

## SQL 활용 통합 실습

※ HR 사용자로 접속하여 다음 지시사항을 SELECT 문으로 작성하시오.

1. 사원테이블에서 사원번호, 이름(last\_name), 업무, 월급여를 출력하시오. 월 급여는 salary의 92%를 계산하고 월 별칭을 MONTHLY\_SALARY로 지정하시오. 결과는 급여를 기준으로 내림차순 정렬하여 출력하시오.

```
SELECT employee_id, last_name, job_id, salary*0.92 AS monthly_salary  
  
FROM employees  
  
ORDER BY 4 DESC;
```

2. 사원테이블에서 LAST\_NAME이 소문자 s로 끝나는 사원의 이름, 업무, 급여, 입사연도를 출력하시오. CONCAT 연산자를 사용하여 이름과 성을 공백을 포함하여 연결해서 월 레이블을 NAME으로 표시하고, 급여는 원화기호와 천 단위 구분기호를 포함하시오. 입사연도는 4자리 모두 표시하시오.

```
SELECT CONCAT(CONCAT(first_name, ' '), last_name) AS name, TO_CHAR(salary, 'L999,999') ,  
  
        TO_CHAR(hire_date, 'yyyy')  
  
FROM employees  
  
WHERE last_name LIKE '%s';
```

3. 1998년 1월 1일 이후에 입사한 사원의 사원번호, 이름(last\_name), 관리자 번호, 부서번호를 출력하시오. 관리자가 없는 사원은 No Manager로 출력하시오.

```
SELECT employee_id, last_name, NVL(TO_CHAR(manager_id), 'No Manager') AS manager, department_id  
  
FROM employees  
  
WHERE hire_date >='98/01/01';
```

4. EMPLOYEES와 DEPARTMENTS를 조인하여 사원번호, 사원의 성(LAST\_NAME), 급여, 부서이름을 출력하시오.

```
SELECT e.employee_id, e.last_name, e.salary, d.department_name  
  
FROM employees e JOIN departments d  
  
ON (e.department_id = d.department_id);
```

5. last\_name이 Rais인 직원과 업무가 동일하면서 급여가 적은 사원의 사원번호, 이름, 업무, 급여를 출력하시오.

```
SELECT employee_id, last_name, job_id, salary FROM employees  
  
WHERE job_id = (SELECT job_id FROM employees  
  
                WHERE last_name = 'Rajs')  
  
AND salary < (SELECT salary FROM employees  
  
              WHERE last_name = 'Rajs');
```

6. 업무ID별 인원수와, 총급여를 표시하시오

```
SELECT job_id, COUNT(*), SUM(salary)

FROM employees

GROUP BY job_id;
```

7. 다음 지시사항을 참조하여 테이블을 생성하시오.

(1) 다음은 웹 사이트의 게시판을 사용하는 회원을 관리하기 위한 회원 테이블 인스턴스입니다.. 회원 테이블 인스턴스를 참고하여 테이블을 생성하시오.

테이블명 : MEMBER

열이름	데이터타입(길이)	제약조건	제약조건이름	설명
USERID	NUMBER(6)	Primary key	-	회원번호
USERNAME	VARCHAR2(20)	Not Null	member_name_nn	회원이름
PASSWD	VARCHAR2(20)	Not Null	member_pw_nn	암호
IDNUM	VARCHAR2(13)	Unique Not Null	member_id_uk member_id_nn	주민등록번호
PHONE	VARCHAR2(13)			전화번호
ADDRESS	VARCHAR2(25)			주소
REGDATE	DATE			가입일
INTEREST	VARCHAR2(15)			관심분야

CREATE TABLE member

(USERID NUMBER(6) primary key,

USERNAME varchar2(20) constraint member\_name\_nn not null,

PASSWD varchar2(20) constraint member\_pw\_nn not null,

IDNUM varchar2(13) constraint member\_id\_nn not null

constraint member\_id\_uk unique,

PHONE varchar2(13),

ADDRESS varchar2(25),

REGDATE date,

INTEREST varchar2(15));

(2) 다음은 웹 사이트의 게시판을 사용하는 회원을 관리하기 위한 게시판 테이블 인스턴스입니다. 인스턴스를 참고하여 테이블을 생성하시오.

테이블명 : BOARD

열이름	데이터타입(길이)	제약조건	제약조건이름	설명
NO	NUMBER(4)	Primary key	-	게시글번호
SUBJECT	VARCHAR2(50)	Not Null	Board_sub_nn	제목
CONTENT	VARCHAR2(2000)			내용
RDATE	DATE			작성일
USERID	NUMBER(6)	Foreign Key	Board_uid_fk	MEMBER(USERID)참조

CREATE TABLE board

(NO number(4) primary key,

```
SUBJECT varchar2(50) constraint board_sub_nn not null,  
  
CONTENT varchar2(2000),  
  
RDATE date,  
  
USERID number(6) constraint board_uid_fk references MEMBER(USERID));
```

8. 게시판의 NO 열에 사용할 board\_no\_seq라는 이름의 시퀀스를 생성하시오. 시퀀스의 시작번호와 증분값은 각각 1로 설정하고 NOCACHE, NOCYCLE 속성을 주시오.

```
CREATE SEQUENCE board_no_seq  
  
INCREMENT BY 1  
  
START WITH 1  
  
NOCACHE  
  
NOCYCLE;
```

9. 회원 테이블과 게시판 테이블을 참조하여 무결성 제약조건을 위반하지 않는 데이터를 3건씩 입력하시오. 게시판의 번호는 board\_no\_seq 시퀀스를 사용하여 입력하시오.

```
INSERT INTO member  
  
VALUES('101','송성광','비밀번호','7906021234567','051-123-1234','부산 수정동',sysdate,'DB');  
  
INSERT INTO member  
  
VALUES('102','김영균','비밀번호1','7903022341567','051-321-1234','창원 사림동',sysdate,'internet');  
  
INSERT INTO member  
  
VALUES('103','전인하','비밀번호2','7901041324668','051-345-3456','부산 동삼동',sysdate,'java');  
  
INSERT INTO board  
  
VALUES(board_no_seq.nextval,'제목','내용',sysdate,'101');  
  
INSERT INTO board  
  
VALUES(board_no_seq.nextval,'제목1','내용1',sysdate,'102');  
  
INSERT INTO board  
  
VALUES(board_no_seq.nextval,'제목2','내용2',sysdate,'103');  
  
COMMIT;
```

10. 회원 테이블에 email 열을 추가하시오. 단, email 열의 데이터 타입은 VARCHAR2(50)이다.

```
ALTER TABLE member ADD email VARCHAR2(50);
```

11. 회원 테이블에 국적을 나타내는 country 열을 추가하시오. 데이터타입은 VARCHAR2(20), 기본값은 'Korea'로 지정하시오. 그리고 나서 모든 행의 국적이 Korea로 저장된 것을 확인하시오.

```
ALTER TABLE member ADD country VARCHAR2(20) DEFAULT 'Korea';
```

```
SELECT * FROM member;
```

12. 회원 테이블에서 IDNUM 열을 삭제하시오.

```
ALTER TABLE member DROP COLUMN idnum;
```

13. 회원 테이블의 address 열의 데이터 크기를 50으로 변경하시오.

```
ALTER TABLE member MODIFY address VARCHAR2(50);
```

14. 게시판 테이블의 userid 열에 대해 board\_userid\_ix 라는 이름의 인덱스를 생성하시오.

```
CREATE INDEX board_userid_ix ON board(userid);
```

15. 회원테이블에서 유저아이디, 이름, 전화번호, 주소를 볼 수 있는 member\_addr\_phone\_list\_vu라는 이름의 뷰를 생성하시오.

```
CREATE VIEW member_addr_phone_list_vu
```

```
AS
```

```
SELECT userid, username, phone, address
```

```
FROM member;
```

```
SELECT * FROM member_addr_phone_list_vu;
```

16. 게시판글번호, 제목, 유저아이디를 볼 수 있는 board\_list\_vu라는 이름의 뷰를 생성하시오.

```
CREATE VIEW board_list_vu
```

```
AS
```

```
SELECT no, subject, userid
```

```
FROM board;
```

```
SELECT * FROM board_list_vu;
```

17. member\_addr\_phone\_list\_vu와 board\_list\_vu라는 이름의 뷰에 대하여 각각 m과 b라는 이름의 동의어를 생성하시오.

```
CREATE SYNONYM m FOR member_addr_phone_list_vu;
```

```
CREATE SYNONYM b FOR board_list_vu;
```

18. board\_list\_vu 라는 뷰에서 작성일(RDATE)가 보이도록 뷰를 수정하시오.

```
CREATE OR REPLACE VIEW board_list_vu AS
```

```
SELECT no, subject, userid, rdate
```

```
FROM board;
```

19. USER\_CONSTRAINTS 및 USER\_OBJECTS를 조회하여 제약조건과 사용자 소유의 객체의 이름, 타입, 상태 등을 조회하시오.

```
SELECT constraint_name, constraint_type  
  
FROM user_constraints  
  
WHERE table_name = 'MEMBER';  
  
SELECT object_name, object_type, status  
  
FROM user_objects  
  
WHERE object_name LIKE 'BOARD%'  
  
OR object_name LIKE 'MEMBER%'  
  
OR object_name IN ('B','M')  
  
ORDER BY 2;
```

20. board 와 member테이블을 삭제하시오.

```
DROP TABLE member;  
  
DROP TABLE board;  
  
DROP TABLE member;
```

21. USER\_OBJECTS를 조회하여 사용자 소유의 객체의 이름, 타입, 상태 등을 조회하면서 테이블 삭제의 결과를 정리하시오.

```
SELECT object_name, object_type, status  
  
FROM user_objects  
  
WHERE object_name LIKE 'BOARD%'  
  
OR object_name LIKE 'MEMBER%'  
  
OR object_name IN ('B','M')  
  
ORDER BY 2;
```