Introduction to RDBMS

By Rahul Barve

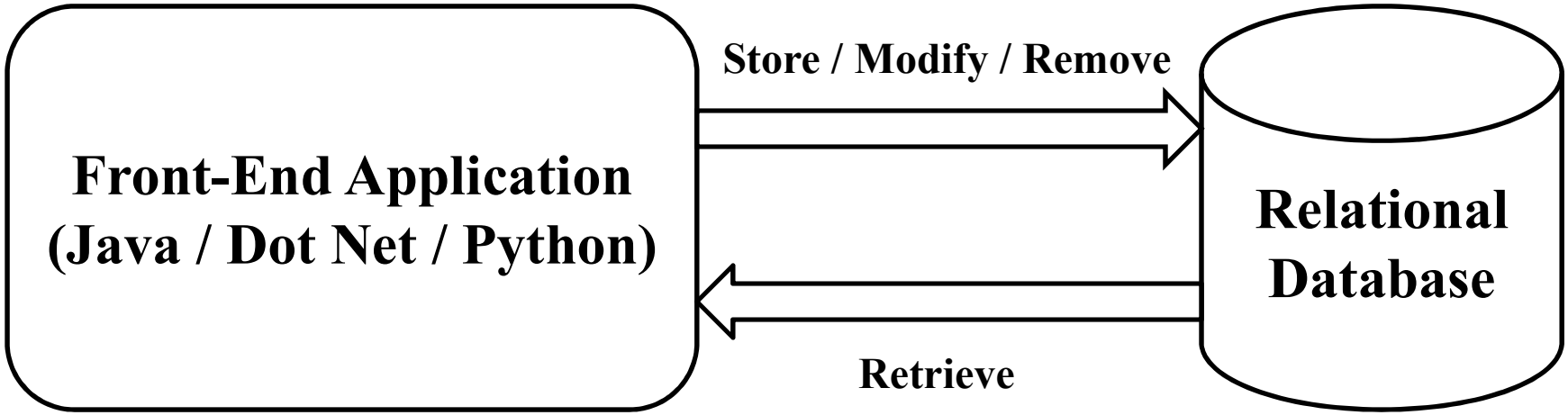
Introduction to RDBMS

- A Relational Database Management System is a special system software that is used to manage the organization, storage, access, security and integrity of a data.

RDBMS

- Allows application systems to emphasize upon the user interface, data validation and screen navigation.
- Whenever there is a need to add, modify, delete or display data, the application system simply makes a "call" to the RDBMS.

RDBMS


Front-End Application
(Java / Dot Net / Python)
Store / Modify / Remove
Retrieve
Relational
Database

Why RDBMS

By Rahul Barve

Why RDBMS

- Since a data is simply stored in a tabular format, retrieval of the data becomes easy.
- Relational model helps in reducing the redundancy.
- It makes possible to apply validation rules on the data with the help of constraints.
- It makes possible to acquire enterprise level services for data management.

Relational Database Services

By Rahul Barve

Relational Database Services

- Simple Design
- Relationships
- Constraints
- Security
- Efficient Searching and Sorting
- Transaction Isolation
- Concurrency
- Locking

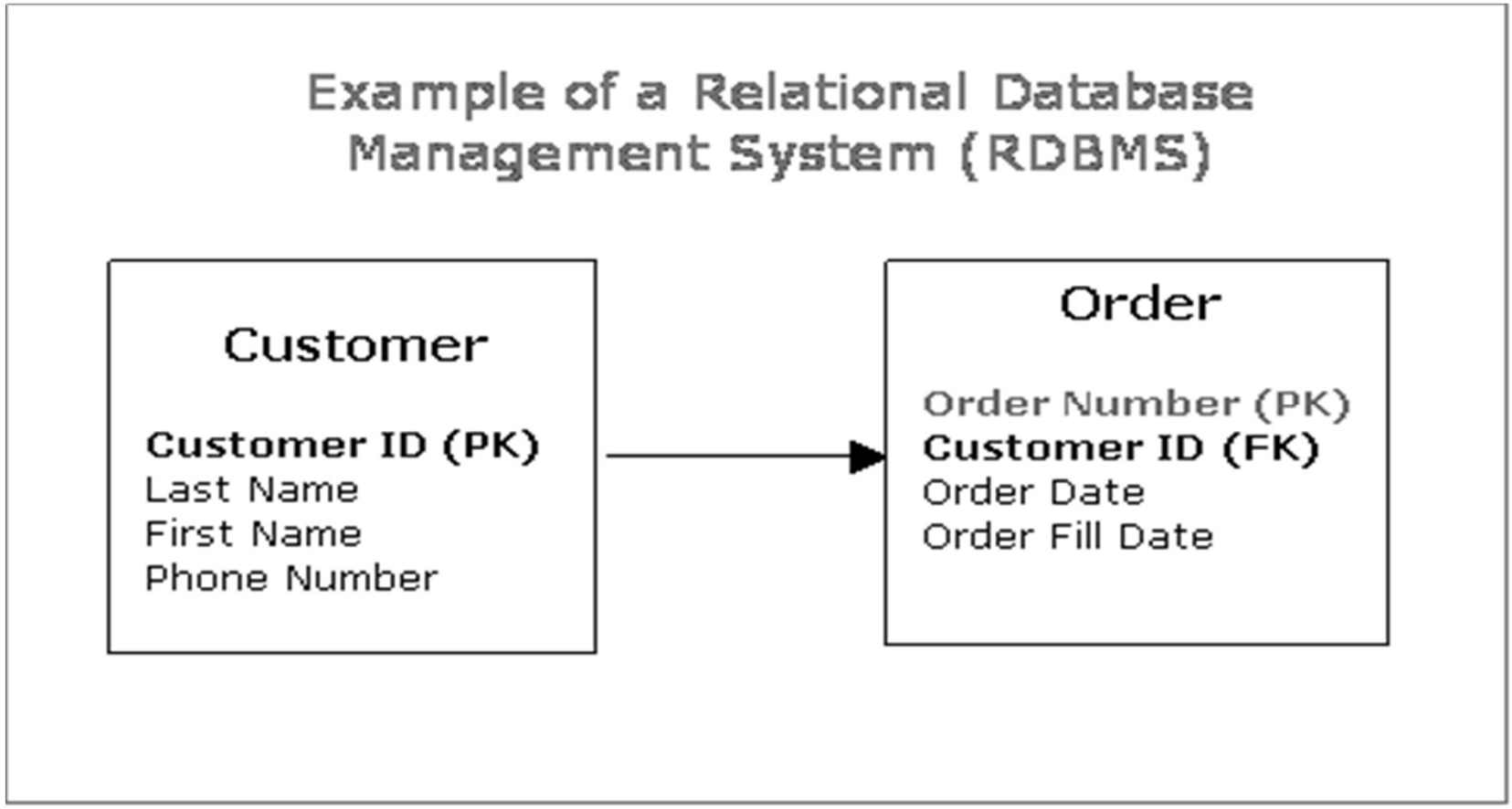
Relational Database Servers

- Oracle by Oracle Corporation
- SQL Server by Microsoft
- DB2 by IBM
- MySQL by Oracle Coporation

Database Design

- A relational database stores an information in a set of `tables`, each of which contains a unique identifier known as a `primary key`.
- These tables are further related to one another by using `foreign keys`.

Relational Database Design



Data Normalization

By Rahul Barve

Data Normalization

- Data normalization is a technique of organizing the data using a systematic approach of decomposing tables to reduce data redundancy.
- Normalization usually involves dividing large tables into smaller ones and defining relationships among them.

Data Normalization

- Data normalization is divided into 3 categories:
  - 1st Normal Form (1NF)
  - 2nd Normal Form (2NF)
  - 3rd Normal Form (3NF)

Data Normalization

- Table without normalization:

Roll No	Name	Branch	HOD	Phone No
1	Bruce	CS	Thomas	856-433-8317
2	Harry	CS	Thomas	856-433-8317
3	Maria	CS	Thomas	856-433-8317
4	Nuria	CS	Thomas	856-433-8317
5	Andrew	CS	Thomas	856-433-8317

Data Normalization – 1NF

By Rahul Barve

Data Normalization – 1NF

- It is the minimum requirement of a Database design otherwise it is considered to be a poor database design.
- Basic Rule: A column must contain a single value.

Data Normalization – 1NF

- Poor Database Design

Roll No	Name	Subject
1	Bruce	Java
2	Harry	Java, Angular
3	Maria	Angular, React
4	Nuria	SQL
5	Andrew	Python

Data Normalization – 1NF

- Database Design with 1NF

Roll No	Name	Subject
1	Bruce	Java
2	Harry	Java
2	Harry	Angular
3	Maria	Angular
3	Maria	React
4	Nuria	SQL
5	Andrew	Python

Data Normalization – 2NF

By Rahul Barve

Data Normalization – 2NF

- There are 2 conditions need to be satisfied so that the tables can be said to be in the 2nd normal form:
  - The tables must be in the 1st normal form.
  - There should not be any partial dependency of any column on a primary key.

Data Normalization – 2NF

- Student_Master

Student_ID	Name	Country
1	Bruce	USA
2	Harry	England
3	Nuria	Spain

Data Normalization – 2NF

- Course_Master

Course_ID	Name
1	Core Java
2	Java EE
3	Angular

Data Normalization – 2NF

- Score_Details

Primary Key

Score_ID	Student_ID	Course_ID	Score	Cost_$
1	1	1	87	400
2	2	3	75	550
3	1	2	77	475
4	3	3	82	550
5	2	2	80	475

Data Normalization – 2NF

- In the `Score_Details` table, column `Cost` indicates a partial dependency.
- Ideally the `Cost` column has to be a part of `Course_Master` table.

Data Normalization – 3NF

By Rahul Barve

Data Normalization – 3NF

- There are 2 conditions need to be satisfied so that the tables can be said to be in the 3rd normal form:
  - The tables must be in the 2nd normal form.
  - There should not be transitive dependency.

Data Normalization – 3NF

- Score_Details

Score_ ID	Student_ ID	Course_ ID	Score	Exam	Total Marks

Data Normalization – 3NF

- In the `Score_Details` table, column `Total_Marks` depends upon the type of the exam such as Theory or Practical i.e. `Exam` column which is a non-prime attribute.

Data Normalization – 3NF

- Ideally columns `Exam` and `Total_Marks` must be taken away from `Score_Details` and maintained in a separate table e.g. `Exam_Details`.