

# Appaserver.org

## Contents:

### **Class Grades Application Appaserver Engine Build Class Grades Application**

- 1) View class grades database schema.
- 2) Define folder
- 3) Define attribute
- 4) Define primary key
- 5) Define relation
- 6) Define foreign key

Transition: Now lets see how Appaserver can help us with the class grades application.

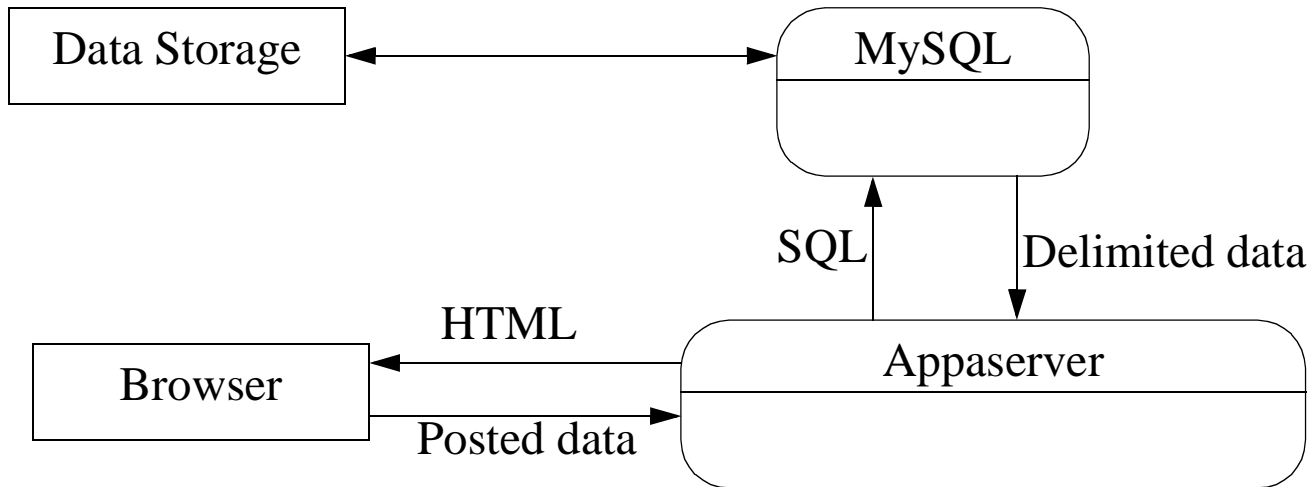
#### <Contents>

Introduction: Appaserver is used to build database applications intended to run in a browser. You do not paint or program screens; instead, you configure Appaserver.

#### <Class grades schema>

Transition: Let's say you've been asked to teach two sections of beginning programming at UC Davis. But maintaining the grade book is going to be a chore. So you pickup a pencil and straight-edge and come up with this database schema.

# Appaserver Data Flow



Appaserver couples MySQL to the browser.

<Appaserver ER>

<Back to screens>

Transition: Now let's build the class grades application.

# Class Grades Folders

- \* Seasons Table
- \* Terms Table
- \* Classes Prompt
- \* Enrollments Prompt
- \* Assignments\_and\_Tests Prompt, Sort
- \* Grades Prompt, Sort

These are the folders needed for the application.

# Class Grades Attributes

* season	char 6
* class_name	char 30
* student_name_last_first_middle	char 40
* assignment_or_test_name	char 30
* counts_for_average_yn	char 1
* year	integer 4
* section	integer 2
* personal_identification_number	integer 4
* max_points	integer 3
* points	integer 3
* assignment_or_test_date	date 10, Hint

These are the attributes  
needed for the application.

# Class Grades Folder Attributes

## \* Seasons

Season Primary Key 1

## \* Terms

Season Primary Key 1

Year Primary Key 2

## \* Classes

Class\_name Primary Key 1

Section Primary Key 2

Season Primary Key 3

Year Primary Key 4

New we assign the attributes to the folders.

# Class Grades Folder Attributes

## \* Enrollments

Student_Name_Last_First_Middle	Primary Key 1
Class_Name	Primary Key 2
Section	Primary Key 3
Season	Primary Key 4
Year	Primary Key 5
Personal_Identification_Number	Display Order 1, omit insert

# Class Grades Folder Attributes

## \* Assignments\_and\_Tests

Assignment_or_Test_Name	Primary Key 1
Class_Name	Primary Key 2
Section	Primary Key 3
Season	Primary Key 4
Year	Primary Key 5
Assignment_or_Test_Date	Display Order 1
Max_Points	Display Order 2
Counts_For_Average_yn	Display Order 3

# Class Grades Folder Attributes

## \* Grades

Student_Name_Last_First_Middle	Primary Key 1
Class_Name	Primary Key 2
Section	Primary Key 3
Season	Primary Key 4
Year	Primary Key 5
Assignment_or_Test_Name	Primary Key 6
Points	Display Order 1



# Class Grades Relations

- |                         |                       |
|-------------------------|-----------------------|
| * Terms                 | Seasons               |
| * Classes               | Terms                 |
| * Enrollments           | Classes               |
| * Assignments_and_Tests | Classes               |
| * Grades                | Enrollments           |
| * Grades                | Assignments_and_Tests |
| * Grades                | Classes (Preprompt)   |

Next, we'll assign the relations.

# Class Grades Operations

- \* Seasons Delete, Detail
- \* Terms Delete, Detail
- \* Classes Delete, Detail
- \* Enrollments Delete, Detail
- \* Assignments\_and\_Tests Delete, Detail
- \* Grades Delete, Detail

Now, we'll enter in the operations -- delete and detail.

# Role Folder Permissions

* New role = Instructor	
* Seasons	Insert, Update
* Terms	Insert, Update
* Classes	Insert, Update
* Enrollments	Insert, Update
* Assignments_and_Tests	Insert, Update
* Grades	Insert, Update

Now, we'll enter in the folder permissions. First, we need to create the role 'Instructor', and then assign everyone in that role full access.

# Role Operation Permissions

* Seasons	Instructor, Delete, Detail
* Terms	Instructor, Delete, Detail
* Classes	Instructor, Delete, Detail
* Enrollments	Instructor, Delete, Detail
* Assignments_and_Tests	Instructor, Delete, Detail
* Grades	Instructor, Delete, Detail

Now, we'll give instructors permission to perform the delete and detail operations.

# Class Grades Processes

- \* `Generate_Random_Pins`  
Folder = Classes
- \* `Average_Report`  
Folder = Classes  
Prompt = `Omit_Student_Name_yn`

There are two processes already written: one to generate the random pins, and another to report the grade averages.

[<View the query screens>](#)

# Process Permissions

- \* Average\_Report Instructor
- \* Generate\_Random\_Pins Instructor

Now we have to give the instructor role permission to execute these processes.

# Creating the Application

\* Run process = Create Application

Now, we can create the application.

# Class Grades Execution

Finally, we'll create a new Appaserver user and assign him or her to the instructor role.

## Conclusion:

The Appaserver database schema contains many elements we didn't cover. Each one of them represents a feature.

Whether your information system needs are six folders or sixty, Appaserver can run it. The rudimentary details of inserting, updating, and deleting data while maintaining database integrity have been written -- you can assume they'll work. All you then need to program are the processes and reports that fill in the gaps.

Consider using Appaserver as your application server.