```
import sys
import mysql.connector
from pykafka import KafkaClient
from pykafka.common import OffsetType
from pykafka.exceptions import SocketDisconnectedError, LeaderNotAvailable


class KafkaMySQLSink:

    def __init__(self, kafka_bootstrap_server, kafka_topic_name, database_host,
database_username, database_password,
                database_name):

        # Initialize Kafka Consumer
        kafka_client = KafkaClient(kafka_bootstrap_server)
        self.consumer = kafka_client \
            .topics[kafka_topic_name] \
            .get_simple_consumer(consumer_group="groupid",
auto_offset_reset=OffsetType.LATEST)

        # Initialize MySQL database connection
        self.db = mysql.connector.connect(
            host=database_host,
            user=database_username,
            password=database_password,
            database=database_name
        )

    # Process single row
    def process_row(self, text):
        # Get the db cursor
        db_cursor = self.db.cursor()

        # DB query for supporting UPSERT operation
        sql = "INSERT INTO texts(text1, text2) VALUES (%s, %s)"
        val = (text, text)
        db_cursor.execute(sql, val)

        # Commit the operation, so that it reflects globally
        self.db.commit()

    # Process kafka queue messages
    def process_events(self):
        try:
            for queue_message in self.consumer:
                if queue_message is not None:
                    msg = queue_message.value
                    print(msg)
                    self.process_row(msg)

        # In case Kafka connection errors, restart consumer ans start 
processing
        except (SocketDisconnectedError, LeaderNotAvailable) as e:
            self.consumer.stop()
            self.consumer.start()
            self.process_events()

    def __del__(self):
        # Cleanup consumer and database connection before termination
        self.consumer.stop()
        self.db.close()


if __name__ == "__main__":

    # Validate Command line arguments
    if len(sys.argv) != 7:
        print("Usage: kafka_mysql.py <kafka_bootstrap_server> <kafka_topic> 
<database_host> "
              "<database_username> <database_password> <database_name>")
        exit(-1)

    kafka_bootstrap_server = sys.argv[1]
    kafka_topic = sys.argv[2]
    database_host = sys.argv[3]
    database_username = sys.argv[4]
    database_password = sys.argv[5]
    database_name = sys.argv[6]

    ad_manager = None

    try:
        kafka_mysql_sink = KafkaMySQLSink(kafka_bootstrap_server, kafka_topic, 
database_host, database_username,
                               database_password, database_name)
        kafka_mysql_sink.process_events()

    except KeyboardInterrupt:
        print('KeyboardInterrupt, exiting...')

    finally:
        if kafka_mysql_sink is not None:
            del kafka_mysql_sink
```