데이터 독립성이 보장된 최초의 데이터 모델이다.

물리(Physical) = Ordering, Indexing, Access Path 종속적

논리(Logical) = 데이터 항목 자체에 집중

테이블 형태, 중복이 최소화 된 데이터 모델이다.

데이터 모델을 봤을 때 이해하는 법

DVD Rental Model 로 보자

Reference Data(사업 환경 설정) : 언어, 카테고리, 나라, 도시 등

Master Data(사업 활동 주체와 대상) : 고객, 상품, 가게, 영화, 고객 주소 등

Transaction Data(사업 활동) : 돈과 관련, 주문 등

재사용률은 위에서 부터 아래 순이다.

**SELECT 문법**

SELECT column FROM table\_name;

모든 데이터를 다 보고 싶다면 column 에 \* 넣으면 된다.

From - >table\_name -> select -> column 순으로 실행된다.

**Alias : 코드의 가독성을 높여준다, SQL 성능을 높여준다**

DBMS -> 옵티마이저(최적화 기능 담당)이 SQL을 읽을 때 실행할 때 가장 빠르고 저비용인 방법을 찾는다.

테이블이 여러개 있을 때 table\_name 뒤에 alias를 붙여주는 걸 의미한다.

SELECT a.column FROM table\_name a;

**ORDER BY 문법 : 데이터를 정렬하는데 사용. 업무 처리상 매우 중요**

SELECT column\_1, column\_2 FROM table\_name ORDER BY column\_1 ASC(오름차순), column\_2 DESC(내림차순)

default값은 ASC이다.

Asc + desc 정렬 : first\_name을 ASC로, last\_name을 DESC로 정렬 가능

ORDER BY 에 정수 넣어도 된다. 정수가 의미하는건 COLUMN을 왼쪽으로 부터 오른쪽으로 읽었을 때 순서이다. 간단하지만 가독성이 좋은 방법은 아니다. (추천X)

ORDER BY COLUMN\_1, COLUMN\_2 : COLUMN\_1을 정렬하고 그걸 기반으로 COLUMN\_2를 정렬한다.

**SELECT DISTINC 문** : **중복 값 제외한 결과값이 출력된다.** 즉, 같은 결과의 행이면 중복을 제거할 수 있다.

SELECT

*//COLUMN\_1 + COLUMN\_2 중복값을 제거한다. 따라서 COLUMN\_1에는 중복값 있을 수 있다.*

DISTINCT COLUMN\_1, COLUMN2

FROM TABLE\_NAME

ORDER BY COLUMN\_1, COLUMN\_2;

참고) create table은 ddl이기에 commit 필요없지만, insert문은 필요하다.

컬럼 한 개 : 한 개의 컬럼의 중복값을 모두 제거한다.

컬럼 두 개 이상 : 모든 컬럼을 합친 ROW 값의 중복값을 제거한다.

**SELECT DISTINCT ON (COLUMN1) COLUMN\_1, COLUMN\_2 : COLUMN\_1 값 기준 중복 제거. COLUMN\_2값은 단 한 개 값 만을 보여줌**

****

****

**Bcolor의 모든 중복값들을 제거하고, 오른편의 fcolor column에는 오름차순 맨 위에 있는 값 단 한 개 만을 둔다.**

**SELECT DISTINCT ON (COLUMN1) COLUMN\_1, COLUMN\_2 … ORDER BY COLUMN\_1, COLUMN\_2 DESC;**

**: FCOLOR 컬럼 값 보여줄 때 내림차순 정렬함.**

****

**WHERE 절 : 집합을 가져올 때 어떤 집합을 가져올 것인가에 대한 조건을 설정하는 절이다.**

SELECT

COLUMN\_1, COLUMN\_2

FROM

TABLE\_NAME

WHERE

<조건> //문자열을 조건으로 작성 시 작은 따옴표 써야함!

;

연산자



조건이 두 개 이상일 때는 AND를 붙인다.

참고) CTRL + SHIFT + F 누르면 정렬된다.