

CSE231 데이터베이스 설계 (Database Design) Practice 03: SQL #2

담당교수: 전강욱(컴퓨터공학부)
kw.chon@koreatech.ac.kr

Sample DB 구축

■ Sample DB 구축

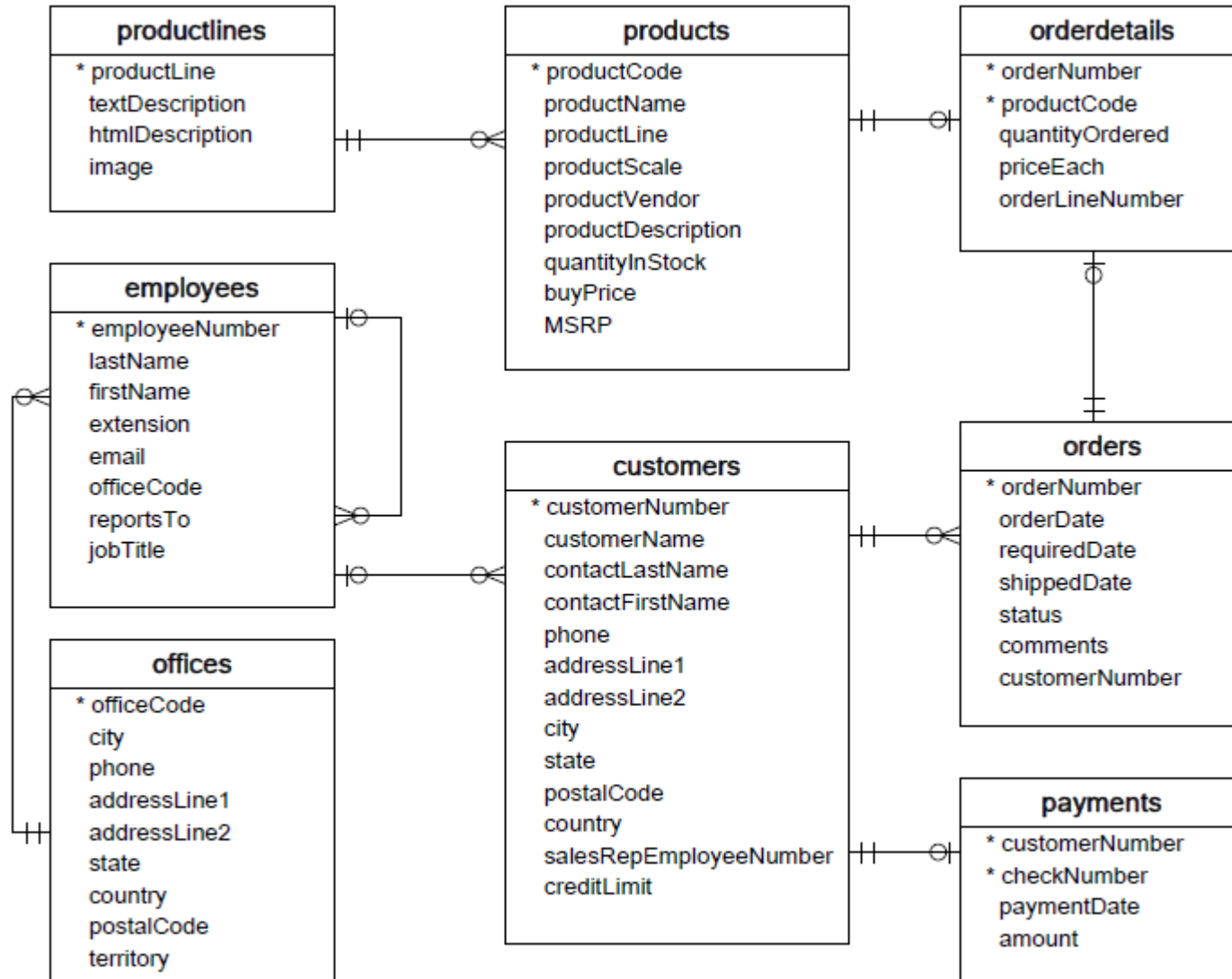
□ `mysqlsampledatabase.sql` 파일 다운로드

■ 출처: <https://github.com/hhorak/mysql-sample-db>

□ 데이터 상세

- Customers: stores customer's data
- Products: stores a list of scale model cars
- ProductLines: stores a list of product line categories
- Orders: stores sales orders placed by customers
- OrderDetails: stores sales order line items for each sales order
- Payments: stores payments made by customers based on their accounts
- Employees: stores all employee information as well as the organization structure such as who reports to whom
- Offices: stores sales office data.

Sample DB 구축 (계속)



출처: <https://www.mysqltutorial.org/mysql-sample-database.aspx>.

Sample DB 구축 (계속)

- Command Line Client 실행
- Password 입력 후 하기 명령어 수행
 - source \$PATH/mysqlsampledatabase.sql
 - \$PATH: mysqlsampledatabase.sql 파일을 포함하는 디렉토리

UNION 연산자

- 여러 개의 테이블을 합치는 연산자 (a.k.a., 합집합)
- 사용법
 - **SELECT** column1, column2 **FROM** table명 **UNION**
SELECT column1, column2 **FROM** table명;
 - UNION 연산자는 기본적으로 중복제거 기능이 있으며, 중복된 결과를 보고싶을 경우 UNION ALL 사용

GROUP BY절

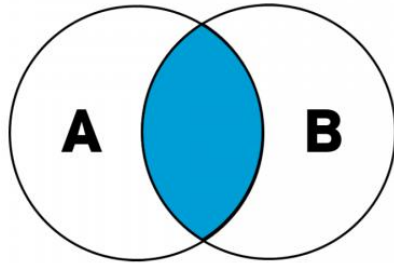
- 동일한 행들을 묶어서 그룹을 만드는데 사용
- GROUP BY는 Aggregate 함수(AVG, COUNT, SUM, MAX, MIN)와 함께 사용
 - i.e., 어떤 그룹의 평균값, 최대값 등
- 사용법
 - **SELECT** column_1, column_2,...,column_n
AGGREGATE_FUNCTION(expression)
FROM table
WHERE where_condition
GROUP BY column_1, column_2,...,column_n;
 - e.g., **SELECT** orders.status, **count**(orderNumber) **AS** count
FROM orders
GROUP BY orders.status
ORDER BY count desc;

HAVING 절

- **SELECT절의 WHERE처럼 조건을 제약하여 Row를 추출하기 위한 절**
 - 보통 GROUP BY와 함께 활용
 - WHERE절과의 차이점은 SELECT뒤에 오는 Aggregate Function이 적용된 column에 대하여 제약을 둘 수 있음
- **사용법**
 - **SELECT** column_1, column_2,...,column_n
FROM Table
GROUP BY column_1
HAVING column_2 > 1000;

INNER JOIN절

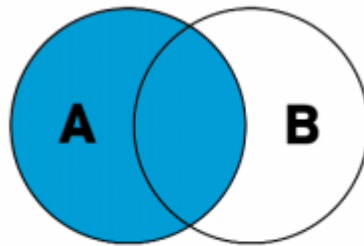
- Join절: 둘 이상의 테이블을 연결해서 데이터를 검색하는 방법
 - 연결하려는 테이블들이 적어도 하나의 column을 공유하는 것이 필요
- 여러 개의 테이블에서 교집합을 만족하는 Row들을 추출하는 절



- 사용법
 - **SELECT** column_1, column_2,...,column_n
FROM table_1
INNER JOIN table_2 **ON** join_condition_1
INNER JOIN table_3 **ON** join_condition_2
...
WHERE conditions;

LEFT JOIN절

- Join절: 둘 이상의 테이블을 연결해서 데이터를 검색하는 방법
 - 연결하려는 테이블들이 적어도 하나의 column을 공유하는 것이 필요
- 여러 개의 테이블에서 아래 그림을 만족하는 Row들을 추출하는 절



LEFT OUTER JOIN

- 사용법
 - **SELECT** column_1, column_2,...,column_n
FROM table_1 A
LEFT JOIN table_2 B **ON** A.column_1 = B.column_1
WHERE conditions;

SELF JOIN절

- **Join절: 둘 이상의 테이블을 연결해서 데이터를 검색하는 방법**
 - 연결하려는 테이블들이 적어도 하나의 column을 공유하는 것이 필요
- **조인되어야 할 데이터들이 동일 테이블에 있을 경우 사용**
- **사용법**
 - **SELECT** column_1, column_2,...,column_n
FROM table_1 A1
LEFT JOIN table_1 A2 **ON** A.column_1 = A2.column_1
WHERE conditions;

JOIN 보충

S	A	B1
	1	가
	2	나
	3	다

T	B2	C
	나	#
	다	%
	라	\$
	마	@

조인

$$V \leftarrow S \bowtie_{B1=B2} T$$

V			
A	B1	B2	C
2	나	나	#
3	다	다	%

왼쪽 외부조인

$$V \leftarrow S \ltimes_{B1=B2} T$$

V			
A	B1	B2	C
1	가	null	null
2	나	나	#
3	다	다	%

오른쪽 외부조인

$$V \leftarrow S \bowtie_{B1=B2} T$$

V			
A	B1	B2	C
2	나	나	#
3	다	다	%
null null		라	\$
null null		마	@

양쪽 외부조인

$$V \leftarrow S \ltimes_{B1=B2} T$$

V			
A	B1	B2	C
1	가	null	null
2	나	나	#
3	다	다	%
null	null	라	\$
null	null	마	@

감사합니다!

숙제

- 아래 query에 대해서 SQL문을 작성하시고, 출력 화면을 캡처하여 제출
 - 고객테이블에서 고객번호, 고객 Lastname정보를 얻은 결과와 직원테이블에서 직원번호, 직원 Lastname정보를 합치고 번호에 대하여 오름차순으로 정렬한 결과를 보이시오
 - 주문의 상태와 그 상태를 갖는 주문의 수를 출력하여라
 - 상태를 갖는 주문의 수가 많은 순으로 출력
 - 상세주문 테이블에서 주문번호별 총 주문수, 총 개당 가격을 추출하시오
 - 총주문수는 1000건 이상이고 총 개당 가격은 600이상인 ROW만 추출
 - INNER JOIN을 활용하여 priceEach가 가장 작은 물품의 이름과 priceEach를 검색하시오
 - 주문 테이블과 상세 주문 테이블로부터 주문번호 별로 주문번호, 상태, 총 주문금액을 검색하시오
 - 하나의 주문번호에는 여러 개의 상품주문이 있기 때문에 총 주문금액은 quantityOrdered*priceEach들의 Sum이 되어야 함
 - 모든 고객의 주문 정보를 검색하시오(추출해야 할 정보는 고객번호, 고객이름, 주문번호, 상품상태)
 - Q: **SELECT** c.customerNumber, c.customerName, orderNumber, o.status **FROM** customers c **LEFT JOIN** orders o **ON** c.customerNumber=o.customerNumber;
 - Q의 결과에서 NULL값이 포함되었다면, 이유를 설명하시오
 - 고객 중 주문을 하지 않은 고객의 고객번호와 고객이름을 추출하시오