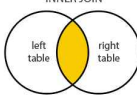
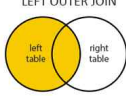
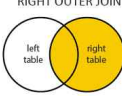


# SQL 정리 표 1차

SQL		설명	예	문법
DDL	데이터베이스	현재 DB server 의 모든 데이터베이스 검색하기	show databases;	
		데이터베이스 생성 하기	create database sqldb;	create database 생성할DB명;
		데이터베이스 가 (로컬)저장되는 위치 검색하기	show variables like 'datadir';	
		데이터베이스 삭제 하기	drop database sqldb;	drop database 삭제할DB명;
		- 만약에 해당 데이터베이스가 존재하면 삭제	drop database if exists sqldb;	drop database if exists 삭제할DB명;
		사용할 데이터베이스 활성화/선택 하기	use sqldb;	use 사용할DB명;
	테이블	테이블 생성 하기	create table table1( 데이터필드1 int , 데이터필드2 varchar(30) );	create table 테이블명( 필드명 타입 제약조건 , 필드명 타입 제약조건 , 관계제약조건 );
		테이블 삭제 하기	drop table table1;	drop table 테이블명;
		- 만약에 해당 테이블이 존재하면 삭제	drop table if exists table1;	drop table if exists 테이블명;
		테이블의 데이터 모두 삭제 하기 * 추후에 취소 불가능	truncate table table1;	truncate table 테이블명;

DML	삽입	모든 필드에 값을 대입하는 레코드 삽입	insert into table1 values( 1 , '유재석' );	insert into 테이블명 values( 값1 , 값2 );
		특정 필드에 값을 대입하는 레코드 삽입	insert into table1( 데이터필드1 ) values( 5 );	insert into 테이블명( 필드명 , 필드명 ) values( 값1 , 값2 );
	검색	모든 필드(*와일드카드) 의 레코드 검색	select * from table1;	select * from 테이블명;
		특정 필드를 표시한 레코드 검색	select 데이터필드2 from table1;	select 필드명1 , 필드명2 from 테이블명;
		특정 필드 값이 조건에 충족하는 레코드만 검색	select * from table1 where 데이터필드1 = 5;	select 필드명1 , 필드명2 from 테이블명 where 조건식
	수정	모든 레코드의 필드 값 변경	update table1 set 데이터필드2 = '강호동';	update 테이블명 set 수정할필드명 = 수정할값 , 수정할필드명 = 수정할값;
		특정 필드 값이 조건에 충족하는 레코드만 특정 필드 값으로 수정	update table1 set 데이터필드2 = '강호동' where 데이터필드1 = 5;	update 테이블명 set 수정할필드명 = 새로운값 , 수정할필드명 = 새로운값 where 조건식;
	삭제	모든 레코드 삭제	delete from table1 ;	delete from 테이블명;
		특정 필드 값이 조건에 충족하는 레코드만 삭제	delete from table1 where 데이터필드1 = 5 ;	delete from 테이블명 where 조건식;

제약조건	조건	레코드 삽입시 필드내 null 포함하지 않는다.	회원번호2 int not null	not null
		레코드 삽입시 필드내 중복된 값을 저장할 수 없다.	회원번호2 int unique	unique
		레코드 삽입시 데이터 생략시 해당 기본값으로 설정하기.	날짜 datetime default now()	default 기본값
		( 기본키/PK 일때만 사용가능 ) 레코드 삽입시 해당 필드의 자동 번호 부여하기	bno int auto_increment	auto_increment
		PK 레코드를 고유하게 구분하기 위한 최소의 정보	constraint board_bno_pk primary key ( bno )	constraint 제약조건명 primary key( pk필드명 )
		FK 다른 테이블의 기본키 정보 와 연결/참조된 정보	constraint board_mno_fk foreign key (mno) references member( mno )	constraint 제약조건명 foreign key( fk필드명 ) references 참조할테이블( 참조할pk필드명 )
	옵션	PK데이터를 삭제/수정할때 PK데이터를 참조하는 FK데이터가 있으면 삭제/수정 취소 처리됨	constraint board_mno_fk foreign key (mno) references member( mno ) on delete restrict on update restrict	on delete restrict , on update restrict
		PK데이터를 삭제/수정할때 PK데이터를 참조하는 FK데이터가 있으면 같이 삭제/수정 처리됨	constraint board_mno_fk foreign key (mno) references member( mno ) on delete cascade on update cascade	on delete cascade , on update cascade
		PK데이터를 삭제/수정할때 PK데이터를 참조하는 FK데이터가 있으면 PK데이터삭제/수정 하고 FK데이터는 NULL 변경됨	constraint board_mno_fk foreign key (mno) references member( mno ) on delete set null on update set null	on delete set null , on update set null

조인	내부조인	- 두개 이상의 테이블을 이용하여 검색할때 사용됩니다.	select * from member m , board b where m.mno = b.mno;	select * from 테이블명A , 테이블명B where 테이블명A.pk필드명 = 테이블명B.fk필드명;
		<div>INNER JOIN</div>  <div>LEFT OUTER JOIN</div>  <div>RIGHT OUTER JOIN</div> 	select * from member natural join board;	select * from 테이블명A natural join 테이블명B;
			select * from member join board on member.mno = board.mno;	select * from 테이블명A join 테이블명B on 테이블명A.pk필드명 = 테이블명B.fk필드명;
			select * from member join board using(mno);	select * from 테이블명A join 테이블명B using( 필드명 );
			select * from member inner join board on member.mno = board.mno ;	select * from 테이블명A inner join 테이블명B on 테이블명A.pk필드명 = 테이블명B.fk필드명;
			select * from member right outer join board on member.mno = board.mno	select * from 테이블명A RIGHT OUTER JOIN 테이블명B on 테이블명A.pk필드명 = 테이블명B.fk필드명;
	외부조인		select * from member left outer join board on member.mno = board.mno ;	select * from 테이블명A LEFT OUTER JOIN 테이블명B on 테이블명A.pk필드명 = 테이블명B.fk필드명;

SQL 정리 표 2차

SQL		설명 및 문법	예	문법
연산자	산술연산자	+더하기 -빼기 *곱하기 /나누기 div몫 mod나머지	select * from member where mname = '불액핑크'; # 1.필드의 값(문자이면 ' )이 일치한 레코드 검색 select * from member where mnumber = 4; # 2.필드의 값(숫자이면 )이 일치한 레코드 검색 select * from member where mheight <= 162; # 3.필드의 값이 이하 이면 레코드 검색 select * from member where mheight >= 165 and mheight <=170; # 4.필드의 값이 이상 이면서 이하 이면 레코드 검색 select * from member where mheight >= 165 or mnumber > 6; # 5.필드의 값이 이상 이거나 초과 이면 레코드 검색 select * from member where maddr = '경기' or maddr = '전남' or maddr = '경남'; select * from member where maddr in( '경기','전남','경남'); select * from member where not maddr in( '경기','전남','경남'); select * from member where mname = '에이핑크'; select * from member where not mname = '에이핑크'; # 6-1. '에이핑크' 가 아니면 select * from member where mname != '에이핑크'; # 6-2. '에이핑크' 가 아닌 select * from member where mname like '에이%'; # 7. '에이' 로 시작하는 문자 select * from member where mname like '%핑크'; # 8. '핑크' 로 끝나는 문자 select * from member where mname like '에이.'; # 9. '에이' 로 시작하는 세글자 select * from member where mname like '.핑크'; # 10. '핑크' 로 끝나는 세글자 select * from member where mname like '%우%'; # 11. '우' 가 포함된 문자 select * from member where mname like '_우_'; # 12. '우' 가 두번째에 위치한 세글자 select mnumber+10, mnumber-10, mnumber * 10, mnumber/10, mnumber div 3, mnumber mod 3, mnumber * mheight from member;	SELECT 산술연산자 테이블명;
	비교연산자	>초과 <미만 >=이상 <=이하 =같다 !=같지않다.		SELECT * FROM 테이블명 WHERE 필드명 비교연산자 값;
	논리연산자	and이면서 or이거나 not부정		SELECT * FROM 테이블명 WHERE 비교연산 AND/OR 비교연산 SELECT * FROM 테이블명 WHERE NOT 논리값(TRUE/FALSE)
	기타연산자	[ 시작값부터 끝값 까지 포함 여부 ] between 시작값 and 끝값		SELECT * FROM 테이블명 WHERE 필드명 BETWEEN 시작값 AND 끝값
		[ 하나라도 포함 여부 ] in( 값, 값, 값 )		SELECT * FROM 테이블명 WHERE 필드명 IN( 값1, 값2, 값3 )
		[ 패턴 검색 ] 필드명 like 패턴 ( % : 모든 문자수 대응, _ : 1개수만큼 문자수 대응 )		SELECT * FROM 테이블명 WHERE 필드명 LIKE '%값%' SELECT * FROM 테이블명 WHERE 필드명 LIKE '_값_'
		[ 필드 NULL 여부 ] 필드명 is null, 필드명 is not null		SELECT * FROM 테이블명 WHERE 필드명 IS NULL SELECT * FROM 테이블명 WHERE 필드명 IS NOT NULL
		[필드 또는 테이블 에 별칭/별명 ] as		SELECT 필드명 AS 별칭 FROM 테이블명 별칭;
		[중복제거] distinct		SELECT DISTINCT 필드명 FROM 테이블명;
검색 키워드	where	특정 조건에 부합하는 데이터들만 조회	select * from buy where bamount >= 3;	SELECT * FROM 테이블명 WHERE 조건절
	group by having	같은 값을 가진 행을 그룹	select mid from buy group by mid; # mid 필드 그룹 select distinct mid from buy; # mid 필드 중복 제거	SELECT 그룹할필드명 FROM 테이블명 GROUP BY 그룹할필드명
		같은 값을 가진 행을 그룹 으로 집계 함수를 사용	select mid as 구매자, sum( bamount ) as 총수량 from buy group by mid;	SELECT 그룹할필드명, 집계함수(집계할필드명) FROM 테이블명 GROUP BY 그룹할필드명
	order by	[Having] 같은 값을 가진 행을 그룹 으로 특정 조건에 부합하는 데이터들만 조회	from buy group by mid having sum( bamount * bprice ) >= 1000;	SELECT 그룹할필드명, 집계함수(집계할필드명) FROM 테이블명 GROUP BY 그룹할필드명 HAVING 그룹조건
		조회 된 데이터를 오름차순 정렬 ASC, 생략가능	select * from member order by mdebut asc; # 데뷔일 필드 기준으로 오름차순	SELECT * FROM 테이블명 ORDER BY 정렬할필드명 ASC
	limit	조회 된 데이터를 내림차순 정렬 DESC	select * from member order by mdebut desc; # 데뷔일 필드 기준으로 내림차순 select * from member order by mheight desc, mdebut asc;	SELECT * FROM 테이블명 ORDER BY 정렬할필드명 DESC
		조회 된 데이터를 제한 수 제한하기	select * from member limit 2; # 검색 결과의 레코드를 2개만 표시 select * from member order by mheight desc limit 3; # 키 상위 3명	SELECT * FROM 테이블명 LIMIT 개수
	같이 사용시 작성 순서	조회 된 데이터를 제한 수 범위 제한하기, 0년부터 시작	select * from member limit 0, 3; # 0(첫번째)레코드 부터 3개만 표시	SELECT * FROM 테이블명 LIMIT 시작번호, 개수
		SELECT * FROM 테이블명 WHERE 조건절 GROUP BY 그룹필드 HAVING 그룹조건 ORDER BY 정렬필드 정렬기준 LIMIT 시작번호, 개수		
집계 함수	sum	필드의 총합계	select sum( bamount ) from buy;	SELECT SUM(필드명) FROM 테이블명;
	avg	필드의 총평균	select avg( bamount ) from buy;	SELECT AVG(필드명) FROM 테이블명;
	max , min	필드의 최댓값	select max( bamount ) from buy;	SELECT MAX(필드명) FROM 테이블명;
		필드의 최솟값	select min( bamount ) from buy;	SELECT MIN(필드명) FROM 테이블명;
	count	필드의 레코드수 ( 필드가 NULL 일때 제외 )	select count( bname ) from buy;	SELECT COUNT(필드명) FROM 테이블명;
		전체 레코드수 ( 필드가 NULL 포함 일때 포함 )	select count( * ) from buy;	SELECT COUNT(*) FROM 테이블명;