DB

PostgreSQL 쿼리 작성

- 기존 스마트빌리지 웹 서버가 MySQL DB와 연결했었지만 새로운 PostgreSQL DB와 연결하게 되면서 새로운 데이터 컬럼 및 데이터를 쿼리에 적용해 서비스를 유지하는 쿼리 작성

주요 성과

- **물, 에너지 사용량 서비스와 빌리지 내 대기환경정보 사용량 2건에 대해 새로운 DB 기반 새로운 쿼리를 작성해 웹 서비스에 적용**
- **물, 에너지 사용량 서비스에서 냉방 탭 추가하여 냉방 에너지 표기 기능 추가**
- **물, 에너지 사용량 서비스의 전기 탭에서 송전량 표기 기능 추가**

- waterEnergy 서비스 변경사항

  mysql(기존 DB)

  tbl_house_energy_hist

Aa 이름	자료형	설명
house_energy_hist_id	NN PK int	id
reg_date	varchar(500)	해당 에너지를 사용한 날짜
house_dong_ho	varchar(500)	해당 에너지를 사용한 동호
water_usage	varchar(500)	해당 날짜의 물 사용량
electric_usage	varchar(500)	해당 날짜의 전기 사용량
heating_usage	varchar(500)	해당 날짜의 난방 사용량
hot_water_usage	varchar(500)	해당 날짜의 온수 사용량

PostgreSQL(변경 후 DB)

household_energy_msrmt_system1_copy

Aa 이름	자료형	설명
reg_date	character varying(50)	에너지 등록 날짜
heating_usage_month	double precision	해당 월의 난방 사용량
water_usage_month	double precision	해당 월의 물 사용량
cooling_usage_month	double precision	해당 월의 냉방 사용량
power_transmission_month	double precision	해당 월의 전기 송전량
hotwater_usage_month	double precision	해당 월의 온수 사용량
power_receive_month	double precision	해당 월의 전기 사용량
dongho	character varying(50)	해당 에너지를 사용한 동 호
제목 없음		

DB 변경사항

- 각 날짜의 에너지 사용량을 받아오다가 달(month) 단위로 합산된 에너지 사용량을
  DB에 저장
- 전기 송전량 데이터(power_transmission_month)도 표기가 필요해 전기 사용량 탭을 대상으로 하는 ExportChartVo 객체 생성
- MySQL에서는 에너지 사용량을 varchar 자료형으로 저장했지만
  PostgreSQL에서는 double precision으로 저장

Query 변경사항

1. 현재 날짜의 에너지 사용량과 전년 같은 날짜의 에너지 사용량을 얻는 쿼리

- MySQL(기존 쿼리)

  https://github.com/hyeonseong0917/smartVillage/blob/main/MySQL/WaterEnergy_SQL.xml

  WaterEnergyDAO.selectWaterUsage 참조

- PostgreSQL(변경 후 쿼리)

```
-- 1. 최종 결과로 반환할 열을 선택하는 부분
SELECT
    date_tab.recent_reg_date AS recent_reg_date,  -- 최근 등록일자
    (
        -- 1-1. 최근 등록일자에 대한 최대 물 사용량을 서브쿼리로 계산
        SELECT MAX("water_usage_month")
        FROM household_energy_msrmt_system1_copy
        WHERE "dongho" = #houseDongHo#
        AND "reg_date" = date_tab.recent_reg_date
        GROUP BY "reg_date"
        ORDER BY MAX("water_usage_month") DESC
        LIMIT 1
    ) AS recent_usage,
    date_tab.prev_reg_date AS prev_reg_date,  -- 이전(1년 전) 등록일자
    (
        -- 1-2. 이전 등록일자에 대한 최대 물 사용량을 서브쿼리로 계산
        SELECT MAX("water_usage_month")
        FROM household_energy_msrmt_system1_copy
        WHERE "dongho" = #houseDongHo#
        AND "reg_date" = date_tab.prev_reg_date
        GROUP BY "reg_date"
        ORDER BY MAX("water_usage_month") DESC
        LIMIT 1
    ) AS prev_usage
FROM
    household_energy_msrmt_system1_copy,  -- 메인 테이블
    (
        -- 2. 서브쿼리: 최근 및 이전 등록일자 계산
        SELECT
            "dongho",
            (
                -- 2-1. 최근 등록일자에 대한 최대 날짜를 서브쿼리로 계산
                SELECT MAX("reg_date")
                FROM household_energy_msrmt_system1_copy
                WHERE TO_TIMESTAMP("reg_date", 'YYYY-MM-DD"T"HH24:MI') BETWEEN (NOW() - IN
```

```
                    AND "dongho" = #houseDongHo#
            ) AS "recent_reg_date",
            (
                -- 2-2. 이전 등록일자에 대한 최대 날짜를 서브쿼리로 계산
                SELECT MAX("reg_date")
                FROM household_energy_msrmt_system1_copy
                WHERE TO_TIMESTAMP("reg_date", 'YYYY-MM-DD"T"HH24:MI') BETWEEN (NOW() - IN
                    AND "dongho" = #houseDongHo#
            ) AS "prev_reg_date"
        FROM
            household_energy_msrmt_system1_copy
        WHERE
            "dongho" = #houseDongHo#
        GROUP BY
            "dongho"
    ) date_tab
WHERE
    household_energy_msrmt_system1_copy."dongho" = date_tab."dongho"  -- house_dong_ho 값이
ORDER BY
    "water_usage_month" DESC  -- 최대 물 사용량을 기준으로 내림차순으로 정렬
LIMIT
    1;  -- 결과를 최상위 1개로 제한
```

결과값

	recent_reg_date text	recent_usage double precision	prev_reg_date text	prev_usage double precision
1	2024-01-01T00:00	12.48	2023-01-01T00:00	25.68

2. 현재 날짜로부터 6개월 전까지의 에너지 사용량과 전년 같은 날짜로부터 6개월 전까지의 에너지 사용량을 얻는 쿼리

- MySQL(기존 쿼리)

https://github.com/hyeonseong0917/smartVillage/blob/main/MySQL/WaterEnergy_SQL.xml

WaterEnergyDAO.selectWaterUsageChartData 참조

- PostgreSQL(변경 후 쿼리)

```
-- 1. 최종 결과로 반환할 열을 선택하는 부분
SELECT
    -- 2. 최근 및 이전(1년 전) 등록연월에 대한 정보 및 물 사용량 계산
    CONCAT(CAST(recent_usage.reg_year AS VARCHAR),'년') AS reg_year,  -- 등록연월의 연 부분
    CONCAT(LPAD(CAST(recent_usage.reg_month AS VARCHAR), 2, '0'),'월') AS reg_month,  -- 등록연
    recent_usage.water_usage_month AS recent_usages,  -- 최근 등록연월의 물 사용량
    prev_usage.water_usage_month AS prev_usages  -- 이전 등록연월의 물 사용량
-- 3. 최근 등록연월에 대한 정보와 이전 등록연월에 대한 정보를 서브쿼리로 가져오고 조인
FROM
    (
        -- 3-1. 최근 등록연월의 정보를 가져오는 서브쿼리
        SELECT
```

```
            EXTRACT(YEAR FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')) AS reg_year,
            EXTRACT(MONTH FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')) AS reg_month,
            MAX(water_usage_month) AS water_usage_month,
            dongho
        FROM
            household_energy_msrmt_system1_copy
        WHERE
            dongho = #houseDongHo#
            AND TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI') BETWEEN DATE_TRUNC('MONTH', CU
        GROUP BY
            EXTRACT(YEAR FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')),
            EXTRACT(MONTH FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')),
            dongho
        ORDER BY
            reg_year DESC, reg_month DESC
        LIMIT 6
    ) recent_usage,
    (
        -- 3-2. 이전 등록연월의 정보를 가져오는 서브쿼리
        SELECT
            EXTRACT(YEAR FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')) AS reg_year,
            EXTRACT(MONTH FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')) AS reg_month,
            MAX(water_usage_month) AS water_usage_month,
            dongho
        FROM
            household_energy_msrmt_system1_copy
        WHERE
            dongho = #houseDongHo#
            AND TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI') BETWEEN DATE_TRUNC('MONTH', CU
        GROUP BY
            EXTRACT(YEAR FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')),
            EXTRACT(MONTH FROM TO_TIMESTAMP(reg_date, 'YYYY-MM-DD"T"HH24:MI')),
            dongho
        ORDER BY
            reg_year DESC, reg_month DESC
        LIMIT 6
    ) prev_usage
-- 4. 최근 등록연월과 이전 등록연월이 일치하는지 확인
WHERE
    recent_usage.reg_month = prev_usage.reg_month
-- 5. 결과를 연도 오름차순, 월 오름차순으로 정렬
ORDER BY
    reg_year ASC, reg_month ASC;
```

결과값(물)

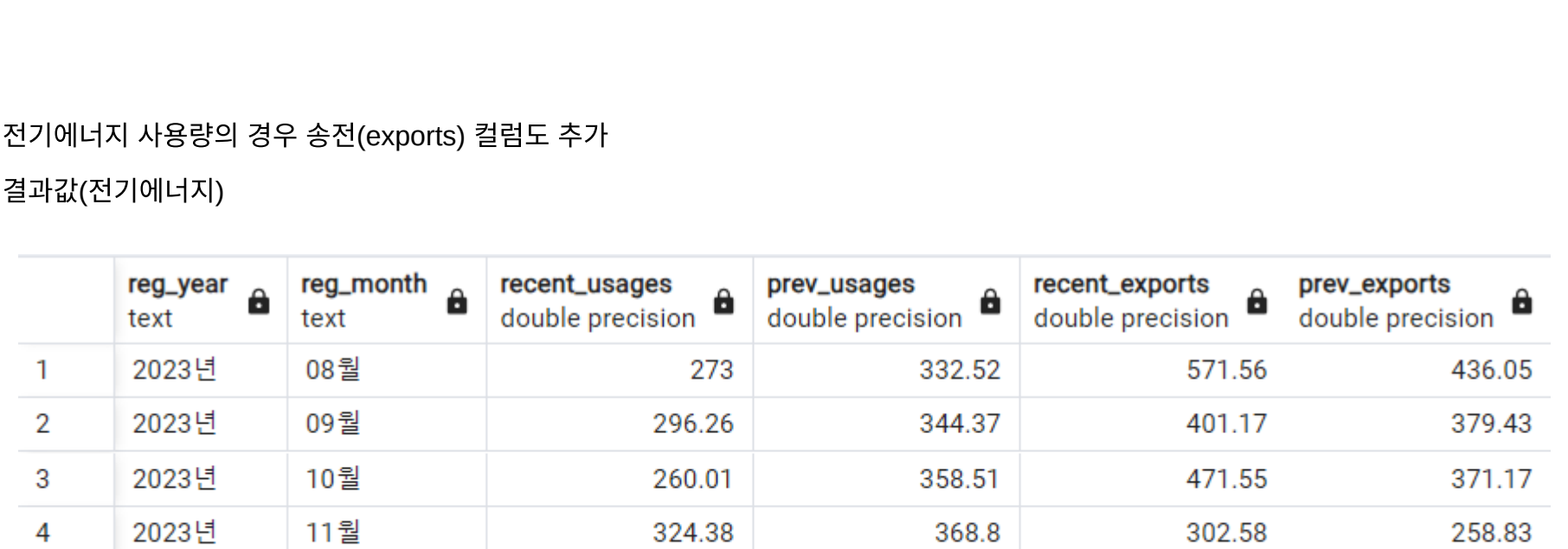
	reg_year text	reg_month text	recent_usages double precision	prev_usages double precision
1	2023년	08월	52.95	45.99
2	2023년	09월	26.91	28.89
3	2023년	10월	23.84	29.04
4	2023년	11월	24.11	28.17
5	2023년	12월	25.39	24.65
6	2024년	01월	12.48	25.68

전기에너지 사용량의 경우 송전(exports) 컬럼도 추가

결과값(전기에너지)

	reg_year text	reg_month text	recent_usages double precision	prev_usages double precision	recent_exports double precision	prev_exports double precision
1	2023년	08월	273	332.52	571.56	436.05
2	2023년	09월	296.26	344.37	401.17	379.43
3	2023년	10월	260.01	358.51	471.55	371.17
4	2023년	11월	324.38	368.8	302.58	258.83
5	2023년	12월	374.74	391.94	213.59	224.59
6	2024년	01월	150.92	480.71	137.61	245.78

차트 표시 웹 서비스 페이지



- environment 서비스 변경사항

mysql

tbl_thing_attr_hist

Aa 이름	자료형	설명
tbl_thing_attr_hist_id	NN PK int	id
thing_model_attribute_id	varchar(50)	측정된 지역의 id값
thing_model_attribute_name	varchar(50)	측정된 지역의 이름
thing_model_attribute_type	varchar(50)	측정된 환경정보 타입
thing_model_attribute_value	varchar(50)	측정된 환경정보 값
thing_model_attribute_time	varchar(50)	측정된 시간

PostgreSQL

smartpole1

Aa 이름	자료형	설명
id	integer	id
thing	character varying	센서 정보
air_temperature	double precision	해당 지역의 기온
humidity	double precision	해당 지역의 습도
pm10	double precision	해당 지역의 미세먼지
pm25	double precision	해당 지역의 초미세먼지
noise	double precision	해당 지역의 소음
iluminance	double precision	해당 지역의 조도
measurement_time	character varying	해당 지역에서 측정된 시간

DB 변경사항(MySQL to PostgreSQL)

- 측정된 지역의 이름 데이터가 PostgreSQL에서 사라짐 - 센서 정보의 마지막 4자리 숫자로 지역 매핑 가능
  - 0001 ~ 0008까지 8개의 지역 존재
  - 0001: 주출입구, 0002: 도보출입구, 0003: 1단지출입구, 0004: 스마트플라자,
    0005: 공용공간, 0006: 코리도1, 0007: 코리도2, 0008: 코리도3
- 측정된 환경정보 타입이 없어지고 모든 타입의 데이터가 주어짐
  - 동적으로 환경정보 타입을 받아 해당 환경정보 타입의 컬럼만 SELECT 필요
- 컬럼 명들이 미세하게 다름
  - ex) attributeType이 fine_dust였지만, PostgreSQL에서는 pm10으로 컬럼을 쿼리해야함

Query 변경사항

1. 환경정보 타입을 동적으로 받아 가장 최근의 값을 얻는 쿼리

- MySQL(기존 쿼리)

https://github.com/hyeonseong0917/smartVillage/blob/main/MySQL/Environment_SQL.xml

selectEnvironmentInfo 참조

- PostgreSQL(변경 후 쿼리)

```
-- 1. 결과로 반환할 열을 선택하는 부분
SELECT
    id AS thing_model_attribute_id,  -- thing_model_attribute_id
    -- 2. thing 값에 따라 thing_model_attribute_name을 결정하는 CASE 문
    CASE
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0001' THEN '주출입구'
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0002' THEN '도보출입구'
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0003' THEN '1단지출입구'
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0004' THEN '스마트플라자'
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0005' THEN '공용공간'
```

```
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0006' THEN '코리도1'
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0007' THEN '코리도2'
        WHEN SUBSTRING(thing FROM LENGTH(thing) - 3) = '0008' THEN '코리도3'
    END AS thing_model_attribute_name,  -- thing_model_attribute_name
    #attributeType# AS thing_model_attribute_type,  -- thing_model_attribute_type
    -- 3. attributeType 값에 따라 thing_model_attribute_value를 결정하는 CASE 문
    CASE
        WHEN #attributeType# = 'humidity' THEN humidity
        WHEN #attributeType# = 'temperature' THEN air_temperature
        WHEN #attributeType# = 'fine_dust' THEN pm10
        WHEN #attributeType# = 'ultra_particles' THEN pm25
        WHEN #attributeType# = 'noise_pollution' THEN noise
        WHEN #attributeType# = 'ilumination' THEN iluminance
    END AS thing_model_attribute_value  -- thing_model_attribute_value
-- 4. 결과를 smartpole1 테이블에서 thing_model_attribute_id를 기준으로 내림차순으로 정렬하고 상위 8개
FROM
    smartpole1
ORDER BY
    thing_model_attribute_id DESC
LIMIT 8;
```

#attributeType#=fine_dust 일 때 결과값

	thing_model_attribute_id integer	thing_model_attribute_name text	thing_model_attribute_type text	thing_model_attribute_value double precision
1	71234	코리도3	fine_dust	37
2	71233	코리도2	fine_dust	39
3	71232	코리도1	fine_dust	40
4	71231	공용공간	fine_dust	44
5	71230	스마트폴라자	fine_dust	41
6	71229	1단지출입구	fine_dust	38
7	71228	도보출입구	fine_dust	38
8	71227	주출입구	fine_dust	46

8개 구역에 대한 가장 최근의 #attribute# 측정값이 쿼리됨

결과 웹 서비스 페이지