

# DAY 2

Tableau

# Topics

- Tableau
- Database
- Small group project

# Tableau

# Number Functions

Function	Example
CEILING (number)	CEILING(2.145) = 3
POWER (number, power)	POWER(5,3) = 125
ROUND (number, [decimals])	ROUND(3.14152,2) = 3.14

# String Functions

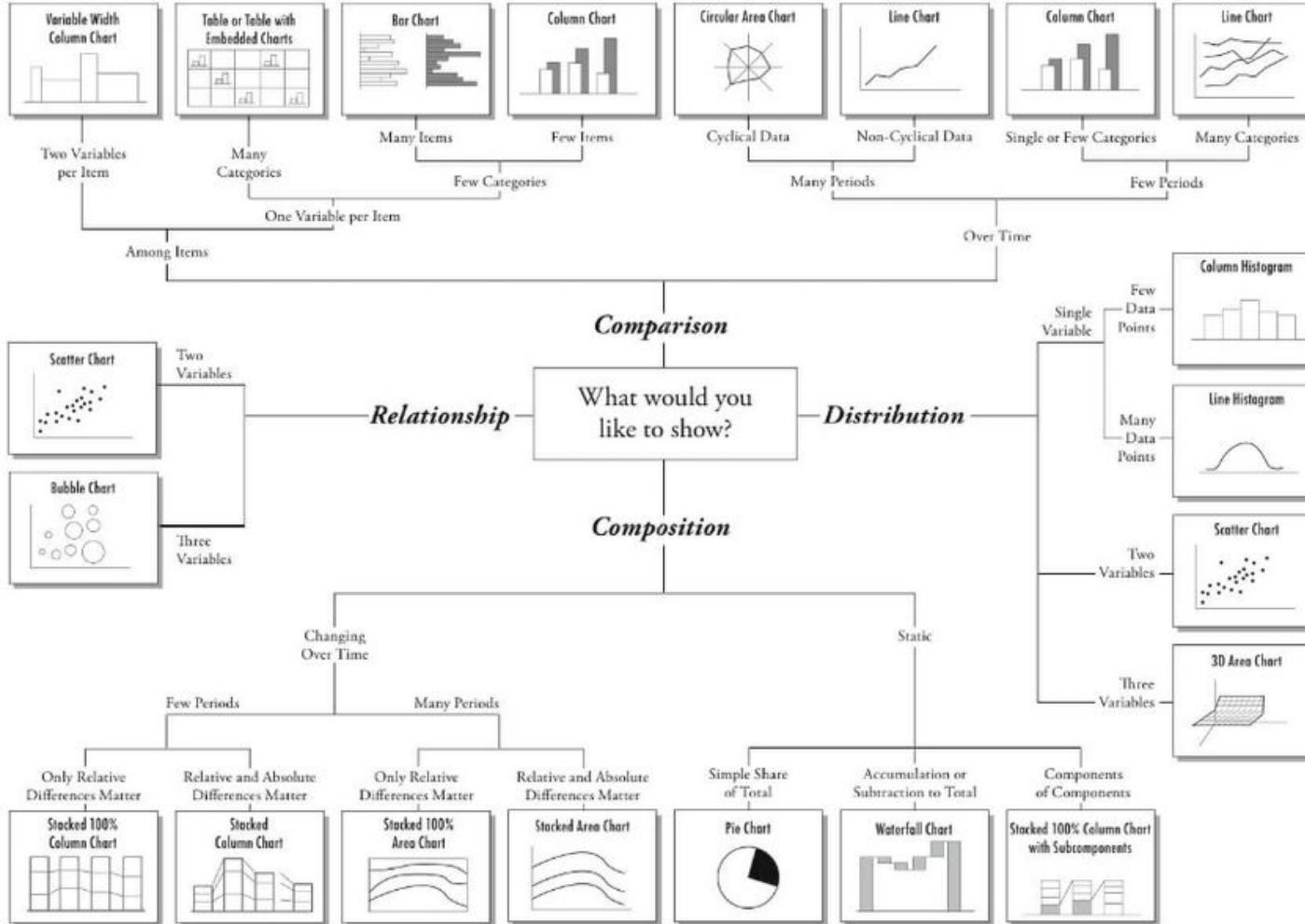
Function	Example
LEN (string)	LEN("Tableau") = 7
LTRIM (string)	LTRIM(" Tableau ") = "Tableau"
REPLACE (string, substring, replacement)	REPLACE("GreenBlueGreen", "Blue", "Red") = "GreenRedGreen"
UPPER (string)	UPPER("Tableau") = "TABLEAU"

# Date Functions

Function	Example
DATEADD (date_part, increment, date)	DATEADD ('month', 3, #2004-04-15#) = 2004-0715 12:00:00 AM
DATENAME (date_part, date, [start_of_week])	DATENAME('month', #200404-15#) = "April"
DAY (date)	DAY(#2004-04-12#) = 12
NOW( )	NOW( ) = 2004-04-15 1:08:21 PM

# Aggregate Functions

Function
AVG(expression)
MIN(expression)
COUNT (expression)
MEDIAN (expression)
STDEV (expression)

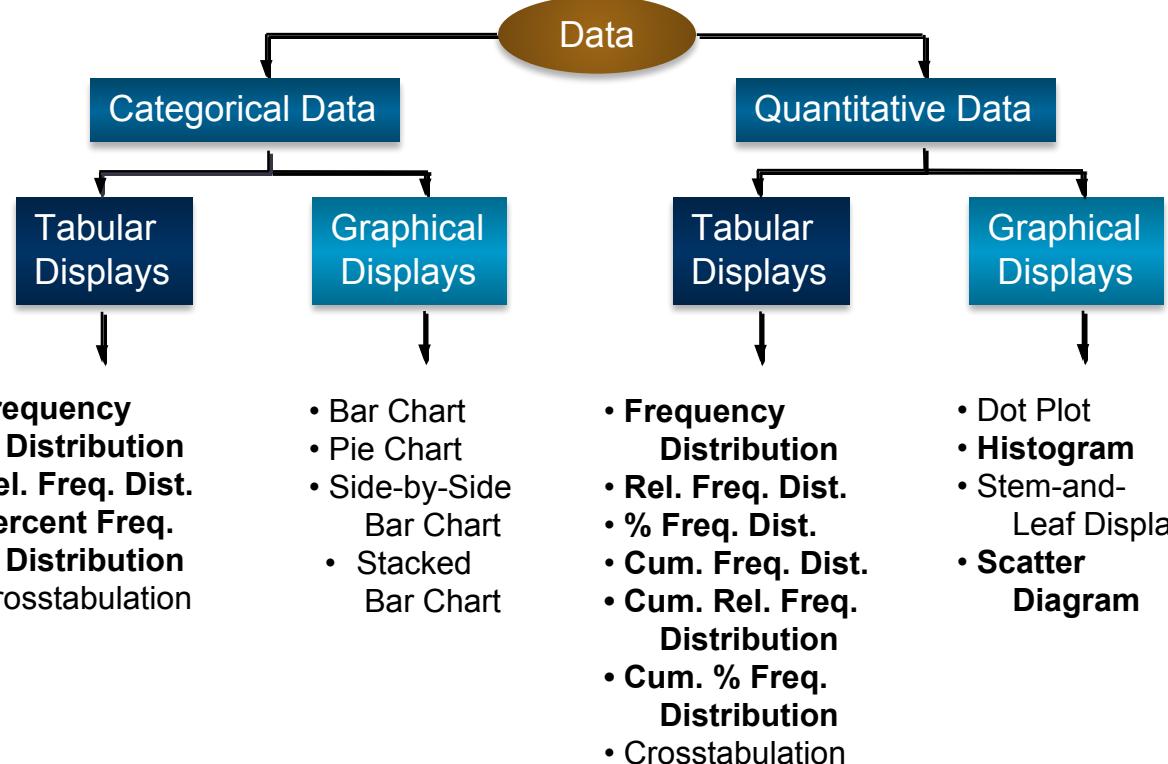


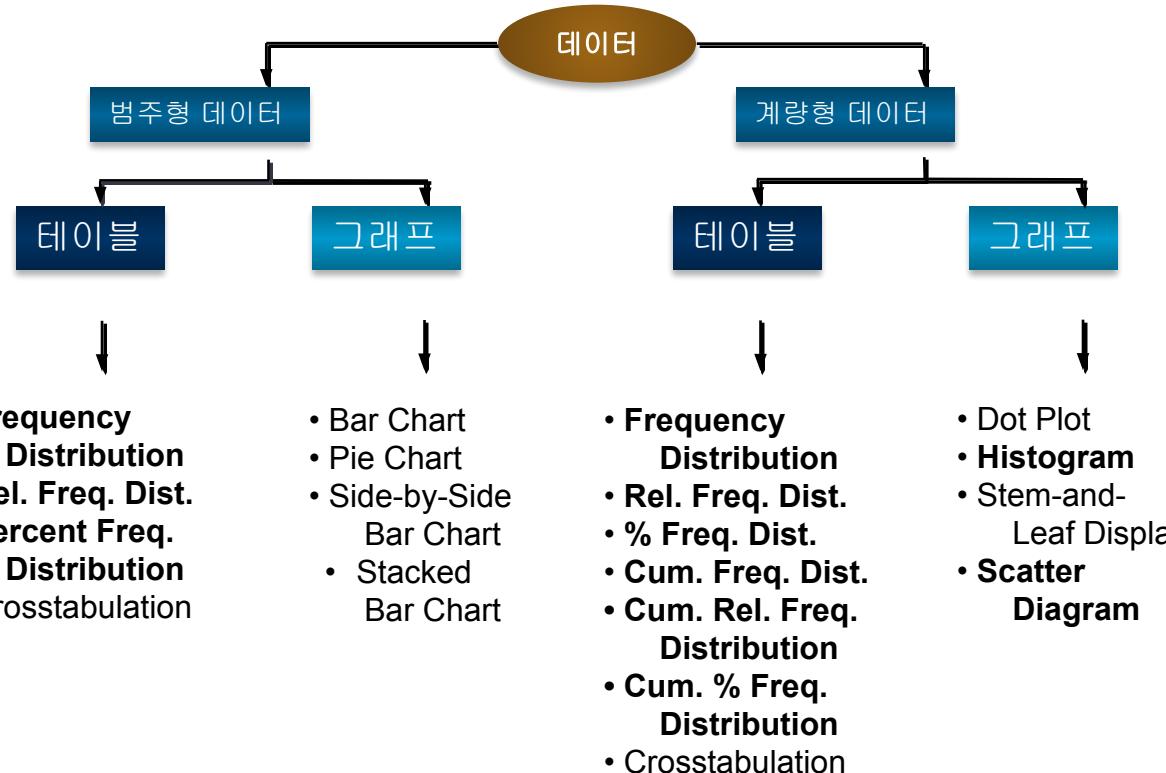
# Presentation and Chart Types

# Categorical and Quantitative Data

(범주형, 계량형 데이터)

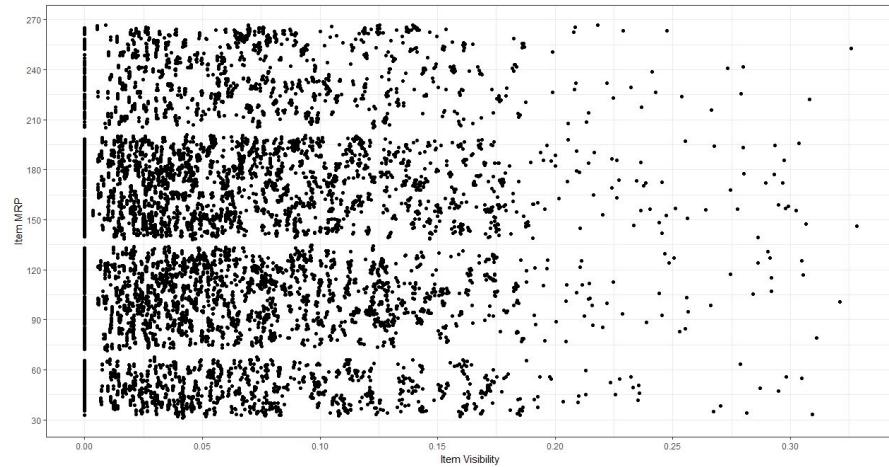
- Data can be further classified as being categorical or quantitative.  
데이터는 범주형, 계량형으로 나뉘어짐
- Categorical data use labels or names to identify categories of like items.  
범주형 데이터는 비슷한 아이템의 카테고리를 구분하는 레이블이나 이름을 사용하는 데이터들
- Quantitative data are numerical values that indicate how much or how many  
계량적 데이터는 얼마나 많은지를 나타내는 숫자들





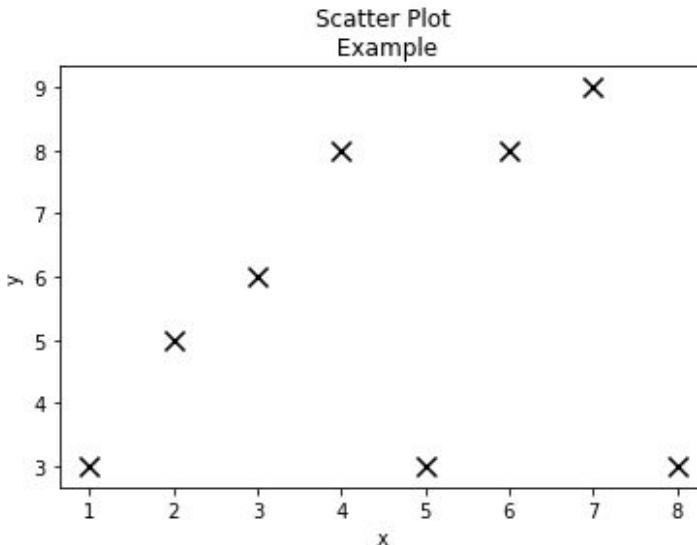
# Scatter Plot (산점도)

- Used to see the relationship between two continuous variables (두 연속변수 사이의 관계)



# Example

```
x = [1, 2, 3, 4, 5, 6, 7 ,8]
y = [3, 5, 6, 8, 3, 8, 9, 3]
plt.scatter(x,y, color='k',
marker='x', s=100)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Scatter Plot\nExample')
plt.show()
```



# Colors

- You can also use hex string

```
color = '#eeeefff'
```

- html names

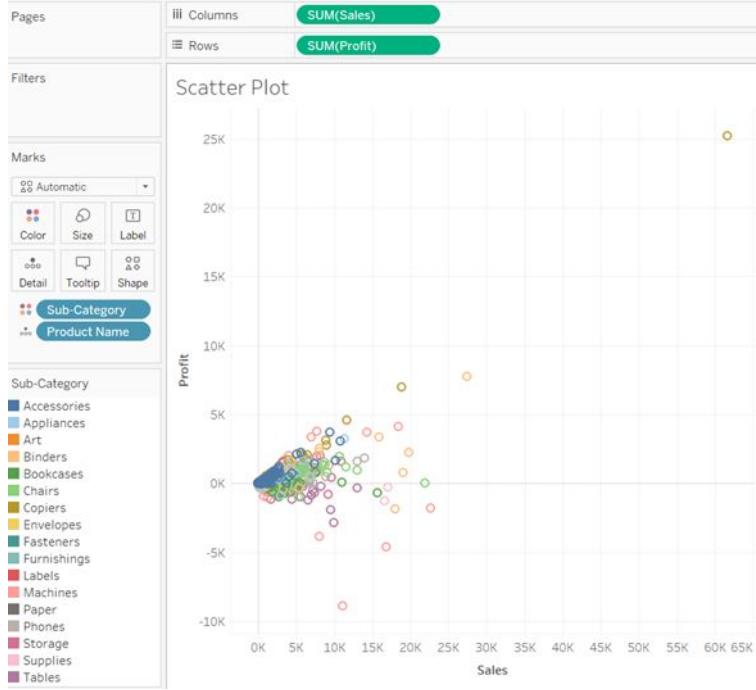
```
color = 'red'
```

```
color = 'burlywood'
```

```
color = 'chartreuse'
```

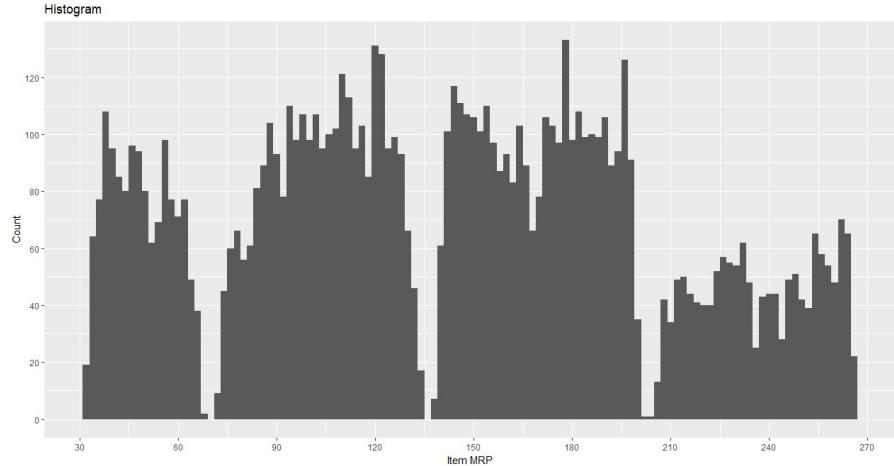
Alias	Color
'b'	blue
'g'	green
'r'	red
'c'	cyan (청록)
'm'	magenta (자홍색)
'y'	yellow
'k'	black
'w'	white

# Exercise #2



- [https://www.tutorialspoint.com/tableau/tableau\\_scatter\\_plot.htm](https://www.tutorialspoint.com/tableau/tableau_scatter_plot.htm)
- X, Y값 집어 넣고 안에 넣을 값을 선택함

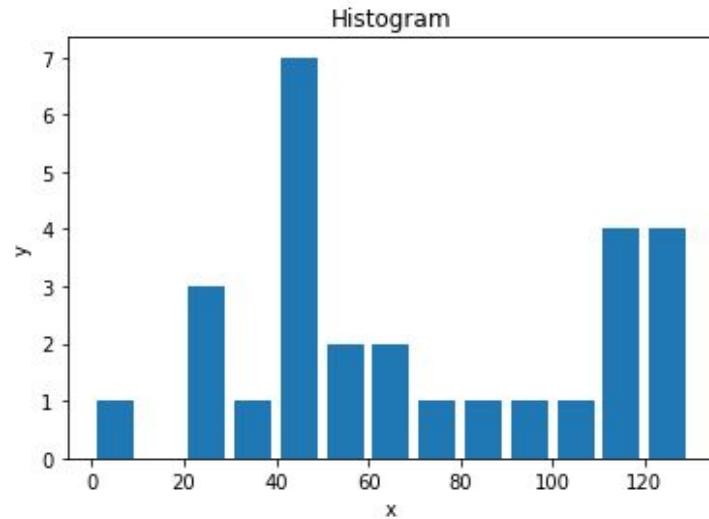
# Histogram



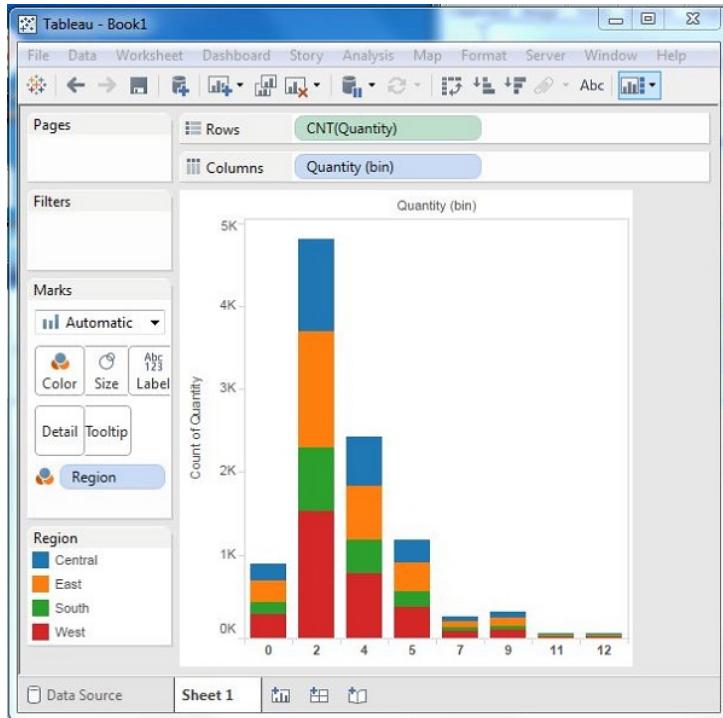
- Used to plot continuous variable  
(한개의 연속변수의 분포를  
보여줌)
- It breaks the data into bins and shows  
frequency distribution of these bins  
(빈도테이블을 그래프로!)

# Example

```
population_ages = [22, 55, 62, 45, 21,  
22, 34, 42, 42, 4, 99, 102, 110, 120,  
121, 122, 130, 111, 115, 112, 80, 75,  
65, 54, 44, 43, 42, 48]  
  
bins = [0, 10, 20, 30, 40, 50, 60,  
70, 80, 90, 100, 110, 120, 130]  
  
plt.hist(population_ages, bins,  
histtype='bar', rwidth=.8)  
  
plt.xlabel('x')  
plt.ylabel('y')  
plt.title('Histogram')  
plt.show()
```



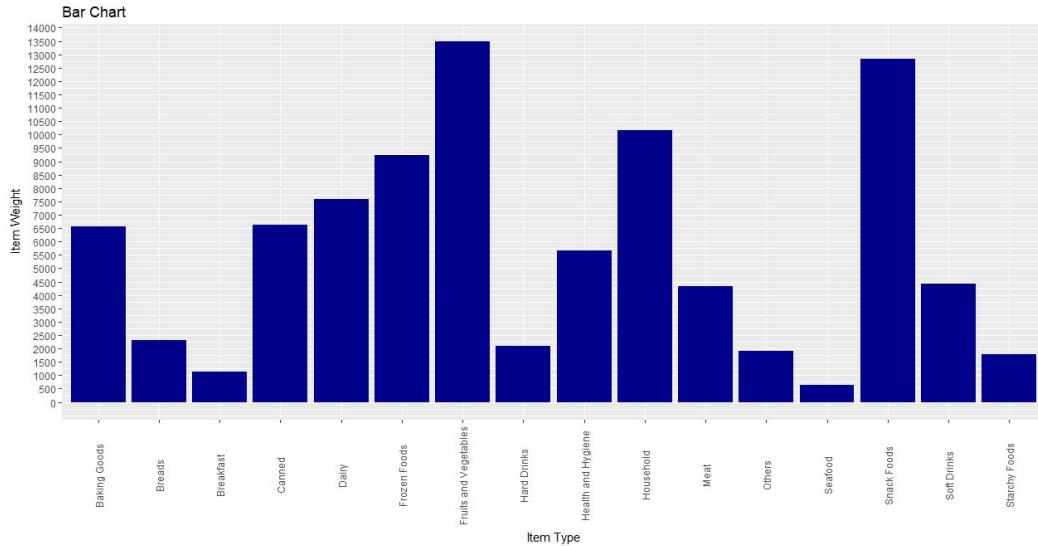
# Exercise #2



- [https://www.tutorialspoint.com/tableau/tableau\\_histogram.htm](https://www.tutorialspoint.com/tableau/tableau_histogram.htm)
- Drag the Measure named Quantity to the Rows shelf. Then open Show Me and select the Histogram chart.  
퀄티티만 행에 갖다놓고  
히스토그램 누름

# Bar Chart

- Plots a categorical variable or a combination of continuous and categorical variable (범주형변수나 연속, 범주변수가 같이 있을때 사용할수 있음)

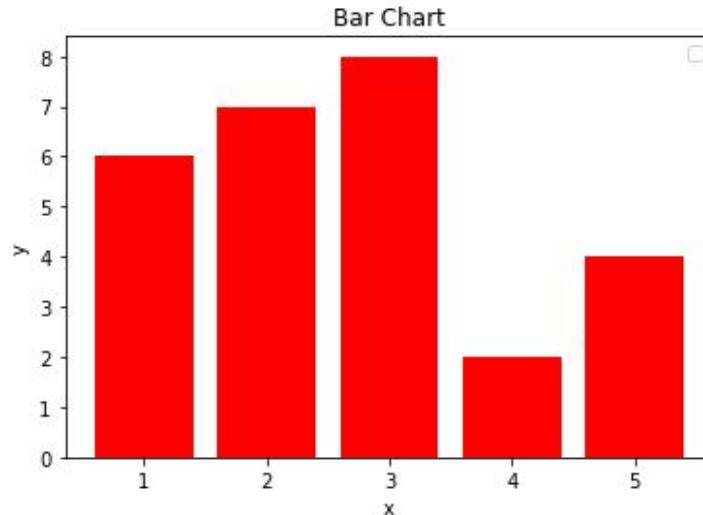


# Example

```
x = [1, 2, 3, 4, 5]
```

```
y = [6, 7, 8, 2, 4]
```

```
plt.bar(x,y, color='r')  
plt.xlabel('x')  
plt.ylabel('y')  
plt.title('Bar Chart')  
plt.legend()  
plt.show()
```

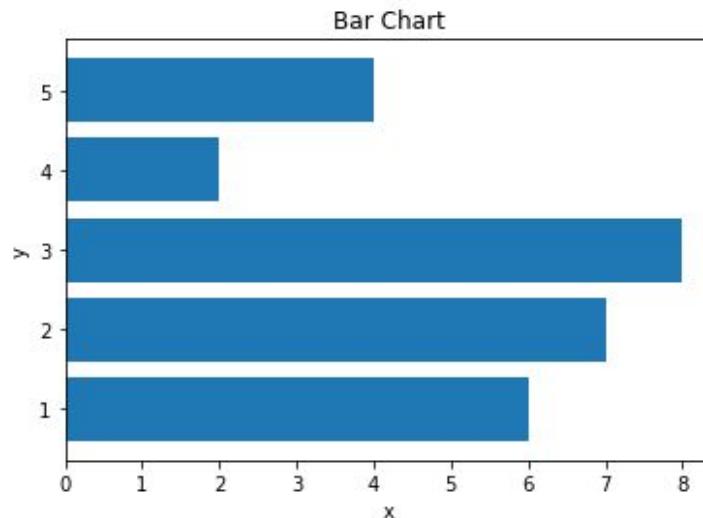


# Horizontal Bar Chart

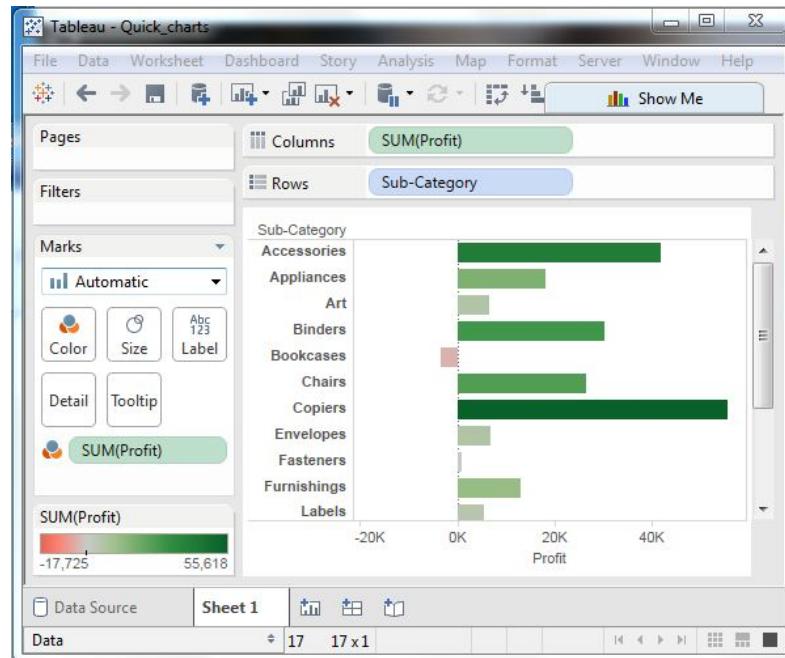
```
x = [1, 2, 3, 4, 5]
```

```
y = [6, 7, 8, 2, 4]
```

```
plt.bart(x,y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Bar Chart')
plt.show()
```



# Exercise #2



- [https://www.tutorialspoint.com/tableau/tableau\\_bar\\_chart.htm](https://www.tutorialspoint.com/tableau/tableau_bar_chart.htm)

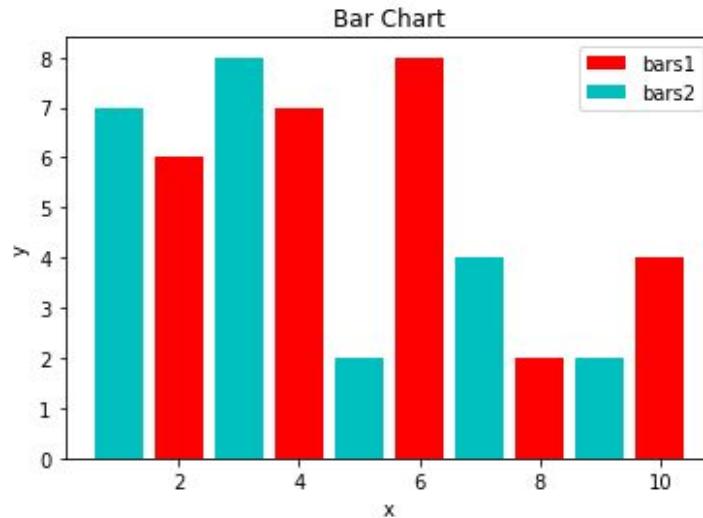
# Combining Charts

```
x = [2, 4, 6, 8, 10]
y = [6, 7, 8, 2, 4]

x2 = [1, 3, 5, 7, 9]
y2 = [7, 8, 2, 4, 2]

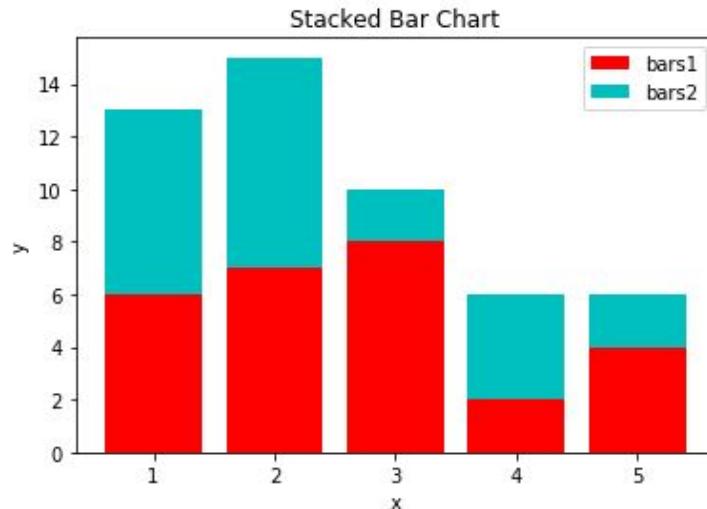
plt.bar(x, y, label='bars1',
        color='r')
plt.bar(x2, y2,
        label='bars2', color='c')

plt.xlabel('x')
plt.ylabel('y')
plt.title('Two Bar Charts')
plt.legend()
plt.show()
```

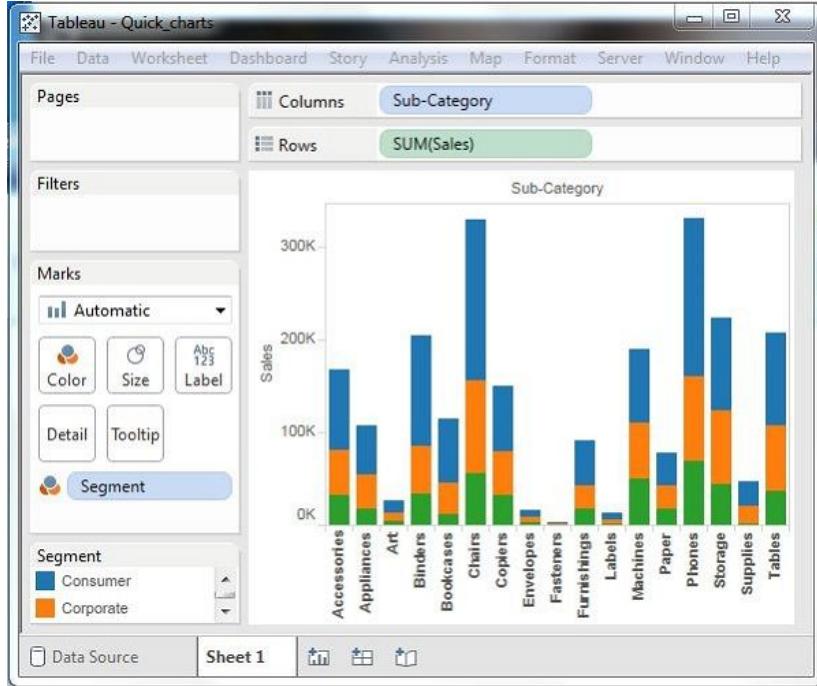


# Stacked Chart

```
x = [1, 2, 3, 4, 5]
y1 = [6, 7, 8, 2, 4]
y2 = [7, 8, 2, 4, 2]
plt.bar(x, y1, label='bars1', color='r')
plt.bar(x, y2, label='bars2', color='c',
        bottom=y1)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Stacked Bar Chart')
plt.legend()
plt.show()
```



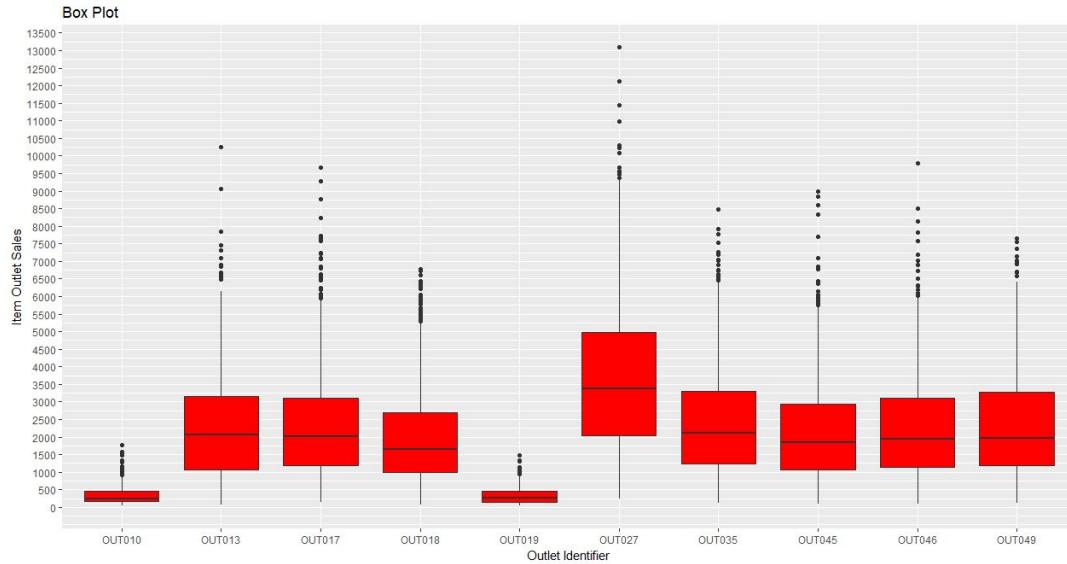
# Exercise #2



- [https://www.tutorialspoint.com/tableau/tableau\\_histogram.htm](https://www.tutorialspoint.com/tableau/tableau_histogram.htm)
- 그래프에 세그먼트를 칼라로  
집어넣음

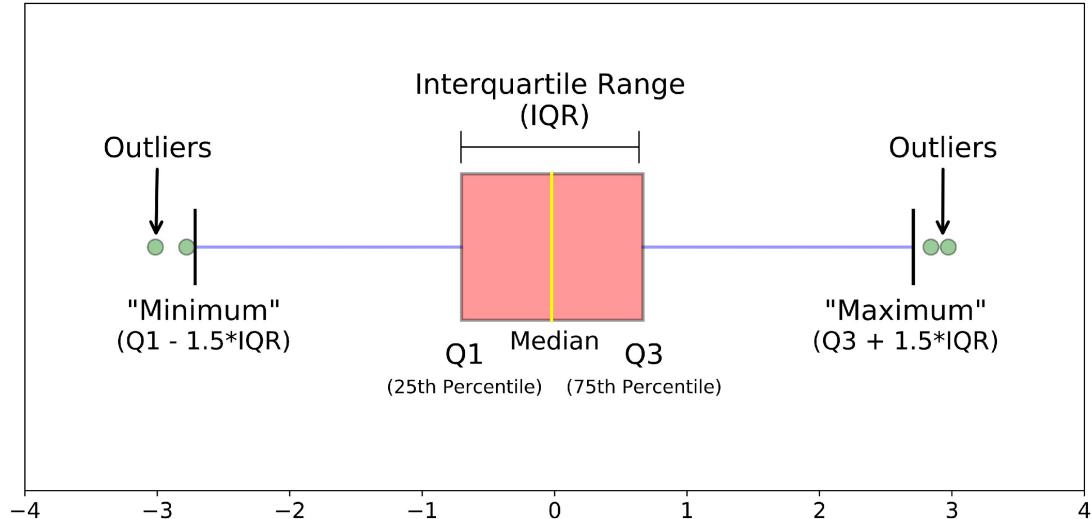
# Box Plot

- Used to plot a combination of categorical and continuous variables (범주형 변수와 연속변수의 관계를 보여줄때)



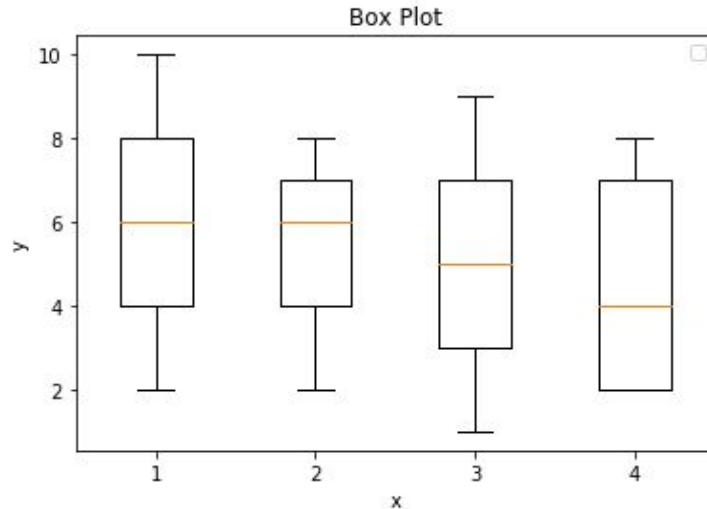
# Boxplot

- The median of a data set is the value in the middle when the data items are arranged in ascending order 중위수는 순서대로 정렬시 가운데 있는 값

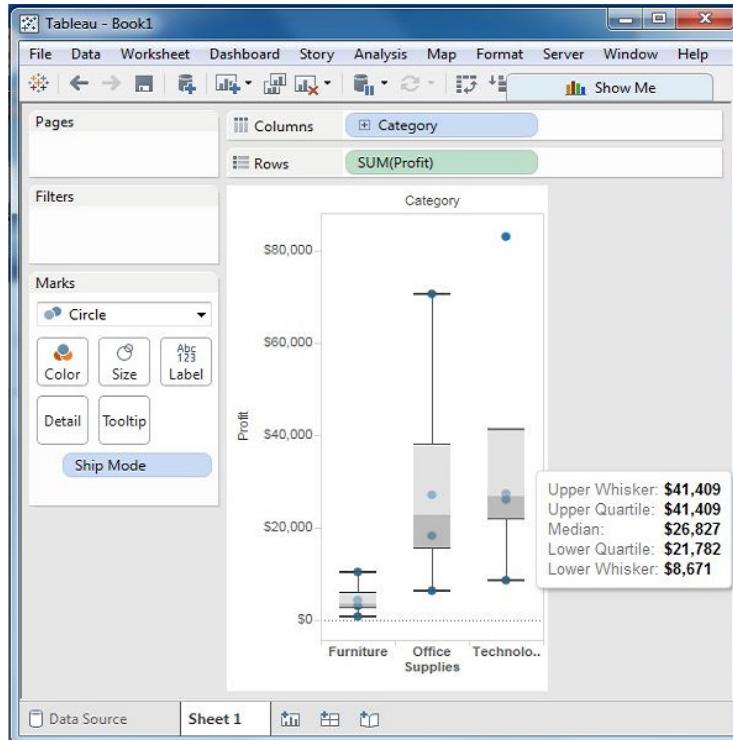


# Example

```
Data = [[2, 4, 6, 8, 10], [6, 7, 8, 2, 4], [1, 3, 5, 7, 9], [7, 8, 2, 4, 2]]  
x = pd.DataFrame(data)  
plt.boxplot(x)  
plt.xlabel('x')  
plt.ylabel('y')  
plt.title('Box Plot')  
plt.show()
```



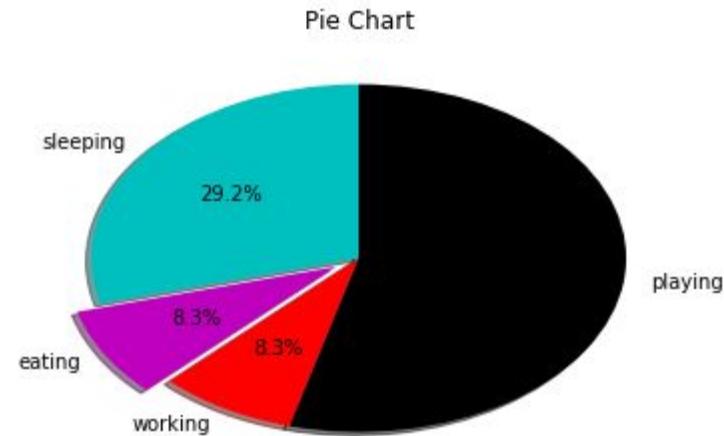
# Exercise #2



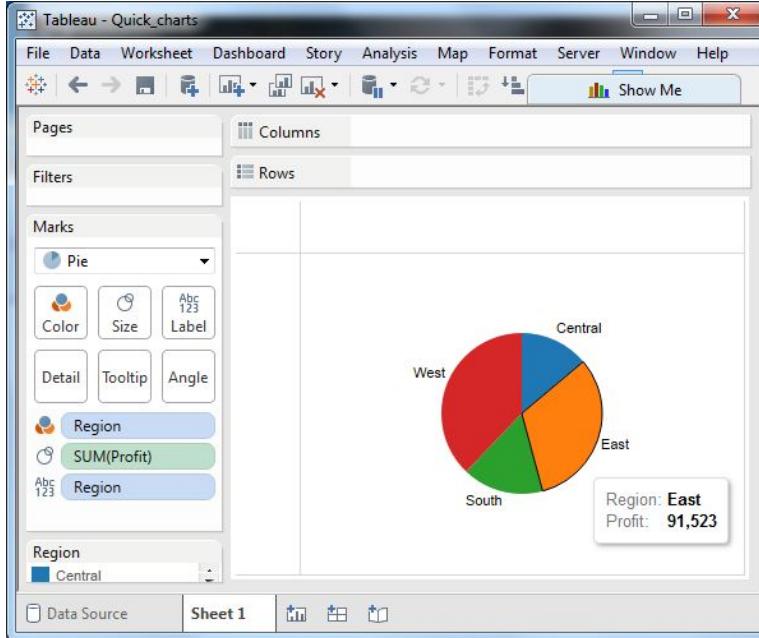
- [https://www.tutorialspoint.com/tableau/tableau\\_box\\_plot.htm](https://www.tutorialspoint.com/tableau/tableau_box_plot.htm)
- 카테고리, 쉽모드를 열어,  
프라핏을 행에 넣은후 박스플랏을  
선택

# Pie Chart

```
slices = [7, 2, 2, 13]
activities =
['sleeping', 'eating', 'working', 'playing']
cols = ['c', 'm', 'r', 'k']
plt.pie(slices,
        labels=activities,
        colors = cols,
        startangle=90,
        shadow=True,
        explode=(0,0.1,0,0),
        autopct='%.1f%%')
plt.title('Pie Chart')
plt.show()
```

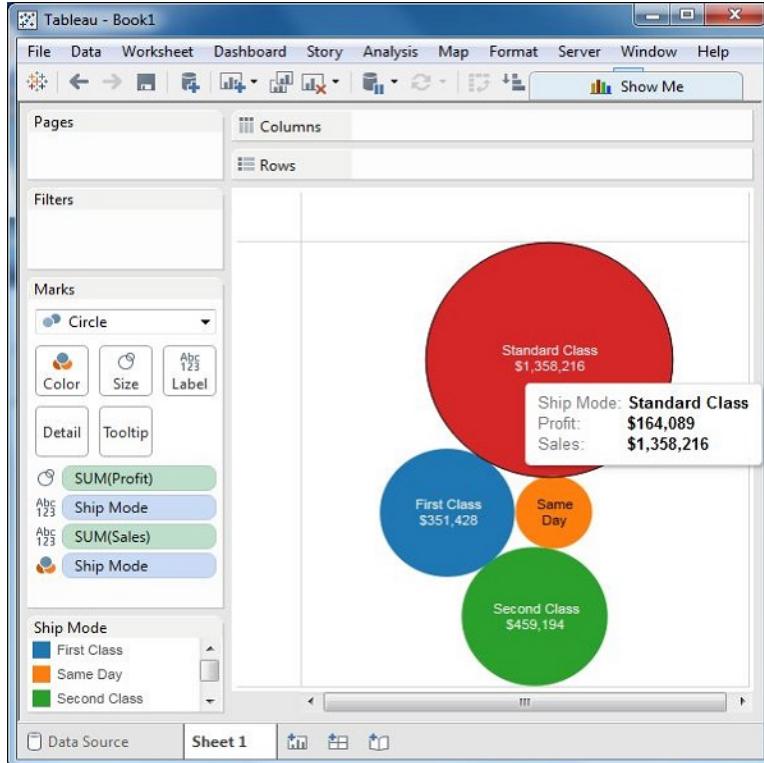


# Exercise #2



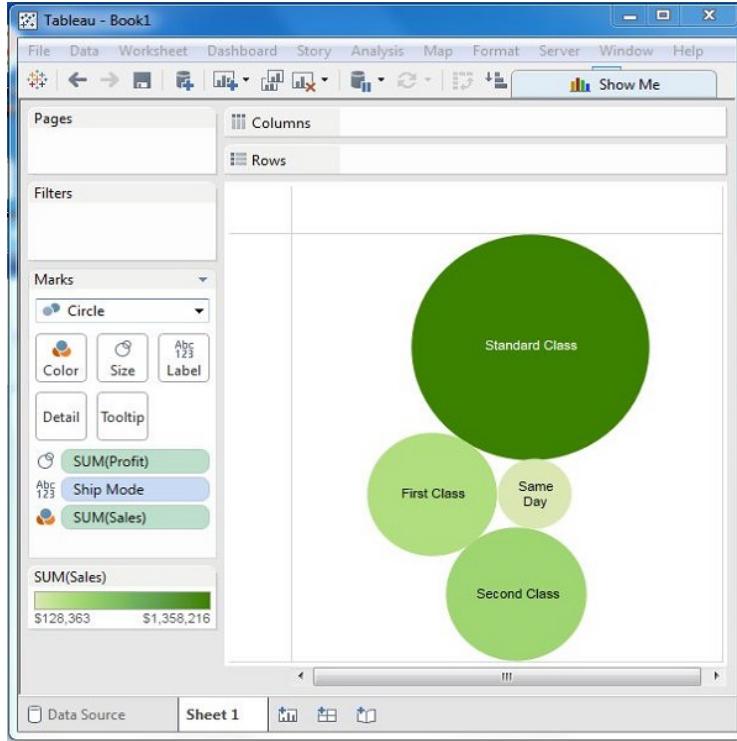
- [https://www.tutorialspoint.com/tableau/tableau\\_pie\\_chart.htm](https://www.tutorialspoint.com/tableau/tableau_pie_chart.htm)
- **프라핏을 사이즈에 넣고 파이차트 체크! 지역을 칼라로 넣어주고 지역을 레이블로 줌**

# Bubble Chart



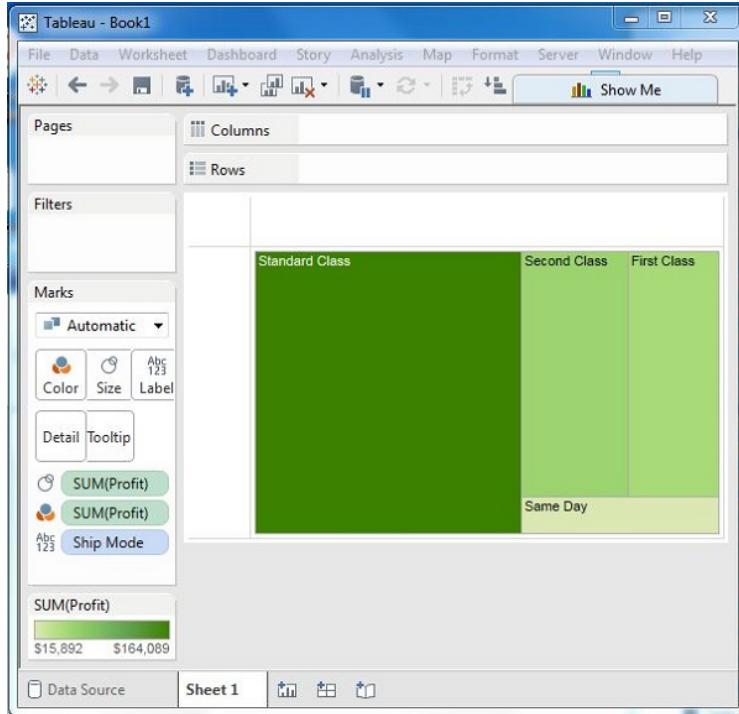
- Can plot a combination of categorical and continuous variables  
(범주형 변수와 연속변수의 관계를 보여줄때)

# Exercise #2



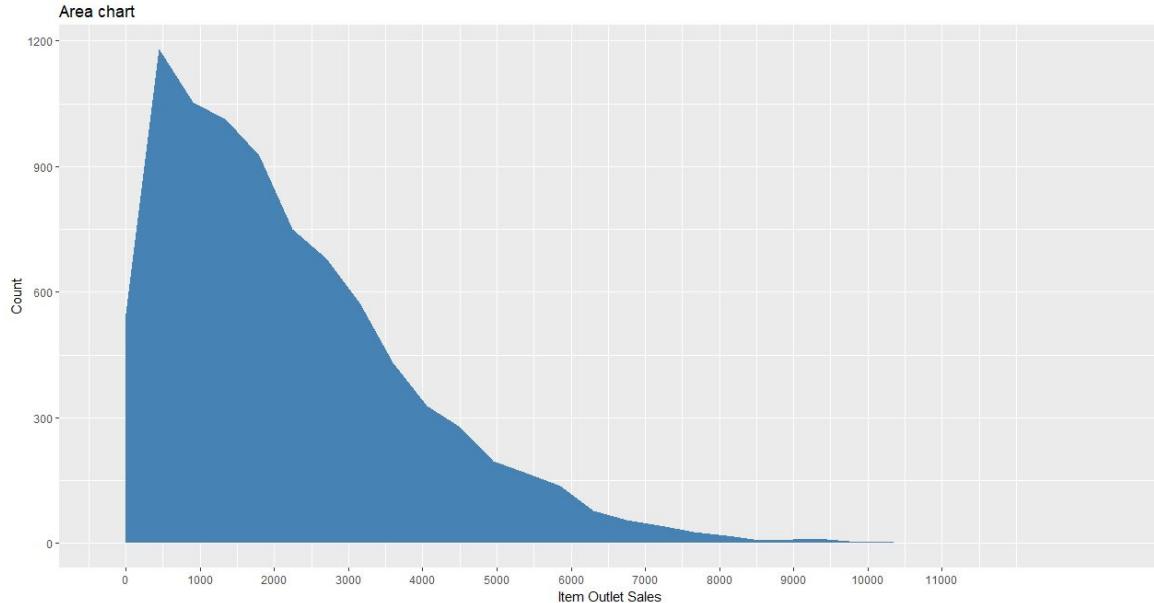
- [https://www.tutorialspoint.com/tableau/tableau\\_bubble\\_chart.htm](https://www.tutorialspoint.com/tableau/tableau_bubble_chart.htm)
- 칼라를 바꿀때는 칼라 □ Edit Color □ Palette를 바꿈

# Tree Map



- [https://www.tutorialspoint.com/tableau/tableau\\_tree\\_map.htm](https://www.tutorialspoint.com/tableau/tableau_tree_map.htm)

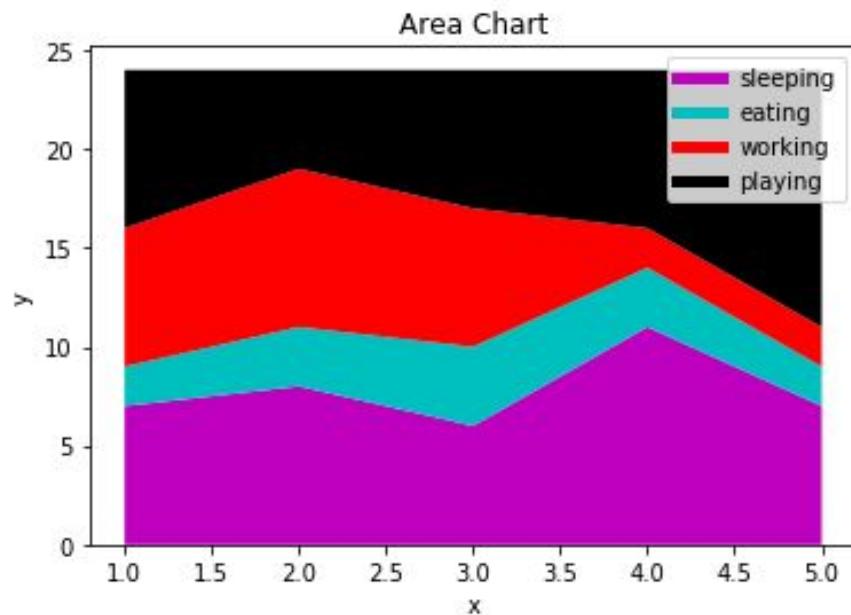
# Area Chart



- Used to show continuity across a variable or data set  
(연속변수의 빈도를 나타낼 때)

# Example

```
days = [1, 2, 3, 4, 5]
sleeping = [7, 8, 6, 11, 7]
eating = [2, 3, 4, 3, 2]
working = [7, 8, 7, 2, 2]
playing = [8, 5, 7, 8, 13]
plt.stackplot(days, sleeping,
              eating, working, playing,
              colors=['m', 'c', 'r', 'k'])
plt.xlabel('x')
plt.ylabel('y')
plt.title('Area Chart')
plt.legend()
plt.show()
```



# Legends

```
plt.plot([],[],color='m',
          label='sleeping',
          linewidth=5)

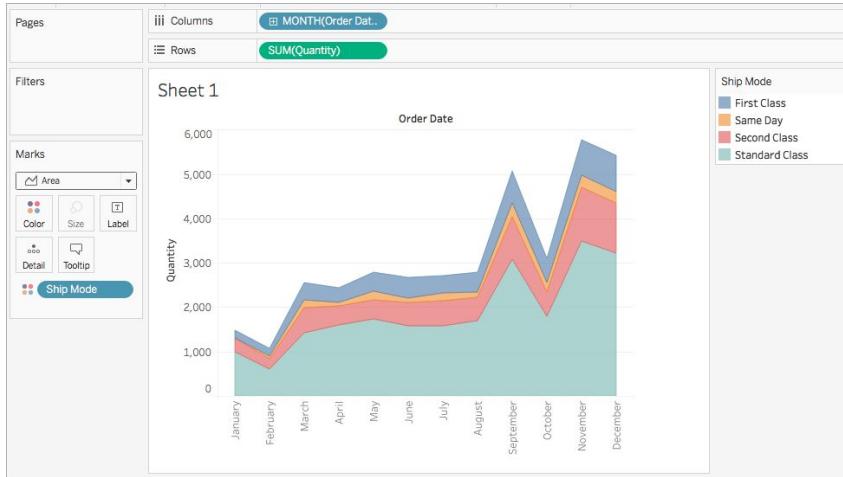
plt.plot([],[],color='c',
          label='eating',
          linewidth=5)

plt.plot([],[],color='r',
          label='working',
          linewidth=5)

plt.plot([],[],color='k',
          label='playing',
          linewidth=5)
```

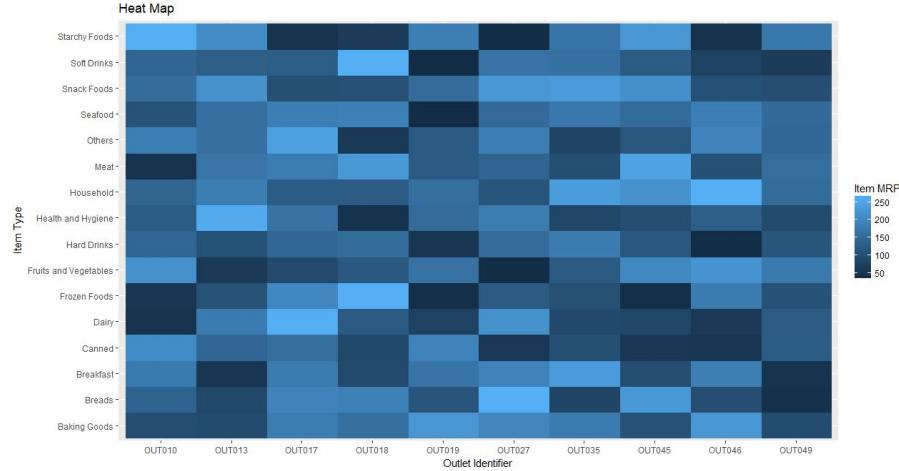
- Place this code any places above legend()  
범례함수전에 코드를 집어 넣어야 함

# Exercise #2



- [https://help.tableau.com/current/pro/desktop/en-us/qs\\_area\\_charts.htm](https://help.tableau.com/current/pro/desktop/en-us/qs_area_charts.htm)

# Heat Map

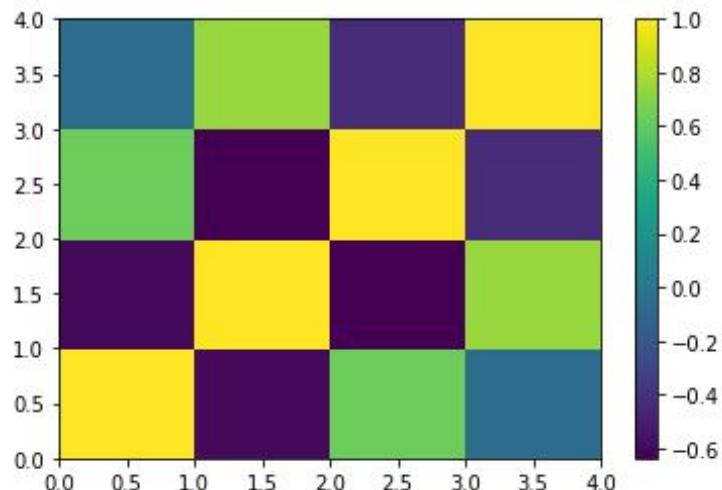


- Uses intensity of colors to display relationship between two or three or many variables in a two dimensional image (2개이상 변수의 관계를 색의 진하기로 나타냄)

Explores two dimensions as the axis and the third dimension by intensity of color 2개의 변수는 각각의 축으로, 3번째 변수는 색의 진하기 정도로

# Example

```
data=[[2,3,4,1],[6,3,5,2],[6,3,5,4]
],[3,7,5,4],[2,8,1,5]] #5x4 matrix  
Index= ['I1', 'I2','I3','I4','I5']  
Cols = ['C1', 'C2', 'C3','C4'] #4  
variables  
  
df = pd.DataFrame(data,  
index=Index, columns=Cols)  
  
plt.pcolor(df.corr())  
#plt.matshow(df.corr())  
  
plt.colorbar()  
  
plt.show()
```

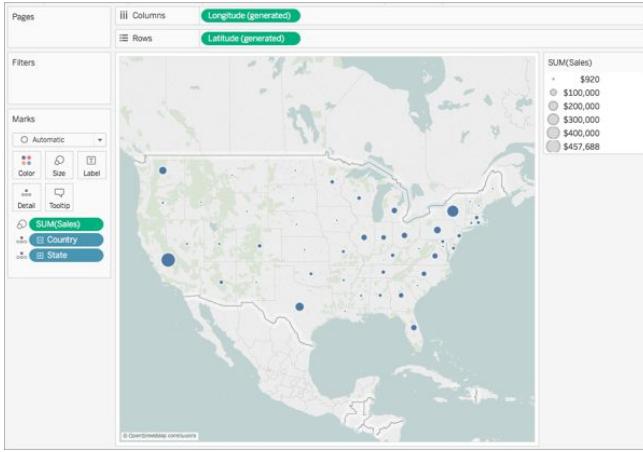


Positive co-relations are displayed in yellow and negative correlations in dark blue color 양의 관계는 노란색, 음의 관계는 어두운 파란색

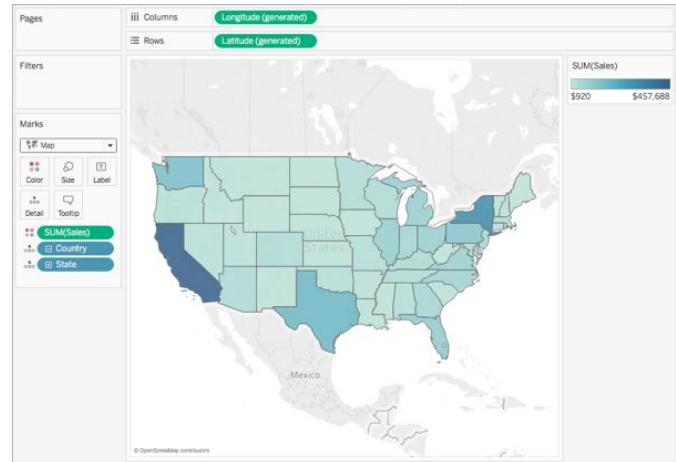
# Exercise #2

- Complete the map tutorial on  
[https://help.tableau.com/current/pro/desktop/en-gb/maps\\_howto\\_simple.htm](https://help.tableau.com/current/pro/desktop/en-gb/maps_howto_simple.htm)

Point Map



Polygon Map



# Exercise #2

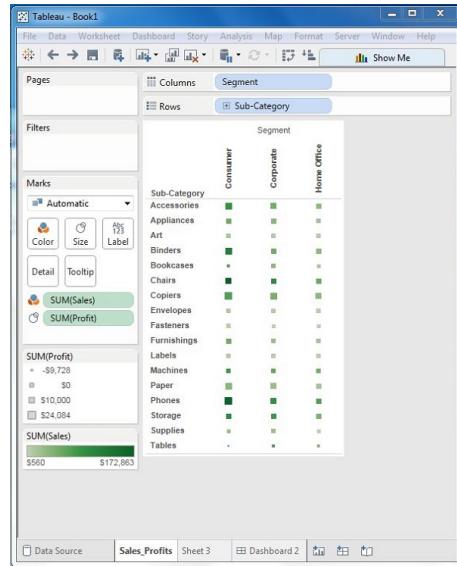
## Chipotle Example

- Analyze the Chipotle data using graphs
  - Scatter plot
  - Histogram
  - Bar chart, horizontal bar chart, stacked bar chart
  - Box plot
  - Area chart
  - Heat map (if possible)

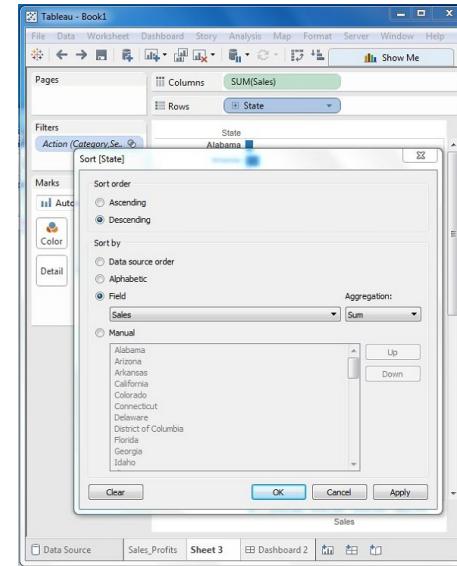
# Exercise #2

[https://www.tutorialspoint.com/tableau/tableau\\_dashboard.htm](https://www.tutorialspoint.com/tableau/tableau_dashboard.htm)

- Sales\_Profits Worksheet



- Sales\_state Worksheet



# Exercise #2

- Profit\_Dashboard



- 대시보드를 만든 후 필터를 이용하여 어떻게 데이터가 연결되는지 확인하시요

# Exercise #2

## Occupation Example

- Analyze the Occupation data using graphs
  - Scatter plot
  - Histogram
  - Bar chart, horizontal bar chart, stacked bar chart
  - Box plot
  - Area chart
  - Heat map (if possible)
  - Point map/polygon map (if possible)
- Create a dashboard

# Exercise #2

## World Food Facts

- Analyze the world food fact data using graphs
  - Scatter plot
  - Histogram
  - Bar chart, horizontal bar chart, stacked bar chart
  - Box plot
  - Area chart
  - Heat map (if possible)
  - Point map/polygon map (if possible)

<https://www.kaggle.com/openfoodfacts/world-food-facts/data>

# Database

# Spreadsheet vs. Database

Spreadsheet

Data  
Entry

Database

Data  
Structure



Data  
Entry

# Difference between Spreadsheet and Database

Common Uses	
Excel	Access
data analysis	data management
mathematical, logical	display data subsets, data structuring, complex queries
calculations, statistical comparisons	automation of common events
easy to generate sharable output	database management with multiple users
conditional formatting, chart management	reports for data summarisation

데이터분석, 수학, 로직,  
계산, 통계, 조건포맷, 차트

데이터관리, 데이터 구조화, 복잡한  
질의, 자동화, 다중사용자, 리포트

# Tables

Clients

Client ID	Client First Name	Client Last Name	Client City	<< other fields >>
9001	Stewart	Jameson	Seattle	.....
9002	Shannon	McLain	Poulsbo	.....
9003	Estela	Pundt	Tacoma	.....
9004	Timothy	Ennis	Seattle	.....
9005	Marvin	Russo	Bellingham	.....
9006	Kendra	Bonnicksen	Tacoma	.....

*Fields*

*Records*

- A record is composed of fields and contains all the data about one particular person, company, or item in a database.  
필드로 구성되어 있고, DB안에 있는 각각의 사람, 회사, 아이템등이 레코드임
- A field is part of a record and contains a single piece of data for the subject of the record.  
레코드의 일부분으로 레코드에 대한 속성을 포함하고 있음

# Database

- Collection of information that you organize and access according to the logical structure of the information

논리적 구조를 갖는 정보  
(테이블)를 모아놓은 것이  
데이터베이스

ORDER FILE							
Order Number	Order Date	Customer Number	Delivery Address	Concrete Type	Amount	Truck Number	Driver ID
100000	9/1/2004	1234	55 Smith Lane	1	8	111	123456789
100001	9/1/2004	3456	2122 E. Biscayne	1	3	222	785934444
100002	9/2/2004	1234	55 Smith Lane	5	6	222	435296657
100003	9/3/2004	4567	1333 Burr Ridge	2	4	333	435296657
100004	9/4/2004	4567	1333 Burr Ridge	2	8	222	785934444
100005	9/4/2004	5678	1222 Westminster	1	4	222	785934444
100006	9/5/2004	1234	222 East Hampton	1	4	111	123456789
100007	9/6/2004	2345	9 W. Palm Beach	2	5	333	785934444
100008	9/6/2004	6789	4532 Lane Circle	1	8	222	785934444
100009	9/7/2004	1234	987 Furlong	3	8	111	123456789
100010	9/9/2004	6789	4532 Lance Circle	2	7	222	435296657
100011	9/9/2004	4567	3500 Tomahawk	5	6	222	785934444

CUSTOMER FILE			
Customer Number	Customer Name	Customer Phone	Customer Primary Contact
1234	Smelding Homes	3333333333	Bill Johnson
2345	Home Builders Superior	3334444444	Marcus Connolly
3456	Mark Akey	3335555555	Mark Akey
4567	Triple A Homes	3336666666	Janille Smith
5678	Sheryl Williamson	3337777777	Sheryl Williamson
6789	Home Makers	3338888888	John Yu

CONCRETE TYPE FILE		
Concrete Type	Type	Description
1		Home foundation and walkways
2		Commercial foundation and infrastructure
3		Premier speckled (concrete with pea-size smooth gravel aggregate)
4		Premier marble (concrete with crushed marble aggregate)
5		Premier shell (concrete with shell aggregate)

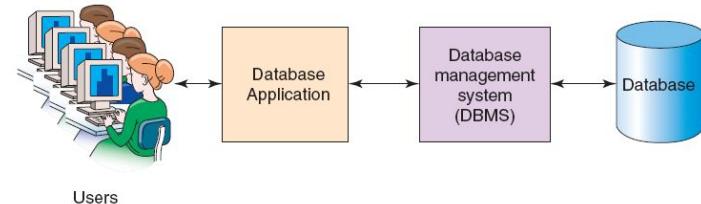
EMPLOYEE FILE			
Employee ID	Employee Last Name	Employee First Name	Date of Hire
123456789	Johnson	Emilio	2/1/1985
435296657	Evarez	Antonio	3/3/1992
785934444	Robertson	John	6/1/1999
984568756	Smithson	Allison	4/1/1997

TRUCK FILE		
Truck Number	Truck Type	Date of Purchase
111	Ford	6/17/1999
222	Ford	12/24/2001
333	Chevy	1/1/2002

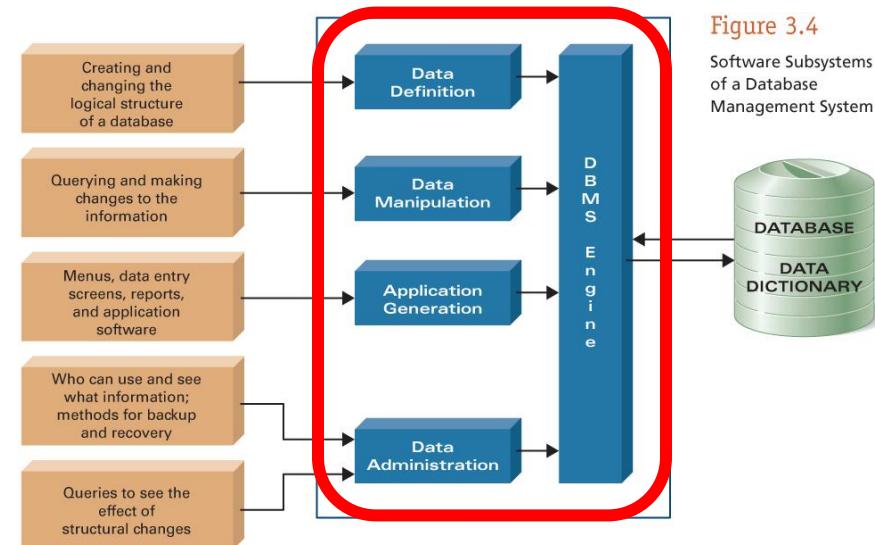
# Database Management System (DBMS)

- A software system designed to create, store, manage, and facilitate access to databases  
데이터베이스를 만들고, 저장하고, 관리할 수 있는 소프트웨어
- Serves as an intermediary between database applications and the database.  
데이터베이스와 어플리케이션 사이에서 중개하는 역할
- The DBMS also manages and controls database activities.  
데이터베이스 사용자 활동도 관리할 수 있음



# 5 Components of a DBMS

- It serves as an intermediary between database applications and the database
- 실제 데이터베이스와  
어플리케이션 사이를  
중재



# Data Definition Subsystem

- You create and maintain the data dictionary and structure of the files in a database  
데이터베이스 안에 있는 데이터의 정의 및 구조를 정함
- The data dictionary helps you define:
  - Field names 필드이름
  - Data types (numeric, etc) 데이터타입
  - Form (do you need an area code) 형식
  - Default value 기본값 설정
  - Is an entry required, etc 필수항목 등

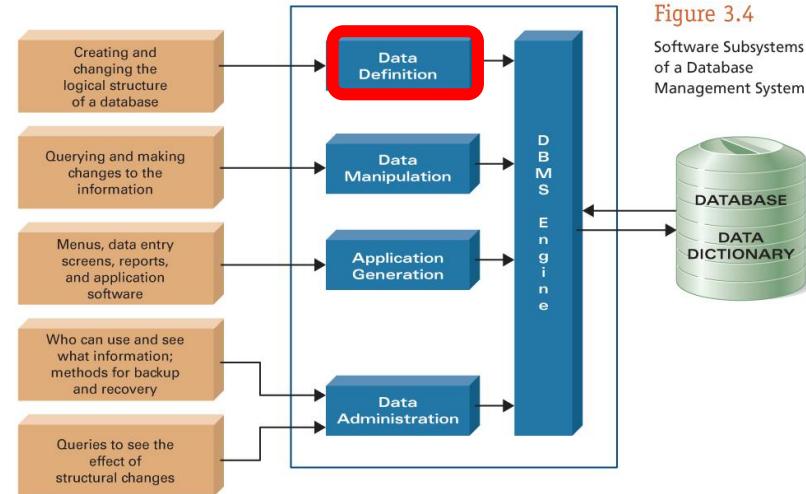
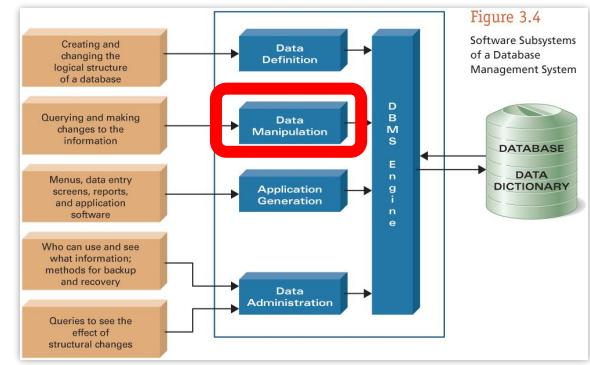


Figure 3.4

Software Subsystems  
of a Database  
Management System

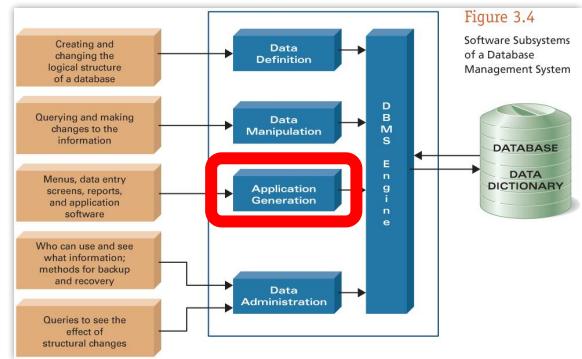
# Data Manipulation Subsystem

- You can add, change, and delete information in a database and query it to find valuable information  
데이터베이스 안에 데이터를 더하거나 바꾸거나 지울 수 있고 검색할 수도 있음
- Includes views, report generators, query-by-example tools, and structured query language 뷰, 리포트, QBE, SQL을 포함



# Application Generation Subsystem

- Contains facilities to help you develop transaction-intensive applications  
어플리케이션을 만들수 있음
- Mainly used by IT professionals  
주로 IT담당자가 만듬

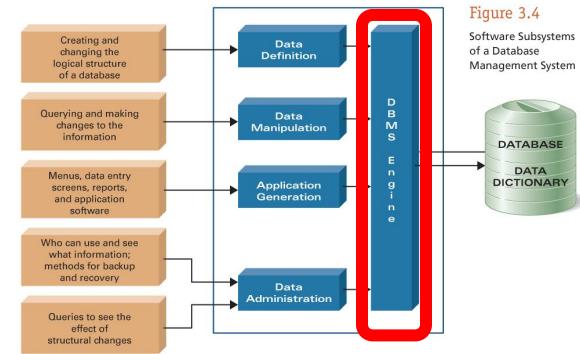


# Data Administration Subsystem

- You can manage the overall database environment by providing facilities for:
  - Backup and recovery 백업 및 복구
  - Security management 보안관리
  - Query optimization - minimizes response times for large, complex queries 쿼리최적화
  - Reorganization - for physically rearranging the structure of the information 재조직
  - Concurrency control - what happens if two people attempt to make changes to the same record 동시에 두사람이 같은 레코드를 바꿀려고 할때
  - Change management - how will structural changes impact the overall database 바꾼것을 관리

# DBMS Engine

- Accepts logical requests and converts them into the physical equivalents, and access the database and data dictionary on a storage device **논리적 요청을 받아서 물리적 요청으로 변환**
  - Physical view – how information is physically arranged, stored, and accessed on a storage device **물리적으로 어떻게 저장됐고 접근할수있나**
  - Logical view – how you need to arrange and access information to meet your needs  
**필요에 따라 정보를 정렬, 접근**



	A	B	C	D	E	F	G
1	LastName	FirstName	Email	AdviserLastName	AdviserEmail	Department	AdminLastName
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
5	Hwang	Terry	Terry.Hwang@ourcampus.edu	Tang	Susan.Tang@ourcampus.edu	Chemistry	Smith
6	Li	Tzu	Tzu.Li@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
7	Manno	Chip	Chip.Manno@ourcampus.edu	Tran	Ken.Tran@ourcampus.edu	InfoSystems	Rogers
8	Thompson	James	James.Thompson@ourcampus.edu	Tang	Susan.Tang@ourcampus.edu	Accounting	Smith
9	???	???	???	???	???	Biology	Kelly

# Problems with DBMS

Redundancy  
반복성

If a particular person is managing 10 projects, his/her information would have to be entered 10 times. 어떤 특정한 사람이 10개의 프로젝트에서 일한다면 그 사람에 대한 정보를 10번 입력해야 함

Multiple Themes  
여러개의 주제가  
한꺼번에 들어있음

A list of Projects may include Project Manager information and Project information in the same row. 프로젝트 리스트에 프로젝트 관리자 정보와 프로젝트 정보가 한줄에 같이 들어감

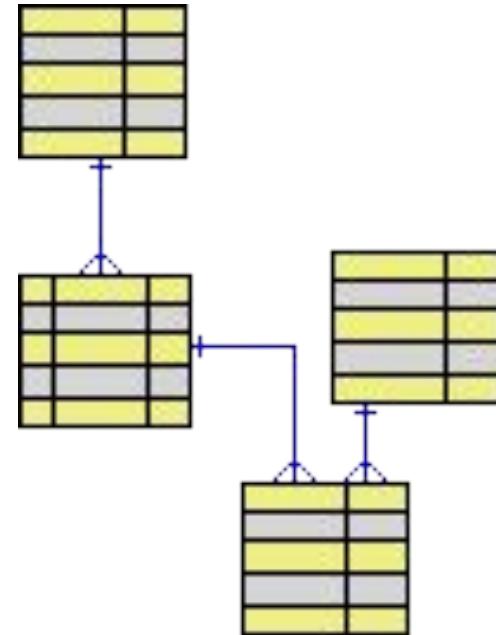
Modification Issues  
수정문제

Deletion problems (data lost), insertion problems (data missing), update problems (need to update all related data multiple times) 데이터를 지우다가 데이터가 사라질 수 있고,  
입력하다가 데이터가 빠질 수 있고, 업데이트 할 때 반복해서 내용을 넣어야 함

# Relational Database

## 관계형 데이터베이스

- Most popular database model  
가장 많이 쓰이는 데이터베이스 모델
- Series of logically related two-dimensional tables or files for storing information  
논리적으로 연관된 2차원 테이블들이나 파일들의 집합
- Tables within a relational database are related to each other.  
관계형 데이터베이스안에 있는 테이블들은 서로 연관되어 있음

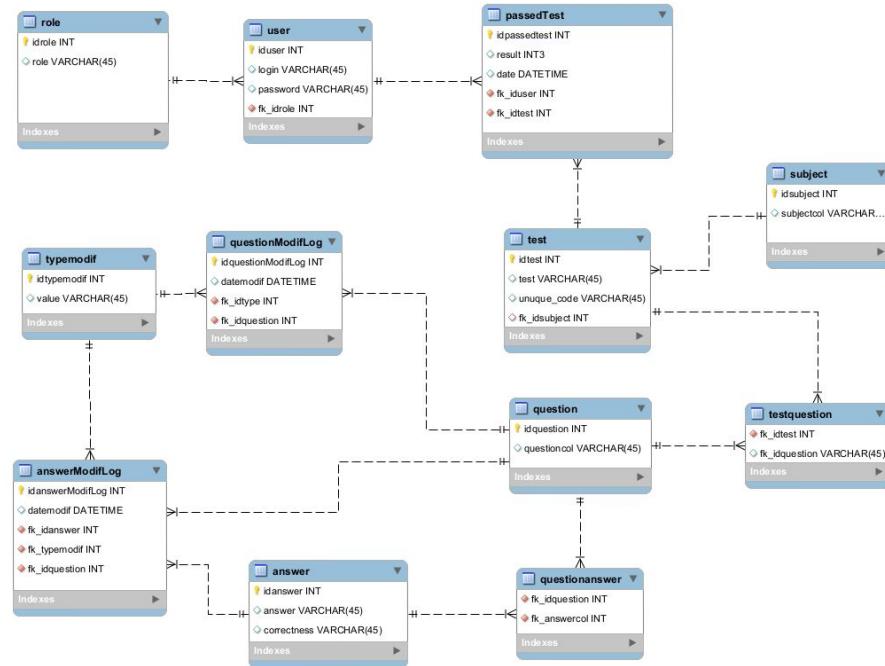


# RDB Example

- For example, a Project List can be divided into a CUSTOMER Table, a PROJECT Table, and a PROJECT\_MANAGER Table.  
프로젝트 리스트는 고객 테이블, 프로젝트 테이블, 프로젝트관리자 테이블로 나눌수 있음
- A relational database breaks up a list into several tables—one table for each theme to minimize data redundancy while preserving complex relationships among themes.  
리스트를 하나의 리스트를 여러개의 테이블로 쪼갬. 데이터중복을 최소화하기 위해 각 테이블에는 한가지 주제만 집어 넣고 관계로 연결

# Relational Database Management Systems (RDBMS)

- An application or collection of applications designed to create, store, manage, and facilitate access to relational databases  
관계형 데이터베이스를 생성, 저장, 관리하는 어플리케이션
- The RDBMS also manages and controls database activities.  
사용자의 데이터베이스 사용활동 관리



# Primary Key

## 기본키

- Field (or group of fields) that uniquely describes each record  
각 레코드를 구분하게 해주는 필드
- Order Number is the primary key for Order and Customer Number is the primary key for Customer  
주문테이블에는 주문번호,  
고객테이블에는 고객번호 등

ORDER FILE								
Order Number	Order Date	Customer Number	Delivery Address	Concrete Type	Amount	Truck Number	Driver ID	
100000	9/1/2004	1234	55 Smith Lane	1	8	111	123456789	
100001	9/1/2004	3456	2122 E. Biscayne	1	3	222	785934444	
100002	9/2/2004	1234	55 Smith Lane	5	6	222	435296657	
100003	9/3/2004	4567	1333 Burr Ridge	2	4	333	435296657	
100004	9/4/2004	4567	1333 Burr Ridge	2	8	222	785934444	
100005	9/4/2004	5678	1222 Westminster	1	4	222	785934444	
100006	9/5/2004	1234	222 East Hampton	1	4	111	123456789	
100007	9/6/2004	2345	9 W. Palm Beach	2	5	333	785934444	
100008	9/6/2004	6789	4532 Lane Circle	1	8	222	785934444	
100009	9/7/2004	1234	987 Furlong	3	8	111	123456789	
100010	9/9/2004	6789	4532 Lance Circle	2	7	222	435296657	
100011	9/9/2004	4567	3500 Tomahawk	5	6	222	785934444	

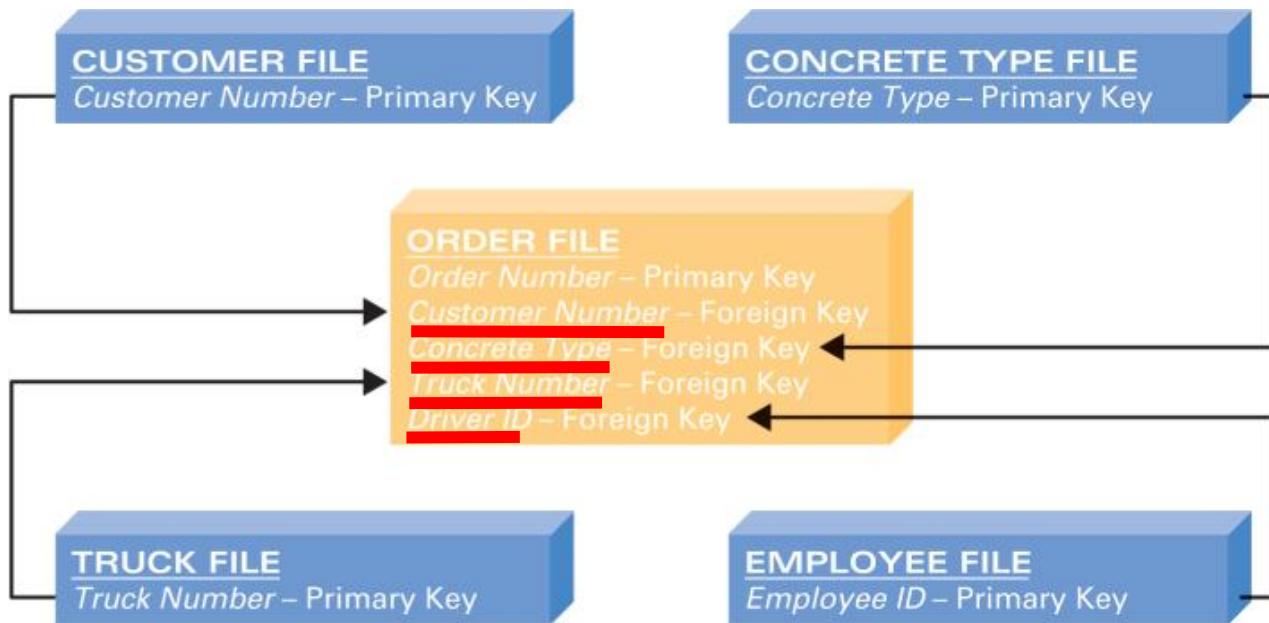
  

CUSTOMER FILE			
Customer Number	Customer Name	Customer Phone	Customer Primary Contact
1234	Smelding Homes	3333333333	Bill Johnson
2345	Home Builders Superior	3334444444	Marcus Connolly
3456	Mark Akey	3335555555	Mark Akey
4567	Triple A Homes	3336666666	Janielle Smith
5678	Sheryl Williamson	3337777777	Sheryl Williamson
6789	Home Makers	3338888888	John Yu

# Foreign Key

## 외래키

- Primary key of one file that appears in another file  
기본키가 다른 테이블에 나타나면 외래키
- Customer Number appears in Order as a foreign key  
주문테이블에 있는 고객번호는 외래키



# Integrity Constraints

- Rules that help ensure the quality of information  
정보의 질을 유지하기 위한 방법
- Data dictionary, for example, defines type of information – numeric, date, and so on  
데이터 정의어를 이용하여 데이터 타입을 정함. 데이터 타입이 안맞으면 에러!!!
- Foreign keys – must be found as primary keys in another file  
외래키로 사용하기 위해서는 다른 테이블의 기본키여야 함
  - E.g., a Customer Number in the Order Table must also be present in the Customer Table  
고객테이블에 고객넘버가 있어야 주문테이블에서 고객넘버를 사용할수 있음

# Exercise #2

- Import XLMD\_Customer.xls and create queries to answer the following questions:

**xlmd\_customer파일을  
임포트한후에 다음 질문에  
답하시요**

- Q1. Open Show all information for only customers who live in the North region.

**북쪽에 사는 고객들의 정보를  
보여주시요**

필드:	CUST ID	REGION	RENT VS# OWN	NUM HOUSEHOLD	ANNUAL INCOME
테이블:	customer	customer	customer	customer	customer
정렬:					
표시:	<input checked="" type="checkbox"/>				
조건:		"North"			

Run!!

Q1	CUST ID	REGION	RENT VS# OWN	NUM HOU!	ANNUAL IN	TOTAL PUR	NUM PURC
	1	North	Own	1	10K-25K	\$38	1
	3	North	Rent	3	25K-50K	\$19	3
	9	North	Own	5	50K-100K	\$30	8
	11	North	Own	1	50K-100K	\$21	4
		North	Own	6	50K-100K	\$129	5
	20	North	Rent	1	50K-100K	\$119	3
	24	North	Rent	6	10K-25K	\$129	7
	28	North	Own	4	50K-100K	\$115	8
	34	North	Own	6	100K+	\$45	3
	35	North	Own	5	10K-25K	\$29	1
	41	North	Own	7	25K-50K	\$19	3
	50	North	Own	6	25K-50K	\$30	4
	53	North	Rent	1	25K-50K	\$40	5

# Exercise #2

- Q2. Show all information for only customers who (a) live in the North region, (b) own their homes, and (c) have only one household member.

북쪽에 살면서 자기집을  
소유하고, 1명의  
가정구성원을 가진 고객의  
정보를 보여주시요

필드:	CUST ID	REGION	RENT VS# OWN	NUM HOUSEHOLD
테이블:	customer	customer	customer	customer
정렬:				
표시:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
조건:		"North"	"OWN"	1



CUST ID	REGION	RENT VS# OWN	NUM HOU:	ANNUAL IN	TOTAL PUR	NUM PURC
11 North	Own	1	10K-25K	\$38	1	4
348 North	Own	1	50K-100K	\$21	6	
484 North	Own	1	50K-100K	\$29	6	
		1	25K-50K	\$61	6	

# Exercise #2

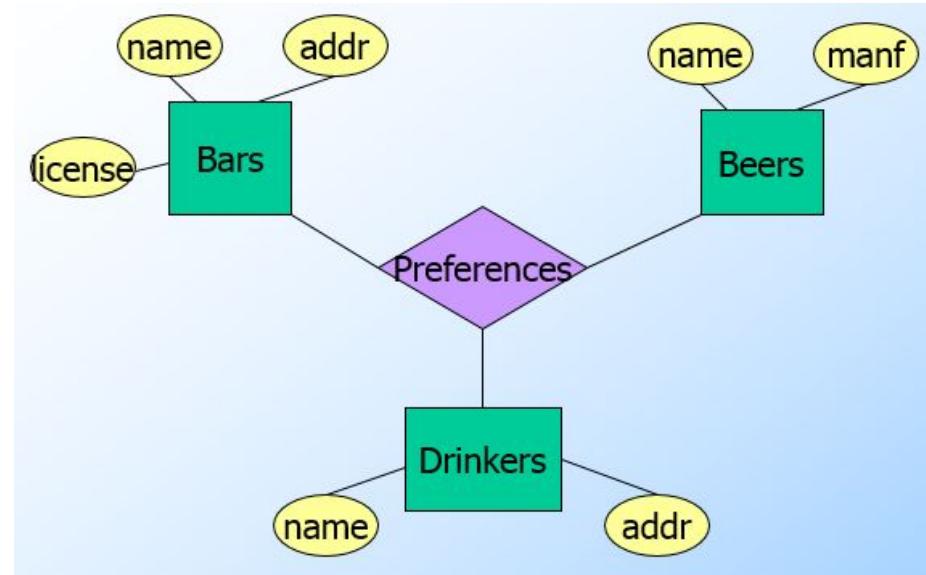
- Q3. Show all information for customers who have at least 4 household members.  
최소 4명의 가족구성원을 가진 고객들의 정보를 보여주시요
- Q4. Show all information for customers who (a) have spent less than \$20 or (b) more than \$100.  
20불미만 100불이상 소비한 고객들의 정보를 보여주시요
- Q5. Provide a two-dimensional table using crosstab that counts the number of customers by the categories of REGION and RENT VS. OWN.  
크로스탭을 이용하여 2차원 테이블을 만들되 지역과 렌트대소유에 따른 고객들의 수를 보여주시요

# Exercise #2

- Q6. Provide a two-dimensional table using crosstab that sums the TOTAL PURCHASES of customers by the categories of REGION and RENT VS OWN.  
고객들의 총구매액의 합을 보여주세요  
크로스탭을 이용하여 2차원 테이블을 만들되 지역과 렌트대소유에 따른
- Q7. Provide a query using Totals that counts the number of customers by the categories of REGION, RENT VS OWN, and NUM HOUSEHOLD.  
고객들의 수를 보여주세요  
토탈스를 이용하여 쿼리를 만들되 지역, 렌트대소유, 가족구성원의 수에 따른
- Q8-Q10. Create three more queries to answer your questions on this database.  
추가적으로 3개 쿼리를 더 만드시요

# ER Diagram

- Entity-Relationship
- Widely used in database design  
데이터베이스 디자인에  
많이 사용됨
- Represent conceptual level  
of a database system  
가장 높은 레벨에서의  
디자인



# Basic Concepts in ER Diagram

- Entity – “thing” or object, e.g., student  
어떤 개체, 예를 들어 학생
- Attribute – property of an entity, e.g. student id, student name, student email, etc.  
개체의 속성, 예를 들어 학생아이디, 이름, 이메일 등
- Relationship – relations among entities from two or more entities  
두개이상의 개체사이의 관계

# Relationships

---

## One-to-one relationship

A record in one table has only one matching record in the second table.  
한 테이블에 있는 한 레코드가 다른 테이블에 있는 한 레코드와 매치

## One-to-many relationship

One record in the first table corresponds to many records in the second table.  
한 테이블에 있는 한 레코드가 다른 테이블에 있는 여러 레코드와 매치

## Many-to-man y relationship

One record in one table has many matching records in a second table, and one record in the related table has many matching records in the first table.  
한 테이블에 있는 한 레코드가 다른 테이블에 있는 여러 레코드와 매치되고 다른 테이블에 있는 한 레코드도 이 테이블에 있는 여러개의 레코드와 매치될 때

---

# Types of ER Diagrams

## Information Engineering (IE) [James Martin 1990]

- Uses “crow’s feet” to show the many sides of a relationship, and it is sometimes called the crow’s foot model.

까마귀 발 모양을  
이용하여 그림

## Integrated Definition 1, Extended 3 (IDEF1X)

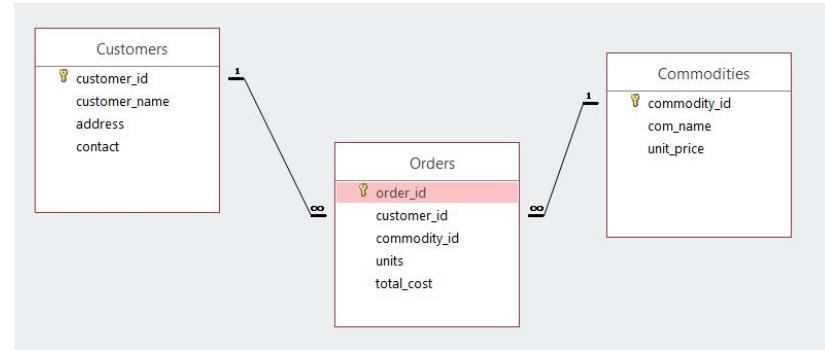
- A version of the E-R model that is a national standard.  
국가표준

## Unified Modeling Language (UML)

- A set of structures and techniques for modeling and designing object-oriented programs (OOP) and applications  
모델링하는 테크닉의  
하나인 **UML**에서  
사용하는 다이어글매

# Exercise #2

- Create tables for online shop and a relationship among these tables, and enter some data  
**온라인숍 테이블과 관계를 만든후 데이터를 입력하시요**
- Refer to Extended Learning Module J on page 473-487 (Implementing a database with MS Access)



# SQL

- SQL is Structured Query Language  
구조화된 질의어
  - Some people pronounce SQL as “sequel” 씨퀄이라고 발음할수도 있고
  - Other people insist that only “ess-cue-ell” is the only correct pronunciation  
에스큐엘이라고 발음해도 됨
- Just about every relational database supports SQL  
거의 모든 관계형 데이터베이스들은 SQL을 지원함

# SQL (cont.)

- An international standard (ANSI) for creating, processing, and querying databases and their tables.

데이터베이스와  
테이블을 만들고,  
처리하고, 질의하기 위한  
국제표준

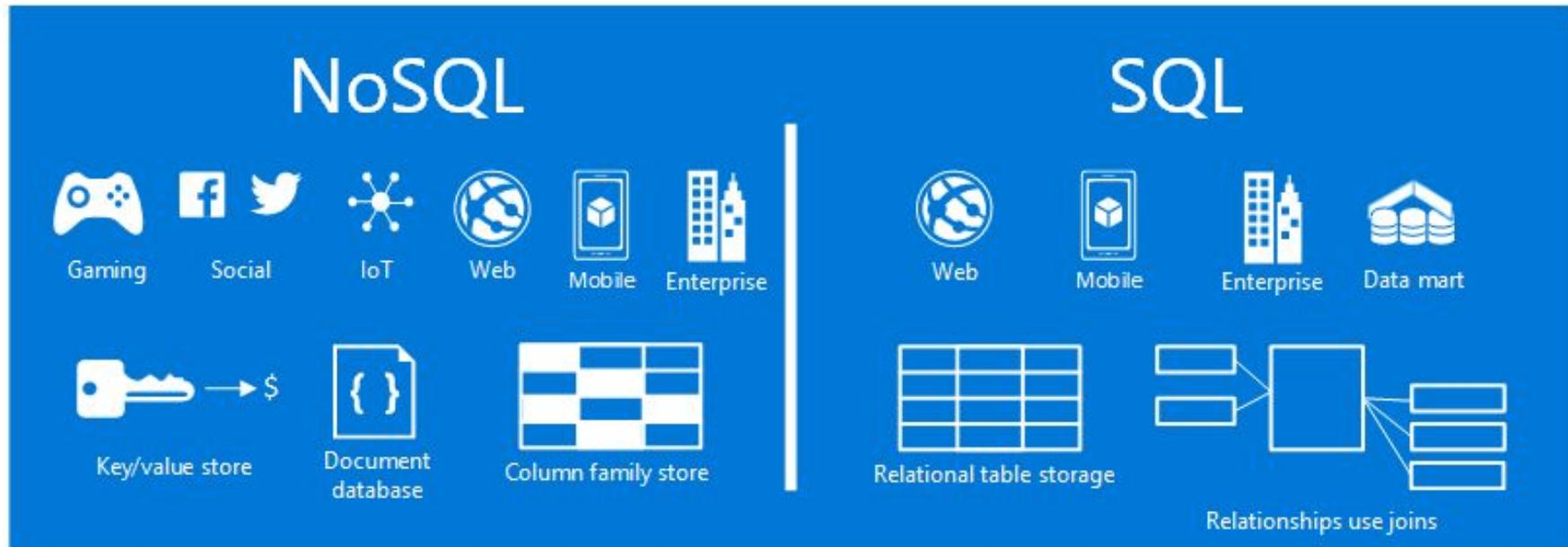
```
SELECT      CUSTOMER.CustomerLastName,
            CUSTOMER.CustomerFirstName,
            CUSTOMER.Phone,
            COURSE.CourseDate,
            ENROLLMENT.AmountPaid,
            COURSE.Course, COURSE.Fee
FROM        CUSTOMER, ENROLLMENT, COURSE
WHERE       CUSTOMER.CustomerNumber
            = ENROLLMENT.CustomerNumber
            AND COURSE.CourseNumber
            = ENROLLMENT.CourseNumber;
```

# NoSQL Database

- A database that does not use SQL. **SQL을 이용하지 않는 데이터베이스**
- It is a non-relational database **관계형 데이터베이스가 아님**
- Facebook and Twitter (Web 2.0 Applications) use Apache Software Foundation Cassandra database to quickly create and store massive amounts of data **페이스북, 트위터는 캐assandra 데이터베이스를 이용하여 많은 양의 데이터를 생성하고 저장함**

```
{
  _id: ObjectId('7df78ad8902c'),
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user:'user1',
      message: 'My first comment',
      dateCreated: new Date(2011,1,20,2,15),
      like: 0
    },
    {
      user:'user2',
      message: 'My second comments',
      dateCreated: new Date(2011,1,25,7,45),
      like: 5
    }
  ]
}
```

# SQL vs. NoSQL



# Commercial DBMS Products

## Desktop DBMS Products

- Microsoft Access

## Organizational DBMS Products

- Microsoft's SQL Server
- Oracle's Oracle
- Sun Microsystem's MySQL
- IBM's DB2

# Microsoft Access 2010

The database name **Art-Course-Database**

The table object **CUSTOMER** is displayed under the All Access Objects

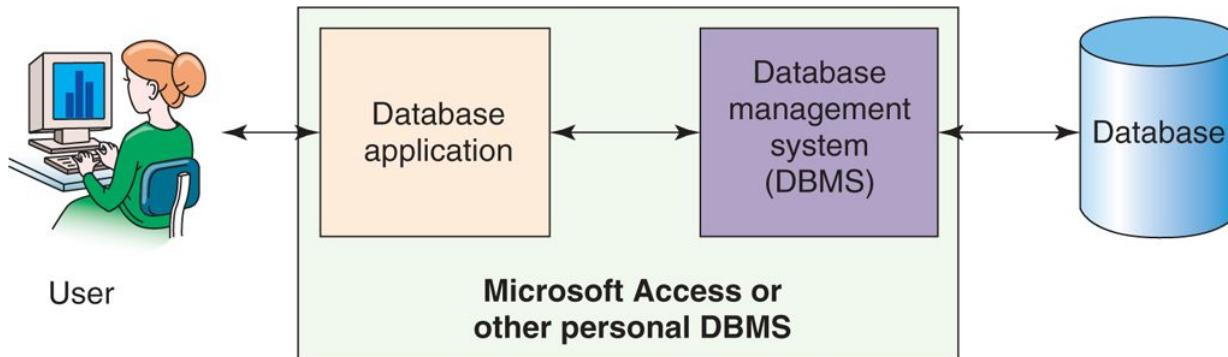
The query object Art Course List stores the query itself

The query results in table format

Art Course List

CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
Johnson	Ariel	206-567-1234	10/1/2015	\$250.00	Adv Pastels	\$500.00
Johnson	Ariel	206-567-1234	3/15/2015	\$350.00	Int Pastels	\$350.00
Green	Robin	425-678-8765	9/15/2015	\$350.00	Beg Oils	\$350.00
Jackson	Charles	360-789-3456	10/1/2015	\$500.00	Adv Pastels	\$500.00
Pearson	Jeffery	206-567-2345	10/1/2015	\$500.00	Adv Pastels	\$500.00
Sears	Miguel	360-789-4567	9/15/2015	\$350.00	Beg Oils	\$350.00
Kyle	Leah	425-678-7654	11/15/2015	\$250.00	Adv Pastels	\$500.00
Myers	Lynda	360-789-5678	10/15/2015	\$0.00	Beg Oils	\$350.00

# Personal Database Systems



# Microsoft SQL Server 2014

The screenshot shows the Microsoft SQL Server Management Studio interface. A tooltip box on the left contains five items:

- Click this button to run the SQL query
- The database object **Art-Course-Database** is displayed in the Object Explorer
- The table object **CUSTOMER** is displayed under the Art-Course-Database object
- The SQL query
- The query results in table format

The Object Explorer on the left shows the database structure, including the **Art\_Course\_Database** and its tables like **CUSTOMER**, **COURSE**, and **ENROLLMENT**. The SQL Query window in the center displays a SELECT statement:

```
SELECT CUSTOMER.CustomerLastName, CUSTOMER.CustomerFirstName, CUSTOMER.Phone, COURSE.CourseDate, ENROLLMENT.AmountPaid, COURSE.Course, COURSE.Fee
FROM CUSTOMER, ENROLLMENT, COURSE
WHERE CUSTOMER.CustomerNumber = ENROLLMENT.CustomerNumber
    AND COURSE.CourseNumber = ENROLLMENT.CourseNumber;
```

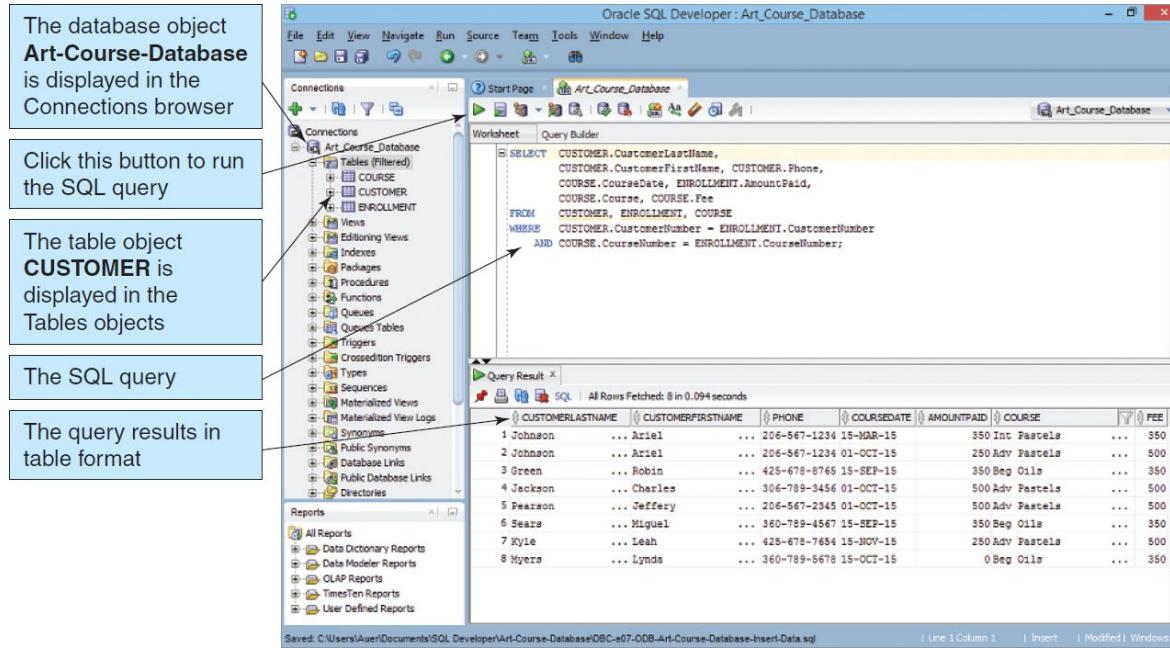
The Results pane on the right shows the query results in a table format:

	CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
1	Johnson	Angie	206-567-1234	2015-10-01	250.00	Adv Pastels	500.00
2	Johnson	Angie	206-567-1234	2015-03-15	350.00	Int Pastels	350.00
3	Green	Robin	425-678-8765	2015-09-15	350.00	Beg Oils	350.00
4	Jackson	Charles	306-789-3456	2015-10-01	500.00	Adv Pastels	500.00
5	Pearson	Jeffrey	206-567-2345	2015-10-01	500.00	Adv Pastels	500.00
6	Sears	Miguel	360-789-4567	2015-09-15	350.00	Beg Oils	350.00
7	Kyle	Leah	425-678-7654	2015-11-15	250.00	Adv Pastels	500.00
8	Myers	Lynda	360-789-5678	2015-10-15	0.00	Beg Oils	350.00

At the bottom, a status bar indicates: Ready, In 2, Col 37, Ch 31, INS.

# Oracle Database Express Edition 11g

## Release 2



# MySQL 5.6

The screenshot illustrates the MySQL Workbench interface for executing SQL queries. A tooltip at the top left indicates to "Click this button to run the SQL query". The SQL query itself is displayed in the central pane:

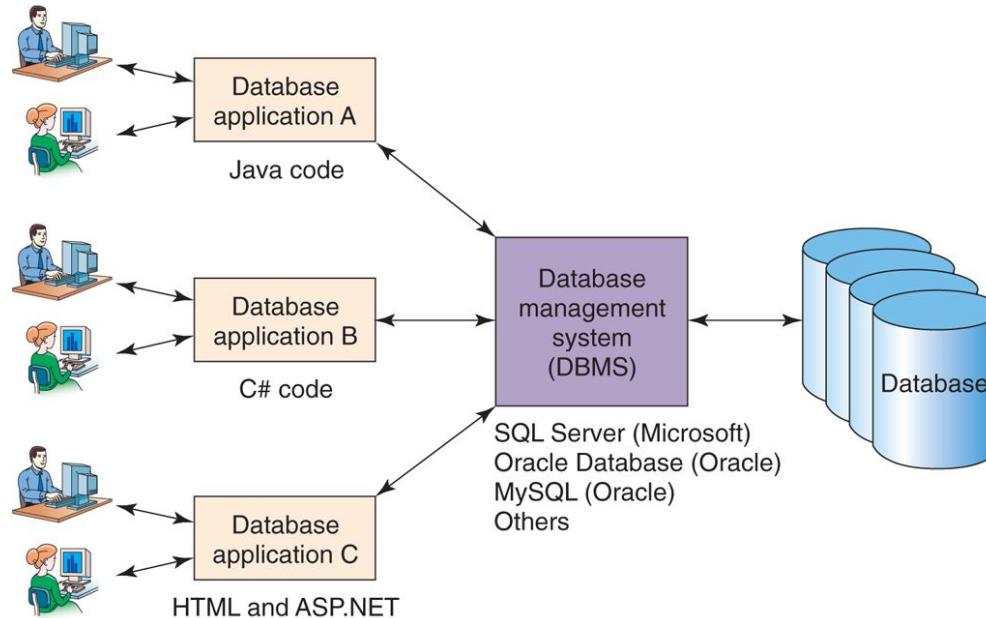
```
SELECT CUSTOMER.CustomerLastName,
       CUSTOMER.CustomerFirstName, CUSTOMER.Phone,
       COURSE.CourseDate, ENROLLMENT.AmountPaid,
       COURSE.Course, COURSE.Fee
  FROM CUSTOMER, ENROLLMENT, COURSE
 WHERE CUSTOMER.CustomerNumber = ENROLLMENT.CustomerNumber
   AND COURSE.CourseNumber = ENROLLMENT.CourseNumber;
```

Annotations provide details about the environment:

- The database object **art\_course\_database** is displayed in the Object Browser.
- The table object **CUSTOMER** is displayed under the art-course-database object.
- The query results are presented in a table format, showing data for various customers and their course enrollments.

CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
Johnson	Ariel	206-567-1234	2015-10-01	250.00	Adv Pastels	500.00
Jackson	Charles	306-789-3456	2015-10-01	500.00	Adv Pastels	500.00
Pearson	Jeffery	206-567-2345	2015-10-01	500.00	Adv Pastels	500.00
Green	Robin	425-678-9876	2015-09-15	350.00	Beg Oils	350.00
Sears	Miguel	360-789-4567	2015-09-15	350.00	Beg Oils	350.00
Johnson	Ariel	206-567-1234	2015-03-15	350.00	Int Pastels	350.00
Myers	Lynda	360-789-5678	2015-10-15	0.00	Beg Oils	350.00
Kyle	Leah	425-678-7654	2015-11-15	250.00	Adv Pastels	500.00

# Organizational Database Systems



# Database Engine Ranking

345 systems in ranking, March 2019

Rank			DBMS	Database Model	Score		
Mar 2019	Feb 2019	Mar 2018			Mar 2019	Feb 2019	Mar 2018
1.	1.	1.	Oracle 	Relational, Multi-model 	1279.14	+15.12	-10.47
2.	2.	2.	MySQL 	Relational, Multi-model 	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	469.81	-3.75	+70.46
5.	5.	5.	MongoDB 	Document	401.34	+6.24	+60.82
6.	6.	6.	IBM Db2 	Relational, Multi-model 	177.20	-2.23	-9.47
7.	↑ 9.	7.	Microsoft Access	Relational	146.20	+2.18	+14.26
8.	↓ 7.	8.	Redis 	Key-value, Multi-model 	146.12	-3.32	+14.90
9.	↓ 8.	9.	Elasticsearch 	Search engine, Multi-model 	142.79	-2.46	+14.25
10.	10.	↑ 11.	SQLite 	Relational	124.87	-1.29	+10.06

# Community Edition Databases

- IBM DB2 Express-C
  - Relational DB with XML support
  - Free edition of IBM DB2
  - <https://www.ibm.com>
- Oracle 11g Express Edition (XE)
  - Leading market share among commercial databases
  - XE is easier to administer than full Oracle
  - <http://www.oracle.com>

# Free RDBMS

- MySQL
  - One of the most popular open-source relational database management systems (RDBMS) commonly used in web applications  
웹개발에 주로 쓰이는 오픈소스 관계형 데이터베이스
  - Free for non-commercial use only, owned by Oracle  
비상업적 용도는 공짜
  - Handles very large databases; very fast performance.  
아주 큰 데이터베이스를 다룰수 있고 아주 빠름
  - Easy to use with Java JDBC  
자바와 사용하기 편함

# Free RDBMS (cont.)

- PostgreSQL
  - Open Source, can be used in commercial apps without a license  
라이센스없이 상업용 앱에 사용할수 있음
  - Included with Linux distros
  - <http://www.postgresql.org>
- Derby
  - Lightweight, pure Java database  
사이즈가 작은, 순수 자바 데이터베이스
  - included with JavaEE as "Java DB"
  - <http://db.apache.org/derby>

# Free RDBMS (cont.)

- Hypersonic SQL
  - Lightweight, fast database written in Java  
자바로 쓰여진 사이즈가 작고, 빠른 데이터베이스
  - Useful for development and "demo" systems
  - <http://hsqldb.org>
- SQLite
  - World's most widely distributed database written in C  
C로 쓰여짐
  - Used by several widespread browsers, operating systems, or mobile phones
  - <https://www.sqlite.org/>

# Free RDBMS (cont.)

- Berkeley DB
  - written in C with language bindings for C++, Perl, Python, Ruby, and more libraries for embedded database using the OS's file system.
  - bought by Oracle in 2006.
  - <http://www.oracle.com/database/berkeley-db/index.html> (still Open Source under the "Sleepycat Public License" and "Sleepycat Commercial License")

# Oracle 11g Express Edition (XE)

- Leading market share among commercial databases  
상업 데이터베이스 중에서 가장 마켓점유율이 높음
- XE is easier to administer than full Oracle
- <http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

# SQL Command Line



The screenshot shows a Windows Start menu search results window titled "Run SQL Command Line". The results list includes the following items:

- Start menu  Programs
- (or All Programs)  Oracle Database 11g Express Edition
- Run SQL Command Line.

The "Run SQL Command Line" item is highlighted with a blue selection bar. Below the search results, a terminal window titled "Run SQL Command Line" is open, showing the following text:

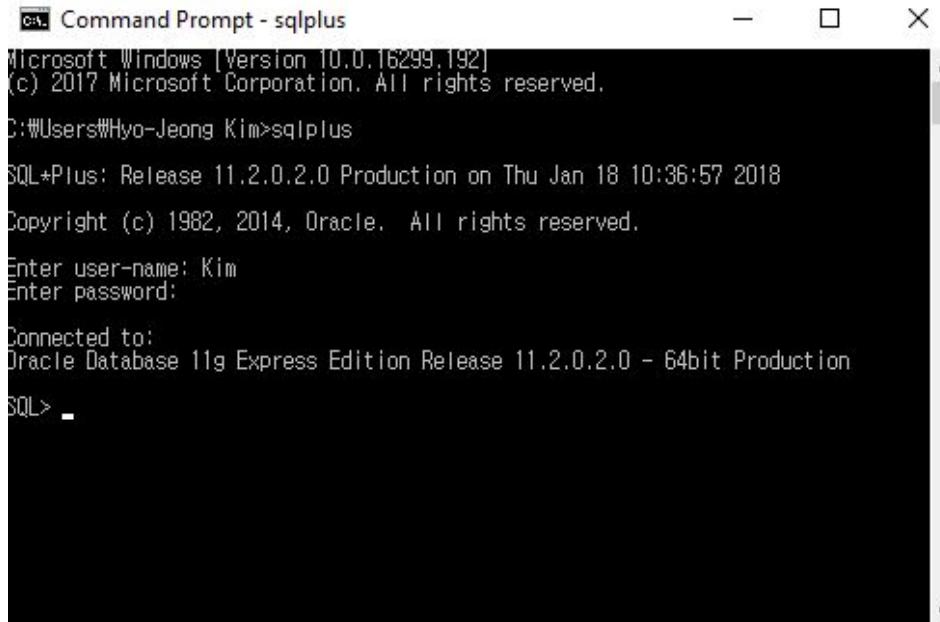
```
SQL*Plus: Release 11.2.0.2.0 Production on Thu Jan 18 10:26:03 2018  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
  
SQL> connect  
Enter user-name: Kim  
Enter password:  
Connected.  
SQL>
```

Start menu  Programs  
(or All Programs)  Oracle Database 11g Express Edition  Run SQL Command Line.

# SQL\*Plus

- Pronounced sequel plus
- An interactive and batch query tool that is installed with every Oracle Database Express Edition installation 오라클 설치할 때 함께 설치됨
- It has a command-line user interface that acts as the client when connecting to the database. 명령어기반 사용자 인터페이스
- You can enter and run SQL\*Plus commands, SQL statements, PL/SQL statements, and operating system commands. 다양한 명령어를 실행할 수 있음

# Connecting to Oracle Database Express Edition from SQL\*Plus



Microsoft Windows [Version 10.0.16299.192]  
(c) 2017 Microsoft Corporation. All rights reserved.  
C:\Users\Hyo-Jeong Kim>sqlplus  
SQL\*Plus: Release 11.2.0.2.0 Production on Thu Jan 18 10:36:57 2018  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Enter user-name: Kim  
Enter password:  
Connected to:  
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production  
SQL> -

Search   
Windows   
cmd   
sqlplus

# Oracle SQL Developer

- A graphical version of SQL\*Plus that gives database developers a convenient way to perform basic tasks. **SQL\*PLUS의 그래픽한 버전**
- You can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. **데이터베이스 오브젝트 검색, 명령어 실행, 명령어 편집, 디버깅 등을 할 수 있음**
- You can also run any number of provided reports, as well as create and save your own. **리포트 생성**

# PL/SQL (Procedural Language/SQL)

- Pronounced P L sequel
- A native Oracle Database Express Edition extension to SQL.
- Bridges the gap between declarative and imperative program control by adding procedural elements, such as conditional control and loops.  
SQL에 프로그램적인 요소를 가미

# Exercise #2

- Download and install Oracle Database Express Edition 11g Release 2
  - File -  
<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>
  - Instructions -  
[https://docs.oracle.com/cd/E17781\\_01/install.112/e18803/toc.htm#XEINW101](https://docs.oracle.com/cd/E17781_01/install.112/e18803/toc.htm#XEINW101)
    - #1 for overview, #4 for installation instructions, #5 for executing Oracle DB XE

# Exercise #2

- Complete p.23-28.
  - Search windows □ type cmd □ type sqlplus in the command windows □ id – system, password – 1234
  - ALTER USER HR IDENTIFIED BY your\_password ACCOUNT UNLOCK;
  - Use employees table instead of emp table.

# Exercise #2

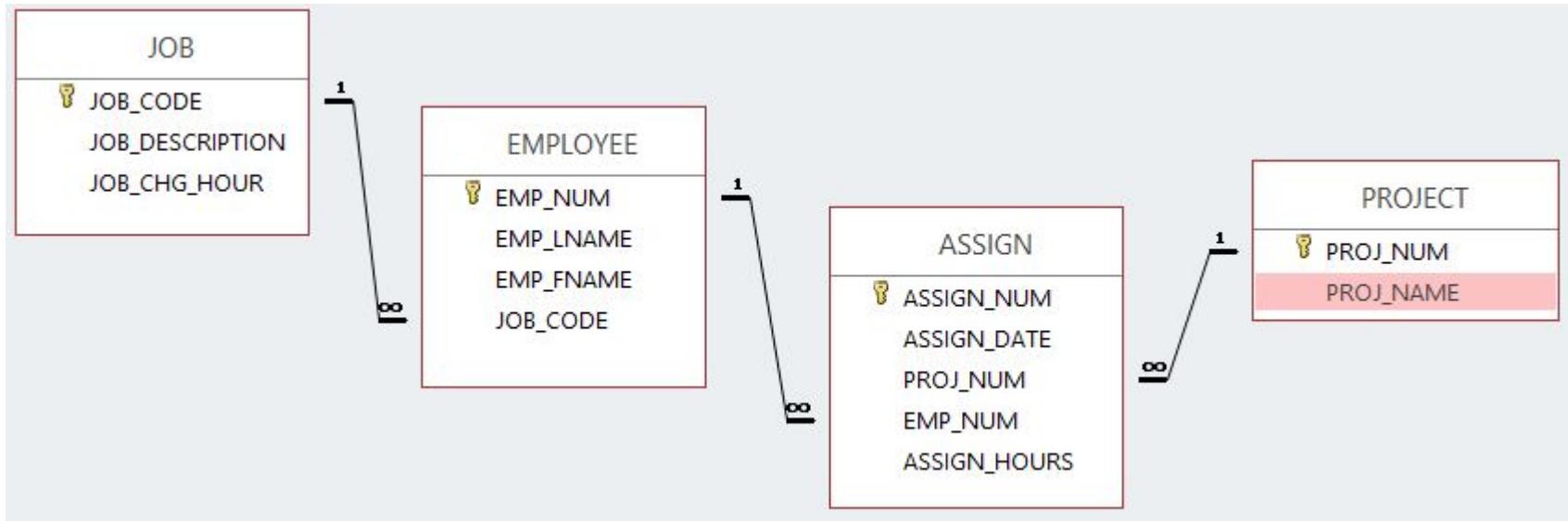
- Install SQL Developer
  - File -  
<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>
  - Instructions -  
[https://docs.oracle.com/cd/E17781\\_01/admin.112/e18585/toc.htm](https://docs.oracle.com/cd/E17781_01/admin.112/e18585/toc.htm)
    - *#1 for user interface*
    - *#3 for installation*
    - *#4.1 for database connection*

# Exercise #2

## Foothills Construction Company Example

The Foothills Construction Company needs a relational database developed to track project details according to project name, hours assigned and charges per hour for each job description. From a list of entity classes (and their primary keys) and a list of integrity constraints, develop a database for Foothills Construction Company.

# ER Diagram



# Job Table

	Field Name	Data Type
PK	JOB_CODE	Short Text
	JOB_DESCRIPTION	Short Text
	JOB_CHG_HOUR	Currency

# Employee Table

	Field Name	Data Type
PK	EMP_NUM	Short Text
	EMP_LNAME	Short Text
	EMP_FNAME	Short Text
	JOB_CODE	Short Text

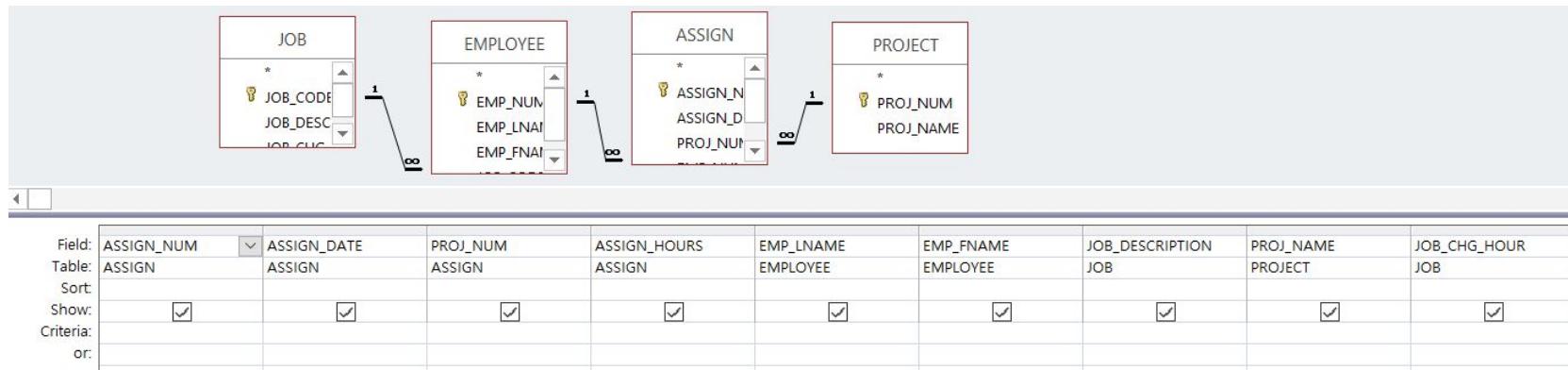
# Assign Table

Field Name	Data Type
ASSIGN_NUM	Number
ASSIGN_DATE	Date/Time
PROJ_NUM	Short Text
EMP_NUM	Short Text
ASSIGN_HOURS	Number

# Project Table

	Field Name	Data Type
	PROJ_NUM	Short Text
	PROJ_NAME	Short Text

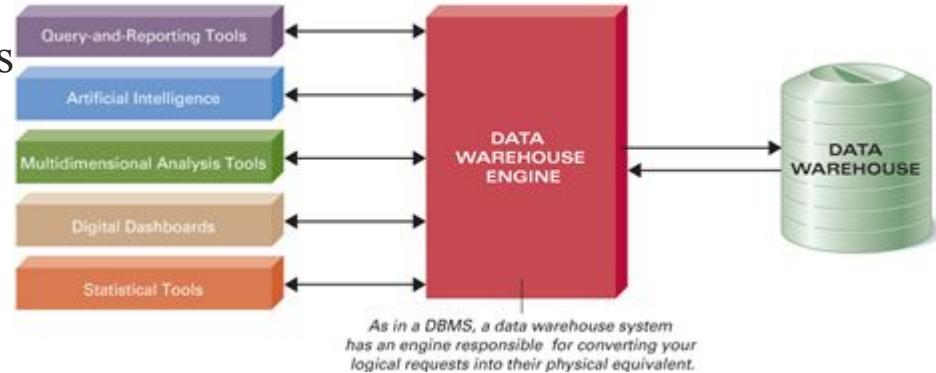
# Query for Report



# Data Mining Tools

# Data-Mining Tools

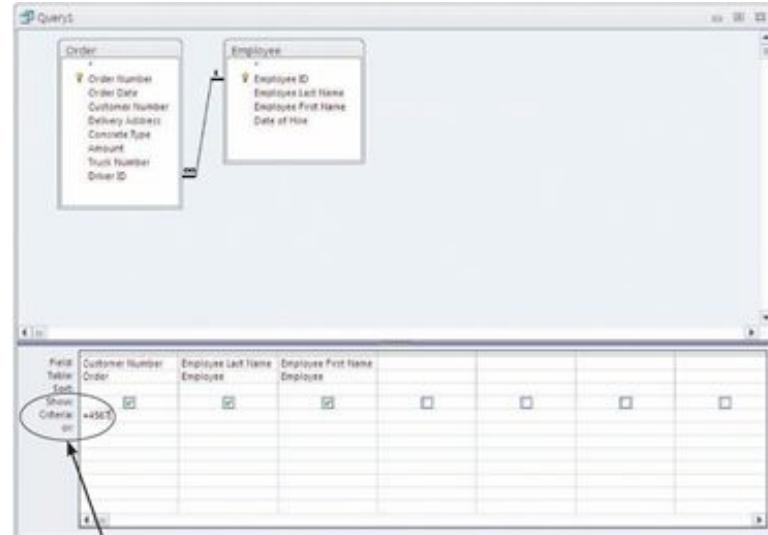
- Software tools you use in a data warehouse environment
  - Query-and-reporting tools
  - Artificial intelligence
  - Multidimensional analysis tools
  - Digital dashboards
  - Statistical tools



# Query-by-Example (QBE) Tool

- You can graphically design the answer to a question

질의어를 만들어  
데이터マイ닝



Customer Number	Employee Last Name	Employee First Name
4567	Evaraz	Antonio
4567	Robertson	John
4567	Robertson	John
4567	Robertson	John

# Report Generator

- You can quickly define formats of reports and what information you want to see in a report

리포트를 만들어  
데이터마이닝



The screenshot shows two windows of the Report Wizard. The left window, titled 'Report Wizard - Select Fields', displays a list of fields from the 'Order' file: Order Number, Order Date, and Amount. The right window, titled 'Report Wizard - Select Report Type', shows options for 'List', 'Table', 'Chart', and 'Query'. Below these windows is a preview of a report titled 'CUSTOMER AND AMOUNT REPORT'.

Selected fields from the Order file

Report formats

Customer Number	Order Number	Order Date	Amount
1234	100000	9/1/2004	8
1234	100002	9/2/2004	6
1234	100006	9/5/2004	4
1234	100009	9/7/2004	8
1234	100015	9/12/2004	8
2345	100007	9/6/2004	5
2345	100012	9/9/2004	8
3456	100001	9/1/2004	3
4567	100003	9/3/2004	4
4567	100004	9/4/2004	8
4567	100011	9/9/2004	6
4567	100013	9/10/2004	4
5678	100005	9/4/2004	4
6789	100008	9/6/2004	8
6789	100010	9/9/2004	7
6789	100014	9/10/2004	6
4567	Robertson	John	
4567	Robertson	John	

# Structured Query Language (SQL)

- An international standard for creating, processing, and querying databases and their tables.

SQL을 이용하여  
데이터마이닝

```
SELECT    CUSTOMER.CustomerLastName,  
          CUSTOMER.CustomerFirstName,  
          CUSTOMER.Phone,  
          COURSE.CourseDate,  
          ENROLLMENT.AmountPaid,  
          COURSE.Course, COURSE.Fee  
FROM CUSTOMER, ENROLLMENT, COURSE  
WHERE    CUSTOMER.CustomerNumber  
        = ENROLLMENT.CustomerNumber  
        AND    COURSE.CourseNumber  
        = ENROLLMENT.CourseNumber;
```

# Artificial Intelligence

- Tools to help you “discover” information and trends

인공지능툴이나  
알고리즘을  
이용하여  
데이터마이닝

## Artificial Intelligence

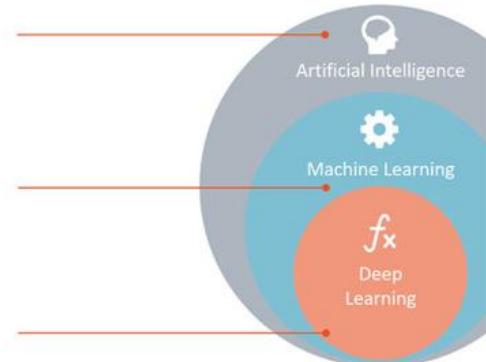
Any technique which enables computers to mimic human behavior.

## Machine Learning

Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

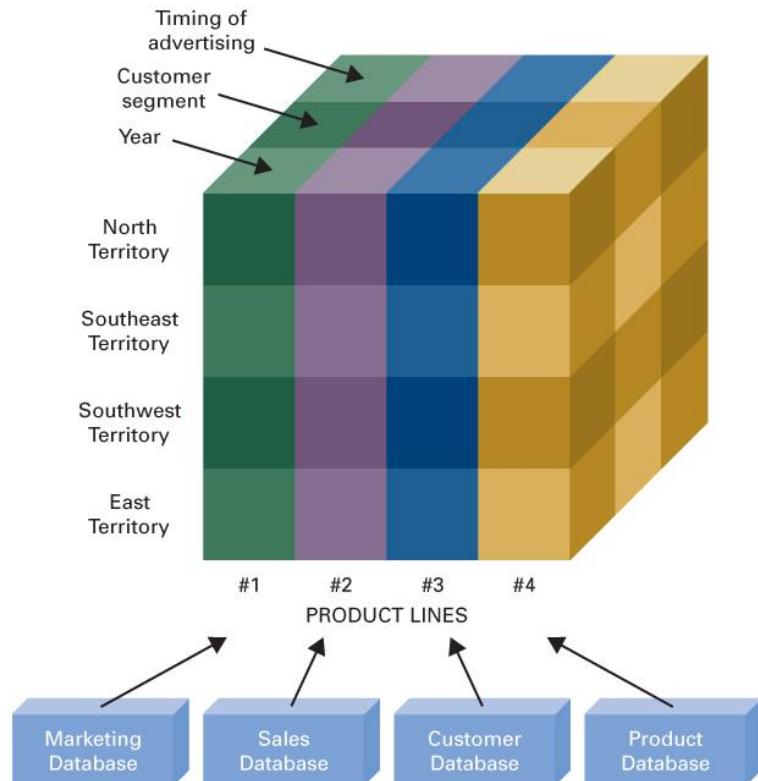
## Deep Learning

Subset of ML which make the computation of multi-layer neural networks feasible.



# Multidimensional Analysis (MDA tools)

- Slice-and-dice techniques for viewing multidimensional information
- 다차원 정보를  
잘라서  
데이터マイ닝



# Digital Dashboard

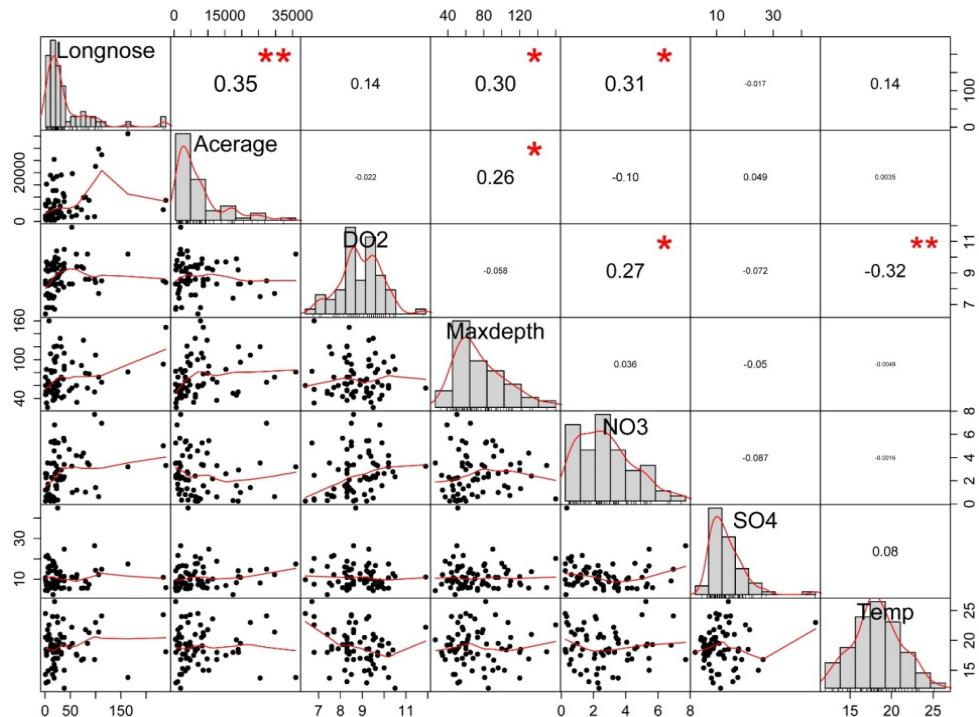
- Displays key information on a computer screen tailored to the needs and wants of an individual  
필요에 따라 주요성과지표를  
요약
- Key performance indicator (KPI)
  - most essential information used in any analytics initiative  
주요성과지표를 시각화



# Statistical Tools

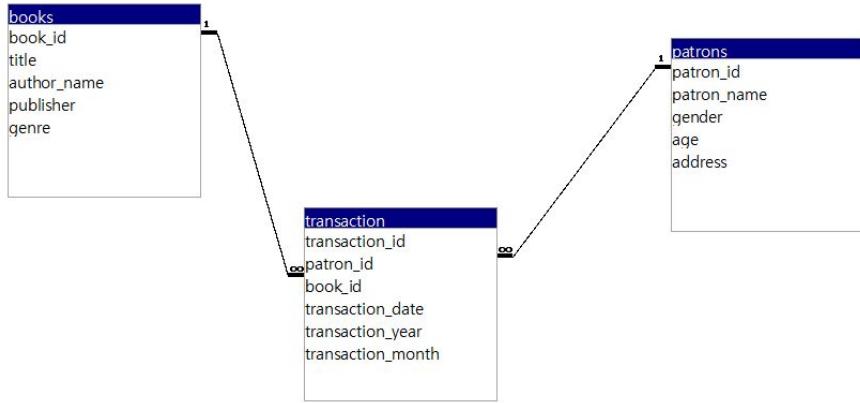
- Applying mathematical models to data warehouse information.  
e.g., linear regression

회귀분석 같은  
수학모델을 이용하여  
데이터마이닝



# Exercise #1

- Create a simple library database with three tables: books, patrons, and transaction  
세 개의 테이블을 가진 도서관 데이터베이스를 만드시요
- You can import the given text files into the tables. If you can't directly import the text files into Access, copy and paste into Excel then import the Excel file  
데이터를 텍스트파일로 만들어서 임포트하던가  
엑셀파일로 만들어서 임포트할수 있음



# Sample SQL Code for books and patrons

```
create table books(  
    book_id char(13),  
    title varchar(100),  
    author_name varchar(50),  
    publisher varchar(50),  
    genre varchar(50),  
    constraint pk_book_id  
primary key (book_id));
```

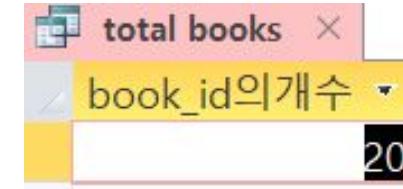
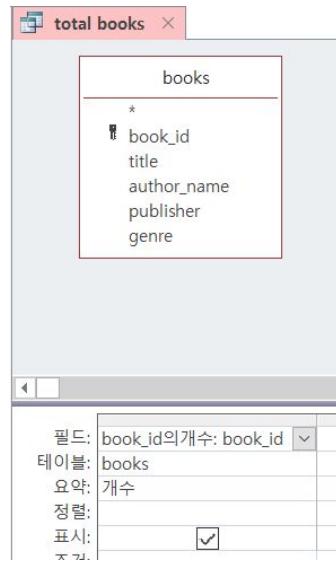
```
create table patrons(  
    patron_id char(5),  
    patron_name varchar(80),  
    gender char(1),  
    age number,  
    address varchar(200),  
    constraint pk_patron_id  
primary key (patron_id));
```

# Sample SQL Code for transaction

```
create table transaction(
    transaction_id char(5),
    patron_id char(5),
    book_id char(13),
    transaction_date char(50),
    transaction_year char(4),
    transaction_month varchar(2),
    constraint pk_transaction_id primary key(transaction_id),
    constraint fk_patron_id foreign key(patron_id) references patron(patron_id),
    constraint fk_book_id foreign key(book_id) references book(book_id));
```

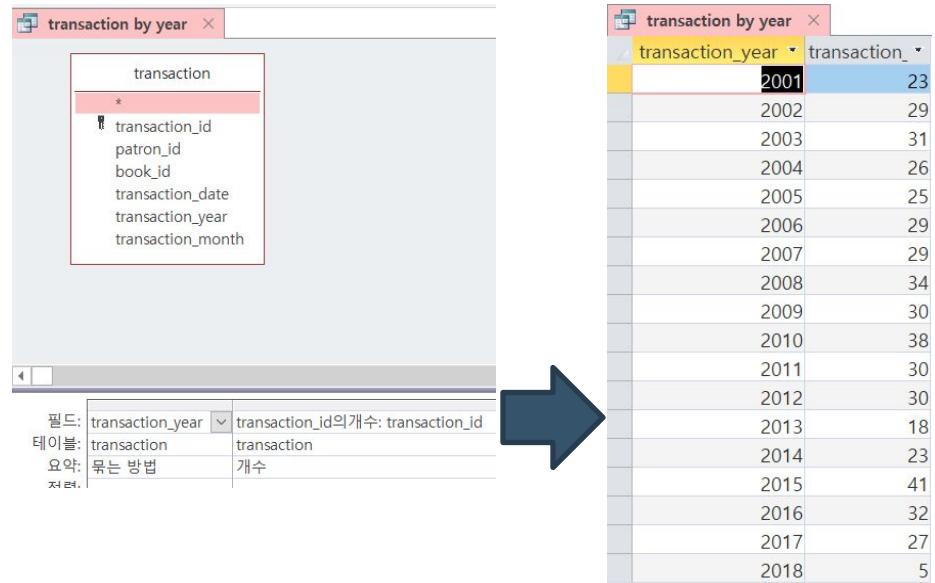
# Exercise #2

- Create relationships among tables and produce a ER Diagram report.  
테이블 사이의 관계를 만들고  
ER다이어그램 리포트를 만드시요
- Create queries to answer the following questions 다음의 질문을  
답할 수 있는 질의어를 만들시요
  - Total number of books?
  - Total number of patrons?
  - Total number of transactions?



# Exercise #2

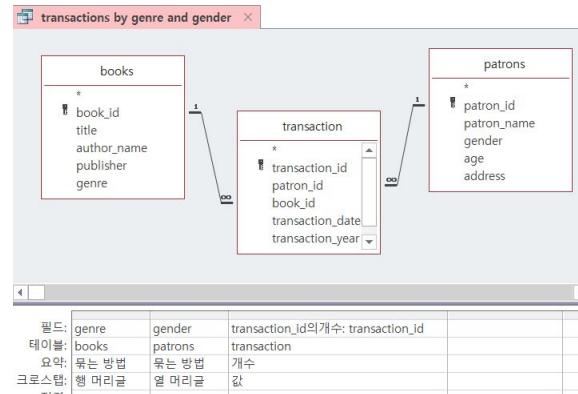
- Number of transactions by year?
- Number of transactions by genre?
- Number of transactions by month?



# Exercise #2

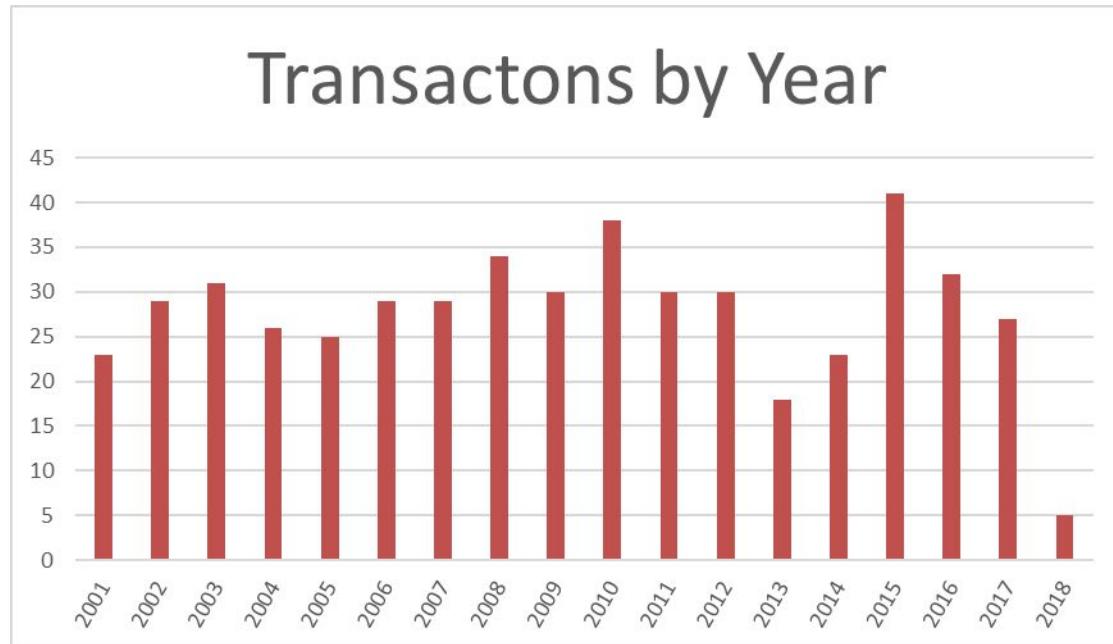
- Number of transactions by genre and gender?
- Number of transactions by month and gender?

genre	F	M
소설	47	53
아동도서	27	23
에세이	91	109
학습서	73	77



# Number of Transactions by Year

transaction_year	transaction_
2001	23
2002	29
2003	31
2004	26
2005	25
2006	29
2007	29
2008	34
2009	30
2010	38
2011	30
2012	30
2013	18
2014	23
2015	41
2016	32
2017	27
2018	5



# Exercise #2

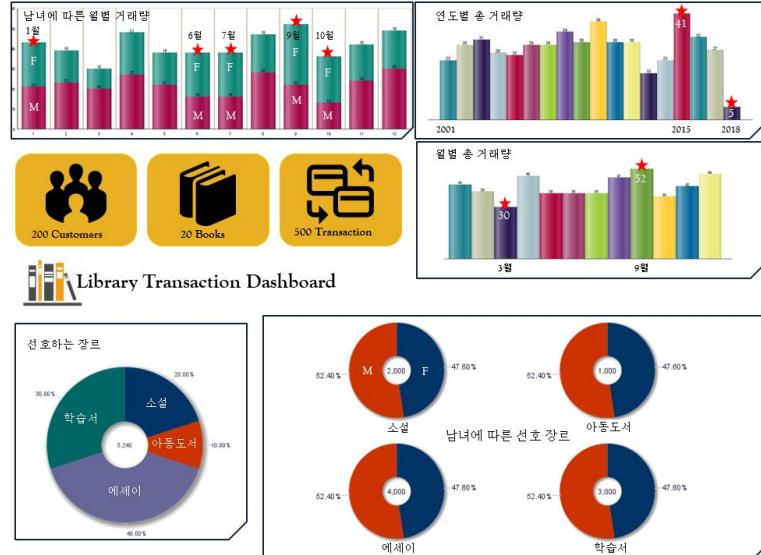
- Build a “Dashboard” that summarize information about the patrons, books, and transactions

고객, 책, 거래내역에 대한 정보를  
요약하는 대시보드를 만드시요

- Export the result of queries to Excel and create appropriate charts for your data  

쿼리결과를 엑셀로 엑스포트해서 적절한  
그래프를 그리시요
- Summarize the result and charts using a PowerPoint slide  

파워포인트를 이용해서 결과와 차트를  
요약하시요



# Group Project

- Design a simplified database for any businesses such as grocery store, motel, insurance, airline, housing, library, etc. and create an actual database using MS Access or Oracle Database 11g Express, populate the data, and write interactive queries, and reports on the database
  - Form a team with 2-3 people
  - Create a new user account “am\_group” with password “1234”

# Group Project

- Create tables
  - The database include at least three tables. For example, books, patrons (people who have library cards), and transactions (checking a book out, returning a book, and so on)
- Create ERD and relational model using reverse engineering
- Insert data into the tables
  - Use a sequence for one of your primary keys
  - Import your data from three different data sources (e.g., text file, excel file, MS Access, or Oracle database)

# Group Project

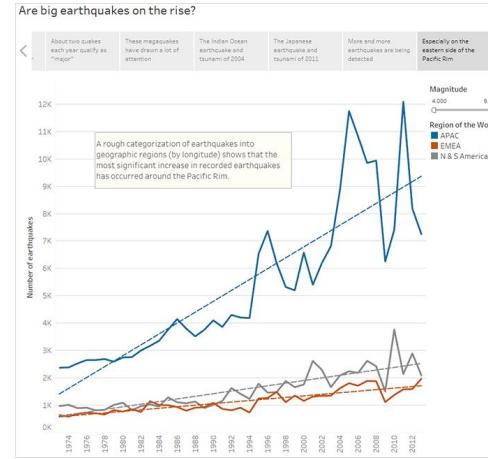
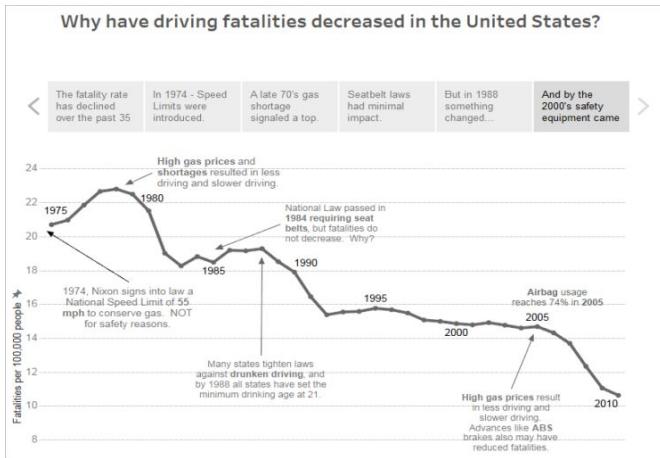
- Create a view that returns information about your customers and their transactions
  - For example, this view queries the PATRONS and TRANSACTIONS tables, and returns rows that contain a patron's ID, last name, and first name, along with a transaction and the transaction type. The rows are ordered by patron ID, and by transaction type within patron IDs

# Group Project

- Create a report to summarize information about customers, products, and their transactions
- Build a “Dashboard” that summarize information about customers, products, and their transactions

# Story

- A sequence of visualizations that work together to convey information. When you share a story – for example, by publishing a workbook to Tableau Public, Tableau Server, or Tableau Online – users can interact with the story to reveal new findings or ask new questions of the data.



# Exercise #2

- Complete the story tutorial on  
[https://help.tableau.com/current/pro/desktop/en-us/story\\_create.htm](https://help.tableau.com/current/pro/desktop/en-us/story_create.htm)