

lab1

数据库系统存储的数据可分为三类：

- 元数据
- 表中的record
- 索引

lab1的核心便是对表和索引的key的编码和解码，其key的形式为：

- 表中的record: tablePrefix+tableID+recordPrefix+handle
- 索引的元素: tablePrefix+tableID+indexPrefix+indexID+indexValue
 - 对于unique索引, indexValue即对应的record存储的值
indexedColumnsValue
 - 对于非unique索引, indexValue = indexColumnsValue+rowID

其解码流程可分为四个模块

1. decodeTableKey
2. decodeRowKey
3. decodeIndexID
4. decodeIndexValue

如此其解码流程为：

```
DecodeRecordKey: decodeTableKey->decodeRowkey  
DecodeIndexKeyPrefix: decodeTableKey->decodeIndexID
```

code如下：

```
func DecodeRecordKey(key kv.Key) (tableID int64, handle int64, err error) {  
    tableID = DecodeTableID(key) //decode tableKey  
    if tableID == 0 {  
        return 0, 0, errInvalidKey.GenWithStack("invalid key - %q",  
key)  
    }  
    handle, err = DecodeRowKey(key) //decode rowKey  
    if err != nil {  
        return 0, 0, errors.Trace(err)  
    }  
    return tableID, handle, nil  
}  
  
func DecodeIndexKeyPrefix(key kv.Key) (tableID int64, indexID  
int64, indexValues []byte, err error) {  
    tableID = DecodeTableID(key) //decode tableKey  
    if tableID == 0 {
```

```
        return 0, 0, nil, errInvalidKey.GenWithStack("invalid key -
%q", key)
    }
    indexID, err = DecodeIndexID(key) //decode indexID
    if tableID == 0 {
        return 0, 0, nil, errInvalidKey.GenWithStack("invalid key -
%q", key)
    }
    indexValues = key[prefixLen+idLen:] //remain indexValues
    return tableID, indexID, indexValues, nil
}
```

测试结果:

```
(base) [xiejian@xiejian-optiplex3070 tablecodec]$ go test .
ok      github.com/pingcap/tidb/tablecodec    0.008s
```