



/ JAVA DB SCHOOL

Databases



/ Data collections

Data collections

1. What is data?
2. Why do we need it?
3. Where can we use it?



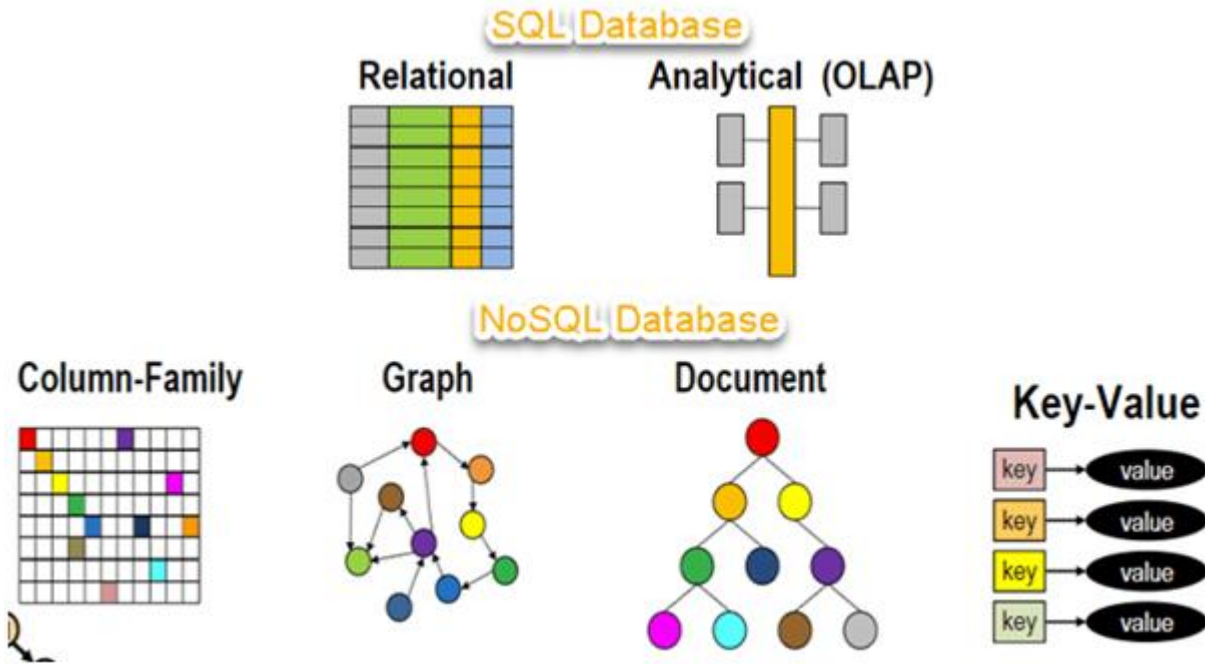
/ Databases

Databases

- Database:
 - Organized data collection
 - Stored / accessed electronically on a computer system
- DBMS:
 - **DataBase Management System**
 - Software that allows users / other programs to interact with a database in terms of reading / writing data and database administration



Database types (1)



Reference: <https://www.gologica.com/elearning/comprehensive-nosql-tutorial-for-beginners/>

Database types (2)

- Relational
- Non-relational

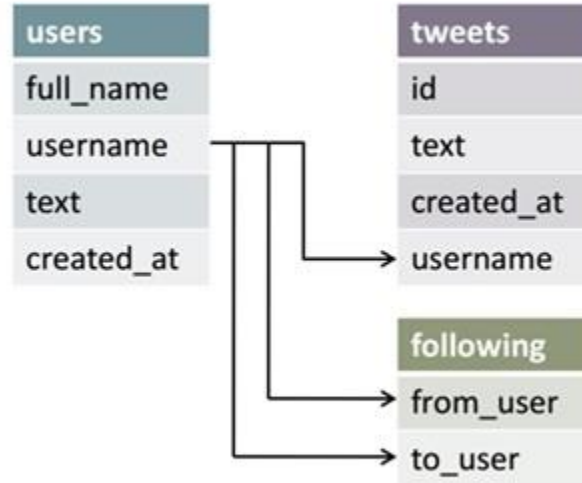


Relational databases (1)

- These databases are categorized by a set of tables where data gets fit into a pre-defined category
- The table consists of rows and columns where the column has an entry for data for a specific category and rows contains instance for that data defined according to the category
- The Structured Query Language (SQL) is the standard user and application program interface for a relational database
- There are various simple operations that can be applied over the table which makes these databases easier to extend, join two databases with a common relation and modify all existing applications



Relational databases (2)



Non-relational databases (1)

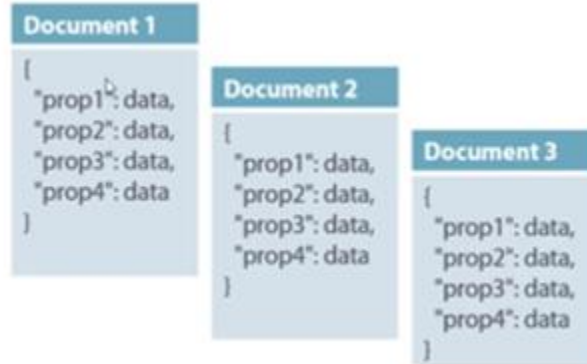
- These databases are categorized by a set of tables where data gets fit into a pre-defined category
- These are used for large sets of distributed data
- There are some big data performance issues which are effectively handled by relational databases, such kind of issues are easily managed by NoSQL databases
- There are very efficient in analyzing large size unstructured data that may be stored at multiple virtual servers of the cloud



Non-relational databases (2)

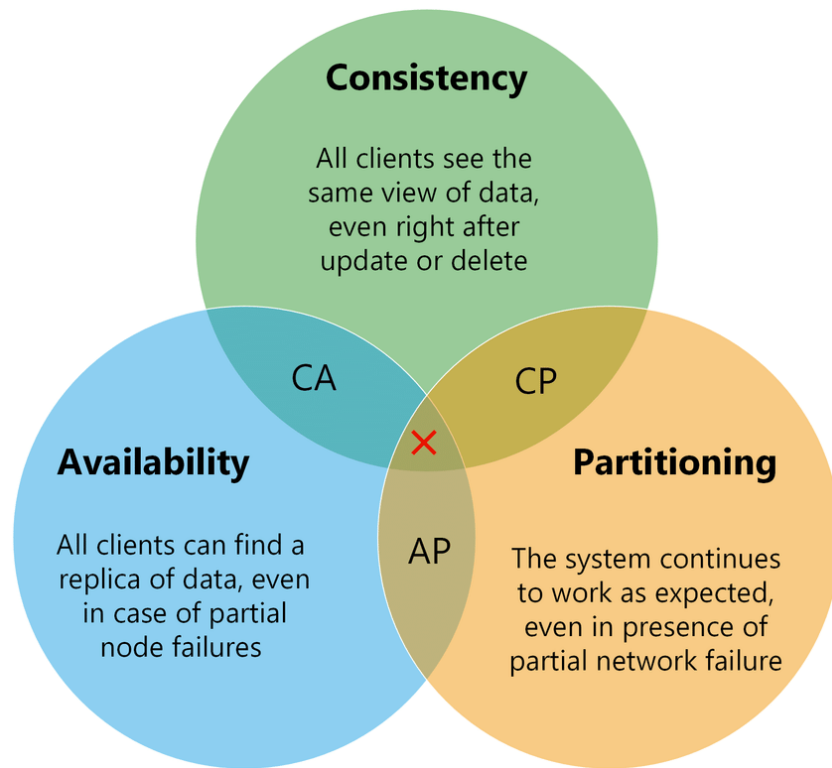
- Document-based – stores and retrieves the data as the key-value pairs but the value part is stored as the document
- The document is stored in the JSON or XML formats

Col1	Col2	Col3	Col4
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data



CAP Theorem

- The CAP theorem – a distributed database system can only guarantee two out of these three characteristics:
 - Consistency
 - Availability
 - Partition Tolerance



/ MySQL



MySQL

- Enables you to implement a database with tables, columns and indexes
- Guarantees the Referential Integrity between rows of various tables
- Updates the indexes automatically
- Interprets an SQL query and combines information from various tables



/ MySQL Workbench



MySQL Workbench

- Visual database design tool
- Integrates SQL development, administration, database design, creation and maintenance
- Install: <https://www.mysql.com/products/workbench/>



/ MySQL Hands-on



Create

- Database

```
CREATE DATABASE db_name;
```

- Table

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



Constraints

- NOT_NULL – ensures a column cannot have null value
- UNIQUE – ensures all values in a column are different
- PRIMARY_KEY – uniquely identifies each row in a table
- FOREIGN_KEY – link between tables



AUTO_INCREMENT

- Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table
- Often this is the primary key field that we would like to be created automatically every time a new record is inserted
- By default, the starting value of AUTO_INCREMENT is 1, and it is incremented by 1 for each record



Common data types – numeric

- INT – A normal-sized integer that can be signed or unsigned
- FLOAT(M,D) – A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D)
- DOUBLE(M,D) – A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D)
- DECIMAL(M,D) – An unpacked floating-point number that cannot be unsigned. In the unpacked decimals, each decimal corresponds to one byte. Defining the display length (M) and the number of decimals (D) is required



Common data types – data

- DATE – A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30
- DATETIME – A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.



Common data types – string

- VARCHAR(M) – A variable-length string between 1 and 255 characters in length. For example, VARCHAR(25)
- BLOB or TEXT – A field with a maximum length of 65535 characters. BLOBs are "Binary Large Objects" and are used to store large amounts of binary data, such as images or other types of files. Fields defined as TEXT also hold large amounts of data. The difference between the two is that the sorts and comparisons on the stored data are case sensitive on BLOBs and are not case sensitive in TEXT fields



Primary key

- Must contain UNIQUE values
- Cannot contain NULL values
- A table can have only one primary key that can consist of single or multiple columns

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);
```



Compound primary key

- A table can have only one primary key that can consist of single or multiple columns

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT PK_Person PRIMARY KEY (ID,LastName)  
);
```



Foreign key

- Defines links between tables
- Is a field or a collection of fields in one table that refers to the PRIMARY_KEY in another table
- Parent table (referenced) – the table with the primary key
- Child table – the table with the foreign key



Foreign key

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);
```



/ Data manipulation



Select

SELECT field1, field2,...fieldN

FROM table_name1, table_name2...

[WHERE Clause]

[OFFSET M][LIMIT N]



Insert

```
INSERT INTO table_name ( field1, field2,...fieldN )
```

```
VALUES
```

```
( value1, value2,...valueN );
```



Update

```
UPDATE table_name SET field1 = new-value1, field2 = new-value2
```

```
[WHERE Clause]
```



Delete

```
DELETE FROM table_name [WHERE Clause]
```



/ SQL Joins

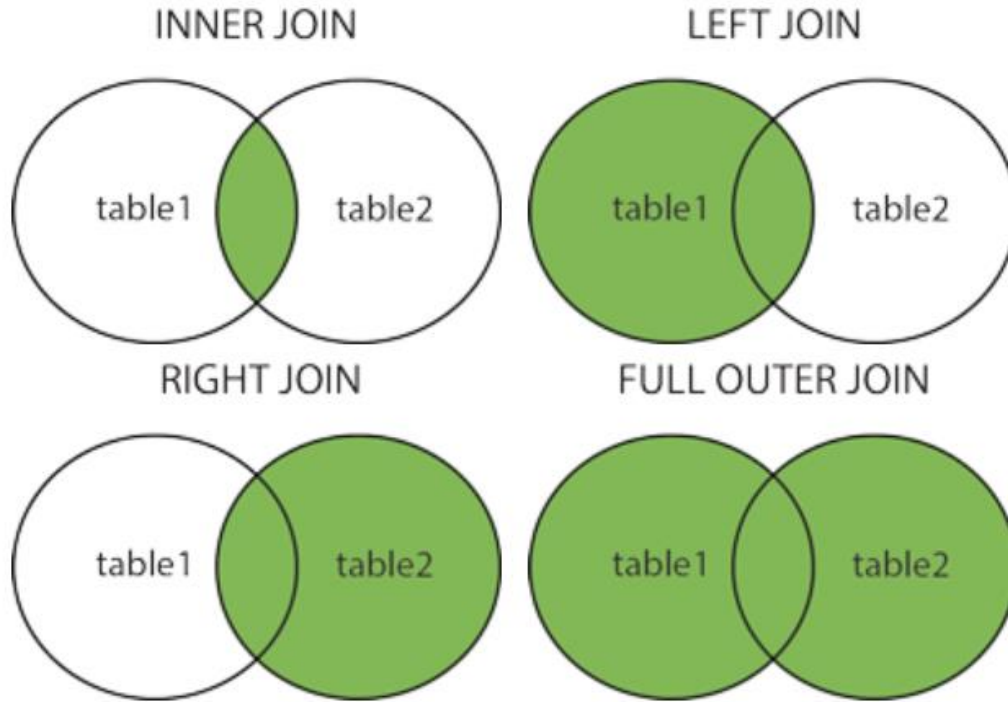


Joins

- A JOIN clause is used to combine rows from two or more tables based on a related column between them
- Types of JOIN:
 - (INNER) JOIN – returns records that have matching values in both tables
 - LEFT (OUTER) JOIN – returns all records from the left table and the matched records from the right table
 - RIGHT (OUTER) JOIN – returns all records from the right table and the matched records from the left table
 - FULL (OUTER) JOIN – returns all records when there is a match in either left or right table



Joins



Reference: https://www.w3schools.com/sql/sql_join.asp



/ Cheat sheet



Cheat sheet (1)

- Database – A database is a collection of tables, with related data
- Table – A table is a matrix with data. A table in a database looks like a simple spreadsheet
- Column – One column (data element) contains data of one and the same kind
- Row – A row (= tuple, entry or record) is a group of related data



Cheat sheet (2)

- Redundancy – Storing data twice, redundantly to make the system faster.
- Primary Key – A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- Foreign Key – A foreign key is the linking pin between two tables



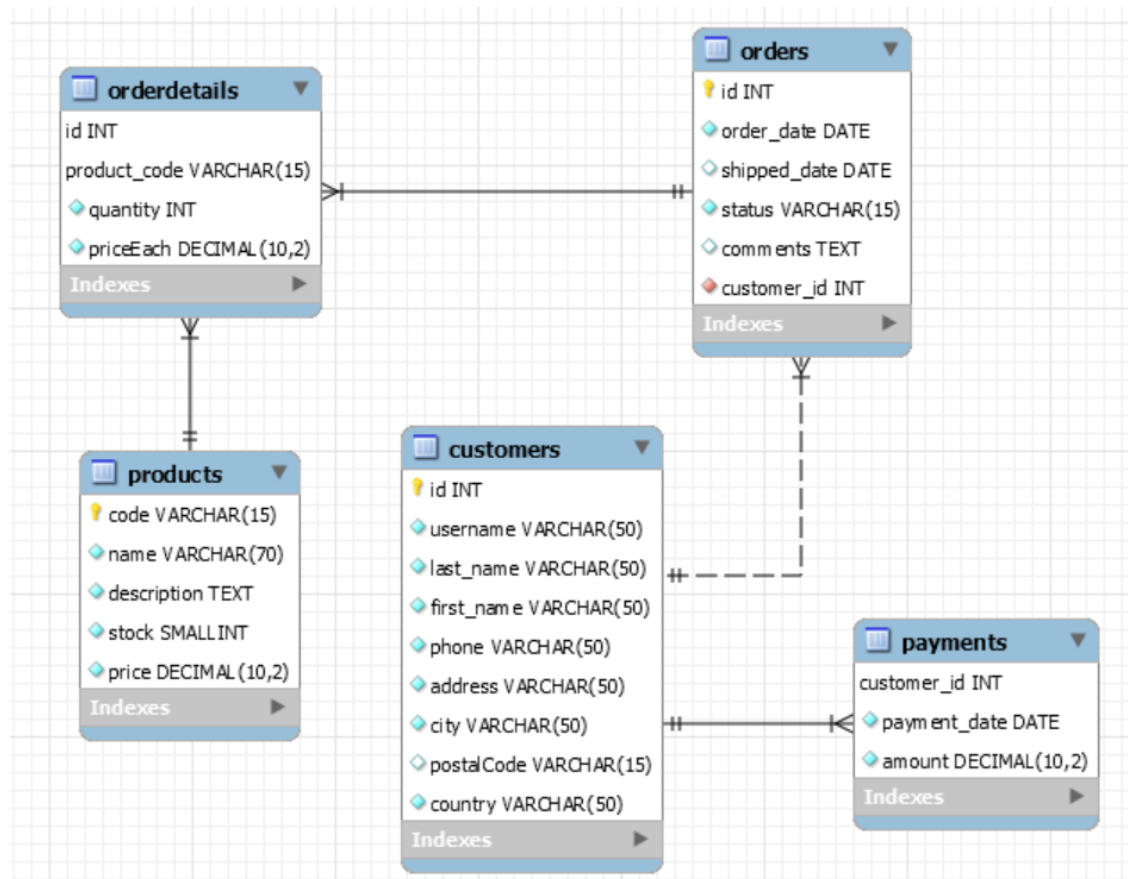
Cheat sheet (3)

- Compound Key – A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique
- Index – Indexes are special lookup tables that the database search engine can use to speed up data retrieval
- Referential Integrity – Referential Integrity makes sure that a foreign key value always points to an existing row.



/ Practice, practice, practice





/ Homework



Homework

- Create the remaining database tables and their associated relationships, if any left
- Implement the following through SQL queries:
 - Add 5 customers, 10 products, 5 orders and 3 payments
 - Update the status of one order
 - Add comments to one order
 - View all customers
 - View all orders for the third customer
 - When placing an order update the stock for the products – try to implement with a small number of queries and explain your choice



/ Q&A





MOBILE / ACADEMY