
Introduction to Oracle9i: SQL (한글판)

부록 3 • 추가 연습

40049KR11
제품 1.1
2002년 2월
D34387

ORACLE®

만든이

Nancy Greenberg
Priya Nathan

기술 제공자 및 검토자

Josephine Turner
Martin Alvarez
Anna Atkinson
Don Bates
Marco Berbeek
Andrew Brannigan
Laszlo Czinkocski
Michael Gerlach
Sharon Gray
Rosita Hanoman
Mozhe Jalali
Sarah Jones
Charbel Khouri
Christopher Lawless
Diana Lorentz
Nina Minchen
Cuong Nguyen
Daphne Nougier
Patrick Odell
Laura Pezzini
Stacey Procter
Maribel Renau
Bryan Roberts
Helen Robertson
Sunshine Salmon
Casa Sharif
Bernard Soleillant
Craig Spoonemore
Ruediger Steffan
Karla Villasenor
Andree Wheeley
Lachlan Williams

발행인

Nita Brozowski

Copyright © Oracle Corporation, 2000, 2001. All rights reserved.

이 문서는 Oracle Corp.의 독점적 정보를 포함하고 있습니다. 이 정보는 사용 제한 및 기밀 유지 규정을 포함하는 사용권 계약에 따라 제공되며 저작권법에 의해 보호됩니다. 이 소프트웨어를 리버스 엔지니어링 하는 것은 금지되어 있습니다. 이 문서를 미국 국방성 내의 정부 기관에 제공할 때는 제한된 권리 규정이 적용되며 다음 범례를 적용할 수 있습니다.

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Box SB-6, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

Oracle and all references to Oracle products are trademarks or registered trademarks of Oracle Corporation.

인용된 모든 다른 회사명 또는 제품명은 명시적으론 사용되었고 각 소유 회사들의 상표일 수 있습니다.

목차

머리말

교과 과정표

입문

- 목표 I-2
- Oracle9i I-3
- Oracle9i Application Server I-5
- Oracle9i Database I-6
- 관계형 데이터베이스 관리 시스템 및 객체 관계형 데이터베이스 관리 시스템 I-7
- 오라클 인터넷 플랫폼 I-8
- 시스템 개발 주기 I-9
- 다양한 데이터 저장 매체 I-11
- 관계형 데이터베이스 개념 I-12
- 관계형 데이터베이스 정의 I-13
- 데이터 모델 I-14
- 엔티티 관계 모델 I-15
- 엔티티 관계 모델링 표기법 I-16
- 여러 테이블 관련시키기 I-18
- 관계형 데이터베이스에서 사용되는 용어 I-19
- 관계형 데이터베이스 특성 I-20
- SQL을 사용하여 RDBMS와 통신 I-21
- 관계형 데이터베이스 관리 시스템 I-22
- SQL 문 I-23
- 이 과정에서 사용되는 테이블 I-24

1 기본 SQL SELECT 문 작성

- 목표 1-2
- SQL SELECT 문의 기능 I-3
- 기본 SELECT 문 1-4
- 모든 열 선택 1-5
- 특정 열 선택 1-6
- SQL 문 작성 1-7
- 열 머리글 기본 값 1-8
- 산술식 1-9
- 산술 연산자 사용 1-10
- 연산자 우선순위 1-11
- 괄호 사용 1-13
- 널 값 정의 1-14
- 산술식의 널 값 1-15
- 열 별칭 정의 1-16
- 열 별칭 사용 1-17
- 연결 연산자 1-18
- 연결 연산자 사용 1-19
- 리터럴 문자열 1-20
- 리터럴 문자열 사용 1-21
- 중복 행(row) 1-22
- 중복 행(row) 제거 1-23

SQL과 iSQL*Plus의 상호 작용 1-24
SQL 문과 iSQL*Plus 명령 비교 1-25
iSQL*Plus 개요 1-26
iSQL*Plus에 로그인 1-27
iSQL*Plus 환경 1-28
테이블 구조 표시 1-29
스크립트 파일과의 상호 작용 1-31
요약 1-34
연습 1 개요 1-35

2 데이터 제한 및 정렬

목표 2-2
선택을 사용한 행(row) 제한 2-3
선택되는 행(row) 제한 2-4
WHERE 절 사용 2-5
문자열 및 날짜 2-6
비교 조건 2-7
비교 조건 사용 2-8
다른 비교 조건 2-9
BETWEEN 조건 사용 2-10
IN 조건 사용 2-11
LIKE 조건 사용 2-12
NULL 조건 사용 2-14
논리 조건 2-15
AND 연산자 사용 2-16
OR 연산자 사용 2-17
NOT 연산자 사용 2-18
우선순위 규칙 2-19
ORDER BY 절 2-22
내림차순으로 정렬 2-23
열 별칭을 기준으로 정렬 2-24
여러 열을 기준으로 정렬 2-25
요약 2-26
연습 2 개요 2-27

3 단일 행(row) 함수

목표 3-2
SQL 함수 3-3
SQL 함수의 두 유형 3-4
단일 행(row) 함수 3-5
단일 행(row) 함수 3-6
문자 함수 3-7
문자 함수 3-8
대소문자 조작 함수 3-9
대소문자 조작 함수 사용 3-10

문자 조작 함수 3-11
 문자 조작 함수 사용 3-12
 숫자 함수 3-13
 ROUND 함수 사용 3-14
 TRUNC 함수 사용 3-15
 MOD 함수 사용 3-16
 날짜 사용 3-17
 날짜 계산 3-19
 날짜에 산술 연산자 사용 3-20
 날짜 함수 3-21
 날짜 함수 사용 3-22
 연습 3, 1부: 개요 3-24
 변환 함수 3-25
 암시적(implicit) 데이터 유형 변환 3-26
 명시적(explicit) 데이터 유형 변환 3-28
 날짜에 TO_CHAR 함수 사용 3-31
 날짜 형식 모델 요소 3-32
 날짜에 TO_CHAR 함수 사용 3-36
 숫자에 TO_CHAR 함수 사용 3-37
 TO_NUMBER 및 TO_DATE 함수 사용 3-39
 RR 날짜 형식 3-41
 RR 날짜 형식 예제 3-42
 중첩 함수 3-43
 일반 함수 3-45
 NVL 함수 3-46
 NVL 함수 사용 3-47
 NVL2 함수 사용 3-48
 NULLIF 함수 사용 3-49
 COALESCE 함수 사용 3-50
 조건 표현식 3-52
 CASE 표현식 3-53
 CASE 표현식 사용 3-54
 DECODE 함수 3-55
 DECODE 함수 사용 3-56
 요약 3-58
 연습 3, 2부: 개요 3-59

4 여러 테이블의 데이터 표시

목표 4-2
 여러 테이블에서 데이터 얻기 4-3
 카티시안 곱 4-4
 카티시안 곱(Cartesian Product) 생성 4-5
 조인 유형 4-6
 오라클 구문을 사용하여 테이블 조인 4-7
 등가 조인이란? 4-8

- 등가 조인으로 레코드 검색 4-9
- AND 연산자를 사용한 추가 검색 조건 4-10
- 모호한 열 이름 자세히 지정 4-11
- 테이블 별칭 사용 4-12
- 세 개 이상의 테이블 조인 4-13
- 비등가 조인 4-14
- 비등가 조인으로 레코드 검색 4-15
- 포괄 조인 4-16
- 포괄 조인 구문 4-17
- 포괄 조인 사용 4-18
- 자체 조인 4-19
- 테이블 자체 조인 4-20
- 연습 4, 1부: 개요 4-21
- SQL: 1999 구문을 사용한 테이블 조인 4-22
- 교차 조인 작성 4-23
- 자연 조인 작성 4-24
- 자연 조인으로 레코드 검색 4-25
- USING 절을 포함하는 조인 작성 4-26
- USING 절로 레코드 검색 4-27
- ON 절을 조인 작성 4-28
- ON 절로 레코드 검색 4-29
- ON 절로 3-way 조인 작성 4-30
- 내부 조인(inner join)과 포괄 조인 비교 4-31
- LEFT OUTER JOIN 4-32
- RIGHT OUTER JOIN 4-33
- FULL OUTER JOIN 4-34
- 추가 조건 4-35
- 요약 4-36
- 연습 4, 2부: 개요 4-37

5 그룹 함수를 사용한 데이터 집계

- 목표 5-2
- 그룹 함수란? 5-3
- 그룹 함수 종류 5-4
- 그룹 함수 구문 5-5
- AVG 및 SUM 함수 사용 5-6
- MIN 및 MAX 함수 사용 5-7
- COUNT 함수 사용 5-8
- DISTINCT 키워드 사용 5-10
- 그룹 함수 및 널 값 5-11
- 그룹 함수에 NVL 함수 사용 5-12
- 데이터 그룹 생성 5-13
- 데이터 그룹 생성: GROUP BY 절 구문 5-14
- GROUP BY 절 사용 5-15
- 여러 열을 기준으로 그룹화 5-17

	여러 열에 GROUP BY 절 사용	5-18
	그룹 함수를 사용한 잘못된 질의	5-19
	그룹 결과 제외	5-21
	그룹 결과 제외: HAVING 절	5-22
	HAVING 절 사용	5-23
	그룹 함수 중첩	5-25
	요약	5-26
	연습 5 개요	5-27
6	서브 쿼리	
	목표	6-2
	문제 해결에 서브 쿼리 사용	6-3
	서브 쿼리 구문	6-4
	서브 쿼리 사용	6-5
	서브 쿼리 사용 지침	6-6
	서브 쿼리 유형	6-7
	단일 행(row) 서브 쿼리	6-8
	단일 행(row) 서브 쿼리 실행	6-9
	서브 쿼리에서 그룹 함수 사용	6-10
	HAVING 절에 서브 쿼리 사용	6-11
	이 명령문은 무엇이 잘못되었을까요?	6-12
	이 명령문은 행(row)을 반환할까요?	6-13
	여러 행(row) 서브 쿼리	6-14
	여러 행(row) 서브 쿼리에 ANY 연산자 사용	6-15
	여러 행(row) 서브 쿼리에 ALL 연산자 사용	6-16
	서브 쿼리에서의 널 값	6-17
	요약	6-18
	연습 6 개요	6-19
7	iSQL*Plus로 읽기 쉬운 출력 작성	
	목표	7-2
	치환 변수	7-3
	& 치환 변수 사용	7-5
	치환 변수를 사용한 문자 및 날짜 값	7-7
	열 이름, 표현식 및 텍스트 지정	7-8
	치환 변수 정의	7-10
	DEFINE 및 UNDEFINE 명령	7-11
	& 치환 변수에 DEFINE 명령 사용	7-12
	&& 치환 변수 사용	7-13
	VERIFY 명령 사용	7-14
	iSQL*Plus 환경 사용자 정의	7-15
	SET 명령 변수	7-16
	iSQL*Plus 형식 명령	7-17
	COLUMN 명령	7-18
	COLUMN 명령 사용	7-19

COLUMN 형식 모델 7-20
BREAK 명령 사용 7-21
TTITLE 및 BTITLE 명령 사용 7-22
스크립트 파일을 작성하여 보고서 실행 7-24
예제 보고서 7-26
요약 7-28
연습 7 개요 7-29

8 데이터 조작

목표 8-2
데이터 조작어 8-3
테이블에 새 행(row) 추가 8-4
INSERT 문 구문 8-5
새 행(row) 삽입 8-6
널 값을 갖는 행(row) 삽입 8-7
특정 값 삽입 8-8
특정 날짜 값 삽입 8-9
스크립트 작성 8-10
다른 테이블에서 행(row) 복사 8-11
테이블의 데이터 변경 8-12
UPDATE 문 구문 8-13
테이블의 행(row) 갱신 8-14
서브 쿼리로 두 열 갱신 8-15
다른 테이블을 기반으로 행(row) 갱신 8-16
행(row) 갱신: 무결성 제약 조건 오류 8-17
테이블에서 행(row) 제거 8-18
DELETE 문 구문 8-19
테이블에서 행(row) 삭제 8-20
다른 테이블을 기반으로 행(row) 삭제 8-21
행(row) 삭제: 무결성 제약 조건 오류 8-22
INSERT 문에 서브 쿼리 사용 8-23
DML 문에 WITH CHECK OPTION 키워드 사용 8-25
명시적(Explicit) 기본 기능 개요 8-26
명시적 (Explicit) 기본값 사용 8-27
MERGE 문 8-28
MERGE 문 구문 8-29
행(row) 병합 8-30
데이터베이스 트랜잭션 8-32
COMMIT 및 ROLLBACK 문의 장점 8-34
트랜잭션 제어 8-35
표시자까지 변경 내용 롤백 8-36
암시적(implicit) 트랜잭션 처리 8-37
COMMIT 또는 ROLLBACK 실행 이전의 데이터 상태 8-38
COMMIT 실행 이후의 데이터 상태 8-39
데이터 커밋 8-40

ROLLBACK 실행 이후의 데이터 상태 8-41

명령문 레벨 롤백 8-42

읽기 일관성 8-43

읽기 일관성 구현 8-44

잠금 8-45

암시적(Implicit) 잠금 8-46

요약 8-47

연습 8 개요 8-48

읽기 일관성 예제 8-53

9 테이블 생성 및 관리

목표 9-2

데이터베이스 객체 9-3

이름 지정 규칙 9-4

CREATE TABLE 문 9-5

다른 사용자의 테이블 참조 9-6

DEFAULT 옵션 9-7

테이블 생성 9-8

오라클 데이터베이스의 테이블 9-9

데이터 디셔너리 질의 9-10

데이터 유형 9-11

DateTime 데이터 유형 9-13

TIMESTAMP WITH TIME ZONE 데이터 유형 9-15

TIMESTAMP WITH LOCAL TIME 데이터 유형 9-16

INTERVAL YEAR TO MONTH 데이터 유형 9-17

INTERVAL DAY TO SECOND 데이터 유형 9-18

서브 쿼리 구문을 사용한 테이블 생성 9-20

서브 쿼리를 사용한 테이블 생성 9-21

ALTER TABLE 문 9-22

열 추가 9-24

열 수정 9-26

열 삭제 9-27

SET UNUSED 옵션 9-28

테이블 삭제 9-29

객체 이름 변경 9-30

테이블 절단 9-31

테이블에 주석 추가 9-32

요약 9-33

연습 9 개요 9-34

10 제약 조건 포함

- 목표 10-2
- 제약 조건이란? 10-3
- 제약 조건 지침 10-4
- 제약 조건 정의 10-5
- NOT NULL 제약 조건 10-7
- UNIQUE 제약 조건 10-9
- PRIMARY KEY 제약 조건 10-11
- FOREIGN KEY 제약 조건 10-13
- FOREIGN KEY 제약 조건 키워드 10-15
- CHECK 제약 조건 10-16
- 제약 조건 추가 구문 10-17
- 제약 조건 추가 10-18
- 제약 조건 삭제 10-19
- 제약 조건 비활성화 10-20
- 제약 조건 활성화 10-21
- 제약 조건 연쇄화 10-22
- 제약 조건 보기 10-24
- 제약 조건과 연관된 열 보기 10-25
- 요약 10-26
- 연습 10 개요 10-27

11 뷰 생성

- 목표 11-2
- 데이터베이스 객체 11-3
- 뷰란? 11-4
- 뷰 사용 목적 11-5
- 단순 뷰 및 복합 뷰 11-6
- 뷰 생성 11-7
- 뷰에서 데이터 검색 11-10
- 뷰 질의 11-11
- 뷰 수정 11-12
- 복합 뷰 생성 11-13
- 뷰를 통한 DML 작업 수행에 관한 규칙 11-14
- WITH CHECK OPTION 절 사용 11-17
- DML 작업 거부 11-18
- 뷰 제거 11-20
- 인라인 뷰 11-21
- “Top-N” 분석 11-22
- “Top-N” 분석 수행 11-23
- Top-N 분석 예제 11-24
- 요약 11-25
- 연습 11 개요 11-26

12 기타 데이터베이스 객체

목표 12-2

데이터베이스 객체 12-3

시퀀스란? 12-4

CREATE SEQUENCE 문 구문 12-5

시퀀스 생성 12-6

시퀀스 확인 12-7

NEXTVAL 및 CURRVAL 의사 열 12-8

시퀀스 사용 12-10

시퀀스 수정 12-12

시퀀스 수정에 대한 지침 12-13

시퀀스 제거 12-14

인덱스란? 12-15

인덱스 생성 방법 12-16

인덱스 생성 12-17

인덱스 생성이 필요한 경우 12-18

인덱스를 생성하지 않아야 할 경우 12-19

인덱스 확인 12-20

합수 기반 인덱스 12-21

인덱스 제거 12-23

동의어 12-24

동의어 생성 및 제거 12-25

요약 12-26

연습 12 개요 12-27

13 사용자 액세스 제어

목표 13-2

사용자 액세스 제어 13-3

권한 13-4

시스템 권한 13-5

사용자 생성 13-6

사용자 시스템 권한 13-7

시스템 권한 부여 13-8

롤이란? 13-9

롤 생성 및 권한 부여 13-10

암호 변경 13-11

객체 권한 13-12

객체 권한 부여 13-14

WITH GRANT OPTION 및 PUBLIC 키워드 사용 13-15

부여된 권한 확인 13-16

객체 권한 취소 방법 13-17

객체 권한 취소 13-18

데이터베이스 링크 13-19

요약 13-21

연습 13 개요 13-22

- 14 SQL 강습**
 - 강습 개요 14-2
- 15 SET 연산자 사용**
 - 목표 15-2
 - SET 연산자 15-3
 - 이 단원에서 사용되는 테이블 15-4
 - UNION 연산자 15-7
 - UNION 연산자 사용 15-8
 - UNION ALL 연산자 15-10
 - UNION ALL 연산자 사용 15-11
 - INTERSECT 연산자 15-12
 - INTERSECT 연산자 사용 15-13
 - MINUS 연산자 15-14
 - SET 연산자 지침 15-16
 - Oracle Server와 SET 연산자 15-17
 - SELECT 문 일치 15-18
 - 행(row) 순서 제어 15-20
 - 요약 15-21
 - 연습 15 개요 15-22
- 16 Oracle9i Datetime 함수**
 - 목표 16-2
 - 시간대 16-3
 - Oracle9i datetime 지원 16-4
 - TZ_OFFSET 16-6
 - CURRENT_DATE 16-8
 - CURRENT_TIMESTAMP 16-9
 - LOCALTIMESTAMP 16-10
 - DBTIMEZONE 및 SESSIONTIMEZONE 16-11
 - EXTRACT 16-12
 - FROM_TZ를 사용하여 TIMESTAMP 변환 16-13
 - TO_TIMESTAMP 및 TO_TIMESTAMP_TZ를 사용하여 STRING을 TIMESTAMP로 변환 16-14
 - TO_YMINTERVAL을 사용하여 시간 간격 변환 16-15
 - 요약 16-16
 - 연습 16 개요 16-17
- 17 GROUP BY 절의 향상된 기능**
 - 목표 17-2
 - 그룹 함수 복습 17-3
 - GROUP BY 절 복습 17-4
 - HAVING 절 복습 17-5
 - GROUP BY에 ROLLUP 및 CUBE 연산자 사용 17-6
 - ROLLUP 연산자 17-7
 - ROLLUP 연산자 예제 17-8

CUBE 연산자 17-9
CUBE 연산자: 예제 17-10
GROUPING 함수 17-11
GROUPING 함수: 예제 17-12
GROUPING SETS 17-13
GROUPING SETS: 예제 17-15
조합 열 17-17
조합 열: 예제 17-19
연결된 그룹화 17-21
연결된 그룹화 예제 17-22
요약 17-23
연습 17 개요 17-24

18 고급 서브 쿼리

목표 18-2
서브 쿼리란? 18-3
서브 쿼리 18-4
서브 쿼리 사용 18-5
여러 열 서브 쿼리 18-6
열 비교 18-7
쌍(pairwise) 비교 서브 쿼리 18-8
비쌍 비교 서브 쿼리 18-9
FROM 절에 서브 쿼리 사용 18-10
스칼라 서브 쿼리식 18-11
스칼라 서브 쿼리: 예제 18-12
상관 서브 쿼리 18-14
상관 서브 쿼리 사용 18-16
EXISTS 연산자 사용 18-18
NOT EXISTS 연산자 사용 18-20
상관 UPDATE 18-21
상관 DELETE 18-24
WITH 절 18-26
WITH 절: 예제 18-27
요약 18-29
연습 18 개요 18-31

19 계층 검색

목표 19-2
EMPLOYEES 테이블의 예제 데이터 19-3
자연 트리 구조 19-4
계층 질의 19-5
트리 검색 19-6
트리 검색: 아래에서 위로 19-8
트리 검색: 위에서 아래로 19-9
LEVEL 의사 열을 사용하여 순위 결정 19-10

LEVEL 및 LPAD를 사용하여 계층 보고서 형식 지정 19-11

분기 제거 19-13

요약 19-14

연습 19 개요 19-15

20 Oracle9i에서 확장된 DML 및 DDL 문 기능

목표 20-2

INSERT 문 복습 20-3

UPDATE 문 복습 20-4

다중 테이블 INSERT 문 개요 20-5

다중 테이블 INSERT 문 개요 20-6

다중 테이블 INSERT 문 유형 20-7

다중 테이블 INSERT 문 20-8

무조건 INSERT ALL 20-10

조건 INSERT ALL 20-11

조건 FIRST INSERT 20-13

피벗 INSERT 20-15

외부 테이블 20-18

외부 테이블 생성 20-19

외부 테이블 생성 예제 20-20

외부 테이블 질의 20-23

CREATE TABLE 문 내의 CREATE INDEX 20-24

요약 20-25

연습 20 개요 20-26

A 해답

B 테이블 설명 및 데이터

C SQL* Plus 사용

D 고급 스크립트 작성

E 오라클의 구조적 구성 요소

인덱스

추가 연습

추가 연습 해답

추가 연습 테이블과 설명

추가 연습

기본적인 SQL SELECT 문, 기본적인 iSQL*Plus 명령 및 SQL 함수에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어보십시오.

1. 1997년 이후에 입사한 사무원(clerk)의 모든 데이터를 표시하십시오.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK

2. 커미션을 받는 사원의 이름, 업무, 급여 및 커미션을 표시하면서 데이터를 급여에 대한 내림차순으로 정렬하십시오.

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
Abel	SA_REP	11000	.3
Zlotkey	SA_MAN	10500	.2
Taylor	SA_REP	8600	.2
Grant	SA_REP	7000	.15

3. 커미션을 받지 않는 사원의 급여를 10% 인상하여 표시하십시오(급여는 반올림).

New salary
The salary of King after a 10% raise is 26400
The salary of Kochhar after a 10% raise is 18700
The salary of De Haan after a 10% raise is 18700
The salary of Hunold after a 10% raise is 9900
The salary of Ernst after a 10% raise is 6600
The salary of Lorentz after a 10% raise is 4620
The salary of Mourgos after a 10% raise is 6380
The salary of Rajs after a 10% raise is 3850
The salary of Davies after a 10% raise is 3410
The salary of Matos after a 10% raise is 2860
The salary of Vargas after a 10% raise is 2750
The salary of Whalen after a 10% raise is 4840
The salary of Hartstein after a 10% raise is 14300
The salary of Fay after a 10% raise is 6600
New salary
The salary of Higgins after a 10% raise is 13200
The salary of Gietz after a 10% raise is 9130

16 rows selected.

4. 모든 사원의 이름과 함께 근무 연수 및 개월 수를 만으로 표시하십시오.

LAST_NAME	YEARS	MONTHS
King	14	3
Kochhar	12	0
De Haan	8	8
Hunold	11	8
Ernst	10	4
Lorentz	2	7
■ ■ ■		
Whalen	14	0
Hartstein	5	7
Fay	4	1
Higgins	7	3
Gietz	7	3

20 rows selected.

5. 이름이 J, K, L 또는 M으로 시작하는 사원을 표시하십시오.

LAST_NAME
King
Kochhar
Lorentz
Mourgos
Matos

6. 모든 사원을 커미션 수령 여부를 나타내는 “Yes” 또는 “No”와 함께 표시하십시오.

LAST_NAME	SALARY	COM
King	24000	No
Kochhar	17000	No
De Haan	17000	No
Hunold	9000	No
Ernst	6000	No
Lorentz	4200	No
Mourgos	5800	No
Rajs	3500	No
■ ■ ■		

(참고: 다음 페이지에 결과가 계속됩니다.)

Davies	3100	No
Matos	2600	No
Vargas	2500	No
Zlotkey	10500	Yes
Abel	11000	Yes
Taylor	8600	Yes
LAST_NAME	SALARY	COM
Grant	7000	Yes
Whalen	4400	No
Hartstein	13000	No
Fay	6000	No
Higgins	12000	No
Gietz	8300	No

20 rows selected.

기본적인 SQL SELECT 문, 기본적인 iSQL*Plus 명령, SQL 함수, 조인 및 그룹 함수에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

7. 1800이라는 위치에 근무하는 사원의 부서 이름, 위치, 이름, 업무 및 급여를 표시하십시오.

DEPARTMENT_NAME	LOCATION_ID	LAST_NAME	JOB_ID	SALARY
Marketing	1800	Hartstein	MK_MAN	13000
Marketing	1800	Fay	MK_REP	6000

8. 이름이 n으로 끝나는 사원이 몇 명인지 표시하는 질의를 두 가지 방법으로 작성하십시오.

COUNT(*)
3

9. 모든 부서의 이름과 위치 및 각 부서에서 근무하는 사원 수를 표시하십시오. 사원이 없는 부서도 함께 표시하십시오.

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	COUNT(E.EMPLOYEE_ID)
10	Administration	1700	1
20	Marketing	1800	2
50	Shipping	1500	5
60	IT	1400	3
80	Sales	2500	3
90	Executive	1700	3
110	Accounting	1700	2
190	Contracting	1700	0

8 rows selected.

10. 부서 10과 20에는 어떤 업무가 있습니까?

JOB_ID
AD_ASST
MK_MAN
MK_REP

11. Administration 및 Executive 부서에는 어떤 업무가 있으며 해당 업무를 맡고 있는 사원 수는 몇 명입니까? 인원이 가장 많은 업무부터 표시하십시오.

JOB_ID	FREQUENCY
AD_VP	2
AD_ASST	1
AD_PRES	1

기본적인 SQL SELECT 문, 기본적인 iSQL*Plus 명령, SQL 함수, 조인, 그룹 함수 및 서브 쿼리에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

12. 매월 상반기(해당 월의 16일 이전)에 입사한 사원을 모두 표시하십시오.

LAST_NAME	HIRE_DATE
De Haan	13-JAN-93
Hunold	03-JAN-90
Lorentz	07-FEB-99
Matos	15-MAR-98
Vargas	09-JUL-98
Abel	11-MAY-96
Higgins	07-JUN-94
Gietz	07-JUN-94

8 rows selected.

13. 모든 사원의 이름, 급여 및 달러로 나타낸 급여(천 단위)를 표시하십시오.

LAST_NAME	SALARY	THOUSANDS
King	24000	24
Kochhar	17000	17
De Haan	17000	17
Hunold	9000	9
Ernst	6000	6
Lorentz	4200	4
Mourgos	5800	5

■ ■ ■

(참고: 다음 페이지에 결과가 계속됩니다.)

■ ■ ■

Rajs	3500	3
Davies	3100	3
Matos	2600	2
Vargas	2500	2
Zlotkey	10500	10
Abel	11000	11
Taylor	8600	8
LAST_NAME	SALARY	THOUSANDS
Grant	7000	7
Whalen	4400	4
Hartstein	13000	13
Fay	6000	6
Higgins	12000	12
Gietz	8300	8

20 rows selected.

14. 관리자의 급여가 \$15,000를 넘는 모든 사원의 이름을, 관리자 이름, 관리자 급여 및 관리자의 급여 등급과 함께 표시하십시오.

LAST_NAME	MANAGER	SALARY	GRA
Kochhar	King	24000	E
De Haan	King	24000	E
Mourgos	King	24000	E
Zlotkey	King	24000	E
Hartstein	King	24000	E
Whalen	Kochhar	17000	E
Higgins	Kochhar	17000	E
Hunold	De Haan	17000	E

8 rows selected.

15. 모든 부서의 부서 번호, 이름, 사원 수 및 평균 급여를 각 부서에서 근무하는 사원의 이름, 급여 및 업무와 함께 표시하십시오.

DEPARTMENT_ID	DEPARTMENT_NAME	EMPLOYEES	AVG_SAL	LAST_NAME	SALARY	J
10	Administration	1	4400.00	Whalen	4400	AD_
20	Marketing	2	9500.00	Fay	6000	MK_
				Hartstein	13000	MK_
50	Shipping	5	3500.00	Davies	3100	ST_
				Matos	2600	ST_
				Mourgos	5800	ST_
				Rajs	3500	ST_
				Vargas	2500	ST_
60	IT	3	6400.00	Ernst	6000	IT_F
				Hunold	9000	IT_F
				Lorentz	4200	IT_F
80	Sales	3	10033.33	Abel	11000	SA_
				Taylor	8600	SA_
				Zlotkey	10500	SA_
DEPARTMENT_ID	DEPARTMENT_NAME	EMPLOYEES	AVG_SAL	LAST_NAME	SALARY	J
90	Executive	3	19333.33	De Haan	17000	AD_
				King	24000	AD_
				Kochhar	17000	AD_
110	Accounting	2	10150.00	Gietz	8300	AC_
				Higgins	12000	AC_
190	Contracting	0	No average			

20 rows selected.

16. 평균 급여가 가장 높은 부서의 부서 번호 및 해당 부서의 최저 급여를 표시하십시오.

DEPARTMENT_ID	MIN(SALARY)
90	17000

17. 영업 사원이 없는 부서의 부서 번호, 이름 및 위치를 표시하십시오.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

7 rows selected.

18. 다음과 같은 부서의 부서 번호, 부서 이름 및 사원 수를 표시하십시오.

a. 사원이 3명 미만인 부서

DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)
10	Administration	1
20	Marketing	2
110	Accounting	2

b. 사원 수가 가장 많은 부서

DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)
50	Shipping	5

c. 사원 수가 가장 적은 부서

DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)
10	Administration	1

19. 모든 사원의 사원 번호, 이름, 급여, 부서 번호 및 소속 부서의 평균 급여를 표시하십시오.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	AVG(S.SALARY)
100	King	90	19333.3333
101	Kochhar	90	19333.3333
102	De Haan	90	19333.3333
103	Hunold	60	6400
104	Ernst	60	6400
107	Lorentz	60	6400
124	Mourgos	50	3500
141	Rajs	50	3500
142	Davies	50	3500
143	Matos	50	3500
144	Vargas	50	3500
149	Zlotkey	80	10033.3333
174	Abel	80	10033.3333
176	Taylor	80	10033.3333
EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	AVG(S.SALARY)
200	Whalen	10	4400
201	Hartstein	20	9500
202	Fay	20	9500
205	Higgins	110	10150
206	Gietz	110	10150

19 rows selected.

20. 가장 많은 사원이 입사한 요일에 입사한 모든 사원을 표시하십시오.

LAST_NAME	DAY
Ernst	TUESDAY
Mourgos	TUESDAY
Rajs	TUESDAY
Taylor	TUESDAY
Higgins	TUESDAY
Gietz	TUESDAY

6 rows selected.

21. 사원의 입사일을 기준으로 기념일을 파악할 수 있는 목록을 오름차순으로 작성하십시오.

LAST_NAME	BIRTHDAY
Hunold	January 03
De Haan	January 13
Davies	January 29
Zlotkey	January 29
Lorentz	February 07
Hartstein	February 17
Matos	March 15
Taylor	March 24
Abel	May 11
Ernst	May 21
Grant	May 24
Higgins	June 07
Gietz	June 07
King	June 17
LAST_NAME	BIRTHDAY
Vargas	July 09
Fay	August 17
Whalen	September 17
Kochhar	September 21
Rajs	October 17
Mourgos	November 16

20 rows selected.

SET 연산자에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

22. 1990년상반기와 1991년 상반기에 인원이 충원된 업무를 찾으십시오.

JOB_ID
IT_PROG

23. 급여 인상율, 사원 ID, 이전 급여 및 새 급여 목록을 생성하는 혼합 질의를 작성하십시오. 부서 10, 50 및 110의 사원은 5%, 부서 60의 사원은 10%, 부서 20 및 80의 사원은 15%, 부서 90의 사원은 0%의 인상율을 적용하십시오.

RAISE	EMPLOYEE_ID	SALARY	NEW_SALARY
05% raise	124	5800	290
05% raise	141	3500	175
05% raise	142	3100	155
05% raise	143	2600	130
05% raise	144	2500	125
05% raise	200	4400	220
05% raise	205	12000	600
05% raise	206	8300	415
10% raise	103	9000	900
10% raise	104	6000	600
10% raise	107	4200	420
15% raise	149	10500	1575
15% raise	174	11000	1650
15% raise	176	8600	1290
RAISE	EMPLOYEE_ID	SALARY	NEW_SALARY
15% raise	201	13000	1950
15% raise	202	6000	900
no raise	100	24000	24000
no raise	101	17000	17000
no raise	102	17000	17000

19 rows selected.

Oracle9i 단일 행(row) 함수에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

참고: 명령이 실행된 날짜에 따라 결과가 다를 수 있습니다.

24. NLS_DATE_FORMAT을 DD-MON-YYYY HH24:MI:SS로 설정하여 세션을 변경하십시오.
25. a. 다음 시간대에 대해 시간대 오프셋(TZ_OFFSET)을 표시하는 질의를 작성하십시오.
- Australia/Sydney

TZ_OFFSET
+10:00

- Chile/Easter Island

TZ_OFFSET
-05:00

- b. TIME_ZONE 파라미터 값을 Australia/Sydney의 시간대 오프셋으로 설정하여 세션을 변경하십시오.
- c. 이 세션의 SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP 및 LOCALTIMESTAMP를 표시하십시오.

참고: 명령이 실행된 날짜에 따라 결과가 다를 수 있습니다.

SYSDATE	CURRENT_DATE	CURRENT_TIMESTAMP	LOCALTIMESTAMP
23-OCT-2001 14:30:23	24-OCT-2001 04:30:23	24-OCT-01 04.30.23.000001 AM +10:00	24-OCT-01 04.30.23.000001 AM

- d. TIME_ZONE 파라미터 값을 Chile/Easter Island의 시간대 오프셋으로 설정하여 세션을 변경하십시오.

참고: 앞의 문제들은 서로 다른 날짜를 기준으로 하므로 결과가 달라지지만 경우에 따라 수강생의 실제 결과와 같을 수도 있습니다. 또한 국가에 따라 일광 절약 시간 때문에 시간대 오프셋이 달라질 수 있습니다.

e. 이 세션의 SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP 및 LOCALTIMESTAMP를 표시하십시오.

참고: 명령이 실행된 날짜에 따라 결과가 다를 수 있습니다.

SYSDATE	CURRENT_DATE	CURRENT_TIMESTAMP	LOCALTIMESTAMP
04-OCT-2001 16:59:19	04-OCT-2001 14:59:19	04-OCT-01 02:59:19.000000 PM -06:00	04-OCT-01 02:59:19.000000 PM

참고: 앞의 문제를 살펴보면 CURRENT_DATE, CURRENT_TIMESTAMP 및 LOCALTIMESTAMP가 세션 시간대에 따라 달라지는 것을 알 수 있습니다. 그러나 SYSDATE는 세션 시간대의 영향을 받지 않습니다.

참고: 앞의 문제들은 서로 다른 날짜를 기준으로 하므로 결과가 달라지지만 경우에 따라 수강생의 실제 결과와 같을 수도 있습니다. 또한 국가에 따라 일광 절약 시간 때문에 시간대 오프셋이 달라질 수 있습니다.

f. NLS_DATE_FORMAT을 DD-MON-YYYY로 설정하여 세션을 변경하십시오.

26. 입사한 연도에 상관 없이 1월에 입사한 사원의 이름, 입사한 달 및 입사일을 표시하는 질의를 작성하십시오.

LAST_NAME	EXTRACT(MONTHFROMHIRE_DATE)	HIRE_DATE
De Haan	1	13-JAN-1993
Hunold	1	03-JAN-1990
Davies	1	29-JAN-1997
Zlotkey	1	29-JAN-2000

GROUP BY 절의 향상된 기능에 대해 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

27. 부서 ID가 80보다 큰 부서에 대해 다음을 표시하는 질의를 작성하십시오.

- 각 부서 내의 업무별 총 급여
- 총 급여
- 부서가 위치해 있는 도시별 총 급여
- 부서와 상관 없는 업무별 총 급여
- 도시와 상관 없는 부서별 총 급여
- 부서가 위치해 있는 도시의 전체 총 급여
- 업무 및 도시와 상관 없는 부서의 전체 총 급여

CITY	DNAME	JOB	SUM(E.SALARY)
Seattle	Accounting	AC_ACCOUNT	8300
Seattle	Accounting	AC_MGR	12000
Seattle	Accounting		20300
Seattle	Executive	AD_PRES	24000
Seattle	Executive	AD_VP	34000
Seattle	Executive		58000
Seattle		AC_ACCOUNT	8300
Seattle		AC_MGR	12000
Seattle		AD_PRES	24000
Seattle		AD_VP	34000
Seattle			78300
	Accounting	AC_ACCOUNT	8300
	Accounting	AC_MGR	12000
	Accounting		20300
CITY	DNAME	JOB	SUM(E.SALARY)
	Executive	AD_PRES	24000
	Executive	AD_VP	34000
	Executive		58000
		AC_ACCOUNT	8300
		AC_MGR	12000
		AD_PRES	24000
		AD_VP	34000
			78300

22 rows selected.

28. 다음 그룹화를 표시하는 질의를 작성하십시오.

- 부서 ID, 업무 ID
- 업무 ID, 관리자 ID

질의에서 각 그룹의 최고 및 최저 급여를 계산해야 합니다.

DEPARTMENT_ID	JOB	MANAGER_ID	MAX(SALARY)	MIN(SALARY)
10	AD_ASST		4400	4400
20	MK_MAN		13000	13000
20	MK_REP		6000	6000
50	ST_CLERK		3500	2500
50	ST_MAN		5800	5800
60	IT_PROG		9000	4200
80	SA_MAN		10500	10500
80	SA_REP		11000	8600
90	AD PRES		24000	24000
90	AD_VP		17000	17000
110	AC_ACCOUNT		8300	8300
110	AC_MGR		12000	12000
	SA_REP		7000	7000
	AC_ACCOUNT	205	8300	8300
DEPARTMENT_ID	JOB	MANAGER_ID	MAX(SALARY)	MIN(SALARY)
	AC_MGR	101	12000	12000
	AD_ASST	101	4400	4400
	AD PRES		24000	24000
	AD_VP	100	17000	17000
	IT_PROG	102	9000	9000
	IT_PROG	103	6000	4200
	MK_MAN	100	13000	13000
	MK_REP	201	6000	6000
	SA_MAN	100	10500	10500
	SA_REP	149	11000	7000
	ST_CLERK	124	3500	2500
	ST_MAN	100	5800	5800

26 rows selected.

고급 서브 쿼리에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

29. EMPLOYEES 테이블에서 가장 많은 급여를 받는 세 명의 사원에 대해 이름과 급여를 표시하는 질의를 작성하십시오.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000

30. California 주에서 근무하는 사원의 사원 ID와 이름을 표시하는 질의를 작성하십시오.

힌트: 스칼라 서브 쿼리를 사용하십시오.

EMPLOYEE_ID	LAST_NAME
124	Mourgos
141	Rajs
142	Davies
143	Matos
144	Vargas

31. JOB_HISTORY 테이블에서 해당 사원에 대한 MIN (START_DATE)를 찾는 방식으로 각 사원의 가장 오래된 JOB_HISTORY 행을 삭제하는 질의를 작성하십시오. 두 번 이상 업무를 바꾼 사원에 대해서만 이러한 레코드를 삭제하십시오. 질의가 제대로 실행되면 피드백을 받게 됩니다.

힌트: 상관 DELETE 명령을 사용하십시오.

32. 트랜잭션을 롤백하십시오.

33. 해당 업무의 최고 급여가 회사 전체의 업무별 최고 급여 중에서 상위 50% 안에 드는 업무 ID를 표시하는 질의를 작성하십시오. WITH 절을 사용하여 이 질의를 작성하십시오. 질의 이름은 MAX_SAL_CALC로 지정하십시오.

JOB_TITLE	JOB_TOTAL
President	24000
Administration Vice President	17000
Marketing Manager	13000

계층 검색에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

34. 사원 번호, 이름, 입사일 및 급여를 표시하면서 다음을 보여주는 SQL 문을 작성하십시오.

a. De Haan가 직접 보고하는 관리자

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	SALARY
103	Hunold	03-JAN-90	9000

b. De Haan(사원 번호 102) 아래에 있는 조직 트리

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	SALARY
103	Hunold	03-JAN-90	9000
104	Ernst	21-MAY-91	6000
107	Lorentz	07-FEB-99	4200

35. De Haan (사원 번호 102)로부터 두 레벨 아래에 있는 모든 사원의 사원 번호, 관리자 번호 및 사원 이름을 표시하는 계층 질의를 작성하십시오. 사원의 레벨도 함께 표시하십시오.

EMPLOYEE_ID	MANAGER_ID	LEVEL	LAST_NAME
104	103	3	Ernst
107	103	3	Lorentz

36. 사원 번호, 관리자 번호, LEVEL 의사열 및 사원 이름을 표시하는 보고서를 생성하십시오. EMPLOYEES 테이블의 모든 행에 대해 사원, 해당 사원의 관리자, 해당 관리자의 관리자 식의 트리 구조를 출력해야 합니다. NAME 열에 대해 들여쓰기를 사용하십시오.

EMPLOYEE_ID	MANAGER_ID	LEVEL	LAST_NAME
100		1	King
101	100	1	Kochhar
100		2	_King
102	100	1	De Haan
100		2	_King
103	102	1	Hunold
102	100	2	_De Haan
100		3	_King
104	103	1	Ernst
103	102	2	_Hunold
102	100	3	_De Haan
100		4	_King
107	103	1	Lorentz
103	102	2	_Hunold
■ ■ ■			
101	100	2	_Kochhar
100		3	_King
201	100	1	Hartstein
100		2	_King
202	201	1	Fay
201	100	2	_Hartstein
100		3	_King
205	101	1	Higgins
101	100	2	_Kochhar
100		3	_King
206	205	1	Gietz
205	101	2	_Higgins
101	100	3	_Kochhar
100		4	_King

56 rows selected.

참고: 이 결과는 단지 예제일 뿐입니다. 실제 출력 결과는 여기에 표시하지 않았습니다.

Oracle 9i에서 확장된 DML 및 DDL 문의 기능에 대해 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

참고: lab 폴더에 있는 cre_special_sal.sql, cre_sal_history.sql, cre_mgr_history.sql 스크립트를 실행하여 SPECIAL_SAL, SAL_HISTORY 및 MGR_HISTORY 테이블을 생성하십시오.

37. 다음을 수행하는 질의를 작성하십시오.
- EMPLOYEES 테이블에서 사원 ID가 200보다 크거나 같은 사원의 사원 ID, 입사일, 급여 및 관리자 ID를 검색하십시오.
 - 급여가 \$5,000 미만인 경우 사원 ID 및 급여를 SPECIAL_SAL 테이블에 삽입하십시오.
 - 사원 ID, 입사일 및 급여를 SAL_HISTORY 테이블에 삽입하십시오.
 - 사원 ID, 관리자 ID 및 급여를 MGR_HISTORY 테이블에 삽입하십시오.
38. SPECIAL_SAL, SAL_HISTORY 및 MGR_HISTORY 테이블을 질의하여 삽입된 레코드를 보십시오.

SPECIAL_SAL 테이블

EMPLOYEE_ID	SALARY
200	4400

SAL_HISTORY 테이블

EMPLOYEE_ID	HIRE_DATE	SALARY
201	17-FEB-96	13000
202	17-AUG-97	6000
205	07-JUN-94	12000
206	07-JUN-94	8300

MGR_HISTORY 테이블

EMPLOYEE_ID	MANAGER_ID	SALARY
201	100	13000
202	201	6000
205	101	12000
206	205	8300

39. 다음 테이블 인스턴스 차트를 기반으로 LOCATIONS_NAMED_INDEX 테이블을 생성하십시오. PRIMARY KEY 열의 인덱스 이름을 LOCATIONS_PK_IDX로 지정하십시오.

열 이름	Deptno	Dname
기본 키	예	
데이터 유형	Number	VARCHAR2
길이	4	30

40. USER_INDEXES 테이블을 질의하여 LOCATIONS_NAMED_INDEX 테이블에 대한 INDEX_NAME을 표시하십시오.

INDEX_NAME	TABLE_NAME
LOCATIONS_PK_IDX	LOCATIONS_NAMED_INDEX

고급 스크립트 작성에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

41. 사용자가 소유한 모든 객체(테이블, 뷰, 인덱스, 시퀀스, 동의어 등)를 삭제하는 SQL 스크립트 파일을 작성하십시오.

참고: 표시된 결과는 단지 지침일 뿐입니다.

DROP INDEX COUNTRY_C_ID_PK;
DROP INDEX DEPT_ID_PK;
DROP INDEX DEPT_LOCATION_IX;
DROP INDEX EMP_DEPARTMENT_IX;
DROP INDEX EMP_ID_IDX;
DROP INDEX EMP_MANAGER_IX;
DROP INDEX JHIST_DEPARTMENT_IX;
DROP INDEX JHIST_EMP_ID_ST_DATE_PK;
DROP INDEX JOB_ID_PK;
DROP INDEX SYS_C002835;
DROP INDEX REG_ID_PK;
DROP INDEX LOC_STATE_PROVINCE_IX;
DROP INDEX LOC_ID_PK;
DROP INDEX LOC_COUNTRY_IX;
DROP INDEX LOC_CITY_IX;
DROP INDEX LOCATIONS_PK_IDX;
DROP INDEX JHIST_JOB_IX;
DROP INDEX JHIST_EMPLOYEE_IX;
■ ■ ■
DROP TABLE LOCATIONS;
DROP TABLE JOB_GRADES;
DROP TABLE EMP_UNNAMED_INDEX;
DROP TABLE EMPLOYEES;
DROP TABLE COUNTRIES;
DROP VIEW EMP_DETAILS_VIEW;

추가 연습 해답

기본적인 SQL SELECT 문, 기본적인 iSQL*Plus 명령 및 SQL 함수에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어보십시오.

1. 1997년 이후에 입사한 사무원(clerk)의 모든 데이터를 표시하십시오.

```
SELECT *
FROM employees
WHERE job_id = 'ST_CLERK'
AND hire_date > '31-DEC-1997';
```

2. 커미션을 받는 사원의 이름, 업무, 급여 및 커미션을 표시하면서 데이터를 급여에 대한 내림차순으로 정렬하십시오.

```
SELECT last_name, job_id, salary, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY salary DESC;
```

3. 커미션을 받지 않는 사원의 급여를 10% 인상하여 표시하십시오(급여는 반올림).

```
SELECT 'The salary of ' || last_name || ' after a 10% raise is '
|| ROUND(salary*1.10) "New salary"
FROM employees
WHERE commission_pct IS NULL;
```

4. 모든 사원의 이름과 함께 근무 연수 및 개월 수를 만으로 표시하십시오.

```
SELECT last_name,
TRUNC(MONTHS_BETWEEN(SYSDATE, hire_date) / 12) YEARS,
TRUNC(MOD(MONTHS_BETWEEN(SYSDATE, hire_date), 12))
MONTHS
FROM employees;
```

5. 이름이 J, K, L 또는 M으로 시작하는 사원을 표시하십시오.

```
SELECT last_name
FROM employees
WHERE SUBSTR(last_name, 1,1) IN ('J', 'K', 'L', 'M');
```

6. 모든 사원을 커미션 수령 여부를 나타내는 “Yes” 또는 “No”와 함께 표시하십시오.

```
SELECT last_name, salary,
       decode(commission_pct, NULL, 'No', 'Yes') commission
FROM   employees;
```

기본적인 SQL SELECT 문, 기본적인 iSQL*Plus 명령, SQL 함수, 조인 및 그룹 함수에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

7. 1800이라는 위치에 근무하는 사원의 부서 이름, 위치, 이름, 업무 및 급여를 표시하십시오.

```
SELECT d.department_name, d.location_id,
       e.last_name, e.job_id, e.salary
FROM   employees e, departments d
WHERE  e.department_id = d.department_id
AND    d.location_id = 1800;
```

8. 이름이 n으로 끝나는 사원이 몇 명인지 표시하는 질의를 두 가지 방법으로 작성하십시오.

```
SELECT COUNT(*)
FROM   employees
WHERE  last_name LIKE '%n';
```

```
SELECT COUNT(*)
FROM   employees
WHERE  SUBSTR(last_name, -1) = 'n';
```

9. 모든 부서의 이름과 위치 및 각 부서에서 근무하는 사원 수를 표시하십시오. 사원이 없는 부서도 함께 표시하십시오.

```
SELECT d.department_id, d.department_name,
       d.location_id, COUNT(e.employee_id)
FROM   employees e, departments d
WHERE  e.department_id(+) = d.department_id
GROUP BY d.department_id, d.department_name, d.location_id;
```

10. 부서 10과 20에는 어떤 업무가 있습니까?

```
SELECT DISTINCT job_id
FROM   employees
WHERE  department_id IN (10, 20);
```

11. Administration 및 Executive 부서에는 어떤 업무가 있으며 해당 업무를 맡고 있는 사원 수는 몇 명입니까? 인원이 가장 많은 업무부터 표시하십시오.

```
SELECT e.job_id, count(e.job_id) FREQUENCY
FROM   employees e, departments d
WHERE  e.department_id = d.department_id
AND    d.department_name IN ('Administration', 'Executive')
GROUP BY e.job_id
ORDER BY FREQUENCY DESC;
```

기본적인 SQL SELECT 문, 기본적인 iSQL*Plus 명령, SQL 함수, 조인, 그룹 함수 및 서브 쿼리에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

12. 매월 상반기(해당 월의 16일 이전)에 입사한 사원을 모두 표시하십시오.

```
SELECT last_name, hire_date
FROM   employees
WHERE  TO_CHAR(hire_date, 'DD') < 16;
```

13. 모든 사원의 이름, 급여 및 달러로 나타낸 급여(천 단위)를 표시하십시오.

```
SELECT last_name, salary, TRUNC(salary, -3)/1000 Thousands
FROM   employees;
```

14. 관리자의 급여가 \$15,000를 넘는 모든 사원의 이름을, 관리자 이름, 관리자 급여 및 관리자의 급여 등급과 함께 표시하십시오.

```
SELECT e.last_name, m.last_name manager, m.salary,
       j.grade_level
FROM   employees e, employees m, job_grades j
WHERE  e.manager_id = m.employee_id
AND    m.salary BETWEEN j.lowest_sal AND j.highest_sal
AND    m.salary > 15000;
```


15. 모든 부서의 부서 번호, 이름, 사원 수 및 평균 급여를 각 부서에서 근무하는 사원의 이름, 급여 및 업무와 함께 표시하십시오.

```
BREAK ON department_id -
ON department_name ON employees ON avg_sal SKIP 1
```

```
SELECT d.department_id, d.department_name,
       count(e1.employee_id) employees,
       NVL(TO_CHAR(AVG(e1.salary), '99999.99'),
          'No average' ) avg_sal,
       e2.last_name, e2.salary, e2.job_id
FROM   departments d, employees e1, employees e2
WHERE  d.department_id = e1.department_id(+)
AND    d.department_id = e2.department_id(+)
GROUP BY d.department_id, d.department_name,
         e2.last_name, e2.salary, e2.job_id
ORDER BY d.department_id, employees;
```

```
CLEAR BREAKS
```

16. 평균 급여가 가장 높은 부서의 부서 번호 및 해당 부서의 최저 급여를 표시하십시오.

```
SELECT department_id, MIN(salary)
FROM   employees
GROUP BY department_id
HAVING AVG(salary) = (SELECT MAX(AVG(salary))
                     FROM   employees
                     GROUP BY department_id);
```

17. 영업 사원이 없는 부서의 부서 번호, 이름 및 위치를 표시하십시오.

```
SELECT *
FROM   departments
WHERE  department_id NOT IN(SELECT department_id
                           FROM   employees
                           WHERE  job_id = 'SA REP'
                           AND    department_id IS NOT NULL);
```

18. 다음과 같은 부서의 부서 번호, 부서 이름 및 사원 수를 표시하십시오.

a. 사원이 3명 미만인 부서

```
SELECT d.department_id, d.department_name, COUNT(*)
FROM   departments d, employees e
WHERE  d.department_id = e.department_id
GROUP BY d.department_id, d.department_name
HAVING COUNT(*) < 3;
```

b. 사원 수가 가장 많은 부서

```
SELECT d.department_id, d.department_name, COUNT(*)
FROM   departments d, employees e
WHERE  d.department_id = e.department_id
GROUP BY d.department_id, d.department_name
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                   FROM   employees
                   GROUP BY department_id);
```

c. 사원 수가 가장 적은 부서

```
SELECT d.department_id, d.department_name, COUNT(*)
FROM   departments d, employees e
WHERE  d.department_id = e.department_id
GROUP BY d.department_id, d.department_name
HAVING COUNT(*) = (SELECT MIN(COUNT(*))
                   FROM   employees
                   GROUP BY department_id);
```

19. 모든 사원의 사원 번호, 이름, 급여, 부서 번호 및 소속 부서의 평균 급여를 표시하십시오.

```
SELECT e.employee_id, e.last_name,
       e.department_id,   AVG(s.salary)
FROM   employees e, employees s
WHERE  e.department_id = s.department_id
GROUP BY e.employee_id, e.last_name, e.department_id;
```

20. 가장 많은 사원이 입사한 요일에 입사한 모든 사원을 표시하십시오.

```
SELECT last_name, TO_CHAR(hire_date, 'DAY') day
FROM   employees
WHERE  TO_CHAR(hire_date, 'Day') =
       (SELECT TO_CHAR(hire_date, 'Day')
        FROM   employees
        GROUP BY TO_CHAR(hire_date, 'Day')
        HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                           FROM   employees
                           GROUP BY TO_CHAR(hire_date, 'Day')));
```

21. 사원의 입사일을 기준으로 기념일을 파악할 수 있는 목록을 오름차순으로 작성하십시오.

```
SELECT last_name, TO_CHAR(hire_date, 'Month DD') BIRTHDAY
FROM   employees
ORDER BY TO_CHAR(hire_date, 'DDD');
```

SET 연산자에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

22. 1990년 상반기와 1991년 상반기에 인원이 충원된 업무를 찾으십시오.

```
SELECT job_id
FROM employees
WHERE hire_date
BETWEEN '01-JAN-1990' AND '30-JUN-1990'
INTERSECT
SELECT job_id
FROM employees
WHERE hire_date BETWEEN '01-JAN-1991'
AND '30-JUN-1991';
```

23. 급여 인상율, 사원 ID, 이전 급여 및 새 급여 목록을 생성하는 혼합 질의를 작성하십시오.
부서 10, 50 및 110의 사원은 5%, 부서 60의 사원은 10%, 부서 20 및 80의 사원은 15%, 부서 90의 사원은 0%의 인상율을 적용하십시오.

```
SELECT '05% raise' raise, employee_id, salary,
salary *.05 new_salary
FROM employees
WHERE department_id IN (10, 50, 110)
UNION
SELECT '10% raise', employee_id, salary, salary * .10
FROM employees
WHERE department_id = 60
UNION
SELECT '15% raise', employee_id, salary, salary * .15
FROM employees
WHERE department_id IN (20, 80)
UNION
SELECT 'no raise', employee_id, salary, salary
FROM employees
WHERE department_id = 90;
```

Oracle9i 단일 행(row) 함수에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

24. NLS_DATE_FORMAT을 DD-MON-YYYY HH24:MI:SS로 설정하여 세션을 변경하십시오.

```
ALTER SESSION
SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';
```

25. a. 다음 시간대에 대해 시간대 오프셋(TZ_OFFSET)을 표시하는 질의를 작성하십시오.

```
- Australia/Sydney
SELECT TZ_OFFSET ('Australia/Sydney') from dual;

- Chile/EasterIsland
SELECT TZ_OFFSET ('Chile/EasterIsland') from dual;
```

- b. TIME_ZONE 파라미터 값을 Australia/Sydney의 시간대 오프셋으로 설정하여 세션을 변경하십시오.

```
ALTER SESSION SET TIME_ZONE = '+10:00';
```

- c. 이 세션의 SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP 및 LOCALTIMESTAMP를 표시하십시오.

참고: 명령이 실행된 날짜에 따라 결과가 다를 수 있습니다.

```
SELECT SYSDATE, CURRENT_DATE,
CURRENT_TIMESTAMP, LOCALTIMESTAMP
FROM DUAL;
```

- d. TIME_ZONE 파라미터 값을 Chile/Easter Island의 시간대 오프셋으로 설정하여 세션을 변경하십시오.

```
ALTER SESSION SET TIME_ZONE = '-06:00';
```

- e. 이 세션의 SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP 및 LOCALTIMESTAMP
를 표시하십시오.

참고: 명령이 실행된 날짜에 따라 결과가 다를 수 있습니다.

```
SELECT SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP, LOCALTIMESTAMP  
FROM DUAL;
```

- f. NLS_DATE_FORMAT을 DD-MON-YYYY로 설정하여 세션을 변경하십시오.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY';
```

참고: 앞의 문제를 살펴보면 CURRENT_DATE, CURRENT_TIMESTAMP 및
LOCALTIMESTAMP가 세션 시간대에 따라 달라지는 것을 알 수 있습니다. 그러나 SYSDATE
는 세션 시간대의 영향을 받지 않습니다.

참고: 앞의 문제들은 서로 다른 날짜를 기준으로 하므로 결과가 달라지지만 경우에 따라
수강생의 실제 결과와 같을 수도 있습니다. 또한 국가에 따라 일광 절약 시간 때문에 시간대
오프셋이 달라질 수 있습니다.

26. 입사한 연도에 상관 없이 1월에 입사한 사원의 이름, 입사한 달 및 입사일을 표시하는 질의
를 작성하십시오.

```
SELECT last_name, EXTRACT (MONTH FROM HIRE_DATE), HIRE_DATE  
FROM employees  
WHERE EXTRACT (MONTH FROM HIRE_DATE) = 1;
```

GROUP BY 절의 향상된 기능에 대해 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

27. 부서 ID가 80보다 큰 부서에 대해 다음을 표시하는 질의를 작성하십시오.

- 각 부서 내의 업무별 총 급여
- 총 급여
- 부서가 위치해 있는 도시별 총 급여
- 부서와 상관 없는 업무별 총 급여
- 도시와 상관 없는 부서별 총 급여
- 부서가 위치해 있는 도시의 전체 총 급여
- 업무 및 도시와 상관 없는 부서의 전체 총 급여

```
COLUMN city FORMAT A25 Heading CITY
COLUMN department_name FORMAT A15 Heading DNAME
COLUMN job_id  FORMAT A10 Heading JOB
COLUMN SUM(salary)  FORMAT $99,99,999.00 Heading SUM(SALARY)
```

```
SELECT  l.city,d.department_name, e.job_id, SUM(e.salary)
FROM    locations l,employees e,departments d
WHERE   d.location_id = l.location_id
AND     e.department_id = d.department_id
AND     e.department_id > 80
GROUP   BY CUBE( l.city,d.department_name, e.job_id);
```

28. 다음 그룹화를 표시하는 질의를 작성하십시오.

- 부서 ID, 업무 ID
- 업무 ID, 관리자 ID

질원에서 각 그룹의 최고 및 최저 급여를 계산해야 합니다.

```
SELECT department_id, job_id, manager_id,max(salary),min(salary)
FROM   employees
GROUP BY GROUPING SETS
((department_id,job_id), (job_id,manager_id));
```


고급 서브 쿼리에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

29. EMPLOYEES 테이블에서 가장 많은 급여를 받는 세 명의 사원에 대해 이름과 급여를 표시하는 질의를 작성하십시오.

```
SELECT last_name, salary
FROM   employees e
WHERE  3 > (SELECT COUNT(*)
           FROM   employees
           WHERE  e.salary < salary);
```

30. California 주에서 근무하는 사원의 사원 ID와 이름을 표시하는 질의를 작성하십시오.

힌트: 스칼라 서브 쿼리를 사용하십시오.

```
SELECT employee_id, last_name
FROM   employees e
WHERE  ((SELECT location_id
         FROM departments d
         WHERE e.department_id = d.department_id )
        IN  (SELECT location_id
             FROM locations l
             WHERE STATE_province = 'California'));
```

31. JOB_HISTORY 테이블에서 해당 사원에 대한 MIN(START_DATE)를 찾는 방식으로 각 사원의 가장 오래된 JOB_HISTORY 행을 삭제하는 질의를 작성하십시오. 두 번 이상 업무를 바꾼 사원에 대해서만 이러한 레코드를 삭제하십시오. 질의가 제대로 실행되면 피드백을 받게 됩니다.

힌트: 상관 DELETE 명령을 사용하십시오.

```
DELETE FROM job_history JH
WHERE employee_id =
      (SELECT employee_id
       FROM employees E
       WHERE JH.employee_id = E.employee_id
       AND START_DATE = (SELECT MIN(start_date)
                        FROM job_history JH
                        WHERE JH.employee_id = E.employee_id)
       AND 3 > (SELECT COUNT(*)
                FROM job_history JH
                WHERE JH.employee_id = E.employee_id
                GROUP BY EMPLOYEE_ID
                HAVING COUNT(*) >= 2));
```

32. 트랜잭션을 롤백하십시오.

```
ROLLBACK;
```

33. 해당 업무의 최고 급여가 회사 전체의 업무별 최고 급여 중에서 상위 50% 안에 드는 업무 ID를 표시하는 질의를 작성하십시오. WITH 절을 사용하여 이 질의를 작성하십시오. 질의 이름은 MAX_SAL_CALC로 지정하십시오.

```
WITH
MAX_SAL_CALC AS (
  SELECT job_title, MAX(salary) AS job_total
  FROM employees, jobs
  WHERE employees.job_id = jobs.job_id
  GROUP BY job_title)
SELECT job_title, job_total
FROM MAX_SAL_CALC
WHERE job_total > (SELECT MAX(job_total) * 1/2
                   FROM MAX_SAL_CALC)
ORDER BY job_total DESC;
```

계층 검색에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

34. 사원 번호, 이름, 입사일 및 급여를 표시하면서 다음을 보여주는 SQL 문을 작성하십시오.

a. De Haan가 직접 보고하는 관리자

```
SELECT employee_id, last_name, hire_date, salary
FROM   employees
WHERE  manager_id = (SELECT employee_id
                    FROM   employees
                    WHERE  last_name = 'De Haan');
```

b. De Haan(사원 번호 102) 아래에 있는 조직 트리

```
SELECT employee_id, last_name, hire_date, salary
FROM   employees
WHERE  employee_id != 102
CONNECT BY manager_id = PRIOR employee_id
START WITH employee_id = 102;
```

35. De Haan(사원 번호 102)로부터 두 레벨 아래에 있는 모든 사원의 사원 번호, 관리자 번호 및 사원 이름을 표시하는 계층 질의를 작성하십시오. 사원의 레벨도 함께 표시하십시오.

```
SELECT employee_id, manager_id, level, last_name
FROM   employees
WHERE  LEVEL = 3
CONNECT BY manager_id = PRIOR employee_id
START WITH employee_id= 102;
```

36. 사원 번호, 관리자 번호, LEVEL 의 사열 및 사원 이름을 표시하는 보고서를 생성하십시오.
EMPLOYEES 테이블의 모든 행에 대해 사원, 해당 사원의 관리자, 해당 관리자의 관리자 식의 트리 구조를 출력해야 합니다. NAME 열에 대해 들여쓰기를 사용하십시오.

```
COLUMN name FORMAT A25
SELECT  employee_id, manager_id, LEVEL,
LPAD(last_name, LENGTH(last_name)+(LEVEL*2)-2, '_ ') LAST_NAME
FROM    employees
CONNECT BY employee_id = PRIOR manager_id;
COLUMN name CLEAR
```

Oracle 9i에서 확장된 DML 및 DDL 문의 기능에 대해 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

37. 다음을 수행하는 질의를 작성하십시오.

- EMPLOYEES 테이블에서 사원 ID가 200보다 크거나 같은 사원의 사원 ID, 입사일, 급여 및 관리자 ID를 검색하십시오.
- 급여가 \$5,000 미만인 경우 사원 ID 및 급여를 SPECIAL_SAL 테이블에 삽입하십시오.
- 사원 ID, 입사일 및 급여를 SAL_HISTORY 테이블에 삽입하십시오.
- 사원 ID, 관리자 ID 및 급여를 MGR_HISTORY 테이블에 삽입하십시오.

```
INSERT ALL
WHEN SAL < 5000 THEN
INTO special_sal VALUES (EMPID, SAL)
ELSE
INTO sal_history VALUES (EMPID, HIREDATE, SAL)
INTO mgr_history VALUES (EMPID, MGR, SAL)
SELECT employee_id EMPID, hire_date HIREDATE,
       salary SAL, manager_id MGR
FROM employees
WHERE employee_id >=200;
```

38. SPECIAL_SAL, SAL_HISTORY 및 MGR_HISTORY 테이블을 질의하여 삽입된 레코드를 보십시오.

```
SELECT * FROM special_sal;
SELECT * FROM sal_history;
SELECT * FROM mgr_history;
```

39. 다음 테이블 인스턴스 차트를 기반으로 LOCATIONS_NAMED_INDEX 테이블을 생성하십시오. PRIMARY KEY 열의 인덱스 이름을 LOCATIONS_PK_IDX로 지정하십시오.

```
CREATE TABLE LOCATIONS_NAMED_INDEX
(location_id NUMBER(4) PRIMARY KEY USING INDEX
(CREATE INDEX locations_pk_idx ON
LOCATIONS_NAMED_INDEX(location_id)),
location_name VARCHAR2(20));
```

40. USER_INDEXES 테이블을 질의하여 LOCATIONS_NAMED_INDEX 테이블에 대한 INDEX_NAME을 표시하십시오.

```
SELECT INDEX_NAME, TABLE_NAME
FROM USER_INDEXES
WHERE TABLE_NAME = 'LOCATIONS_NAMED_INDEX';
```

고급 스크립트 작성에 대한 내용을 배운 후 아래의 추가 연습 문제를 풀어 보십시오.

41. 사용자가 소유한 모든 객체(테이블, 뷰, 인덱스, 시퀀스, 동의어 등)를 삭제하는 SQL 스크립트 파일을 작성하십시오. 참고: 표시된 결과는 단지 지침일 뿐입니다.

```
SET HEADING OFF ECHO OFF FEEDBACK OFF
SET PAGESIZE 0
```

```
SELECT 'DROP ' || object_type || ' ' || object_name || ';'
FROM user_objects
ORDER BY object_type
/
```

```
SET HEADING ON ECHO ON FEEDBACK ON
SET PAGESIZE 24
```

추가 연습

테이블 설명
및 데이터

COUNTRIES 테이블

```
DESCRIBE countries
```

Name	Null?	Type
COUNTRY_ID	NOT NULL	CHAR(2)
COUNTRY_NAME		VARCHAR2(40)
REGION_ID		NUMBER

```
SELECT * FROM countries;
```

CO	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

DEPARTMENTS 테이블

DESCRIBE departments

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

SELECT * FROM departments;

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

EMPLOYEES 테이블

DESCRIBE employees

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

EMPLOYEES 테이블(계속)

```
SELECT * FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94

20 rows selected.

EMPLOYEES 테이블(계속)

JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
AD_PRES	24000			90
AD_VP	17000		100	90
AD_VP	17000		100	90
IT_PROG	9000		102	60
IT_PROG	6000		103	60
IT_PROG	4200		103	60
ST_MAN	5800		100	50
ST_CLERK	3500		124	50
ST_CLERK	3100		124	50
ST_CLERK	2600		124	50
ST_CLERK	2500		124	50
SA_MAN	10500	.2	100	80
SA_REP	11000	.3	149	80
SA_REP	8600	.2	149	80
SA_REP	7000	.15	149	
AD_ASST	4400		101	10
MK_MAN	13000		100	20
MK_REP	6000		201	20
AC_MGR	12000		101	110
AC_ACCOUNT	8300		205	110

20 rows selected.

JOBS 테이블

```
DESCRIBE jobs
```

Name	Null?	Type
JOB_ID	NOT NULL	VARCHAR2(10)
JOB_TITLE	NOT NULL	VARCHAR2(35)
MIN_SALARY		NUMBER(6)
MAX_SALARY		NUMBER(6)

```
SELECT * FROM jobs;
```

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20000	40000
AD_VP	Administration Vice President	15000	30000
AD_ASST	Administration Assistant	3000	6000
AC_MGR	Accounting Manager	8200	16000
AC_ACCOUNT	Public Accountant	4200	9000
SA_MAN	Sales Manager	10000	20000
SA_REP	Sales Representative	6000	12000
ST_MAN	Stock Manager	5500	8500
ST_CLERK	Stock Clerk	2000	5000
IT_PROG	Programmer	4000	10000
MK_MAN	Marketing Manager	9000	15000
MK_REP	Marketing Representative	4000	9000

12 rows selected.

JOB_GRADES 테이블

DESCRIBE job_grades

Name	Null?	Type
GRADE_LEVEL		VARCHAR2(3)
LOWEST_SAL		NUMBER
HIGHEST_SAL		NUMBER

SELECT * FROM job_grades;

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

6 rows selected.

JOB_HISTORY 테이블

DESCRIBE job_history

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
START_DATE	NOT NULL	DATE
END_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
DEPARTMENT_ID		NUMBER(4)

SELECT * FROM job_history;

EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
102	13-JAN-93	24-JUL-98	IT_PROG	60
101	21-SEP-89	27-OCT-93	AC_ACCOUNT	110
101	28-OCT-93	15-MAR-97	AC_MGR	110
201	17-FEB-96	19-DEC-99	MK_REP	20
114	24-MAR-98	31-DEC-99	ST_CLERK	50
122	01-JAN-99	31-DEC-99	ST_CLERK	50
200	17-SEP-87	17-JUN-93	AD_ASST	90
176	24-MAR-98	31-DEC-98	SA_REP	80
176	01-JAN-99	31-DEC-99	SA_MAN	80
200	01-JUL-94	31-DEC-98	AC_ACCOUNT	90

10 rows selected.

LOCATIONS 테이블

DESCRIBE locations

Name	Null?	Type
LOCATION_ID	NOT NULL	NUMBER(4)
STREET_ADDRESS		VARCHAR2(40)
POSTAL_CODE		VARCHAR2(12)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_ID		CHAR(2)

SELECT * FROM locations;

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	CO
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario	CA
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK

REGIONS 테이블

```
DESCRIBE regions
```

Name	Null?	Type
REGION_ID	NOT NULL	NUMBER
REGION_NAME		VARCHAR2(25)

```
SELECT * FROM regions;
```

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

