IFT 530 Advanced SQL

Team Elite Techies

Ramya Varakantham

Vipul Sarin

Nupur Bhargava

Arizona State University

Submission on: 04/02/2016

**Create a data model for an Online Marketplace for Services:**

* **Person (User) data**
* **New Request data**
* **Pending Request data**
* **Completed Request data**
* **Service data**
* **Category data**
* **Message data**

**Create attributes (four or more) for each entity. Using your choice of drawing tool/platform, create the relationships between the entities, name the relationships, and create an entity­relationship diagram for the store.**

This project deals with creating a database to support an online marketplace for services. The team looked at key aspects of the website [www.fiverr.com](http://www.fiverr.com). We recognized various entities from this application, modeled appropriate relationships between them and created an Entity-Relationship diagram as well as a Partial Conceptual Model.

**List of all Entities, Primary Keys and Foreign Keys:**

|  |  |  |
| --- | --- | --- |
| **Entity** | **Primary Key** | **Foreign Key** |
| Person | Email\_id |  |
| Message | Message\_id | Author\_id, Request\_id |
| New\_Request | Request\_id | Email\_id, category\_id |
| Pending\_Request | Pending\_request\_id | Requestor\_id, provider\_id, category\_id |
| Category | Category\_id |  |
| Completed\_Request | Completed\_request\_id | Requestor\_id, provider\_id, category\_id |
| Service | Service\_id | Email\_id, category\_id |
| Placeholder | Placeholder\_id |  |
| Person\_Message | Email\_id, Message\_id | Email\_id, Message\_id, placeholder\_id |
| Person\_Category | Email\_id, category\_id | Email\_id, category\_id |

**Relationship between different entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Entity1** | **Entity2** | **Relationship Degree** |
| 1 | Person | New\_Request | 1:M |
| 2 | Person | Pending\_Request | 1:M |
| 3 | Person | Completed\_Request | 1:M |
| 4 | Person | Service | 1:M |
| 5 | Person | Person\_Category | 1:M |
| 6 | Person | Person\_Message | 1:M |
| 7 | Person | Message | 1:M |
| 8 | Person\_Category | Category | M:1 |
| 9 | Category | New\_Request | 1:1 |
| 10 | Category | Pending\_Request | 1:1 |
| 11 | Category | Completed\_Request | 1:1 |
| 12 | Category | Service | 1:1 |
| 13 | Message | Person\_Message | 1:M |
| 14 | Message | New\_Request | 1:1 |
| 15 | Person\_Message | Placeholder | M:1 |

# Design Decisions

1. One Person may or may not make multiple New\_Requests, so therefore there is a 1:M relationship between the table Person and New\_Requests.
2. One Person may or may not have multiple Pending\_Requests, so therefore there is a 1:M relationship between the table Person and Pending \_Requests.
3. One Person may or may not have multiple Completed\_Requests, so therefore there is a 1:M relationship between the table Person and Completed \_Requests.
4. One Person may or may not offer multiple Services, so therefore there is a 1:M relationship between the table Person and Service.
5. One person may or may not subscribe to multiple categories, so therefore there is a 1:M relationship between the table Person and Person\_Category.
6. One person may or may not have multiple messages, so therefore there is a 1:M relationship between the table Person and Person\_Message.
7. One person may or may not be the author of one or more messages. Therefore there is a 1:M relationship between the table Person and Message.
8. One category can be subscribed by multiple Persons, therefore there is a M:1 relationship between the table Person\_Category and Category.
9. Every New\_Request can belong to only one particular category. Therefore, there is a 1:1 relationship between the table Category and New\_Request.
10. Every Pending\_Request can belong to only one particular category. Therefore, there is a 1:1 relationship between the table Category and Pending \_Request.
11. Every Completed\_Request can belong to only one particular category. Therefore, there is a 1:1 relationship between the table Category and Completed\_Request.
12. Every Service can belong to only one particular category. Therefore, there is a 1:1 relationship between the table Category and Service.
13. Since every message is between two Person’s, each message can appear in multiple rows of Person\_Message. Therefore there is a 1:M mapping between Message and Person\_Message.
14. A Message may or may not allow a Person to accept
15. Every Person’s Placeholder(Inbox, Sent, etc..) can hold multiple messages. Therefore, there is a M:1 relationship between Person\_Message and Placeholder.

**Design Explanation**

Requests are the major component of the Online Marketplace for Services that we have designed. A Person may place one or more Requests for a service which they require. The states that a request goes through are:

1. New\_Request – When a request is newly created it is added as a row to this table. So far, the Request has not been serviced i.e no Person is working on it currently. A Person who is willing to work on it, i.e a Provider, would send a special message to the Requestor indicating that they are willing to work on this request. The Requestor can accept or decline the request. If the Requestor accepts the request, we move to state 2.
2. Pending\_Request – A New\_Request which has been accepted is deleted from the New\_Request table, and added to the Pending\_Request table. This table stores all the request which are currently being worked on, including the deadline associated with these requests. When the deadline expires, or the provider indicates that work is finished, we move to state 3.
3. Completed\_Request – A Completed\_Request indicates that work on a request has finished either through deadline expiry or by the provider indicating that work was finished. There is a column ‘Accepted’ to indicate whether the work was finished satisfactorily or not.

Messages are the second major component of the application. A brief description of each table associated with sending/receiving messages follows:

1. Messages – This table stores the actual information associated with each message. It includes the message\_id and author\_id. Each message will be a part of a thread between 2 users. A thread can encapsulate multiple messages.
2. Placeholder – This table stores multiple folders that a message may be a part of. For example, a Sent or an Inbox folder.
3. Person\_Message – This table associates each message with one or more Persons. For example, if a Person (the sender) composes a message to another Person (the receiver) there will be 2 entries made in the Person\_Message table, and 1 entry in the Message Table. If a person indicates a message is to be deleted, then deletion occurs only in the Person\_Message table. Therefore, it will only delete the message for that Person. All other Person’s associated with that message will still be able to view the Message.

# Partial Conceptual Model



# Logical Model



Note: Please find attached the copies of Partial Conceptual Model and Logical Model as .pdf files in our submission.

**References**

www.fiverr.com

http://stackoverflow.com/questions/3370647/database-design-for-email-messaging-system