



SQLD 중요부분 정리

SQL 명령문 개괄

From-where-group by-having-select-orderby

DDL-select, insert, delete, update

DDL-alter, create, modify, drop

TCL-rollback, commit

DCL-grant, revoke

SELECT

distinct(집약) → 1010202030 → 102030

AS -select - 1. as 생략가능 2. 컬럼명에 띄어쓰기 ("직원 번호") , from - as 사용불가

concat(인수가2개임 무조건) - + SQL server , || oracle

논리연산자

and(A and B) , or (A or B), not (A,B 둘다 아님)

연산순위 1.NOT(NOT <조건> 2.AND(AND<조건>) 3.OR(OR<조건>) = NAO

ex) 조건 and 조건 and 조건 or 조건

SQL 연산자

A Between = 1 and 2 1 사이 2

A IN (1,2,3) = A=1 , or A=2, or A=3

중요! LIKE _ - 미지의 한글자, % - 0 이상 글자, escape 와일드카드인(_ %) 를 문자로 취급하는 함수 - `ename like 'A_A' ⇒ 'A@_A ⇒ '@'` 예시 - 도둑 = @(아무 문자 가능)
 , 흠칠물건 = _

Rownum (oracle) → 1, where 조건절에서 rownum = 1 포함인 경우

Rownum 이 3 인경우 order by sal 가장 마지막 실행 → 1900 1500 1300

TOP (SQL server) → select TOP(n) 상위n개 n에는 칼럼명 삽입

ex) _L% = 이름의 2번째 글자가 L인 사원들 다 고르기 인 문제

NULL

1. null의 정의 - 부재, 모르는값
2. $\text{null} + 2 = ??$, $\text{null} - 4 =$, $\text{null} * \text{null} \Rightarrow \text{null}$ (산술연산)
3. where(조건) = False = 비교연산
4. 정렬상의 의미 - oracle : 무한대 , SQL sever : - 무한대
5. NVL, NVL2, isnull, nullif, coalsece = NVL, NVL2, isnull 은 널뛰기로 이해 nullif은 같이놀자! coalsece은 널 아닌 첫번째값 으로 이해하기
6. 값1 is null = 값2 아니라면 값1
7. 값 is null = 값3
8. 값 is not null =값2
9. 같이놀자 = 같으면 null , 다르면 값1

정렬

1. 정렬의 특성 - 가장 마지막에 실행, 성능이 느려질 가능성, null 값과의 관계
2. 컬럼 번호정렬 - 출력되는 칼럼의 수보다 큰 값이 불허함
3. 인수 두개의 정렬 - sal desc , ename asc - sal 이 같으면 ename 오름차순
4. 예시 - select ename orderby sal 은 가능하다

숫자함수

Round 자릿수 확인

예시 - Round() 인수!

ceil - oracle

celing - sql server

문자열 함수

upper

lower

LPad

Rpad

Ltrim

Rtrim

substr

nstr

날짜함수

TO_Char

TO_date 이 두가지 실습하기 예시 - 다음중 데이터에 형변환을 일으키는 함수

Sysdate(oracle)

getdate(sql server)

날짜데이터 + 100 - 100일 이후(day)로 인식

DECODE/CASE

case만!

case when then

case when then

else 없을 경우 조건1, 조건2 만족하지 않는경우 null 출력

end

집계함수 - null과의 관계

→ null과의 관계

	A	B	C
1	null	null	1
2	3	2	2
3	null	2	3

$\text{Sum}(A) = 3$
 $\text{Sum}(B) = 4$
 $\text{Count}(A) = 1$
 $\text{Count}(B) = 3$

sum(A+B+C) 이 나오는 경우 null , 7 , null 이라고 생각하고 풀어라

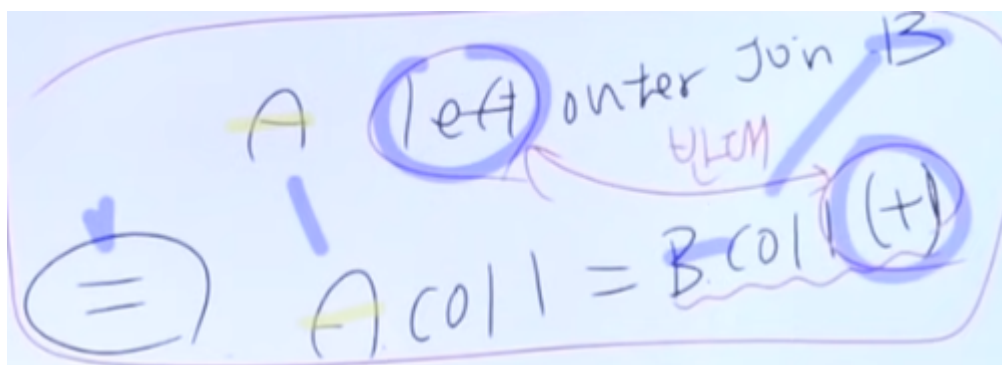
GROUP BY

집약기능이 있음

그룹수준으로 정보를 바꿈

JOIN

1. natural join , using - 중복된 컬럼 “하나” 출력 , 제일 앞에 등장 , alias 사용 안됨



A 는 선행조건자 left가 +로 바뀜

join 하면 할수록 똥똥해짐

From A,B 조인 다음 C 새로 조인 - 순서 기억하기

서브쿼리

select - Scalar

from - Inline view (메인쿼리에 칼럼 사용가능)

where - 거의 모든 서브쿼리가 들어감(중첩서브쿼리)

groupby X

having - 거의 모든 서브쿼리 들어감(중첩서브쿼리)

orderby - Scalar

In

any/some

all

exist (l , x , a \Rightarrow True) , (0 rows \Rightarrow False)

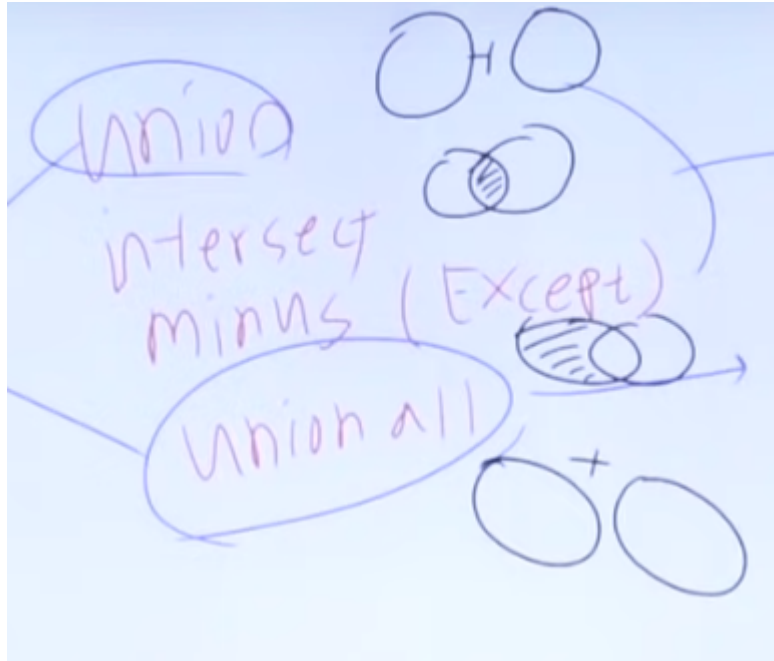
집합연산자

union - 정렬작업 , 느리다

intersect - 정렬작업 , 느리다

minus (Except) - 정렬작업 , 느리다

union all \rightarrow 중복데이터 O , 정렬작업 x , 속도가 빠름



union 은 중복데이터 사라짐 union all 은 중복데이터 남아있음

DDL

TCL 과 연관짓기

Truncate(입주민퇴거(구조남음)) vs Drop(철거(구조삭제))

Truncate vs delete

DDL vs DML - rollback ,commit 연관되서 문제 출제

DML

insert - insert values 오류 상황 문제 출제될수도잇음(가끔안나옴)

update

delete

rollback ,commit 연관되서 문제 출제\

Merge

제약조건

Plc - unique + notnull (하나만 존재한다)

unique

not null

DCL

grant, revoke 정의 문제

role 특징 5가지 , 명령어는 아님 object임

VIEW (SQL 명령문 저장)

독편보

1. 독립성

2. 편리성

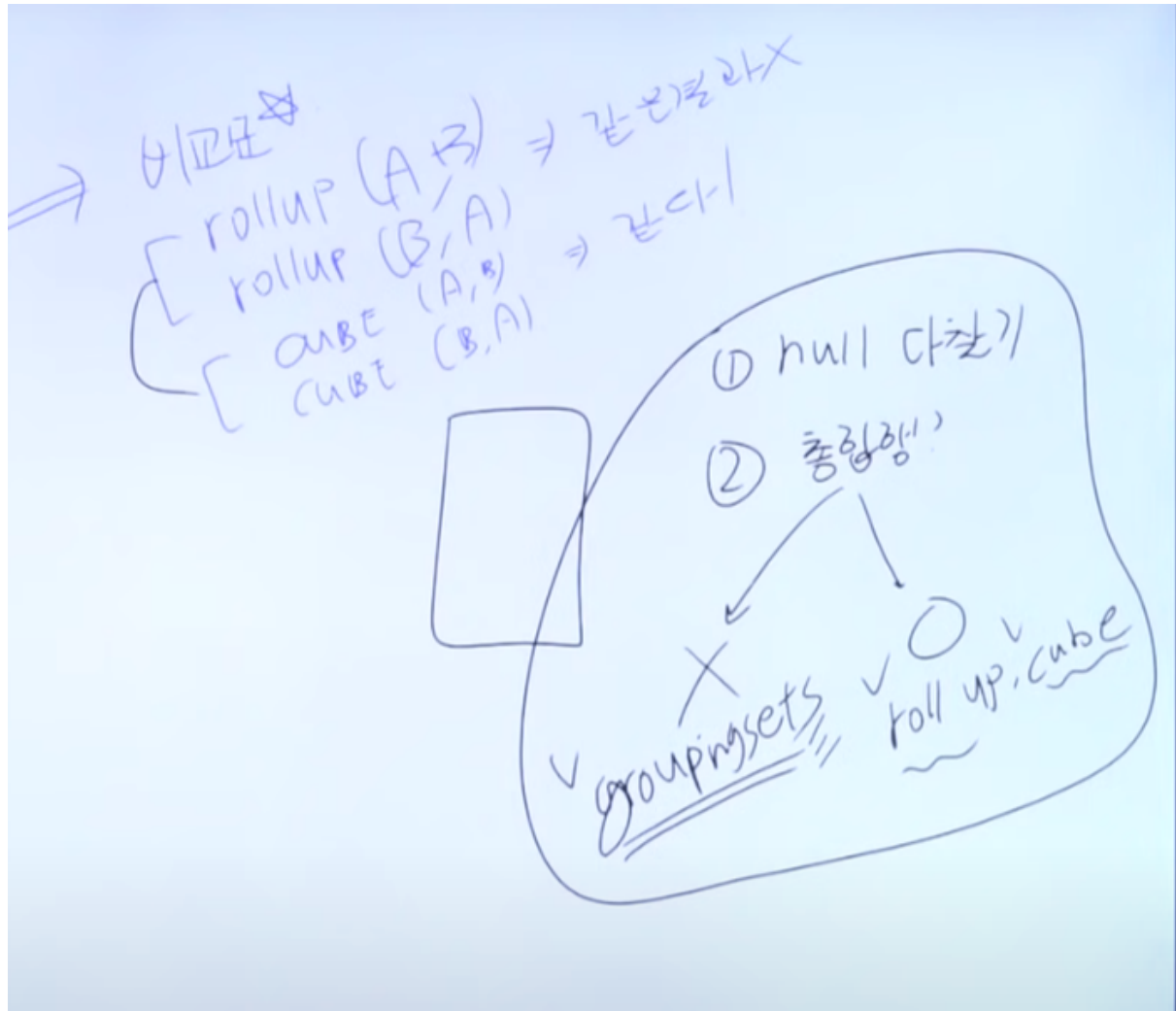
3. 보안성

그룹함수

roll up

cube

groupingsets



위 3개 결과값 주고 어떤걸 씻는지 나오는 문제 예상

grouping (자주 안나오는 추세)

TCL

commit - auto commit off and Begin transaction (DDL 커밋기능 없앴)

rollback

윈도우 함수(무조건 문제풀기)

Rows

Rank

차이점 중요 / 같은값 있는지 판별하기

Rank - 중복 건너뛰기 ex) 1, 1, 3, 4

Dense_Rank - 건너뛰기 없음 ex) 1, 1, 2, 3

계층형 질의

Prior 자식데이터 = 부모데이터 - 프 자 부

부모에서 자식으로 가면 순방향 - 부 자 순

실습 문제 예시

level1 - king (empno)

level2 - james (mgr)

level3 - scott

절차형 PL/SQL

Exception ⇒ 생략 가능

procednre , 트리거 , userdefinend 차이점

데이터모델링

1절 데이터 모델의 이해

1. 모델링 : 현실세계를 단순화하여 표현하는 기법

■ 특징

- 단순화, 추상화, 명확화

■ 관점

- 데이터관점 : 업무와 데이터, 데이터 사이의 관계
- 프로세스관점 : 진행되고 있거나 진행되어야 하는 업무
- 상관관점 : 데이터에 대한 업무 처리 방식의 영향

2. 데이터 모델링 : 정보시스템 구축을 위한 데이터 관점의 업무 분석 기법

■ 중요성 및 유의점

- 중복 : 같은 시간 같은 데이터 제공
- 비유연성 : 사소한 업무변화에 데이터 모델이 수시로 변경되면 안됨
- 비밀관성 : 신용상태에 대한 갱신 없이 고객의 납부 이력 정보 갱신 안됨

■ 데이터 모델링 3단계

- 개념적 모델링 : 엔터티, 속성 도출. ERD 작성. 업무중심, 포괄적 수준의 모델링
- 논리적 모델링 : 식별자 도출. 속성, 관계 정의. 정규화 수행
- 물리적 모델링 : DB구축. 성능 및 보안 등 물리적인 성격 고려

계획 > 분석 > 설계 > 개발 > 테스트 > 전환/이행

개념 (논리) 물리

■ 데이터 독립성 요소

- 외부 스키마 : 개개 사용자가 보는 개인적 DB 스키마
- 개념 스키마 : 모든 사용자 관점을 통합한 전체 DB
- 내부 스키마 : 물리적 장치에서 데이터가 실제적 저장

■ 데이터 독립성

- 필요성 : 데이터의 중복성과 데이터 복잡도 증가로 인한 유지비용 증가, 요구사항 대응 저하
- 논리적 독립성 : 개념 스키마가 변경되어도 외부 스키마에 영향 X
- 물리적 독립성 : 내부 스키마가 변경되어도 외부/개념 스키마는 영향 X

엔터티

1. 엔터티의 특징

2. 특징

- 반드시 해당 업무에서 필요하고 관리하고자 함
- 유일한 식별자에 의해 식별 가능
- 두개 이상의 인스턴스의 집합
- 업무 프로세스에 의해 이용되어야 함
- 반드시 속성이 있어야 함
- 다른 엔터티와 최소 1개 이상의 관계가 존재해야 함

2. 분류

3. 분류

■ 유무형에 따른 분류

- 유형 : 물리적 형태 (ex. 사원, 물품, 강사)
- 개념 : 개념적 정보 (ex. 조직, 보험상품)
- 사건 : 업무 수행시 발생 (ex. 주문, 청구, 미납)

■ 발생시점에 따른 분류

- 기본 : 원래 존재하는 정보 (ex. 사원, 부서)
- 중심 : 기본 엔터티로부터 발생 (ex. 계약, 사고, 주문)
- 행위 : 2개 이상의 부모 엔터티로부터 발생 (ex. 주문목록, 사원변경이력)

