CSC472B, Introduction to Database Systems

Project Database Design and Implementation

Julie C. King
UIN 650972718

December 10, 2012

1. Project Description and Purpose:

Project description and purpose. Choose an application that you think is useful that has relevance in database system design. The application should have a minimum of three entity sets. Each entity set must have at least 2 attributes.

I chose a project that used information I've encountered doing legal document review projects. It was clear from using the various database programs on various projects that no two projects were set up the same way with regard to basic information about a document. I've created a very simple version of how I would prefer to set up selected entities and relationships about email documents.

Entity Sets:

- Email identified with an ID number; author(from); several types of recipients: to, CC, BCC; date; subject; and identification of any attachments
- Person identified with an ID number, name, department, and job title
- Document identified with an ID number, document type, author(s), and date created

Relationship Sets:

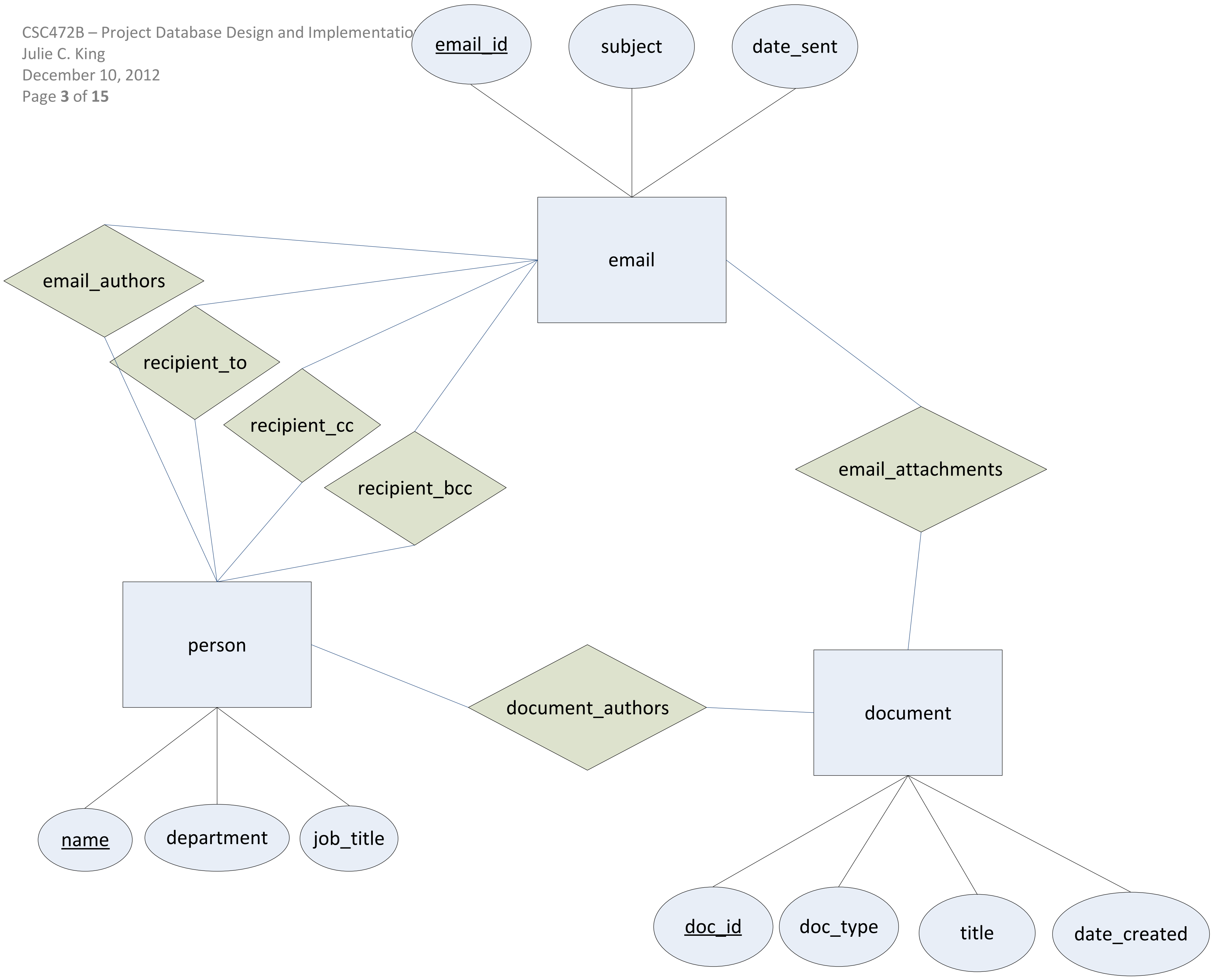
- Email will be linked to Person (To, From, CC, BCC) and Document (Attachment)
- Document will be linked to Person (Author(s))

Because the attributes author(both email and document), to, cc, bcc, and attachment can each contain multiple values, it doesn't make practical sense to include them as true attributes of either email or document because that would lead to redundancies and dependencies and thus make entering data into the tables a mess. Thus, I will implement that data using relationships.

2. Basic ER Diagram:

Produce a basic ER diagram of your database which reflects entity sets, attributes, and relationships among them. Underline the primary key on the ER diagram.

Please see the next page for the ER diagram.


email_id
subject
date_sent
email
email_authors
recipient_to
recipient_cc
recipient_bcc
email_attachments
person
document_authors
document
name
department
job_title
doc_id
doc_type
title
date_created

3. Database Schema:

Convert the ER diagram into a relational database schema in unnormalized form. Underline primary keys.

The Relations:

Entity Sets:

```
email (email_ID, date, subject)
person (name, department, job_title)
document(doc_ID, doc_type,  title, date_created)
```

Relationship Sets:

```
recipient_to (email_ID, email_to)
recipient_cc (email_ID, email_cc)
recipient_bcc (email_ID, email_bcc_)
email_authors (email_ID, email_from)
email_attachments (email_ID, doc_ID)
document_authors (doc_ID, author)
```

4. Normal Form:

Specify any 3NF, 2NF, BCNF or unnormalized relations.

The way the entities and relationship sets are set up, it removes any duplication or dependencies from the entity sets, so everything is already in 3NF.

Entity Sets:

```
email (email_ID, date, subject)
person (name, department, job_title)
document(doc_ID, doc_type,  title, date_created)
```

Relationship Sets:

```
recipient_to (email_ID, email_to)
recipient_cc (email_ID, email_cc)
recipient_bcc (email_ID, email_bcc_)
email_authors (email_ID, email_from)
email_attachments (email_ID, doc_ID)
document_authors (doc_ID, author)
```

5. Normalized Form:

Perform normalization or decomposition. Eliminate any partial functional dependencies, transitive functional dependences, repetitive repetition, and anomalies.

Entity Sets:

```
email (email_ID, date, subject)
person (name, department, job_title)
document(doc_ID, doc_type,  title, date_created)
```

Relationship Sets:

```
recipient_to (email_ID, email_to)
recipient_cc (email_ID, email_cc)
recipient_bcc (email_ID, email_bcc )
email_authors (email_ID, email_from)
email_attachments (email_ID, doc_ID)
document_authors (doc_ID, author)
```

6. Schemas:

Specify all the schemas of the resulting relations in your database. Specify the name, attributes separated by commas and place in parenthesis for each relation. Underline the primary key(s).

Entity Sets:

```
email (email_ID, date, subject)
person (name, department, job_title)
document(doc_ID, doc_type,  title, date_created)
```

Relationship Sets:

```
recipient_to (email_ID, email_to)
recipient_cc (email_ID, email_cc)
recipient_bcc (email_ID, email_bcc )
email_authors (email_ID, email_from)
email_attachments (email_ID, doc_ID)
document_authors (doc_ID, author)
```

7. Create Relations:

Create at least two relations using SQL statements including constraints. Show the SQL commands on how to create the two relations.

```
CREATE TABLE person
     (name                  varchar2(30)      NOT NULL,
      department            varchar2(30),
      job_title             varchar2(30),
      PRIMARY KEY(name));

CREATE TABLE document
     (doc_ID                number            NOT NULL,
      doc_type              varchar2(30)      NOT NULL,
      title                 varchar2(50),
      date_created          date,
      PRIMARY KEY(doc_ID));

CREATE TABLE email
     (email_ID              number            NOT NULL,
      subject               varchar2(50),
      date_sent             date              NOT NULL,
      PRIMARY KEY(email_ID));

CREATE TABLE recipient_to
     (email_ID              number            NOT NULL,
      email_to              varchar2(30)      NOT NULL,
      PRIMARY KEY(email_ID, email_to),
      FOREIGN KEY(email_ID) REFERENCES email(email_ID),
      FOREIGN KEY(email_to) REFERENCES person (name));

CREATE TABLE recipient_cc
     (email_ID              number            NOT NULL,
      email_cc              varchar2(30)      NOT NULL,
      PRIMARY KEY(email_ID, email_cc),
      FOREIGN KEY(email_ID) REFERENCES email(email_ID),
      FOREIGN KEY(email_cc) REFERENCES person (name));


CREATE TABLE recipient_bcc
     (email_ID              number            NOT NULL,
      email_bcc             varchar2(30)      NOT NULL,
      PRIMARY KEY(email_ID, email_bcc),
      FOREIGN KEY(email_ID) REFERENCES email(email_ID),
      FOREIGN KEY(email_bcc) REFERENCES person (name));

CREATE TABLE email_authors
     (email_ID              number            NOT NULL,
      email_from            varchar2(30)      NOT NULL,
      PRIMARY KEY(email_ID, email_from),
      FOREIGN KEY(email_ID) REFERENCES email(email_ID),
      FOREIGN KEY(email_from) REFERENCES person (name));
```

```
CREATE TABLE email_attachments
      (email_ID               number            NOT NULL,
       doc_ID                 number            NOT NULL,
       PRIMARY KEY(email_ID, doc_ID),
       FOREIGN KEY(email_ID) REFERENCES email(email_ID),
       FOREIGN KEY(doc_ID) REFERENCES document (doc_ID));

CREATE TABLE document_authors
      (doc_ID                       number            NOT NULL,
       author                       varchar2(30)      NOT NULL,
       PRIMARY KEY(doc_ID, author),
       FOREIGN KEY(doc_ID) REFERENCES document(doc_ID),
       FOREIGN KEY(author) REFERENCES person (name));
```

8. Populate Relations:

Insert three records of at least two relations. Show SQL commands on how to populate the relations.

I inserted five records (emails) to experiment with various relations a bit more.

```
/* populate relations */

insert into person values ('Jane Doe',    'Admin',          'CEO');
insert into person values ('John Doe',    'Admin',          'COO');
insert into person values ('Jill Smith',  'Admin',    'Assistant to CEO');
insert into person values ('Jack Smith',  'Accounting', 'Accountant');
insert into person values ('Kate Jones',  'Human Resources', 'Head of HR');
insert into person values ('Bob Jackson',       'Legal', 'Attorney');
insert into person values ('Beth Williams', 'Engineering', Test Specialist');
insert into person values ('Jim Williamson', 'Legal', 'Compliance Officer');
insert into person values ('Ann Johnson',       'none', 'Outside Counsel');


insert into document values (1, 'Report', 'Top Secret Report', '01-JAN-
2010');
insert into document values (2, 'Spreadsheet', '2009 Incidents', '01-JAN-
2010');
insert into document values (3, 'Memo', 'Company Holidays 2010', '01-DEC-
2009');
insert into document values (4, 'Photo', 'none', '25-JAN-2010');
insert into document values (5, 'Report', '2012 Projections', '30-JUN-2010');
insert into document values (6, 'Spreadsheet', '2012 Projections', '30-JUN-
2010');

insert into email values (1, 'Do not distribute',     '01-JAN-2010');
insert into email values (2, '2010 holidays',         '01-DEC-2009');
insert into email values (3, 'what is this?',         '25-JAN-2010');
insert into email values (4, '2012 projections',      '01-JUL-2010');
insert into email values (5, 'now with attachment',   '01-JUL-2010');
insert into recipient_to values (1, 'John Doe');
insert into recipient_to values (2, 'Jane Doe');
insert into recipient_to values (2, 'John Doe');
insert into recipient_to values (2, 'Jill Smith');
insert into recipient_to values (2, 'Jack Smith');
insert into recipient_to values (2, 'Bob Jackson');
insert into recipient_to values (2, 'Beth Williams');
insert into recipient_to values (2, 'Jim Williamson');
insert into recipient_to values (2, 'Ann Johnson');
insert into recipient_to values (3, 'Kate Jones');
insert into recipient_to values (4, 'John Doe');
insert into recipient_to values (5, 'John Doe');


insert into recipient_cc values (1, 'Jim Williamson');
insert into recipient_cc values (4, 'Jack Smith');
insert into recipient_cc values (5, 'Jill Smith');
insert into recipient_cc values (5, 'Bob Jackson');
insert into recipient_cc values (5, 'Beth Williams');
insert into recipient_cc values (5, 'Jim Williamson');


insert into recipient_bcc values (1, 'Bob Jackson');
insert into recipient_bcc values (1, 'Ann Johnson');


insert into email_authors values (1, 'Jane Doe');
insert into email_authors values (2, 'Kate Jones');
insert into email_authors values (3, 'Jane Doe');
insert into email_authors values (4, 'Jane Doe');
insert into email_authors values (5, 'Jane Doe');
insert into email_authors values (1, 'John Doe');


insert into email_attachments values (1, 1);
insert into email_attachments values (2, 2);
insert into email_attachments values (3, 3);
insert into email_attachments values (4, 4);
insert into email_attachments values (5, 5);
insert into email_attachments values (5, 6);


insert into document_authors values (1, 'Jane Doe');
insert into document_authors values (1, 'John Doe');
insert into document_authors values (2, 'Jane Doe');
insert into document_authors values (3, 'Kate Jones');
insert into document_authors values (5, 'John Doe');
insert into document_authors values (5, 'Jack Smith');
insert into document_authors values (6, 'Jack Smith');
```

9. SQL Query:

Formulate at least three algebraic queries in SQL statements that are relevant as report to the application you have chosen.

```
select email.email_ID, email.subject, email.date_sent,
email_authors.email_from, recipient_to.email_to,
recipient_cc.email_cc, recipient_bcc.email_bcc,
email_attachments.doc_id from email, email_authors,
recipient_to, recipient_cc, recipient_bcc, email_attachments
where email.email_ID = 1;

select email.email_ID, email.subject, email_authors.email_from
from email, email_authors where email.email_ID = 1;

select document.doc_ID, document.doc_type, document.title,
document_authors.author from document, document_authors where
document_authors = 'Jane Doe';
```

10. XPath Expression:

Convert three algebraic queries into XPath expressions.

SQL expression: `select * from email_attachments where doc_ID = 6;`
XPath expression: /email/attachments[@doc_ID = 6]

SQL expression: `select email_ID from email_authors where email_from = 'Jane Doe';`
XPath expression: /email_authors/email_IDs[@`email_from` =' Jane Doe']

SQL expression: `select * from documents where doc_type = 'spreadsheet';`
XPath expression: /documents[@doc_type='spreadsheet']

11. Describe Challenges:

Briefly describe what were the least and most challenging parts of the above and explain why. Give any recommendation about the project.

The least challenging part was deciding which entities I would use, which attributes I wanted each entity to have, and the relationships I needed to establish between the entities. I think this is because I have worked with document-related databases for the last 11 years, and I've seen so many setup variations that I already had a very strong idea in my head even before beginning this course about how I would prefer to set up those entities and relationships. Also, I find there is a natural organization to the entities, attributes, and relationships that made the setup fairly straightforward.

I found it most challenging to try to think of the database in XML form and then generate the complex queries I found fairly easy to generate in SQL. I found it very difficult to construct XPath expressions that would return the same multiple attributes from multiple entities and relationships I could return using my SQL queries. I think part of the problem is that the relationships are all many-to-many, which makes the XML form seem more complicated and cumbersome than the SQL form.

This essentially is an oversimplification of the data and how it might be set up in a real document review database designed to be used for the litigation discovery process. So many additional pieces of information might need to be collected about each document, each person, and each attachment, depending on the needs of the project and on the sophistication of the people who will be using the information. Questions constantly arise about how to treat email chains, email threads, drafts, etc. Also, no list of people ever seems to be complete or completely accurate. Furthermore, information about attachments could include modification date(s), version(s), etc. This doesn't even touch the data that could be entered about the substantive content of the emails and attachments.

This project was an interesting way to dip my toes in the pool of the complexity of setting up a litigation document review database. It certainly reinforced my supposition that taking the time to think very carefully about all the information that should be collected, how it will be collected, and how it will be used, so that setting up the entities, attributes, and relationships will be of the most utility is crucial for effective database function and management. Without the precisely proper logical foundation, the database will fail at least some of its users. This is true for vastly complex databases, but also true for simple ones like the one in this project. If you don't know what you will need and how you will need it, you won't know how to construct the database so you can get what you need when you need it and how you need it.

Appendix: JKproject.sql

This appendix contains the entirety of the sql file I used to set up and test my database via the Oracle sqlplus server at uisacad5.uis.edu.

```
/* clean up old tables */

delete from recipient_to;
drop table recipient_to CASCADE CONSTRAINTS;

delete from recipient_cc;
drop table recipient_cc CASCADE CONSTRAINTS;

delete from recipient_bcc;
drop table recipient_bcc CASCADE CONSTRAINTS;

delete from email_authors;
drop table email_authors CASCADE CONSTRAINTS;

delete from email_attachments;
drop table email_attachments CASCADE CONSTRAINTS;

delete from document_authors;
drop table document_authors CASCADE CONSTRAINTS;

delete from person;
drop table person CASCADE CONSTRAINTS;

delete from document;
drop table document CASCADE CONSTRAINTS;

delete from email;
drop table email CASCADE CONSTRAINTS;



/* create tables */

CREATE TABLE person
      (name                  varchar2(30)      NOT NULL,
       department            varchar2(30),
       job_title             varchar2(30),
       PRIMARY KEY(name));

CREATE TABLE document
      (doc_ID                number            NOT NULL,
       doc_type        varchar2(30)      NOT NULL,
       title                 varchar2(50),
       date_created          date,
       PRIMARY KEY(doc_ID));

CREATE TABLE email
      (email_ID              number            NOT NULL,
       subject               varchar2(50),
       date_sent              date        NOT NULL,
       PRIMARY KEY(email_ID));

CREATE TABLE recipient_to
      (email_ID               number            NOT NULL,
       email_to               varchar2(30)      NOT NULL,
       PRIMARY KEY(email_ID, email_to),
       FOREIGN KEY(email_ID) REFERENCES email(email_ID),
       FOREIGN KEY(email_to) REFERENCES person (name));

CREATE TABLE recipient_cc
      (email_ID               number            NOT NULL,
       email_cc               varchar2(30)      NOT NULL,
       PRIMARY KEY(email_ID, email_cc),
       FOREIGN KEY(email_ID) REFERENCES email(email_ID),
       FOREIGN KEY(email_cc) REFERENCES person (name));


CREATE TABLE recipient_bcc
      (email_ID               number            NOT NULL,
       email_bcc              varchar2(30)      NOT NULL,
       PRIMARY KEY(email_ID, email_bcc),
       FOREIGN KEY(email_ID) REFERENCES email(email_ID),
       FOREIGN KEY(email_bcc) REFERENCES person (name));

CREATE TABLE email_authors
      (email_ID               number            NOT NULL,
       email_from             varchar2(30)      NOT NULL,
       PRIMARY KEY(email_ID, email_from),
       FOREIGN KEY(email_ID) REFERENCES email(email_ID),
       FOREIGN KEY(email_from) REFERENCES person (name));

CREATE TABLE email_attachments
      (email_ID               number            NOT NULL,
       doc_ID                       number            NOT NULL,
       PRIMARY KEY(email_ID, doc_ID),
       FOREIGN KEY(email_ID) REFERENCES email(email_ID),
       FOREIGN KEY(doc_ID) REFERENCES document (doc_ID));

CREATE TABLE document_authors
      (doc_ID                       number            NOT NULL,
       author                       varchar2(30)      NOT NULL,
       PRIMARY KEY(doc_ID, author),
       FOREIGN KEY(doc_ID) REFERENCES document(doc_ID),
       FOREIGN KEY(author) REFERENCES person (name));



/* populate relations */

insert into person values ('Jane Doe',         'Admin',          'CEO');
insert into person values ('John Doe',         'Admin',          'COO');
insert into person values ('Jill Smith',  'Admin',         'Assistant to
CEO');
```

```
insert into person values ('Jack Smith',  'Accounting',
      'Accountant');
insert into person values ('Kate Jones',  'Human Resources',      'Head of
HR');
insert into person values ('Bob Jackson',       'Legal',
      'Attorney');
insert into person values ('Beth Williams',     'Engineering',          'Test
Specialist');
insert into person values ('Jim Williamson',    'Legal',          'Compliance
Officer');
insert into person values ('Ann Johnson',       'none',           'Outside
Counsel');


insert into document values (1, 'Report',       'Top Secret Report',
      '01-JAN-2010');
insert into document values (2, 'Spreadsheet',  '2009 Incidents',
      '01-JAN-2010');
insert into document values (3, 'Memo',   'Company Holidays 2010',      '01-
DEC-2009');
insert into document values (4, 'Photo',  'none',                 '25-JAN-
2010');
insert into document values (5, 'Report',       '2012 Projections',
      '30-JUN-2010');
insert into document values (6, 'Spreadsheet',  '2012 Projections',
      '30-JUN-2010');

insert into email values (1, 'Do not distribute',     '01-JAN-2010');
insert into email values (2, '2010 holidays',         '01-DEC-2009');
insert into email values (3, 'what is this?',         '25-JAN-2010');
insert into email values (4, '2012 projections',      '01-JUL-2010');
insert into email values (5, 'now with attachment',   '01-JUL-2010');


insert into recipient_to values (1, 'John Doe');
insert into recipient_to values (2, 'Jane Doe');
insert into recipient_to values (2, 'John Doe');
insert into recipient_to values (2, 'Jill Smith');
insert into recipient_to values (2, 'Jack Smith');
insert into recipient_to values (2, 'Bob Jackson');
insert into recipient_to values (2, 'Beth Williams');
insert into recipient_to values (2, 'Jim Williamson');
insert into recipient_to values (2, 'Ann Johnson');
insert into recipient_to values (3, 'Kate Jones');
insert into recipient_to values (4, 'John Doe');
insert into recipient_to values (5, 'John Doe');


insert into recipient_cc values (1, 'Jim Williamson');
insert into recipient_cc values (4, 'Jack Smith');
insert into recipient_cc values (5, 'Jill Smith');
insert into recipient_cc values (5, 'Bob Jackson');
insert into recipient_cc values (5, 'Beth Williams');
insert into recipient_cc values (5, 'Jim Williamson');
insert into recipient_bcc values (1, 'Bob Jackson');
insert into recipient_bcc values (1, 'Ann Johnson');


insert into email_authors values (1, 'Jane Doe');
insert into email_authors values (2, 'Kate Jones');
insert into email_authors values (3, 'Jane Doe');
insert into email_authors values (4, 'Jane Doe');
insert into email_authors values (5, 'Jane Doe');
insert into email_authors values (1, 'John Doe');


insert into email_attachments values (1, 1);
insert into email_attachments values (2, 2);
insert into email_attachments values (3, 3);
insert into email_attachments values (4, 4);
insert into email_attachments values (5, 5);
insert into email_attachments values (5, 6);


insert into document_authors values (1, 'Jane Doe');
insert into document_authors values (1, 'John Doe');
insert into document_authors values (2, 'Jane Doe');
insert into document_authors values (3, 'Kate Jones');
insert into document_authors values (5, 'John Doe');
insert into document_authors values (5, 'Jack Smith');
insert into document_authors values (6, 'Jack Smith');

/* check tables */

select * from email;
select * from person;
select * from document;
select * from recipient_to;
select * from recipient_cc;
select * from recipient_bcc;
select * from email_authors;
select * from email_attachments;
select * from document_authors;

select email.email_ID, email.subject, email.date_sent,
email_authors.email_from, recipient_to.email_to, recipient_cc.email_cc,
recipient_bcc.email_bcc, email_attachment.doc_id from email, email_authors,
recipient_to, recipient_cc, recipient_bcc, email_attachments where
email.email_ID = 1;

select email.email_ID, email.subject, email.date_sent,
email_authors.email_from, recipient_to.email_to, recipient_cc.email_cc,
recipient_bcc.email_bcc, email_attachment.doc_id from email, email_authors,
recipient_to, recipient_cc, recipient_bcc, email_attachments where
email.email_ID = 2;
```

```
select email.email_ID, email.subject, email.date_sent,
email_authors.email_from, recipient_to.email_to, recipient_cc.email_cc,
recipient_bcc.email_bcc, email_attachment.doc_id from email, email_authors,
recipient_to, recipient_cc, recipient_bcc, email_attachments where
email.email_ID = 3;

select email.email_ID, email.subject, email.date_sent,
email_authors.email_from, recipient_to.email_to, recipient_cc.email_cc,
recipient_bcc.email_bcc, email_attachment.doc_id from email, email_authors,
recipient_to, recipient_cc, recipient_bcc, email_attachments where
email.email_ID = 4;

select email.email_ID, email.subject, email.date_sent,
email_authors.email_from, recipient_to.email_to, recipient_cc.email_cc,
recipient_bcc.email_bcc, email_attachment.doc_id from email, email_authors,
recipient_to, recipient_cc, recipient_bcc, email_attachments where
email.email_ID = 5;


select email.email_ID, email.subject, email_authors.email_from from email,
email_authors where email.email_ID = 1;
```