Workbook

실무 실습

A SQL 실습
B 데이터 모델링
C 데이터베이스 프로젝트

PREVIEW

워크북은 1~9장까지 배운 내용을 실무 전문가가 추천하는 문제로 직접 실습해 볼 수 있도록 구성하였다. 각각 다루는 내용은 다음과 같다.

A. SQL 실습 | 기초적인 SQL 문과 주제별 질의문 작성

0. 개요
1. 데이터 검색: SELECT
2. 데이터 제한 및 정렬: WHERE, ORDER BY
3. 단일행 함수 및 변환 함수
4. 집계된 데이터 보고: 집계 함수
5. 여러 테이블의 데이터 표시: JOIN
6. 부속질의
7. 집합 연산자
8. 조건부 논리 표현식 제어: CASE & DECODE
9. 다차원 그룹 데이터 검색: ROLLUP, CUBE
10. 분석 함수: NTILE, RANK
11. 계층 검색

B. 데이터 모델링 | 다양한 업무별 데이터 모델링 수행

0. 개체, 관계, 속성의 이해
1. 푸드코트 ER 다이어그램
2. 병원 업무 관리 ER 다이어그램
3. 학사 관리 ER 다이어그램
4. 은행 업무 관리 ER 다이어그램
5. 캠핑카 대여 관리 ER 다이어그램
6. 정규화 수행 연습: 도서 주문

C. 데이터베이스 프로젝트 | 모델링 이후 데이터베이스 구축

1. 병원 업무 관리 프로젝트
2. 학사 관리 프로젝트
3. 은행 업무 관리 프로젝트
4. 캠핑카 대여 관리 프로젝트

A SQL 실습

0 개요

실습은 오라클 데이터베이스에서 제공하는 기업의 인사부서Human Resource, hr 데이터베이스를 사용한다. 오라클 21c 버전부터는 hr 데이터베이스가 기본으로 탑재되어 있지 않으므로 직접 hr 계정을 만들고 데이터베이스를 설치해야 한다. 먼저 hr 계정을 만들고, 데이터베이스를 생성하는 과정을 SQL Developer 기준으로 준비해 보자.

TIP 예제 소스와 함께 제공되는 oracle-hr-schema_CDB.sql 파일을 이용하여 계정을 직접 만들고 데이터도 함께 설치하여 진행한다.

❶ [그림 A-1]의 스크립트를 system 계정에서 실행하여 hr 계정을 만든다.

```
-- [system 계정에서 - Oracle 21c 이상인 경우] -- hr 스키마 생성
-- ORACLE 21c 이상 CDB 환경에서 사용
-- CDB user 사용, 사용자 이름에 c##을 붙이고, GRANT UNLIMITED 명령을 추가
-- id: c##hr, password: hr
-- DROP USER c##hr CASCADE; -- 이미 hr 계정이 있으면 삭제하는 명령어

CREATE USER c##hr IDENTIFIED BY hr DEFAULT TABLESPACE users
            TEMPORARY TABLESPACE temp PROFILE DEFAULT;
GRANT CONNECT, RESOURCE TO c##hr;
GRANT CREATE VIEW, CREATE SYNONYM TO c##hr;
GRANT UNLIMITED TABLESPACE TO c##hr;
ALTER USER c##hr ACCOUNT UNLOCK;
```

그림 A-1 system 계정에서 hr 계정 생성 스크립트

❷ SQL Developer에 hr 계정(c##hr)에 대한 접속 정보를 등록하여 실습 준비를 마무리한다.

접속 이름	hr
사용자 이름	c##hr
비밀번호	hr
비밀번호 저장	체크
호스트 이름	localhost
포트	1521
SID	FREE

그림 A-2 PDB1_hr 데이터베이스 접속 정보 등록

❸ c##hr 계정으로 샘플 데이터베이스에 접속하여 스크립트를 실행시킨다.

오라클 데이터베이스의 hr 스키마를 ER 다이어그램으로 나타내면 [그림 A-3]과 같다. 본격적인 실습에 앞서 각 테이블의 기본키와 외래키, 그리고 테이블 간의 관계를 파악해 보기 바란다.

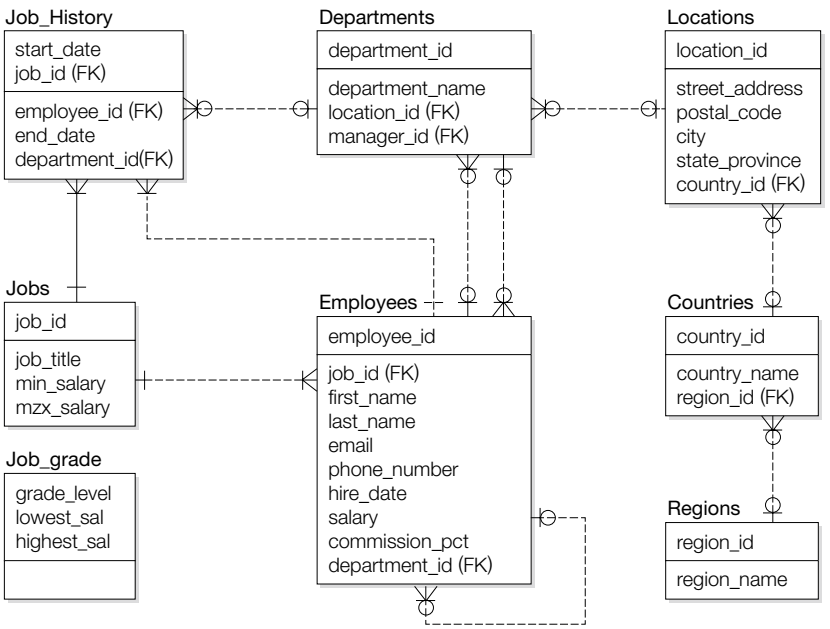


그림 A-3 hr 스키마의 ER 다이어그램

1 데이터 검색: SELECT

데이터베이스에서 테이블에 저장된 데이터를 조회하려면 SELECT 문을 이용한다. SELECT 문을 이용하면 원하는 열만 선택하여 출력할 수 있다.

여기서 잠깐! | SELECT 문을 통해 출력되는 결과의 표현 방식

- 테이블의 모든 행과 열을 출력할 수 있다.
- 열에 표현식 및 수식을 사용할 경우 별칭을 붙여 알기 쉽게 출력할 수 있다.

샘플 문제

사원정보(Employees) 테이블에서 사원번호, 이름, 급여, 업무, 입사일, 상사의 사원번호를 출력하시오. 이때 이름은 성과 이름을 연결하여 'Name'이라는 별칭으로 출력하시오(107행).

풀이

```
SELECT  employee_id, first_name || last_name as "Name",
        salary, job_id, hire_date, manager_id
FROM    Employees;
```

	EMPLOYEE_ID	Name	SALARY	JOB_ID	HIRE_DATE	MANAGER_ID
1	100	StevenKing	24000	AD PRES	03/06/17	(null)
2	101	NeenaKochhar	17000	AD VP	05/09/21	100
3	102	LexDe Haan	17000	AD VP	01/01/13	100
4	103	AlexanderHunold	9000	IT PROG	06/01/03	102
5	104	BruceErnst	6000	IT PROG	07/05/21	103
			... (생략) ...			
105	204	HermannBaer	10000	PR REP	02/06/07	101
106	205	ShelleyHiggins	12008	AC MGR	02/06/07	101
107	206	WilliamGietz	8300	AC ACCOUNT	02/06/07	205

지금부터 문제와 결과 화면을 보고 SQL 문을 직접 작성해 보자.

문제 1

사원정보(Employees) 테이블에서 사원의 성과 이름은 'Name', 업무는 'Job', 급여(월급)는 'Salary', 연봉에 $100 보너스를 추가하여 계산한 연봉은 'Increased Ann_salary', 급여(월급)에 $100 보너스를 추가하여 계산한 연봉은 'Increased Salary'라는 별칭으로 출력하시오(107행).

HINT 연산자는 우선순위를 갖는다.

풀이

	Name	Job	Salary	Increased Ann_salary	Increased Salary
1	StevenKing	AD PRES	24000	288100	289200
2	NeenaKochhar	AD VP	17000	204100	205200
3	LexDe Haan	AD VP	17000	204100	205200
			... (생략) ...		
105	HermannBaer	PR REP	10000	120100	121200
106	ShelleyHiggins	AC MGR	12008	144196	145296
107	WilliamGietz	AC ACCOUNT	8300	99700	100800

문제 2

사원정보(Employees) 테이블에서 모든 사원의 성(last_name)과 연봉을 '성: 1 Year Salary = $ 연봉' 형식으로 출력하고, '1 Year Salary'라는 별칭을 붙여 출력하시오(107행).

HINT 모든 문자, 열, 숫자, 특수기호 등은 연결할 수 있다.

풀이

	1 Year Salary
1	King: 1 Year Salary = $288000
2	Kocchar: 1 Year Salary = $204000
3	De Haan: 1 Year Salary = $204000
4	Hunold: 1 Year Salary = $10800
5	Ernst: 1 Year Salary = $72000
	... (생략) ...

문제 3

부서별로 담당하는 업무를 한 번씩만 출력하시오(20행).

HINT DISTINCT로 열의 중복된 값을 배제한다.

풀이

	DEPARTMENT_ID	JOB_ID
1	90	AD PRES
2	90	AD VP
3	60	IT PROG
	... (생략) ...	
18	70	PR REP
19	110	AC MGR
20	110	AC ACCOUNT

2 데이터 제한 및 정렬: WHERE, ORDER BY

WHERE 절을 조건절이라고 부르며, 데이터베이스의 수많은 정보 중에 원하는 정보만 필터링하여 얻고자 할 때 사용한다.

여기서 잠깐! | WHERE 절의 표현

- 쿼리(질의)에서 반환되는 행을 제한할 수 있다.
- 반드시 FROM 절 뒤에 사용해야 하며 조건이 참인 경우 만족하는 행을 반환한다.
- 조건식은 컬럼 이름, 비교 연산자, 비교 대상 등으로 구성된다.
- 날짜와 문자는 작은따옴표(' ')로 묶어서 표현한다.

샘플 문제

HR 부서에서 예산 편성 문제로 급여 정보 보고서를 작성하려고 한다. 사원정보(Employees) 테이블에서 급여가 $7,000~$10,000 범위에 포함되지 않는 사람의 성과 이름('Name'으로 별칭) 및 급여를 급여가 적은 순서로 출력하시오(75행).

HINT 연산자의 범위 출력은 BETWEEN A AND B 연산자를 이용하고, 부정에 대한 표현은 NOT 연산자를 이용한다.

풀이

```
SELECT   first_name || ' ' || last_name, salary
FROM     Employees
WHERE    salary NOT BETWEEN 7000 AND 10000
ORDER BY salary ;
```

	FIRST_NAME||''||LAST_NAME	SALARY
1	TJ Olson	2100
2	Steven Markle	2200
3	Hazel Philtanker	2200
	... (생략) ...	
73	Neena Kochhar	17000
74	Lex De Haan	17000
75	Steven King	24000

문제 1

사원의 이름(last_name) 중에 'e'나 'o' 글자가 포함된 사원을 출력하시오. 이때 머리글은 'e and o Name'이라고 출력하시오(8행).

HINT LIKE 연산자는 특정 패턴에 맞는 정보를 출력할 때 유용하다.

풀이

	e and o Name
1	Colmenares
2	Geoni
3	Jones
4	Lorentz
5	OConnell
6	Rogers
7	Seo
8	Zlotkey

문제 2

날짜 함수를 이용해 2006년 05월 20일부터 2007년 05월 20일 사이에 고용된 사원들의 성과 이름('Name'으로 별칭), 사원번호, 고용일자를 출력하시오. 단, 입사일이 빠른 순으로 정렬하시오(18행).

HINT 현재 날짜는 SYSDATE 함수를 이용하고, 날짜는 현재 날짜 형식에 맞게 조회한다.

풀이

	Name	EMPLOYEE_ID	HIRE_DATE
1	Kevin Feeney	197	06/05/23
2	Julia Dellinger	186	06/06/24
3	Samuel McCain	194	06/07/01
	... (생략) ...		
16	Danielle Greene	163	07/03/19
17	Elizabeth Bates	172	07/03/24
18	TJ Olson	132	07/04/10

문제 3

수당을 받는 모든 사원의 성과 이름('Name'으로 별칭), 급여(salary), 업무, 수당률(commission_pct)을 출력하시오. 급여가 많은 순서대로 정렬하되, 급여가 같으면 수당률이 큰 순서대로 정렬하시오(35행).

HINT 테이블에는 NULL 값이 포함되어 있으며, NULL 값은 조건절에서 활용할 수 있다. 또한 다중 컬럼 정렬은 첫 번째 정렬 기준으로 정렬한 후, 해당 기준값이 동일한 경우에만 두 번째 정렬 기준이 적용된다.

풀이

	Name	SALARY	JOB_ID	COMMISSION_PCT
1	John Russell	14000	SA MAN	0.4
2	Karen Partners	13500	SA MAN	0.3
3	Alberto Errazuriz	12000	SA MAN	0.3
		... (생략) ...		
33	Amit Banda	6200	SA REP	0.1
34	Charles Johnson	6200	SA REP	0.1
35	Sundita Kumar	6100	SA REP	0.1

3 단일행 함수 및 변환 함수

오라클에서 제공하는 여러 함수를 사용해 복잡한 계산도 쉽게 처리할 수 있다. 함수는 크게 행 단위로 처리되는 단일행 함수와 그룹 단위로 처리되는 집계 함수로 나눌 수 있다. 여기에서는 단일행 함수에 대해 알아본다.

여기서 잠깐! | 단일행 함수의 종류

- 문자 조작 함수: UPPER(), LOWER(), REPLACE(), SUBSTR(), LPAD(), CONCAT() 등
- 숫자 관련 함수: ROUND(), TRUNC(), MOD(), POWER() 등
- 날짜 관련 함수: SYSDATE(), ADD_MONTHS(), LAST_DAY(), MONTHS_BETWEEN() 등
- 변환 함수: TO_CHAR(), TO_DATE(), TO_NUMBER() 등

샘플 문제

신규 프로그램을 개발해 회사에 공헌한 IT 부서의 사원 급여를 12.3% 인상하기로 하였다. 60번 IT 부서 사원의 급여를 12.3% 인상하여 정수만(반올림) 표시하는 보고서를 작성하시오. 출력 형식은 사번, 이름과 성('Name'으로 별칭), 급여, 인상된 급여('Increased Salary'로 별칭) 순으로 출력한다(5행).

HINT ROUND() 함수를 이용하여 반올림 처리한다.

풀이

```
SELECT  employee_id, last_name || ' ' || first_name as "Name", salary,
        ROUND(salary * 1.123, 0) "Incresed Salary"
FROM    Employees
WHERE   department_id = 60;
```

	EMPLOYEE_ID	Name	SALARY	Increased Salary
1	103	Hunold Alexander	9000	10107
2	104	Ernst Bruce	6000	6738
3	105	Austin David	4800	5390
4	106	Pataballa Valli	4800	5390
5	107	Lorentz Diana	4200	4717

문제 1

성(last_name)이 's'로 끝나는 사원의 이름과 업무를 'Michael Rogers is a ST_CLERK'의 형태로 출력하고자 한다. 이름(first_name)과 성(last_name)은 첫 자만 대문자, 업무는 모두 대문자, 머리글은 'Employee JOBs.'라는 별칭으로 표시하시오(18행).

HINT INITCAP, UPPER, SUBSTR 함수를 사용한다. SUBSTR 함수에서 문자열의 마지막 문자를 추출할 때는 두 번째 인자에 −1을 사용할 수 있다.

풀이

	Employee JOBs.
1	Shelley Higgins is a AC MGR
2	Susan Mavris is a HR REP
3	Sigal Tobias is a PU CLERK
	... (생략) ...
16	Peter Vargas is a ST CLERK
17	Matthew Weiss is a ST MAN
18	Kevin Mourgos is a ST Man

문제 2

모든 사원의 연봉을 표시하는 보고서를 작성하려고 한다. 보고서에 사원의 성과 이름('Name'으로 별칭), 급여, 수당 여부에 따른 연봉을 포함하여 출력하시오. 수당 여부는 수당이 있으면 'Salary + Commission', 수당이 없으면 'Salary only'라고 표시하고, 별칭은 적절히 붙인다. 또한 출력 시 연봉이 높은 순으로 정렬한다(107행).

HINT CASE 문이나 DECODE를 사용한다. 오라클의 경우 NVL(), NVL2() 함수를 사용하면 더 간단하다. NVL() 함수를 이용해 NULL 값을 실제 연산 가능한 값으로 변환할 수 있고, NVL2() 함수를 이용해 원래 값이 NULL이었는지 아니었는지를 식별한다.

풀이 NVL(exp1, exp2)

- exp1: NULL을 포함할 수 있는 소스 값 또는 표현식
- exp2: NULL을 변환하기 위한 대상 값

NVL2(exp1, exp2, exp3)

- exp1: NULL을 포함할 수 있는 소스 값 또는 표현식
- exp2: exp1이 NULL이 아닌 경우에 반환되는 값
- exp3: exp1이 NULL인 경우에 반환되는 값

	Name	SALARY	Annual Salary	Commission?
1	Steven King	24000	288000	Salary only
2	Neena Kochhar	17000	204000	Salary only
3	Lex De Haan	17000	204000	Salary only
4	John Russell	14000	235200	Salary + Commission
5	Karen Partners	13500	210600	Salary + Commission
... (생략) ...				
105	Hazel Philtanker	2200	26400	Salary only
106	Steven Markle	2200	26400	Salary only
107	TJ Olson	2100	25200	Salary only

문제 3

모든 사원의 성과 이름('Name'으로 별칭), 입사일, 입사 요일을 출력하시오. 이때 주(week)의 시작인 일요일부터 출력되도록 정렬하시오(107행).

HINT TO_CHAR() 함수는 날짜와 숫자를 문자로 변환해 주는 함수이다.

풀이

	Name	HIRE_DATE	Day of the week
1	Valli Pataballa	06/02/05	일요일
2	James Landry	07/01/14	일요일
3	Douglas Grant	08/01/13	일요일
... (생략) ...			
105	Nancy Greenberg	02/08/17	토요일
106	Lex De Haan	01/01/13	토요일
107	David Austin	05/06/25	토요일

4 집계된 데이터 보고: 집계 함수

집계 함수(또는 그룹 함수)는 테이블의 행row에 대한 집계(합, 평균, 최대, 최소, 개수 등)를 낼 때 사용한다. 테이블 집합을 더 작은 그룹별로 조건 처리를 수행할 때 유용하다.

여기서 잠깐! | 집계 함수와 GROUP BY 절

- 집계 함수 종류: SUM(), AVG(), COUNT(), MAX(), MIN() 등
- GROUP BY 절: 테이블을 더 작게 그룹화하고, 그룹을 나누는 기준으로 열, 계산식, 함수 등을 사용할 수 있다.
- HAVING 절: 집계 함수에 대한 조건을 지정할 수 있다.
- 오라클 10g 버전 이후부터 GROUP BY 절은 자동 정렬을 포함하지 않는다.

샘플 문제

모든 사원은 직속 상사 및 직속 직원이 있다. 단, 최상위 또는 최하위 직원은 직속 상사 또는 직속 직원이 없다. 소속된 사원들 중 어떤 사원의 상사로 근무 중인 사원의 총수를 출력하시오(1

풀이

```
SELECT  COUNT(DISTINCT manager_id) "Count Managers"
FROM    Employees;
```

	Count Managers
1	18

문제 1

각 사원의 부서별로 급여 합계, 급여 평균, 급여 최댓값, 급여 최솟값을 집계하고자 한다. 계산 값은 여섯 자리와 세 자리 구분기호, $와 함께 출력하고 부서번호의 오름차순으로 정렬하시오. 단, 부서에 소속되지 않은 사원에 대한 정보는 제외하고, 출력 시 머리글은 DEPARTMENT_ID, Sum Salary, Avg Salary, Max Salary, Min Salary로 별칭 처리하시오(11행).

HINT TO_CHAR() 함수의 출력 양식을 정하는 방법은 TO_CHAR(SUM(salary), '$999,999.00')와 같다. GROUP BY 절은 특정 열을 기준으로 작은 그룹으로 나누는 역할을 한다.

풀이

	DAPARTMENT_ID	Sum Salary	Avg Salary	Max Salary	Min Salary
1	10	$4,400.00	$4,400.00	$4,400.00	$4,400.00
2	20	$19,000.00	$9,500.00	$13,000.00	$6,000.00
3	30	$24,900.00	$4,150.00	$11,000.00	$2,500.00
4	40	$6,500.00	$6,500.00	$6,500.00	$6,500.00
5	50	$156,400.00	$3,475.56	$8,200.00	$2,100.00
6	60	$28,800.00	$5,760.00	$9,000.00	$4,200.00
7	70	$10,000.00	$10,000.00	$10,000.00	$10,000.00
8	80	$304,500.00	$8,955.88	$14,000.00	$6,100.00
9	90	$58,000.00	$19,333.33	$24,000.00	$17,000.00
10	100	$51,608.00	$8,601.33	$12,008.00	$6,900.00
11	110	$20,308.00	$10,154.00	$12,008.00	$8,300.00

문제 2

업무별 전체 급여 평균이 $10,000보다 큰 경우를 조회하여 업무와 급여 평균을 출력하시오. 단, 업무에 사원(CLERK)이 포함된 경우는 제외하고 전체 급여 평균이 높은 순서대로 출력하시오(7행).

HINT GROUP BY, HAVING, ORDER BY 절 등을 사용한다.

풀이

	JOB_ID	Avg Salary
1	AD PRES	24000
2	AD VP	17000
3	MK MAN	13000
4	SA MAN	12200
5	FI MGR	12008
6	AC MGR	12008
7	PU MAN	11000

5 여러 테이블의 데이터 표시: JOIN

데이터를 다루다 보면 간혹 하나 이상의 테이블을 연결하여 데이터를 추출해야 하는 경우가 있다. 이 방법을 '조인'이라고 한다. 조인 수행 시 양쪽 테이블 간의 연결 조건을 정확히 지정하지 않으면 (n×m) 연산이 발생하여 아주 큰 양의 데이터를 출력하게 된다. 이는 성능 저하로 이어진다.

여기서 잠깐! | 조인

- 동등조인: 양쪽 테이블의 공통 컬럼을 기준으로 같은 행들을 연결한다.
- 비동등조인: 양쪽 테이블의 공통 컬럼을 기준으로 동등이 아닌 조건으로 연결한다.
- 셀프조인: 하나의 테이블에 공통된 컬럼이 두 개 이상 있는 경우의 동등조인을 수행한다.
- 외부조인: 동등조인에서 누락된 양쪽 테이블 정보를 출력하기 위한 조인으로 LEFT, RIGHT, FULL OUTER JOIN이 있다.
- 자연조인: 양쪽 테이블에서 데이터 타입 및 컬럼명이 같은 값이 존재할 때 자동으로 동등조인을 수행한다.
- USING 절: 두 개 이상의 공통 컬럼이 존재할 경우 하나의 컬럼만 동등조인하도록 지정해 주는 절이다.

조인의 종류는 크게 ANSI 표준 SQL과 오라클 전용 SQL로 분리하여 생각할 수 있다. 각각에 대해 실습해 보자.

샘플 문제

hr 스키마에 존재하는 Employees, Departments, Locations 테이블의 구조를 파악한 후 'Oxford'에 근무하는 사원의 성과 이름('Name'으로 별칭), 업무, 부서명, 도시명을 출력하시오. 이때 첫 번째 열은 회사명인 'Han-Bit'이라는 상숫값이 출력되도록 하시오(34행).

HINT 테이블의 각 구조를 파악하여 연결 가능한 컬럼을 찾아서 조인 연결 조건을 지정해 동등조건을 수행한다.

풀이

```
DESC employees ;
DESC departments ;
DESC locations ;

(ANSI 표준 SQL)
   SELECT   'Han-Bit', e.first_name ¦¦ ' ' ¦¦ e.last_name as "Name",
            e.job_id, d.department_name, l.city
   FROM           Employees e JOIN Departments d
            ON (e.department_id = d.department_id)
            JOIN locations l
            ON (d.location_id = l.location_id)
            AND l.city = 'Oxford';

(오라클 전용 SQL)
   SELECT   'Han-Bit', e.first_name ¦¦ ' ' ¦¦ e.last_name as "Name",
            e.job_id, d.department_name, l.city
   FROM     Employees e , Departments d , Locations l
   WHERE    e.department_id = d.department_id
   AND      d.location_id = l.location_id
   AND      l.city = 'Oxford';
```

	'HAN-BIT'	Name	JOB_ID	DEPARTMENT_NAME	CITY
1	Han-Bit	John Russell	SA MAN	Sales	Oxford
2	Han-Bit	Karen Partners	SA MAN	Sales	Oxford
3	Han-Bit	Alberto Errazuriz	SA MAN	Sales	Oxford
			... (생략) ...		
32	Han-Bit	Jonathon Taylor	SA REP	Sales	Oxford
33	Han-Bit	Jack Livingston	SA REP	Sales	Oxford
34	Han-Bit	Charles Johnson	SA REP	Sales	Oxford

문제 1

hr 스키마에 있는 Employees, Departments 테이블의 구조를 파악한 후 사원 수가 5명 이상인 부서의 부서명과 사원 수를 출력하시오. 이때 사원 수가 많은 순으로 정렬하시오(5행).

HINT 기본적으로 SQL 각 절의 사용 방법을 이해해야 한다. 요구하는 컬럼이 서로 다른 테이블에 있고, 동등한 조건으로 데이터를 처리해야 적합한 데이터를 출력할 수 있다. 또한 어떤 컬럼을 기준으로 그룹화하는지 확인하고, 집계 결과에 대한 조건 처리도 해야 한다.

풀이

	DEPARTMENT_NAME	COUNT(E.EMPLOYEE_ID)
1	Shipping	45
2	Sales	34
3	Purchasing	6
4	Finance	6
5	IT	5

문제 2

각 사원의 급여에 따른 급여 등급을 보고하려고 한다. 급여 등급은 Job_Grades 테이블에 표시된다. 해당 테이블의 구조를 살펴본 후 사원의 성과 이름('Name'으로 별칭), 업무, 부서명, 입사일, 급여, 급여 등급을 급여 등급의 오름차순으로 출력하시오(106행). 실습하기 전에 다음을 참고하여 Job_Grades 테이블을 먼저 생성하고, 비동등조인으로 문제를 풀어보시오.

HINT 정보를 요구하는 컬럼이 어떤 테이블에 존재하는지 확인하고 적합한 조인 방법을 선택한다.

```
[Job_Grades 테이블 생성]
CREATE TABLE Job_Grades (grade_level VARCHAR2(3), lowest_sal NUMBER,
       highest_sal NUMBER);
INSERT INTO Job_Grades VALUES('A', 1000, 2999);
INSERT INTO Job_Grades VALUES('B', 3000, 5999);
INSERT INTO Job_Grades VALUES('C', 6000, 9999);
INSERT INTO Job_Grades VALUES('D', 10000, 14999);
INSERT INTO Job_Grades VALUES('E', 15000, 24999);
INSERT INTO Job_Grades VALUES('F', 25000, 40000);
COMMIT;
SELECT * FROM JOB_GRADES;
```

풀이

	Name	JOB_ID	DEPARTMENT_NAME	HIRE_DATE	SALARY	GRADE_LEVEL
1	TJ Olson	ST CLERK	Shipping	07/04/10	2100	A
2	Steven Markle	ST CLERK	Shipping	08/03/08	2200	A
3	Michael Rogers	ST CLERK	Shipping	06/08/26	2900	A
4	Shelli Baida	PU CLERK	Purchasing	05/12/24	2900	A
5	Vance Jones	SH CLERK	Shipping	07/03/17	2800	A
6	Girard Geoni	SH CLERK	Shipping	08/02/03	2800	A
7	Mozhe Atkinson	ST CLERK	Shipping	05/10/30	2800	A
8	Sigal Tobias	PU CLERK	Purchasing	05/07/24	2800	A
9	John Seo	ST CLERK	Shipping	06/02/12	2700	A
10	Irene Mikkilineni	ST CLERK	Shipping	06/09/28	2700	A
11	Douglas Grant	SH CLERK	Shipping	08/01/13	2600	A
12	Donald OConnell	SH CLERK	Shipping	07/06/21	2600	A
13	Randall Matos	ST CLERK	Shipping	06/03/15	2600	A
14	Guy Himuro	PU CLERK	Purchasing	06/11/15	2600	A
15	Randall Perkins	SH CLERK	Shipping	07/12/19	2500	A
	... (생략) ...					

문제 3

각 사원과 직속 상사와의 관계를 이용하여 다음과 같은 형식의 보고서를 작성하고자 한다.

예 홍길동은 허균에게 보고한다 → Eleni Zlotkey report to Steven King

이 예를 참고하여 어떤 사원이 어떤 사원에게 보고하는지 출력하시오. 단, 보고할 상사가 없는 사원이 있다면 그 정보도 포함하여 출력하고, 상사의 이름은 대문자로 출력하시오(107행).

HINT 외부조인(OUTER JOIN)은 동등조인의 조건에서 누락된 정보를 포함하여 출력하고자 할 때 사용한다.

풀이

	Employee vs Manager
1	Neena Kochhar report to STEVEN KING
2	Lex De Haan report to STEVEN KING
3	Den Raphaely report to STEVEN KING
	... (생략) ...
105	Pat Fay report to MICHAEL HARTSTEIN
106	William Gietz report to SHELLEY HIGGINS
107	Steven King report to

6 부속질의

특정 조건은 반드시 조회를 통해 그 값을 확인한 다음 처리해야 하는 경우가 있다. 이때 부속질의를 이용하면 알 수 없는 조건 값을 대체하여 손쉽게 처리할 수 있다. 부속질의는 SQL의 SELECT, FROM, WHERE, HAVING 절을 비롯하여 DML과 DDL 문에도 사용할 수 있고, 연산자와 위치에 따라 여러 가지 형태로 만들어 활용할 수 있다. 실행 순서는 부속질의가 먼저 수행되고 그 결과를 주 질의에 전달하여 조건절에 따라 처리된다.

여기서 잠깐! | 부속질의의 종류

- 단일행 부속질의: 부속질의 결과 오직 하나의 행만 출력하여 전달한다.
- 다중행 부속질의: 부속질의 결과 하나 이상의 행을 출력하여 전달한다.
- 다중열 부속질의: 부속질의로부터 두 개 이상의 컬럼을 비교 출력하여 전달한다.
- 상관관계 부속질의: 바깥쪽 쿼리의 컬럼 중 하나가 안쪽 부속질의의 조건에 이용되는 처리 방식이다.
- FROM 절의 부속질의: 뷰처럼 동작하여 인라인 뷰라고 부르기도 한다. 하나의 테이블 데이터가 많을 경우 FROM 절에 테이블 전체를 기술하면 효율이 떨어질 수 있다. 이 경우 필요한 행과 열만 선택하여 FROM 절에 기술하면 오라클 서버의 최적화 단계에서 효율적인 검색을 할 수 있다.
- 스칼라 부속질의: 하나의 행에서 하나의 열값만 반환하는 부속질의를 말한다.

샘플 문제

HR 부서의 어떤 사원은 급여 정보를 조회하는 업무를 맡고 있다. Tucker(last_name) 사원보다 급여를 많이 받고 있는 사원의 성과 이름('Name'으로 별칭), 업무, 급여를 출력하시오(15행).

HINT Tucker의 급여를 알 수 없기 때문에 해당 조건 값 대신 부속질의를 활용하여 조건 처리를 한다.

풀이

```
SELECT  first_name || ' ' || last_name, job_id, salary
FROM    Employees
WHERE   salary > (SELECT salary
                  FROM   Employees
                  WHERE  last_name = 'Tucker');
```

	FIRST_NAME||''||LAST_NAME	JOB_ID	SALARY
1	Steven King	AD PRES	24000
2	Neena Kochhar	AD VP	17000
3	Lex De Haan	AD VP	17000
	... (생략) ...		
13	Ellen Abel	SA REP	11000
14	Michael Hartstein	MK MAN	13000
15	Shelly Higgins	AC MGR	12008

문제 1

사원의 급여 정보 중 업무별 최소 급여를 받고 있는 사원의 성과 이름('Name'으로 별칭), 업무, 급여, 입사일을 출력하시오(21행).

HINT 부속질의에는 그룹 데이터 처리가 가능하다.

풀이

	Name	JOB_ID	SALARY	HIRE_DATE
1	Steven King	AD PRES	24000	03/06/17
2	Neena Kochhar	AD VP	17000	05/09/21
3	Lex De Haan	AD VP	17000	01/01/13
		... (생략) ...		
19	Hermann Baer	PR REP	10000	02/06/07
20	Shelley Higgins	AC MGR	12008	02/06/07
21	William Gietz	AC ACCOUNT	8300	02/06/07

문제 2

소속 부서의 평균 급여보다 많은 급여를 받는 사원의 성과 이름('Name'으로 별칭), 급여, 부서번호, 업무를 출력하시오(38행).

HINT 상관관계 부속질의는 바깥쪽 쿼리의 컬럼이 안쪽 부속질의에 사용되는 경우로, 테이블에서 행을 먼저 읽어서 각 행의 값을 관련된 데이터와 비교하는 방법 중 하나이다.

풀이

	Name	SALARY	DEPARTMENT_ID	JOB_ID
1	Steven King	24000	90	AD PRES
2	Alexander Hunold	9000	60	IT PROG
3	Bruce Ernst	6000	60	IT PROG
		... (생략) ...		
36	Britney Everett	3900	50	SH CLERK
37	Michael Hartstein	13000	20	MK MAN
38	Shelley Higgins	12008	110	AC MGR

문제 3

사원들의 지역별 근무 현황을 조회하고자 한다. 도시 이름이 영문 'O'로 시작하는 지역에 살고 있는 사원의 사번, 성과 이름('Name'으로 별칭), 업무, 입사일을 출력하시오(34행).

HINT 요구사항에 나타나는 정보에 대해 조인을 이용하지 않고 하나하나 연결하여 부속질의로 처리할 수 있다. 도시정보와 부서정보, 부서정보와 사원정보는 서로 연관이 가능한 구조이므로 부속질의로 처리 가능하다.

풀이

	EMPLOYEE_ID	FIRST_NAME||''||LAST_NAME	JOB_ID	HIRE_DATE
1	145	John Russell	SA MAN	04/10/01
2	146	Karen Partners	SA MAN	05/01/05
3	147	Alberto Errazuriz	SA MAN	05/03/10
... (생략) ...				
32	176	Jonathon Taylor	SA REP	06/03/24
33	177	Jack Livingston	SA REP	06/04/23
34	179	Charles Johnson	SA REP	08/01/04

문제 4

모든 사원의 소속 부서 평균연봉을 계산하여 사원별로 성과 이름('Name'으로 별칭), 업무, 급여, 부서번호, 부서 평균연봉('Department Avg Salary'로 별칭)을 출력하시오. 급여의 오름차순으로 출력하시오(107행).

HINT 스칼라 서브쿼리는 부속질의의 SELECT 절에 포함되는 항목 값이며, 하나의 행에서 하나의 컬럼 값만 출력하는 부속질의이다.

풀이

	Name	JOB_ID	SALARY	DEPARTMENT_ID	Department Avg Salary
1	TJ Olson	ST CLERK	2100	50	3476
2	Steven Markle	ST CLERK	2200	50	3476
3	Hazel Philtanker	ST CLERK	2200	50	3476
4	Ki Gee	ST CLERK	2400	50	3476
5	James Landry	ST CLERK	2400	50	3476
... (생략) ...					

7 집합 연산자

데이터베이스에서 테이블은 행과 열의 집합이다. 이는 테이블을 집합 개념으로도 처리 가능하다는 의미이다. 쿼리의 결과값은 합집합, 교집합, 차집합을 이용하여 하나의 결과로 통합할 수 있다. 테이블에 집합 연산을 사용하려면 표현식의 개수와 데이터 타입이 서로 일치해야 한다. 특히 ORDER BY 절은 명령문 끝에만 사용할 수 있다.

여기서 잠깐! | 집합 연산자의 종류

- UNION: 두 쿼리에서 중복된 행을 배제한 쿼리의 통합 결과를 출력한다.
- UNION ALL: 두 쿼리에서 중복된 행을 포함한 쿼리의 통합 결과를 출력한다.
- INTERSECT: 두 쿼리에 공통으로 선택된 값 중 중복된 값을 배제한 통합 결과를 출력한다.
- MINUS: 첫 번째 비교 쿼리문에서 선택되고, 두 번째 비교 쿼리문에서 선택되지 않는 중복된 값을 배제한 결과를 출력한다. 쿼리의 순서에 따라 결과는 달라진다.

샘플 문제

hr 스키마에 있는 Job_History 테이블은 사원들의 업무 변경 이력을 나타내는 테이블이다. 이 정보를 이용하여 모든 사원의 현재 및 이전의 업무 이력 정보를 출력하고자 한다. 중복된 튜플이 있을 경우 한 번만 표시하여 출력하시오(115행).

HINT Employees, Job_History 테이블의 구조와 공통적으로 출력할 수 있는 컬럼을 확인한 후 중복된 사원정보를 배제하기 위해 UNION 연산자를 이용하여 처리한다.

풀이

```
SELECT  employee_id, job_id
FROM    Employees
UNION
SELECT  employee_id, job_id
FROM    Job_History;
```

	EMPLOYEE_ID	JOB_ID
1	206	AC ACCOUNT
2	205	AC MGR
3	200	AD ASST
	... (생략) ...	
113	176	SA MAN
114	114	ST CLERK
115	122	ST CLERK

문제 1

앞의 [샘플 문제]에서 각 사원의 업무 이력 정보를 확인하였다. 하지만 모든 사원의 업무 이력 전체를 보지는 못했다. 여기에서는 모든 사원의 업무 이력 변경 정보 및 업무 변경에 따른 부서정보를 사번이 빠른 순서대로 출력하시오(107행).

HINT UNION ALL 연산자는 중복된 정보를 포함한 합집합을 출력한다.

풀이

	Name	JOB_ID	SALARY	DEPARTMENT_ID	Department Avg Salary
1	TJ Olson	ST CLERK	2100	50	3476
2	Steven Markle	ST CLERK	2200	50	3476
3	Hazel Philtanker	ST CLERK	2200	50	3476
			... (생략) ...		
105	Lex De Haan	AD VP	17000	90	19333
106	Neena Kochhar	AD VP	17000	90	19333
107	Steven King	AD PRES	24000	90	19333

문제 2

사원정보(Employees) 테이블의 job_id는 사원의 현재 업무를 뜻하고, Job_History의 job_id는 사원의 이전 업무를 뜻한다. 이 두 테이블을 비교해 보면 어떤 사원의 업무가 변경되었고 그대로 유지되고 있는지 확인할 수 있다. 두 테이블을 이용하여 업무 변경 없이 같은 업무를 하고 있는 사원의 사번과 업무를 출력하시오(2행).

HINT INTERSECT 연산자는 양쪽 집합의 공통된 정보를 출력할 수 있다.

풀이

	EMPLOYEE_ID	JOB_ID
1	200	AD ASST
2	176	SA REP

문제 2-1

앞의 [문제 2]의 결과를 이용하여 출력된 101번 사원의 현재 업무와 업무 이력의 변경 날짜 이력을 조회하시오(3행).

HINT 집합 구현 시 양쪽 테이블의 컬럼 수 및 데이터 타입은 반드시 같아야 한다. 만일 공통된 컬럼이 없을 경우 NULL 처리를 하고 필요에 따라 데이터 타입을 변환하여 일치시킨다.

풀이

	EMPLOYEE_ID	JOB_ID	Start Date	End Date
1	101	AD VP	(null)	(null)
2	101	AC ACCOUNT	97/09/21	01/10/27
3	101	AC MGR	01/10/28	05/03/15

문제 3

우리 회사는 1년에 한 번 업무를 변경하여 전체적인 회사 업무를 사원들이 익히도록 하는 Role change 정책을 시행하고 있다. 이번 인사 이동 때 아직 업무가 변경된 적이 없는 사원들을 적합한 업무로 이동시키려고 한다. HR 부서의 사원정보 테이블과 업무 이력 정보 테이블을 이용하여 한 번도 업무가 변경되지 않은 사원의 사번을 출력하시오(100행).

HINT MINUS 연산자를 이용한다.

풀이

	EMPLOYEE_ID
1	100
2	103
3	104
4	105
5	106
... (생략) ...	

8 조건부 논리 표현식 제어: CASE & DECODE

프로그램 작성에 큰 역할을 하는 구문 중에 IF-THEN-ELSE 조건 표현식이 있다. 하지만 SQL에서는 직접적인 작성이 불가능하기 때문에 CASE와 DECODE 함수를 이용한다. CASE 구문은 ANSI SQL 표준 구문(표현식)이고, DECODE는 오라클 전용 함수이다. 언뜻 보기에 같아 보이지만 분명한 차이점이 있다. CASE 구문은 대부분의 비교 연산(<, >, = 등) 및 검색형 조건 비교 처리가 가능한 반면, DECODE 함수는 '='을 이용한 동등 비교 연산만 가능하고 검색형 조건 비교 처리는 불가능하다.

여기서 잠깐! | SQL의 조건 표현식 CASE & DECODE

IF-THEN-ELSE 조건 표현식을 SQL에서 구현하기 위해 CASE와 DECODE를 선택적으로 사용할 수 있다.

❶ CASE 구문

단순 비교 처리뿐만 아니라 검색 조건을 이용한 비교 처리를 수행할 수 있다.

```
CASE 대상값  WHEN 비교값1 THEN 처리1
             WHEN 비교값2 THEN 처리2
             ...
             [ELSE 디폴트 처리]
END
```

▲ 단순 비교(=) 처리 구문

```
CASE  WHEN 비교조건1 THEN 처리1
      WHEN 비교조건2 THEN 처리2
      ...
      [ELSE 디폴트 처리]
END
```

▲ 검색 조건을 이용한 비교 처리 구문

❷ DECODE 함수

대상값이 search1이면 result1을, search2면 result2를 출력하고, 모든 조건에 해당하지 않으면 default_result를 출력한다. 만약 default_result를 생략하면 Null을 출력한다.

```
DECODE(대상값, search1, result1, search2, result2, ..., default_result)
```

샘플 문제 1

HR 부서에서는 신규 프로젝트를 성공으로 이끈 해당 업무자들의 급여를 인상하기로 결정하였다. 사원은 현재 107명이며 19개의 업무에 소속되어 근무 중이다. 급여 인상 대상자는 회사의 업무(Distinct job_id) 중 다음 5개 업무에서 일하는 사원에 해당된다. 나머지 업무에 대해서는 급여가 동결된다. 5개 업무의 급여 인상안은 다음과 같다(107행).

HR_REP (10%), MK_REP (12%), PR_REP (15%), SA_REP (18%), IT_PROG (20%)

HINT CASE와 DECODE를 이용하여 조건을 만족하는 구문을 작성해 본다.

풀이

```
-- CASE
SELECT  employee_id, last_name || ' ' || first_name as "Name", job_id, salary,
        CASE job_id
             WHEN 'HR_REP' THEN 1.10 * salary
             WHEN 'MK_REP' THEN 1.12 * salary
             WHEN 'PR_REP' THEN 1.15 * salary
             WHEN 'SA_REP' THEN 1.18 * salary
             WHEN 'IT_PROG' THEN 1.20 * salary
             ELSE salary
        END  "New Salary"
FROM    Employees;

-- DECODE
SELECT  employee_id, last_name || ' ' || first_name as "Name", job_id, salary,
        DECODE (job_id, 'HR_REP', 1.10 * salary
             , 'MK_REP' , 1.12 * salary
             , 'PR_REP' , 1.15 * salary
             , 'SA_REP' , 1.18 * salary
             , 'IT_PROG' , 1.20 * salary
             , salary) "New Salary"
FROM    Employees;
```

	EMPLOYEE_ID	Name	JOB_ID	SALARY	New Salary
1	100	King Steven	AD PRES	24000	24000
2	101	Kochhar Neena	AD VP	17000	17000
3	102	De Haan Lex	AD VP	17000	17000
		... (생략) ...			
105	204	Baer Hermann	PR REP	10000	11500
106	205	Higgins Shelley	AC MGR	12008	12008
107	206	Gietz William	AC ACCOUNT	8300	8300

샘플 문제 2

HR 부서에서는 최상위 입사일에 해당하는 2001년부터 2003년까지 입사자들의 급여를 각각 5%, 3%, 1% 인상하여 지분에 따른 배당금으로 지급하고자 한다. 전체 사원들 중 해당 연도에 해당하는 사원들의 급여를 검색하여 적용하고, 입사일자에 따른 오름차순 정렬을 수행하시오(107행).

HINT CASE 구문을 이용한 검색 조건 비교를 이용하여 해당 연도별 조건 처리를 수행한다.

풀이

```
SELECT employee_id, last_name ||' '|| first_name as "Name", hire_date, salary,
       CASE WHEN hire_date < TO_DATE('2002-01-01', 'YYYY-MM-DD')
            THEN salary * 1.05
            WHEN hire_date < TO_DATE('2003-01-01', 'YYYY-MM-DD')
            THEN salary * 1.03
            WHEN hire_date < TO_DATE('2004-01-01', 'YYYY-MM-DD')
            THEN salary * 1.01
            ELSE salary
       END  "New Salary"
FROM   Employees
ORDER BY hire_date;
```

	EMPLOYEE_ID	Name	HIRE_DATE	SALARY	New Salary
1	102	De Haan Lex	01/01/13	17000	17850
2	206	Gietz William	02/06/07	8300	8549
3	204	Baer Hermann	02/06/07	10000	10300
4	203	Mavris Susan	02/06/07	6500	6695
5	205	Higgins Shelley	02/06/07	12008	12368.24
... (생략) ...					

문제 1

부서별 급여 합계를 구하고, 그 결과를 다음과 같이 표현하시오(12행).

- Sum Salary > 100000이면, 'Excellent'
- Sum Salary > 50000이면, 'Good'
- Sum Salary > 10000이면, 'Medium'
- Sum Salary <= 10000이면, 'Well'

HINT 우선 인라인 뷰를 이용하여 부서별 급여 합계를 구하고, 상위 쿼리에서 CASE 구문을 이용하여 조건 비교를 통해 급여 합계에 따른 각각의 표현을 할 수 있다.

풀이

	DEPARTMENT_ID	SUM_SAL	Department Grade Salary
1	90	58000	Good
2	60	28800	Medium
3	100	51608	Good
4	30	24900	Medium
5	50	156400	Excellent
6	80	304500	Excellent
7	(null)	7000	Well
8	10	4400	Well
9	20	19000	Medium
10	40	6500	Well
11	70	10000	Well
12	110	20308	Medium

문제 2

2005년 이전에 입사한 사원 중 업무에 'MGR'이 포함된 사원은 급여를 15% 인상하고, 'MAN'이 포함된 사원은 급여를 20% 인상한다. 또한 2005년부터 근무를 시작한 사원 중 'MGR'이 포함된 사원은 급여를 25% 인상하는 쿼리를 작성하시오. 단, 조건에 해당하는 사원들만 출력한다(7행).

HINT IF-THEN-ELSE 표현식에는 다중 조건을 처리하는 다중 IF 구문이 있다. CASE 또는 DECODE도 마찬가지로 다중 조건에 대한 처리가 가능하다. 다만, 동등 조건으로 처리가 가능한지, 또는 가능하지 않은지에 따라 CASE와 DECODE를 선택하면 된다.

풀이

	EMPLOYEE_ID	Name	JOB_ID	HIRE_DATE	SALARY	Department Grade Salary
1	108	Greenberg Nancy	FI MGR	02/08/17	12008	13809.2
2	114	Raphaely Den	PU MAN	02/12/07	11000	13200
3	120	Weiss Matthew	ST MAN	04/07/18	8000	9600
4	122	Kaufling Payam	ST MAN	03/05/01	7900	9480
5	145	Russell John	SA MAN	04/10/01	14000	16800
6	201	Hartstein Michael	MK MAN	04/02/17	13000	15600
7	205	Higgins Shelley	AC MGR	02/06/07	12008	13809.2

문제 3

월별로 입사한 사원 수를 다음과 같은 2가지 방식으로 출력하시오.

(방식 1) 월별로 입사한 사원 수가 다음과 같이 각 행별로 출력되도록 하시오(12행).

(방식 2) 첫 행에 모든 월별 입사 사원 수가 출력되도록 하시오(1행).

HINT DECODE, CASE 함수는 IF THEN ELSE 로직을 구현할 수 있도록 만든 함수이다. 기본 IF 구문의 흐름처럼 조건에 맞는 행은 각 DECODE 절에서 COUNT(*)하여 구하고, 아닌 것은 0으로 처리하는 IF 구문을 작성하는 것과 유사하다. 1~12월까지의 IF 구문이 있다고 보고, 각 IF 문, 즉 각 DECODE 함수의 합을 구하면 된다.

풀이

(방식 1)

	1 Month	2 Month	3 Month	4 Month	5 Month	6 Month	7 Month	8 Month	9 Month	10 Month	11 Month	12 Month
1	14	0	0	0	0	0	0	0	0	0	0	0
2	0	13	0	0	0	0	0	0	0	0	0	0
3	0	0	17	0	0	0	0	0	0	0	0	0
4	0	0	0	7	0	0	0	0	0	0	0	0
5	0	0	0	0	6	0	0	0	0	0	0	0
6	0	0	0	0	0	11	0	0	0	0	0	0
7	0	0	0	0	0	0	7	0	0	0	0	0
8	0	0	0	0	0	0	0	9	0	0	0	0
9	0	0	0	0	0	0	0	0	5	0	0	0
10	0	0	0	0	0	0	0	0	0	6	0	0
11	0	0	0	0	0	0	0	0	0	0	5	0
12	0	0	0	0	0	0	0	0	0	0	0	7

(방식 2)

	1 Month	2 Month	3 Month	4 Month	5 Month	6 Month	7 Month	8 Month	9 Month	10 Month	11 Month	12 Month
1	14	13	17	7	6	11	7	9	5	6	5	7

9 다차원 그룹 데이터 검색: ROLLUP, CUBE

ROLLUP과 CUBE는 본문에서 배우지 않았지만 데이터 웨어하우스data warehouse를 구축하는 데 많이 사용하는 쿼리 기법이다. GROUP BY 절에 ROLLUP을 지정하면 그룹화 대상의 소계와 합계를 N+1개 구해 주고, CUBE를 지정하면 가능한 모든 그룹의 조합을 만들어 소계와 합계를 2N개 생성해 준다. 이 기능은 GROUP BY 절의 확장된 형태로 사용하기 쉬우며 다중 컬럼 지정이 가능하기 때문에 매우 효과적일 뿐만 아니라 계층적 분류를 포함하고 있는 데이터의 집계에 적합하다.

여기서 잠깐! | 그룹 함수 정리

- ROLLUP: 지정된 컬럼들을 오른쪽 → 왼쪽 순서로 그룹을 만들어 각 그룹에 대한 집계를 수행한다.
- CUBE: 지정된 모든 컬럼에 대해 가능한 모든 조합 그룹을 만들어 집계를 수행한다.
- GROUPING: 지정된 컬럼이 GROUP BY 절에 사용되었는지 여부를 0(사용됨)과 1(사용 안 됨)로 표시한다.
- GROUPING SETS: 여러 개의 GROUP BY 절을 포함한 쿼리에 대해 UNION ALL 처리와 동일한 통합 결과 집합을 생성한다.

샘플 문제 1

HR 부서에서는 부서와 업무별 급여 합계를 구하여 신년도 급여 수준 레벨을 지정하고자 한다. 부서 번호와 업무를 기준으로 전체 행을 그룹별로 나누어 급여 합계와 인원수를 출력하시오(20행).

풀이

```
SELECT   department_id, job_id,
         TO_CHAR(SUM(salary),'$999,999.00') as "Salary SUM",
         COUNT(employee_id) as "COUNT EMPs"
FROM     Employees
GROUP BY department_id, job_id
ORDER BY department_id;
```

	DEPARTMENT_ID	JOB_ID	Salary SUM	COUNT EMPs
1	10	AD ASST	$4,400.00	1
2	20	MK MAN	$13,000.00	1
3	20	MK REP	$6,000.00	1
4	30	PU CLERK	$13,900.00	5
5	30	PU MAN	$11,000.00	1
... (생략) ...				

샘플 문제 2

앞의 [샘플 문제 1] 결과에서 부서번호와 업무를 기준으로 전체 행을 그룹별로 나누어 급여 합계와 인원수를 출력하고, 부서번호와 업무 각각에 대하여 그룹화된 합계와 인원수를 출력하시오(33행).

HINT ROLLUP은 지정된 컬럼 수(N)+1개의 결과 집합을 생성한다.

풀이

```
SELECT   department_id, job_id,
         TO_CHAR(SUM(salary),'$999,999.00') as "Salary SUM",
         COUNT(employee_id) as "COUNT EMPs"
FROM     Employees
GROUP BY ROLLUP(department_id, job_id)
ORDER BY department_id;
```

	DEPARTMENT_ID	JOB_ID	Salary SUM	COUNT EMPs
1	10	AD ASST	$4,400.00	1
2	10	(null)	$4,400.00	1
3	20	MK_MAN	$13,000.00	1
4	20	MK_REP	$6,000.00	1
5	20	(null)	$19,000.00	2
		... (생략) ...		
28	110	AC_ACCOUNT	$8,300.00	1
29	110	AC_MGR	$12,008.00	1
30	110	(null)	$20,308.00	2
31	(null)	SA_REP	$7,000.00	1
32	(null)	(null)	$7,000.00	1
33	(null)	(null)	$691,416.00	107

부서 20, job: MK_REP의 group by 결과

부서: 20의 group by 결과

부서: null, job: SA_REP의 group by 결과

부서: null의 group by 결과

모든 자료의 group by 결과

TIP DEPARTMENT_ID의 null 값은 부서가 지정되지 않은 대기발령자이다.

문제 1

앞의 [샘플 문제]의 결과를 보면 job_id에 대한 집계는 나타나지 않았다. 업무에 대해서도 샘플과 같은 총집계가 이루어지도록 쿼리를 수정해 보시오(52행).

HINT CUBE는 가능한 모든 차원의 결과 집합(2N)을 만드는 상위 집계를 출력한다.

풀이

	DEPARTMENT_ID	JOB_ID	Salary SUM	COUNT EMPs
1	10	AD ASST	$4,400.00	1
2	10	(null)	$4,400.00	1
3	20	MK MAN	$13,000.00	1
4	20	MK REP	$6,000.00	1
5	20	(null)	$19,000.00	2
		... (생략) ...		
47	(null)	SA_REP	$250,500.00	30
48	(null)	SH_CLERK	$64,300.00	20
49	(null)	ST_CLERK	$55,700.00	20
50	(null)	ST_MAN	$7,000.00	1
52	(null)	(null)	$691,416.00	107

문제 2

GROUPING 함수를 이용하여 앞에서 나온 결과들이 GROUP BY 절에 의한 결과인지 여부를 식별하시오(52행).

HINT GROUPING 함수는 ROLLUP 또는 CUBE 집계 결과에서 특정 컬럼 값이 집계에 의해 생성된 NULL인지 여부를 0 또는 1로 반환한다. 0은 실제 그룹 값, 1은 집계로 생성된 행을 의미한다.

풀이

	DEPARTMENT_ID	JOB_ID	Salary SUM	COUNT EMPs	Dept#	Job ID
1	10	AD ASST	$4,400.00	1	0	0
2	10	(null)	$4,400.00	1	0	1
3	20	MK MAN	$13,000.00	1	0	0
4	20	MK REP	$6,000.00	1	0	0
5	20	(null)	$19,000.00	2	0	1
... (생략) ...						

여기서 잠깐! | ROLLUP과 CUBE의 차이

ROLLUP은 N+1개의 부분집합을 생성하고, CUBE는 2N의 부분집합을 생성한다(N은 그룹 기준 컬럼 수).

- group by cube(a, b, c);
  = group by grouping sets((a, b, c), (a, b), (b, c), (a, c), (a), (b), (c), ());
- group by rollup(a, b, c);
  = group by grouping sets((a, b, c), (a, b), (a), ());

문제 3

앞의 [샘플 문제]의 결과를 근거로 부서에 대한 집계 결과가 아니면 (ALL-DEPTS), 업무에 대한 집계 결과가 아니면 (ALL-JOBS)라고 출력하며, 해당 집계에 대한 사원수와 평균연봉을 구하시오(33행).

HINT GROUPING 함수가 갖는 0과 1 기능을 이용하여 DECODE를 이용한 조건부 처리를 수행한다.

풀이

	Dept#	Jobs	Count Emps	Avg Ann_sal
1	90	AD PRES	1	288000
2	90	AD VP	2	204000
3	60	IT PROG	5	69120
4	100	FI MGR	1	144096
5	100	FI ACCOUNT	5	95040
... (생략) ...				

10 분석 함수: NTILE, RANK

기존 RDBMS에서는 컬럼과 컬럼 간의 연산, 비교, 연결은 쉽게 처리되는 반면 행과 행 간의 관계 정의나 비교, 연산 등은 난해한 문제였다. 이러한 부분은 분석 함수analytic function를 이용할 경우 하나의 SQL로 쉽게 처리할 수 있다. 분석 함수는 쿼리의 결과 집합을 기준으로 각 그룹 내에서 Window 범위에 포함된 행들을 기준으로 각 행마다 계산 값을 생성해 준다. 일

반적으로 누적 계산, 집계 및 보고용 결과를 출력할 때 유용하게 사용할 수 있으며 복잡한 쿼리를 보다 간편하고 빠르게 실행할 수 있도록 도와준다.

```
SELECT
        분석 함수(파라미터1, 파라미터2, ...)
        OVER([PARTITION BY 컬럼] [ORDER BY 절] [WINDOWING 절])
FROM    테이블명;
```

- 분석 함수(파라미터) : 분석 함수에 값을 전달하여 처리하는 것을 '파라미터'라고 한다. 파라미터는 보통 0~3개 정도 사용될 수 있다.
- PARTITION BY : 동일 그룹으로 묶어 줄 컬럼명을 지정한다.
- ORDER BY : PARTITION BY 절에 지정된 컬럼에 대한 정렬을 수행한다.
- WINDOWING : PARTITION BY 절로 묶인 그룹을 또 다른 소그룹으로 나누는 역할을 한다.

여기서 잠깐! | 분석 함수의 종류

분석 함수는 그 역할과 용도에 따라 다음과 같은 종류가 있다.

- NTILE(): PARTITION을 bucket이라 불리는 그룹으로 나누고 PARTITION 내의 각 행들을 bucket에 배치하는 함수이다. 각 bucket에는 동일한 수의 행이 배치된다.
- RANK(): 값이 같은 경우 동일한 순위를 부여하며, 이후 순위는 앞선 동등 순위 개수를 고려하여 건너뛴 순위 값으로 출력한다.
- DENSE_RANK(): 동등 순위 번호는 같게 나오고 그다음 순위를 다음 번호로 출력한다.
- ROW_NUMBER(): 동등 순위를 인식하지 않고 매번 증가되는 번호를 출력한다.
- LEAD() / LAG(): 지정된 컬럼의 이전, 이후의 행 값을 출력한다.
- FIRST() / LAST(): 각 그룹별 첫 번째와 마지막 값 하나만 출력한다.

샘플 문제

분석 함수 NTILE()을 이용하여 부서별 급여 합계를 구하시오. 단, 급여 합계가 제일 큰 것이 1, 제일 작은 것이 4가 되도록 bucket(등급)을 나누어 출력하고, bucket(등급)을 기준으로 오름차순으로 정렬하시오(12행).

HINT NTILE() 함수는 지정된 값을 bucket(등급)으로 나누어 저장한다.

풀이

```
SELECT   department_id, SUM(salary) as "Sum Salary",
         NTILE (4) over (order by SUM(salary) DESC) as "Bucket#"
FROM     Employees
GROUP BY department_id
ORDER BY 3;
```

	DEPARTMENT_ID	Sum Salary	Bucket#
1	50	156400	1
2	80	304500	1
3	90	58000	1
4	60	28800	2
5	100	51608	2
... (생략) ...			

문제 1

각 사원이 소속된 부서별로 급여를 기준으로 내림차순 정렬하시오. 이때 다음 세 가지 함수를 이용하여 순위를 다음과 같이 출력하시오(107행).

- RANK(): 동등 순위가 발생하면 다음 순위는 중복된 값만큼 증가시켜 매긴다.
- DENSE_RANK(): 동등 순위가 발생하면 다음 순위는 중복된 값을 무시하고 바로 다음 번호로 매긴다.
- ROW_NUMBER(): 동등 순위 자체를 인식하지 않고 매번 번호를 증가시켜 매긴다.

HINT RANK(), DENSE_RANK(), ROW_NUMBER() 함수의 차이점을 이해하고 작성한다.

풀이

	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID	Rank	Dense_Rank	Row_Number
1	200	Whalen	4400	10	1	1	1
2	201	Hartstein	13000	20	1	1	1
3	202	Fay	6000	20	2	2	2
4	114	Raphaely	11000	30	1	1	1
5	115	Khoo	3100	30	2	2	2
... (생략) ...							

문제 2

지정된 개수의 이전, 이후 행의 값을 가져오는 LAG(), LEAD() 함수를 이용하여 50번 부서의 사원정보를 급여순으로 내림차순 정렬하고 이전, 다음 행의 급여를 함께 출력하시오(45행).

HINT LAG() 함수를 통해 해당 행의 salary 컬럼의 값을 한 칸씩 밑에 할당하고, 반대로 LEAD() 함수를 통해 한 칸씩 위로 할당하면 된다.

- LAG(salary, 1, 0): 1(한 칸씩 아래 설정), 0(받을 값이 없을 경우 NULL 출력 대신 0 출력)

풀이

	EMPLOYEE_ID	LAST_NAME	SALARY	Prev_Sal	Next_Sal
1	121	Fripp	8200	0	8000
2	120	Weiss	8000	8200	7900
3	122	Kaufling	7900	8000	6500
4	123	Vollman	6500	7900	5800
5	124	Mourgos	5800	6500	4200
... (생략) ...					

문제 3

FIRST(), LAST() 함수를 이용하여 사원이 소속된 부서별로 급여를 가장 많이 받거나 가장 적게 받는 사원의 이름을 출력하시오(12행).

HINT FIRST(), LAST() 함수를 통해 가장 크거나 작은 값을 구할 수 있다.

풀이

	DEPARTMENT_ID	COUNT(*)	Max_Emp_name	Min_Emp_name
1	10	1	Whalen	Whalen
2	20	2	Hartstein	Fay
3	30	6	Raphaely	Colmenares
4	40	1	Mavris	Mavris
5	50	45	Fripp	Olson
6	60	5	Hunold	Lorentz
7	70	1	Baer	Baer
8	80	34	Russell	Kumar
9	90	3	King	De Haan
10	100	6	Greenberg	Popp
11	110	2	Higgins	Gietz
12	(null)	1	Grant	Grant

11 계층 검색

계층 검색은 본문에서 배우지는 않았지만 유용한 질의 기법이다. 사원 테이블에서 사원의 상

관의 상관을 찾거나 가족 관계 테이블에서 부모의 부모를 찾는 등 트리 구조 데이터가 릴레이션에 저장되어 있을 때 트리의 조상이나 자손을 찾을 수 있다. 계층 검색hierarchical query을 이용하면 트리Tree 형태의 보고서를 생성할 수 있다. 일반적으로 RDBMS는 행을 저장할 때 계층적인 방법으로 저장하지 않는다. 그러나 행들 사이에 어떤 계층적 관련성이 있다면 트리 형태의 탐색으로 계층을 만들 수 있다.

여기서 잠깐! | 계층형 쿼리

계층형 쿼리는 탐색의 시작과 방향 및 조건을 이용하여 가지 제거 등의 기능을 구현한다.

- START WITH: 계층 탐색의 시작 지점을 지정한다.
- CONNECT BY: 탐색의 방향을 아래 → 위, 위 → 아래로 지정한다.
- LEVEL: 오라클 지정 의사열을 통해 트리 구조의 깊이, 즉 행 등급이나 레벨을 명시적으로 지정하여 보고서를 더욱 읽기 쉽게 해 준다.
- PRIOR: CONNECT BY 절에 지정된 컬럼 중 PRIOR로 지정된 컬럼 바로 다음에 오는 행부터 시작하여 검색한다.
- 조건: 조건의 위치(WHERE 절/CONNECT BY 절)에 따라 특정 행, 또는 그 하위의 분기를 모두 제거할 수 있다.
- 관련 함수: SYS_CONNECT_BY_PATH()

샘플 문제

HR 부서에서는 신규 채용 이후 회사 조직도를 새로 작성하고자 한다. 사원정보(Employees) 테이블을 이용하여 대표이사부터 최하위 사원까지의 보고 체계를 다음 예시와 같은 형태로 출력하시오(108행).

예 Kochhar reports to King

HINT START WITH 절은 트리 탐색의 시작점을 지정하고, CONNECT BY PRIOR 절은 탐색의 방향을 결정한다.

풀이

```
SELECT   last_name || ' reports to ' || PRIOR last_name "Walk"
FROM     employees
START WITH last_name = 'King'
CONNECT  BY PRIOR employee_id = manager_id;
```

	Walk
1	King reports to
2	King reports to
3	Kochhar reports to King
4	Greenberg reports to Kochhar
5	Faviet reports to Greenberg
6	Chen reports to Greenberg
7	Sciarra reports to Greenberg
8	Urman reports to Greenberg
... (생략) ...	

문제 1

전체 조직도를 앞의 [샘플 문제]를 통해 확인한 후 가장 급여를 적게 받는 사원 'Olson'의 보고 체계를 'Olson'부터 시작하여 직속으로 최상위까지 출력하시오(3행).

HINT PRIOR 절의 이동으로 탐색의 방향을 바꿀 수 있다.

풀이

	Walk
1	reports to Olson
2	Olson reports to Fripp
3	Fripp reports to King

문제 2

LEVEL 의사 열(pseudo column)을 이용하여 사원들의 전체 계층의 행 등급, 즉 레벨을 명시적으로 출력하시오(107행).

HINT LEVEL 의사 열을 사용하면 계층 구조에서 각 행의 등급이나 레벨을 표현할 수 있다. 또한 계층적 보고서를 들여쓰기 형태의 트리 구조로 출력하기 위해서는 LPAD( ) 함수를 활용한다.

풀이

	Hierarchical Query
1	King
2	Kochhar
3	Greenberg
4	Faviet
5	Chen
	... (생략) ...
105	Johnson
106	Hartstein
107	Fay

문제 2-1

앞의 출력 내용 중 'Kochhar'을 제외하고 출력하시오(95행).

HINT 조건 지정을 WHERE 절로 하면 해당 행만 제거되고, CONNECT BY 절에 추가 시 해당 분기(가지)가 모두 제거된다.

풀이

	Hierarchical Query
1	King
2	King
3	==De Haan
4	====Hunold

5	======Ernst
6	======Austin
7	======Pataballa
	... (생략) ...

문제 3

다음 결과와 같이 사원의 이름, 부서명, 업무를 합쳐서 출력하고자 하는 계층형 쿼리에 대해 [문제 2]를 변형하여 작성하시오(107행).

HINT 모든 함수와 연산자, 문자열은 연결할 수 있다.

풀이

	Hierarchical Query
1	★King job is SA REP, department name is Sales
2	★King job is AD PRES, department name is Executive
3	★★★★Kochhar job is AD VP, department name is Executive
4	★★★★★★★Greenberg job is FI MGR, department name is Finance
5	★★★★★★★★★★Faviet job is FI ACCOUNT, department name is Finance
6	★★★★★★★★★★Chen job is FI ACCOUNT, department name is Finance
7	★★★★★★★★★★Sciarra job is FI ACCOUNT, department name is Finance
	... (생략) ...

문제 4

앞의 문제와 다르게 LEVEL 의사 열(pseudo column)을 사용하지 않고 각 사원의 전체 직속 계층을 표현하는데, SYS_CONNECT_BY_PATH() 함수를 이용하여 다음 예시와 같이 출력하시오(107행).

예 King-Kochhar-Greenberg-Faviet

HINT SYS_CONNECT_BY_PATH(column, char) 함수는 계층형 쿼리에서만 유효하며, column의 절대 경로를 char로 지정한 문자로 분리하여 반환한다.

풀이

	NAME_PATH
1	King
2	King-Kochhar
3	King-Kochhar-Greenberg
4	King-Kochhar-Greenberg-Faviet
	... (생략) ...
105	King-Zlotkey-Johnson
106	King-Hartstein
107	King-Hartstein-Fay

B 데이터 모델링

0 개체, 관계, 속성의 이해

ER 다이어그램Entity Relationship Diagram이란 개체와 개체 간의 관계를 알기 쉽게 나타낸 데이터 표현의 공통된 규칙으로, 약속된 도형으로 표현한다.

- **개체**: 현실 세계에서 식별 가능한 대상을 의미한다. 관리하거나 표현해야 할 대상을 명사형으로 정의하며, 서식이나 업무 문서, 정보 시스템 분석 등을 통해 도출한다.
- **관계**: 여러 개체 간의 연관성을 나타내는 개념으로, 특정 업무에서 적용되는 규칙을 표현한다. 예를 들어 ❶, ❷ 관계를 ER 다이어그램으로 표현하면 다음과 같다.

❶ 사원과 부서의 관계를 보면 사원은 반드시 하나 이상의 부서에 소속되어야 하고,
❷ 모든 부서는 한 명 이상의 사원을 배정받거나 그렇지 않을 수도 있다.

- **속성**: 개체를 설명하는 더 이상 분해할 수 없는 최소 단위의 데이터를 말한다. 예를 들어 사람을 개체로 보았을 때 '주민번호'는 개체를 식별하는 식별 속성, 직업·나이·학위·전화번호 등은 개체의 특징을 나타내는 기술 속성이다.

샘플 문제 1

다음 단어를 보고 각각 개체인지 관계인지 속성인지 구별하시오.

입원일자, 환자, 주민번호, 입원하다, 처방받다, 의사, 진료하다, 간호사, 재직연수

풀이

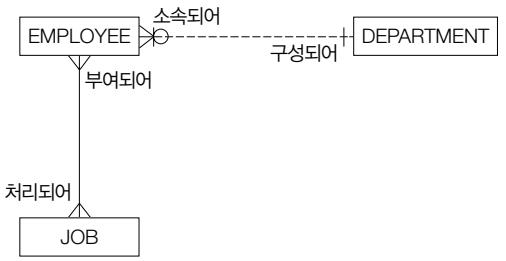
- 의사, 환자, 간호사는 식별 가능한 대상으로 볼 수 있기 때문에 개체이다.
- 입원일자, 주민번호, 재직연수는 식별가능 대상보다 식별 대상의 설명 요소로 볼 수 있기 때문에 속성이다.
- 입원하다, 처방받다, 진료하다 등은 의사가 환자를 보고 판단하는 것이므로 관계로 볼 수 있다.

샘플 문제 2

A절에서 사용한 hr 스키마의 인사 구조에 해당하는 다음 내용을 읽고, 사원(Employee), 부서(Department), 업무(Job) 간의 ER 다이어그램을 작성하시오.

❶ 모든 사원은 하나의 부서에 속할 수도 있고, 속하지 않을 수도 있다.
❷ 모든 부서는 한 명 이상의 사원들로 구성될 수도 있고, 그렇지 않을 수도 있다.
❸ 모든 사원에게는 하나 이상의 업무가 부여된다.
❹ 모든 업무는 한 명 이상의 사원들에 의해 처리된다.

풀이



1 푸드코트 ER 다이어그램

문제

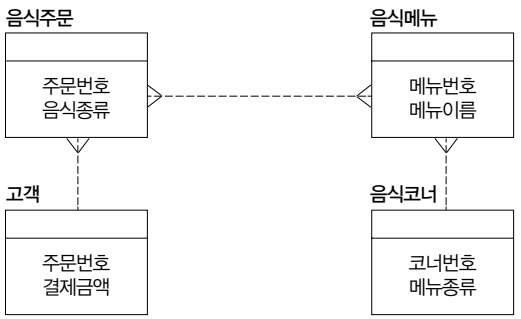
다음의 요구사항과 ER 다이어그램을 참고해 관계에 대한 문장을 만드시오.

나는 주말에 아이들과 A마트 푸드코트를 자주 이용한다. 여기에는 상하이, 한성면옥, 전주한식, 돈이돈가스 등의 음식코너가 있고, 각각 5개 이상의 메뉴를 갖고 있다. 우리는 각자 원하는 음식을 주문 기계로 주문한다. 큰아이는 두 가지 음식을 시키기도 한다. 주문을 한 후 주문번호가 적힌 쿠폰을 가지고 기다리다가 음식이 나오면 쿠폰과 교환하여 받아온다.

[문장 예시]

우리 가족(고객)은 각자 주문 기계를 통해 반드시 하나 이상의 음식메뉴를 선택할 수 있다.

[ER 다이어그램]



HINT 저장해야 할 데이터를 찾아본다. 음식메뉴, 음식코너, 고객이 있고 고객과 음식메뉴 사이에 음식주문의 관계가 성립한다. 그리고 각 개체 간의 카디널리티(cardinality) 분석을 위하여 다음과 같이 테이블을 정리해 본다.

- 모든 음식주문은 (하나 이상의) 음식메뉴를 (반드시 포함하고 있다).
- 모든 음식메뉴는 (하나 이상의) 음식주문에 (포함될 수 있다).
- 모든 음식주문은 (단 한 명의) 고객에 (반드시 속해야 한다).
- 모든 고객은 (하나 이상의) 음식을 (주문할 수 있다).
- 모든 음식메뉴는 (단 하나의) 음식코너에 (반드시 지정되어 있다).
- 모든 음식코너는 (하나 이상의) 음식메뉴를 (보유하고 있다).

2 병원 업무 관리 ER 다이어그램

마당병원은 데이터베이스를 구축해 의사, 환자, 간호사가 정보를 쉽게 검색, 관리할 수 있다.

문제

다음 요구사항을 읽고 개체, 속성, 관계를 식별한 후 〈병원 업무 관리〉 ER 다이어그램을 작성하시오.

1. 의사 한 명은 여러 명의 환자를 맡고, 환자 한 명은 한 명의 의사에게 진료를 받는다. 또한 의사는 환자에 대한 모든 정보를 검색할 수 있다.
2. 환자는 진료내용 등의 정보를 검색할 수 있으며 담당의사에 대해서도 조회할 수 있다.
3. 간호사는 병원 진료 차트 정보를 검색할 수 있으며 의사와 더불어 환자를 관리한다. 간호사 업무는 여러 환자들의 진료접수, 환자관리, 차트관리 등을 수행하는 것이다.
4. 모든 의사는 의사ID를 부여하여 식별하며 담당진료과목, 성명, 성별, 연락처, 이메일, 직급을 가진다.
5. 모든 환자는 환자ID를 부여하여 식별하며 담당의사, 담당간호사, 환자성명, 주민번호, 성별, 주소, 연락처, 이메일, 직업을 가진다.
6. 모든 간호사는 간호사ID를 부여하여 식별하며 담당업무, 성명, 성별, 전화번호, 이메일, 직급을 가진다.
7. 진료ID는 날짜+진료순번으로 부여하고 진료한 의사ID, 환자ID, 진료내용, 진료날짜를 포함한다.
8. 진료를 받은 환자에 대한 내용은 차트에 기록하고, 차트는 차트번호를 부여하여 식별하며 담당의사ID, 간호사ID와 환자ID, 진료ID 그리고, 의사 소견을 차트 내용에 기록한다.

3 학사 관리 ER 다이어그램

한빛대학은 학사 관리 시스템을 구축하여 학사 정보를 체계적으로 관리한다.

문제

다음 요구사항을 읽고 개체, 속성, 관계를 식별한 후 〈학사 관리〉 ER 다이어그램을 작성하시오.

❶ 학생은 담당교수와 소속학과가 있어야 하고, 수강신청을 하여 강좌를 수강해야 한다. 모든 학생은 학생번호를 부여하여 식별하며 학생이름, 학생주민번호, 학생주소, 학생전화번호, 학생이메일, 학생사진, 소속학과번호 정보를 가진다.

❷ 교수는 소속학과가 있어야 하고, 학생을 담당할 수도 있어야 하며, 강좌에 대한 강의를 해야 한다. 모든 교수는 교수번호를 부여하여 식별하며 교수이름, 교수주민번호, 교수주소, 교수전화번호, 교수이메일, 교수사진, 소속학과번호 정보를 가진다. 교수와 학생은 학생번호와 교수번호를 이용한 관계를 갖고, 현재 학년/학기 정보를 포함한다.

❸ 학과는 교수와 학생들로 이루어졌으며, 해당 학과에 대한 강좌가 있어야 한다. 그리고 학과번호를 부여하여 식별하며 학과명, 학과전화번호 정보를 가진다.

❹ 강좌는 강의하는 교수가 있어야 하지만, 간혹 강의가 진행되지 않는 강좌들도 있을 수 있다. 그리고 학생이 수강을 해야 하고, 강좌를 진행할 학과가 있어야 한다. 또한 강좌는 강좌번호와 담당 교수번호를 부여하여 식별하며 강좌명, 취득학점, 강의시간, 강의실 정보를 가진다.

❺ 수강(수강내역)은 학생번호, 강좌번호, 교수번호를 부여하여 식별하며 출석점수, 중간고사점수, 기말고사점수, 기타점수, 총점, 평점 정보를 가진다.

4 은행 업무 관리 ER 다이어그램

한빛은행은 지점별로 기본적인 은행 업무를 수행한다. 고객은 통장을 가지고 입출금이나 카드 서비스 등을 이용하고, 직원은 고객 업무를 도와 은행 내부 업무를 수행한다. 한빛은행은 고객과 내부 직원의 원활한 업무를 위해 데이터베이스를 구축하고자 한다.

문제 1

다음 두 개체에 대한 설명을 읽고 주 식별자를 선정하시오. 그리고 두 개체 사이의 관계를 표시하시오.

❶ 예금계좌는 고객의 통장을 관리하기 위한 개체이다.
❷ 예금계좌거래내역은 각 예금계좌에 대한 입출금 내역을 관리하기 위한 개체이다.
❸ 하나의 입금이나 출금 내역은 반드시 거래건별로 날짜와 함께 계좌에 남긴다.

예금계좌

예금계좌ID
주민번호
예금계좌종류
예금잔고
카드신청여부
예금개설일
예금자이름
전화번호
이메일

예금계좌거래내역

예금계좌ID
입출금날짜
거래번호
예금구분
예금내용
거래금액
예금잔고

문제 2

다음 요구사항을 읽고 개체, 속성, 관계를 식별한 후 〈은행 업무 관리〉 ER 다이어그램을 작성하시오.

❶ 한빛은행은 예금 및 카드 서비스를 고객에게 제공한다.
❷ 고객은 고객주민번호를 부여하여 식별하며 고객명, 고객주소, 고객생년월일, 고객이메일, 고객전화번호, 고객직업 등의 정보를 가진다.
❸ 카드(업무)는 고유의 카드ID(번호)를 부여하여 식별하며 카드신청일자, 카드한도금액, 카드결제일자, 카드종류, 고객주민번호, 예금계좌ID 정보를 가진다.
❹ 예금계좌는 예금계좌ID를 부여하여 식별하며 예금계좌종류, 예금잔고, 카드신청여부, 예금개설일자, 예금자이름, 전화번호, 이메일, 고객주민번호 정보를 포함한다.
❺ 예금계좌거래내역은 예금계좌ID, 입출금날짜, 거래번호를 부여하여 식별하며 예금구분, 예금내용, 거래금액, 예금잔고 정보를 가진다.

5 캠핑카 대여 관리 ER 다이어그램

마당 캠핑카 회사에서는 캠핑카 중계 사이트를 열어 여러 캠핑카 대여 회사가 캠핑카를 등록하고 고객들이 대여할 수 있도록 하려고 한다. 또한 캠핑카의 특수성을 고려하여 캠핑카 정비업체와도 연계할 계획이다. 이 사이트에서 구축할 데이터베이스는 캠핑카 대여 회사, 캠핑카 정비업소, 고객을 연결해 주는 역할을 할 것이다.

문제

〈마당 캠핑카 중계 사이트〉에 등록될 정보에 대한 설명을 읽고 개체, 속성, 관계를 식별한 후 〈캠핑카 대여 관리〉 ER 다이어그램을 작성하시오.

❶ 캠핑카 대여 회사는 캠핑카대여회사ID를 부여받고 회사명, 주소, 전화번호, 담당자이름, 담당자이메일 정보를 등록한다.

❷ 캠핑카는 캠핑카등록ID를 부여받고 캠핑카이름, 캠핑카차량번호, 캠핑카승차인원수, 캠핑카이미지, 캠핑카상세정보, 캠핑카대여비용 정보를 포함한다. 또한 어느 캠핑카 대여 회사의 소유인지 알기 위해 캠핑카대여회사ID와 캠핑카등록일자를 등록한다.

❸ 고객은 고객이 보유한 운전면허증번호를 등록하고 고객명, 고객주소, 고객전화번호, 고객이메일, 이전에 캠핑카 사용날짜 및 캠핑카 종류 정보를 등록한다.

❹ 고객은 원하는 캠핑카를 대여할 수 있다. 캠핑카 대여를 하게 되면 고유 대여번호를 부여받고 캠핑카등록ID, 운전면허증번호, 캠핑카대여회사ID, 대여시작일, 대여기간, 청구요금, 납입기한, 기타청구내역, 기타청구요금 정보를 등록한다.

❺ 캠핑카 정비소는 캠핑카정비소ID를 부여받고 정비소명, 정비소주소, 정비소전화번호, 담당자 이름 및 이메일 정보를 등록한다.

❻ 고객이 특정 캠핑카 정비소에 정비를 의뢰하면 정비정보를 남긴다. 정비정보는 고유 정비번호를 부여받고 캠핑카등록ID, 캠핑카정비소ID, 캠핑카대여회사ID, 고객운전면허증번호, 정비내역, 수리날짜, 수리비용, 납입기한, 기타정비내역 정보를 등록한다. 캠핑카 대여 회사는 캠핑카대여회사ID를 부여받고 회사명, 주소, 전화번호, 담당자이름, 담당자이메일 정보를 등록한다.

6 정규화 수행 연습: 도서 주문

정규화normalization는 관계형 데이터베이스의 개념적 데이터 모델링의 한 요소이다. 단계별 규칙을 적용하여 각 개체가 갖는 속성의 위치를 적합하게 배정하는 작업을 말한다. 정규화 작업을 통해 잘 만들어진 ER 모델은 물리적 설계로 옮겨 관계형 데이터베이스로 구축할 수 있다.

여기서 잠깐! | 정규화

- 정의: 이상 발생 요인인 중복성을 최소화하기 위해 일종의 제약조건을 체계화하여 이를 바탕으로 릴레이션에 이상현상이 없어질 때까지 분해해 나가는 과정이다.
- 목적: 데이터 중복을 최소화하기 위해 수행한다(중복은 데이터의 일관성 문제를 일으키고, 불필요한 트랜잭션을 유발할 수 있음).
- 종류

제1정규형: 원자값이 아닌 속성 제거	• 모든 속성값이 더 이상 쪼갤 수 없는 원자값을 가지는 상태 • 여전히 삽입, 삭제, 변경 작업에 이상현상이 발생할 가능성 높음
제2정규형: 부분 함수 종속 제거	• 제1정규화 이후 기본키가 아닌 속성이 기본키에 완전 함수 종속되는 경우 • 이행적 함수 종속에 따른 데이터 중복 문제는 여전히 남아 있음
제3정규형: 이행적 함수 종속 제거	• 제2정규화 이후 기본키 외에 속성 간에 종속성이 존재하지 않는 경우 • 대부분 데이터베이스 설계의 목표 단계임

- 효과
  - 필요한 조인을 피하여 입출력(I/O) 성능이 개선된다.
  - 데이터 중복을 최소화하여 공간 소비량을 절감하는 효과를 얻을 수 있다.

문제

다음 도서 주문 테이블을 보고 정규화를 수행하시오(제1정규형, 제2정규형, 제3정규형의 단계로 수행한다).

주문번호	주문날짜	회원 이름	회원 번호	회원 주소	전화번호	도서SNO	도서명	수량	단가
2025021 3123	2025 0213	Lee	L-123	서울 강남구	010- 3664-1234	97889123	NoSQL의 진수를 만나다	1	25000
2025050 9238	2025 0509	Park	P-234	서울 성동구	010- 5301-3456	97889234 97889235	데이터베이스 개론과 실습 Java for Beginner	1 2	27000 18000
2025072 1376	2025 0721	Kim	K-345	경기도 용인	010- 7341-2345	97889345	정보보안 개론과 실습	5	24000
2025072 9401	2025 0729	Nam	N-456	강원도 속초	010- 9279-3456	97889234	데이터베이스 개론과 실습	1	27000
2025100 5456	2025 1005	Park	P-234	서울 성동구	010- 5301-3456	97889567	오라클 프로그래밍	4	27000

C 데이터베이스 프로젝트

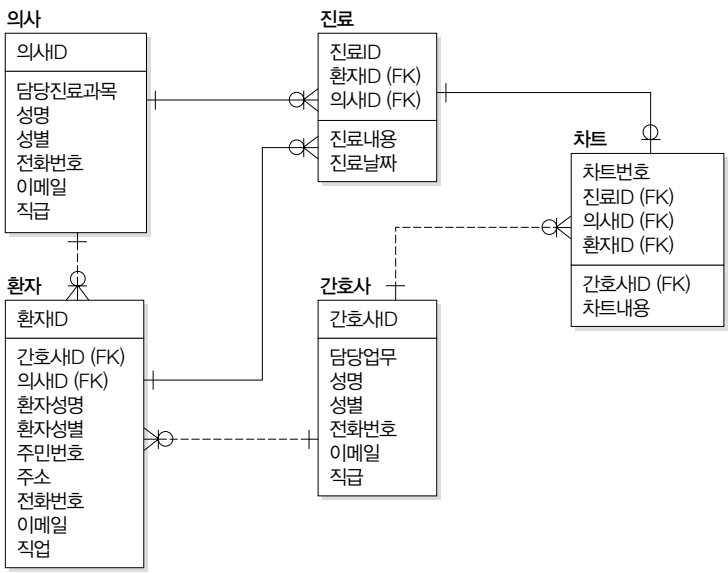
B절에서 만든 관계 데이터 모델을 실제 데이터베이스로 구축하는 과정을 실습해 보자.

1 병원 업무 관리 프로젝트

여기에서는 B.2절에서 만든 〈병원 업무 관리〉 ER 다이어그램을 이용하여 병원 업무 관리 데이터베이스를 구축해 본다. ER 다이어그램을 이용하여 데이터베이스의 테이블을 생성한 후 레코드를 삽입, 갱신, 삭제하는 작업을 하고 해당 계정schema을 생성하여 관련 테이블의 오브젝트를 생성하여 저장하는 작업을 한다. 그리고 별도의 저장영역인 테이블스페이스tablespace를 생성하여 개별 저장 관리하는 실습도 진행한다.

과제 1: ER 다이어그램 구축 및 계정(schema) 생성

B.2절에서 만든 〈병원 업무 관리〉 ER 다이어그램을 이용하여 테이블을 설계한다. 계정은 U_HOSPITAL로 하기로 한다.



과제 2: 테이블 구축

[과제 1]에서 구성한 ER 다이어그램을 근거로 각각의 테이블 구조 및 관계, 제약조건 등을 설정한다.

❶ 테이블명: Doctors(의사)

개체	Doctors(의사)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	의사ID	doc_id	INTEGER	10	Not Null / Unique
외래키					
속성	담당진료과목	major_treat	VARCHAR	25	Not Null
	성명	doc_name	VARCHAR	20	Not Null
	성별	doc_gen	CHAR	1	Not Null
	전화번호	doc_phone	VARCHAR	15	
	이메일	doc_email	VARCHAR	50	Unique
	직급	doc_position	VARCHAR	20	Not Null

테이블 생성 구문

```
CREATE TABLE Doctors (
   doc_id NUMBER(10) NOT NULL,
   major_treat VARCHAR2(25) NOT NULL,
   doc_name VARCHAR2(20) NOT NULL,
   doc_gen CHAR(1) NOT NULL,
   doc_phone VARCHAR2(15) NULL,
   doc_email VARCHAR2(50) UNIQUE,
   doc_position VARCHAR2(20) NOT NULL
);

ALTER TABLE Doctors
   ADD CONSTRAINT doc_id_pk PRIMARY KEY (doc_id);
```

❷ 테이블명: Nurses(간호사)

개체	Nurses(간호사)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	간호사ID	nur_id	INTEGER	10	Not Null / Unique
외래키					
속성	담당업무	major_job	VARCHAR	25	Not Null
	성명	nur_name	VARCHAR	20	Not Null
	성별	nur_gen	CHAR	1	Not Null
	전화번호	nur_phone	VARCHAR	15	
	이메일	nur_email	VARCHAR	50	Unique
	직급	nur_position	VARCHAR	20	Not Null

테이블 생성 구문

```
CREATE TABLE Nurses (
   nur_id NUMBER(10) NOT NULL,
   major_job VARCHAR2(25) NOT NULL,
   nur_name VARCHAR2(20) NOT NULL,
   nur_gen CHAR(1) NOT NULL,
   nur_phone VARCHAR2(15) NULL,
   nur_email VARCHAR2(50) UNIQUE,
   nur_position VARCHAR2(20) NOT NULL
);

ALTER TABLE Nurses
   ADD CONSTRAINT nur_id_pk PRIMARY KEY (nur_id);
```

❸ 테이블명: Patients(환자)

개체	Patients(환자)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	환자ID	pat_id	INTEGER	10	Not Null / Unique
외래키	간호사ID	nur_id	INTEGER	10	Not Null
	의사ID	doc_id	INTEGER	10	Not Null
속성	환자성명	pat_name	VARCHAR	20	Not Null
	환자성별	pat_gen	CHAR	1	Not Null
	주민번호	pat_jumin	VARCHAR	14	Not Null
	주소	pat_addr	VARCHAR	100	Not Null
	전화번호	pat_phone	VARCHAR	15	
	이메일	pat_email	VARCHAR	50	Unique
	직업	pat_job	VARCHAR	20	Not Null

테이블 생성 구문

```
CREATE TABLE Patients (
   pat_id NUMBER(10) NOT NULL,
   nur_id NUMBER(10) NOT NULL,
   doc_id NUMBER(10) NOT NULL,
   pat_name VARCHAR2(20) NOT NULL,
   pat_gen CHAR(1) NOT NULL,
   pat_jumin VARCHAR2(14) NOT NULL,
   pat_addr VARCHAR2(100) NOT NULL,
   pat_phone VARCHAR2(15) NULL,
   pat_email VARCHAR2(50) UNIQUE,
   pat_job VARCHAR2(20) NOT NULL
);

ALTER TABLE Patients
   ADD CONSTRAINT pat_id_pk PRIMARY KEY (pat_id);

ALTER TABLE Patients
   ADD (CONSTRAINT R_2 FOREIGN KEY (doc_id) REFERENCES Doctors (doc_id));

ALTER TABLE Patients
   ADD (CONSTRAINT R_3 FOREIGN KEY (nur_id) REFERENCES Nurses (nur_id));
```

❹ 테이블명: Treatments(진료)

개체	Treatments(진료)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	진료ID	treat_id	INTEGER	15	Not Null / Unique
	환자ID	pat_id	INTEGER	10	Not Null / Unique
	의사ID	doc_id	INTEGER	10	Not Null / Unique
외래키	환자ID	pat_id	INTEGER	10	
	의사ID	doc_id	INTEGER	10	
속성	진료내용	treat_contents	VARCHAR	1000	Not Null
	진료날짜	treat_date	DATE		Not Null

테이블 생성 구문

```
CREATE TABLE Treatments (
   treat_id NUMBER(15) NOT NULL,
   pat_id NUMBER(10) NOT NULL,
   doc_id NUMBER(10) NOT NULL,
   treat_contents VARCHAR2(1000) NOT NULL,
   treat_date DATE NOT NULL
);

ALTER TABLE Treatments
   ADD CONSTRAINT treat_pat_doc_id_pk PRIMARY KEY (treat_id,pat_id,doc_id);

ALTER TABLE Treatments
   ADD (CONSTRAINT R_5 FOREIGN KEY (pat_id) REFERENCES Patients (pat_id));

ALTER TABLE Treatments
   ADD (CONSTRAINT R_6 FOREIGN KEY (doc_id) REFERENCES Doctors (doc_id));
```

❺ 테이블명: Charts(차트)

개체	Charts(차트)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	차트번호	chart_id	VARCHAR	20	Not Null / Unique
	진료ID	treat_id	INTEGER	15	Not Null / Unique
	의사ID	doc_id	INTEGER	10	Not Null / Unique
	환자ID	pat_id	INTEGER	10	Not Null / Unique
외래키	진료ID	treat_id	INTEGER	15	
	의사ID	doc_id	INTEGER	10	
	환자ID	pat_id	INTEGER	10	
	간호사ID	nur_id	INTEGER	10	Not Null
속성	차트내용	chart_contents	VARCHAR	1000	Not Null

테이블 생성 구문

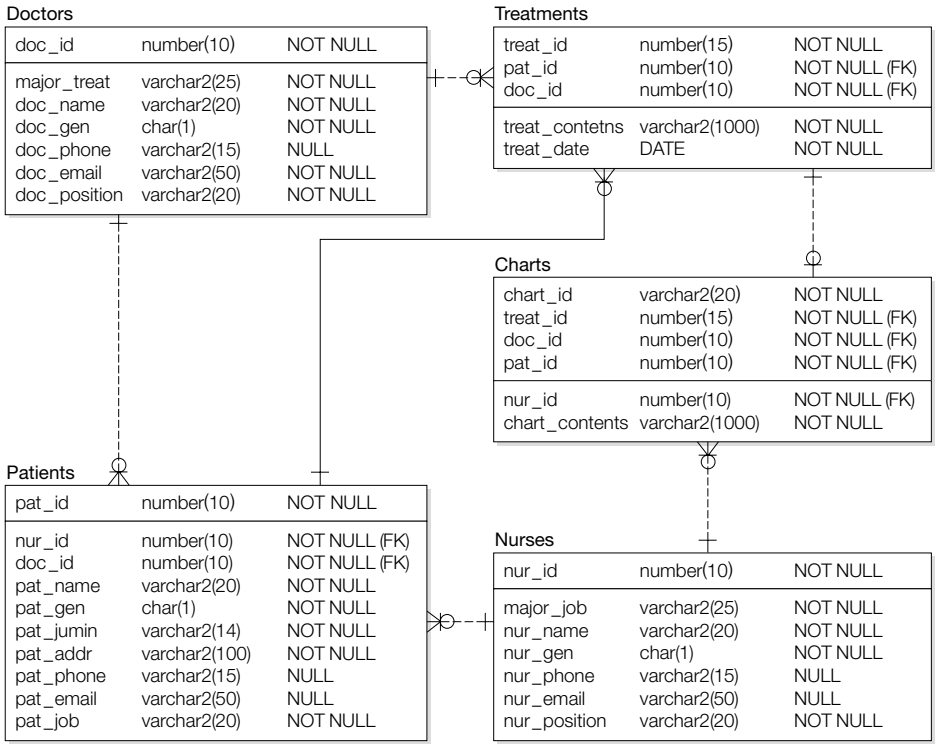
```
CREATE TABLE Charts (
   chart_id VARCHAR2(20) NOT NULL,
   treat_id NUMBER(15) NOT NULL,
   doc_id NUMBER(10) NOT NULL,
   pat_id NUMBER(10) NOT NULL,
   nur_id NUMBER(10) NOT NULL,
   chart_contents VARCHAR2(1000) NOT NULL
);

ALTER TABLE Charts
   ADD CONSTRAINT chart_treat_doc_pat_id_pk PRIMARY KEY (chart_id,treat_id,doc_
   id,pat_id);

ALTER TABLE Charts
   ADD (CONSTRAINT R_4 FOREIGN KEY (nur_id) REFERENCES Nurses (nur_id));

ALTER TABLE Charts
   ADD (CONSTRAINT R_7 FOREIGN KEY (treat_id, pat_id, doc_id) REFERENCES
   Treatments (treat_id, pat_id, doc_id));
```

실제 데이터베이스에 접속하여 각각의 테이블을 해당 계정schema에 생성한다.



과제 3: 데이터 입력(INSERT)

생성된 다섯 개의 테이블에 다음과 같은 정보를 입력한다. 각 테이블 컬럼의 데이터 타입 및 사이즈를 확인한 후 정확히 입력하도록 한다.

예 의사 테이블에 데이터를 입력하는 경우

```
INSERT INTO Doctors
        VALUES (070576, '피부과', '홍길동', 'F', '010-333-7263', 'lja@hanbh.
                com', '전문의');

SELECT * FROM Doctors;
```

	DOC_ ID	MAJOR_ TREAT	DOC_ NAME	DOC_ GEN	DOC_PHONE	DOC_EMAIL	DOC_ POSITION
1	70576	피부과	홍길동	F	010-333-7263	lja@hanbh.com	전문의

	Doctors(의사)						
	doc_id	major_treat	doc_name	doc_gen	doc_phone	doc_email	doc_position
1	980312	소아과	이태정	M	010-333-1340	ltj@hanbh.com	과장

2	000601	내과	안성기	M	010-222-0987	ask@hanbh.com	과장
3	001208	외과	김민종	M	010-333-8743	kmj@han.com	과장
4	020403	피부과	이태서	M	010-777-3764	lts@hanbh.com	과장
5	050900	소아과	김연아	F	010-555-3746	kya@habhn.com	전문의
6	050101	내과	차태현	M	010-222-7643	cth@hanbh.com	전문의
7	062019	소아과	전지현	F	010-999-1265	jh@hanbh.com	전문의
8	070576	피부과	홍길동	M	010-333-7263	hgd@hanbh.com	전문의
9	080543	방산선과	유재석	M	010-222-1263	yjs@hanbh.com	과장
10	091001	외과	김병만	M	010-555-3542	kbm@hanbh.com	전문의

	Nurses(간호사)						
	nur_id	major_job	nur_name	nur_gen	nur_phone	nur_email	nur_position
1	050302	소아과	김은영	F	010-555-8751	key@hanbh.com	수간호사
2	050021	내과	윤성애	F	010-333-8745	ysa@hanbh.com	수간호사
3	040089	피부과	신지원	M	010-666-7646	sjw@hanbh.com	주임
4	070605	방사선과	유정화	F	010-333-4588	yjh@hanbh.com	주임
5	070804	내과	라하나	F	010-222-1340	nhn@hanbh.com	주임
6	071018	소아과	김화경	F	010-888-4116	khk@hanbh.com	간호사
7	100356	소아과	이선용	M	010-777-1234	lsy@hanbh.com	간호사
8	104145	외과	김현	M	010-999-8520	kh@hanbh.com	간호사
9	120309	피부과	박성완	M	010-777-4996	psw@hanbh.com	간호사
10	130211	외과	이서연	F	010-222-3214	lsy2@hanbh.com	간호사

	Patients(환자)									
	pat_id	nur_id	doc_id	pat _ name	pat _gen	pat_ jumin	pat _ addr	pat_phone	pat_email	pat_ job
1	2345	050302	980312	안상건	M	232345	서울	010-555-7845	ask@ab.com	회사원
2	3545	040089	020403	김성룡	M	543545	서울	010-333-7812	ksr@bb.com	자영업
3	3424	070605	080543	이종진	M	433424	부산	010-888-4859	ljj@ab.com	회사원
4	7675	100356	050900	최광석	M	677675	당진	010-222-4847	cks@cc.com	회사원
5	4533	070804	000601	정한경	M	744533	강릉	010-777-9630	jhk@ab.com	교수
6	5546	120309	070576	유원현	M	765546	대구	010-777-0214	ywh@cc.com	자영업
7	4543	070804	050101	최재정	M	454543	부산	010-555-4187	cjj@bb.com	회사원
8	9768	130211	091001	이진희	F	19768	서울	010-888-3675	ljh@ab.com	교수
9	4234	130211	091001	오나미	F	234234	속초	010-999-6541	onm@cc.com	학생
10	7643	071018	062019	송성묵	M	987643	서울	010-222-5874	ssm@bb.com	학생

	Treatments(진료)				
	treat_id	pat_id	doc_id	treat_contents	treat_date
1	130516023	2345	980312	감기, 몸살	2023-05-16
2	130628100	3545	020403	피부 트러블 치료	2023-06-28
3	131205056	3424	080543	목 디스크로 MRI 촬영	2023-12-05
4	131218024	7675	050900	중이염	2023-12-18
5	131224012	4533	000601	장염	2023-12-24
6	140103001	5546	070576	여드름 치료	2024-01-03
7	140109026	4543	050101	위염	2024-01-09
8	140226102	9768	091001	화상 치료	2024-02-26
9	140303003	4234	091001	교통사고 외상 치료	2024-03-03
10	140308087	7643	062019	장염	2024-03-08

	Charts(차트)					
	chart_id	treat_id	doc_id	pat_id	nur_id	chart_contents
1	p_130516023	130516023	980312	2345	050302	감기 주사 및 약 처방
2	d_130628100	130628100	020403	3545	040089	피부 감염 방지 주사
3	r_131205056	131205056	080543	3424	070605	주사 처방
4	p_131218024	131218024	050900	7675	100356	귓속 청소 및 약 처방
5	i_131224012	131224012	000601	4533	070804	장염 입원 치료
6	d_140103001	140103001	070576	5546	120309	여드름 치료약 처방
7	i_140109026	140109026	050101	4543	070804	위내시경
8	s_140226102	140226102	091001	9768	130211	화상 크림약 처방
9	s_140303003	140303003	091001	4234	130211	입원 치료
10	p_140308087	140308087	062019	7643	071018	장염 입원 치료

과제 4: 데이터 변경(UPDATE) 및 삭제(DELETE)

홍길동 의사가 맡고 있던 담당진료과목이 피부과에서 소아과로 변경되어 내일부터 진료를 시작할 예정이다. 이 정보에 대한 테이블 정보를 변경하시오.

```
UPDATE Doctors
SET    major_treat = '소아과'
WHERE  doc_name = '홍길동';

SELECT *
FROM   Doctors;
```

	DOC_ID	MAJOR_TREAT	DOC_NAME	DOC_GEN	DOC_PHONE	DOC_EMAIL	DOC_POSITION
1	70576	소아과	홍길동	F	016-333-7263	lja@hanbh.com	전문의

김은영 간호사는 대학원 진학으로 오늘까지만 근무하고 퇴사하게 되었다. 이 정보에 대한 테이블 정보를 변경하시오.

```
DELETE
FROM    Nurses
WHERE   nur_name = '김은영';
```

과제 5: 데이터 검색(QUERY)

담당진료과목이 '소아과'인 의사에 대한 정보를 출력하시오.

```
SELECT  *
FROM    Doctors
WHERE   major_treat = '소아과';
```

홍길동 의사에게 진료를 받은 환자에 대한 모든 정보를 출력하시오.

```
SELECT  p.*, d.doc_name
FROM    Doctors d JOIN Patients p ON (d.doc_id = p.doc_id)
WHERE   d.doc_name = '홍길동';
```

진료날짜가 2023년 12월인 환자에 대한 모든 정보를 오름차순 정렬하여 출력하시오.

```
SELECT  p.*, t.treat_date
FROM    Treatments t JOIN Patients p ON (t.pat_id = p.pat_id)
WHERE   t.treat_date between '2023-12-01' and '2023-12-31'
ORDER BY t.treat_date ASC;
```

간호사ID가 5로 시작하는 모든 간호사 정보를 출력하시오.

```
SELECT  *
FROM    Nurses
WHERE   ROUND(nur_id / 10000, 0)=5;
```

2 학사 관리 프로젝트

여기에서는 B.3절에서 만든 〈학사 관리〉 ER 다이어그램을 이용하여 학사 업무 관리 데이터베이스를 구축해 본다. ER 다이어그램을 이용하여 데이터베이스의 테이블을 생성한 후 레코드를 삽입, 갱신, 삭제하는 작업을 하고 해당 계정을 생성하여 관련 테이블의 오브젝트를 생성하여 저장하는 작업을 한다. 그리고 별도의 저장영역인 테이블스페이스를 생성하여 개별 저장 관리하는 실습도 진행한다. 계정 이름은 U_HAKSA로 한다.

과제 1: ER 다이어그램 구축 및 계정(Schema) 생성

B.3절에서 만든 〈학사 관리〉 ER 다이어그램으로 테이블을 설계한다.

과제 2: 테이블 구축

[과제 1]에서 구성한 ER 다이어그램을 근거로 각각의 테이블 구조 및 관계, 제약조건 등을 설정한다.

과제 3: 데이터 입력(INSERT)

생성된 다섯 개의 테이블에 다음과 같은 정보를 입력한다. 각 테이블 컬럼의 데이터 타입 및 사이즈를 확인한 후 정확히 입력하도록 한다.

예 INSERT INTO HS_STUDENTS

```
VALUES ('04154123', '145', '1234', '서창호', '750813-1234567', '서울시 성동구',
'010-4322-9876', 'sch@ab.com', null, '2-1');
```

과제 4: 데이터 검색(Query)

❶ 교수번호 145에게 수업을 받는 학생에 대한 정보를 출력하시오.

❷ 컴퓨터공학과에 재학중인 학생과 강의하는 교수에 대한 모든 정보를 출력하시오.

❸ 현재 2학년 2학기에 재학 중인 학생에 대한 정보를 오름차순 정렬하여 출력하시오.

❹ 학생ID가 04로 시작하는 모든 학생 정보를 출력하시오.

3 은행 업무 관리 프로젝트

여기에서는 B.4절에서 만든 〈은행 업무 관리〉 ER 다이어그램을 이용하여 은행 업무 관리 데이터베이스를 구축해본다. ER 다이어그램을 이용하여 데이터베이스의 테이블을 생성한 후 레코드를 삽입, 갱신, 삭제하는 작업을 하고 해당 계정을 생성하여 관련 테이블의 오브젝트를 생성하여 저장하는 작업을 한다. 그리고 별도의 저장영역인 테이블스페이스를 생성하여 개별 저장 관리하는 실습도 진행한다. 계정 이름은 U_BANK로 하기로 한다.

과제 1: ER 다이어그램 구축 및 계정(Schema) 생성

B.4절에서 만든 〈은행 업무 관리〉 ER 다이어그램으로 테이블을 설계한다.

과제 2: 테이블 구축

[과제 1]에서 구성한 ER 다이어그램을 근거로 각각의 테이블 구조 및 관계, 제약조건 등을 설정한다.

과제 3: 데이터 입력(INSERT)

생성된 네 개의 테이블에 다음과 같은 정보를 입력한다. 각 테이블 컬럼의 데이터 타입 및 사이즈를 확인한 후 정확히 입력하도록 한다.

예 INSERT INTO B_CUSTOMERS

```
VALUES ('850813-1234567', '서창호', '서울시 성동구', '85-08-13', 'sch@ab.com', '010-4322-9876', '주방장');
```

과제 4: 데이터 검색(Query)

❶ 카드를 신청한 모든 고객에 대한 정보를 출력하시오.

❷ 예금 잔고가 1,000,000 ~ 5,000,000인 고객에 대한 정보를 출력하시오.

❸ 카드 종류가 'VISA'인 고객의 예금 잔고를 출력하시오.

4 캠핑카 대여 관리 프로젝트

과제 1: ER 다이어그램 구축 및 계정(schema) 생성

B.5절에서 만든 〈캠핑카 대여 관리〉 ER 다이어그램을 이용하여 테이블을 설계한다. 계정 이름은 U_CRENT로 하기로 한다.

과제 2: 테이블 구축

[과제 1]에서 구성한 ER 다이어그램을 근거로 각각의 테이블 구조 및 관계, 제약조건 등을 설정한다.

과제 3: 데이터 입력(INSERT)

생성된 여섯 개의 테이블에 다음과 같은 정보를 입력한다. 각 테이블 컬럼의 데이터 타입 및 사이즈를 확인한 후 정확히 입력하도록 한다.

예 INSERT INTO C_CAR_RENT_COMP

```
VALUES ('33345', '마당캠핑카', '서울시 마포구', '010-5344-6231', '이태정', 'ltj@
cc.com');
```

과제 4: 데이터 검색(Query)

❶ 2023년 7월 ~ 8월까지 캠핑카를 대여한 모든 고객에 대한 정보를 출력하시오.

❷ 서울에 있는 캠핑카 정비소에 대한 모든 정보를 출력하시오.

❸ 캠핑카 승차 인원이 5인 이상인 캠핑카의 모든 정보를 출력하시오.