

Diese PDF-Vorabversion meiner Folien verzichtet auf Hintergrundbilder zugunsten schnellerer Downloads. Die endgültige HTML-Version wird ab 19. November 2019 online unter folgendem Link verfügbar sein:

[ogobrecht.github.io/posts/2019-11-19-doag-konferenz](https://ogobrecht.github.io/posts/2019-11-19-doag-konferenz)

# **SCHNELLSTART**

## **VERSIONSKONTROLLE FÜR EXISTIERENDE ORACLE PROJEKTE**

Ottmar Gobrecht [@ogobrecht](#)

DOAG Konferenz Nürnberg  
19. November 2019

# ZU MEINER PERSON

- Oracle APEX Entwickler seit 2008 (APEX 3.1)
- Seit 2013 im Headquarter der Linde AG
- Individualsoftware für Fachbereiche
- Aktiv im [Open Source Bereich](#)

# MOTIVATION

*„There is no clean (database)  
development without Version Control“*

*Samuel Nitsche ([Blog Post](#))*

**DEPLOYMENT PAIN?!?**

# JUST DO IT



**PeterRaganitsch**

@PeterRaganitsch



Source code control is just awesome and saved me again. Having a history of all old versions of your code is invaluable!

Can't stress that point enough, get your shit together and use git, or anything else. Just do it.

[#code](#) [#orclAPEX](#) [#oracle](#) [#programming](#)

Tweet vom 7. Oktober 2019

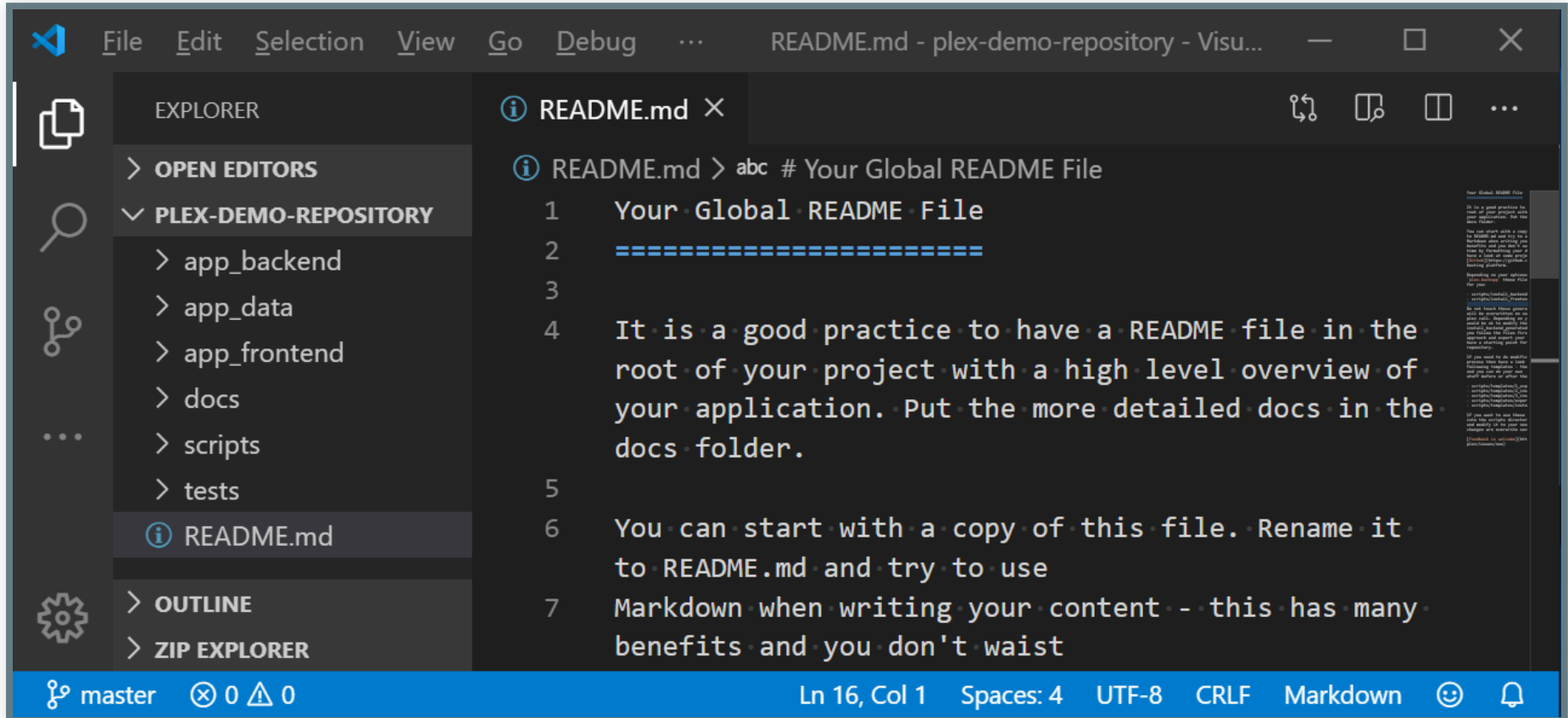
# UNSER ZIEL

Export „all in one“ für Versionskontrolle

- Backend (Schema DDL)
- Daten (Katalogdaten)
- Frontend (APEX App, zerlegt)
- Deployment Templates
- Wiederanlauffähigkeit
- Übersichtliche Dateistruktur



# GEWÜNSCHTE VERZEICHNISSTRUKTUR



Kurze Wege, alle Skripte vereint, übersichtlich

# TOOLS

# TOOL-VERGLEICH DDL EXPORT - KRITERIEN

Tools: SQL Dev, PL/SQL Dev, Toad, PLEX

- Eine Skript-Datei pro Objekt?
- Unterverzeichnisse pro Objekttyp?
- Eigene Dateien FK Constraints?
- "Object already exist" verhinderbar?
- Daten exportierbar?
- APEX App exportierbar?

# TOOL-VERGLEICH DDL EXPORT

Kriterium	SQL Dev.	PL/SQL Dev.	Toad	PLEX
Datei pro Objekt	Ja	Ja	Ja	Ja
Unterverz. pro Typ	Ja	Nein	Ja	Ja
FK Constr. extra	Ja	Nein	Ja	Ja
Verhi. object exist	Nein	Nein	Nein	Ja
Export Daten	Ja	Nein	<i>Jein</i>	Ja
Export APEX App	Ja	Nein	Nein	Ja

# ANMERKUNGEN PLEX

- Ist ein Package (PL/SQL Export Utilities)
- Ausgabeverzeichnisstruktur anpassbar
- APEX App zerlegt  
(Änderungen im VCS nachvollziehbar)
- [Projekt auf GitHub](#)
- Wenig Nacharbeit erforderlich

# ANMERKUNGEN TOAD

- Zwei Exportmöglichkeiten (mindestens)
  - Entweder Unterverzeichnisse pro Objekttyp...
  - ... oder Daten
- Daten nur als Insert Statements
- Umfangreich konfigurierbar, unübersichtlich
- Viel Nacharbeit erforderlich

# ANMERKUNGEN PL/SQL DEVELOPER

- Wenig konfigurierbar
- Enttäuscht für Aufbau Quellcode-Repos
- Viel Nacharbeit erforderlich

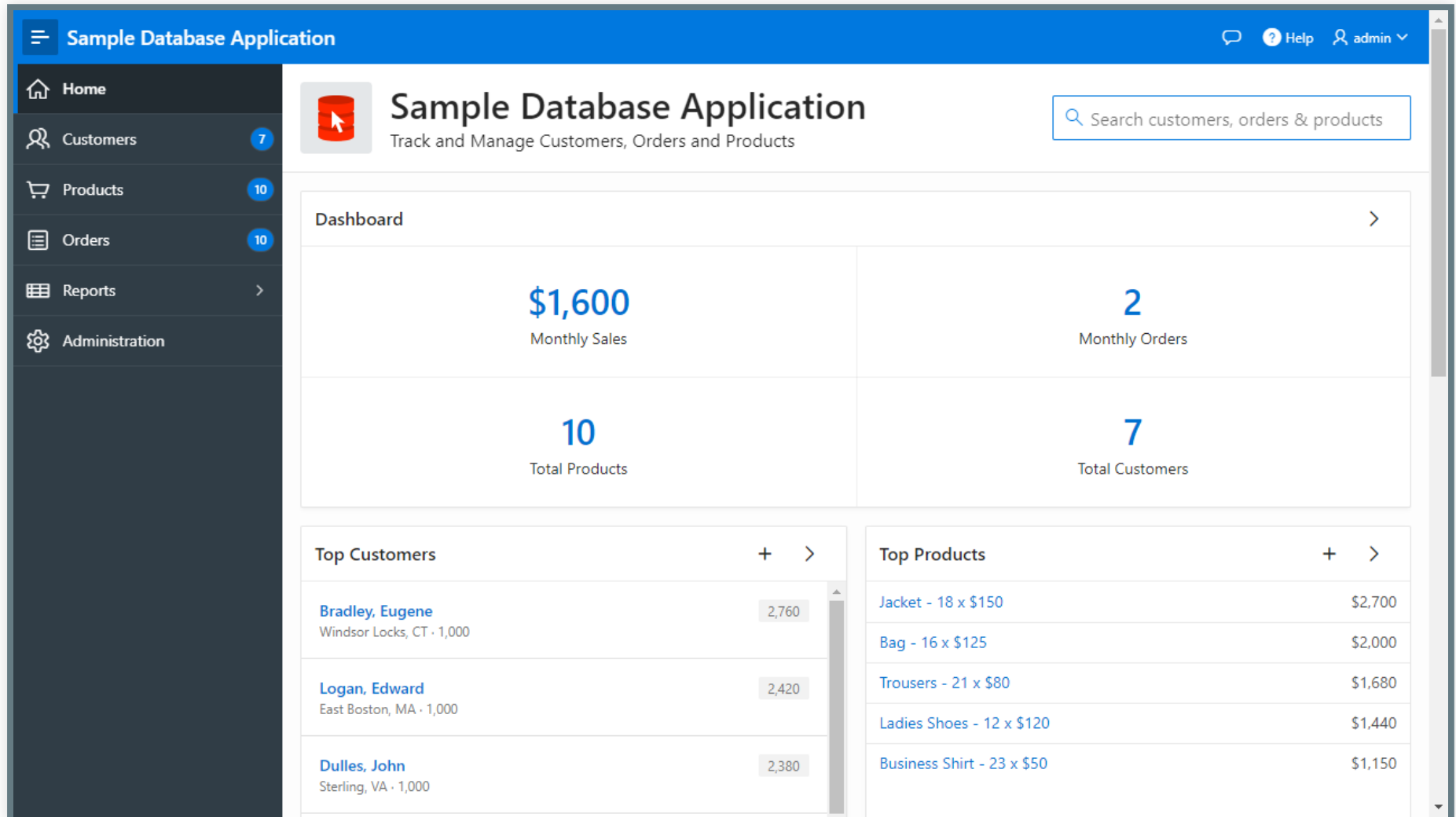
# ANMERKUNGEN SQL DEVELOPER

- Ist am übersichtlichsten
- Viele Formate für Datenexport (auch CSV)
- Umfangreich konfigurierbar
- Blain Carter: [CI/CD for Database Developers – Export Database Objects into Version Control](#)
- Nacharbeit erforderlich



**PLEX**

# AUSGANGSBASIS: SAMPLE DB APP



# MÖGLICHER ERSTEXPORT

```
WITH
  FUNCTION backapp RETURN BLOB IS
  BEGIN
    RETURN plex.to_zip(plex.backapp(
      p_app_id           => 100,
      p_include_object_ddl => true,
      p_include_templates => true,
      p_include_runtime_log => true,
      p_include_data      => true,
      p_data_table_name_like => 'DEMO_PRODUCT_INFO,DEMO_STATES'
    ));
  END backapp;

SELECT backapp FROM dual;
```

[Blog Post](#)

# DEMO

Create Repo & Export App

# SPEICHERN ALS ZIP

The screenshot shows the Oracle SQL Developer interface with a PL/SQL function named `backapp` in the main editor. The function returns a BLOB and includes a call to `plex.to_zip`. A red circle and the number '1' highlight the BLOB column in the query result grid.

A 'View Value' dialog box is open, displaying the 'Saved Data' section. The 'Download' button is circled in red, and a red circle with the number '2' highlights the 'View As' dropdown menu, which is currently set to 'Text'.

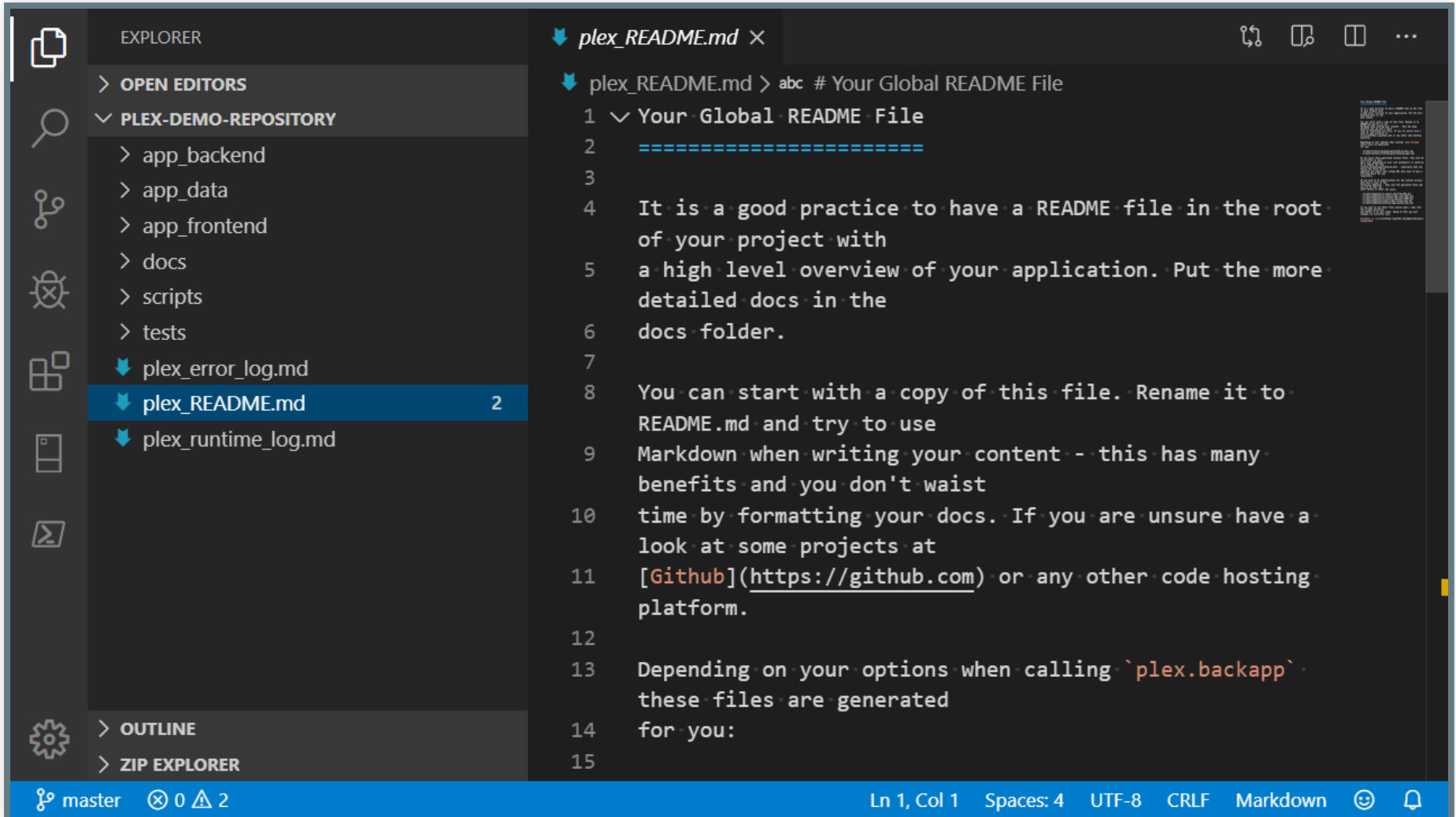
The 'Saved Data' section shows the following properties:

Property	Value
Length	403893
Empty	false
Temporary	true
Open	false

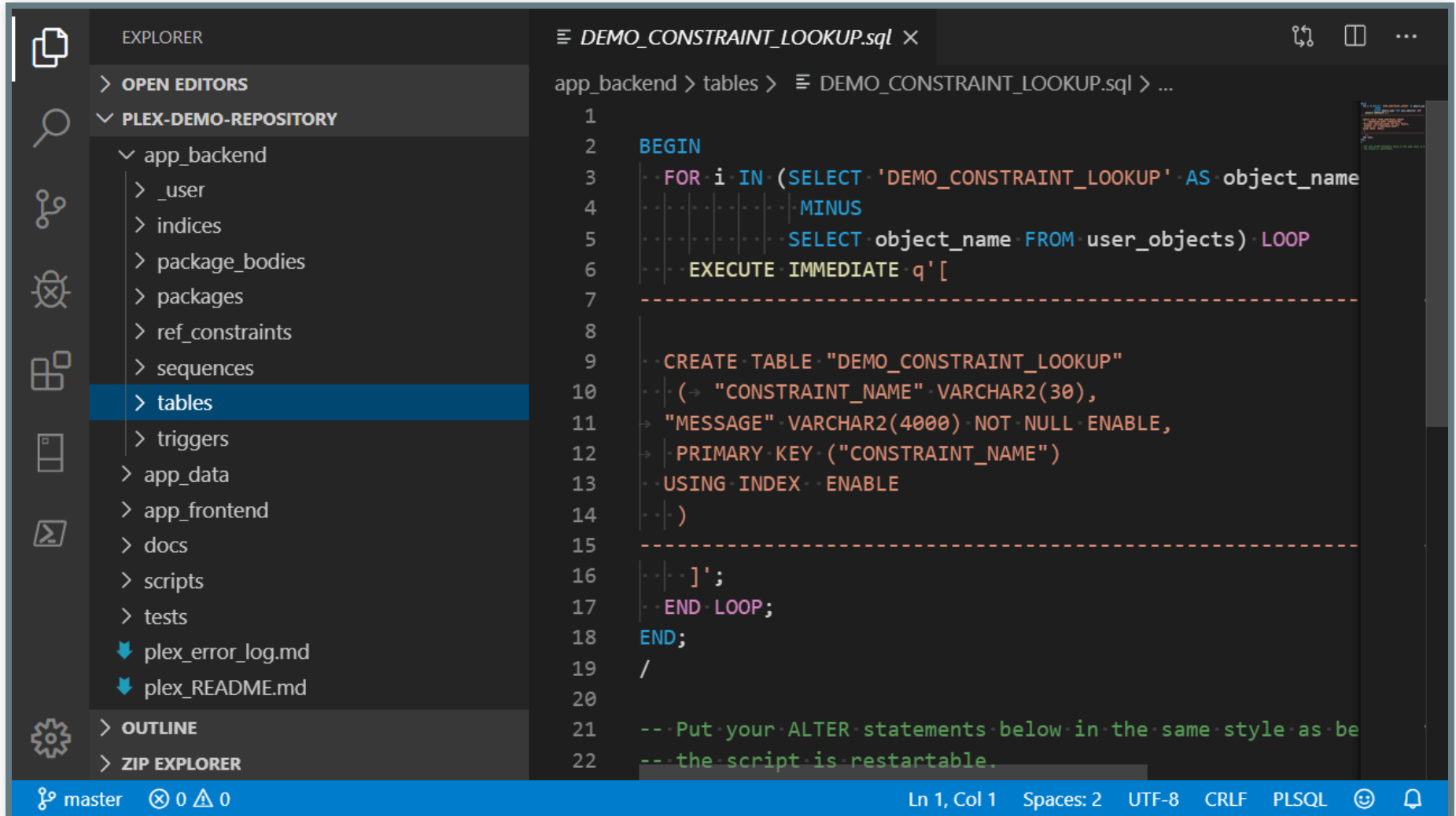
The 'Local Data' section is also visible, showing 'File Name' and 'File Size'.

The status bar at the bottom indicates 'Line 10 Column 20 | Insert | Windows: C'.

# DAS ENTPACKTE ZIP FILE



# BACKEND



The image shows a screenshot of an IDE interface. On the left is the EXPLORER panel, and on the right is the editor panel displaying a SQL script.

**EXPLORER Panel:**

- OPEN EDITORS
- PLEX-DEMO-REPOSITORY
  - app\_backend
    - \_user
    - indices
    - package\_bodies
    - packages
    - ref\_constraints
    - sequences
    - tables** (selected)
    - triggers
  - app\_data
  - app\_frontend
  - docs
  - scripts
  - tests
  - plex\_error\_log.md
  - plex\_README.md
- OUTLINE
- ZIP EXPLORER

**Editor Panel:**

File: DEMO\_CONSTRAINT\_LOOKUP.sql

Path: app\_backend > tables > DEMO\_CONSTRAINT\_LOOKUP.sql > ...

```
1
2 BEGIN
3   FOR i IN (SELECT 'DEMO_CONSTRAINT_LOOKUP' AS object_name
4             MINUS
5             SELECT object_name FROM user_objects) LOOP
6     EXECUTE IMMEDIATE q'[
7
8
9     CREATE TABLE "DEMO_CONSTRAINT_LOOKUP"
10    (
11      ("CONSTRAINT_NAME" VARCHAR2(30),
12      "MESSAGE" VARCHAR2(4000) NOT NULL ENABLE,
13      PRIMARY KEY ("CONSTRAINT_NAME")
14      USING INDEX ENABLE
15    )
16
17    ]';
18   END LOOP;
19 /
20
21 -- Put your ALTER statements below in the same style as be
22 -- the script is restartable.
```

**Status Bar:** master 0 0 Ln 1, Col 1 Spaces: 2 UTF-8 CRLF PLSQL

# KATALOGDATEN

The image shows a screenshot of a Visual Studio Code editor interface. On the left, the Explorer sidebar is open, showing a project structure under 'PLEX-DEMO-REPOSITORY'. The 'app\_data' folder is expanded, and 'DEMO\_STATES.csv' is selected. The main editor area displays the contents of 'DEMO\_STATES.csv', which is a CSV file with 22 lines of data. The status bar at the bottom shows the current file is 'master', has 0 errors and 0 warnings, and is using 'CSVLint' for linting. The status bar also indicates the current line and column (Ln 1, Col 1), the number of spaces (Spaces: 4), the encoding (UTF-8), the line ending (CRLF), and the file type (CSV).

EXPLORER

> OPEN EDITORS

▼ PLEX-DEMO-REPOSITORY

- > app\_backend
- ▼ app\_data
  - DEMO\_PRODUCT\_INFO.csv
  - DEMO\_STATES.csv**
- > app\_frontend
- > docs
- > scripts
- > tests
- plex\_error\_log.md
- plex\_README.md
- plex\_runtime\_log.md

> OUTLINE

> ZIP EXPLORER

DEMO\_STATES.csv

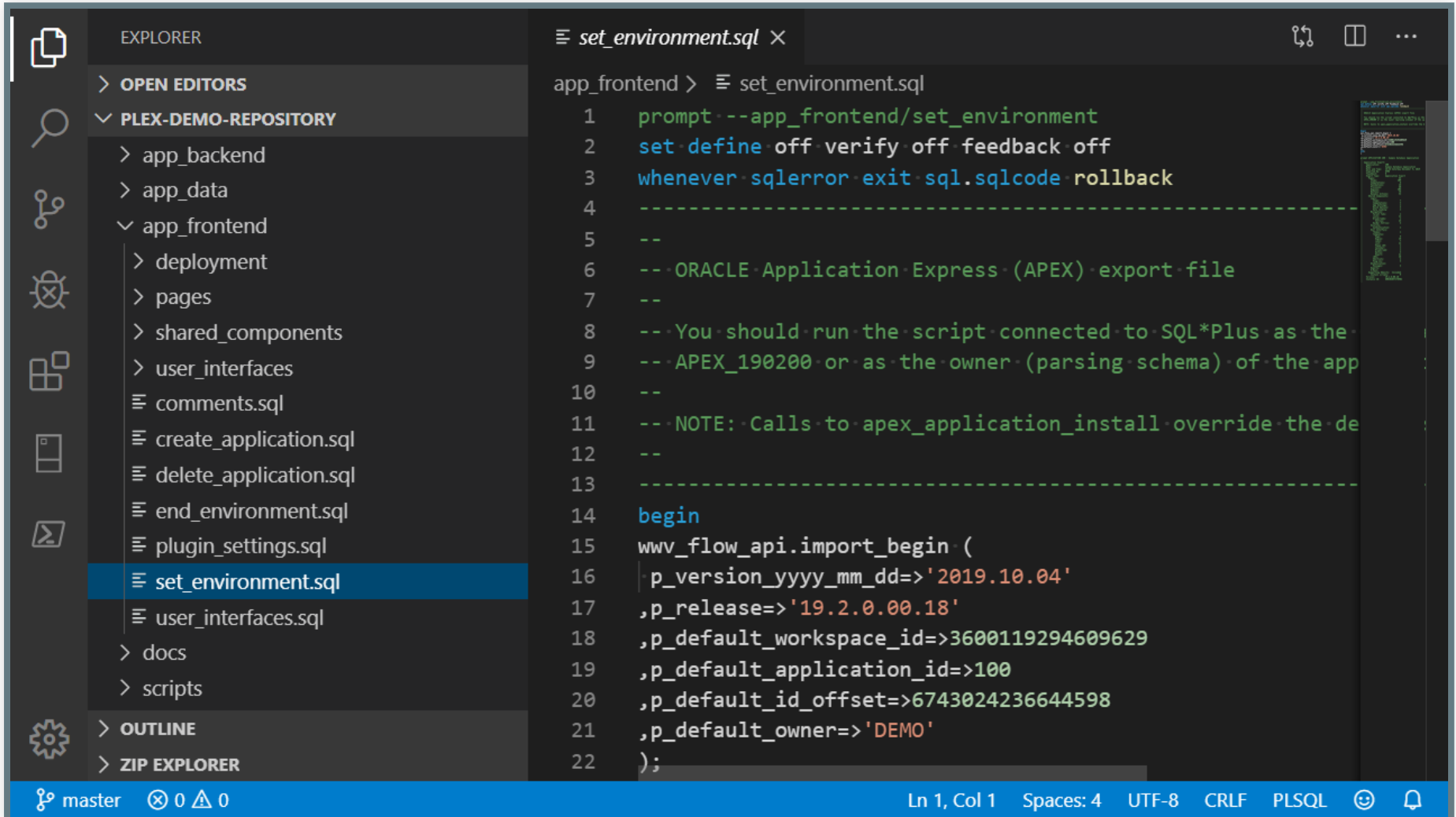
app\_data > DEMO\_STATES.csv

```
1 ST,STATE_NAME
2 AK,ALASKA
3 AL,ALABAMA
4 AR,ARKANSAS
5 AZ,ARIZONA
6 CA,CALIFORNIA
7 CO,COLORADO
8 CT,CONNECTICUT
9 DC,DISTRICT OF COLUMBIA
10 DE,DELAWARE
11 FL,FLORIDA
12 GA,GEORGIA
13 HI,HAWAII
14 IA,IOWA
15 ID,IDAHO
16 IL,ILLINOIS
17 IN,INDIANA
18 KS,KANSAS
19 KY,KENTUCKY
20 LA,LOUISIANA
21 MA,MASSACHUSETTS
22 MD,MARYLAND
```

master 0 0 CSVLint RBQL Align Rainbow OFF Ln 1, Col 1 Spaces: 4 UTF-8 CRLF CSV



# FRONTEND ZERLEGT



The screenshot shows an IDE interface with a dark theme. On the left is a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Accounts, and Settings. The Explorer panel is open, showing a project structure under 'PLEX-DEMO-REPOSITORY'. The 'app\_frontend' folder is expanded, listing files like 'comments.sql', 'create\_application.sql', 'delete\_application.sql', 'end\_environment.sql', 'plugin\_settings.sql', 'set\_environment.sql' (which is selected and highlighted in blue), and 'user\_interfaces.sql'. Below the Explorer is an 'OUTLINE' and 'ZIP EXPLORER' section. The main editor area displays the content of 'set\_environment.sql'. The script is a PL/SQL block that sets environment variables for an APEX application. It includes comments in green and SQL code in blue and black. The status bar at the bottom indicates the current file is 'master', there are 0 errors and 0 warnings, and the cursor is at line 1, column 1. The status bar also shows 'Spaces: 4', 'UTF-8', 'CRLF', 'PLSQL', and icons for a smiley face and a bell.

EXPLORER

> OPEN EDITORS

▼ PLEX-DEMO-REPOSITORY

- > app\_backend
- > app\_data
- ▼ app\_frontend
  - > deployment
  - > pages
  - > shared\_components
  - > user\_interfaces
  - ≡ comments.sql
  - ≡ create\_application.sql
  - ≡ delete\_application.sql
  - ≡ end\_environment.sql
  - ≡ plugin\_settings.sql
  - ≡ set\_environment.sql
  - ≡ user\_interfaces.sql
- > docs
- > scripts

OUTLINE

ZIP EXPLORER

≡ set\_environment.sql ×

app\_frontend > ≡ set\_environment.sql

```
1  prompt --app_frontend/set_environment
2  set define off verify off feedback off
3  whenever sqlerror exit sql.sqlcode rollback
4  -----
5  --
6  --ORACLE Application Express (APEX) export file
7  --
8  --You should run the script connected to SQL*Plus as the
9  --APEX_190200 or as the owner (parsing schema) of the app
10 --
11 --NOTE: Calls to apex_application_install override the de
12 --
13 -----
14 begin
15 wwv_flow_api.import_begin (
16   p_version_yyyy_mm_dd=>'2019.10.04'
17   ,p_release=>'19.2.0.00.18'
18   ,p_default_workspace_id=>3600119294609629
19   ,p_default_application_id=>100
20   ,p_default_id_offset=>6743024236644598
21   ,p_default_owner=>'DEMO'
22 );
```

master 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PLSQL

# BACKEND MASTER SCRIPT

The screenshot shows an IDE interface with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORER', shows a project structure for 'PLEX-DEMO-REPOSITORY'. It includes folders for 'app\_backend', 'app\_data', 'app\_frontend', 'docs', 'scripts', 'logs', and 'templates'. The 'scripts' folder is expanded, showing 'install\_backend\_generated\_by\_plex.sql' (selected), 'install\_frontend\_generated\_by\_apex.sql', and 'tests'. Below the explorer are sections for 'OUTLINE' and 'ZIP EXPLORER'. The code editor displays the content of 'install\_backend\_generated\_by\_plex.sql', which is a SQL script for installing a backend. The script includes comments in green, SQL commands in blue, and prompts in green. The status bar at the bottom indicates the current file is 'master', there are 0 errors and 0 warnings, and the cursor is at line 1, column 1. The status bar also shows 'Spaces: 4', 'UTF-8', 'CRLF', 'PLSQL', and icons for a smiley face and a bell.

EXPLORER

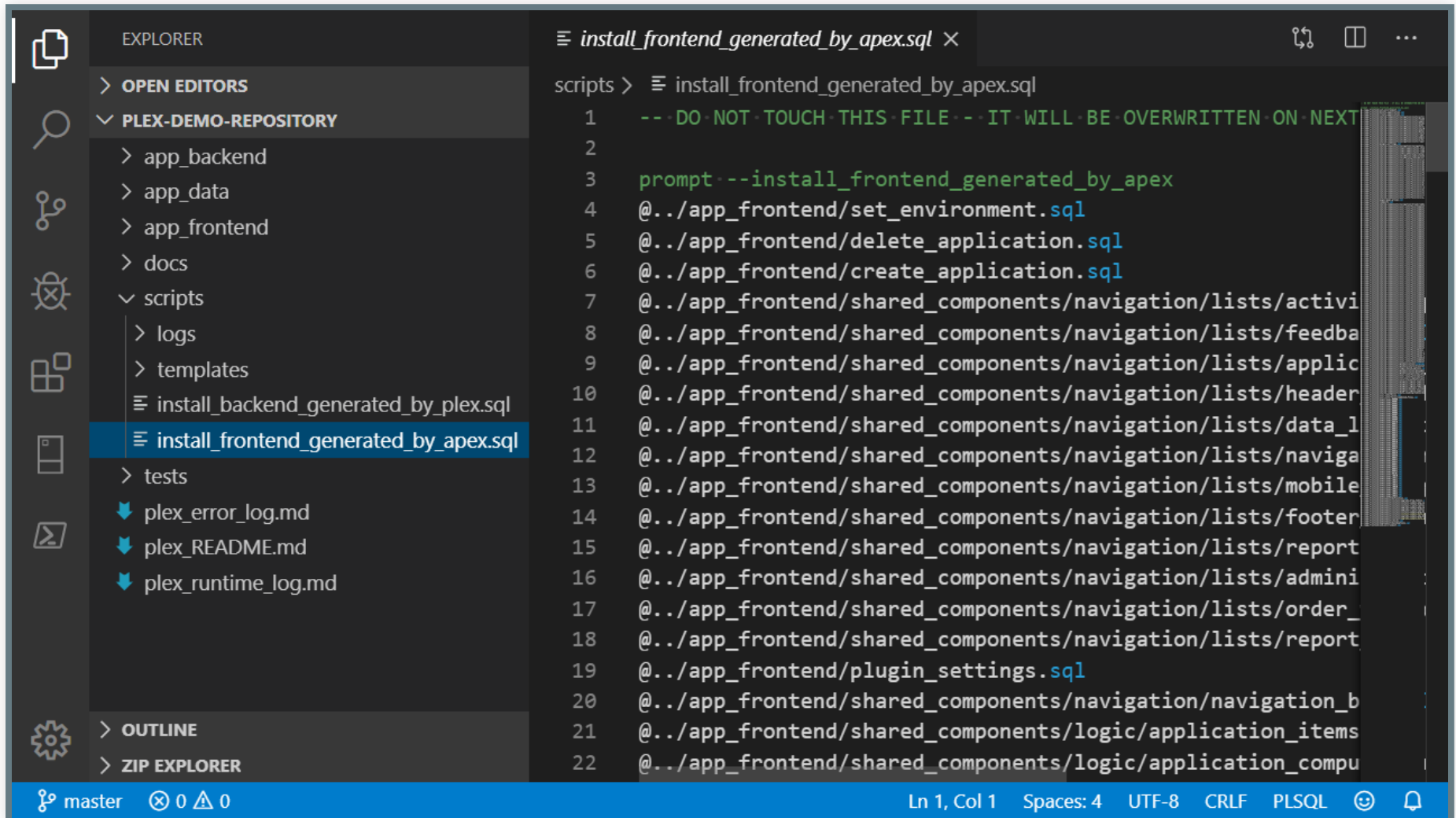
- > OPEN EDITORS
- ▼ PLEX-DEMO-REPOSITORY
  - > app\_backend
  - > app\_data
  - > app\_frontend
  - > docs
  - ▼ scripts
    - > logs
    - > templates
    - ≡ install\_backend\_generated\_by\_plex.sql
    - ≡ install\_frontend\_generated\_by\_apex.sql
    - > tests
  - ▼ plex\_error\_log.md
  - ▼ plex\_README.md
  - ▼ plex\_runtime\_log.md
- > OUTLINE
- > ZIP EXPLORER

scripts > ≡ install\_backend\_generated\_by\_plex.sql

```
1  /* A-T-T-E-N-T-I-O-N
2  DO NOT TOUCH THIS FILE or set the PLEX.BackApp parameter p
3  to false -- otherwise your changes would be overwritten on
4  call. It is recommended to export your object ddl only one
5  repository creation and then start to use the "files first
6  */
7
8  set define off verify off feedback off
9  whenever sqlerror exit sql.sqlcode rollback
10
11  prompt --install_backend_generated_by_plex
12
13  prompt --app_backend/sequences/DEMO_CUST_SEQ
14  @../app_backend/sequences/DEMO_CUST_SEQ.sql
15
16  prompt --app_backend/sequences/DEMO_ORDER_ITEMS_SEQ
17  @../app_backend/sequences/DEMO_ORDER_ITEMS_SEQ.sql
18
19  prompt --app_backend/sequences/DEMO_ORD_SEQ
20  @../app_backend/sequences/DEMO_ORD_SEQ.sql
21
22  prompt --app_backend/sequences/DEMO_PROD_SEQ
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PLSQL

# FRONTEND MASTER SCRIPT



The image shows a screenshot of an IDE interface. On the left is the 'EXPLORER' sidebar with a tree view of a project named 'PLEX-DEMO-REPOSITORY'. The tree includes folders like 'app\_backend', 'app\_data', 'app\_frontend', 'docs', 'scripts', 'logs', 'templates', 'tests', and files like 'plex\_error\_log.md', 'plex\_README.md', and 'plex\_runtime\_log.md'. The file 'install\_frontend\_generated\_by\_apex.sql' is selected and highlighted in blue. On the right is the main editor window, titled 'install\_frontend\_generated\_by\_apex.sql', showing the content of the selected file. The script is a SQL file with comments and file paths. The status bar at the bottom indicates the current file is 'master', there are 0 errors and 0 warnings, and the cursor is at line 1, column 1. The status bar also shows 'Spaces: 4', 'UTF-8', 'CRLF', 'PLSQL', and icons for search and help.

EXPLORER

> OPEN EDITORS

▼ PLEX-DEMO-REPOSITORY

- > app\_backend
- > app\_data
- > app\_frontend
- > docs
- ▼ scripts
  - > logs
  - > templates
  - ≡ install\_backend\_generated\_by\_plex.sql
  - ≡ install\_frontend\_generated\_by\_apex.sql
  - > tests
- ▼ plex\_error\_log.md
- ▼ plex\_README.md
- ▼ plex\_runtime\_log.md

> OUTLINE

> ZIP EXPLORER

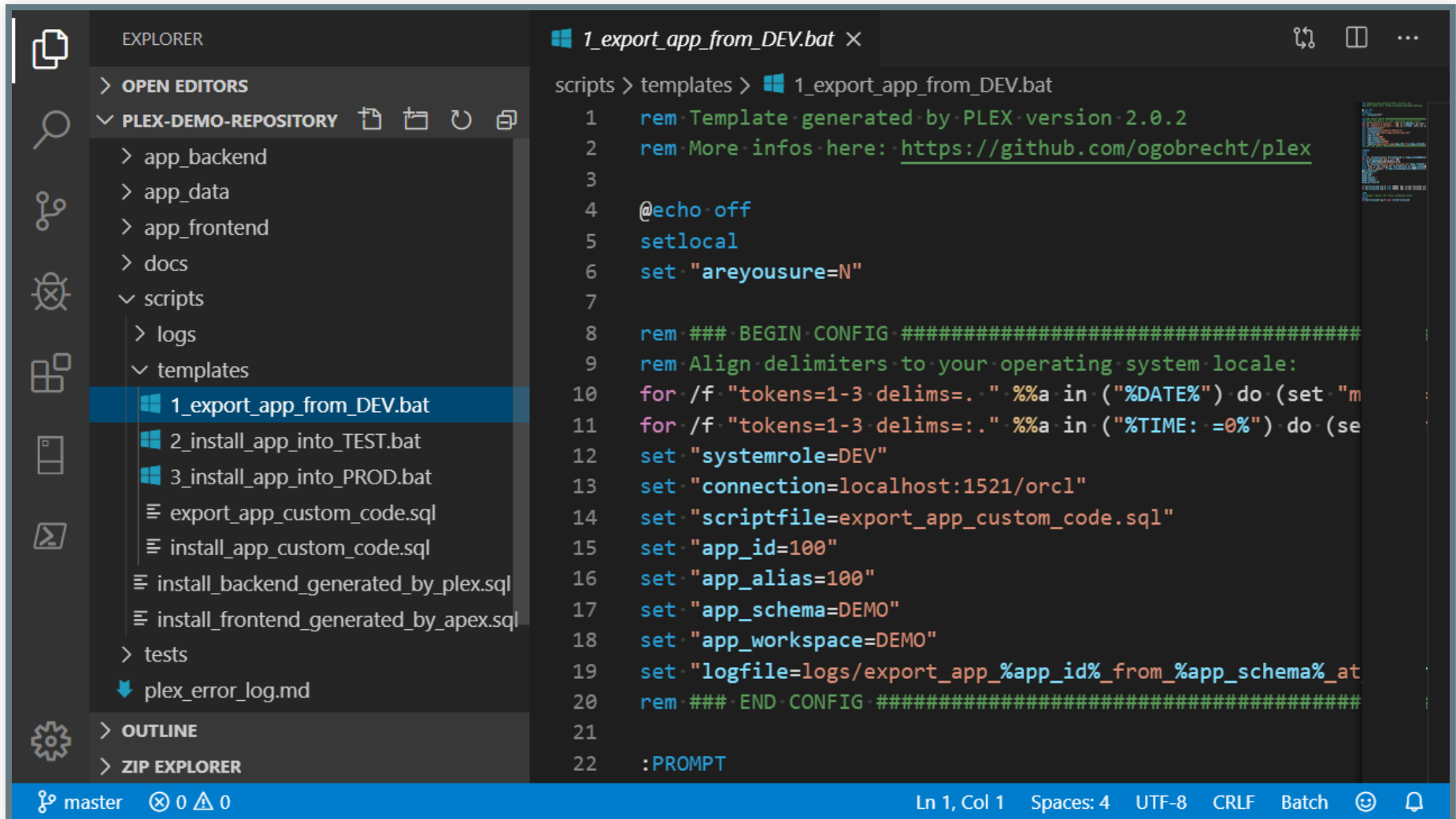
≡ install\_frontend\_generated\_by\_apex.sql ×

scripts > ≡ install\_frontend\_generated\_by\_apex.sql

```
1  --DO NOT TOUCH THIS FILE-- IT WILL BE OVERWRITTEN ON NEXT
2
3  prompt --install_frontend_generated_by_apex
4  @../app_frontend/set_environment.sql
5  @../app_frontend/delete_application.sql
6  @../app_frontend/create_application.sql
7  @../app_frontend/shared_components/navigation/lists/activi
8  @../app_frontend/shared_components/navigation/lists/feedba
9  @../app_frontend/shared_components/navigation/lists/applic
10 @../app_frontend/shared_components/navigation/lists/header
11 @../app_frontend/shared_components/navigation/lists/data_l
12 @../app_frontend/shared_components/navigation/lists/naviga
13 @../app_frontend/shared_components/navigation/lists/mobile
14 @../app_frontend/shared_components/navigation/lists/footer
15 @../app_frontend/shared_components/navigation/lists/report
16 @../app_frontend/shared_components/navigation/lists/admini
17 @../app_frontend/shared_components/navigation/lists/order_
18 @../app_frontend/shared_components/navigation/lists/report
19 @../app_frontend/plugin_settings.sql
20 @../app_frontend/shared_components/navigation/navigation_b
21 @../app_frontend/shared_components/logic/application_items
22 @../app_frontend/shared_components/logic/application_compu
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PLSQL

# DEPLOYMENT TEMPLATES



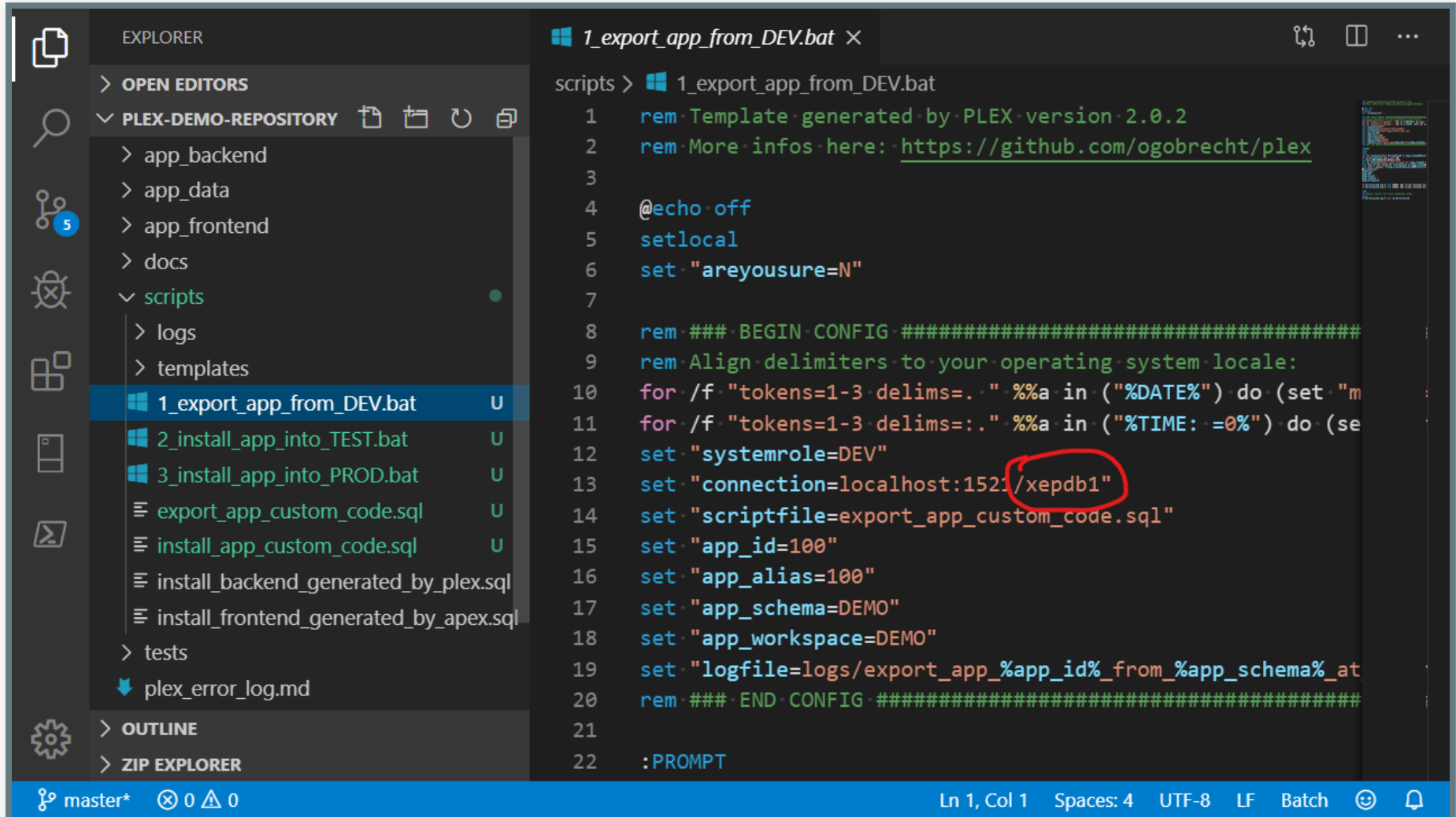
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar displays the file structure of the 'PLEX-DEMO-REPOSITORY', with the 'scripts' folder expanded and 'templates' subfolder selected. The file '1\_export\_app\_from\_DEV.bat' is highlighted. The Editor window shows the content of this batch file, which is a deployment template for a Plex application. The script includes comments about the version and a link to the GitHub repository, followed by configuration settings for the application, such as the system role, connection string, script file, app ID, app alias, app schema, and app workspace. The script also sets the log file path and ends with a prompt.

```
1  rem Template generated by PLEX version 2.0.2
2  rem More infos here: https://github.com/ogobrecht/plex
3
4  @echo off
5  setlocal
6  set "areyousure=N"
7
8  rem ### BEGIN CONFIG #####
9  rem Align delimiters to your operating system locale:
10 for /f "tokens=1-3 delims=." %%a in ("%DATE%") do (set "m
11 for /f "tokens=1-3 delims=." %%a in ("%TIME: =0%") do (se
12 set "systemrole=DEV"
13 set "connection=localhost:1521/orcl"
14 set "scriptfile=export_app_custom_code.sql"
15 set "app_id=100"
16 set "app_alias=100"
17 set "app_schema=DEMO"
18 set "app_workspace=DEMO"
19 set "logfile=logs/export_app_%app_id%_from_%app_schema%.at
20 rem ### END CONFIG #####
21
22 :PROMPT
```

master 0 0

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Batch

# TEMPLATES KOPIERT & ANGEPASST



EXPLORER

OPEN EDITORS

PLEX-DEMO-REPOSITORY

- app\_backend
- app\_data
- app\_frontend
- docs
- scripts
  - logs
  - templates
  - 1\_export\_app\_from\_DEV.bat U
  - 2\_install\_app\_into\_TEST.bat U
  - 3\_install\_app\_into\_PROD.bat U
  - export\_app\_custom\_code.sql U
  - install\_app\_custom\_code.sql U
  - install\_backend\_generated\_by\_plex.sql
  - install\_frontend\_generated\_by\_apex.sql
- tests
- plex\_error\_log.md

OUTLINE

ZIP EXPLORER

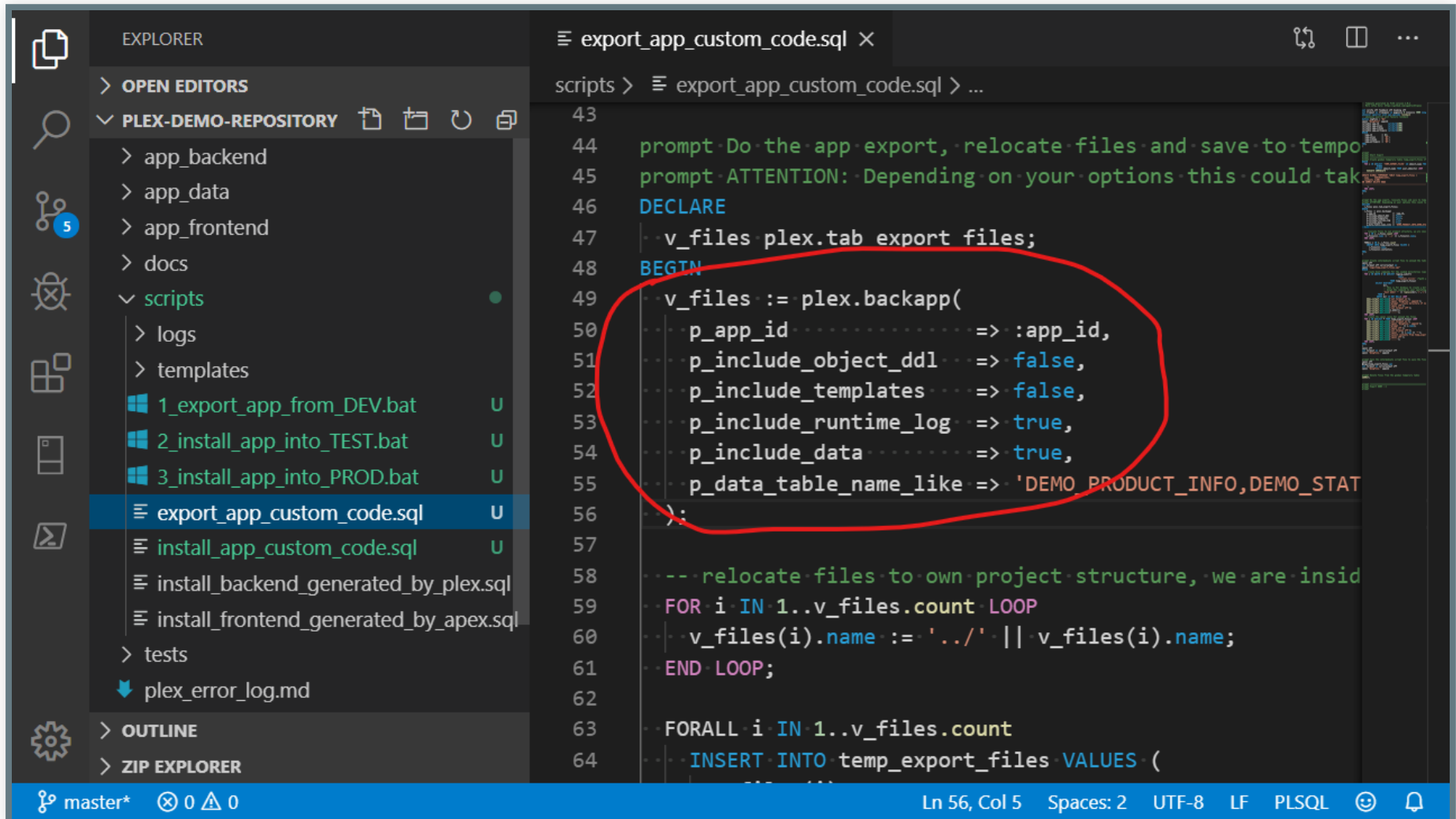
1\_export\_app\_from\_DEV.bat

```
1 rem Template generated by PLEX version 2.0.2
2 rem More infos here: https://github.com/ogobrecht/plex
3
4 @echo off
5 setlocal
6 set "areyousure=N"
7
8 rem ### BEGIN CONFIG #####
9 rem Align delimiters to your operating system locale:
10 for /f "tokens=1-3 delims=." %%a in ("%DATE%") do (set "m
11 for /f "tokens=1-3 delims=." %%a in ("%TIME: =0%") do (se
12 set "systemrole=DEV"
13 set "connection=localhost:1521/xepdb1"
14 set "scriptfile=export_app_custom_code.sql"
15 set "app_id=100"
16 set "app_alias=100"
17 set "app_schema=DEMO"
18 set "app_workspace=DEMO"
19 set "logfile=logs/export_app_%app_id%_from_%app_schema%.at
20 rem ### END CONFIG #####
21
22 :PROMPT
```

master\* 0 0

Ln 1, Col 1 Spaces: 4 UTF-8 LF Batch

# ZUKÜNFTIGE EXPORTE KONFIGURIEREN



The screenshot displays an IDE interface with a file explorer on the left and a code editor on the right.

**File Explorer (Left):**

- EXPLORER
  - OPEN EDITORS
  - PLEX-DEMO-REPOSITORY
    - app\_backend
    - app\_data
    - app\_frontend
    - docs
    - scripts
      - logs
      - templates
      - 1\_export\_app\_from\_DEV.bat
      - 2\_install\_app\_into\_TEST.bat
      - 3\_install\_app\_into\_PROD.bat
      - export\_app\_custom\_code.sql** (selected)
      - install\_app\_custom\_code.sql
      - install\_backend\_generated\_by\_plex.sql
      - install\_frontend\_generated\_by\_apex.sql
    - tests
    - plex\_error\_log.md
  - OUTLINE
  - ZIP EXPLORER

**Code Editor (Right):**

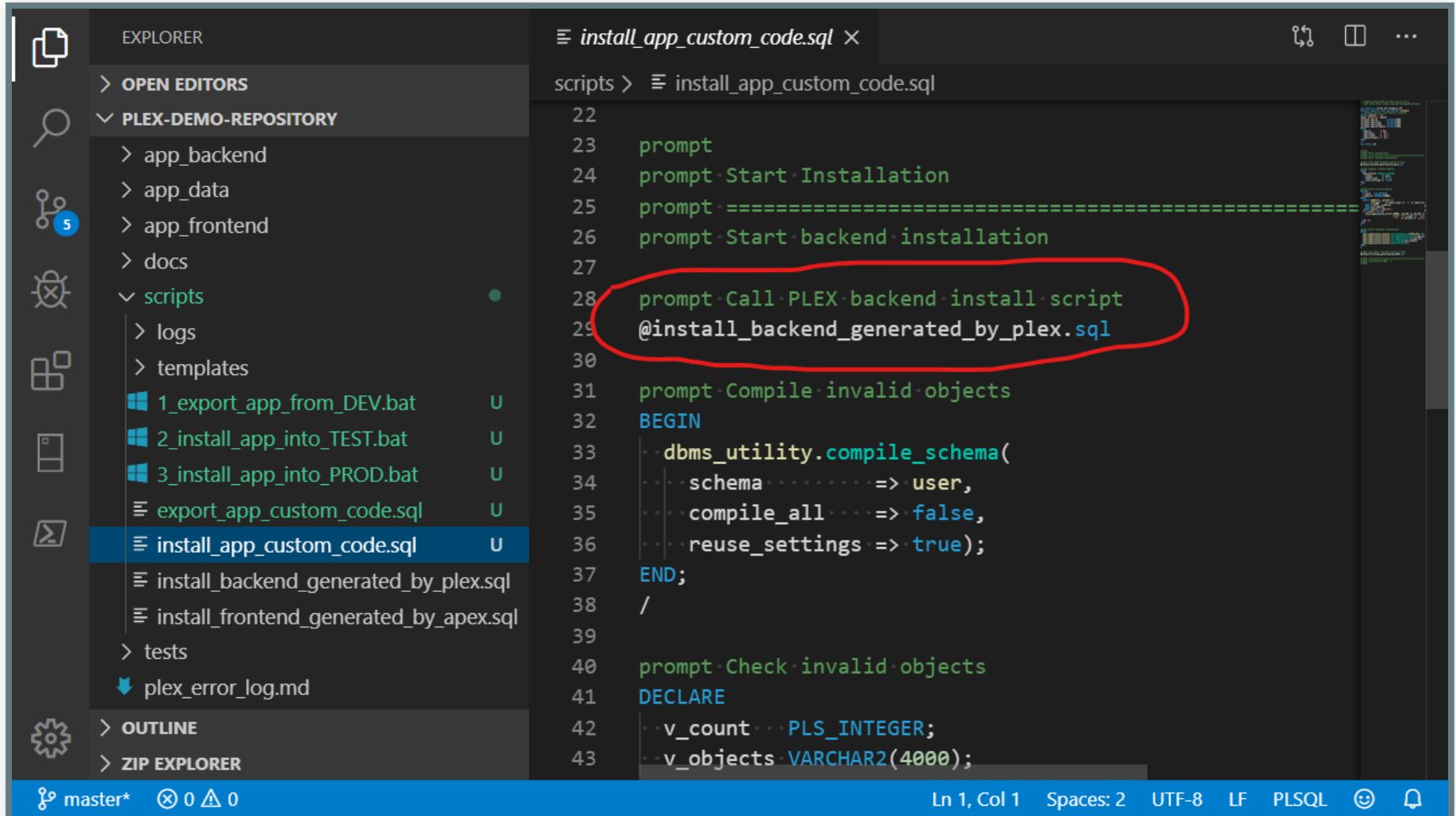
The code editor shows the file `export_app_custom_code.sql`. The code is as follows:

```
43
44 prompt Do the app export, relocate files and save to tempo
45 prompt ATTENTION: Depending on your options this could tak
46 DECLARE
47   v_files plex.tab export files;
48 BEGIN
49   v_files := plex.backapp(
50     p_app_id => :app_id,
51     p_include_object_ddl => false,
52     p_include_templates => false,
53     p_include_runtime_log => true,
54     p_include_data => true,
55     p_data_table_name_like => 'DEMO_PRODUCT_INFO,DEMO_STAT
56   );
57
58   ---relocate files to own project structure, we are insid
59   FOR i IN 1..v_files.count LOOP
60     v_files(i).name := '../' || v_files(i).name;
61   END LOOP;
62
63   FORALL i IN 1..v_files.count
64     INSERT INTO temp_export_files VALUES (
```

A red circle highlights the `plex.backapp()` function call and its parameters, indicating the configuration for future exports.

The status bar at the bottom shows: master\* 0 0 Ln 56, Col 5 Spaces: 2 UTF-8 LF PLSQL

# DEPLOYMENT MASTER SCRIPT 1/2



The screenshot displays an IDE interface with the following components:

- EXPLORER Panel:**
  - OPEN EDITORS
  - PLEX-DEMO-REPOSITORY
    - app\_backend
    - app\_data
    - app\_frontend
    - docs
    - scripts
      - logs
      - templates
      - 1\_export\_app\_from\_DEV.bat U
      - 2\_install\_app\_into\_TEST.bat U
      - 3\_install\_app\_into\_PROD.bat U
      - export\_app\_custom\_code.sql U
      - install\_app\_custom\_code.sql U**
      - install\_backend\_generated\_by\_plex.sql
      - install\_frontend\_generated\_by\_apex.sql
    - tests
    - plex\_error\_log.md
  - OUTLINE
  - ZIP EXPLORER

- Editor Panel:** Shows the file `install_app_custom_code.sql` with the following content:

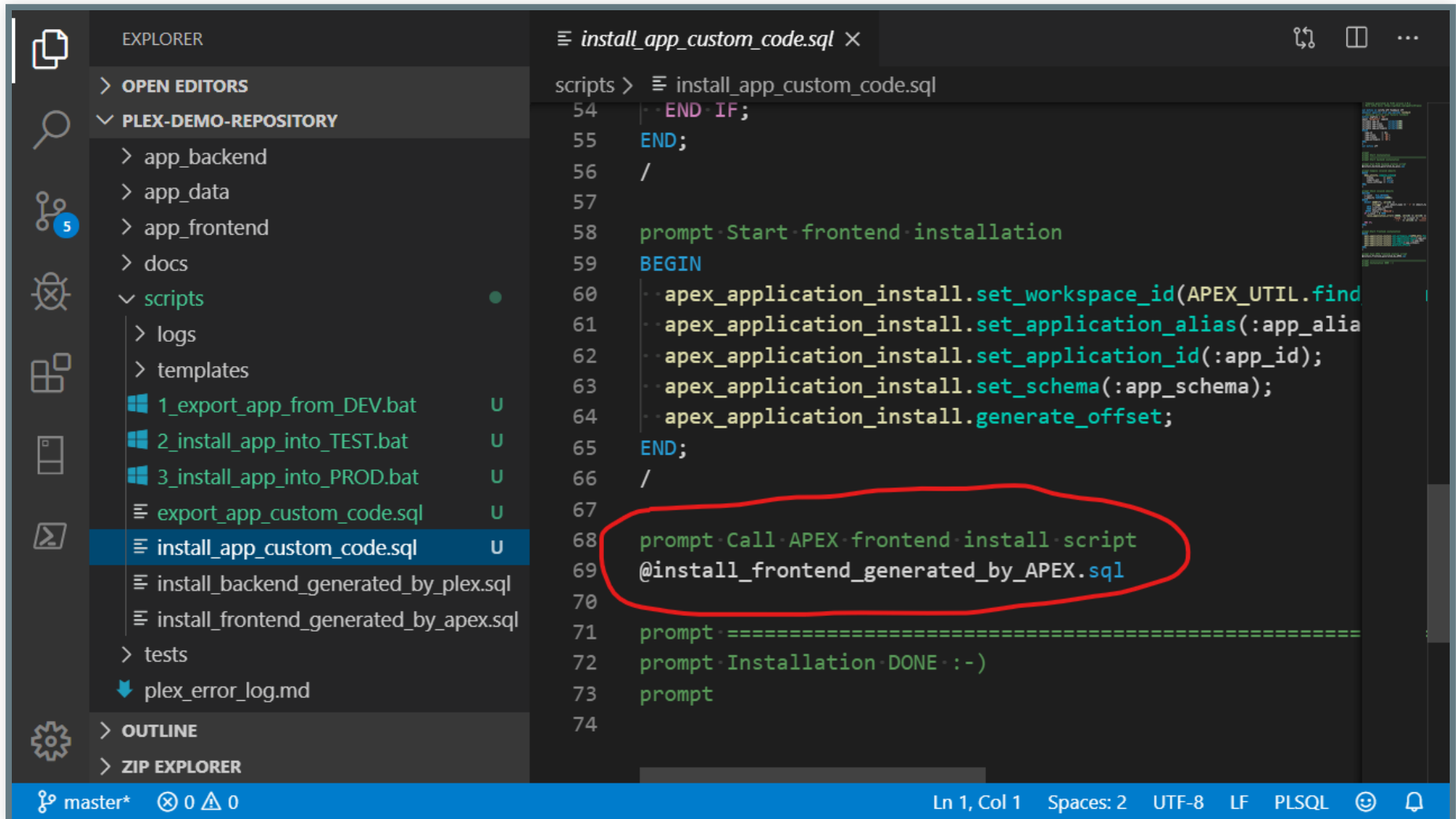
```
22
23 prompt
24 prompt Start Installation
25 prompt =====
26 prompt Start backend installation
27
28 prompt Call PLEX backend install script
29 @install_backend_generated_by_plex.sql
30
31 prompt Compile invalid objects
32 BEGIN
33   dbms_utility.compile_schema(
34     schema => user,
35     compile_all => false,
36     reuse_settings => true);
37 END;
38 /
39
40 prompt Check invalid objects
41 DECLARE
42   v_count PLS_INTEGER;
43   v_objects VARCHAR2(4000);
```

A red circle highlights the line `prompt Call PLEX backend install script @install_backend_generated_by_plex.sql`.

- Status Bar:** Shows `master*`, `0` errors, `0` warnings, `Ln 1, Col 1`, `Spaces: 2`, `UTF-8`, `LF`, `PLSQL`, and icons for search and help.



# DEPLOYMENT MASTER SCRIPT 2/2



EXPLORER

> OPEN EDITORS

▼ PLEX-DEMO-REPOSITORY

- > app\_backend
- > app\_data
- > app\_frontend
- > docs
- ▼ scripts
  - > logs
  - > templates
  - 1\_export\_app\_from\_DEV.bat U
  - 2\_install\_app\_into\_TEST.bat U
  - 3\_install\_app\_into\_PROD.bat U
  - ≡ export\_app\_custom\_code.sql U
  - ≡ **install\_app\_custom\_code.sql** U
  - ≡ install\_backend\_generated\_by\_plex.sql
  - ≡ install\_frontend\_generated\_by\_apex.sql
- > tests
- plex\_error\_log.md

> OUTLINE

> ZIP EXPLORER

≡ install\_app\_custom\_code.sql ×

scripts > ≡ install\_app\_custom\_code.sql

```
54  -- END IF;  
55  END;  
56  /  
57  
58  prompt Start frontend installation  
59  BEGIN  
60      apex_application_install.set_workspace_id(APEX_UTIL.find  
61      apex_application_install.set_application_alias(:app_alia  
62      apex_application_install.set_application_id(:app_id);  
63      apex_application_install.set_schema(:app_schema);  
64      apex_application_install.generate_offset;  
65  END;  
66  /  
67  
68  prompt Call APEX frontend install script  
69  @install_frontend_generated_by_APEX.sql  
70  
71  prompt =====  
72  prompt Installation DONE :-)  
73  prompt  
74
```

master\* 0 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF PLSQL



**DDL**

# AB HIER DATEIBASIERTES ARBEITEN

- Keine Änderungen über Klickibunti-Tools
- Alle Änderungen per Skript
- Nur noch deklarativen Code exportieren
  - APEX-Frontend
  - ORDS-REST-Services
  - ...

# WIEDERANLAUFFÄHIGKEIT

The screenshot shows an IDE interface with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORER', shows a project named 'PLEX-DEMO-REPOSITORY' with a folder 'app\_backend' containing several SQL files. The file 'DEMO\_STATES.sql' is selected and highlighted in blue. The code editor, titled 'DEMO\_STATES.sql', shows the following SQL script:

```
1
2 BEGIN
3   --FOR i IN (SELECT 'DEMO_STATES' AS object_name FROM dual
4   --           MINUS
5   --           SELECT object_name FROM user_objects) LOOP
6   --   EXECUTE IMMEDIATE q'[
7   -----
8
9   --CREATE TABLE "DEMO_STATES"
10  --  ( "ST" VARCHAR2(30),
11  --    "STATE_NAME" VARCHAR2(30)
12  --  )
13  -----
14  -- ];
15  --END LOOP;
16 END;
17 /
18
19 --Put your ALTER statements below in the same style as be
20 --the script is restartable.
21
22
```

The status bar at the bottom indicates the current file is 'master\*', there are 0 errors and 0 warnings, and the cursor is at line 1, column 1. The status bar also shows 'Spaces: 2', 'UTF-8', 'CRLF', 'PLSQL', and icons for a smiley face and a bell.

# DIE IDEE HINTER DER SKRIPTEREI

- Agile DB-Entwicklung
- Jede Änderung ist eine Migration
  - [Wikipedia: Schema migration](#)
  - [Artikel Samuel Nitsche](#)
  - [Artikel Martin Fowler](#)

Unser Ansatz ist nur eine mögliche Ausprägung der Idee

**GESCHWINDIGKEIT**

# MEHRARBEIT, DIE SICH AUSZAHLT

- Nur Skripte
- Kein manueller App Export/Import
- Alle Skripte wiederaanlauffähig
- Gesamtablauf getestet
- Reduzierte „Deployment Pain“

# DEMO

## APEX Export & Deployment

# APROPOS DEPLOYMENT PAIN

- Jede Umgebung ist individuell
- Mit den Skripten anfangen
- Kleine Schritte
- Immer besser werden
- Nicht stehenbleiben



**MEHR TOOLS**

# GIT VERSUS SVN

- Git ist schneller
- SVN braucht weniger Platz
- Git funktioniert offline
- SVN Rechteverwaltung ist flexibler
- Entscheidungshilfe: [Artikel zum Thema](#)
- Tipp Windows Server: [Git](#), [SVN](#)

# GITHUB DESKTOP

- Multi-Plattform (Linux in Arbeit)
- Reduziert auf das Wesentliche
- Übersichtlich
- Funktioniert mit eigenen Git-Servern
- [Homepage](#)

# GITHUB DESKTOP

The screenshot displays the GitHub Desktop application window. The top menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu, the 'Current repository' is set to 'plex' and the 'Current branch' is 'development'. A 'Fetch origin' button indicates the last fetch was 'just now'.

The main interface is divided into three sections:

- Left Panel (Commit History):** Lists recent commits with their titles, authors, and dates. The selected commit is 'new build script' by Ottmar Gobrecht, committed on Sep 2, 2019.
- Center Panel (File Changes):** Shows a list of 10 changed files for the selected commit. The files include README.md, package.json, plex\_install.sql, PLEX.pkb, PLEX.pks, src\build.js, lib\as\_zip.sql, lib\as\_zip.txt, src\plex\_install.sql, and src\plex\_uninstall.sql. Each file has an icon indicating its type (e.g., README, package, SQL, JavaScript, text) and a status icon (yellow for new, blue for modified, green for added).
- Right Panel (Diff View):** Displays the diff for the selected file, 'PLEX.pks'. It shows the changes between the current branch and the previous commit, highlighting the addition of a new section titled 'PL/SQL Export Utilities'.

The diff view for 'PLEX.pks' shows the following changes:

```
@@ -1,4 +1,4 @@
-<!-- DO NOT EDIT THIS FILE DIRECTLY - it is generated from source file PLEX.pks -->
+<!-- DO NOT EDIT THIS FILE DIRECTLY - it is generated from source file src/PLEX.pks -->

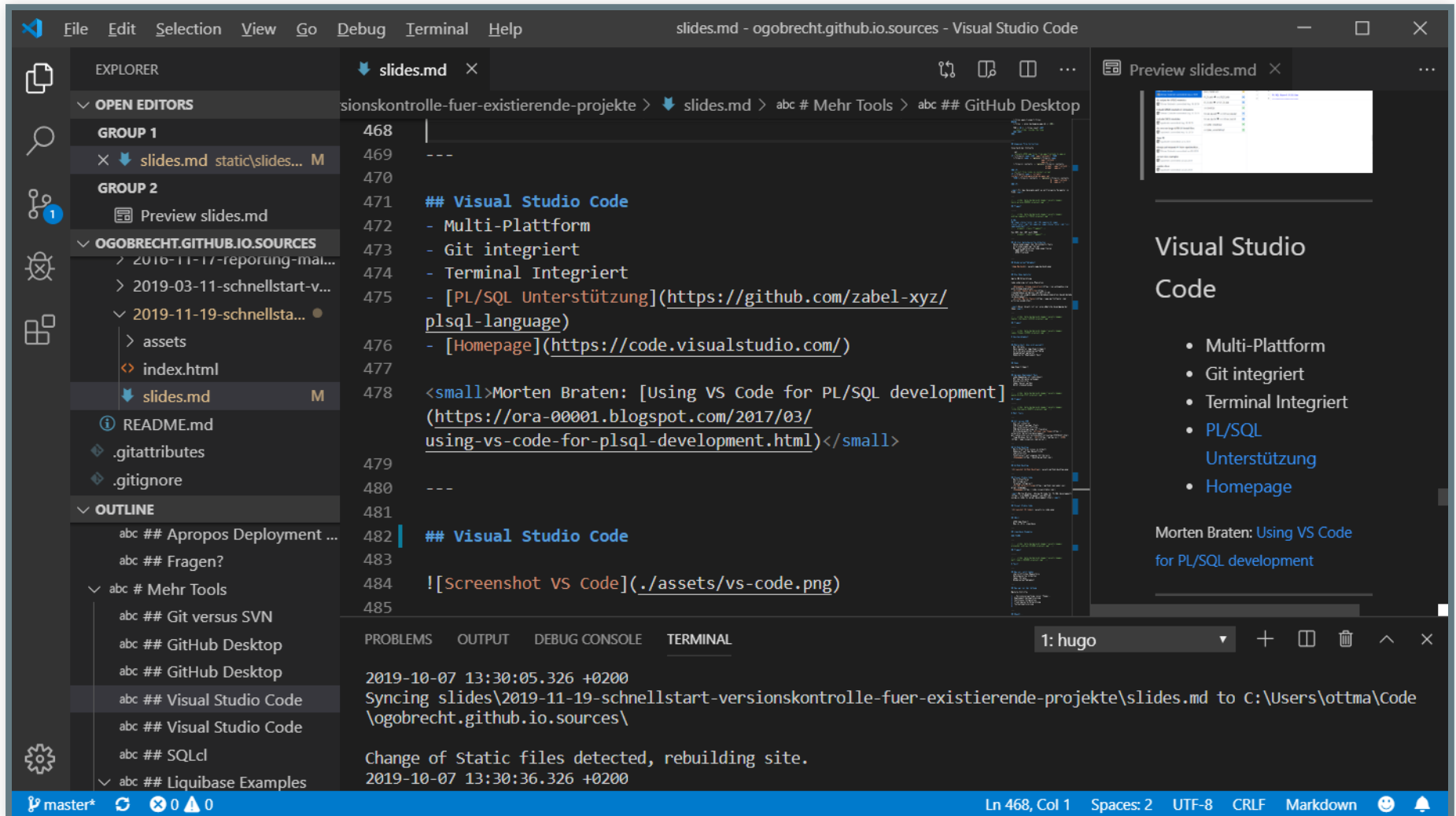
PL/SQL Export Utilities
=====
```

# VISUAL STUDIO CODE

- Multi-Plattform
- Git integriert
- Terminal Integriert
- [PL/SQL Unterstützung](#)
- [Homepage](#)

Morten Braten: [Using VS Code for PL/SQL development](#)

# VISUAL STUDIO CODE



# LIQUIBASE

*„Database schema change management“*

- In SQLcl integriert (v19.2+)
- Jeff Smith: [Liquibase and SQLcl](#)
- [www.liquibase.org](http://www.liquibase.org)

**FAZIT**



# DIE REISE HAT BEGONNEN

1. ~~Versionsverwaltung~~
2. Deployment-Automatisierung
3. Continuous Integration
4. Trunk-basierte Entwicklung
5. Testautomatisierung
6. ...

Siehe Anhang „DevOps“

# WEITERGEDACHT

Versionskontrolle nach DevOps meint  
„Alle Produktionsartefakte“

- ~~Anwendungscode~~
- Build-Skripte
- Deployment Pipelines
- Systemkonfigurationen
- ...
- Kurz: Infrastruktur als Code

Übrigens: Diese Folien sind auch Code - geschrieben in Markdown

# LESESTOFF 1

- Ottmar Gobrecht: **PLEX - PL/SQL Export Utilities & Schnellstart - Versionskontrolle für existierende Oracle-Projekte**
- Samuel Nitsche: **There is no clean (database) development without Version Control & “One does not simply update a database” – migration based database development**

# LESESTOFF 2

- Blain Carter: [Tips to help PL/SQL developers get started with CI/CD & CI/CD for Database Developers – Export Database Objects into Version Control](#)
- Denis Savenko: [Oracle APEX and ORDS deployments automation](#)

# LESESTOFF 3

- Martin Fowler: [Evolutionary Database Design](#)
- Jeff Smith: [Liquibase and SQLcl](#)
- Antti Kirmanen: [Git vs. Subversion \(SVN\): Welches Versionskontrollsystem sollten Sie nutzen?](#)
- Morten Braten: [Using VS Code for PL/SQL development](#)

**THE END**

# ANHANG - DEVOPS

# DEVOPS - STATISTIKEN

Leistungsstarke gegenüber leistungsschwachen Unternehmen laut [State of DevOps Report 2017](#)

- 46-mal häufigere Code Deployments
- 440-mal schneller von Commit zu Deployment
- 96-mal schnellere Wiederherstellung nach Ausfällen
- 5-mal niedrigere Ausfallrate bei Änderungen



# KANN MAN DAS GLAUBEN?

Buchtip: [Das Mindset von DevOps - Accelerate](#)

- Performance der Softwarebereitstellung
- 24 Schlüsselkompetenzen in 5 Kategorien
- Wissenschaftlich belegt

Der Kern: Auf Kompetenzen, nicht auf Reife fokussieren.  
Verbesserungen kontinuierlich vorantreiben.

# DIE KOMPETENZ-KATEGORIEN

1. Continuous Delivery
2. Architektur
3. Produkt und Prozess
4. Lean Management und Monitoring
5. Kultur

# CONTINUOUS DELIVERY-KOMPETENZEN

## Die ersten Schritte

1. Versionsverwaltung
2. Deployment-Automatisierung
3. Continuous Integration
4. Trunk-basierte Entwicklung
5. Testautomatisierung
6. ...

# KULTURELLE KOMPETENZEN - BEISPIEL ;-)

Kapitel 11: Führungskräfte und Manager, Transformationale Führung

*„Ermutigen Sie Ihre Belegschaft,  
mindestens einmal im Jahr technische  
Konferenzen zu besuchen und das dort  
Gelernte für das gesamte Team  
zusammenzufassen.“*

# LESESTOFF

- Buchtip Heise Developer: [Das Mindset von DevOps - Accelerate - 24 Schlüsselkompetenzen, um leistungsstarke Technologieunternehmen zu entwickeln und zu skalieren](#)
- Gareth Rushgrove: [Macht DevOps Unternehmen erfolgreicher?](#)
- Puppet: State of DevOps Report [2017](#), [2018](#), [2019](#)