Database Design and Creation for The Journal of Computing Professionals Using Entity-Relationship Modeling

By:
Evan Bonsignori

Submitted to:
Dr. Hashemi

As
Assignment #3
for
Database Systems

April 30th, 2018

Table of Contents

Subject **Page#**

Table of Figures

Table of Tables

I. Introduction

As a reoccurring journal with many records, you are an enterprise who requires a database system. A database system is an electronic record keeping system. A database system contains a database, which is your collection of persistent data regarding your enterprise. Persistent data is the data that cannot be 100% removed from your database. Persistent data is broken down into two types of data: operation data and decision support data. Your operational data is the data used in the day to day operations of your enterprise, and your decision support data is the data used by any level of management for making a decision. Operational data consists of core data and associate data. Core data represents the entities vital to the existence of your enterprise, and associate data is the data connecting any two entities or representing the association between two pieces of data within a given entity. An entity is anything in the world that exists. Your journal issues, scientific articles, interns, etc. are the core data of your enterprise. It is our job as your database designer to define the best storage structure of your database [1].

Entity Relationship (E-R) Modelling is a top-down database design paradigm commonly used in the western US because of its origins from University of California, Berkeley. In E-R modelling, we describe your database in blocks, convert each block into an E-R diagram, until we arrive at a final E-R diagram that represents your database. Previously, we designed your database using the Universal Relation (UR) approach. The UR approach is a bottom-up database design paradigm that is commonly used in the eastern US because of its origins from MIT. In this project, we will demonstrate design of a database from the same data source as last time (the two journal excerpts), but using the E-R approach rather than the UR approach [2].

We begin the E-R modelling approach by writing a description of your database and blocking those descriptions by entities. We will create this description using the two excerpts from your journal that you provided us with. From these block descriptions, we will work towards a final E-R diagram that represents your database, and then convert that diagram into a set of relations [2].

A relation is defined as, “a subset of cartesian product of n not necessarily distinct domains.” A domain is a pool of data where all data in a domain must have the same type [3]. Each relation contains attributes and tuples. A tuple is defined as, “A number of values for one instance of an entity represented by a relation.” An attribute is a subset of a domain where there is an underlying, not necessarily distinct domain for every attribute of a relation [3]. Each attribute has two properties of type and instance. The type of an attribute is not time dependent (does not change with time), while the instance of an attribute is time dependent (changes with time). Additionally, an entity also has two properties of type and instance. The type (time independent) and instance (time dependent). There is only one type and many instances for an entity [2].

II. Database Description Blocks

The following items in bold are the result of systematically blocking the information about your database based on entities and their descriptions. Each block contains all the sentences pertaining to an entity and may include sentence redundancies from sentences where multiple participating entities are mentioned. Each block begins with the entity title in bold followed by "*block:*". Everything after the colons and before the next bold item are the sentences describing the entity [2].

Journal Block:
A journal is uniquely identified by its name. Each journal publishes multiple issues with a certain frequency.

Issue Block:
An issue is uniquely identified by its issue number in a given journal. An issue has a volume number and date that is was published on. An issue's date has a month, year, and day. Each issue lists multiple scientific articles, call for papers, events, jobs, and student interns.

Scientific Article Block:
A scientific article is uniquely described by an article number and has multiple authors and references multiple references. A reference is described by its reference number in a given scientific article. Each scientific article may have multiple keywords and topics. A scientific article always has a title, the page number of where it begins in an issue, and its written contents.

Reference Block:
A reference is described by its reference number in a given scientific article. Each reference has a title and written contents. Each reference may have multiple authors.

Call for Papers Block:
A call for papers is uniquely identified by its call for papers identification number and has its written content. Each call for papers has multiple requested topics. Each call for papers is requested for a specific conference and uses a specific secretariat for a given call for papers. Each call for papers may have to meet multiple deadlines.

Secretariat Block:
A secretariat has a title for a given call for papers. Each secretariat has a contact and address. A contact has an email, fax, phone number, and an address has a country, city, street, and may have a state and zip code. A secretariat may belong to an organization.

Deadline Block:
A deadline has a deadline number, label, and the date that the labelled deadline is due. A date has a day, month, and year.

Conference Block:
A conference is described by a conference number. Each conference has a title, start date, end date, and conference location. Each conference contains multiple chairs that sit on it, and multiple participating organizations. A participating organization has a type of participation that can either be participating or sponsoring a given conference. A start date and end date have a day, month, and year. A conference location has a heading and may have a country, city, street, state, and zip code.

Conference Chair Block:
A conference chair is collectively described by the participating chair person's name and role for a given conference. Each chair belongs to an organization.

Organization Block:
An organization has a tax ID, name, an organization location, and corporation type. The corporation type of an organization can be a school, LLC, non-profit, etc. An organization location has a heading, and may have a country, city, street, state, and zip code.
Event Block:
An event is described by its event identification number. Each event has an event location, title, start date, end date, and a contact person. Each event may have multiple sponsoring organizations. A contact person has a contact title, email, fax, phone number, country, city, street, and may have a state and zip code. A start date and end date both have a day, month, and year. An event location has a heading, and may have a country, city, street, state, and zip code
Job Block:
A job has a job identification number, position, number of references required, job status (tenue or not tenue track), salary, contact person, and its written contents. Each job has a job location. Each job may require multiple qualifications for a given job and may need to meet multiple deadlines. A job location has a heading, and may have a country, city, street, state, and zip code
Qualification Block:
A qualification is described by a requirement for a given job and has the value that the requirement requires.
Student Intern Block:
A student intern is described by their social security number, name, graduating year, degree type, degree title, and written biography contents. Each intern has a hometown location. A hometown location has a country, city, and may have a state.

III. Description Blocks to E-R Diagram

In E-R modelling, we have three abstractions: an entity, relationship, and attribute. We have already defined entity and attribute, but in the context of E-R modelling, an entity is an object, a relationship is a meaningful interaction among entities, and attributes describe entities and relationships [2]. In this section, we will derive an E-R diagram from each block description in the previous section, until we reach the final block.

Each entity must have an entity identifier, which is an attribute whose value is unique for every instance of that entity. We will represent an entity identifier by underlining the attribute in the E-R diagram. Each attribute may be: single-valued (one value for each instance of an entity), multi-valued (may have more than one value for each instance on an entity), simple (is atomic), composite (it is not atomic and can form a hierarchy), or derived (calculated by using another attribute or attributes on the fly) [2]. For an attribute to be atomic, it must not be decomposable [3].

An entity is either a regulator or weak entity. A regular entity can stand alone, while a weak entity cannot. A weak entity always depends on a regular entity (called its owner) and the entity identifier of a weak entity has unique values only within the owner's entity identifier. We represent a weak entity's entity identifier with a dotted underlined attribute [2]. The process by which a weak entity's entity identifier becomes unique will be covered in section **IV.2. Step 2**.

We represent an entity with a rectangle, a weak entity with a double border-ed rectangle, an attribute with an oval, a multi-valued attribute with a double border-ed oval, a relationship with a diamond, and a weak entity (determining) relationship with a double-bordered diamond [2].

A relationship has two properties of type (time dependent) and instance (time independent). Like an entity, there is only one type and many instances for a relationship. A relationship type always associates two or more entity types, say k entity types. Relationship R can have many instances, say n instances of r1, r2, …, rn. Relationship type R is a named set of relationship instances, ri (for i = 1, …, n) such that each ri associates k entity instances of (e1, e2, … ek) and each entity instance ej in ri is an instance of entity type Ej (for j = 1 to k). Therefore, R is a subset of cartesian product of E1 X E2 X …. X Ek [2].

Each relationship type has cardinality ratio and Participation. Cardinality ratio specifies the number of relationship instances that an entity can participate in. (1:1, 1:M, and M:N). Participation specifies whether the existence on an entity depends on the entity being related to another entity via the relationship type. We determine participation using the following algorithm:

IF:

(relationship type R exist between two entites of type Ei and Ej) and (every instance of entity type Ei is related to an instance of entity type Ej)

THEN:

entity type Ei is totally participating in the relationship type R.

ELSE:

entity type Ei is partially participating in the relationship type R [2].

In our E-R diagram, we represent cardinality using the space on the line coming from a relationship to an entity that is closest to an entity. If the line closest to an entity is horizontal, it implies that that entity's participation is 1. If the line closest to an entity is two lines that form a triangle with a vertical line through the center, then it implies that the entity's participation is M. Figure 1 provides a reference for each representation described above.

To demonstrate the process of achieving the final E-R diagram, we will work through the conversion of the regular entity description block, Scientific Article, and its weak entity dependent, Reference into an E-R diagram in the section, **III.1 Description Block to E-R Demonstration**. For brevity, we will assume sufficient demonstration is displayed after III.1, and proceed to the final E-R diagram that is derived from converting every E-R description block.

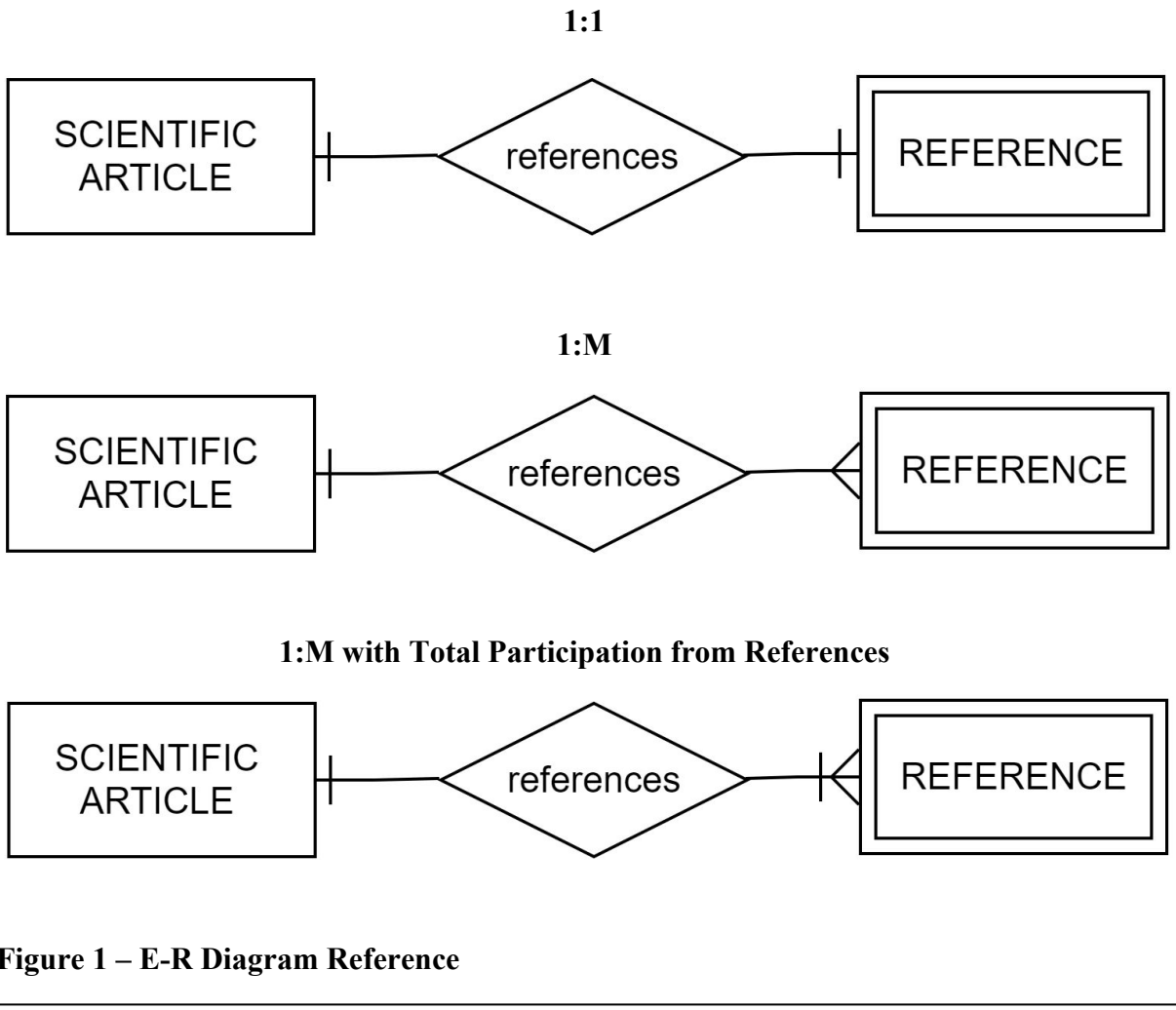

Figure 1 – E-R Diagram Reference

III.1 Description Block to E-R Demonstration

In this section we will walk through the process of converting the Scientific Article and Reference block into E-R diagrams. The description blocks are as follows:

Scientific Article Block:

A scientific article is uniquely described by an article number and has multiple authors and references multiple references. A reference is described by its reference number in a given scientific article. Each scientific article may have multiple keywords and topics. A scientific article always has a title, the page number of where it begins in an issue, and its written contents.

Reference Block:

A reference is described by its reference number in a given scientific article. Each reference has a title and written contents. Each reference may have multiple authors.

We begin by identifying that scientific article is a regular entity, represented by a rectangle with simple attributes: article number, title, contents, and page number represented by ovals. We infer from the phrase “uniquely described by” that article number is the entity identifier of Scientific Article, so we underline it within its oval (Figure 2). We have changed the name of article number to Art# in the diagram for brevity.

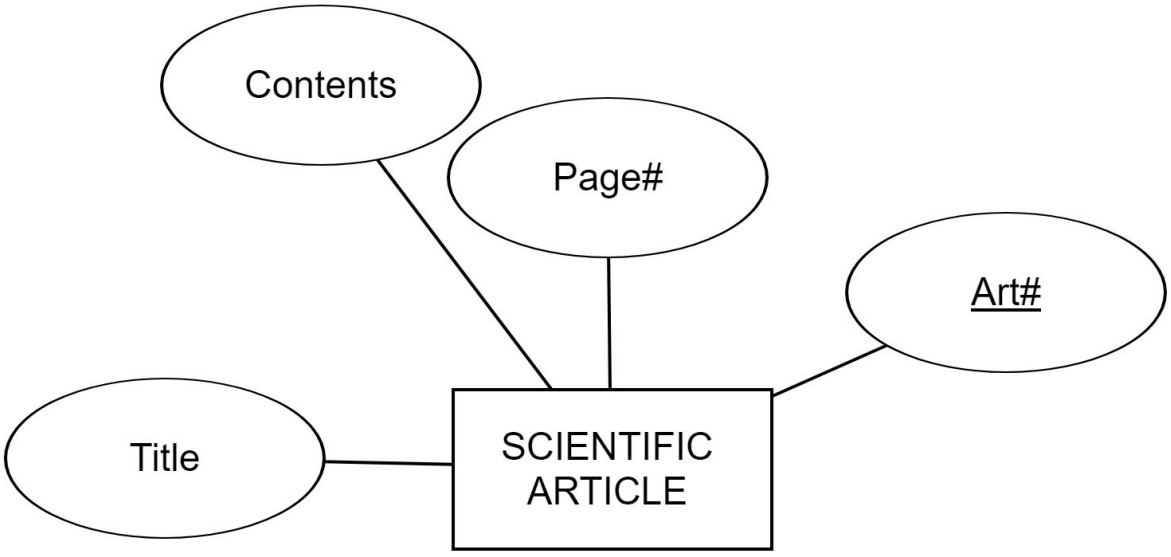

Figure 2 – Description Block to E-R Diagram Demo Part 1

Next, we represent the multi-valued attributes from the description block. We know that topics, author, and keywords aren't individual entities, because they have no attributes that describe them. Thus, we use double border-ed ovals to represent them as multi-valued attributes. However, reference does have attributes, but is described by a, "reference number in a given specific article." The "given" keyword implies that reference is a weak entity whose owner entity is Scientific Article. We represent the weak entity relationship using the verb *references* from the following phrase, "references multiple references" as the title of our relationship. Since Reference is a weak entity type, its entity identifier is Art# and Ref# collectively. We show this by dotted underlining the attribute Ref#.

The weak entity relationship, *references*, is represented as a double border-ed diamond that connects reference and scientific article, and the weak entity, *reference*, is represented as a double-bordered rectangle. Reference has simple attributes: reference number, title, and written contents. It also has the multi-valued attribute authors. We represent the different attributes with ovals and double-bordered ovals respectively (Figure 3). We change the name of refence number to Ref# and author to RefAuthor for brevity and clarity respectively.

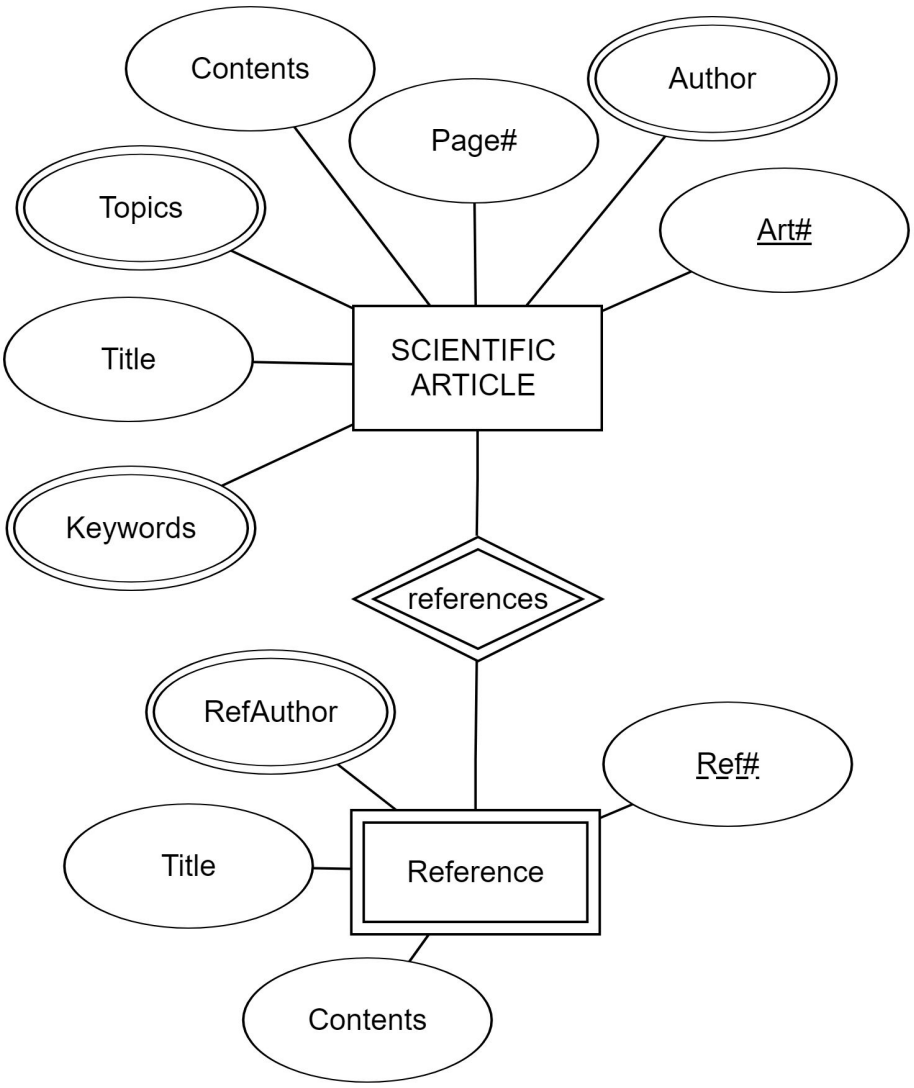

Figure 3 – Description Block to E-R Diagram Demo Part 2

The final step in our conversion is to find cardinality and participation. Since scientific article has multiple references, we know that the relationship, *Scientific Article:Reference* is 1:M because 1 article has many references, but not vise-versa. We represent Scientific Article's 1 cardinality with a single horizontal line, and references M cardinality with the three lines as shown in Figure 1's 1:M demonstration. We determine that References has total participation in a scientific article, because no reference instance does not belong to a scientific article. We demonstrate this with a horizontal line after the M cardinality on the relationship line closest to references (Figure 4). Scientific Article also has total participation in references, because every scientific article must have a reference by design choice, because an article without references is nearly impossible.

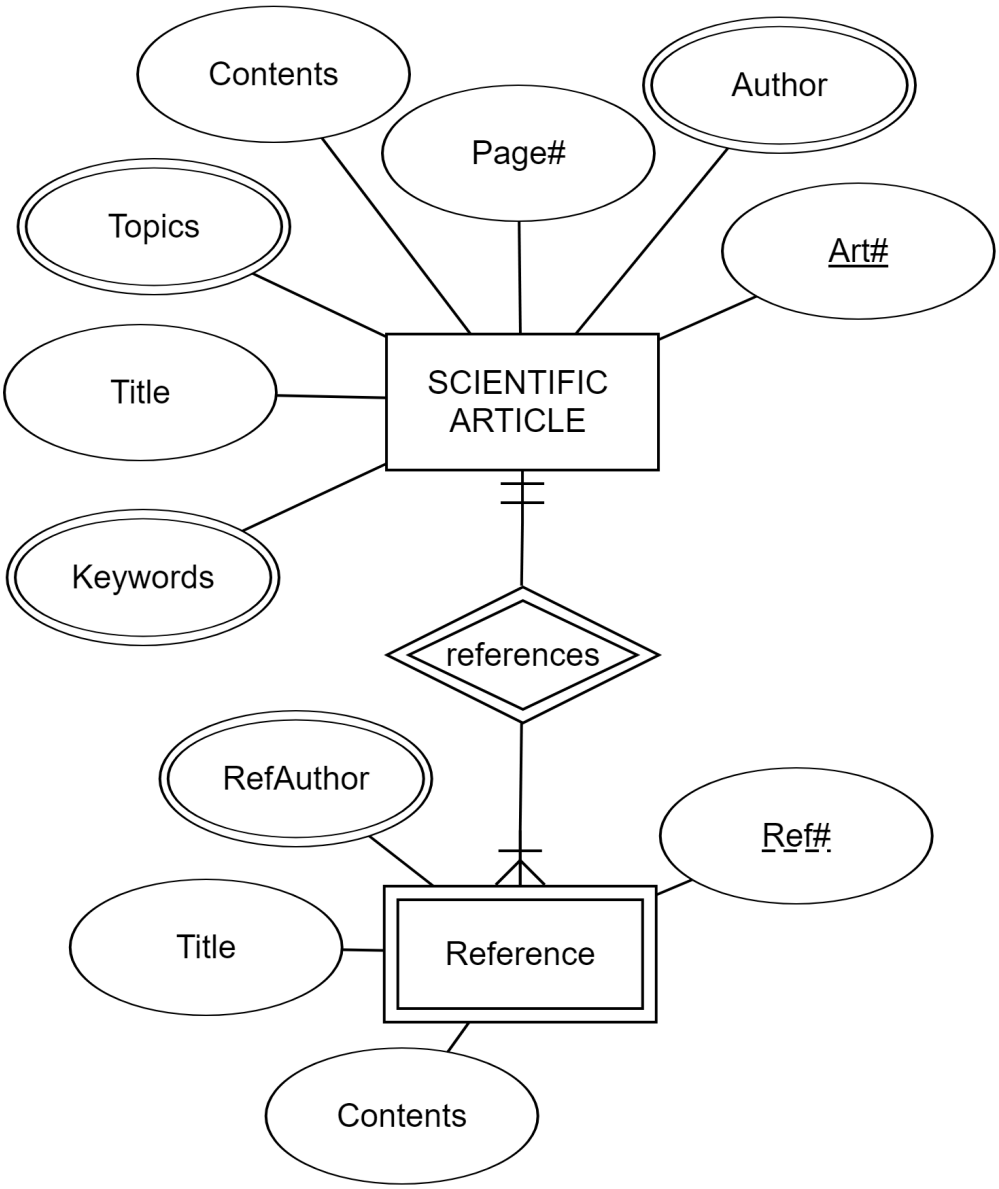

Figure 4 – Description Block to E-R Diagram Demo Part 3

III.2 Final E-R Diagram

Figure 5 contains the final E-R diagram that is the result of converting each description block into an E-R diagram. The steps followed in the previous section demonstrate how we achieved the final E-R diagram. The only two E-R diagraming topics not covered were relationships with attributes (in which case we add an oval coming from a diamond), and a composite attribute (which is represented by an oval attribute that has additional oval attributes branching from it).

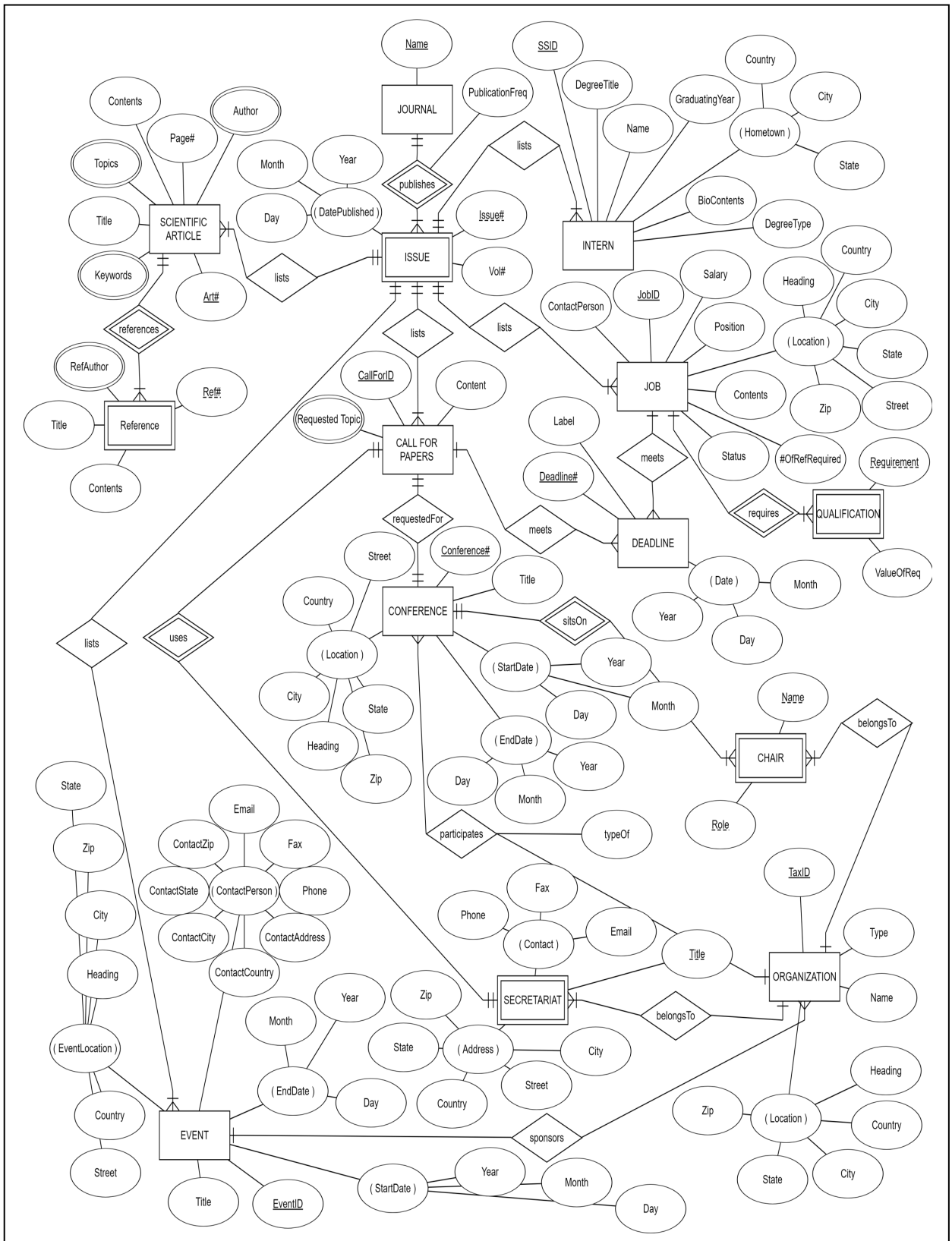

Figure 5 – Final E-R Diagram

IV. E-R Model into Relations

There are eight steps that we follow to convert our final E-R digram into a set of relations [4]. We will list the process for each step, followed by the results of our relations at each step. The results will be a table containing intensions of relations. In each step except the first, the table will contain the work from the prior step, and the changes specific to the step will be highlighted in yellow.

Intensions of a relation are the relation name with the all the relations attributes contained in parenthesis. An intension of a relation doesn't show instances of a relation, only the types of the attributes, with the relation's primary key (entity identifier) underlined [5].

A primary key is the chosen candidate key, and there can only be one primary key in a relation. A candidate key is a key that satisfies both the uniqueness and minimality conditions of a key. A key is a set of attributes that have a unique value for every tuple. A key satisfies the uniqueness condition if it's a set of attributes that uniquely identifies every tuple in a relation, and a key satisfies the minimality condition if the numbers of attributes participating in a key is minimal. A foreign key is an attribute in a relation that is not the primary key of that relation, but whose data comes from a primary domain. A primary domain is the domain from which a primary key has been derived [3].

IV.1. Step 1

For each regular entity type E, create a relation P that includes all the simple attributes of composite attributes of E. Include only the simple component attributes of composite attributes of E. Chose one of the key attributes of E as primary key for P. If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of P [4].

JOURNAL(Name)
ARTICLE(Art#, Title, Contents, Page#)
INTERN(SSID, Name, BioContents, DegreeTitle, DegreeType, GraduatingYear, Country, City, State)
JOB(JOBID, Position, Salary, Contents, Status, #OfRefRequired, Heading, Country, City, State Street, Zip, ContactPerson)
DEADLINE(Deadline#, Label, Year, Month, Day)
CALLFOR(CallForID, Content)
CONFERENCE(Conference#, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip)
ORGANIZATION(TaxID, Name, Type)
EVENT(EventID, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, ContactCountry, ContactCity, ContactState, ContactAddress, ContractZip, Email, Fax, Phone)

Table 1 – Step 1 of E-R to Relations

IV.2. Step 2

For each weak entity type W with owner entity type E, create a relation P, and include all simple attributes (or simple components of composite attributes) of W as attributes of P. In addition, include as foreign key attributes of P the primary key attribute(s) of the relation(s) that correspond to the owner entity types(s). The primary key of P is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any [4].

JOURNAL(Name)
ISSUE(Name, Issue#, Vol#, Year, Month, Day)
ARTICLE(Art#, Title, Contents, Page#)
REFERENCE(Issue#, Ref#, Title, Contents)
INTERN(SSID, Name, BioContents, DegreeTitle, DegreeType, GraduatingYear, Country, City, State)
JOB(JOBID, Position, Salary, Contents, Status, #OfRefRequired, Heading, Country, City, State Street, Zip, ContactPerson)
QUALIFICATION(Requirement, ValueOfReq)
DEADLINE(Deadline#, Label, Year, Month, Day)
CALLFOR(CallForID, Content)
CONFERENCE(Conference#, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip)
SECRETARIAT(CallForID, Title, Country, City, State, Street, Zip, Email, Fax, Phone)
CHAIR(Conference#, ChairName, Role)
ORGANIZATION(TaxID, Name, Type)
EVENT(EventID, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, ContactCountry, ContactCity, ContactState, ContactAddress, ContractZip, Email, Fax, Phone)
Table 2 – Step 2 of E-R to Relations

IV.3. Step 3

For each binary 1:1 relationship type R, identify the relations S and T that represent the entity types participating in R. Choose one of the relations-S, say- and include as foreign key in S the primary key of T. (It is better to choose an entity type with total participation in R in the role of S.) Include all the simple attributes and simple components of composite attributes of the 1:1 relationship type R as attributes of S [4].

JOURNAL(Name)
ISSUE(Name, Issue#, Vol#, Year, Month, Day)
ARTICLE(Art#, Title, Contents, Page#)
REFERENCE(Issue#, Ref#, Title, Contents)
INTERN(SSID, Name, BioContents, DegreeTitle, DegreeType, GraduatingYear, Country, City, State)
JOB(JOBID, Position, Salary, Contents, Status, #OfRefRequired, Heading, Country, City, State Street, Zip, ContactPerson)
QUALIFICATION(Requirement, ValueOfReq)
DEADLINE(Deadline#, Label, Year, Month, Day)
CALLFOR(CallForID, Content, Conference#)
CONFERENCE(Conference#, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip)
SECRETARIAT(CallForID, Title, Country, City, State, Street, Zip, Email, Fax, Phone)
CHAIR(Conference#, ChairName, Role)
ORGANIZATION(TaxID, Name, Type)
EVENT(EventID, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, ContactCountry, ContactCity, ContactState, ContactAddress, ContractZip, Email, Fax, Phone)
Table 3 – Step 3 of E-R to Relations

IV.4. Step 4

For each binary 1:M relationship type R, identify the relations S and T that represent the entity types participating in R (S is in the N-side and T is in the 1-side). Include as foreign key in S the primary key of the relation T. Include also any simple attribute and simple components of composite attributes of type R as attributes of S [4].

JOURNAL(Name)

ISSUE(Name, Issue#, Vol#, Year, Month, Day, PublicationFreq)
ARTICLE(Art#, Title, Contents, Page#, Issue#)
REFERENCE(Issue#, Ref#, Title, Contents)
INTERN(SSID, Name, BioContents, DegreeTitle, DegreeType, GraduatingYear, Country, City, State#, Issue#)
JOB(JOBID, Position, Salary, Contents, Status, #OfRefRequired, Heading, Country, City, State Street, Zip, ContactPerson, Issue#)
QUALIFICATION(Requirement, ValueOfReq)
DEADLINE(Deadline#, Label, Year, Month, Day, JobID, CallForID)
CALLFOR(CallForID, Content, Conference#, Title, Issue#)
CONFERENCE(Conference#, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, TaxID)
SECRETARIAT(CallForID, Title, Country, City, State, Street, Zip, Email, Fax, Phone, TaxID)
CHAIR(Conference#, ChairName, Role, TaxID)
ORGANIZATION(TaxID, Name, Type, EventID)
EVENT(EventID, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, ContactCountry, ContactCity, ContactState, ContactAddress, ContractZip, Email, Fax, Phone, Issue#)
Table 4 – Step 4 of E-R to Relations

IV.5. Step 5

For each binary 1:M relationship type R (recursive), identify the relations S and T that represent the entity types participating in R (S is in the N-side and T is in 1-side). Include as foreign key in S the primary key of the relation T. Include also any simple attributes and simple components of composite attributes of type R as attributes of S [4].

A recursive relationship is when an entity has a relationship with itself [4]. There exists no recursive 1:M relationships in the E-R diagram, so there are no changes from Table 4 in **Step 5**.

IV.6. Step 6

For each binary M:N relationship type R, identify the relations S and T that represent the entity types participating in R. Create a new relation P to represent R. Include the primary keys of S and T in P that collectively makes the primary key of P and they also act as foreign keys. Include

also the simple attributes and simple components of composite attributes of type R as attributes of P [4].
There exists no N:M relationships in the E-R diagram, so there are no changes from Table 4 in **Step 6**.

IV.7. Step 7

For each n-ary (n>2) relationship type R, identify the relations S1, …, Sn correspond to the n entity types participating in R. If entity type Ei has the max participation of 1, then mark relation Si. Create a new relation P to represent R. Include the primary keys of S1, …, Sn in P as foreign keys. Include also the simple attributes and simple components of composite attributes of type R as attributes of P. If there is a marked relation Si, then the key of Si in P is the primary key of P; otherwise all the foreign keys in P collectively make the primary key of P [4].
An n-ary relationship is when a relationship has more than 2 (n) participating entities [4]. There exists no n-ary relationships in the E-R diagram, so there are no changes from Table 4 in **Step 7**.

IV.8. Step 8

For each multivalued attribute A that belongs to the entity type E or relationship type R (let S be the relation that represent E or R), create a new relation P that includes an attribute corresponding to A plus the primary key, K, of S (as the foreign key). Combination of A and K makes the primary key of relation P [4].

JOURNAL(Name)
ISSUE(Name, Issue#, Vol#, Year, Month, Day, PublicationFreq)
ARTICLE(Art#, Title, Contents, Page#, Issue#)
ARTICLEAUTHOR(Art#, Author)
ARTICLETOPIC(Art#, Topic)
ARTICLEKEYWORD(Art#, Keyword)
REFERENCE(Issue#, Ref#, Title, Contents)
REFERENCEAUTHOR(Art#, Ref#, RefAuthor)
INTERN(SSID, Name, BioContents, DegreeTitle, DegreeType, GraduatingYear, Country, City, State#, Issue#)
JOB(JOBID, Position, Salary, Contents, Status, #OfRefRequired, Heading, Country, City, State Street, Zip, ContactPerson, Issue#)
QUALIFICATION(Requirement, ValueOfReq)
DEADLINE(Deadline#, Label, Year, Month, Day, JobID, CallForID)

CALLFOR(CallForID, Content, Conference#, Title, Issue#)
CALLFORTOPIC(CallForID, RequestedTopic)
CONFERENCE(Conference#, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, TaxID)
SECRETARIAT(CallForID, Title, Country, City, State, Street, Zip, Email, Fax, Phone, TaxID)
CHAIR(Conference#, ChairName, Role, TaxID)
ORGANIZATION(TaxID, Name, Type, EventID)
EVENT(EventID, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, ContactCountry, ContactCity, ContactState, ContactAddress, ContractZip, Email, Fax, Phone, Issue#)
Table 5 – Step 8 of E-R to Relations

IV.9 Final Step

Typically, in this step we would collect semantic rules from the relations found in **Step 8,** build a universal relation, and then proceed to normalize it. However, we have already carried out this task for your company and are simply demonstrating a different design approach, E-R Modelling, to deign a database for you [4].

V. Final Set of Relations

The final set of relations can be found below in Table 6. Table 6 contains the sub-relations of derived from our final E-R diagram (Figure 5) as intensions of a relation. They are not guaranteed to be normalized since we did not follow the **Final Step** in our E-R model database design process.

JOURNAL(Name)
ISSUE(Name, Issue#, Vol#, Year, Month, Day, PublicationFreq)
ARTICLE(Art#, Title, Contents, Page#, Issue#)
ARTICLEAUTHOR(Art#, Author)
ARTICLETOPIC(Art#, Topic)
ARTICLEKEYWORD(Art#, Keyword)
REFERENCE(Issue#, Ref#, Title, Contents)
REFERENCEAUTHOR(Art#, Ref#, RefAuthor)

INTERN(SSID, Name, BioContents, DegreeTitle, DegreeType, GraduatingYear, Country, City, State#, Issue#)
JOB(JOBID, Position, Salary, Contents, Status, #OfRefRequired, Heading, Country, City, State Street, Zip, ContactPerson, Issue#)
QUALIFICATION(Requirement, ValueOfReq)
DEADLINE(Deadline#, Label, Year, Month, Day, JobID, CallForID)
CALLFOR(CallForID, Content, Conference#, Title, Issue#)
CALLFORTOPIC(CallForID, RequestedTopic)
CONFERENCE(Conference#, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, TaxID)
SECRETARIAT(CallForID, Title, Country, City, State, Street, Zip, Email, Fax, Phone, TaxID)
CHAIR(Conference#, ChairName, Role, TaxID)
ORGANIZATION(TaxID, Name, Type, EventID)
EVENT(EventID, Title, StartYear, StartMonth, StartDay, EndYear, EndMonth, EndDay, Heading, Country, City, State, Street, Zip, ContactCountry, ContactCity, ContactState, ContactAddress, ContractZip, Email, Fax, Phone, Issue#)
Table 6 – Final Set of Relations

References

[1] Ray Hashemi, Lecture #2, January 10, 2018.
[2] Ray Hashemi, Lecture #21, April 2, 2018
[3] Ray Hashemi, Lecture #4, January 22, 2018.
[4] Ray Hashemi, Lecture #22, April 4, 2018
[5] Ray Hashemi, Lecture #5, February 27, 2018.