Appendix

Appendix 1. Procedure to identify green, digital and green-digital with WIPO data

1. **Include docdb_family_id**: I have stored in my sql server a table called 'DGPT_WIPO_matching' that has the appln_id and the DGPTs category. I created a new table called 'DGPT_WIPO_matching_families' including the number of families associated with each apln_id.

```
CREATE TABLE `menendez`.`DGPT_WIPO_matching_families` (
 `docdb_family_id` INT NOT NULL,
 `appln_id` INT NOT NULL);

insert into menendez.DGPT_WIPO_matching_families (docdb_family_id, appln_id)
select m.docdb_family_id, a.appln_id
from menendez.DGPT_WIPO_matching a
join patstat2024b.tls201_appln m ON m.appln_id = a.appln_id;
```

2. Identification of green-digital: In this SQL script, I create a new table called new_green_digital_patents_wipo_new in the menendez schema to store information about a filtered subset of patent applications. The goal is to extract and store data on green and digital technology patent families (identified by CPC codes starting with Y02 or Y04) that have international relevance. Specifically, I identify patent families filed between 2000 and 2019 that have applications from multiple patent authorities, with at least one application filed at the European Patent Office (EP). This subset is selected using a Common Table Expression (CTE) called qualifying_families. You then join this with WIPO-matched application CPC classification data to extract detailed information—such as the docdb_family_id, appln_id, appln_auth, earliest_filing_year, and cpc_class_symbol—for only those applications that are part of these qualifying families and are classified in green or digital CPC categories.

```
CREATE TABLE `menendez`.`new_green_digital_patents_wipo_new` (
 `docdb_family_id` INT NOT NULL,
 `appln_id` INT NOT NULL,
 `appln_auth` VARCHAR(2),
 `earliest_filing_year` INT NOT NULL,
`cpc_class_symbol` VARCHAR(15)
);

INSERT INTO menendez.new_green_digital_patents_wipo_new (docdb_family_id, appln_id, appln_auth,
earliest_filing_year, cpc_class_symbol)

WITH qualifying_families AS (
 SELECT
  a.docdb_family_id
 FROM menendez.DGPT_WIPO_matching m
 JOIN patstat2024b.tls201_appln a ON m.appln_id = a.appln_id
 WHERE a.earliest_filing_year BETWEEN 2000 AND 2019
 GROUP BY a.docdb_family_id
 HAVING COUNT(DISTINCT a.appln_auth) > 1
   AND SUM(a.appln_auth = 'EP') > 0
)
```

```
-- Step 2: Select detailed records
SELECT
  a.docdb_family_id,
  m.appln_id,
  a.appln_auth,
  a.earliest_filing_year,
  b.cpc_class_symbol
FROM menendez.DGPT_WIPO_matching m
JOIN patstat2024b.tls201_appln a ON m.appln_id = a.appln_id
JOIN qualifying_families q ON a.docdb_family_id = q.docdb_family_id
JOIN patstat2024b.tls224_appln_cpc b ON m.appln_id = b.appln_id
WHERE a.earliest_filing_year BETWEEN 2000 AND 2019
and m.appln_id IN (
        SELECT t4.appln_id
        FROM patstat2024b.tls224_appln_cpc t4
        WHERE t4.cpc_class_symbol LIKE 'Y02%' OR t4.cpc_class_symbol LIKE 'Y04%'
    );
```

In this SQL script, I created a new table named new_green_digital_patents_wipo_new_abs in the menendez schema to enrich your previously filtered green and digital patents dataset with **English-language abstracts**. This new table extends the structure of new_green_digital_patents_wipo_new by adding two columns: appln_abstract (the full abstract text) and appln_abstract_lg (the language code of the abstract). Then, I populated this new table by selecting records from new_green_digital_patents_wipo_new and joining them with the tls203_appln_abstr table from PATSTAT, which contains application abstracts. The join ensures that only applications with available English ('en') abstracts are included.

```
CREATE TABLE `menendez`.`new_green_digital_patents_wipo_new_abs` (
  `docdb_family_id` INT NOT NULL,
  `appln_id` INT NOT NULL,
  `appln_auth` VARCHAR(2),
  `earliest_filing_year` INT NOT NULL,
  `cpc_class_symbol` VARCHAR(15),
  `appln_abstract` TEXT NOT NULL ,
`appln_abstract_lg` TEXT NOT NULL)
;
INSERT INTO menendez.new_green_digital_patents_wipo_new_abs (docdb_family_id, appln_id,
appln_auth,earliest_filing_year, cpc_class_symbol, appln_abstract_lg,appln_abstract )
SELECT f1.docdb_family_id, f1.appln_id, f1.appln_auth, f1.earliest_filing_year, f1.cpc_class_symbol,
f2.appln_abstract_lg, f2.appln_abstract FROM new_green_digital_patents_wipo_new f1
inner join patstat2024b.tls203_appln_abstr f2 on f2.appln_id=f1.appln_id
where f2.appln_abstract_lg like 'en';
```

3. **Identification digital**: I followed a similar procedure by creating two new tables called new_digital_patents_wipo_new, retrieving similar information as table new_green_digital_patents_wipo_new, but indicating **not to** extract and store data on green and digital technology patent families (identified by CPC codes starting with Y02 or Y04).

```
CREATE TABLE `menendez`.`new_digital_patents_wipo_new` (
  `docdb_family_id` INT NOT NULL,
  `appln_id` INT NOT NULL,
  `appln_auth` VARCHAR(2),
  `earliest_filing_year` INT NOT NULL,
`cpc_class_symbol` VARCHAR(15)
);

INSERT INTO menendez.new_digital_patents_wipo_new (docdb_family_id, appln_id, appln_auth,
earliest_filing_year, cpc_class_symbol)

WITH qualifying_families AS (
  SELECT
    a.docdb_family_id
  FROM menendez.DGPT_WIPO_matching m
  JOIN patstat2024b.tls201_appln a ON m.appln_id = a.appln_id
  WHERE a.earliest_filing_year BETWEEN 2000 AND 2019
  GROUP BY a.docdb_family_id
  HAVING COUNT(DISTINCT a.appln_auth) > 1
     AND SUM(a.appln_auth = 'EP') > 0
)

-- Step 2: Select detailed records
SELECT
  a.docdb_family_id,
  m.appln_id,
  a.appln_auth,
  a.earliest_filing_year,
  b.cpc_class_symbol
FROM menendez.DGPT_WIPO_matching m
JOIN patstat2024b.tls201_appln a ON m.appln_id = a.appln_id
JOIN qualifying_families q ON a.docdb_family_id = q.docdb_family_id
JOIN patstat2024b.tls224_appln_cpc b ON m.appln_id = b.appln_id
WHERE a.earliest_filing_year BETWEEN 2000 AND 2019
and m.appln_id IN (
          SELECT t4.appln_id
          FROM patstat2024b.tls224_appln_cpc t4
          WHERE t4.cpc_class_symbol NOT LIKE 'Y02%' OR t4.cpc_class_symbol NOT LIKE 'Y04%'
      )
AND a.docdb_family_id NOT IN (
        SELECT DISTINCT docdb_family_id FROM menendez.new_green_digital_patents_wipo_new
    );

SELECT * FROM menendez.new_digital_patents_wipo_new;

CREATE TABLE `menendez`.`new_digital_patents_wipo_new_RAND` (
  `docdb_family_id` INT NOT NULL,
  `appln_id` INT NOT NULL,
  `earliest_filing_year` INT NOT NULL
);
```

As there are a lot of records, I retrieve information for a random sample of **50,000 unique digital patent applications** into a new table called new_digital_patents_wipo_new_RAND. The data is selected from the existing table new_digital_patents_wipo_new, and I am using SELECT DISTINCT to ensure no duplicate combinations of docdb_family_id, appln_id, and earliest_filing_year are included. The ORDER BY RAND() function randomly shuffles the rows, and LIMIT 50000 ensures only a fixed number (50,000) of these shuffled, distinct records are inserted.

```
use menendez;
INSERT INTO menendez.new_digital_patents_wipo_new_RAND (docdb_family_id, appln_id,
earliest_filing_year)

SELECT DISTINCT docdb_family_id, appln_id, earliest_filing_year
 FROM menendez.new_digital_patents_wipo_new
 ORDER BY RAND()
 LIMIT 50000  ;
```

Finally, I created a new table called new_digital_patents_wipo_new_abs in the menendez schema to store **abstracts of a random sample of digital patents**, specifically those written in English. The table includes fields for the patent family ID (docdb_family_id), application ID (appln_id), filing year, and both the abstract text (appln_abstract) and its language (appln_abstract_lg). I populated this table by joining my previously created random sample (new_digital_patents_wipo_new_RAND) with the tls203_appln_abstr table from PATSTAT, which contains the actual abstract content. The WHERE clause filters for records where the abstract language is English ('en'), ensuring the resulting dataset is ready for English-language text analysis

```
CREATE TABLE `menendez`.`new_digital_patents_wipo_new_abs` (
 `docdb_family_id` INT NOT NULL,
 `appln_id` INT NOT NULL,
 `earliest_filing_year` INT NOT NULL,
 `appln_abstract` TEXT NOT NULL ,
`appln_abstract_lg` TEXT NOT NULL)
;
INSERT INTO menendez.new_digital_patents_wipo_new_abs (docdb_family_id, appln_id, earliest_filing_year,
appln_abstract_lg,appln_abstract )
SELECT f1.docdb_family_id,f1.appln_id,f1.earliest_filing_year, f2.appln_abstract_lg, f2.appln_abstract FROM
new_digital_patents_wipo_new_RAND f1
inner join patstat2024b.tls203_appln_abstr f2 on f2.appln_id=f1.appln_id
where f2.appln_abstract_lg like 'en';
```

4. **Identification green:** The identification of the green patents relies on information retrieved directly from patstat2024b. The query I run has two steps. First, I randomly choose 10.000 green patent families. I created a temporary result set of docdb_family_ids from the tls201_appln table, selecting families that have at least one application between **2000 and 2019**, that are classified under **green or digital CPC codes** (Y02% or Y04%), include applications filed in **more than one jurisdiction** (COUNT(DISTINCT appln_auth) > 1) and explicitly include at least one filing at the **European Patent Office (EP)** (checked using FIND_IN_SET in a GROUP_CONCAT of authorities).

   I extract detailed application-level data from those selected families—docdb_family_id, appln_id, appln_auth, and earliest_filing_year—by joining the full applications table again. I ensure that each application is still between **2000 and 2019** and classified as **green or digital** using a second filter on CPC codes.

```
WITH selected_families AS (
  SELECT t3.docdb_family_id
  FROM patstat2024b.tls201_appln t3
  WHERE t3.earliest_filing_year BETWEEN 2000 AND 2019
   AND t3.appln_id IN (
      SELECT t4.appln_id
      FROM patstat2024b.tls224_appln_cpc t4
      WHERE t4.cpc_class_symbol LIKE 'Y02%' OR t4.cpc_class_symbol LIKE 'Y04%'
    )
  GROUP BY t3.docdb_family_id
  HAVING COUNT(DISTINCT t3.appln_auth) > 1
    AND FIND_IN_SET('EP', GROUP_CONCAT(DISTINCT t3.appln_auth)) > 0
  ORDER BY RAND()
  LIMIT 10000
)

-- Step 2: Get all appln_id for those selected families
SELECT
  t1.docdb_family_id,
  t1.appln_id,
  t1.appln_auth,
  t1.earliest_filing_year
FROM patstat2024b.tls201_appln t1
JOIN selected_families sf ON t1.docdb_family_id = sf.docdb_family_id
WHERE t1.earliest_filing_year BETWEEN 2000 AND 2019
 AND t1.appln_id IN (
    SELECT t2.appln_id
    FROM patstat2024b.tls224_appln_cpc t2
    WHERE t2.cpc_class_symbol LIKE 'Y02%' OR t2.cpc_class_symbol LIKE 'Y04%'
 )
ORDER BY t1.docdb_family_id, t1.appln_id;
```

Then, I created a new table called new_green_patents_wipo1 in the menendez schema to store **green patent applications excluding** those whose docdb_family_id appears in the DGPT_WIPO_matching_families table. Essentially, this operation filters out **green-digital overlapping patents**, leaving you with patents that are **green only**—i.e., **purely green patents** not associated with digital technology based on my existing WIPO matching.

```
CREATE TABLE `menendez`.`new_green_patents_wipo1` (
 `docdb_family_id` INT NOT NULL,
 `appln_id` INT NOT NULL,
 `appln_auth` VARCHAR(2),
 `earliest_filing_year` INT NOT NULL
);
INSERT INTO menendez.new_green_patents_wipo1 (docdb_family_id, appln_id, appln_auth, earliest_filing_year)
SELECT * FROM new_green_patents f1
where f1.docdb_family_id NOT IN (
     SELECT DISTINCT f2.docdb_family_id FROM menendez.DGPT_WIPO_matching_families f2
  );
```

After that, I created a new table called new_green_patents_wipo2 in the menendez schema to **add CPC classification information** to the previously filtered set of **green-only patents** stored in new_green_patents_wipo1. This table includes patent metadata (docdb_family_id, appln_id, appln_auth, earliest_filing_year) and the cpc_class_symbol indicating the technological field of the patent.

```
CREATE TABLE `menendez`.`new_green_patents_wipo2` (
 `docdb_family_id` INT NOT NULL,
 `appln_id` INT NOT NULL,
 `appln_auth` VARCHAR(2),
 `earliest_filing_year` INT NOT NULL,
 `cpc_class_symbol` VARCHAR(15)
);
INSERT INTO menendez.new_green_patents_wipo2 (docdb_family_id, appln_id, appln_auth,
earliest_filing_year, cpc_class_symbol)
SELECT f1.docdb_family_id, f1.appln_id, f1.appln_auth,f1.earliest_filing_year, f2.cpc_class_symbol FROM
new_green_patents_wipo1 f1
inner join patstat2024b.tls224_appln_cpc f2 on f2.appln_id=f1.appln_id;
```

Finally, I created a new_green_patents_wipo3 in the menendez schema, consolidating green-only **patent applications** with their **English-language abstracts**. This table includes standard patent metadata (docdb_family_id, appln_id, appln_auth, earliest_filing_year), the **CPC classification** (cpc_class_symbol), and both the **abstract text** (appln_abstract) and its **language code** (appln_abstract_lg). To populate it, I join the previously built new_green_patents_wipo2 (which contains metadata and CPC info for green-only patents) with the tls203_appln_abstr table from PATSTAT, which holds patent abstracts.

```
CREATE TABLE `menendez`.`new_green_patents_wipo3` (
 `docdb_family_id` INT NOT NULL,
 `appln_id` INT NOT NULL,
 `appln_auth` VARCHAR(2),
 `earliest_filing_year` INT NOT NULL,
 `cpc_class_symbol` VARCHAR(15),
 `appln_abstract` TEXT NOT NULL ,
`appln_abstract_lg` TEXT NOT NULL)
;
INSERT INTO menendez.new_green_patents_wipo3 (docdb_family_id, appln_id,
appln_auth,earliest_filing_year, cpc_class_symbol, appln_abstract_lg,appln_abstract )
SELECT f1.docdb_family_id, f1.appln_id, f1.appln_auth, f1.earliest_filing_year, f1.cpc_class_symbol,
f2.appln_abstract_lg, f2.appln_abstract FROM new_green_patents_wipo2 f1
inner join patstat2024b.tls203_appln_abstr f2 on f2.appln_id=f1.appln_id
where f2.appln_abstract_lg like 'en';
```