Laboratory Enterprise Database Systems	
Laboratory 02	**National Language Support (NLS)** ***Language, territory, dates, numbers and currencies in Oracle XE***

Laboratory preparation

Requirements
Docker Desktop with the Docker Engine available and running
Oracle Database 21c XE container based on `gvenzl/oracle-xe:21`
Docker terminal and SQL Developer or Visual Studio Code with the Oracle extension
Access to the `SYSTEM` account and the `XEPDB1` database service

Introduction and theoretical background

Purpose. In this laboratory you will examine how Oracle Database presents dates, numbers, currencies and text to users working in different countries. You will connect to Oracle XE running inside Docker, prepare a small table, change National Language Support (NLS) settings and compare the results. No separate database account is required. All database exercises use `SYSTEM` in the `XEPDB1` pluggable database. You will learn more about pluggable databases in future laboratory sessions. For now, think of a pluggable database as a separate database with its own users and database objects.

1. What is a database, a table and a session?

A **database management system (DBMS)** is software that stores data, allows applications to read and change that data, and controls access. Oracle Database is a DBMS. A **table** consists of rows and columns. In this exercise, one row represents one employee. Columns store a person's name, employment date and salary. Each column has a defined **data type**, such as `VARCHAR2` for text, `DATE` for a date and time, and `NUMBER` for numeric values.

The **SQL** language is used to work with the data. A `SELECT` statement reads data, `CREATE TABLE` defines a table, `INSERT` adds a row and `ALTER SESSION` changes settings of the current connection. A **session** is created when a user successfully logs in. Two SQL Developer worksheets are not automatically two different sessions, whereas two separate database connections usually are. Session identity matters in this laboratory because NLS changes do not automatically propagate to other connections.

Oracle Database 21c XE uses a **container database (CDB)** and a **pluggable database (PDB)**. The default CDB is named `XE`, and the pre-created PDB is `XEPDB1`. You will connect specifically to `XEPDB1`. An Oracle PDB is not the same thing as a Docker container. A PDB is an Oracle database component, while a Docker container is an isolated operating environment running Oracle software.

> **The account for this laboratory:** `SYSTEM` is an administrative Oracle account with extensive privileges. It is not a normal application account. It is used here only to simplify a controlled classroom experiment. In production, application tables should be created in dedicated schemas with restricted privileges. Connect as `SYSTEM` using the *Default* role, not `SYSDBA`.

2. Why is National Language Support necessary?

The same information can be presented differently in different countries. The value representing an amount of money may appear as `1,250.50` or `1.250,50`. A date may be displayed as `2026-11-05`, `05.11.2026` or as a sentence containing the name of a weekday and a month. Currency signs, the language of month names and calendars may also differ.

Oracle calls the mechanisms responsible for many of these settings **National Language Support (NLS)**. NLS settings govern language-dependent behavior and default formatting. This is useful for international applications, reporting and data exchange. It does **not** mean that a number or a date changes every time its display format is changed. Oracle stores values using their data types, while the client or an SQL conversion function may show these values using a chosen format.

The following comparison is central to the laboratory.

Stored value	Possible presentation
`NUMBER(9,2)` containing `1250.50`	`1,250.50`, `1.250,50`, `PLN 1 250,50`
`DATE` containing 5 Nov 2026	`05.11.2026`, `11/05/2026`, `Thursday 05 November 2026`

3. Where do Oracle NLS settings come from?

NLS information can be inspected at three levels. **Database settings** describe values recorded for the database. **Instance settings** represent settings for a running Oracle instance. **Session settings** are the values effective for the connected client. The following Oracle views expose these levels.

View	Meaning
`NLS_DATABASE_PARAMETERS`	Values associated with the database, including the database character set
`NLS_INSTANCE_PARAMETERS`	Values associated with the running instance
`NLS_SESSION_PARAMETERS`	NLS values in the session executing the query

These views do not necessarily contain identical sets of parameter names. In particular, the database character set is not a normal modifiable session character-set parameter. Session values can depend on the client and on the session's `ALTER SESSION` statements. If two people obtain different displays from the same database, their connection settings may explain the difference.

```
SELECT parameter, value
FROM nls_session_parameters
ORDER BY parameter;
```

4. NLS settings used in this laboratory

Parameter	What it controls
`NLS_LANGUAGE`	General session language, including defaults for some language-dependent operations
`NLS_TERRITORY`	Regional conventions and defaults for date, currency and number formatting
`NLS_DATE_FORMAT`	Default format when a `DATE` is converted to text
`NLS_DATE_LANGUAGE`	Language of spelled-out weekday and month names
`NLS_TIMESTAMP_FORMAT`	Default text format of `TIMESTAMP` without time-zone information
`NLS_NUMERIC_CHARACTERS`	Decimal character followed by grouping character
`NLS_CURRENCY`	Local currency text used by the `L` number-format element
`NLS_ISO_CURRENCY`	Territory that determines the ISO currency code used by `C`
`NLS_CALENDAR`	Calendar system used for certain date operations and date presentations

Language is not territory. Selecting French for `NLS_LANGUAGE` does not mean that the monetary conventions must be those of France. A company may use a French-language application with United States regional

formatting. Likewise, NLS_DATE_LANGUAGE can be specified independently to display month names in a selected language.

5. Display formats and explicit conversions

The TO_CHAR function converts a date or number into text. Its format model determines the desired output. The TO_DATE function converts text into an Oracle DATE. An explicit format model usually makes an SQL statement easier to understand and more portable than relying on a client's current settings.

```
SELECT TO_CHAR(DATE '2026-11-05', 'YYYY-MM-DD') AS iso_date
FROM dual;

SELECT TO_CHAR(1250.5, '9G999D99') AS amount
FROM dual;
```

The special table DUAL is used to evaluate an expression without reading an application table. In number formats, D is the current decimal separator, G is the grouping separator, L is the local currency symbol or text and C is an ISO currency code. In date formats, DD means day, MM means month number, MON means an abbreviated month name, MONTH a full month name, and YYYY a four-digit year. HH24, MI and SS describe hours, minutes and seconds.

DATE contains a date and time to whole seconds. TIMESTAMP can additionally contain fractional seconds. TIMESTAMP WITH TIME ZONE also includes time-zone information. For this reason SYSTIMESTAMP, CURRENT_TIMESTAMP and LOCALTIMESTAMP are not identical. The ordinary NLS_TIMESTAMP_FORMAT setting does not control every time-zone-aware display.

6. Sessions, data storage and character sets

ALTER SESSION SET ... changes the effective setting of *one connected session*. It does not permanently change the database for everybody. When a new connection is created, that session receives its own initial settings. An existing connection retains its changes until they are changed again or the session ends.

A **character set** defines which characters can be represented in the database and how they are encoded. Oracle stores its database character set in NLS_CHARACTERSET, visible in NLS_DATABASE_PARAMETERS. This differs from the operating system's console code page and from NLS display formats. Changing a Windows terminal's environment variable does not change the database character set. Character-set migration is an administrative operation outside this laboratory.

7. Reading and executing the examples

SQL statements in the boxed listings can be copied into SQL*Plus, Oracle SQL Developer or the Oracle extension for VS Code. In SQL*Plus, the semicolon at the end completes a SQL statement. SQL Developer or VS Code can execute a selected statement from a worksheet. SQL*Plus commands such as SHOW USER are not standard SQL, so a graphical client's SQL worksheet may handle them differently. The exercises therefore normally include equivalent SELECT statements.

8. Changes you make and changes you only see

Three different actions will be performed in the exercise. Connecting with a username creates a session. Creating the sample table and inserting data changes database objects and rows. Changing an NLS session parameter changes the presentation rules of a particular connection. Understanding the difference prevents misleading conclusions after you reconnect.

Action	What happens
`SELECT`	Reads or calculates a result without modifying a stored table
`CREATE TABLE`	Creates a persistent database object in the current schema
`INSERT` and `COMMIT`	Add rows and save the transaction
`ALTER SESSION`	Changes one session's settings, not everyone's settings
Disconnecting	Ends the session, but does not delete a committed table

A useful experiment therefore has four steps. First, observe the current setting or data value. Second, change one variable. Third, repeat the same query and compare its output. Finally, open an independent session or restore the original setting to find out which changes were temporary. If several settings are changed at once, the result may be harder to interpret because more than one parameter could explain the difference.

Do not confuse **formatting** with **currency conversion**. Showing the number `1250.50` with a dollar, euro or yen label does not recalculate its amount using an exchange rate. A currency format element adds a label according to the selected NLS setting. It is still the same stored number. For the same reason, a different calendar representation of a `DATE` does not mean that Oracle moved the employee's hiring date.

Connecting to Oracle XE in Docker

1. Docker container and the first startup

Docker runs software in containers. A **Docker image** is a prepared software package, while a **Docker container** is a running or stopped instance created from that image. The `gvenzl/oracle-xe:21` image provides Oracle Database 21c XE. Docker Desktop and its engine must be running before any connection to Oracle can succeed.

The instructor's startup command, without any additional published ports, is:

```
docker run -d --name oracle-xe --restart unless-stopped -p 1521:1521 -e ORACLE_PASSWORD=Oracle123 -v
    oracle-xe-data:/opt/oracle/oradata gvenzl/oracle-xe:21
```

The same command divided into colour-underlined components:

```
docker run -d -name oracle-xe -restart unless-stopped -p 1521:1521 -e ORACLE_PASSWORD=Oracle123
-v oracle-xe-data:/opt/oracle/oradata gvenzl/oracle-xe:21
```

- **`docker run`, `-d`, `-name`** create a new container, start it in the background and give it the name `oracle-xe`.
- **`-restart unless-stopped`** restart the container after an engine restart unless it has been explicitly stopped.
- **`-p 1521:1521`** publish host port `1521` and forward connections to port `1521` inside the container. The left number refers to the computer, the right number to the container. Publishing a port does not create an Oracle listener or a database service. Publishing a port does not create an Oracle listener or a database service. These topics will be covered in future laboratory lessons.
- **`-e ORACLE_PASSWORD=Oracle123`** provide the initial password for the built-in `SYS` and `SYSTEM` accounts. The environment variable is used when the database is first initialized, not as a password reset on every restart.
- **`-v oracle-xe-data:/opt/oracle/oradata`** mount a named Docker volume so that the database files remain available after the container is deleted and recreated using the same volume.
- **`gvenzl/oracle-xe:21`** select the image and its Oracle 21c tag.

> **Do not execute `docker run` again if the `oracle-xe` container already exists.** Use `docker start oracle-xe` to start an existing stopped container. The default credentials in this guide correspond to the provided classroom command. A reused database volume may retain an earlier password.

Open Windows PowerShell or the Windows terminal and check:

```
docker ps -a
docker start oracle-xe
docker logs oracle-xe
```

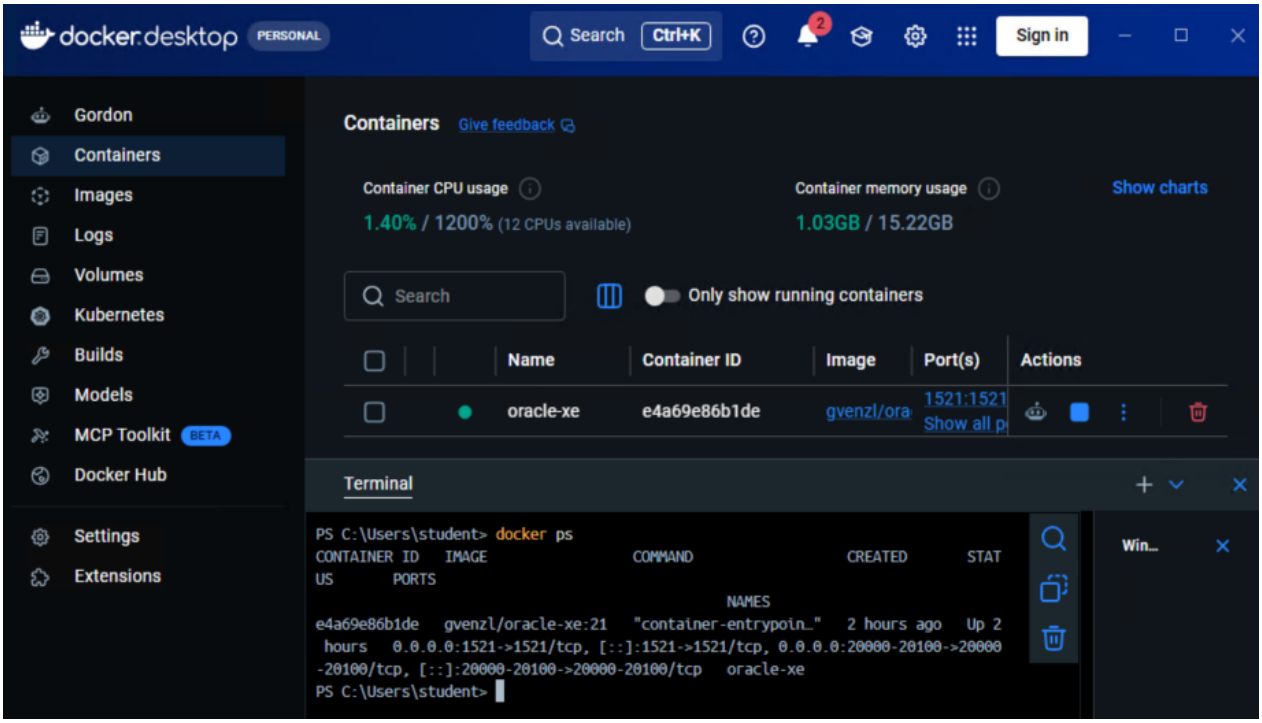


Figure 1: Docker Desktop app

2. Console mode: shell inside the Docker container

This mode uses SQL*Plus installed inside the Oracle image. Run the first command in **Windows PowerShell** or another terminal on the host computer.

```
docker exec -it oracle-xe bash
```

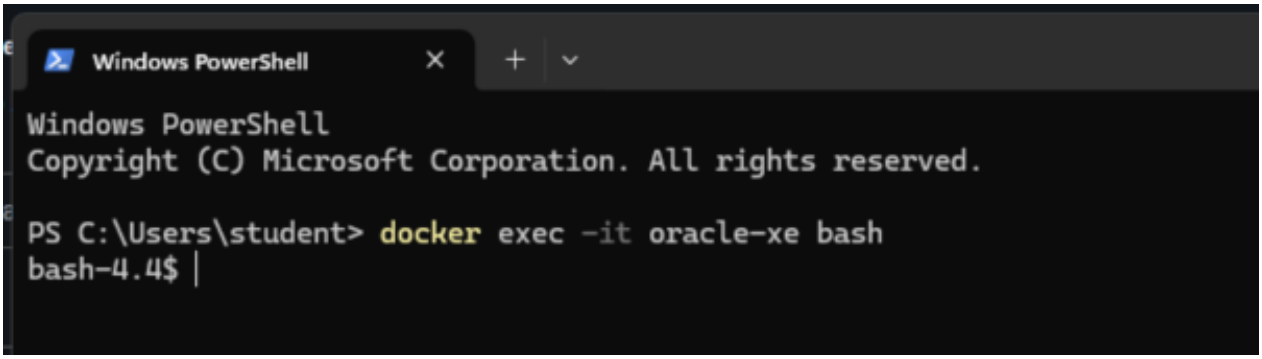


Figure 2: Linux shell

You are now inside the Linux shell of the running container. Start SQL*Plus and connect directly to the PDB service. Type the password when prompted (Oracle123).

```
sqlplus system@//localhost:1521/XEPDB1
```

`system` is the account, `localhost` is the computer name from the container's perspective, `1521` is the listener port and `XEPDB1` is the Oracle database service name. The initial password in this laboratory is `Oracle123`. The password is not visible as you type it.

Once connected, run:

```
SELECT USER AS connected_user, SYS_CONTEXT('USERENV','CON_NAME') AS container_name FROM dual;
```

The expected user is `SYSTEM`, and the expected PDB is `XEPDB1`. To leave SQL*Plus enter `EXIT`. To leave the Linux shell enter `exit`. A shorter one-step alternative from the Windows host is `docker exec -it oracle-xe sqlplus system@//localhost:1521/XEPDB1`.

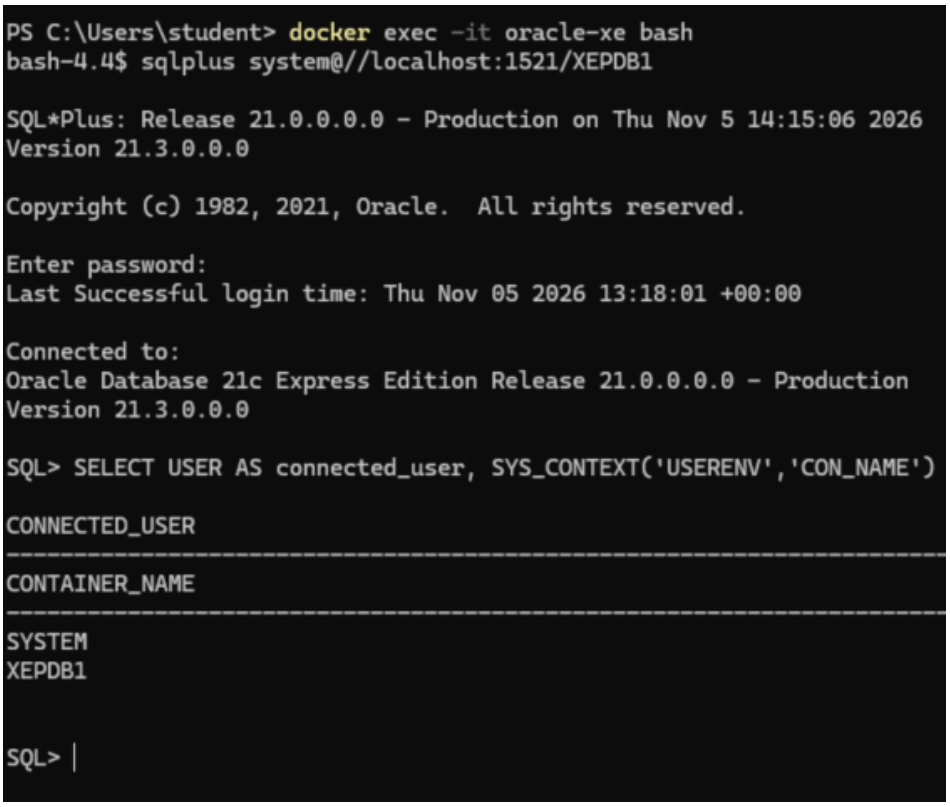


Figure 3: System user session in SQL*Plus using Docker

3. Graphical mode: Oracle SQL Developer

Start the installed Oracle SQL Developer application. In the **Connections** panel choose **New Connection** or click the plus icon. Enter these values. A connection name is only a label stored in the client. You can choose your own connection name, but the recommended naming convention is `Container-Service`, for example `XE-XEPDB1`.

Field	Value
Connection Name	`XE-SYSTEM-NLS`
Username	`SYSTEM`
Password	`Oracle123` for a newly initialized database
Role	**Default**, not SYSDBA
Connection Type	`Basic`
Hostname	`localhost` or `127.0.0.1`
Port	`1521`
Connect by / Type	**Service name**
Service name	`XEPDB1`

Click **Test** and check that the status indicates a successful connection. Then click **Connect**. Open an SQL worksheet and run the identity query above using the *Run Statement* command (usually `Ctrl+Enter`). When executing several SQL statements, select the intended text or use the script execution control.

4. Graphical mode: Visual Studio Code

Visual Studio Code does not provide an Oracle connection by itself. Install the **Oracle SQL Developer Extension for VS Code** first. In Visual Studio Code open **Extensions** with `Ctrl+Shift+X`, search for `SQL Developer`, select the extension published by Oracle and choose **Install**. After installation, open its Oracle SQL Developer panel and create a new database connection.

Use the same connection properties as in the SQL Developer table. Select **SYSTEM**, password `Oracle123`, host `localhost`, port `1521`, **Service name** `XEPDB1` and **Default** role. Test or save the connection, open a SQL worksheet and execute the identity query. The exact labels may vary between extension releases.

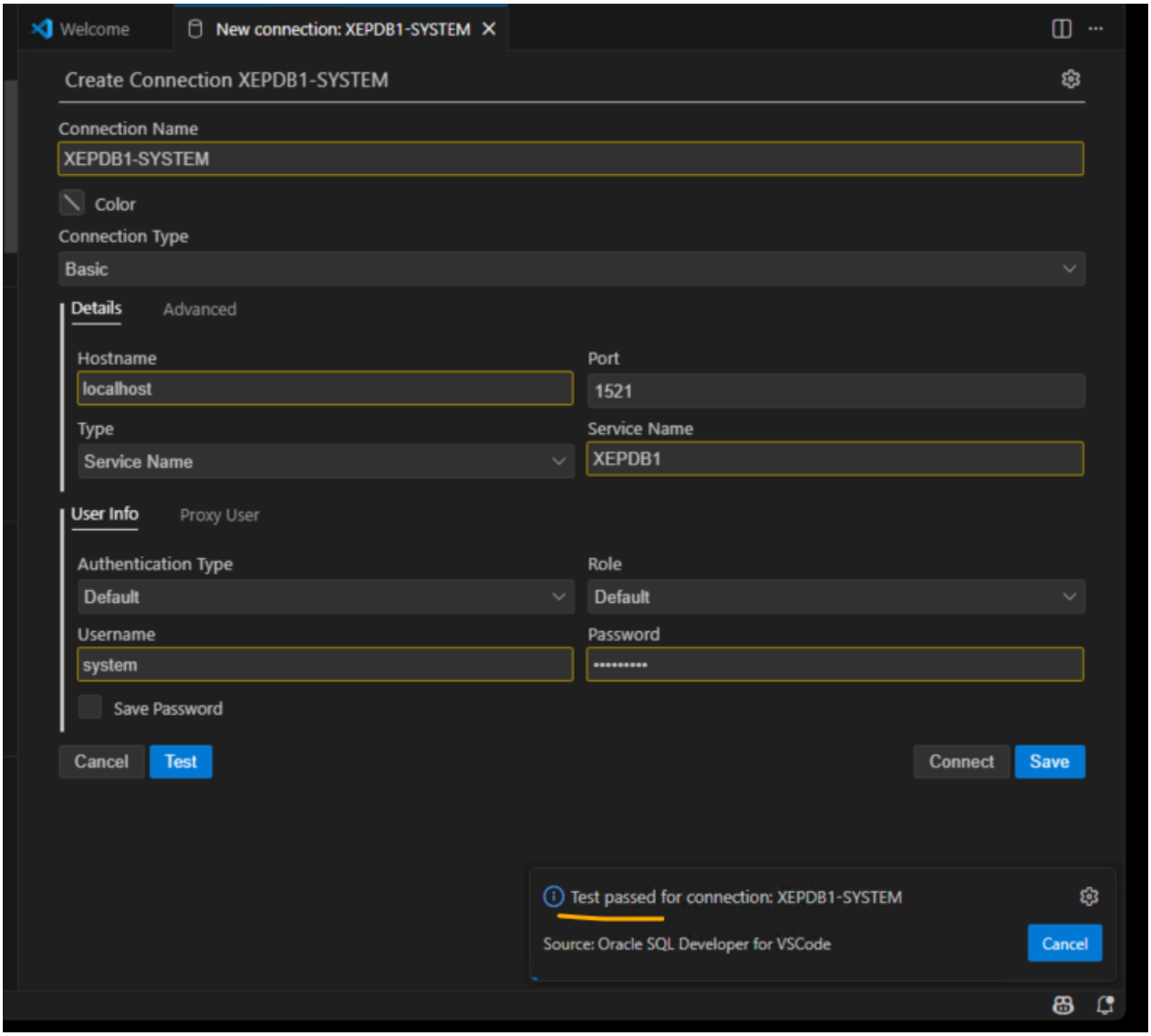


Figure 4: Oracle SQL Developer connection

5. How to read the login and what the password and port mean

The SQL*Plus login string can be divided into separate pieces:

```
sqlplus system@//localhost:1521/XEPDB1
```

Part	Interpretation
`sqlplus`	Starts the command-line Oracle client
`system`	Oracle database account, not the Windows or Linux login
`@//localhost`	Easy Connect syntax, target host for the listener
`:1521`	TCP listener port
`/XEPDB1`	Name of the PDB service to connect to
Password prompt	Database password configured by `ORACLE_PASSWORD` on first initialization

The Docker command and the SQL login are different operations. Docker creates the environment and publishes the port. SQL*Plus connects to Oracle's listener through that port and authenticates to an Oracle account. A database user, an Oracle database service and a Docker container therefore have three different

names. In graphical clients you enter the same connection pieces into separate fields rather than writing one connection string.

6. Simple diagnostics

If a connection fails, first check whether Docker Desktop is running and the container is started with `docker ps`. Then check `docker logs oracle-xe` and confirm that port `1521` is published. A listener error often means the service is not yet ready or the host or port is wrong. An authentication error often means the username or password is wrong. A service-name error can mean that `XE` was entered instead of `XEPDB1`, or that the requested PDB service is not available. Remember that the `ORACLE_PASSWORD` environment variable does not overwrite the account password in an already initialized volume.

Laboratory tasks

Working rule. Use `SYSTEM` connected to `XEPDB1` for every task. Do not change instance-wide settings, restart the database to apply NLS changes or modify the database character set. Start each task by checking the session if you have opened another connection. Unless the task explicitly asks you to work independently, the examples provide the commands to run and the observations to make.

Evidence rule. Screenshots must show a current result from your own running database.

Part I. Environment and starting data

1. Verify the user, container and date.

Connect through SQL*Plus or one of the graphical clients. Run the query and identify each returned value.

```
SELECT USER AS connected_user,
       SYS_CONTEXT('USERENV','CON_NAME') AS container_name,
       CURRENT_TIMESTAMP AS current_time
FROM dual;
```

Record: the account name, PDB service and current timestamp. Explain why `XE` and `XEPDB1` are not interchangeable names.

2. Compare NLS values across the database, instance and session.

Compare selected parameters in three views. A missing value is not automatically an error.

```
SELECT s.parameter,
       s.value AS session_value, i.value AS instance_value, d.value AS database_value
FROM nls_session_parameters s
LEFT JOIN nls_instance_parameters i
  ON i.parameter = s.parameter
LEFT JOIN nls_database_parameters d
  ON d.parameter = s.parameter
WHERE s.parameter IN
  ('NLS_LANGUAGE', 'NLS_TERRITORY',
   'NLS_DATE_FORMAT', 'NLS_NUMERIC_CHARACTERS',
   'NLS_CURRENCY', 'NLS_CALENDAR')
ORDER BY s.parameter;
```

Record: at least two parameters with equal or different values across levels. State which column describes your current connection.

3. Create a small employee table and insert test rows.

Create the NLS_EMPLOYEES table in XEPDB1 as SYSTEM. The DATE literals below do not depend on the NLS date format. Replace YOUR_NAME and YOUR_SURNAME with your details.

```
CREATE TABLE nls_employees (
  first_name VARCHAR2(30),
  last_name  VARCHAR2(40),
  hire_date  DATE NOT NULL,
  salary     NUMBER(9,2)
) TABLESPACE USERS;

INSERT INTO nls_employees VALUES
  ('YOUR_NAME', 'YOUR_SURNAME', DATE '2005-09-30', 7700.50);
INSERT INTO nls_employees VALUES
  ('Adam', 'Nowak', DATE '2019-04-15', 1250.75);
INSERT INTO nls_employees VALUES
  ('Anna', 'Kowalska', DATE '2024-12-03', 8900.25);
COMMIT;

SELECT * FROM nls_employees;
```

Record: three rows, including your own name. If the table already exists from an earlier attempt, reuse it instead of creating duplicate tables.

4. Compare a stored value with formatted text.

Run both queries. Notice that TO_CHAR returns characters rather than the original numeric or date data type.

```
SELECT last_name, hire_date, salary FROM nls_employees;

SELECT last_name,
       TO_CHAR(hire_date, 'YYYY-MM-DD') AS formatted_date,
       TO_CHAR(salary, 'L99G999D99') AS formatted_salary
FROM nls_employees;
```

Record: one date and one salary in both representations. Which SQL expression explicitly controls the format?

Part II. Language and territory

5. Change the language of the current session.

Execute the following sequence and inspect the output after each change. The language controls the textual elements of the date, not the stored DATE value.

```
ALTER SESSION SET NLS_LANGUAGE = 'FRENCH';
SELECT TO_CHAR(hire_date, 'FMDay DD FMMonth YYYY')
FROM nls_employees;

ALTER SESSION SET NLS_LANGUAGE = 'GERMAN';
SELECT TO_CHAR(hire_date, 'FMDay DD FMMonth YYYY')
FROM nls_employees;

ALTER SESSION SET NLS_LANGUAGE = 'POLISH';
SELECT TO_CHAR(hire_date, 'FMDay DD FMMonth YYYY')
FROM nls_employees;
```

Record: the day and month names in at least two languages.

6. Test three regional conventions for salaries.

Set the territory to FRANCE, GERMANY and AMERICA in turn. Run the two queries after each change.

```
ALTER SESSION SET NLS_TERRITORY = 'FRANCE';

SELECT TO_CHAR(salary, 'L99G999D99') AS amount
FROM nls_employees;

SELECT parameter, value
FROM nls_session_parameters
WHERE parameter IN ('NLS_TERRITORY',
                    'NLS_CURRENCY',
                    'NLS_NUMERIC_CHARACTERS');
```

Record: the currency indicator and separators for all three territories. Explain which part is selected by NLS_TERRITORY.

7. Combine a language and a different territory.

Test a French-language session using the American territory, then a Polish-language session using the French territory. Inspect the parameter values and format one salary and one long date in each combination.

```
ALTER SESSION SET NLS_LANGUAGE = 'FRENCH';
ALTER SESSION SET NLS_TERRITORY = 'AMERICA';

SELECT parameter, value FROM nls_session_parameters
WHERE parameter IN ('NLS_LANGUAGE','NLS_TERRITORY',
                    'NLS_DATE_LANGUAGE','NLS_CURRENCY');

SELECT TO_CHAR(hire_date, 'FMDay DD FMMonth YYYY')
       AS long_date,
       TO_CHAR(salary, 'L99G999D99') AS salary_text
FROM nls_employees;
```

Record: the difference between a language choice and a territorial choice.

8. Formatting a date in a selected language.

The TO_CHAR() function allows you to specify the language used to format a date without changing the session settings. Execute the following statement:

```
SELECT TO_CHAR(SYSDATE, 'DD FMMonth YYYY', 'NLS_DATE_LANGUAGE=FRENCH') AS FRENCH_DATE
FROM DUAL;
```

Record: Observe the differences between the French and German date formats. Does this operation change the NLS_DATE_LANGUAGE parameter of the current session?

9. Check what Oracle reports about the session language.

Compare the information in USERENV('LANGUAGE') with the explicit values from NLS_SESSION_PARAMETERS. Repeat after switching from FRENCH to GERMAN.

```
SELECT USERENV('LANGUAGE') AS language_info FROM dual;

SELECT parameter, value FROM nls_session_parameters
WHERE parameter IN ('NLS_LANGUAGE','NLS_TERRITORY');
```

Record: which parts change. Explain why the character set reported by USERENV('LANGUAGE') must not be treated as proof of a client's full NLS_LANG setting.

Part III. Dates, timestamps and session scope

10. Compare three date display formats.

Set NLS_DATE_FORMAT successively to DD.MM.YYYY, DD-MON-RR and YYYY/MM/DD. Run the query after each setting.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD.MM.YYYY';
SELECT last_name, hire_date FROM nls_employees;
```

Record: one result for each date format. What is the difference between RR and YYYY in these examples?

11. Control the language of month names independently.

Test FRENCH, ITALIAN and GERMAN as NLS_DATE_LANGUAGE. Keep a date format with spelled-out names.

```
ALTER SESSION SET NLS_DATE_LANGUAGE = 'FRENCH';
SELECT TO_CHAR(hire_date, 'FMDay DD FMMonth YYYY')
FROM nls_employees;
```

Record: the names of weekdays and months. Which parameter controls these words more directly, NLS_TERRITORY or NLS_DATE_LANGUAGE?

12. Examine timestamp formats.

Read three current-time expressions. Then change NLS_TIMESTAMP_FORMAT and compare the output again.

```
SELECT SYSTIMESTAMP, CURRENT_TIMESTAMP,
       LOCALTIMESTAMP FROM dual;

ALTER SESSION SET NLS_TIMESTAMP_FORMAT =
  'YYYY-MM-DD HH24:MI:SS.FF3';

SELECT LOCALTIMESTAMP,
       CAST(SYSTIMESTAMP AS TIMESTAMP) AS local_cast,
       SYSTIMESTAMP AS with_timezone
FROM dual;
```

Record: which columns follow the ordinary timestamp format and which retain time-zone formatting.

13. Prove that NLS changes are session-specific.

In your first session set NLS_DATE_FORMAT to YYYY/MM/DD and NLS_LANGUAGE to FRENCH. Without closing it, create a second, independent database connection as SYSTEM to XEPDB1. Query the same settings in both sessions.

```
SELECT parameter, value FROM nls_session_parameters
WHERE parameter IN ('NLS_LANGUAGE','NLS_DATE_FORMAT');
```

Record: the values in session 1 and session 2. Explain why opening another worksheet inside the same physical connection may not be a sufficient test.

Part IV. Numbers, currencies, calendars and encodings

14. Change the decimal and grouping separators.

Test both sequences of NLS_NUMERIC_CHARACTERS, where the *first* character is the decimal character and the *second* is the group character.

```
ALTER SESSION SET NLS_NUMERIC_CHARACTERS = '.,';
SELECT salary, TO_CHAR(salary,'999G999D99')
FROM nls_employees;

ALTER SESSION SET NLS_NUMERIC_CHARACTERS = ',.';
SELECT salary, TO_CHAR(salary,'999G999D99')
```

```
FROM nls_employees;
```

Record: both text formats and the unchanged numeric result of `salary`.

15. Compare local currency and ISO currency formatting.

Use the local indicator before and after the number. Then select an ISO currency territory and compare `L` with `C`.

```
ALTER SESSION SET NLS_CURRENCY = 'PLN';
ALTER SESSION SET NLS_ISO_CURRENCY = 'JAPAN';

SELECT TO_CHAR(salary, 'L99G999D99') AS local_first,
       TO_CHAR(salary, '99G999D99L') AS local_last,
       TO_CHAR(salary, 'C99G999D99') AS iso_code
FROM nls_employees;
```

Record: why the `L` and `C` format elements return different currency labels. No currency conversion occurs.

16. Compare calendar systems.

Check `NLS_CALENDAR`, then test `GREGORIAN`, `THAI BUDDHA` and `PERSIAN`. Run the query after each setting and finally restore the initial value.

```
SELECT value FROM nls_session_parameters
WHERE parameter = 'NLS_CALENDAR';

ALTER SESSION SET NLS_CALENDAR = 'THAI BUDDHA';
SELECT TO_CHAR(hire_date, 'YYYY-MM-DD') AS calendar_date
FROM nls_employees;
```

Record: whether the displayed year changes. Explain whether the stored `DATE` value was replaced.

17. Check the database character set and the client context.

Read the database character set, national character set and session's language information. Do not attempt to alter these character sets.

```
SELECT parameter, value FROM nls_database_parameters
WHERE parameter IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET');

SELECT USERENV('LANGUAGE') AS language_info FROM dual;
```

Record: both character set names. Explain why setting a different `NLS_LANGUAGE` or terminal environment variable does not re-encode the stored database.

Part V. Independent exercises

Complete the following exercises independently, without a complete example solution. You may use the SQL functions and commands introduced earlier. For each exercise, provide the SQL statements, actual results and a brief explanation. All exercises use the `NLS_EMPLOYEES` table unless otherwise stated.

18. One date in three languages.

Create *one SELECT statement* that displays the same `hire_date` in *French, German and Polish* in three separate columns.

Use the `TO_CHAR()` function with an explicit NLS conversion argument, without changing the current session settings.

Record: Are the displayed dates different? Do they represent the same date stored in the database?

19. Date formatting and sorting.

Display the employees' last names and hiring dates using the format `DD.MM.YYYY`. Sort the results by `hire_date` from the *oldest to the newest* employee.

Next, change the displayed date format to `YYYY-MM-DD` and execute the query again.

Record: Does changing the displayed date format affect the order of the records? Explain your result.

20. Numeric separators.

Configure the current session to display salary values with a *comma as the decimal separator* and a *dot as the thousands separator*.

Display the salaries using the format model `99G999D99`.

Next, *reverse the separators* and display the same values again.

Record: Compare both results. Has the actual numeric value of the salary changed?

21. Custom currency symbols.

Set `NLS_CURRENCY` to `PLN` and display the salaries with the currency label *at the end* of each value.

Next, change the currency label to another currency *(excluding EUR)* and display it *before* each value.

Record: Does changing the currency label convert the salary into another currency? Explain your answer.

22. Prepare a formatted employee report.

Create *one SELECT statement* that displays the following information from `NLS_EMPLOYEES`:

- Employee's first and last name
- Hiring date in the `DD.MM.YYYY` format
- Salary with two decimal places
- Currency label `PLN` after the salary

Sort the employees by salary in *descending order*. Use appropriate aliases for all displayed columns.

Record: Present the complete SQL statement and its result. Explain which functions and format models were used to prepare the report.

23. Session settings after reconnecting.

Check and record the current values of `NLS_LANGUAGE`, `NLS_TERRITORY` and `NLS_DATE_FORMAT` using `NLS_SESSION_PARAMETERS`.

Next, modify the current session as follows:

- Set `NLS_LANGUAGE` to `FRENCH`
- Set `NLS_TERRITORY` to `GERMANY`
- Set `NLS_DATE_FORMAT` to `DD.MM.YYYY`

Verify that the new values are active.

Then, *disconnect from the database* and establish a *new connection* using the same `SYSTEM` account and `XEPDB1` service.

Check the three NLS parameters again.

Record: Compare the parameter values before the changes, after the changes and after reconnecting. Were the modified settings preserved in the new session? Explain your observations.

Assessment and report requirements

The laboratory is assessed according to completed and correctly explained tasks, documented results and an instructor's live verification. Completing a task means executing the SQL, obtaining the expected type of result and being able to explain it. It does not mean merely copying the statement. Screenshots are part of a short report, not a replacement for a working Oracle session.

Requirements by grade

ECTS	Completed tasks	Minimum screenshots	Required live check
E	1–17	3	Connection, data, NLS settings, dates, timestamps, numbers, currencies, calendars and character sets
D	1–19	4	Above plus multilingual date formatting and date sorting
C	1–20	5	Above plus independent numeric separator configuration and comparison
B	1–22	6	Above plus currency formatting and an independent employee report
A	1–23	7	Above plus verification of NLS session settings after reconnecting, with interpretation
F / FX	Requirements for E not met	–	Work not accepted or correction required

The minimum screenshots are **3, 4, 5, 6 or 7 in total**, not additional counts for each grade. Include the completed task numbers and relevant results in the report. Screenshots must document actual SQL execution and its results. A screenshot taken from a connection tutorial is not counted as task evidence unless it includes a laboratory result. Do not show passwords in screenshots.

ECTS and Polish grading scale

ECTS	ECTS descriptor	Polish grade	Description
A	Excellent	**5.0** (very good)	Outstanding performance with only minor errors
B	Very Good	**4.5** (good plus)	Above-average work with a few errors
C	Good	**4.0** (good)	Reliable work overall, with some shortcomings
D	Satisfactory	**3.5** (satisfactory plus)	Acceptable work with notable gaps
E	Pass	**3.0** (satisfactory)	Minimum requirements met
F / FX	Fail	**2.0** (unsatisfactory)	Not passed or correction required

Keep the database client and Docker container available until the check has been completed. Outputs copied from another student's environment are not accepted.