

## ACS 1803 Supplementary Notes

### DATA MANAGEMENT IN A COMPUTER

One of the main uses of computers in running organizations is the ability to produce meaningful *reports* on different aspects of the organization. However, the reports are the “finished product”. The “raw material” for these reports is *data*, stored efficiently on a disk. Then specially written *programs* take this data from disk, put it in memory and process it to produce the reports that are desired.

When we start off computerizing our business, we need to do a thorough *data analysis*. We go to every person that will need any kind of output from the computer when doing his / her job. We ask this person what data items will be in the output he needs. We then write down each data item from each user on a piece of scrap paper.

For example, say in a school, we talked to five administrators and found that the following data items, in total, were needed in their outputs: Student Number, Student Name, Student Street address, Student City and Province, Student major area (only one), and for each course the student takes, the Course Number, Course Name and Grade that the student got. This therefore is the raw material that we will need to store efficiently on disk so as to get the various reports that the different administrators need.

We now learn how to organize real-world data so it could be efficiently stored on disk in a computer.

The smallest item of data is a **character**: e.g., A,b,9,%,\$,” “

But storing just characters alone in a computer won't do; we need to group them

A group of characters is a **field**: The field has a *name*, a *type* (how the computer will represent it internally) and a *length*. e.g., First Name, Text type, length 15 (incl. blanks). E.g., “John” is an example of this field.

But storing just fields alone in a computer won't do; we need to group them.

A group of fields, side-by-side, is known as a **record**.

A record needs to have a *layout* (structure, or design).

e.g.:

STUDNO	SNAME	STREET	CITY	MAJOR	CNO	CNAME	GRADE
Text	Text	Text	Text	Text	Text	Text	Text
4	25	25	20	15	4	20	2

Then there is the *data record itself*, that is, the real world data for one record that follows this layout

Data record:

1234          Lam A          491 Castle    Wpg    Economics    1803 Computers A

(some blanks have been omitted)

But storing just one record alone in a computer won't do; we need a group of several records that follow the same layout and are somehow related (say students of the same class)

A group of records that follow the same layout and that it makes sense to keep together is known as a **file** or **table**.

e.g., of a table

STUDNO	SNAME	STREET	CITY	MAJOR	CNO	CNAME	GRADE
1234	Lam A	491 Castle	Wpg	Economics	1803	Computers	A
9876	Kelly R	22 Miller Cr	Wpg	Biology	2002	Finance	B
4567	Ng J	399 High St	Wpg	Business	1304	Accounting	C

So, in order to store data on disk meaningfully, so that we can get a variety of useful information out, we see that we must set up our data, at least in a file.

We have also seen that

- storing just characters alone in a computer won't do; we need to group them
- storing just fields alone in a computer won't do; we need to group them.
- storing just one record alone in a computer won't do; we need a group of several records that follow the same layout, i.e. we need to store data in a file

But, we need to ask ourselves at this point

***Will storing records in one file do? Or, will we need to group the files?***

Consider the following situation:

STUDNO	SNAME	STREET	CITY	MAJOR	CNO	CNAME	GRADE
1234	Lam A	491 Castle	Wpg	Economics	1803	Computers	A
1234	Lam A	491 Castle	Wpg	Economics	1901	Pascal	B
1234	Lam A	491 Castle	Wpg	Economics	2002	Finance	A
9876	Kelly R	22 Miller Cr	Wpg	Biology	2002	Finance	B
9876	Kelly R	22 Miller Cr	Wpg	Biology	3421	Entomology	B
9876	Kelly R	22 Miller Cr	Wpg	Biology	4523	Zoology	C
4567	Ng J	399 High St	Wpg	Business	1304	Accounting	C
4567	Ng J	399 High St	Wpg	Business	2233	Marketing	A

{Take it on faith that when we set up a table on disk, every intersection of a row and a column must be filled-in [can't use dittos etc.]}

So, we have a problem

So, is this a good way to store the above data on disk? Why or why not?

Problem: repeating data: use storage and an update problem too

So we conclude that

Once we know which fields we need to store, putting all the fields in one record and then **storing all records in one file will not do; we need to use several connected files**

So we need to **break up the above file (table) into two tables** in such a way that we do not lose any information that is contained in the one table.

<<TRY THIS NOW.>>

In our one table above, we see that Sname, Street, City and Major are closely related (1-1) to Studno, since for each value of Studno, there will be only one value of these other fields; However, for each value of Studno, there may be more than one value of Cno, Cname, and Grade; so these fields will go into a separate table, since they have a one to many (1-m) relationship to student number.

So our data will now consist of two related tables, as below:

1. Name of the table : STUDENTS

<i>STUDNO</i>	<i>SNAME</i>	<i>STREET</i>	<i>CITY</i>	<i>MAJOR</i>
1234	Lam A	491 Castle	Wpg	Economics
9876	Kelly R	22 Miller Cr	Wpg	Biology
4567	Ng J	399 High St	Wpg	Business

and

2. Name of the table : GRADES

<i>STUDNO</i>	<i>CNO</i>	<i>CNAME</i>	<i>GRADE</i>
1234	1803	Computers	A
1234	1901	Pascal	B
1234	2002	Finance	A
9876	2002	Finance	B
9876	3421	Entomology	B
9876	4523	Zoology	C
4567	1304	Accounting	C
4567	2233	Marketing	A

What is the advantage of setting up our data on disk in this way instead of having one table, as above?

Remember, what we are doing is not about output; it is about efficient storage of data on disk, so that after this the computer could produce many kinds of desired output.

Question: If the computer is asked to provide as output, a list that contains fields from both tables (e.g., SNAME, MAJOR, CNAME, GRADE), could the computer do it, if the data were set up on disk as above?

Yes, because the data in both tables is linked by a common column, in this case STUDNO.

When we have two or more tables on disk, that are related by a common column, the collection of (container for) these tables is called a **database**, more specifically, a relational database. The database has its own name (say, SCHOOL).

In this design, STUDNO is the common column linking both tables.

STUDNO is also a primary key in the STUDENTS table, since every record in that table will have a unique value of STUDNO.

Primary key – a field such that when we have its value, we can retrieve only that one record (when we know 1234 is Studno, then we can retrieve the entire record for this student and only his record)

In the GRADES table, STUDNO is a foreign key; it is in the GRADES table to link data from STUDENTS with data in GRADES.

{The common column, in this case STUDNO is a primary key in one table and a foreign key in the other table}

In the GRADES table, we have a compound (concatenated) primary key, consisting of the combination of **STUDNO and CNO**. {Do you see why we need a compound key in this case? Because a combination of Studno and Cno will be unique for each record in the GRADES table}

### Determining the layout for each table

Now we know we will need two separate tables, STUDENTS and GRADES in our database. Next, we have to determine the layout or structure of each table, before we will be able to implement this database in a computer.

Structure of the STUDENTS table:

<u>Field Name</u>	<u>Data Type</u>	<u>Length</u>
STUDNO	Text	4
SNAME	Text	25
STREET	Text	25
CITY	Text	20
MAJOR	Text	15

The primary key is STUDNO

Structure of the GRADES table:

<u>Field Name</u>	<u>Data Type</u>	<u>Length</u>
STUDNO	Text	4
CNO	Text	4
CNAME	Text	20
GRADE	Text	2

The primary key is STUNDO-CNO (compound key)

{Could also have

BDATE	Date / Time	
GPA	Numeric	4.2 (4 chars; 2 after dec pt. Eg. 3.82)
FOREIGN	Logical	1 }

---

## THE DATABASE MANAGEMENT SYSTEM (dbms)

With all this planning, we know we want to set up our data on disk in a container (database) called SCHOOL which contains two related tables, STUDENTS and GRADES.

But, what kind of software will do this for us? It will be a **database management system**, that is, software that 'manages' a database i.e., sets up the container, then the structure (skeleton) for each table, fills each table structure with data and then *produces a variety of output from this data, on screen or paper, without much programming.*

**MS Access is such a dbms.**

We have now done the design work that is necessary and we are now ready to use Access to implement the database as above.

In using Access, we have two main steps:

1. Setting up the database

- create a database (the 'container')
- create each table
  - provide the layout
  - identify the primary key
- link the tables
  
- enter the data for each table

## 2. Using the database

- **queries** quick questions, but with primitive format, more often on screen
- **reports** – more formal, with headings etc. can be on paper more often

## CUSTOMIZED INFORMATION SYSTEMS

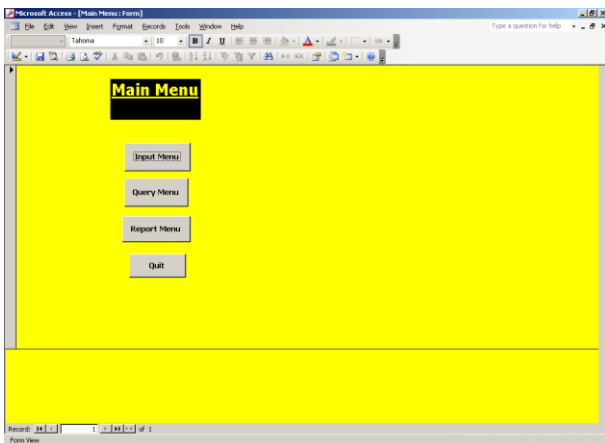
We can use Access at level 1 – just as above, that is we can use it **as a dbms**, where we have to know the features for creating a database, tables, and designing queries and reports.

However, we can also use MS Access **as an information system developer**. We use it to build a customized information system for some specific purpose (e.g., managing a golf course). The system will have up-front: its own menus, input screens, output (query) screens, and reports and, in the background, it will have a database with related tables, and programs that take the raw data from the database and convert it to the required queries and reports; the system will also have controls, that is, mechanisms that try to ensure that the output is correct and that the data is safe from accidental or deliberate destruction.

<<Example of building parts of a system - handout>>

Then <<demonstration of a working system built in Access>>

Note that a person building such a system does not only have to know how to use Access at Level 1 but also how to use it at Level 2. however, the *user* of such a system does not even realize that Access is in the background. S/he knows only how to use the menus and interfaces of the customized system itself. However, note that each such customized system uses a database (through a database management system) in the background, as shown in the diagram below:



Soon, we will be learning about different kinds of information systems that can assist in running organizations.

\*\*\*\*\*