Homework Problem Set 10: Data Normalization

Overview

In this lab, we will explore how to normalize data and then migrate data to new tables in the process.

Learning Objectives

Upon completion of the lab, you should be able to:

- Identify data dependencies and normal forms.
- Resolve data dependencies by placing data in a higher normal form.
- Move data into new normalized tables.

What You Will Need

To complete this lab, you will need the learn-databases environment up and running, specifically:

- Microsoft SQL Server DBMS.
- Provision the **demo** and **cheepwebhosting** databases using the database provisioner application https://localhost:5000.
- Azure Data Studio connected to SQL Server with an open query window.
- **Fudgenbooks.sql** file from where you got this lab.
- Please review the first lab if you require assistance with these tools.

Walkthrough

In this walkthrough, we will normalize the Fudgenbooks database. Execute the **fudgenbooks.sql** script to create the table **fudgenbooks** in the **demo** database. The **isbn** column is the primary key of this table.

isbn	title	price	author1	author2	author3	subjects	pages	pub_no	pub_name	pub_webs
372317842	Introduction to Money Laundering	29.9500	Mandafort	Made-Off	*NULL*	scam,money laundering	367	101	Rypoff	http://r
472325845	Imbezzle Like a Pro	34.9500	Made-Off	Moneesgon	*NULL*	imbezzle,scam	670	101	Rypoff	http://r
535621977	The Internet Scammer's Bible	44.9500	Screwm	Sucka	*NULL*	phising,id theft,scams	944	102	BS Press	http://r
635619239	Art of the Ponzi Scheme	39.9500	Dewey	Screwm	Howe	scams, ponzi	450	102	BS Press	http://r

To review, for the data normalization process, we must perform these steps:

- Resolve any columns not dependent on the key (if they exist), then
- Resolve any partial key dependencies (if they exist), then
- Resolve any transitive dependencies (if they exist), and finally

- Add the foreign key constraints.

In general, each of the three resolution processes are similar. It involves splitting the data in the original table by placing the columns with dependencies into a new table. How the new tables are created will depend on the type of dependency. We figure this out by first writing the SELECT statement to get the desired data.

Next, we write a migration script to create the table and insert the data:

- Write a DROP TABLE before the SELECT as part of the up/down process.
- Use the SELECT statement you wrote but add the INTO clause to create a new table and insert the query output.
- ALTER the table after the SELECT to add the primary key (we want entity integrity).
- Run a SELECT on the new table to verify the data are correct.

Step 1: Resolve Columns With No Dependency on the Key

In the fudgenbooks example, you cannot get a single atomic value for **author** for the given key **isbn.** Likewise, you cannot get a single **subject**, either. This is because these columns contain multivalued attributes, which are not desirable in a relational table because it is not trivial to query the data.

Because both **subjects** and **authors** are hidden many-to-many relationships, we must resolve each to a bridge table and lookup table. When we are finished, we should have five tables:

- The 1NF version of fudgenbooks with the no key dependencies removed
- A lookup table of unique authors
- A bridge table connecting each **isbn** to its many **author**(s)
- A lookup table of unique subjects
- A bridge table connecting each **isbn** to its many **subject**(s)

Step 1.1: fudgenbooks_1nf

The 1NF version of fudgenbooks is simple:

```
select isbn, title, price pages, pub_no, pub_name, pub_website
    from fudgenbooks
```

esults Messages

isbn	title	pages	pub_no	pub_name	pub_website
372317842	Introduction to Money Launde…	29.9500	101	Rypoff	http://www.rypoffpub
472325845	Imbezzle Like a Pro	34.9500	101	Rypoff	http://www.rypoffpub
535621977	The Internet Scammer's Bible	44.9500	102	BS Press	http://www.bspress.c
635619239	Art of the Ponzi Scheme	39.9500	102	BS Press	http://www.bspress.c

Then we write our migration script using the four-step resolution process from above (drop, make table, add PK constraint, select to verify).

```
28  drop table if exists fudgenbooks_1nf
29  go
30  select isbn, title, price, pages, pub_no, pub_name, pub_website
31      into fudgenbooks_1nf
32      from fudgenbooks
33  GO
34  alter table fudgenbooks_1nf add constraint pk_fudgenbooks_1nf primary
35  GO
36  select * from fudgenbooks_1nf
37  GO
```

esults Messages

isbn	title	price	pages	pub_no	pub_name	pub_website
372317842	Introduction to Money Launde…	29.9500	367	101	Rypoff	http://www.rypoffp
472325845	Imbezzle Like a Pro	34.9500	670	101	Rypoff	http://www.rypoffp
535621977	The Internet Scammer's Bible	44.9500	944	102	BS Press	http://www.bspress
635619239	Art of the Ponzi Scheme	39.9500	450	102	BS Press	http://www.bspress

The **INTO** clause on line 31 will create a new table from the query output. This is the key to migrating the normalized data.

Step 1.2: fb_authors Lookup Table

To create the lookup table, we must combine the unique values from columns **author1**, **author2**, and **author3**. Here, a UNION query does the trick. This is the common approach to use when there are multiple columns.

```
select author1 as author_name from fudgenbooks where author1 i
    union
select author2 from fudgenbooks where author2 is not null
    union
select author3 from fudgenbooks where author3 is not null
```

esults Messages

author_name
Dewey
Howe
Made-Off
Mandafort
Moneesgon
Screwm
Sucka

With the desired output, we then turn this into a migration script with our four-step process once more:

```
drop table if exists fb_authors
GO
select a.author_name
    into fb_authors
from  (
    select author1 as author_name from fudgenbooks where author1 i
        union
    select author2 from fudgenbooks where author2 is not null
        union
    select author3 from fudgenbooks where author3 is not null
) as a
GO
alter table fb_authors alter column author_name varchar(20) not NU
GO
alter table fb_authors add constraint pk_fb_authors primary key (a
GO
select * from fb_authors
```

esults Messages

author_name
Dewey
Howe
Made-Off
Mandafort
Moneesgon
Screwm
Sucka

Notice in this migration script we had to alter the **author_name** column, setting it to **not null**. This is required because the table created from the INTO clause allows null on all columns except the original primary key, **isbn**.

Step 1.3: fb_book_authors Bridge Table

In the last step of the resolution of the **authors** columns, we must create the bridge table, assigning each **isbn** and **author_name** a row in the table. When the values are in multiple columns, we use the UNPIVOT clause to build the bridge table:

```
select isbn, author_name
    from fudgenbooks unpivot (
        author_name for author_column in (author1,author2
    ) as upvt
```

esults Messages

isbn	author_name
372317842	Mandafort
372317842	Made-Off
472325845	Made-Off
472325845	Moneesgon
535621977	Screwm
535621977	Sucka
635619239	Dewey
635619239	Screwm
635619239	Howe

Notice this data still line up with the original data (three authors of book with isbn 635619239, for example), only now the data are easier to query.

Next, we transform this query into a migration script:

```sql
drop table if exists fb_book_authors
go
select isbn, author_name
    into fb_book_authors
    from fudgenbooks unpivot (
        author_name for author_column in (author1,author2,author3)
    ) as upvt
GO
alter table fb_book_authors alter column author_name varchar(20) not NULL
GO
alter table fb_book_authors add constraint pk_fb_book_authors primary key (isbn,
GO
select * from fb_book_authors
```

esults Messages

isbn	author_name
372317842	Made-Off
372317842	Mandafort
472325845	Made-Off
472325845	Moneesgon
535621977	Screwm
535621977	Sucka
635619239	Dewey
635619239	Howe
635619239	Screwm

Note how we used a composite primary key in this migration script. This makes sense because the **fb_book_authors** table is a bridge table.

Step 1.4: fb_subjects Lookup Table

Because the **subjects** column is a multivalued single column, we use the STRING_SPLIT function to help us extract the values. We also need distinct to pare down the number of unique values in the lookup table:

```
73  select distinct value as subject from fudgenbooks cross apply string_split(s
```

esults Messages

subject
id theft
imbezzle
money laundering
phising
ponzi
scams

It is left to the reader to turn this into a migration script that creates the table **fb_subjects**, populated with data and having the appropriate primary key set.

Step 1.5: fb_book_subjects Bridge Table

Here is the SQL to for the bridge table, which is similar to the lookup table.

```
75  select isbn, value as subject from fudgenbooks cross apply string_split
```

esults Messages

isbn	subject
372317842	scams
372317842	money laundering
472325845	imbezzle
472325845	scams
535621977	phising
535621977	id theft
535621977	scams
635619239	scams
635619239	ponzi

It is left to the reader to turn this into a migration script that creates the **fb_book_subjects** table.

When we are finished, we should have five tables, and with all columns key dependent, we are

in first normal form (1NF).

Step 2: Resolve Any Partial Key Dependencies

We can skip this step because partial dependencies apply only to composite primary keys. The only composite primary keys are in the bridge tables. At this point there are no partial key dependencies. We are now in second normal form (2NF)

Step 3: Resolve Any Transitive dependencies

In this final step, we must resolve any transitive dependencies. These occur when a non-key acts as a key for other non-key columns. Observe:

isbn	title	pages	pub_no	pub_name	pub_website
372317842	Introduction to Money Laundе…	29.9500	101	Rypoff	http://www.rypoff
472325845	Imbezzle Like a Pro	34.9500	101	Rypoff	http://www.rypoff
535621977	The Internet Scammer's Bible	44.9500	102	BS Press	http://www.bspres
635619239	Art of the Ponzi Scheme	39.9500	102	BS Press	http://www.bspres

The **pub_no** (publisher number) column acts as a key for the **pub_name** and **pub_website** columns. In actuality, this table has four books and two publishers. The first two books were published by Rypoff Publishing, and the last two books were published by B. S. Press. Transitive dependencies are hidden 1-M relationships, like this one that we have between publisher and book. To resolve, we must:

- Create a 3NF version of fudgenbooks_1nf. We will call this table **fb_books** with the transitive dependencies removed. We must leave **pub_no** in the table as the FK. This is the “many” side table of the hidden 1-M relationship.
- The table **fb_publishers** should contain the **pub_no** as primary key and the transitively dependent columns. This table is the “one” side of this hidden 1-M relationship.

Step 3.1: fb_books Table From the Original Table

First, remove the transitive dependencies from the **fudgenbooks_1nf** table to create the table **fb_books**.

```
select isbn, title, price, pages, pub_no
    from fudgenbooks_1nf
```

sults Messages

isbn	title	price	pages
372317842	Introduction to Money Laundering	29.9500	367
472325845	Imbezzle Like a Pro	34.9500	670
535621977	The Internet Scammer's Bible	44.9500	944
635619239	Art of the Ponzi Scheme	39.9500	450

The migration script for the table **fb_books** is left to the reader to complete.

Step 3.2: fb_publishers Table From Transitive Dependencies

```
select distinct pub_no, pub_name, pub_web
    from fudgenbooks_1nf
```

sults Messages

pub_no	pub_name	pub_website
101	Rypoff	http://www.rypoffpublishing.…
102	BS Press	http://www.bspress.com/books

The migration script for the table **fb_publishers** is left to the reader to complete.

At this point, there are no transitive dependencies, so we are now in third normal form (3NF).

Step 4: Add the Foreign Keys

At this point, our tables are in third normal form. Our normalized model is complete, and so now we should reintroduce our foreign keys back into the model. Here are our tables:

```
select * from INFORMATION_SCHEMA.TABL
    where TABLE_NAME like 'fb_%'
```

Results Messages

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYI
demo	dbo	fb_authors	BASE TAE
demo	dbo	fb_book_authors	BASE TAE
demo	dbo	fb_book_subjects	BASE TAE
demo	dbo	fb_books	BASE TAE
demo	dbo	fb_publishers	BASE TAE
demo	dbo	fb_subjects	BASE TAE

Left to the reader, write an up/down script to add the following foreign keys:

Table	Column	FK Name	References
fb_book_authors	isbn	fk_book_authors_isbn	fb_books(isbn)
fb_book_authors	author_name	fk_book_authors_author_name	fb_authors(author_name)
fb_book_subjects	isbn	fk_book_subjects_isbn	fb_books(isbn)
fb_book_subjects	subject	fk_book_subjects_subject	fb_subjects(subject)
fb_books	pub_no	fk_books_pub_no	fb_publishers(pub_no)

Your script should alter the tables, add the FKs at the bottom of the script, and drop the foreign keys (if they exist) before you start the migration. Code like this, which soft-drops the FK, should appear at the top, before any migrations.

```
-- foreign keys
if exists(select * from INFORMATION_SCHEMA.CONSTRAINT_TABI
    where CONSTRAINT_NAME= 'fk_books_pub_no')
    alter table fb_books drop fk_books_pub_no
```

(Repeat for each foreign key.)

When you are finished, you should have a single script to normalize and migrate the data to new tables!

Questions

Answer these questions using the problem set submission template. For any screen shots provided, please follow the guidelines for submitting a screen shot.

- Provide a screen shot of your working migrations for Steps 1.4, 3.1, and 3.2 in the walkthrough.

```
SELECT DISTINCT VALUE AS subject
FROM fudgenbooks
CROSS APPLY string_split(subjects, ',')
```

Results | Messages

	subject
1	id theft
2	imbezzle
3	money laundering
4	phising
5	ponzi
6	scams

*Untitled - ...

File Edit Format View

pwalsh03

Windows (CRLF) UT

```
-- First, remove the transitive dependencies f
SELECT isbn, title, price, pages, pub_no
FROM fudgenbooks_1nf
```

pwalsh03

Windows (CRLF) UTF-8

Results | Messages

	isbn	title	price	pages	pub_no
1	372317842	Introduction to Money Laundering	29.95	367	101
2	472325845	Imbezzle Like a Pro	34.95	670	101
3	535621977	The Internet Scammer's Bible	44.95	944	102
4	635619239	Art of the Ponzi Scheme	39.95	450	102

```
149   -- The migration script for the table fb_books is left to the reader to comp
150   SELECT DISTINCT pub_no, pub_name, pub_website
151   FROM fudgenbooks_1nf
152
153
```

Results | Messages

	pub_no	pub_name	pub_website
1	101	Rypoff	http://www.rypoffpublishing.com
2	102	BS Press	http://www.bspress.com/books

*Untitled -
File Edit Forr
pwalsh03
Windows (CRLI

- Provide a screen shot of your adding foreign keys in Step 4 and a separate screen shot of your code to drop the foreign keys.

```
249   ALTER TABLE fb_book_authors
250     ADD CONSTRAINT fk_book_authors_isbn
251     FOREIGN KEY (isbn) REFERENCES fudgenbooks(isbn);
252
253   ALTER TABLE fb_book_authors
254     ADD CONSTRAINT fk_book_authors_author_name
255     FOREIGN KEY (author_name) REFERENCES fb_authors(author_name);
256
257   -- Table: fb_book_subjects
258   ALTER TABLE fb_book_subjects
259     ADD CONSTRAINT fk_book_subjects_isbn
260     FOREIGN KEY (isbn) REFERENCES fb_books(isbn);
261
262   -- Table: fb_books
263   ALTER TABLE fb_books
264     ADD CONSTRAINT fk_books_pub_no
265     FOREIGN KEY (pub_no) REFERENCES fb_publishers(pub_no);
266
267
268
```

*Untitled -
File Edit Forn
pwalsh03
Windows (CRLF

Messages

```
12:44:48 PM    Started executing query at Line 249
               Commands completed successfully.
               Total execution time: 00:00:00.146
```

```
284   -- the FK, should appear at the top, before any migrations. (Repeat for each foreign k
285   IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE
286       WHERE CONSTRAINT_NAME = 'fk_books_pub_no')
287   ALTER TABLE fb_books DROP fk_books_pub_no
288
289   IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE
290       WHERE CONSTRAINT_NAME = 'fk_book_authors_author_name')
291   ALTER TABLE fb_book_authors DROP fk_book_authors_author_name
292
293   IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE
294       WHERE CONSTRAINT_NAME = 'fk_book_authors_isbn')
295   ALTER TABLE fb_book_authors DROP fk_book_authors_isbn
296
297   IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE
298       WHERE CONSTRAINT_NAME = 'fk_book_subjects_isbn')
299   ALTER TABLE fb_book_subjects DROP fk_book_subjects_isbn
300
301
```

*Untitled -

File Edit For

pwalsh03

Windows (CRL

Messages

```
12:43:51 PM    Started executing query at Line 285
               Commands completed successfully.
               Total execution time: 00:00:00.092
```

Normalize the **xyz_consulting** database. You can get this script in the same place you got this lab.

- Provide a screen shot of your migration scripts (if any) to 1NF.

```
334 DROP TABLE IF EXISTS projects
335 GO
336 CREATE TABLE projects (
337     project_id INT,
338     project_name VARCHAR(50) NOT NULL
339     CONSTRAINT pk_projects PRIMARY KEY (project_id)
340 )
341 -- Create the employees table
342 DROP TABLE IF EXISTS employees
343 GO
344 CREATE TABLE employees (
345     employee_id INT,
346     employee_name VARCHAR(50) NOT NULL
347     CONSTRAINT pk_employees PRIMARY KEY (employee_id)
348 )
349 -- Create the consulting_rates table
350 DROP TABLE IF EXISTS consulting_rates
351 GO
```

*Untitled - Nc

File Edit Format

pwalsh03

100% Windows (

Messages

```
11:45:43 PM    Started executing query at Line 334
               Commands completed successfully.
11:45:43 PM    Started executing query at Line 336
               Commands completed successfully.
11:45:43 PM    Started executing query at Line 344
               Commands completed successfully.
11:45:43 PM    Started executing query at Line 352
               (3 rows affected)
               (5 rows affected)
               (8 rows affected)
               Total execution time: 00:00:00.143
```

```
352 CREATE TABLE consulting_rates (
353     project_id INT NOT NULL,
354     employee_id INT NOT NULL,
355     rate_category CHAR(1) NOT NULL,
356     rate_amount MONEY NOT NULL,
357     billable_hours INT NOT NULL,
358     total_billed MONEY NOT NULL,
359     CONSTRAINT pk_consulting_rates PRIMARY KEY (project_id, employee_id),
360     CONSTRAINT fk_consulting_rates_project FOREIGN KEY (project_id) REFERENCES projects (project_id),
361     CONSTRAINT fk_consulting_rates_employee FOREIGN KEY (employee_id) REFERENCES employees (employee_id)
362 )
363 -- Populate the projects table
364 INSERT INTO projects (project_id, project_name)
365 SELECT DISTINCT project_id, project_name
366 FROM xyz_consulting
367 -- Populate the employees table
368 INSERT INTO employees (employee_id, employee_name)
369 SELECT DISTINCT employee_id, employee_name
370 FROM xyz_consulting
371 -- Populate the consulting_rates table
372 INSERT INTO consulting_rates (project_id, employee_id, rate_category, rate_amount, billable_hours, total_billed)
373 SELECT project_id, employee_id, rate_category, rate_amount, billable_hours, total_billed
374 FROM xyz_consulting
375
```

*Untitled - Notepad

File Edit Format View Help

pwalsh03

Messages

```
11:45:43 PM    Started executing query at Line 334
               Commands completed successfully.
11:45:43 PM    Started executing query at Line 336
               Commands completed successfully.
11:45:43 PM    Started executing query at Line 344
               Commands completed successfully.
```

- Provide a screen shot of your migration scripts (if any) to 2NF.

```sql
373 -- Normlalize to 2NF
374 DROP TABLE IF EXISTS consulting_rates
375 GO
376 DROP TABLE IF EXISTS projects
377 GO
378 DROP TABLE IF EXISTS employees
379 GO
380 DROP TABLE IF EXISTS consultants
381 GO
382
383 -- Create the projects table
384 CREATE TABLE projects (
385     project_id INT PRIMARY KEY,
386     project_name VARCHAR(50) NOT NULL
387 )
388 -- Create the consultants table
389 CREATE TABLE consultants (
390     employee_id INT PRIMARY KEY,
391     employee_name VARCHAR(50) NOT NULL
392 )
```

*Untitled - N

File Edit Forma

pwalsh03

100% Windows

Messages

```
11:57:17 PM    Started executing query at Line 384
               (3 rows affected)
               (5 rows affected)
               (8 rows affected)
               Total execution time: 00:00:00.128
```

```sql
394 CREATE TABLE consulting_rates (
395     project_id INT NOT NULL,
396     employee_id INT NOT NULL,
397     rate_category CHAR(1) NOT NULL,
398     rate_amount MONEY NOT NULL,
399     billable_hours INT NOT NULL,
400     total_billed MONEY NOT NULL,
401     CONSTRAINT pk_consulting_rates PRIMARY KEY (project_id, employee_id),
402     CONSTRAINT fk_consulting_rates_project FOREIGN KEY (project_id) REFERENCES projects (project_id),
403     CONSTRAINT fk_consulting_rates_consultant FOREIGN KEY (employee_id) REFERENCES consultants (employee_id)
404 )
405 -- Populate the projects table
406 INSERT INTO projects (project_id, project_name)
407 SELECT DISTINCT project_id, project_name
408 FROM xyz_consulting
409 -- Populate the consultants table
410 INSERT INTO consultants (employee_id, employee_name)
411 SELECT DISTINCT employee_id, employee_name
412 FROM xyz_consulting
413 -- Populate the consulting_rates table
414 INSERT INTO consulting_rates (project_id, employee_id, rate_category, rate_amount, billable_hours, total_billed)
415 SELECT project_id, employee_id, rate_category, rate_amount, billable_hours, total_billed
416 FROM xyz_consulting
417
```

Messages

```
11:57:17 PM    Started executing query at Line 384
               (3 rows affected)
               (5 rows affected)
               (8 rows affected)
               Total execution time: 00:00:00.128
```

*Untitled - Notepad

File Edit Format View Help

pwalsh03

- Provide a screen shot of your migration scripts (if any) to 3NF.

```sql
-- Normalize to 3NF
DROP TABLE IF EXISTS consulting_rates
GO
DROP TABLE IF EXISTS projects
GO
DROP TABLE IF EXISTS employees
GO
DROP TABLE IF EXISTS consultants
GO

-- Create the projects table
CREATE TABLE projects (
    project_id INT PRIMARY KEY,
    project_name VARCHAR(50) NOT NULL
)
-- Create the consultants table
CREATE TABLE consultants (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(50) NOT NULL
)
-- Create the consulting_rates table
CREATE TABLE consulting_rates (
    project_id INT NOT NULL,
    employee_id INT NOT NULL,
    rate_category CHAR(1) NOT NULL,
    rate_amount MONEY NOT NULL,
    billable_hours INT NOT NULL,
    total_billed MONEY NOT NULL,
    CONSTRAINT pk_consulting_rates PRIMARY KEY (project_id, employee_id),
    CONSTRAINT fk_consulting_rates_project FOREIGN KEY (project_id) REFERENCES projects (project_id),
    CONSTRAINT fk_consulting_rates_consultant FOREIGN KEY (employee_id) REFERENCES consultants (employee_id)
)
```

*Untitled - Notepad

File Edit Format View Help

pwalsh03

100% Windows (CRLF) UTF-8

Messages

```
12:00:21 AM    Started executing query at Line 431
               (3 rows affected)
               (5 rows affected)
               (8 rows affected)
```

```sql
-- Populate the projects table
INSERT INTO projects (project_id, project_name)
SELECT DISTINCT project_id, project_name
FROM xyz_consulting
-- Populate the consultants table
INSERT INTO consultants (employee_id, employee_name)
SELECT DISTINCT employee_id, employee_name
FROM xyz_consulting
-- Populate the consulting_rates table
INSERT INTO consulting_rates (project_id, employee_id, rate_category, rate_amount, billable_hours, total_billed)
SELECT project_id, employee_id, rate_category, rate_amount, billable_hours, total_billed
FROM xyz_consulting
```

*Untitled - Notepad

File Edit Format View Help

pwalsh03

Messages

- Provide a list of tables in 3NF.

Projects, Consultants, Consulting_rates

- Add all foreign keys back to the new model.

Foreign keys have already been added.