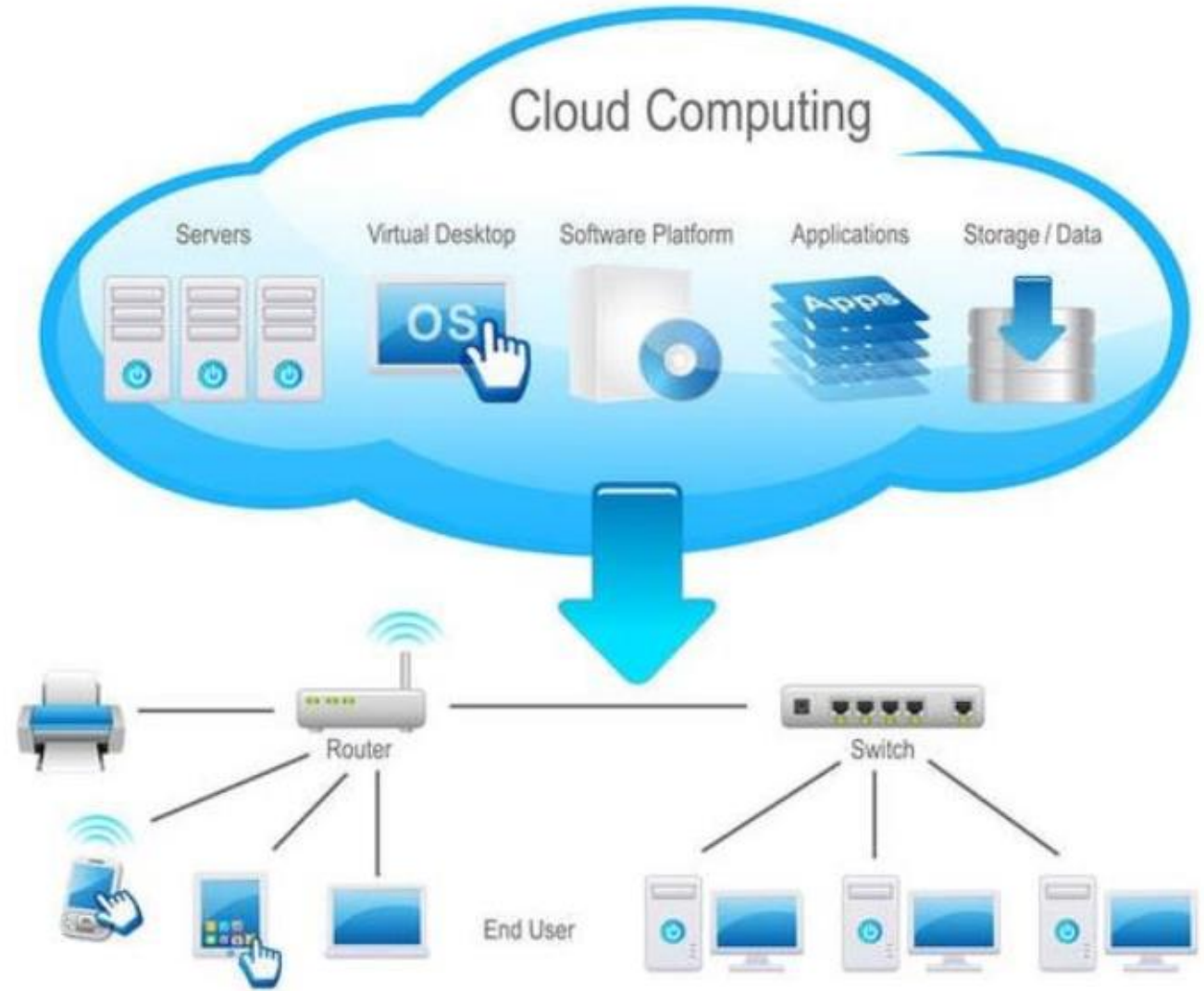
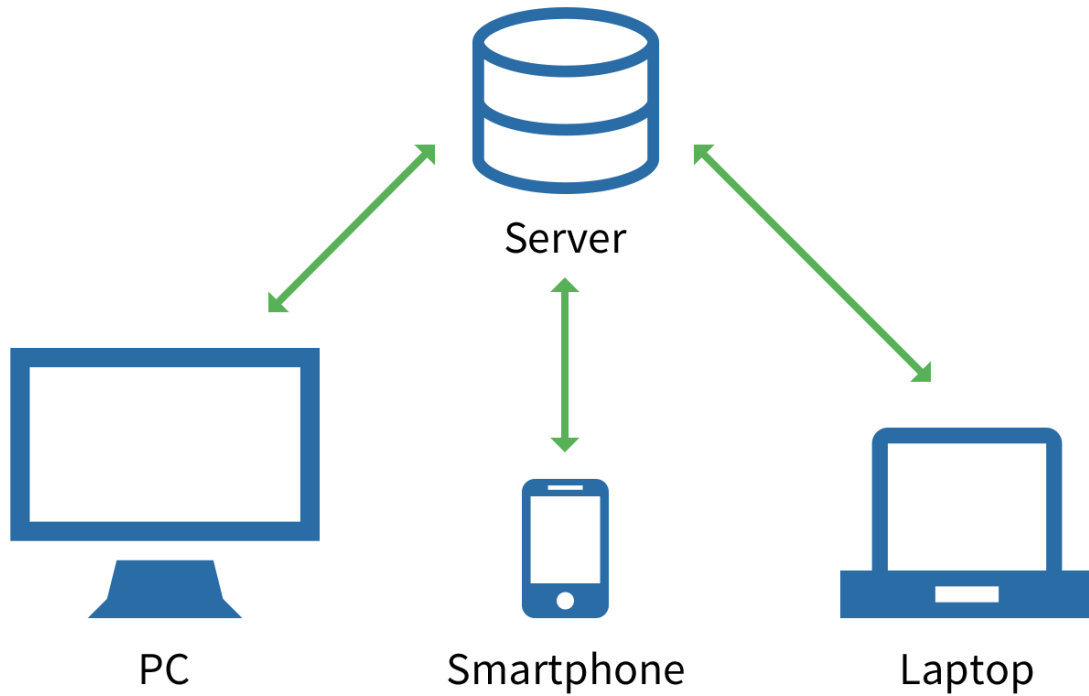
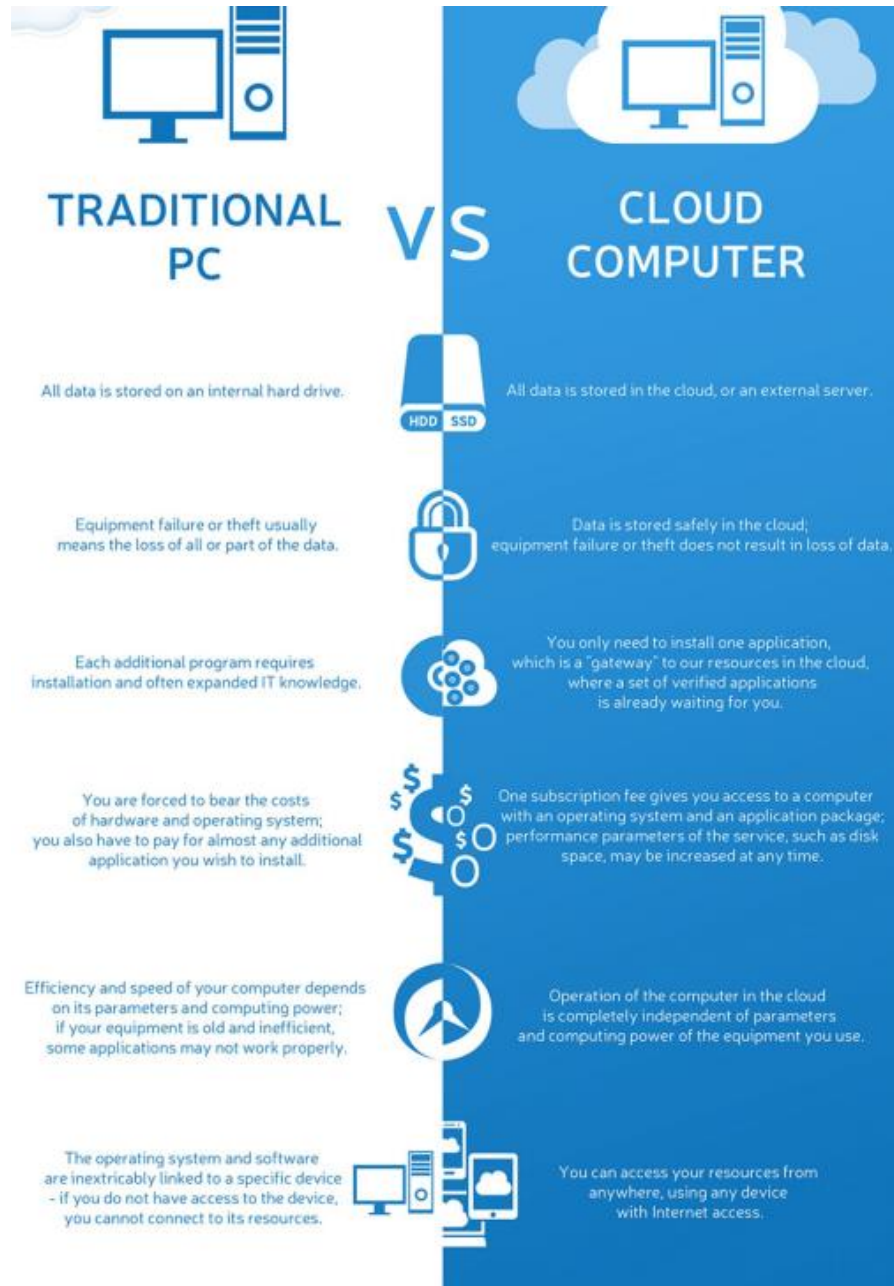


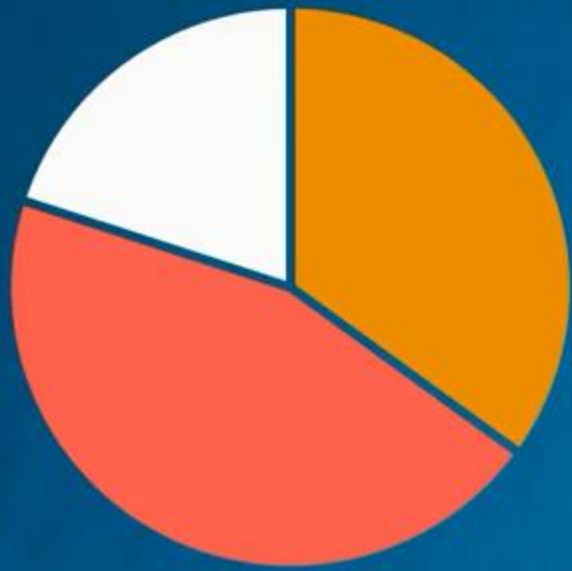


# Client-Server Model









**Business Intelligence (BI)**



**Web Development**

**Why study SQL?**



**Data Science**

**Database Administration**



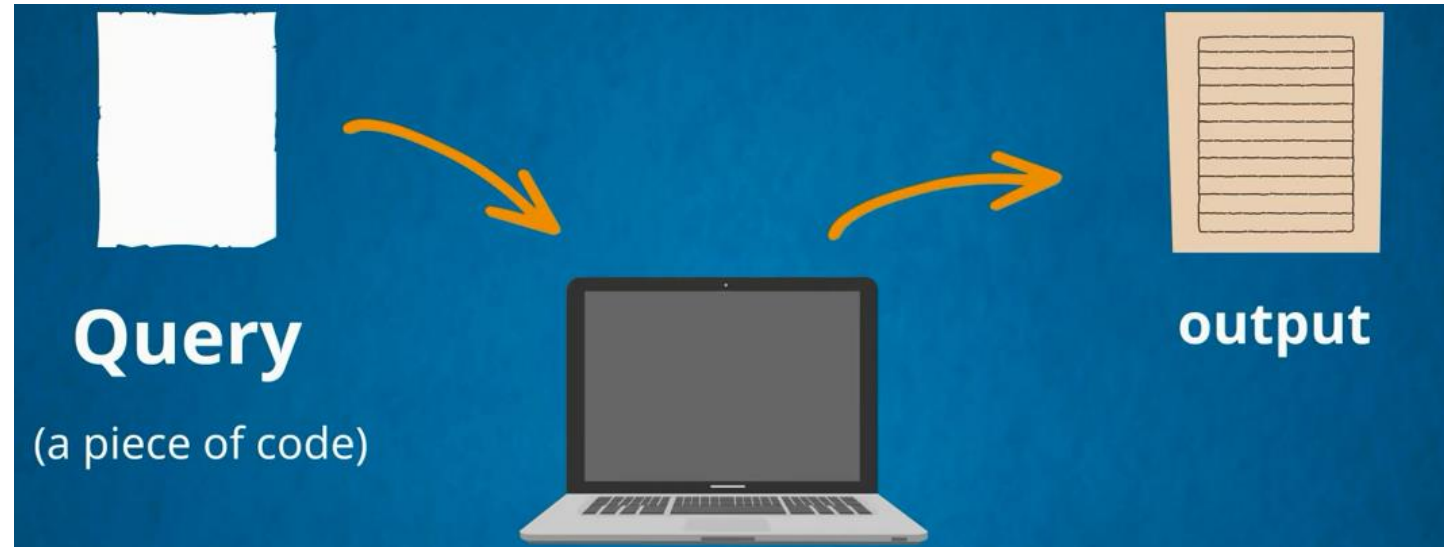
# SQL = Structured Query Language

1. a programming language specifically designed for working with databases

- create

- manipulate **DATA**

- share **DATA** from Relational Database Management Systems



# DBMS



ORACLE



# The Client-Server Model

The program we will be working with in this course is called MySQL Workbench. It is the Oracle visual tool for database design, modelling, creation, manipulation, maintenance, and administration. Professionals refer to this type of software as “Integrated Development Environment” or IDE. So, Workbench will be our IDE.

And, if you wonder what *Oracle* is, this is the software company that owns the MySQL version of SQL.

## Client Program:

MySQL Workbench



provided by **ORACLE**

You could also wonder why we would need a server. Sticking to the basic theory of operation of computer networks, MySQL Workbench acts as a client program - a client of a MySQL Server.

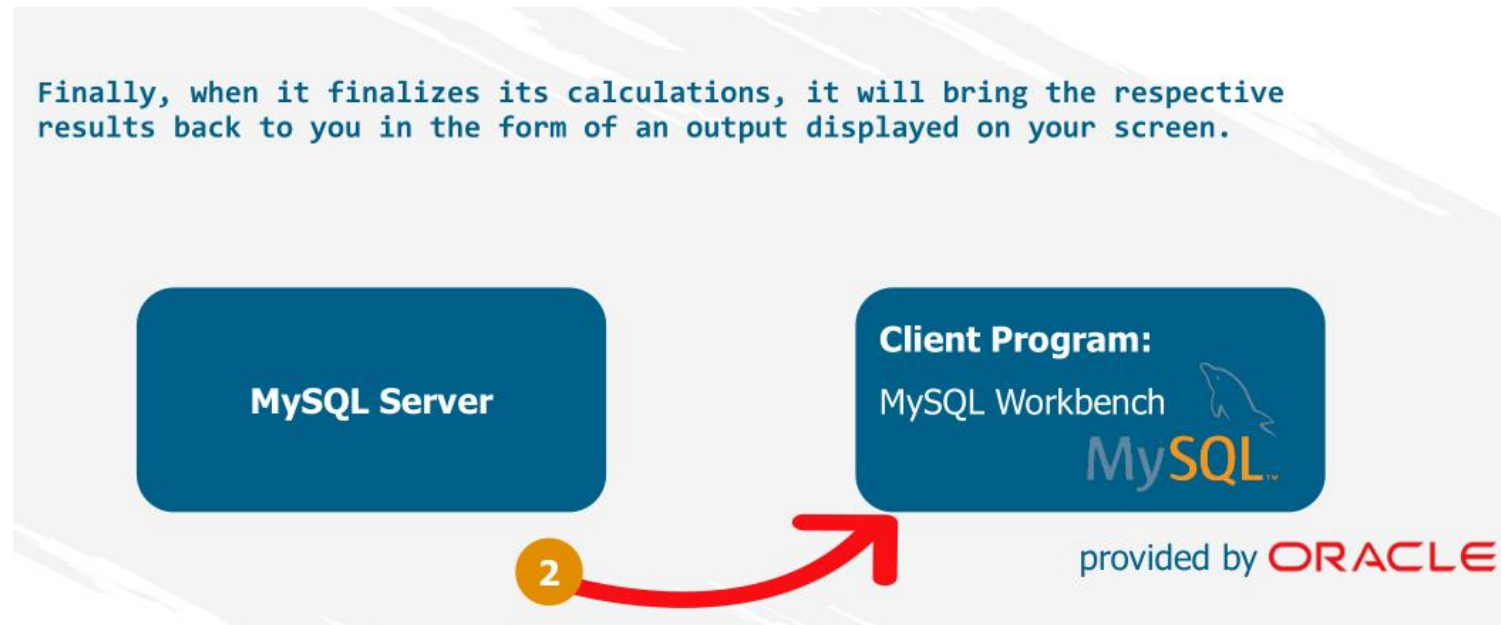
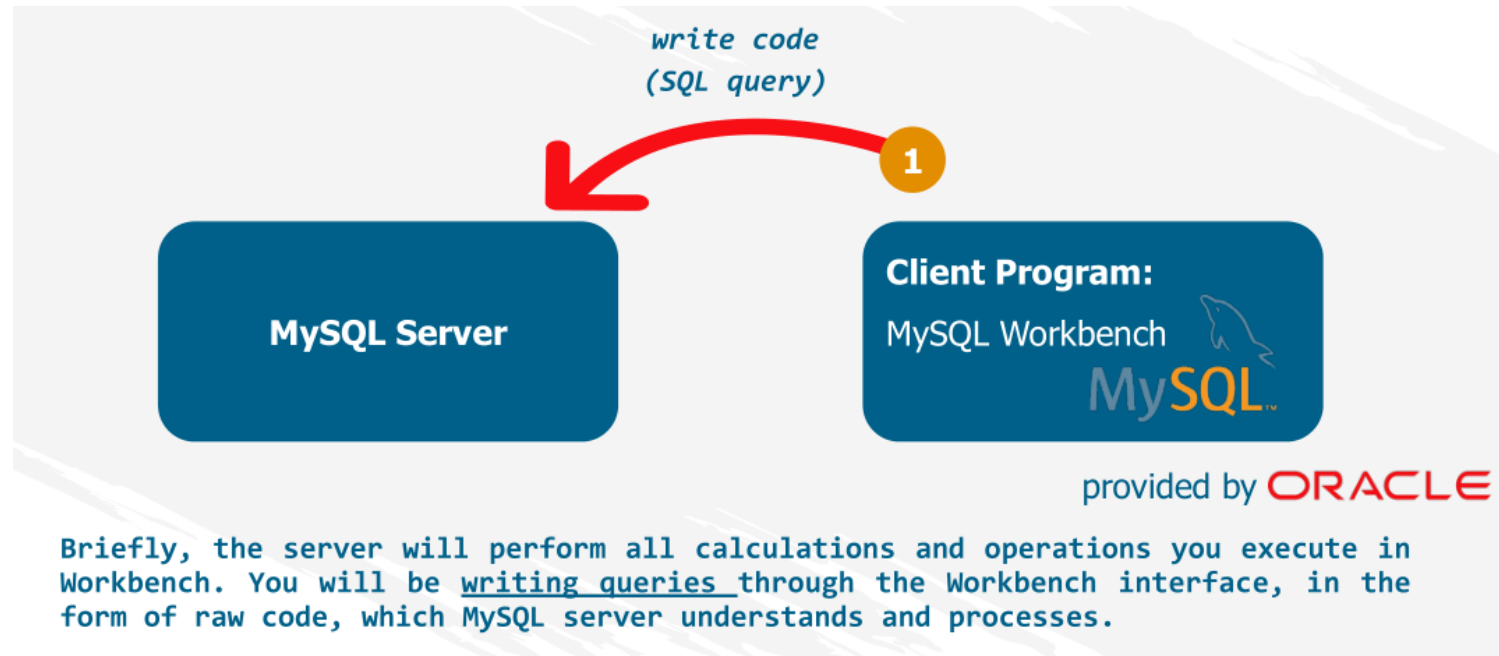
**MySQL Server**

## Client Program:

MySQL Workbench

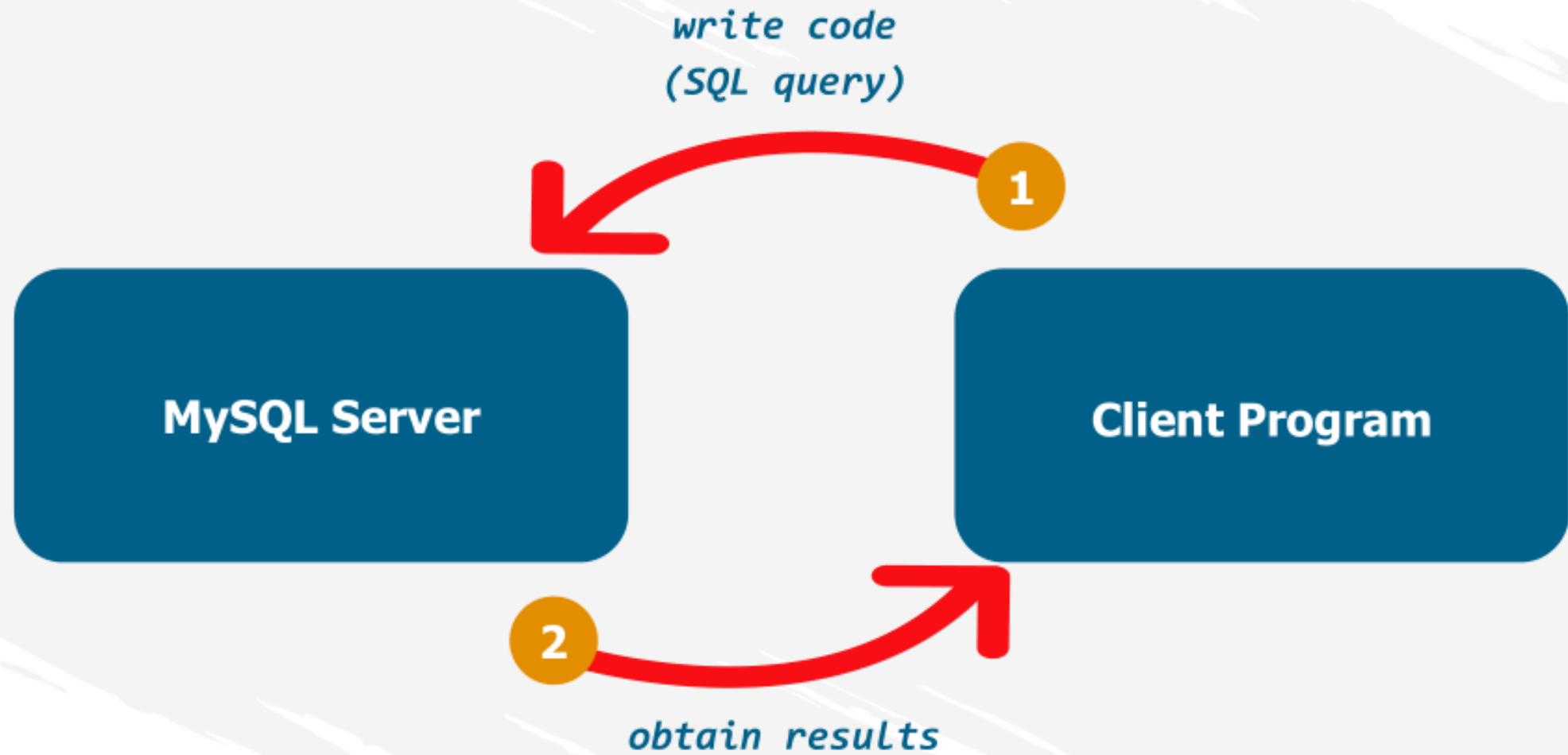


provided by **ORACLE**





# The Client-Server Model



record  
record

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2016	1	A_1
2	02/12/2016	2	C_1
3	15/04/2017	3	D_1
4	24/05/2017	1	B_2
5	25/05/2017	4	B_2
6	06/06/2017	2	B_1
7	10/06/2017	4	A_2
8	13/06/2017	3	C_1
9	20/07/2017	1	A_1
10	11/08/2017	2	B_1

field

= a column in a table containing specific information about every record in the table

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2016	1	A_1
2	02/12/2016	2	C_1
3	15/04/2017	3	D_1
4	24/05/2017	1	B_2
5	25/05/2017	4	B_2
6	06/06/2017	2	B_1
7	10/06/2017	4	A_2
8	13/06/2017	3	C_1
9	20/07/2017	1	A_1
10	11/08/2017	2	B_1

Customer												
purchase_number	date_of_purchase	customer_id	first_name	last_name	email_address	number_of_complaints	item_id	item	unit_price_usd	company	headquarters	phone_number
1	03/09/2016	cust_1	John	McKinley	john.mckinley@365care.com	0	A_1	Lamp	20	Company A		+1 (202) 555-0196
2	02/12/2016	cust_2	Elizabeth	McFarlane	e.mcfarlane@365care.com	2	C_1	Chair	150	Company C		+1 (229) 853-9913
3	15/04/2017	cust_3	Kevin	Lawrence	kevin.lawrence@365care.com	1	D_1	Loudspeakers	400	Company D		+1 (618) 369-7392
4	24/05/2017	cust_1	John	McKinley	john.mckinley@365care.com	0	B_2	Desk	350	Company B		+1 (202) 555-0152
5	25/05/2017	cust_4	Catherine	Winnfield	c.winnfield@365care.com	0	B_2	Desk	350	Company B		+1 (202) 555-0152
6	06/06/2017	cust_2	Elizabeth	McFarlane	e.mcfarlane@365care.com	2	B_1	Lamp	30	Company B		+1 (202) 555-0152
7	10/06/2017	cust_4	Catherine	Winnfield	c.winnfield@365care.com	0	A_2	Desk	250	Company A		+1 (202) 555-0196
8	13/06/2017	cust_3	Kevin	Lawrence	kevin.lawrence@365care.com	1	C_1	Chair	150	Company C		+1 (229) 853-9913
9	20/07/2017	cust_1	John	McKinley	john.mckinley@365care.com	0	A_1	Lamp	20	Company A		+1 (202) 555-0196
10	11/08/2017	cust_2	Elizabeth	McFarlane	e.mcfarlane@365care.com	2	B_1	Lamp	30	Company B		+1 (202) 555-0152

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	<a href="mailto:john.mackinley@365careers.com">john.mackinley@365careers.com</a>	0
2	Elizabeth	McFarlane	<a href="mailto:e.mcfarlane@365careers.com">e.mcfarlane@365careers.com</a>	2
3	Kevin	Lawrence	<a href="mailto:kevin.lawrence@365careers.com">kevin.lawrence@365careers.com</a>	1
4	Catherine	Winnfield	<a href="mailto:c.winnfield@365careers.com">c.winnfield@365careers.com</a>	0

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2016	1	A_1
2	02/12/2016	2	C_1
3	15/04/2017	3	D_1
4	24/05/2017	1	B_2
5	25/05/2017	4	B_2
6	06/06/2017	2	B_1
7	10/06/2017	4	A_2
8	13/06/2017	3	C_1
9	20/07/2017	1	A_1
10	11/08/2017	2	B_1

Items					
item_code	item	unit_price_usd	company_id	company	headquarters_phone_number
A_1	Lamp	20	1	Company A	+1 (202) 555-0196
A_2	Desk	250	1	Company A	+1 (202) 555-0196
B_1	Lamp	30	2	Company B	+1 (202) 555-0152
B_2	Desk	350	2	Company B	+1 (202) 555-0152
C_1	Chair	150	3	Company C	+1 (229) 853-9913
D_1	Loudspeakers	400	4	Company D	+1 (618) 369-7392

**relational algebra** allows us to retrieve data efficiently

## SALES

3 tables

"items"

item\_code

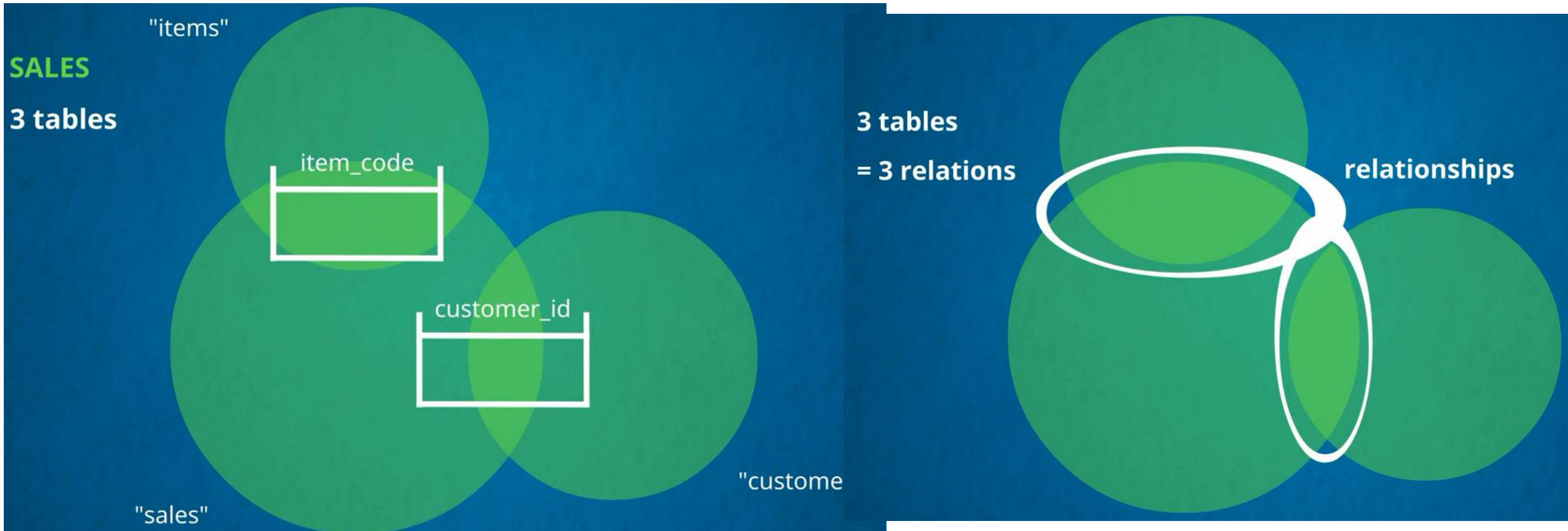
customer\_id

"sales"

"custome

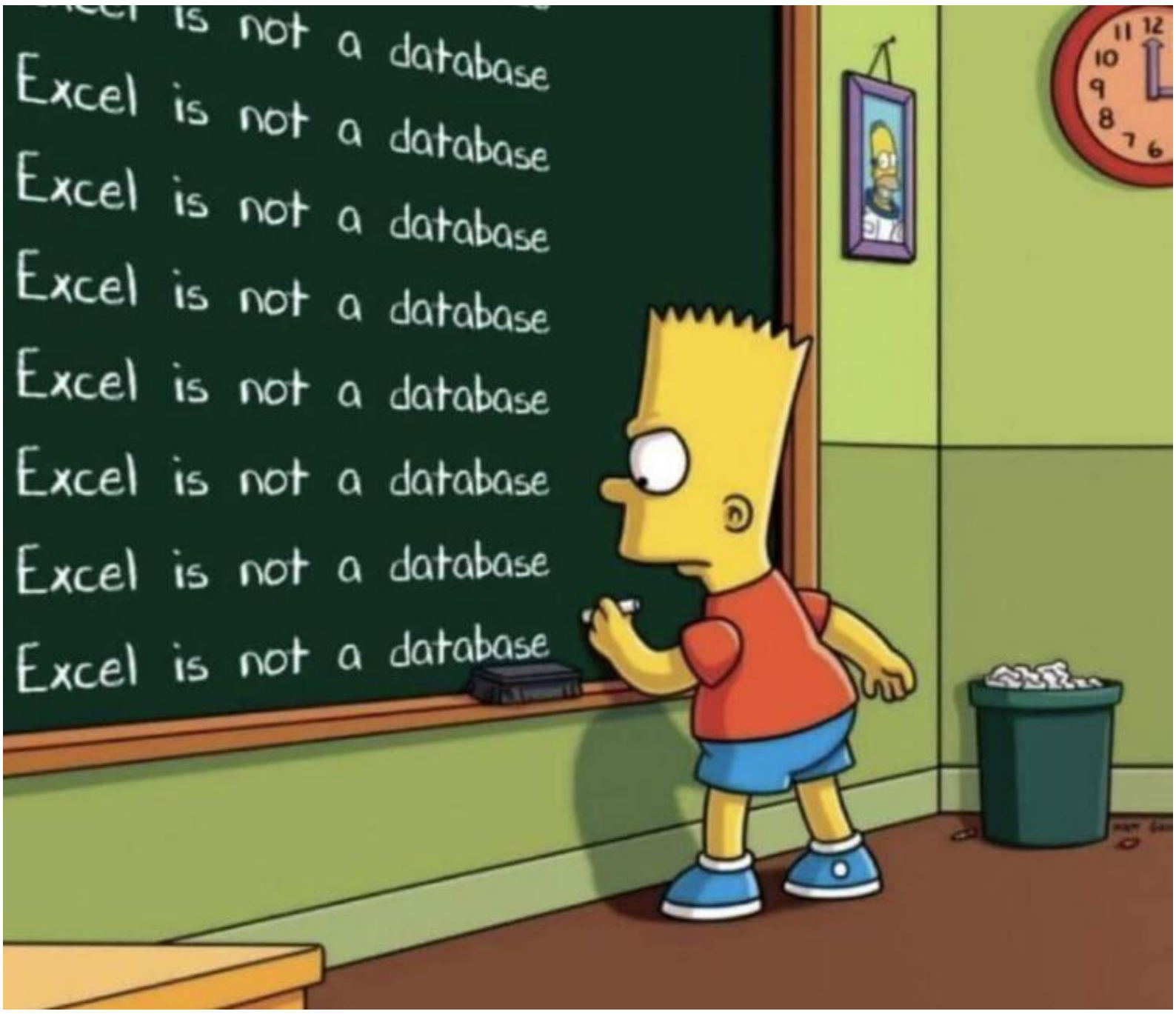
3 tables  
= 3 relations

relationships





**IS EXCEL A  
DATABASE?**



# Databases vs. Spreadsheets

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2017	1	A_1
2	03/09/2017	2	C_1
1	04/09/2017	3	D_1
2	04/09/2017	1	B_2
3	04/09/2017	4	B_2
4	04/09/2017	2	B_1
1	05/09/2017	4	A_2
1	06/09/2017	3	C_1
2	06/09/2017	1	A_1
3	06/09/2017	2	B_1

A screenshot of a spreadsheet application, likely Microsoft Excel, showing a grid of data. The data is organized into columns and rows, similar to the database table shown on the left, but presented in a more flexible, user-editable format.

relational databases



Spreadsheets

**Both:**

can contain a large amount of tabular data

can use existing data to make calculations

are used by many users

relational databases



Spreadsheets

pre-set the type of data contained  
in a certain field

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2017	1	A_1
2	03/09/2017	2	C_1
3	03/09/2017	3	D_1
4	03/09/2017	4	B_2
1	04/09/2017	2	B_1
2	04/09/2017	4	A_2
3	05/09/2017	3	C_1
4	06/09/2017	1	A_1
1	06/09/2017	2	B_1

A screenshot of the Microsoft Excel interface. The 'Date' cell in column B, row 3 is selected, and the date '03/05/2018' is entered. The ribbon at the top shows 'File', 'Home', 'Insert', 'Page Layout', 'Formulas', 'Data', 'Review', and 'View'.

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/05/2018	1	A_1
2		2	C_1
3		3	D_1
4		4	B_2
1		2	B_1
2		4	A_2
3		3	C_1
4		1	A_1
1		2	B_1

relational databases



Spreadsheets

data stored in a record of a table

data stored in a cell

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2017	1	A_1
2	03/09/2017	2	C_1
1	04/09/2017	3	D_1
2	04/09/2017	1	B_2
3	04/09/2017	4	B_2
4	04/09/2017	2	B_1
1	05/09/2017	4	A_2
1	06/09/2017	3	C_1
2	06/09/2017	1	A_1
3	06/09/2017		B_1

A screenshot of the Microsoft Excel interface showing a grid of cells. A single cell is highlighted with a black border.

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2017	1	A_1
2	03/09/2017	2	C_1
1	04/09/2017	3	D_1
2	04/09/2017	1	B_2
3	04/09/2017	4	B_2
4	04/09/2017	2	B_1
1	05/09/2017	4	A_2
1	06/09/2017	3	C_1
2	06/09/2017	1	A_1
3	06/09/2017		B_1

formatting

formatting ✓



relational databases



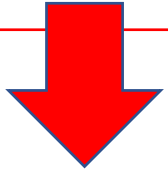
Spreadsheets

different cells can contain calculations  
(functions and formulas)


all calculations and operations  
are done after data retrieval

you can do calculations in "views"

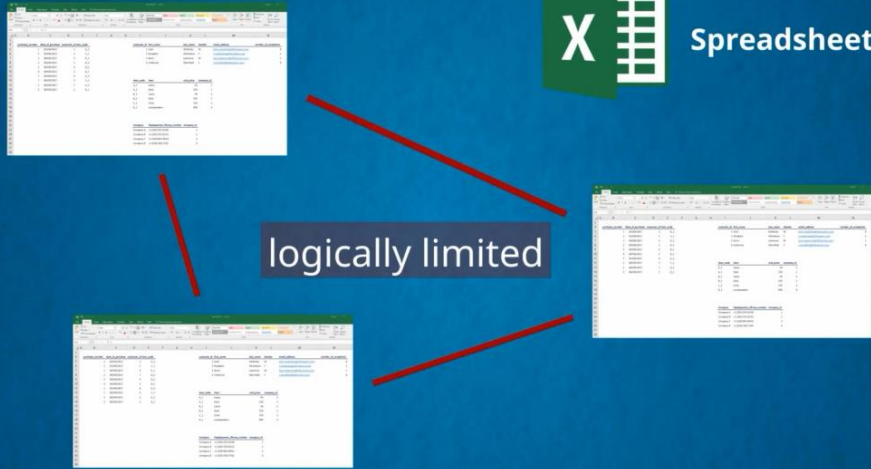
record of data  $\neq$  calculation



DATA INTEGRITY




# Spreadsheets



logically limited

# relational databases



customer_id	first_name	last_name	Gender	email_address	number_of_complaints
1	John	McKinley	M	<a href="mailto:john.mckinley@365careers.com">john.mckinley@365careers.com</a>	0
2	Elizabeth	McFarlane	F	<a href="mailto:e.mcfarlane@365careers.com">e.mcfarlane@365careers.com</a>	2
3	Kevin	Lawrence	M	<a href="mailto:kevin.lawrence@365careers.com">kevin.lawrence@365careers.com</a>	1
4	Catherine	Wynfield	F	<a href="mailto:c.wynfield@365careers.com">c.wynfield@365careers.com</a>	0

purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2017	1	A_1
2	03/09/2017	2	C_1
1	04/09/2017	3	D_1
2	04/09/2017	1	B_2
3	04/09/2017	4	B_2
4	04/09/2017	2	B_1
1	05/09/2017	4	A_2
1	06/09/2017	3	C_1
2	06/09/2017	1	A_1
3	06/09/2017	2	B_1

item_code	Item	unit_price	company_id
A_1	Lamp	20	1
A_2	Desk	250	1
B_1	Lamp	30	2
B_2	Desk	350	2
C_1	Chair	150	3
D_1	Loudspeakers	400	4

# relational databases



# Spreadsheets



storing and keeping track of data

retrieval of data

updating of data

efficiency

data consistency

data integrity

speed

security

extensive analysis

=

## Relational Schemas: Primary Key

Each purchase is unique!

Sales			
purchase_number	date_of_purchase	customer_id	item_code
1	9/3/2016	1	A_1
2	12/2/2016	2	C_1
3	4/15/2017	3	D_1
4	5/24/2017	1	B_2
5	5/25/2017	4	B_2
6	6/6/2017	2	B_1
7	6/10/2017	4	A_2
8	6/10/2017	3	C_1
9	7/20/2017	1	A_1
10	8/11/2017	2	B_1

all the numbers in this column will be different

# Relational Schemas: Primary Key

## Primary Key

a column (or a set of columns) whose value exists and is unique for every record in a table is called a **primary key**

- each table can have one and only one primary key
- in one table, you cannot have 3 or 4 primary keys

Sales			
purchase_number	date_of_purchase	customer_id	item_code
1	9/3/2016	1	A_1
2	12/2/2016	2	C_1
3	4/15/2017	3	D_1
4	5/24/2017	1	B_2
5	5/25/2017	4	B_2
6	6/6/2017	2	B_1
7	6/10/2017	4	A_2
8	6/10/2017	3	C_1
9	7/20/2017	1	A_1
10	8/11/2017	2	B_1



# Relational Schemas: Primary Key

## ● Primary Key

a column (or a set of columns) whose value exists and is unique for every record in a table is called a **primary key**

- each table can have one and only one primary key
- in one table, you cannot have 3 or 4 primary keys
- primary keys are the unique identifiers of a table
- cannot contain null values!

## Sales

purchase number

date\_of\_purchase

customer\_id (FK)

item\_code (FK)

Table name: Sales

Primary key: purchase number

Other fields: date\_of\_purchase, customer\_id, item\_code

## Sales

purchase number

date\_of\_purchase

customer\_id (FK)

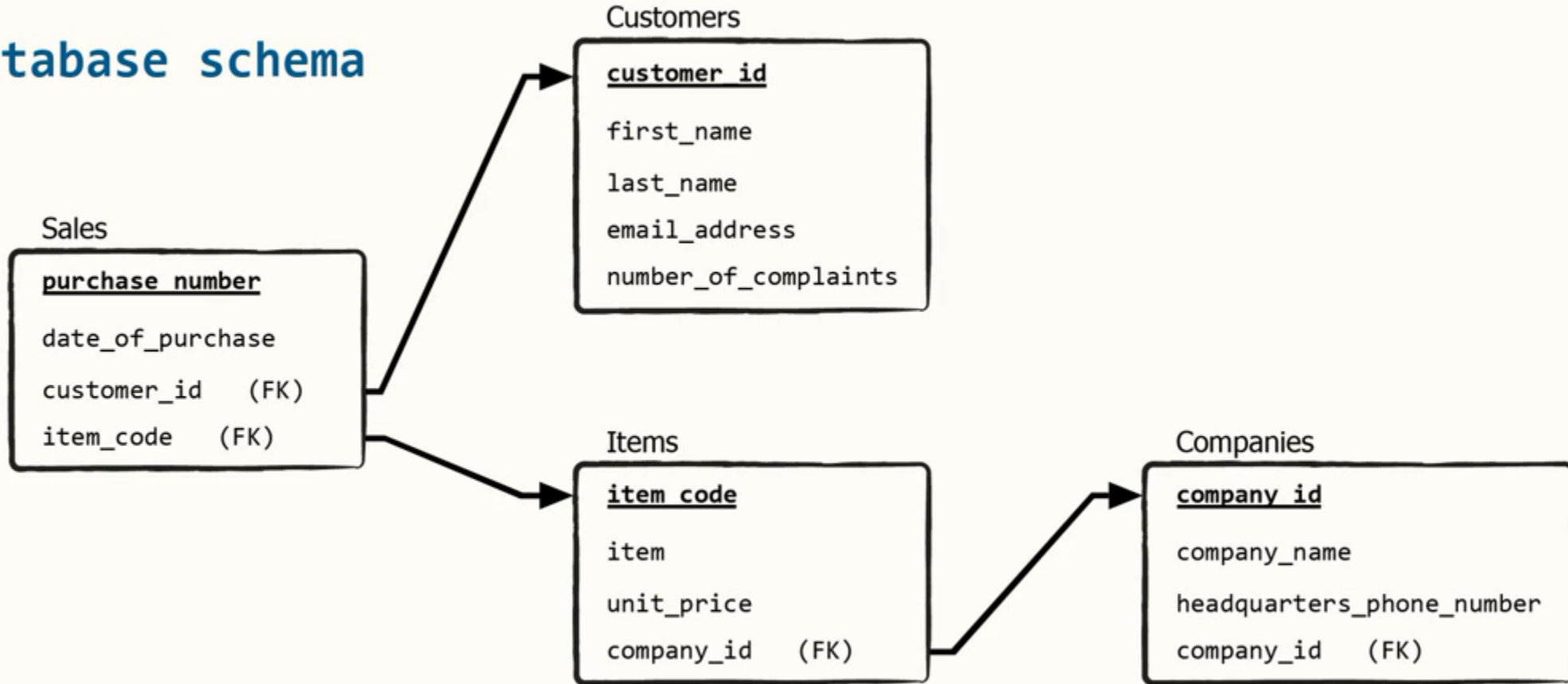
item\_code (FK)

=

## Sales

purchase_number	date_of_purchase	customer_id	item_code
1	9/3/2016	1	A_1
2	12/2/2016	2	C_1
3	4/15/2017	3	D_1
4	5/24/2017	1	B_2
5	5/25/2017	4	B_2
6	6/6/2017	2	B_1
7	6/10/2017	4	A_2
8	6/10/2017	3	C_1
9	7/20/2017	1	A_1
10	8/11/2017	2	B_1

## database schema

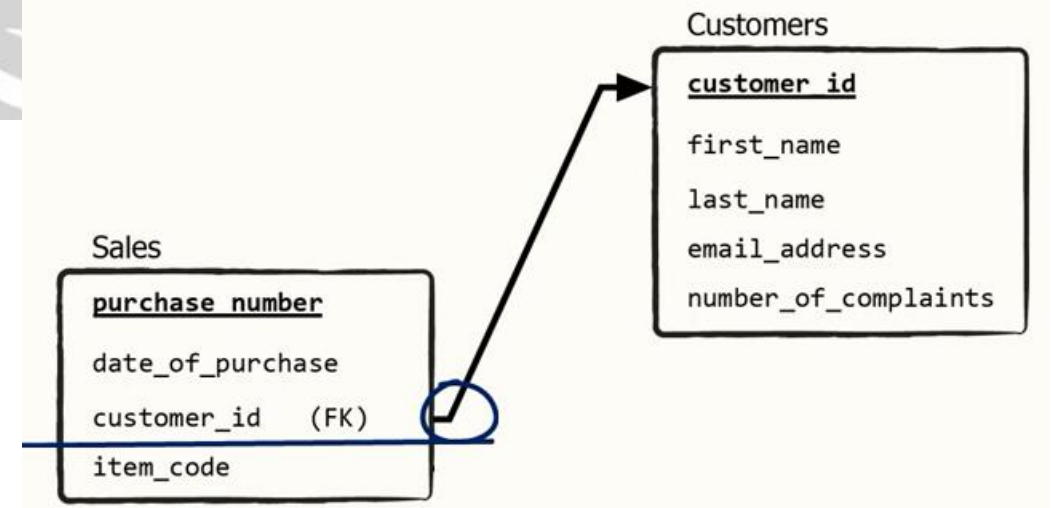


primary key

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	<a href="mailto:john.mackinley@365careers.com">john.mackinley@365careers.com</a>	0
2	Elizabeth	McFarlane	<a href="mailto:e.mcfarlane@365careers.com">e.mcfarlane@365careers.com</a>	2
3	Kevin	Lawrence	<a href="mailto:kevin.lawrence@365careers.com">kevin.lawrence@365careers.com</a>	1
4	Catherine	Winnfield	<a href="mailto:c.winnfield@365careers.com">c.winnfield@365careers.com</a>	0

Sales			
purchase_number	date_of_purchase	customer_id	item_code
1	9/3/2016	1	A_1
2	12/2/2016	2	C_1
3	4/15/2017	3	D_1
4	5/24/2017	1	B_2
5	5/25/2017	4	
6	6/6/2017	2	
7	6/10/2017	4	
8	6/13/2017	3	C_1
9	7/20/2017	1	A_1
10	8/11/2017	2	B_1

foreign key



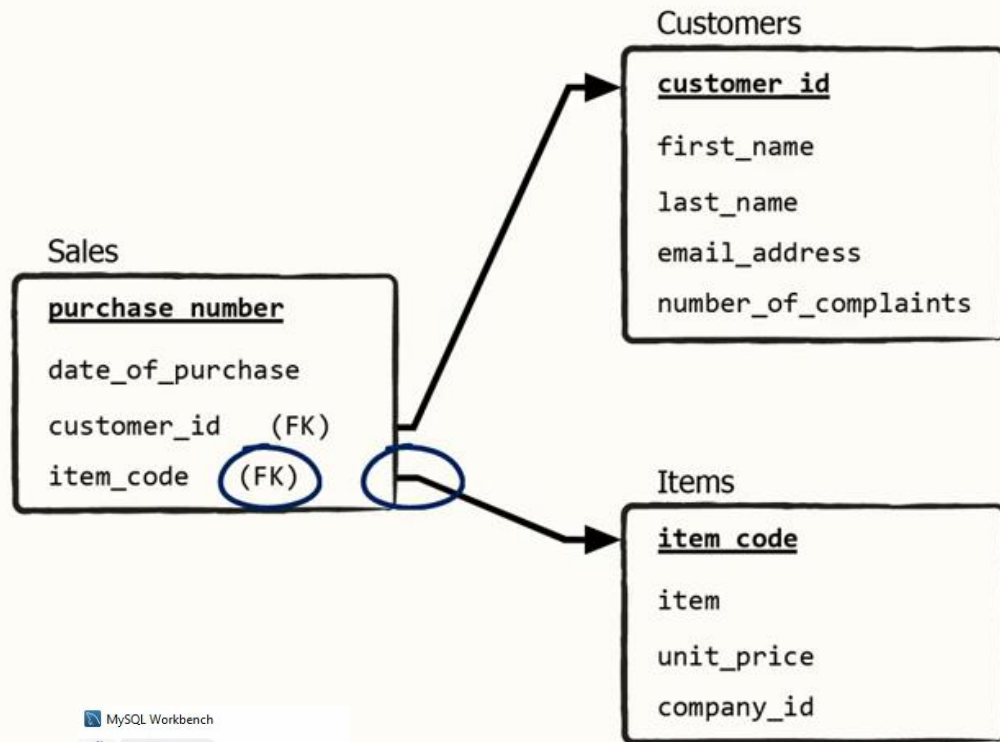


Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	<a href="mailto:john.mackinley@365careers.com">john.mackinley@365careers.com</a>	0
2	Elizabeth	McFarlane	<a href="mailto:e.mcfarlane@365careers.com">e.mcfarlane@365careers.com</a>	2
3	Kevin	Lawrence	<a href="mailto:kevin.lawrence@365careers.com">kevin.lawrence@365careers.com</a>	1
4	Catherine	Winnfield	<a href="mailto:c.winnfield@365careers.com">c.winnfield@365careers.com</a>	0

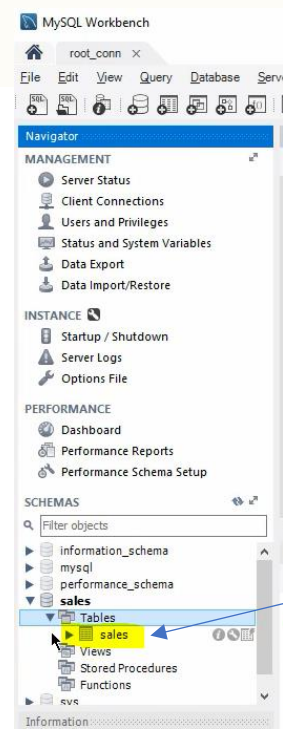
Sales				
purchase_number	date_of_purchase	customer_id	item_code	
1	9/3/2016	1	A_1	
2	12/2/2016	2	C_1	
3	4/15/2017	3	D_1	
4	5/24/2017	1	B_2	
5	5/25/2017	4	B_2	
6	6/6/2017	2	B_1	
7	6/10/2017	4	A_2	
8	6/10/2017	3	C_1	
9	7/20/2017		A_1	
10	8/11/2017	2	B_1	



~~5~~  
~~10~~  
~~100~~



	primary key	unique key
NULL VALUES	no	yes
NUMBER OF KEYS	1	0, 1, 2...
APPLICATION TO MULTIPLE COLUMNS	yes	yes



```

1 • CREATE TABLE sales
2 (
3   purchase_number INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
4   date_of_purchase DATE NOT NULL,
5   customer_id INT,
6   item_code VARCHAR(10) NOT NULL
7 );
  
```

MySQL Workbench

root\_conn x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- information\_schema
- mysql
- performance\_schema
- sales**
- sys

Using Tables\*

```
1
2 • CREATE DATABASE IF NOT EXISTS Sales;
3
4 • USE Sales;
5
6 • SELECT * FROM customers;
```

Result Grid

customer_id	first_name	last_name	email_address	number_of_complaints
-------------	------------	-----------	---------------	----------------------

customers 8 x

Output

Action Output

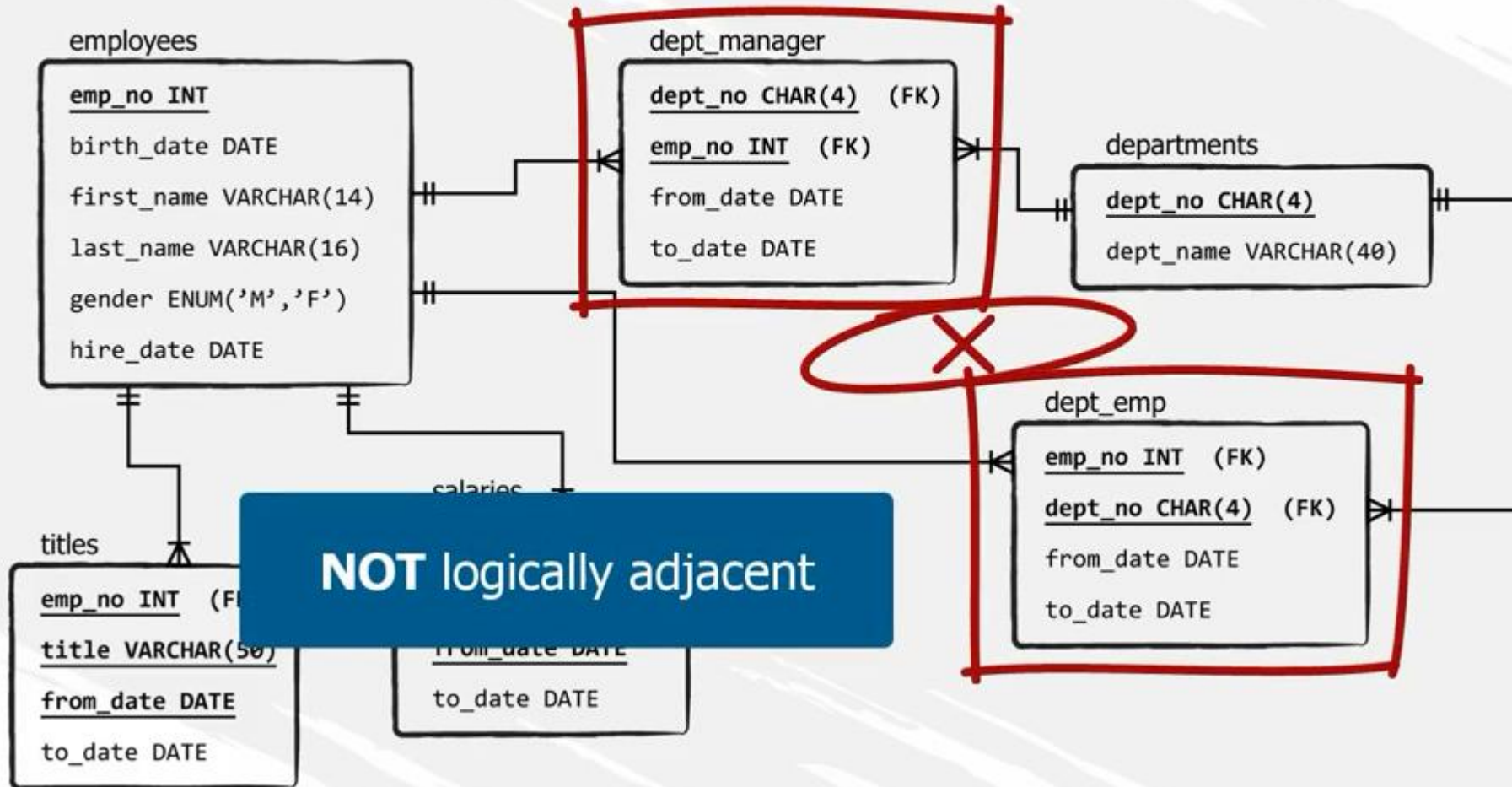
#	Time	Action	Message	Duration / Fetch
✓ 1	15:42:30	SELECT * FROM customers	0 row(s) returned	0.000 sec / 0.000 sec

# Introduction to Joins

The diagram illustrates the relationships between several database tables:

- employees**:
  - emp\_no INT
  - birth\_date DATE
  - first\_name VARCHAR(14)
  - last\_name VARCHAR(16)
  - gender ENUM('M','F')
  - hire\_date DATE
- dept\_manager**:
  - dept\_no CHAR(4) (FK)
  - emp\_no INT (FK)
  - from\_date DATE
  - to\_date DATE
- departments**:
  - dept\_no CHAR(4)
  - dept\_name VARCHAR(40)
- dept\_emp**:
  - emp\_no INT (FK)
  - dept\_no CHAR(4) (FK)
  - from\_date DATE
  - to\_date DATE
- titles**:
  - emp\_no INT (FK)
  - title VARCHAR(50)
  - from\_date DATE
  - to\_date DATE
- salaries**:
  - emp\_no INT (FK)
  - dept\_no CHAR(4) (FK)
  - from\_date DATE
  - to\_date DATE

Relationships are indicated by lines with crow's foot notation. A red box highlights the **dept\_manager** and **dept\_emp** tables, which are not logically adjacent. A red 'X' is placed over the relationship between them, and a blue box with the text **NOT logically adjacent** is overlaid on the diagram.





## INNER JOIN

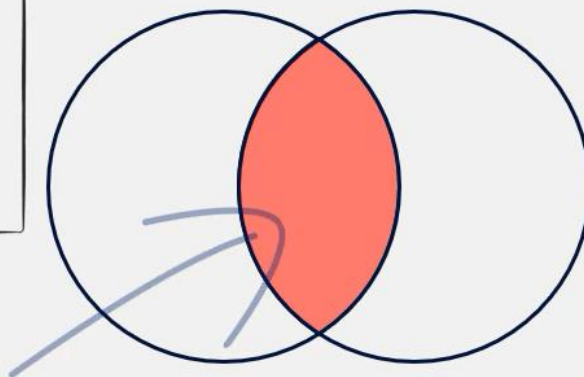
dept\_manager\_dup

dept\_no CHAR(4)

emp\_no INT

from\_date DATE

to\_date DATE



departments\_dup

dept\_no CHAR(4)

dept\_name VARCHAR(40)

### result set

the area that belongs to both circles, which is filled with **red**, represents all records belonging to *both* the “Department Manager Duplicate” and the “Departments Duplicate” tables

## INNER JOIN

dept\_manager\_dup

dept\_no CHAR(4)

emp\_no INT

from\_date DATE

to\_date DATE

dept\_no CHAR(4)



departments\_dup

dept\_no CHAR(4)

dept\_name VARCHAR(40)

## LEFT JOIN

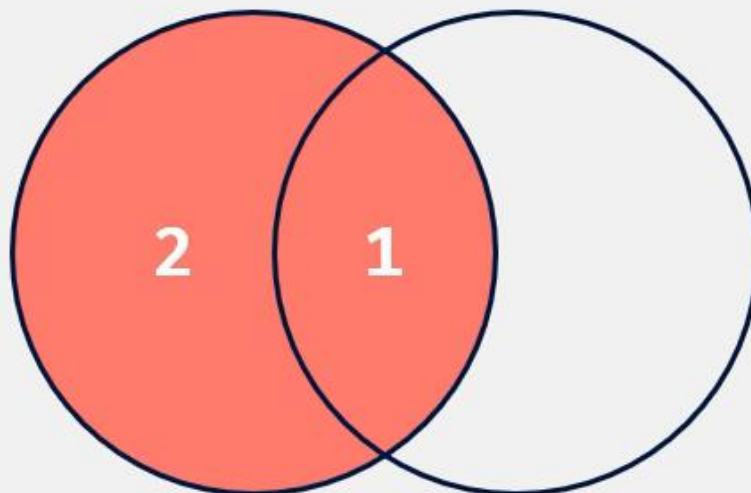
dept\_manager\_dup

dept\_no CHAR(4)

emp\_no INT

from\_date DATE

to\_date DATE



departments\_dup

dept\_no CHAR(4)

dept\_name VARCHAR(40)

- 1) all matching values of the two tables +
- 2) all values from the left table that match no values from the right table

## RIGHT JOIN

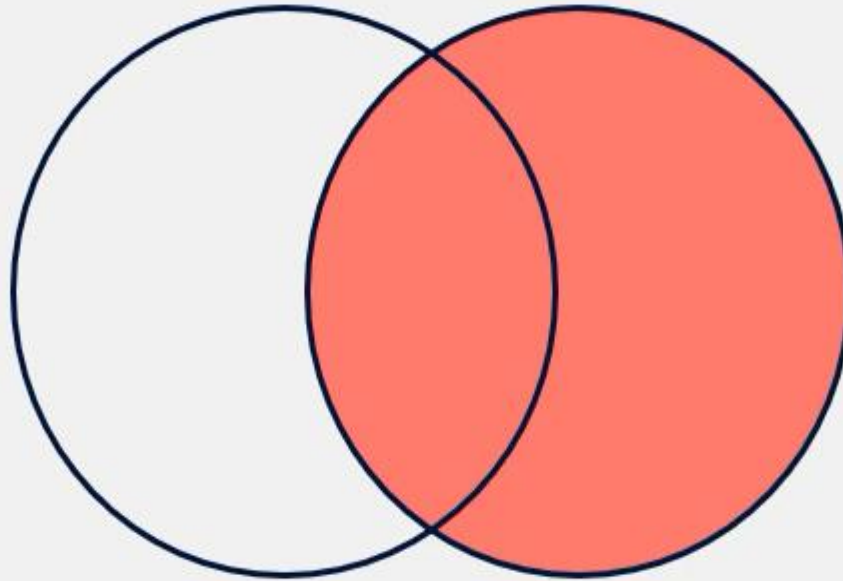
dept\_manager\_dup

dept\_no CHAR(4)

emp\_no INT

from\_date DATE

to\_date DATE

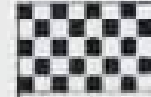


departments\_dup

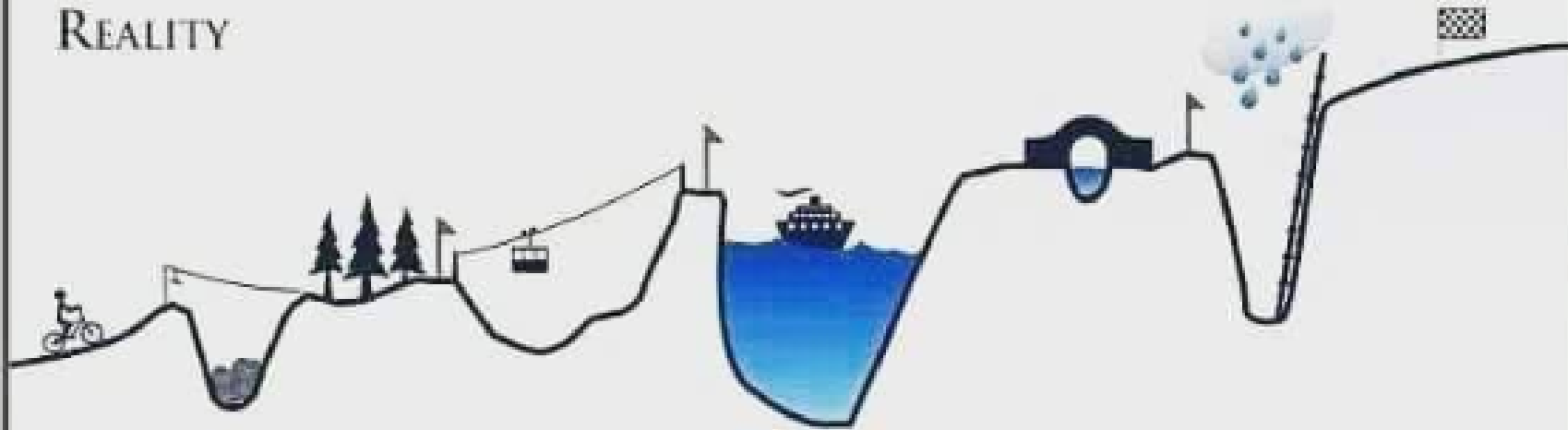
dept\_no CHAR(4)

dept\_name VARCHAR(40)

**DATA**



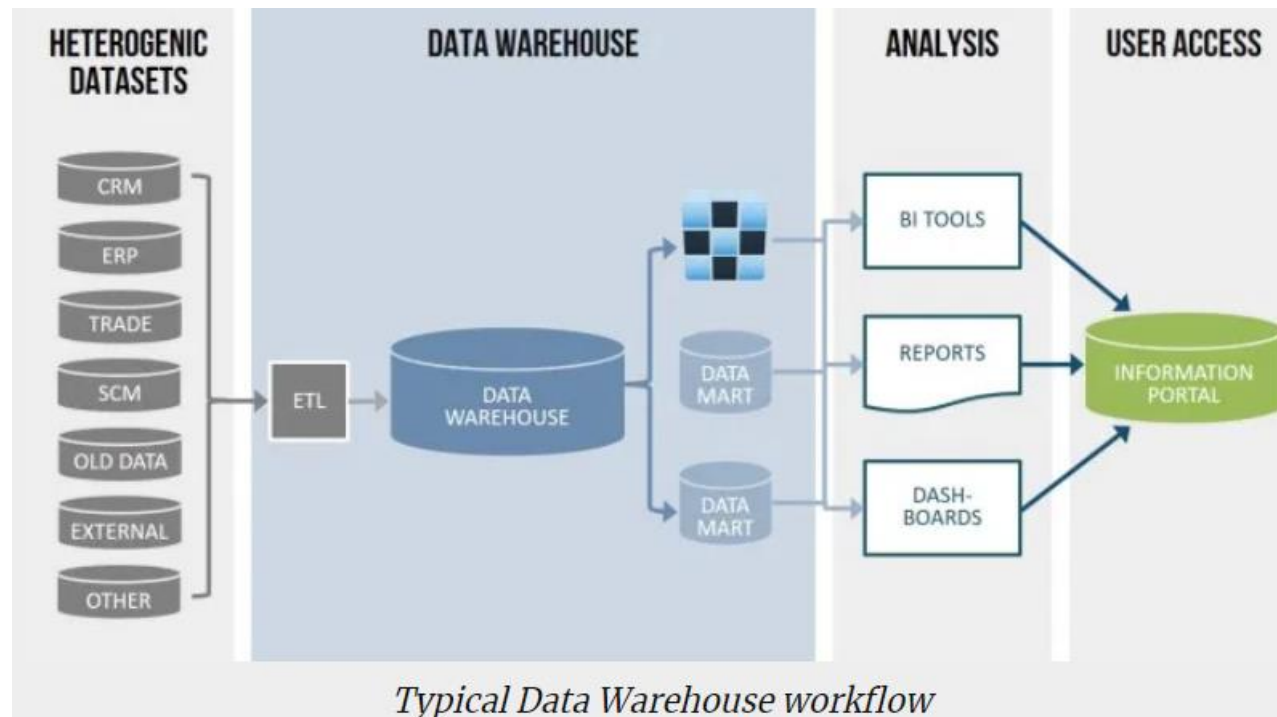
**REALITY**



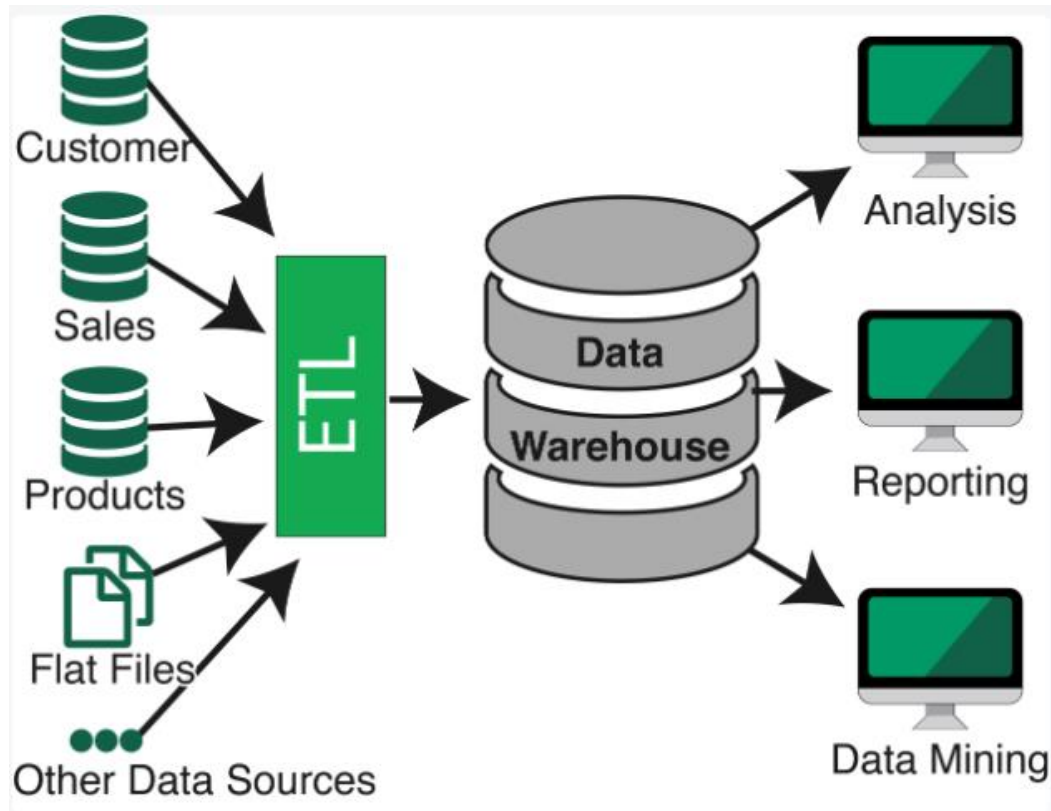




*Typical Data Science workflow*



*Typical Data Warehouse workflow*



**ETL = Extract, Transform, Load**

**Why ETL?**

Need to load the data warehouse regularly (daily/weekly) so that it can serve its purpose of facilitating business analysis.

**Extract** - data from one or more OLTP systems and copied into the warehouse



**Transform** – removing inconsistencies, assemble to a common format, adding missing fields, summarizing detailed data and deriving new fields to store calculated data.

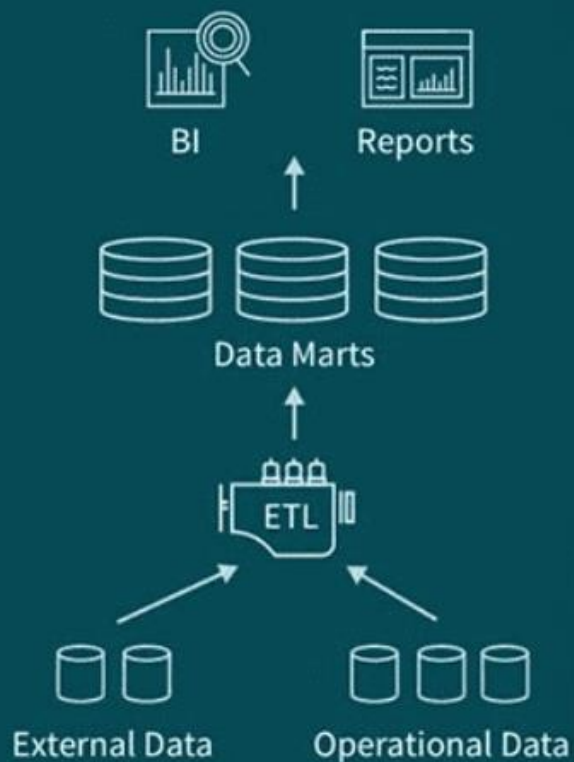


**Load** – map the data and load it into the DW



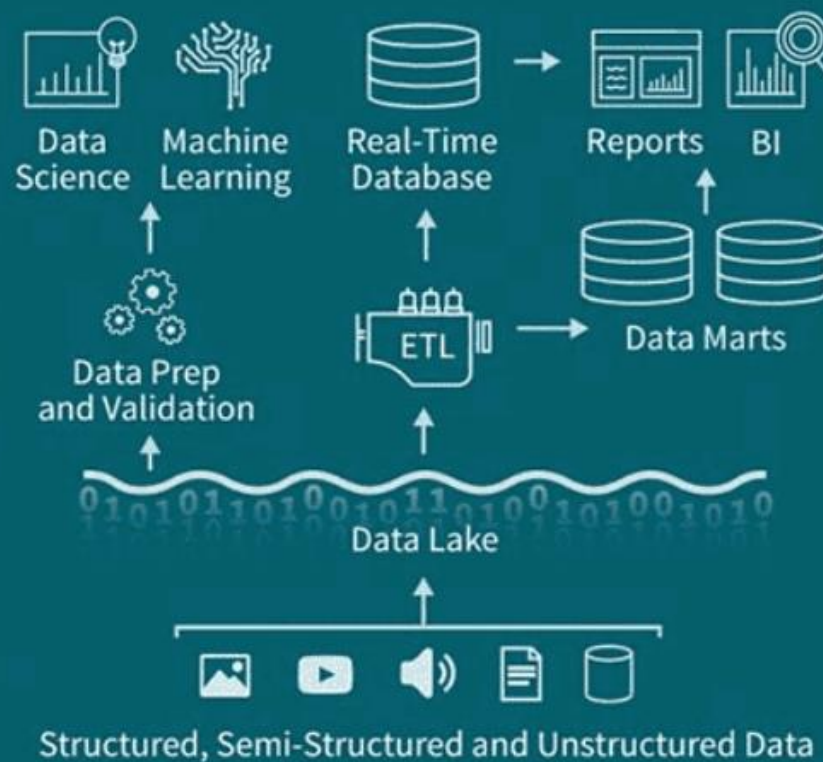
Late 1980's

## Data Warehouse



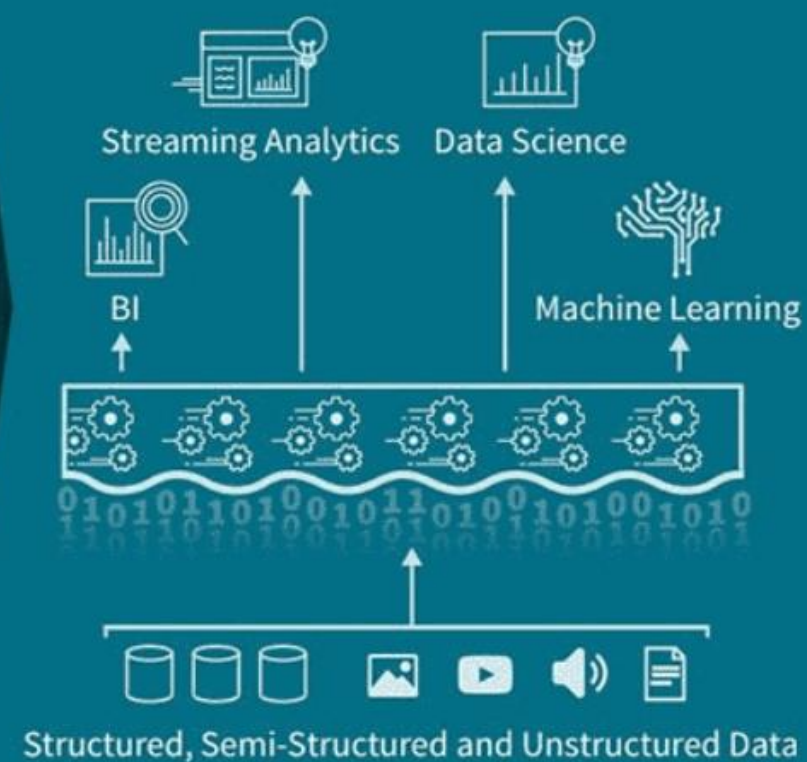
2011

## Data Lake



2020

## Lakehouse

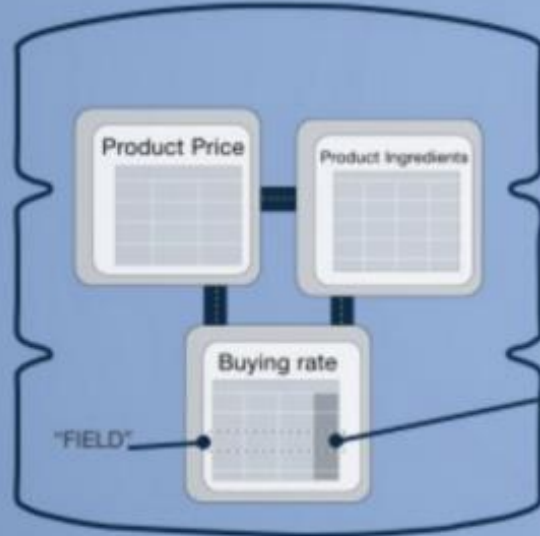


SOURCE: DATABRICKS

# SQL

# VS

# NoSQL



A NON-RELATIONAL DATABASE DOES NOT INCORPORATE THE TABLE MODEL. INSTEAD, DATA CAN BE STORED IN A SINGLE DOCUMENT FILE.

A RELATIONAL DATABASE TABLE ORGANIZES STRUCTURED DATA FIELDS INTO DEFINED COLUMNS.





		Relational		Non-Relational
Analytics	Proprietary Storage	Amazon Redshift EMC Greenplum HP Vertica	IBM Netezza Oracle Teradata MPP	
	Hadoop Storage	Cloudera Impala Presto	Hive SQL-on-Hadoop	MapReduce
Operational		Traditional SQL	NewSQL	NoSQL
	Proprietary Storage	Oracle DB2 SQL Server MySQL	User-Sharded MySQL NuoDB Clustrix On-Disk MemSQL VoltDB In-Memory	Key Value: Aerospike, Riak Column Family: Cassandra Document: MongoDB Graph: Neo4j, InfiniteGraph
	Hadoop Storage		Splice Machine On-Hadoop	Column Family: HBase

## Relational

Tend to be larger,  
monolithic



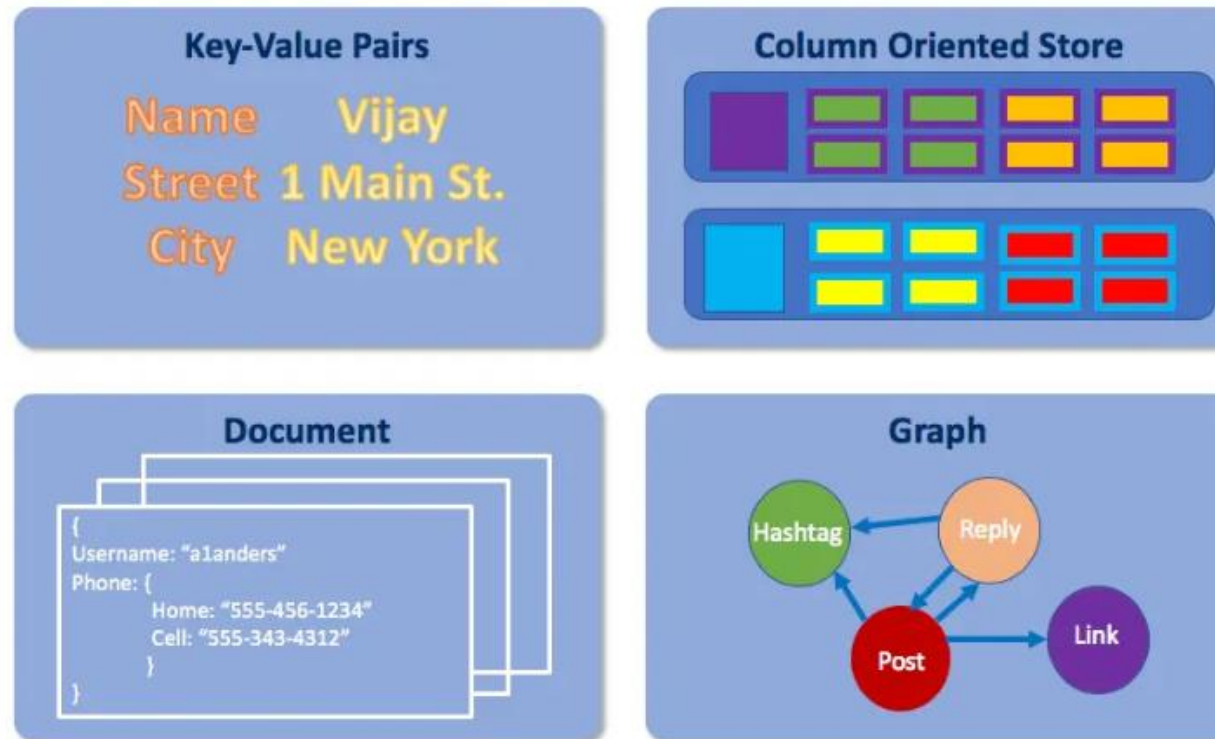
## Non-relational

Newer field, lots  
of players



The key difference between a NoSQL and SQL is that a SQL database is considered a relational database. **A relational database stores data in tables, which are organized into columns.** Each column stores one datatype (integer, real number, string, date etc.) and each row represents an instance of the table. Non-relational databases do not store data in tables- instead there are multiple ways to store data in NoSQL databases (Key-value, Document-based, Column-based).

### Types of Non-relational Databases



# PROGRAMMING + SOFTWARE APPLICATIONS



- specifically designed for the domain of the RDBMS



**Tableau**

- business intelligence and analytics
- visualizations of datasets



- operating systems
- graphic design applications



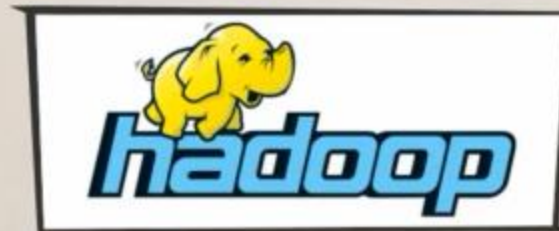
- complex mathematical computations
- business, statistics, finance



# PROGRAMMING + SOFTWARE APPLICATIONS



Tableau







# Tableau

- graphs
- charts
- reports
- dashboards

allow end users  
to understand the core of a business  
+ extract insights

# PROBLEM STRUCTURE

- 1.** Receive a business task
- 2.** Use SQL to execute a query retrieving a relevant dataset from the database
- 3.** Export the newly obtained data in a CSV file to be used in Tableau
- 4.** Create a professional and understandable visualization in Tableau

Data source

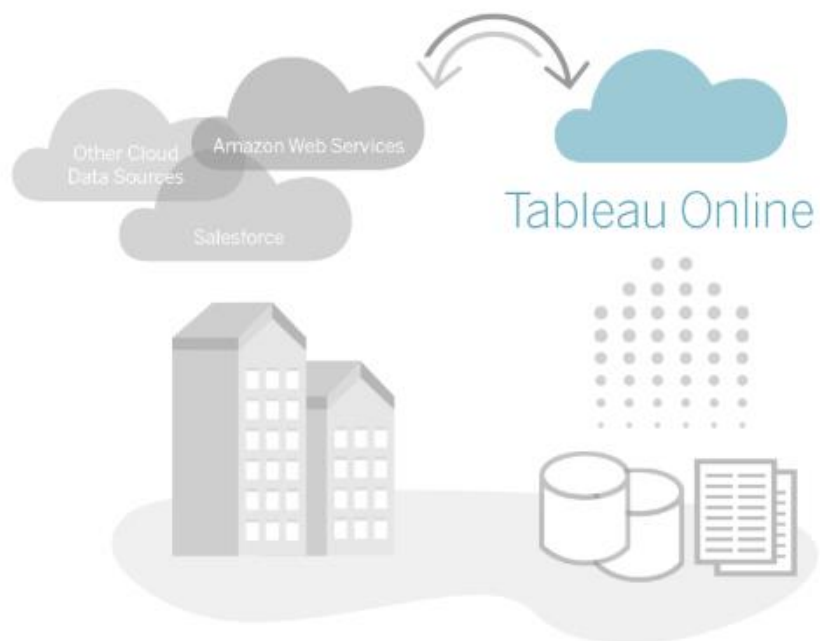
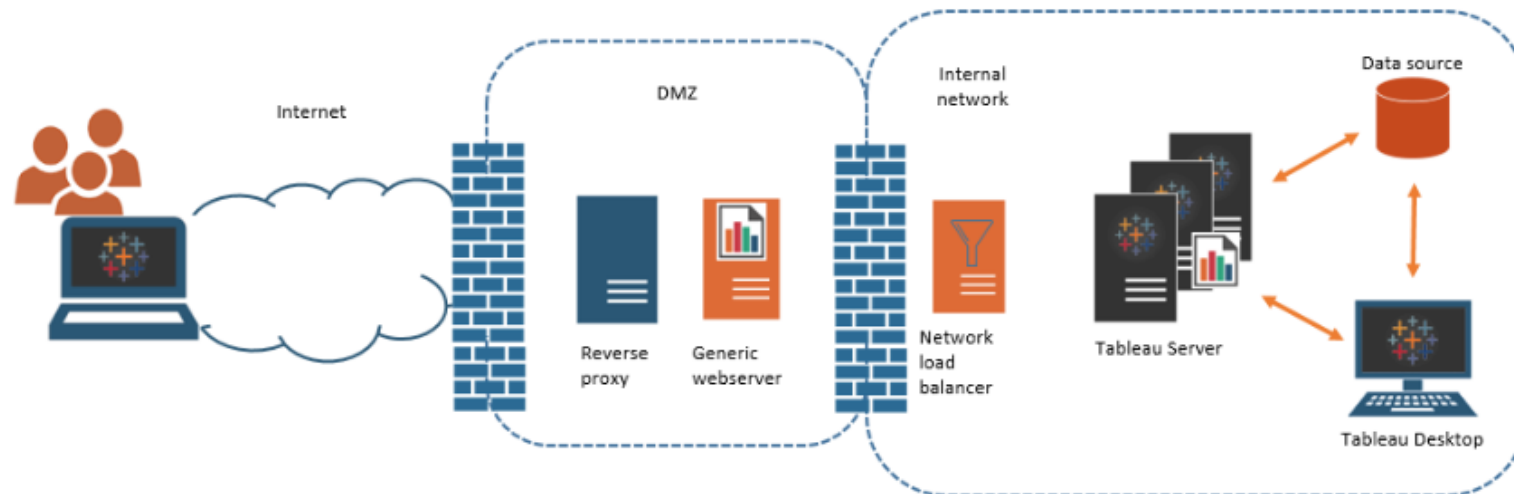


Tableau Desktop



Version 8.3

Tableau Reader



**Desktop**  
Create Your Visualizations



**Mobile**



**Server**  
Consume Your Visualizations Using  
a Variety of Delivery  
Methods

**Server**  
Standardize  
Your  
Visualizations



**Web**



**Desktop**



**Tableau Online**



Live connections



Redshift



MySQL



PostgreSQL



SQLServer



BigQuery



