

# 한 번에 끝내는 블록체인 개발 A to Z

---

Chapter 1

Blockchain 1.0 - Bitcoin

Chapter 1

Blockchain 1.0 - Bitcoin

# LevelDB

# Key-Value Database

- Google에서 개발한 빠르고 가벼운 Key-Value DB
- C++ 언어로 구현됨
- File 형태로 관리되며, RDBMS와 다른 형태로 구현
- 읽기, 쓰기 성능이 빠르다.
- 제한점
  - 1) NoSQL 데이터베이스이기 때문에, 관계형 검색 불가능
  - 2) Single Processing : 하나의 프로세스만이 한번에 특정 데이터베이스에 접근 가능

# LevelDB Sample Code



```
import leveldb
db = leveldb.LevelDB('test_leveladb',create_if_missing=True)

#Data 입력
db.Put('apple'.encode(), '10'.encode())

#Data 조회
get_result = db.Get(b'apple')
print(get_result.decode())

#Data 삭제
db.Delete(b'apple')
```

# Bitcoin 디렉토리 구조

File Name	Field
Banlist.dat	금지된 Node의 IP 리스트 저장
bitcoin.conf	bitcoind Config 파일
bitcoind.pid	Process Id값 저장 파일
blocks/blk000???.dat	Block Data(128MB)
blocks/rev000???.dat	사용된 UTXO 리스트 데이터
blocks/index/*	Block Index 정보 파일(LevelDB)
chainstate/*	블록체인 상태 정보 DB(LevelDB) UTXO 저장
database/*	Wallet 관련 정보를 저장하는 파일

# Bitcoin 파일 디렉토리 구조

File Name	Field
db.log	Wallet Database 관련 Log 파일
debug.log	Bitcoin의 Debug Log 파일
fee_estimates.dat	네트워크에 적합한 최소 수수료 저장 파일
indexes/txindex/*	Transaction data Index 파일(LevelDB)
mempool.dat	Mempool에 있는 Transaction Dump한 파일
peers.dat	Peer의 IP Address 관리 Database
Wallet 관련 디렉토리	Locktime

# LevelDB 저장 구조

Key	Value	Description
'b' + 32-byte block hash	Block header, 블록높이, Tx 수, block Validation 여부, block Data 저장 파일 위치, rev 데이터 저장 파일 위치	Block Index 기록
'f' + 4-byte file number	파일 내 블록 수, 블록 파일 크기, rev 파일 크기, 블록 파일 내 블록 최고*최저 높이, 블록 파일 내 최고*최저 시간	파일 정보 기록
'l' -> 4-byte file number	마지막 블록 파일 숫자 Blk + 0010.dat	
'R' -> 1-byte boolean	1(True) 0(False)	Reindexing 여부
'F' + 1-byte flag name length + flag name string	1(True) 0(False)	Txindex On/Off 여부
't' + 32-byte transaction hash	블록 파일 넘버, 파일 내 offset 위치, 블록 내 offset 위치	Transaction index 기록 (TxIndex On인 경우에만)

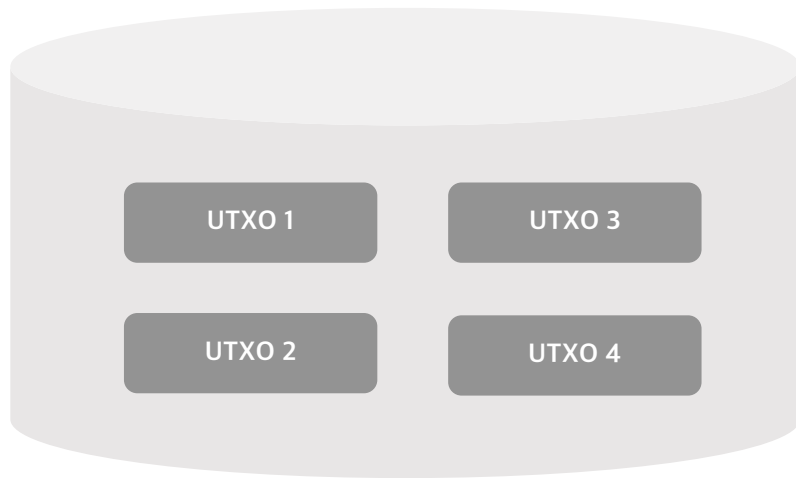
# LevelDB 저장 구조

Key	Value	Description
'c' + 32-byte transaction hash	Tx Version, Coinbase Tx 여부, Tx가 포함된 블록 높이, Tx 내 UTXO, UTXO의 scriptPubkey와 amount	트랜잭션 내 UTXO 데이터 조회용 (UTXO가 남은 경우)
'B' -> 32-byte block hash	가장 최신 Block Header Hash	최신 Block이 있는지 확인 용



# Mempool

- 아직 블록에 포함되지 않은 Pending Transaction들을 저장 및 관리하는 방법
- 채굴자들은 Mempool 중에서 Transaction을 선택해서 신규 Block에 포함시킨다.
- Mempool 에 들어가고도 14일동안 처리 되지 않고 남아있는 Transaction은 Expired 된다.



LevelDB

# Bitcoin 검색의 한계

- 우리가 일반적으로 생각하는 검색의 조건은 단순검색, 범위 검색, 조건 검색등이 있다.
- 하지만 Bitcoin 은 Key-Value 기반 Database를 사용하고 있기 때문에, 검색에는 Key값으로만 검색이 가능
- 과거에는 Bitcoin-Qt, Paper wallet 등이 많이 사용되어 있는데, 이는 실제로 Transaction이 전송 중인지, 내가 전송을 잘 받았는지, 6-Confirm이 지났는지에 대한 확인이 어렵다.
- 이런 검색 측면에서의 문제점을 개선하고자 나온 것이 Explorer 사이트이다.  
(대표적으로 bitinfochart.com, blockchain.info 등이 존재)