

# Bank system

---



# Web store (eCommerce)



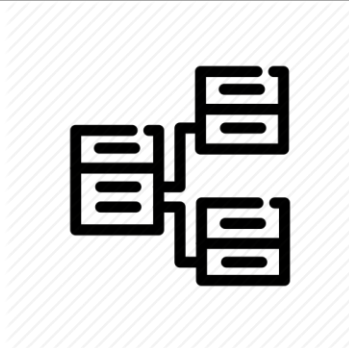
# Database



A **database** is an organized collection of data. It is the collection of schemas, tables, queries, views, stored procedures, and other objects.



A **database management system (DBMS)** is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data.



**Relational database management system (RDBMS)** is a type of dbms having relationships between the tables using indexes and different constraints like primary key, foreign key etc.

# Table

A **table** is a collection of related data held in a structured format within a database. It consists of columns, and rows.

CustomerId	FirstName	LastName	Email	Phone	AltPhone	FaxNumber	ZipCode	NewsLetter
1	John	Smith	John@gmail.com	703-543-3302	703-543-3302	NULL	22201	1
2	Jeremy	Smith	Jeremy@gmail.com	723-543-3302	NULL	NULL	22203	0
3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	NULL	NULL	22031	1
4	Bob	James	bob@microsoft.com	703-366-9632	NULL	703-455-9632	22221	0
5	Adam	Marcos	adam@Marcos.com	703-566-0000	NULL	703-366-0000	22001	1

# Database design

## Customers + Schedules relationship

CustomerId	FirstName	LastName	Email	Phone	ZipCode	ScheduleDescription	DateNeeded	JobType
1	John	Smith	John@gmail.com	703-543-3302	22201	Kitchen remodel needed	2013-10-10	Remodeling
2	Jeremy	Smith	Jeremy@gmail.com	723-543-3302	22203	Decorating help for dinig room	2013-10-15	Decorating
3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	22031	Kitchen remodel needed	2015-11-29	Remodeling
3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	22031	Garade rebuild	2016-12-31	Rebuild

### Problems:

1. Duplicated data
2. Updated problem
3. Possible data ambiguity

# Database design (Normalization)

## Solution

Each table contains information about single functional item.

Primary key

## Customers

CustomerId	FirstName	LastName	Email	Phone	ZipCode
1	John	Smith	John@gmail.com	703-543-3302	22201
2	Jeremy	Smith	Jeremy@gmail.com	723-543-3302	22203
3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	22031

Relationship



## Schedules

ID	CustomerId	Description	DateNeeded	Job Type
1	1	Kitchen remodel needed	2013-10-10	Remodeling
2	2	Decorations help for dinig room	2013-10-15	Decorating
3	3	Kitchen remodel needed	2015-11-29	Remodeling
4	3	Garade rebuild	2016-12-31	Rebuild

Foreign Key

# HomePro

HomePro.Customers

Primary Key	CustomerId	FirstName	LastName	Email	Phone	AltPhone	FaxNumber	ZipCode	NewsLetter	State	Age
	1	John	Smith	John@gmail.com	703-543-3302	703-543-3302	NULL	22201	1	VA	18
	2	Jeremy	Smith	Jeremy@gmail.com	723-543-3302	NULL	NULL	22203	0	NY	23
	3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	NULL	NULL	22031	1	CA	64
	4	Bob	James	bob@microsoft.com	703-366-9632	NULL	703-455-9632	22221	0	VA	37
	5	Adam	Marcos	adam@Marcos.com	703-566-0000	NULL	703-366-0000	22001	1	NC	41

HomePro.Schedules

Primary Key	ID	CustomerId	Description	DateNeeded	Job Type
	1	1	Kitchen remodel needed	2013-10-10	Remodeling
	2	2	Decorating help for dinig room	2013-10-15	Decorating
	3	3	Kitchen remodel needed	2015-11-29	Remodeling
	4	3	Garade rebuild	2016-12-31	Rebuild

Foreign key

HomePro.Quites

Primary Key	ID	CustomerId	Description	Estimation
	1	1	Kitchen remodel	210.55
	2	3	Quote with discount	875.55
	3	3	Quote with additional work	10000.00

Foreign key

# Primary key

---

**Primary key** – the column(s) that has completely unique data throughout the table

The main role of a primary key in a data table is to maintain the internal integrity of a data table.

Table can have only one primary key.



# Foreign key

---

**Foreign key** – the column that links one table to another table's primary key or unique constraint

Table can have any number of foreign keys defined.

# Structured Query Language (SQL)

SQL is a language used for creating, storing, fetching and updating of data and database objects in RDBMS.

---

# SELECT

---

**SELECT** is used to retrieve rows selected from one or more tables.

Basic syntax:

```
SELECT <columns>  
FROM <Table>  
WHERE <condition>  
ORDER BY <columns>
```

# Select (examples)

---

```
Select * from HomePro.Customers;
```

```
Select FirstName, LastName  
From HomePro.Customers  
Order by LastName;
```

# Where (char, varchar)

---

Select \* from HomePro.Customers

Where LastName = 'Smith'

Select \* from HomePro.Customers

Where LastName like 'S%'

Select \* from HomePro.Customers

Where LastName like '\_m%'

# Where (numbers)

---

1. Where Age = 10
2. Where Age > 10
3. Where Age > 10 and Age < 40
4. Where Age >= 10 and Age <= 40
5. Where Age between 10 and 40
6. Where Age in (10, 20, 30)

# Where (date)

---

1. Where DateNedeed = '2015-11-29'
2. Where DateNedeed > '2014-12-30'
3. Where DateNedeed between '2015-12-01' and '2015-12-30'

# NULL values

---

- NULL is an unknown and undefined value.
- Arithmetic operation with NULL in SQL will return a NULL.
- ~~Where Value = Null~~ -> Where Value is Null

Question: How much money do you have?

- Answer 1. I have \$10                      Meaning: \$10
- Answer 3. I have no money              Meaning: \$0
- Answer 3. I don't tell you              Meaning: **Null**



# Query with NULL

---

Select \*

from **HomePro.Customers**

where AltPhone is null;

# Aggregate functions

---

Count ()

Sum ()

Max()

Min ()

Avg()

# Aggregate (examples)

---

Select Count(\*), Avg(Age)

From **Bank.Client**;

Select Count(\*), Avg(Age)

From **Bank.Client**

Where Type = 'private';

# Aggregate (examples)

---

Select ClientId, Max(Balance), Min(Balance)

From **Bank.Client**

Where Type = 'CREDIT'

Group by ClientId;

Select ClientId, Avg(Balance), Sum(Balance)

From **Bank.Client**

Group by ClientId

Having Avg(Balance) > 10000;