SE Factory

Week 3 - Assignments

`version 03BD_1701`

```
Change Log
- Removed delivery dates
- Fixed DELETE Command Syntax
- Added the assumptions discussed in class
```

Relational Databases

Exercise 1

For this exercise we will be creating a simple database. The underlying **data structure** of this database is up to you to define, however, your database should have (at least) the following features (discussed in today's lecture):

- CRUD
- Consistency
- Validity
- Speed (All queries to return results in under 2 seconds for a maximum of 10,000 records)
- Persistence

Your database will implement the following functionality:

- Create a database
- Delete a database
- Create a table
- Add a record
- Get/Retrieve a record
- Delete a record

The database will not use SQL, instead it will use a similar but more basic language defined as follows:

Sample Input

```
-- Create a Database
CREATE,DATABASE,"Database Name"

-- Delete a Database
DELETE,DATABASE,"Database Name"

-- Create a table (Number of columns is indefinite)
CREATE,TABLE,"TABLENAME",COLUMNS,"Column1","Column2","Column3"

-- Add a record (The table has a non-null constraint on all columns)
ADD,"10","Bassem","Dghaidi","SEF Instructor"

-- Retrieve a record
GET,"10"

-- Delete a record
DELETE,ROW,"10"
```

Sample Output

```
"Database Name" CREATED

"Database Name" DELETED

"Table Name" CREATED

Record ADDED

"10","Bassem","Dghaidi","SEF Instructor"

Record DELETED
```

Assumptions

- If we create `n` tables or database, we always retrieve or execute subsequent commands on the `nth`
- First column is always an `Integer` and it is always considered a `Primary Key`
- `CREATE,TABLE,COLUMNS` can take an infinite number of column inputs
- `GET` will search for the input value in `all` columns
- `DELETE,ROW` will always take an `Integer` as input which is always the primary key
- `Primary Key` cannot be auto-increment and the value has to be inserted by the user