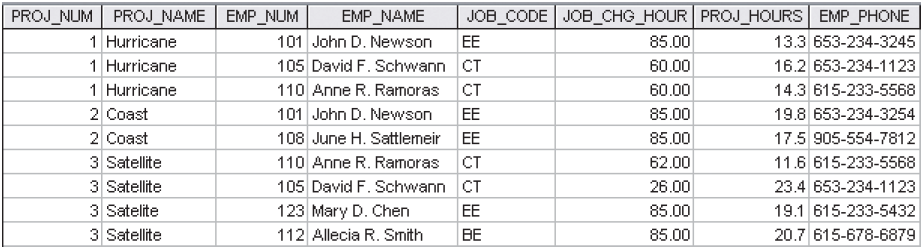
1. 아래 파일을 보고 질문에 답하시오. (10점)



A.1. 자료 중복 문제를 찾아 설명하시오. (5점)

프로젝트를 관리하는 테이블에 사번, 사원 이름, 사원 휴대폰 번호 등

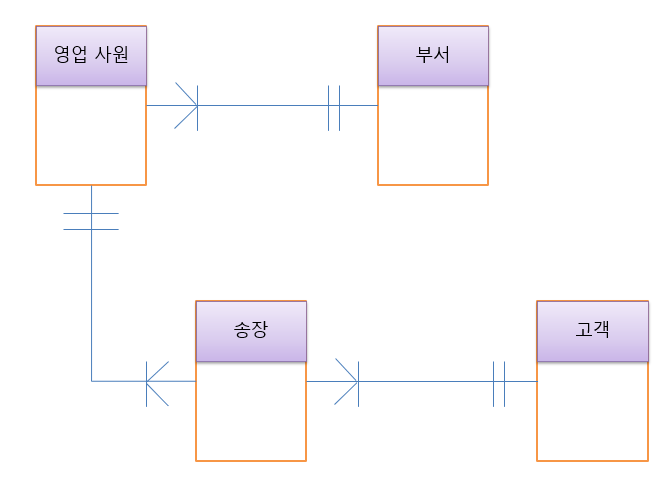
사원에 관한 정보들이 중복되서 출력되고 있다.

A.2. 자료 중복 문제를 해결할 수 있는 방안을 제시하시오. (5점)

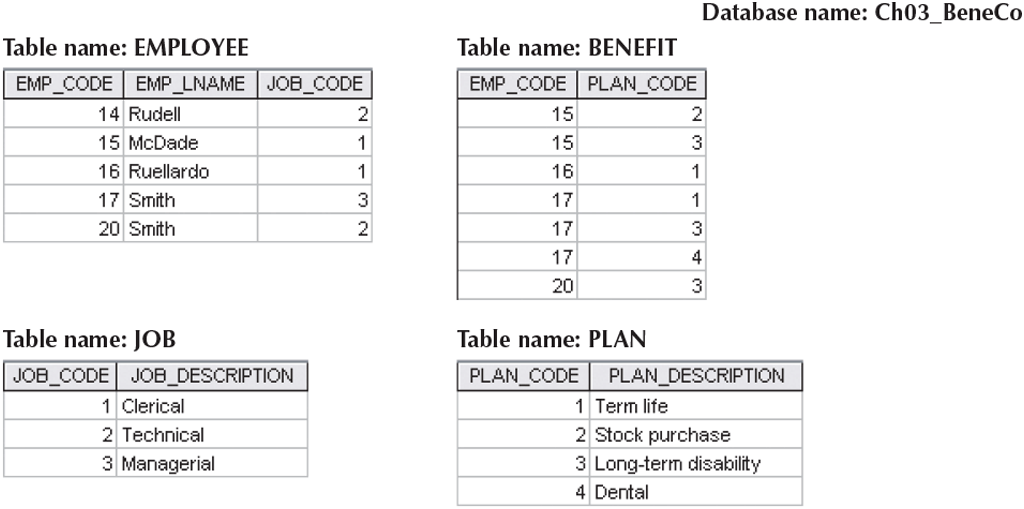
별도의 사원 테이블을 만들어 사원의 개인 정보들을 입력한 뒤,

foreign key를 통해 프로젝트 테이블에 연결한다.

1. 아래 사업규칙(business rules)을 충족시키는 실체관계도(Entity Relationship Diagram)를 그리시오. (10점)
2. 각 영업사원은 여러 송장을 작성한다.
3. 각 송장은 한 명의 영업사원에 의해 작성된다.
4. 각 영업사원은 한 부서에 배치된다.
5. 각 부서에는 여러 명의 영원사원들이 소속된다.
6. 각 고객은 여러 건의 송장을 발생시킨다.
7. 각 송장은 한 명의 고객을 위해 생성된다.



1. 아래 관계표(table)들로 구성된 DB에 대해 질문에 답하시오. (10점)



4개의 관계표는 아래의 관계를 반영하였다:

* 한 명의 직원(EMPLOYEE)은 단 하나의 직무코드(JOB\_CODE)를 갖지만, 여러 명의 직원이 같은 직무코드(JOB\_CODE)를 가질 수 있다.
* 한 명의 직원(EMPLOYEE)은 다수의 혜택프로그램(PLAN)에 가입할 수 있고, 하나의 혜택프로그램(PLAN)에는 다수의 직원(EMPLOYEE)이 가입할 수 있다.
* BENEFIT 관계표는 다:다 관계를 실체로 변환한 associative entity 이다.

C.1. 각 관계표에 대해 주 키(primary key)와 외래키(foreign key)를 찾아내시오. 외래키가 없으면 ‘없음(None)’으로 나타내시오. (4점)

|  |  |  |
| --- | --- | --- |
| **TABLE** | **PRIMARY KEY** | **FOREIGN KEY(S)** |
| EMPLOYEE | EMP\_CODE | JOB\_CODE |
| BENEFIT | 없음(None) | EMP\_CODE, PLAN\_CODE |
| JOB | JOB\_CODE | 없음(None) |
| PLAN | PLAN\_CODE | 없음(None) |

C.2. 각 관계표는 실체 무결성(entity integrity)을 보이고 있는가? ‘네/아니오’로 답하고 답을 선택한 이유를 설명하시오. (3점)

|  |  |  |
| --- | --- | --- |
| **TABLE** | **ENTITY INTEGRITY** | **EXPLANATION** |
| EMPLOYEE | Yes | PRIMARY KEY가 고유한 값을 가지고 있으며, Null값도 존재하지 않는다. |
| BENEFIT | No | 연결테이블이라서 PRIMARY KEY가 존재하지 않는다. |
| JOB | Yes | PRIMARY KEY가 고유한 값을 가지고 있으며, Null값도 존재하지 않는다. |
| PLAN | Yes | PRIMARY KEY가 고유한 값을 가지고 있으며, Null값도 존재하지 않는다. |

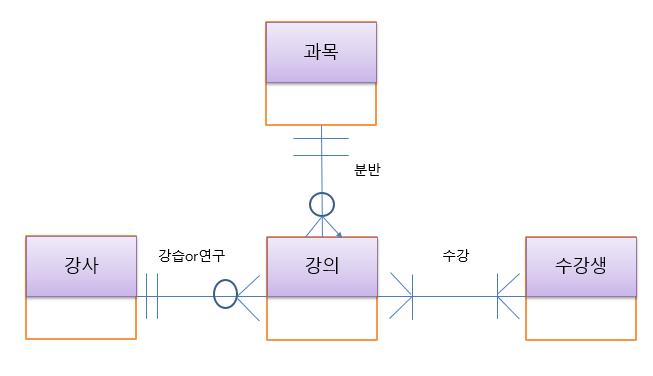
C.3. 각 관계표는 참조 무결성(referential integrity)을 보이고 있는가? ‘네/아니오’로 답하고 답을 선택한 이유를 설명하시오. 관계표가 외래키를 갖지 않으면 NA(Not Applicable)로 답하시오. (3점)

|  |  |  |
| --- | --- | --- |
| **TABLE** | **REFERENTIAL INTEGRITY** | **EXPLANATION** |
| EMPLOYEE | Yes | Foreign Key로 다른 테이블과 연결되어있다. |
| BENEFIT | Yes | Foreign Key로 다른 테이블과 연결되어있다. |
| JOB | No | 다른 테이블을 참조하지 않는다. |
| PLAN | No | 다른 테이블을 참조하지 않는다. |

1. HEG사는 교육/훈련프로그램에 관한 DB를 구축하기 위해 개념모형(conceptual model)을 만들려고 한다. HEG사의 관리자가 교육훈련 프로그램의 운영에 관해 다음과 같이 설명하였다.

HEG사는 12명의 강사를 보유하고 있으며 한 강의에는 30명까지의 수강생을 수용할 수 있다. HEG사는 고급기술 5과목을 개설하고 있으며, 각 과목은 대여섯 개의 분반으로 나누어 강의가 개설될 수 있다. 수강생이 10명 미만인 강의는 취소된다. 따라서, 강의가 개설되지 않는 과목도 있을 수 있다. 각 강의는 한 명의 강사가 담당한다. 각 강사는 2개의 강의까지 가르칠 수 있는데 강의하지 않고 연구만 하는 것도 가능하다. 각 수강생은 1년에 2과목까지 수강할 수 있다.

개념모형을 그리고, ‘강사’와 ‘강의’의 관계에서 connectivity, cardinality, and existence dependence를 설명하시오. (10 점)



Connectivity :

강사 -> 강의 : 1:M 관계

강의 -> 강사 : 1:1 관계

Cardinality :

강사(0,2) - 1년에 최대 2개의 강의를 하거나 쉴 수 있음

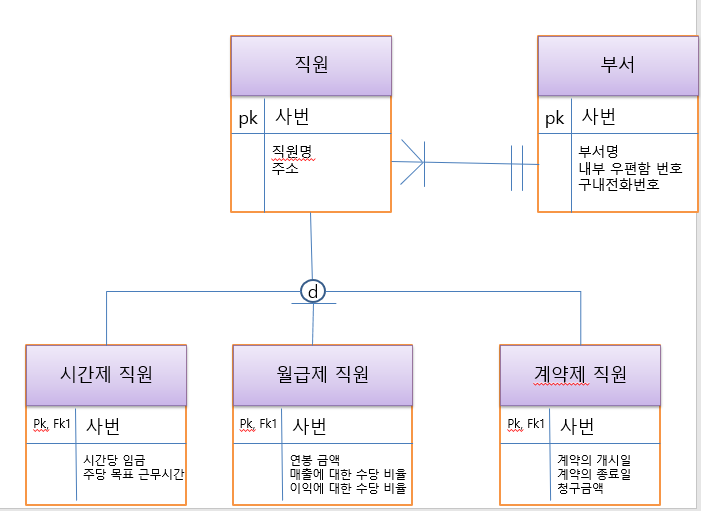
강의(1,1) – 강의는 한 명의 강사가 담당해야함.

existence dependence :

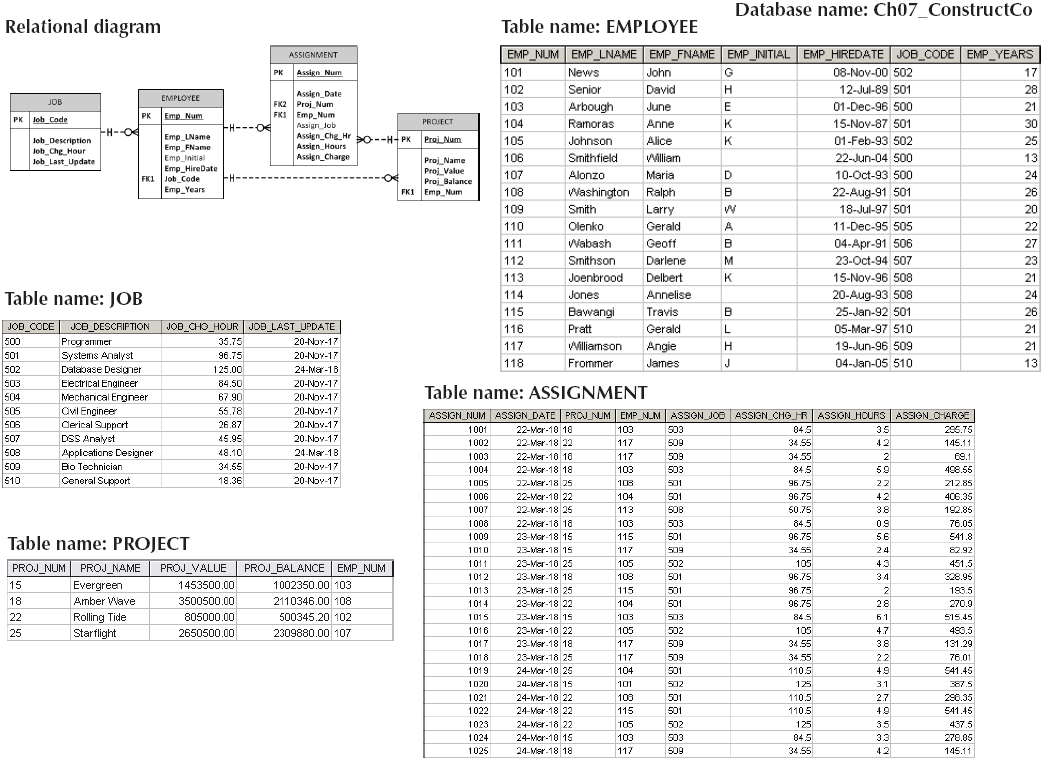
과목 – 강의(분반) 의 관계는 존재 종속(existence dependence)의 원칙을 따른다.

(과목이 없으면 강의가 존재할 수 없음)

1. 다음 시나리오를 읽고, 필요하면 specialization hierarchy를 포함하는 실체관계도(ERD)를 그리시오. Granite사는 직원과 소속 부서에 관한 정보를 유지한다. 각 부서에 대해서는 부서명, 내부 우편함 번호, 구내전화번호를 보관한다. 한 부서에는 다수의 직원이 소속될 수 있고, 한 직원은 한 부서에만 소속된다. 직원들은 월급제, 시간제, 계약제 직원으로 분류된다. 모든 직원은 사번을 부여받고, 직원명과 주소를 기록한다. 시간제 직원에 대해서는 시간당 임금과 주당 목표 근무시간이 저장된다. 예를 들면 어떤 직원은 주당 40시간을 근무하고 다른 직원은 32시간 또는 20시간을 근무하도록 한다. 일부 월급제 직원은 기본급에 수당을 추가하여 받는 영업사원이다. 모든 월급제 직원에 대해서는 연봉 금액을 기록한다. 영업사원에 대해서는 매출에 대한 수당 비율과 이익에 대한 수당 비율을 저장한다. 예를 들면 존은 기본 연봉 5만$과 매출액의 2%와 매출로부터의 이익의 5%를 수당으로 받는다. 계약제 직원에 대해서는 계약의 개시일과 종료일, 계약기간동안 청구금액 등을 기록한다. (10 점)



1. 다음의 Ch07\_ConstructCo DB는 프로젝트에 대한 모든 비용을 추적하는 컨설팅회사에 대한 자료를 저장한다. 비용은 각 직원이 각 프로젝트에 투입한 시간을 기반으로 산정된다. DB 구조는 아래와 같다.



ASSIGNMENT 관계표는 자료 이력의 정확성을 유지하기 위해 (ASSIGN\_CHG\_HR) 속성에 JOB\_CHG\_HOUR 값을 저장한다. JOB\_CHG\_HOUR 값은 시간이 지남에 따라 바뀐다. 사실 JOB\_CHG\_HOUR 값이 변하면 ASSIGNMENT 관계표에 반영될 것이다. 당연히 직원의 주 직무 배정도 바뀔 수 있는데 (ASSIGN\_JOB) 속성에 저장된다. 이들 속성은 자료 이력의 정확성을 유지하기 위해 필요하므로 자료 중복이 아니다. 이 사례에 대해 아래의 각 질문에 대해 SQL문을 작성하시오. (총 24 점)

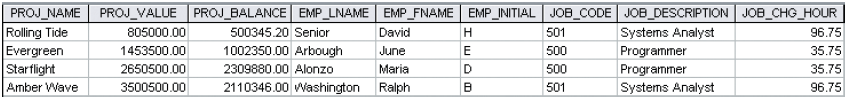
1. 성(last name)이 Smith로 시작하는 직원 모두의 사번, 성, 이름, 중간이름 첫자를 보여주시오. 예를 들면, Smith와 Smithfield 라는 직원은 검색 결과에 포함될 것이다. 검색결과를 사번 순으로 정렬하시오. (4 점)

SELECT EMP\_NUM, EMP\_LNAME, EMP\_FNAME, EMP\_INITIAL

FROM EMPLOYEE

WHERE EMP\_LNAME LIKE ‘Smith%’;

1. Using the EMPLOYEE, JOB, PROJECT 관계표를 사용하여 아래와 같이 project value 순으로 정렬한 검색 결과를 얻을 수 있도록 SQL문을 작성하시오. EMPLOYEE와 PROJECT 관계표를 join 할 때는 EMP\_NUM을 공통자료항목으로 사용하시오. (4 점)



SELECT PROJECT.PROJ\_NAME, PROJECT.PROJECT\_VALUE, PROJECT.PROJ\_BALANCE, EMPLOYEE.EMP\_LNAME, EMPLOYEE.EMP\_FNAME, EMPLOYEE.EMP\_INITIAL, JOB.JOB\_CODE, JOB.JOB\_DESCRIPTION, JOB.JOB\_CHG\_HOUR

FROM EMPLOYEE JOIN PROJECT ON EMPLOYEE.EMP\_NUM = PROJECT.EMP\_NUM JOIN JOB ON EMPLOYEE.JOB\_CODE = JOB.JOB\_CODE;

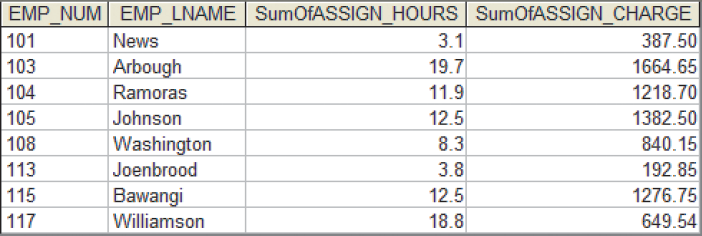
1. ASSIGNMENT 관계표에서 ASSIGN\_CHARGE 값을 확인하기 위한 SQL문을 작성하시오. 검색 결과로 assignment number, employee number, project number, 저장된 assignment charge (ASSIGN\_CHARGE)와 계산된 assignment charge (ASSIGN\_CHG\_HR 와 ASSIGN\_HOURS를 곱한 값)를 assignment number 순으로 정렬하여 보여주시오. (4 점)

SELECT ASSIGN\_NUM, EMP\_NUM, PROJ\_NUM, ASSIGN\_CHARGE, ASSIGN\_CHG\_HR \* ASSIGN\_HOURS AS NEW\_ ASSIGN\_CHARGE

FROM ASSIGNMENT

ORDER BY ASSIGN\_NUM;

1. ASSIGNMENT 관계표의 자료를 사용하여, 각 직원의 총 근무시간과 총 급여액을 사번 순으로 정렬하여 보여주는 SQL문을 작성하시오. SQL문을 실행한 결과는 다음과 같아야 한다. (4 점)

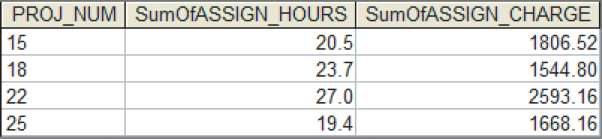


SELECT ASSIGNMENT.EMP\_NUM, EMPLOYEE.EMP\_LNAME, SUM(ASSIGN\_HOURS), SUM(ASSIGN\_CHARGE)

FROM ASSIGNMENT JOIN EMPLOYEE ON EMPLOYEE.EMP\_NUM = ASSIGNMENT.EMP\_NUM

GROUP BY EMP\_NUM;

1. ASSIGNMENT 관계표에 나타난 각 프로젝트에 대해 총 투입시간과 총 비용을 프로젝트 번호 순으로 정렬하여 보여주는 SQL문을 작성하시오. 검색 결과는 다음과 같아야 한다. (4 점)



SELECT PROJ\_NUM, SUM(ASSIGN\_HOURS), SUM(ASSIGN\_CHARGE)

FROM ASSIGNMENT

GROUP BY PROJ\_NUM

ORDER BY PROJ\_NUM;

1. 모든 직원의 총 근무시간과 총 비용을 보여주는 SQL문을 작성하여 다음과 같은 결과가 나타나게 하시오. (4 점)

D:\lecture\2020-1\DB Management\시험문제\9781337627900_ch07_0222-t3.png

SELECT SUM(ASSIGN\_HOURS), SUM(ASSIGN\_CHARGE)

FROM ASSIGNMENT;

1. 다음 질문들에 대해 SQL문을 작성하시오. (총 16 점)

G.1. 아래에서 보여준 구조를 갖는 EMP\_1 관계표를 생성하는 SQL문을 작성하시오. Use EMP\_NUM을 주 키(primary key)로 하고, JOB\_CODE는 외래키로 JOB 관계표를 참조하며 참조무결성을 확보해야 한다. EMP\_LNAME과 EMP\_FNAME은 NULL값을 허용할 수 없다. (4 점)

EMP\_1 Table structure

|  |  |
| --- | --- |
| **ATTRIBUTE (FIELD) NAME** | **DATA DECLARATION** |
| EMP\_NUM | CHAR(3) |
| EMP\_LNAME | VARCHAR(15) |
| EMP\_FNAME | VARCHAR(15) |
| EMP\_INITIAL | CHAR(1) |
| EMP\_HIREDATE | DATE |
| JOB\_CODE | CHAR(3) |

CREATE TABLE EMP\_1 (

EMP\_NUM CHAR(3) CONSTRAINT EMP\_1\_EMP\_NUM\_PK PRIMARY KEY,

EMP\_LNAME VARCHAR(15) NOT NULL,

EMP\_FNAME VARCHAR(15) NOT NULL,

EMP\_INITIAL CHAR(1),

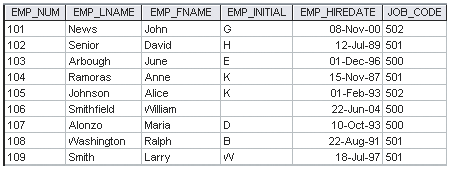
EMP\_HIREDATE DATE,

JOB\_CODE CHAR(3),

CONSTRAINT EMP\_1\_JOB\_CODE\_FK

FOREIGN KEY(JOB\_CODE)REFERENCES JOB);

G.2. 다음에서 보여준 row들 중 첫 2개의 row를 앞에서 생성한 관계표 EMP\_1에 추가하는 SQL문을 작성하시오. (4 점)



INSERT INTO EMP\_1 VALUES(‘101’, ‘News’, ‘John’, ‘G’, ‘08-Nov-00’, ‘502’);

INSERT INTO EMP\_1 VALUES(‘102’, ‘Senior’, ‘David’, ‘H’, ‘12-JUL-89’, ‘501’);

G.3. 사번(EMP\_NUM)이 107 인 사원의 job code를 501로 변경하는 SQL문을 작성하시오. (4 점)

UPDATE EMP\_1

SET JOB\_CODE = 501

WHERE EMP\_NUM = 107;

G.4. EMP\_1 관계표의 모든 자료를 포함하여 복제한 관계표 EMP\_2를 생성하는 SQL문을 작성하시오. (4 점)

CREATE TABLE EMP\_2 AS SELECT \* FROM EMP\_1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE NAME: PRODUCT | |  | TABLE NAME: PART | |
| PROD\_CODE | PROD\_QOH |  | PART\_CODE | PART\_QOH |
| ABC | 1,205 |  | A | 567 |
|  |  |  | B | 98 |
|  |  |  | C | 549 |

H.상품 ABC는 세가지 부품 A, B, C 1개를 조립하여 만들어진다. ABC 상품이 한 개 새로 만들어질 때마다 PRODUCT 관계표의 현 상품재고량(PROD\_QOH)이 하나 증가해야 하고, 동시에 PART 관계표의 A, B, C의 현 부품재고량(PART\_QOH)이 하나씩 감소해야 한다. DB의 관계표는 아래와 같다. 다음 질문에 답하시오. (총 10 점)

H.1. PRODUCT, PART 두 관계표의 재고량을 갱신하기 위해 DB 요청이 몇 개 필요한가? (5 점)

답 : 2개

H.2. 문제 H.1.의 답에 따라 SQL문을 작성하시오. (5 점)

UPDATE PRODUCT

SET PROD\_QOH = PROD\_QOH + 1

WHERE PROD\_CODE = ‘ABC’;

UPDATE PART

SET PART\_QOH = PART\_QOH - 1

WHERE PART\_CODE = ‘A’OR PART\_CODE = ’B’OR PART\_CODE = ’C’;