대용량 데이터 처리를 위한 기술 활용 (Kafka, ES

대용량 데이터 처리 및 Kafka + Bulk 활용

데이터 처리 흐름

1. **DB -> Kafka -> ElasticSearch** 순서로 데이터가 전달됨
2. **Kafka**를 **bucket** 단위로 나누어 데이터 전송 하여 효율적인 처리
3. **ElasticSearch**에는 **Bulk API**를 활용하여 대량의 데이터를 일괄 삽입

주요 개선 사항

1. **Kafka**를 **Bucket** 단위로 전송 → 데이터 스트리밍 성능 최적화
2. **ElasticSearch**에 **Bulk** 처리 적용 → 인덱싱 성능 향상 및 네트워크 부하 감소
3. 대용량 데이터**(150,000건 이상)**를 빠르고 안정적으로 처리

Kafka Bucket 처리 예시

```
qryCnt := models.QryCount(y.Db, cntStr)
	fmt.Println("qryCnt:", qryCnt)

	bucketCnt := 0
	if qryCnt == 0 {
		bucketCnt = 0
	} else {
		bucketCnt = 1 + (int(qryCnt) / p.Limit)
	}

    fmt.Println("bucketCnt:", bucketCnt)
	for i := 0; i < bucketCnt; i++ {
		// for i := 0; i < 3; i++ {
		p.Offset = p.Limit * i
		sqlStr, _, _ := models_join.ListType1PlainQryStrGet(y, lt, p, q, "/list")
```

```
    if err := t.ProduceaBucketToKafka(y, "kkk", headers, sqlStr, topic); err != nil {
        return errors.New(e.PageQryErr("vrt452", ": "+err.Error()))
    }
}
```

Bucket □ □□□□ □□ □□ □□!

ElasticSearch Bulk API □□

```
for _, row := range v.Page {
    row.ClientCode = clientCode
    fmt.Printf("BuyerId: %d, LastSorderDate: %s, ClientCode : %s\n", row.Id, row.LastSorderDate, row.ClientCode)
    row.LastVistDate = row.LastSorderDate
    // Meta □□□ □□
    meta := fmt.Sprintf(`{ "index": { "_index": "%s", "_id": "%d" } }%s`, esIndex, row.Id, "\n")
    bulkRequest.WriteString(meta)

    // □□ □□□ □□
    data, _ := json.Marshal(row)
    bulkRequest.WriteString(string(data) + "\n")
    fmt.Println(string(data)) // □□□□ □□□□ □□

    fmt.Printf("Prepared document for BuyerId: %d\n", row.Id)
}

// Bulk □□ □□
res, err := es.Bulk(bytes.NewReader(bulkRequest.Bytes()))
if err != nil {
    return fmt.Errorf("bulk request failed: %w", err)
}
defer res.Body.Close()

// Elasticsearch □□ □□
if res.IsError() {
    log.Printf("Bulk indexing failed: %s", res.String())
    return fmt.Errorf("bulk indexing failed")
}
```

Bulk API □ □□□□ □ □□ □□ □□□□ □□ □□!

□□ □□□ □□□ □□□□ □□□□□ □□□□ □□□ □□□□ □ □□□ □□□.

Revision #1
Created 7 February 2025 03:02:29 by □□
Updated 7 February 2025 03:06:08 by □□