

Appendix A: Documentation related to the E-R schema of BEyOND

1. CORE: contains information about the sediment core.

Attributes:

CoreID: sequential numbers identifying of the cores, it is a number ID (primary key).
Name: core name.
NameComplete: the name complete of the core.
Depth: depth to which the core has been taking.
Size: core length.
Latitude: the latitude position recorded in decimal degrees.
Longitude: the longitude position recorded in decimal degrees.
Penetration: length of the core penetration.
Recovery: length of sediment retrieved.
Data: generic date/time.
Year:
Note: comments regarding core.

2. AREA: sea area where the core has been taking.

Attributes:

AreaID: sequential numbers identifying the area (primary key).
Name: area name.
Sea: sea name.

3. REGION: specifies the zone geographic the core.

Attributes:

RegionName: region name of the sea (primary key).
CoordinateRegion: localization of the region.

4. PROJECT: indicates information about the oceanographic campaigns.

Attributes:

ProjectName: oceanographic campaign name (primary key).
CodeProject: identification code of the oceanographic campaign.
DataStart: start of the oceanographic campaign.
DataEnd: end of the oceanographic campaign.
Ship: ship name.

5. PAPER: contains information that concerning at the publications.

Attributes:

PaperID: sequential numbers identifying the papers (primary key).
Title: article title.
Author: author of the article.
Citation: author and co-authors of the article.
Year: published date.
Journal: journal in which it has been published.
Issues: issue of the journal.
Volume: volumen of the journal.
Pages:
Doi: publication digital object identifier.

6. PROXY: comprehends the feature of the proxies.

Attributes:

MeasurementID: sequential numbers identifying the proxies (primary key).
Name: proxy name.
Description: generic name for description of proxy.
Method: method used to obtain the values of proxy.
Class: proxy class.
SubClass: proxy subclass.
Material: origin of the proxy.
Units: measurement unit.

7. MEASUREMENT: records the values analyzed in the core depending on the depth.

Attributes:

DepthStart: depth start of the section analyzed (primary key).
DepthEnd: depth end of the section analyzed (primary key).
Value: value obtained of the section analyzed.
Notes: additional Information.
DeltaError: margin of error.

8. SAPROPEL: defines the sapropel event.

Attributes:

Top: depth where sapropel section end.
Bottom: depth where sapropel section start.
SapropelName: sapropel event name (primary key).

9. CORE_MEASUREMENT: records the average value of the cores.

Attributes:

Value: average value obtained in the core.
Notes: additional Information.

Appendix B: SQL Queries

Plot Fig.13 (a)

Table with cores that have the TOC proxy and Age

```
create view CoreIDwithAge as
SELECT distinct m1.CoreID
From MEASUREMENT m1, MEASUREMENT m2, AREA, REGION, CORE
WHERE AREA.ArealID = REGION.ArealID
AND REGION.RegionName = CORE.RegionName
AND CORE.CoreID = m1.CoreID
AND m1.CoreID = m2.CoreID
AND AREA.Name = 'Eastern Mediterranean'
AND (m1.MeasurementID = '1518' or m1.MeasurementID = '592')
AND m2.MeasurementID = '78'
```

Create standardized depth:

```
create view TOT1 as
(
SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, CoreIDwithAge
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND CORE.CoreID = CoreIDwithAge.CoreID
AND (MEASUREMENT.MeasurementID = 1518 or MEASUREMENT.MeasurementID = 592)
AND SAPROPEL.CoreID = CORE.CoreID
)
UNION
(
SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, CoreIDwithAge
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND CORE.CoreID = CoreIDwithAge.CoreID
AND (MEASUREMENT.MeasurementID = 1518 or MEASUREMENT.MeasurementID = 592)
AND SAPROPEL.CoreID = CORE.CoreID
);
```

```
create view TOT2 as
SELECT d1, min( d2 ) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the TOC results:

```
create view t_toc as
SELECT MEASUREMENT.Value TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, MEASUREMENT.PaperID PaperID
FROM CORE, MEASUREMENT, TOT2, SAPROPEL, CoreIDwithAge, PAPER
WHERE CORE.CoreID = CoreIDwithAge.CoreID
AND CORE.CoreID = MEASUREMENT.CoreID
AND PAPER.PaperID = MEASUREMENT.PaperID
AND SAPROPEL.CoreID= CORE.CoreID
AND (MEASUREMENT.MeasurementID = 1518 or MEASUREMENT.MeasurementID = 592)
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom -SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
```

Table with the Age results:

```
create view t_year as
SELECT MEASUREMENT.Value Age, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, MEASUREMENT.PaperID PaperID
FROM CORE, MEASUREMENT, TOT2, SAPROPEL, CoreIDwithAge, PAPER
WHERE CORE.CoreID = CoreIDwithAge.CoreID
AND CORE.CoreID = MEASUREMENT.CoreID
AND PAPER.PaperID = MEASUREMENT.PaperID
AND SAPROPEL.CoreID= CORE.CoreID
AND MEASUREMENT.MeasurementID = 78
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom -SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
```

Union table with TOC-Age results:

```
SELECT *
FROM t_toc, t_year
WHERE t_toc.d1=t_year.d1
AND t_toc.d2_ok=t_year.d2_ok
AND t_toc.PaperID=t_year. PaperID
```

Plot Fig.13 (b)

Table with cores that have the TOC proxy and Age

```
create view CoreIDwithAge as
SELECT distinct m1.CoreID
From MEASUREMENT m1, MEASUREMENT m2, AREA, REGION, CORE
```

```
WHERE AREA.ArealID = REGION.ArealID
AND REGION.RegionName = CORE.RegionName
AND CORE.CoreID = m1.CoreID
AND m1.CoreID = m2.CoreID
AND AREA.Name = 'Eastern Mediterranean'
AND (m1.MeasurementID = '1518' or m1.MeasurementID = '592')
AND m2.MeasurementID = '78'
```

Create standardized depth:

```
create view TOT1 as
(
SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, CoreIDwithAge
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND CORE.CoreID = CoreIDwithAge.CoreID
AND (MEASUREMENT.MeasurementID = 1518 or MEASUREMENT.MeasurementID = 592)
AND SAPROPEL.CoreID = CORE.CoreID
)
UNION
(
SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, CoreIDwithAge
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND CORE.CoreID = CoreIDwithAge.CoreID
AND (MEASUREMENT.MeasurementID = 1518 or MEASUREMENT.MeasurementID = 592)
AND SAPROPEL.CoreID = CORE.CoreID
);
```

```
create view TOT2 as
SELECT d1, min( d2 ) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with average values of the TOC:

```
create view t_toc as

SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, TOT2, SAPROPEL, CoreIDwithAge
WHERE CORE.CoreID = CoreIDwithAge.CoreID
AND CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND (MEASUREMENT.MeasurementID = 1518 or MEASUREMENT.MeasurementID = 592)
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom -SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Table with average values of the Age:

```
create view t_year as

SELECT avg(MEASUREMENT.Value) Age, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, TOT2, SAPROPEL, CoreIDwithAge
WHERE CORE.CoreID = CoreIDwithAge.CoreID
AND CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND MEASUREMENT.MeasurementID = 78
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom -SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Union table with average values TOC-Age:

```
SELECT *
FROM t_toc, t_year
WHERE t_toc.d1=t_year.d1 AND t_toc.d2_ok=t_year.d2_ok
```

Plot Fig. 14**Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.ArealID = REGION.ArealID
```

```

AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)

```

```

create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1

```

Table with the results:

```

SELECT avg(MEASUREMENT.Value) TOC , TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok

```

Plot Fig. 15**Create standardized depth:**

```

create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)

```

```

create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1

```

Table with the results:

```

SELECT avg(MEASUREMENT.Value) TOC , TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-

```

ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok

Plot Fig. 16**Depth 0-1000 m****Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 1000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 1000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND CORE.Depth >= 1000
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Depth 1000-2000 m**Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 1000
AND CORE.Depth <= 2000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean')
```



```
AND CORE.Depth >= 1000
AND CORE.Depth <= 2000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND CORE.Depth >= 1000
AND CORE.Depth <= 2000
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Depth 2000-3000 m**Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 2000
AND CORE.Depth <= 3000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 2000
AND CORE.Depth <= 3000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND CORE.Depth >= 2000
AND CORE.Depth <= 3000
```

```
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Depth >3000 m**Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 3000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, SAPROPEL, AREA, REGION
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND CORE.Depth >= 3000
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Name = 'Eastern Mediterranean'
AND MEASUREMENT.MeasurementID =1518
AND CORE.Depth >= 3000
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Plot Fig. 17**Adriatic sea****Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Adriatic'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Adriatic'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Adriatic'
AND MEASUREMENT.MeasurementID =1518
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Levantine sea**Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Levantine'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Levantine'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC, TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Levantine'
AND MEASUREMENT.MeasurementID =1518
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```

Ionian and Libyan sea**Create standardized depth:**

```
create view TOT1 as
(SELECT (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
```




```
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Ionian'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
UNION
(SELECT (MEASUREMENT.DepthEnd - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) d1
FROM CORE, MEASUREMENT, AREA, REGION, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Ionian'
AND MEASUREMENT.MeasurementID =1518
AND SAPROPEL.CoreID= CORE.CoreID)
```

```
create view TOT2 as
SELECT d1, min(d2) d2_ok
FROM (
SELECT t1.d1 d1, t2.d1 d2
FROM `TOT1` t1, TOT1 t2
WHERE t1.d1 < t2.d1) t
group by d1
```

Table with the results:

```
SELECT avg(MEASUREMENT.Value) TOC , TOT2.d1 standardised_start, TOT2.d2_ok standardised_end, count(MEASUREMENT.Value) number_of_core, std
(MEASUREMENT.Value) STD
FROM CORE, MEASUREMENT, AREA, REGION, TOT2, SAPROPEL
WHERE CORE.CoreID = MEASUREMENT.CoreID
AND SAPROPEL.CoreID= CORE.CoreID
AND AREA.AreaID = REGION.AreaID
AND CORE.RegionName = REGION.RegionName
AND AREA.Sea = 'Ionian'
AND MEASUREMENT.MeasurementID =1518
AND TOT2.d1 >= (MEASUREMENT.DepthStart - SAPROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top) AND TOT2.d2_ok <= (MEASUREMENT.DepthEnd - SAP-
ROPEL.Top) / (SAPROPEL.Bottom - SAPROPEL.Top)
GROUP BY TOT2.d1, TOT2.d2_ok
```