```sql
-- Maja Gensow 22.01.2021
-- Data collection, -management, - visualization lecture

-- SQL code for calculating the temperature indices:
drop view if exists avg_temp; --mean temperature
create temporary view avg_temp as
select
    (select device_id
        from metadata
        where metadata.id=data.meta_id) as HOBO_id,
    AVG(value) as avg_t
from data
join metadata on metadata.id = data.meta_id
where metadata.term_id = 11 -- term_id 11= Nov/Dec 2020
group by meta_id
order by HOBO_id ASC;
select * from avg_temp;

drop view if exists avg_dtemp; --mean day temperature
create temporary view avg_dtemp as
SELECT
    (select device_id
        from metadata
        where metadata.id=data.meta_id) as HOBO_id,
    avg(value) as avg_dt
from data
join metadata on metadata.id = data.meta_id
where metadata.term_id = 11
and EXTRACT(hour FROM data.tstamp) >= 6 -- from 6am
and EXTRACT(hour FROM data.tstamp) < 18 -- to 6pm
group by HOBO_id
order by HOBO_id ASC;
select * from avg_dtemp;

drop view if exists avg_ntemp; -- mean night temperature
create temporary view avg_ntemp as
SELECT
    (select device_id
        from metadata
        where metadata.id=data.meta_id) as HOBO_id,
    avg(value) as avg_nt
from data
join metadata on metadata.id = data.meta_id
where metadata.term_id = 11
and (EXTRACT(hour FROM data.tstamp) < 6 -- to 6am
or EXTRACT(hour FROM data.tstamp) >= 18) -- from 6pm
group by HOBO_id
order by HOBO_id ASC;
select * from avg_ntemp;
```

```sql
drop view if exists temp_ind; -- final view
create view temp_ind as
select
a.hobo_id,
a.avg_t,
d.avg_dt,
n.avg_nt,
(d.avg_dt-n.avg_nt) as dif_temp -- calculates the difference
from
avg_temp a
join avg_dtemp d
on d.hobo_id = a.hobo_id
join avg_ntemp n
on n.hobo_id = a.hobo_id;
select * from temp_ind;

-- SQL code for calculating the correlation of temperature

-- determines closest HOBO logger:
DROP VIEW IF EXISTS metadistance cascade;
CREATE VIEW metadistance AS
WITH metadistance AS (
    SELECT *,
    (SELECT id FROM metadata ly WHERE term_id=9
    ORDER BY st_distance(m.location, ly.location) ASC LIMIT 1)
        as close_meta20_id,
    (SELECT id FROM metadata ly WHERE term_id=7
    ORDER BY st_distance(m.location, ly.location) ASC LIMIT 1)
        as close_meta19_id
    FROM metadata m
    WHERE term_id=11 AND sensor_id=1
    )
SELECT *
FROM metadistance;
SELECT * FROM metadistance;

-- normalises the values with their respective group mean value
DROP VIEW IF EXISTS data_normal cascade;
CREATE VIEW data_normal AS
SELECT
    row_number() OVER (PARTITION BY meta_id, variable_id
        ORDER BY tstamp ASC) as measurement_index,
    *,
    value - avg(value) OVER (PARTITION BY meta_id, variable_id)
        AS norm,
    avg(value) OVER (PARTITION BY meta_id, variable_id)
        AS group_avg
FROM data;
SELECT * FROM data_normal;
```

```sql
-- Calculates the correlation of each HOBO logger and its closest of each term
DROP VIEW IF EXISTS data_cor CASCADE;
CREATE VIEW data_cor AS
    SELECT
        metadistance.id,
        avg(d.value) AS "mean",
        corr(d.norm, d19.norm) AS "Tcorr1Y",
      corr(d.norm, d20.norm) AS "Tcorr2Y"
    FROM data_normal d
    JOIN metadistance on metadistance.id = d.meta_id
    JOIN metadata m19 on metadistance.close_meta19_id=m19.id
    JOIN metadata m20 on metadistance.close_meta20_id=m20.id
-- correlation between term 11 and term 7(2020 ~ 2018)
-- and between term 11 and term 9 (2020 ~ 2019)
    JOIN data_normal d19 on m19.id=d19.meta_id
AND d.measurement_index=d19.measurement_index
    JOIN data_normal d20 on m20.id=d20.meta_id
AND d.measurement_index=d20.measurement_index

GROUP BY metadistance.id;
SELECT * FROM data_cor;

-- creates final view and adds important logger info
DROP VIEW IF EXISTS final_cor CASCADE;
CREATE VIEW final_cor AS
    SELECT
        cor.id, "Tcorr1Y", "Tcorr2Y",
        d.device_id, d.location, d.term_id
    FROM data_cor cor
    JOIN metadata d ON d.id=cor.id;
SELECT * FROM final_cor
-- done :)
```