**DML (Insert, Update, Delete) 문 실습 과제**

**HR4 계정을 생성후 아래의 테이블과 값을 넣고 문제를 푸세요.**

**✅ 예제 테이블: EMPLOYEES, DEPARTMENTS, PROJECTS**

|  |
| --- |
| -- 부서 테이블  CREATE TABLE DEPARTMENTS (  DEPT\_ID NUMBER PRIMARY KEY,  DEPT\_NAME VARCHAR2(50)  );  -- 사원 테이블  CREATE TABLE EMPLOYEES (  EMP\_ID NUMBER PRIMARY KEY,  EMP\_NAME VARCHAR2(50),  SALARY NUMBER,  DEPT\_ID NUMBER,  HIRE\_DATE DATE,  CONSTRAINT fk\_dept FOREIGN KEY (DEPT\_ID) REFERENCES DEPARTMENTS(DEPT\_ID)  );  -- 프로젝트 테이블  CREATE TABLE PROJECTS (  PROJ\_ID NUMBER PRIMARY KEY,  PROJ\_NAME VARCHAR2(100),  EMP\_ID NUMBER,  START\_DATE DATE,  CONSTRAINT fk\_proj\_emp FOREIGN KEY (EMP\_ID) REFERENCES EMPLOYEES(EMP\_ID)  ); |

**✅ 초기 데이터 입력**

sql

|  |
| --- |
| -- DEPARTMENTS  INSERT INTO DEPARTMENTS VALUES (10, '개발팀');  INSERT INTO DEPARTMENTS VALUES (20, '영업팀');  INSERT INTO DEPARTMENTS VALUES (30, '디자인팀');  -- EMPLOYEES  INSERT INTO EMPLOYEES VALUES (1001, '김철수', 3500000, 10, TO\_DATE('2020-01-10', 'YYYY-MM-DD'));  INSERT INTO EMPLOYEES VALUES (1002, '이영희', 4000000, 10, TO\_DATE('2019-03-12', 'YYYY-MM-DD'));  INSERT INTO EMPLOYEES VALUES (1003, '박지민', 2800000, 20, TO\_DATE('2021-06-01', 'YYYY-MM-DD'));  INSERT INTO EMPLOYEES VALUES (1004, '정국', 5000000, 30, TO\_DATE('2018-09-21', 'YYYY-MM-DD'));  INSERT INTO EMPLOYEES VALUES (1005, '최유리', 3200000, 20, TO\_DATE('2022-02-14', 'YYYY-MM-DD'));  -- PROJECTS  INSERT INTO PROJECTS VALUES (2001, '홈페이지 리뉴얼', 1001, TO\_DATE('2023-01-15', 'YYYY-MM-DD'));  INSERT INTO PROJECTS VALUES (2002, '고객관리 시스템', 1002, TO\_DATE('2023-03-01', 'YYYY-MM-DD'));  INSERT INTO PROJECTS VALUES (2003, '신제품 마케팅', 1003, TO\_DATE('2023-05-10', 'YYYY-MM-DD'));  INSERT INTO PROJECTS VALUES (2004, '브랜드 리디자인', 1004, TO\_DATE('2022-11-05', 'YYYY-MM-DD'));  INSERT INTO PROJECTS VALUES (2005, '데이터 분석 시스템', 1005, TO\_DATE('2023-07-01', 'YYYY-MM-DD')); |

**✅ DML 실습 문제 15제**

**🔹 INSERT 관련**

1. 새로운 부서 기획팀(DEPT\_ID=40)을 추가하세요.

|  |
| --- |
| // 코드 작성 [답안]  INSERT INTO DEPARTMENTS (DEPT\_ID, DEPT\_NAME)  VALUES (40, '기획팀'); |

1. 사원 김민수(EMP\_ID=1006)를 기획팀에 추가하세요. (급여 300만원, 입사일: 2024-01-01)

|  |
| --- |
| // 코드 작성 [답안]  INSERT INTO EMPLOYEES (EMP\_ID, EMP\_NAME, SALARY, DEPT\_ID, HIRE\_DATE)  VALUES (1006, '김민수', 3000000, 40, TO\_DATE('24/01/01', 'YY/MM/DD')); |

1. 프로젝트 사내 교육 시스템(PROJ\_ID=2006)을 김민수에게 배정하세요.

|  |
| --- |
| // 코드 작성 [답안]  INSERT INTO PROJECTS (PROJ\_ID, PROJ\_NAME, EMP\_ID, START\_DATE)  VALUES (2006, '사내 교육 시스템', 1006, SYSDATE); |

1. 새로운 사원 고유정(1007)을 영업팀에, 급여 310만원으로 추가하세요.

|  |
| --- |
| // 코드 작성 [답안]  INSERT INTO EMPLOYEES (EMP\_ID, EMP\_NAME, SALARY, DEPT\_ID, HIRE\_DATE)  VALUES (1007, '고유정', 3100000, 20, SYSDATE); |

1. 정우성(1008)을 부서 없이 등록해보세요. (DEPT\_ID를 NULL로)

|  |
| --- |
| // 코드 작성 [답안]  INSERT INTO EMPLOYEES (EMP\_ID, EMP\_NAME, SALARY, DEPT\_ID, HIRE\_DATE)  VALUES (1008, '정우성', 0, NULL, SYSDATE); |

**🔹 UPDATE 관련**

1. 김철수의 급여를 380만원으로 수정하세요.

|  |
| --- |
| // 코드 작성 [답안]  UPDATE EMPLOYEES  SET SALARY = 3800000  WHERE EMP\_NAME = '김철수'; |

1. 박지민의 부서를 디자인팀으로 변경하세요.

|  |
| --- |
| // 코드 작성 [답안]  UPDATE EMPLOYEES  SET DEPT\_ID = 30  WHERE EMP\_NAME = '박지민'; |

1. 홈페이지 리뉴얼 프로젝트를 정국에게 재배정하세요.

|  |
| --- |
| // 코드 작성 [답안]  UPDATE PROJECTS  SET EMP\_ID = 1004  WHERE PROJ\_NAME = '홈페이지 리뉴얼'; |

1. 최유리의 이름을 최윤아로 변경하세요.

|  |
| --- |
| // 코드 작성 [답안]  UPDATE EMPLOYEES  SET EMP\_NAME = '최윤아'  WHERE EMP\_NAME = '최유리'; |

1. 모든 영업팀 직원의 급여를 10% 인상하세요.

|  |
| --- |
| // 코드 작성 [답안]  UPDATE EMPLOYEES  SET SALARY = SALARY \* 1.1  WHERE DEPT\_ID = 20; |

**🔹 DELETE 관련**

1. 급여가 300만원 이하인 사원을 삭제하세요.

|  |
| --- |
| // 코드 작성 [답안]  DELETE FROM EMPLOYEES  WHERE SALARY <= 3000000; |

1. 기획팀의 부서를 삭제하려면 어떤 제약 조건을 고려해야 하나요? 삭제를 시도해보세요.

|  |
| --- |
| // 코드 작성 [답안]  DELETE FROM DEPARTMENTS  WHERE DEPT\_ID = 40; |

1. 사원 번호가 1008인 사람을 삭제하세요.

|  |
| --- |
| // 코드 작성 [답안]  DELETE FROM EMPLOYEES  WHERE EMP\_ID = 1008; |

1. 김민수의 프로젝트를 삭제하세요.

|  |
| --- |
| // 코드 작성 [답안]  DELETE FROM PROJECTS  WHERE EMP\_ID = 1006; |

1. 부서가 없는 사원을 모두 삭제하세요.

|  |
| --- |
| // 코드 작성 [답안]  DELETE FROM EMPLOYEES  WHERE DEPT\_ID IS NULL; |

**✅ 추가 팁**

* DELETE 전에 SELECT로 미리 확인하기
* ROLLBACK으로 실습 후 복원 가능
* 실습 중 CONSTRAINT 오류 발생 시 CASCADE CONSTRAINTS 활용