DATABASES
SQL

INFOTEK SOLUTIONS TEAM

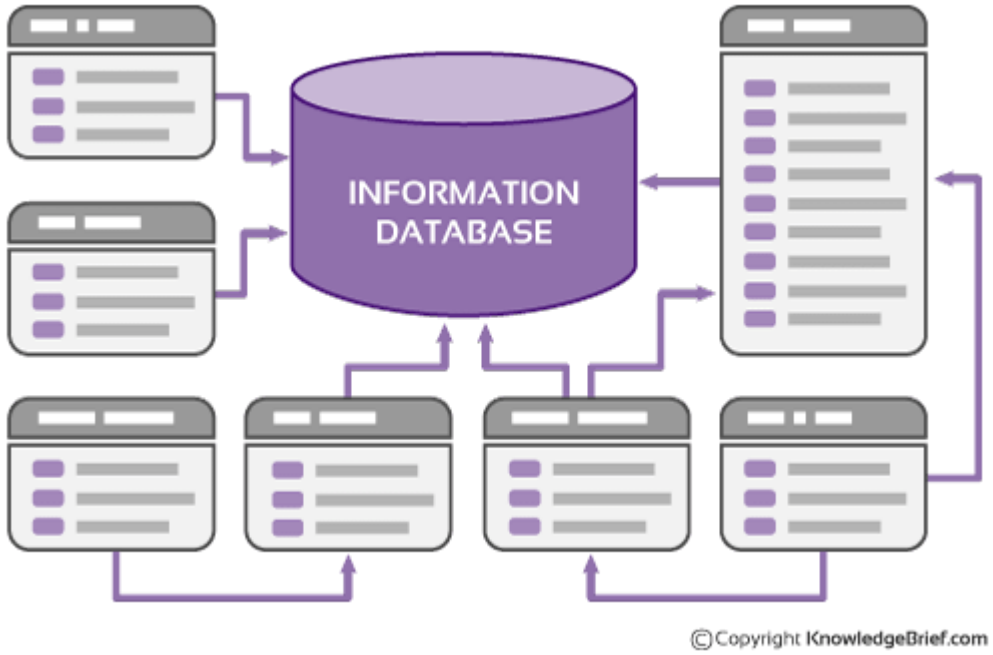
TRAINING@INFOTEK-SOLUTIONS.COM

Databases

1. Introduction in databases
2. Relational databases (SQL databases)
3. Database management system (DBMS)
4. Database design
5. Constrains
6. ACID (Atomicity, Consistency, Isolation, and Durability)
7. SQL (DCL, DDL, DML)
8. Deep dive with DML

Introduction into databases

A *database* is an organized collection of data that is organized so that it can be easily accessed, managed and updated.



Database management system (DBMS)

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.

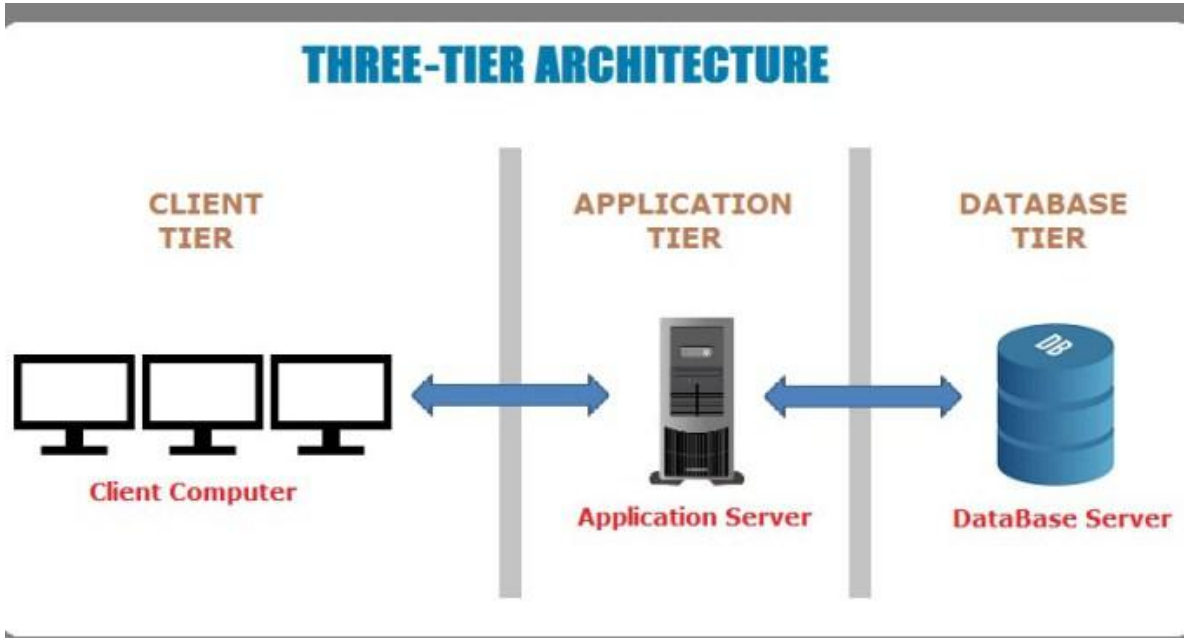
SQL databases

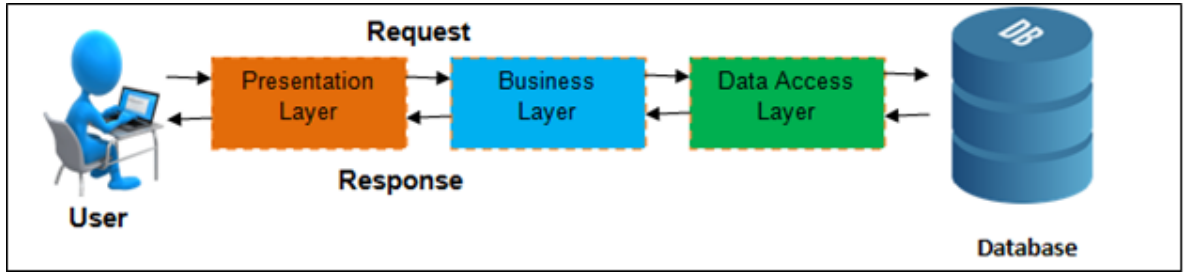
Microsoft SQL Server (MS SQL), Oracle Database, IBM DB2, MySQL, PostgreSQL

Non-SQL databases

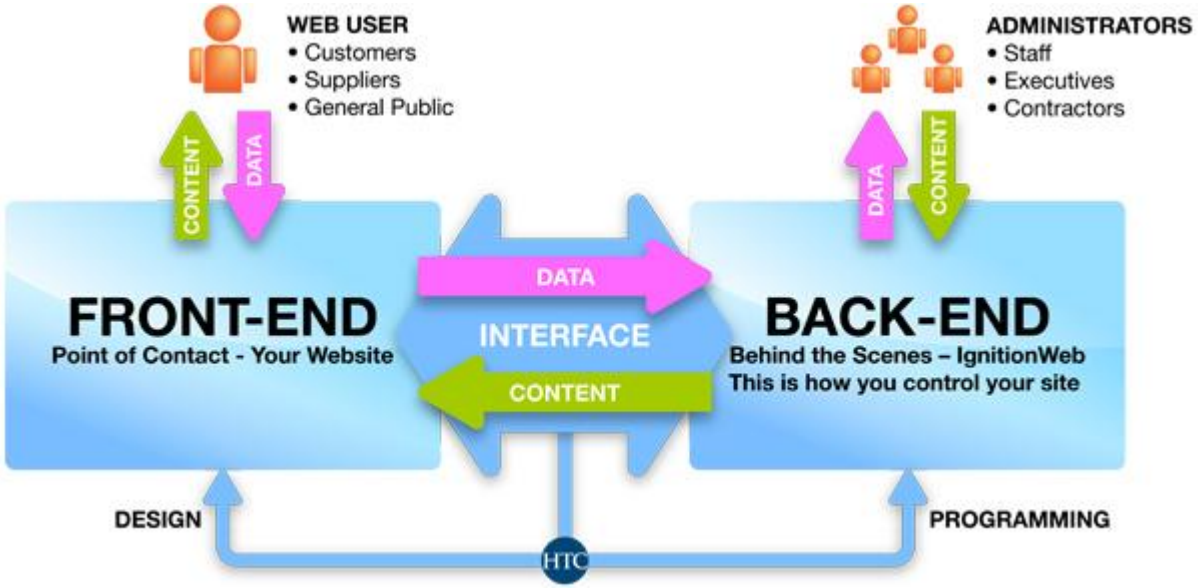
MongoDB, DynamoDB, Cassandra, Couchbase

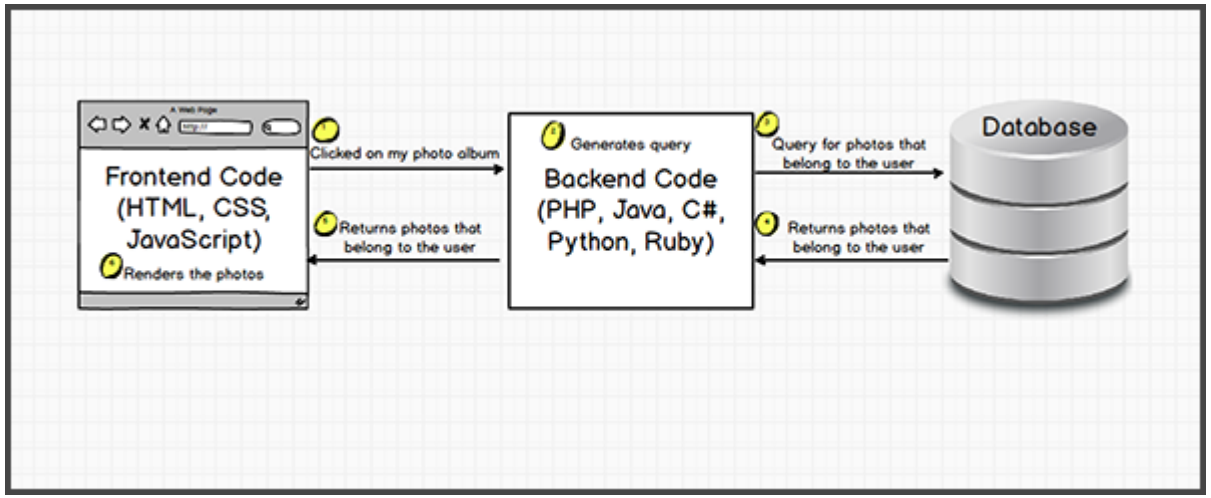
Three Tier Architecture





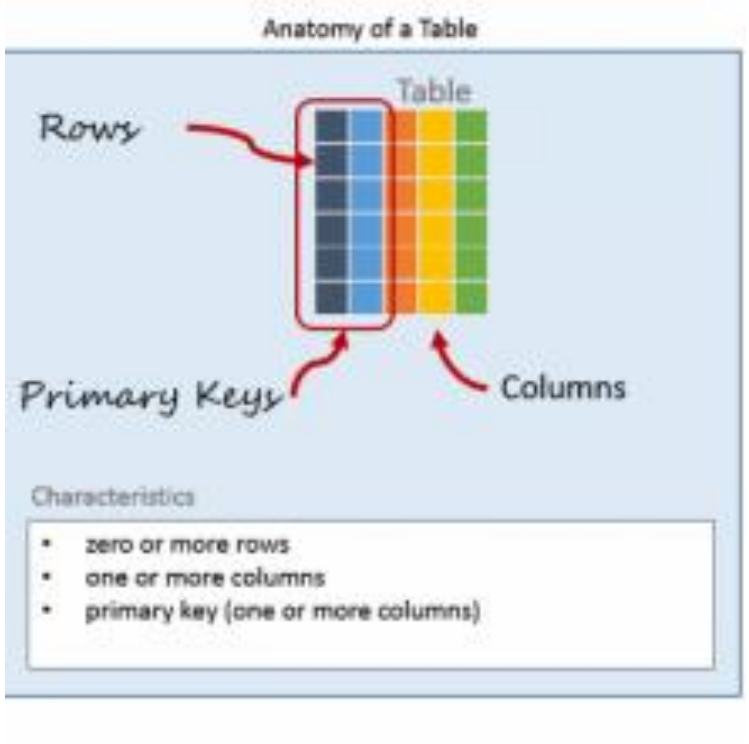
Front-End vs Back-End

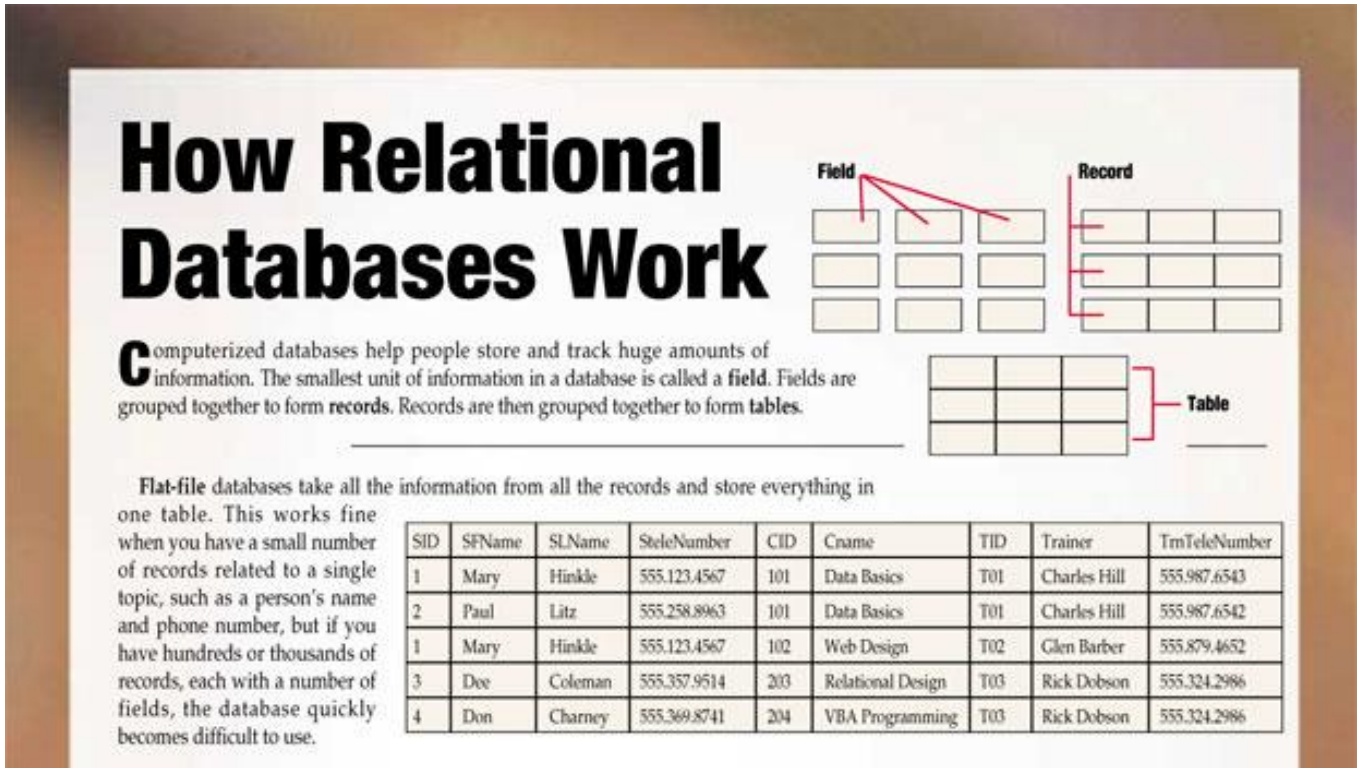




Relational databases

A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements.



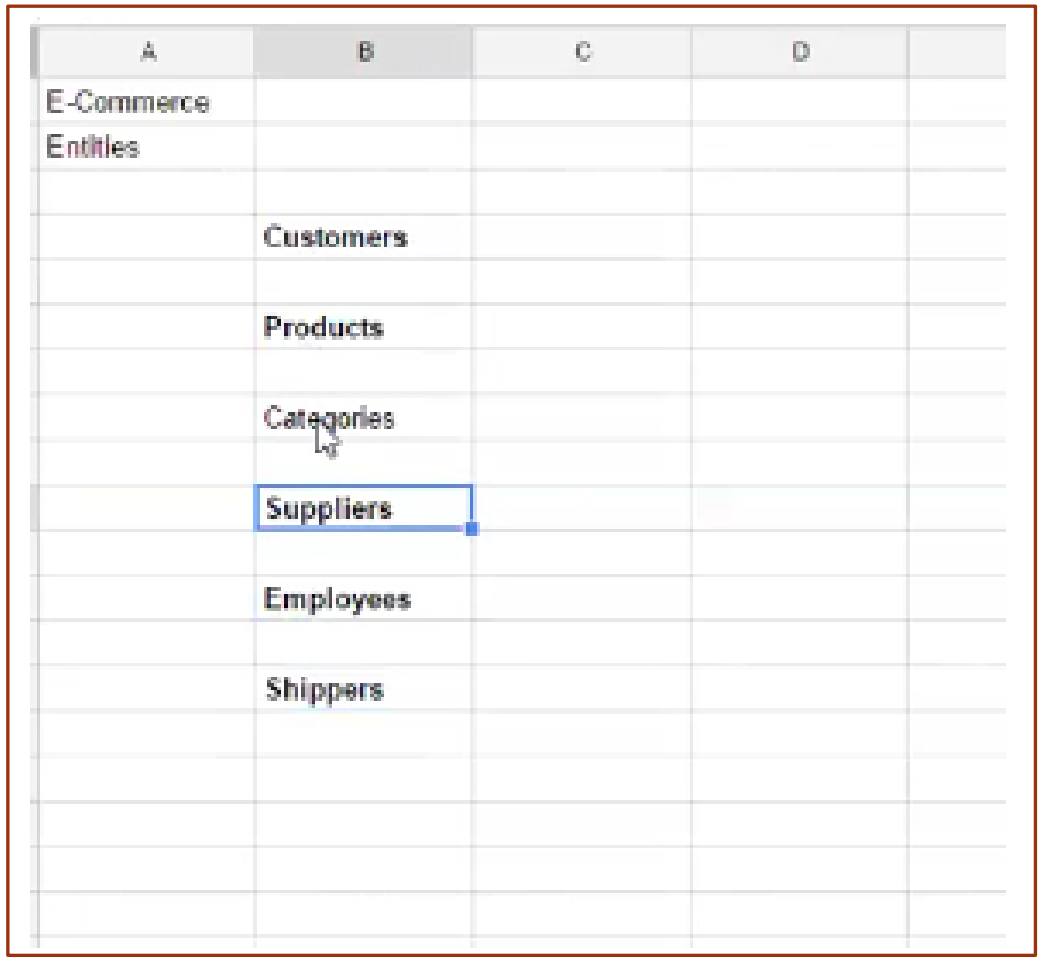


Database Design Life Cycle

1. Do requirement Gathering
    a. From paper
    b. From their website
    c. From Backlog
    d. From their current system
2. Define Entities: Table Names going to be included
3. Define Attributes of each Entity: Column Names to be included for each table
4. Define Constraints of each Entity
    a. Data type of each Entity
    b. Primary Key
    c. Null value
    d. Unique
5. Create ER (Entity Relationship) Diagram Between Available Entities
6. Manage Newborn Tables from ER diagram
7. Do normalization to improve efficiency, security and performance of your Database
    a. First level
    b. Second level
    c. Third level
    d. Fourth level
    e. Fifth level
8. Create final schema of your database
    a. Ship foreign keys between tables
    b. And decide the final schema
9. Design the Actual database on any **RDBMS**
    a. Can be **MySQL**
    b. Oracle
    c. MS SQL Server
    d. DB2
    e. PostgreSQL
10. Populate data and manipulate with programming languages

Complete Guide to Design E-Commerce DB

Step 1: Define Entities



Complete Guide to Design E-Commerce DB

Step 2: Define Attributes of Each Entity

E-Commerce Database									
		Define Attributes							
Entities	**Customers**	CustomerID	CustomerName	ContactName	City	PostalCode	Country		
	Products	ProductID	Productname	Unit	Price				
	Categories	CategoryID	CategoryName	Description					
	Suppliers	SupplierID	SupplierName	ConatctName	Address	City	PostalCode	Country	Phone
	Employees	EmployeeID	LastName	FirstName	BirthDate	Photo	Notes		
	Shippers	ShipperID	ShipperName	Phone					

Complete Guide to Design E-Commerce DB

Step 3: Assign Data Type & Constraints for Attributes

Integer
Varchar(n)
Double
Date
Byte
Boolean

E-Commerce Database									
		Define Attributes							
Entities	Customers	CustomerID	CustomerName	Contactname	Address	City	PostalCode	Country	
		int	varchar(50)	Varchar(50)	Varchar(100)	Varchar(25)	int	varchar(25)	
	Products	ProductID	ProductName	Unit	Price				
		int	varchar(50)	Varchar(50)	double				
	Categories	CategoryID	CategoryName	Descripton					
		int	varchar(50)	varchar(100)					
	Suppliers	SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
		int	varchar(50)	Varchar(50)	varchar(100)	varchar(25)	int	varchar(25)	varchar(15)
	Employees	EmployeeID	LastName	FiirstName	BirthDate	Photo	Notes		
		int	varchar(50)	varchar(50)	date	bitmap	varchar(100)		
	Shippers	ShipperID	ShipperName	Phone					
		int	varchar(50)	varchar(15)					

Constraints

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

The following constraints are commonly used in SQL:

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

FOREIGN KEY - Uniquely identifies a row/record in another table

CHECK - Ensures that all values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column when no value is specified

INDEX - Used to create and retrieve data from the database very quickly

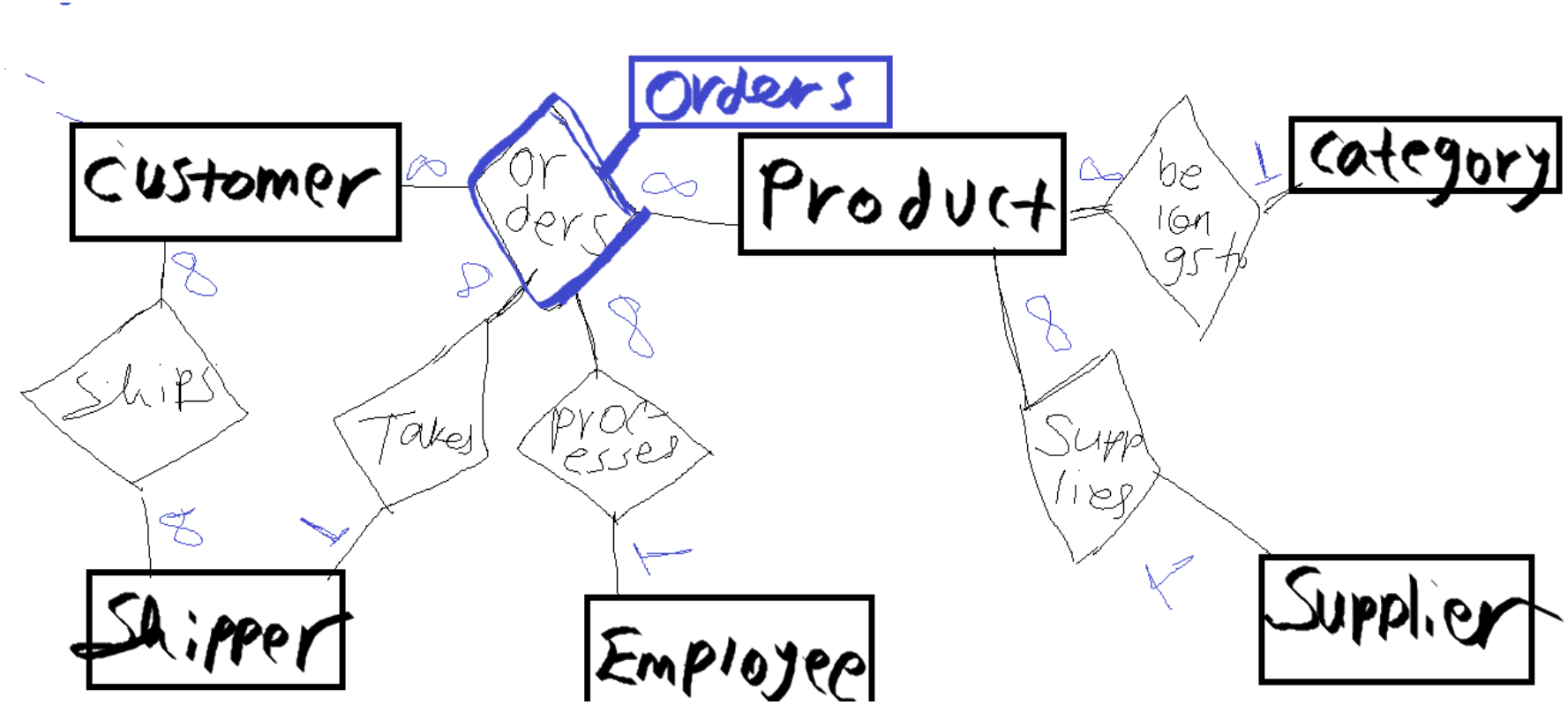
Complete Guide to Design E-Commerce DB

Setting Primary key and Not Null Constraint

E-Commerce Database			
		Define Attributes	
Entities	**Customers**	CustomerID	CustomerName
		int, Primary key	varchar(50), Not
	Products	ProductID	ProductName
		int, Primary key	varchar(50), not
	Categories	CategoryID	CategoryName
		int, Primary key	varchar(50)
	Suppliers	SupplierID	SupplierName
		int, Primary key	varchar(50)
	Employees	EmployeeID	LastName
		int	varchar(50)
	Shippers	ShipperID	ShipperName
		int, Primary key	varchar(50)

Complete Guide to Design E-Commerce DB

Step 4: ER-Diagram: Develop Entity Relationship Diagram



Complete Guide to Design E-Commerce DB

Normalization is a process of breaking down tables into small in size to increase efficiency, remove redundancy, remove null value, save storage, increase security.

Example for our case:

Because customer and product has many to many relationship, I have added a new table named Orders(orderID primarykey int(11), CustomerID int(11), ProductID int(11), OrderDate date)

For management purpose, I have added OrderDetail(OrderID, ProductID, amount)

Normalization

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data.

It divides larger tables to smaller tables and links them using relationships.

1NF (First Normal Form) Rules

Each table cell should contain a single value.

Each record needs to be unique.

2NF (Second Normal Form) Rules

Rule 1- Be in 1NF

Rule 2- Single Column Primary Key

3NF (Third Normal Form) Rules

Rule 1- Be in 2NF

Rule 2- Has no transitive functional dependencies

Complete Guide to Design E-Commerce DB

Step 6: Define Final Schema on Paper :

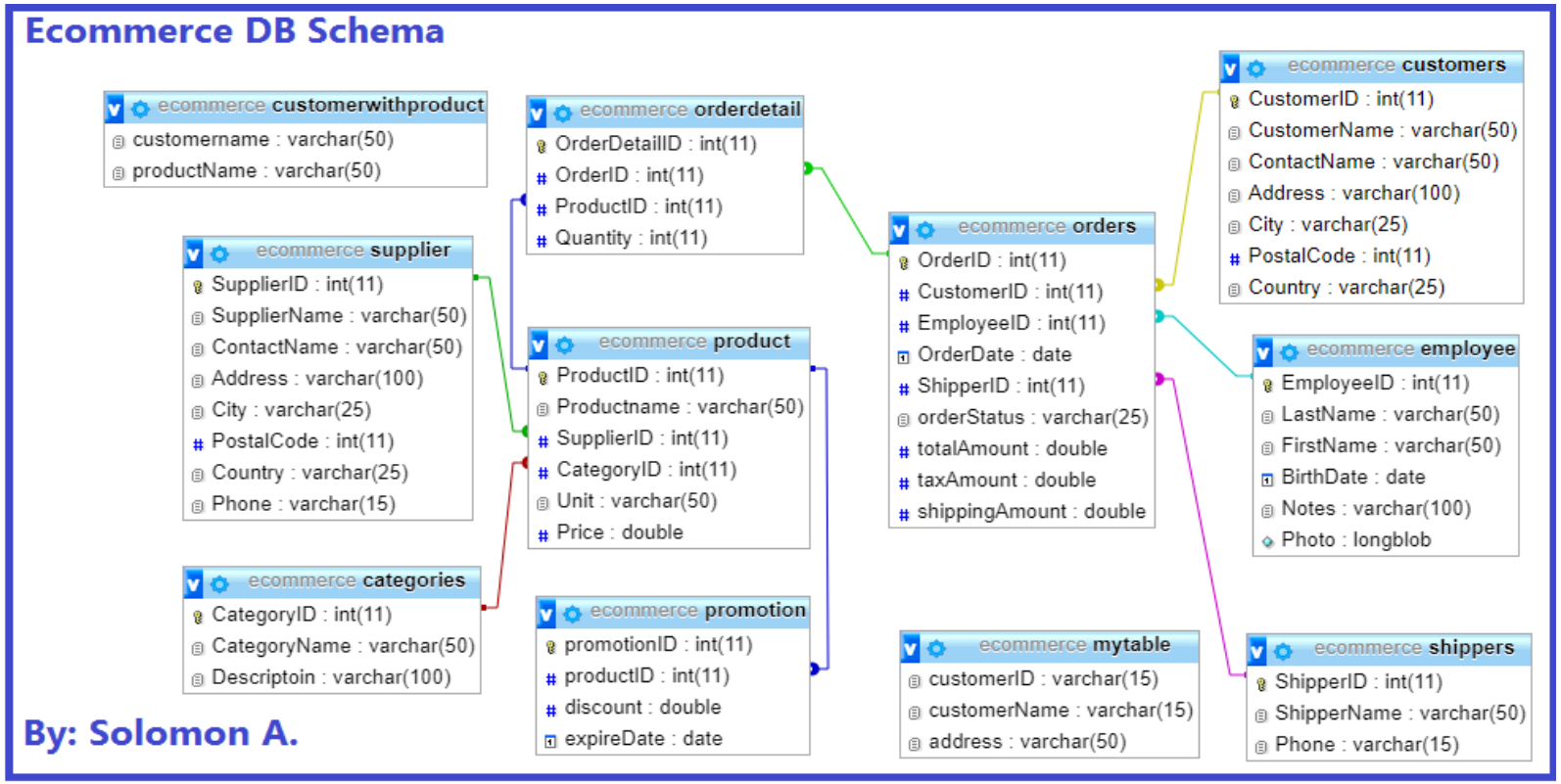
Show foreign keys

	Final Schema On Paper of E-Commerce Databse							
Customers	CustomerID	CustomerName	Contactname	Address	City	PostalCode	Country	
	int, Primary Key	varchar(50), Not null	Varchar(50), not null	Varchar(100)	Varchar(25)	int	varchar(25)	
Products	ProductID	ProductName	Unit	Price	CategoryID	SupplierID		
	int, Primary Key	varchar(50)	Varchar(50)	double				
Categories	CategoryID	CategoryName	Descripton					
	int, Primary Key	varchar(50)	varchar(100)					
Supplier	SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
	int, Primary Key	varchar(50)	Varchar(50)	varchar(100)	varchar(25)	int	varchar(25)	varchar(15)
Employees	EmployeeID	LastName	FiirstName	BirthDate	Photo	Notes		
	int, Primary Key	varchar(50)	varchar(50)	date, not null	bitmap	varchar(100)		
Shippers	ShipperID	ShipperName	Phone					
	int, Primary Key	varchar(50)	varchar(15)					
Orders	OrderID	OrderDate	CustomerID	EmployeeID	ShipperID	Status		
OrderDetailsw	OrderDetailID	ProductID	Quantity	OrderID		varchar(10)		
Email	CustomerID	Email						
	int, Primary Key	varchar(50), Not null						

Complete Guide to Design E-Commerce DB

Step 7: Design Final Schema on any RDBMS

Following is from MySQL



ACID

The acronym ACID describes some ideal properties of a database transaction: Atomicity, Consistency, Isolation, and Durability.

Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged. An atomic system must guarantee atomicity in each and every situation, including power failures, errors and crashes. To the outside world, a committed transaction appears (by its effects on the database) to be indivisible ("atomic"), and an aborted transaction does not happen.

The **consistency** property ensures that any transaction will bring the database from one valid state to another. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof. This does not guarantee correctness of the transaction in all ways the application programmer might have wanted (that is the responsibility of application-level code), but merely that any programming errors cannot result in the violation of any defined rules.

The **isolation** property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed sequentially, i.e., one after the other. Providing isolation is the main goal of concurrency control. Depending on the concurrency control method (i.e., if it uses strict - as opposed to relaxed - serializability), the effects of an incomplete transaction might not even be visible to another transaction.

The **durability** property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors. In a relational database, for instance, once a group of SQL statements execute, the results need to be stored permanently (even if the database crashes immediately thereafter). To defend against power loss, transactions (or their effects) must be recorded in a non-volatile memory.

SQL

SQL stands for Structured Query Language. **SQL** lets you access and manipulate data held in a relational database management system (RDBMS).

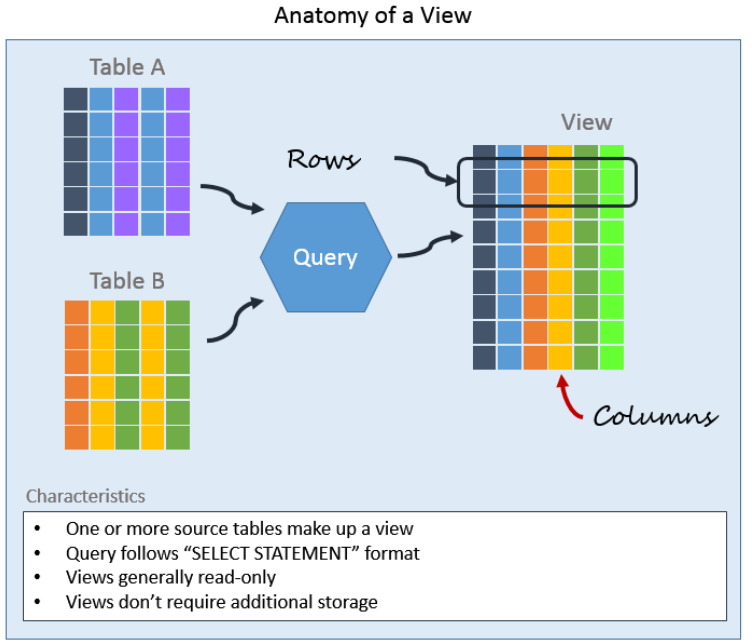
Data control language (DCL) – controls access to data;

Data definition language (DDL) – defines data types such as creating, altering, or dropping and the relationships among them;

Data manipulation language (DML) – performs tasks such as inserting, updating, or deleting data occurrences;

Then study by following a tutorial on training portal:

http://www.training.qaonlinetraining.com/database/sql-queries/sql/dml/



Install MySQL database

1) Download XAMPP
2) Install
3) Run Apache and MySQL
4) Open localhost/phpmyadmin
5) Create your database

Install XAMPP on Mac OS
1) google: "xampp download"
2) download optional for Mac OS with PHP 5.4
3) once it downloaded launch installation (it is behind browser window)
4) follow installation process, switch back to xampp application
5) choose tab "Manage Servers" in xampp application
6) check that Apache is running
7) start MySQL
8) open tab "General" and click "Go to application" (browser localhost/phpmyadmin)
9) work with your databases in browser

Thank you

Questions

Suggestions