

## Structure of p:

The screenshot shows the phpMyAdmin interface for the 'cs275\_wasingej' database. The 'Structure' tab is selected for table 'p'. The table has 5 columns: pno, pname, color, weight, and city. Below the column list, the 'Information' section shows space usage and row statistics.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	pno	char(6)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	pname	char(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	color	char(6)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	weight	smallint(6)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	city	char(15)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Space usage		Row statistics	
Data	0 B	Format	static
Index	1 KiB	Collation	latin1_swedish_ci
Total	1 KiB	Rows	0
		Creation	Apr 10, 2014 at 05:13 PM
		Last update	Apr 10, 2014 at 05:13 PM

## structure of s:

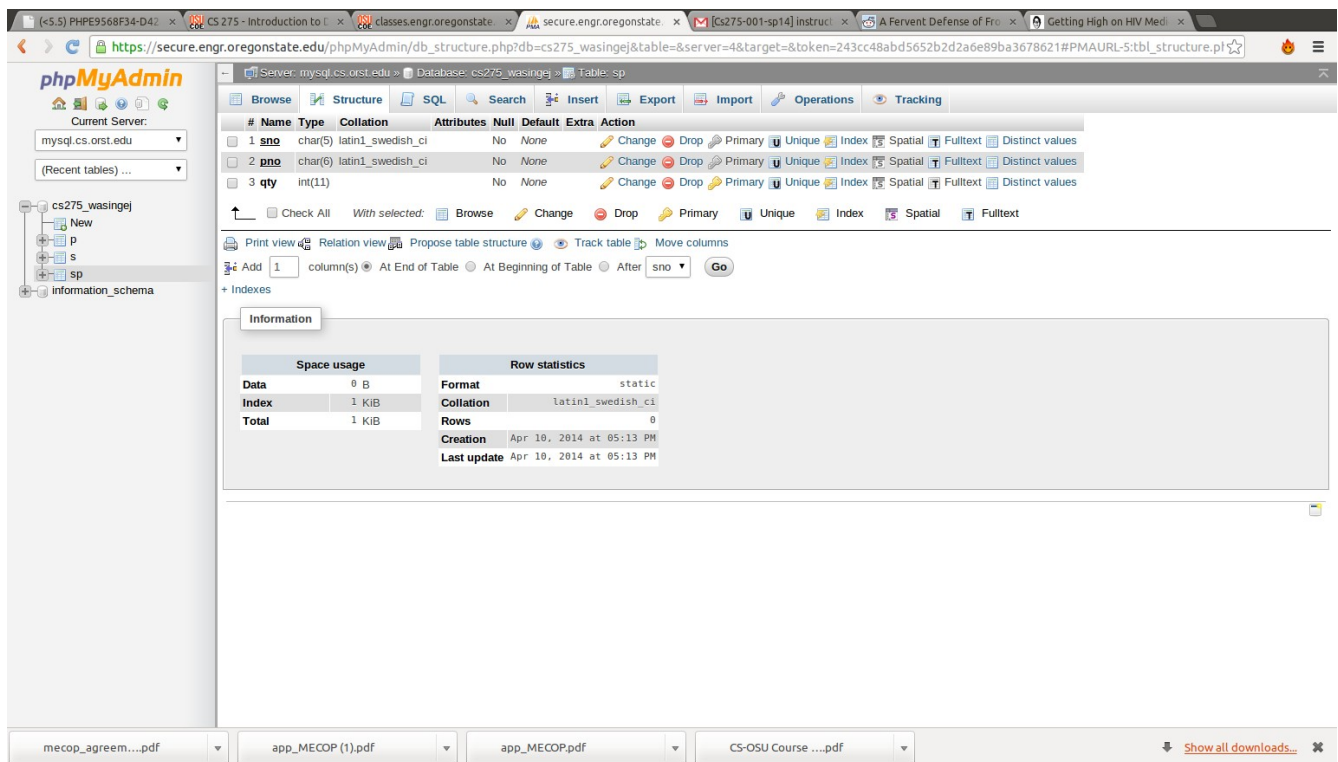
The screenshot shows the phpMyAdmin interface for the 'cs275\_wasingej' database. The 'Structure' tab is selected for table 's'. The table has 4 columns: sno, sname, status, and city. Below the column list, the 'Information' section shows space usage and row statistics.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	sno	char(5)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	sname	char(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	status	smallint(6)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	city	char(15)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Space usage		Row statistics	
Data	0 B	Format	static
Index	1 KiB	Collation	latin1_swedish_ci
Total	1 KiB	Rows	0
		Creation	Apr 10, 2014 at 05:13 PM
		Last update	Apr 10, 2014 at 05:13 PM

## strucutre of sp:



### Exercise 1.1:

Operating system files are not designed for rapid access/storage of large amounts of table-structured data. In addition, a standard operating system may not possess some or most of the functionality that would be needed by a database program.

A DBMS (Database Management System) would be a much better option for storing a large amount of data for several seasons. First, a DBMS is optimized for the kind of data access and modification that an application accessing large amounts of data (possibly concurrently) needs. Second, a DBMS possesses a lot of built in functionality that would be useful for developers meaning that overall development time is reduced. Third, a DBMS is built to be secure meaning that data organized in a DBMS would have more security governing its access and modification.

A situation where a DBMS would not be useful is if the size/importance of data in a database is not large enough to warrant investing resources in a DBMS.

### Exercise 1.3:

Logical independence allows the user to be shielded from the programmatic organization of data. Take for example, a database which contains information about the classes OSU students are currently taking. The database is ordered so that there are two separate tables: one to hold entries for each student and the CRN's of the classes they are taking, another to hold entries where each entry contains a CRN and the name of an instructor teaching the course that associates with that CRN. Perhaps, a DBA decides to make a change where instead of having multiple entries relating one instructor one course CRN, the new database would have entries with an instructor in one column and CRNs for the courses that the instructor taught in the other columns for the entry. The end user is unaware of this change in this logical structure of the data.

Physical independence refers to a separation between the end user's perception of the application and the physical structure of database data in a DBMS. Although the underlying details of

data access may vary (in order to best optimize data access/manipulation), the application is unaware of this change as queries are still made to the data through the DBMS in the same manner.

#### Exercise 1.4:

External schema specifies how data relations contained in the database is presented to the end-user. While the database itself contains all of the data entries necessary to run the application, access to this data can be restricted to prevent the user of the application from having access to data that they shouldn't. In addition, external schema specifies how data contained in the application database is visualized and presented to the user.

Conceptual schema details how data is to be related in the database. The conceptual schema sits at a level of abstraction below that of the external schema. Conceptual data specifies the relationships between entities in a database in terms of the data model of the DBMS.

Physical schema specifies how data outlined in the conceptual schema is physically stored on hard disks. Physical schema sits at a level of abstraction lower than conceptual schema and typically deals with structuring data in the DBMS in a manner for most efficient access by an application.

As with the concepts of physical and logical data independence, the idea of separating design tasks into separate schema naturally facilitates the development of secure and maintainable code. The design of a given schema can be considered to be independent from the details governing a lower level schema. For example, a change in the type of file in which data is stored (change in physical schema) should not change the structure of code relating to the conceptual schema.

#### Exercise 1.6:

1. A security facility- Scrooge should buy this feature. Scrooge has explicitly stated that the data contained in his database is to be viewed by nobody. If he wants to ensure that his data remains secure from prying eyes, he should invest in security.
2. Concurrency Control- Scrooge should not buy concurrency control because he will be the only person making transactions on his database.
3. Crash recovery- Scrooge should not invest in crash recovery. Because he will be the only user making transactions on his database and is unlikely to update multiple entries in the data in one query, a worst case scenario crash could result in unfilled data for one entry. Although it would be nice to be able to instantly rollback whatever data has been corrupted, the monetary costs of having this feature are too great to warrant purchasing it.
4. A view mechanism- Scrooge should not purchase a view mechanism. If he is the only user accessing and modifying data entries about his employees, his knowledge about the database will be intimate enough to allow him not to purchase a view mechanism.
5. A query language- Scrooge should purchase a query language. If he needs to select and modify select entries in his database having a query language would allow him to do it quickly and efficiently. If he didn't have a query language, maintaining the data by hand would be almost impossible when the database became large enough.

#### Part G:

## 1. Introduction

Recent decades have seen huge economic developments in places that were once deemed “3rd World Countries”. The rapid expansion of the Internet to newly developed parts of the world has opened a business niche for digital advertising of tourism opportunities. My web application will provide an online interface for an island town to advertise opportunities which may be of interest to potential consumers.

The application will consist of an 'index' page which renders a colorful interactive map of the tropical island with different points of interest. Each of these points of interest can be clicked on by a visitor to the website. When clicked on, a point of interest will direct the user to a page where they can more information about the venue.

Overall, this web interface is designed to give prospective foreign tourists a visually appealing representation of interesting and necessary tourist facilities. It serves two main purposes. First, it allows local businesses to advertise themselves and gain wider recognition through the Internet. Second of all, it allows a prospective tourist to be able to see key information concerning all of the necessary amenities of a vacation destination such lodging, food and entertainment opportunities.

## 2. Detailed Application Requirements

There are five different objects that can be marked as a “point of interest” on the island: Restaurants, general stores, hotels, transportation hubs and important landmarks. Restaurants have the following properties: open time (integer, army time), close time (integer, army time), name (string), address (string, street address). General stores have following properties: open time (integer, army time), close time (integer, army time), address (string, street address), supermarket (boolean: true if the store specializes in food, false if it doesn't). Hotels have the following properties: name (string), price per night (float, local currency), vacancy (boolean), smoking (boolean), street address (string). Transport hubs have the following properties: next trip time (integer, army time), next trip location (street address), price per ticket (float, local currency), street address (string), name (string).

There will need to be a DBA working closely with local businesses and government to make sure that values reflected in the database for businesses remain valid. In addition, street addresses will need to be verified upon the modification of a field containing an address using a third party map API to ensure that they are valid. When open/close times are modified, there will need to be validation ensuring that the new times are in fact valid times.