

Hanbit
RealTime
126

개발자를 위한 오라클 SQL 튜닝

이경오 지음



개발자를 위한
오라클 SQL 튜닝

이경오 지음



표지 사진 강수진

이 책의 표지는 강수진님이 보내 주신 풍경사진을 담았습니다.
리얼타임은 독자의 시선을 담은 풍경사진을 책 표지로 보여주고자 합니다.

사진 보내기 ebookwriter@hanbit.co.kr

개발자를 위한 **오라클 SQL 튜닝**

초판발행 2016년 2월 16일

지은이 이경오 / **펴낸이** 김태현

펴낸곳 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부

전화 02-325-5544 / **팩스** 02-336-7124

등록 1999년 9월 30일 제10-1779호

ISBN 978-89-6848-799-6 15000 / **평가** 14,000원

총괄 전태호 / **책임편집** 김창수 / **기획·편집** 정지연 / **교정** 이미연

디자인 표지/내지 여동일, 조판 최송실

마케팅 박상용, 송경석 / **영업** 김형진, 김진불, 조유미

이 책에 대한 의견이나 오타자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주세요.

한빛미디어 홈페이지 www.hanbit.co.kr / **이메일** ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2016 이경오 & HANBIT Media, Inc.

이 책의 저작권은 이경오와 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

지은이 **이경오**

광운대학교 컴퓨터소프트웨어학과를 졸업하였으며 2009년 키움증권 전산실에서 사회생활을 시작하였습니다. 이후 흥국생명 전산실에서 SM 업무를 수행하였고, KG모빌리언스에서 차세대 시스템 구축 프로젝트를 담당하였습니다. 현재는 국내 유일의 오픈소스 DBMS인 CUBRID의 벤더사인 (주)큐브리드에 합류하여 DBMS 컨설팅 업무를 수행하고 있습니다. 6년간의 소프트웨어 개발 경험과 DBMS 시스템에 대한 심도 있는 학습을 바탕으로 대한민국 오픈소스 DBMS의 확산과 발전을 위해 하루하루 최선을 다하고 있습니다.

보유 자격증

- SQL 개발자(국가공인 SQL Developer, 한국데이터베이스진흥원)
- SQL 전문가(국가공인 SQL Professional, 한국데이터베이스진흥원)
- 리눅스마스터 1급(Linux Master 1st, 한국정보통신진흥협회)

운영 블로그 <http://blog.naver.com/dbmsexpert>

저는 지난 2009년 9월 키움증권 전산실에서 IT 개발자 생활을 시작하였습니다. 업무를 진행하면서 정확한 고성능 SQL문을 작성하는 것이 프로그래밍 언어만큼이나 중요하다고 생각해 왔습니다. 그리하여 개발자로 일하면서도 SQL 튜닝 분야에 대한 심도 있는 학습을 진행해 왔습니다.


대다수 IT 개발자는 SQL문을 작성할 때 업무 요건에만 충족하면 작업을 멈추고 해당 SQL문을 실제 운영 환경에 적용합니다. 결과 집합은 충족하지만 성능을 고려하지 않은 SQL문이 하나둘씩 쌓여갈 때 시스템은 견잡을 수 없는 부하에 시달리게 됩니다. 정작 해당 SQL을 튜닝해야 할 DBA는 해당 업무를 잘 몰라서 SQL 튜닝을 하기가 쉽지 않습니다.

이러한 실무 환경이나 현실을 생각해 보면 해당 비즈니스 로직을 누구보다 잘 아는 사람인 개발자가 SQL 튜닝을 하는 것이 맞습니다. 저 또한 이러한 상황에서 SQL 튜닝 학습을 시작하였습니다. 시중에 있는 SQL 튜닝 관련 책을 보고 학습하면서 다음과 같은 힘든 점이 있었습니다.

- SQL 기초를 다루는 책은 개발자가 읽기에는 너무 쉽습니다.
- SQL 튜닝을 다룬 책은 개발자가 읽기에는 너무 어렵고, 책에 나온 내용을 실무 환경에서 실습 하기가 어렵습니다(권한 문제 등).

즉, 개발자에게 최적화된 SQL 튜닝 책이 시중에는 없었습니다. 이러한 이유로 SQL 튜닝을 학습하는 데 있어 커뮤니티에서 만난 스터디 그룹이나 인터넷 검색에 의존하였으며 학습하는 과정 또한 쉽지 않았습니다. SQL 튜닝 분야를 꼭 학습해야 하거나 관심이 있는 사람들이 보기에 가장 적합한 책이 있었으면 좋겠다고 판단하였습니다.

SQL문을 튜닝하는 데 화려한 UI를 가지고 있는 성능 모니터링 툴이나 고도의 SQL Tracing 툴이 꼭 필요한 것은 아닙니다(물론 있으면 많은 도움이 됩니다). 실행 계획만 보



고도 SQL문을 튜닝할 수 있습니다. 실행 계획은 개발자에게도 주어지는 권한입니다. 이 책의 내용은 대한민국 개발자의 업무 환경에 초점을 맞춘 책이며 이 책의 이론과 실습 내용을 차근차근 학습해 나간다면 누구나 SQL 튜닝을 할 수 있습니다.

이 책의 특징을 살펴보면 다음과 같습니다.

- 개발자에게 모든 초점과 난이도를 맞췄습니다.
- 기초 SQL문을 작성할 수 있는 사람이라면 누구나 이 책의 내용을 학습할 수 있습니다.
- DBA 권한이 반드시 필요한 부분이나 사용빈도가 극히 낮고 어려운 부분은 다루지 않습니다.
- 이 책에서 다루는 SQL 튜닝 기법은 실무에 바로 적용할 수 있으며, 이는 전체 SQL 튜닝의 80~90%를 차지합니다.

즉, 이 책에서 다루고 있는 이론과 실습을 모두 학습한다면 약간의 노력과 시간 투자로 엄청난 업무 효율을 발휘할 수 있습니다(필자의 경험입니다).

이 책에서 다루는 내용은 SQL 튜닝 분야에서 가장 기초적이면서 가장 널리 쓰이고 있는 내용입니다. 1장~7장까지의 모든 학습을 마무리하면 현재 개발하는 시스템 또는 유지보수하는 시스템에서 튜닝이 필요한 SQL문이 눈에 보이게 됩니다. 그러한 비효율적인 SQL문을 튜닝해 나가는 것이 시스템 부하를 최소화하고 안정된 시스템을 만들어 가는 과정일 것입니다.

이 책을 학습하는 모든 개발자가 SQL문을 작성할 때 튜닝적인 관점에서 접근하여 시스템 성능을 향상시키고 더 나아가 대한민국 IT 시스템이 한 단계 발전할 수 있는 계기가 되기를 간절히 바랍니다.

이 책의 실습 환경

- **Windows 8.1 K** Windows 8.1 K 환경에 오라클을 설치하여 실습을 진행하였습니다.
Linux 또는 Unix 기반에 설치된 오라클 환경에서도 실습할 수 있습니다.
- **Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production** 오라클 11g 버전을 기준으로 기술되었으며 오라클 11g가 설치된 서버 또는 PC라면 이 책의 내용을 실습할 수 있습니다. 이 책에서는 오라클 11g의 설치에 관한 설명은 생략합니다.

스크립트 다운로드

이 책에서 사용한 스크립트는 다음에서 다운로드할 수 있습니다.

- <https://www.hanbit.co.kr/exam/2799>

chapter 1 SQL 튜닝을 위한 준비 ——— 013

- 1.1 테이블 스페이스 및 계정 생성 ——— 013
 - 1.1.1 테이블 스페이스 생성 ——— 013
 - 1.1.2 사용자 계정 생성 ——— 014
- 1.2 실습 테이블 구성 ——— 015
 - 1.2.1 NOLOGGING 모드 설정 ——— 015
 - 1.2.2 APPEND 힌트 ——— 016
 - 1.2.3 데이터 복제 ——— 016
 - 1.2.4 RANDOM 함수의 사용 ——— 017
- 1.3 실행 계획 및 통계정보 생성 ——— 018
 - 1.3.1 실행 계획 ——— 018
 - 1.3.2 실행 계획 분석 ——— 019
 - 1.3.3 통계정보 생성 ——— 020

chapter 2 인덱스 튜닝 ——— 021

- 2.1 인덱스 스캔 튜닝 ——— 021
 - 2.1.1 B-Tree 인덱스 ——— 021
 - 2.1.2 인덱스와 테이블의 관계 ——— 023
 - 2.1.3 인덱스 스캔 튜닝 ——— 024
 - 2.1.4 인덱스 스캔 튜닝 관련 힌트 ——— 025
 - 실습 2-1 인덱스를 최대한 활용하여 원하는 결과 검색하기 ——— 026
 - 실습 2-2 인덱스 구성 컬럼을 추가하여 테이블 랜덤 액세스 제거하기 ——— 035
- 2.2 인덱스 풀 스캔 튜닝 ——— 041
 - 2.2.1 인덱스 풀 스캔 ——— 041
 - 2.2.2 인덱스 풀 스캔의 종류 ——— 042
 - 2.2.3 인덱스 풀 스캔 튜닝 ——— 043
 - 2.2.4 인덱스 풀 스캔 튜닝 관련 힌트 ——— 044

실습 2-3 인덱스 패스트 풀 스캔을 활용하여 집계결과 검색하기 — 045

2.3 테이블 풀 스캔 튜닝 — 050

2.3.1 선택도 — 050

2.3.2 인덱스 손익 분기점 — 050

2.3.3 테이블 풀 스캔 튜닝 — 051

실습 2-4 테이블 풀 스캔을 유도하여 비효율적인 인덱스 스캔 예방하기 — 051

chapter 3 조인 튜닝 — 057

3.1 중첩 루프 조인 튜닝 — 057

3.1.1 중첩 루프 조인 — 057

3.1.2 Outer 테이블과 Inner 테이블 — 058

3.1.3 인라인 뷰 — 058

3.1.4 중첩 루프 조인 튜닝 — 059

3.1.5 중첩 루프 조인 튜닝 관련 힌트 — 060

실습 3-1 효율적인 중첩 루프 조인으로 결과 도출하기 — 063

3.2 해시 조인 튜닝 — 070

3.2.1 해시 조인 — 070

3.2.2 해시 조인의 특성 — 071

3.2.3 Build Input과 Probe Input — 071

3.2.4 해시 조인을 위한 메모리 관리 — 072

3.2.5 해시 조인 튜닝 — 072

3.2.6 해시 조인 튜닝 관련 힌트 — 072

실습 3-2 해시 조인으로 성능 극대화하기 — 074

실습 3-3 인라인 뷰를 이용한 해시 조인으로 성능 극대화하기 — 080

3.3 세미 조인 튜닝 — 086

3.3.1 세미 조인 — 086

3.3.2 EXISTS문과 NOT EXISTS문 — 086

3.3.3 세미 조인 튜닝 — 086

3.3.4 서버쿼리 Unnesting	087
3.3.5 세미 조인 튜닝 관련 힌트	087
실습 3-4 세미 조인 기법을 이용하여 성능 극대화하기	090

3.4 아우터 조인 튜닝 099

3.4.1 아우터 조인	099
3.4.2 Left 아우터 조인	100
3.4.3 Right 아우터 조인	101
3.4.4 아우터 조인 튜닝	101
실습 3-5 아우터 조인으로 테이블 스캔을 최소화하여 성능 개선하기	102
실습 3-6 아우터 조인을 스칼라 서버쿼리 방식으로 변환하여 성능 극대화하기	109

chapter 4 함수 튜닝 115

4.1 분석 함수 튜닝 115

4.1.1 집계 함수의 한계	115
4.1.2 분석 함수의 유용성	116
4.1.3 분석 함수 튜닝	117
4.1.4 주요 분석 함수	118
실습 4-1 RANK 함수를 이용하여 반복적인 테이블 스캔 제거하기	121
실습 4-2 SUM 함수를 이용하여 반복적인 테이블 스캔 제거하기	126

4.2 사용자 정의 함수 튜닝 130

4.2.1 사용자 정의 함수	130
4.2.2 사용자 정의 함수의 재귀 호출 부하	131
4.2.3 사용자 정의 함수 튜닝	131
실습 4-3 재귀 호출 부하 최소화하기	132

chapter 5 부분 범위 처리 튜닝 — 139

- 5.1 부분 범위 처리 — 139
 - 5.1.1 부분 범위 처리의 기초 — 139
 - 5.1.2 부분 범위 처리의 구현 — 141
- 5.2 최대값/최소값 스캔 튜닝 — 143
 - 5.2.1 최대값/최소값 스캔 튜닝 — 143
 - 실습 5-1 부분 범위 처리 기법을 이용하여 최대값/최소값 검색하기 — 143
- 5.3 페이징 처리 튜닝 — 149
 - 5.3.1 페이징 처리 — 149
 - 5.3.2 페이징 처리 튜닝 — 149
 - 실습 5-2 부분 범위 처리 기법을 이용하여 페이징 처리하기 — 149

chapter 6 파티셔닝 튜닝 — 157

- 6.1 파티셔닝 — 157
- 6.2 파티션 프루닝 튜닝 — 158
 - 6.2.1 파티션 프루닝 — 158
 - 실습 6-1 파티션 프루닝이 동작하도록 조건절 튜닝하기 — 158
- 6.3 파티션 인덱스 튜닝 — 163
 - 6.3.1 파티션 인덱스의 정의와 종류 — 163
 - 6.3.2 파티션 인덱스의 생성 방식 — 165
 - 6.3.3 파티션 인덱스 유형 정리 — 166
 - 6.3.4 파티션 인덱스 튜닝 — 166
 - 실습 6-2 파티션 인덱스 스캔으로 성능 극대화하기 — 166

chapter 7 병렬 처리 튜닝 — 173

- 7.1 병렬과 병렬 처리 — 173
- 7.2 병렬 스캔 튜닝 — 174
 - 7.2.1 병렬 스캔 튜닝 — 174
 - 7.2.2 병렬 스캔 튜닝 관련 힌트 — 174
 - 실습 7-1 대용량의 테이블을 병렬 스캔으로 검색하기 — 175
- 7.3 병렬 인덱스 스캔 튜닝 — 179
 - 7.3.1 병렬 인덱스 스캔 — 179
 - 7.3.2 병렬 인덱스 스캔 튜닝 — 180
 - 7.3.3 병렬 인덱스 스캔 튜닝 관련 힌트 — 180
 - 실습 7-2 인덱스 병렬 스캔을 유도하여 성능 극대화하기 — 181

SQL 튜닝을 위한 준비

SQL 튜닝 실습을 진행하기 위해서는 다음의 3가지가 필요합니다.

- 대용량 데이터를 저장할 저장소(테이블 스페이스)와 해당 저장소를 사용할 사용자 계정
- SQL문만으로 대용량의 테이블을 구성하는 방법
- 오라클의 통계정보를 분석하는 방법

이번 장에서 이 3가지에 대해 자세히 다루게 됩니다. 이번 장을 완벽히 이해한 후 본격적인 튜닝 학습을 시작하기 바랍니다.

1.1 테이블 스페이스 및 계정 생성

1.1.1 테이블 스페이스 생성

테이블 스페이스 *Table Space*는 테이블을 저장하는 공간입니다. 오라클은 테이블 스페이스 내에 테이블을 저장하며 테이블에는 데이터가 저장됩니다. 실습을 진행하기 위해서 별도의 테이블 스페이스를 생성합니다. 테이블 스페이스의 생성 방법은 다음과 같습니다.

(1) 관리자 권한으로 로그인

‘sysdba’ 권한으로 접속합니다.

```
sqlplus "/as sysdba"
```

(2) 테이블 스페이스 생성

'dbmsexpert'라는 오라클 인스턴스에 총 4GB 용량의 테이블 스페이스를 생성 하였습니다.

```
CREATE TABLESPACE DBMSEXPERT_DATA
DATAFILE 'C:\app\dbmsexpert\oradata\orc1\DBMSEXPERT_DATA.DBF' SIZE 4G
AUTOEXTEND ON NEXT 1G MAXSIZE UNLIMITED
LOGGING
ONLINE
PERMANENT
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8K
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

(3) 임시 테이블 스페이스 생성

'dbmsexpert'라는 오라클 인스턴스에 총 1GB 용량의 임시 테이블 스페이스를 생성하였습니다.

```
CREATE TEMPORARY TABLESPACE DBMSEXPERT_TMP
TEMPFILE 'C:\app\dbmsexpert\oradata\orc1\DBMSEXPERT_TMP.DBF' SIZE 1G
AUTOEXTEND ON NEXT 100M MAXSIZE UNLIMITED;
```

1.1.2 사용자 계정 생성

앞에서 생성한 테이블 스페이스를 기본 설정으로 하는 오라클 계정을 생성합니다. 생성 방법은 다음과 같습니다.

(1) 관리자 권한으로 로그인

'sysdba' 권한으로 접속합니다.

```
sqlplus "/as sysdba"
```

(2) 사용자 계정 생성

‘DBMSEXPART_DATA’와 ‘DBMSEXPART_TMP’ 테이블 스페이스를 Default로 하는 ‘DBMSEXPART’ 계정을 신규로 생성하였습니다. 지금부터 ‘DBMSEXPART’ 계정으로 로그인하여 생성하는 테이블 및 인덱스는 모두 ‘DBMSEXPART_DATA’에 생성됩니다. 해당 계정으로 작업하다 임시 영역이 필요한 경우에는 ‘DBMSEXPART_TMP’ 영역을 사용하게 됩니다.

```
CREATE USER DBMSEXPART IDENTIFIED BY DBMSEXPART
DEFAULT TABLESPACE DBMSEXPART_DATA
TEMPORARY TABLESPACE DBMSEXPART_TMP
PROFILE DEFAULT
ACCOUNT UNLOCK;
```

(3) 권한 주기

‘DBMSEXPART’ 계정에 권한을 주었습니다.

```
GRANT RESOURCE TO DBMSEXPART;
GRANT CONNECT TO DBMSEXPART;
GRANT CREATE VIEW TO DBMSEXPART;
GRANT CREATE SYNONYM TO DBMSEXPART;
```

(4) 생성된 계정으로 접속

생성된 계정으로 오라클에 접속합니다.

1.2 실습 테이블 구성

1.2.1 NOLOGGING 모드 설정

오라클에서 테이블에 NOLOGGING 모드를 설정하면 해당 테이블에 INSERT 작업 시 Redo 로그 작업을 최소화합니다. 따라서 대용량의 데이터를 INSERT 작업할

때 데이터 입력 시간을 줄일 수 있습니다. 사용법은 다음과 같습니다. 대용량의 데이터를 INSERT 전에 해당 테이블을 NOLOGGING 모드로 설정합니다.

```
ALTER TABLE 테이블명 NOLOGGING;
```

1.2.2 APPEND 힌트

오라클이 테이블에 데이터를 입력할 때 다음 단계를 거치게 됩니다.

- 1) 데이터 버퍼 캐시(Data Buffer Cache)를 경유합니다.
- 2) 테이블 세그먼트의 비어 있는 블록(Free Block)을 검색합니다.
- 3) 비어 있는 블록에 데이터를 저장합니다.

APPEND 힌트를 사용한다면 세그먼트의 HWM(High Water Mark) 바로 뒤부터 데이터를 입력하게 되는데, HWM은 세그먼트의 가장 끝이라고 이해하시면 됩니다. 또한, 데이터 버퍼 캐시를 경유하지 않고 바로 데이터를 저장하게 되므로 데이터의 입력 시간을 단축할 수 있습니다.

APPEND 힌트를 사용하려면 다음과 같이 INSERT 바로 뒤에 APPEND 힌트를 입력합니다.

```
INSERT /*+ APPEND */ INTO 테이블명
```

1.2.3 데이터 복제

대용량의 테이블을 구성하기 위해서는 데이터 복제 기법을 정확히 알아야 합니다.

카티션 곱 조인(Cartesian Product Join)

N건의 데이터로 구성된 'A'라는 테이블과 M건의 데이터를 가진 'B'라는 테이블을 아무런 조인 조건 없이 조인하면 'N건 × M건'의 데이터를 출력하게 됩니다.

다음 예제에서 테이블 A에 100건, 테이블 B에 1,000건의 데이터가 있다고 가정하면, 총 10만 건($100\text{건} \times 1,000\text{건} = 100,000\text{건}$)의 결과 건수가 나오게 됩니다.

```
SELECT * FROM A, B;
```

계층형 쿼리 사용

오라클에서 사용하는 계층형 쿼리를 이용하여 인위적으로 여러 개(N)의 행을 출력할 수 있습니다. 다음 예제는 총 1,000개의 행을 출력하게 됩니다.

```
SELECT * FROM DUAL CONNECT BY LEVEL <= 1000;
```

카티션 곱 조인과 계층형 쿼리의 혼용

카티션 곱 조인과 계층형 쿼리를 혼용하면 특정 테이블의 내용을 복제할 수 있습니다. 다음 예제에서 테이블 'A'에 100건의 데이터가 있다고 가정하면, 총 200건 ($100\text{건} \times 2 = 200\text{건}$)의 행이 생기고 테이블 'A'의 내용을 복제합니다.

```
SELECT * FROM A, (SELECT LEVEL FROM DUAL CONNECT BY LEVEL <= 2);
```

1.2.4 RANDOM 함수의 사용

테이블 구성 시 특정 값을 인위적으로 만들기 위해서 RANDOM 함수를 이용합니다.

랜덤 숫자

다음은 랜덤 숫자를 발생시키는 예제로, 1~100까지의 숫자 중 특정 숫자를 리턴합니다. 기본으로 실수를 리턴하기 때문에 TRUNC 함수로 덮어씌어 주면 정수를 리턴하게 됩니다.

```
SELECT TRUNC(DBMS_RANDOM.VALUE(1, 100)) FROM DUAL;
```

랜덤 문자열

랜덤 문자열을 발생시키는 예제로, 대문자로 된 10자리의 랜덤 문자열을 리턴합니다.

```
SELECT DBMS_RANDOM.STRING('U', 10) FROM DUAL;
```

다음 예제는 소문자로 된 10자리의 랜덤 문자열을 리턴합니다.

```
SELECT DBMS_RANDOM.STRING('L', 10) FROM DUAL;
```

1.3 실행 계획 및 통계정보 생성

1.3.1 실행 계획

오라클의 옵티마이저^{Optimizer}는 사용자가 호출한 SQL에 대해 최적의 실행 계획을 도출해 줍니다. 도출 기준은 SQL문 자체 분석과 각종 통계정보입니다.

실행 계획이 도출되면 해당 실행 계획대로 SQL문에 대한 연산을 수행하게 됩니다. 오라클의 옵티마이저는 타 DBMS보다 월등한 성능을 자랑하며 아무리 복잡한 SQL문이라도 최소한의 비용으로 해당 결과를 도출할 수 있습니다.

하지만 옵티마이저가 모든 SQL문에 대해서 최적의 실행 계획을 도출하는 것은 아닙니다. 때때로 옵티마이저도 비효율적인 실행 계획을 도출하며 해당 SQL문은 DBMS에 과부하의 원인이 되기도 합니다. 실행 계획을 분석하여 옵티마이저가 미처 최적화하지 못한 부분을 찾아 튜닝하는 것이 이 책의 주 목적입니다(이 책은 SQL 기초를 다루는 책이 아니므로 실행 계획을 출력하는 방법은 다루지 않았습니다).

1.3.2 실행 계획 분석

실행 계획 분석은 다음의 두 가지 기본 원칙을 바탕으로 합니다.

- Operation 항목 중 가장 오른쪽에 있는 문자열부터 수행합니다.
- Operation 항목 중 가장 오른쪽에 있는 문자열이 두 개 이상이라면(즉, 같은 Depth에 있다면) 위에서부터 수행합니다.

다음 SQL문에 대한 실행 계획을 분석해 보겠습니다.

SQL문

```
SELECT *  
FROM  
    EMP A, DEPT B  
WHERE  
    A.DEPTNO = B.DEPTNO;
```

실행 계획

ID	Operation	Name	Cost
0	SELECT STATEMENT		7
1	HASH JOIN		7
2	TABLE ACCESS FULL	DEPT	3
3	TABLE ACCESS FULL	EMP	3

수행 순서(ID 기준)

2 → 3 → 1 → 0

실행 계획 설명

ID	설명
2	가로로 기준으로 가장 오른쪽에 위치한 연산이 2번과 3번입니다. 동일한 가로 깊이일 경우 세로를 기준으로 위부터 시작하므로 2번 연산이 가장 먼저 시작하고, DEPT 테이블을 테이블 풀 스캔(TABLE ACCESS FULL)합니다.

ID	설명
3	2번과 동일하게 가로를 기준으로 가장 오른쪽에 있으면서 2번보다 아래에 있는 3번이 수행됩니다. EMP 테이블을 테이블 풀 스캔(TABLE ACCESS FULL)합니다.
1	가로를 기준으로 2번과 3번 바로 왼쪽에 위치한 1번을 수행합니다. 2번과 3번 연산을 해시 조인(HASH JOIN)하였습니다. 옵티마이저가 DEPT 테이블과 EMP 테이블의 조인 연산은 해시 조인이 가장 유리하다고 판단하였습니다(해시 조인에 대한 설명은 3장 2절 참고).
0	가로를 기준으로 1번보다 왼쪽에 있는 0번이 수행됩니다. SELECT절에 대한 연산을 수행합니다.

1.3.3 통계정보 생성

오라클의 옵티마이저가 최적의 실행 계획을 생성하기 위해서는 통계정보가 미리 생성되어 있어야 합니다. 통계정보의 생성 방법은 다음과 같습니다.

(1) 테이블 통계정보 생성

EMP 테이블에 대한 통계정보를 생성합니다.

```
ANALYZE TABLE EMP COMPUTE STATISTICS;
```

(2) 인덱스 통계정보 생성

PK_EMP 인덱스에 대한 통계정보를 생성합니다.

```
ANALYZE INDEX PK_EMP COMPUTE STATISTICS;
```

(3) 특정 테이블과 테이블 내의 인덱스에 대한 통계정보 생성

EMP 테이블과 EMP 테이블이 가지고 있는 모든 인덱스에 대한 통계정보를 생성합니다.

```
ANALYZE TABLE EMP COMPUTE STATISTICS
FOR TABLE FOR ALL INDEXES FOR ALL INDEXED COLUMNS SIZE 254;
```

한빛 리얼타임은 IT 개발자를 위한 전자책입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾기도 쉽지 않습니다. 또한, 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1 eBook First - 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 좀 더 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한, 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

무료로 업데이트되는 전자책 전용 서비스입니다

2 종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3 독자의 편의를 위해 DRM-Free로 제공합니다

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 어려운 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리하고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가겠습니다.

지금 보시는 전자책에 소유 권한을 표시한 문구가 없거나 타인의 소유권함을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 큼니다. 이 경우 저작권법에 따라 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한, 한빛미디어 사이트에서 구매하신 후에는 횡수와 관계없이 내려받을 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오탈자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려 드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구매하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허락되지 않습니다.