

데이터베이스와 SQL

데이터베이스란?

여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합을 말한다. 자료항목의 중복을 없애고 자료를 구조화하여 저장함으로써 자료 검색과 갱신의 효율을 높인다. 현대적인 의미의 데이터베이스 개념을 확립한 사람은 당시 제너럴일렉트릭사(社)에 있던 C.바크만으로 그는 1963년 IDS(Integrated Data Store)라는 데이터베이스 관리시스템을 만들었다.

[네이버 지식백과] 데이터베이스 [data base] (두산백과)

관계형 데이터베이스

관계형 데이터베이스 시스템은 가장 많이 쓰이고 가장 대표적인 데이터베이스 시스템 (Mysql, Oracle)
2차원 구조의 모델을 가지고 SQL이라는 언어를 이용 데이터에 접근한다.

이름	전화번호	구입일	상품	수량
홍길동	010-3242-5931	2019-01-02	우유	1
홍길동	010-3242-5931	2019-01-03	식빵	2
이동진	010-3943-1992	2018-12-21	치즈	1
박철우	010-6123-4453	2018-12-23	소금	1
박철우	010-6123-4453	2018-12-25	우유	3

이름	전화번호
홍길동	010-3242-5931
이동진	010-3943-1992
박철우	010-6123-4453

이름	구입일	상품	수량
홍길동	2019-01-02	우유	1
홍길동	2019-01-03	식빵	2
이동진	2018-12-21	치즈	1
박철우	2018-12-23	소금	1
박철우	2018-12-25	우유	3

테이블의 구조

column, 열



이름	전화번호	구입일	상품	수량
홍길동	010-3242-5931	2019-01-02	우유	1
홍길동	010-3242-5931	2019-01-03	식빵	2
이동진	010-3943-1992	2018-12-21	치즈	1
박철우	010-6123-4453	2018-12-23	소금	1
박철우	010-6123-4453	2018-12-25	우유	3

← row, record, 행

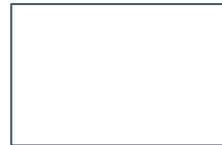
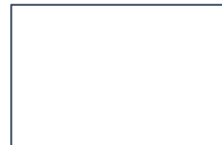
MySQL 구조

데이터베이스(Database)
or 스키마 (Schemas)

데이터베이스 서버
(Database Server)

이름	전화번호
홍길동	010-3242-5931
이동진	010-3943-1992
박철우	010-6123-4453

이름	구입일	상품	수량
홍길동	2019-01-02	우유	1
홍길동	2019-01-03	식빵	2
이동진	2018-12-21	치즈	1
박철우	2018-12-23	소금	1
박철우	2018-12-25	우유	3

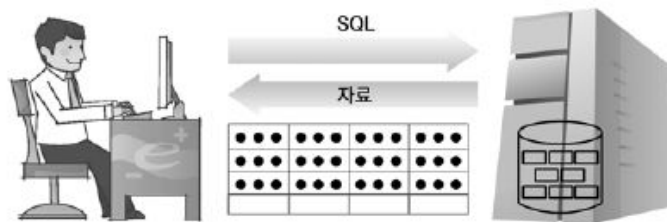


NoSql

NoSQL 데이터베이스는 전통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 이용하는 데이터의 저장 및 검색을 위한 매커니즘을 제공한다. 이러한 접근에 대한 동기에는 디자인의 단순화, 수평적 확장성, 세세한 통제를 포함한다. NoSQL 데이터베이스는 빅데이터와 실시간 웹 애플리케이션의 상업적 이용에 널리 쓰인다. 또, NoSQL 시스템은 SQL 계열 쿼리 언어를 사용할 수 있다는 사실을 강조한다는 면에서 "Not only SQL"로 불리기도 한다.

ex) MongoDB, Hadoop, Elasticsearch

SQL 이란?



[그림 II-1-1] 사용자와 데이터베이스 간의 대화 과정

DBMS와 대화를 위해 필요한 것이 SQL
여러 종류의 DBMS중 RDBMS를 조작할 때 사용함

ANSI SQL

sql 표준화 : 1986년 SQL-86의 명칭(별칭 SQL-87)으로 ANSI에 의한 최초의 표준화가 되었다.

각 나라별 표준협회 : 일본규격협회(JIS) **미국표준협회(ANSI)** 한국표준협회(KS) 유럽(DIN)

여러 데이터베이스들의 SQL들이 제공해주는 특성에 따라 SQL이 다를 수 있다. 하지만 ANSI 표준으로 작성된 SQL은 모든 데이터베이스에서 호환된다.

그러므로 ANSI SQL 작성하게끔 습관을 들이는 것이 좋다.

ANSI SQL

Oracle에서의 쿼리

```
SELECT a.ename  
       ,b.ename  
FROM scott.emp a, scott.emp b  
WHERE a.mgr = b.empno
```

ANSI SQL

```
SELECT a.ename,b.ename  
FROM scott.emp a JOIN scott.emp b  
ON a.mgr = b.empno
```

두 쿼리는 같은 출력을 낸다.

SQL 문장 종류

[표 II-1-1] SQL 문장들의 종류

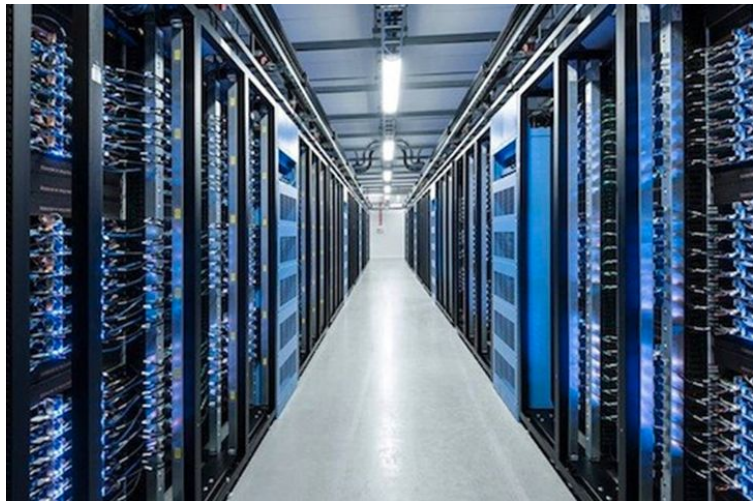
명령어의 종류	명령어	설명
데이터 조작어 (DML: Data Manipulation Language)	SELECT	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어를 말하는 것으로 RETRIEVE 라고도 한다.
	INSERT UPDATE DELETE	데이터베이스의 테이블에 들어 있는 데이터에 변형을 가하는 종류의 명령어들을 말한다. 예를 들어 데이터를 테이블에 새로운 행을 집어넣거나, 원하지 않는 데이터를 삭제하거나 수정하는 것들의 명령어들을 DML이라고 부른다.
데이터 정의어 (DDL: Data Definition Language)	CREATE ALTER DROP RENAME	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어들로 그러한 구조를 생성하거나 변경하거나 삭제하거나 이름을 바꾸는 데이터 구조와 관련된 명령어들을 DDL이라고 부른다.
데이터 제어어 (DCL: Data Control Language)	GRANT REVOKE	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어를 DCL이라고 부른다.
트랜잭션 제어어 (TCL: Transaction Control Language)	COMMIT ROLLBACK	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위(트랜잭션) 별로 제어하는 명령어를 말한다.

실습

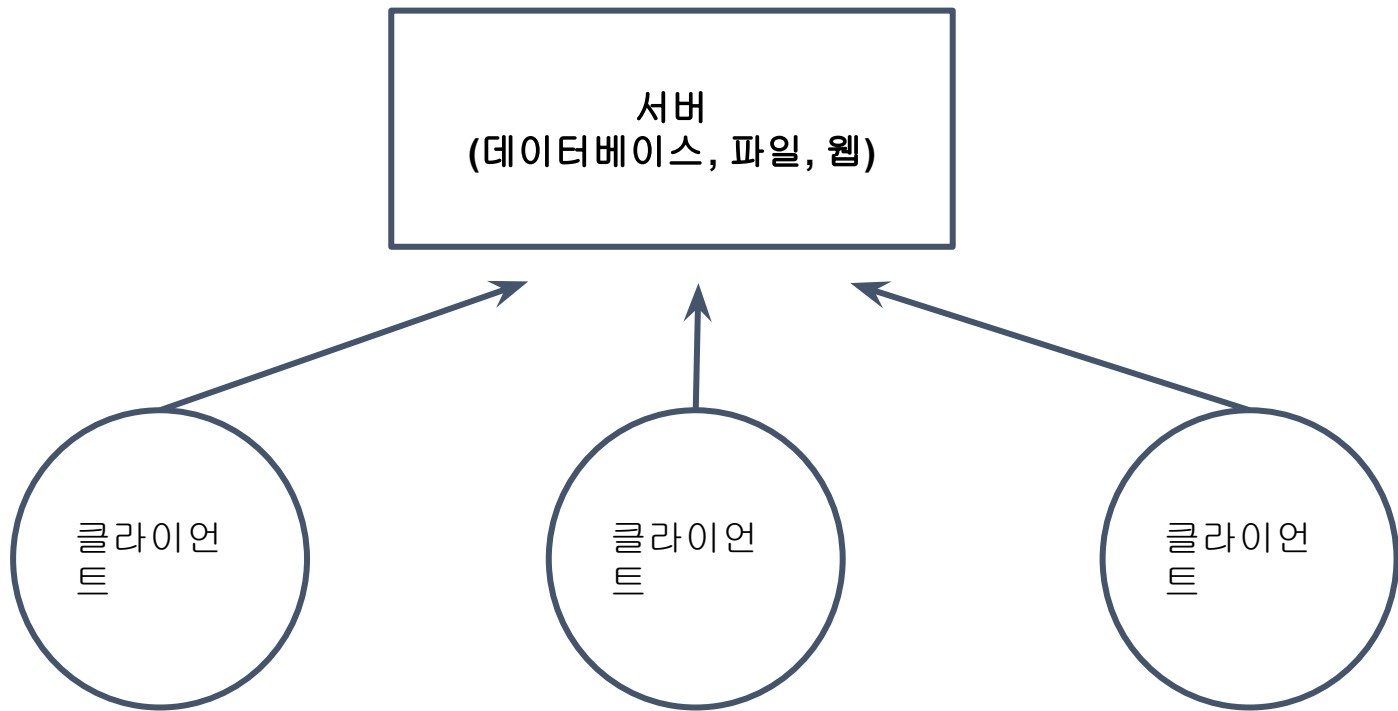
https://www.w3schools.com/sql/trysql.asp?filename=trysql_asc

데이터베이스 설치 (서버)

서버란?



서버란?



MYSQL 설치 (1)

<https://dev.mysql.com/downloads/mysql/>

General Availability (GA) Releases Archives

MySQL Community Server 8.0.19

Select Operating System:
Microsoft Windows

Looking for previous GA versions?

Recommended Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 8.0 the MySQL Installer package replaces the standalone MS packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

Other Downloads:

Download	Version	Size	Action
Windows (x86, 64-bit), ZIP Archive	8.0.19	187.8M	Download
<small>(mysql-8.0.19-winx64.zip) MD5: f15c52e7b499f8a0cf108e0e09ca0 Signature</small>			
Windows (x86, 64-bit), ZIP Archive	8.0.19	406.7M	Download
<small>Debug Binaries & Test Suite (mysql-8.0.19-winx64-debug-test.zip) MD5: 9d8855587acc429e7770e79d1801 Signature</small>			

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

General Availability (GA) Releases Archives

MySQL Installer 8.0.19

Select Operating System:
Microsoft Windows

Looking for previous GA versions?

Download	Version	Size	Action
Windows (x86, 32-bit), MSI Installer	8.0.19	18.6M	Download
<small>(mysql-installer-web-community-8.0.19.0.msi) MD5: 32043776a229b4b45f109a3860e081 Signature</small>			
Windows (x86, 32-bit), MSI Installer	8.0.19	398.9M	Download
<small>(mysql-installer-community-8.0.19.0.msi) MD5: 1a332015da71a89120a4717a03b6817c Signature</small>			

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »](#)
using my Oracle Web account

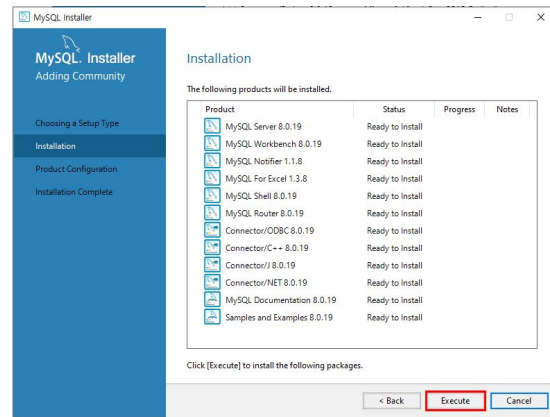
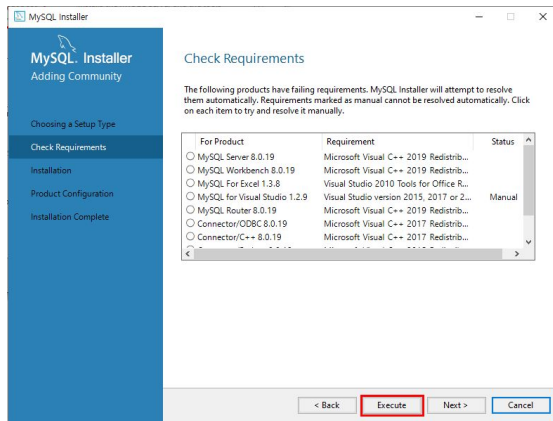
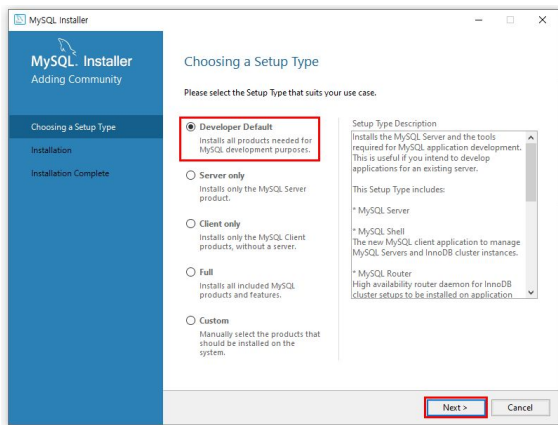
[Sign Up »](#)
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

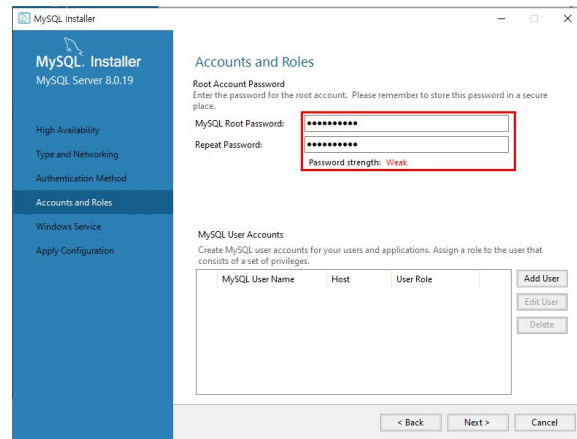
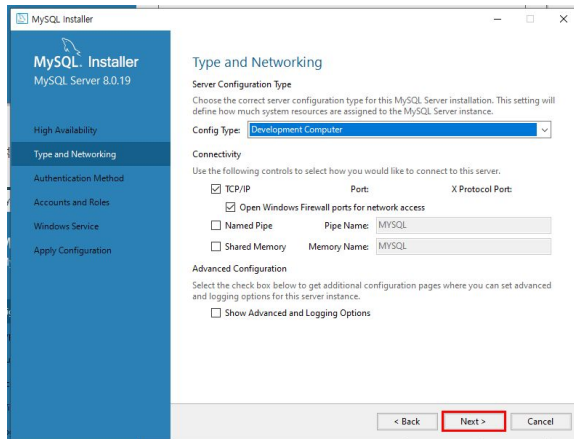
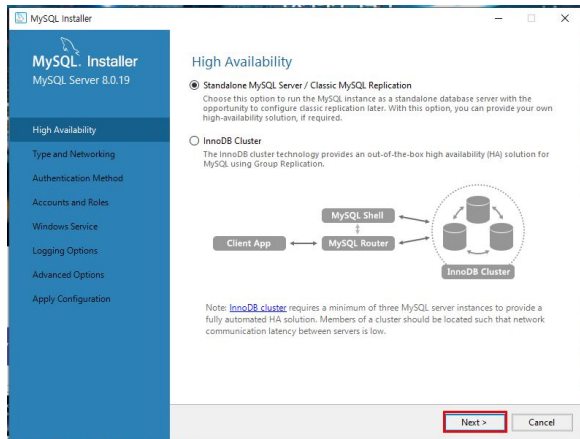
[No thanks, just start my download.](#)

ORACLE © 2020, Oracle Corporation and/or its affiliates.
Legal Policies | Your Privacy Rights | Terms of Use | Trademark Policy | Contributor Agreement | Cookie Preferences

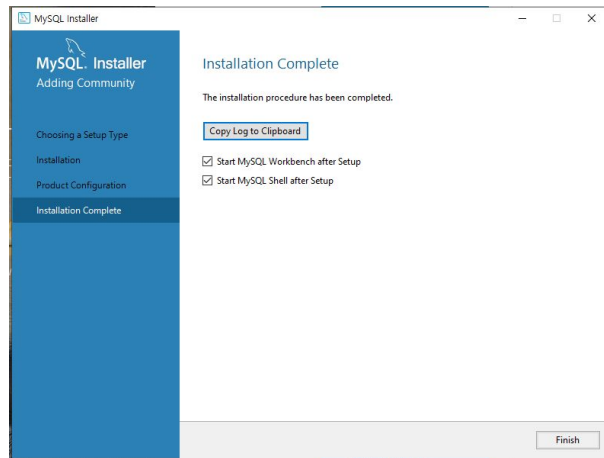
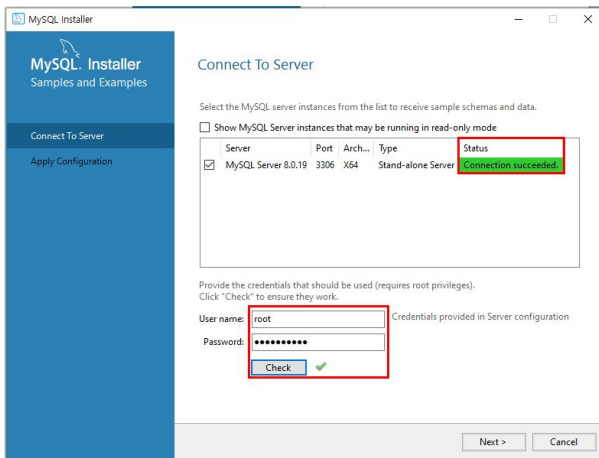
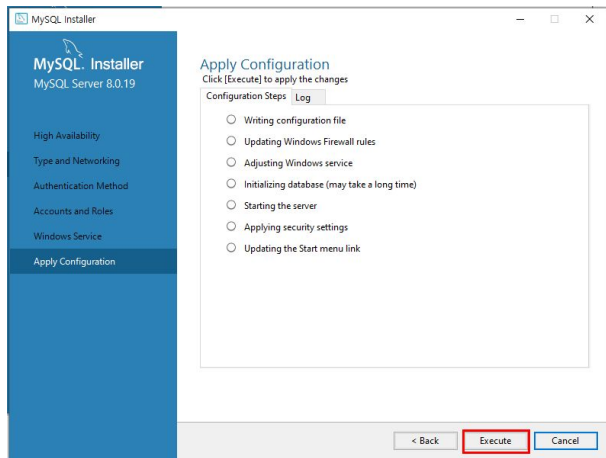
MYSQL 설치 (2)



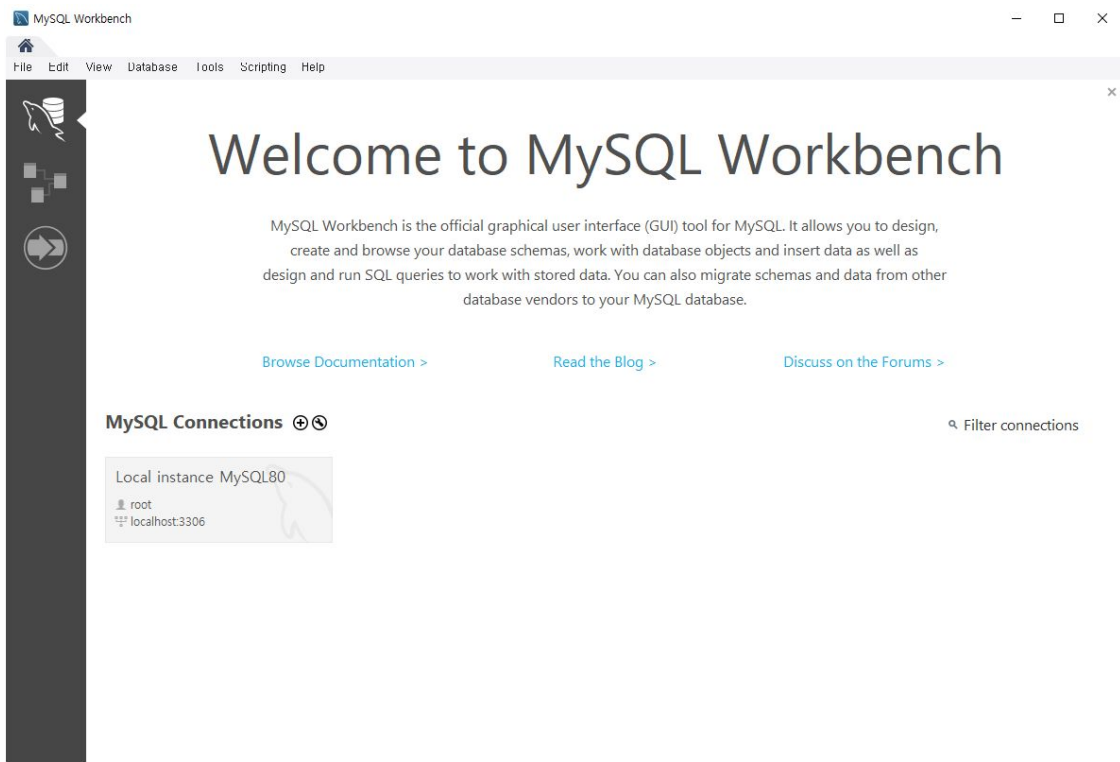
MYSQL 설치 (3)



MYSQL 설치 (4)



MySQL Workbench



데이터베이스 서버세팅

데이터베이스 생성

```
CREATE DATABASE DB명 CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

사용자 생성

CREATE USER '사용자명'@'호스트명' IDENTIFIED BY '사용자패스워드';



아이피 - 일부 아이피에서만 접속가능
localhost - 로컬에서만 접속가능
% - 외부에서 접속가능

권한부여

GRANT [권한부여1,권한부여2....] ON [데이터베이스명].[테이블명] TO [유저명]@[호스트주소];

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON **DB명**.* TO
'**사용자명**'@'**localhost**';

모든권한

GRANT ALL PRIVILEGES ON **DB명**.* TO '**사용자명**'@'**localhost**';

권한갱신

FLUSH PRIVILEGES

권한관련

계정권한조회

SHOW GRANTS FOR [유저명]@[호스트주소]

계정 삭제 명령어

DROP USER [유저명]@[호스트주소]

권한 삭제 명령어

REVOKE [권한1,권한2....] ON [데이터베이스명].[테이블명] FROM [유저명]

모든권한삭제

REVOKE ALL ON *.* FROM [유저명]@[호스트주소]

유저 및 권한조회

SELECT GRANTEE, PRIVILEGE_TYPE, IS_GRANTABLE
FROM INFORMATION_SCHEMA.USER_PRIVILEGES;

테이블 작성, 삭제, 변경

Create Table

데이터베이스 내 테이블 생성

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

```
CREATE TABLE table_name (  
    column1 datatype NOT NULL AUTO_INCREMENT,  
    column2 datatype,  
    column3 datatype,  
    PRIMARY KEY(column1)  
);
```

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

```
CREATE TABLE Persons (  
    PersonID int NOT NULL AUTO_INCREMENT,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255),  
    PRIMARY KEY(SEQ)  
);
```

Create Table

기존 테이블을 이용해 테이블 생성

```
CREATE TABLE new_table_name AS  
  SELECT column1, column2,...  
  FROM existing_table_name  
  WHERE ....;
```

```
CREATE TABLE TestTable AS  
  SELECT customername, contactname  
  FROM customers;
```

데이터 타입 (숫자형)

TYPE	사용되는 바이트	최소값 (signed/unsigned)	최대값 (signed/unsigned)
TINYINT	1	-128 0	127 255
SMALLINT	2	-32768 0	32767 65535
MEDIUMINT	3	-8388608 0	8388607 16777215
INT or INTEGER	4	-2147483648 0	2147483647 4294967295
BIGINT	8	-9223372036854775808 0	9223372036854775807 18446744073709551615
FLOAT	4	-3.40E+45 (no unsigned)	3.40E+45 (no unsigned)
DOUBLE or REAL	8	-1.7976E+320 (no unsigned)	1.7976E+320 (no unsigned)

데이터 타입 (문자형)

TYPE	표현범위	설명
CHAR(N)	정확히 n (<=255)	무조건 크기만큼 메모리사용
VARCHAR(N)	최대 n (<=65535)	안에 데이터만큼 메모리사용
TINYTEXT(N)	최대 n (<=255)	문자열
TEXT(N)	최대 n (<=65535)	문자열
MEDIUMTEXT(N)	최대 n (<=16777215)	문자열
LONGTEXT(N)	최대 n (<=4294967295)	문자열

데이터 타입 (바이너리)

TYPE	표현범위	예제
TINYBLOB(N)	최대 n (≤ 255)	바이너리
BLOB(N)	최대 n (≤ 65535)	바이너리
MEDIUMBLOB(N)	최대 n (≤ 16777215)	바이너리
LOB(N)	최대 n (≤ 4294967295)	바이너리

데이터 타입 (날짜형)

TYPE	표현범위	예제
DATE	1000-01-01 ~ 9999-12-31	YYYY-MM-DD
DATETIME	1000-01-01 00:00:00 ~ 9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS 문자로 저장
TIMESTAMP	1970-01-01 ~ 2037-01-01 임의시간	YYYY-MM-DD HH:MM:SS 숫자로 저장, 타임존
TIME	-838:59:59 ~ 838:59:59	
YEAR	901~2155	

Drop Table

데이터베이스 내 테이블 삭제

```
DROP TABLE table_name;  
TRUNCATE TABLE table_name;
```

Alter Table

데이터베이스 내 테이블 수정

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

```
ALTER TABLE table_name  
MODIFY column_name datatype;
```

```
ALTER TABLE Customers  
ADD Email int(10);
```

```
ALTER TABLE Customers  
DROP COLUMN Email;
```

```
ALTER TABLE Customers  
MODIFY Email varchar(255);
```

실습

Q. 아래와 같이 테이블을 두개 만드세요

테이블명 : Students

studentNumber - 학번(자동증가)
name - 이름
age - 나이
address - 주소

테이블명 : scores

studentNumber - 학번
math - 수학점수
english - 영어점수
science - 과학점수

데이터 추가, 삭제, 갱신

INSERT INTO

지정 컬럼을 기준으로 입력

```
INSERT INTO table_name (column1, column2,  
column3, ...)  
VALUES (value1, value2, value3, ...);
```

컬럼 순서별로 입력

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

조회 결과를 다시 입력

```
INSERT INTO table2 (column1, column2,  
column3, ...)  
SELECT column1, column2, column3, ...  
FROM table1  
WHERE condition;
```

DELETE

DELETE 문을 사용하여 조건에 맞는 데이터 삭제

조건에 맞는 데이터 삭제

```
DELETE FROM table_name WHERE condition;
```

Q. 독일에 살고 있는 고객정보 삭제

UPDATE

UPDATE 문을 사용하여 조건에 맞는 데이터 수정

조건에 맞는 데이터 수정

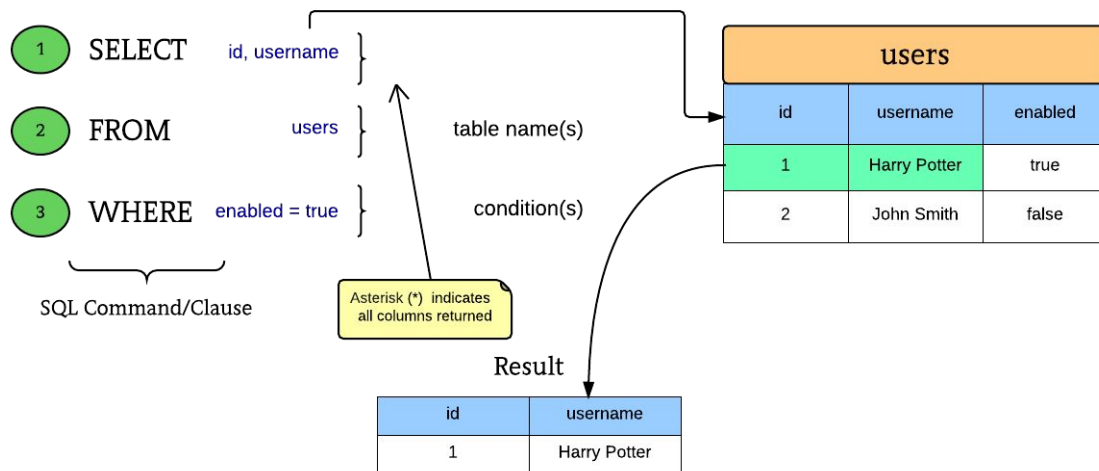
```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Q. 베를린(Berlin)에 살고 있는 고객의 우편번호를
12210 으로 수정

데이터 검색

SELECT

Anatomy of the SELECT Statement



SELECT

SELECT 문을 사용하여 데이터 조회

컬럼을 지정하여 조회

```
SELECT column1, column2, ...  
FROM table_name;
```

전체컬럼을 조회

```
SELECT * FROM table_name;
```

지정컬럼의 유일한 값 조회

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

Q. 고객(Customers)의 이름과 국가를 조회

Table - Customers

Q. 고객(Customers) 정보 전체 조회

Table - Customers

Q. 고객(Customers)의 국가(Country) 목록 조회 (중복X)

Table - Customers

WHERE

WHERE 문을 사용하여 조건에 맞는 데이터 조회

employees

조건에 맞는 데이터 조회

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Q. 국가(Country)가 France 고객(Customers) 조회

Table - Customers

패턴(조건)에 맞는 데이터 조회

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column LIKE pattern;
```

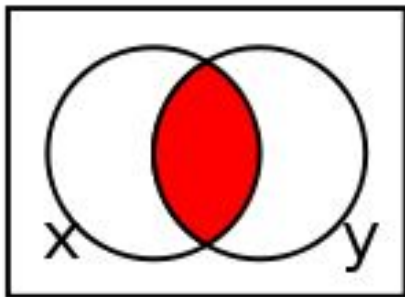
고객(Customers) 테이블에서

Q. 이름이 'Mar'로 시작하는 직원(ContactName) 조회

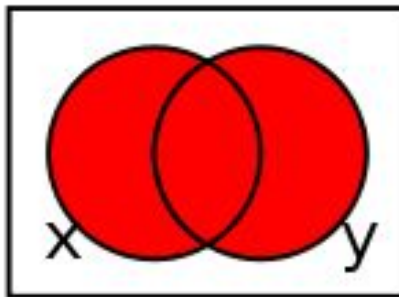
Q. 이름이 'et'로 끝나는 직원(ContactName) 조회

Table - Customers

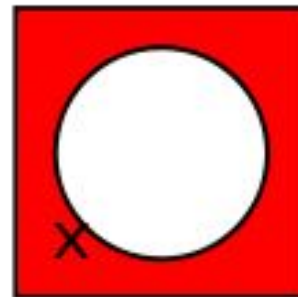
AND, OR, NOT



AND



OR



NOT

AND, OR, NOT

AND, OR, NOT을 사용하여 세밀한 조건 설정

AND 조건에 맞는 데이터 조회

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

OR 조건에 맞는 데이터 조회

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT 조건에 맞는 데이터 조회

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

Q. 국가(Country)가 France이고 이름
(CustomerName)이 'La'로 시작하는 고객
(Customers) 조회

Table - Customers

Q. 국가(Country)가 Germany이거나 이름
(CustomerName)이 'Ma'로 시작하는 고객
(Customers) 조회

Table - Customers

Q. 국가(Country)가 France가 아닌 고객
(Customers) 조회

Table - Customers

IN, BETWEEN

IN, BETWEEN 사용하여 조건 설정

IN 조건에 맞는 데이터 조회

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

IN 조건에 맞는 데이터 조회

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

BETWEEN 사이 조건에 맞는 데이터 조회

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

Q. 국가(Country)가 France 혹은 Spain 사는 고객 (Customers) 조회

Table - Customers

Q. 가격(Price)이 15에서 20사이인 상품 (Products) 의 상품명(ProductName) 목록 조회

Table - Products

Q. 가격(Price)이 15에서 20사이인 상품 (Products) 조회

Table - Products

NULL

NULL 값 처리 (다른 자료형과는 별도로 처리)

컬럼 데이터가 NULL인 데이터 조회

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

Q. 우편번호(PostalCode)가 NULL 인 고객(Customers) 목록
조회

Table - Customers

컬럼 데이터가 NULL이 아닌 데이터 조회

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

Q. 우편번호(PostalCode)가 NULL 이 아닌 고객(Customers)
목록 조회

Table - Customers

데이터 정렬과 연산

ORDER BY

조회시 정렬 적용 (오름차순 / 내림차순)

지정 컬럼을 기준으로 오름/내림차순 조회

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

컬럼 별로 오름/내림차순 조회

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1 ASC|DESC, column2 ASC|DESC,  
... ;
```

Q. 고객이름(CustomerName) 오름차순(ASC)으로 조회
Table - Customers

Q. 상품가격(Price) 내림차순(DESC)으로 조회
Table - Products

Q. 고객이름(CustomerName) 오름차순(ASC)으로, 나라
(Country) 내림차순(DESC)으로 조회

Table - Cusotmers

TOP, LIMIT, ROWNUM

조회 건수를 제한하여 데이터 조회

SQL Server / MS Access Syntax:

```
SELECT TOP number | percent column_name(s)  
FROM table_name  
WHERE condition;
```

MySQL Syntax:

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```

Oracle Syntax:

```
SELECT column_name(s)  
FROM table_name  
WHERE ROWNUM <= number;
```

Q. 국가(Country) 가 UK 고객 중 이름순 3명 조회

Table - Customers

CASE

조건에 따라 값을 변환

MySQL Syntax:

```
SELECT
*,
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END
FROM tablename;
```

Q. 상품가격(Price) 이 30 미만, '저', 30~50 '중',
50 초과는 '고'로 조회

Table - Products

데이터 집계

COUNT, AVG, SUM

집계 함수(COUNT, AVG, SUM) 사용

데이터 갯수 조회

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

Q. France 에 거주하는 고객(Customers)수 조회
: 국가명(Country), 고객수

Table - Customers

데이터 평균 값 계산

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

Q. 전체상품(Price) 평균가격 계산
: 평균가격

Table - Products

데이터 합산 값 계산

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

Q. 주문상품(Quantity) 수량 합계 계산
: 주문수량 합계

Table - OrderDetails

MIN, MAX

집계 함수(MIN, MAX) 사용

최소값 조회

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

Q. 상품가격(Price) 중 최소값 조회
: 최소가격

Table - Products

최대값 조회

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

Q. 상품가격(Price) 중 최대값 조회
: 최대가격

Table - Products

GROUP BY

열기준 데이터 그룹핑

그룹핑

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Q. 국가별 고객수 조회 (고객수 기준 오름차순)
: 국가명(Country), 고객수

Table - Customers

Q. 국가 별, 도시 별 고객수 조회 (고객수 기준 내림차순)
: 국가명(Country), 도시명(City), 고객수

Table - Customers

HAVING

집계데이터를 기준으로 조건설정

집계데이터로 조건 설정

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

Q. 국가별 고객수를 조회하고 그 중 5명 초과인
국가만 조회 (고객수 내림 차순)
: 국가명, 고객수

Table - Customers

기타

Comment

```
-- Select all:  
SELECT * FROM Customers;
```

```
SELECT * FROM Customers -- WHERE City='Berlin';
```

```
-- SELECT * FROM Customers;  
SELECT * FROM Products;
```

Alias

```
SELECT column_name AS alias_name  
FROM table_name;
```

```
SELECT column_name(s)  
FROM table_name AS alias_name;
```

UNION

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;;
```

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```

실습

Q. 상품(Products) 중 상품명(ProductName) 가 C 로 시작하고 가격(Price) 이 20보다 큰 상품의 상품명과 가격을 가격이 비싼순으로 정렬하세요

:상품명(ProductName), 가격(Price)

Table - Products

Q. 상품(Products) 의 카테고리아이디(CategoryID) 별로 상품가격의 합, 가장비싼상품 가격, 가장저렴한상품 가격을 구하세요

: 카테고리아이디(CategoryID), 상품가격이—합, 가장비싼상품 가격, 가장저렴한상품 가격

Table - Products

데이터 조인

조인?

이름	전화번호
홍길동	010-3242-5931
이동진	010-3943-1992
박철우	010-6123-4453

이름	구입일	상품	수량
홍길동	2019-01-02	우유	1
홍길동	2019-01-03	식빵	2
이동진	2018-12-21	치즈	1
박철우	2018-12-23	소금	1
박철우	2018-12-25	우유	3



이름	전화번호	구입일	상품	수량
홍길동	010-3242-5931	2019-01-02	우유	1
홍길동	010-3242-5931	2019-01-03	식빵	2
이동진	010-3943-1992	2018-12-21	치즈	1
박철우	010-6123-4453	2018-12-23	소금	1
박철우	010-6123-4453	2018-12-25	우유	3

INNER JOIN

두 테이블간 INNER JOIN

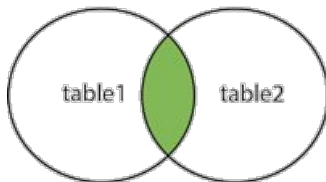
INNER JOIN

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

Q. 주문이력이 있는 고객명(CustomerName)과
주문일(OrderDate)
:고객명(CustomerName), 주문일(OrderDate)

Table - Customers, Orders [CustomerID]

INNER JOIN



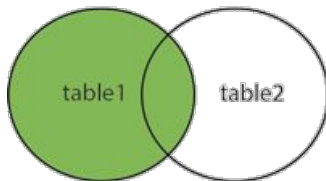
LEFT OUTER JOIN

(구문상) 왼쪽에 위치한 테이블을 기준으로 데이터 조회 (오른쪽 테이블에 NULL 값이 있어도 조회됨)

LEFT OUTER JOIN

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

LEFT JOIN



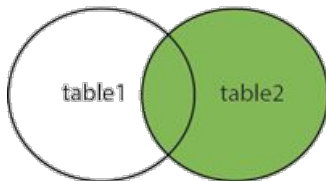
RIGHT OUTER JOIN

(구문상) 오른쪽에 위치한 테이블을 기준으로 데이터 조회 (왼쪽 테이블에 NULL 값이 있어도 조회됨)

RIGHT OUTER JOIN

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

RIGHT JOIN



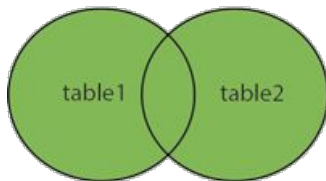
FULL OUTER JOIN

두 테이블간 NULL 값이 있어도 모두 조회

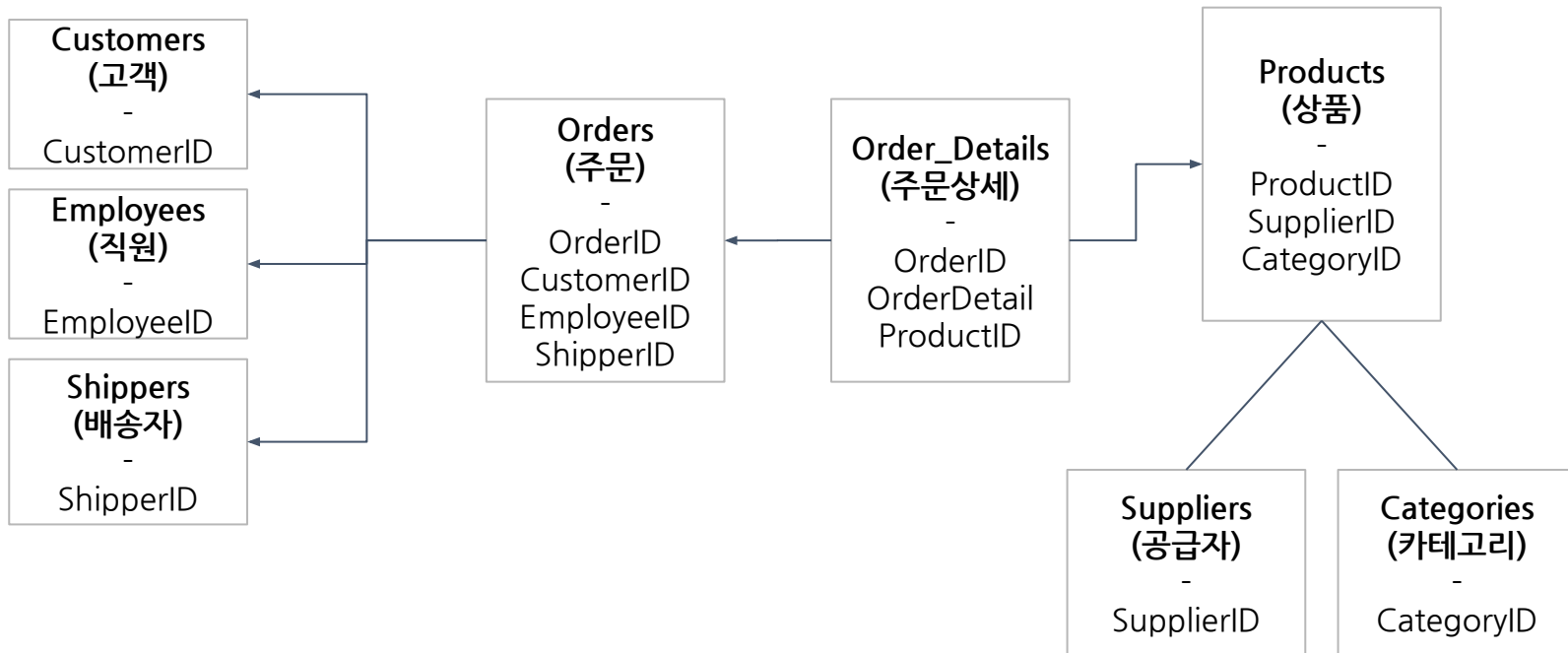
FULL OUTER JOIN

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

FULL OUTER JOIN



ERD



실습

관계를 파악하고 JOIN을 활용한 실습예제

Q. 도쿄에 위치한 공급자(Supplier)가 생산한 상품 목록 조회 :

도시명(City), 공급자명(SupplierName), 상품명(ProductName), 상품가격(Price)

Table - Suppliers, Products

Q. 분류(Categories)가 Seafood 인 상품명(ProductName) :

분류명(CategoryName), 상품명(ProductName)

Table - Categories, Products

Q. 공급자(Supplier) 가 공급한 상품의 공급자 국가(Country) 별로 상품 카테고리(CategoryName) 의 상품건수와 평균가격 :

국가명(Country), 카테고리명(CategoryName), 상품건수, 평균가격 (3개 테이블 조인)

Table - Suppliers, Products, Categories

Q. 주문별 주문자명(CustomerName), 직원명(LastName), 배송자명(ShipperName), 주문상세갯수 :

주문아이디(OrderID), 주문자명(CustomerName), 직원명(LastName), 배송자명(ShipperName), 주문상세갯수

Table - Orders, Order_Details, Customers(고객,주문자), Employees(직원), Shippers(배송자)

실습

관계를 파악하고 JOIN을 활용한 실습예제

Q. 판매량(Quantity) 상위 TOP3 공급자(supplier) 목록 :
공급자명(SupplierName), 판매량(Quantity)

Table - Suppliers, Product, Order_Details

Q. 상품분류(Category)별, 고객지역별(City) 판매량(Quantity) 많은순으로 정렬 :
카테고리명(CategoryName), 고객지역명(City), 판매량(Quantity)

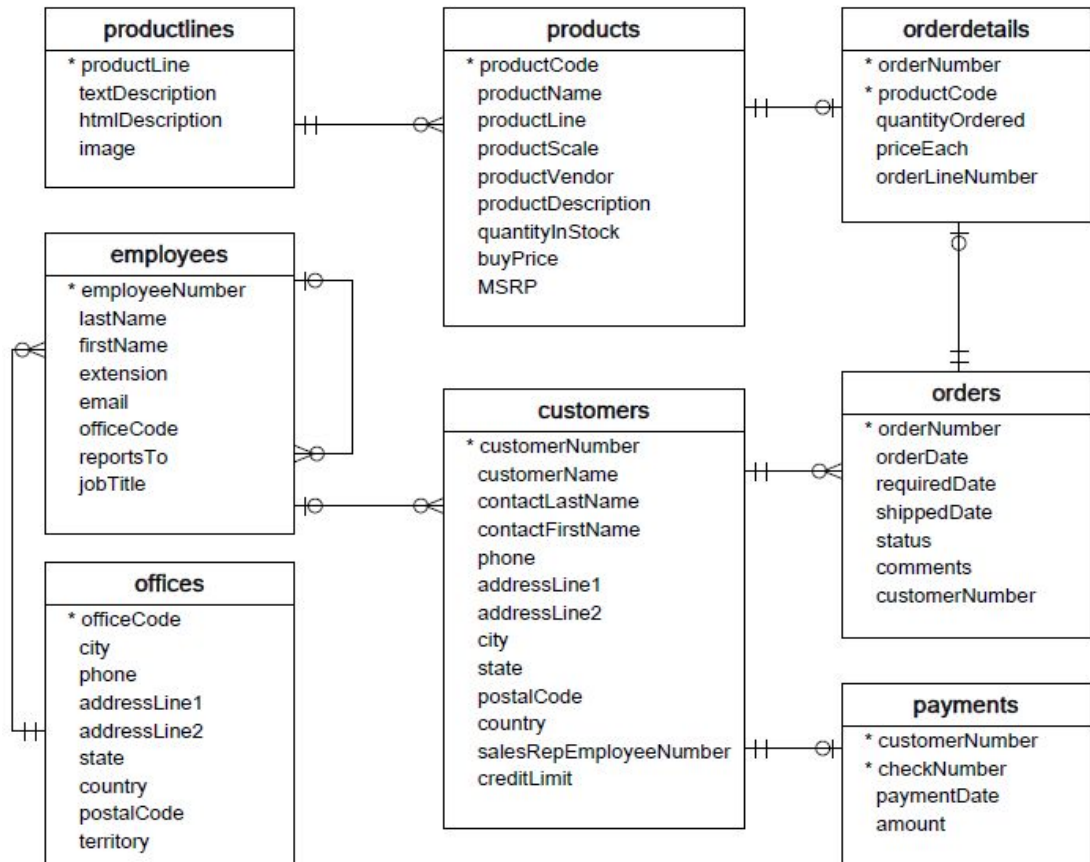
Table - Order, Order_Details, Customers, Categories, Products

Q. 고객국가(Country) 가 USA이고, 상품별판매량(Quantity)의 합이 많은순으로 정렬 :
국가명(Country), 상품명(ProductName), 판매량합(Quantity), 판매금액합(Price)

Table - Customers, Products, Orders, Order_Details

classicmodel 데이터베이스 실습

ERD



실습

Q. 사무실이 위치한 국가(Country) 별 직원수 : 국가(Country), 직원수

Table - Employees, Offices

Q. 상품별 일자별 판매수 추이 : 상품명, 판매일자, 판매수량(QuantityOrdered), 판매금액(QuantityOrdered * PriceEach)

Table - Orders, OrderDetails, Products

Q. 고객별 Credit 잔여한도 : 고객명, 잔여한도(CreditLimit - Amount 합계) , 한도소진율 (CreditLimit 대비 카드사용량)

Table - Customers, Payments

Q. 직원별 판매금액 랭킹 (영업왕 찾기) Top 10 : 랭킹, 직원명, 판매수량, 판매금액, percentile (=1등의판매금액 대비 %)

VIEW

VIEW

실제 테이블기반으로 만드는 가상테이블 (논리적인 테이블)
VIEW를 통한 데이터 수정은 가능하나 제한적임
복잡한 SQL을 간단하게 표시할수있음
필요한 데이터만 사용자에게 보여줄 수 있음

생성

```
CREATE VIEW view_name (column1, column2, ...)  
AS  
SELECT column1, column2, ...  
FROM table_name;
```

삭제

```
DROP VIEW view_name
```

실습

아래의 쿼리결과를 VIEW 로 만들어보세요

Q. 도쿄에 위치한 공급자(Supplier)가 생산한 상품 목록 조회 :

도시명(City), 공급자명(SupplierName), 상품명(ProductName), 상품가격(Price)

Table - Suppliers, Products

Q. 분류(Categories)가 Seafood 인 상품명(ProductName) :

분류명(CategoryName), 상품명(ProductName)

Table - Categories, Products

외래키 (Foreign Key)

외래키 (Foreign Key)

두개의 테이블을 하나로 합칠때 사용
부모가 되는 테이블의 칼럼은 **unique** 1:다 해야한다.

Foreign Key

이름	전화번호
홍길동	010-3242-5931
이동진	010-3943-1992
박철우	010-6123-4453

이름	구입일	상품	수량
홍길동	2019-01-02	우유	1
홍길동	2019-01-03	식빵	2
이동진	2018-12-21	치즈	1
박철우	2018-12-23	소금	1
박철우	2018-12-25	우유	3

외래키 (Foreign Key)

생성

```
CREATE TABLE table_name
(  
    ...  
    FOREIGN KEY (column) REFERENCES table_name (column)  
)
```

```
ALTER TABLE table_name ADD CONSTRAINT key_name FOREIGN KEY(column)  
REFERENCES table_name (column)
```

삭제

```
ALTER TABLE table_name DROP FOREIGN KEY key_name
```

INDEX

INDEX

인덱스는 큰 테이블에서 소량 데이터를 검색할 때 사용
테이블에서 자주 사용하는 속성에 사용 (조건문, 조인)
SELECT 에션 이득을 얻지만 INSERT, UPDATE, DELETE 에션 속도가 떨어진다.

조회시 조회정보확인 **explain**

인덱스 생성

```
CREATE INDEX index_name ON table_name(column);
```

인덱스 확인

```
SHOW INDEX FROM table_name;
```

인덱스 삭제

```
DROP INDEX index_name ON table_name
```