# Thanks Corporation Database Project
CMSI 486 Enterprise Project
Fall 2014

Edward Bramanti

This project designs and implements a database for an employee recognition system for companies to use internally.

# Contents

# Chapter III

# Description of the Enterprise

Thanks is an effective, entirely digital, multi-purpose employee recognition system. The database will consist of many different companies who will pay for a service that makes recognizing employees simple and meaningful. This will allow the Thanks database to be used in a way that is unique to each company using the service.

The enterprise in question will make it much easier for employees to recognize one another across their company. As companies grow, it becomes difficult to maintain an atmosphere of employee worth. This growing size represents a problem as individual employees can feel like their work goes unnoticed in their company. Thanks provides a remedy for that by providing a digital way to send recognition to any employee quickly.

When opening the thanks form, it will be necessary to present a list of all employees in the company. A list of employees would appear with these attributes: name, position and department. If they saw a fellow employee do something, they should not feel uncertain about what position that employee currently occupies. One of the major goals of Thanks is to help people recognize each other regardless of rank in the company and title. Informing users about a person's job title will provide clarity into who they are thanking and their role in the organization. Users will be able to send other users the primary data type known as Thanks. For each thank, we have a user that gives the Thanks and a user that receives the Thanks. This demonstrates the personal aspect of recognition, as an employee recognizes a specific employee directly.

A Thanks also contains an area for an employee to write a message so they can explain what they are thanking the employee for. This allows for personalization of the Thanks so that employees can be detailed on the outstanding work their coworkers are doing. Finally, companies will be able to define a custom attribute for the last part of the Thanks, which will be represented as values the company wants to promote. The advantage of this value is that companies, depending on their mission statements and core beliefs, will be able to tailor this field to encourage specific attributes that represent the company within employees.

Thanks represents a social network within a company than it is a private thanking system. Messages will display on a live feed all of the Thanks being given around the company. Employees will be able to easily see the interaction and encouragement being spread amongst their acquaintances. The potential of Thanks is enormous, which is why a specific structure around a public space of thanks combined with personalized instances and explanations of employee worth are necessary in this database design.

Here are a set of questions employees may pose when retrieving data:

1. List the names of all employees in the company.
2. Which department has received the most Thanks in the company?
3. Return a list of all employees who have nicknames.
4. List all company values for the company.
5. Which employee has received the most Thanks in the company?
6. Which one of an employee's Thanks they gave received the most likes?
7. Who has never received a Thanks within a company?
8. Who is the newest employee to have joined the company?
9. What Thanks were posted in the database on October 20, 2014?
10. List the most awarded company value.

# Chapter IV

# Definition of Environment

## IV.1 Input and Report Forms

- Thanks Form
  - Name of Employee Being Thanked
  - Message
  - Company Value (optional)
    Represents a custom value that the company wants to encourage in their employees
- Employee Profile Page
  Allows for editing of a specific employee's profile. Some of the values stored in the database will not be able to be changed, such as Real Name and Job Title, and must be updated by an admin.
  - Edit Employee's Nickname
  - Edit Employee's Photo
- "Thanks Feed" of Company
  Allows employees to see the "Thanks Feed" of the company, a place where signed-in employees can like Thanks messages that have been given and comment/like those thanks messages.
- Thanks Item in "Thanks Feed"
  Allows an employee to take action on a Thanks item within the "Thanks Feed".
  - Like Current Message
    Allows employee to like a Thanks Given, whether addressed directly or external from the employee.
  - Comment on Current Message
    A comment is a text response to a Thanks Given, whether addressed directly or external from the employee.
- Admin Panel for Department
  Allows for department heads to manage information about their department.
  - Edit Department Title
    If the title of a department changes slightly, a department head is able to change this.

  - Edit Department Description
    Allows department head to change the department description information.
  - Edit Department Employee's Job Title
    Allows department head to select a specific employee's job title in their department and edit it.

- Admin Panel for Executive
  Allows for executives to manage information about their company, and to change even department-level information.
  - Edit Company Title
    Allows executives to change their company name if their corporation undergoes a name change.
  - Edit Company Founded Date
    Allows executives to set the founding date of the company.
  - Set Department Heads
    Allows executives to set the head of each department.

- Webmaster Panel for DB Manager
  Since Thanks Corporation is a business that provides its database service to other companies, a webmaster needs an admin panel to manage the many companies in the database.
  - View Companies in Database
    Allows webmaster to view all companies in the database
  - Edit Companies in Database
    Allows webmaster to deactivate companies or remove companies from database.

## IV.2 Assumptions

1. Fellow employees address thanks to other employees.
2. Each employee is assigned an account, which has predefined data and some limited customization/personalization.
3. Employees can view a newsfeed-like interface of all thanks being given throughout the company.
4. Employees have the ability to view and like/comment on all Thanks company-wide.
5. Department heads will be able to manage their department info and certain employee info.
6. Executives will be able to manage department heads and company info.

## IV.3 User-Oriented Data Dictionary

| Datum | Information Definition |
|---|---|
| comment_data | Text data contained within a comment on a Thanks data type |
| companies | View of companies for a webmaster of Thanks |
| company_title | Title of a company using Thanks |
| company_value | Value being exuding that represents the company in the Thanks |
| department_description | Description of particular department in text form |
| department_title | The name of a particular department within the company |
| employee_department | Department employee works in |
| employee_name | Name of employee in the company, form |
| employee_nickname | Nickname of an employee |
| employee_photo | Photo of an employee for the Thanks database |
| employee_title | Job title of an employees position |
| founded_date | Date that a company was founded |
| like_data | Stored when an employee likes a Thanks another employee game |
| message_data | Text data contained within the body of a Thanks data type |
| thanks_feed | List of all Thanks in a corporation |

## IV.4 Cross-Reference Table

| **Datum** | **Form/Screen** | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Thanks Form** | **Employee Profile** | **"Thanks Feed"** | **Thanks Item** | **Department Admin Panel** | **Executive Admin Panel** | **Webmaster Admin Panel** |
| comment_data | | | X | X | | | |
| companies | | | | | | | X |
| company_title | | | X | | | X | |
| company_value | X | | | | | | |
| department_description | | | | | X | | |
| department_title | | | | | X | | |
| employee_department | | X | | X | | | |
| employee_name | X | X | | X | X | X | |
| employee_nickname | | X | | X | X | X | |
| employee_photo | | X | | X | | | |
| employee_title | | X | | X | X | X | |
| founded_date | | | | | | X | |
| like_data | | | X | X | | | |
| message_data | X | | X | X | | | |
| thanks_feed | | | X | | | | |

# Chapter V

# Enterprise Database Design

## V.1 Logical Model of the Enterprise

### V.1.1 List of Entities and Attributes

- Employee
  - eid: Employee ID
  - name: Employee Name
  - job_title: Employee Job Title
  - photo: Employee Photo
  - nickname: Employee Nickname
  - started: Date the Employee Started
- Department
  - did: Department ID
  - depTitle: Department Title
  - depDescription: Department Description
- Company
  - cid: Company ID
  - cTitle: Company Title
  - founded_date: Date Company was Founded
- Thanks
  - tid: Thanks ID
  - thanksdate: Date Thanks was Given
- Like
  - likeid: Like ID
  - likedate: Date Like was Given
- Comment

  - commentid: Comment ID
  - comment_data: Comment Message
  - commentdate: Date Comment was Given

- Message
  - mid: Message ID
  - message_text: Text of Message

- Company Value
  - vid: Company Value ID
  - value_type: type of company value

### V.1.2 List of Relationships and Attributes

- COMPANY_OF_DEPARTMENT(did, cid)
  CK - did

- DEPARTMENT_OF_EMPLOYEE(eid, did)
  CK - eid

- THANKS_TO(to,eid)
  CK - to

- THANKS_FROM(from,eid)
  CK - from

- THANKS_MESSAGE(tid,mid)
  CK - tid

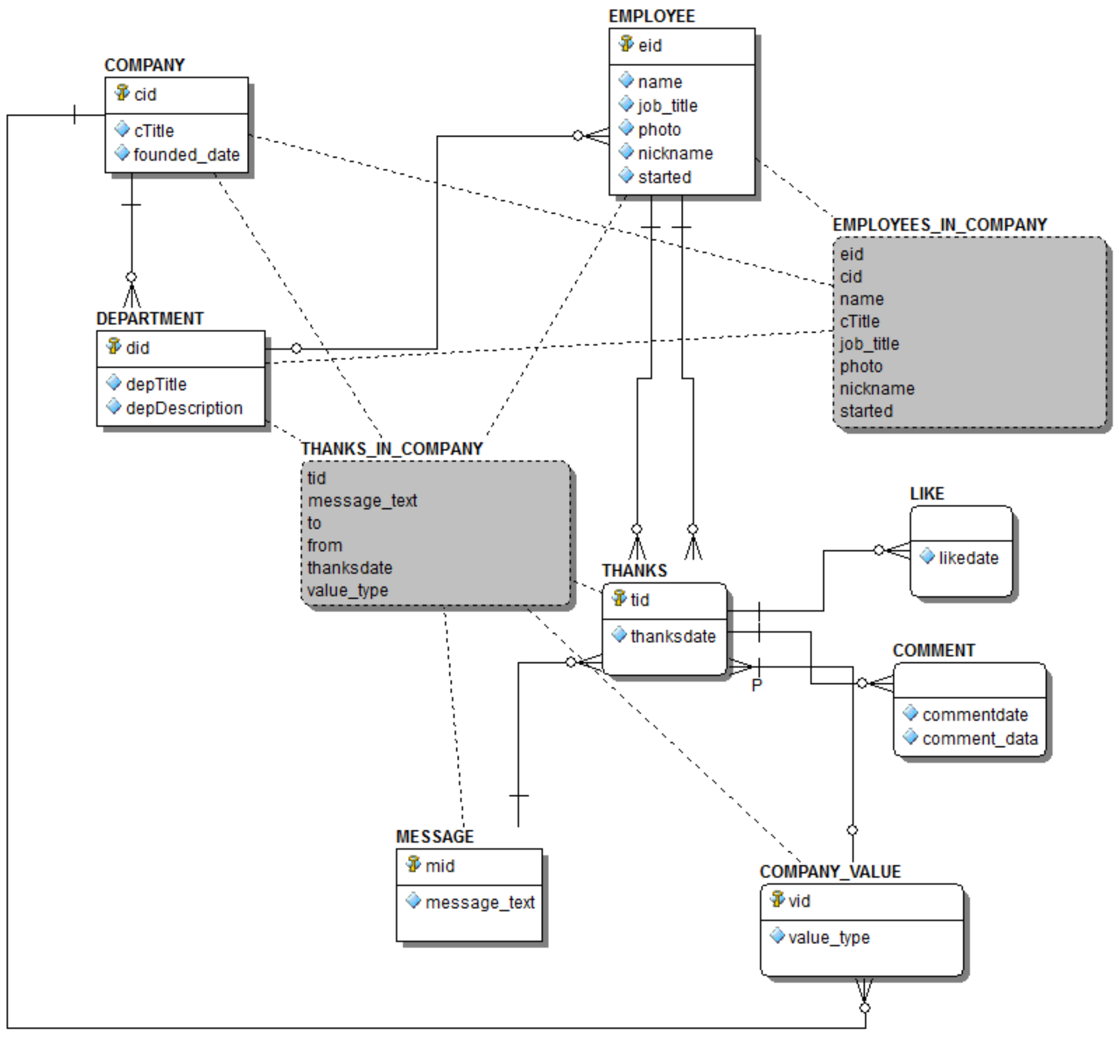
- THANKS_LIKE(likeid,tid)
  CK - likeid

- THANKS_COMMENT(commentid,tid)
  CK - commentid

- THANKS_VALUE(thanks.vid,company_value.vid)
  CK - thanks.vid

- VALUES_OF_COMPANY(cid, vid)
  CK - cid

### V.1.3 Entity-Relationship Diagram of the Enterprise



## V.2 Conceptual Model of the Enterprise

A conceptual model of the Thanks database.

company(cid, cTitle, founded_date)
PK - cid
CK - cid, cTitle
AK - cTitle

department(did, depTitle, depDescription)
PK - did
CK - did
FK - department.headid REFERENCES departmenthead.headid
department.cid REFERENCES company.cid

employee(eid, name, job_title, photo, nickname, started)
PK - eid
CK - eid, name, photo
AK - name, photo
FK - employee.did REFERENCES department.did

thanks((tid, mid), thanksdate)
PK - (tid, mid)
CK - tid
FK - thanks.mid REFERENCES message.mid
thanks.to REFERENCES employee.eid
thanks.from REFERENCES employee.eid
thanks.vid REFERENCES company_value.vid

like((likeid, tid), likedate)
PK - (likeid, tid)
CK - likeid
FK - like.tid REFERENCES thanks.tid

comment((commentid, tid), comment_data, commentdate)
PK - (commentid, tid)
CK - commentid
FK - comment.tid REFERENCES thanks.tid

message(mid, message_text)
PK - mid
CK - mid

company_value(vid, value_type)
PK - vid
CK - vid
FK - company_value.cid REFERENCES company.cid

## V.3 Table Dictionary

| Table | Attributes | Definition |
|---|---|---|
| COMPANY | cid, cTitle, founded_date | Represents a company, which is the starting data point |
| DEPARTMENT | did, depTitle, depDescription, headid, cid | Department in a company |
| EMPLOYEE | eid, name, job_title, photo, nickname, started, did | Employee in a company |
| THANKS | tid, thanksdate | Data type used to recognize another employee |
| LIKE | likeid, tid, likedate | A like on a Thanks post |
| COMMENT | commentid, tid, commentdate, comment_data | A comment on a Thanks post |
| MESSAGE | mid, message_text | Message of a Thanks |
| COMPANY_VALUE | vid, value_type | Company value of a Thanks |
| EMPLOYEES_IN_COMPANY | eid, cid, name, cTitle, job_title, photo, nickname, started | View of employees and their respective companies |
| THANKS_IN_COMPANY | tid, message_text, to, from, thanksdate, value_type | View of Thanks and their respective attributes |

## V.4 Attribute Dictionary

| Attribute | Tables Used In | Description |
|---|---|---|
| eid | EMPLOYEE, EMPLOYEES_IN_COMPANY | Unique identifier of an employee |
| name | EMPLOYEE, EMPLOYEES_IN_COMPANY | Name of an employee |
| job_title | EMPLOYEE, EMPLOYEES_IN_COMPANY | Job title of employee |
| photo | EMPLOYEE, EMPLOYEES_IN_COMPANY | Photo of employee |
| nickname | EMPLOYEE, EMPLOYEES_IN_COMPANY | Nickname of employee |
| started | EMPLOYEE, EMPLOYEES_IN_COMPANY | Date that the employee started at their company |
| did | DEPARTMENT | Unique identifier of the department of a company |
| depTitle | DEPARTMENT | Name (title) of the department |
| depDescription | DEPARTMENT | Description of the department |
| cid | COMPANY, EMPLOYEES_IN_COMPANY | Unique identifier of a company |
| cTitle | COMPANY, EMPLOYEES_IN_COMPANY | Name (title) of a company |
| founded_date | COMPANY | Date a company was founded |
| tid | THANKS, THANKS_IN_COMPANY | Unique identifier of a Thanks |
| thanksdate | THANKS, THANKS_IN_COMPANY | Date the Thanks was given |
| likeid | LIKE | Unique identifier of a like on a Thanks |
| likedate | LIKE | Date the like was given |
| commentid | COMMENT | Unique identifier of a comment |
| comment_data | COMMENT | Message of comment |
| commentdate | COMMENT | Date the comment was added to the Thanks |
| mid | MESSAGE | Unique identifier of the message of a Thanks |
| message_text | MESSAGE, THANKS_IN_COMPANY | Body text of the message of a Thanks |
| vid | COMPANY_VALUE | Unique identifier of a company value |
| value_type | COMPANY_VALUE, THANKS_IN_COMPANY | Name (type) of company value |

# Chapter VI

# Database and Query Definition

## VI.1 Database Definition

```
--
-- ER/Studio Data Architect 9.6 SQL Code Generation
-- Project :      ThanksCorp.DM1
--
-- Date Created : Tuesday, November 25, 2014 23:37:27
-- Target DBMS : MySQL 5.x
--

--
-- TABLE: COMMENT
--

CREATE TABLE COMMENT(
    commentid       INT             NOT NULL AUTO_INCREMENT,
    tid             INT             NOT NULL,
    commentdate     DATE            NOT NULL,
    comment_data    VARCHAR(255)    NOT NULL,
    PRIMARY KEY (commentid, tid)
)ENGINE=INNODB
;



--
-- TABLE: COMPANY
--

CREATE TABLE COMPANY(
    cid             INT             NOT NULL AUTO_INCREMENT,
    cTitle          VARCHAR(30)     NOT NULL,
    founded_date    DATE            NOT NULL,
    PRIMARY KEY (cid)
```

```
)ENGINE=INNODB
;



--
-- TABLE: COMPANY_VALUE
--

CREATE TABLE COMPANY_VALUE(
    vid           INT            NOT NULL AUTO_INCREMENT,
    cid           INT            NOT NULL,
    value_type    VARCHAR(30)    NOT NULL,
    PRIMARY KEY (vid, cid)
)ENGINE=INNODB
;



--
-- TABLE: DEPARTMENT
--

CREATE TABLE DEPARTMENT(
    did               INT             NOT NULL AUTO_INCREMENT,
    depTitle          VARCHAR(30)     NOT NULL,
    depDescription    VARCHAR(255),
    cid               INT             NOT NULL,
    PRIMARY KEY (did)
)ENGINE=INNODB
;



--
-- TABLE: EMPLOYEE
--
-- note: photo changed from BLOB to VARCHAR for ease of demo

CREATE TABLE EMPLOYEE(
    eid          INT            NOT NULL AUTO_INCREMENT,
    name         VARCHAR(30)    NOT NULL,
    job_title    VARCHAR(30)    NOT NULL,
    photo        VARCHAR(30),
    nickname     VARCHAR(30),
    started      DATE           NOT NULL,
    did          INT,
    PRIMARY KEY (eid)
```

```
)ENGINE=INNODB
;



--
-- TABLE: LIKE
--

CREATE TABLE ‘LIKE‘(
    likeid      INT            NOT NULL AUTO_INCREMENT,
    tid         INT            NOT NULL,
    likedate    DATETIME       NOT NULL,
    PRIMARY KEY (likeid, tid)
)ENGINE=INNODB
;



--
-- TABLE: MESSAGE
--

CREATE TABLE MESSAGE(
    mid             INT           NOT NULL AUTO_INCREMENT,
    message_text    VARCHAR(255),
    PRIMARY KEY (mid)
)ENGINE=INNODB
;



--
-- TABLE: THANKS
--

CREATE TABLE THANKS(
    tid           INT            NOT NULL AUTO_INCREMENT,
    mid           INT            NOT NULL,
    ‘to‘          INT            NOT NULL,
    ‘from‘        INT            NOT NULL,
    vid           INT,
    thanksdate    DATE           NOT NULL,
    cid           INT,
    PRIMARY KEY (tid, mid)
)ENGINE=INNODB
;
```

```
--
-- VIEW: EMPLOYEES_IN_COMPANY
--

CREATE VIEW EMPLOYEES_IN_COMPANY AS (
    SELECT e.eid
             , c.cid
             , e.name
             , c.cTitle
             , e.job_title
             , e.photo
             , e.nickname
             , e.started
    FROM COMPANY as c
            INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
            INNER JOIN EMPLOYEE AS e ON e.did = d.did
)
;

--
-- VIEW: THANKS_IN_COMPANY
--

CREATE VIEW THANKS_IN_COMPANY AS (
    SELECT t.tid
         , m.message_text
         , t.‘to‘
         , t.‘from‘
         , t.thanksdate
         , cv.value_type
    FROM COMPANY as c
        INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
        INNER JOIN EMPLOYEE AS e ON e.did = d.did
        INNER JOIN THANKS AS t ON t.to = e.eid
        INNER JOIN MESSAGE AS m ON t.mid = m.mid
        INNER JOIN COMPANY_VALUE as cv ON t.vid = cv.vid
)
;

--
-- TABLE: COMMENT
--

ALTER TABLE COMMENT ADD CONSTRAINT RefTHANKS20
    FOREIGN KEY (tid)
    REFERENCES THANKS(tid)
;
```

```
--
-- TABLE: COMPANY_VALUE
--

ALTER TABLE COMPANY_VALUE ADD CONSTRAINT RefCOMPANY33
    FOREIGN KEY (cid)
    REFERENCES COMPANY(cid)
;


--
-- TABLE: DEPARTMENT
--

ALTER TABLE DEPARTMENT ADD CONSTRAINT RefCOMPANY32
    FOREIGN KEY (cid)
    REFERENCES COMPANY(cid)
;


--
-- TABLE: EMPLOYEE
--

ALTER TABLE EMPLOYEE ADD CONSTRAINT RefDEPARTMENT30
    FOREIGN KEY (did)
    REFERENCES DEPARTMENT(did)
;


--
-- TABLE: LIKE
--

ALTER TABLE ‘LIKE‘ ADD CONSTRAINT RefTHANKS19
    FOREIGN KEY (tid)
    REFERENCES THANKS(tid)
;


--
-- TABLE: THANKS
--

ALTER TABLE THANKS ADD CONSTRAINT RefCOMPANY_VALUE17
    FOREIGN KEY (vid, cid)
    REFERENCES COMPANY_VALUE(vid, cid)
```

```
;

ALTER TABLE THANKS ADD CONSTRAINT RefEMPLOYEE22
    FOREIGN KEY (‘to‘)
    REFERENCES EMPLOYEE(eid)
;

ALTER TABLE THANKS ADD CONSTRAINT RefEMPLOYEE24
    FOREIGN KEY (‘from‘)
    REFERENCES EMPLOYEE(eid)
;

ALTER TABLE THANKS ADD CONSTRAINT RefMESSAGE31
    FOREIGN KEY (mid)
    REFERENCES MESSAGE(mid)
;
```

## VI.2 Database Queries

Given below are 14 example English queries, with their SQL DML used to retrieve the necessary data.

1. List the names of all employees in the company “I Love Thanks”.

```
SELECT name
FROM   EMPLOYEES_IN_COMPANY
WHERE  cTitle = "I Love Thanks"
;
```

2. Show all department names in the corporation “Blitz”.

```
SELECT d.depTitle
FROM COMPANY AS c
    INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
WHERE c.cTitle = "Blitz"
;
```

3. Return a list of all employees in the database who have nicknames.

```
SELECT e.nickname
     , e.name
FROM EMPLOYEE AS e
WHERE e.nickname IS NOT NULL
;
```

4. Show names of employees who do not have photos in the “I Love Thanks” company.

```
SELECT e.name
FROM COMPANY AS c
    INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
    INNER JOIN EMPLOYEE   AS e ON d.did = e.did
WHERE e.photo IS NULL
    AND c.cTitle = "I Love Thanks"
;
```

5. Show all comments on Julia Crow’s Thanks to Richard Baker on January 2, 2011.

```
SELECT commentdate
     , comment_data
FROM COMMENT
WHERE tid = (
    SELECT t.tid
    FROM THANKS_IN_COMPANY AS t
    WHERE t.to = (
        SELECT eid
        FROM   EMPLOYEES_IN_COMPANY
        WHERE  name = "Richard Baker"
    )
    AND t.from = (
        SELECT eid
        FROM   EMPLOYEES_IN_COMPANY
        WHERE  name = "Julia Crow"
    )
    AND t.thanksdate = "2011-1-2"
)
;
```

6. Show all of the Thanks that Julia Crow from “First America” gave.

```
SELECT thanks.‘from‘
     , addressed_to.name as ‘to‘
     , thanks.message
     , thanks.thanksdate
     , thanks.value
FROM (
    SELECT e.name AS ‘from‘
         , t.‘to‘
         , m.message_text AS message
         , t.thanksdate
         , cv.value_type AS value
    FROM COMPANY AS c
```

```
        INNER JOIN DEPARTMENT    AS d  ON c.cid    = d.cid
        INNER JOIN EMPLOYEE      AS e  ON d.did    = e.did
        INNER JOIN THANKS        AS t  ON t.‘from‘ = e.eid
        INNER JOIN MESSAGE       AS m  ON t.mid    = m.mid
        INNER JOIN COMPANY_VALUE AS cv ON t.vid    = cv.vid
    WHERE c.cTitle = "First America"
        AND e.name = "Julia Crow"
) AS thanks
, EMPLOYEE AS addressed_to
WHERE addressed_to.eid = thanks.‘to‘
;
```

7. Show all of the Thanks that Emma Cross from “Lightning Corporation” received.

```
SELECT thanks.‘to‘
     , addressed_to.name AS ‘from‘
     , thanks.message
     , thanks.thanksdate
     , thanks.value
FROM (
    SELECT e.name as ‘to‘
         , t.‘from‘
         , m.message_text AS message
         , t.thanksdate
         , cv.value_type AS value
    FROM COMPANY AS c
        INNER JOIN DEPARTMENT    AS d  ON c.cid  = d.cid
        INNER JOIN EMPLOYEE      AS e  ON d.did  = e.did
        INNER JOIN THANKS        AS t  ON t.‘to‘ = e.eid
        INNER JOIN MESSAGE       AS m  ON t.mid  = m.mid
        INNER JOIN COMPANY_VALUE AS cv ON t.vid  = cv.vid
    WHERE c.cTitle = "Lightning Corporation"
        AND e.name = "Emma Cross"
) AS thanks
, EMPLOYEE as addressed_to
WHERE addressed_to.eid = thanks.‘from‘
;
```

8. Show all thanks that have been given in the corporation “First America” before 2013.

```
SELECT t.to
     , t.from
     , m.message_text
     , t.thanksdate
     , cv.value_type
FROM COMPANY AS c
```

```
    INNER JOIN DEPARTMENT    AS d  ON c.cid  = d.cid
    INNER JOIN EMPLOYEE      AS e  ON d.did  = e.did
    INNER JOIN THANKS        AS t  ON t.‘to‘ = e.eid
    INNER JOIN COMPANY_VALUE AS cv ON t.vid  = cv.vid
    INNER JOIN MESSAGE       AS m  ON t.mid  = m.mid
WHERE c.cTitle = "First America"
    AND t.thanksdate < ’2013-1-1’
;
```

9. List the company values of “Blitz”.

```
SELECT cv.value_type
FROM   COMPANY AS c
    INNER JOIN COMPANY_VALUE as cv on c.cid = cv.cid
WHERE  c.cTitle = "Blitz"
;
```

10. Who is the newest employee to have joined the company “Lightning Corporation”?

```
SELECT e.name
FROM COMPANY AS c
    INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
    INNER JOIN EMPLOYEE   AS e ON d.did = e.did
WHERE c.cTitle = "Lightning Corporation"
    AND e.started = (
        SELECT MAX(e.started)
        FROM COMPANY AS c
            INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
            INNER JOIN EMPLOYEE   AS e ON d.did = e.did
        WHERE c.cTitle = "Lightning Corporation"
    )
;
```

11. List all Thanks “I Love Thanks” employees gave in October 2011.

```
SELECT t.to
     , t.from
     , m.message_text
     , t.thanksdate
     , cv.value_type
FROM COMPANY AS c
    INNER JOIN DEPARTMENT    AS d  ON c.cid  = d.cid
    INNER JOIN EMPLOYEE      AS e  ON d.did  = e.did
    INNER JOIN THANKS        AS t  ON t.‘to‘ = e.eid
    INNER JOIN COMPANY_VALUE AS cv ON t.vid  = cv.vid
    INNER JOIN MESSAGE       AS m  ON t.mid  = m.mid
WHERE MONTHNAME(t.thanksdate) = "October"
    AND YEAR(t.thanksdate) = 2011
;
```

12. Find the name of the employee who received the most Thanks in “Blitz”.

```
SELECT e.name
FROM COMPANY AS c
    INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
    INNER JOIN EMPLOYEE   AS e ON d.did = e.did
    INNER JOIN THANKS     AS t ON t.to  = e.eid
WHERE c.cTitle = "Blitz"
    AND t.to = (
        SELECT ‘to‘ FROM (
            SELECT ‘to‘, count(‘to‘) AS counted
            FROM COMPANY AS c
                INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
                INNER JOIN EMPLOYEE   AS e ON d.did = e.did
                INNER JOIN THANKS     AS t ON t.to  = e.eid
            WHERE c.cTitle = "Blitz"
            GROUP BY ‘to‘
            ORDER BY counted DESC
            LIMIT 1
        ) as received_most_thanks
    )
    GROUP BY ‘to‘
;
```

13. Find the average number of likes each Thanks in the corporation “Blitz” received in 2011.

```
SELECT AVG(avg.counter) as average
FROM (
    SELECT l.tid, count(l.likeid) as counter
    FROM COMPANY AS c
        INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
        INNER JOIN EMPLOYEE   AS e ON d.did = e.did
        INNER JOIN THANKS     AS t ON t.to  = e.eid
        INNER JOIN ‘LIKE‘     AS l ON t.tid = l.tid
    WHERE c.cTitle = "Blitz"
    AND   YEAR(t.thanksdate) = 2011
    GROUP BY l.tid
) AS avg
;
```

14. Find the most awarded company value in “Blitz”.

```
SELECT cv.value_type
FROM COMPANY AS c
    INNER JOIN DEPARTMENT    AS d  ON c.cid = d.cid
    INNER JOIN EMPLOYEE      AS e  ON d.did = e.did
    INNER JOIN THANKS        AS t  ON e.eid = t.to
    INNER JOIN COMPANY_VALUE AS cv ON t.vid = cv.vid
WHERE c.cTitle = "Blitz"
GROUP BY value_type
ORDER BY COUNT(VALUE_TYPE) DESC
LIMIT 1
;
```

## VI.3 Design Tradeoffs and Limitations

One of the biggest design tradeoffs is that an employee can only be associated with one company. This can become difficult if an employee leaves a company and wants to take their data with them to a new company. An example of this can be seen more clearly when the attribute “started” is taken into account. If an employee is starting in a new position, this date has to change and is no longer preserved. However, since the idea of Thanks Corporation is to maintain a private way to share Thanks within a company, this seemed to be a fair tradeoff. The “started” example is a more difficult limitation of the database, but necessary without another entity.

Another tradeoff of the database is the lack of multiple job titles. Multiple photos are also not allowed, so if transferring between companies were ever a feature in the future, these would not work in a future implementation. Another design limitation is the lack of Thanks being given between corporations. Since I wanted to keep the database as small communities of corporations rather than one large social network, it seemed to be the right tradeoff.

Chapter VII

# Database Integrity and Security

## VII.1 Functional Dependencies

- cid → cTitle, founded_date
- did → depTitle, depDescription
- eid → name, job_title, photo, nickname, started
- tid → thanksdate
- likeid → likedate
- commentid → comment_data, commentdate
- mid → message_text
- vid → value_type

## VII.2 Adjustments for Normalization

According to these functional dependencies and after looking at the ERD, it is clear that the database has already been normalized. Therefore, no adjustments will be needed on top of what already is in normalizing this database.

## VII.3 Integrity and Security

1. Users and Granted Privileges
   There will multiple types of access privileges to the database. These access types are listed below.

   - Webmaster User
     A webmaster represents access privileges for the owner of Thanks Corporation. He has complete access to all data at any time.
   - Executive User
     Allows access to an executive's own company entity, and allows for modification of its attributes.

- Department Head User
  Allows access to a department head's own department entity, and allows for modification of its attributes.
- Employee User
  Allows normal access for an employee: modification of own employee entity and insert/update/delete privileges for own Thanks and related entities.

2. Assertion
   MySQL does not directly provide assertions, however triggers have been used to provide assertion-like functionality in MySQL. Assertions the Thanks Corporation needs are listed below.

   - Determine whether an employee giving a Thanks with a company value is using a company value from the employee's own company.

3. Views
   Two views are utilized in this database.

   - EMPLOYEES_IN_COMPANY
     Returns employees from all departments associated with a company.
   - THANKS_IN_COMPANY
     Returns all thanks from within a company.

# Chapter VIII

# Implementation Notes

## VIII.1 Indices

Refer to section IV.1 for a list of used indices.

## VIII.2 Data

```
--
-- DATA LOAD for Thanks Corporation
--

CREATE TABLE IF NOT EXISTS COMPANY(
    cid             INT             NOT NULL AUTO_INCREMENT,
    cTitle          VARCHAR(30)     NOT NULL,
    founded_date    DATE            NOT NULL,
    PRIMARY KEY (cid)
)ENGINE=INNODB
;

INSERT INTO COMPANY (cTitle, founded_date) VALUES
("Blitz", "2001-8-31"),
("I Love Thanks", "2005-8-15"),
("Playa Medical", "1930-5-2"),
("First America", "1993-7-17"),
("Lightning Corporation", "2001-9-12")
;


CREATE TABLE IF NOT EXISTS COMPANY_VALUE(
    vid           INT             NOT NULL AUTO_INCREMENT,
    cid           INT             NOT NULL,
    value_type    VARCHAR(30)     NOT NULL,
    PRIMARY KEY (vid, cid)
)ENGINE=INNODB
;
```

```
INSERT INTO COMPANY_VALUE(cid, value_type) VALUES
(1, "Integrity"),
(1, "Creativity"),
(1, "Ingenuity"),
(1, "Success"),
(2, "Helpfulness"),
(2, "Thankfulness"),
(2, "Caring"),
(2, "Focused"),
(3, "Helpful"),
(3, "Serving"),
(3, "Invaluable"),
(3, "Medical Star"),
(4, "Hardworking"),
(4, "Encouraging"),
(4, "Strategic"),
(4, "Successful"),
(5, "Fast"),
(5, "Comprehensive"),
(5, "Timely"),
(5, "Innovative")
;

CREATE TABLE IF NOT EXISTS DEPARTMENT(
    did               INT             NOT NULL AUTO_INCREMENT,
    depTitle          VARCHAR(30)     NOT NULL,
    depDescription    VARCHAR(255),
    cid               INT             NOT NULL,
    PRIMARY KEY (did)
)ENGINE=INNODB
;

INSERT INTO DEPARTMENT(depTitle, depDescription, cid) VALUES
("Administration", "The executive branch", 1),
("Technology", "We build the apps for Blitz.", 1),
("Creative", "The brain of the operation", 1),
("Communications", "We communicate well with others.", 1),
("Thanks1", "Thanks Department 1", 2),
("Thanks2", "Thanks Department 2", 2),
("Thanks3", "Thanks Department 3", 2),
("Thanks4", "Thanks Department 4", 2),
("Board", "Board of the Hospital", 3),
("Outpatient", "Outpatient Team", 3),
("Inpatient", "Inpatient Team", 3),
("ER", "Emergency Room Team", 3),
("Car Loan", "Giving people wheels.", 4),
("Home Loan", "Giving people homes.", 4),
("Banking", "Banking section of First America", 4),
("Mutual Funds", "Mutual Funds of First America", 4),
("Electricity", "We power this company.", 5),
("Human Relations", "HR Department of Lightning Corporation", 5),
("Power", "We humbly thank Electricity", 5),
("Thunder", "Lightning's Best Friend", 5)
;

INSERT INTO DEPARTMENT(depTitle, cid) VALUES
("Strategy", 1),
("Thanks5", 2),
("Cancer Center", 3),
("Administration", 4),
("Lightning Execs", 5)
;

CREATE TABLE IF NOT EXISTS EMPLOYEE(
    eid          INT            NOT NULL AUTO_INCREMENT,
    name         VARCHAR(30)    NOT NULL,
    job_title    VARCHAR(30)    NOT NULL,
    photo        VARCHAR(30),
    nickname     VARCHAR(30),
    started      DATE           NOT NULL,
    did          INT,
    PRIMARY KEY (eid)
)ENGINE=INNODB
;

INSERT INTO EMPLOYEE(name, job_title, photo, nickname, started, did) VALUES
("William Smith", "CEO", "mountain tops", "Will", "2013-1-1", 1),
("Florence Johnson", "Department Head", "mountain tops", "Flo", "2010-4-6", 5),
("George Williams", "Finance", "mountain tops", "GW", "2008-9-15", 9),
("Thomas Brown", "Finance Manager", "mountain tops", "Tom", "2005-3-10", 13),
("Annie Jones", "Lead Engineer", "mountain tops", "Ann", "2013-1-1", 17)
;

INSERT INTO EMPLOYEE(name, job_title, photo, started, did) VALUES
("Frederick Martinez", "President", "mountain tops", "2008-9-15", 1),
("Elsie Anderson", "Assistant Director", "mountain tops", "2013-1-1", 5),
("Charles Taylor", "Public Relations", "mountain tops", "2005-3-10", 9),
("Dorothy Thomas", "Finance Associate", "mountain tops", "2013-1-1", 13),
("Albert Hernandez", "Electric Manager", "mountain tops", "1993-7-17", 17),
("Ethel Moore", "Senior Technology Officer", "mountain tops", "2010-2-9", 2),
("Robert Martin", "Department Head", "mountain tops", "2010-2-11", 6),
("Doris Jackson", "Head Nurse", "mountain tops", "2011-3-15", 10),
("Joseph Thompson", "Financial Manager", "mountain tops", "2008-11-15", 14),
("Margaret White", "HR Manager", "mountain tops", "2011-3-15", 18),
("Alfred Lopez", "Creative Director", "mountain tops", "2010-4-6", 3),
("Gladys Lee", "Department Head", "mountain tops", "2008-11-15", 7),
("Henry Gonzales", "Inpatient Manager", "mountain tops", "2013-1-1", 11),
("Sarah Harris", "Funds Manager", "mountain tops", "2008-9-15", 15),
("Ernest Clark", "Power Manager", "mountain tops", "2011-3-15", 19),
("Lillian Lewis", "Communications Manager", "mountain tops", "1993-7-17", 4),
("Harry Robinson", "Department Head", "mountain tops", "2013-1-1", 8),
("Ellen Walker", "MD", "mountain tops", "2008-9-15", 12),
("Harold Perez", "Mutual Fund Manager", "mountain tops", "2013-1-1", 16),
("Hilda Hall", "Thunder Manager", "mountain tops", "2014-5-6", 20),
("Edward Young", "Strategy Consultant", "mountain tops", "2010-4-6", 21),
("Lily Allen", "Department Head", "mountain tops", "2013-1-1", 22),
("Walter Sanchez", "Lead MD", "mountain tops", "1993-7-17", 23),
("Frank Wright", "Chief Financial Officer", "mountain tops", "2013-1-1", 24),
("Violet King", "CEO", "mountain tops", "2008-9-15", 25)
;

INSERT INTO EMPLOYEE(name, job_title, nickname, started, did) VALUES
("Henry Gonzales", "Strategy Consultant", "H", "2010-4-6", 21),
("Herbert Scott", "Thanks Assistant", "Herb", "2011-3-15", 22),
("Ada Green", "Assistant MD", "A", "2011-3-10", 23),
("Richard Baker", "Chief of Operations", "Rich", "2013-1-1", 24),
("Emily Adams", "Lightning Board Member", "Emma", "2005-3-10", 25)
;

INSERT INTO EMPLOYEE(name, job_title, started, did) VALUES
("May Roberts", "HR Manager", "2010-2-11", 1),
("Samuel Carter", "Thanks Associate", "2008-11-15", 5),
("Mabel Phillips", "Company Effectiveness Officer", "2013-1-1", 9),
("David Evans", "Loan Associate", "1993-7-17", 13),
("Ivy Turner", "Electrician", "2011-3-10", 17),
("Sidney Torres", "Senior Developer", "2010-4-6", 2),
("Rose Parker", "Thanks Worker", "2011-3-10", 6),
("Francis Collins", "Outpatient Desk Manager", "2008-9-15", 10),
("Gertrude Edwards", "Home Loan Associate", "2013-1-1", 14),
("Edward Bisiani", "HR Associate", "2014-1-1", 18),
("Stanley Stewart", "Main Strategy Officer", "2010-2-11", 21),
("Jane Flores", "Thanks Associate", "2005-3-10", 22),
("Fred Morris", "Front Desk", "2007-10-10", 23),
("Julia Crow", "Assistant CEO", "1993-7-17", 24),
("Emma Cross", "Executive Lightning Specialist", "2013-1-1", 25)
;

CREATE TABLE IF NOT EXISTS MESSAGE(
    mid             INT          NOT NULL AUTO_INCREMENT,
    message_text    VARCHAR(255),
    PRIMARY KEY (mid)
)ENGINE=INNODB
;
```

```
INSERT INTO MESSAGE(message_text) VALUES
("GREAT JOB!"),
("Keep up the good work!"),
("You really followed instructions well"),
("I’m very impressed with your work"),
("Keep it up!"),
("You have done a service for the company"),
("Keep working hard!"),
("Nice work"),
("You really helped me out"),
("Thanks for working with my team!"),
("You are really creative"),
("Good work"),
("Good work, keep it up!"),
("Nice work!"),
("Quickly finished this project"),
("Great job on the project!"),
("You collaborate well!"),
("You’re a good employee and friend."),
("Thank you so much!"),
("You did awesome work on that project"),
("You’re the MVP of this company"),
("Good at keeping pace and a pleasure to work with"),
("Thank you so much for your continuous hard work!"),
("Thanks again for the effort!"),
("Thoroughly impressed with your work.")
;

CREATE TABLE IF NOT EXISTS THANKS(
    tid           INT           NOT NULL AUTO_INCREMENT,
    mid           INT           NOT NULL,
    ‘to‘          INT           NOT NULL,
    ‘from‘        INT           NOT NULL,
    vid           INT,
    thanksdate    DATETIME      NOT NULL,
    cid           INT,
    PRIMARY KEY (tid, mid)
)ENGINE=INNODB
;

INSERT INTO THANKS(mid, ‘to‘, ‘from‘, vid, thanksdate, cid) VALUES
(1, 50, 25, 17, "2010-1-2", 5),
(2, 50, 10, 17, "2010-5-6", 5),
(3, 50, 45, 18, "2010-11-1", 5),
(4, 34, 49, 14, "2011-1-2", 4),
(5, 24, 49, 15, "2012-1-1", 4),
(6, 39, 19, 16, "2012-2-4", 4),
(7, 4, 9, 13, "2012-12-31", 4),
(8, 2, 12, 5, "2011-10-1", 2),
(9, 12, 2, 6, "2011-10-2", 2),
(10, 1, 6, 1, "2013-2-4", 1),
(11, 1, 36, 1, "2014-1-1", 1),
(12, 1, 41, 1, "2012-2-2", 1),
(13, 41, 36, 1, "2014-2-3", 1),
(14, 6, 1, 1, "2010-1-2", 1),
(15, 11, 1, 2, "2010-2-10", 1),
(16, 11, 16, 3, "2009-1-3", 1),
(17, 11, 21, 4, "2008-5-19", 1),
(18, 11, 11, 2, "2010-12-31", 1),
(19, 6, 16, 2, "2011-1-1", 1),
(20, 16, 6, 3, "2011-1-2", 1),
(21, 36, 41, 4, "2011-1-3", 1),
(22, 16, 11, 2, "2011-1-4", 1),
(23, 38, 22, 9, "2011-1-5", 3),
(24, 18, 38, 10, "2011-1-6", 3),
(25, 8, 18, 11, "2010-2-2", 3)
;

CREATE TABLE IF NOT EXISTS ‘LIKE‘(
    likeid      INT            NOT NULL AUTO_INCREMENT,
    tid         INT            NOT NULL,
    likedate    DATETIME       NOT NULL,
    PRIMARY KEY (likeid, tid)
)ENGINE=INNODB
;

INSERT INTO ‘LIKE‘(tid, likedate) VALUES
(19, "2011-2-4"),
(19, "2011-2-10"),
(19, "2011-2-23"),
(19, "2011-3-1"),
(20, "2011-6-21"),
(20, "2011-8-21"),
(21, "2011-9-1"),
(21, "2011-9-11"),
(21, "2011-11-2"),
(1, "2013-1-2"),
(5, "2014-5-6"),
(7, "2013-2-3"),
(8, "2014-8-9"),
(8, "2011-10-1"),
(10, "2013-2-4")
;

CREATE TABLE IF NOT EXISTS COMMENT(
    commentid       INT            NOT NULL AUTO_INCREMENT,
    tid             INT            NOT NULL,
    commentdate     DATE           NOT NULL,
    comment_data    VARCHAR(255)   NOT NULL,
    PRIMARY KEY (commentid, tid)
)ENGINE=INNODB
;

INSERT INTO COMMENT(tid, commentdate, comment_data) VALUES
(4, "2011-1-2", "I totally agree!"),
(4, "2011-1-2", "Such kind words"),
(4, "2011-1-3", "Yeah! You go!"),
(10, "2013-5-6", "This was a thoughtful thing to say."),
(10, "2013-5-7", "I second this, keep it up"),
(14, "2010-1-2", "You’re so nice!"),
(15, "2010-2-10", "Keep it up"),
(16, "2009-1-3", "So much encouragement!"),
(17, "2008-5-19", "Nice!")
;
```

## VIII.3 Query Trace

A query trace is found on the next page.

Your SQL query has been executed successfully

```sql
-- List the names of all employees in the company "I Love Thanks".
SELECT name
FROM EMPLOYEES_IN_COMPANY
WHERE cTitle = "I Love Thanks"
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| name |
|---|
| Florence Johnson |
| Elsie Anderson |
| Samuel Carter |
| Robert Martin |
| Rose Parker |
| Gladys Lee |
| Harry Robinson |
| Lily Allen |
| Herbert Scott |
| Jane Flores |

Your SQL query has been executed successfully

```sql
-- Show all department names in the corporation "Blitz".
SELECT d.depTitle
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
WHERE c.cTitle = "Blitz"
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| depTitle |
|---|
| Administration |

| depTitle |
|---|
| Technology |
| Creative |
| Communications |
| Strategy |

Your SQL query has been executed successfully

```
-- Return a list of all employees in the database who have nicknames.
SELECT e.nickname, e.name
FROM EMPLOYEE AS e
WHERE e.nickname IS NOT NULL
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| nickname | name |
|---|---|
| Will | William Smith |
| Flo | Florence Johnson |
| GW | George Williams |
| Tom | Thomas Brown |
| Ann | Annie Jones |
| H | Henry Gonzales |
| Herb | Herbert Scott |
| A | Ada Green |
| Rich | Richard Baker |
| Emma | Emily Adams |

Your SQL query has been executed successfully

```sql
-- Show names of employees who do not have photos in the "I Love Thanks"
company.
SELECT e.name
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
WHERE e.photo IS NULL
AND c.cTitle = "I Love Thanks"
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| name |
|---|
| Samuel Carter |
| Rose Parker |
| Herbert Scott |
| Jane Flores |

Your SQL query has been executed successfully

```sql
-- Show all comments on Julia Crow's Thanks to Richard Baker on January 2,
2011.
SELECT commentdate, comment_data
FROM COMMENT
WHERE tid = (
SELECT t.tid
FROM THANKS_IN_COMPANY AS t
WHERE t.to = (
SELECT eid
FROM EMPLOYEES_IN_COMPANY
WHERE name = "Richard Baker" )
AND t.from = (
SELECT eid
FROM EMPLOYEES_IN_COMPANY
WHERE name = "Julia Crow" )
AND t.thanksdate = "2011-1-2" )
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| commentdate | comment_data |
|---|---|
| 2011-01-02 | I totally agree! |

| commentdate | comment_data |
|---|---|
| 2011-01-02 | Such kind words |
| 2011-01-03 | Yeah! You go! |

Your SQL query has been executed successfully

```
-- Show all of the Thanks that Julia Crow from "First America" gave.
SELECT e.name, t.`to` , m.message_text
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.`from` = e.eid
INNER JOIN MESSAGE AS m ON t.mid = m.mid
WHERE c.cTitle = "First America"
AND e.name = "Julia Crow"
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| name | to | message_text |
|---|---|---|
| Julia Crow | 34 | I'm very impressed with your work |
| Julia Crow | 24 | Keep it up! |

Your SQL query has been executed successfully

```
-- Show all of the Thanks that Emma Cross from "Lightning Corporation"
received.
SELECT e.name, t.`from` , m.message_text
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.`to` = e.eid
INNER JOIN MESSAGE AS m ON t.mid = m.mid
WHERE c.cTitle = "Lightning Corporation"
AND e.name = "Emma Cross"
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| name | from | message_text |
|---|---|---|
| Emma Cross | 25 | GREAT JOB! |

| name | from | message_text |
|---|---|---|
| Emma Cross | 10 | Keep up the good work! |
| Emma Cross | 45 | You really followed instructions well |

Your SQL query has been executed successfully

```
-- Show all thanks that have been given in the corporation "First America" before
2013.
SELECT t.to, t.from, m.message_text, t.thanksdate, cv.value_type
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.`to` = e.eid
INNER JOIN COMPANY_VALUE AS cv ON t.vid = cv.vid
INNER JOIN MESSAGE AS m ON t.mid = m.mid
WHERE c.cTitle = "First America"
AND t.thanksdate < '2013-1-1'
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| to | from | message_text | thanksdate | value_type |
|---|---|---|---|---|
| 4 | 9 | Keep working hard! | 2012-12-31 00:00:00 | Hardworking |
| 39 | 19 | You have done a service for the company | 2012-02-04 00:00:00 | Successful |
| 24 | 49 | Keep it up! | 2012-01-01 00:00:00 | Strategic |
| 34 | 49 | I'm very impressed with your work | 2011-01-02 00:00:00 | Encouraging |

Your SQL query has been executed successfully

```
-- List the company values of "Blitz".
SELECT cv.value_type
FROM COMPANY AS c
INNER JOIN COMPANY_VALUE AS cv ON c.cid = cv.cid
WHERE c.cTitle = "Blitz"
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

**value_type**

| value_type |
|---|
| Integrity |
| Creativity |
| Ingenuity |
| Success |

Your SQL query has been executed successfully

```
-- Who is the newest employee to have joined the company "Lightning
Corporation"?
SELECT e.name
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
WHERE c.cTitle = "Lightning Corporation"
AND e.started = (
SELECT MAX( e.started )
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
WHERE c.cTitle = "Lightning Corporation" )
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| name |
|---|
| Hilda Hall |

Your SQL query has been executed successfully

```
-- List all Thanks "I Love Thanks" employees gave in October 2011.
SELECT t.to, t.from, m.message_text, t.thanksdate, cv.value_type
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.`to` = e.eid
INNER JOIN COMPANY_VALUE AS cv ON t.vid = cv.vid
INNER JOIN MESSAGE AS m ON t.mid = m.mid
WHERE MONTHNAME( t.thanksdate ) = "October"
AND YEAR( t.thanksdate ) =2011
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| to | from | message_text | thanksdate | value_type |
|---|---|---|---|---|
| 2 | 12 | Nice work | 2011-10-01 00:00:00 | Helpfulness |
| 12 | 2 | You really helped me out | 2011-10-02 00:00:00 | Thankfulness |

Your SQL query has been executed successfully

```
-- Find the name of the employee who received the most Thanks in "Blitz".
SELECT e.name
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.to = e.eid
WHERE c.cTitle = "Blitz"
AND t.to = (
SELECT `to`
FROM (

SELECT `to` , COUNT( `to` ) AS counted
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.to = e.eid
WHERE c.cTitle = "Blitz"
GROUP BY `to`
ORDER BY counted DESC
LIMIT 1 ) AS received_most_thanks
)
GROUP BY `to`
LIMIT 0 , 30
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| name |
|---|
| Ethel Moore |

Your SQL query has been executed successfully

```sql
-- Find the average number of likes each Thanks in the corporation "Blitz"
received in 2011.
SELECT AVG( avg.counter )
FROM (

SELECT l.tid, COUNT( l.likeid ) AS counter
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON t.to = e.eid
INNER JOIN `LIKE` AS l ON t.tid = l.tid
WHERE c.cTitle = "Blitz"
AND YEAR( t.thanksdate ) =2011
GROUP BY l.tid
) AS avg
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| AVG(avg.counter) |
|---|
| 3.0000 |

Your SQL query has been executed successfully

```sql
-- Find the most awarded company value in "Blitz".
SELECT cv.value_type
FROM COMPANY AS c
INNER JOIN DEPARTMENT AS d ON c.cid = d.cid
INNER JOIN EMPLOYEE AS e ON d.did = e.did
INNER JOIN THANKS AS t ON e.eid = t.to
INNER JOIN COMPANY_VALUE AS cv ON t.vid = cv.vid
WHERE c.cTitle = "Blitz"
GROUP BY value_type
ORDER BY COUNT( VALUE_TYPE ) DESC
LIMIT 1
```

[ Inline ] [ Edit ] [ Create PHP Code ]

| value_type |
|---|
| Integrity |

## VIII.4 Implementation Assessment

I had to adjust some of the data types, as my primary keys were not autoincrementing when I inserted new data. I also switched datetime to date for a few data types in order to maintain some consistency. Finally, I modified a database relation since it was conflicting with what I wanted. Overall, after those fixes all of the implementations went smoothly and I was able to get my query trace up and running quickly.

# Chapter IX

# Lessons Learned

One thing I would do differently if I approached the project again would be to put more effort into concepting the initial idea. Instead of depending on revisions in order to hash out a good database design, I should have made an effort to expand my knowledge and find good examples of how to represent my entities and relationships before assuming they were right. While I was able to come up with a solid database design, it ended up taking much longer and required a few hours extra spent on database design later on in the project. I would also start the project working in LaTeX and ER Studio from the beginning. I ended up dividing the information in my deliverables, and towards the fourth deliverable it became difficult to manage all of the different sections and edit my information accordingly. In regards to ER Studio, I eventually had to transcribe my entity-relationship diagram anyways in order to cut down on the time needed to write the SQL code. It also provided a nice diagram for my project section, which I would have had to hand-draw otherwise. While these two things were extremely difficult in the beginning of the project, by the time I reached the fifth and sixth deliverable I found my adjustments to be way less challenging, especially with the project all in one document.

For time spent on the project, I probably spent somewhere between fifty and sixty hours on the project. As I mentioned in my difficulties in the project, I also ended up migrating the database project into LaTeX. This took much longer than expected due to my lack of experience and lack of examples, so approximately five of these hours were spent in formatting alone. Additional time beyond this initial setup was spent in figuring out table formatting and resizing. However, I found this to be time well spent. Moving to LaTeX allowed me to improve my workflow in later deliverables, and helped me worry less about my formatting. Overall, I believe I spent a fair amount of time on the project considering the constraints of time from other classes, such as the Group Senior Project course.

The project was useful in helping me learn the course material in a few ways. Writing the database queries necessary for my database improved my SQL query knowledge. As for completing the project and turning it into a production environment, several things need to be considered. The lack of true assertions in MySQL make it difficult to validate data, which requires a great deal of validation work to be done server side before the INSERT queries are made. In addition, there are some tradeoffs to the database design, such as employees being exclusive to one company, which would complicate things for a server and client when an employee moves to a different company. Finally, adding more views would make it easier to access the necessary data a client needs to display information for this corporation.

Overall, I learned a lot through the project and had a great time realizing Thanks Corporation in MySQL.