Projects	Group Database Projects
CS206 Database Management System, Winter'2018, IIIT Vadodara	

Objective of this project is to learn to apply theoretical principles in solving a real life situation. To get hands on, it is expected that you design and implement database system for a relatively large and real-life situation.

For Database Projects, you work in a team. Students themselves form teams, and permitted team size is 3 to 4 (not less, not more).

Project Evaluation Parameters:

1. Size and Complexity of Schema.
2. Realistic to the real-word
3. Complexity of queries answered
4. Goodness of the solutions

Project Milestones -

1. **Write a description of the scenario**. [Submission Date: 09-Feb-18]

   There is no standard format for scenario description, idea is you must be able to state the scope of database, and capture all data requirements from the description. Sample scenario is being provided for your reference.

   Important Note: In order to identity scope of the database project; try to understand database in users perspective that what purpose this database will serve to the user, what kind of queries database will be able to answer, and so forth. Do not think in terms of table or entities.

2. Draw **ER Diagram** for the scenario [Submission Date: 23-Feb-18].
   ER diagram should include Cardinality and participation constraints. Use Ternary Relationship, Generalization/Specializations judiciously; use them only when you do not find any other way out.

   Use software Dia (http://dia-installer.de/) for creating ERDs.

3. Create **relational schema diagram** from your ER diagram. Identify all Functional Dependencies in your Database and prove that all relations, you have created are in BCNF. If not in BCNF decompose it to bring in BCNF, if you cannot, then give reason, why you can't. [Submission Date: 09-Mar-18]

4. **Implement the database** as separate schema on the PostgreSQL server. Save DDL scripts for creating schema. You will have to submit schema diagram and DDL scripts of your project database. [Submission Date: 23-Mar-18]

5. **Retrieval queries on your project database**. You should include as many queries as your scenario may demand. I expect each scenario should have about 10-15 GOOD

queries. Test your queries on sample database (Populate your database with sample data through INSERT statements and save INSERT statements in a script file) [Submission Date: 13-Apr-18]

6. **Stored procedures and Triggers** in your project scenario. Identify some functionality and constraints that can be better accomplished by stored procedures and triggers. [Submission Date: 13-Apr-18]

7. Create a simple console based program (No GUI required) using embedded SQL in C that accesses your project database [Tentative Submission Date: 13-Apr-18]

8. Final document containing ERD and relational schema with normalization proofs. SQL statements of all the queries including DDL INSERT queries. Also submit code of your stored procedures and console application.
   [Tentative Submission Date: 13-Apr-18]

9. Final Presentation and Viva: Dates to be announced. For final viva and presentations, you are to bring hard copies as following –
   a. ERD (1 copies)
   b. Relational Schema Diagram (2 copies)
   c. Note: Relational schema should be at least in 3NF; BCNF desirable.
   d. Set of FDs (no MVDs) normalization proofs. (1 copy).
   e. If any relation is not in BCNF, give reason why it could not be decomposed into BCNF
   f. SQL Scripts of SELECT queries (1 copy)
   g. Source Code of your stored procedures and console application. (1 copy)

Instructions for console application in C-

Create following set of functions using embedded SQL in C.

```
void ExecuteQuery(SQL_String) {
     //Executes SELECT type of sql statement and displays
     //returned rows one row on a line on "stdout" if no error,
     //otherwise displays error details */
}

void ExecuteUpdate(SQL_String) {
     //Executes INSERT/UPDATE/DELETE type of sql statement
     //and displays "successful" on stdout if no error,
     //otherwise displays error details.
}

void main() {
     /*
     Does following tasks:
     1. test ExecuteQuery and ExecuteUpdate for few of your
     sample queries.
     2. Add couple of rows in one of the table using
     "Prepared Statement"
     */
}
```