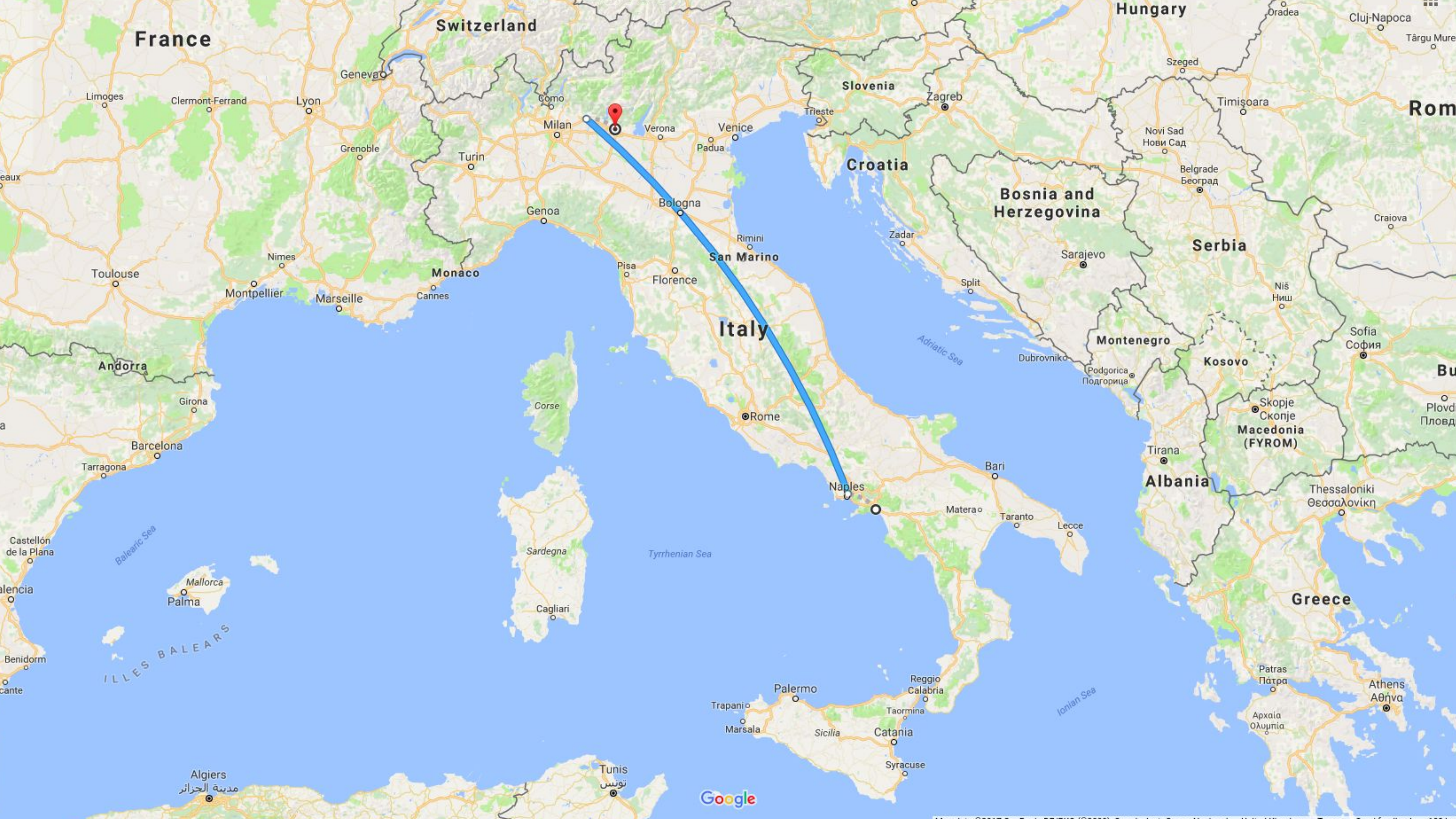
zalando

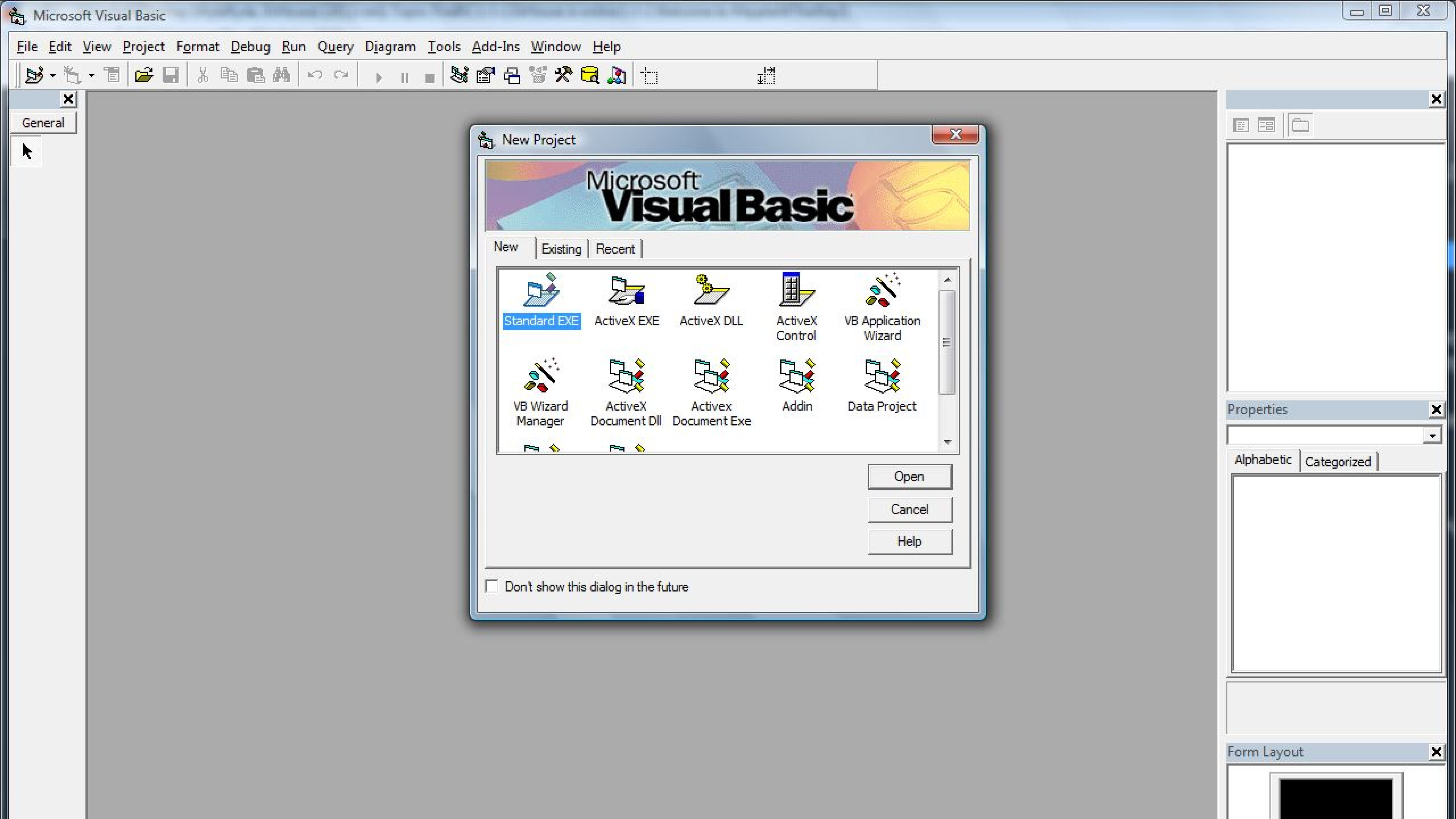
SQL DEVELOPERS: WHAT DO THEY WRITE? DO THEY WRITE GOOD SQL?? LET'S FIND OUT!

FRANCESCO MUCIO

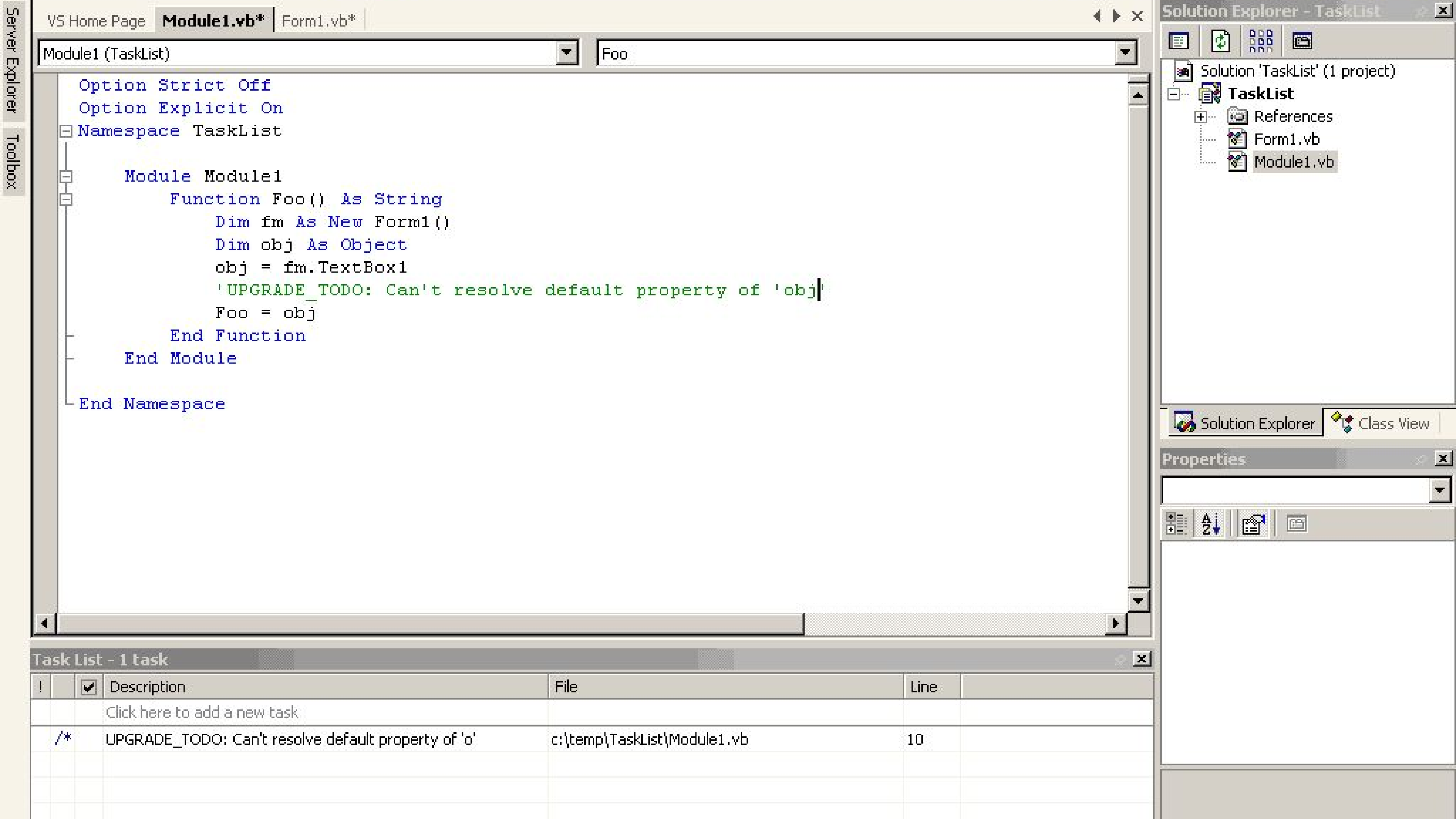
26-10-2017

WHO AM I?


France
Switzerland
Hungary
Slovenia
Croatia
Bosnia and Herzegovina
Serbia
Montenegro
Kosovo
Macedonia (FYROM)
Albania
Greece
Italy
San Marino
Monaco
Andorra
Milan
Verona
Venice
Padua
Bologna
Florence
Rome
Naples
Bari
Genoa
Turin
Geneva
Lyon
Marseille
Toulouse
Barcelona
Tyrrhenian Sea
Adriatic Sea
Ionian Sea
Balearic Sea
Corse
Sardegna
Sicilia
Palermo
Catania
Tunis
Algiers
Athens
Sofia
Belgrade
Sarajevo
Zagreb
Tirana
Skopje
Google


Microsoft Visual Basic
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
General
New Project
Microsoft Visual Basic
New
Existing
Recent
Standard EXE
ActiveX EXE
ActiveX DLL
ActiveX Control
VB Application Wizard
VB Wizard Manager
ActiveX Document Dll
Activex Document Exe
Addin
Data Project
Open
Cancel
Help
Don't show this dialog in the future
Properties
Alphabetic
Categorized
Form Layout

Melangerie
Röstkaffee in
ganzen Bohnen
Lidl Stiftung & Co. KG

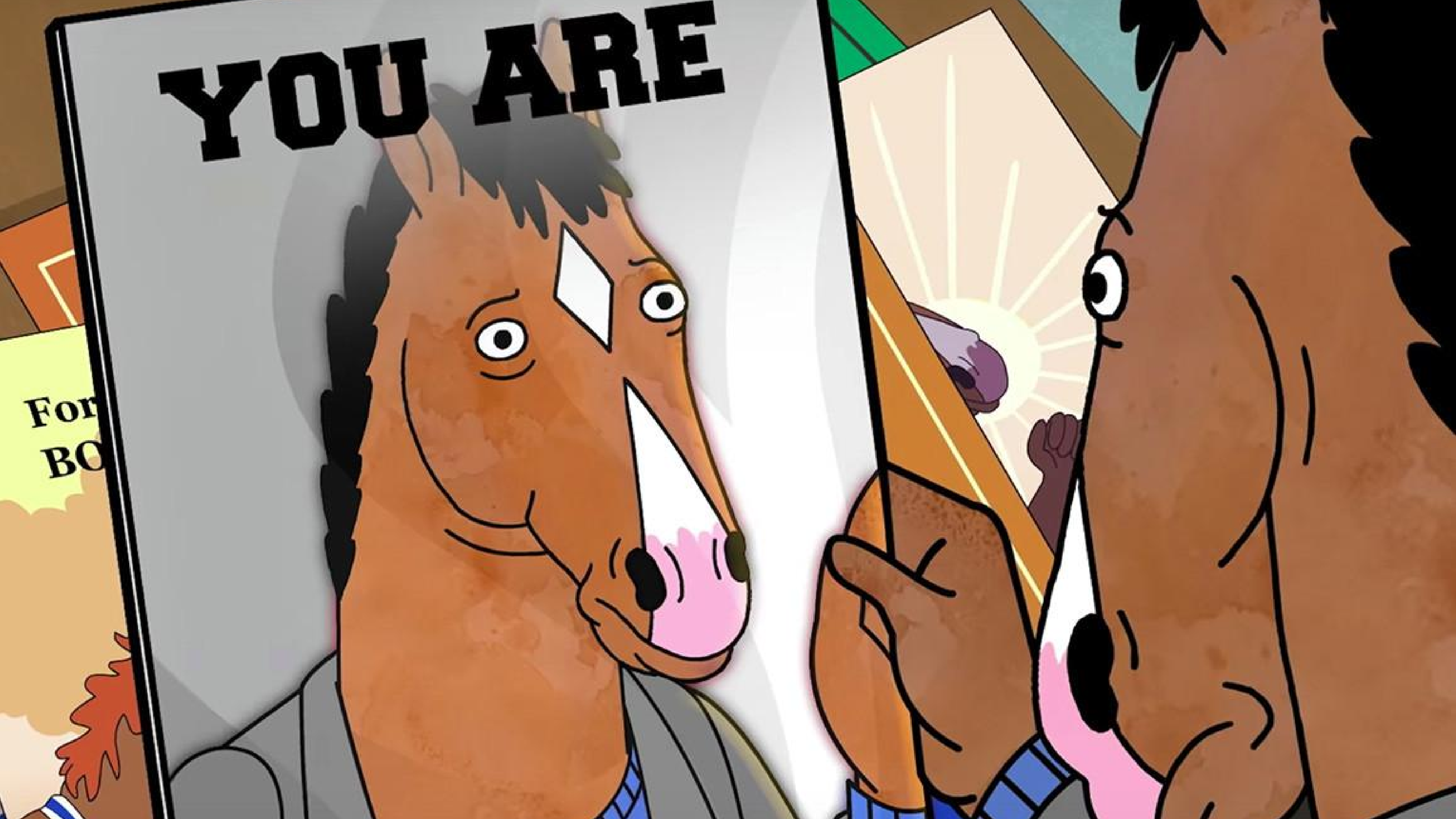


```
Option Strict Off
Option Explicit On
Namespace TaskList

    Module Module1
        Function Foo() As String
            Dim fm As New Form1()
            Dim obj As Object
            obj = fm.TextBox1
            'UPGRADE_TODO: Can't resolve default property of 'obj'
            Foo = obj
        End Function
    End Module

End Namespace
```

Task List - 1 task

!		✓	Description	File	Line
			Click here to add a new task		
	/*		UPGRADE_TODO: Can't resolve default property of 'o'	c:\temp\TaskList\Module1.vb	10

Yale Edu


YOU ARE
For
BO

JD SALINGER

PRESENTS

HOLLYWOO

STARS AND CELEBRITIES

WHAT DO THEY KNOW?

DO THEY KNOW THINGS??

LET'S FIND OUT!

WHAT IS NOT CODE IN YOUR CODE?

zalando

COMMENTS

```
-- Fool me once.
SELECT col1
  FROM my_table;
```

zalando

COMMENTS

```
-- Fool me once.
SELECT col1
  FROM my_table;

/* Fool me twice.
Fool me chicken soup with rice.
*/
SELECT col1
  FROM my_table;
```

zalando

**BEING ANTISOCIAL IS NOT A GOOD EXCUSE
TO AVOID COMMENTING.
ACTUALLY IT IS QUITE THE OPPOSITE.**

COMMENTS BAD IDEAS

```
/* Extract all distinct cities
   from the weather table.
   Sorted.
*/
    SELECT DISTINCT city -- avoid duplication
      FROM weather
  ORDER BY city; -- sorted by city
```

SUBTITLES FOR YOUR CODE

zalando

COMMENTS BAD IDEAS

```
/* author: Bojack Horseman
    email: SecretariatRulez96@hotmail.com
  created: 2013-02-20
   reason: Extract all the cities
   -----------
  changed: 2015-11-21
   author: Todd Chavez
   reason: Removed DISTINCT
   -----------
  changed: 2015-11-22
   author: Todd Chavez
   reason: Added back DISTINCT, I'm sorry Bojack
*/
    SELECT DISTINCT city
      FROM weather
  ORDER BY city;
```

YOUR SOURCE CONTROL SYSTEM

zalando

COMMENTS BAD IDEAS

```
/*--------------------------------------------*\
|   author: Bojack Horseman                    |
|    email: SecretariatRulez96@hotmail.com     |
|  created: 2013-02-20                         |
|   reason: Extract all the cities             |
\*--------------------------------------------*/
   SELECT DISTINCT city
     FROM weather
 ORDER BY city;
```

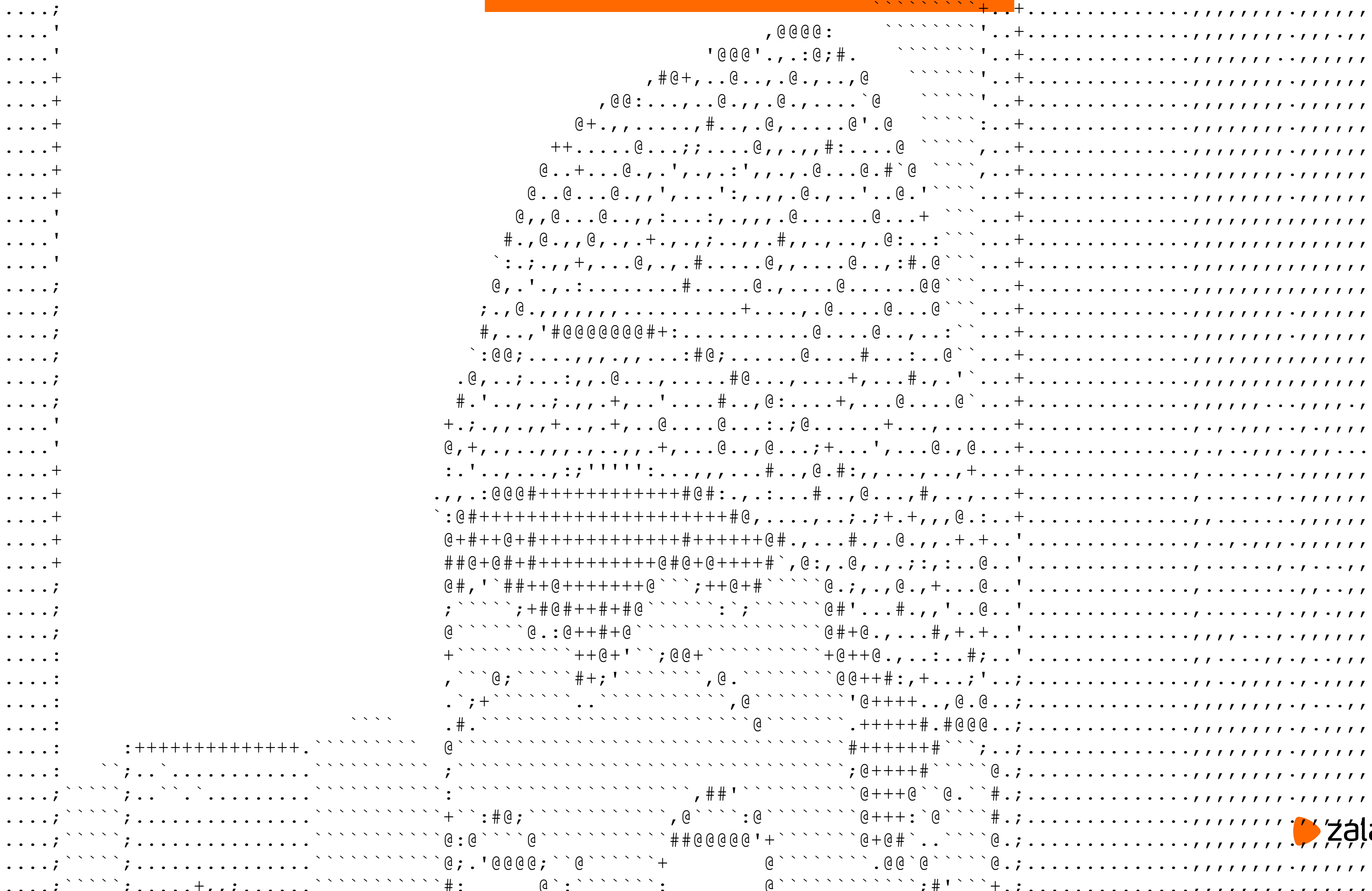
A WAY TO EXPRESS YOUR INNER ARTIST

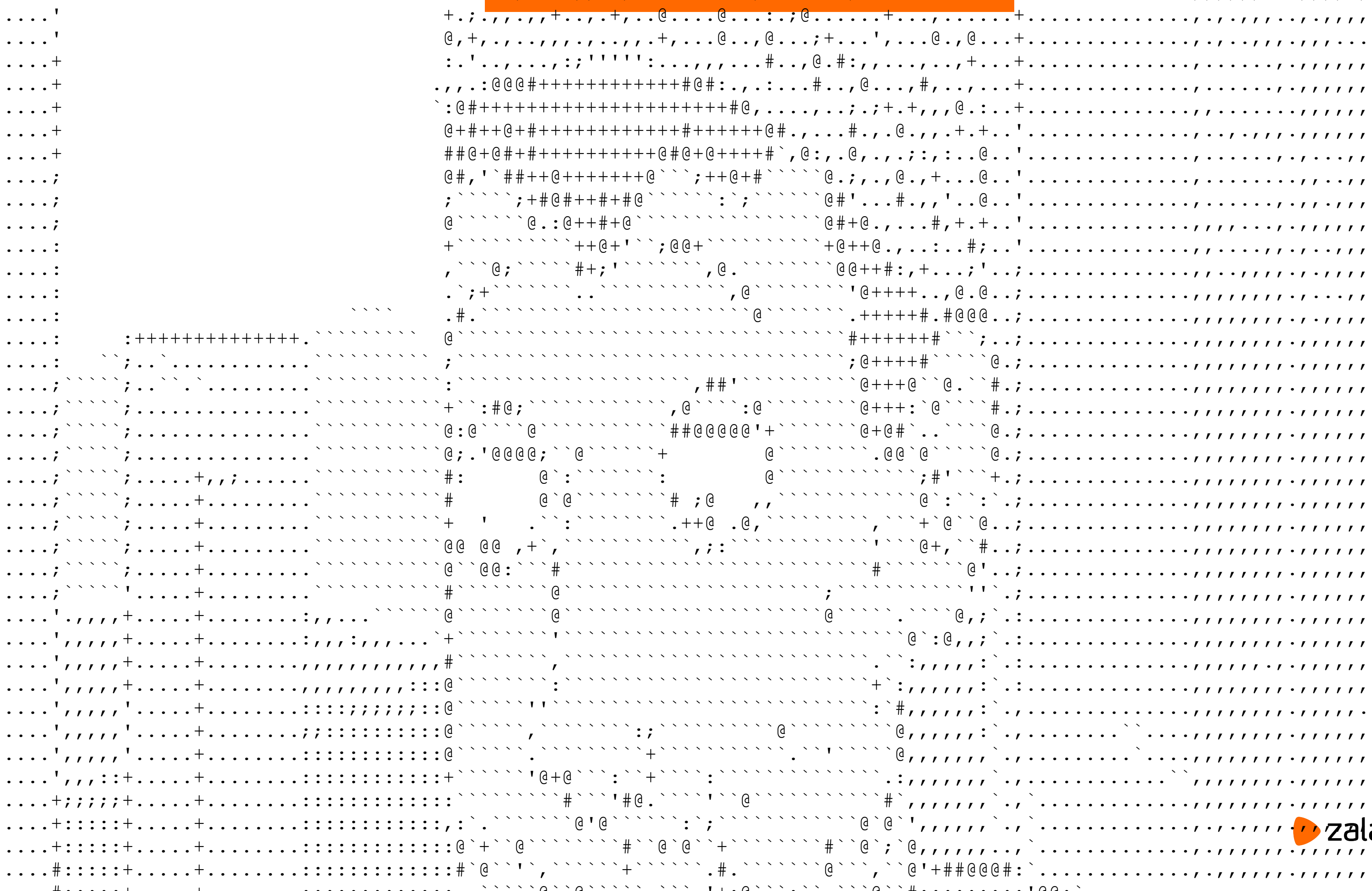
zalando

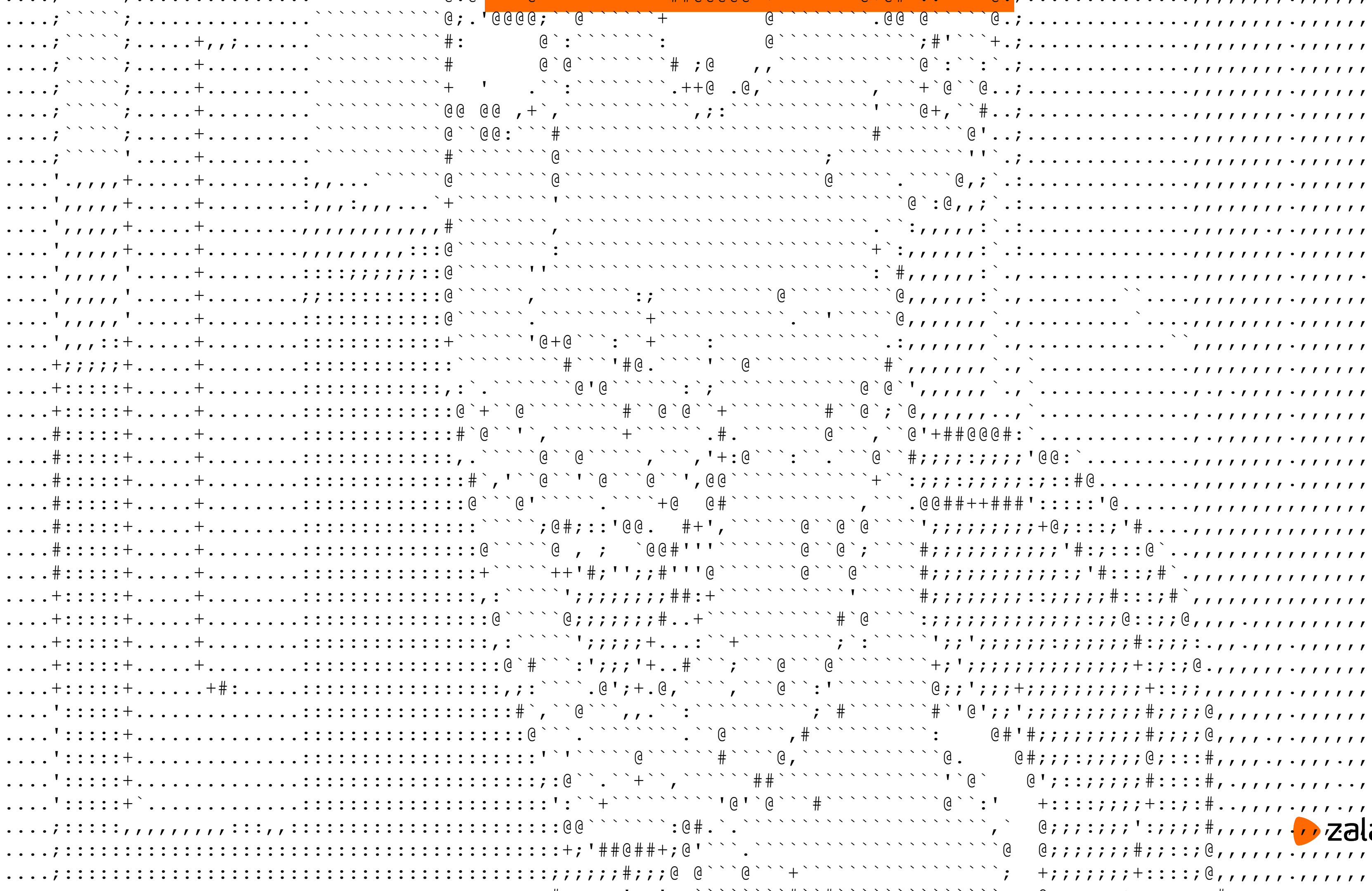
COMMENTS BAD IDEAS

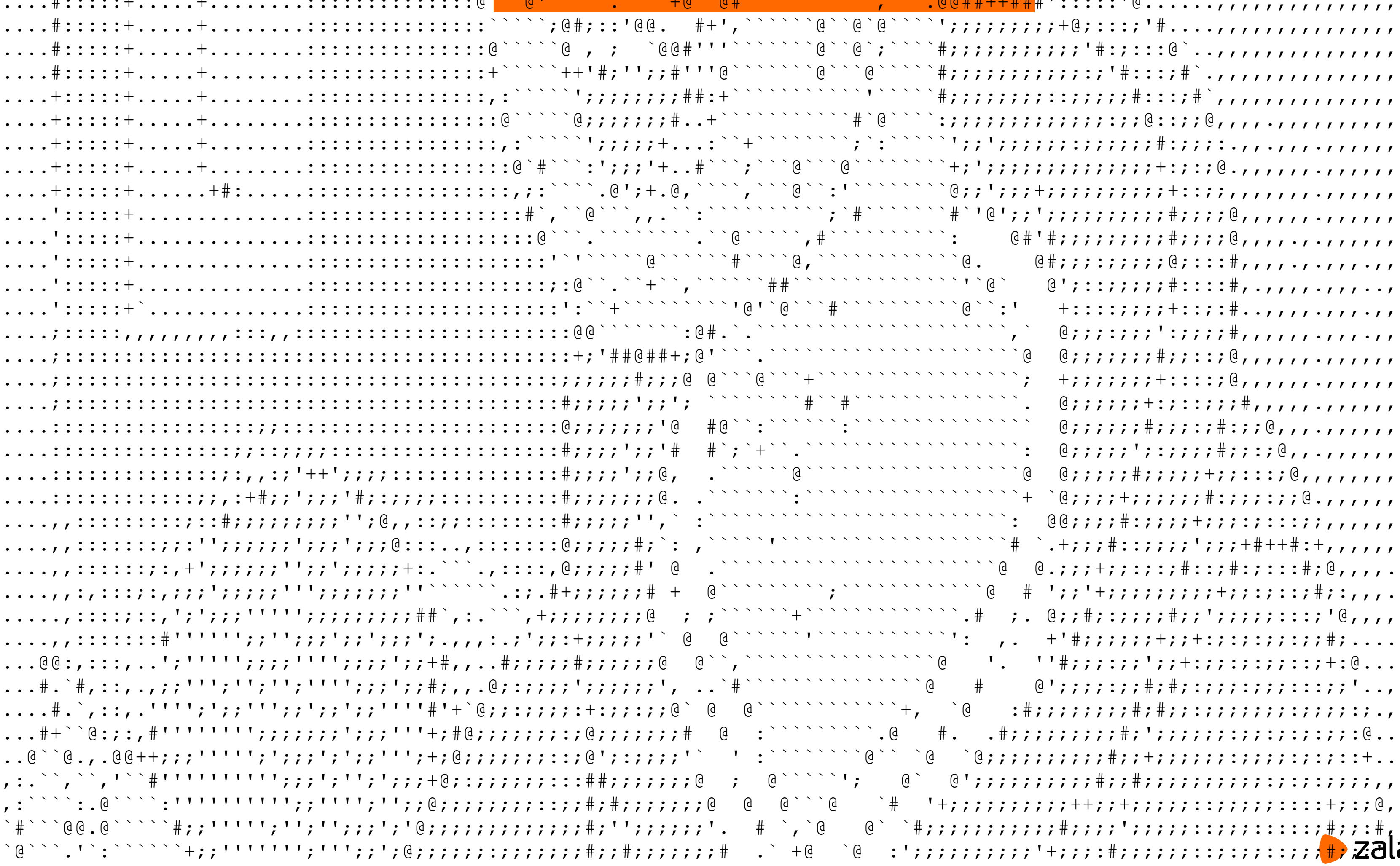
```
/*              __
               / _)   This query is sooo old
        .-^^^-/ /
     __/       /
    <__.|_|-|_|
*/
    SELECT DISTINCT city
      FROM weather
  ORDER BY city;
```

A WAY TO EXPRESS YOUR INNER ARTIST









zalando

COMMENTS BAD IDEAS

```
    SELECT DISTINCT city
      FROM weather
     WHERE country = ‘US’
-- keeping this in case we need it in the future
--   WHERE country <> ‘US’
  ORDER BY city;
```

BE A CODE HOARDER

zalando

COMMENTS GOOD PRACTICES

- COMMENTS ARE TO DESCRIBE WHY NOT HOW
- CONSIDER YOUR COMMENTS AS CODE SMELL
- IF YOU COMMENT, TRY TO EXTRACT YOUR CODE

QUICK TIP I

USE STANDARD SQL

COALESCE (NOT NVL OR ISNULL)

SUBSTR (NOT LEFT OR RIGHT)

CURRENT_TIMESTAMP (NOT NOW)

zalando

CODE FORMATTING

- INDENT YOUR CODE
- FOLLOW A STANDARD
- INDENT YOUR CODE
- USE PROPER ALIASES
- INDENT YOUR CODE

zalando

INDENT YOUR CODE

```
SELECT  a.newsID, a.title, a.clicked, a.newsDate, c.sectionName, a.sectionID
FROM News a INNER JOIN newsSection c ON a.sectionID = c.SectionID WHERE (c.SectionID
= 21)
GROUP BY c.sectionName, a.newsID, a.title, a.clicked, a.newsDate, a.sectionID ORDER
BY a.newsDate DESC
```

zalando

INDENT YOUR CODE

```
SELECT a.newsID,
       a.title,
       a.clicked,
       a.newsDate,
       c.sectionName,
       a.sectionID
FROM News a
INNER JOIN newsSection c ON a.sectionID = c.SectionID
WHERE c.SectionID = 21
GROUP BY c.sectionName,
         a.newsID,
         a.title,
         a.clicked,
         a.newsDate,
         a.sectionID
ORDER BY a.newsDate DESC
```

zalando

INDENT YOUR CODE

```
SELECT a.newsID,
       a.title,
       a.clicked,
       a.newsDate,
       c.sectionName,
       a.sectionID
FROM News a
     INNER JOIN newsSection c ON a.sectionID = c.SectionID
WHERE c.SectionID = 21
GROUP BY c.sectionName,
         a.newsID,
         a.title,
         a.clicked,
         a.newsDate,
         a.sectionID
ORDER BY a.newsDate DESC
```

zalando

INDENT YOUR CODE

```
    SELECT a.newsID,
           a.title,
           a.clicked,
           a.newsDate,
           c.sectionName,
           a.sectionID
      FROM News a
INNER JOIN newsSection c
        ON a.sectionID = c.SectionID
     WHERE c.SectionID = 21
  GROUP BY c.sectionName,
           a.newsID,
           a.title,
           a.clicked,
           a.newsDate,
           a.sectionID
  ORDER BY a.newsDate DESC
```

INDENT YOUR CODE

```
SELECT a.newsID
       , a.title
       , a.clicked
       , a.newsDate
       , c.sectionName
       , a.sectionID
FROM News a
INNER JOIN newsSection c ON a.sectionID = c.SectionID
WHERE c.SectionID = 21
GROUP BY c.sectionName
         , a.newsID
         , a.title
         , a.clicked
         , a.newsDate
         , a.sectionID
ORDER BY a.newsDate DESC
```

zalando

INDENT YOUR CODE

LEARN YOUR IDE SHORTCUT TO INDENT YOUR CODE

FOLLOW A STANDARD

EXAMPLE OF CODE STANDARD:

- UPPER CASE FOR SQL KEYWORDS
- IDENTIFIERS IN LOWER CASE
- USE _ TO SEPARATE WORDS IN YOUR NAMES
- INNER JOIN > JOIN
- DO NOT USE HUNGARIAN NOTATION (T_, V_, PKG_, ...)

USE PROPER ALIASES

```
     SELECT a.newsID,
            a.title,
            a.clicked,
            a.newsDate,
            c.sectionName,
            a.sectionID
       FROM News a
 INNER JOIN newsSection c
         ON a.sectionID = c.SectionID
      WHERE c.SectionID = 21
   GROUP BY c.sectionName,
            a.newsID,
            a.title,
            a.clicked,
            a.newsDate,
            a.sectionID
   ORDER BY a.newsDate DESC
```

USE PROPER ALIASES

```
     SELECT n.newsID,
            n.title,
            n.clicked,
            n.newsDate,
            ns.sectionName,
            n.sectionID
       FROM News n
 INNER JOIN newsSection ns
         ON n.sectionID = ns.SectionID
      WHERE ns.SectionID = 21
   GROUP BY ns.sectionName,
            n.newsID,
            n.title,
            n.clicked,
            n.newsDate,
            n.sectionID
   ORDER BY n.newsDate DESC
```

zalando

USE PROPER ALIASES

```
    SELECT DISTINCT g.id,
                    g.description
      FROM gallery g
INNER JOIN gallery_to_tag g2t_0
        ON g2t_0.gallery_id = g.id
INNER JOIN tag t_0
        ON t_0.id = g2t_0.tag_id
INNER JOIN gallery_to_tag g2t_1
        ON g2t_1.gallery_id = g.id
INNER JOIN tag t_1
        ON t_1.id = g2t_1.tag_id
     WHERE t_0.term = 'hi'
       AND t_1.term = 'hey'
```

FROM STACKOVERFLOW

zalando

USE PROPER ALIASES

```
     SELECT DISTINCT g.id,
                     g.description
       FROM gallery g
-- tag 1
 INNER JOIN gallery_to_tag to_tag1
         ON g.id = to_tag1.gallery_id
 INNER JOIN tag tag1
         ON to_tag1.id = tag1.tag_id
-- tag 2
 INNER JOIN gallery_to_tag to_tag2
         ON g.id = to_tag2.gallery_id
 INNER JOIN tag tag2
         ON to_tag2.tag_id = tag2.id
      WHERE tag1.term = 'hi'
        AND tag2.term = 'hey'
```

FROM STACKOVERFLOW

zalando

USE PROPER ALIASES

```
SELECT DISTINCT g.id,
                g.description
           FROM gallery g
     -- tag 1
     INNER JOIN gallery_to_tag to_tag1
             ON g.id = to_tag1.gallery_id
     INNER JOIN tag tag1
             ON to_tag1.id = tag1.tag_id
     -- tag 2
     INNER JOIN gallery_to_tag to_tag2
             ON g.id = to_tag2.gallery_id
     INNER JOIN tag tag2
             ON to_tag2.tag_id = tag2.id
          WHERE tag1.term = 'hi'
            AND tag2.term = 'hey'
```

- **CHANGED THE ALIASES**
- **ORDERED THE JOIN CONDITIONS**
- **ADDED COMMENTS**

FROM STACKOVERFLOW

zalando

USE PROPER ALIASES

```
     SELECT g.id,
            g.description
       FROM gallery g
 INNER JOIN gallery_to_tag to_tag
         ON g.id = to_tag.gallery_id
 INNER JOIN tag
         ON tag.id = to_tag.tag_id
      WHERE tag.term IN ('hi', 'hey')
   GROUP BY g.id,
            g.description
     HAVING COUNT(1) = 2
```

FROM STACKOVERFLOW

zalando

QUICK TIP II

WHEN WRITING
INSERT INTO
ALWAYS
LIST YOUR COLUMNS

zalando

EVERYTHING YOU ALWAYS WANTED TO KNOW ABOUT SELECT * BUT WERE AFRAID TO ASK

SELECT * - WHY NOT

- UNNECESSARY DATA
- NOT USING INDEXES
- YOU CANNOT RELY ON COLUMN ORDER

zalando

SELECT * - WHEN OK

- * AS ROW - COUNT(*)
- AD-HOC QUERIES

zalando

QUICK TIP III

ORDER BY AND GROUP BY USING ORDINAL IS A BAD IDEA

zalando

RIIING… RIIING… RIIING...

Unknown Caller
?
Nope
Yep

SABRINA

JOIN SYNTAX

```
SELECT p.id,
       a.id,
       a.address_1
  FROM person p,
       address a
 WHERE p.id = a.id
   AND p.name = ‘Bojack’
```

zalando

JOIN SYNTAX

```
    SELECT p.id,
           a.id,
           a.address_1
      FROM person p
INNER JOIN address a
        ON p.id = a.id
     WHERE p.name = ‘Bojack’
```

zalando

JOIN SYNTAX

```
    SELECT p.id,
           a.id,
           a.address_1
      FROM person p
INNER JOIN address a
        ON p.id = a.id
     WHERE p.name = ‘Bojack’
```

zalando

JOIN SYNTAX

```
    SELECT p.id,
           a.id,
           a.address_1
      FROM person p
INNER JOIN address a
        ON p.id = a.id
     WHERE p.name = ‘Bojack’
```

- **SEPARATE WHERE FROM JOIN CONDITIONS**
- **AVOID CROSS JOINS**

zalando

JOIN SYNTAX

```
SELECT p.id,
       a.id,
       a.address_1
  FROM person p,
       address a
 WHERE p.id (+) = a.id
   AND p.name = ‘Bojack’
```

```
    SELECT p.id,
           a.id,
           a.address_1
      FROM person p
RIGHT JOIN address a
        ON p.id = a.id
     WHERE p.name = ‘Bojack’
```

JOIN SYNTAX

```
SELECT p.id,
       a.id,
       a.address_1
  FROM person p,
       address a
 WHERE p.id = a.id (+)
   AND p.name = ‘Bojack’
```

```
   SELECT p.id,
          a.id,
          a.address_1
     FROM person p
LEFT JOIN address a
       ON p.id = a.id
    WHERE p.name = ‘Bojack’
```

BUT WHY?

JOIN SYNTAX

REASONS WHY PEOPLE USE OLD JOIN SYNTAX (FROM STACKOVERFLOW):

- PEOPLE ARE USED TO IT
- PEOPLE ARE LAZY: "OLD STYLE" = LESS TYPING
- BEGINNERS OFTEN HAVE PROBLEMS TO UNDERSTAND THE SQL-92 JOIN SYNTAX
- PEOPLE ARE UNAWARE OF THE BENEFITS: YOU FILTER A TABLE *BEFORE* YOU DO AN OUTER JOIN, AND NOT AFTER IT WHEN ALL YOU HAVE IS THE WHERE CLAUSE
- PEOPLE DON'T SWITCH TO NEW SYNTAX JUST BECAUSE IT IS THERE
- IT’S AN ORACLE THING

zalando

JOIN SYNTAX

MY QUESTION

Dear Oracle Masters,

here is a poor disciple looking for guidance, I know the way to reach the true knowledge does not have an end, but I would appreciate few words to make my journey more safe, especially for my fellow travelers.

Here is my question, **we are already in the second half of year 2017 and I still meet people who think it is a good idea to write joins like**:

…

I see the advantages of the second syntax, even without playing with outer joins, and it is easy to explain what are the benefits. Often the answer I get is that the first one is the Oracle "recommended" syntax.

FROM ASK TOM

zalando

JOIN SYNTAX

THEIR ANSWER (PART 1 of 2)

Oracle style joins may have been recommended in the past, but the current guidance is:

Oracle recommends that you use the FROM clause OUTER JOIN syntax rather than the Oracle join operator

http://docs.oracle.com/database/122/SQLRF/Joins.htm#SQLRF30046

This is because there are number of restrictions that apply to the (+) operator, but not the "outer join" clause. Follow the link for the full list.

FROM ASK TOM

JOIN SYNTAX

THEIR ANSWER (PART 2 of 2)

Materialized View Query Rewrite

Currently ANSI syntax isn't fully supported for query rewrite. So if you use MVs a lot you're better off sticking with Oracle style.

Personally I prefer ANSI style. But I think it's more important for your code to be consistent. This makes it easier to follow.

If working on a legacy app all coded using Oracle syntax, I'd use that.

FROM ASK TOM

JOIN SYNTAX

OTHER THINGS THAT COULD GO WRONG:

- ORDER OF COLUMNS
- MIX WHERE AND JOIN CONDITIONS
- USING DISTINCT

JOIN SYNTAX

```
SELECT * FROM (
    SELECT TP.TOPIC_ID, CK.NAME
    FROM TD_TOPIC TP
    INNER JOIN TD_CIRCLE CK on CK.CIRCLE_ID =
TP.CIRCLE_ID AND CK.VALID = 1 AND SYSDATE > CK.EFF_TIME
    WHERE TP.VALID = 1 AND TP.FORWARD_FROM_TOPIC_ID = 0
    ORDER BY TP.CREATE_TIME DESC
) WHERE ROWNUM<21
```

zalando

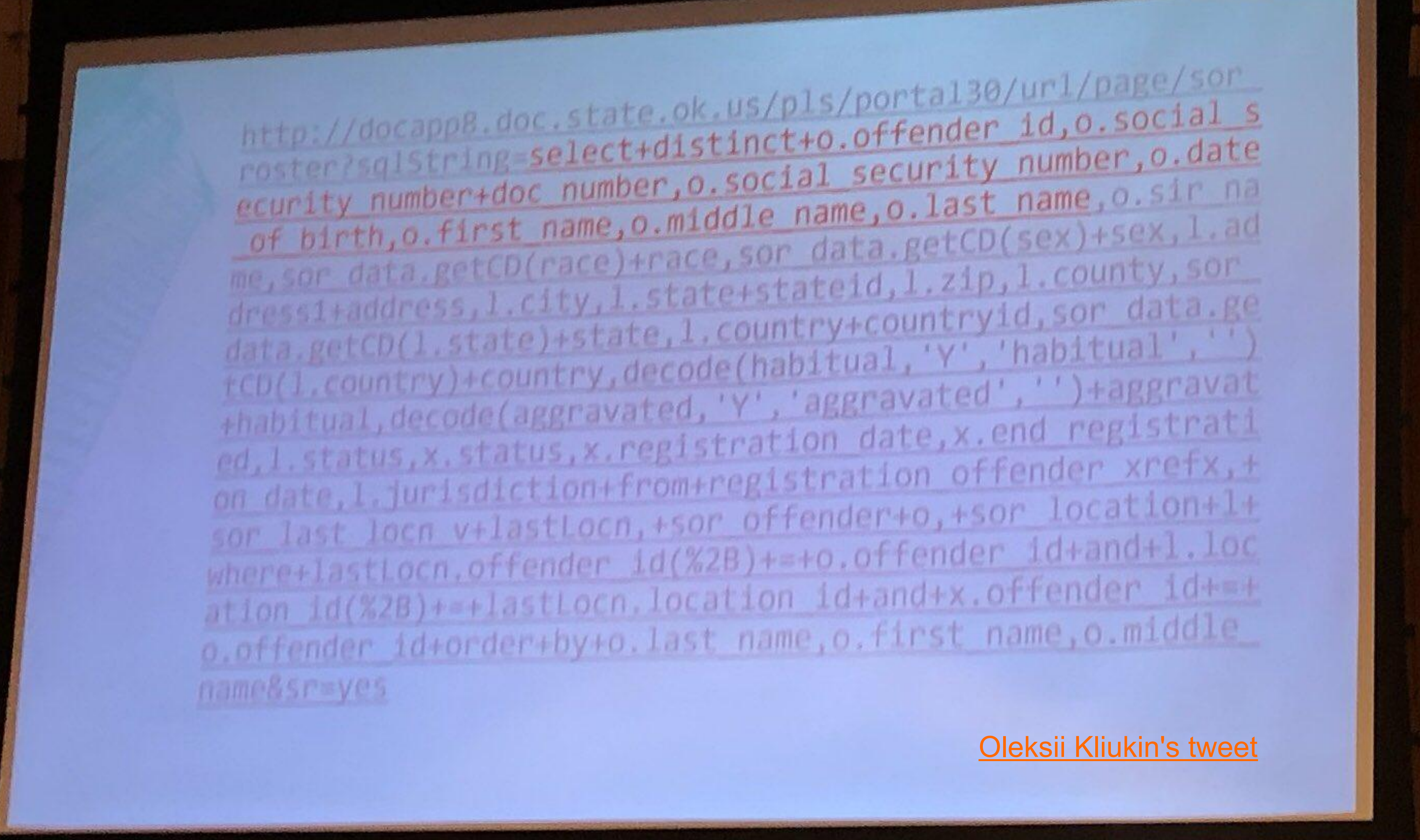
JOIN SYNTAX

```
SELECT *
  FROM (
             SELECT TP.TOPIC_ID,
                    CK.NAME
               FROM TD_TOPIC TP
         INNER JOIN TD_CIRCLE CK
                 ON CK.CIRCLE_ID = TP.CIRCLE_ID
                AND CK.VALID = 1
                AND SYSDATE > CK.EFF_TIME
              WHERE TP.VALID = 1
                AND TP.FORWARD_FROM_TOPIC_ID = 0
           ORDER BY TP.CREATE_TIME DESC
       )
 WHERE ROWNUM<21
```

zalando

JOIN SYNTAX

```
SELECT *
  FROM (
              SELECT TP.TOPIC_ID,
                     CK.NAME
                FROM TD_TOPIC TP
          INNER JOIN TD_CIRCLE CK
                  ON CK.CIRCLE_ID = TP.CIRCLE_ID
               WHERE TP.VALID = 1
                 AND TP.FORWARD_FROM_TOPIC_ID = 0
                 AND CK.VALID = 1
                 AND CK.EFF_TIME < SYSDATE
            ORDER BY TP.CREATE_TIME DESC
       )
 WHERE ROWNUM<21
```



```
http://docapp8.doc.state.ok.us/pls/portal30/url/page/sor_
roster?sqlString=select+distinct+o.offender_id,o.social_s
ecurity_number+doc_number,o.social_security_number,o.date
_of_birth,o.first_name,o.middle_name,o.last_name,o.sir_na
me,sor_data.getCD(race)+race,sor_data.getCD(sex)+sex,l.ad
dress1+address,l.city,l.state+stateid,l.zip,l.county,sor_
data.getCD(l.state)+state,l.country+countryid,sor_data.ge
tCD(l.country)+country,decode(habitual,'Y','habitual','')
+habitual,decode(aggravated,'Y','aggravated','')+aggravat
ed,l.status,x.status,x.registration_date,x.end_registrati
on_date,l.jurisdiction+from+registration_offender_xrefx,+
sor_last_locn_v+lastLocn,+sor_offender+o,+sor_location+l+
where+lastLocn.offender_id(%2B)+=+o.offender_id+and+l.loc
ation_id(%2B)+=+lastLocn.location_id+and+x.offender_id+=+
o.offender_id+order+by+o.last_name,o.first_name,o.middle_
name&sr=yes
```

Oleksii Kliukin's tweet

USING DISTINCT

```
SELECT DISTINCT o.offender_id,
                o.social_security_number doc_number,
                o.social_security_number,
                o.date_of_birth,
                o.first_name,
                o.middle_name,
                o.last_name,
                o.sir_name,
                sor_data.getCD(race) race,
                sor_data.getCD(sex) sex,
                l.address1 address,
                l.city,
                l.state stateid,
                l.zip,
                l.county,
                sor_data.getCD(l.state) state,
                l.country countryid,
                sor_data.getCD(l.country) country,
                decode(habitual,'Y','habitual','') habitual,
                decode(aggravated,'Y','aggravated','') aggravated,
                l.status,
                x.status,
                x.registration_date,
                x.end_registration_date,
                l.jurisdiction
FROM registration_offender_xref x,
     sor_last_locn_v lastLocn,
     sor_offender o,
     sor_location l
WHERE lastLocn.offender_id(%2B) = o.offender_id
  AND l.location_id(%2B) = lastLocn.location_id
  AND x.offender_id = o.offender_id
ORDER BY o.last_name,
         o.first_name,
         o.middle_name
```

https://2017.pgconf.eu/f

QUESTIONS

FRANCESCO MUCIO

BI CORE

BI ARCHITECT

francesco.mucio@zalando.com

+49 176 1275 8695

26-10-2017