**SQL stands for Structured Query Language**. It is used for storing and managing data in relational database management system (RDMS). It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.

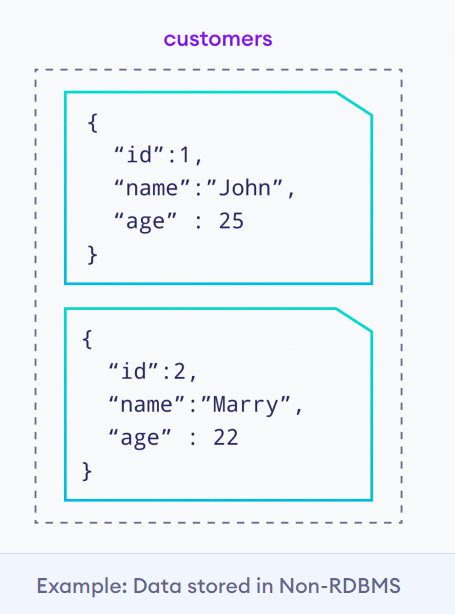
## **Types of DBMS - Database Management System**

In general, there are two common types of databases:

* Non-Relational - Non-RDMS - NoSQL
* Relational - RDMS - SQL

### **Non-Relational Database Management System (Non-RDBMS)**

In Non-RDBMS, data is stored in key-value pairs. For example:



Commonly used Non-RDBMS: MongoDB, Amazon DynamoDB, Redis, etc.

## 

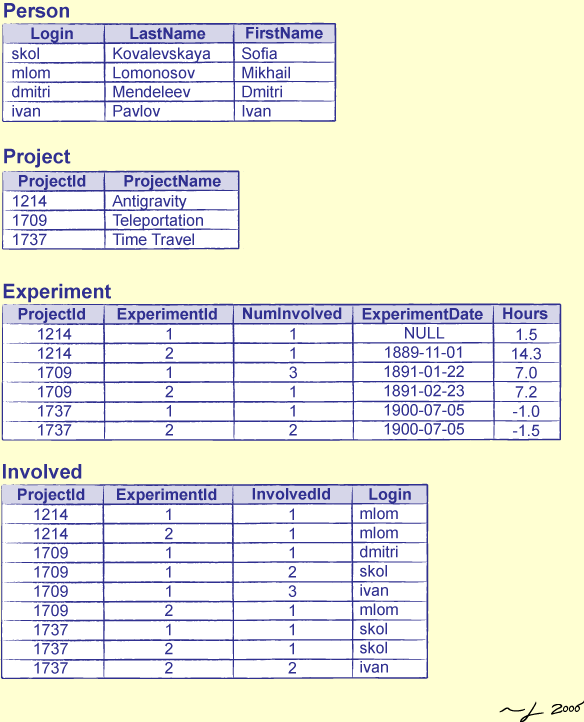
## **RDBMS**

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.



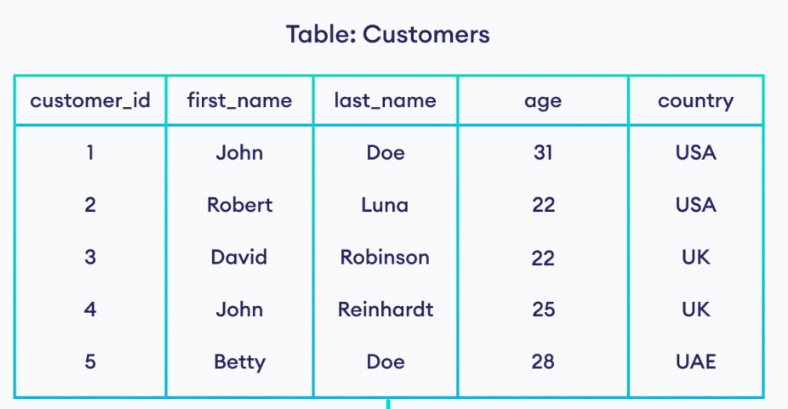


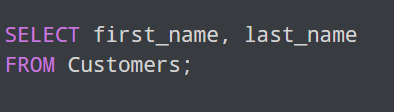
Structured Query Language (SQL) is a standard query language that is used to work with relational databases.

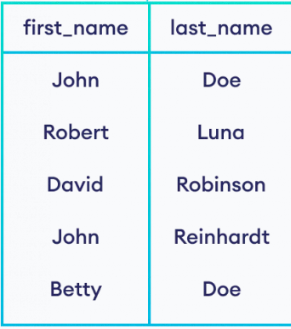
We use SQL to

* create databases
* create tables in a database
* read data from a table
* insert data in a table
* update data in a table
* delete data from a table
* delete database tables
* delete databases

The SQL SELECT statement is used to select (retrieve) data from a database table. For example,







# About MySQL

MySQL is a open-source, free and very popular relational database management system which is developed, distributed and supported by Oracle corporation.

# Syntax help

## Commands

### 1. CREATE

**CREATE** **TABLE** table\_name (

column1 datatype,

column2 datatype,

....);

### Example

**CREATE** **TABLE** EMPLOYEE (

empId **INTEGER** **PRIMARY** **KEY**,

name TEXT **NOT** **NULL**,

dept TEXT **NOT** **NULL**

);

### 2. ALTER

**ALTER** **TABLE** Table\_name **ADD** column\_name datatype;

### Example

**INSERT** **INTO** EMPLOYEE **VALUES** (0001, 'Dave', 'Sales');

### 3. TRUNCATE

**TRUNCATE** **table** table\_name;

### 4. DROP

**DROP** **TABLE** table\_name;

### 5. RENAME

RENAME **TABLE** table\_name1 **to** new\_table\_name1;

### 6. COMMENT

### Single-Line Comments:

*--Line1;*

### Multi-Line comments:

*/\* Line1,*

*Line2 \*/*

## DML Commands

### 1. INSERT

**INSERT** **INTO** table\_name (column1, column2, column3, ...) **VALUES** (value1, value2, value3, ...);

Note: Column names are optional.

### Example

**INSERT** **INTO** EMPLOYEE **VALUES** (0001, 'Ava', 'Sales');

### 2. SELECT

**SELECT** column1, column2, ...

**FROM** table\_name

[**where** **condition**];

### Example

**SELECT** \* **FROM** EMPLOYEE **where** dept ='sales';

### 3. UPDATE

**UPDATE** table\_name

**SET** column1 = value1, column2 = value2, ...

**WHERE** **condition**;

### Example

**UPDATE** EMPLOYEE **SET** dept = 'Sales' **WHERE** empId='0001';

### 4. DELETE

**DELETE** **FROM** table\_name **where** **condition**;

### Example

**DELETE** **from** EMPLOYEE **where** empId='0001';