OPTIMALBI EXERCISE

SOLUTIONS

By:

Subhramitra Borkakati

The three most important things about code documentation, according to me are:

A code's purpose

The code should be documented enough that anyone who is trying to use that code, modify that code or review that code should understand what the code is supposed to do and not do, without going through each line of the code. A few things that I feel are very important while documenting a code are:

- Using a documentation tools like JavaDoc or JSDoc. Not only they do make it easier to document the params, return values and exceptions, IDEs can read these documentations and provide the developer with the relevant information about the code without even opening the file. I personally like this a lot.
- Providing comments for sections of the code that have many conditional statements or runs some complex algorithm. This little effort comes in handy while refactoring or rewriting the code even when it is done my the same person who wrote the original code; for instance, I had to rewrite the Java 7 code in Java 8 and to use the appropriate Java 8 api, a good understanding of the original code is a must.
- On a side note, I personally, consider the Java API as a very good example of how code should be documented.

A developer's promise – Unit tests

- Unit tests, in my opinion and experience, are one of the most important pieces of documentation (in the form of code) for a code. They ensure that the developer delivers what he or she has promised to deliver. Unit tests are technical documentation for other developers to understand the intent and purpose of the code and thus, if need be, refactor the code without losing sleep!

What, why and when of the code execution - Logging

- This is coming from the standpoint of a production support engineer. During my short time as a support engineer, I performed quite a few root cause analyses on interesting production issues and often, either I don't find any trace of the error or exception in the logs at all or the information logged is not helpful enough to go forward. Proper logging is also important from security perspective (PII). Thus, I consider logging an important part of code documentation.

Best approach: Using standard JavaScript built-in object – NumberFormat.

Reasons:

1. It's a JavaScript built-in object, which means it has good browser compatibility and we need not bloat our application package with external dependencies.

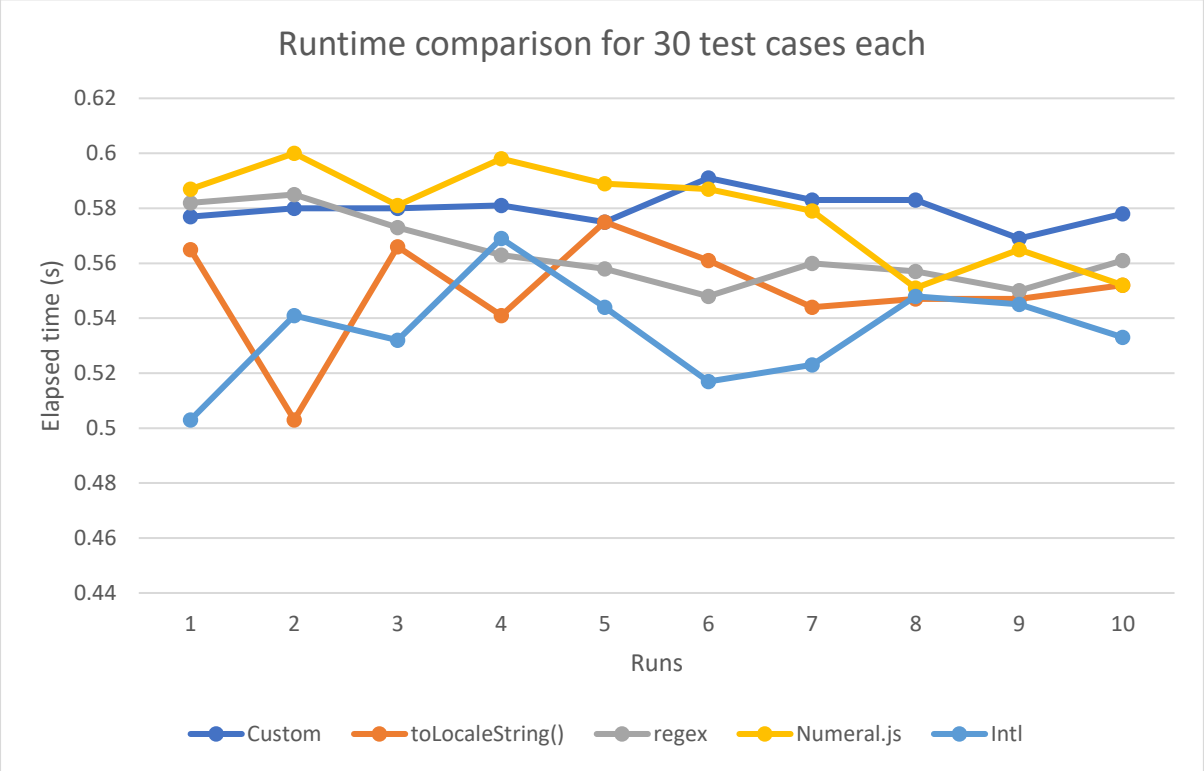
2. Additionally, it supports wide range of parameters for custom formatting and i18n for currencies.

3. *Intl.NumberFormat.prototype.format* property returns a getter function, which enables us to apply the function on individual array elements and format an array of numbers conveniently.

Summary:

Approach	Pros	Cons
Intl.NumberFormat.prototype.format or Number.prototype.toLocaleString()	1. JavaScript built-in object 2. Supports wide range of parameters for custom formatting and i18n of currencies 3. Apply formatting on array of numbers 4. Good performance	1. Does not support change values such as 99¢ etc. 2. Input needs to be sanitized and validated before passing in.
Custom implementation	1. Supports change values like 99¢. 2. Unsanitized inputs are handled gracefully. 3. Easy to understand, debug and customizable code 4. Unit tested 5. Comparable performance with respect to other approaches. 6. No external dependency	1. Development and Maintenance overhead 2. Does not support a wide range of customization options. 3. Does not support i18n
Regex	1. Concise code 2. No external dependency.	1. Difficult to understand, debug and customize 2. Input needs to be sanitized 3. Does not support i18n
Numeral.js	1. Provides a wide range of options to format numbers as per the requirement. 2. Unit tested 3. Useful as a one stop solution if the application requires a variety of number formatting.	1. Overkill for our purpose 2. External dependency 3. Difficult to understand and customize 4. Not actively maintained

Please note, the run time indicates the time taken for each function to complete all 30 test cases in their respective suite. Also, all the functions, except the custom implementation, contain the bare minimum code required for formatting numbers.



Steps to test the custom implementation:

1. Inside OptimalBI -> Formatter: Perform **npm install**

2. Run the following command to execute the test suite: **jest formatter.test.js**

3. Run the following command to execute all the 5 test suites: **jest test**

NOTE: The code for all other methods can be found inside the '**theOthers'** directory.

Code snippet:

```
const CURRENCY_DELIMITER = ','
const DECIMAL_DELIMITER = '.'
const PRECISION = 2
const THRESHOLD = 1000
const PAD_LIMIT = 3
/**
 * Formats the provided number and returns a string representation
of the number based on the provided options
 * @example
 * // returns -99¢
 * formatter(-.99)
```

```
 * @example
 * // returns $1,234,567.00
 * formatter(1234567)
 * @param {number} number - The number to format
 * @param {Object} options - provide currency and change symbols
to use. Example: {currency: '$', change: '¢'}
 * @returns {string}
 * @throws logs an error and returns the input, if the input type
is not of type {number}
 */
formatter = (number, options = {currency: '$', change: '¢'}) => {
    console.log(`INFO:: Formatting number...`)
    if (typeof number !== 'number') {
        console.log('ERROR::Received input is not a Number.')
        return number
    }
    if (!Number.isSafeInteger(number)) {
        console.log('WARNING: Number exceeds the safe integer
value of ± 9007199254740991. Precision of the result will be
effected.')
    }
    const { currency, change } = options
    const sign = Math.sign(number) === -1 ? '-' : ''
    // Extracting the cents value and storing it to be
concatenated to the final result
    const cents = (number -
Math.trunc(number)).toFixed(PRECISION).toString().split(DECIMAL_DE
LIMITER)[1]
    let dollars = Math.trunc(number)
    if (dollars === 0) {
        return sign + cents + change
    }
    dollars = format(Math.abs(dollars)).slice(0, -1)
    console.log(`DEBUG::Formatted ${number} is ${currency +
dollars + DECIMAL_DELIMITER + cents}.`)
    console.log(`INFO::Formatting complete.`)
    return currency + sign + dollars + DECIMAL_DELIMITER + cents
}

/**
 * Divides the input number by the set THRESHOLD and
```

```
 * then recursively divides the result with the THRESHOLD until
the result is 0.
 * When result reaches 0, the remainder of the division operation
is returned with the delimiter appended.
 * @param {number} number - The integer part of the original
number to format
 * @returns {string} Formatted number
 */
format = (number) => {
    if (number < THRESHOLD)  {
        return number + CURRENCY_DELIMITER
    }
    const quotient = Math.floor(number / THRESHOLD)
    const remainder = number % THRESHOLD
    //The remainder should always be of 3 digits so we are
explicitly padding it with zeros.
    //For example: Dividing 1003 by 1000 will give us 3 as
remainder but we need 003 for our formatting
    let res =
format(quotient).concat(remainder.toString().padStart(PAD_LIMIT,
'0')).concat(CURRENCY_DELIMITER)
    return res
}

module.exports = formatter
```

Database: MySQL (Community v8.0.12, NOTE: REGEX functions may not work on older versions)

SQL queries used:

1. Create table - Had to allocate space for an extra character for **month_name** because of dirty data in source CV and no application layer code to handle it before hand

```
create table CustomerSales (
        trans_date date primary key,
   cust_name varchar(10),
   product char(7),
   price tinyint
);

create table CustomerSalesDates (
        trans_date date primary key,
   day_name varchar(9),
   month_name varchar(9),
   quarter char(2)
);
```

2. **Insert data from CSV** - **LOAD DATA INFILE** was not working and **data set was small**, hence decided to do it manually instead of debugging LOAD DATA INFILE)

CustomerSales records

```
insert into CustomerSales values(20180101,'Tim','Widget1',10);
insert into CustomerSales values(20180102,'Geoff','Widget2',7);
insert into CustomerSales values(20180103,'Fred','Widget3',5);
insert into CustomerSales values(20180105,'Bob','Widget5',8);
insert into CustomerSales values(20180107,'Sue','Widget7',20);
insert into CustomerSales values(20180109,'Cat','Widget9',2);
insert into CustomerSales values(20180110,'Jan','Widget0',9);
insert into CustomerSales values(20180111,'Allan','Widget1',10);
insert into CustomerSales values(20180112,'Robert','Widget2',7);
insert into CustomerSales values(20180113,'Chris','Widget3',5);
insert into CustomerSales values(20180114,'Sam','Widget4',16);
insert into CustomerSales values(20180116,'Gandolf','Widget6',6);
insert into CustomerSales values(20180117,'Tim','Widget7',20);
insert into CustomerSales values(20180118,'Geoff','Widget8',17);
insert into CustomerSales values(20180119,'Fred','Widget9',2);
insert into CustomerSales values(20180120,'Bob','Widget0',9);
insert into CustomerSales values(20180121,'Sue','Widget1',10);
insert into CustomerSales values(20180123,'Cat','Widget3',5);
insert into CustomerSales values(20180124,'Jan','Widget4',16);
insert into CustomerSales values(20180125,'Allan','Widget5',8);
insert into CustomerSales values(20180126,'Robert','Widget6',6);
insert into CustomerSales values(20180127,'Chris','Widget7',20);
insert into CustomerSales values(20180128,'Sam','Widget8',17);
insert into CustomerSales values(20180130,'Gandolf','Widget0',9);
insert into CustomerSales values(20180131,'Tim','Widget1',10);
```

```
insert into CustomerSales values(20180201,'Geoff','Widget1',10);
insert into CustomerSales values(20180202,'Fred','Widget2',7);
insert into CustomerSales values(20180203,'Bob','Widget3',5);
insert into CustomerSales values(20180205,'Sue','Widget5',8);
insert into CustomerSales values(20180206,'Cat','Widget6',6);
insert into CustomerSales values(20180207,'Jan','Widget7',20);
insert into CustomerSales values(20180209,'Allan','Widget9',2);
insert into CustomerSales values(20180210,'Robert','Widget0',9);
insert into CustomerSales values(20180211,'Chris','Widget1',10);
insert into CustomerSales values(20180212,'Sam','Widget2',7);
insert into CustomerSales values(20180213,'Gandolf','Widget3',5);
insert into CustomerSales values(20180214,'Hans','Widget4',16);
```

CustomerSalesDates records

```
insert into CustomerSalesDates values(20180101,'Monday','Jan!uary','Q1');
insert into CustomerSalesDates values(20180102,'Tuesday','January','Q1');
insert into CustomerSalesDates values(20180103,'Wednesday','January','Q1');
insert into CustomerSalesDates values(20180104,'Thursday','January','Q1');
insert into CustomerSalesDates values(20180105,'Friday','January','Q1');
insert into CustomerSalesDates values(20180106,'Saturday','January','Q1');
insert into CustomerSalesDates values(20180107,'Sunday','January','Q1');
insert into CustomerSalesDates values(20180108,'Monday','January','Q1');
insert into CustomerSalesDates values(20180109,'Tuesday','Jan!uary','Q1');
insert into CustomerSalesDates values(20180110,'Wednesday','January','Q1');
insert into CustomerSalesDates values(20180111,'Thursday','January','Q1');
insert into CustomerSalesDates values(20180112,'Friday','January','Q1');
insert into CustomerSalesDates values(20180113,'Saturday','January','Q1');
insert into CustomerSalesDates values(20180114,'Sunday','January','Q1');
insert into CustomerSalesDates values(20180115,'Monday','Janu?ary','Q1');
insert into CustomerSalesDates values(20180116,'Tuesday','January','Q1');
insert into CustomerSalesDates values(20180117,'Wednesday','January','Q1');
insert into CustomerSalesDates values(20180118,'Thursday','January','Q1');
insert into CustomerSalesDates values(20180119,'Friday','January','Q1');
insert into CustomerSalesDates values(20180120,'Saturday','Janu?ary','Q1');
insert into CustomerSalesDates values(20180121,'Sunday','January','Q1');
insert into CustomerSalesDates values(20180122,'Monday','January','Q1');
insert into CustomerSalesDates values(20180123,'Tuesday','January','Q1');
insert into CustomerSalesDates values(20180124,'Wednesday','January','Q1');
insert into CustomerSalesDates values(20180125,'Thursday','January','Q1');
insert into CustomerSalesDates values(20180126,'Friday','January','Q1');
insert into CustomerSalesDates values(20180127,'Saturday','January','Q1');
insert into CustomerSalesDates values(20180128,'Sunday','January','Q1');
insert into CustomerSalesDates values(20180129,'Monday','January','Q1');
insert into CustomerSalesDates values(20180130,'Tuesday','January','Q1');
insert into CustomerSalesDates values(20180131,'Wednesday','January','Q1');
insert into CustomerSalesDates values(20180201,'Thursday','February','Q1');
insert into CustomerSalesDates values(20180202,'Friday','February','Q1');
insert into CustomerSalesDates values(20180203,'Saturday','February','Q1');
insert into CustomerSalesDates values(20180204,'Sunday','February','Q1');
insert into CustomerSalesDates values(20180205,'Monday','February','Q1');
```

```
insert into CustomerSalesDates values(20180206,'Tuesday','Feb!ruary','Q1');
insert into CustomerSalesDates values(20180207,'Wednesday','February','Q1');
insert into CustomerSalesDates values(20180208,'Thursday','February','Q1');
insert into CustomerSalesDates values(20180209,'Friday','February','Q1');
insert into CustomerSalesDates values(20180210,'Saturday','February','Q1');
insert into CustomerSalesDates values(20180211,'Sunday','February','Q1');
insert into CustomerSalesDates values(20180212,'Monday','February','Q1');
insert into CustomerSalesDates values(20180213,'Tuesday','February','Q1');
insert into CustomerSalesDates values(20180214,'Wednesday','February','Q1');
```

Test:

```
select * from CustomerSalesDates;
```

3. Cleaning up dirty data in CustomerSalesDates table's month_name column – Initial plan was to just clean up the data when fetching i.e in SELECT statements but then I needed to use the **month_name field in the GROUP BY clause**, hence decided to clean up the data using UPDATE. Also, this **made the select statements leaner**.

```
update CustomerSalesDates set month_name = REGEXP_REPLACE(month_name, '[^a-zA-z]', '')
where trans_date > '2017-12-31';
```

Test:

```
select count(month_name) from CustomerSalesDates where REGEXP_LIKE(month_name, '[^a-zA-
z]');
```

4. Creating temporary table – Decided to create a **temporary table** that holds the following modified data that I will be using for the final query. This step helped me **break the problem** in steps and made the **final query leaner** – kind of like a helper or util method.

1. Day value **truncated** from trans_date

2. **Add sales price** grouped by cust_name and the modified trans_date. For example: Tim's total sales record for 201801 (January) is 40 while for Sam it's 33 for 201801 and 7 for 201802 (February)

3. **cust_name** values are unchanged

```
create temporary table monthly_customer
select DATE_FORMAT(cs.trans_date, '%Y%m') AS TRANS_DATE, sum(price) as sum_price,
cust_name
from customersales as cs
group by cust_name, DATE_FORMAT(cs.trans_date, '%Y%m') order by sum_price DESC;
```

Test:

```
select * from monthly_customer;
```

5. Final query: Performed sum of the (sum of) monthly customer sales grouped by the **cust_name**, **day_name**, **trans_date** and **month_name**.

```
select mc.trans_date, mc.cust_name, sum(mc.sum_price) as sum_price, csd.day_name,
csd.month_name, csd.quarter as qtr
from monthly_customer as mc
cross join CustomerSalesDates as csd
GROUP BY mc.cust_name, csd.day_name, mc.trans_date, csd.month_name
order by sum_price DESC;
```

Result: 336 rows returned (24 x 14 rows, 14 is the number of unique rows for day to month mapping and 24 rows from the temp table)

TRANS_DATE	CUST_NAME	SUM_PRICE	DAY_NAME	MONTH_NAME	QTR
201801	Tim	200	Monday	January	Q1
201801	Tim	200	Tuesday	January	Q1
201801	Tim	200	Wednesday	January	Q1
201801	Sam	165	Monday	January	Q1
201801	Sam	165	Tuesday	January	Q1
201801	Sam	165	Wednesday	January	Q1
201801	Tim	160	Thursday	January	Q1
201801	Tim	160	Friday	January	Q1
201801	Tim	160	Saturday	January	Q1
201801	Tim	160	Sunday	January	Q1
201801	Sue	150	Monday	January	Q1
201801	Sue	150	Tuesday	January	Q1
201801	Sue	150	Wednesday	January	Q1
201801	Sam	132	Thursday	January	Q1
201801	Sam	132	Friday	January	Q1
201801	Sam	132	Saturday	January	Q1
201801	Sam	132	Sunday	January	Q1
201801	Chris	125	Monday	January	Q1
201801	Chris	125	Tuesday	January	Q1
201801	Chris	125	Wednesday	January	Q1
201801	Jan	125	Monday	January	Q1
201801	Jan	125	Tuesday	January	Q1
201801	Jan	125	Wednesday	January	Q1
201801	Sue	120	Thursday	January	Q1
201801	Sue	120	Friday	January	Q1
201801	Sue	120	Saturday	January	Q1
201801	Sue	120	Sunday	January	Q1
201801	Geoff	120	Monday	January	Q1
201801	Geoff	120	Tuesday	January	Q1

201801	Geoff	120	Wednesday	January	Q1
201801	Chris	100	Thursday	January	Q1
201801	Chris	100	Friday	January	Q1
201801	Chris	100	Saturday	January	Q1
201801	Chris	100	Sunday	January	Q1
201801	Jan	100	Thursday	January	Q1
201801	Jan	100	Friday	January	Q1
201801	Jan	100	Saturday	January	Q1
201801	Jan	100	Sunday	January	Q1
201802	Jan	100	Monday	January	Q1
201802	Jan	100	Tuesday	January	Q1
201802	Jan	100	Wednesday	January	Q1
201801	Geoff	96	Thursday	January	Q1
201801	Geoff	96	Friday	January	Q1
201801	Geoff	96	Saturday	January	Q1
201801	Geoff	96	Sunday	January	Q1
201801	Allan	90	Monday	January	Q1
201801	Allan	90	Tuesday	January	Q1
201801	Allan	90	Wednesday	January	Q1
201801	Bob	85	Monday	January	Q1
201801	Bob	85	Tuesday	January	Q1
201801	Bob	85	Wednesday	January	Q1
201801	Tim	80	Thursday	February	Q1
201801	Tim	80	Friday	February	Q1
201801	Tim	80	Saturday	February	Q1
201801	Tim	80	Sunday	February	Q1
201801	Tim	80	Monday	February	Q1
201801	Tim	80	Tuesday	February	Q1
201801	Tim	80	Wednesday	February	Q1
201802	Hans	80	Monday	January	Q1
201802	Hans	80	Tuesday	January	Q1
201802	Hans	80	Wednesday	January	Q1
201802	Jan	80	Thursday	January	Q1
201802	Jan	80	Friday	January	Q1
201802	Jan	80	Saturday	January	Q1
201802	Jan	80	Sunday	January	Q1
201801	Gandolf	75	Monday	January	Q1
201801	Gandolf	75	Tuesday	January	Q1
201801	Gandolf	75	Wednesday	January	Q1
201801	Allan	72	Thursday	January	Q1

201801	Allan	72	Friday	January	Q1
201801	Allan	72	Saturday	January	Q1
201801	Allan	72	Sunday	January	Q1
201801	Bob	68	Thursday	January	Q1
201801	Bob	68	Friday	January	Q1
201801	Bob	68	Saturday	January	Q1
201801	Bob	68	Sunday	January	Q1
201801	Sam	66	Thursday	February	Q1
201801	Sam	66	Friday	February	Q1
201801	Sam	66	Saturday	February	Q1
201801	Sam	66	Sunday	February	Q1
201801	Sam	66	Monday	February	Q1
201801	Sam	66	Tuesday	February	Q1
201801	Sam	66	Wednesday	February	Q1
201801	Robert	65	Monday	January	Q1
201801	Robert	65	Tuesday	January	Q1
201801	Robert	65	Wednesday	January	Q1
201802	Hans	64	Thursday	January	Q1
201802	Hans	64	Friday	January	Q1
201802	Hans	64	Saturday	January	Q1
201802	Hans	64	Sunday	January	Q1
201801	Gandolf	60	Thursday	January	Q1
201801	Gandolf	60	Friday	January	Q1
201801	Gandolf	60	Saturday	January	Q1
201801	Gandolf	60	Sunday	January	Q1
201801	Sue	60	Thursday	February	Q1
201801	Sue	60	Friday	February	Q1
201801	Sue	60	Saturday	February	Q1
201801	Sue	60	Sunday	February	Q1
201801	Sue	60	Monday	February	Q1
201801	Sue	60	Tuesday	February	Q1
201801	Sue	60	Wednesday	February	Q1
201801	Robert	52	Thursday	January	Q1
201801	Robert	52	Friday	January	Q1
201801	Robert	52	Saturday	January	Q1
201801	Robert	52	Sunday	January	Q1
201801	Chris	50	Thursday	February	Q1
201801	Chris	50	Friday	February	Q1
201801	Chris	50	Saturday	February	Q1
201801	Chris	50	Sunday	February	Q1
201801	Chris	50	Monday	February	Q1
201801	Chris	50	Tuesday	February	Q1

201801	Chris	50	Wednesday	February	Q1
201802	Geoff	50	Monday	January	Q1
201802	Geoff	50	Tuesday	January	Q1
201802	Geoff	50	Wednesday	January	Q1
201801	Jan	50	Thursday	February	Q1
201801	Jan	50	Friday	February	Q1
201801	Jan	50	Saturday	February	Q1
201801	Jan	50	Sunday	February	Q1
201801	Jan	50	Monday	February	Q1
201801	Jan	50	Tuesday	February	Q1
201801	Jan	50	Wednesday	February	Q1
201802	Chris	50	Monday	January	Q1
201802	Chris	50	Tuesday	January	Q1
201802	Chris	50	Wednesday	January	Q1
201801	Geoff	48	Thursday	February	Q1
201801	Geoff	48	Friday	February	Q1
201801	Geoff	48	Saturday	February	Q1
201801	Geoff	48	Sunday	February	Q1
201801	Geoff	48	Monday	February	Q1
201801	Geoff	48	Tuesday	February	Q1
201801	Geoff	48	Wednesday	February	Q1
201802	Robert	45	Monday	January	Q1
201802	Robert	45	Tuesday	January	Q1
201802	Robert	45	Wednesday	January	Q1
201802	Sue	40	Monday	January	Q1
201802	Sue	40	Tuesday	January	Q1
201802	Sue	40	Wednesday	January	Q1
201802	Geoff	40	Thursday	January	Q1
201802	Geoff	40	Friday	January	Q1
201802	Geoff	40	Saturday	January	Q1
201802	Geoff	40	Sunday	January	Q1
201802	Chris	40	Thursday	January	Q1
201802	Chris	40	Friday	January	Q1
201802	Chris	40	Saturday	January	Q1
201802	Chris	40	Sunday	January	Q1
201802	Jan	40	Thursday	February	Q1
201802	Jan	40	Friday	February	Q1
201802	Jan	40	Saturday	February	Q1
201802	Jan	40	Sunday	February	Q1
201802	Jan	40	Monday	February	Q1

201802	Jan	40	Tuesday	February	Q1
201802	Jan	40	Wednesday	February	Q1
201801	Allan	36	Thursday	February	Q1
201801	Allan	36	Friday	February	Q1
201801	Allan	36	Saturday	February	Q1
201801	Allan	36	Sunday	February	Q1
201801	Allan	36	Monday	February	Q1
201801	Allan	36	Tuesday	February	Q1
201801	Allan	36	Wednesday	February	Q1
201802	Robert	36	Thursday	January	Q1
201802	Robert	36	Friday	January	Q1
201802	Robert	36	Saturday	January	Q1
201802	Robert	36	Sunday	January	Q1
201801	Fred	35	Monday	January	Q1
201801	Fred	35	Tuesday	January	Q1
201801	Fred	35	Wednesday	January	Q1
201801	Cat	35	Monday	January	Q1
201801	Cat	35	Tuesday	January	Q1
201801	Cat	35	Wednesday	January	Q1
201802	Fred	35	Monday	January	Q1
201802	Fred	35	Tuesday	January	Q1
201802	Fred	35	Wednesday	January	Q1
201802	Sam	35	Monday	January	Q1
201802	Sam	35	Tuesday	January	Q1
201802	Sam	35	Wednesday	January	Q1
201801	Bob	34	Thursday	February	Q1
201801	Bob	34	Friday	February	Q1
201801	Bob	34	Saturday	February	Q1
201801	Bob	34	Sunday	February	Q1
201801	Bob	34	Monday	February	Q1
201801	Bob	34	Tuesday	February	Q1
201801	Bob	34	Wednesday	February	Q1
201802	Sue	32	Thursday	January	Q1
201802	Sue	32	Friday	January	Q1
201802	Sue	32	Saturday	January	Q1
201802	Sue	32	Sunday	January	Q1
201802	Hans	32	Thursday	February	Q1
201802	Hans	32	Friday	February	Q1
201802	Hans	32	Saturday	February	Q1
201802	Hans	32	Sunday	February	Q1

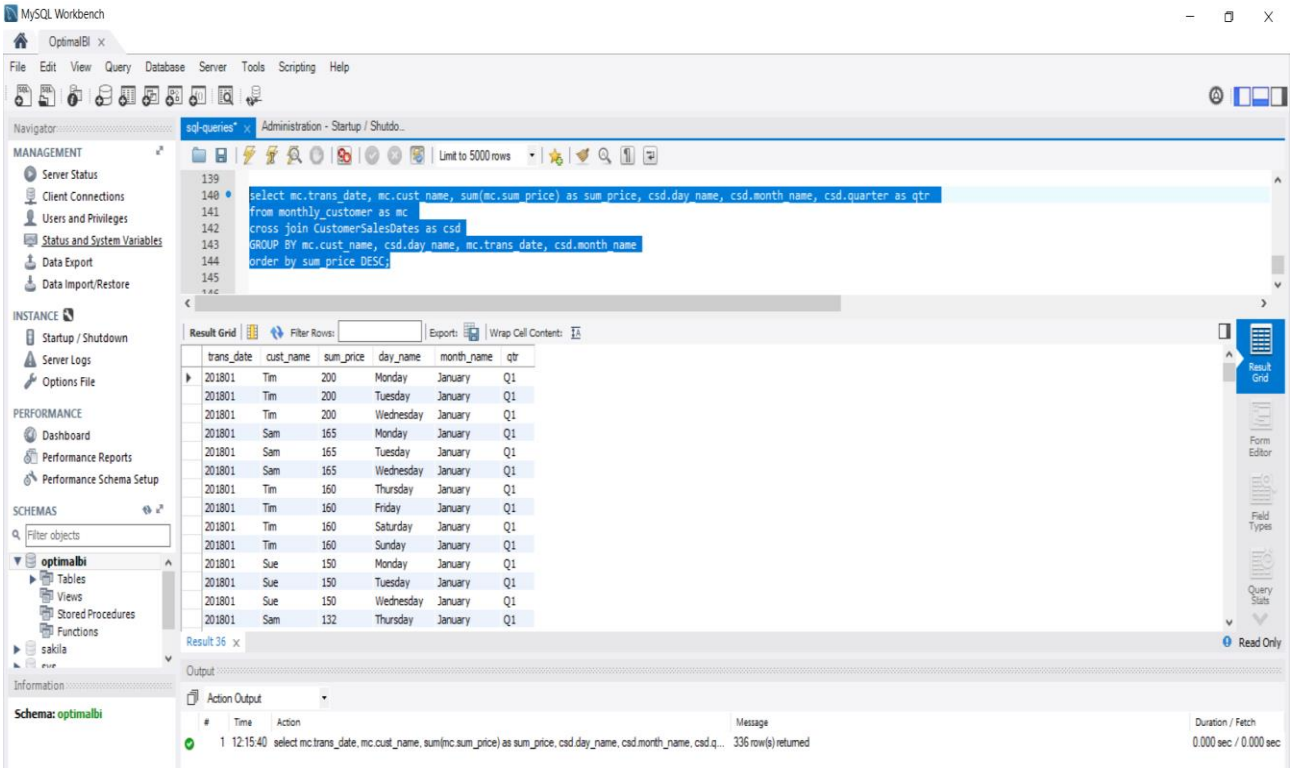
201802	Hans	32	Monday	February	Q1
201802	Hans	32	Tuesday	February	Q1
201802	Hans	32	Wednesday	February	Q1
201801	Gandolf	30	Thursday	February	Q1
201801	Gandolf	30	Friday	February	Q1
201801	Gandolf	30	Saturday	February	Q1
201801	Gandolf	30	Sunday	February	Q1
201801	Gandolf	30	Monday	February	Q1
201801	Gandolf	30	Tuesday	February	Q1
201801	Gandolf	30	Wednesday	February	Q1
201802	Cat	30	Monday	January	Q1
201802	Cat	30	Tuesday	January	Q1
201802	Cat	30	Wednesday	January	Q1
201801	Fred	28	Thursday	January	Q1
201801	Fred	28	Friday	January	Q1
201801	Fred	28	Saturday	January	Q1
201801	Fred	28	Sunday	January	Q1
201801	Cat	28	Thursday	January	Q1
201801	Cat	28	Friday	January	Q1
201801	Cat	28	Saturday	January	Q1
201801	Cat	28	Sunday	January	Q1
201802	Fred	28	Thursday	January	Q1
201802	Fred	28	Friday	January	Q1
201802	Fred	28	Saturday	January	Q1
201802	Fred	28	Sunday	January	Q1
201802	Sam	28	Thursday	January	Q1
201802	Sam	28	Friday	January	Q1
201802	Sam	28	Saturday	January	Q1
201802	Sam	28	Sunday	January	Q1
201801	Robert	26	Thursday	February	Q1
201801	Robert	26	Friday	February	Q1
201801	Robert	26	Saturday	February	Q1
201801	Robert	26	Sunday	February	Q1
201801	Robert	26	Monday	February	Q1
201801	Robert	26	Tuesday	February	Q1
201801	Robert	26	Wednesday	February	Q1
201802	Bob	25	Monday	January	Q1
201802	Bob	25	Tuesday	January	Q1
201802	Bob	25	Wednesday	January	Q1
201802	Gandolf	25	Monday	January	Q1
201802	Gandolf	25	Tuesday	January	Q1

201802	Gandolf	25	Wednesday	January	Q1
201802	Cat	24	Thursday	January	Q1
201802	Cat	24	Friday	January	Q1
201802	Cat	24	Saturday	January	Q1
201802	Cat	24	Sunday	January	Q1
201802	Geoff	20	Thursday	February	Q1
201802	Geoff	20	Friday	February	Q1
201802	Geoff	20	Saturday	February	Q1
201802	Geoff	20	Sunday	February	Q1
201802	Geoff	20	Monday	February	Q1
201802	Geoff	20	Tuesday	February	Q1
201802	Geoff	20	Wednesday	February	Q1
201802	Bob	20	Thursday	January	Q1
201802	Bob	20	Friday	January	Q1
201802	Bob	20	Saturday	January	Q1
201802	Bob	20	Sunday	January	Q1
201802	Chris	20	Thursday	February	Q1
201802	Chris	20	Friday	February	Q1
201802	Chris	20	Saturday	February	Q1
201802	Chris	20	Sunday	February	Q1
201802	Chris	20	Monday	February	Q1
201802	Chris	20	Tuesday	February	Q1
201802	Chris	20	Wednesday	February	Q1
201802	Gandolf	20	Thursday	January	Q1
201802	Gandolf	20	Friday	January	Q1
201802	Gandolf	20	Saturday	January	Q1
201802	Gandolf	20	Sunday	January	Q1
201802	Robert	18	Thursday	February	Q1
201802	Robert	18	Friday	February	Q1
201802	Robert	18	Saturday	February	Q1
201802	Robert	18	Sunday	February	Q1
201802	Robert	18	Monday	February	Q1
201802	Robert	18	Tuesday	February	Q1
201802	Robert	18	Wednesday	February	Q1
201802	Sue	16	Thursday	February	Q1
201802	Sue	16	Friday	February	Q1
201802	Sue	16	Saturday	February	Q1
201802	Sue	16	Sunday	February	Q1
201802	Sue	16	Monday	February	Q1
201802	Sue	16	Tuesday	February	Q1
201802	Sue	16	Wednesday	February	Q1
201801	Fred	14	Thursday	February	Q1

201801	Fred	14	Friday	February	Q1
201801	Fred	14	Saturday	February	Q1
201801	Fred	14	Sunday	February	Q1
201801	Fred	14	Monday	February	Q1
201801	Fred	14	Tuesday	February	Q1
201801	Fred	14	Wednesday	February	Q1
201801	Cat	14	Thursday	February	Q1
201801	Cat	14	Friday	February	Q1
201801	Cat	14	Saturday	February	Q1
201801	Cat	14	Sunday	February	Q1
201801	Cat	14	Monday	February	Q1
201801	Cat	14	Tuesday	February	Q1
201801	Cat	14	Wednesday	February	Q1
201802	Fred	14	Thursday	February	Q1
201802	Fred	14	Friday	February	Q1
201802	Fred	14	Saturday	February	Q1
201802	Fred	14	Sunday	February	Q1
201802	Fred	14	Monday	February	Q1
201802	Fred	14	Tuesday	February	Q1
201802	Fred	14	Wednesday	February	Q1
201802	Sam	14	Thursday	February	Q1
201802	Sam	14	Friday	February	Q1
201802	Sam	14	Saturday	February	Q1
201802	Sam	14	Sunday	February	Q1
201802	Sam	14	Monday	February	Q1
201802	Sam	14	Tuesday	February	Q1
201802	Sam	14	Wednesday	February	Q1
201802	Cat	12	Thursday	February	Q1
201802	Cat	12	Friday	February	Q1
201802	Cat	12	Saturday	February	Q1
201802	Cat	12	Sunday	February	Q1
201802	Cat	12	Monday	February	Q1
201802	Cat	12	Tuesday	February	Q1
201802	Cat	12	Wednesday	February	Q1
201802	Allan	10	Monday	January	Q1
201802	Allan	10	Tuesday	January	Q1
201802	Allan	10	Wednesday	January	Q1
201802	Bob	10	Thursday	February	Q1
201802	Bob	10	Friday	February	Q1
201802	Bob	10	Saturday	February	Q1
201802	Bob	10	Sunday	February	Q1

201802	Bob	10	Monday	February	Q1
201802	Bob	10	Tuesday	February	Q1
201802	Bob	10	Wednesday	February	Q1
201802	Gandolf	10	Thursday	February	Q1
201802	Gandolf	10	Friday	February	Q1
201802	Gandolf	10	Saturday	February	Q1
201802	Gandolf	10	Sunday	February	Q1
201802	Gandolf	10	Monday	February	Q1
201802	Gandolf	10	Tuesday	February	Q1
201802	Gandolf	10	Wednesday	February	Q1
201802	Allan	8	Thursday	January	Q1
201802	Allan	8	Friday	January	Q1
201802	Allan	8	Saturday	January	Q1
201802	Allan	8	Sunday	January	Q1
201802	Allan	4	Thursday	February	Q1
201802	Allan	4	Friday	February	Q1
201802	Allan	4	Saturday	February	Q1
201802	Allan	4	Sunday	February	Q1
201802	Allan	4	Monday	February	Q1
201802	Allan	4	Tuesday	February	Q1
201802	Allan	4	Wednesday	February	Q1

Snapshot of my workbench:



Situations where this approach might be applied: In my opinion, this kind of approach is followed to generate reports that provide a projection of the data or may be identify purchasing/sales trend or provide estimates based on the available data.