

Question1

The Student_ID and Module Code fields follow no numeric order which is generally confusing to look at. Following a pattern e.g 1000001, 1000002 would be more logical.

The following three fields contain the similar flaw of too much data per record. Too much data per record increases the chances of data redundancy creating a problematic database.

The Student_Name field is cluttered and should be divided into two fields Student_First_Name and Student_Last_Name. The Student_Address field like the Student_Name field is cluttered with data and should be divided into three fields Number/Street/Address.

The records in the DOB field also contain too much data limiting the information that can be derived from the database. As mentioned above splitting the data in these records into more fields will alleviate this problem. E.g Day/Month/Year creating three fields allowing one piece of data per field.

There are repeated records within the next couple of fields causing data redundancy to occur. Data redundancy occurs in the Course_Code field from the repeated data anomaly 'KSOFT'. Data redundancy also occurs in the field Course_Name from the data anomaly 'B.Sc. Level 7 in SW development' Finally the field Course_Year contains data redundancy from the repeated data anomaly '1'.

There is one more issue I can see and that is with the Average_Result field. The field presents a lack of options to extract detailed information. This field will only give the average percentage of the overall course, basically the average of five modules per student per year. In the context of this assignment the main issue with the lack of information is a lecturer cannot search results per module. If a student has failed a particular module the lecturer cannot access this information from the current state of the field.

The problems listed above could have occurred through bad database design and or communication between the database designer and end user. A lot of these problems can be avoided if the initial planning and design stage is executed professionally.

Question 3

I have uploaded a PDF document to dropbox I created using the ERD modeling tool from the link on My Moodle page. My thoughts behind this is to give my following answer a little more clarity. In the PDF document the primary keys are in yellow and foreign keys are in black. The Student_ID field also works as a foreign key in later tables so I highlighted this in blue.

I have attempted to keep my table relationships to One to Many or One to One. I will list them here in text format and this combined with the PDF will give a clear picture of the table relationships I have attempted to implement. In the following text the One to One relationships are in **blue** and the One to Many are in **red**.

Student ID /Student First Name /Student Last Name/ Student Address Number/ SA Street,Town/SA County/Student DOB Day/ SDOB Month/ SDOB Year

Course Code /Course Name

Course Code (M) / Module Code /Module Name

KSOFT(M) Y1	“	/	“
KSOFT(M) Y2	“	/	“
KSOFT(M) Y3	“	/	“
HDIP(M) Y1	“	/	“
HDIP(M) Y2	“	/	“
HDIP(M) Y3	“	/	“

Course Code (SR) Year (SR) Student ID /Module Code (SR) X5 fields/ Average Result

KSOFT (SR)	Y1 KSOFT (SR)
	Y2 KSOFT (SR)
	Y3 KSOFT (SR)

HDIP (SR)	Y1 HDIP (SR)
	Y2 HDIP (SR)
	Y3 HDIP (SR)

Course Code (SA1)	Year (SA1)	Student ID / Module Code (SA1) X5 fields/ Avg. Attendance
KSOF (SA1)	Y1 KSOF (SA1)	
	Y2 KSOF (SA1)	
	Y3 KSOF (SA1)	
HDIP (SA1)	Y1 HDIP (SA1)	
	Y2 HDIP (SA1)	
	Y3 HDIP (SA1)	

I couldn't think of a better to write this in a word document so hopefully this will be clear with the PDF document.

Question 4

For my solution I started by looking at the fields to see if I could group them into smaller more relevant tables. I looked at the title for each field and from this I came up with the following tables.

Table 1

Student ID / Student Name / Student Address / Student DOB

Table 2

Course Code / Course Name / Course Year

Table 3

Module Code / Module Name

Table 4

Attendance / Avg. Results

In the next stage I looked at each field to see if the data contained in the corresponding records could be adjusted to make the database more coherent. At this stage I also looked to see if any data redundancy was occurring in the current fields. I decided to further split up some of the current fields to allow for more information to be gathered from the database. I came up with the following fields.

Table 1

Student ID /Student First Name /Student Last Name/ Student Address Number/ SA Street,Town/SA County/Student DOB Day/ SDOB Month/ SDOB Year

Table 2

Course Code /Course Name

At this stage I started to add the information for the Module_Codes and Student_ID's etc so I could see the data and decide how to group it. I titled table 3 as Course_Code(M) M standing for module. I displayed the years for each course under this and beside these fields the Module_Code and Module_Name fields.

Table 3

Course Code(M) / Module Code / Module Name

KSOF(M)Y1

KSOF(M)Y2

KSOF(M)Y3

HDIP(M)Y1

HDIP(M)Y2

HDIP(M)Y3

For the next stage I thought about how I could break up the average attendance and average results to allow the end user to gather more specific information. I came up with the idea that both of the tables could follow the same structure. The first table is labeled Results Course_Code(SR) SR standing for Student Results, below this I have displayed KOFT(SR) and HDIP(SR). The next field contains the individual years for each subject e.g Y1 KSOF(TSR). Each year then links to the Student_ID field and each Student_ID then links to the corresponding 5 modules for that year. The modules are all in separate fields. The final field in this table is the Avg._Result and the data contained in the records is derived from an = Average calculation. The =Average calculation takes the entries from the preceding module fields and displays the average. The student attendance table works the same way and allows the end user to search for attendance per module and gives out the average attendance.

Basically the end user can now search for attendance and results per module. This in turn allows the end user to see if a student is excelling or falling behind in various modules.

Table 4

Results

<u>Course Code(SR) / Year (SR) / Student ID /MC 1/ MC 2/MC 3/MC 4/MC 5/ Avg. Result</u>	
KSOFT(SR)	Y1 KSOFT(SR) Y2 KSOFT(SR) Y3 KSOFT(SR)
HDIP(SR)	Y1 HDIP(SR) Y2 HDIP(SR) Y1 HDIP(SR)

Table 5

Attendance

<u>Course Code(SA1) / Year (SA1) /Student ID /MC 1/ MC 2/MC 3/MC 4/MC 5/ Avg. Attendance</u>	
KSOFT(SA1)	Y1 KSOFT(SA1) Y2 KSOFT(SA1) Y3 KSOFT(SA1)
HDIP(SA1)	Y1 HDIP(SA1) Y2 HDIP(SA1) Y1 HDIP(SA1)

The final stage was to look at the record detail of the Student_ID and Module_Code fields and see if I could make some numeric patterns from the current data. I basically put them in order as follows

Student_ID	Student_First Name	KSOFT(M) Y1	
S00000001	Paul		COMP1234
S00000002	Edgar		COMP1345
S00000003	Stephen		COMP1456
S00000004	Scott		COMP1567
S00000005	Sinead		COMP1678
S00000006	Gerard	KSOFT(M) Y2	COMP2173
			COMP2174
			COMP2175
			COMP2176
			COMP2177
		KSOFT(M) Y3	COMP3001
			COMP3002
			COMP3003
			COMP3004
			COMP3005

The Student_ID has an obvious pattern and the new data in the Module_Codes is based off the original Module_Codes.

In summary I found this challenging but I think I have made the data in the database more accessible for the end user. From what I have learnt to date I'm confident that I have used the relevant tools to find a solution for the database. When making databases in the future I will definitely use the ERD modeling tool before I start entering data.