

Humane DB design and programming

A talk about how to program your DB to make it fast, flexible and right.

With examples from SAP development and the SAP community.

~~A talk about how to program your DB to make it fast, flexible and right.~~

~~With examples from SAP development and the SAP community.~~

Some thoughts about writing, asking questions, and getting answers.



@LBREDDEMANN



Data Process Insights

Lars Breddemann

[@LBREDDemann](#)

<https://lbreddemann.org/>

<https://dataprocessinsights.com.au/>

lars@dataprocessinsights.com.au

UNFINISHED WORK

FRIDAY EVENING



PERFECT!
I'LL FINISH
THIS ON
MONDAY



MONDAY MORNING...



WHAT DOES
THIS MEAN!?!?

```

create FUNCTION "F_KEEP_DATE"( TDATE DATE, TTODAY DATE)
RETURNS RETVALUE CHAR
LANGUAGE SQLSCRIPT
SQL SECURITY INVOKER AS
/*****
NAME: F_KEEP_DATE
PURPOSE: RETURNS T/F WHETHER OR NOT THE DATE MEETS THE CRITERIA TO MAINTAIN DATA.
REVISIONS:
VER DATE AUTHOR DESCRIPTION
-----
1.0 6/18/2008 1. CREATED THIS FUNCTION.
*****/
BEGIN
RETVALUE := 'F';
IF :TDATE BETWEEN ADD_DAYS (TO_DATE (:TTODAY, 'YYYY-MM-DD'), -30) AND :TTODAY
OR TO_CHAR( EXTRACT (MONTH FROM ADD_DAYS (TO_DATE (:TDATE, 'YYYY-MM-DD'), 1)))<
TO_CHAR( EXTRACT (MONTH FROM TO_DATE (TDATE, 'YYYY-MM-DD'))
THEN
RETVALUE := 'T';
END IF;
--RETURN :RETVALUE;
END ; |

```

```

1  create FUNCTION "F_KEEP_DATE"
2      (TDATE DATE, TTODAY DATE)
3  RETURNS RETVALUE CHAR
4  LANGUAGE SQLSCRIPT
5  SQL SECURITY INVOKER AS
6  BEGIN
7      RETVALUE := 'F';
8
9      IF :TDATE BETWEEN ADD_DAYS (TO_DATE (:TTODAY, 'YYYY-MM-DD')
10                                     , -30)
11          AND :TTODAY
12          OR TO_CHAR(EXTRACT(MONTH FROM ADD_DAYS
13                          (TO_DATE (:TDATE, 'YYYY-MM-DD'), 1)))
14              <> TO_CHAR(EXTRACT(MONTH FROM
15                          TO_DATE (:TDATE, 'YYYY-MM-DD')))
16      THEN
17          RETVALUE := 'T';
18      END IF;
19  END;

```

```
1  create FUNCTION "F_KEEP_DATE"
2      (TDATE DATE, TTODAY DATE)
3  RETURNS RETVALUE CHAR
4  LANGUAGE SQLSCRIPT
5  SQL SECURITY INVOKER AS
6  BEGIN
7      RETVALUE := 'F';
8
9      IF :TDATE BETWEEN ADD_DAYS (TO_DATE (:TTODAY, 'YYYY-MM-DD')
10                                     , -30)
11          AND :TTODAY
12      OR TO_CHAR(EXTRACT(MONTH FROM ADD_DAYS
13                      (TO_DATE (:TDATE, 'YYYY-MM-DD'), 1)))
14          <> TO_CHAR(EXTRACT(MONTH FROM
15                          TO_DATE (:TDATE, 'YYYY-MM-DD'))))
16      THEN
17          RETVALUE := 'T';
18      END IF;
19  END;
```

```
1  create FUNCTION "F_KEEP_DATE"
2      (TDATE DATE, TTODAY DATE)
3  RETURNS RETVALUE CHAR
4  LANGUAGE SQLSCRIPT
5  SQL SECURITY INVOKER AS
6  BEGIN
7      RETVALUE := 'F';
8
9      IF :TDATE BETWEEN ADD_DAYS (:TTODAY, -30) AND :TTODAY
10         OR
11         EXTRACT (MONTH FROM ADD_DAYS (:TDATE, 1))
12         <> EXTRACT (MONTH FROM :TDATE)
13     THEN
14         RETVALUE := 'T';
15     END IF;
16 END;
```

```
1  create FUNCTION "F_KEEP_DATE"
2      (TDATE DATE, TTODAY DATE)
3  RETURNS RETVALUE CHAR
4  LANGUAGE SQLSCRIPT
5  SQL SECURITY INVOKER AS
6  BEGIN
7      RETVALUE := 'F';
8
9      IF :TDATE BETWEEN ADD_DAYS (:TTODAY, -30) AND :TTODAY
10         OR
11         EXTRACT (MONTH FROM ADD_DAYS (:TDATE, 1))
12         <> EXTRACT (MONTH FROM :TDATE)
13     THEN
14         RETVALUE := 'T';
15     END IF;
16 END;
```

```
1  create FUNCTION "F_KEEP_DATE"  
2      (TDATE DATE, TTODAY DATE)  
3  RETURNS RETVALUE CHAR  
4  LANGUAGE SQLSCRIPT  
5  SQL SECURITY INVOKER AS  
6  BEGIN  
7      RETVALUE := 'F';  
8  
9      IF :TDATE BETWEEN ADD_DAYS (:TTODAY, -30) AND :TTODAY  
10         OR :TDATE = LAST_DAY(:TDATE)  
11         THEN  
12             RETVALUE := 'T';  
13         END IF;  
14 END;
```

```
1  create FUNCTION "F_KEEP_DATE"  
2      (TDATE DATE, TTODAY DATE)  
3  RETURNS RETVALUE CHAR  
4  LANGUAGE SQLSCRIPT  
5  SQL SECURITY INVOKER AS  
6  BEGIN  
7      RETVALUE := 'F';  
8  
9      IF :TDATE BETWEEN ADD_DAYS (:TTODAY -30) AND :TTODAY  
10         OR :TDATE = LAST_DAY(:TDATE)  
11      THEN  
12         RETVALUE := 'T';  
13      END IF;  
14  END;
```

```
1  create FUNCTION "dataInEditPeriod"(checked_day DATE)
2  RETURNS result INTEGER
3  LANGUAGE SQLSCRIPT
4  SQL SECURITY INVOKER AS
5  BEGIN
6      result := 0;
7      IF      (:checked_day BETWEEN ADD_DAYS (current_date, -30)
8              AND current_date)
9              OR (:checked_day = LAST_DAY(:checked_day))
10     THEN
11         result := 1;
12     END IF;
13 END;
```

```
1  create FUNCTION "dataInEditPeriod"(checked_day DATE)
2  RETURNS result INTEGER
3  LANGUAGE SQLSCRIPT
4  SQL SECURITY INVOKER AS
5  BEGIN
6      result := 0;
7      IF      (:checked_day BETWEEN ADD_DAYS (current_date, -30)
8              AND current_date)
9              OR (:checked_day = LAST_DAY(:checked_day))
10     THEN
11         result := 1;
12     END IF;
13 END;
```

```
1  create FUNCTION "dataInEditPeriod"(checked_day DATE)
2  RETURNS result INTEGER
3  LANGUAGE SQLSCRIPT
4  SQL SECURITY INVOKER AS
5  BEGIN
6
7      IF      (:checked_day BETWEEN ADD_DAYS (current_date, -30)
8              AND current_date)
9      OR      (:checked_day = LAST_DAY(:checked_day))
10 THEN
11     result := 1;
12 ELSE
13     RESULT := 0;
14 END IF;
15 END;
```

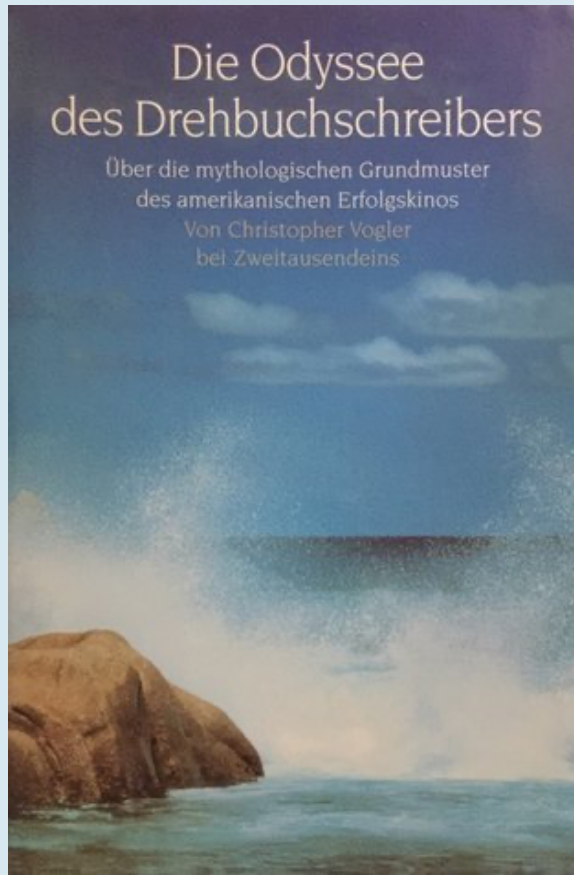
```
1  create FUNCTION "dataInEditPeriod"(checked_day DATE)
2  RETURNS result INTEGER
3  LANGUAGE SQLSCRIPT
4  SQL SECURITY INVOKER AS
5  BEGIN
6
7      IF      (:checked_day BETWEEN ADD_DAYS (current_date, -30)
8              AND current_date)
9      OR      (:checked_day = LAST_DAY(:checked_day))
10     THEN
11         result := 1;
12     ELSE
13         RESULT := 0;
14     END IF;
15 END;

SUM ("FeeAmount" * "dataInEditPeriod"("FeeIssueDate"))
      AS "FeesOpenToChange"
```

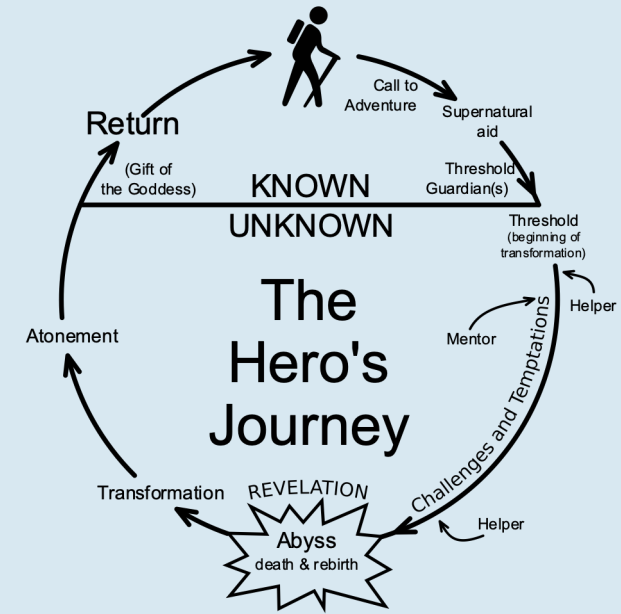
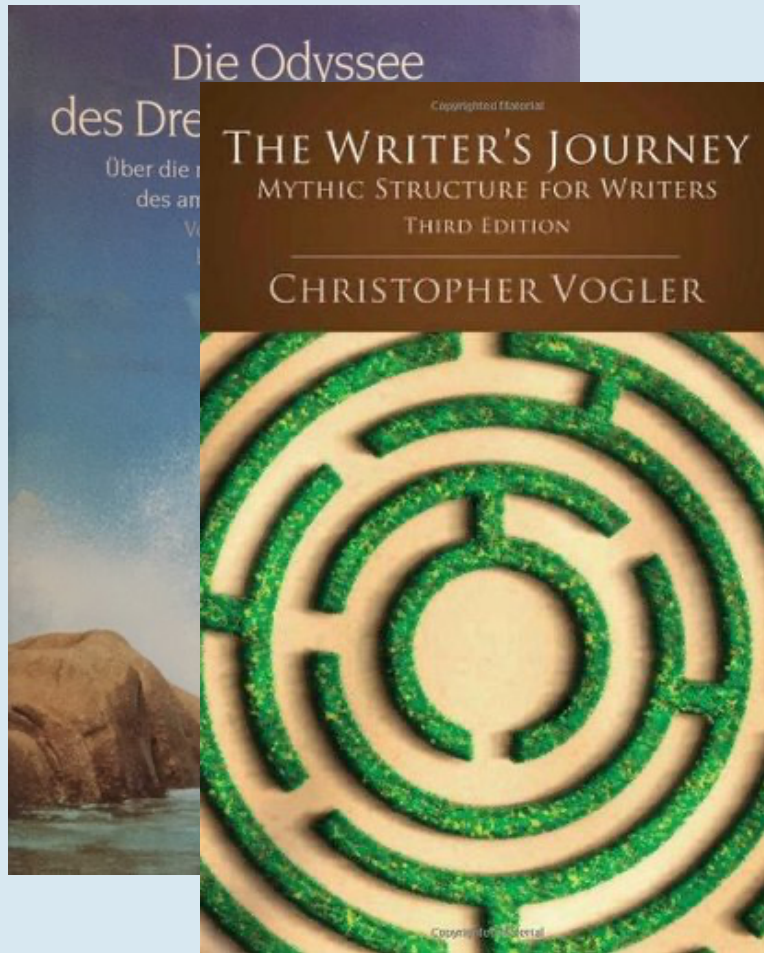
```
1  create FUNCTION "dataInEditPeriod"(checked_day DATE)
2  RETURNS result INTEGER
3  LANGUAGE SQLSCRIPT
4  SQL SECURITY INVOKER AS
5  BEGIN
6
7      IF      (:checked_day BETWEEN ADD_DAYS (current_date, -30)
8              AND current_date)
9      OR      (:checked_day = LAST_DAY(:checked_day))
10     THEN
11         result := 1;
12     ELSE
13         RESULT := 0;
14     END IF;
15 END;
```

```
1 CREATE OR REPLACE FUNCTION "dataInEditPeriod"(checked_day DATE)
2 RETURNS result INTEGER
3 LANGUAGE SQLSCRIPT
4 SQL SECURITY INVOKER
5 AS
6 BEGIN
7
8     result := CASE
9         -- editing is allowed when checked_day is within the past 30 days
10        -- or if checked_day is the last day of it's month (?)
11            WHEN ( (:checked_day BETWEEN ADD_DAYS (current_date, -30)
12                  AND current_date)
13                OR (:checked_day = LAST_DAY(:checked_day)) )
14            THEN 1
15            ELSE 0
16        END;
17
18 END;
19 |
```

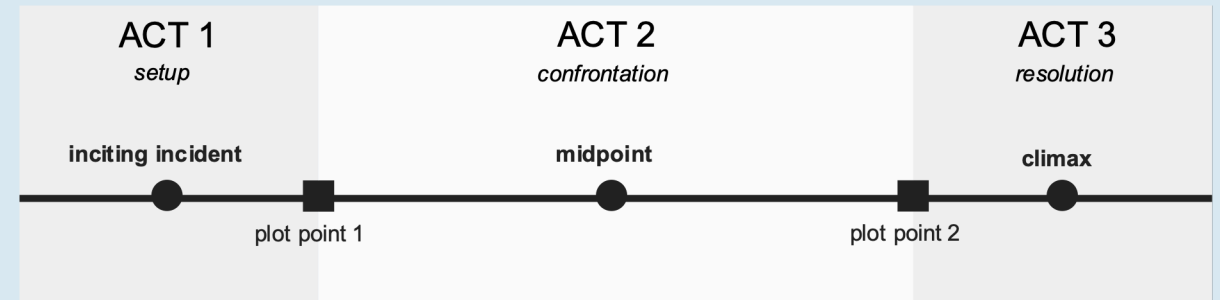
How did we get here?



How did we get here?



Joseph Campbell



"A whole is what has a beginning and middle and end"
Aristotle, Poetics (335 BCE)

Act I – "The setup"

- You need to use an SQL DB

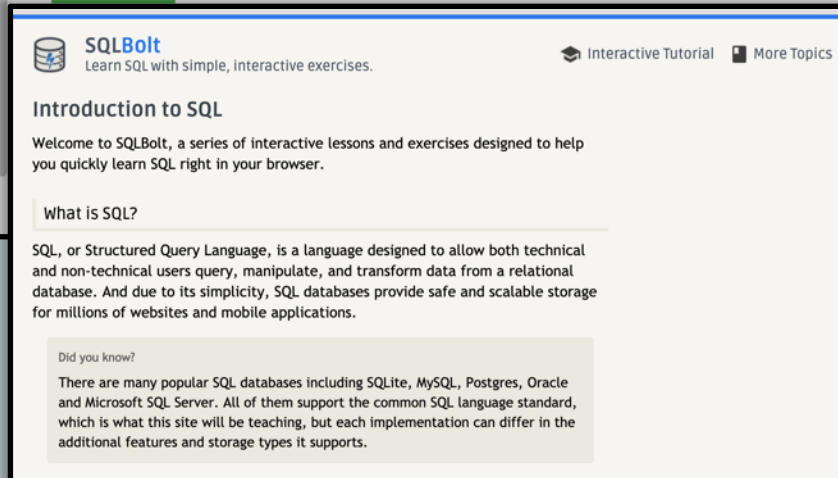
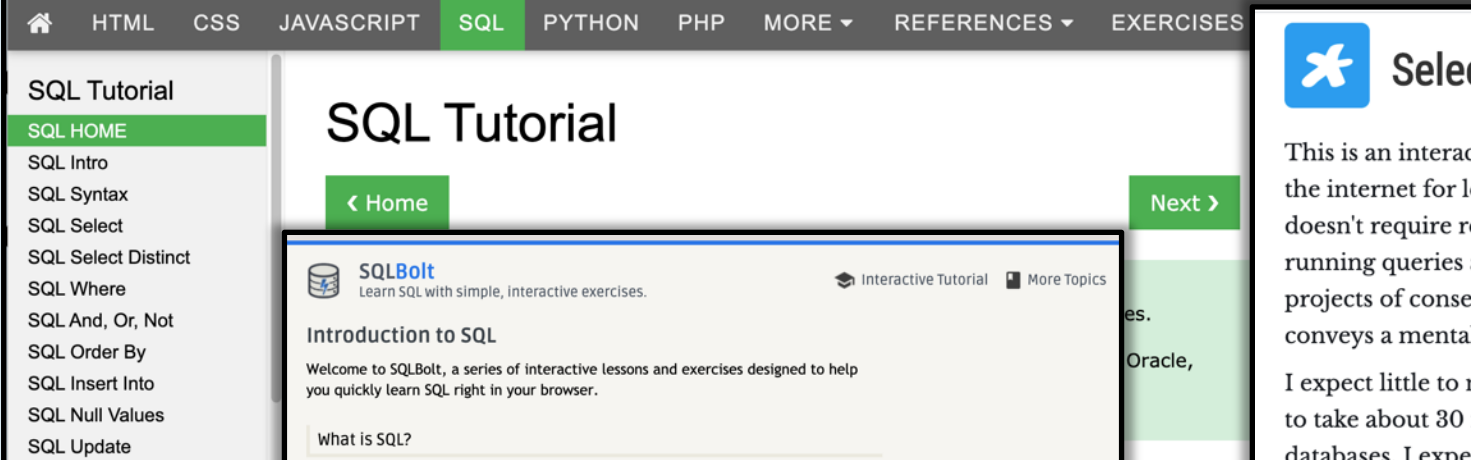
Learning English

Syntax, vocabulary
"Hello, my name is!"

Learning SQL

Syntax, vocabulary
`SELECT "Hello, world!" FROM ...`

You need to use an SQL DB



Select Star SQL

This is an interactive book which aims to be the best place on the internet for learning SQL. It is free of charge, free of ads and doesn't require registration or downloads. It helps you learn by running queries against a real-world dataset to complete projects of consequence. It is not a mere reference page — it conveys a mental model for writing SQL.

I expect little to no coding knowledge. Each chapter is designed to take about 30 minutes. As more of the world's data is stored in databases, I expect that this time will pay rich dividends!

Front Matter

The philosophical and technical underpinnings of this project.

Beazley's Last Statement

Chapter 1 examines Napoleon Beazley's argument against capital punishment. We learn the structure of a SQL query and row operations on a table.

You need to use an SQL DB

w3schools.com
THE WORLD'S LARGEST WEB DEVELOPER SITE

HTML CSS JAVASCRIPT **SQL** PYTHON PHP MORE REFERENCES EXERCISES

SQL Tutorial

SQL HOME

SQL Intro

SQL Syntax

SQL Select

SQL Select Distinct

SQL Where

SQL And, Or, Not

SQL Order By

SQL Insert Into

SQL Null Values

SQL Update

SQL Tutorial

Home

Next

SQLBolt
Learn SQL with simple, interactive exercises.

Interactive Tutorial More Topics

Introduction to SQL

Welcome to SQLBolt, a series of interactive lessons and exercises designed to help you quickly learn SQL right in your browser.

What is SQL?

SQL, or Structured Query Language, is a language designed to allow both technical and non-technical users query, manipulate, and transform data from a relational database. And due to its simplicity, SQL databases provide safe and scalable storage for millions of websites and mobile applications.

Did you know?

There are many popular SQL databases including SQLite, MySQL, Postgres, Oracle and Microsoft SQL Server. All of them support the common SQL language standard, which is what this site will be teaching, but each implementation can differ in the additional features and storage types it supports.

Select Star SQL

This is an interactive book which aims to be the internet for learning SQL. It is free of charge and doesn't require registration or downloads. It's about running queries against a real-world database and projects of consequence. It is not a mere reference; it conveys a mental model for writing SQL.

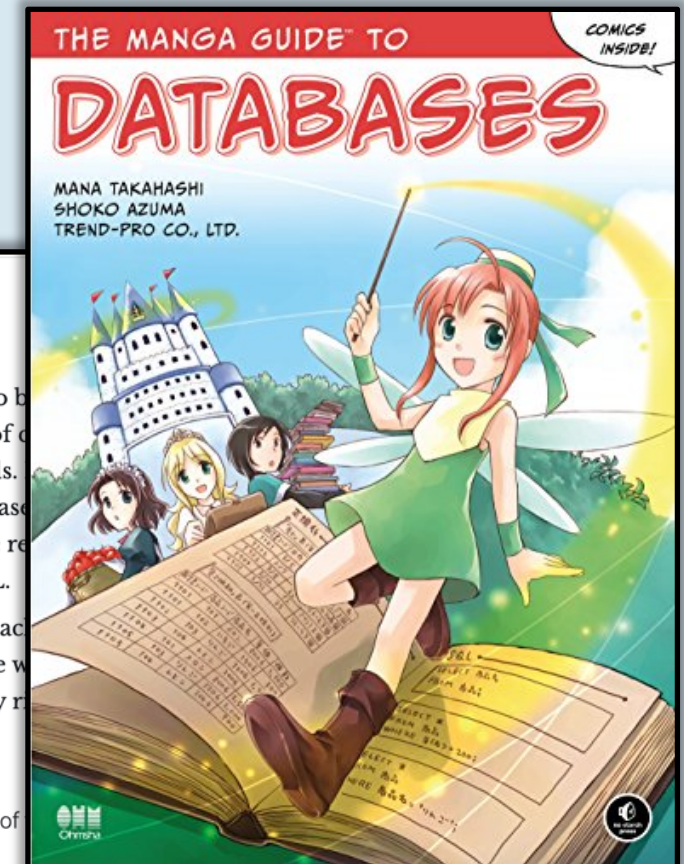
I expect little to no coding knowledge. Each chapter takes about 30 minutes. As more of the world uses databases, I expect that this time will pay for itself.

Front Matter

The philosophical and technical underpinnings of SQL.

Beazley's Last Statement

Chapter 1 examines Napoleon Beazley's argument against capital punishment. We learn the structure of a SQL query and row operations on a table.



Act I – "The setup"

- You need to use an SQL DB
- Getting stuff done™
- Looking up the tricky bits

Learning English

Short reading/writing, figures of
speech
"Bob's your uncle."

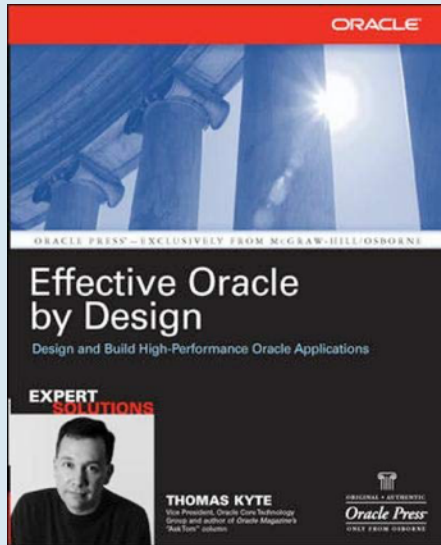
Syntax, vocabulary
"Hello, my name is!"

Learning SQL

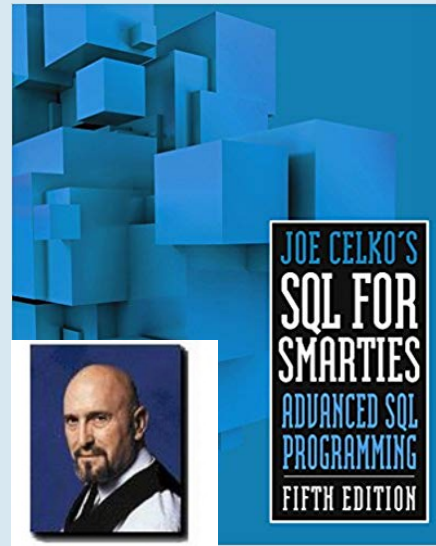
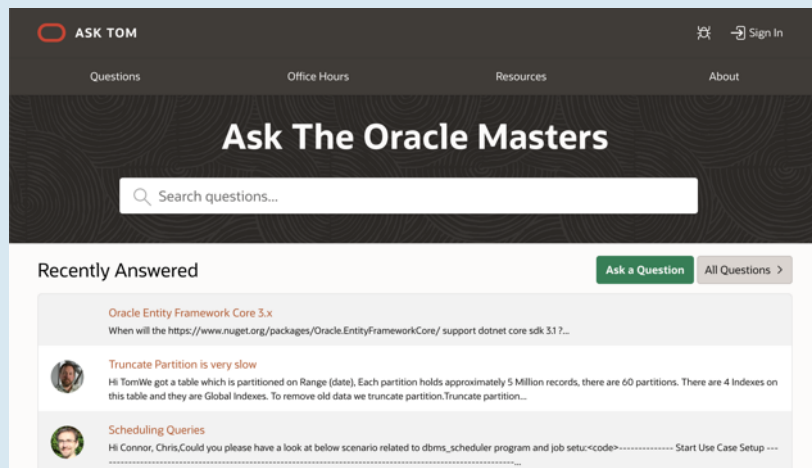
Recipes/templates
`WHERE 1=1, SUM(money) OVER()`

Syntax, vocabulary
`SELECT "Hello, world!" FROM ...`

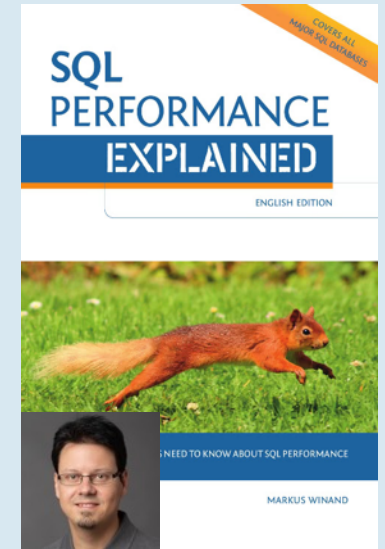
Getting stuff done™



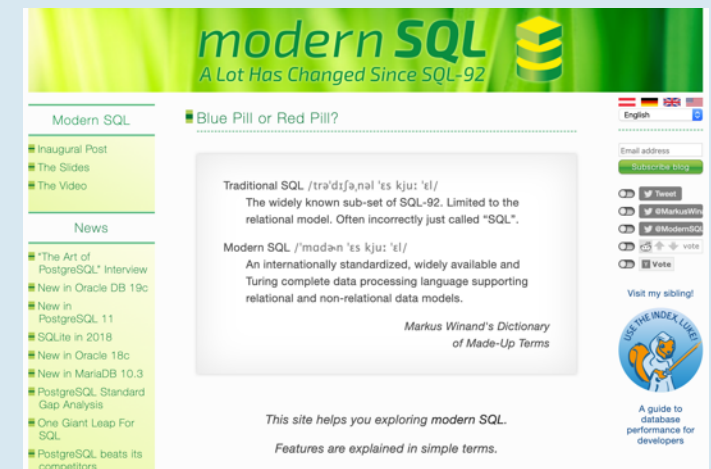
Tom Kyte



Joe Celko



Markus Winand



Getting stuff done™

Haki Benita

AboutSubscribe

21 NOVEMBER 2019 / 9 MINUTES / POSTGRESQL, SQL, PERFORMANCE

12 Common Mistakes and Missed Optimization Opportunities in SQL

Made by Developers and Non-Developers

Most programming languages are designed for professional developers with knowledge of algorithms and data structure. **SQL is different.**

SQL is used by analysts, data scientists, product managers, designers and many others. These professionals have access to databases, but they don't always have the intuition and understanding to write efficient queries.

In an effort to make my team write better SQL, I went over reports written by non-developers and code reviews, and gathered **common mistakes and missed optimization opportunities in SQL.**



Avoid painful mistakes...

► Table of Contents

Be Careful When Dividing Integers

In PostgreSQL, **dividing an integer by an integer results in an integer:**

❌ DONT

```
db=# (  
SELECT tax / price AS tax_ratio  
FROM sale  
)  
  
tax_ratio  
-----  
0
```

stackoverflow

Products

[sql]

6,486 2 9 25

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

Free 30 Day Trial

Questions tagged [sql]

Ask Question

Structured Query Language (SQL) is a language for querying databases. Questions should include code examples, table structure, sample data, and a tag for the DBMS implementation (e.g. MySQL, PostgreSQL, Oracle, MS SQL Server, IBM DB2, etc.) being used. If your question relates solely to a specific DBMS (uses specific extensions/features), use that DBMS's tag instead. Answers to questions tagged with SQL should use ISO/IEC standard SQL.

Unwatch TagIgnore Tag

Learn more...Improve tag infoTop usersSynonyms (6)sql jobs

527,432 questions

NewestActiveBountiedUnansweredMoreFilter

0 votesPossible SQL injection method
I am trying to SQL inject my code with WHERE email = 'aben101@icloud.com'
phphtmlsqlsql-injection
13 views

0 votesSQL: How to design the database
I'm trying to set up a database for a company. I have a table of Genres (id, name, ...)
Table? Genres (id, name, ...)
How do I design the database?
sqlsql-injection
13 views

Blog

Hello World: Curing imposter syndrome by embracing the suck

Podcast: Make my Monolith a Micro

Featured on Meta

TLS 1.0 and TLS 1.1 removal for Stack Exchange services

Did Stack Exchange cut the number of negative comments nearly in half

SAP

Community

Topics

Answers

Blogs

Events

Programs

Resources

What's New

[Search]

[Profile]

Home

Community

Questions

Ask a Question

Write a Blog Post

...

SAP HANA

23754 Questions | 14327 Contributors | 955 Followers

Find questions, answers and discussions

Ask a Question

RSS

23754 results

Sort by: UpdatedCreatedHottestVotes

Janardhana Reddy asked

PO value shows EUR but in stock standard value shows INR kindly advise

Jerry Janda commented | 1 hour ago

SAP ERP | SAP HANA

0 Answers0 Votes

Reinis Verbelis Reinis Verbelis asked

Unable to stop/start HANA

Reinis Verbelis Reinis Verbelis commented | 5 hours ago

SAP HANA

2 Answers0 Votes

QUESTION TYPES

All Questions

All Answered Questions

With Accepted Answer

Without Accepted Answer

No Answers

No Responses

Getting stuff done™

- Development guidelines
- Naming conventions
 - What do you learn about your data model when there is a CV_ at the start of the name?
- Only allowed to use X / have to use Y
- Copy & Modify approach

Top 10 reasons to choose SAP HANA

SAP HANA layered architecture (Great to have) and naming conventions

As per my view in Native HANA great to have Three (3) Layers like SAP HANA Live models

1. Base View (1 layer – Raw data 1:1 source to HANA Base layer, Global Filters)
2. Reuse View (Max 3- 4 layers view good to have – Logics, BTL, Mappings, complex)
3. Query View/Report View (1 layer – Final View – Thin layer)

P.S – Master data will be maintained as individual with Dimensions with Star model and consumed in relevant transactional HANA Models

Calculation View: CV_<Layer>_<BUSINESS_NAME> e.g. CV_BV_FSR_1

Naming Conventions – Keep the name as short as possible (preferably under 15 chars)

Name every element in CAPITAL LETTERS. Give meaningful business names

- CV_BV_XYZ
- CV_RV_XYZ1, 2, 3 (Based on the complexity layers will increase – max 5 nice to have.
Possible chunk to smaller

<https://blogs.sap.com/2019/06/05/sap-native-hana-best-practices-and-guidelines-for-significant-performance/>

Act I – "The setup"

- You need to use an SQL DB
- Getting stuff done™
- Looking up the tricky bits
- Query code is right when there's no error

Learning English

Short reading/writing, figures of speech
"Bob's your uncle."

Syntax, vocabulary
"Hello, my name is!"

Learning SQL

Recipes/templates
`WHERE 1=1, SUM(money) OVER()`

Syntax, vocabulary
`SELECT "Hello, world!" FROM ...`

Act I – "The setup"

- You need to use an SQL DB
- Getting stuff done™
- Looking up the tricky bits
- Query code is right when there's no error
- Why is the query slow?
 - Explain plan, tracing, relational algebra!?

Learning English

Independent reading and writing –
grasp of concept, form, ideas
Jule Verne, Mark Twain

Short reading/writing, figures of
speech
"Bob's your uncle."

Syntax, vocabulary
"Hello, my name is!"

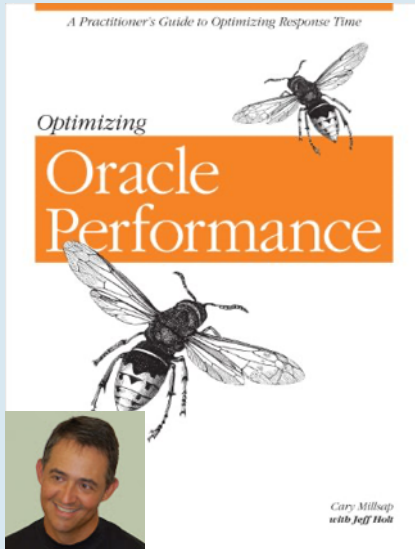
Learning SQL

Deeper grasp of concepts,
awareness of complex functions
and problems
`GROUP BY GROUPING,
HIERARCHIES`

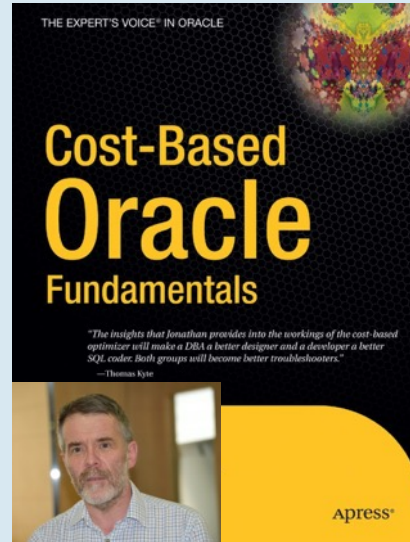
Recipes/templates
`WHERE 1=1, SUM(money) OVER()`

Syntax, vocabulary
`SELECT "Hello, world!" FROM ...`

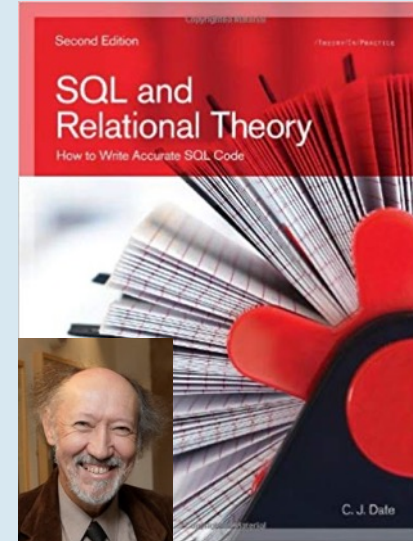
Getting stuff done – performance Pro™



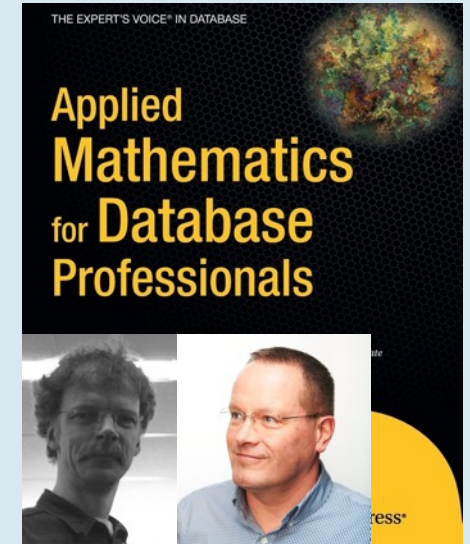
Cary Millsap



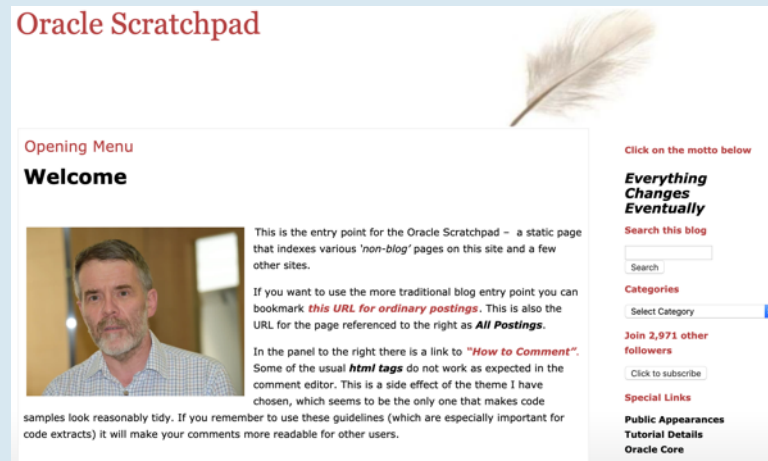
Jonathan Lewis



C. J. Date



Lex de Haan
Toon Koppelaars



Act II – "The confrontation"

- All those recipes and templates are not flexible/fast enough
- It's still complicated and huge SQL statements are hard to understand
- "Technical" tuning only gets us so far
- How much better does "*Huckleberry Finn*" get when you run a spellcheck over it?

Learning English

Independent reading and writing –
grasp of concept, form, ideas
Jule Verne, Mark Twain

Short reading/writing, figures of
speech
"Bob's your uncle."

Syntax, vocabulary
"Hello, my name is!"

Learning SQL

Deeper grasp of concepts,
awareness of complex functions
and problems
GROUP BY GROUPING,
HIERARCHIES

Recipes/templates
WHERE 1=1, SUM(money) OVER()

Syntax, vocabulary
SELECT "Hello, world!" FROM ...

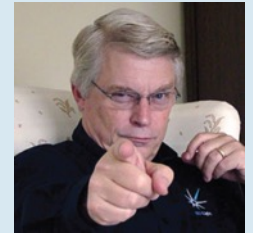
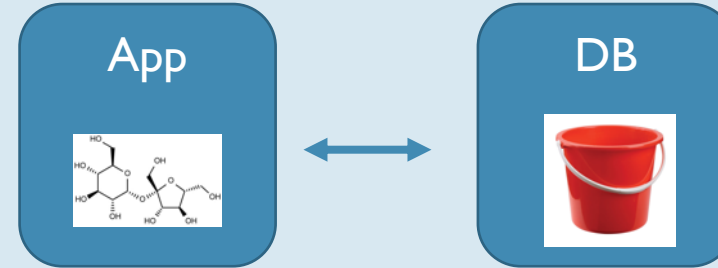
Act II – "The confrontation"

- Operating the DB is hard – this is DBA country
- Changing the app requires changing the DB.
 - (it's called developing an app)
- What about noSQL and schema-less DBs?
 - Schema-less → schema-on-read
 - Write what you want, read what you can.



Schema-free, problem-free?

Where do you keep the knowledge about your data?



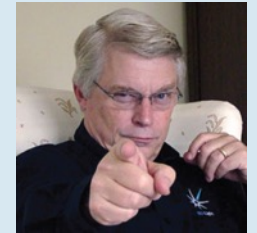
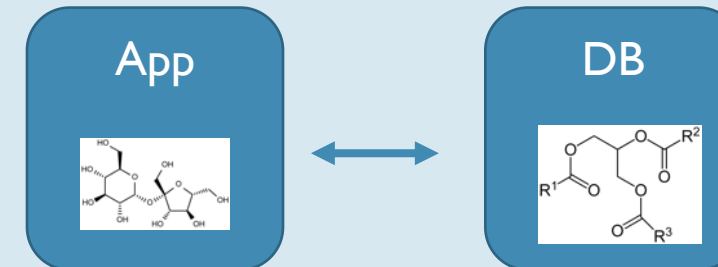
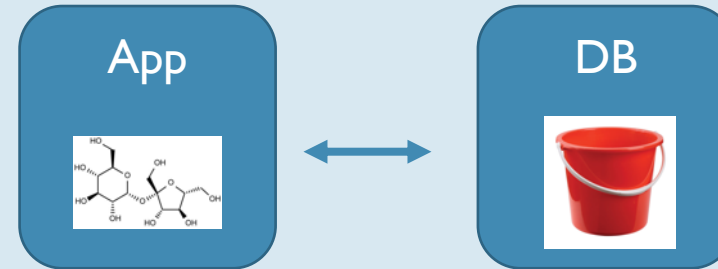
Robert 'Uncle Bob' Martin

Schema-free, problem-free?

Where do you keep the knowledge about your data?

Schema-on-read: only the app(s) need to know the data.

Schema-on-write: app(s) need to know how to map to DB structures.



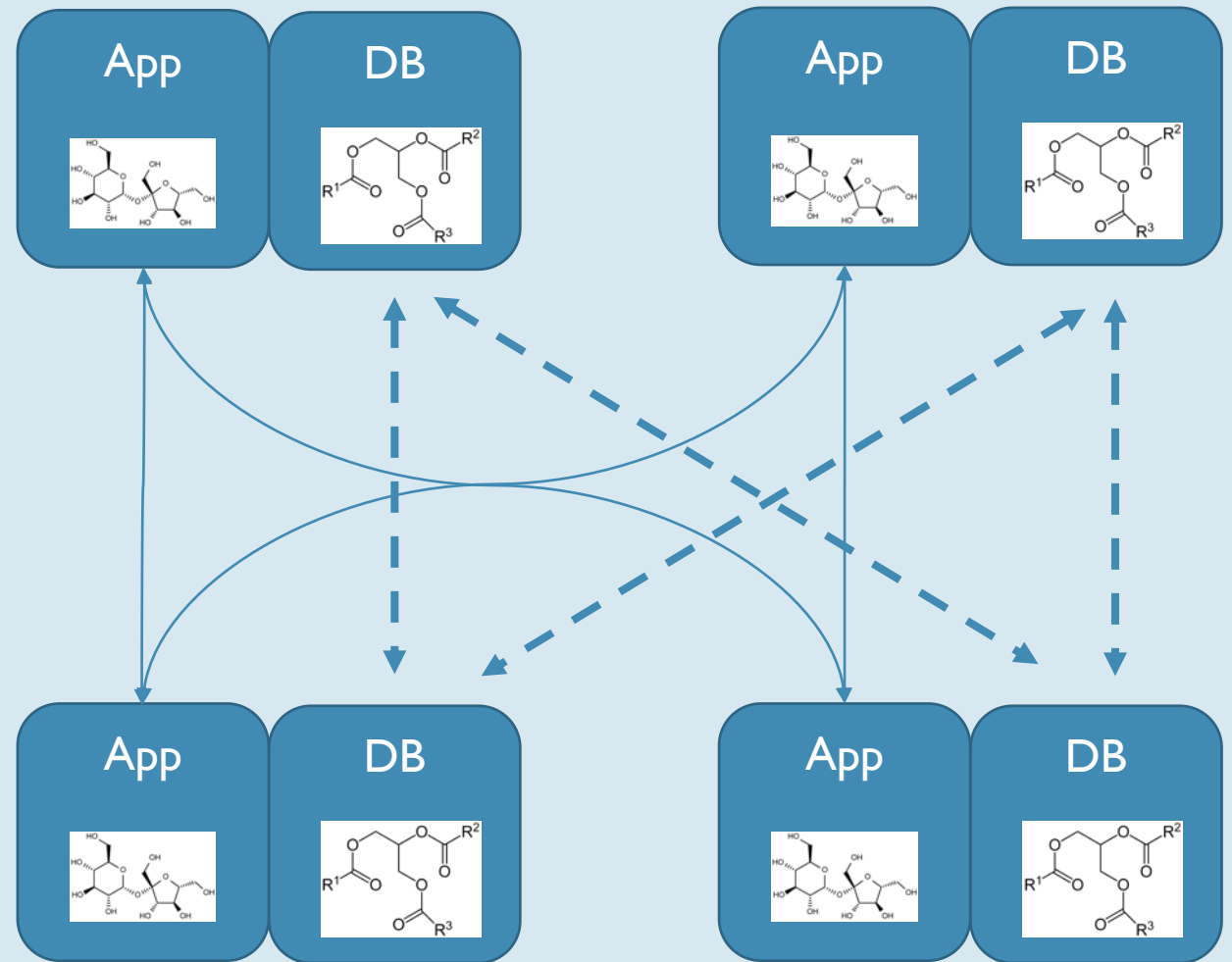
Robert 'Uncle Bob' Martin

Schema-free, problem-free?

It's never just one app/DB.

Now, where's the knowledge about the data?

Often:
everywhere - differently



Pretty sure I stole that diagram from Rich Hickey, but couldn't find the presentation again :-)

Act III – "The resolution"

- SQL DBs are **schema-on-write and meaning-on-read**
- Asking different questions produces new information
- It's not just "the data" but always **"the data" & "the query"** that makes meaning

Learning English

Considering context of text,
author, situation
Hemingway, Steinbeck, Foucault

Independent reading and writing –
grasp of concept, form, ideas
Jule Verne, Mark Twain

Short reading/writing, figures of
speech
"Bob's your uncle."

Syntax, vocabulary
"Hello, my name is!"

Learning SQL

Context of information, questions
and answers/results.
What does the data mean?

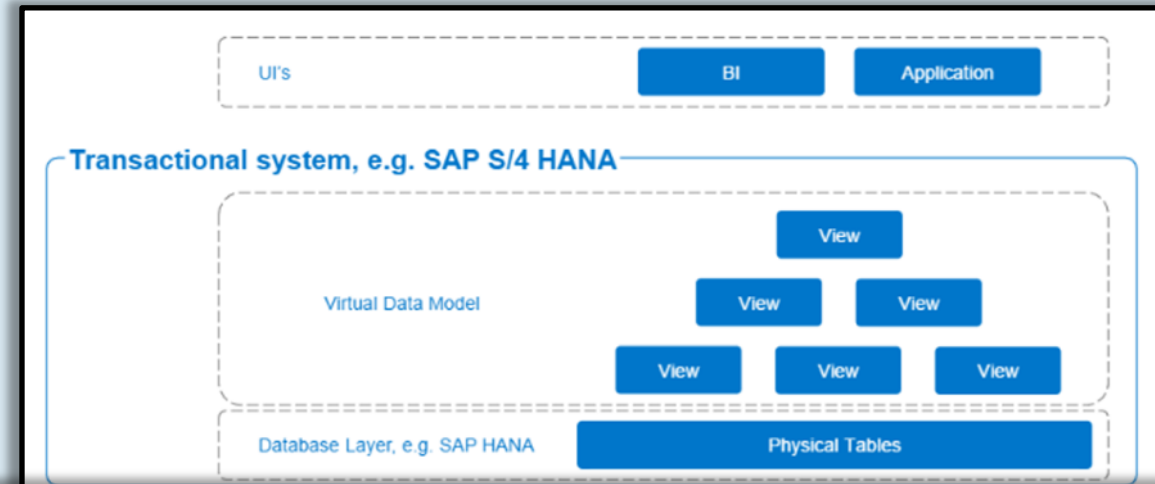
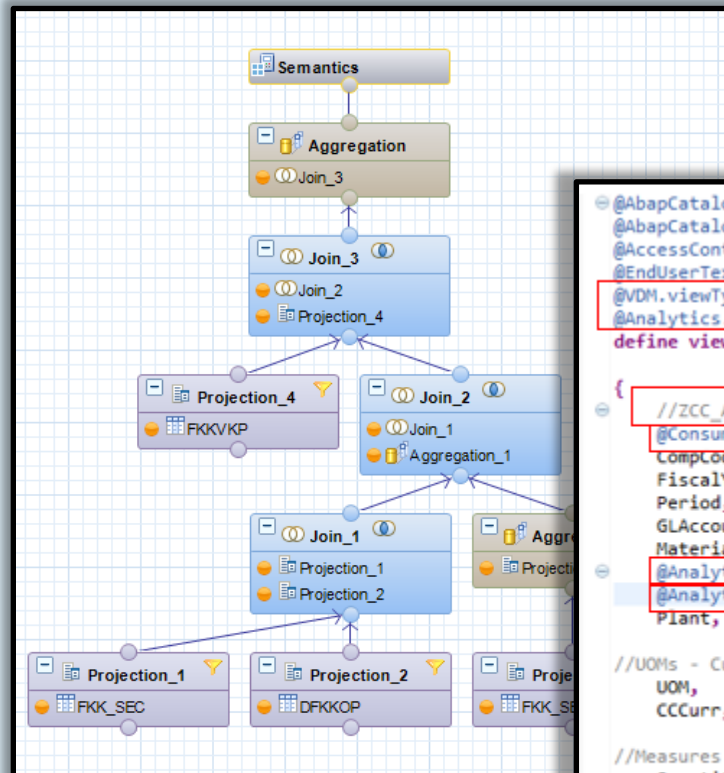
Deeper grasp of concepts,
awareness of complex functions
and problems
`GROUP BY GROUPING,
HIERARCHIES`

Recipes/templates
`WHERE 1=1, SUM(money) OVER()`

Syntax, vocabulary
`SELECT "Hello, world!" FROM ...`

SQL DBs are **schema-on-write** and **meaning-on-read**

- SAP's approach: virtual data models



```
@AbapCatalog.sqlViewName: 'ZCCACDOCAXXQ001'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Universal Journal Entry, Query'
@VDM.viewType: #CONSUMPTION
@Analytics.query: true
define view ZCC_ACDOCA_XX_Q001 as select from ZCC_ACDOCA_XX

{
    //ZCC_ACDOCA_XX
    @Consumption.filter: {mandatory: false, selectionType: #SINGLE, multipleSelections: true}
    CompCode,
    FiscalYear,
    Period,
    GLAccount,
    Material,
    @AnalyticsDetails.query.display: #KEY TEXT
    @AnalyticsDetails.query.axis: #ROWS
    Plant,

    //UOMs - Currencies
    UOM,
    CCCurr,

    //Measures
    Quantity,
    AmtCC
}
```

Indicates a #CONSUMPTION view type, and it's #TRUE so it can be read by the analytic manager (RSRT)

Commented code is prefixed with " //"

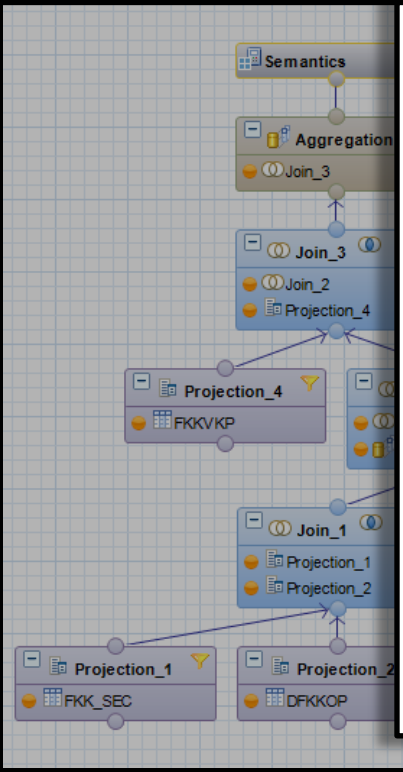
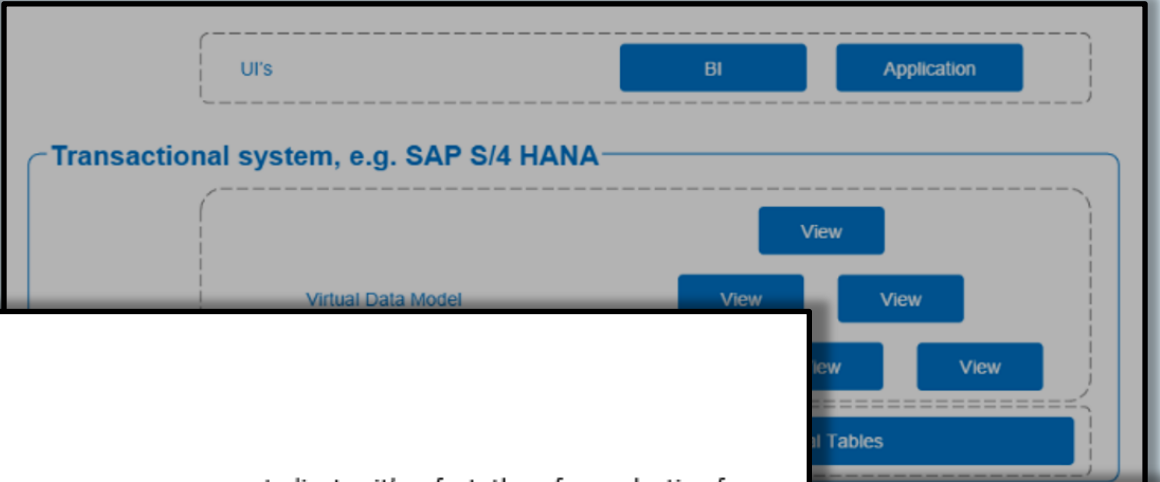
Definition of a query filter/prompt/variable

Definition of standard field display

Definition of standard layout, #ROWS, #COLUMNS or #FREE

SQL DBs are schema-on-write and meaning-on-read

- SAP's approach: virtual data models



```
[D] [ECD] ZBF_ACDOCA_XX
@AbapCatalog.sqlViewName: 'ZBFACDOCAXX'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Universal Journal Entry, Basic'
@VDM.viewType: #BASIC
@Analytics.dataCategory: #FACT
@Analytics.dataExtraction.enabled: true

define view ZBF_ACDOCA_XX as select from acdoca

{
  rbukrs as CompCode,
  gjahr as FiscalYear,
  poper as Period,
  racct as GLAccount,
  matnr as Material,
  werks as Plant,

  // UOMs - Currencies
  runit as UOM,
  rhcur as CCCurr,

  // Measures
  msl as Quantity,
  hsl as AmtCC
}
```

Indicates it's a fact, therefore selecting from a transactional source (table or view)

Selecting fields from the source table and providing meaningful aliases

view type, and it's #TRUE so
ic manager (RSRT)

d with " //"

prompt/variable

display

it, #ROWS, #COLUMNS or

Quantity,
AmtCC

<https://blogs.sap.com/2018/03/19/s4-embedded-analytics-the-virtual-data-model/>

Does that give us "self-service" answers?

"Average sales
volumes per work
day in this fiscal
year?"



GJAHR, TFAC,
JOIN... ?



Wasn't that the promise?



You are looking inside the world's most remarkable business machine . . . the IBM Electronic Calculator. It solves accounting and research problems faster than any other commercial calculator in general use.

GETTING YOUR ANSWERS
... at electronic speed!

IBM's vast engineering know-how is helping American business, industry and the Armed Forces get the answers . . . fast. Through its leadership in applying electronic principles to calculators and other types of punched card business machines, IBM has given greater speed, accuracy and economy to the nation's vital processes of calculating and accounting.

Already thousands of IBM Electronic Business Machines are in everyday use. We are continuing to manufacture them in quantity . . . as fast as quality production will permit.

IBM INTERNATIONAL BUSINESS MACHINES
590 MADISON AVENUE • NEW YORK 22, N. Y.



Operation in Real-Time . . .

In the field of missile development, there's a need for a computer capable of real-time performance — the famous Univac® Scientific. It's the ideal system for flight simulation and for on-line data reduction. It solves complex problems from purely sensed data at speeds that are compatible with real-time control.

Because of its ability to reduce volumes of data at tremendous speeds, the Univac Scientific System easily handles even the most difficult research problems. Furthermore, it offers many other outstanding characteristics, including: superb operating efficiency, obtained through large storage capacity . . . great programming versatility . . . the ability to operate simultaneously with a wide variety of input-output devices . . . and far greater reliability than any computer of its type.

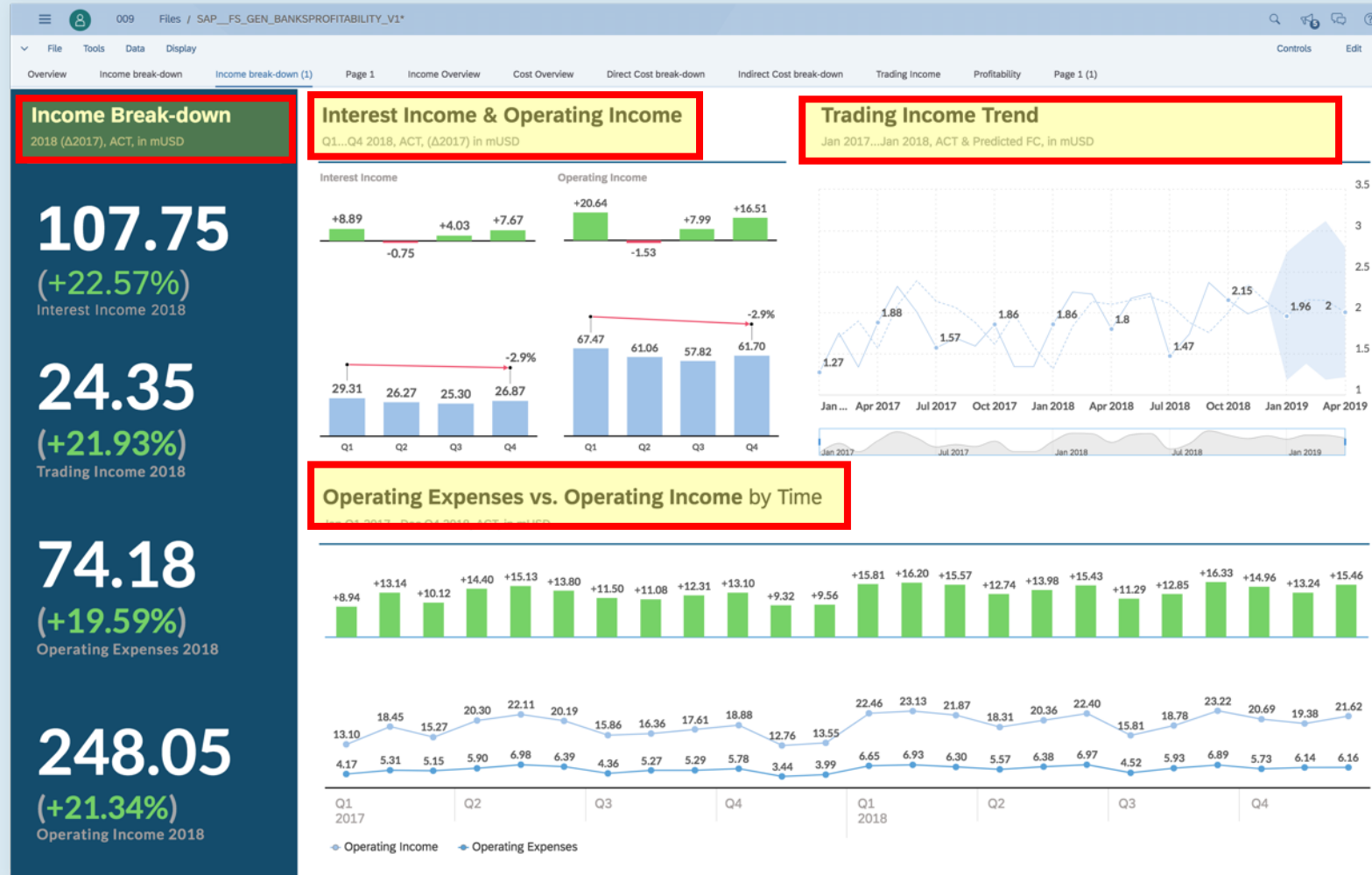
For more information about the Univac Scientific System or for information about ways in which you might apply the system to your particular problems, write on your business letterhead to . . .

ROOM 1315, 315 FOURTH AVE., NEW YORK 10
Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

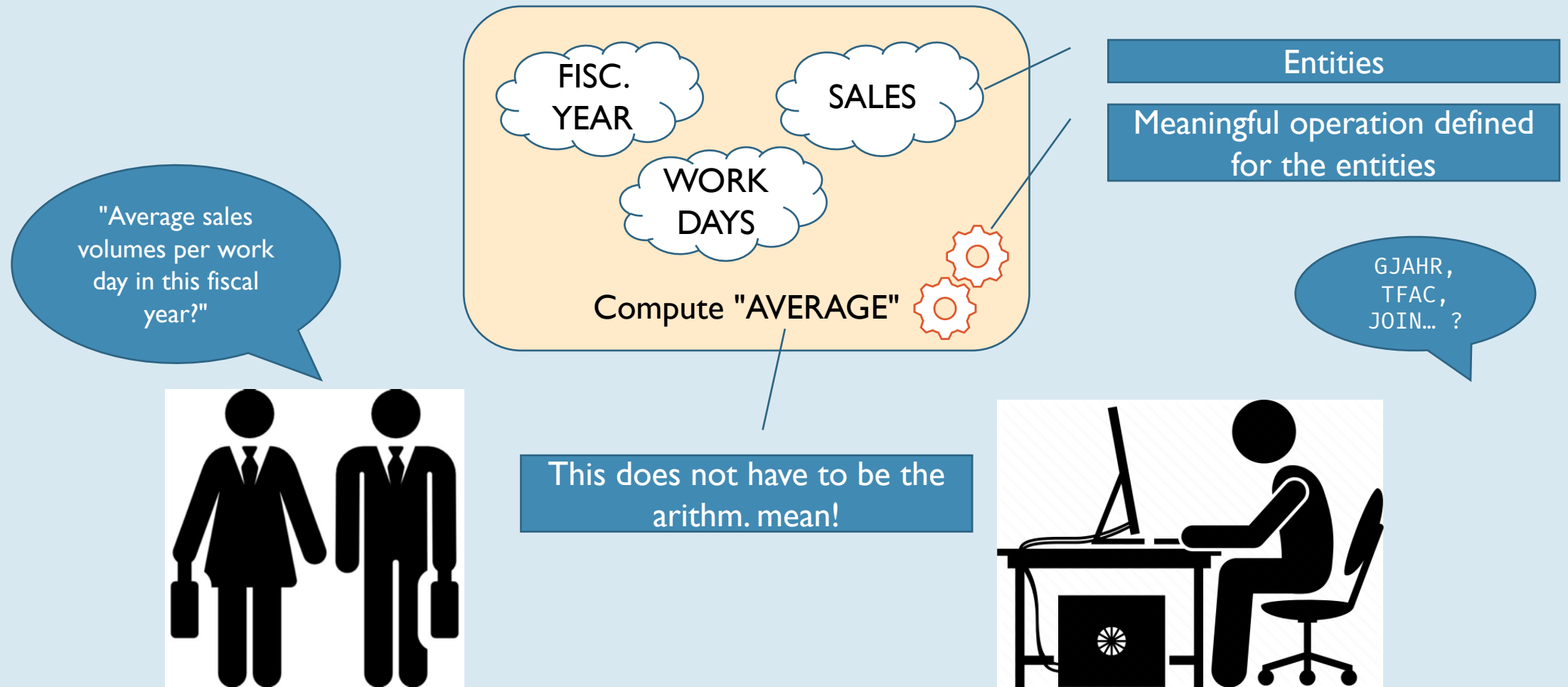
Who thinks this is the answer to a single question? What is it about?



Who thinks this is the answer to a single question? What is it about?

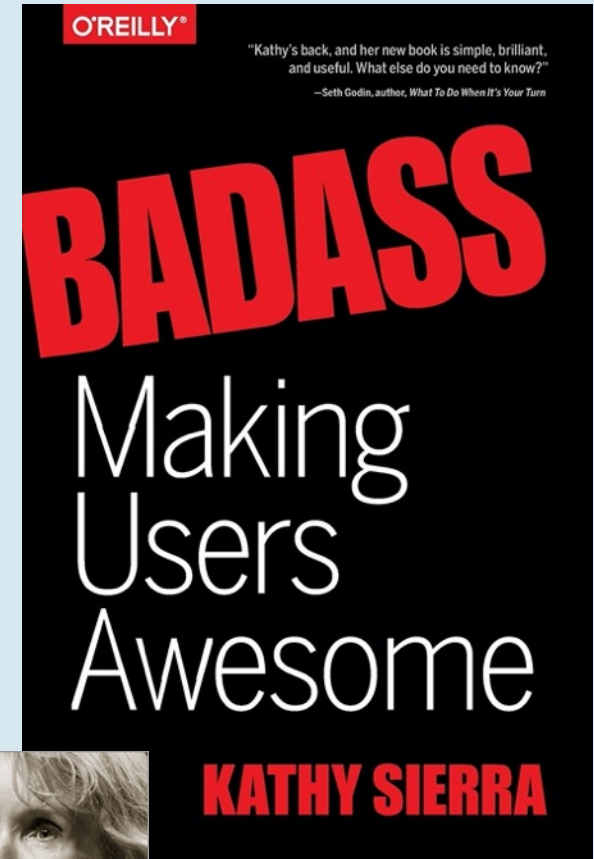


Does that give us "self-service" answers?



Make Users (and yourself) Awesome

- Help users of your code move forward – empower them
 - Including other developers and future-you
- Remove blocks and mental load

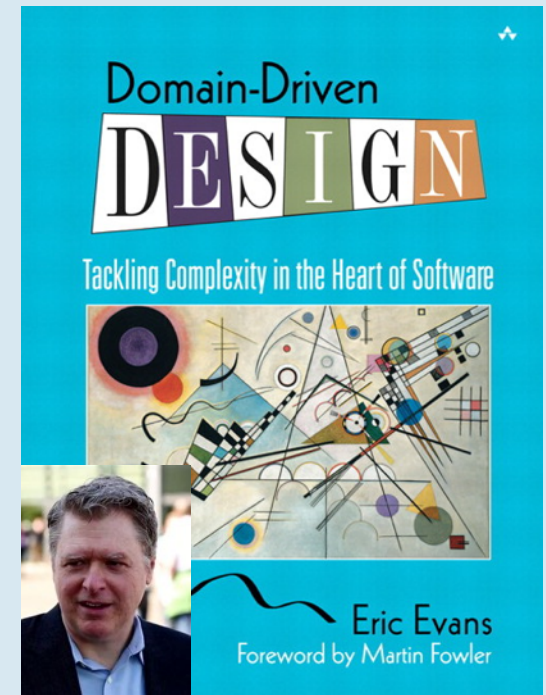


Kathy Sierra

Domain Driven Design

- "entity"
An object fundamentally defined not by its attributes, but by a thread of continuity and identity.

→ It's not the columns of the tables!
- "The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise"
– *Dijkstra, The humble programmer, Turing lecture 1972*



Eric Evans

Ask about your stuff – don't query tables

- Queries should focus on what they are about – not how the abstraction works
- *"Simple is not about having only one instance or operation, but the lack of interleaving. It's about single role, task, concept or dimension. It's about one **objective**."*
 - Rich Hickey

Simple Made Easy

LIKE 25 BOOKMARKS

View Presentation Speed: 1X 1.5X 2X



Download MP3 ANDROID APP 01:01:26

Summary

Rich Hickey emphasizes simplicity's virtues over easiness', showing that while many choose easiness they may end up with complexity, and the better way is to choose easiness along the simplicity path.

Simple

- One fold/braid
- One role
- One task
- One concept
- One dimension
- But not
 - One instance
 - One operation
 - About lack of interleaving, not cardinality
 - Objective

Refactor to better understanding

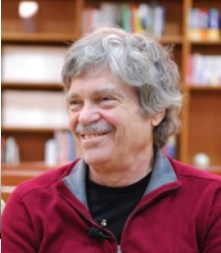
- By writing a query you are making your very own reading-schema
- Make the problem simpler by using a better abstraction
- "Point of view is worth 80 IQ"
- "You get simplicity by finding a slightly more sophisticated building block to build your theories out of."
- Alan Kay

Anyone Can Make A Doghouse



But what works for 1-1/2 feet usually cannot be scaled even by a factor of 100

The Superdome is several 100 times larger than a doghouse and has a clear span, but had to be made very differently



59 / 51:06

Act III – "The resolution"

- Having a common interpretation/meaning of data **is required** to make it useful
- One model to rule them all isn't it.
- DB models that correspond to domain concepts empower users and communication
- Allows to consider broader concerns, i.e. ethical limits/imperatives, structural unfairness/biases

Learning English

Considering context of text,
author, situation
Hemmingway, Steinbeck, Foucault

Independent reading and writing –
grasp of concept, form, ideas
Jule Verne, Mark Twain

Short reading/writing, figures of
speech
"Bob's your uncle."

Syntax, vocabulary
"Hello, my name is!"

Learning SQL

Context of information, questions
and answers/results.
What does the data mean?

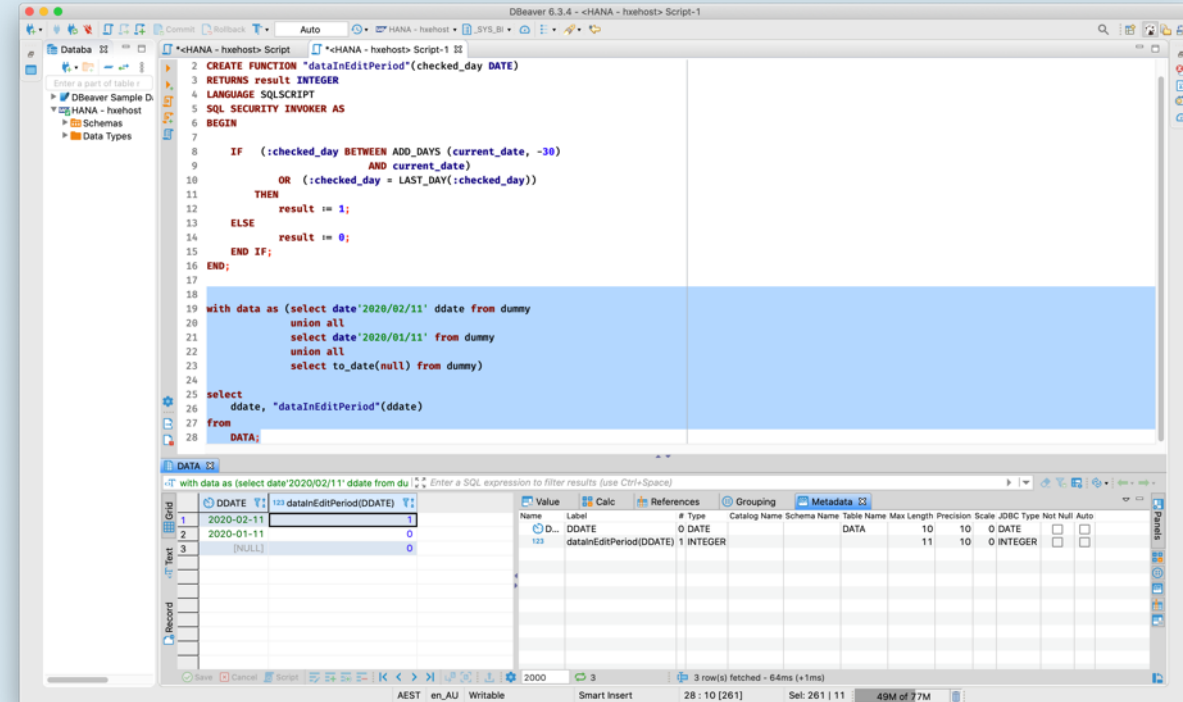
Deeper grasp of concepts,
awareness of complex functions
and problems
GROUP BY GROUPING,
HIERARCHIES

Recipes/templates
WHERE 1=1, SUM(money) OVER()

Syntax, vocabulary
SELECT "Hello, world!" FROM ...

Where to from here?

- What to do if you cannot change the "big model"?
- Make use of better tools
 - [DBeaver](#) - covers a vast range of features and functions across many DBMS.
 - [VSCode](#)
 - [SQLSquirrel](#)

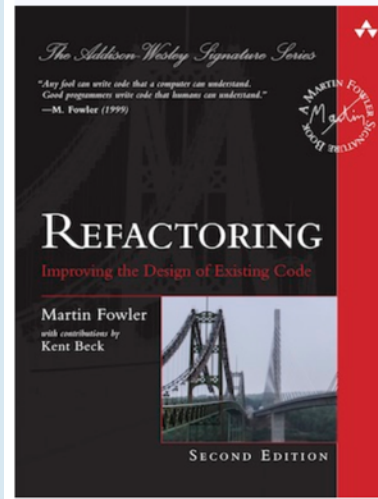


Where to from here?

- Make use of better tools
 - ~~automatic query formatting~~ (wanted: formatting based on **meaning**)
 - syntax highlighting
 - ligature-enabled font rendering ($\neq$, $\geq$, $\leq$ instead of $\!=$, $\>=$, $\<=$)
 - Code folding
 - Multi-cursor editing, advanced search & replace, etc.
 - Versioning control

Refactor towards understanding

- Changing code structure to make it easier to understand and change
- Not about making code run faster but to make development faster
- Creating options for alternative solutions



Refactoring

Improving the Design of Existing Code

by Martin Fowler, with Kent Beck
2018

The guide to how to transform code with safe and rapid process, vital to keeping it cheap and easy to modify for future needs.

amazon

informIT
icon



Kent Beck



Martin Fowler

Literate SQL, readable SQL

- Intention revealing names
- Visually prominent in code
- In order of human logic

<https://modern-sql.com/use-case/literate-sql>

```
WITH _customer_credits_debits
> ("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS ...
> SELECT ...
FROM
|...| _customer_credits_debits
GROUP BY
|...| "Customer Code"
|...| , "Customer Name"
HAVING
|...| SUM("SYS Balance Due") >= 0
ORDER BY
|...| "Customer Code";
```

Literate SQL, readable SQL

- Intention revealing names
- Visually prominent in code
- In order of human logic

<https://modern-sql.com/use-case/literate-sql>

```
WITH _customer_credits_debits
> ("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS ...
> SELECT ...
FROM
  ... _customer_credits_debits
GROUP BY
  ... "Customer Code"
  ... , "Customer Name"
HAVING
  ... SUM("SYS Balance Due") <math>\neq</math> 0
ORDER BY
  ... "Customer Code";
```

WITH – naming part-query

WITH – projected column names

Code folding hides implementation detail

Example

Original
SQL

```
<HANA - hxehost> ref_step00 - orig post
4 SELECT OCRD."CardCode" AS "Customer Code", OCRD."CardName" AS "Customer Name", SUM(
5     CASE
6         WHEN "SYS Cred" < 0 THEN "SYS Cred" * -1
7         ELSE "SYS Deb"
8     END) AS "Balance Due", IFNULL(SUM(
9     CASE
10        WHEN DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 0 THEN
11        CASE
12            WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
13            ELSE JDT1."BalDueDeb"
14        END
15        END), 0.00) AS "Future Remit", IFNULL(SUM(
16        CASE
17            WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 0 AND
18            DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 30) THEN
19            CASE
20                WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
21                ELSE JDT1."BalDueDeb"
22            END
23            END), 0.00) AS "0-30 days", IFNULL(SUM(
24            CASE
25                WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 30 AND
26                DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 60) THEN
27                CASE
28                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
29                    ELSE JDT1."BalDueDeb"
30                END
31                END), 0.00) AS "31 to 60 days", IFNULL(SUM(
32                CASE
33                    WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 60 AND
34                    DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 90) THEN
35                    CASE
36                        WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
37                        ELSE JDT1."BalDueDeb"
38                    END
39                END)
40    ) AS "Balance Due", IFNULL(SUM(
41        CASE
42            WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 90) THEN
43            CASE
44                WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
45                ELSE JDT1."BalDueDeb"
46            END
47            END)
48    ) AS "91+ days"
49 ) FROM OCRD
50 LEFT JOIN JDT1 ON OCRD."CardCode" = JDT1."CardCode"
51 WHERE OCRD."CardCode" <> ''
```

	Customer Code	Customer Name	Balance Due	Future Remit	0-30 days	31 to 60 days
1	CC0001	Robel, Hermiston and Smith	-627,931.508	-291,028.86	-43,153.118	-33,409.51
2	CC0007	Hand-Murazik	-632,987.217	-288,355.601	-41,058.501	-38,910.4
3	CC0013	Beatty, Kris and Wolff	-620,051.886	-276,076.3	-42,590.106	-32,784.61
4	CC0019	Leffler-Koch	-635,646.453	-285,300.567	-35,670.232	-33,576.91
5	CC0020	O'Conner, Swaniawski and R	-622,834.399	-288,415.426	-33,284.658	-37,022.71
6	CC0022	Brakus-Braun	-620,072.744	-292,218.733	-41,318.508	-39,999.21
7	CC0026	Kautzer, Wolf and Conn	-624,792.548	-284,289.679	-41,438.527	-38,270.61
8	CC0043	Collier-Haley	-629,535.548	-298,654.913	-36,114.367	-40,696.1
9	CC0045	Hodkiewicz, Cummerata and	-622,480.056	-306,292.792	-43,441.827	-39,978.91
10	CC0067	Eichmann, Mann and Bins	-639,821.278	-302,998.601	-35,672.301	-39,816.21
11	CC0073	Sporer Group	-627,416.476	-290,245.451	-38,965.736	-36,862.11
12	CC0078	Stracke Group	-622,963.823	-291,549.189	-36,085.845	-38,656.11
13	CC0079	Farrell LLC	-611,338.027	-294,790.706	-42,381.414	-39,889.1

Original
result

Save Cancel Script 2000 1,382
1382 row(s) fetched - 2.534s (+53ms)

Refactored
SQL

```
<HANA - hxehost> ref_step04 - ANSI join syntax
1 set schema refactor;
2 -- ORIG POST
3 -- 1 - replace "current_timestamp" with current_timestamp
4 --
5 -- 2 - reformat for readability
6 -- 3 - replace <=> with <=> for editors with ligature support
7 -- 4 - change to ANSI join syntax
8
9 SELECT
10     OCRD."CardCode" AS "Customer Code"
11     , OCRD."CardName" AS "Customer Name"
12     , SUM(CASE
13         WHEN "SYS Cred" < 0 THEN "SYS Cred" * -1
14         ELSE "SYS Deb"
15     END) AS "Balance Due"
16     , IFNULL(SUM(
17         CASE
18             WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
19             CASE
20                 WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
21                 ELSE JDT1."BalDueDeb"
22             END
23             END), 0.00) AS "Future Remit"
24     , IFNULL(SUM(
25         CASE
26             WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 0 AND
27             DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
28             CASE
29                 WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
30                 ELSE JDT1."BalDueDeb"
31             END
32             END), 0.00) AS "0-30 days"
33     , IFNULL(SUM(
34         CASE
35             WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 30 AND
36             DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
37             CASE
38                 WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
39                 ELSE JDT1."BalDueDeb"
40             END
41             END), 0.00) AS "31 to 60 days"
42     , IFNULL(SUM(
43         CASE
44             WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 60 AND
45             DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
46             CASE
47                 WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
48                 ELSE JDT1."BalDueDeb"
49             END
50             END), 0.00) AS "91+ days"
51 ) FROM OCRD
52 LEFT JOIN JDT1 ON OCRD."CardCode" = JDT1."CardCode"
53 WHERE OCRD."CardCode" <> ''
```

	Customer Code	Customer Name	Balance Due	Future Remit	0-30 days	31 to 60 days
1	CC0001	Robel, Hermiston and Smith	-627,931.508	-291,028.86	-43,153.118	-33,409.51
2	CC0007	Hand-Murazik	-632,987.217	-288,355.601	-41,058.501	-38,910.4
3	CC0013	Beatty, Kris and Wolff	-620,051.886	-276,076.3	-42,590.106	-32,784.61
4	CC0019	Leffler-Koch	-635,646.453	-285,300.567	-35,670.232	-33,576.91
5	CC0020	O'Conner, Swaniawski and Rogahn	-622,834.399	-288,415.426	-33,284.658	-37,022.71
6	CC0022	Brakus-Braun	-620,072.744	-292,218.733	-41,318.508	-39,999.21
7	CC0026	Kautzer, Wolf and Conn	-624,792.548	-284,289.679	-41,438.527	-38,270.61
8	CC0043	Collier-Haley	-629,535.548	-298,654.913	-36,114.367	-40,696.1
9	CC0045	Hodkiewicz, Cummerata and Will	-622,480.056	-306,292.792	-43,441.827	-39,978.91
10	CC0067	Eichmann, Mann and Bins	-639,821.278	-302,998.601	-35,672.301	-39,816.21
11	CC0073	Sporer Group	-627,416.476	-290,245.451	-38,965.736	-36,862.11
12	CC0078	Stracke Group	-622,963.823	-291,549.189	-36,085.845	-38,656.11
13	CC0079	Farrell LLC	-611,338.027	-294,790.706	-42,381.414	-39,889.1

Refactored
result

Save Cancel Script 2000 1,382
1382 row(s) fetched - 793ms (+13ms)

Original

```
SELECT OCRD."CardCode" AS "Customer Code", OCRD."CardName" AS "Customer Name", SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) AS "Balance Due", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 0 THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1 +
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "Future Remit", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) ≥ 0 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 30) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "0-30 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) ≥ 30 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 60) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "31 to 60 days", IFNULL(SUM(
```

```
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) ≥ 60 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 90) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "61 to 90 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) ≥ 90 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 120) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "91 to 120 days", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) ≥ 120 THEN
CASE
WHEN "BalDueCred" <> 0 THEN "BalDueCred" * -1
ELSE "BalDueDeb"
END
END), 0.00) AS "120+ days"
FROM JDT1,
OCRD
WHERE JDT1."ShortName" = OCRD."CardCode" AND OCRD."CardType" = 'c'
GROUP BY OCRD."CardCode", OCRD."CardName" HAVING SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) > 0 OR SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) < 0
ORDER BY OCRD."CardCode";
```

Original query result

Customer Code	Customer Name	Balance Due	Future Remit	0-30 days	31 to 60 days	61 to 90 days	91 to 120 days	120+ days
-----	-----	-----	-----	-----	-----	-----	-----	-----
CC0001	Robel, Hermiston and Smith	-627931.508	-292253.3320	-43440.598	-33361.374000	-36461.363000	-37044.144000	-182889.416
CC0007	Hand-Murazik	-632987.217	-290324.8610	-41162.541	-37237.034000	-38708.927000	-41575.572000	-182358.897
CC0013	Beatty, Kris and Wolff	-620051.886	-276851.2240	-44209.015	-30972.273000	-39446.000000	-36691.526000	-185799.788
CC0019	Leffler-Koch	-635646.453	-285973.0680	-35738.858	-34188.120000	-39915.037000	-39743.413000	-186057.643
CC0020	O'Conner, Swaniawski and Rogahn	-622834.399	-289215.2410	-33202.611	-38646.829000	-38930.113000	-37384.327000	-184100.311
CC0022	Brakus-Braun	-620072.744	-293423.0760	-40851.999	-39740.231000	-42606.646000	-34726.326000	-170138.382
CC0026	Kautzer, Wolf and Conn	-624792.548	-285367.4400	-41901.141	-38295.999000	-39945.978000	-34297.624000	-180138.311
CC0043	Collier-Haley	-629535.548	-300841.4300	-35856.967	-39892.859000	-34951.770000	-35664.465000	-169146.612
CC0045	Hodkiewicz, Cummerata and Will	-622480.056	-308358.7810	-43362.080	-39065.686000	-31912.466000	-39727.909000	-168080.452
CC0067	Eichmann, Mann and Bins	-639821.278	-304936.9680	-35080.969	-39500.653000	-31518.377000	-33243.902000	-181726.008
[...]								

➔ Customer Aging /Accounts Receivables aging report

Step 1 – replace "now()" with current_timestamp

```
SELECT OCRD."CardCode" AS "Customer Code", OCRD."CardName" AS "Customer Name", SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) AS "Balance Due", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 0 THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1 +
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "Future Remit", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 0 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 30) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "0-30 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 30 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 60) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "31 to 60 days", IFNULL(SUM(
```

```
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 60 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 90) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "61 to 90 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 90 AND
DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) < 120) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "91 to 120 days", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", (SELECT NOW() "NOW" FROM DUMMY)) >= 120 THEN
CASE
WHEN "BalDueCred" <> 0 THEN "BalDueCred" * -1
ELSE "BalDueDeb"
END
END), 0.00) AS "120+ days"
FROM JDT1,
OCRD
WHERE JDT1."ShortName" = OCRD."CardCode" AND OCRD."CardType" = 'c'
GROUP BY OCRD."CardCode", OCRD."CardName" HAVING SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) > 0 OR SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) < 0
ORDER BY OCRD."CardCode";
```

Step 1 – replace "now()" with current_timestamp

```
SELECT OCRD."CardCode" AS "Customer Code", OCRD."CardName" AS "Customer Name", SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) AS "Balance Due", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "Future Remit", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 0 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "0-30 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 30 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "31 to 60 days", IFNULL(SUM(
```

```
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 60 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "61 to 90 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 90 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "91 to 120 days", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 120 THEN
CASE
WHEN "BalDueCred" <> 0 THEN "BalDueCred" * -1
ELSE "BalDueDeb"
END
END), 0.00) AS "120+ days"
FROM JDT1,
OCRD
WHERE JDT1."ShortName" = OCRD."CardCode" AND OCRD."CardType" = 'c'
GROUP BY OCRD."CardCode", OCRD."CardName" HAVING SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) > 0 OR SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) < 0
ORDER BY OCRD."CardCode";
```

Step 2 – reformat for readability

```
SELECT OCRD."CardCode" AS "Customer Code", OCRD."CardName" AS "Customer Name", SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) AS "Balance Due", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "Future Remit", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 0 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "0-30 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 30 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "31 to 60 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 60 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "61 to 90 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 90 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "91 to 120 days", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 120 THEN
CASE
WHEN "BalDueCred" <> 0 THEN "BalDueCred" * -1
ELSE "BalDueDeb"
END
END), 0.00) AS "120+ days"
FROM JDT1,
OCRD
WHERE JDT1."ShortName" = OCRD."CardCode" AND OCRD."CardType" = 'c'
GROUP BY OCRD."CardCode", OCRD."CardName" HAVING SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) > 0 OR SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) < 0
ORDER BY OCRD."CardCode";
```

```
SELECT OCRD."CardCode" AS "Customer Code", OCRD."CardName" AS "Customer Name", SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) AS "Balance Due", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "Future Remit", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 0 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "0-30 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 30 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "31 to 60 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 60 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "61 to 90 days", IFNULL(SUM(
CASE
WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 90 AND
DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
CASE
WHEN JDT1."BalDueCred" <> 0 THEN JDT1."BalDueCred" * -1
ELSE JDT1."BalDueDeb"
END
END), 0.00) AS "91 to 120 days", IFNULL(SUM(
CASE
WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) >= 120 THEN
CASE
WHEN "BalDueCred" <> 0 THEN "BalDueCred" * -1
ELSE "BalDueDeb"
END
END), 0.00) AS "120+ days"
FROM JDT1,
OCRD
WHERE JDT1."ShortName" = OCRD."CardCode" AND OCRD."CardType" = 'c'
GROUP BY OCRD."CardCode", OCRD."CardName" HAVING SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) > 0 OR SUM(
CASE
WHEN "SYSCred" <> 0 THEN "SYSCred" * -1
ELSE "SYSDeb"
END) < 0
ORDER BY OCRD."CardCode";
```

Step 2 – reformat for readability

```
SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , SUM(CASE
        WHEN "SYS Cred" < 0 THEN "SYS Cred" * -1
        ELSE "SYS Deb"
        END) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
            CASE
                WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                ELSE JDT1."BalDueDeb"
            END
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
                CASE
                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
                CASE
                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "31 to 60 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
                CASE
                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "61 to 90 days"
```

```

-- , IFNULL(SUM(
--     CASE
--     WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
--          DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
--     CASE
--     WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
--     ELSE JDT1."BalDueDeb"
--     END
--     END), 0.00) AS "91 to 120 days"
-- , IFNULL(SUM(
--     CASE
--     WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
--     CASE
--     WHEN "BalDueCred" < 0 THEN "BalDueCred" * -1
--     ELSE "BalDueDeb"
--     END
--     END), 0.00) AS "120+ days"
FROM JDT1,
-- OCRD
WHERE
-- JDT1."ShortName" = OCRD."CardCode"
-- AND OCRD."CardType" = 'c'
GROUP BY
-- OCRD."CardCode", OCRD."CardName"
HAVING SUM(
CASE
-- WHEN "SYSCred" < 0 THEN "SYSCred" * -1
-- ELSE "SYSDeb"
END) > 0
OR SUM(
CASE
-- WHEN "SYSCred" < 0 THEN "SYSCred" * -1
-- ELSE "SYSDeb"
END) < 0
ORDER BY
-- OCRD."CardCode";

```

Step 3 – replace < > with != for editors with ligature support

```

SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , SUM(CASE
        WHEN "SYS Cred" < 0 THEN "SYS Cred" * -1
        ELSE "SYS Deb"
        END) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
            CASE
                WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                ELSE JDT1."BalDueDeb"
            END
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
                CASE
                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
                CASE
                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "31 to 60 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
                CASE
                    WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "61 to 90 days"

```

```

-- , IFNULL(SUM(
--     CASE
--     WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
--          DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
--     CASE
--     WHEN JDT1."BalDueCred" < 0 THEN JDT1."BalDueCred" * -1
--     ELSE JDT1."BalDueDeb"
--     END
--     END), 0.00) AS "91 to 120 days"
-- , IFNULL(SUM(
--     CASE
--     WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
--     CASE
--     WHEN "BalDueCred" < 0 THEN "BalDueCred" * -1
--     ELSE "BalDueDeb"
--     END
--     END), 0.00) AS "120+ days"
FROM JDT1,
-- OCRD
WHERE
-- JDT1."ShortName" = OCRD."CardCode"
-- AND OCRD."CardType" = 'c'
GROUP BY
-- OCRD."CardCode", OCRD."CardName"
HAVING SUM(
    CASE
    WHEN "SYSCred" < 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
    END) > 0
OR SUM(
    CASE
    WHEN "SYSCred" < 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
    END) < 0
ORDER BY
-- OCRD."CardCode";

```

Step 3 – replace < > with != for editors with ligature support

```
SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , SUM(CASE
        WHEN "SYS Cred" ≠ 0 THEN "SYS Cred" * -1
        ELSE "SYS Deb"
        END) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
            CASE
                WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                ELSE JDT1."BalDueDeb"
            END
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
                CASE
                    WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
                CASE
                    WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "31 to 60 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
                CASE
                    WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                    ELSE JDT1."BalDueDeb"
                END
            END), 0.00) AS "61 to 90 days"
```

```

-- , IFNULL(SUM(
--     CASE
--     WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
--          DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
--         CASE
--         WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
--         ELSE JDT1."BalDueDeb"
--         END
--     END), 0.00) AS "91 to 120 days"
-- , IFNULL(SUM(
--     CASE
--     WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
--         CASE
--         WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
--         ELSE "BalDueDeb"
--         END
--     END), 0.00) AS "120+ days"
FROM JDT1,
-- OCRD
WHERE
-- JDT1."ShortName" = OCRD."CardCode"
-- AND OCRD."CardType" = 'c'
GROUP BY
-- OCRD."CardCode", OCRD."CardName"
HAVING SUM(
    CASE
    WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
    END) > 0
OR SUM(
    CASE
    WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
    END) < 0
ORDER BY
-- OCRD."CardCode";

```

Step 4 – ANSI join syntax

```
SELECT
  .... OCRD."CardCode" AS "Customer Code"
  .... OCRD."CardName" AS "Customer Name"
  .... SUM(CASE
    >      .... WHEN "SYS Cred" ≠ 0 THEN "SYS Cred" * -1
    >      .... ELSE "SYS Deb"
    >      .... END) AS "Balance Due"
  .... , IFNULL(SUM(CASE
    >      .... WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
    >      .... CASE
    >      ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    >      ....   .... ELSE JDT1."BalDueDeb"
    >      ....   .... END
    >      .... END), 0.00) AS "Future Remit"
  .... , IFNULL(SUM(
    >      .... CASE
    >      ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
    >      ....   ....   DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
    >      ....   .... CASE
    >      ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    >      ....   ....   .... ELSE JDT1."BalDueDeb"
    >      ....   ....   .... END
    >      ....   .... END), 0.00) AS "0-30 days"
  .... , IFNULL(SUM(
    >      .... CASE
    >      ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
    >      ....   ....   DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
    >      ....   .... CASE
    >      ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    >      ....   ....   .... ELSE JDT1."BalDueDeb"
    >      ....   ....   .... END
    >      ....   .... END), 0.00) AS "31 to 60 days"
  .... , IFNULL(SUM(
    >      .... CASE
    >      ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
    >      ....   ....   DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
    >      ....   .... CASE
    >      ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    >      ....   ....   .... ELSE JDT1."BalDueDeb"
    >      ....   ....   .... END
    >      ....   .... END), 0.00) AS "61 to 90 days"
```

```
    .... , IFNULL(SUM(
    >      .... CASE
    >      ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
    >      ....   ....   DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
    >      ....   .... CASE
    >      ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    >      ....   ....   .... ELSE JDT1."BalDueDeb"
    >      ....   ....   .... END
    >      ....   .... END), 0.00) AS "91 to 120 days"
  .... , IFNULL(SUM(
    >      .... CASE
    >      ....   .... WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
    >      ....   .... CASE
    >      ....   ....   .... WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
    >      ....   ....   .... ELSE "BalDueDeb"
    >      ....   ....   .... END
    >      ....   .... END), 0.00) AS "120+ days"

FROM JDT1,
  .... OCRD
WHERE
  .... JDT1."ShortName" = OCRD."CardCode"
  .... AND OCRD."CardType" = 'c'
GROUP BY
  .... OCRD."CardCode", OCRD."CardName"
HAVING SUM(
  >      CASE
  >      .... WHEN "SYS Cred" ≠ 0 THEN "SYS Cred" * -1
  >      .... ELSE "SYS Deb"
  >      .... END) > 0
  OR SUM(
  >      CASE
  >      .... WHEN "SYS Cred" ≠ 0 THEN "SYS Cred" * -1
  >      .... ELSE "SYS Deb"
  >      .... END) < 0
ORDER BY
  .... OCRD."CardCode";
```

Step 4 – ANSI join syntax

```

SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , SUM(CASE
        WHEN "SYS Cred" ≠ 0 THEN "SYS Cred" * -1
        ELSE "SYS Deb"
        END) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
        CASE
            WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
            ELSE JDT1."BalDueDeb"
            END
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
            DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
            CASE
                WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                ELSE JDT1."BalDueDeb"
            END
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
            DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
            CASE
                WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                ELSE JDT1."BalDueDeb"
            END
            END), 0.00) AS "31 to 60 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
            DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
            CASE
                WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
                ELSE JDT1."BalDueDeb"
            END
            END), 0.00) AS "61 to 90 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND

```

```

        ...END), 0.00) AS "91 to 120 days"
    , IFNULL(SUM(
        CASE
            WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
                CASE
                    WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
                    ELSE "BalDueDeb"
                END
            END), 0.00) AS "120+ days"
FROM
    JDT1 INNER JOIN OCRD
ON JDT1."ShortName" = OCRD."CardCode"
AND OCRD."CardType" = 'c'

GROUP BY
    OCRD."CardCode", OCRD."CardName"
HAVING SUM(
    CASE
        WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
        ELSE "SYSDeb"
    END) > 0
OR SUM(
    CASE
        WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
        ELSE "SYSDeb"
    END) < 0
ORDER BY
    OCRD."CardCode";

```

Step 5 – compact having clause

```
SELECT
  OCRD."CardCode" AS "Customer Code"
  , OCRD."CardName" AS "Customer Name"
  , SUM(CASE
    WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
    END) AS "Balance Due"
  , IFNULL(SUM(CASE
    WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
      CASE
        WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
        ELSE JDT1."BalDueDeb"
      END
    END), 0.00) AS "Future Remit"
  , IFNULL(SUM(
    CASE
      WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
        DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
        CASE
          WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
          ELSE JDT1."BalDueDeb"
        END
    END), 0.00) AS "0-30 days"
  , IFNULL(SUM(
    CASE
      WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
        DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
        CASE
          WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
          ELSE JDT1."BalDueDeb"
        END
    END), 0.00) AS "31 to 60 days"
  , IFNULL(SUM(
    CASE
      WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
        DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
        CASE
          WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
          ELSE JDT1."BalDueDeb"
        END
    END), 0.00) AS "61 to 90 days"
  , IFNULL(SUM(
    CASE
      WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
```

```
    DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
      CASE
        WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
        ELSE JDT1."BalDueDeb"
      END
    END), 0.00) AS "91 to 120 days"
  , IFNULL(SUM(
    CASE
      WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
        CASE
          WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
          ELSE "BalDueDeb"
        END
    END), 0.00) AS "120+ days"

FROM
  JDT1 INNER JOIN OCRD
  ON JDT1."ShortName" = OCRD."CardCode"
  AND OCRD."CardType" = 'c'

GROUP BY
  OCRD."CardCode", OCRD."CardName"

HAVING SUM(
  CASE
    WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
  END) > 0
  OR SUM(
  CASE
    WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
    ELSE "SYSDeb"
  END) < 0

ORDER BY
  OCRD."CardCode";
```

Step 5 – compact having clause

```
SELECT
  .... OCRD."CardCode" AS "Customer Code"
  .... , OCRD."CardName" AS "Customer Name"
  .... , SUM(CASE
  ....   .... WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
  ....   .... ELSE "SYSDeb"
  ....   .... END) AS "Balance Due"
  .... , IFNULL(SUM(CASE
  ....   .... WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
  ....   .... CASE
  ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
  ....   ....   .... ELSE JDT1."BalDueDeb"
  ....   .... END
  ....   .... END), 0.00) AS "Future Remit"
  .... , IFNULL(SUM(
  ....   .... CASE
  ....   ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
  ....   ....   ....   .... DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
  ....   ....   .... CASE
  ....   ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
  ....   ....   ....   .... ELSE JDT1."BalDueDeb"
  ....   ....   .... END
  ....   .... END), 0.00) AS "0-30 days"
  .... , IFNULL(SUM(
  ....   .... CASE
  ....   ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
  ....   ....   ....   .... DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
  ....   ....   .... CASE
  ....   ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
  ....   ....   ....   .... ELSE JDT1."BalDueDeb"
  ....   ....   .... END
  ....   .... END), 0.00) AS "31 to 60 days"
```

```
    .... , IFNULL(SUM(
    ....   .... CASE
    ....   ....   .... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
    ....   ....   ....   .... DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
    ....   ....   .... CASE
    ....   ....   ....   .... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    ....   ....   ....   .... ELSE JDT1."BalDueDeb"
    ....   ....   .... END
    ....   .... END), 0.00) AS "91 to 120 days"
    .... , IFNULL(SUM(
    ....   .... CASE
    ....   ....   .... WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
    ....   ....   .... CASE
    ....   ....   ....   .... WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
    ....   ....   ....   .... ELSE "BalDueDeb"
    ....   ....   .... END
    ....   .... END), 0.00) AS "120+ days"

FROM
  .... JDT1 INNER JOIN OCRD
  .... ON JDT1."ShortName" = OCRD."CardCode"
  .... AND OCRD."CardType" = 'c'

GROUP BY
  .... OCRD."CardCode", OCRD."CardName"
HAVING SUM(
  .... CASE
  ....   .... WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
  ....   .... ELSE "SYSDeb"
  .... END) ≠ 0

ORDER BY
  .... OCRD."CardCode";
```

Step 6 - pull out CASE-WHEN-ELSE credit-debit-balance into function

```
SELECT
  ... OCRD."CardCode" AS "Customer Code"
  ... OCRD."CardName" AS "Customer Name"
  ... SUM(CASE
    ... WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
    ... ELSE "SYSDeb"
    ... END) AS "Balance Due"
  ... , IFNULL(SUM(CASE
    ... WHEN DAYS BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
    ... CASE
    ... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    ... ELSE JDT1."BalDueDeb"
    ... END
    ... END), 0.00) AS "Future Remit"
  ... , IFNULL(SUM(
    ... CASE
    ... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
    ... DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
    ... CASE
    ... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    ... ELSE JDT1."BalDueDeb"
    ... END
    ... END), 0.00) AS "0-30 days"
  ... , IFNULL(SUM(
    ... CASE
    ... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
    ... DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
    ... CASE
    ... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
    ... ELSE JDT1."BalDueDeb"
    ... END
    ... END), 0.00) AS "31 to 60 days"
```

```
... , IFNULL(SUM(
  ... CASE
  ... WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
  ... DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
  ... CASE
  ... WHEN JDT1."BalDueCred" ≠ 0 THEN JDT1."BalDueCred" * -1
  ... ELSE JDT1."BalDueDeb"
  ... END
  ... END), 0.00) AS "91 to 120 days"
  ... , IFNULL(SUM(
    ... CASE
    ... WHEN DAYS BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
    ... CASE
    ... WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
    ... ELSE "BalDueDeb"
    ... END
    ... END), 0.00) AS "120+ days"

FROM
  ... JDT1 INNER JOIN OCRD
  ... ON JDT1."ShortName" = OCRD."CardCode"
  ... AND OCRD."CardType" = 'c'

GROUP BY
  ... OCRD."CardCode", OCRD."CardName"
HAVING SUM(
  ... CASE
  ... WHEN "SYSCred" ≠ 0 THEN "SYSCred" * -1
  ... ELSE "SYSDeb"
  ... END) ≠ 0

ORDER BY
  ... OCRD."CardCode";
```

Step 6 - pull out CASE-WHEN-ELSE credit-debit-balance into function

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then :CREDITS * -1.0
        else :DEBITS
        end;
end;

SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , SUM(balance_from_credits_debits(JDT1."SYS Cred", JDT1."SYS Deb")) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
            balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "31 to 60 days"
```

```
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "61 to 90 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "91 to 120 days"
    , IFNULL(SUM(
        CASE
            WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
                CASE
                    WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
                    ELSE "BalDueDeb"
                END
            END), 0.00) AS "120+ days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'

    GROUP BY
        OCRD."CardCode", OCRD."CardName"
    HAVING
        SUM(balance_from_credits_debits(JDT1."SYS Cred", JDT1."SYS Deb")) ≠ 0
    ORDER BY
        OCRD."CardCode";
```

Step 7 - pull out WITH clause for customer_debits_credits, DUE DAYS calculation

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6))
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then :CREDITS * -1.0
        else :DEBITS
        end;
end;

SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , SUM(balance_from_credits_debits(JDT1."SYS Cred", JDT1."SYS Deb")) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 0 THEN
            balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 0 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 30) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 30 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 60) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "31 to 60 days"
```

```
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 60 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 90) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "61 to 90 days"
    , IFNULL(SUM(
        CASE
            WHEN (DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 90 AND
                DAYS_BETWEEN(JDT1."DueDate", current_timestamp) < 120) THEN
                balance_from_credits_debits(JDT1."BalDueCred", JDT1."BalDueDeb")
            END), 0.00) AS "91 to 120 days"
    , IFNULL(SUM(
        CASE
            WHEN DAYS_BETWEEN(JDT1."DueDate", current_timestamp) ≥ 120 THEN
                CASE
                    WHEN "BalDueCred" ≠ 0 THEN "BalDueCred" * -1
                    ELSE "BalDueDeb"
                END
            END), 0.00) AS "120+ days"
FROM
    JDT1 INNER JOIN OCRD
    ON JDT1."ShortName" = OCRD."CardCode"
    AND OCRD."CardType" = 'c'

GROUP BY
    OCRD."CardCode", OCRD."CardName"
HAVING
    SUM(balance_from_credits_debits(JDT1."SYS Cred", JDT1."SYS Deb")) ≠ 0
ORDER BY
    OCRD."CardCode";
```

Step 7 - pull out WITH clause for customer_debits_credits, DUE DAYS calculation

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then :CREDITS * -1.0
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
    ON JDT1."ShortName" = OCRD."CardCode"
    AND OCRD."CardType" = 'c'
)
```

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) AS "Balance Due"
    , IFNULL(SUM(
        CASE
            WHEN "Due Days" < 0
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 0 AND "Due Days" < 30)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 30 AND "Due Days" < 60)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "31 to 60 days"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 60 AND "Due Days" < 90)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "61 to 90 days"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 90 AND "Due Days" < 120)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "91 to 120 days"
    , IFNULL(SUM(
        CASE
            WHEN "Due Days" ≥ 120
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "120+ days"
    FROM
        _customer_debits_credits
    GROUP BY
        "Customer Code"
        , "Customer Name"
    HAVING
        SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) ≠ 0
    ORDER BY
        "Customer Code";
```

Step 8 - eliminate IFNULL by ELSE clause

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then :CREDITS * -1.0
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)

```

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) AS "Balance Due"
    , IFNULL(SUM(CASE
        WHEN "Due Days" < 0
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END), 0.00) AS "Future Remit"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 0 AND "Due Days" < 30)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "0-30 days"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 30 AND "Due Days" < 60)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "31 to 60 days"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 60 AND "Due Days" < 90)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "61 to 90 days"
    , IFNULL(SUM(
        CASE
            WHEN ("Due Days" ≥ 90 AND "Due Days" < 120)
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "91 to 120 days"
    , IFNULL(SUM(
        CASE
            WHEN "Due Days" ≥ 120
            THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
            END), 0.00) AS "120+ days"
FROM
    _customer_debits_credits
GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) ≠ 0
ORDER BY
    "Customer Code";
```

Step 8 - eliminate IFNULL by ELSE clause by nothing

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then :CREDITS * -1.0
        else :DEBITS
        end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    (SELECT
        , OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
    )
)
```

```
SELECT
    , "Customer Code"
    , "Customer Name"
    , SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) AS "Balance Due"
    , SUM(CASE
        WHEN "Due Days" < 0
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        ELSE 0.00
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" ≥ 0 AND "Due Days" < 30)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" ≥ 30 AND "Due Days" < 60)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" ≥ 60 AND "Due Days" < 90)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" ≥ 90 AND "Due Days" < 120)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN "Due Days" ≥ 120
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "120+ days"
FROM
    _customer_debits_credits

GROUP BY
    , "Customer Code"
    , "Customer Name"
HAVING
    SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) ≠ 0
ORDER BY
    , "Customer Code";
```

Step 9 – change $x \geq Y$ and $x < Z$ to x between $Y+1$ and Z

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS  $\neq$  0.0
        then :CREDITS * -1.0
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)

```

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) AS "Balance Due"
    , SUM(CASE
        WHEN "Due Days" < 0
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        ELSE 0.00
    END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days"  $\geq$  0 AND "Due Days" < 30)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days"  $\geq$  30 AND "Due Days" < 60)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days"  $\geq$  60 AND "Due Days" < 90)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days"  $\geq$  90 AND "Due Days" < 120)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN "Due Days"  $\geq$  120
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "120+ days"
FROM
    _customer_debits_credits

GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM( balance_from_credits_debits("SYSCred", "SYSDeb") )  $\neq$  0
ORDER BY
    "Customer Code";
```

Step 9 – change $x \geq Y$ and $x < Z$ to x between $Y+1$ and Z

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6))
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS  $\neq$  0.0
        then TO_DECIMAL (:CREDITS * -1.0, 19, 6)
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)

```

Careful here!

At this point, the groupings are changed and that changes the results per column.
Direct comparison of results with the original query does not work from here on.

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM(balance_from_credits_debits("SYSCred", "SYSDeb")) AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "121+ days"
    FROM
        _customer_debits_credits

GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM(balance_from_credits_debits("SYSCred", "SYSDeb"))  $\neq$  0
ORDER BY
    "Customer Code";

```

Step 10 – change current_timestamp to current_date

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6))
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then TO_DECIMAL (:CREDITS * -1.0, 19, 6)
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)

```

DAYS_BETWEEN works on timestamps (not dates) and counts how often a day's worth of seconds goes into the difference between the inputs.

This can lead to counterintuitive results when calendar days are expected to be used.

<https://lbreddemann.org/the-time-is-now-isnt-it/>

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "121+ days"
FROM
    _customer_debits_credits

GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) ≠ 0
ORDER BY
    "Customer Code";

```

Step 10 – change current_timestamp to current_date

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6))
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then TO_DECIMAL (:CREDITS * -1.0, 19, 6)
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)

```

DAYS_BETWEEN works on timestamps (not dates) and counts how often a day's worth of seconds goes into the difference between the inputs.

This can lead to counterintuitive results when calendar days are expected to be used.

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM(balance_from_credits_debits("SYSCred", "SYSDeb")) AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "121+ days"
FROM
    _customer_debits_credits
GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM(balance_from_credits_debits("SYSCred", "SYSDeb")) ≠ 0
ORDER BY
    "Customer Code";

```

<https://lbreddemann.org/the-time-is-now-isnt-it/>

Step 11 - pull customer debit function up, _customer_debits_credits → _customer_credits_debits

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then TO_DECIMAL (:CREDITS * -1.0, 19, 6)
        else :DEBITS
    end;
end;

WITH _customer_debits_credits
("Customer Code", "Customer Name", "SYSCred", "SYSDeb", "BalDueCred", "BalDueDeb", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , JDT1."SYSCred"
        , JDT1."SYSDeb"
        , JDT1."BalDueCred"
        , JDT1."BalDueDeb"
        , DAYS_BETWEEN(JDT1."DueDate", current_timestamp) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)

```

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN balance_from_credits_debits("BalDueCred", "BalDueDeb")
        END) AS "121+ days"
FROM
    _customer_debits_credits

GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM( balance_from_credits_debits("SYSCred", "SYSDeb") ) ≠ 0
ORDER BY
    "Customer Code";
```

Step 11 – pull customer debit function up, _customer_debits_credits → _customer_credits_debits

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6)
)
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then TO_DECIMAL (:CREDITS * -1.0, 19, 6)
        else :DEBITS
    end;
end;

WITH _customer_credits_debits
("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , balance_from_credits_debits("SYSCred", "SYSDeb") as "SYS Balance Due"
        , balance_from_credits_debits("BalDueCred", "BalDueDeb") as "Cust Balance Due"
        , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)
```

Renaming the CTE (common table expression) here is just to match the order of parameters in the function call.
It's CREDITS first and then DEBITS.

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM("SYS Balance Due") AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN "Cust Balance Due"
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN "Cust Balance Due"
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN "Cust Balance Due"
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN "Cust Balance Due"
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN "Cust Balance Due"
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN "Cust Balance Due"
        END) AS "121+ days"
FROM
    _customer_credits_debits
GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM("SYS Balance Due") ≠ 0
ORDER BY
    "Customer Code";
```

Step 12 - remove UDF call with SQL expression right in the CTE

```
create function balance_from_credits_debits
(
    IN CREDITS DECIMAL(19,6)
    , IN DEBITS DECIMAL(19,6))
returns balance DECIMAL(19,6)
as
begin
    balance = case
        when :CREDITS ≠ 0.0
        then TO_DECIMAL(:CREDITS * -1.0, 19, 6)
        else :DEBITS
    end;
end;

WITH _customer_credits_debits
("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS
(
    SELECT
        OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , balance_from_credits_debits("SYSCred", "SYSDeb") as "SYS Balance Due"
        , balance_from_credits_debits("BalDueCred", "BalDueDeb") as "Cust Balance Due"
        , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
    FROM
        JDT1 INNER JOIN OCRD
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
)
```

Checking the runtime with the UDF (user defined function) it turned out to be slower. Depending on the data volume that might be OK, but let's try for a faster query.

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM("SYS Balance Due") AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN "Cust Balance Due"
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN "Cust Balance Due"
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN "Cust Balance Due"
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN "Cust Balance Due"
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN "Cust Balance Due"
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN "Cust Balance Due"
        END) AS "121+ days"
FROM
    _customer_credits_debits
GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM("SYS Balance Due") ≠ 0
ORDER BY
    "Customer Code";
```

Step 12 – remove UDF call with SQL expression right in the CTE

```
WITH _customer_credits_debits
("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS
(SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , TO_DECIMAL(
        IFNULL(case
            when "SYSCred" ≠ 0.0
            then "SYSCred" * -1.0
            else "SYSDeb"
        end
        , 0.00)
    , 19, 6) as "SYS Balance Due"
    , TO_DECIMAL(
        IFNULL(case
            when "BalDueCred" ≠ 0.0
            then "BalDueCred" * -1.0
            else "BalDueDeb"
        end
        , 0.00)
    , 19, 6) as "Cust Balance Due"
    , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
FROM
    OCRD INNER JOIN JDT1
ON JDT1."ShortName" = OCRD."CardCode"
AND OCRD."CardType" = 'c'
)
```

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM("SYS Balance Due") AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN "Cust Balance Due"
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN "Cust Balance Due"
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN "Cust Balance Due"
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN "Cust Balance Due"
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN "Cust Balance Due"
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN "Cust Balance Due"
        END) AS "121+ days"
FROM
    _customer_credits_debits
GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM("SYS Balance Due") ≠ 0
ORDER BY
    "Customer Code";
```

Step 13 – fix type conversion from CASE expressions

```
WITH _customer_credits_debits
("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS
(SELECT
  OCRD."CardCode" AS "Customer Code"
  , OCRD."CardName" AS "Customer Name"
  , TO_DECIMAL(
    IFNULL(case
      when "SYSCred" <> 0.0
      then "SYSCred" * -1.0
      else "SYSDeb"
    end
    , 0.00)
  , 19, 6) as "SYS Balance Due"
  , TO_DECIMAL(
    IFNULL(case
      when "BalDueCred" <> 0.0
      then "BalDueCred" * -1.0
      else "BalDueDeb"
    end
    , 0.00)
  , 19, 6) as "Cust Balance Due"
  , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
FROM
  OCRD INNER JOIN JDT1
ON JDT1."ShortName" = OCRD."CardCode"
AND OCRD."CardType" = 'c'
)
```

In EXPLAIN PLAN another automatic type cast occurs:

```
TO_DECIMAL(IFNULL(
  (CASE WHEN JDT1.BalDueCred <> 0.0
    THEN JDT1.BalDueCred * -1.0
    ELSE TO_DECIMAL(JDT1.BalDueDeb, 22, 7)
  END, ...
```

```
SELECT
  "Customer Code"
  , "Customer Name"
  , SUM("SYS Balance Due") AS "Balance Due"
  , SUM(CASE
    WHEN ("Due Days" < 0)
    THEN "Cust Balance Due"
    END) AS "Future Remit"
  , SUM(CASE
    WHEN ("Due Days" BETWEEN 0 AND 30)
    THEN "Cust Balance Due"
    END) AS "0-30 days"
  , SUM(CASE
    WHEN ("Due Days" BETWEEN 31 AND 60)
    THEN "Cust Balance Due"
    END) AS "31 to 60 days"
  , SUM(CASE
    WHEN ("Due Days" BETWEEN 61 AND 90)
    THEN "Cust Balance Due"
    END) AS "61 to 90 days"
  , SUM(CASE
    WHEN ("Due Days" BETWEEN 91 AND 120)
    THEN "Cust Balance Due"
    END) AS "91 to 120 days"
  , SUM(CASE
    WHEN ("Due Days" > 121)
    THEN "Cust Balance Due"
    END) AS "121+ days"
FROM
  _customer_credits_debits
GROUP BY
  "Customer Code"
  , "Customer Name"
HAVING
  SUM("SYS Balance Due") <> 0
ORDER BY
  "Customer Code";
```

Step 13 – fix type conversion from CASE expressions

```
WITH _customer_credits_debits
("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS
(
    (SELECT
        , OCRD."CardCode" AS "Customer Code"
        , OCRD."CardName" AS "Customer Name"
        , (case
            when "SYSCred" ≠ 0.0
            then TO_DECIMAL("SYSCred" * -1.0, 19, 6)
            else TO_DECIMAL("SYSDeb", 19, 6)
            end) as "SYS Balance Due"
        , (case
            when "BalDueCred" ≠ 0.0
            then TO_DECIMAL("BalDueCred" * -1.0, 19, 6)
            else TO_DECIMAL("BalDueDeb", 19, 6)
            end) as "Cust Balance Due"
        , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
    FROM
        , OCRD INNER JOIN JDT1
        ON JDT1."ShortName" = OCRD."CardCode"
        AND OCRD."CardType" = 'c'
    )
)
```

```
SELECT
    , "Customer Code"
    , "Customer Name"
    , SUM("SYS Balance Due") AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN "Cust Balance Due"
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN "Cust Balance Due"
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN "Cust Balance Due"
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN "Cust Balance Due"
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN "Cust Balance Due"
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN "Cust Balance Due"
        END) AS "121+ days"
FROM
    _customer_credits_debits
GROUP BY
    , "Customer Code"
    , "Customer Name"
HAVING
    SUM("SYS Balance Due") ≠ 0
ORDER BY
    "Customer Code";
```

STOP!

```
WITH _customer_credits_debits
("Customer Code", "Customer Name", "SYS Balance Due", "Cust Balance Due", "Due Days") AS
(SELECT
    OCRD."CardCode" AS "Customer Code"
    , OCRD."CardName" AS "Customer Name"
    , (case
        when "SYSCred" <> 0.0
        then TO_DECIMAL("SYSCred" * -1.0, 19, 6)
        else TO_DECIMAL("SYSDeb", 19, 6)
        end) as "SYS Balance Due"
    , (case
        when "BalDueCred" <> 0.0
        then TO_DECIMAL("BalDueCred" * -1.0, 19, 6)
        else TO_DECIMAL("BalDueDeb", 19, 6)
        end) as "Cust Balance Due"
    , DAYS_BETWEEN(JDT1."DueDate", current_date) AS "Due Days"
FROM
    OCRD INNER JOIN JDT1
ON JDT1."ShortName" = OCRD."CardCode"
AND OCRD."CardType" = 'c'
)
```

Time to have a look at the statement again.

What does it do?

Note that there are in fact two aggregation levels:

- 1) Based on the full "SYS Balance Due" per customer ("Cust. Code" and "Cust. Name" always build the same group)
- 2) Based on the time slots.

```
SELECT
    "Customer Code"
    , "Customer Name"
    , SUM("SYS Balance Due") AS "Balance Due"
    , SUM(CASE
        WHEN ("Due Days" < 0)
        THEN "Cust Balance Due"
        END) AS "Future Remit"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 0 AND 30)
        THEN "Cust Balance Due"
        END) AS "0-30 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 31 AND 60)
        THEN "Cust Balance Due"
        END) AS "31 to 60 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 61 AND 90)
        THEN "Cust Balance Due"
        END) AS "61 to 90 days"
    , SUM(CASE
        WHEN ("Due Days" BETWEEN 91 AND 120)
        THEN "Cust Balance Due"
        END) AS "91 to 120 days"
    , SUM(CASE
        WHEN ("Due Days" > 121)
        THEN "Cust Balance Due"
        END) AS "121+ days"
FROM
    _customer_credits_debits
GROUP BY
    "Customer Code"
    , "Customer Name"
HAVING
    SUM("SYS Balance Due") ≠ 0
ORDER BY
    "Customer Code";
```

STOP!

Original query result

Customer Code	Customer Name	Balance Due	Future Remit	0-30 days	31 to 60 days	61 to 90 days	91 to 120 days	120+ days
CC0001	Robel, Hermiston and Smith	-627931.508	-292253.3320	-43440.598	-33361.374000	-36461.363000	-37044.144000	-182889.416
CC0007	Hand-Murazik	-632987.217	-290324.8610	-41162.541	-37237.034000	-38708.927000	-41575.572000	-182358.897
CC0013	Beatty, Kris and Wolff	-620051.886	-276851.2240	-44209.015	-30972.273000	-39446.000000	-36691.526000	-185799.788
CC0019	Leffler-Koch	-635646.453	-285973.0680	-35738.858	-34188.120000	-39915.037000	-39743.413000	-186057.643
CC0020	O'Conner, Swaniawski and Rogahn	-622834.399	-289215.2410	-33202.611	-38646.829000	-38930.113000	-37384.327000	-184100.311
CC0022	Brakus-Braun	-620072.744	-293423.0760	-40851.999	-39740.231000	-42606.646000	-34726.326000	-170138.382
CC0026	Kautzer, Wolf and Conn	-624792.548	-285367.4400	-41901.141	-38295.999000	-39945.978000	-34297.624000	-180138.311
CC0043	Collier-Haley	-629535.548	-300841.4300	-35856.967	-39892.859000	-34951.770000	-35664.465000	-169146.612
CC0045	Hodkiewicz, Cummerata and Will	-622480.056	-308358.7810	-43362.080	-39065.686000	-31912.466000	-39727.909000	-168080.452
CC0067	Eichmann, Mann and Bins	-639821.278	-304936.9680	-35080.969	-39500.653000	-31518.377000	-33243.902000	-181726.008
[...]								

Refactored query result

Customer Code	Customer Name	Balance Due	Future Remit	0-30 days	31 to 60 days	61 to 90 days	91 to 120 days	121+ days
CC0001	Robel, Hermiston and Smith	-627931.5080	-294650.9240	-43278.701	-33306.089000	-35092.160000	-36947.865000	-180389.534
CC0007	Hand-Murazik	-632987.2170	-291009.0010	-41292.249	-37292.444000	-39640.456000	-41434.543000	-179722.923
CC0013	Beatty, Kris and Wolff	-620051.8860	-278176.1200	-44524.200	-30563.171000	-38893.552000	-36694.032000	-183180.450
CC0019	Leffler-Koch	-635646.4530	-287049.8810	-35757.327	-35088.686000	-38701.060000	-40181.777000	-183672.690
CC0020	O'Conner, Swaniawski and Rogahn	-622834.3990	-289982.7740	-33637.519	-38596.452000	-39262.480000	-36788.164000	-182935.555
CC0022	Brakus-Braun	-620072.7440	-295057.5930	-40550.984	-39128.104000	-42858.681000	-35183.487000	-167773.835
CC0026	Kautzer, Wolf and Conn	-624792.5480	-285923.0200	-42298.153	-38684.974000	-38929.167000	-35894.637000	-176688.802
CC0043	Collier-Haley	-629535.5480	-301422.0290	-35825.168	-39473.902000	-37462.314000	-33455.688000	-167871.078
CC0045	Hodkiewicz, Cummerata and Will	-622480.0560	-309145.9970	-43916.288	-39318.572000	-32411.329000	-38930.249000	-165828.971
CC0067	Eichmann, Mann and Bins	-639821.2780	-306494.0610	-34437.052	-39837.614000	-31280.390000	-33656.206000	-179099.721
[...]								

What causes the differences?

- Change of time-slot ranges
- Change to use `current_date` instead of `current_timestamp`

Needs to be confirmed with users !

Further options?

- Put mapping of ranges to text into a helper table and join it, allowing to be "flexible"
- Don't calculate both aggregation levels in statement and don't perform the pivot. Let the UI do the formatting and simple aggregation:

Customer Code	Customer Name	Due Range	Balance Due	Customer Balance Due
CC0001	Robel, Hermiston and Smith	0-30 days	-38238.12400	-37708.0520000
CC0001	Robel, Hermiston and Smith	121+ days	-163492.2860	-169697.4130000
CC0001	Robel, Hermiston and Smith	31 to 60 days	-41199.45300	-35254.8160000
CC0001	Robel, Hermiston and Smith	61 to 90 days	-35146.90600	-38366.9070000
CC0001	Robel, Hermiston and Smith	91 to 120 days	-38980.91300	-35379.9750000
CC0001	Robel, Hermiston and Smith	Future Remit	-310873.8260	-309043.0640000
CC0007	Hand-Murazik	0-30 days	-41570.45400	-39600.3580000
CC0007	Hand-Murazik	121+ days	-168389.7590	-166371.3600000
[...]				

Summed up this is the
"Balance Due" per
customer from the
original query

The finale

- Refactor for understanding
- Use good tools
- Understand what the query is about
- Try make "simple" queries about one thing (concept)
- Avoid being "flexible" for the sake of it - SQL is statically typed
 - Your "general" solution likely is just generic
 - If you absolutely need to be flexible, put the flexibility into your application code and generate the right queries from there
- Try to not get stuck on "Getting Stuff Done™"-level; use the existing material (books/blogs/comics) to learn solutions to common problems
- Read other people's SQL (user forums, open source)

Ride into sunset



Code examples can be found at https://github.com/LarsBr/sitMel2020_refactor



Data Process Insights

Lars Breddemann

[@LBREDDemann](#)

<https://lbreddemann.org/>

<https://dataprocessinsights.com.au/>

lars@dataprocessinsights.com.au