

# Real-time IFRS system implementation with PHP and MySQL

## [초록]

현재 많은 회계 시스템은 여전히 많은 데이터와 정보를 필요로 하며 기존 데이터 처리와 같이 스프레드 시트로 처리됩니다. 회계시스템이 가리키는 것은 IFRS 를 의미합니다. IFRS 이란, 회계 시스템은 기업의 금융 시스템에 적용된 국제 회계 표준입니다.

회계라는 정보에 데이터베이스시스템을 적용한다면 실시간성을 가질 수 있다고 봅니다. 데이터베이스 시스템이 갖는 특성을 활용하여 연구를 수행하였습니다.

MySQL 5.0.77-community-nt 데이터베이스 프로그램이 지원하는 데이터베이스 엔진에는 MyISAM, 메모리, BLACKHOLE, ARCHIVE, FEDERATED 및 MRG\_MYISAM 이 있습니다. MyISAM 엔진을 적용하여 기존의 스프레드시트가 처리할 수 있는 자료의 한계 등을 개선하고 실시간성을 가질 수 있는 시스템을 PHP 와 MySQL 을 활용하여 개선하였습니다.

## [키워드]

**Index Terms**— Object-oriented programming, DBMS, PHP Programming, IFRS, International Financial Reporting System

## 정도윤(예명: Narae Kim)

### [소개]

현재의 관계형 데이터베이스는 MySQL, MSSQL, Cubrid, Oracle 등 종류만 해도 수 십 가지가 된다. 이러한 데이터베이스에서의 핵심은 엔진이라고 할 수 있다. MySQL 5.0.77 커뮤니티-NT 버전에서 지원하는 엔진은 크게 6 가지로 구성된다. MyISAM, BLACKHOLE, ARCHIVE, FEDERATED, MRG\_MYISAM 이 있다. 기존에 스프레드시트가 가지는 자료구조는 파일 기반의 자료를 가지고 있다. 예를 들면, xlsx 등의 단일 파일로 구성되어 있으며 시트라는 단위로 데이터베이스를 구성하고 있다. Xlsx 이 가지는 한계는 행과 열의 메모리 제한사항이 존재한다는 점이다. 스프레드시트에서 가장 많이 사용하는 확장자인 xlsx 로 처리하면 문제점이 프로그래밍 처리가 복잡하고 매우 느리다는 점이다. 이러한 문제를 개선하기 위해서는 자료구조 등의 방법으로 해결할 수 있다고 주장한다. IFRS 란 국제회계기준으로 회계보고서 규격이다. 재무제표라고도 불리는 IFRS 시스템에는 크게 4 가지 항목을 만족해야 한다. 재무상태표, 손익계산서, 현금흐름표, 자본변동표가 있다. 이러한 4 가지 항목의 보고서를 생성하기 위해서는 기초 정보가 있어야 한다. 가계부라고 불리는 기본정보가 최소 1 개는 있어야 한다. 회계보고서는 작게는 경영자, 투자자, 이해관계자 등으로부터 재무상태에 대한 유용한 정보를 도출할 수 있는 시스템을 말한다. PHP, MySQL 의 프로그램을 활용하면, 스프레드시트 구조에서 생성되는 기존의 문제점을 개선하고 더 효율적인 지식 정보 처리가 가능할 수 있을 거라고 보인다.

### [1. PHP: Hypertext Preprocessor]

PHP 는 하이퍼텍스트 기반의 프로그래밍 언어로서 객체지향 프로그래밍으로 코드를 쉽게 작성할 수 있다. PHP 를 사용하는 방법을 소개하는 코드로써, 0 부터 5 까지 출력하는 코드이다.

```
<?php
    $i = 1;
    $max = 5;
    for ( $i = 0; $i < $max; $i++)
        echo $i . "<br>";
?>
```

PHP 로 객체지향 프로그래밍 형태로 코드 작성하는 방법이다.

```
<?php
    Class hama{
        public function __construct(){
        }
        public function __destruct(){
        }
    }
?>
```

PHP 로 MySQL 기반의 프로그래밍을 했을 때 사용되는 코드이다.

```
<?php
// we connect to example.com and port 3307
$link = mysql_connect('example.com:3307',
    'mysql_user', 'mysql_password');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);

// we connect to localhost at port 3307
$link = mysql_connect('127.0.0.1:3307', 'mysql_user',
    'mysql_password');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
```

```
mysql_close($link);
?>
```

PHP 를 사용하면, 쉽고 빠르게 프로그램을 개발할 수 있다.

### [2. MySQL] - MyISAM

MySQL 은 관계형 데이터베이스 프로그램입니다.  
MySQL5.096 에서의 MyISAM 은 기본 스토리지 엔진입니다.  
구형 (더 이상 사용할 수없는) ISAM 저장소 엔진을 기반으로하지만 많은 유용한 확장 기능을 제공합니다.  
각 MyISAM 테이블은 3 개의 파일로 디스크에 저장됩니다.  
파일의 이름은 테이블 이름으로 시작하고 파일 유형을 나타내는 확장자가 있습니다. .frm 파일은 테이블 형식을 저장합니다. 데이터 파일의 확장자는 .MYD (MYData)입니다. 색인 파일에는 .MYI (MYIndex) 확장자가 있습니다.

MyISAM 테이블을 명시 적으로 지정하려면 ENGINE 테이블 옵션을 사용하여이를 지정하십시오.  
CREATE TABLE t (i INT) ENGINE = MYISAM;  
이전 용어 TYPE 은 이전 버전과의 호환성을 위해 ENGINE 과 동의어로 지원되지만 ENGINE 이 선호되는 용어이고 TYPE 은 더 이상 사용되지 않습니다.

mysqlcheck 클라이언트 나 myisamchk 유틸리티로 MyISAM 테이블을 점검하거나 복구 할 수있다.  
myisampack 으로 MyISAM 테이블을 압축하여 훨씬 적은 공간을 차지할 수도 있습니다.

MyISAM 테이블의 특징은 다음과 같습니다.

- 모든 데이터 값은 먼저 낮은 바이트로 저장됩니다. 이렇게하면 데이터 시스템과 운영 체제가 독립적입니다. 바이너리 이식성에 대한 유일한 요구 사항은 기계가 2 의 보수 부호있는 정수와 IEEE 부동 소수점 형식을 사용한다는 것입니다. 이러한 요구 사항은 주류 기계간에 널리 사용됩니다. 바이너리 호환성은 임베디드 시스템에는 적용되지 않을 수 있으며 때로는 특유의 프로세서가 있습니다.
- 낮은 바이트 먼저 데이터를 저장하는 데 상당한 속도 저하가 없습니다. 테이블 행의 바이트는 일반적으로 정렬되지 않고 정렬되지 않은 바이트를 역순으로 읽는 데 약간의 처리가 필요합니다. 또한 열 값을 가져 오는 서버의 코드는 다른 코드에 비해 시간이 중요하지 않습니다.
- 더 나은 인덱스 압축을 위해 모든 숫자 키 값이 먼저 높은 바이트로 저장됩니다.
- 대형 파일 (최대 63 비트 파일 길이)은 대용량 파일을 지원하는 파일 시스템 및 운영 체제에서 지원됩니다.
- MyISAM 테이블에는 최대 232 행 (~ 4.295E + 09)이있다.

### 3 B+TREE : DATA STRUCTURE

컴퓨터 과학에서 B+ 트리(Quaternary Tree)는 키에 의해서 각각 식별되는 레코드의 효율적인 삽입, 검색과 삭제를 통해 정렬된 데이터를 표현하기 위한 트리 자료구조 중 하나이다. B+트리는 동적이며, 각각의 인덱스 세그먼트 (보통 블록 또는 노드) 내에 최대와 최소범위의 키의 개수를 가지는 다계층 인덱스(multilevel index)로 구성된다.

B 트리와 대조적으로 B+트리는, 모든 레코드들이 트리의 가장 하위 레벨에 정렬되어있다. 오직 키들만이 내부 블록에 저장된다.

B+트리에서 중요한 가치는 블록-지향적인 storage context(예: filesystem)에서 검색을 효율적으로 할 수 있다는 점이다. 바이너리 서치 트리보다 B+트리 노드의 fanout(한 노드의 자식 노드의 수)이 훨씬 높아서 검색에 필요한 I/O 동작 회수를 줄일 수 있기 때문이다.

위의 파일시스템들은 모두 블록 인덱싱을 위해 B+트리 타입을 사용한다. 관계 데이터베이스들도 테이블 인덱스를 위해 B+트리 타입을 가끔 사용한다.

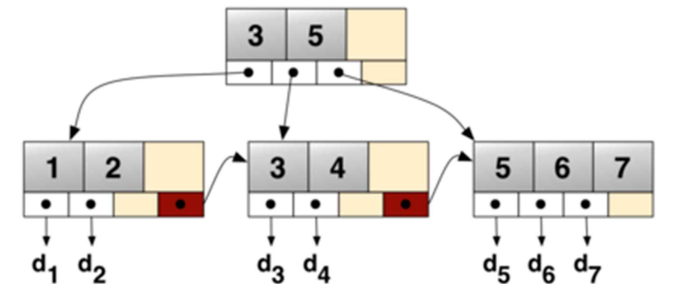


그림 1. B+Tree 구조

### 4 SPREADSHEET

오늘 날 스프레드시트는 정말 많은 곳에서 광범위하게 사용되고 있다.

이러한 스프레드시트에서 핵심적으로 구성하고 있는 것은 시트의 행과 열이다.

스프레드시트의 예로 마이크로소프트 오피스 엑셀 2007 의 경우에는 메모리 제한을 두고 있다.

기능	최대 한도
Open workbooks	사용 가능한 메모리와 시스템 리소스에 따라 제한됨
워크시트에서 행과 열의 총 수	1,048,576 행 x16,384 열
열 너비	255 자
행 높이	409 포인트
페이지 나누기	가로/세로 1,026 개
프로세서 코어	64

그림 2. 워크시트와 통합 문서 사양 및 제한

### 5 IFRS: INTRODUCE

회계기준이 되고 있는 것은 IFRS 이다.

IFRS (International Financial Reporting Standards)은 국제 회계 기준위원회 (IASB)가 국제 회계 기준을 이해하고 비교할 수 있도록 업무 관련 공통 언어를 제공하기 위해 발행 한 표준입니다. 그것들은 국제적인 주식 보유 및 무역의 증가의 결과이며 여러 나라에서 거래를 하는 기업에게 특히 중요합니다.

### 6. Ifrs : Elements of financial statements

재무 상태 보고서의 측정과 직접 관련된 요소는 다음과 같다고 볼 수 있다. 자산은 과거 사건으로 인하여 자산이 통제하고 미래 경제적 효익이 기업에 유입될 것으로 기대되는 자산이다.

부채는 과거 사건으로 인해 발생하는 기업의 현재 의무이며, 그 정산으로 인해 경제적 효익이 담보된 자산, 즉 자산이 유출될 것으로 예상하는 시점이다.

자본은 모든 부채를 명목 가치로 차감한 후 자산에 대한 잔여 지분이라고 한다.

손익계산서(Statement of Profit / Loss)란 재무 성과를 보고하는 문서이다. 이 문서에는 크게 수익과 비용, 세금의 항목으로 재무 성과의 결과를 도출할 수 있다. 비용은 유출, 자산의 고갈 또는 자본의 감소를 초래하는 부채 발생과 같은 회계 기간 경제적 이익의 감소이다.

## 7 IFRS: REQUIREMENT

IFRS 는 4 가지 항목과 필수요소와 주서를 갖추게 되면 기능적으로 성립이 된다.

데이터 기반의 지식 정보 처리를 위해서는 재무 제표, 손익 계산서와 포괄 손익 계산서, 자본변동표, 현금 흐름표를 요구한다. 참고로 IFRS 재무제표는 (IAS1.8)에 의해 구성된다.

Fig 3 은 재무상태표에 관한 보고서 양식 및 식 구조를 작성하였다.

Financial Statement	
Asset	Debt = {categoryDebt1 + categoryDebtn}
{categoryAsset1}	{categoryDebt1}
{categoryAsset2}	{categoryDebtn}
{categoryAsset3}	Capital = {categoryCapital1} + {categoryCapital2} + {categoryCapitalSP}
{categoryAsset4}	{categoryCapital1}
...	{categoryCapital2}
{categoryAssetn}	{categoryCapitalSP}
	= $\sum \text{categoryAsset} - \sum \text{categoryDebt} - (\text{categoryCapital1} + \text{categoryCapital2})$
Asset = $\sum \text{categoryAsset}$	Asset = if ( Asset == (Debt + Capital) }

Fig 3. Financial Statement

그림 4 는 손익과 손실 상태표를 논리식으로 표현한 보고서 양식을 작성하였다.

Profit and loss statement (\$startDate)~ (\$endDate)	
Income	$\sum \text{categoryValue I}$
$\sum \text{category I}$	CategoryValueI{n}

Expenditure	$\sum \text{categoryValueE}$
$\sum \text{categoryE}$	CategoryValueE{n}
Profit and loss (Tax income before)	Income - Expenditure
Tax	TaxValue
Profit and loss (Tax deducted)	(Income - Expenditure) - Tax

Fig 4. Profit and loss statement

그림 5 은 현금흐름표를 작성한 것이다. 몇 가지 구성으로 논리식을 수립했다. 현금흐름표에서 다소 복잡한 점은 시작일자 시점의 현금흐름에 대한 정보가 하나 필요하다.

Cashflow statement (\$startDate)~ (\$endDate)	
1. Income	Income - Expenditure
Cost without cash flow	
- Depreciation cost	\$value1
- Loss on disposal of fixtures	\$value2 \$lod = \$value1 + \$value2
Revenue without cash flow	
- Interest revenue	\$irval = \$value
Changes in assets and liabilities due to operating activities	
- Accounts Receivable	\$value1
- Inventories Payable	\$value2
- Accounts Payable	-\$value3 \$account Pay = \$value 1 + \$value2 + \$value3
Cash generated from sales	\$cash = \$income - \$lod - \$irval - \$account Pay
- Interest payment	\$iPay = \$value1

- Tax payment	\$tax = \$tResult \$value2 = \$cash - \$iPay - \$tax
2. Investment activity cash flow	
Cash inflows from investing activities	
- Disposal of fixtures	\$value
Addition of expenses without cash outflow	
- Acquisition of equipment	-\$value \$result = \$dof + \$aoe
...	
Cash and cash equivalents	\$tResult + \$result
startDate{ Cash and cash equivalents}	\$query1
endDate{ Cash and cash equivalents}	\$query2

Fig 5. Cash flow statement

그림 6 은 자산변동표에 관한 것이다.

Changes in equity (\$startDate)~ (\$endDate)			
	Capital	Retained Earnings	Total
\$startDate			
{ \$capital.category1 }	\$value1		\$value1
{ \$capital.category2 }	\$value2		\$value2
{ \$capital.categoryn }	\$valueN		\$value3
Net income / Net loss		{ \$income and loss .profit }	{ \$income and loss .profit }
Cash Divided		\$capital. divided	$\sum$ cashDiv
Total \$endDate	$\sum$ capital	$\sum$ earning	\$capital + \$earnin g

Fig 6. Changes in equity

다소 복잡한 항목으로 구성되어 있으나

## 8 IMPLEMENT

IFRS 의 “재무상태표”, “손익계산서”, “현금흐름표, 자본변동표”에 대해서 공통으로 사용할 수 있는 추상형 자료 구조를 고안하였다.

```

abstract class IAccount{
    abstract public function sumOfvalue($type,
    $category, $boardName, $startDate, $endDate);
    abstract public function
    prevBalance($boardName, $targetDate);
    abstract public function
    listContent($boardName, $startDate, $endDate,
    $userName, $balance);
    abstract public function write($boardName,
    $account);
    abstract public function modify($boardName,
    $account);
    abstract public function remove($boardName,
    $account);
}

```

Fig 7. IAccount - 추상형 구조

ADT 에 있는 함수 “sumOfvalue”의 경우에는 합계에 대한 데이터를 도출할 수 있다. 함수 “prevBalance()”는 이전의 잔액 합계에 대해서 출력할 수 있다. 함수들이 공통으로 가지는 중요한 점은 일자에 관한 정보이다.

## 9 CATEGORY

IFRS 보고서의 중복된 기능 항목을 최소화하고 항목을 아래의 구조로 분류하였다.

```

Class Category{
    private $minSizeType;
    private $maxSizeType;

    private $minSizeACategory; // 수입(Income) - 최소
    private $maxSizeACategory; // 수입(Income) - 최대
    private $minSizeBCategory; // 지출(Outcome) - 최소
    private $maxSizeBCategory; // 지출(Outcome) - 최대
    private $minSizeCCategory; // 자본(Capital) - 최소
    private $maxSizeCCategory; // 자본(Capital) - 최대
    private $minSizeDCategory; // 부채(Debt) -

```

```

최소
private $maxSizeDCategory;      // 부채(Debt) - 최대
private $minSizeECategory;      // 자산(Asset) - 최소
private $maxSizeECategory; // 자산(Asset) - 최대
}

```

Fig 8. Category - Pseudocode

그림 8 은 카테고리에 관한 항목들을 공통으로 분류하기 위해서 min, Max 의 형태로 크기를 결정할 수 있다. 이러한 카테고리의 구조는 IFRS 에 필요한 AccountBook 이 가지고 있는 지식기반의 데이터를 처리하기 위해서 중요하다.

## 10 CATEGORY

그림 9 는 AccountBook 에 대한 ADT 설계이다.

```

class AccountBook{
    private $id;
    private $type;
    private $category;
    private $accountName;
    private $accountNo;
    private $subject;
    private $price;
    private $taxRate;
    private $count;
    private $regidate;
}

```

Fig 10. AccountBook - abstract type structure

AccountBook 의 정보를 통해서 IFRS 의 보고서를 효율적으로 도출할 수 있다.

## 11 데이터처리를 위한 SQL

listContent() 함수 내에 구현된 SQL 에 관한 코드이다.

```

$query = "SELECT id, type, category, accountName,
accountNo, " .
        "subject, price, taxRate, count, regidate
" .
        "FROM account_ %s WHERE regidate
BETWEEN '%s' AND '%s' " .
        "ORDER BY regidate, id ASC");

```

Fig 10. listContent() 함수에 구현된 SQL

“prevBalance(\$boardName, \$target Date)” 함수 내 구현된 SQL 코드이다.

```

$query = "SELECT id, type, subject, accountName,
accountNo, " .
        "price, taxRate, count, regidate " .
        "FROM account_ %s WHERE regidate

```

```

BETWEEN '%s' AND '%s' ORDER BY regidate, id ASC";

```

Fig 11. prevBalance() 함수에 구현된 SQL

“prevBalance()” 함수 내에서 SQL 문을 지탱해주는 논리식을 소개한다.

이전의 합계를 알기 위해서는 이전의 정보에 대한 합이 요구된다.

```

while ($row = mysql_fetch_array($result,
MYSQL_ASSOC)) {

    $sumOfPrice = $row['price'] * $row['count'];
    $sumOfPriceTax = $sumOfPrice * $row['taxRate'];
    $resultOfPrice = $sumOfPrice + $sumOfPriceTax;

    if ($index == 0){
        $tmp_price = $resultOfPrice;
    }

    if ($index != 0){
        $resultOfPrice = $accountFn-
>calculate($row['type'], $tmp_price, $resultOfPrice );
        $tmp_price = $resultOfPrice;
    }

    $index = $index + 1;
}

```

Fig 12. prevBalance() - Logics

그림 13 에는 ifrsSystem 에 관한 방법을 몇 가지의 형태로 표현하였다.

```

/// 0. 공통 임시변수
$usrType;
$category;
/// 1. 대차대조표(Balance Sheets)

// 차변(왼쪽 - Left)
$usrType = $accountFn-
>getSearchTypeID('자산(Asset)');
$b_l_balance = $jasper->prevBalance($boardName,
$endDate); // 잔액(Balance)

$category = $accountFn-
>getSearchCategoryID($usrType, '토지(Land)');
$b_l_land = $jasper->sumOfvalue($usrType,
$category, $boardName, $initialStartDate, $endDate);
$category = $accountFn-
>getSearchCategoryID($usrType, '매출채권(Accounts
Receivable)');
$b_l_ar = $jasper->sumOfvalue($usrType, $category,
$boardName, $initialStartDate, $endDate);

```

.....

그림 13의 로직은 ifrsSystem에 관한 로직이다. “7. Requirement”에서 소개된 로직을 컴퓨터 코드로 변환하였다. 조회를 여러 번 할 것으로 보인다. 그림 13의 슈도코드에서는 가독성을 사람이 이해할 수 있는 형태로 표현하였다.

## 12 DATA MODEL

그림 8을 바탕으로 타입과 카테고리에 관계 모델을 그림 14와 같이 정의하였다.

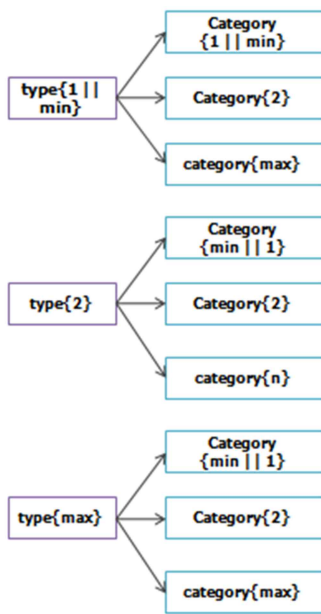


Fig 14. 타입과 카테고리의 관계

타입과 카테고리의 관계는 IFRS 보고서를 생성하는 데 있어서 매우 중요한 역할을 수행한다. 재무상태표, 손익계산서, 현금흐름표, 자본변동표에서 공통으로 사용될 수 있는 분류이다. 이러한 분류 체계를 정립함으로써 데이터의 중복성을 제거할 수 있었다.

## 13 PERFORMANCE

퍼포먼스는 지식 기반의 데이터처리에 있어서 중요하다고 할 수 있다. IFRS 보고서를 출력하는데 걸리는 시간을 측정했다.

번호	구분	개수	초(화면)	초(PHP)
1	Website	100,000	5.5 초	0.000771
2	Website	250,000	5.5 초	0.000610
3	Website	500,000	26 초	0.000752
4	Website	1,040,350	51 초	0.000608

Fig 15. 퍼포먼스 측정 결과

PHP와 "Print"에 관한 결과를 차트로 출력하였다.

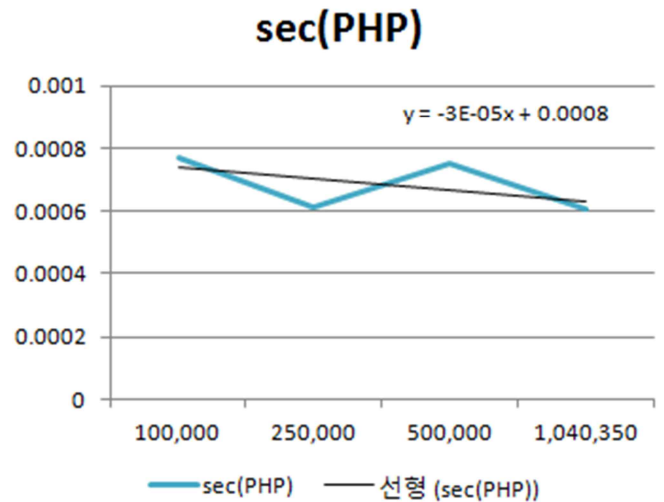


Fig 16. PHP에서 측정된 결과(단위: 초)

그림 16에서 측정된 결과는 실제 그림 15에 있는 개수를 MySQL에 Insert 명령어로 삽입하였다. 실제 컴퓨터에서 해당 데이터처리를 하였을 때 수행한 결과이다.

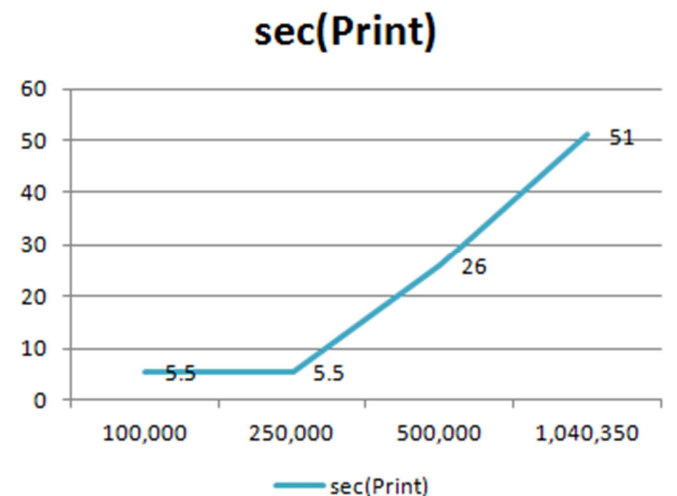


Fig 17. 사용자 화면에서 측정된 결과(단위: 초)

그림 17은 그림 15에 있는 데이터로 도출한 결과이다. 그림 16과 그림 17은 하나의 실험에서 얻어진 데이터이다. “sec(print)”와 “sec(PHP)”가 가지는 차이점을 sec(print)의 경우에는 실제 사용자의 체감시간을 측정한 것이고, sec(PHP)는 PHP에서 구동되었을 때의 시간을 측정한 것이다.

그림 17을 분석해보면, 50만 건과 100만 건의 조회에서 약 2배 정도의 실험 결과를 얻어낼 수 있었다.

## 14 CONCLUSION

국제회계표준 보고서를 생성하는 데 있어서 개별적인 정보 분류 등의 낭비되는 문제를 개선할 수 있었다.

더불어 회계표준 보고서를 실시간으로 처리함으로써 데이터베이스가 가지는 무결성 등의 특징이 적용되어 지식정보를 처리할 수 있는 기반을 정립하였다.

이러한 특징을 기존의 스프레드시트로 처리했던 IFRS 보고서 생성보다 데이터마이닝을 효율적으로 할 수 있을 것으로

보인다. IFRS의 보고시스템이 좀 더 나은 퍼포먼스를 구현할 수 있다면 더욱 생산성이 높은 데이터처리를 기대할 수 있을 것이다.

조금 더 응용한다면, 기업을 넘어서 개인의 가계부 등에서도 충분히 IFRS 시스템이 구현될 수 있을 것이다.

## 15. Acknowledgment

저자는 해당 연구를 수행하면서 부모님께 감사의 말을 전하고 싶다.

<https://downloads.mysql.com/docs/refman-5.0-en.pdf>,  
*Accessed by 2018-08-05*

<http://php.net/manual/en/function.mysql-connect.php>,  
*Accessed by 2018-08-05*

[https://en.wikipedia.org/wiki/B+\\_tree](https://en.wikipedia.org/wiki/B+_tree), Accessed by  
2018-08-05

<https://support.office.com/ko-kr/article/excel-%EC%82%AC%EC%96%91-%EB%B0%8F-%EC%A0%9C%ED%95%9C-1672b34d-7043-467e-8e27-269d656771c3>, Accessed by 2018-08-05

<https://support.office.com/en-us/article/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3?ui=en-US&rs=en-US&ad=US>,  
Accessed by 2018-08-05

[https://en.wikipedia.org/wiki/International\\_Financial\\_Reporting\\_Standards#Concepts\\_of\\_capital](https://en.wikipedia.org/wiki/International_Financial_Reporting_Standards#Concepts_of_capital), Accessed by  
2018-08-05