<SQL 활용>

1. Select 구문을 사용한 데이터 검색	[문법] select * | 컬럼1, 컬럼2, 컬럼3 from 테이블명; - 테이블 구조 조회 : S> desc 테이블명 - null값 : 정의되지 않은 값, 모르는 값, 알 수 없는 값 모든 데이터타입에 사용 가능함 0 또는 공백과는 다른 특수한 값 산술식에 null값이 포함된 경우에는 전체 결과도 null임 - 컬럼 alias : 컬럼명 [as] alias 컬럼명 [as] "Alias" => 대소문자, 공백, 특수문자 포함 - 리터럴값 : 쿼리구문에 포함된 일반 문자, 숫자, 날짜값 문자, 날짜 리터럴값은 작은따옴표(' ')로 묶어야함 - 연결연산자(||) - distinct 키워드 : 중복값 제거
2. WHERE절과 ORDER BY절	[문법] select 컬럼1, 컬럼2, 컬럼3, ... from 테이블명 where 좌변 = 우변 order by 컬럼명 [ASC | DESC] - 좌변 : 비교할 컬럼명 - 우변 : 리터럴값 => 문자 : 대소문자 구분 날짜 : DD-MON-RR 형식으로 작성 - 비교연산자 : =, >, >=, <, <=, <>, !=, (not) between A and B, (not) in, (not) like, is (not) null - order by : 쿼리구문 마지막에 작성 정렬할 컬럼명 대신에 표현식, alias, 위치표기법 작성 가능
3. 단일 행 함수	1. 문자함수 1) 대소문자 변환함수 : lower, upper, initcap 2) 문자 조작함수 : concat, substr, instr, length, lpad, rpad, replace, trim 2. 숫자함수 : round, trunc, mod 3. 날짜함수 : sysdate, months_between, add_months, last_day, next_day, round, trunc 4. 변환함수 : to_char, to_number, to_date 5. 일반함수 : nvl, nvl2, nullif, coalesce
4. GROUP BY절과 HAVING절	[문법] select 컬럼1, 컬럼2, 컬럼3, ... from 테이블명 where 좌변 = 우변 group by 컬럼명 having 좌변 = 우변 //그룹함수가 포함된 조건문 order by 컬럼명 [ASC | DESC]; -그룹함수 : avg, sum, min, max, count -그룹함수와 group by절 사용 시 규칙(문법) => select절의 컬럼리스트들 중 그룹함수에 포함된 컬럼과 그룹함수에 포함되지 않은 컬럼이 같이 출력되려면 그룹함수에 포함되지 않은 컬럼은 반드시 group by절에 포함되어 있어야 함!!!

5. JOIN	[문법] select a.컬럼1, b.컬럼2 from 테이블1 a join 테이블2 b on a.컬럼명 = b.컬럼명; <조인유형> -natural join -using절 join -on절 join -self join -equi join vs non-equi join -inner join vs outer join => left, right, full outer join -cross join
6. Subquery	[문법] select 컬럼1, 컬럼2, 컬럼3 from 테이블명 where 컬럼명 = (select 컬럼1 from 테이블명 where 조건문); -subquery가 포함될 수 있는 곳 => select절, from절(Inlineview), where절, having절, order by절, insert구문, update구문, delete구문, create구문 등... [Subquery 유형] -단일컬럼 서브쿼리 vs 다중컬럼 서브쿼리 : 비쌍 비교 : 쌍 비교 -단일행 서브쿼리 vs 다중행 서브쿼리 : 단일행 비교연산자 : 다중행 비교연산자 =, >, >=, <, <=, <> IN, NOT IN, ANY, ALL
7. 데이터조작어(DML)	[DML 문법] insert into 테이블명(컬럼1, 컬럼2, 컬럼3, ...) values (값1, 값2, 값3, ...); update 테이블명 set 컬럼명 = 값 where 조건문; delete from 테이블명 where 조건문; -트랜잭션 : 하나의 논리적인 작업 단위 여러 개의 DML이 모여서 하나의 트랜잭션 구성 하나의 DDL, 하나의 DCL이 하나의 트랜잭션 구성 -트랜잭션 제어 명령어 : commit, rollback, savepoint
8. Table(DDL)	[테이블 생성] - create table ---; - 테이블명, 컬럼명, 데이터타입, 컬럼사이즈 설정 - 옵션 : default, 제약조건 - 제약조건 : PK, FK, UK, CK, NN

	[테이블 수정]
	- alter table add ---; //컬럼 추가, 제약조건 추가(테이블 레벨 문법)
	- alter table modify ---; //컬럼 수정, 제약조건 추가(컬럼 레벨 문법)
	- alter table drop ---; //컬럼 삭제, 제약조건 삭제
	
	[테이블 삭제]
	- drop table 테이블명 [purge];
	- recyclebin(휴지통)에서 삭제된 테이블 조회 가능
	SQL> show recyclebin
	- 삭제된 테이블 되살리는 방법
	SQL> flashback table 테이블명 to before drop;
	
	[테이블 절단]
	- truncate table 테이블명; (=) delete from 테이블명;
	- 테이블의 모든 행을 삭제하는 명령어

9. View, Sequence, Index, Synonym(DDL)	[View] -하나 이상의 Base table을 기반으로 생성은 되었으나 물리적으로 존재하지 않고 Data Dictionary에 select 구문 형태로 정의만 되어 있는 가상의 논리적인 테이블 -사용 이유 : 공간적인 효율성, 보안성, 편의성, 다양성 -create [or replace] view 뷰명 as subquery; -drop view 뷰명; [Sequence] -자동으로 고유한 번호를 반환해 주는 번호 생성기 -사용 이유 : PK, UK 제약조건이 정의된 컬럼에 이용 -create sequence 시퀀스명 increment by N //시퀀스 번호 간격, default = 1 start with N //시퀀스 번호의 시작값, default = 1 maxvalue N | nomaxvalue //시퀀스 번호의 최대값 minvalue N | nominvalue //시퀀스 번호의 최소값 cycle | nocycle //시퀀스 순환 여부 cache N | nocache //default => cache 20 -alter sequence ---; => start with 옵션 수정 X -drop sequence ---; -사용방법 : 시퀀스명.nextval, 시퀀스명.currval [Index] -테이블 내 행의 검색 속도를 높이기 위해서 사용하는 B-tree 구조의 객체로 생성되어 있다면 오라클이 자동으로 유지/관리 및 사용함 -사용 이유 : 테이블의 데이터 검색 속도를 높이기 위함(성능) -자동 생성 : PK, UK 제약조건이 정의된 컬럼에 자동 생성되어 있음 -수동 생성 : PK, UK 제약조건이 정의되지 않은 컬럼 중 자주 사용되는 컬럼에 수동 생성함 SQL> create index 인덱스명 on 테이블명(컬럼명); -alter index, drop index [Synonym] -객체들에게 또 다른 이름(별명, 동의어)을 부여함 -create synonym, drop synonym

10. Data Dictionary 사용	[Data Dictionary 구성요소] -Base tables : DB의 모든 내용을 Oracle이 보기 쉬운 형태로 기록해 놓은 테이블 -Data Dictionary views : Base table을 사용자가 보기 쉬운 형태로 만들어 놓은 뷰, 읽기전용(select) user_xxx, all_xxx, dba_xxx, v$xxx -사용방법 SQL> desc 딕셔너리명 SQL> select ------ from 딕셔너리명 where 조건문 ;