개발자 역량 강화를 위한 - **SYS4U 기술 능력 평가 방안**

**1. 도입 취지**

- 직급별 요구기술 명시

- 개발자 기술역량 배양

- 기술 기반의 품질관리 중요성 고취

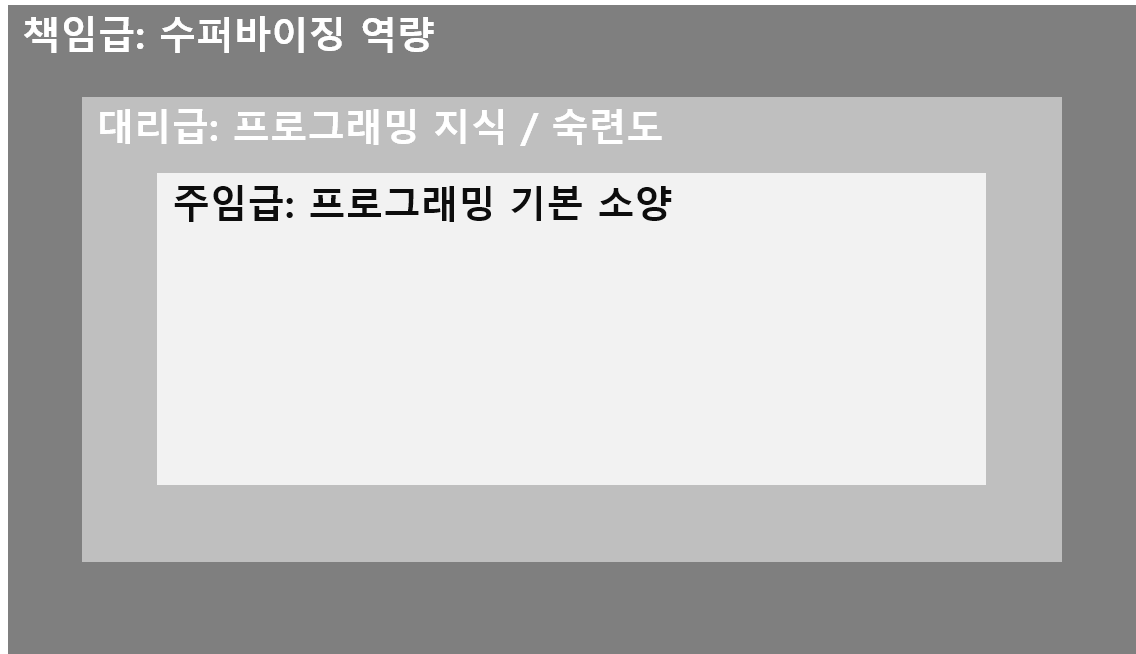
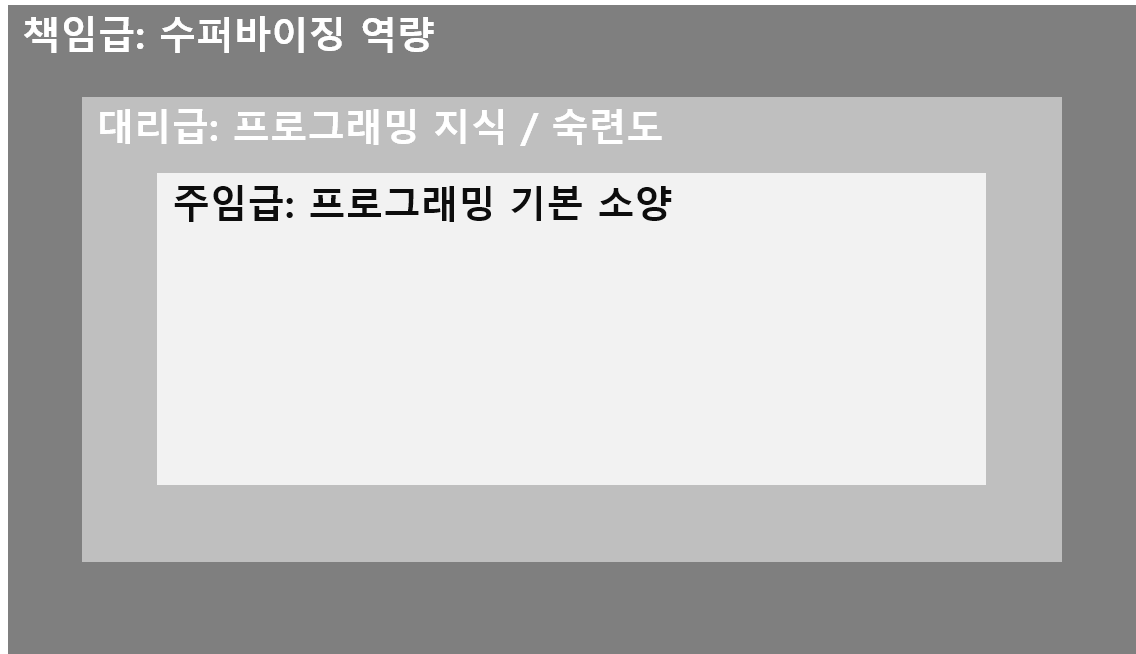
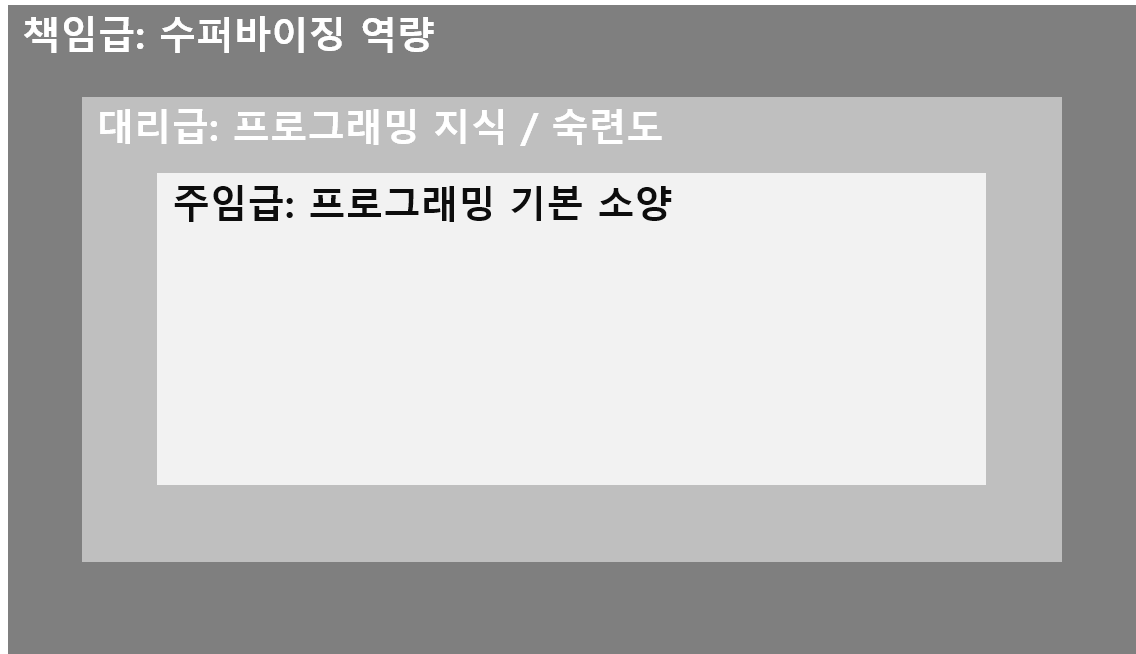
**2. 테스트 대상**

- 주임 진급 예정자 이상 전 개발자

- 단, 팀장, 실장, 소장 및 임원은 테스트 대상에서 제외

**3. 방안**

- 주임, 대리, 책임, 수석 등 직급별 테스트 진행

 - 1년에 최소 1회 최대 3회까지 테스트 수검 가능. 1회라도 테스트 통과 시 동일 직급 내에서는 추가 테스트 없음

- 2018년은 시행 첫해이므로 모든 개발직군 직원 대상으로 테스트 실시

- 진급 예정자는 진급 1년 전에 테스트를 수검해야 승진/연봉협상 가능   
(진급 예정자는 경영지원실에서 사전 고지)

- 경력직 신규 입사자 중 상반기 입사자는 당해에 테스트 수검 (입사 시 고지 필요)

- 테스트 종류 / 시간

|  |  |  |
| --- | --- | --- |
|  | 종류 | 시간 |
| 주임 | 주임급 문항 100% | 1시간 30분 |
| 대리 | 주임급 문항 20%  대리급 문항 80% | 2시간 |
| 책임 | 주임급 문항 10%  대리급 문항 20%  책임급 문항 70% | 3시간 |
| 수석 | 책임 진급자와 동일 | 2시간 30분 |

- 테스트 일정은 개별 요청 후 조율 - 일정이 겹치지 않는 한 상시 가능

- 테스트 기준, 참고서적, 예제 문항 등 사전 고지

- 테스트용 노트북은 본사에서 제공

- 테스트 과정에서의 인터넷 사용은 JAVA API, Spring API 등 API 검색으로 제한 (블로그 등 검색 불가)

- 테스트를 통과하지 못할 경우 향후 1년간 승진 불가, 연봉 동결. 다음 해 재수검 필수

**예외사항**

1. 2018년 진급 예정자만 예외적으로 2018년 초 승진/연봉협상 후 2018년 내에 테스트 수검. 탈락 시 2019년 연봉 동결
2. 2019년 진급 예정자는 2018년 테스트 시 승진 직급 대상 테스트 수검. 그 외는 2018년 당시 직급 대상 테스트 수검.  
   예1) 2018년 당시 주임 2년차: 대리급 시험 수검  
   예2) 2018년 당시 대리 1년차: 대리급 시험 수검  
   예3) 2018년 당시 사원 2년차: 주임급 시험 수검

**별첨: 대상별 테스트 준거 및 예제 문제**

**1. 주임 (진급자) 대상: 프로그래머 기본 소양 테스트**

- JAVA API를 이용하여 문자열을 조작할 수 있는가

- JAVA Collection Framework를 이용할 수 있는가

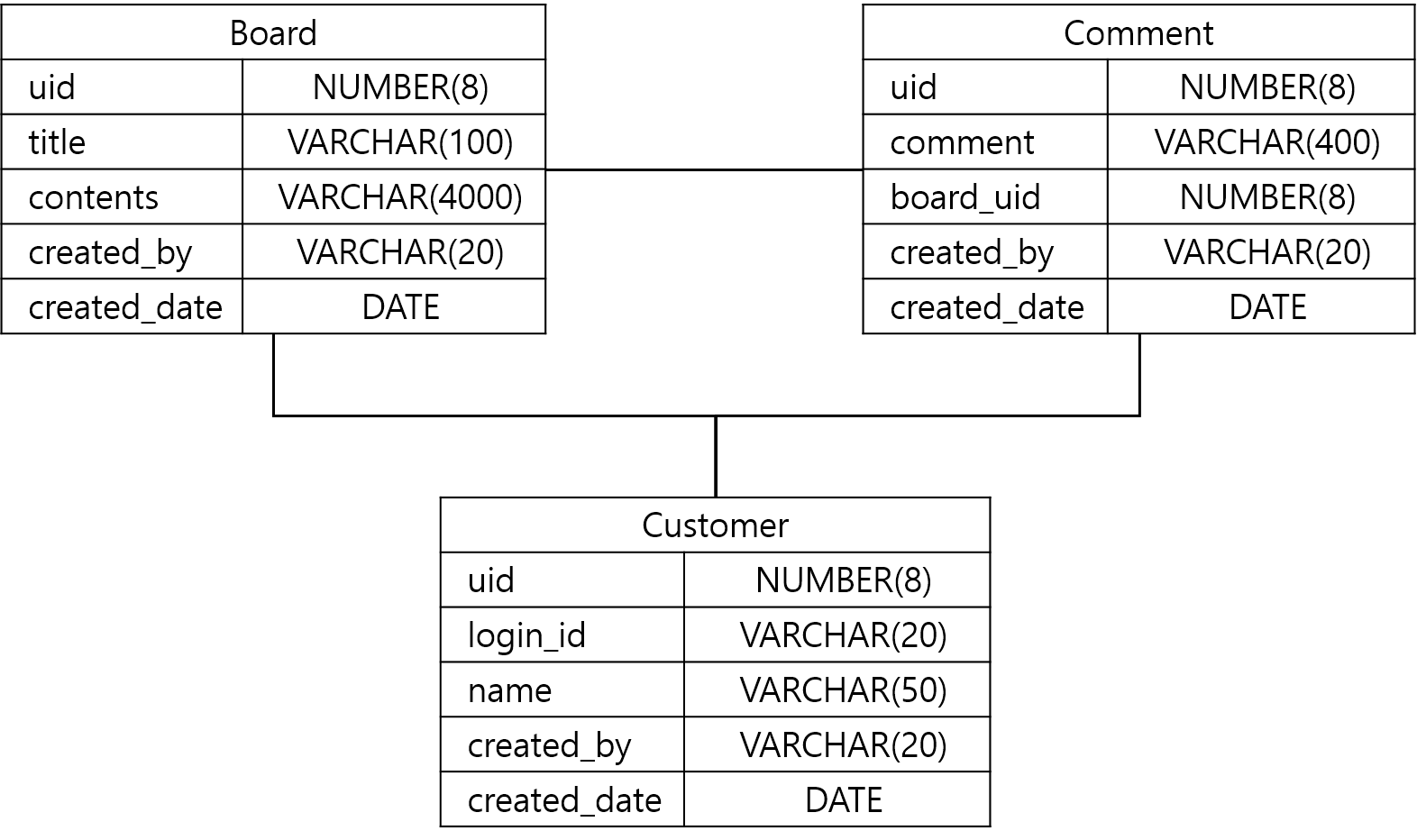
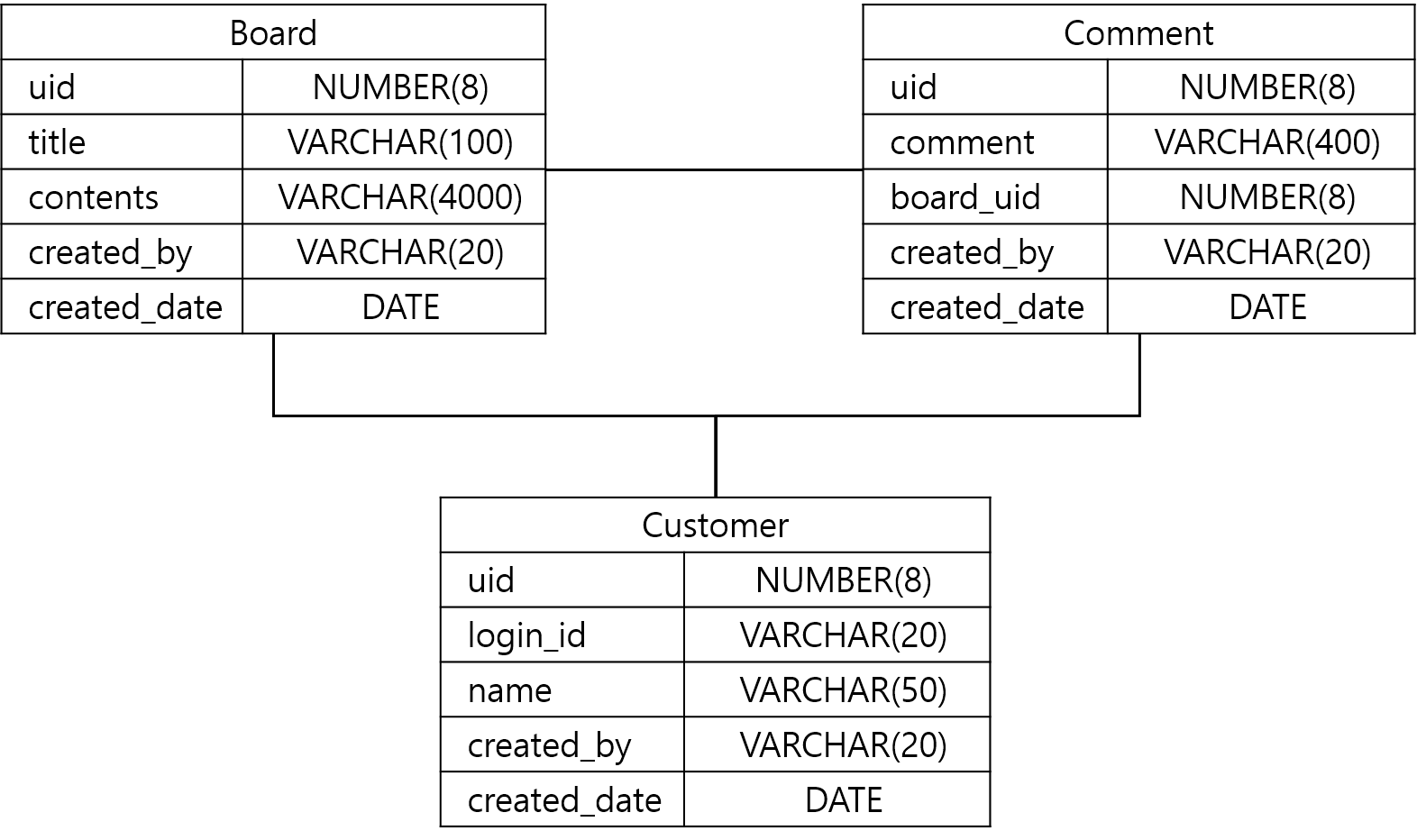
- JDBC를 이용하여 기본적인 SQL을 작성할 수 있는가

- 기본적인 Oracle Function을 이용할 수 있는가.

- null에 대한 검증과 처리를 제대로 할 수 있는가.

- 소스 정적 분석 결과를 바탕으로 코드를 수정할 수 있는가

문제 예시)

1. 하나의 문자열 내에서 동일한 문자가 두 개 이상 존재하는 경우 해당 문자에 대해 [**문자:개수**]의 형식으로 출력하는 메소드 **public void printDuplicatedChars(String value)** 를 작성하라. 하나만 존재하는 문자인 경우 출력하지 않는다. 단, 입력 데이터가 null인 경우 IllegalArgumentException을 throw한다.   
    4  
   입력 예) 가나다라가나가다마바  
   실행 결과)   
   가:3  
   나:2  
   다:2
2. 하나의 문자열이 회문인지 확인하는 메소드 **public boolean isPalindrome(String value)**를 작성하라. 회문이란 거꾸로 뒤집어도 원래 문자열과 동일한 문자열을 말한다. 단, 입력 데이터가 null인 경우 IllegalArgumentException을 throw한다.   
     
   입력 예) 가나다라  
   리턴 값) false  
     
   입력 예) 나다라다나  
   리턴 값) true  
     
   입력 예) 1221  
   리턴 값) true
3. 주어진 문자열 내에서 반복되는 문자열을 제거하는 메소드 **public String removeDuplicated(String value)**를 작성하라. 단, 입력 데이터가 null인 경우 IllegalArgumentException을 throw한다.   
   입력 예) bananas  
   리턴 값) bans  
     
   입력 예) 가나다가나라가나마  
   리턴 값) 가나다라마
4. 주어진 두 개의 문자열의 구성요소가 동일한지 확인하는 메소드 **public boolean isIdenticalElements(String a, String b)**를 작성하라. 문자열의 길이는 결과에 영향을 주지 않는다. 단, 입력 데이터가 null인 경우 IllegalArgumentException을 throw한다.   
   입력 예) a = abcde / b = edbca  
   리턴 값) true   
     
   입력 예) a = abcde / b = abc1e  
   출력 값) false
5. 다음과 같이 세 개의 테이블이 있다. (예시를 위해 간략하게 구성함)  
   위 테이블에 대해 아래 조건과 출력 데이터에 부합하는 SQL을 작성하라.   
   조건1: 게시물(Board)에는 0개 이상의 댓글(Comment)가 달릴 수 있다.  
   조건2: 게시물과 댓글은 고객(Customer)만이 작성할 수 있다.  
   조건3: 게시물 및 댓글을 작성한 고객의 login\_id는 created\_by 컬럼에 저장된다.  
   **조회 조건:** 게시물 번호(Board.uid)가 1인 경우  
   **출력 데이터**: 게시물 번호 / 게시물 타이틀 / 게시물 내용 / 게시물 작성자 이름 / 게시물 작성일  
    댓글 번호 / 댓글 내용 / 댓글 작성자 이름 / 댓글 작성일
6. 위 테이블에 대해, 댓글을 10개 이상 작성한 사람의 목록을 출력하는 SQL을 작성하라.   
   **출력 데이터:** 댓글 작성자 이름 / 댓글 개수
7. 다음 Oracle 데이터 타입과 관련된 내용에 대해 설명하라.
   1. CHAR와 VARCHAR2 타입의 차이
   2. DATE 타입을 VARCHAR2 타입으로 변환하기 위해 TO\_CHAR 함수를 이용하는 방법
   3. TO\_CHAR 함수나 TO\_DATE 함수를 이용할 때 DATE\_FORMAT을 입력하지 않은 경우 처리되는 방식
8. PRIMARY KEY와 UNIQUE KEY에 대해 설명하고 차이점을 기술하라.
9. 숫자를 총 10자리의 문자로 변환하려 한다. 남는 자릿수는
10. 0으로 채우려 할 때, 어떤 함수를 이용하면 되나?  
    예) 145 -> 0000000145
11. CU\_CUSTOMER 테이블에는 LOGIN\_ID라는 UNIQUE KEY 컬럼이 존재한다고 하자. 이 때, CU\_CUSTOMER 테이블 내에 ‘sys4u’라는 LOGIN\_ID를 갖는 행이 있는지 여부를 확인하는 쿼리를 작성하라.   
    COUNT 함수를 이용하는 방법과 EXISTS 키워드를 이용하는 방법 두 가지를 각각 사용하고, 그 두 방법의 차이에 대해 설명하라.
    1. **주임급 개발자 평가 문항 항목 및 상세 레퍼런스**

(1) java.lang.String / java.lang.StringBuilder

레퍼런스: <https://docs.oracle.com/javase/tutorial/java/data/strings.html> 및 그 이하 링크

- java.lang.String 클래스 사용법

- 숫자를 문자열로, 문자열을 숫자로 변환하는 방법

- StringBuilder 사용법

(2) java.util.List

레퍼런스: <https://docs.oracle.com/javase/tutorial/collections/interfaces/list.html>

- java.util.List 인터페이스 사용법

- java.util.List에서 특정 인스턴스와 동일한(equals) 엘리먼트의 위치를 구하는 방법

- JDK8부터 제공하는 Stream API 사용법

- java.util.Iterator 사용법

- Collections 클래스가 제공하는 각종 알고리즘들

(3) java.util.Map

레퍼런스: <https://docs.oracle.com/javase/tutorial/collections/interfaces/map.html>

- java.util.Map 인터페이스 사용법

- java.util.Map.Entry를 이용한 Map 사용법

- java.util.Map에서 key 목록 얻기 / value 목록 얻기 / key-value 쌍 목록 얻기

- java.util.Map에서 특정 value 값과 연결된 모든 value 찾기

(4) Exceptions

레퍼런스: <https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html> 및 그 이하 링크

<http://tutorials.jenkov.com/java-exception-handling/exception-wrapping.html>

- Java SDK에서 제공하는 Exception 처리 방법

- 사용자 정의 Exception 만들기

- Exception throw하는 방법 / Exception을 wrapping하는 방법

(5) SQL 기본

레퍼런스: <https://www.w3schools.com/sql/> 및 이하 링크

연구소에서 제공한 SQL 관련 문서

- Row에 대한 Create / Read / Update / Delete 기본 문법

- Where 조건

- Join 종류 및 방법

- Inline-View

- Scalar Query (<https://docs.oracle.com/cd/B19306_01/server.102/b14200/expressions010.htm>)

- Primary Key / Unique Key

- Index 기본

**2. 대리 (진급자) 대상: 프로그래머로서의 지식/숙련도 테스트**

- JAVA SDK의 다양한 API를 제대로 이해하고 이를 이용하여 구현할 수 있는가.

- 각종 검색/정렬 알고리즘에 대해 이해하고 구현할 수 있는가.

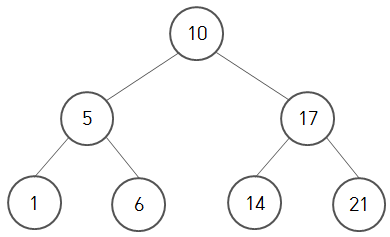
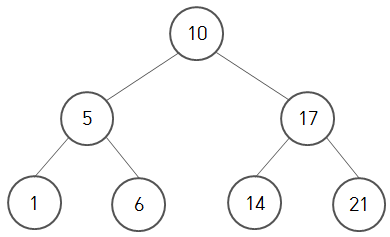
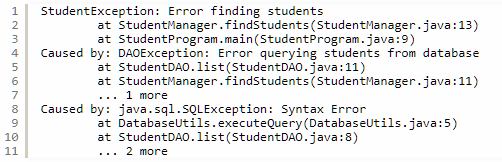
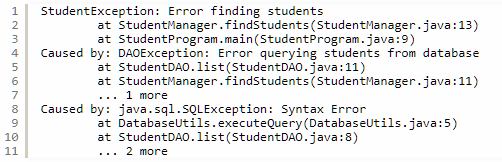
- Exception에 대해 이해하고 있으며 Exception Stacktrace를 통해 원인을 추론할 수 있는가.

- 스프링 프레임워크를 이용하여 프로그램을 구현할 수 있는가.

- Oracle Procedure를 작성할 수 있는가.

- 데이터베이스 트랜잭션에 대해 명확하게 이해하고 있는가.

문제 예시)

1. java.util.Vector 클래스와 java.util.ArrayList 클래스, 그리고 java.util.LinkedList의 차이에 대해 기술하라. 그리고 Java5 이후로 java.util.Vector를 사용하지 말아야 하는 이유에 대해서도 설명하라.
2. java.util.ArrayList 클래스와 java.util.LinkedList 클래스의 중, 엘리먼트의 추가와 삭제가 빈번히 일어나는 경우 어떤 List가 유리한지 명시하고 그 이유를 설명하라.
3. 병합 정렬(merge sort)은 전형적인 분할 정복 알고리즘으로 다음과 같은 절차를 따른다.  
   a. 리스트의 길이가 0 또는 1이면 이미 정렬된 것으로 간주한다.  
   b. 정렬되지 않은 리스트를 절반으로 잘라 비슷한 크기의 두 리스트를 만든다.  
   c. 각 부분 리스트에 대해 재귀적으로 병합 정렬을 수행한다.  
   d. 두 부분 리스트를 다시 하나의 정렬된 리스트로 병합한다. (정렬하며 병합한다.)  
   JAVA 언어를 이용하여 병합 정렬 메소드 **public void mergeSort(List<Integer> list)**를 구현하라.
4. 이진 탐색 트리(Binary Search Tree)란 아래과 같이 하나의 노드가 두 개의 자식 노드를 갖는 이진 트리를 이용하여 탐색을 수행하는 검색 알고리즘을 의미한다.  
   이진트리는 다음과 같은 성질을 갖는다.  
   a. 모든 원소는 서로 다른 유일한 키를 갖는다.  
   b. 왼쪽 서브 트리에 있는 원소의 키는 그 루트의 키보다 작다.  
   c. 오른쪽 서브 트리에 있는 원소의 키는 그 루트의 키보다 크다.  
   d. 왼쪽 서브 트리와 오른쪽 서브 트리도 이진탐색 트리다.  
   이진 탐색 트리는 최적의 경우 O(log2N)의 시간 복잡도를 보이기 때문에 검색 알고리즘으로 자주 사용된다.  
   이진 탐색 트리를 이용하여 이진 트리 내에 특정 int 값이 존재하는지 확인하는 **public boolean exists(BinaryTree tree, int value)** 메소드를 작성하라. 이 때 **BinaryTree** 클래스도 같이 작성해야 한다.
5. 다음 Exception StackTrace를 보고 어떤 코드에서 어떤 문제가 발생했는지 기술하라.  
   
6. 스프링 웹 MVC를 이용하여 웹 어플리케이션을 개발할 때, “특별한 이유가 없으면” Controller / Service / DAO에서 Exception을 catch하지 않을 것을 권장한다. 이와 관련하여 다음 세 가지에 대해 설명하라.  
   a. Exception을 catch하지 말 것을 권장하는 이유는 무엇인가?  
   b. Exception을 catch하지 않아도 어플리케이션이 정상적으로 동작하는 이유는 무엇인가?  
   c. Exception을 catch해야 하는 “특별한 이유”는 무엇인가?
7. PMD나 SonarQube와 같은 소스 정적 분석 도구들은 Exception을 catch했을 때 printStackTrace로 출력하면 안 된다고 규정하고 있다.   
   a. 그 이유는 무엇인가?  
   b. catch한 Exception을 어떻게 처리해야 하는가?
8. 데이터베이스 트랜잭션은 정의상 ACID해야 한다. ACID는 Atomic, Consistent, Isolated, Durable에서 따온 단어이다. ACID 각각에 대해 설명하라.
9. 특정 비즈니스 로직에 대해 데이터베이스 트랜잭션 관리를 해야 할 때 다음 정보들을 지정해야 한다. 이들에 대해 설명하라.  
   a. Propagation  
   b. Isolation  
   c. Timeout  
   d. Read-Only
10. 스프링 프레임워크의 Transaction Manager와 Thread는 어떤 관계가 있는지 기술하라.
11. ORACLE의 SEQUENCE에 대해 설명하라. CURRVAL과 NEXTVAL의 차이에 대해 설명하고, CURRVAL을 얻기 위한 제한 조건을 설명하라.
12. ST\_MENU 테이블에는 MENU\_ID와 PARENT\_MENU\_ID 컬럼이 있다. MENU\_ID는 자신의 메뉴 번호를, PARENT\_MENU\_ID는 부모 메뉴의 번호를 의미한다. (즉, ST\_MENU 테이블은 계층 구조라 할 수 있다.) 최상위 MENU\_ID가 ‘ROOT\_01’ 이라고 할 때, 전체 메뉴 계층 정보를 출력하는 SQL을 작성하라. 조회할 컬럼은 다음과 같다.  
    조회 대상 컬럼: MENU\_ID, MENU\_NAME, PARENT\_MENU\_ID, LEVEL
13. SUM OVER 분석함수를 사용하여 2018년 01월 고객실적을 지역별로 출력하는 쿼리를 작성하라.   
    테이블명: ORD\_ORDER  
    테이블 컬럼: 주문번호, 고객번호, 지역코드, 주문금액, 주문일자  
    출력컬럼: 지역코드, 고객번호, 고객 총 주문금액, 지역 총 주문금액
14. 2018년도 주문실적을, 월별로 출력하는 SQL을 작성하라. 단, 주문일자 컬럼은 YYYYMMDD의 VARCHAR2 형식이라고 가정한다.  
    테이블명: ORD\_ORDER  
    테이블 컬럼: 주문번호, 고객번호, 지역코드, 금액, 주문일자  
    출력 컬럼: 01월, 02월, 03월, ….., 10월, 11월, 12월
15. RANK 함수를 이용하여, 2018년 1월 주문액 TOP 10인을 조회하는 쿼리를 작성하라.  
    테이블명: ORD\_ORDER  
    테이블 컬럼: 주문번호, 고객번호, 지역코드, 금액, 주문일자

**2.1 대리급 개발자 평가 문항 항목 및 상세 레퍼런스**

(1) java.util.ArrayList / java.util.LinkedList / java.util.Vector

레퍼런스: <https://dzone.com/articles/arraylist-vs-linkedlist-vs>

- 특징, 차이점 사용법 등

(2) 정렬 알고리즘

레퍼런스: <http://www.vogella.com/tutorials/JavaAlgorithmsMergesort/article.html> - 병합정렬

<https://introcs.cs.princeton.edu/java/23recursion/> - 재귀호출

<https://en.wikipedia.org/wiki/Divide_and_conquer_algorithm> - 분할정복 알고리즘

- 병합정렬 알고리즘 개념 및 구현

- 재귀호출 방법

- 분할정복 알고리즘 이해

(3) 이진 검색 트리 (Binary Search Tree)

레퍼런스: <http://algorithms.tutorialhorizon.com/binary-search-tree-complete-implementation/>

- 이진 검색 트리 특징, 형상

- 삽입

- 조회

(4) Exception Stacktrace

레퍼런스: <http://www.codejava.net/java-core/exception/understanding-exception-stack-trace-in-java-with-code-examples>

- Exception Stacktrace의 이해

- Exception Wrapping을 통한 전달

- 바람직한 Exception Handling 방법

(5) Database Transaction and Spring Framework

레퍼런스: <https://www.tutorialspoint.com/spring/spring_transaction_management.htm> 및 연관 링크

(6) Oracle PL-SQL

레퍼런스: <http://www.oracle.com/technetwork/topics/o67sql-101793.html> - PL/SQL 작성 및 사용법

연구소에서 제공한 SQL 관련 문서

- PL/SQL 작성 방법

- 에러 발생시 처리 방법

**3. 책임 (진급자) 대상: 개발자 슈퍼바이징 가능 여부 테스트**

- 코드 리뷰를 수행할 수 있는가.

- 코드 리팩토링을 수행할 수 있는가.

- 특정한 상황에 적합한 디자인 패턴을 제안할 수 있는가.

- 적절한 커뮤니케이션 능력을 보유하고 있는가.

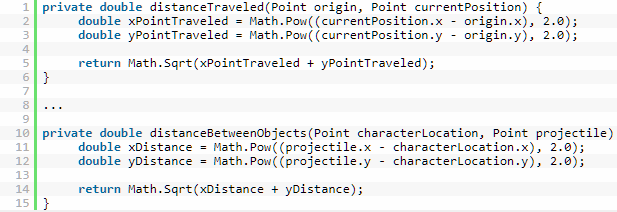
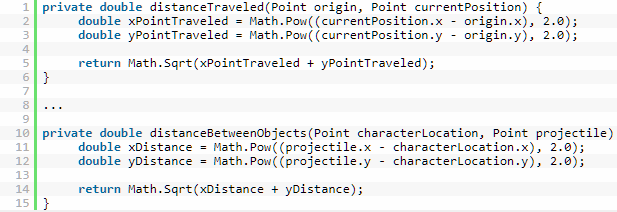
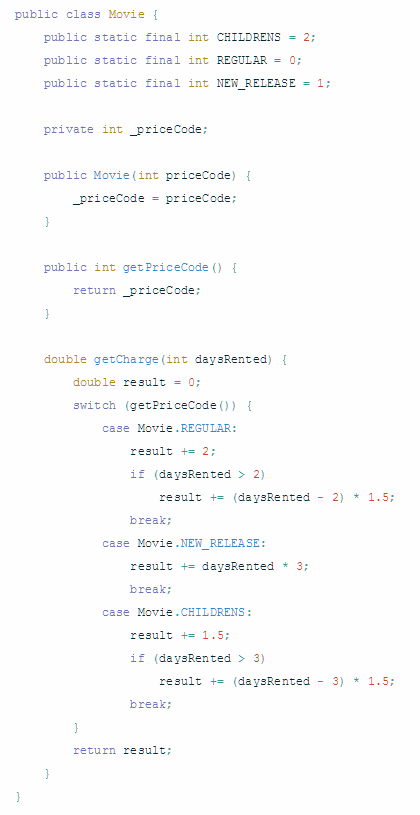
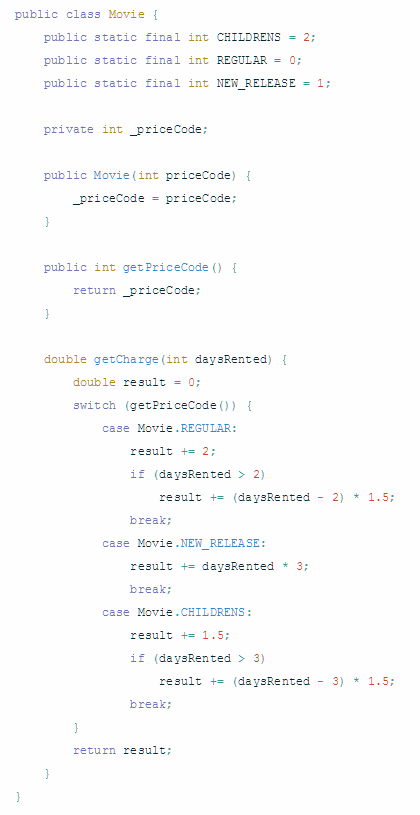
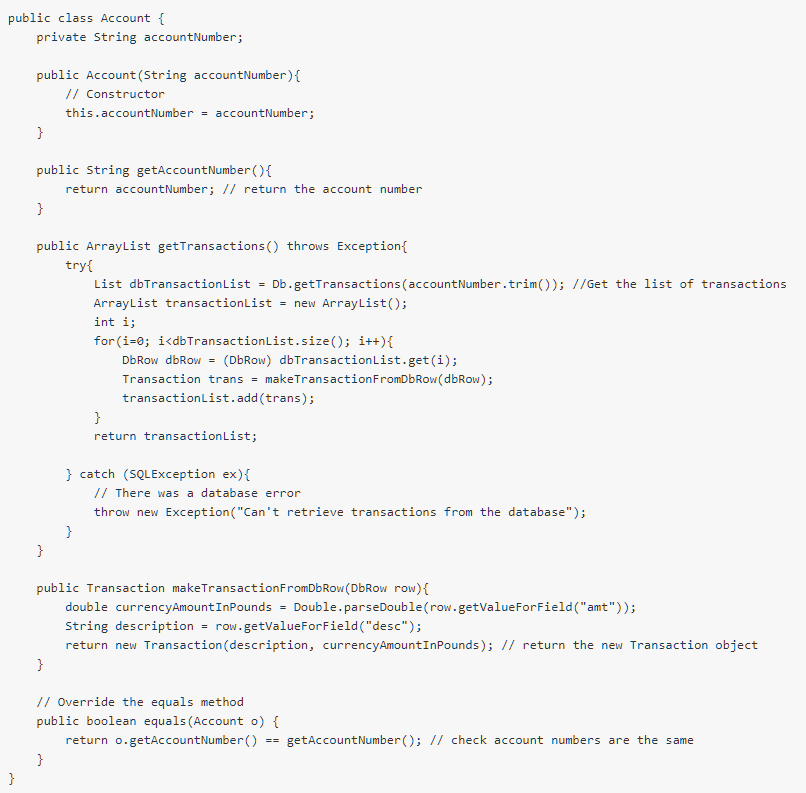
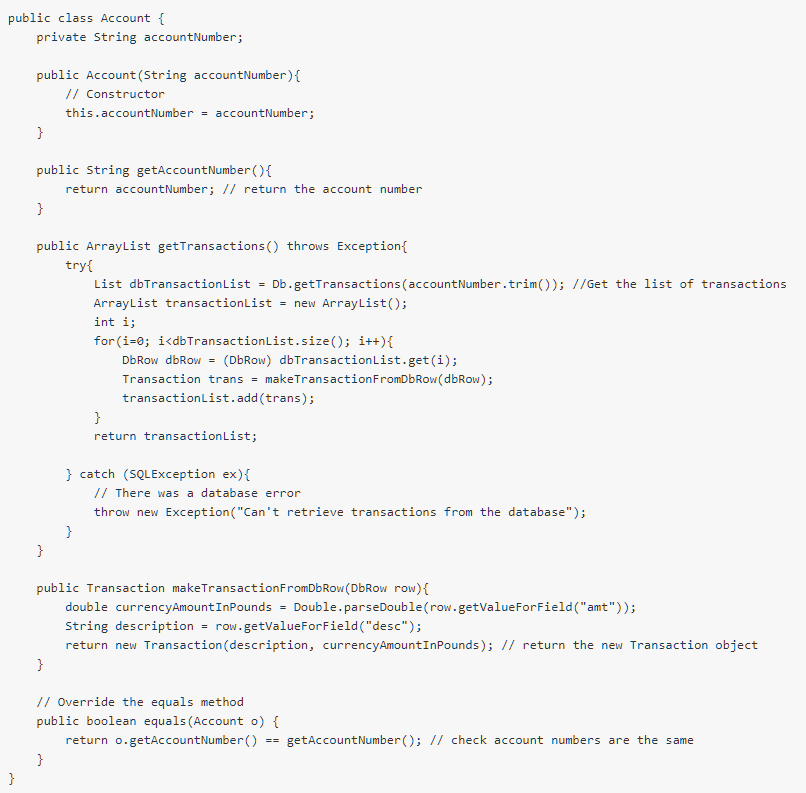
- 스프링 프레임워크 혹은 스프링 부트를 이용하여 어플리케이션 개발을 시작할 수 있는가.

- Lock과 Dead-Lock의 차이를 이해하고 있으며 Dead-Lock을 회피하는 방법을 알고 있는가.

- Thread-Safety에 대해 이해하고 있으며 Thread와 관련하여 어떤 문제가 발생할 수 있는지 알고 있는가.

- Explain Plan을 보고 SQL Tuning을 수행할 수 있는가.

문제 예시)

1. 코드 리팩토링이란 무엇이고, 언제 수행해야 하며, 왜 필요한지에 대해 기술하라.
2. 다음 코드를 리팩토링 하라. 참고로 Point는 2차원 좌표 정보(x,y)를 담고 있는 클래스이다.  
   
3. 다음 코드를 리팩토링하라.  
   https://diplograph.net/posts/a\_better\_approach\_to\_martin\_fowlers\_refactoring\_chapter\_1  
   
4. 다음 코드에서 잘못된 부분을 수정하라. 로직에는 틀린 부분이 없으며 정상적으로 컴파일된다고 가정한다. (https://codebox.net/pages/java-interview-question)  
   
5. 브라우저에서 <http://localhost:8080/index.do> 에 접속하면 Hello World를 출력하도록 SpringBoot 어플리케이션을 작성하라. SpringBoot의 버전은 중요하지 않다.
6. 브라우저에서 [http://localhost:8080/dual.do 에 접](http://localhost:8080/dual.do%20에%20접)속하면 1을 출력하도록 SpringBoot 어플리케이션을 작성하라. (5)에서 작성했던 프로젝트를 이용하되, 다음 조건에 부합하도록 한다.  
   a. Controller / Service / Mybatis Mapper / JSP를 각각 작성한다.  
   b. Mybatis Mapper는 SELECT 1 FROM DUAL를 실행하여 그 결과값을 리턴한다.  
   c. Service는 b.에서 작성한 Mybatis Mapper를 호출하고 그 결과값을 리턴한다.  
   d. Controller는 c.에서 작성한 Service를 호출하고 그 결과값을 JSP로 forwarding 한다.
7. 컴퓨팅 환경에서 Lock은 무엇이고 왜 필요한지 설명하라.
8. NESTED LOOP 조인과 HASH JOIN을 비교 설명하라. NESTED LOOP 조인을 사용해야 하는 경우와 HASH JOIN을 사용해야 하는 경우를 비교 설명하라.
9. 다음 EXPLAIN PLAN의 실행 순서를 기술하고 그 내용에 대해 설명하라.  
   ----------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost |

----------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 5 | 270 | 3 |

| 1 | NESTED LOOPS | | 5 | 270 | 3 |

| 2 | TABLE ACCESS BY INDEX ROWID| DEPT | 1 | 21 | 1 |

|\* 3 | INDEX UNIQUE SCAN | PK\_DEPT | 4 | | |

|\* 4 | TABLE ACCESS FULL | EMP | 5 | 165 | 2 |

----------------------------------------------------------------------------

1. 데이터베이스 인덱스에 대해 설명하고, 어떤 경우에 인덱스를 이용하는 것이 유리하고 어떤 경우에 불리한지를 설명하라.
2. 대량의 데이터를 LOAD하거나 UPDATE, DELETE할때 일반적으로 사용하는 loop문보다 수십배 이상 좋은 방법중에 하나인 BULK SQL이다. 일반적인 LOOP문 안에 있는 SQL은 매 LOOP마다 1회씩 수행되는데 BULK SQL은 LOOP없이 1번의 SQL으로 수행할 수 있다. BULK SQL 사용하는 방법을 설명하라.
3. ORACLE PROCEDURE에서 커서를 생성하고 사용하는 방법에는 세 가지가 있다. 각각의 방법을 기술하라.

**3.1 책임/수석급 개발자 평가 문항 항목 및 상세 레퍼런스**

(1) 리팩토링

레퍼런스: <http://www.yes24.com/24/goods/55147071> - 자바로 배우는 리팩토링 입문

- 리팩토링과 버그패치의 차이점 이해

- 1장 매직넘버를 기호 상수로 치환

- 2장 제어 플러그 삭제

- 5장 메소드 추출

- 7장 분류코드를 클래스(혹은 enum)로 치환

- 9장 분류코드를 상태/전략 패턴으로 치환

- 10장 에러코드를 예외로 치환

(2) 클린 코드

레퍼런스:

<http://www.yes24.com/24/goods/11681152?scode=032&OzSrank>=1 – 클린 코드

- 1장 깨끗한 코드

- 2장 의미있는 이름

- 3장 함수

- 4장 주석

- 5장 형식 맞추기

- 6장 객체와 자료구조

- 7장 오류처리

- 9장 클래스

- 부록II 동시성

(3) Spring Boot

레퍼런스: https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/ – SpringBoot

- Maven으로 Spring Boot 프로젝트 생성 / 실행하기   
([https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/#using-boot-maven](https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/))

- XML 대신 Java Config로 빈 설정하기  
([https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/#using-boot-importing-configuration](https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/))

- 웹 어플리케이션 개발을 위한 환경 설정하기  
([https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/#boot-features-spring-mvc](https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/))

- JSP 뷰를 이용하기 (<http://coding-slave.blogspot.kr/2016/01/web-spring-spring-boot.html>)

- Validation하기  
([https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/#boot-features-validation](https://docs.spring.io/spring-boot/docs/1.5.9.RELEASE/reference/htmlsingle/))

(4) 동시성 (Concurrency)

레퍼런스: <https://docs.oracle.com/javase/tutorial/essential/concurrency/> - Concurrency

- 프로세스와 스레드

- 스레드 사용 방법

- 동기화 방법

- Lock과 Dead Lock의 차이

- 불변 객체

- Executors와 ThreadPool

(5) Oracle Explain Plain

레퍼런스: <http://www.nextree.co.kr/oracle-explain-plan-sayong/> – Explain Plan

<http://www.dba-oracle.com/t_hash_join_vs_nested_loops_join.htm> - Hash Join vs Nested Loop

<http://www.dba-oracle.com/t_tuning_sql_nested_loops_joins.htm> - Using Nested Loop join

(6) Oracle Bulk SQL  
 레퍼런스: <http://www.dba-oracle.com/t_oracle_bulk_collect.htm>

(7) Oracle PL/SQL and Cursor  
 레퍼런스: <http://www.gurubee.net/oracle/plsql>