

# CS5300 PROGRAMMING PROJECT01

## DATABASE NORMALIZATION

**Team:** Nandini Gopu(12603457), Jonas Edward Jangam(12611244), Sindhu Reddy Kontham(12602099), Manaswini Nara(12611698), Hari Sumanth Thalluru(12611723)

### Sample Input:

**Dataset:** The sample data contains attributes like StudentID, FirstName, Lastname, Course, Professor, ProfessorEmail, CourseStart, CourseEnd. It contains 5 tuples. The dataset is taken as .csv file(comma separated values).

**Functional Dependencies:** The functional dependencies used for the dataset are:

StudentId->FirstName, LastName.

Course->CourseStart, CourseEnd, Professor.

Professor->ProfessorEmail.

**Multi-valued dependencies:** The multi-valued dependencies used for the dataset are:

Course ->> CourseStart

Course ->> CourseEnd

The functional dependencies and multi-valued dependencies are taken as .txt file(input.txt).

### User Input:

- ☐ Choice of the highest normal form to read(1NF,2NF,3NF,BCNF,4NF,5NF).
- ☐ Primary key used(StudentID, Course)

### Core Components:

1. **Input Parser:** The parser function is used to read the csv file and the txt file which is used to store the functional dependencies. It breaks down the text into recognized strings of characters for further analysis.
2. **Normalizer:** The Normalizer decomposes the input dataset into the required normal form based on the given functional dependencies. The normalization methods are used to decompose the given dataset into the user-required normal form.
3. **SQL Query Generator:** It refers to a tool or functionality that helps to generate the SQL queries.

### Deliverables:

**Source Code:** Here, we have used the Python programming language to normalize the given dataset.

### Code Description:

- ☐ The Python libraries are imported, to perform the required operations.
- ☐ The input dataset is given as .csv file. The read method is used to read the data from the file.
- ☐ The open method is used to read the functional dependencies txt file.
- ☐ Getting the input of the user required normal form and the primary keys.
- ☐ The Parser method is used to convert the given input files into the string characters.

- ☐ Checking for the multi-valued dependencies.
- ☐ is\_superkey is defined to check for the superkey.
- ☐ bcnf\_decomposition function is defined to decompose the relation into BCNF normal form.
- ☐ in\_1nf method is defined to check whether the dataset is in 1NF or not.
- ☐ in\_2nf method is defined to check whether the dataset is in 2NF or not.
- ☐ in\_3nf method is defined to check whether the dataset is in 3NF or not.
- ☐ in\_bcnf method is defined to check whether the dataset is in BCNF or not.
- ☐ in\_4nf method is defined to check whether the dataset is in 4NF or not.
- ☐ in\_5nf method is defined to check whether the dataset is in 5NF or not.
- ☐ first\_normal\_form function is defined to decompose the given dataset into the 1NF.
- ☐ second\_normal\_form function is defined to decompose the given dataset into the 2NF.
- ☐ third\_normal\_form function is defined to decompose the given dataset into the 3NF.
- ☐ bc\_normal\_form function is defined to decompose the given dataset into the BCNF.
- ☐ fourth\_normal\_form function is defined to decompose the given dataset into 4NF.
- ☐ decomposing\_to\_5nf function is defined to check for the lossless decomposition of 5NF.
- ☐ fifth\_normal\_form function is defined to decompose the given dataset into 5NF.
- ☐ output function is defined to generate the user required output results.
- ☐ sql\_query\_1NF function is defined to print the 1NF sql query as an output.
- ☐ sql\_query\_2\_3 function is defined to print the 2NF, 3NF sql queries as output.
- ☐ sql\_query\_BCNF\_4\_5 function is defined to print the BCNF, 4NF, 5NF sql queries as output.
- ☐ Checking for the normal form of the given dataset whether it is in any type of normal form or not.
- ☐ The highest normal form of the given dataset can be found by using the functions in\_1nf, in\_2nf, in\_3nf, in\_4nf, in\_bcnf and in\_5nf from the variable high\_normalform.

### Code Execution:

- ☐ Enter the highest normal form to normalize the given table press 1 for 1NF, 2 for 2NF, 3 for 3NF, B for BCNF, 4 for 4NF and 5 for 5NF.
- ☐ Then press 1 to get the highest normal form of the given relation and 2 if the highest normal form is not required.
- ☐ Then enter the primary key values which is separated by commas for this relation the primary key should be given as StudentID,Course.
- ☐ If the given table is not in the normal form that is provided then it decomposes the table until it satisfies the given normal form then it provides the query of the decomposed tables.
- ☐ Else if the given table satisfies the provided normal form then it returns the query of the original table.
- ☐ For the 5NF, the candidate keys are provided for each and every table. For the student table the candidate key is StudentID, for the course table the candidate key is Course and for the professor table the candidate key is Professor.
- ☐ At the end the highest normal form satisfied by the given table is displayed. The given table satisfies only 1NF.