

Vlad Predovic

CS 340

10/6/2015

## Assignment 1

A. Ok

B.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'cs340\_predoviv'. The left sidebar shows the database structure with three tables: 'p', 's', and 'sp'. The main panel displays the structure of these tables. The 'p' table has 6 rows, 's' has 5 rows, and 'sp' has 12 rows. The total size of the database is 6.7 KiB.

Table	Action	Rows	Type	Collation	Size
p	Browse Structure Search Insert Empty Drop	6	MyISAM	latin1_swedish_ci	2.3 KiB
s	Browse Structure Search Insert Empty Drop	5	MyISAM	latin1_swedish_ci	2.2 KiB
sp	Browse Structure Search Insert Empty Drop	12	MyISAM	latin1_swedish_ci	2.2 KiB

3 tables Sum 23 MyISAM latin1\_swedish\_ci 6.7 KiB

The screenshot shows a SQL query result for the query 'SELECT \* FROM `p`'. The result is displayed as a table with columns 'sno', 'pno', and 'qty'. The table contains 12 rows of data.

sno	pno	qty
s1	p1	300
s1	p2	200
s1	p3	400
s1	p4	200
s1	p5	100
s1	p6	100
s2	p1	300
s2	p2	400
s3	p2	200
s4	p2	200
s4	p4	300
s4	p5	400

The screenshot shows a SQL query result for the query 'SELECT \* FROM `p`'. The result is displayed as a table with columns 'pno', 'pname', 'color', 'weight', and 'city'. The table contains 6 rows of data.

pno	pname	color	weight	city
p1	nut	red	12	London
p2	bolt	green	17	Paris
p3	screw	blue	17	Rome
p4	screw	red	14	London
p5	cam	blue	12	Paris
p6	cog	red	19	London

The screenshot shows a SQL query result for the query 'SELECT \* FROM `s`'. The result is displayed as a table with columns 'sno', 'sname', 'status', and 'city'. The table contains 5 rows of data.

sno	sname	status	city
s1	Smith	20	London
s2	Jones	10	Paris
s3	Blake	30	Paris
s4	Clark	20	London
s5	Adams	30	Athens

C. Why would you choose a database system instead of simply storing data in operating system files? When would it make sense *not* to use a database system?

For large amounts of data it is unwise to use main memory to store this, and in many cases you may not have enough main memory. Additionally, you cannot refer to more than 4GB of main memory at a time (ram?). Special programs would have to be written to answer questions about the data. Data would have to be restored and protected as well using operating system files. Database systems are used to make these tasks easier. DBMS's give you the advantages of giving you an abstract view of the data, efficient data access, data integrity, and the ability to administrate and provide data in a much easier fashion than file systems.

However, A DBMS is optimized for a certain type of work, and becomes cumbersome when being used for specialized applications like those with real-time constraints and when you have to write custom-code applications. Another possible drawback of using a DBMS is that you might need to manipulate the data in a way that the query language of the DBMS does not support.

D. Exercise 1.3 Explain the difference between logical and physical data independence.

Logical data independence is the ability to change the relationship structure between objects while still presenting the same results to a user query.

Physical data independence refers to hiding the details of how data is saved to memory while presenting the same conceptual schema to the user. However, physical changes can affect performance both positively and negatively.

E. Exercise 1.4

Explain the difference between external, internal, and conceptual schemas. How are these different schema layers related to the concepts of logical and physical data independence?

- **External Schema:** Allows data access to be customized at the user level. A database can have many external schemas with different access levels depending on who the users are. The ES is guided by end user requirements.

- **Internal Schema:** Specifies the storage the details. It tells us how the data from the Conceptual Schema is stored in physical memory (Ex: disk, tape).

- **Conceptual Schema** (logical): Stored data is described in terms of the data model with relationships between objects displayed.

F. Exercise 1.6

Scrooge McNugget will need a security facility since he does not want to share his list. Not having one is a risk that leaves him vulnerable to intruders. Since he is the only user, McNugget will not need concurrency control. Crash recovery will be necessary as without that he risks losing everything in case of a system failure. He will need a view mechanism in order to see the results of his queries. Without the view mechanism McNugget will have trouble deciphering his data. Finally, if using a DBMS McNugget will need a query language in order to manipulate all the data.

## G. Database serving a Spanish to English services company

- Employees: EID, Name, PW, phone number, admin(binary)
- Clients: CID, Client Name, password, Email, phone number, address
- Job: JID, Date started, job type, Client, Lead employee, comp date, status (bin), hours worked, File
- Service: SID, Date started, service type, Client, Lead, comp date, status (binary), hours worked
- Documentation(files): DID, Description, Type, Client, Date received

### Introduction:

The database project will support a company that provides services for bridging the gap between English and Spanish speakers. This could be something as simple as translating a document or personal like communicating between a sick patient and their doctor and even developing learning material for non-English speakers. The company provides both jobs and services to its clients, and needs a way to keep track of this. Additionally, long-term clients will have accounts so they can automatically request additional items. The company needs a way to keep track of the many documents being worked on, be able to add and remove users, and keep track of hours worked on a job for all the employees.

### Detailed Application Requirements:

The first implementation is a way for employees to see the work being done in table form. This will require some form of login/session-protected page with a table drawing joined values from the database. The table will portray jobs by creation date and have details like who is the 'lead' working the job, links to a file attached if applicable, and the status of the job/service to be completed. Additionally the owners of the company want to be able to adjust values themselves such as the time worked on the job, changing current parameters, and adding additional jobs. These user-oriented applications can be done through the implementation of form queries in php.

Another requirement that was decided upon with my sponsors was a way for clients to view the jobs they have requested, those they have completed, and be able to request more work. The issue here is that the company does not want to receive emails from regular clients each time a new job or service is needed. Regulars should have a way to submit work they want done and view their current history. Accounts for each long-term client will be added requiring that they be added to the database as a 'client' object. Once logged on a load query and some php code will bring up a table showing the client their history with a form option to add another request.

The company wants to be able to upload and keep track of information. This company will be dealing with documents a lot of the time due to the translating services that are provided. These need to be archived and connected to the rest of the database. The user will have a way to upload a file in relation to each job. Once this is complete, the document will automatically be related to a client, a work date, and a 'lead' employee through the implementation of the relational database. However, one of the limitations is the storage capacity of the servers. A

work-around in consideration is using an existing system such as the Google Drive or Microsoft's One Drive to upload files and then providing the URL link to these to be saved into the database. To users a file link would be displayed for the current job that would lead them to where the file that is being worked on is located.