SESRIC

SESRIC IT Department

Statistical Database Design and Visualization Tools Development

DATABASE DESIGN

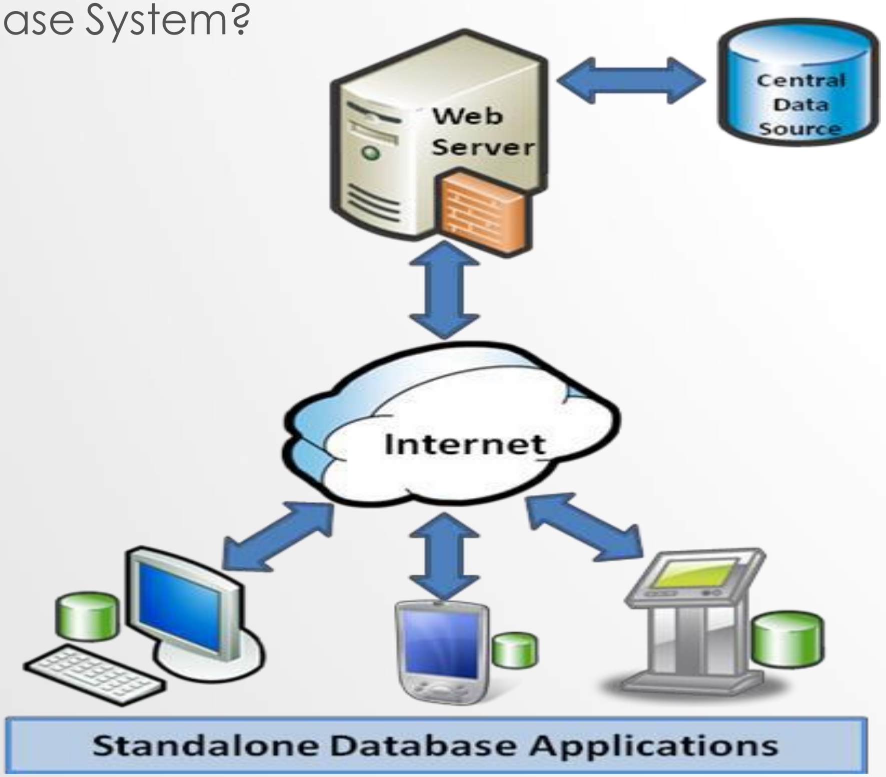
Relational Database Systems and the Life Cycle

DATABASE DESIGN

- **Relational Database Systems and the Life Cycle**
  - **General definition of Database Systems**
  - **Information system life cycle**
  - **How to design Database System?**

RELATIONAL DATABASE SYSTEMS AND THE LIFE CYCLE

- **General definition of Database Systems**
- Information system life cycle
- How to design Database System?

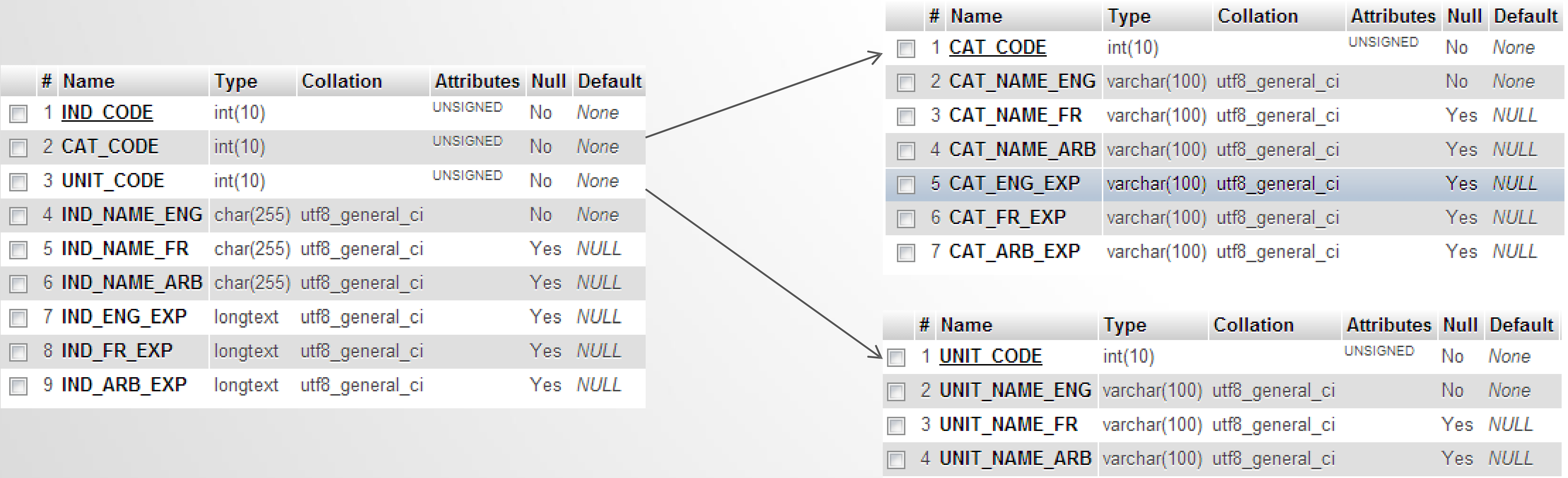


- **A database system is an overall collection of different database software components and databases containing the following parts:**
  - **Database application programs**
  - **Client components**
  - **Database server(s)**
  - **Databases**

RELATIONAL DATABASE SYSTEMS AND THE LIFE CYCLE

- **General definition of Database Systems**
  - Information system life cycle
  - How to design Database System?

- **Relational Database**



	#	Name	Type	Collation	Attributes	Null	Default
□	1	IND_CODE	int(10)		UNSIGNED	No	*None*
□	2	CAT_CODE	int(10)		UNSIGNED	No	*None*
□	3	UNIT_CODE	int(10)		UNSIGNED	No	*None*
□	4	IND_NAME_ENG	char(255)	utf8_general_ci		No	*None*
□	5	IND_NAME_FR	char(255)	utf8_general_ci		Yes	*NULL*
□	6	IND_NAME_ARB	char(255)	utf8_general_ci		Yes	*NULL*
□	7	IND_ENG_EXP	longtext	utf8_general_ci		Yes	*NULL*
□	8	IND_FR_EXP	longtext	utf8_general_ci		Yes	*NULL*
□	9	IND_ARB_EXP	longtext	utf8_general_ci		Yes	*NULL*

	#	Name	Type	Collation	Attributes	Null	Default
□	1	CAT_CODE	int(10)		UNSIGNED	No	*None*
□	2	CAT_NAME_ENG	varchar(100)	utf8_general_ci		No	*None*
□	3	CAT_NAME_FR	varchar(100)	utf8_general_ci		Yes	*NULL*
□	4	CAT_NAME_ARB	varchar(100)	utf8_general_ci		Yes	*NULL*
□	5	CAT_ENG_EXP	varchar(100)	utf8_general_ci		Yes	*NULL*
□	6	CAT_FR_EXP	varchar(100)	utf8_general_ci		Yes	*NULL*
□	7	CAT_ARB_EXP	varchar(100)	utf8_general_ci		Yes	*NULL*

	#	Name	Type	Collation	Attributes	Null	Default
□	1	UNIT_CODE	int(10)		UNSIGNED	No	*None*
□	2	UNIT_NAME_ENG	varchar(100)	utf8_general_ci		No	*None*
□	3	UNIT_NAME_FR	varchar(100)	utf8_general_ci		Yes	*NULL*
□	4	UNIT_NAME_ARB	varchar(100)	utf8_general_ci		Yes	*NULL*

RELATIONAL DATABASE SYSTEMS AND THE LIFE CYCLE

- General definition of Database Systems
- **Information system life cycle**
- How to design Database System?

❑Feasibility analysis

❑Requirement collection and analysis

❑Design

❑Validation and acceptance testing

RELATIONAL DATABASE SYSTEMS AND THE LIFE CYCLE

- General definition of Database Systems
- **Information system life cycle**
- How to design Database System?

➢ **Information system**

Information System includes all resources involved in the collection, management, use and dissemination of the information resources of the organization

INFORMATION SYSTEM LIFE CYCLE

❑ **Feasibility analysis**

❑ Requirement collection and analysis

❑ Design

❑ Validation and acceptance testing

- Analyze potential application areas
- Identify the cost for information gathering and dissemination
- Determine the complexity of data and process
- Perform cost-benefit studies
- Set up priorities among applications

INFORMATION SYSTEM LIFE CYCLE

- ❑ Feasibility analysis
- ❑ **Requirement collection and analysis**
- ❑ Design
- ❑ Validation and acceptance testing

- Interact with potential users and user groups to identify problems and needs
- Identify inter application dependencies, communication and reporting procedures.

INFORMATION SYSTEM LIFE CYCLE

- ❑ Feasibility analysis
- ❑ Requirement collection and analysis
- ❑ **Design**
- ❑ Validation and acceptance testing

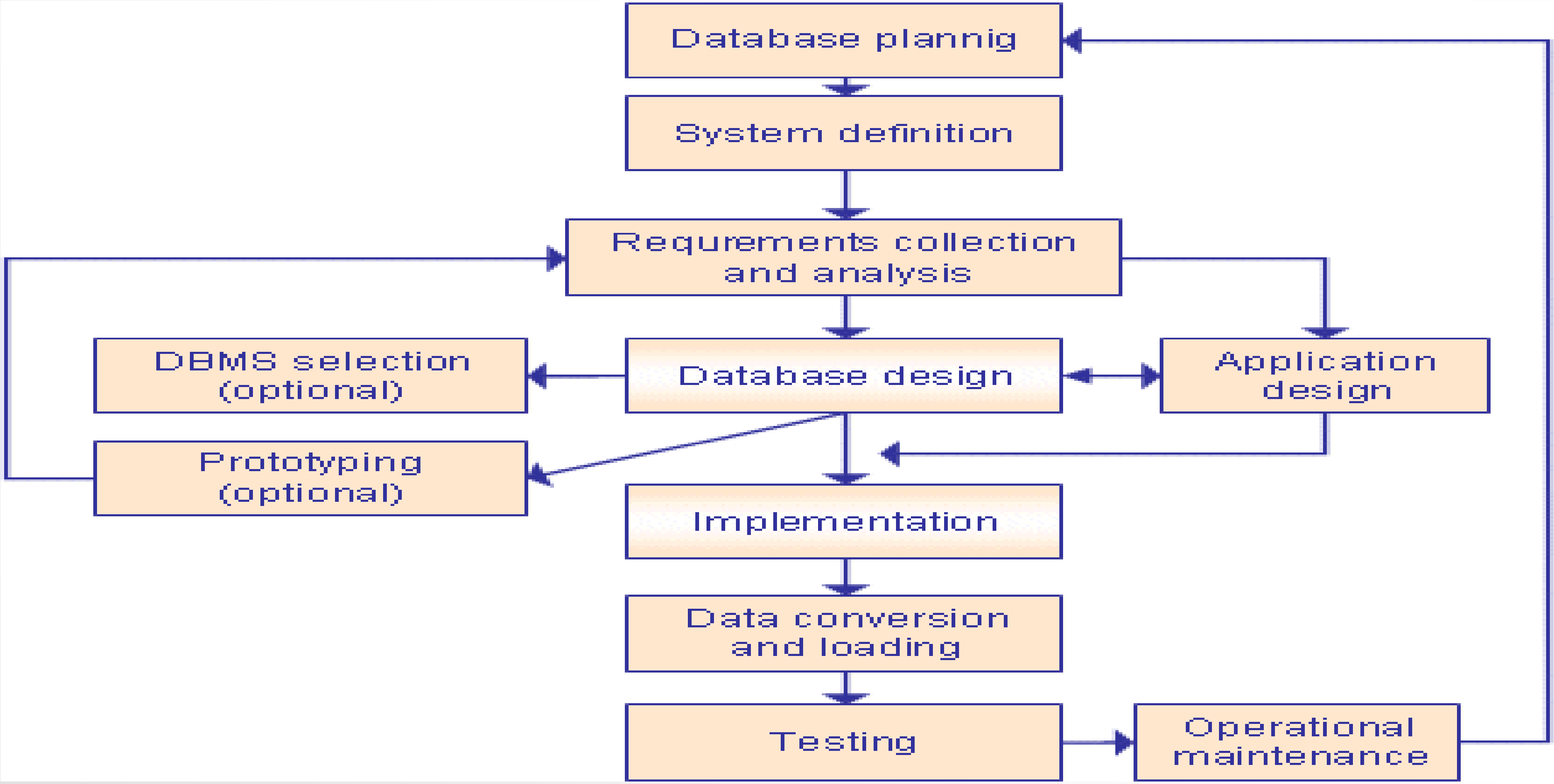
- Design database system
- Design application systems (programs)

INFORMATION SYSTEM LIFE CYCLE

- ❑ Feasibility analysis
- ❑ Requirement collection and analysis
- ❑ Design
- ❑ **Validation and acceptance testing**

DATABASE DESIGN

How to design Database System?



End of 1st Session

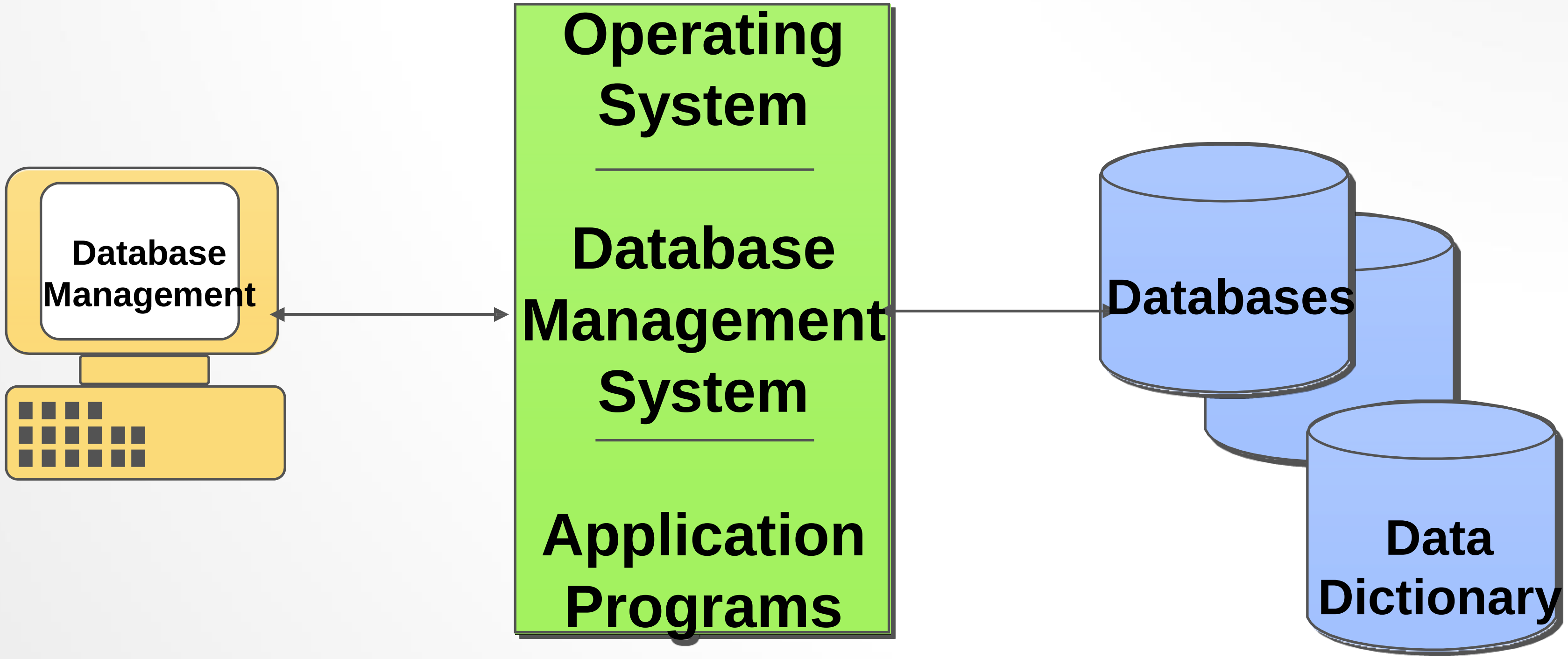
DATABASE DESIGN

Database Management Principles

DATABASE MANAGEMENT PRINCIPLES

Database management systems (DBMS) is a collection of programs that allows users to create and maintain a database, e.g.,

- Construction
- Manipulation
- Sharing
- Protection
- Maintenance



DATABASE MANAGEMENT PRINCIPLES

– **Construction**

– Manipulation

– Sharing

– Protection

– Maintenance

Which database is appropriate for the application?

Does your database need 24x7 availability?

Is the database mission critical, and no data loss can be tolerated?

Is the database large? (backup recovery methods)

What data types do you need? (binary, large objects?)

DATABASE MANAGEMENT PRINCIPLES

- Construction
- **Manipulation**
- Sharing
- Protection
- Maintenance

- Inserting Data
- Retrieving Existing Data
- Updating Data
- Deleting Data

DATABASE MANAGEMENT PRINCIPLES

– Construction
– Manipulation
– **Sharing**

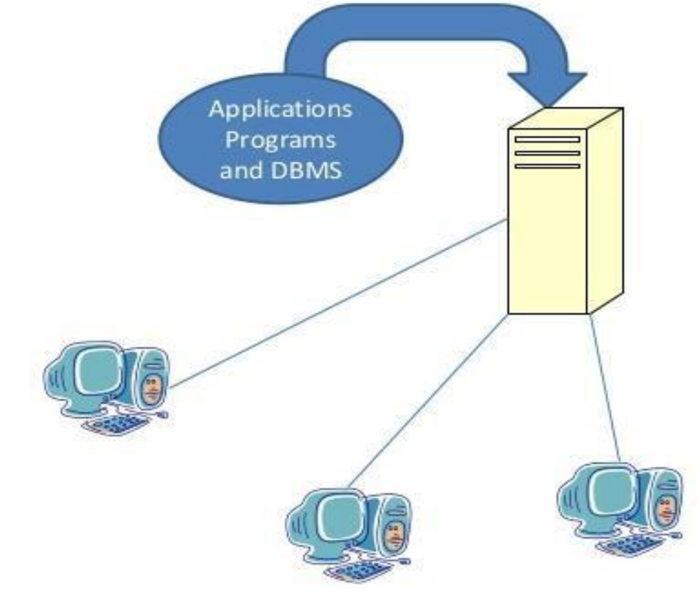
Support for Multiple Users

Multiple Ways of Interfering to the System

– Protection
– Maintenance

Support for Multiple Users

A true RDBMS allows effective sharing of data. That is, it ensures that several users can concurrently access the data in the database without affecting the speed of the data access.



DATABASE MANAGEMENT PRINCIPLES

- Construction
- Manipulation
- **Sharing**
  - **Support for Multiple Users**
  - **Multiple Ways of Interfering to the System**
- Protection
- Maintenance

Multiple Ways of Interfering to the System

For example we can access to MySQL Database server through mysqldump, mysqladmin and mysqlshow etc.

DATABASE MANAGEMENT PRINCIPLES

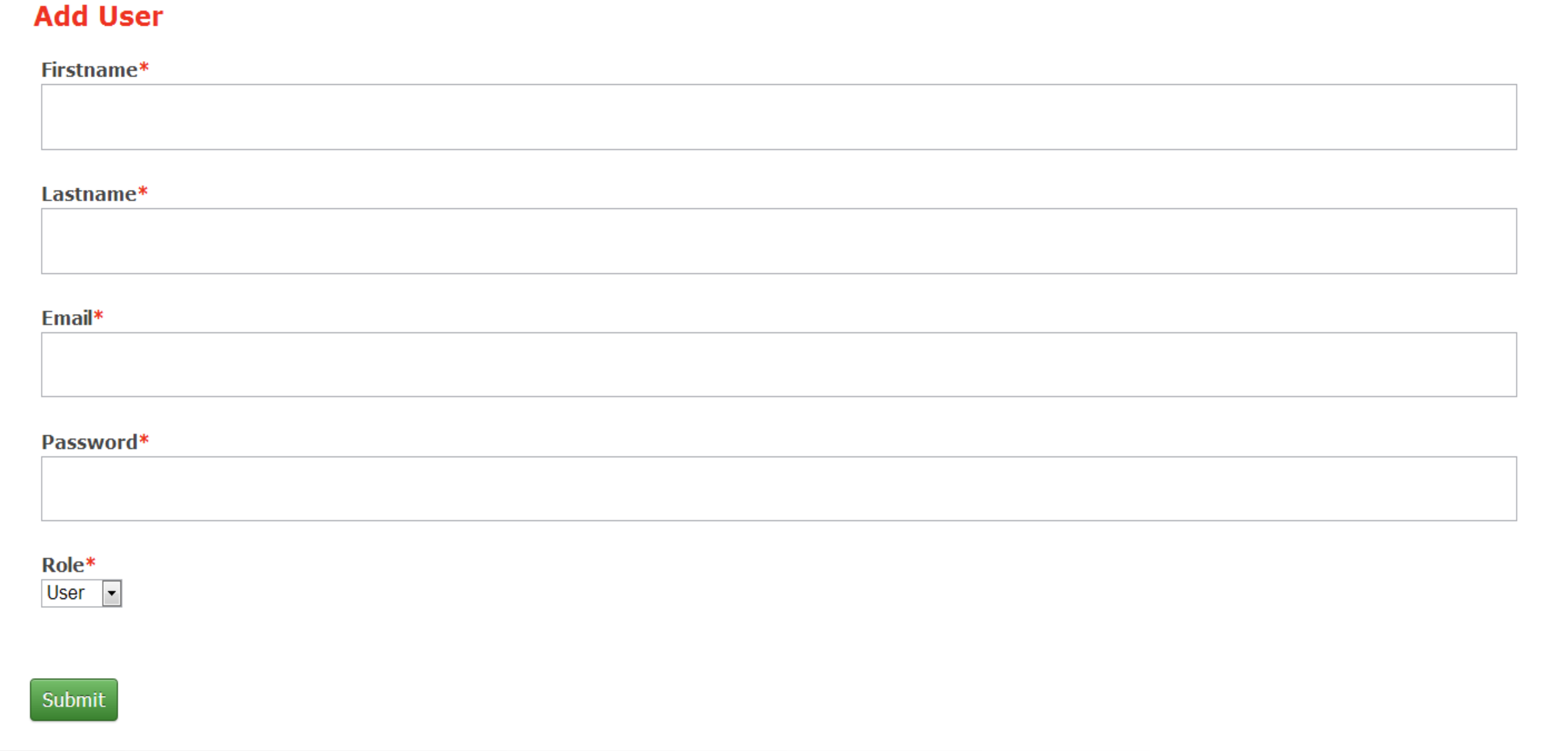
– Construction
– Manipulation
– Sharing
– **Protection**
– Maintenance

- Data entry
- Passwords
- Viruses and worms
- Backups
- System redundancy
- Physical protection
- Private network/wiring
- Encryption
- Training

PROTECTION

- Data entry
  - Passwords
  - Viruses and worms
  - System redundancy
  - Physical protection
  - Private network/wiring
  - Encryption
  - Training

- Both Frontend (Client-side) and Backend (Server-side) validation should be made



Add User

Firstname*

Lastname*

Email*

Password*

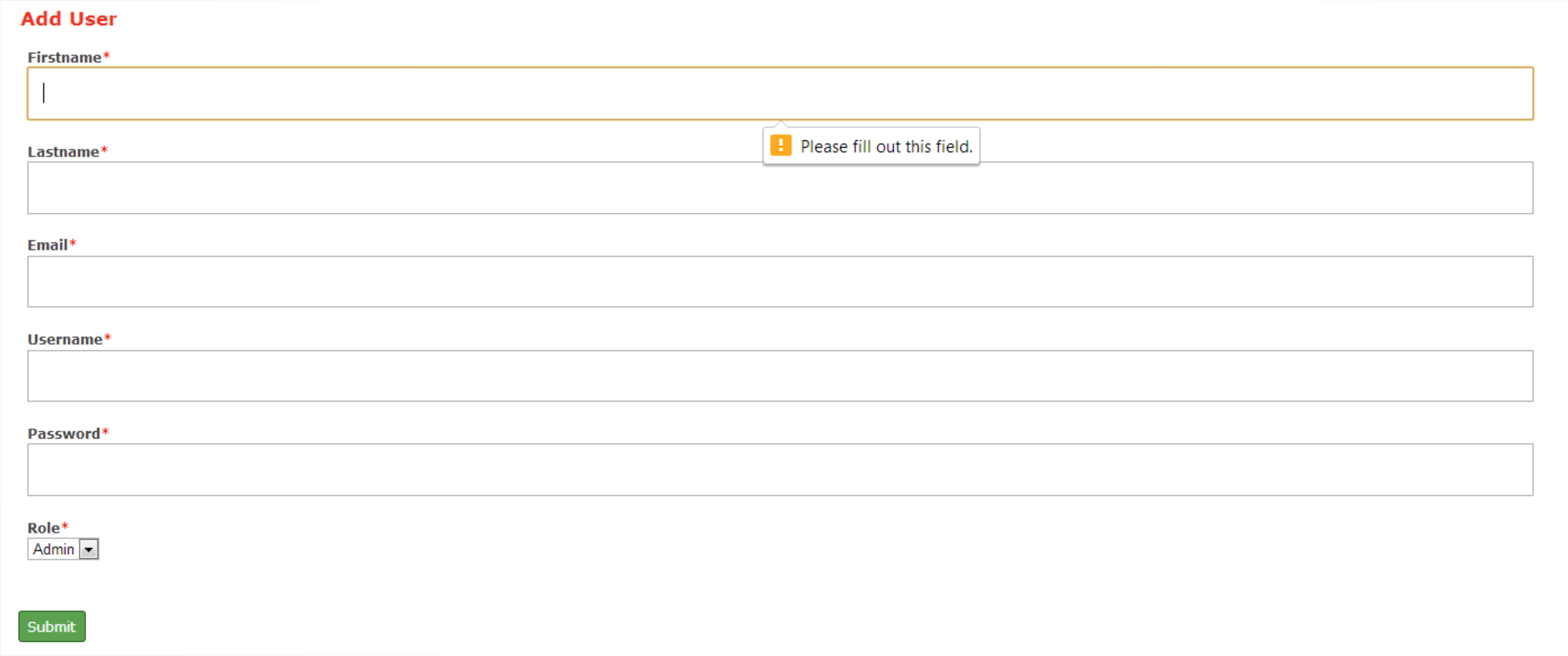
Role*

User

PROTECTION

- **Data entry**
  - Passwords
  - Viruses and worms
  - System redundancy
  - Physical protection
  - Private network/wiring
  - Encryption
  - Training

- Both Frontend (Client-side) and Backend (Server-side) validation should be made



Add User

Firstname*

Please fill out this field.

Lastname*

Email*

Username*

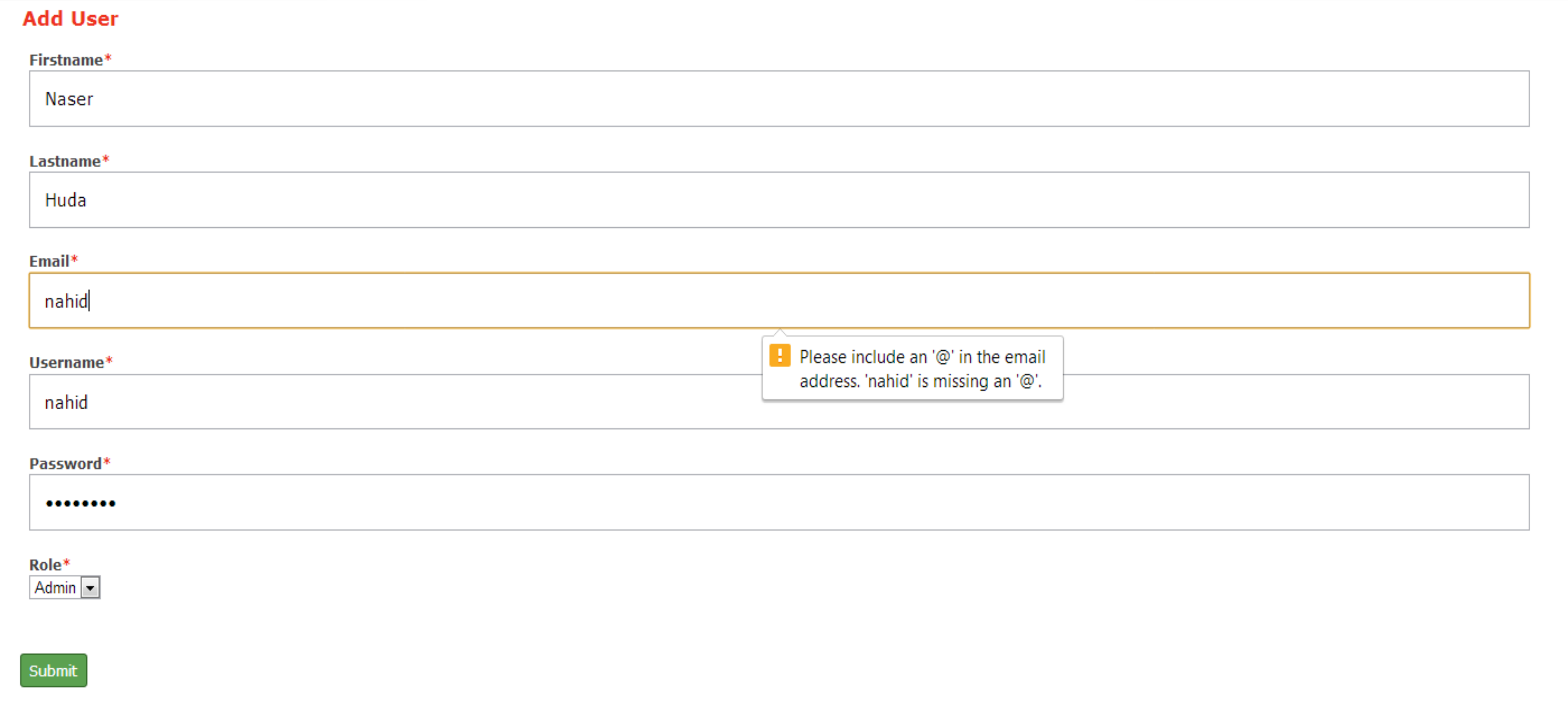
Password*

Role*
Admin

PROTECTION

- **Data entry**
  - Passwords
  - Viruses and worms
  - System redundancy
  - Physical protection
  - Private network/wiring
  - Encryption
  - Training

- **Both Frontend (Client-side) and Backend (Server-side) validation should be made**



Add User

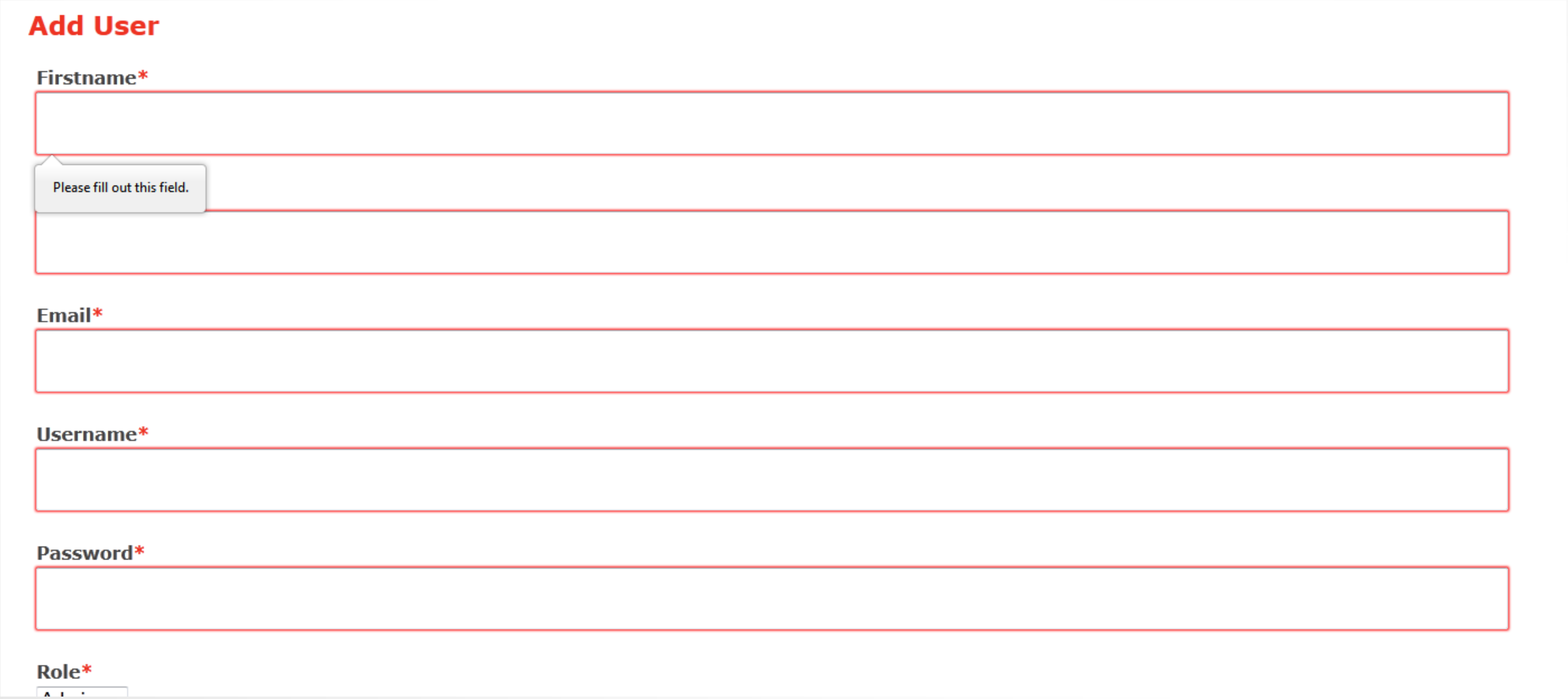
Firstname*	Naser
Lastname*	Huda
Email*	nahid
Username*	nahid
Password*	••••••••
Role*	Admin

Please include an '@' in the email address. 'nahid' is missing an '@'.

PROTECTION

- **Data entry**
  - Passwords
  - Viruses and worms
  - System redundancy
  - Physical protection
  - Private network/wiring
  - Encryption
  - Training

- **Both Frontend (Client-side) and Backend (Server-side) validation should be made**



Add User

Firstname*

Please fill out this field.

Email*

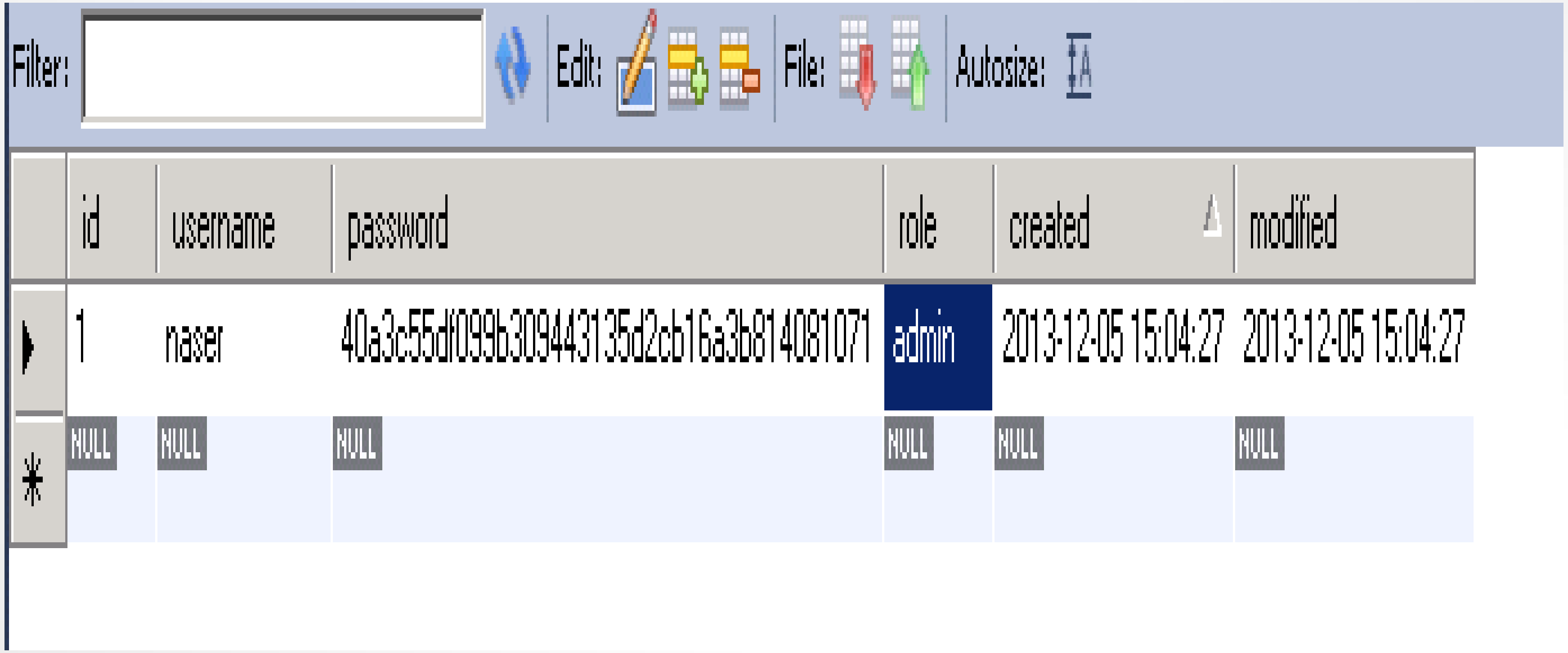
Username*

Password*

Role*

PROTECTION

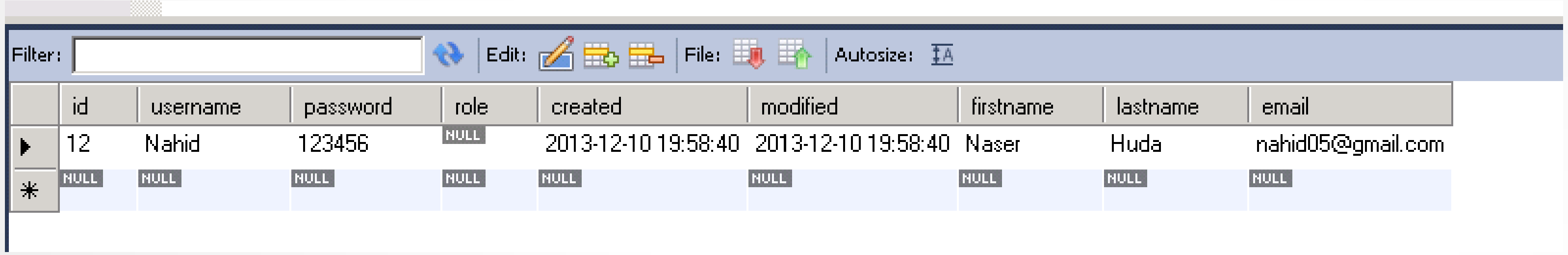
- Data entry
- **Passwords**
- Viruses and worms
- System redundancy
- Physical protection
- Private network/wiring
- Encryption
- Training



id	username	password	role	created	modified
1	naser	40a3c55df099b3094431 35d2cb16a3b814081071	admin	2013-12-05 15:04:27	2013-12-05 15:04:27
NULL	NULL	NULL	NULL	NULL	NULL

PROTECTION

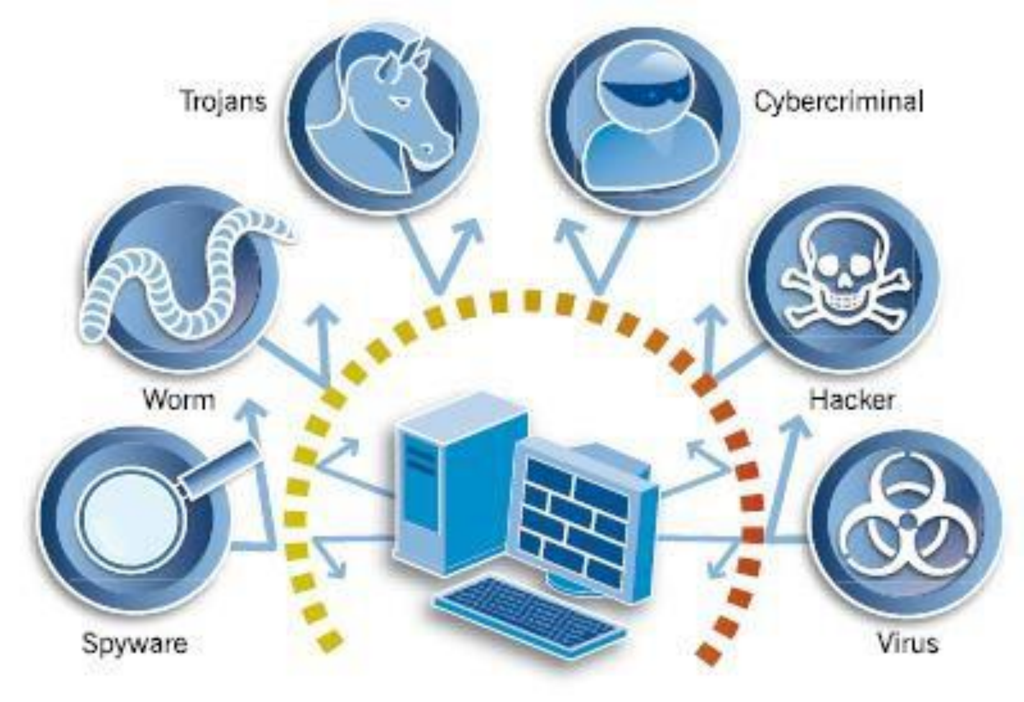
- Data entry
- **Passwords**
- Viruses and worms
- System redundancy
- Physical protection
- Private network/wiring
- Encryption
- Training



	id	username	password	role	created	modified	firstname	lastname	email
▶	12	Nahid	123456	NULL	2013-12-10 19:58:40	2013-12-10 19:58:40	Naser	Huda	nahid05@gmail.com
✳	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

PROTECTION

- Data entry
- Passwords
- **Viruses and worms**
- System redundancy
- Physical protection
- Private network/wiring
- Encryption
- Training



DATABASE MANAGEMENT PRINCIPLES

– Construction
– Manipulation
– Sharing
– Protection

– Maintenance

What is Backup and Recovery?

- In general, backup and recovery refers to the various strategies and procedures involved in protecting your database against data loss and reconstructing the database after any kind of data loss.
- Physical Backups
- Logical Backups

End of 2nd Session

DATABASE DESIGN

What Makes a Good Database?

WHAT MAKES A GOOD DATABASE?

- Storage Needs Met
- Data is Available
- Data Protected
- Data is Accurate
- Acceptable Performance

WHAT MAKES A GOOD DATABASE?

- **Storage Needs Met**
  - Data Is Available
  - Data Protected
  - Data Is Accurate
  - Acceptable Performance

- Have all storage needs been met for the database?
- Has all data been stored effectively?
- Is the hardware adequate for storage needs?
- Does the database software meet the storage needs?
- How easy is it to access offline data storage?

WHAT MAKES A GOOD DATABASE?

- Storage Needs Met
- **Data is Available**
- Data Protected
- Data Is Accurate
- Acceptable Performance

Data availability is also related to the user’s expected and perceived performance of the data- base. Consider this: A user starts an application and performs some function that requests data. Depending on the nature of the request, it may take several minutes or seconds for data to become available, or it might appear instantly on the screen. In the production environment, perception is reality, and if a user feels an application or database is slow, it will create the perception of poor performance.

WHAT MAKES A GOOD DATABASE?

- Storage Needs Met
- Data Is Available
- **Data Protected**
- Data Is Accurate
- Acceptable Performance
- Redundant Data Is Minimized

- Does security exist in the database?
- Is the data protected from outside users?
- Is the data protected from internal users?
- How easy is it for unauthorized users to access the data?
- How easy is it to grant and revoke data access to various groups of users?

WHAT MAKES A GOOD DATABASE?

- Storage Needs Met
- Data Is Available
- Data Protected
- **Data is Accurate**
- Acceptable Performance

- Has referential integrity been applied (primary key and foreign key constraints)?
- What other constraints have been established to check the uniqueness or validity of data?
- Are data relationships easily maintained within the database?
- How easy is it for the end user to enter invalid data into the database?

WHAT MAKES A GOOD DATABASE?

- Storage Needs Met
- Data Is Available
- Data Protected
- Data Is Accurate
- **Acceptable Performance**

- What is the expected response time for transactions and small queries?
- How does the database perform overall according to the end user?
- How does the database perform during high peak times of transactional activity?
- How does the database perform during batch operations, such as massive data loads and queries against large amounts of data?

DATABASE DESIGN

Statistical Database requirements analysis

STATISTICAL DATABASE REQUIREMENTS ANALYSIS

A statistical database management system (SDBMS) is a database management system that can model, store and manipulate data in a manner well suited to the needs of users who want to perform statistical analyses on the data. Statistical databases have some special characteristics and requirements that are not supported by existing commercial database management systems. For example, while basic aggregation operations like SUM and AVG are part of SQL

STATISTICAL DATABASE REQUIREMENTS ANALYSIS

Objectives:

- Appreciate the Features of Statistical database system (DBMS)
- Define Statistical database system.
- Use High-Level Concept of Statistics data in policies formulation.
- State some major concepts of Statistical database Models.
- Design the major components of Statistical database and Modeling.

STATISTICAL DATABASE REQUIREMENTS ANALYSIS

The design of a Statistical Database (Micro, Macro and Metadata Modeling)

- Micro data as primary or basis data on individuals, objects or events representing sampled, census or collected data.
- Macro data as grouped or aggregated data (summarized data) which are cross-classified by a set of categorical attributes (variables). The summary attribute represents counts (frequencies), means, indices or other statistics characterizing a set (population) of individuals, objects or events.
- Metadata describing the micro- and macro data on the semantic, structural, statistical and physical level in such a way that they can be stored transformed retrieved and transmitted in a reasonable way. It covers the whole data life cycle, i.e. the data collecting from the data source, the data storing, the data processing and retrieval, and the data disseminating within the electronic data interchange (EDI)

STATISTICAL DATABASE REQUIREMENTS ANALYSIS

Microdata

- List name, age, sex
- From labourcensusemployees
- Where industry = 'whole industry' and year = 1980

Macrodata

- List number (employees), average (employees.income)
- From labourcensus
- Where industry - 'whole industry' and year - 1980
- Cross-classified by age^group and sex

Metadata

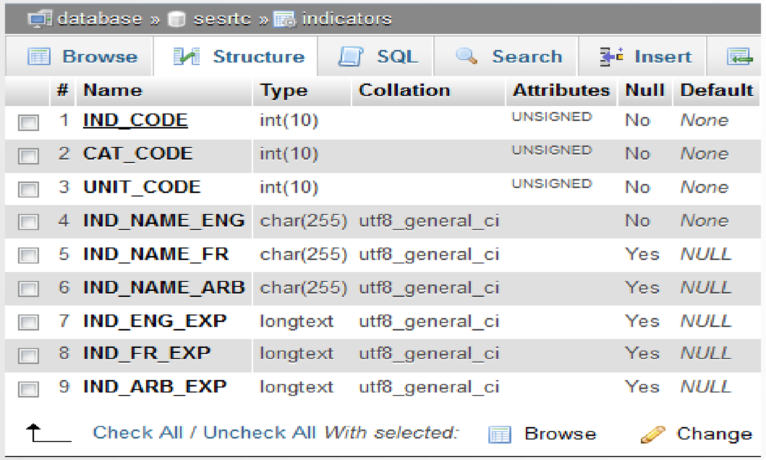
- Household All the people belong to a household who live there together and have a joint budget
- Each person who has an own
- Budget forms her own household,
- Summary-attribute (employees)
- Income categoryattribute (employees)
- Domain (industry)

DATABASE DESIGN

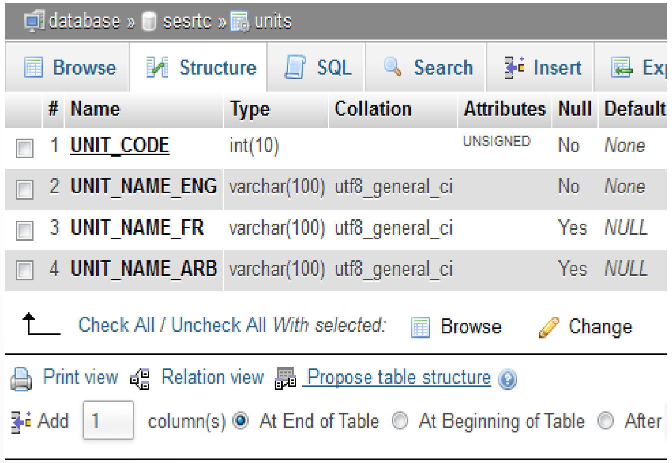
Statistical Data Uploading

STATISTICAL DATA UPLOADING

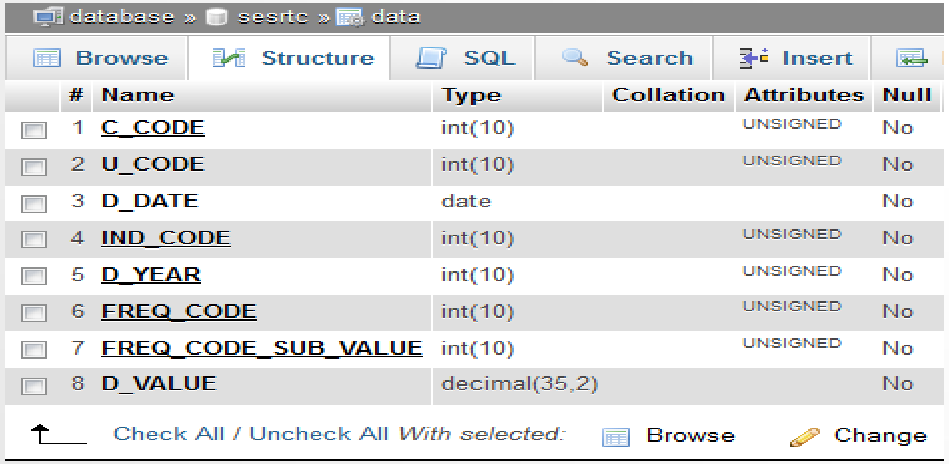
SESRIC has built a relational database to store statistical data collected from various sources.


database » sesrtc » indicators
Browse
Structure
SQL
Search
Insert
Check All / Uncheck All *With selected:*
Browse
Change

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	IND_CODE	int(10)		UNSIGNED	No	*None*
☐	2	CAT_CODE	int(10)		UNSIGNED	No	*None*
☐	3	UNIT_CODE	int(10)		UNSIGNED	No	*None*
☐	4	IND_NAME_ENG	char(255)	utf8_general_ci		No	*None*
☐	5	IND_NAME_FR	char(255)	utf8_general_ci		Yes	*NULL*
☐	6	IND_NAME_ARB	char(255)	utf8_general_ci		Yes	*NULL*
☐	7	IND_ENG_EXP	longtext	utf8_general_ci		Yes	*NULL*
☐	8	IND_FR_EXP	longtext	utf8_general_ci		Yes	*NULL*
☐	9	IND_ARB_EXP	longtext	utf8_general_ci		Yes	*NULL*



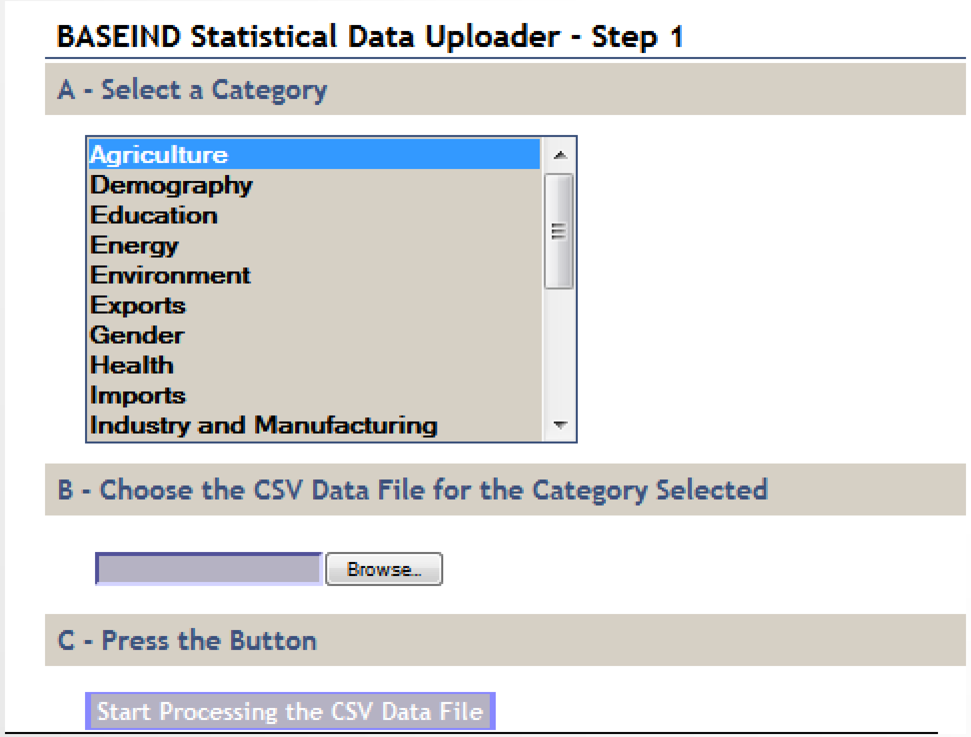
	#	Name	Type	Collation	Attributes	Null	Default
☐	1	UNIT_CODE	int(10)		UNSIGNED	No	*None*
☐	2	UNIT_NAME_ENG	varchar(100)	utf8_general_ci		No	*None*
☐	3	UNIT_NAME_FR	varchar(100)	utf8_general_ci		Yes	*NULL*
☐	4	UNIT_NAME_ARB	varchar(100)	utf8_general_ci		Yes	*NULL*



	#	Name	Type	Collation	Attributes	Null
☐	1	C_CODE	int(10)		UNSIGNED	No
☐	2	U_CODE	int(10)		UNSIGNED	No
☐	3	D_DATE	date			No
☐	4	IND_CODE	int(10)		UNSIGNED	No
☐	5	D_YEAR	int(10)		UNSIGNED	No
☐	6	FREQ_CODE	int(10)		UNSIGNED	No
☐	7	FREQ_CODE_SUB_VALUE	int(10)		UNSIGNED	No
☐	8	D_VALUE	decimal(35,2)			No

STATISTICAL DATA UPLOADING

- 18 categories
- 266 socio-economic variables
- 57 OIC member countries



BASEIND Statistical Data Uploader - Step 1

A - Select a Category

B - Choose the CSV Data File for the Category Selected

C - Press the Button

End of 3rd Session

XML

What is XML?
What made XML necessary?
What does XML provide?
XML with favorite programming language
XML Data Structure for SMC

WHAT IS XML?

Extensible Markup Language:

An activity of the World Wide Web Consortium (W3C) organized and led by Sun Microsystems

Objective:

move the Web to its next stage of evolution by adapting existing ISO standards for markup, linking, and formatting

WHAT MADE XML NECESSARY?

```
<p><b>Mrs. Mary McGoon</b>
<br> 1401 Main Street <br>
Anytown, NC 34829</p>
```

```
<address>
        <name>
                <title>Mrs.</title>
                <first-name> Mary </first-name>
                <last-name> McGoon </last-name>
        </name>
        <street> 1401 Main Street </street>
        <city>Anytown</city>
        <state>NC</state>
        <postal-code> 34829 </postal-code>
</address>
```

WHAT MADE XML NECESSARY?

New data-centric Web applications

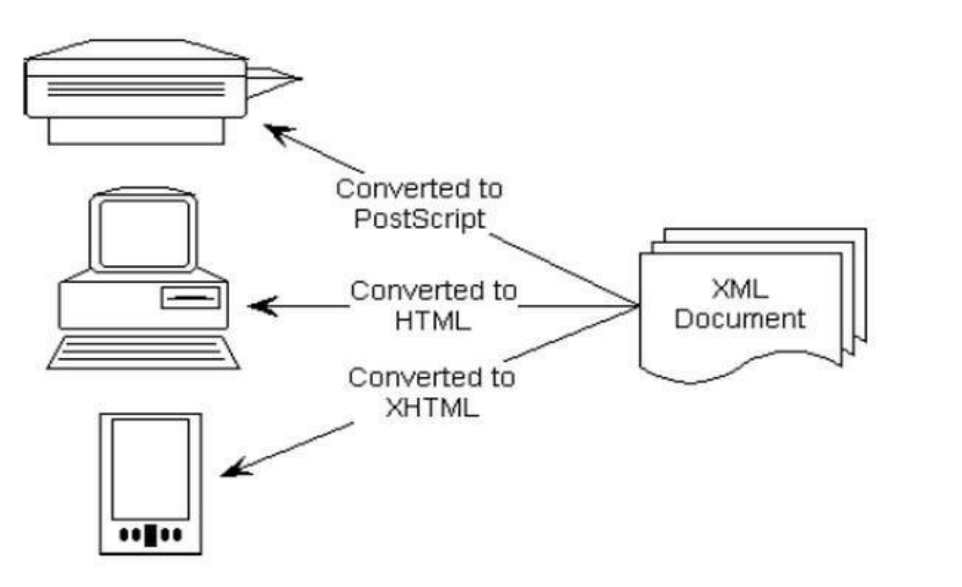
- Data exchange
- Share Data
- Store data

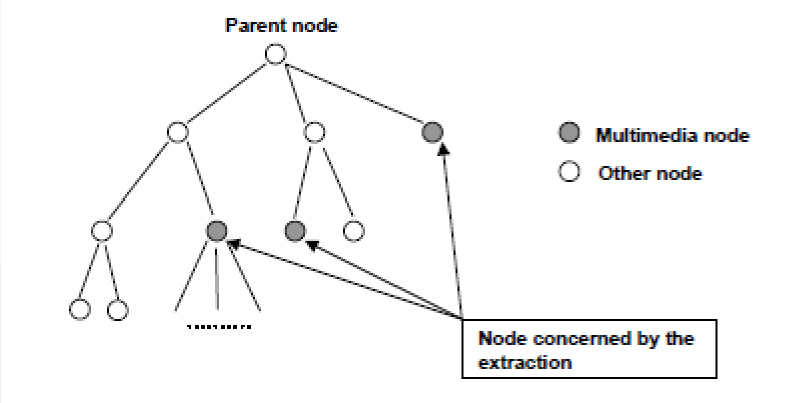
WHAT DOES XML PROVIDE?

XML provides key features needed for a new generation of Web applications:

- Platform-independent
- Language-independent
- Media-independent

```
<bibliography>
        <book>
                <title> Foundations </title>
                <author> Abiteboul </author>
                <author> Hull </author>
                <author> Vianu </author>
                <publisher> Addison Wesley </publisher>
                <year> 1995 </year>
        </book>
</bibliography>
```





XML WITH FAVORITE PROGRAMMING LANGUAGE

We can use XML with any kind of server-side technology, including PHP pages, JavaServer Pages (JSPs), Java servlets, Ruby on Rails, and Microsoft ASP pages etc.

XML WITH FAVORITE PROGRAMMING LANGUAGE

PHP Perl

ASP.NET

Ruby on Rails

XML, FLEX, HTTPSERVICE, PHP AND SQL

We can use the following components to manage and insert data

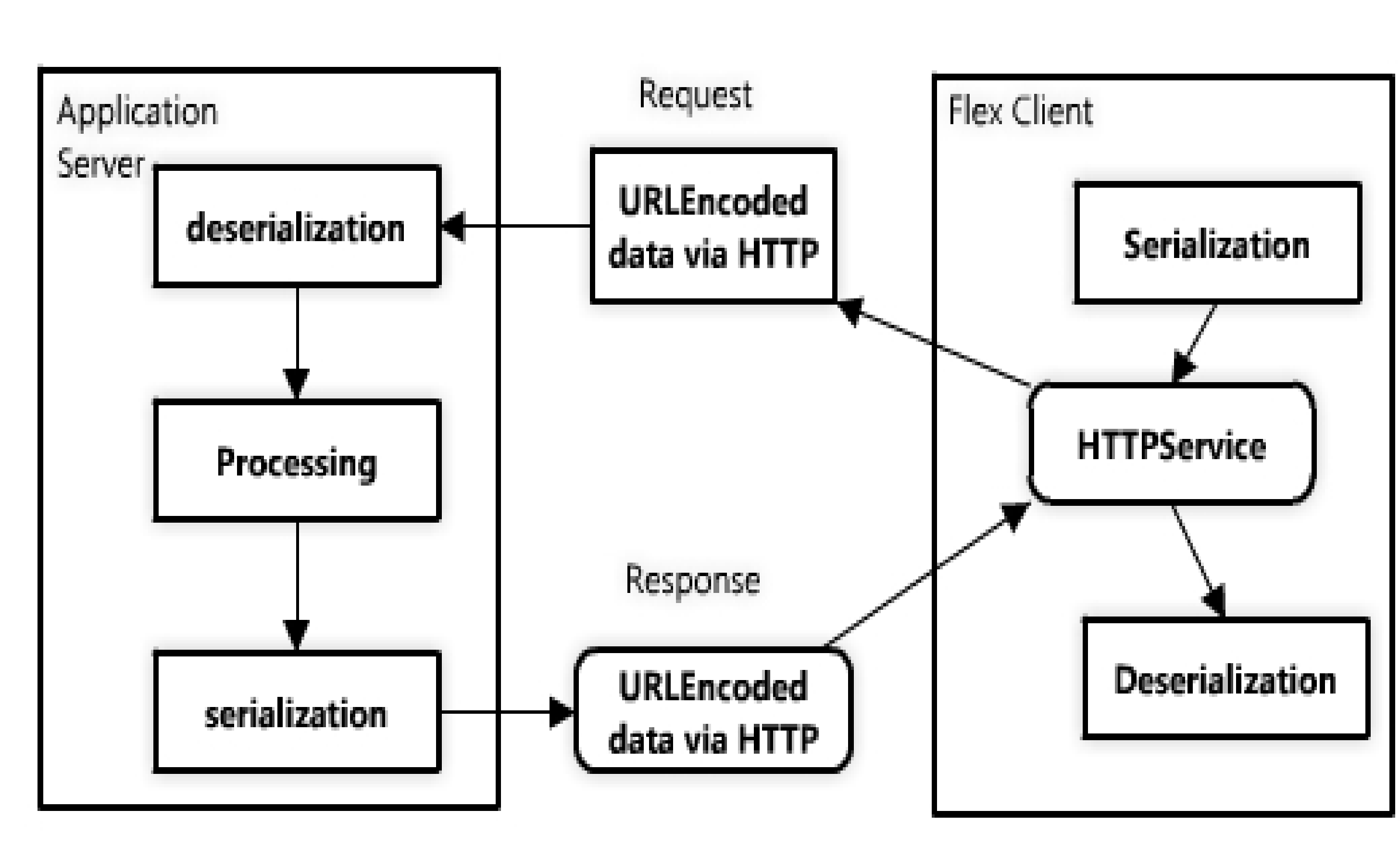
- Flex HTTPService
- PHP and
- SQL database

XML, FLEX, HTTPSERVICE, PHP AND SQL

```
<?xml version="1.0"  encoding="utf-8"?>
<mx:Application xmlns:m.x=''http://www.adobe.com/2006/m.xml'' l.ayout=''absolute''
xmlns="*'" creationCormplete="send_data()">
    <mx:Script>
        <![CDATA[
      privat:e function send_dat:a(): void {
        userRequest:.send();
       }
        ]]>
    </mx.:Script>
    <mx:Form x="22" y="lO" wi.dth="493">
        <mx:HBox>
            <mx:Label text=''Username"/>
            <mx:Textinput id="username"/>
        </mx:HBox>
        <mx:HBox>
            <mx:La.bel text="Email Address"/>
            <mx:Textinput id="emailaddress"/>
        </mx:HBox>
        <mx:Button la.bel="Submit" click="send_data()"/>
    </mx:Form>
  <n1x:Dat:aGrid id="dgUserRequest:" x="22" y="128" dat:aProvider="{userRequest:last:Result:.users.user}">
        <mx:columns>
            <mx:DataGridColum:n headerText="User ID" dataField="userid"/>
            <mx:DataGridColumn .headerText="User Name" dataField="username"/>
        </mx:columns>
    </mx:DataGrid>
    <mx:Textinput x="22" y="292" id="selectedemailaddress"
        text="{dgUserRequest.selecteditem.emailaddress}"/>
  <n1x: HTTPService id="userReque·st:" url="http: //local host:/rnyproj/request:_post:2. php" useProxy="fal se"
  rnet:hod="POST">
    <Il'Ix: reque·st: XIlillns="">
      <usernarne>{usernanle. t:ext:}</usernarne>
      <enlailaddress>{ellilailaddresst:ext:}</ernailaddress>
    </Il'Ix:request:>
  </IlilX: HTTPService>
</mx:Application>
```

XML, FLEX, HTTPSERVICE, PHP AND SQL

Data Communication between PHP and Flex



XML DATA STRUCTURE FOR SMC

```
<dataset>
	<item>
		<Who>Turkey</Who>
		<When>1980</When>
		<DataX>32.94</DataX>
		<DataY>18394.00</DataY>
		<DataR>44105216.00</DataR>
		<Estimated>0</Estimated>
	</item>
	<item>
		<Who>Turkey</Who>
		<When>1981</When>
		<DataX>33.11</DataX>
		<DataY>18472.00</DataY>
		<DataR>45130008.00</DataR>
		<Estimated>0</Estimated>
	</item>
	…
</dataset>
```

END

DAY

VISUALIZATION TOOLS DEVELOPMENT

- Visualization Tools Essentials
- Flex in Visual Programming (Flex and PHP)
- Time Series Statistical Data Visualization
- SESRIC SMC Source Code Analysis

VISUALIZATION TOOLS ESSENTIALS

The Trend Toward of Visualization-based Data Discovery Tools

- Big data is creating unprecedented opportunities for businesses to achieve deeper, faster insights that can strengthen decision making, improve the customer experience, and accelerate the pace of innovation. But today, most big data yields neither meaning nor value. Businesses are so overwhelmed by the amount and variety of data cascading into and through their operations that they struggle just to store the data—much less analyze, interpret, and present it in meaningful ways.

VISUALIZATION TOOLS ESSENTIALS

The Struggle to Make Meaning Out of Big Data

Key Results from IT Manager Survey

- 33% of companies surveyed are working with very large amounts of data (500 TB or more)
- 84% of IT managers are analyzing unstructured data.
- 44% of those who are not analyzing unstructured data expect to do so in the next 12 to 18 months
- By 2015, IT managers expect that 63% of all analytics will be done in real time
- Of seven possibilities, IT managers indicated that they would find the most value in receiving help deploying cost-effective data visualization methods

VISUALIZATION TOOLS ESSENTIALS

Key Features of Visualization-based Data Discovery Tools

- Enable real-time data analysis
- Support real-time creation of dynamic, interactive presentations and reports
- Allow end users to interact with data, often on mobile devices
- Hold data in-memory, where it is accessible to multiple users
- Allow users to share and collaborate securely

VISUALIZATION TOOLS ESSENTIALS

Protecting Data Quality

Data security and governance have always been part of BI, but big data introduces added legal, ethical, and regulatory issues. Visualization based data discovery tools further those concerns, particularly in the area of data quality.

The risk to data quality stems from one of the great benefits of visualization-based data discovery tools: their ease of use. The tools facilitate self-service BI, enabling more users to perform advanced analyses.

FLEX IN VISUAL PROGRAMMING

What does it do?

Flex is a powerful, open source application framework that allows you to easily build mobile applications for iOS, Android™, and BlackBerry® Tablet OS devices, as well as traditional applications for browser and desktop using the same programming model, tool, and codebase.

FLEX IN VISUAL PROGRAMMING

Enterprise-class programming model

Use constructs like strong typing, inheritance, and interfaces to program more efficiently. Extensive mobile and traditional components help speed development. Flex applications can access device capabilities such as GPS, accelerometer, camera, and local database. Cross-platform and native experience Create applications that run consistently across Android, BlackBerry Tablet OS, and iOS devices, as well as inside the browser and on traditional desktop computers. Although cross platform, with Flex you get an uncompromised native experience on each platform.

FLEX IN VISUAL PROGRAMMING

End-to-end tooling

Build Flex applications more easily using Adobe Flash Builder, PowerFlasher FDT, Flash Develop or JetBrains IntelliJ IDEA. Productivity features in most IDEs include on-device debugging and mobile simulators for testing across screen sizes and resolutions. You can even use our command-line tooling with your favorite text editor!

FLEX IN VISUAL PROGRAMMING

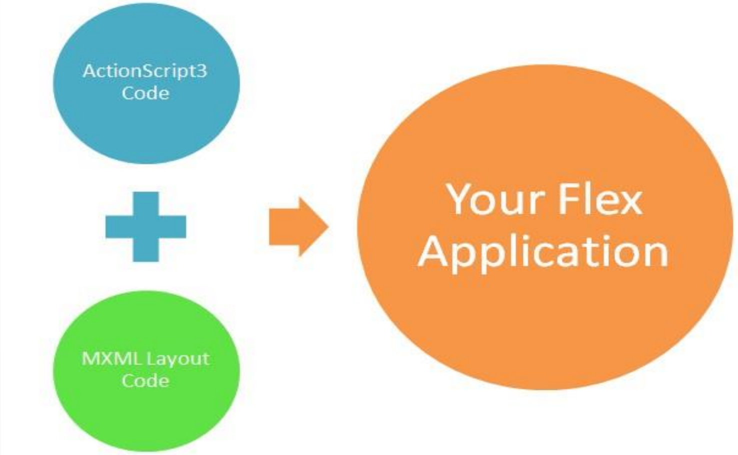
Server integration

Integrate with all major back ends including Java™, Spring, Hibernate, PHP, Ruby, .NET, Adobe ColdFusion®, and SAP using industry standards such as REST, SOAP, JSON, JMS, and AMF.

FLEX IN VISUAL PROGRAMMING

How it works?

Apache Flex is comprised of a few different components. One component is the compiler which combines MXML (layout) documents with Action Script files to output a SWF application. SWF file as a stand-alone application to be presented by the Adobe Flash Player in the browser, OR you can compile it with Adobe AIR to make native applications on Windows, MacOSX, Android, iOS, or BlackBerry platforms.



FLEX IN VISUAL PROGRAMMING

Flex in Visual Programming

- Unlike a set of HTML templates created using JSPs and servlets, ASP, PHP, or CFML, Flex separates client code from server code. The application user interface is compiled into a binary SWF file that is sent to the client.
- When the application makes a request to a data service, the SWF file is not recompiled and no page refresh is required. The remote service returns only data. Flex binds the returned data to user interface components in the client application.
- For example, in Flex, when a user clicks a Button control in an application, client-side code calls a web service. The result data from the web service is returned into the binary SWF file without a page refresh. Thus, the result data is available to use as dynamic content in the application.

TIME SERIES STATISTICAL DATA VISUALIZATION

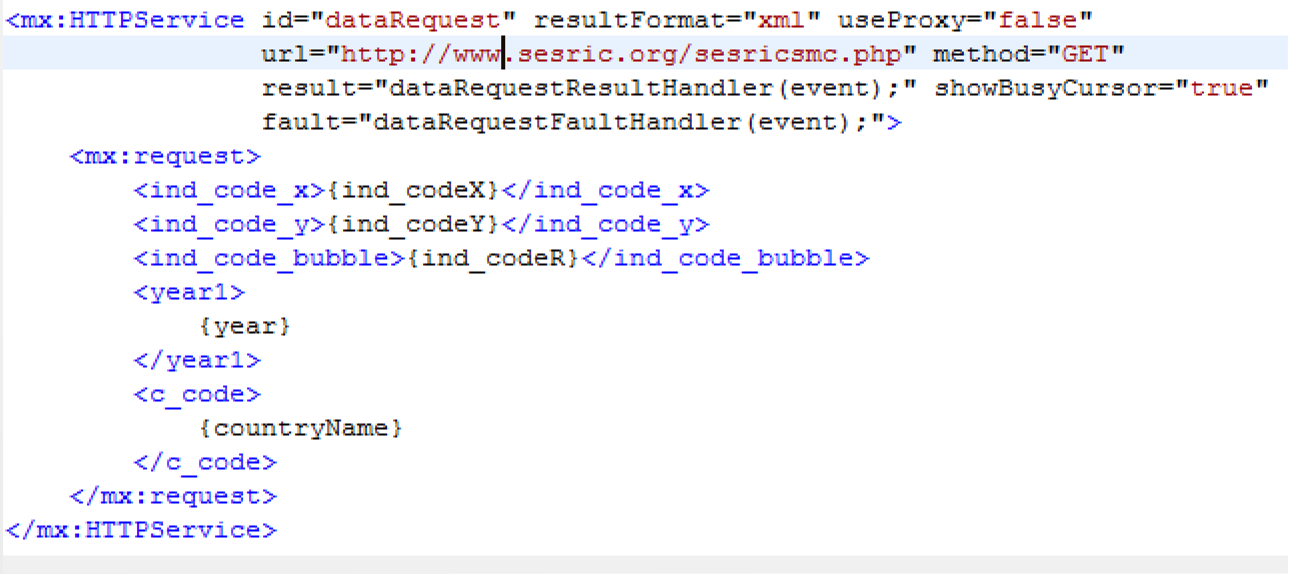
Example will be given from BASEIND Time Series

SESRIC SMC MODULE SHARING

- Platform Independent
- Technology
- Customizable

SESRIC MC SOURCE CODE SHARING

General Overview of Codes and Object Oriented Classes:



```
<mx:HTTPService id="dataRequest" resultFormat="xml" useProxy="false"
                url="http://www.sesric.org/sesricsmc.php" method="GET"
                result="dataRequestResultHandler(event);" showBusyCursor="true"
                fault="dataRequestFaultHandler(event);">
    <mx:request>
        <ind_code_x>{ind_codeX}</ind_code_x>
        <ind_code_y>{ind_codeY}</ind_code_y>
        <ind_code_bubble>{ind_codeR}</ind_code_bubble>
        <year1>
            {year}
        </year1>
        <c_code>
            {countryName}
        </c_code>
    </mx:request>
</mx:HTTPService>
```

SESRIC MC SOURCE CODE SHARING

```
<mx:BubbleChart id="bubbleChart"
            showDataTips="true"
            paddingRight="S"
            paddingLeft="S"
            maxRadius="25"
            minRadius="5"
            dataTipFunction="bubbleChart_dataTipFunc"
            width="693"
            height="442"
            dataTipMode="single" mouseOut="hideTipData(}"
            baseline="2"
            initialize="initBubbleChart();" itemClick:="ltemClicked(event);">
            <mx:verticalAxis>
                <mx:LinearAxis  baseAtZero="true"  labelFunction="linearAxis labelFunc" autoAdjust="true"  />
            </mx:verticalAxis>
            <mx:horizontalAxis>
                <mx:LinearAxis  baseAtZero="true"  labelFunction="linearAxis labelFunc" autoAdjust="true" />
            </mx:horizontalAxis>
            <mx:radiusAxis>
                <mx:LinearAxis />
            </mx:radiusAxis>
    <mx:series>
        <components:CustomBubbleSeries id="bubbleSeries"
            displayName="SERIC1C" fillFunction="myFillFunction"
            xField="DataX"
            yField="DataY"
            radiusField="DataR" CountryName="Who">
        </components:CustomBubbleSeries>
    </mx:series>
</mx:BubbleChart>
```

SESRIC MC SOURCE CODE SHARING

```
<?xml  version="l.O"  encoding="utf-8"?>
<mx:Application xrolns:mx="http://www.adobe.com/2006/mxml." layout="absolute" height="608"
    widtb="941" backgroundColor="white" creationComplete="dataRequest2.send();" initialize="initApp()"  xmlns:components="components.<">
     <mx:Style>
     Corr..bo3ox  {
            dropdownStyleName: myDropdownStyleNom;
        }
        . yDropdoSty:eNo    {
            borderColor: haloBlue;
            borderThickness:   3;
            fontWeight: normal;
        }
        .colorBubble {
            color: #9933FF;
        }
      .com..boStyle{
        fontFamily:myPlai::1Fo::1t;
        fontSize: 12pt;
        font-weight:normal;
      }
      .sizelbl {
        padding-left: 40px;
        padding-top: 30px;
      }
      .si:der  {
        slide-duration:2000;
        slide-easing-function:"Elastic.ease!nOut";
      }
  </mx:Style>
```

SESRIC MC SOURCE CODE SHARING

```
var countryFlag:Striug = "<img src=\"http://www.sesric.org/imgs/country_flags/B.jpg\" alt=\"Country Flag\" />";
var toolTipBubble:String = "";
toolTipBubble += "<img src='../assets/8.jpg' hspace='3' vspace='3'/>";
toolTipBubble += "<font color='ft076baa'><b>";
toolTipBubbl.e += cSI.item.Wbo + " ("+ (yearSl.ider.value + Number(year) - 1).toString() + ")</b></font>";
return toolTipBubble;

          publ.ic function yindicatorChangeB(e:Event):void{
              var Col.Ser:Col.umnSeries =
              barCbart.series[0]; var l.nAxis:LinearAxis =
              new Linear.Axis();
              var l.gAxis:LogAxis = new LogAxis();
              lgAxis.l.abelFunction = logAxis_labelFunc;
              if (yindicatorB.sel.ectedindex == 0){
                  if (vertical.ComboB.sel.ectedlndex == 1) {
                      lgAxis.maximum = l.inear.AxisMaximum;
                      lgAxis.minimum = l.inearAxisHinimum;
                      lgAxis.interval. = 10;
                  } el.se {
                      l.nAxis.maximum   l.inearAxisHaximum + Number(l.inearAxisHaximum/10);
                      lnAxis.minimum   l.inear.Axislofinimum   Number(l.inear.Axislofinimum/25);
                  }
                  ColSer.yField = "DataX";
              } else if (yindicatorB.sel.ectedlndex   2){
                  if (verticalComboB.sel.ectedindex   1) {
                      l.gAxis.maximum '= RAxisHaximum;
                      lgAxis.minimum = RAxisHinimum;
                      lgAxis.interval. = 10;
                  } el.se {
                      lnAxis.maximum   RAxisHaximum +
                      lnAxis.minimum   Number(RAxisHaximum/10); RAxisl1inimum
                  }                                           Number(RAxisl1inimum/25)
                                      ;
```

```
ColSer.yField   "DataR";
```

SESRIC MOTION CHART

THANK YOU