

# INFO 6210

## Data Management and Database Design

### Normalization and Optimization

### Assignment 3

Professor: Nik Bear Brown

Due March 23, 2019

In this assignment, you will normalize and optimize the database you created in Assignment 2.

#### *Database normalization*

Database normalization is the process of organizing the attributes and tables of a relational database to minimize data redundancy.

Normalization involves refactoring a table into smaller (and less redundant) tables but without losing information; defining foreign keys in the old table referencing the primary keys of the new ones. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys.

Edgar F. Codd, the inventor of the relational model (RM), introduced the concept of normalization and what we now know as the First normal form (1NF) in 1970. Codd went on to define the Second normal form (2NF) and Third normal form (3NF) in 1971.

#### Design Requirements

You will check whether your tables are in First normal form (1NF), Second normal form (2NF) and Third normal form (3NF). If not, you'll restructure your database so that all of the tables are in Third normal form; that is, you normalize the database.

Your submission must include:

- A check that your tables are in First normal form (1NF)
- A check that your tables are in Second normal form (2NF)
- A check that your tables are in Third normal form (3NF)
- The final SQL of a normalized database in Third normal form (3NF).
- Views for all of your use-cases

#### First normal form (1NF)

- Each table has a primary key: minimal set of attributes which can uniquely identify a record
- The values in each column of a table are atomic (No multi-value attributes allowed).

- There are no repeating groups: two columns do not store similar information in the same table.

#### Second normal form (2NF)

- All requirements for 1<sup>st</sup> NF must be met.
- No partial dependencies.
- No calculated data

#### Third normal form (3NF)

- All requirements for 2<sup>nd</sup> NF must be met.
- Eliminate fields that do not directly depend on the primary key; that is no transitive dependencies.

#### Create Views Syntax

```
CREATE VIEW Miley_Tweets AS
SELECT Tweets.tweetid, Tweets.text, Users.screen_name
FROM Tweets
JOIN Users ON (Users.userid = Tweets.userid)
WHERE Users.screen_name= "MileyCyrus";
```

#### Scoring Rubric

- (15 points) A check that your tables are in First normal form (1NF)
- (15 points) A check that your tables are in Second normal form (2NF)
- (15 points) A check that your tables are in Third normal form (3NF) and the final SQL
- (10 points) Views for all of your use-cases
- (10 points) Each student must create 5 indexes and you must show that they improve the performance of at least one query for each index.
- (10 points) Each student must create 5 functions and you must show that they are practically useful
- (10 points) Each student must create 5 stored procedures and you must show that they are practically useful
- (15 points) Professionalism and documentation

#### Submission of Assignments

Your submission should be a zip file with all of the required work. You will submit your assignments via BlackBoard. Click the title of assignment (blackboard -> assignment -> <Title of Assignment>), to go to the submission page. You will know your score on an assignment, project or test via BlackBoard. BlackBoard represents only the raw scores. Not normalized or curved grades.