

# 240610

--1. 회원(member) 테이블을 만들고 다음과 같이 값을 삽입하는 명령

```
CREATE TABLE MEMBER(  
    ID NUMBER PRIMARY KEY  
    , NAME VARCHAR2(20)  
    , SAL NUMBER  
    , BIRTHDAY DATE  
);
```

```
INSERT INTO MEMBER(ID, NAME, SAL, BIRTHDAY) VALUES(1, '사랑해',  
INSERT INTO MEMBER(ID, NAME, SAL, BIRTHDAY) VALUES(2, '최보람',  
INSERT INTO MEMBER(ID, NAME, SAL, BIRTHDAY) VALUES(3, '고마와',  
INSERT INTO MEMBER(ID, NAME, SAL, BIRTHDAY) VALUES(4, '오감사',  
INSERT INTO MEMBER(ID, NAME, SAL, BIRTHDAY) VALUES(5, '이가을',  
INSERT INTO MEMBER(ID, NAME, BIRTHDAY) VALUES(6, '배우자', '1994년07월08일');
```

--2. select sum(sal), avg(sal), count(\*), count(sal) from member;  
select sum(sal), avg(sal), count(\*), count(sal) from member;

--3. sal의 값이 null인 튜플의 모든 정보를 출력하는 명령

```
SELECT * FROM MEMBER WHERE SAL IS NULL;
```

--4. select \* from member where sal=''; 의 결과를 쓰고 그 이유를 조사  
select \* from member where sal='';

--5. member 테이블에서 생일이 1994년 7월 8일에서 7월 9일 사이에 태어난

--회원번호, 이름, 봉급, 생일 검색 명령. 단, 생일은 ex) '1994년07월09일'  
SELECT ID, NAME, SAL, TO\_CHAR(BIRTHDAY, 'YYYY"년"MM"월"DD"일" )

--6. DBMS 서버에 설정된 현재 시간과 오늘 날짜를 확인ex) 2018년 09월 17

```
SELECT TO_CHAR(SYSDATE, 'YYYY"년" MM"월" DD"일" DY HH24:MI:SS')
```

--7. member 테이블에서 모든 정보를 앞의 두 명만 검색

```

SELECT *
FROM (SELECT * FROM MEMBER ORDER BY ID)
WHERE ROWNUM <= 2;

--8. select * from (select * from member order by sal) bwhere
-- select * from (select * from member where rownum<=3) order

--9. sal 이 가장 많은 두 명의 모든 정보를 검색
SELECT *
FROM (SELECT * FROM MEMBER ORDER BY SAL DESC)
WHERE ROWNUM <= 2 AND SAL IS NOT NULL;

--10. member테이블에 party라는 속성을 추가하고 생일에서 각각 100일 지난
--ALTER TABLE MEMBER MODIFY PARTY DATE;
--UPDATE MEMBER SET BIRTHDAY = DATE_ADD(BIRTHDAY, INTERVAL '1

--11. 스칼라 부속질의를 사용하여 마당서점의 도서별 도서명과 판매액 합계를 :

--12. 도서번호가 4 이하인 도서의 판매액을 보이는 명령 (결과는 도서명과 도

--13. 대한미디어에서 출판한 도서의 총판매액을 구하는 명령
-- (where 부속 질의 사용!)
SELECT SUM(SALEPRICE) FROM ORDERS WHERE BOOKID IN (SELECT BOO
-- (join사용!)
SELECT SUM(SALEPRICE) FROM BOOK A JOIN ORDERS B ON(A.BOOKID=B

--14. 2번 도서의 최고 가격보다 더 비싼 금액의 도서명과 가격 검색
SELECT BOOKNAME, PRICE FROM BOOK WHERE PRICE > (SELECT MAX(PR

--15. 고객번호가 3번 이하인 고객들에게 판매한 판매액 합계(exists 사용!)
SELECT SUM(SALEPRICE) FROM ORDERS A WHERE EXISTS (SELECT 1 FR

--16. EXISTS 연산자로 대한민국에 거주하지 않는 고객에게 판매한 도서의 총
SELECT SUM(SALEPRICE) FROM ORDERS A WHERE EXISTS (SELECT 1 FR

--17. 대한미디어에서 출판한 도서의 모든 정보를 보여주는 뷰인 vw_b_d 생성

```

```
CREATE VIEW vw_b_d AS SELECT * FROM BOOK WHERE PUBLISHER = '다
```

```
--18. 대한미디어에서 출판한 도서의 모든 정보 중 가격이 2만원대인 도서정보를  
SELECT * FROM vw_b_d WHERE PRICE BETWEEN 20000 AND 29999;
```

```
--19. vw_b_d 뷰를 삭제하는 명령  
DROP VIEW vw_b_d;
```

```
--20. 판매가격이 15,000원 이상인 도서의 도서번호, 도서이름, 고객이름, 출  
CREATE VIEW vw_o AS  
SELECT A.BOOKID, A.BOOKNAME, C.NAME, A.PUBLISHER, B.SALEPRIC  
JOIN CUSTOMER C ON(B.CUSTID=C.CUSTID)  
WHERE SALEPRICE >= 15000;
```

```
--21. 생성한 뷰를 이용하여 판매된 도서의 이름과 고객의 이름을 출력하는 SQL  
SELECT BOOKNAME, NAME FROM vw_o;
```

```
--22. vw_o 뷰를 변경하고자 한다. 판매가격 속성을 삭제하는 명령, 삭제 후
```