

기본 쿼리문

DB 수업 목차

SQL

책 추천

세팅

User 계정 생성

oracle_exam01.sql 불러오기

sqlDeveloper 세팅

hr계정 unlock

scott 계정 불러오기

이후 계정 세팅

기본 쿼리문

실습 1

정렬

실습 2

조건절

연산자

AND

BETWEEN AND

OR

IN

NOT

IS NOT NULL

LIKE

실습 3

과제

DB 수업 목차

SQL

- 기본 SELECT
- 오라클 함수, 연산자
- 테이블 제약 조건
- 조인

- 서브쿼리문
- 시퀀스, 뷰, 인덱스
- DB모델링

책 추천

- 오브젝트, 조영호
- 마틴 파울러 객체 지향 책
- 객체 지향의 사실과 오해

세팅

```
sqlplus/nolog  
conn sys as sysdba
```

```
C:\Users\COM>sqlplus/nolog  
  
SQL*Plus: Release 11.2.0.2.0 Production on 목 2월 9 16:14:14 2023  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
  
SQL> conn sys as sysdba  
Enter password:  
Connected.  
SQL> _
```

User 계정 생성

```
// 비밀번호로 사용자 생성  
create user kosa01 identified by 1234;  
// 권한 부여  
grant connect, resource, dba to kosa01;  
// 접속  
conn kosa01/1234;
```

```

connected.
SQL> create user kosa01 identified by 1234;

User created.

SQL> grant connect, resource, dba to kosa01;

Grant succeeded.

SQL> conn kosa01/1234;
Connected.
SQL> _

```

- 나오기: ctrl + c

oracle_exam01.sql 불러오기

@D:\oracle_exam2.sql

```

Table created.

DROP TABLE emp3
      *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

DROP TABLE st_score
      *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

Recyclebin purged.

Session altered.

SQL>

```

SQL > commit

sqlDeveloper 세팅

```
C:\Program Files\Java\jdk-11.0.12
```

hr계정 unlock

```
conn /as sysdba
alter user hr identified by hr account unlock;
conn hr/hr
```

```
SQL> ^C
C:\Users\COM>sqlplus/nolog

SQL*Plus: Release 11.2.0.2.0 Production on 목 2월 9 16:22:55 2023

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> conn /as sysdba
Connected.
SQL> alter user hr identified by hr account unlock;

User altered.

SQL> conn hr/hr
Connected.
SQL>
```

```
select * from tab;
```

```

TNAME
-----
CLUSTERID
-----
COUNTRIES                                TABLE

DEPARTMENTS                            TABLE

EMPLOYEES                              TABLE

TNAME
-----
CLUSTERID
-----
EMP_DETAILS_VIEW                        VIEW

JOBS                                    TABLE

JOB_HISTORY                            TABLE

TNAME
-----
CLUSTERID
-----
LOCATIONS                              TABLE

REGIONS                                TABLE

8 rows selected.

SQL> _

```

scott 계정 불러오기

- 경로 복사: C:\oracle\app\oracle\product\11.2.0\server\rdbms\admin

```

conn /as sysdba
@C:\oracle\app\oracle\product\11.2.0\server\rdbms\admin\scott.sql
alter user scott identified by tiger;
conn scott/tiger
select * from tab;

```

```

SQL> conn /as sysdba
Connected.
SQL> @C:\oracle\exe\app\oracle\product\11.2.0\server\rdbms\admin\scott.sql
SQL> alter user scott identified by tiger;

User altered.

SQL> conn scott/tiger
Connected.
SQL> select * from tab;

```

TNAME	TABTYPE
CLUSTERID	
BONUS	TABLE
DEPT	TABLE
EMP	TABLE
TNAME	TABTYPE
CLUSTERID	
SALGRADE	TABLE

이후 계정 세팅

- hr 계정 불러오기

새로 만들기/데이터베이스 접속 선택

접속 이름: 접속 세부정보

Name: hr

데이터베이스 유형: Oracle

사용자 정보 | 프록시 사용자

인증 유형: 기본값

사용자 이름(U): hr

비밀번호(P): ..

롤(L): 기본값

☒ 비밀번호 저장(Y)

접속 유형(Y): 기본

세부정보 | 고급

호스트 이름(A): localhost

포트(R): 1521

☒ SID(I): xe

☐ 서비스 이름(E):

상태: 성공

도움말(H) | 저장(S) | 지우기(C) | 테스트(T) | 접속(O) | 취소

- kosa 계정 불러오기

새로 만들기/데이터베이스 접속 선택

접속 이름: hr

접속 세부정보: hr@//localhos...

Name: kosa01

데이터베이스 유형: Oracle

사용자 정보 | 프록시 사용자

인증 유형: 기본값

사용자 이름(U): kosa01

비밀번호(P):

롤(L): 기본값

☐ 비밀번호 저장(Y)

접속 유형(Y): 기본

세부정보 | 고급

호스트 이름(A): localhost

포트(R): 1521

☒ SID(I): xe

☐ 서비스 이름(E):

상태: 성공

도움말(H) | 저장(S) | 지우기(C) | 테스트(T) | 접속(O) | 취소

- scott 계정 불러오기

새로 만들기/데이터베이스 접속 선택

접속 이름	접속 세부정보
hr	hr@//localhos...
kosa01	kosa01@//loc...

Name Color ☐

데이터베이스 유형

사용자 정보 프록시 사용자

인증 유형

사용자 이름(U) 룰(L)

비밀번호(P) ☒ 비밀번호 저장(Y)

접속 유형(Y)

세부정보 고급

호스트 이름(A)

포트(P)

☒ SID(J)

☐ 서비스 이름(E)

상태: 성공

기본 쿼리문

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/907ac98b-6501-4e8d-8030-5a2ea3e2b3ad/%EA%B0%81_%ED%85%8C%EC%9D%B4%E%B8%94_%EC%83%81%EC%84%B8%EA%B5%AC%EC%A1%B0.hwp

각 테이블 상세구조

emp(사원)		dept(부서)		salgrade(급여등급)	
컬럼명	내용	컬럼명	내용	컬럼명	내용
eno	사번	dno	부서번호	grade	급여등급
ename	이름	dname	부서명	losal	하한값
job	업무	loc	지역	hosal	상한값
mgr	관리자 사번	director	부서장사번		
hdate	입사일				
sal	급여(월)				
comm	보너스				
dno	부서번호				

student(학생)		professor(교수)		course(과목)	
컬럼명	내용	컬럼명	내용	컬럼명	내용
sno	학번	pno	교수번호	cno	과목번호
sname	이름	pname	이름	cname	과목명
sex	성별	section	소속학과	st_num	학점수
syear	학년	orders	직위	pno	교수번호
major	학과	hiredate	부임일		
avr	평점				

score(기말고사 점수)		scgrade(점수등급)	
컬럼명	내용	컬럼명	내용
sno	학번	grade	등급
cno	과목번호	hiscore	상한값
result	점수	loscore	하한값

- 테이블 목록 불러오기

```
SELECT * FROM tab;
```

- 테이블 구조 불러오기

```
DESC 테이블명;
```

- 별칭 ⇒ AS, ""

```
SELECT employee_id AS 사원번호, last_name "사원번호" FROM employees;
```

실습 1

- 각 학생 평점 검색하기 (학번, 이름, 학점): student → 별칭 사용

```
SELECT SNO AS 학번, SNAME AS 이름, AVR AS 학점 FROM STUDENT;
```

- 각 과목 학점 검색하기 (과목번호, 과목명, 학점수): course → 별칭

```
SELECT CNO AS 과목번호, CNAME AS 과목명, ST_NUM AS 학점수 FROM COURSE;
```

- 각 교수의 직위 검색 (교수 번호, 교수 이름, 직위): professor → 별칭

```
SELECT PNO AS 교수번호, PNAME AS 교수이름, ORDERS AS 직위 FROM PROFESSOR;
```

- 급여 10% 인상했을 때 각 직원마다 연간 지급되는 급여 검색(사원 번호, 사원 이름, 연봉): emp → 별칭

```
SELECT eno "사원번호", ename "사원이름", sal*1.1 "연봉" FROM emp;
```

- 현재 학생 평점 4.0 만점. 이를 4.5 만점으로 환산하여 검색. student(학번, 이름, 환산 학점) → 별칭

```
SELECT SNO AS 학번, SNAME AS 이름, AVR*4.5/4.0 AS 환산학점 FROM STUDENT;
```

정렬

- 묶음 검색
- ORDER BY절

```
SELECT 컬럼들 FROM 테이블명 ORDER BY 절;  
ORDER BY [컬럼명] [인덱스] [옵션(DESC)]  
// 첫 번째 컬럼으로 정렬 후에 두 번째 컬럼으로 정렬  
ORDER BY [컬럼명], [컬럼명]
```

실습 2

- 각 학과별로 교수의 정보를 부임일자 순으로 검색

```
SELECT section, pname, hiredate FROM professor ORDER BY section, hiredate;
```

조건절

- 조건에 맞는 일부 데이터 불러오기
- WHERE 절

```
SELECT employee_id, last_name, hire_date FROM employees  
WHERE hire_date >= '03/01/01'  
AND last_name = 'King';
```

연산자

AND

- 연봉이 5000 이상 10000 이하 사이 직원들만 출력

```
SELECT employee_id, last_name, salary FROM employees  
WHERE salary >= 5000 AND salary <= 10000
```

BETWEEN AND

```
SELECT employee_id, last_name, salary FROM employees  
WHERE salary BETWEEN 5000 AND 10000;
```

OR

```
SELECT employee_id, last_name, job_id
FROM employees
WHERE job_id = 'FI_MGR' OR job_id = 'FI_ACCOUNT';
```

IN

- OR 연산자를 **대체** 가능
- 컬럼명을 중복으로 쓰지 않아도 됨

```
SELECT employee_id, last_name, job_id
FROM employees
WHERE job_id IN('FI_MGR', 'FI_ACCOUNT')
```

NOT

- **!=**, **<>**, **<**, **>**, **<=**, **>=**, **NOT** 으로 표현 가능

```
SELECT department_id, department_name FROM departments
WHERE department_id != 10;
WHERE department_id <> 10;
WHERE NOT department_id = 10;
```

IS NOT NULL

```
SELECT employee_id, last_name, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL;
```

LIKE

- 2007년도에 입사한 사람들 출력
- **BETWEEN AND** 로 대체 가능

```
SELECT employee_id, last_name, hire_date
FROM employees
WHERE hire_date BETWEEN '07/01/01' AND '07/12/31';

SELECT employee_id, last_name, hire_date
FROM employees
WHERE hire_date LIKE '07%';
```

실습 3

- LAST_NAME 컬럼에 'a'가 없는 사원 출력

```
SELECT employee_id, last_name
FROM employees
WHERE NOT last_name LIKE '%a%';
```

- 화학과 학생 중에 성이 '관'씨인 학생 검색

```
SELECT * FROM STUDENT
WHERE major = '화학' AND sname LIKE '관%';
```

- 부임일이 1995년 이전의 정교수 검색

```
SELECT * FROM PROFESSOR
WHERE hiredate < '95/01/01' AND ORDERS = '정교수'
```

- 성과 이름이 각각 한 글자인 교수 검색 (이름이 2자)
 - LIKE __

```
SELECT * FROM PROFESSOR
WHERE pname LIKE '__';
```

- 화학과 학생 중에 4.5 환산 학점이 3.5 이상인 학생 검색

```
SELECT * FROM STUDENT
WHERE avr * (4.5 / 4.0) >= 3.5;
```

- 화학과 이외 학과 학생의 평점을 각 학과별, 학년별 순서로 출력

```
SELECT SNO, SNAME, MAJOR, AVR FROM STUDENT
WHERE major != '화학'
ORDER BY MAJOR, SYEAR;
```

과제

- scott 계정

1. 부서 번호가 10번인 부서의 사람의 사원번호, 이름, 월급을, 출력

```
SELECT EMPNO AS 사원번호, ENAME AS 이름, SAL AS 월급 FROM EMP
WHERE DEPTNO = 10;
```

2. 사원번호가 7369인 사람 중 이름, 입사일, 부서번호를 출력하라.

```
SELECT empno "사원번호", ename "이름", hiredate "입사일", deptno "부서번호"
FROM EMP
WHERE empno = 7369;
```

3. 이름이 ALLEN인 사람의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE ENAME = 'ALLEN';
```

4. 직업이 MANAGER가 아닌 사람의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE JOB != 'MANAGER';
```

5. 81/04/02 이후에 입사한 사원의 정보를 출력하라.

```
SELECT * FROM EMP
WHERE HIREDATE > '81/04/02';
```

6. 급여가 \$800 이상인 사람의 이름, 급여, 부서번호를 출력하라.

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE SAL >= 800;
```

7. 부서번호가 20번 이상인 사원의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE DEPTNO >= 20;
```

8. 입사일이 81/12/09 보다 먼저 입사한 사람들의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE HIREDATE < '81/12/09';
```

9. 입사번호가 7698보다 작거나 같은 사람들의 입사번호와 이름을 출력하라.

```
SELECT EMPNO AS "입사번호", ENAME AS "이름" FROM EMP
WHERE EMPNO <= 7698;
```

10. 입사일이 81/04/02 보다 늦고 82/12/09 보다 빠른 사원의 이름, 월급, 부서번호를 출력하라.

```
SELECT ENAME "이름", SAL "월급", DEPTNO "부서번호" FROM EMP
WHERE HIREDATE > '81/04/02' AND HIREDATE < '82/12/09';
```

11. 급여가 \$1,600보다 크고 \$3,000보다 작은 사람의 이름, 직업, 급여를 출력하라.

```
SELECT ENAME "이름", JOB "직업", SAL "급여" FROM EMP
WHERE SAL > 1600 AND SAL < 3000;
```

12. 사원번호가 7654와 7782 사이가 아닌 사원의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE NOT EMPNO >= 7654 AND EMPNO <= 7782;

SELECT * FROM EMP
WHERE empno BETWEEN 7654 AND 7782;
```

13. 이름이 B와 J 사이의 모든 사원의 정보를 출력하라.

```
SELECT * FROM EMP
WHERE ename > 'B' AND ename < 'J';
```

14. 입사일이 81년 이외에 입사한 사람의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE EXTRACT(YEAR FROM hiredate) NOT LIKE '1981';

SELECT * FROM EMP
WHERE hiredate NOT LIKE '81%';
```

15. 직업이 MANAGER와 SALESMAN인 사람의 모든 정보를 출력하라.

```
SELECT * FROM EMP
WHERE job = 'MANAGER' OR job = 'SALESMAN';

SELECT * FROM EMP
WHERE JOB IN('MANAGER', 'SALESMAN');
```

16. 부서번호 20, 30번을 제외한 모든 사람의 이름, 사원번호, 부서번호를 출력하라.

```
SELECT ename "이름", empno "사원번호", deptno "부서번호" FROM emp
WHERE deptno <> 20 AND deptno <> 30;
```

17. 이름이 S로 시작하는 사원의 사원번호, 이름, 입사일, 부서번호를 출력하라.

```
SELECT empno "사원번호", ename "이름", hiredate "입사일", deptno "부서번호"
FROM emp
WHERE ename LIKE 'S%';
```

18. 입사일이 81년도인 사람의 모든 정보를 출력하라

```
SELECT * FROM emp
WHERE EXTRACT(YEAR FROM hiredate) = '1981';

SELECT * FROM emp
WHERE hiredate LIKE '81%';
```

19. 이름 중 S자가 들어가 있는 사람만 모든 정보를 출력하라

```
SELECT * FROM emp
WHERE ename LIKE '%S%';
```

20. 첫 번째 문자는 관계없고, 두 번째 문자가 A인 사람의 정보를 출력하라.

```
SELECT * FROM emp
WHERE ename like '_A%';
```

21. 커미션이 NULL인 사람의 정보를 출력하라.


```
SELECT * FROM emp
WHERE comm IS NULL;
```

22. 커미션이 NULL이 아닌 사람의 모든 정보를 출력하라.

```
SELECT * FROM emp
WHERE NOT comm IS NULL;

SELECT * FROM emp
WHERE comm IS NOT NULL;
```

23. 부서가 30번 부서이고 급여가 \$1,500 이상인 사람의 이름, 부서, 월급을 출력하라.

```
SELECT ename "이름", job "부서", sal "월급"
FROM emp
WHERE deptno = 30 AND sal >= 1500;
```

24. 이름 첫 글자가 K로 시작하거나 부서번호가 30인 사람의 사원번호, 이름, 부서번호를 출력하라.

```
SELECT empno as 사원번호, ename as 이름, deptno as 부서번호
FROM emp
WHERE ename like 'K%' OR deptno = 30;
```

25. 급여가 \$1,500 이상이고 부서번호가 30번인 사원 중 직업이 MANAGER인 사람 정보를 출력하라.

```
SELECT * FROM emp
WHERE sal >= 1500 AND deptno = 30 AND job = 'MANAGER';
```

26. 부서번호가 30인 사람 중 사원번호를 SORT하라.

```
SELECT * FROM emp
WHERE deptno = 30
ORDER BY empno;
```

27. 급여가 많은 순으로 SORT하라.

```
SELECT * FROM emp  
ORDER BY sal DESC;
```