Lecture 4.2b. In-lecture exercise. Developing a conceptual data schema
Two extensive worked-out examples

Note: In neither example did I go into how the entity values would be identified. For most entity types this is fairly obvious.

Developing a conceptual data schema. Example 1. Information system for a computer users group

Step 1: From anticipated questions to conceptual data schema raw material collection

The purpose of this step to derive from anticipated questions a first rough list of entity types and relationship types that serves as the raw material for Step 2,

Column 1 lists questions from students and the students' analysis of entity types and relationship types, except that I omitted the entity types by themselves – they can be seen from the relationship types.

Entity types and relationship types not already listed for a previous question are underlined. For each question, the new entity types and relationship types are copied to column 2 and 3, respectively. If column 2 is blank, all the entity types needed for the questions are already listed earlier in column 2, and analogously for column 3. columns to give a very rough draft of the conceptual data schema, the material from which the final schema can be built. This is a first step of distillation.

Questions and E-R analysis from students	Entity types	Relationship types
Who knows about printers? Person <*knowsAbout*> Subject	Person Subject	Person <*knowsAbout*> Subject
I am looking for a review of Microsoft Word 12 Document <*reviews*> SoftwareMake&Model	Document SoftwareMake&Model	Document <*reviews*> SoftwareMake&Model
What is a good word processor for Red Hat Linux SoftwareMake&Model <*servesFunction*> Function SoftwareMake&Model <*worksWith*> SoftwareMake&Model SoftwareMake&Model <*hasQuality*> QualityScore SoftwareMake&Model <*hasAttribute*> Quality	Function QualityScore	SoftwareMake&Model <*servesFunction*> Function SoftwareMake&Model <*worksWith*> SoftwareMake&Model SoftwareMake&Model <*hasQuality*> QualityScore SoftwareMake&Model <*hasAttribute*> (Quality, QualityScore) ▸Note: <hasQuality> is more specific than <*hasAttribute*>, may not be needed
Where will the Red Hat Linux group meet? ▸Note: Question is underspecified. Should be When and where UserGroup <*meetsIn*> (DateTime, Location UserGroup <*takesplacein*> (DateTime, Location)	UserGroup DateTime Location	**UserGrou**p <*meetsIn*> (DateTime, Location UserGroup <*takesplacein*> (DateTime, Location) ▸Note: May not need both relationships
When is the meeting for the Red Hat Linux user group? MeetingDate, UserGroup, SoftwareMake&Model UserGroup <*dealsWith*> SoftwareMake&Model UserGroup <*meetsOn*> MeetingDate ▸Note: MeetingDate is not a good entity type because it includes a role, date of a meeting. DateTime is a good entity type (plain date is DateTime given with low precision).		

Questions and E-R analysis from students	Entity types	Relationship types
When does the IT Club meet? Club <*meetsOn*> MeetingDate ▸Note: Same as previous.		
When does the next computer group meeting take place and whom can I call for more information? user group <*meetsOn*> meeting date person <*hasKnowledgeOf*> subject person <*contactBy*> phone number	PhoneNumber	Person <*contactBy*> PhoneNumber
Who will write the newsletter? Newsletter <*authoredBy*> Person Person <*isAuthorOf*> Newsletter	Newsletter	Newsletter <*authoredBy*> Person Person <*isAuthorOf*> Newsletter ▸Note: <isAuthorOf> is the reciprocal of <*authoredBy*> *;* need only one
What is the phone number to get help with word processing? Person <*knowsAbout*> Subject Person <*contactBy*> PhoneNumber		
Who wrote the product review on MS Word? Document <*reviews*> SoftwareMake&Model Document <*authoredBy*> Person		
Where can I get the best price for the printer I selected? HardwareMake&Model <*hasPrice*> (Organization, MoneyAmount)	Organization MoneyAmount	HardwareMake&Model <*hasPrice*> (Organization, MoneyNumber)
What is the phone number of the company I found? Organization <*hasPhoneNumber*> PhoneNumber		Organization <*hasPhoneNumber*> PhoneNumber
Is there a fee to join the club? ▸Note: **Not a good question to ask of the club's database**. Look in the club's bylaws		
Who is in the computer club? Club <*hasMember*> Person	Club	Club <*hasMember*> Person

Questions and E-R analysis from students	Entity types	Relationship types
How can members be contacted? Person <*isMemberof*> Club Person <*hasEmail*> Contact	EmailAddress ▸Note: Contact is not a good entity type	Person <isMemberof> Club ▸Note: <hasMember> is reciprocal of <*isMemberof*>; only one is needed. Person <hasEmail> EmailAddress
What articles have been published by members? Person <*has Published*> Document ▸Note: Means <*has Authored*>, which is the same as <*AuthorOf*> Document <*authoredBy*> Person		
What graphics cards are available for the Lenovo W510? HardwareMake&Model <*worksWith*> HardwareMake&Model	HardwareMake&Model	HardwareMake&Model <*worksWith*> HardwareMake&Model
Does SimCity work with the Invidia 700 graphics card? SoftwareMake&Model <*worksWith*> HardwareMake&Model		SoftwareMake&Model <*worksWith*> HardwareMake&Model
What member can set up a computer network? Person <*hasKnowledge*> Subject ▸Note: Already covered as <*knowsAbout*>		
What subjects are the experts knowledgeable in? Person <*hasKnowledgeOf*> Subject ▸Note: Curiously phrased. Should say What subject is expert A knowledgeable in? OR Who knows about a given subject?		
I'm searching for an article about creating websites. Document <*dealsWith*> Subject		Document <*dealsWith*> Subject
Who is the creator of the Website? Person <*creatorOf*> Object ▸Note: Object seems too general for Website. Should use Website for now and think about this when reorganizing the initial schema.	Website	Person <*creatorOf*> Website

Questions and E-R analysis from students	Entity types	Relationship types
Where is the Chairpersons office? Person <*locatedAt*> Location ▸Note: How do you know who is the chair person. Need Person <*servesInRole*> (PersonRole, Organization) A role is always in the context of an Organization, so this needs a three-way relationship.	PersonRole	Person <*servesInRole*> (PersonRole, Organization) Person <*locatedAt*> Location
Who is the contact person for the group purchasing program? user group <*carriesOut*> program person <*affiliatedWith*> program ▸Note: <*servesInRole*> is better here; the role is contact person. person <*contactBy*> phone number	Program	UserGroup <carriesOut> Program
Who can fix my computer, how do I contact them? Person <knowsAbout> Subject Person <contactBy> Phone Number ▸Note: This could also be an organization, such as a computer store or computer repair company, so we also need Organization <*knowsAbout*> Subject Organization <*contactBy*> Phone Number		Organization <*knowsAbout*> Subject Organization <*contactBy*> Phone Number
Who services the printer and how can we contact for service? ▸Note: What printer? HardwareMake&Model <*servesFunction*> Function Company <*knowsAbout*> Subject Company <*contactBy*> Email ▸Note: Organization is the better entity type. Also need generic entity types ServiceType and ServiceEvent Organization <*performs*> (ServiceType, HardwareMake&Model, MoneyAmount)	ServiceType ServiceEvent	HardwareMake&Model <*servesFunction*> Function
What is the best printer for printing photos? Hardware <*bestFor*> Function ▸Note: Instead use HardwareMake&Model <*hasQuality*> (Function, QualityScore), an extension of the two-way relationship <*hasQuality*>; the system can then find the HardwareMake&Model with the best score.		HardwareMake&Model <*hasQuality*> (Function, QualityScore>)

Questions and E-R analysis from students	Entity types	Relationship types
How much will repair cost? Repair <*hasFeeof*> Cost ▸Note: What repair? What would a statement in the database look like? What are the values of the entity type Repair? Organization <*performs*> (ServiceType, HardwareMake&Model, MoneyAmount) where ServiceType would be Repair (as opposed to, for example, maintenance); in real life it would be much more specific: the specific repair for this specific printer model.		Organization <*performs*> (ServiceType, HardwareMake&Model, MoneyAmount)
Where do I find an article about fixing a computer? Document <*dealsWithSubject*> Subject		
What is the best anti-virus program for a Windows PC? Software <*doesFunction*> Function Software <*worksWith*> Hardware Software <*bestQuality*> Quality ▸Note: It needs to be SoftwareMake&Model. Also, <*worksWith*> needs to be with the operating system, not the hardware. <*bestQuality*> treated as in the previous question		
What is the phone number to get help with word processing? Person <knowsAbout> Subject Person <contactBy> PhoneNumber		
Whom do I contact for IT support? Person <*hasKnowledgeOf*> Subject Person <*contactBy*> Phone number		
What is the phone number for canceled meetings? Person, PhoneNumber, Subject Person <*knowsAbout*> Subject Person <*contactBy*> PhoneNumber		

Example 1. Computer users group**. Step 2: Refining the conceptual data schema**

This step starts from the raw entity types and relationship types that were distilled from the original questions; there is no need to look at the many original questions any more.

In this step, I first arranged the draft relationship types into groups that address similar types of data. The groups are indicated by color in the draft **Relationship types, raw** column. In the **Scratchpad** column, relationship types are rearranged to show all relationship types in a group together. Now one can see relationship types that are the same or almost the same and can be consolidated. One also sees patterns that suggest how the conceptual data schema can be simplified (see the notes after the table). There are many other types of editing. The results are seen in the column **Relationship types, close to final**. I first extracted the purple group and dealt with it, then the red group, etc. The audio gives more explanation..

The **Scratchpad** column and the **Relationship types, final** are arranged in correspondence. Relationship types marked with strike through have been consolidated; they are encompassed in a one of the final relationship types. A final relationship type is next to an empty line in the scratchpad column has been added to make the schema more complete.

HardwareMake&Model is an entity type for such values as Lenovo (Make) W510 (Model). When necessary, it is abbreviated as HardwareM&M. Same for SoftwareMake&Model and ProductMake&Model

Relationship types related to **Person and Organization, Hardware and Software, Document, Organization activities and membership**

Raw list of entity types and relationship types		**Scratchpad**	**Final conceptual data schema**	
Entity types, raw	**Relationship types, raw**	**Relationship types, rearranged**	**Relationship types, final**	**Entity types, final**
Person Subject Document SoftwareMake&Model Function QualityScore UserGroup DateTime Location Newsletter PhoneNumber Organization MoneyNumber Club EmailAddress HardwareMake&Model Website PersonRole Organization Program ServiceType ServiceEvent	Person <*knowsAbout*> Subject Document <*reviews*> SoftwareMake&Model SoftwareMake&Model <*servesFunction*> Function SoftwareMake&Model <*worksWith*> SoftwareMake&Model SoftwareMake&Model <*hasQuality*> QualityScore SoftwareMake&Model <*hasAttribute*> (Quality, QualityScore) ▸Note: <hasQuality> is more specific than <*hasAttribute*> and may not be needed UserGroup <*meetsIn*> (DateTime, Location UserGroup <*takesplacein*> (DateTime, Location) ▸Note: May not need both relationships Newsletter <*authoredBy*> Person Person <*isAuthorOf*> Newsletter ▸Note: <isAuthorOf> is the reciprocal of <*authoredBy*> ; need only one Person <*contactBy*> PhoneNumber HardwareMake&Model <*hasPrice*> (Organization, MoneyNumber) Organization <*hasPhoneNumber*> PhoneNumber Club <*hasMember*> Person Person <isMemberof> Club ▸Note: <hasMember> is reciprocal of <*isMemberof*>; only one is needed. Person <hasEmail> Contact ▸Note: Contact is a poor entity type; what would be its values? Should be EmailAddress HardwareMake&Model <*worksWith*>	**Purple** Person <*knowsAbout*> Subject Person <isC*reatorOf*> Entity Person <*isAuthorOf*> Document ~~Person <*creatorOf*> Website~~ ~~Person <*contactBy*> PhoneNumber~~ ~~Organization <*contactBy*> Phone Number~~ Person <*hasPhoneNumber*> PhoneNumber Organization <*hasPhoneNumber*> PhoneNumber ~~Person <*hasEmail*> Contact~~ Person <*hasEmail*> EmailAddress Organization <*knowsAbout*> Subject **Red** SoftwareMake&Model <*servesFunction*> Function HardwareMake&Model <*servesFunction*> Function SoftwareMake&Model <*worksWith*> SoftwareMake&Model HardwareMake&Model <*worksWith*> HardwareM&M SoftwareMake&Model <*worksWith*> HardwareM&M SoftwareMake&Model <*hasQuality*> QualityScore SoftwareMake&Model <*hasAttribute*> (Quality, QualityScore) ▸Note: <hasQuality> is more specific than <*hasAttribute*> and may not be needed HardwareMake&Model <*hasQuality*> (Function,	LegalEntity <*knowsAbout*> Subject LegalEntity <isC*reatorOf*> Entity LegalEntity <*isAuthorOf*> Document LegalEntity <*hasPhoneNumber*> PhoneNumber LegalEntity <*hasEmailAddress*> *EmailAddress* ProductMake&Model <*servesFunction*> Function ProductMake&Model <*worksWith*> ProductM&M ProductMake&Model <*hasAttribute*> (Attribute, Function, AttributeMeasurement)	LegalEntity Person Organization Subject Document DocumentM&M DocumentType PhoneNumber EmailAddress ProductMake&Model SoftwareM&M HardwareM&M DocumentM&M Function Attribute AttributeMeasurement MoneyAmount ServiceType ServiceEvent Event EventType DateTime Location LegalEntityRole Program

	HardwareMake&Model SoftwareMake&Model <*worksWith*> HardwareMake&Model Document <*dealsWith*> Subject Person <*creatorOf*> Website Person <*servesInRole*> (PersonRole, Organization) Person <*locatedAt*> Location UserGroup <carriesOut> Program Organization <*knowsAbout*> Subject Organization <*contactBy*> Phone Number HardwareMake&Model <*servesFunction*> Function HardwareMake&Model <*hasQuality*> (Function, QualityScore> Organization <*performs*> (ServiceType, HardwareMake&Model, MoneyAmount)	QualityScore> HardwareM&M <*hasPrice*> (Organization, MoneyNumber) Organization <*performs*> (ServiceType, HardwareMake&Model, MoneyAmount) **Blue** Document <*reviews*> SoftwareMake&Model Document <*dealsWith*> Subject **Green** UserGroup <*meetsIn*> (DateTime, Location UserGroup <*takesplacein*> (DateTime, Location) ▸Note: May not need both relationships Club <*hasMember*> Person Person <isMemberof> Club ▸Note: <hasMember> is reciprocal of <*isMemberof*>; only one is needed. Person <*servesInRole*> (PersonRole, Organization) Person <*locatedAt*> Location UserGroup <carriesOut> Program	ProductM&M <*hasPrice*> (Organization, MoneyAmount) LegalEntity <*performs*> (ServiceType, ProductM&M, MoneyAmount) Document <*reviews*> Entity Document <*dealsWith*> Subject (LegalEntity-1, LE-2, LE-3, . . .) <*participateIn*> Event Event <*isa*> EventType Event <*takesPlaceIn*> (DateTime, Location) LegalEntity <*isMemberof*> Organization LegalEntity <*servesInRole*> LegalEntityRole, LegalEntity) LegalEntity <*locatedAt*> Location LegalEntity <carriesOut> Program	

Student-submitted questions with entity-relationship analysis

What to do and what not to do. Common mistakes to avoid. (read)

These questions are for different kinds of systems, not just computer users group or school

Note: **Entity types should be defined based on <u>intrinsic characteristics,</u>**
<u>not the role the entity plays in some context</u>

Question. I need to find information on nursing home ratings to determine which place is best for my mother to live.

Entity Types:
Document
Subject
Rating
Accrediting Bodies Not an entity type, but a role. Entity type is Organization
Location

Relationship Types:
Document <*dealsWith*> Subject
Subject <*includes*> Nursing Home
Rating <*determinedBy*> Accrediting Body
Location <isNear> Home

Your relationship types are problematic. Try to write sample statements. For example, what Rating? Whose Home?

Better relationship types

(1) Organization <*locatedAt*> Location

(2) Organization <*isa*> OrganizationType (Among the values of OrganizationType is NursingHome)

(3) Organization <*hasRating*> (Rating, Organization)

(4) Organization <*isRatingAuthorityFor*> OrganizationType)
If the system knows the location of your home, it can use (1) to find organizations close to your home and (2) to limit the list to nursing homes. Then it can use (3) to find ratings of these organizations and (4) to check whether the rating comes from an organization that is an authority for rating nursing homes.

Student-submitted questions with entity-relationship analysis

Question: **I am looking for information on different employees that work within a company.**

Entity Types:

Employee
Not an entity type, but a role in a relationship. The entity type is Person, relationship type Person <employedBy> Organization
Supervisor same comment
Wage not an entity type, entity type is MoneyAmount, Person <isPaid> MoneyAmount
Job Title
Also not an entity type. The entity type is Text: Job <*hasTitle*> Text
Job A specific job in the organization, usually identified by some number in the human resources (HR) system. Often there are many jobs with the same title, such as Store Manager.
Insurance Policy (identified by organization and policy number)

Relationship types

Employee <works for> Supervisor Person <*worksFor*> Person
Employee <earns> Wage Person <*earns*> MoneyAmount

Employee <*has*> Job Title Employee <*haJob*> Job, Job <*hasTitle*> Text

Employee <*hasInsurancePolicy*> InsurancePolicy

The following two questions are fine for the **movie company** example

Student-submitted questions with entity-relationship analysis

Question: I need to find which director(s) is attached to drama projects with R rating.

Entity Types:

Director — Not an entity type but a role
Producer — ditto
ProjectTitle — The entity types are Project and Text. Project <hasTitle> Text
Location — Needed for question #2, not for question #1
Genre
MovieRating

Relationship types:

Director <assigned> ProjectTitle — Person <*isDirectorOf*> Project
ProjectTitle < assigned > Genre — Project <*hasGenre*> Genre
ProjectTitle < assigned > MovieRating — Project <*hasRating*> MovieRating

All these relationships are with the Project, not the Text that is the title of the Project.

Question: I need to find which projects are shot in NY location with a Producer Joe Smith.

Entity Types:

Director
Producer
ProjectTitle
Location
Genre
MovieRating

Relationship types:

Location< hasAttached > ProjectTitle — Project <*isFilmedIn*> Location
ProjectTitle <assigned> Producer — Person <*isProducerOf*> Project

Developing a conceptual data schema. Example 2. Integrated information system for a school

The close-to-final version of this conceptual data schema contains a number of ideas that would be useful in designing an actual system.

Step 1: From anticipated questions to conceptual data schema raw material collection

The purpose of this step to derive from anticipated questions a first rough list of entity types and relationship types that serves as the raw material for Step 2,

Column 1 lists questions from students and the students' analysis of entity types and relationship types, except that I omitted the entity types by themselves – they can be seen from the relationship types.

Entity types and relationship types not already listed for a previous question are underlined. For each question, the new entity types and relationship types are copied to column 2 and 3, respectively. If column 2 is blank, all the entity types needed for the questions are already listed earlier in column 2, and analogously for column 3. Columns 2 an 3 give a very rough draft of the conceptual data schema, the material from which the final schema can be built. This is a first step of distillation.

Questions and E-R analysis from students	Entity types	Relationship types
I need to gather books on the 1920's for an 8th grade social studies project. Document <dealsWithSubject> Subject Subject <suitableFor> GradeLevel Document <suitableFor> GradeLevel	Document Subject GradeLevel	Document <dealsWithSubject> Subject Subject <suitableFor> GradeLevel Document <suitableFor> GradeLevel
My student has failing grades in my history course for the subject of WW1. I would like to find supplemental documents to help this student understand the subject. Course <*hasInstructor*> Person Good for the overall system but not needed for this question Course should be CourseOffering Course <*hasStudent*> Person " (same comment) Person <*has Attribute*> Grade " The relationship is incomplete, a grade belongs with a course offering Document <*dealsWithSubject*> HistorySubject Should be just Subject Document <*suitableFor*> Course	Course CourseOffering Grade	Person <*has Attribute*> Grade Document <*suitableFor*> Course
I need to find nonfiction resource books on the atomic bomb to assist students in their research debate for global history. Document <*dealsWithSubject*> Subject Subject <*suitableFor*> Grade Level Document <*suitableFor*> Grade Level Document <*belongsTo*> DocumentGenre	DocumentGenre	Document <belongsTo> DocumentGenre

Questions and E-R analysis from students	Entity types	Relationship types
A young girl, who loves cooking, is looking for a cookbook regarding foods from around the world to practice her culinary skills while prepping for a home and careers test. Document <dealswithSubject> HomeAndCareersSubject Just Subject Person <hasInterest> Cooking should be Subject Document <suitableFor> Course Document <helpfulFor> CourseExam A problematic entity type. Improve	CourseExam	Person <hasInterest> Subject Document <helpfulFor> CourseExam
I am looking for materials for high school students creating a presentation to teach their health class about eating disorders. *EducationObject <addressesCourse> CurriculumCourse Better: LearningObject. Just Course *EducationObject <addressesSubject> CourseSubject Just Subject *EducationObject <belongsTo> EducationObjectType *EducationObject <isComponentOf> EducationObject *EducationObject <containsReadingLevel> AgeLevel Perhaps better: ReadingLevel Good relationship types, in the final conceptual data schema, these appear with names edited	LearningObjec LearningObjectType AgeLevel ReadingLevel	LearningObject <addressesCourse> Course LearningObject <addressesSubject> Subject LearningObject <belongsTo> LearingObjectType LearningObjec <isComponentOf> LearningObjec LearningObjec <containsReadingLevel> AgeLevel
The next two questions are from the Lecture Notes 4.2b		
I am concerned about this girl who is the victim of bullying. Could you recommend a fiction book she could read to help her to stand up for herself. Person *<hasCondition>* Condition Document <*usefulFor*> Condition Document <*belongsToGenre*> DocumentGenre	Person Condition For example, being bullied	Person *<hasCondition>* Condition Document <*usefulFor*> Condition
I am looking for a group activity in a lesson on the dangers and consequences of stereotyping / labeling / name calling in sixth grade LearningObject <dealsWithSubject> Subject, LearningObject <suitableFor> GradeLevel LearningObject <belongsToType> LearningObjectType LearningObject <isPartOf> LearningObject LearningObject <dealsWithStandard> CurriculumStandard LearningObject <hasQuality> Quality (could set values as 1 – 5 stars).	LearningObject LearningObjectType CurriculumStandard Quality	LearningObject <dealsWithSubject> Subject, LearningObject <suitableFor> GradeLevel LearningObject <belongsToType> EducationObjectType LearningObject <isPartOf> LearningObject LearningObject <dealsWithStandard> CurriculumStandard LearningObject <hasQuality> Quality
I added the next two to go beyond just the school library		

Questions and E-R analysis from students	Entity types	Relationship types
For a learning activity in a specific course offering the system should schedule and deliver necessary equipment, library materials, etc.; solicit parental permission where needed; and do anything else required. LearningActivityInstance <*isa*> LearningActivityType LearningActivityInstance <*isScheduledFor*> (Date&Time, Date&Time, Place) Note: a LearningActivityInstance may be scheduled for several time intervals (given as beginning Date&Time and ending Date&Time at different places (for example, a field trip). LearningActivityInstance <*isPartOf*> CourseOffering CourseOffering <*hasStudent*> Person CourseOffering <*hasInstructor*> Person Person <*hasGuardian*> Person LearningActivityType <*requires*> EquipmentOrMaterial	LearningActivityInstance LearningActivityType Date&Time Place EquipmentOrMaterial	LearningActivityInstance <*isa*> LearningActivityType LearningActivityInstance <*isScheduledFor*> (Date&Time, Date&Time, Place) LearningActivityInstance <*isPartOf*> CourseOffering CourseOffering <*hasStudent*> Person CourseOffering <*hasInstructor*> Person Person <*hasGuardian*> Person LearningActivityType <*requires*> EquipmentOrMaterial
For all students, prepare a report of learning objectives (expressed in terms of curriculum standards) the student has achieved so far in the present school year, with degree of achievement given as a letter grade. Assignment <isEvidenceFor> LearningOutcome TestItem <isEvidenceFor> LearningOutcome Person <completed> (Assignment, Date, Grade) Person <completed> (TestItem, Date, Grade)	Assignment LearningOutcome TestItem	Assignment <isEvidenceFor> LearningOutcome TestItem <isEvidenceFor> LearningOutcome Person <completed> (Assignment, Date, Grade) Person <completed> (TestItem, Date, Grade)
For each student who shows deficiencies the system should put together a set of learning objects that would help the student overcome the deficiency. Person <achieved> (LearningOutcome, Grade) LearningObject <supports> LearningOutcome LearningObject <hasPrerequisite> LearningOutcome		Person <achieved> (LearningOutcome, Grade) LearningObject <supports> LearningOutcome LearningObject <hasPrerequisite> LearningOutcome

Example 2. Integrated information system for a school**. Step 2: Refining the conceptual data schema**

This step starts from the raw entity types and relationship types that were distilled from the original questions; there is no need to look at the many original questions any more.

In this step, I first arranged the draft relationship types into groups that address similar types of data. The groups are indicated by color in the draft **Relationship types, raw** column. In the **Scratchpad** column, relationship types are rearranged to show all relationship types in a group together. Now one can see relationship types that are the same or almost the same and can be consolidated. One also sees patterns that suggest how the conceptual data schema can be simplified (see the notes after the table). There are many other types of editing. The results are seen in the column **Relationship types, close to final**. I first extracted the purple group and dealt with it, then the red group, etc. The notes give more explanation.

The **Scratchpad** column and the **Relationship types, close to final** are arranged in correspondence. Relationship types marked with strike through have been consolidated; they are encompassed in a one of the final relationship types. A final relationship type is next to an empty line in the scratchpad column has been added to make the schema more complete.

I have added many relationship types from the University Database (Soergel, Organizing Information), but the conceptual data schema given here still covers only a small part of what is needed for a Totally Integrated Information System for a School.

Relationships revolving around Document, LearningObject, and Subject; **LearningActivity;** Person

Raw list of entity types and relationship types		**Scratchpad**	**Final conceptual data schema**	
Entity types, raw	**Relationship types, raw**	**Relationship types, rearranged**	**Relationship types, close to final**	**Entity types, ~final**
Document Subject GradeLevel DocumentGenre Course CourseOffering Grade (Assessment) DocumentGenre [but see below] Test (changed from Course Exam) EducationObject EducationObjectType Age AgeLevel ReadingLevel Person Condition (Being bullied) LearningObject LearningObjectType CurriculumStandard Quality (e.g., 1 – 5 stars) LearningActivityInstance [need to check LearningActivity against LearningObject] LearningActivityType	Document <*dealsWithSubject*> Subject Subject <*suitableFor*> GradeLevel Document <*suitableFor*> GradeLevel Document <*belongsTo*> DocumentGenre Person <*hasInterest*> Subject Document <*suitableFor*> Course Document <helpfulFor> Test EducationObject <*addressesCourse*> CurriculumCourse EducationObject <*addressesSubject*> CourseSubject EducationObject <*belongsTo*> EducationObjectType EducationObject <*isComponentOf*> EducationObject EducationObject <*containsReadingLevel*> AgeLevel Person <*hasCondition*> Condition Document <*usefulFor*> Condition LearningObject <*dealsWithSubject*> CurriculumSubject LearningObject <*suitableFor*> GradeLevel LearningObject <*belongsToType*> LearningObjectType LearningObject <*isPartOf*> LearningObject LearningObject <*dealsWithStandard*> CurriculumStandard LearningObject <*hasQuality*> Quality	**Purple** (see Note #1) Document <*dealsWithSubject*> Subject Document <*suitableFor*> GradeLevel Document <*belongsTo*> DocumentGenre Document <*suitableFor*> Course Document <*helpfulFor*> Test Document <*usefulFor*> Condition ~~LearningObject <*dealsWithSubject*> Subject~~ ~~LearningObject <suitableFor> GradeLevel~~ ~~LearningObject <*belongsToType*> LearningObjectType~~ LearningObject <*isPartOf*> LearningObject ~~LearningObject <*dealsWithStandard*> CurriculumStandard~~ LearningObject <*hasQuality*> Quality ~~LearningObject <*supports*> LearningOutcome~~ LearningObject <*hasPrerequisite*> LearningOutcome ~~EducationObject <*addressesCourse*> CurriculumCourse~~ ~~EducationObject <*addressesSubject*> CourseSubject~~ ~~EducationObject <*belongsTo*> EducationObjectType~~ ~~EducationObject <*isComponentOf*> EducationObject~~ ~~EducationObject <*containsReadingLevel*> AgeLevel~~ Subject <*suitableFor*> GradeLevel	Document <*dealsWithSubject*> Subject Document <*suitableFor*> GradeLevel Document <*suitableFor*> ReadingLevel Document <*belongsTo*> DocumentGenre Document <*suitableFor*> Course Document <*usedIn*> CourseOffering Document <*helpfulFor*> Test Document <*usefulFor*> Condition (*Note #2*) Document <*isPartOf*> Document Document <*hasQuality*> Quality Document <*hasPrerequisite*> LearningOutcome Document <*hasPrerequisite*> Document Subject <suitableFor> GradeLevel Subject <*hasNarrowerTerm*> Subject	EquipmentOrMaterial (*includes library materials)* . Document . . LearningObject DocumentGenre . LearningObjectType Subject . LearningOutcome . . CurriculumStandard AgeLevel (RT Age) GradeLevel ReadingLevel

Note: A learning activity could be whole class period, an activity, such as showing a video, a field trip, etc. Date&Time Place EquipmentOrMaterial (includes library materials) Assignment (all assignments from all courses in the school have their own ID) LearningOutcome TestItem	LearningActivityInstance <*isa*> LearningActivityType LearningActivityInstance <*isScheduledFor*> (Date&Time, Date&Time, Place) *** LearningActivityInstance <*isPartOf*> CourseOffering CourseOffering <*hasStudent*> Person CourseOffering <*hasInstructor*> Person Person <hasGuardian> Person LearningActivityType <*requires*> EquipmentOrMaterial Assignment <*isEvidenceFor*> LearningOutcome TestItem <*isEvidenceFor*> LearningOutcome Person <*completed*> (Assignment, Date, Grade) Person <*completed*> (TestItem, Date, Grade) Person <*achieved*> (LearningOutcome, Grade) LearningObject <*supports*> LearningOutcome LearningObject <hasPrerequisite> LearningOutcome	**Red** LearningActivityInstance <*isa*> LearningActivityType LearningActivityInstance <*isScheduledFor*> (Date&Time, Date&Time, Place) LearningActivityInstance <*isPartOf*> CourseOffering LearningActivityType <*requires*> EquipmentOrMaterial Assignment <*isEvidenceFor*> LearningOutcome TestItem <*isEvidenceFor*> LearningOutcome CourseOffering <*hasStudent*> Person CourseOffering <*hasInstructor*> Person **Green** Person <*hasInterest*> Subject Person <*hasCondition*> Condition Person <*hasGuardian*> Person Person <*completed*> (Assignment, Date, Grade) Person <*completed*> (TestItem, Date, Grade) Person <achieved> (LearningOutcome, Grade)	Subject <*hasComponent*> Subject LearningActivityIntellectual <*isa*> LearningActivityType LearningActivityInstance <*instanceOf*> LearningActivityIntellectual LearningActivityInstance <*isScheduledFor*> (Date&Time, Date&Time, Place) (*Note #3*`) LearningActivityInstance <*isPartOf*> CourseOffering LearningActivityIntellectual <*requires*> EquipmentOrMaterial LearningActivityIntellectual <*dealsWith*> Subject Assignment <*isEvidenceFor*> LearningOutcome TestItem <*isEvidenceFor*> LearningOutcome TestItem <*isPartOf*> Test CourseOffering <*hasStudent*> Person CourseOffering <*hasInstructor*> Person Course <*dealsWith*> Subject Course <*hasPrerequisite*> Course Course <*isOfferedAs*> CourseOffering CourseOffering <*usesText*> Document Person <*hasInterest*> Subject Person <*hasCondition*> Condition Person <*hasGuardian*> Person Person <*completed*> (Assignment, Date, Grade) Person <*completed*> (TestItem, Date, Grade) Person <achieved> (LearningOutcome, Grade)	LearningActivityType *(Note #4* LearningActivityIntellectual LearningActivityInstance Course CourseOffering Assignment (*Note #5*) Test TestItem Grade (Assessment) Person Age Condition (e.g., : BeingBullied Quality (*e.g. 1 – 5 stars)* Date&Time Place

Notes (print these so you can look at them as you look at the table)

Note #1 on Document and on Subject
Looking at the entity types and relationships, we can see right away that EducationObject and LearningObject are the same thing, with LearningObject the preferred term, so I listed its group of relationships first. Furthermore, we can see that the relationship types for Document and for LearningObject are partially the same; in any event, all relationship types listed for either could be used with both, so I made LearningObject a narrower term of Document and consolidated all relationship types under Document. Likewise I made LearningObjectType a narrower term of DocumentGenre so that statements about the type of a LearningObject can now be made as

Document <belongsTo> DocumentGenre

To further consolidate relationship types, we can recognize that Subject and LearningOutcome, while not the same, relate to Document and LearningActivity in similar ways, so LearningOutcome can be a narrower term of Subject. Furthermore, a CurriculumStandard is a LearningOutcome within a particular scheme promulgated by a state or a school district. All of this simplifies the conceptual data schema without limiting the information that can be stored in the database in any way. For example,

Document <*dealsWithSubject*> Subject

can now be defined broadly to encompass

LearningObject <*dealsWithSubject*> Subject
LearningObject <*supports*> LearningOutcome
LearningObject <*dealsWithStandard*> CurriculumStandard

Note #2. On Document <*usefulFor*> Condition
Documents, esp. books, may be helpful to a person who is bullied or is overly shy or has some behavioral disorder. Reading a book in which a major character successfully deals with such issues may help the reader deal with his or her problem. Such use of books is called bibliotherapy; bibliotherapy can be used in conjunction with (adjunct to) other measures or therapies

Note #3. On Age, AgeLevel, GradeLevel, and ReadingLevel
These are interrelated but different entity types: Age (applied to people and other objects), AgeLevel (applied to objects that are suitable / appropriate for people of a certain age, values are usually age intervals), GradeLevel, and ReadingLevel (for which there are several measurement scales, such as grade level, Lexiles, etc.) The last three are often confused.

Note #4. Entity types related to LearningActivity
A LearningActivity is a process, a LearningObject is an object. A LearningActivityType is the general type of a learning activity, such as GroupActivity, Exploration, ShowingVideo, FieldTrip. LearningActivityIntellectual is the abstract idea of a learning activity, such as may be described in a LearningObject. A learning activity may be supported by one or more LearningObjects. A LearningActivity could be designed to take a whole semester, a class period, or part of a class period. A LearningActivityInstance is an actual activity happening in a specific time and place, it may be scheduled for several time intervals (given as beginning Date&Time and ending Date&Time at different places (for example, a field trip)

Note #5 Assignment
All assignments from all courses in the school have their own ID