst_num := ROUND(DBMS_RANDOM.VALUE(1, 5));
    menu_num := ROUND(DBMS_RANDOM.VALUE(1, 5));

    -- 주문서를생성한다.
    insert into real_ord( ord_seq, cst_id, mnu_id, mnu_size, mnu_ice,
      qty, price, total_price, point_use, point_add)
    values( 1, r_cst_info(cst_num).cst_id, r_menu(menu_num).mnu_id, 'N', 'N',
      cst_num, r_menu(menu_num).mnu_price, cst_num * r_menu(menu_num).mnu_price , 0, 0
    );
  End Loop;

end;
```

데이터 이관 (Migration)

데이터 이관(Migration)에서도 PL/SQL 은 강력한 힘을 발휘합니다.

시나리오 : 매일 새벽 어제까지의 주문서 데이터를 기준으로 분석 데이터를 생성한다.

-- 분석데이터 , 테이블 생성

```
create table DA_ORDER(  
  ord_no number(10),  
  ord_seq number(5),  
  cst_id varchar2(20),  
  name varchar2(20),  
  mobile varchar2(20),  
  mnu_id varchar2(20),  
  mnu_type varchar2(20),  
  mnu_nm  varchar2(20),  
  mnu_price number(10),  
  qty number(10),  
  price number(10),  
  total_price number(10),  
  point_use number(10),  
  minor_yn varchar2(10), -- 미성년자여부  
  minor_alcohol_yn varchar2(10), -- 미성년자에게 술판매여부  
  reg_day date  
);
```

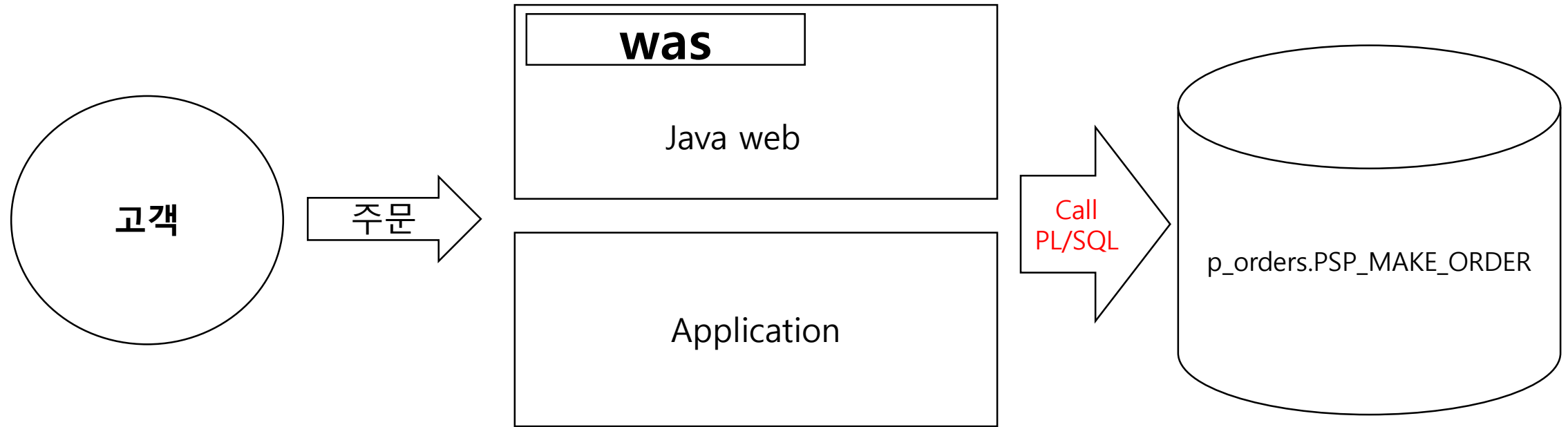
-- 기본쿼리

```
select a.ord_no, a.ord_seq, a.cst_id,  
       b.name, b.mobile, a.mnu_id,  
       c.mnu_type, c.mnu_nm, c.mnu_price,  
       a.qty, a.price, a.total_price, a.point_use, a.reg_day  
from real_ord a, cst_info b, menu c  
where a.cst_id=b.cst_id and a.mnu_id=c.mnu_id  
;
```

Java - Call PL/SQL

외부에서 DB 패키지 사용하기

시나리오 : Java 웹 개발에서 PL/SQL 을 호출하여 사용한다.



Java - Call PL/SQL

```
import java.sql.*;

/**
 * JDBC 이용 PL/SQL 사용하기
 * by IT늦공 김부장
 */

declare v_rtn varchar2(20); v_msg varchar2(200); begin P_ORDERS.PSP_MAKE_ORDER('C001', v_rtn, v_msg); dbms_output.put_line(v_rtn); dbms_output.put_line(v_msg); end;
*
*/

public class JdbcOraclePLSQL {

    public static void main(String[] args) throws SQLException {
        // TODO Auto-generated method stub

        String driver = "oracle.jdbc.driver.OracleDriver";
        String dbURL="jdbc:oracle:thin:@localhost:1521:xe";
        String user_id="User_PLSql";
        String user_pw="User_PLSql";
        String qry="";
        int result =0;

        Connection conn = null; // DB 에 connection 된 객체를 저장
        Statement stmt = null; // connection 객체에 실행문을 던지는 역할(창구)
        ResultSet rs = null; // select 결과값을 받아옴
        CallableStatement callableStatement = null; // PL/SQL 실행을 위한 인터페이스 CallableStatement 사용

        try {

            /* Driver Loading */
            Class.forName(driver);
            /* Oracle Connection
             * 여기까지만, 실행해보고, 실제 오라클 접속이 되었는지 체크
             */
            conn = DriverManager.getConnection(dbURL, user_id, user_pw);
            /* jdbc 는 기본적으로 자동커밋을 한다. 자동커밋 해제 */
            conn.setAutoCommit(false); // 자동 커밋 해제

            stmt = conn.createStatement();

            System.out.println("Connection Success");

            /* Statement - select, insert, update, delete */
        }
    }
}
```

Java - Call PL/SQL

```
/* Result Set , Print */
qry = "select * from menu";
rs = stmt.executeQuery(qry);

while(rs.next()) {
    System.out.printf("MNU_ID : %s, MNU_TYPE : %s, MNU_NM : %s \n",
        rs.getString("mnu_id"),
        rs.getString("mnu_type"),
        rs.getString("mnu_nm"));
}

callableStatement = conn.prepareCall("{call p_orders.PSP_MAKE_ORDER(?,?,?)}");
callableStatement.registerOutParameter(1, Types.VARCHAR);
callableStatement.registerOutParameter(2, Types.VARCHAR);
callableStatement.setString(3, "C002");
callableStatement.execute();

String strRtnCode = callableStatement.getString(1);
String strRtnMsg = callableStatement.getString(2);

System.out.println("strRtnCode > "+strRtnCode);
System.out.println("strRtnMsg > "+strRtnMsg);

// Transaction 처리는 Application 에서
if("0".equals(strRtnCode)){
    conn.commit(); // 자동커밋 사용
}else{
    conn.rollback(); // 자동커밋 사용
}

}catch (Exception e) {
    System.out.println("Error =>"+e);
    conn.rollback();
}finally {
    /* Close */
    try {
        if(rs != null) rs.close();
        if(stmt != null) stmt.close();
        if(callableStatement != null) callableStatement.close();
        if(conn != null) conn.close();
    }catch (Exception e2) {
    }
}
}
}
```

스케줄러 (DBMS_SCHEDULER)

참고 https://docs.oracle.com/en/database/oracle/oracle-database/12.2/arpls/DBMS_SCHEDULER.html#GUID-73622B78-EFF4-4D06-92F5-E358AB2D58F3

<https://oracle-base.com/articles/10g/scheduler-10g>

SCHEDULER

- Application (java, php..)
- Crontab (Linux , unix ..)
- Oracle Scheduler

▪ DBMS_JOB

- Job 하나를 통해 스케줄링 작업
- 특정시간이 아닌 실행주기(interval)만 가능

▪ DBMS_SCHEDULER (10g new feature)

- 작업 (DBMS_SCHEDULER.create_job)
 - 단독으로 실행 가능
 - 프로그램, 일정을 이용하여 작업 가능
 - `select * from user_scheduler_jobs ;`
- 프로그램 (DBMS_SCHEDULER.create_program)
 - `select * from user_scheduler_programs ;`
- 일정 (DBMS_SCHEDULER.create_schedule)
 - `select * from user_scheduler_schedules ;`