Virginia Tech. **CS 4604 – Introduction to DBMS**
Computer Science **Spring 2016, Prakash**

Project Assignment 2
(due April 6th, 2016, 4:00pm, in class—hard-copy please)

Reminders:

a. Out of 100 points. Contains 6 pages.
b. Rough time-estimates: 7-9 hours. Has to be done in groups.
c. Please type your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.
d. There could be more than one correct answer. We shall accept them all.
e. Whenever you are making an assumption, please state it clearly.
f. Unless otherwise mentioned, you may use any SQL operator seen in class. Feel free to create intermediate views for SQL.
g. **Important:**
    a. For E/R diagrams, use only the style and notation given in the lecture slides.
    b. A useful tool for creating E/R diagrams: http://logicnet.dk/DiagramDesigner/. You may have to manually draw-in some things though (like adding proper constraints etc.). There are other such programs too.
h. Lead TA for the project: Sorour Amiri.

Q1. Designing the schema [40 points]

We now want to design an appropriate schema for our BiblioVT system. The DBLP dataset contains information about approximately 1.4 million publications in the computer science literature. Here is a complete description of what your E/R diagram should model.

- Each publication has a unique string called the dblp_key that identifies it. It also has a title, a year of publication, and one or more authors.
- Some types of publications do not have authors: they have editors (see below).
- The order in which authors appear in a publication is important and must be recorded. In each publication, each author appears at most once. The rank of a author is unique within the publication. Within a publication, ranks must start at 0 and be consecutive. For some publications, the authors have not been recorded.
- A publication may also have a URL and a Digital Object Identifier (DOI).
- Each publication can cite one or more publications (these are the list of references that appear at the end of a typical publication). In addition, each publication can be associated with one or more topics. Topics are themselves arranged hierarchically, e.g., see the Computing Classification System. A topic can be a sub-topic of more than one "parent" topic and itself have one or more specialized topics as "children".
- Each publication belongs to one of the following categories:

- *article*: This type corresponds to a journal article. The publication will have an associated journal name, a volume and a number specifying the issue of the journal, page numbers, and a publisher of the journal.

- *book*: As the name indicates, this type of publication is a book. It also has a publisher, series and an ISBN number. Book has two types “Author book” and “Editor book”. Author book is a regular book with one or more authors. The contents of an “Editor book” is gathered by one or more editors.

- *incollection:* This type indicates a publication contained within a collection as chapter. Each incollection will have its own page numbers and authors. A chapter in a collection has a cross reference to the collections it was published in.

- *inproceedings:* This type indicates a paper published in the proceedings of a scientific conference. It is very similar to a publication of type "incollection". A publication of type "inproceedings" has a cross reference to the proceedings it was published in.

- *collections*: An example of a collection is a book that contains different chapters written by different authors (note that every book is not necessarily a collection). Each chapter in a collection will have the type incollection. The entire collection itself is considered a separate publication and has its own title, a list of editors, and a publisher. It is not possible for a person to be an author of a collection, i.e., collections only have editors. Within a single collection, an editor appears at most once. Within a single collection, ranks of editors are also unique and consecutively numbered starting at 0.

- *proceedings:* The conference “proceedings” is itself a separate publication with its own title, editors, and publisher. Editors and their ranks for a "proceeding" have the same function and constraints as for a "collection".

- *mastersthesis:* This publication is a Master's thesis, with a specific author, and publisher and year. Publisher has the information about department and/or university.

- *phdthesis:* This publication is a PhD thesis, with a specific author, publisher and year. Publisher has the information about the department and/or university.

- *www:* This type of "publication" is just a pointer to a web page, possibly with a title and one or more authors. It must have a URL.

Q1.1 (15 points) Draw an ER diagram for this database. Make sure to indicate primary keys, cardinality constraints, weak entities (if any), and participation constraints. There might be extra constraints which cannot be captured by the E/R diagram, make sure you mention them below the diagram. List any assumptions you make in the process.

Hint: The E/R diagram should contain ~15 (may be more/less) entities.

Q1.2 (10 points) For each entity set and relationship, write a short description in plain English of what it represents or models. One or two sentences per entity set and relationship is enough. These descriptions are primarily to help us understand that you are modeling the BiblioVT database correctly.

Q1.3 (15 points) Translate the ER diagram in Q2.1 into relational database tables (i.e. give the SQL DDL statements). Make sure that the translation captures key constraints (primary keys and foreign keys if applicable) and participation constraints in the ER diagram. Identify constraints, if any, that you are not able to capture.

Common Mistakes to avoid in design:

1. Modeling a database administrator explicitly in your E/R diagram. The DBMS usually has its own internal representation for administrators.
2. Missing arrows or rounded arrows in a many-one and/or a one-one relationship.
3. Missing arrows from a weak set to the set(s) that provide its key attribute(s).
4. Using inheritance when there is no "ISA" connection between two sets.
5. Forgetting that when entity set *B* inherits from entity set *A*, *B* inherits set *A*, *B* inherits **everything** that *A* has. In addition, *B* can define its own attributes of its own. Therefore, there is no need to repeat all the attributes/relationships that *A* has again for *B*.
6. "Cooking up" multi-way relationships, weak entity sets, or inheritance when they are not needed.
7. Forgetting to underline key attributes in the E/R model.
8. Repeating (reusing) names for different entity sets or for different relationships within the same entity set, i.e., using the same name to denote two different things.
9. When converting a multiway relationship to many two-way relationships using a connecting entity set, forgetting to introduce many-one relationships!

Q2. Refining the schema [30 points]

In this question we want to refine relational schemas using functional dependencies and normalization. You should also be able see how our original 'recipe' for converting ER-diagrams to relational schemas does in fact do a good job of getting good schemas. Assuming your ER diagram is correct, the following questions should be straightforward.

Let's call the schema you get in Q1.3 as `Schema 1`.

Q2.1 (10 points) Using the description given in Q1, for *each* of the relations in `Schema 1`, list all completely non-trivial Functional Dependencies (FDs) that apply to that relation. Only list FDs that have one attribute on the right hand side. It is enough to list only the FDs in a minimal basis.

Q2.2 (10 points) Using the FDs you got in Q2.1, for each relation in `Schema 1`, write down if it is in BCNF. Explain each answer. If any of relations are not in BCNF, decompose and normalize them to BCNF.

Q2.3 (5 points) Are the resulting relations from Q2.2 in 3NF?

Q2.4 (5 points) List (any) differences of the schema you get in Q2.2 above from `Schema 1`.

Q3. Creating the database [30 points]

We now want to create the database for using it in the final assignment.

Let's call the schema you get in Q2.2 as `Schema 2`.

We have stored all the data you will use in your project in the following **three (3)** tables (`Schema 3,` with primary key underlined`)` on our PostgreSQL server. Note that `Schema 3` obviously seems pretty bad.

`Schema 3` **dblp_pub_new** (id, dblp_key, title, type, source, series, year, volume, number, month, pages, url, publisher, isbn, crossref doi)

dblp_author_ref_new (id, author, editor, author_num)

dblp_ref_new (id, ref_id)

Note 1: Ignore the topic and sub-topic of the ER diagram (Q1) as it is not available in the database. Some tables in the `Schema 2` will be empty in your database and it is OK.

Note 2: Do not use the id attribute in the cs4604_project tables in your own databases. You must use the dblp_key attribute as the unique key for a publication. Note that this requirement means that you will have to convert the id attribute in each of the tables dblp_author_ref_new and dblp_ref_new into the dblp_key attribute. To assist you with this translation, we have declared the dblp_key attribute in dblp_pub_new as an unique not null attribute.

The description of the table attributes:

In **dblp_pub_new:**

- id: An internal key in the database (Refer **Note 2**).

- dblp_key: the DBLP key value,
- title: Title of the publication,
- type: Type of publication, i.e. article, proceedings, etc.,
- source: For journals, this is the name of journal. You can ignore this attribute for other types of publications.
- series: The series of the publication,
- year: The year of the publication,
- volume: Volume of the source where the publication was published,
- number: Number of the source where the publication was published,
- month: Month(s) when the publication was published,
- pages: Page numbers of the publication,
- publisher: Name of the publisher of the publication. If type of the publication is 'phdthesis' or 'mastersthesis' then publisher has information about university and department.
- url: external URL to the electronic edition of the publication,
- isbn: ISBN number,
- crossref: dblp_key crossreference to the other publication in which this publication was published (for incollection and inproceedings; for others it will be NULL).
- doi: The DOI of the publication.

In **dblp_author_ref_new:**

- id: the internal database key in dblp_pub_new (Refer **Note 2**).
- author: The author name
- editor: Bool being true when the author is editor of the book. If editor value is 0, it means this publication is written by author, and if value is 1, it means it is edited by editor.
- author_num: The author number (from the implicit order in the dblp xml file).

In **dblp_ref_new**:

- id: the internal database key in dblp_pub_new (Refer **Note 2**).
- ref_id: DBLP key of the publication being cited by source (note that this is the *dblp_key* of the cited publication)

We have stored the project dataset in Schema 3 form in the 'cs4604s16_project' database on the server. Similar to HW4, this time you can download all of these tables to your *group database* by using the following at the *command-prompt (NOT at the psql prompt)* of cs4604.cs.vt.edu (i.e. run this on the first prompt you get after ssh-ing to the server):

"pg_dump -U YOUR-PID cs4604s16_project | psql -d your-group_database"

Note: Please contact the TAs, in case you are unsure about your group database name.

Q3.1 (30 points) First create the BiblioVT refined schema (`Schema 2`) in the PGSQL server. Then we have to insert the right tuples into them: but we want you to insert the data *from* the 3 tables in `Schema 3`.

For each table in `Schema 2`, first show your DDL statements (CREATE TABLE and INSERT INTO), and then output only *the total number* of tuples for each table.

Example: Suppose your Schema 2 has a table reference, which stores the publication's dblp_key and the paper it referred (cited). Clearly it should get data from the dblp_pub_new and dblp_ref_new tables in Schema 3. So show the following in your answers:

1. Create the table reference:

```
CREATE TABLE REFERENCE (
id character varying(150),
ref_id character varying(150) ,
primary key(id, ref_id));
```

   (note that this create table is incomplete and just an example, as we haven't shown foreign keys)

2. Insert proper data;

```
INSERT INTO REFERENCE (id, ref_id )
(select p.dblp_key, r.ref_id  from dblp_pub_new
p,  dblp_ref_new r where  p.id=r.id);
```

3. Count the number of tuples;

```
SELECT COUNT(*) FROM REFERENCE;
```

   Output: 112317

Note 3: We have mentioned this before, but as a reminder, please note that some fields of the three tables may be NULLs for some rows; for example, for dblp_pub_new, volume would be NULL if it is not a book. Similarly, all publication may not have series, number or month. So consider the NULL values for these fields while creating the smaller tables.

Note 4: In the cs4604_project database there is a sequence named "dblp_pub_new_id_seq". Just ignore it.

Important: After creating your tables, please delete your copy of the Schema 3 tables in your group database to save space. Also, please do not use your personal database for the project data (we won't have enough space if everyone copies data to their personal databases).