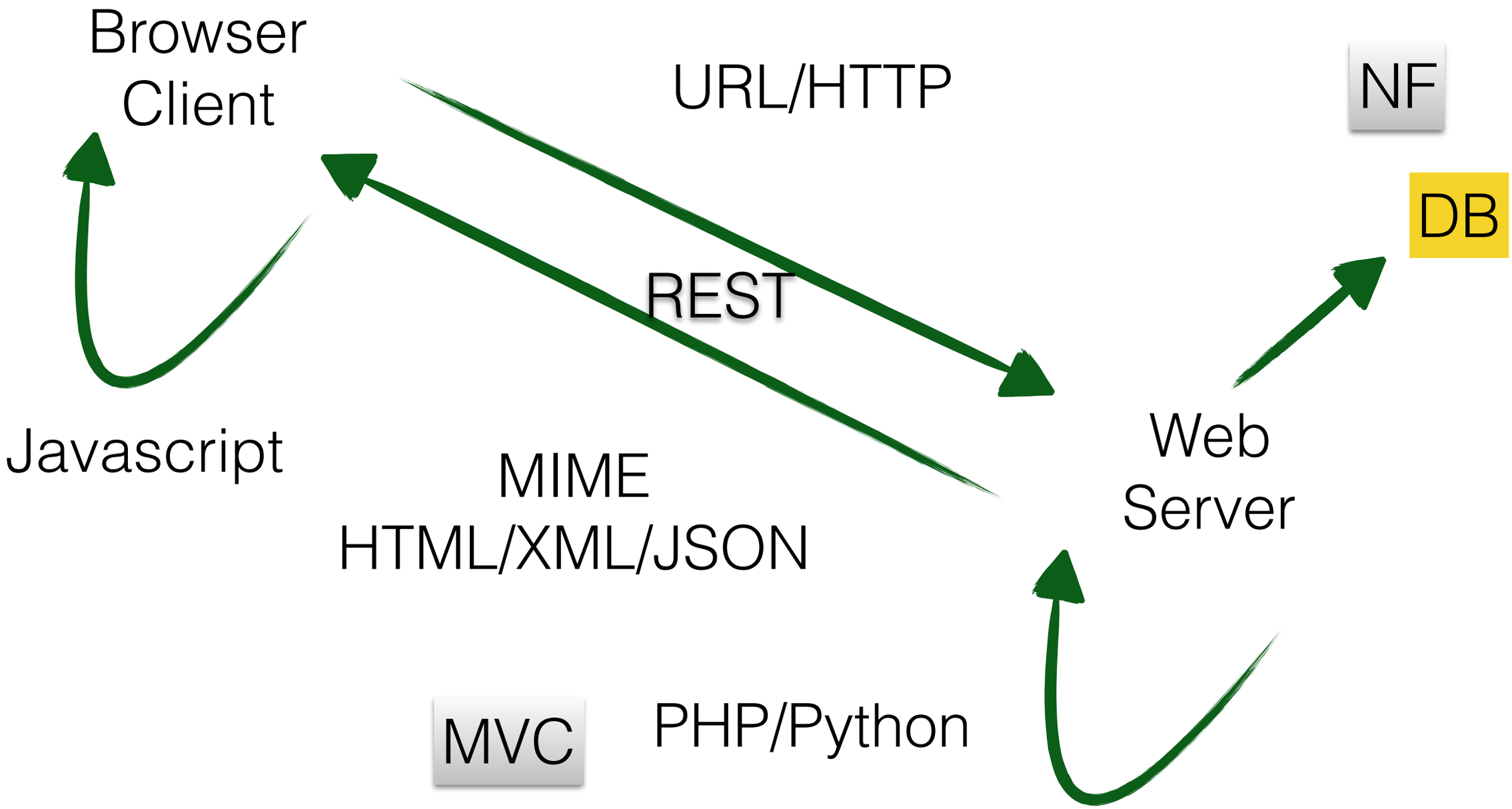
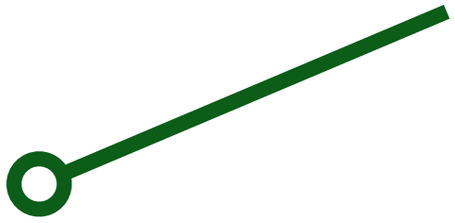
Web dynamic
SQL

Frédéric Le Mouël & Razmig Kechichian

The Big Picture



SQL
Structured Query Language



American National Standards Institute

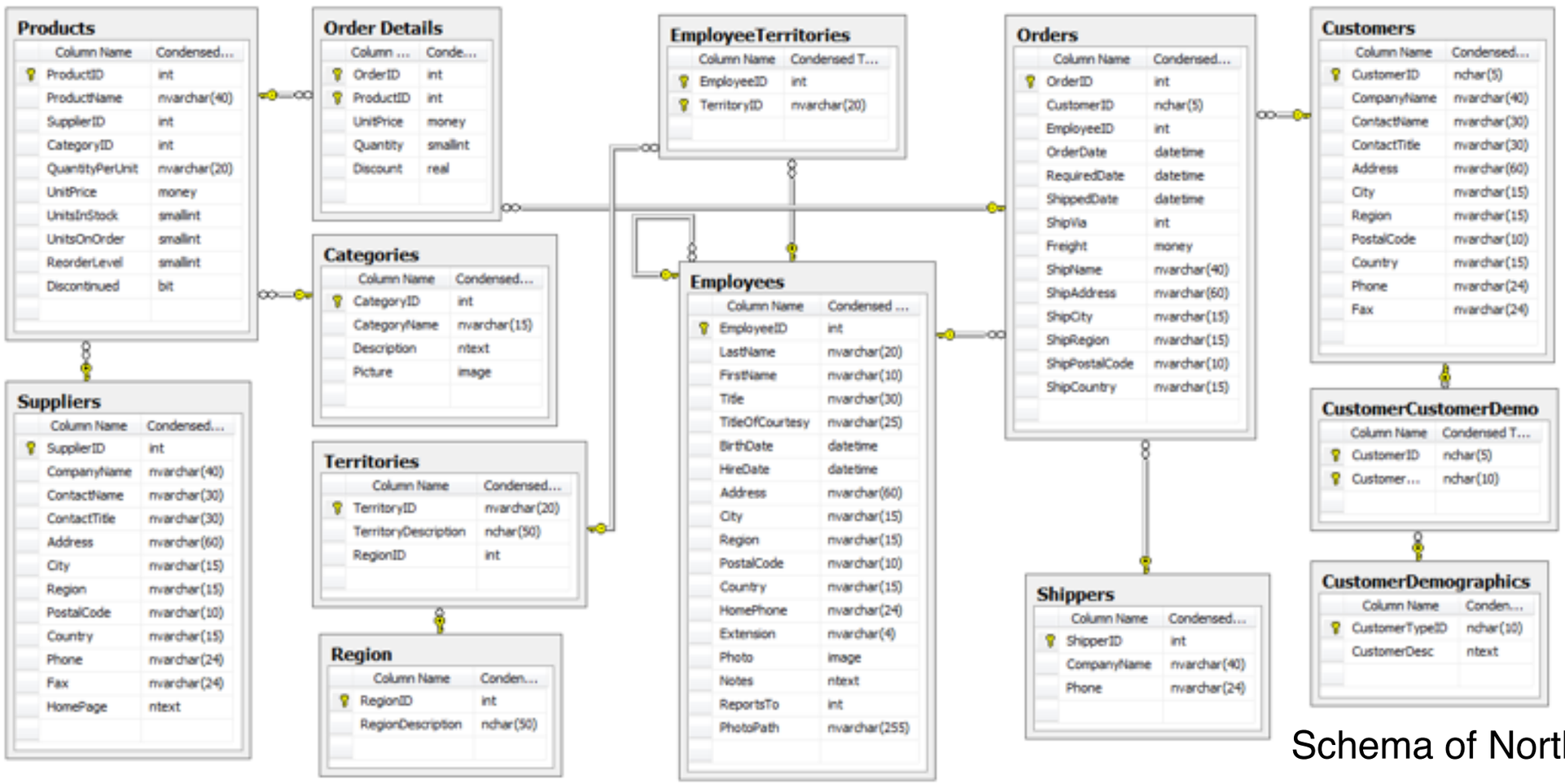
- ANSI recommandation + ISO standard …
- … but a bunch of versions & proprietary extensions!

SQL

- SQL: **declarative** language
  - Specify *what* to do, but *not how* to do it!
- SQL: accesses and manipulates **relational** databases

Relational Databases?

- A Relational Database (RDB): a set of interrelated tables
- Schema of a database: structure of the database



Schema of Northwind Sample database

http://northwinddatabase.codeplex.com

Table?

- Table: a set of entities defined by their attributes
  - Entities = records = table rows
  - Attributes = table column
- Schema of a table: structure of the table (#attributes, keys, types, etc.)
  - Types = INTEGER, REAL, VARCHAR(n), DATE, TIME, FILE, etc.

nchar(5)

key

Table?

nvarchar(10)

Customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)

No order !

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	5021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	5023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Schema of the Customers Table from the Northwind Sample database

http://northwinddatabase.codeplex.com

Key?

- Table keys: set of attributes that uniquely identify records of a table

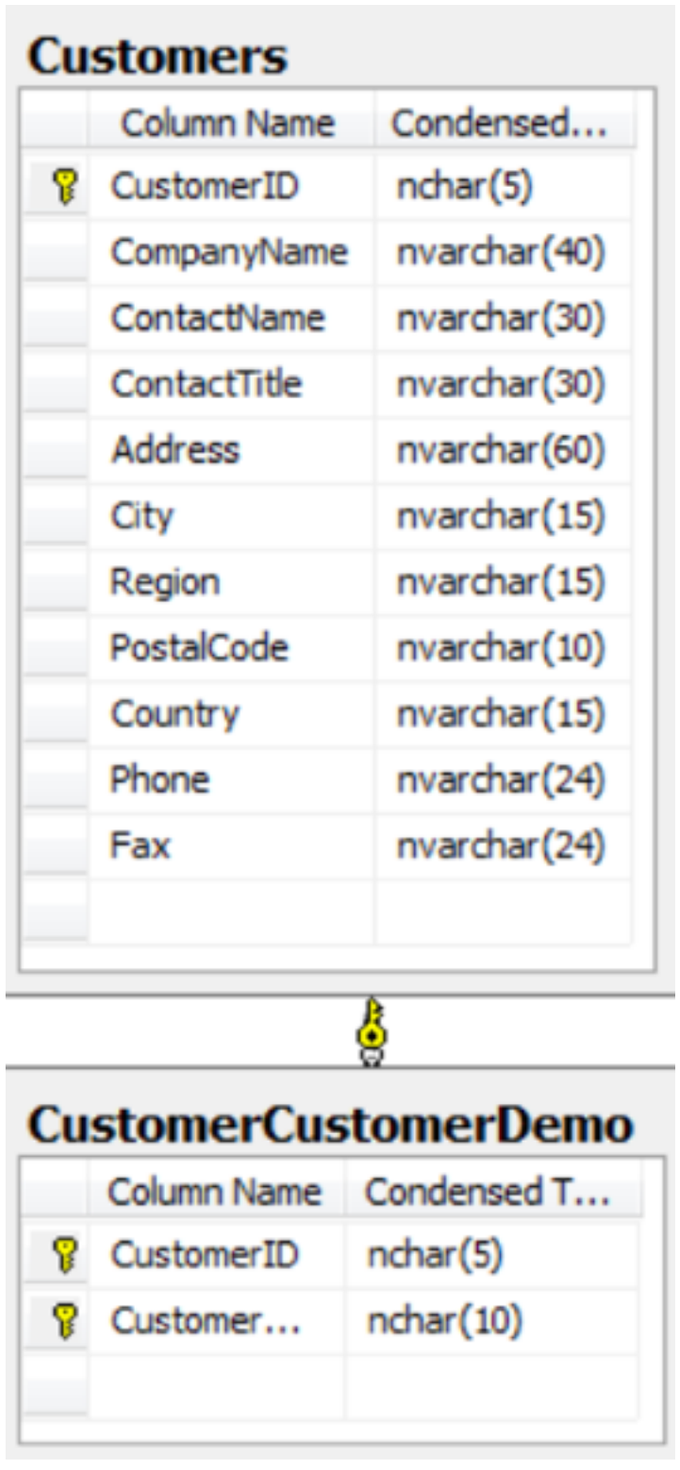
  - Keys can not be NULL value

```
CREATE TABLE  Customers (
  CustomerID nchar(5) NOT NULL,
  …
  PRIMARY KEY (CustomersID)
)
```

Foreign Keys?

- Foreign Key: a table attribute that points a primary key of another table

```
CREATE TABLE  CustomerCustomerDemo (
   CustomerID nchar(5) NOT NULL,
   CustomerPostalCode nchar(10) NOT NULL,
   …
   FOREIGN KEY (CustomersID) REFERENCES
   Customers(CustomerID),
   FOREIGN KEY (CustomerPostalCode)
   REFERENCES Customers(PostalCode)
)
```



What can SQL do?

Data Definition Language (DDL)

Create table **CREATE**
Modify table **ALTER**
Delete table **DROP**
…

Data Modification Language (DML)

Insert records **INSERT**
Update records **UPDATE**
Delete records **DELETE**
…

Queries

Search records **SELECT**

Everything is in SELECT

```
SELECT {Attribute List}
FROM {Table List}
Where {Logical Conditions}
```

Inputs: tables

Output: logical table

SELECT

```
SELECT CustomerID, CustomerName
FROM Customers
WHERE City = ‘México D.F.’
```

2	Ana Trujillo Emparedados y helados
3	Antonio Moreno Taquería

Let’s join tables!

CCDemo

Customers

CustomerID	CustomerName	Country
1	Alfreds Futterkiste	Germany
2	Ana Trujillo Emparedados y helados	Mexico
3	Antonio Moreno Taquería	Mexico

CustomerID	CustomerPostalCode
2	9002
4	18500

Let’s join tables!

```
SELECT *
FROM Customers, CCDemo
```

All possible combinations!

Customers.CustomerID	Customers.CustomerName	Customers.Country	CCDemo.CustomerID	CCDemo.CustomerPostal Code
1	Alfreds Futterkiste	Germany	2	9002
2	Ana Trujillo Emparedados y helados	Mexico	4	18500
3	Antonio Moreno Taquería	Mexico	2	9002
1	Alfreds Futterkiste	Germany	4	18500
2	Ana Trujillo Emparedados y helados	Mexico	2	9002
3	Antonio Moreno Taquería	Mexico	4	18500

Natural join :-)

Common rows!

```
SELECT *
FROM Customers, CCDemo
WHERE Customers.CustomerID
= CCDemo.CustomerID
```

Customers.CustomerID	Customers.CustomerName	Customers.Country	CCDemo.CustomerPostalCode
2	Ana Trujillo Emparedados y helados	Mexico	9002

Left join?

```
SELECT *
FROM Customers LEFT JOIN CCDemo
WHERE Customers.CustomerID =
CCDemo.CustomerID
```

Keep the first table and complete !

Customers.CustomerID	Customers.CustomerName	Customers.Country	CCDemo.CustomerID	CCDemo.CustomerPostalCode
1	Alfreds Futterkiste	Germany	NULL	NULL
2	Ana Trujillo Emparedados y helados	Mexico	2	9002
3	Antonio Moreno Taquería	Mexico	NULL	NULL

Right join?

```
SELECT *
FROM Customers RIGHT JOIN CCDemo
WHERE Customers.CustomerID =
CCDemo.CustomerID
```

Keep the second table and complete !

CCDemo.CustomerID	CCDemo.CustomerPostalCode	Customers.CustomerID	Customers.CustomerName	Customers.Country
2	9002	2	Ana Trujillo Emparedados y helados	Mexico
4	18500	NULL	NULL	NULL

Webography

- W3Schools SQL Tutorial
  - http://www.w3schools.com/sql/
- SQL as understood by SQLite
  - https://www.sqlite.org/lang.html
- Onto more fun and games!
  - https://chinookdatabase.codeplex.com/
  - Downloads, grab ChinookDatabase1.4_Sqlite.zip