DATABASE MANAGEMENT SYSTEMS PROJECT REPORT

1.

Our team has three members:

- Uğur Kızılkuş
- Serdar Okunakol
- Mehmet Emen

Our project consists of all conflicts records between Ukraine and pro-Russian groups since 2014. Dataset has 50 columns and over 3000 rows. There are different types of columns (varchar, float etc.) in the dataset.

2. In the first phase;
We did not define primary key for each table.
We did not define seperate tables for multivalued columns.
We put wrong type of variable for some columns.
There were no views.
There were no store procedures.
We did not insert our data into tables.

In the second phase;
We defined primary key for each table.
We created seperate tables for columns that has multivalued cells.
We edited type of some columns.
We implemented two views.
We implemented two store procedures.
We inserted our data into tables.

3. Dataset Source: https://data.humdata.org/dataset/ucdp-data-for-ukraine

We had one .csv file. We divided this .csv file into seperate .csv files according to tables. For example, our main .csv file has 50 columns and our event table has 2 columns, event_id and number_of_sources. We extracted these two columns from 50 columns and created a new .csv file. We inserted this new .csv file into event table on MySQL Workbench. Before loading data, we added new rows for every multivalued columns. We loaded whole data succesfully.

4. Serdar and Mehmet used MySQL Workbench during the semester but Uğur used a program called DBeaver since MySQL Workbench didn't work properly on Macbooks with M1 Apple Silicon chip. But Uğur also did final presentation with MySQL Workbench on Serdar's Windows PC.

5.

Views:
casualty_with_time_report: It is listing all conflicts with its date and how many people died.
conflictoverview: It is listing all conflicts and their general informations.

Stored Procedures:
GetDeathCountAndPercentage: It takes event_id as parameter and returns death percentage of the event in all events.
GetSourceDetailsByEvent: It takes event_id as parameter and returns all resource details about the event.

6.

'conflictoverview' view: select event_id = 147997 returns;
number_of_sources -> 5, conflict_unique_id -> 25000, type_of_violence -> 1, side_A -> Government of Ukraine, side_B -> Maidan

'casualty_with_time_report' stored procedure: input = event_id -> 149248 returns;
source_date -> 2019-02-20, source_headline -> Two days of bloody unrest in Ukraine, source_id -> 8013, source_office_id -> 23000

7.

There were some problems in our project. Here is our solution suggestions;

We created new primary keys by auto incrementing. These primary keys could have been made composite of two columns.

There were multivalued cells in the dataset. We made atomic them by adding new rows but it can also be made atomic by adding new columns.

Delimiter was both semi comma and comma for multivalued values which makes hard to seperate values. It can be define one common delimiter.

There were null values in the project. Null values can be provided.

8.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema dbFinal
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema dbFinal
-- -----------------------------------------------------
```

```
CREATE SCHEMA IF NOT EXISTS `dbFinal` DEFAULT CHARACTER SET utf8 ;
USE `dbFinal` ;

-- -----------------------------------------------------
-- Table `dbFinal`.`Event`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Event` (
  `event_id` INT NOT NULL,
  `number_of_sources` INT NOT NULL,
  PRIMARY KEY (`event_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`Sides`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Sides` (
  `side_id` INT NOT NULL,
  `side_name` VARCHAR(300) NOT NULL,
  PRIMARY KEY (`side_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`Conflict`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Conflict` (
  `event_id` INT NOT NULL,
  `conflict_unique_id` INT NOT NULL,
  `conflict_id` INT NULL,
  `type_of_violence` INT NULL,
  `dyad_id` INT NULL,
  `sidea_id` INT NULL,
  `sideb_id` INT NULL,
  PRIMARY KEY (`conflict_unique_id`),
  INDEX `fk_Conflict_Event1_idx` (`event_id` ASC) VISIBLE,
  INDEX `fk_Conflict_Sides1_idx` (`sidea_id` ASC) VISIBLE,
  INDEX `fk_Conflict_Sides2_idx` (`sideb_id` ASC) VISIBLE,
  CONSTRAINT `fk_Conflict_Event1`
    FOREIGN KEY (`event_id`)
    REFERENCES `dbFinal`.`Event` (`event_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Conflict_Sides1`
    FOREIGN KEY (`sidea_id`)
    REFERENCES `dbFinal`.`Sides` (`side_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Conflict_Sides2`
    FOREIGN KEY (`sideb_id`)
    REFERENCES `dbFinal`.`Sides` (`side_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `dbFinal`.`SourceOffice`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`SourceOffice` (
  `office_name` VARCHAR(200) NOT NULL,
  `office_id` INT NOT NULL,
  PRIMARY KEY (`office_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`Source`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Source` (
  `event_id` INT NOT NULL,
  `source_date` DATE NULL,
  `source_headline` VARCHAR(500) NULL,
  `source_id` INT NOT NULL,
  `source_office_id` INT NULL,
  PRIMARY KEY (`source_id`),
  INDEX `fk_Source_Event1_idx` (`event_id` ASC) VISIBLE,
  INDEX `fk_Source_table11_idx` (`source_office_id` ASC) VISIBLE,
  CONSTRAINT `fk_Source_Event1`
    FOREIGN KEY (`event_id`)
    REFERENCES `dbFinal`.`Event` (`event_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Source_table11`
    FOREIGN KEY (`source_office_id`)
    REFERENCES `dbFinal`.`SourceOffice` (`office_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`Country`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Country` (
  `country_id` INT NOT NULL,
  `iso_3` VARCHAR(3) NOT NULL,
  `country` VARCHAR(45) NOT NULL,
  `region` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`country_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`Location`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Location` (
  `event_id` INT NOT NULL,
```

```
  `location_id` INT NOT NULL,
  `where_prec` INT NULL,
  `where_coordinates` VARCHAR(300) NULL,
  `where_description` VARCHAR(500) NULL,
  `adm_1` VARCHAR(200) NULL,
  `adm_2` VARCHAR(200) NULL,
  `latitude` FLOAT NULL,
  `longitude` FLOAT NULL,
  `priogrid_gid` INT NULL,
  `country_id` INT NOT NULL,
  PRIMARY KEY (`location_id`),
  INDEX `fk_Location_Event1_idx` (`event_id` ASC) VISIBLE,
  INDEX `fk_Location_table11_idx` (`country_id` ASC) VISIBLE,
  CONSTRAINT `fk_Location_Event1`
    FOREIGN KEY (`event_id`)
    REFERENCES `dbFinal`.`Event` (`event_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Location_table11`
    FOREIGN KEY (`country_id`)
    REFERENCES `dbFinal`.`Country` (`country_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`Casualty`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Casualty` (
  `casualty_id` INT NOT NULL,
  `event_id` INT NOT NULL,
  `deaths_a` INT NOT NULL,
  `deaths_b` INT NOT NULL,
  `deaths_civilians` INT NOT NULL,
  `death_unknown` INT NOT NULL,
  `best` INT NOT NULL,
  `high` INT NOT NULL,
  `low` INT NOT NULL,
  PRIMARY KEY (`casualty_id`),
  INDEX `fk_Casualty_Event1_idx` (`event_id` ASC) VISIBLE,
  CONSTRAINT `fk_Casualty_Event1`
    FOREIGN KEY (`event_id`)
    REFERENCES `dbFinal`.`Event` (`event_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 1;


-- -----------------------------------------------------
-- Table `dbFinal`.`Time`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`Time` (
```

```
  `event_id` INT NOT NULL,
  `active_year` INT NOT NULL,
  `date_prec` INT NOT NULL,
  `time_unique_id` INT NOT NULL,
  `date_start` DATE NULL,
  `date_end` DATE NULL,
  INDEX `fk_Time_Event1_idx` (`event_id` ASC) VISIBLE,
  PRIMARY KEY (`time_unique_id`),
  CONSTRAINT `fk_Time_Event1`
    FOREIGN KEY (`event_id`)
    REFERENCES `dbFinal`.`Event` (`event_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 1;


-- -----------------------------------------------------
-- Table `dbFinal`.`source_originals`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`source_originals` (
  `source_original_name` VARCHAR(150) NOT NULL,
  `original_id` INT NOT NULL,
  PRIMARY KEY (`original_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `dbFinal`.`source_connection`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbFinal`.`source_connection` (
  `connection_id` INT NOT NULL,
  `source_id` INT NOT NULL,
  `original_id` INT NOT NULL,
  PRIMARY KEY (`connection_id`),
  INDEX `fk_source_connection_source_originals1_idx` (`original_id` ASC) VISIBLE,
  INDEX `fk_source_connection_Source1_idx` (`source_id` ASC) VISIBLE,
  CONSTRAINT `fk_source_connection_Source1`
    FOREIGN KEY (`source_id`)
    REFERENCES `dbFinal`.`Source` (`source_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_source_connection_source_originals1`
    FOREIGN KEY (`original_id`)
    REFERENCES `dbFinal`.`source_originals` (`original_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

9. All SQL codes used in our project is too long. That's why we are adding one part of the SQL code. Since we have already added tables code in 8th question, we are only adding insert codes here.

```
-- Data for table `dbFinal`.`Event`
-- -----------------------------------------------------
START TRANSACTION;
USE `dbFinal`;
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (147997, 5);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149221, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149248, 9);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149249, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149251, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149250, 8);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153030, 4);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149866, 3);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149868, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (419757, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149876, 3);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149879, 2);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (398179, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (398183, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (149973, 3);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (152957, 10);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (399045, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (399046, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (399063, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (399053, 2);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153027, 2);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153037, 2);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153149, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153143, 3);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153433, 1);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153154, 7);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (153435, 6);
INSERT INTO `dbFinal`.`Event` (`event_id`, `number_of_sources`) VALUES (400158, 1);
…
…
…
-- Data for table `dbFinal`.`Conflict`
-- -----------------------------------------------------
START TRANSACTION;
USE `dbFinal`;
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (147997, 25000, 13219, 1, 14085, 61, 5817);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149221, 25001, 13219, 1, 14085, 61, 5817);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149248, 25002, 13219, 1, 14085, 61, 5817);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149249, 25003, 13219, 1, 14085, 61, 5817);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149251, 25004, 13219, 1, 14085, 61, 5817);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149250, 25005, 13219, 1, 14085, 61, 5817);
```

```
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (153030, 25006, 13240, 2, 14113, 5837, 5836);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149866, 25007, 13246, 1, 14124, 61, 5840);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149868, 25008, 13246, 1, 14124, 61, 5840);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (419757, 25009, 13246, 1, 14124, 61, 5840);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149876, 25010, 13246, 1, 14124, 61, 5840);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (149879, 25011, 13246, 1, 14124, 61, 5840);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (398179, 25012, 13246, 1, 14124, 61, 5840);
INSERT INTO `dbFinal`.`Conflict` (`event_id`, `conflict_unique_id`, `conflict_id`, `type_of_violence`, `dyad_id`,
`sidea_id`, `sideb_id`) VALUES (398183, 25013, 13246, 1, 14124, 61, 5840);
…
…
…
-- -----------------------------------------------------
-- Data for table `dbFinal`.`SourceOffice`
-- -----------------------------------------------------
START TRANSACTION;
USE `dbFinal`;
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Agence France Presse', 23000);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Reuters News', 23001);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Reuters Pictures', 23002);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Associated Press Newswires', 23003);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('ITAR-TASS World Service', 23004);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('BBC News', 23005);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('BBC', 23006);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Office of the United Nations High
Commissioner for Human Rights', 23007);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('MemoryBook', 23008);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Kyiv Post', 23009);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Voice of Russia', 23010);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('NaN ', 23011);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('The Independent', 23012);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Xinhua News Agency', 23013);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('Euromaidan Press', 23014);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('MeoryBook', 23015);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('CNN', 23016);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('RIA Novosti', 23017);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('RT', 23018);
INSERT INTO `dbFinal`.`SourceOffice` (`office_name`, `office_id`) VALUES ('BBC Monitoring Ukraine & Baltics',
23019);
…
…

-- Data for table `dbFinal`.`Source`
-- -----------------------------------------------------
START TRANSACTION;
USE `dbFinal`;
```

INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (147997, '22.01.2014', 'Police disperse Kiev protest- activists say one shot dead', 8000, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (147997, '22.01.2014', 'Two activists shot dead during Ukraine protests: prosecutors sjw/ma/dh', 8001, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (147997, '22.01.2014', 'Two shot dead as Ukraine police storm Kiev protest', 8002, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (147997, '22.01.2014', 'Five dead as Ukraine police launch assault on protesters', 8003, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (147997, '25.01.2014', 'One more Ukrainian protester dies of wounds in hospital: official', 8004, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149221, '27.01.2014', 'Dead man found hanging on Kiev\'s Independence Square - police', 8005, 23001);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '18.02.2014', 'UKRAINE PARLIAMENTARY DEPUTY SAYS ON FACEBOOK THAT THREE DEAD BODIES OF PROTESTERS ARE IN A BUILDING CLOSE TO PARLIAMENT', 8006, 23001);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '19.02.2014', NULL, 8007, 23002);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '18.02.2014', 'Ukraine police battle protesters after nation\'s bloodiest day', 8008, 23001);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '19.02.2014', 'Four protesters killed in police assault on Kiev camp: opposition', 8009, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '19.02.2014', 'Biden urges Ukraine gov\'t to show restraint', 8010, 23003);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '20.02.2014', 'WRAPUP 2-Ukraine president agrees truce with opponents as U.S. imposes visa bans', 8011, 23001);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '19.02.2014', 'Ukraine must keep its military away from protests- Pentagon says', 8012, 23001);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '19.02.2014', 'Two days of bloody unrest in Ukraine', 8013, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149248, '19.02.2014', 'Two days of bloody unrest in Ukraine', 8014, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149249, '20.02.2014', 'Ukraine death toll rises to 28: ministry', 8015, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149251, '22.02.2014', 'WRAPUP 9-Ukraine parliament removes Yanukovich- who flees Kiev in \"coup\",,NULL\n149250,21.02.2014,\"Ukraine rivals sign deal to end crisis', 8016, 23001);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149250, '21.02.2014', 'Three days of deadly unrest in Ukraine', 8017, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149250, '20.02.2014', 'Two police commandoes die in fire at riot police base in Ukraine?s Lvov', 8018, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149250, '20.02.2014', 'At least 67 killed in this week\'s Kiev clashes: authorities', 8019, 23004);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149250, '20.02.2014', 'Dozens dead on Ukraine\'s deadliest day', 8020, 23000);
INSERT INTO `dbFinal`.`Source` (`event_id`, `source_date`, `source_headline`, `source_id`, `source_office_id`)
VALUES (149250, '20.02.2014', 'WRAPUP 9-Ukraine truce shattered- death toll hits 67', 8021, 23000);
…’
…

10. We did not implement GUI interface for the project, we implemented store procedures instead. But we wrote script to make multivalued cells atomic by adding new rows. Script is below.

main.py

```python
file_path = '/Users/ugurkizilkus/Desktop/db_dataset/ukraine_conflict_dataset.csv'

columns_to_split = ['source_article', 'source_office', 'source_date', 'source_headline', 'source_original']
df_full = pd.read_csv(file_path, encoding=used_encoding, error_bad_lines=False, warn_bad_lines=True)


def expand_multivalued_rows(df, columns):
    new_rows = []

    for _, row in df.iterrows():
        split_values = {col: str(row[col]).split(';') for col in columns}
        max_len = max(len(values) for values in split_values.values())
        for i in range(max_len):
            new_row = row.to_dict()
            for col in columns:
                new_row[col] = split_values[col][i] if i < len(split_values[col]) else split_values[col][-1]
            new_rows.append(new_row)

    new_df = pd.DataFrame(new_rows)
    return new_df


df_atomic = expand_multivalued_rows(df_full, columns_to_split)
output_file_path = '/Users/ugurkizilkus/Desktop/db_dataset/atomic_ukraine_dataset.csv'
df_atomic.to_csv(output_file_path, index=False, encoding='utf-8')

```