

## Ejercicios SQL

Una vez ejecutado el script pollutant.sql tendremos 3 tablas:

- Tabla Item: contiene los contaminantes que podrían ser sensados por estaciones.

```
CREATE TABLE item
(
    code integer NOT NULL,
    name VARCHAR(5) NOT NULL,
    uom VARCHAR(15) NOT NULL,
    good FLOAT ,
    normal FLOAT ,
    bad FLOAT ,
    very_bad FLOAT ,
    PRIMARY KEY (code),
    UNIQUE (name)
);
```

Su código y su nombre no puede repetirse en dicha tabla (son claves).

El atributo uom indica la unidad de medida y los otros atributos los umbrales correspondientes a límite inferiores.

Ejemplo: Hasta ahora hay 8 contaminantes de interés.

Item (pollutant)						
Code	Name	Uom	Good	Normal	Bad	Very bad
1	'SO2'	'ppm'	0.02	0.05	0.15	1.0
3	'NO2'	'ppm'	0.03	0.06	0.2	2.0
5	'CO'	'ppm'	2.0	9.0	15.0	50.0
6	'O3'	'ppm'	0.03	0.09	0.15	0.5
8	'PM10'	'Microgram/m3'	30.0	80.0	150.0	600.0
9	'PM2.5'	'Microgram/m3'	15.0	35.0	75.0	500.0
11	'dummy'	'ppm'	null	null	null	1.2
12	'bis'	'ppm'	null	0.3	0.4	0.5

El pollutant 'SO2' se consideraría **Very Bad** si su valor sensado supera (>) 1.0.

Si el sensado da un valor superior (>) a 0.15 pero no supera (<=) 1.0, entonces es considerado **Bad**.

Si en cambio, es superior (>) a 0.05 pero no supera (<=) 0.15, se lo considera **Normal**.

Si fuera superior (>) 0.02 pero no supera (<=) 0.05, es **Bueno**.

Finalmente, si resultara que no supera (<=) 0.02 sería **Excellent**.

- Tabla Station: contiene las estaciones de sensado

```
CREATE TABLE station
(
  code integer NOT NULL,
  name VARCHAR(20) NOT NULL,
  address VARCHAR(100) NOT NULL,
  lat double precision NOT NULL,
  lon double precision NOT NULL,
  PRIMARY KEY (code),
  UNIQUE (lat, lon)
);
```

Su código no puede repetirse en dicha tabla. Tampoco la combinación lat+lon (son claves).

Ejemplo: Hasta ahora hay 3 estaciones.

Station				
Code	Name	address	Lat	lon
101	'Jongno-gu'	'19 Jong-ro 35ga-gil, Jongno-gu, Seoul, Republic of Korea'	37.57201639	127.0050074
102	'Jung-gu'	'15; Deoksugung-gil, Jung-gu, Seoul, Republic of Korea'	37.56426289	126.9746756
103	'Yongsan-gu'	'136; Hannam-daero, Yongsan-gu, Seoul, Republic of Korea'	37.54003270	127.00485

- Tabla Measurement: contiene las mediciones en diferentes momentos. Es decir, es un histórico del valor de cierta polución medida por cierta estación en cierto momento. No se indica la unidad de medida porque eso está indicado en la tabla Item. Asocia ítem\_code con station\_code con cierto date.

```
CREATE TABLE measurement
(
    date timestamp NOT NULL,
    station_code integer NOT NULL,
    item_code integer NOT NULL,
    value float NOT NULL,
    PRIMARY KEY (date, station_code, item_code),
    FOREIGN KEY (item_code)
        REFERENCES item ON DELETE CASCADE,
    FOREIGN KEY (station_code)
        REFERENCES station ON DELETE CASCADE
);
```

Ejemplo: Hasta ahora hay 53 mediciones históricas.

Notar que el ítem\_code 1 ('SO2') fue medido por la estación 101 ('Jongno-gu') varias veces, pero en diferentes instantes de tiempo.

Measurement			
Date	Station_code	Item_code	value
'2017-01-01 00:00'	101	1	0.04
'2017-01-01 00:00'	101	3	0.590
'2017-01-01 00:00'	101	5	2.5
'2017-01-01 00:00'	101	6	0.2
'2017-01-01 00:00'	101	8	93.0
'2017-01-01 00:00'	101	9	7.0
'2017-01-01 00:00'	102	1	0.6
'2017-01-01 00:00'	102	3	0.068
'2017-01-01 00:00'	102	5	1.3
'2017-01-01 00:00'	102	6	0.002
'2017-01-01 00:00'	102	8	77.0
'2017-01-01 00:00'	102	9	63.0
'2017-01-01 00:00'	103	1	0.005
'2017-01-01 00:00'	103	3	0.039
'2017-01-01 00:00'	103	5	1.4
'2017-01-01 00:00'	103	6	0.002
'2017-01-01 00:00'	103	8	70.0
'2017-01-01 00:00'	103	9	68.0
'2017-01-01 01:00'	101	1	0.004
'2017-01-01 01:00'	101	3	0.579
'2017-01-01 01:00'	101	5	1.2
'2017-01-01 01:00'	101	6	0.002
'2017-01-01 01:00'	101	8	71.0
'2017-01-01 01:00'	101	9	59.0
'2017-01-01 01:00'	102	1	0.006

'2017-01-01 01:00 '	102	3	0.066
'2017-01-01 01:00 '	102	5	1.4
'2017-01-01 01:00 '	102	6	0.002
'2017-01-01 01:00 '	102	8	76.0
'2017-01-01 01:00 '	102	9	63.0
'2017-01-01 01:00 '	103	1	0.004
'2017-01-01 01:00 '	103	3	0.038
'2017-01-01 01:00 '	103	5	1.4
'2017-01-01 01:00 '	103	6	0.002
'2017-01-01 01:00 '	103	8	73.0
'2017-01-01 02:00 '	101	1	0.004
'2017-01-01 02:00 '	101	3	0.0559
'2017-01-01 02:00 '	101	5	1.2
'2017-01-01 02:00 '	101	6	0.002
'2017-01-01 02:00 '	101	8	70.0
'2017-01-01 02:00 '	101	9	59.0
'2017-01-01 02:00 '	102	1	0.005
'2017-01-01 02:00 '	102	3	0.063
'2017-01-01 02:00 '	102	5	1.2
'2017-01-01 02:00 '	102	6	0.002
'2017-01-01 02:00 '	102	8	73.0
'2017-01-01 02:00 '	102	9	57.0
'2017-01-01 02:00 '	103	1	0.005
'2017-01-01 02:00 '	103	3	0.037
'2017-01-01 02:00 '	103	5	1.4
'2017-01-01 02:00 '	103	6	0.002
'2017-01-01 02:00 '	103	8	67.0
'2017-01-01 02:00 '	103	9	65.0

Importante: si un ítem no fue medido por cierta estación NO APARECE en la tabla measurement (ni siquiera con valor 0)

**Ejercicio 1 (clausula SELECT FROM ORDER BY)**

- 1.1) La tabla measurement tiene 53 mediciones. Tiene el ítem\_code pero no el nombre del contaminante. Para poder listar el nombre de dicho contaminante se precisa usar la información de 2 tablas: measurement y item.

¿Por qué la siguiente consulta no arroja lo esperado, es decir, no muestra 53 valores una columna más con el nombre del contaminante? ¿Por qué se obtienen 424 ( 53 \* 8) tuplas?

Explicar

```
SELECT measurement.*, name  
FROM measurement, item
```

Rta

- 1.2) Escribir la consulta en forma correcta. En el ejemplo, deberían obtenerse 53 tuplas. Las mismas de la tabla measurement pero con una columna adicional que indique el nombre del contaminante.

Rta

- 1.3) Mostrar del histórico de mediciones solo hasta las 8 más recientes, desplegando el nombre del ítem también

Rta

- 1.4) Escribir una consulta SQL que permita obtener: nombre de la estación (no código), nombre del contaminante (no código), fecha de la medición y el valor medido, mostrando primero las más reciente y de repetirse la fecha, ordenar luego por nombre del ítem ascendentemente y luego por nombre de la estación ascendentemente. Con los datos del ejemplo, deberíamos obtener sólo 53 tuplas. Mostramos solo un subconjunto

name	name	date	value
Jongno-gu	CO	2017-01-01T02:00:00Z	1.2
Jung-gu	CO	2017-01-01T02:00:00Z	1.2
Yongsan-gu	CO	2017-01-01T02:00:00Z	1.4
Jongno-gu	NO2	2017-01-01T02:00:00Z	0.05599999999999994
Jung-gu	NO2	2017-01-01T02:00:00Z	0.063
Yongsan-gu	NO2	2017-01-01T02:00:00Z	0.037000000000000005

Rta

1.5) Ídem al anterior, pero se espera que en el resultado los nombres de las columnas se muestren diferentes. El nombre de la estación debe aparecer como “EstacionNombre”, el del ítem “Contaminante”, el de la fecha “Instante”. El valor no cambiarlo.

Así cambiaría el header:

estacionnombre	contaminante	instante	value
----------------	--------------	----------	-------

Rta

## Ejercicio 2 (Uso de WHERE)

2.1) La misma consulta que realizamos en el ítem 1.4

```
SELECT name, name, date, value  
FROM (measurement JOIN item ON item_code = code) JOIN station  
     ON station_code= code  
ORDER BY date DESC, name ASC, name ASC
```

podría realizarse colocando la condición de “matcheo” en la cláusula WHERE. Es decir, usar un producto cartesiano entre dichas 3 tablas, para luego establecer en el WHERE las restricciones esperadas.

Reescribir dicha consulta de esta manera

Rta

2.2) Mostrar los nombres de los **contaminantes cuyos valores registrados** fueron Very Bad (superaron el valor VeryBad de la tabla indicada por ITEM). Mostrar dicho valor y el valor del umbral correspondiente (el cual depende de cada ítem)

Con los datos del ejemplo no se obtiene ninguno

Rta

2.3) Mostrar los nombres de los contaminantes cuyos valores registrados fueron Bad (superaron el valor Bad pero no el valor VeryBad de los umbrales indicados por la tabla ITEM). Mostrar dicho valor y los 2 umbrales correspondientes.

Con los datos del ejemplo se obtendría

name	value	bad	very_bad
NO2	0.5900000000000001	0.2	2
O3	0.2	0.15	0.5
SO2	0.6	0.15	1
NO2	0.58	0.2	2

Rta

2.4) Mostrar el nombre de los contaminantes que no se miden en unidades ppm

Con los datos del ejemplo sólo debería obtenerse 2 tuplas.

name

PM10

PM2.5

Rta

2.5) Mostrar aquellos nombres de contaminantes que todavía no tienen ingresados algún valor de sus umbrales.

Obtendremos 2: dummy y bis

Rta



2.6) Mostrar aquellos código y nombres de contaminantes que todavía no tienen mediciones asociadas. Con los datos del ejemplo debería obtenerse

Query #1 Execution time: 1ms

code	name
11	dummy
12	bis

Rta

2.7) Notar que para ver cuáles son los contaminantes (código y nombre) que sí tuvieron mediciones podemos proceder de diferentes formas también:

- a) con algún tipo de Join
- b) con un IN

Realizar las 2 versiones. Debe obtenerse:

code	name
1	SO2
3	NO2
5	CO
6	O3
8	PM10
9	PM2.5

Rta

2.8) Mostrar los nombres de contaminantes que tuvieron algún valor registrado considerado excelente ( $\leq$  good), ordenados por su código numérico

Con los datos del ejemplo obtendría

! name
SO2
CO
O3
PM2.5

Rta

2.9) Reescribir la consulta anterior para no solo mostrar el nombre del contaminante sino también cuál es ese valor que lo hizo excelente en dicha medición. Para verificar el resultado, mostrar también el valor del umbral Good

Deberíamos obtener

name	medido	umbralexcelente
SO2	0.005	0.02
SO2	0.006	0.02
SO2	0.005	0.02
SO2	0.005	0.02
SO2	0.004	0.02
SO2	0.004	0.02
SO2	0.004	0.02

CO	1.4	2
CO	1.3	2
CO	1.2	2
CO	1.2	2
CO	1.2	2
CO	1.4	2
CO	1.4	2
CO	1.4	2
O3	0.002	0.03
O3	0.002	0.03
O3	0.002	0.03
O3	0.002	0.03
O3	0.002	0.03
O3	0.002	0.03
O3	0.002	0.03
O3	0.002	0.03
PM2.5	7	15

Rta.

2.10) Mostrar los nombres de las estaciones que siempre midieron contaminantes que dieron valores aceptables, es decir TODAS sus mediciones arrojaron excelente, good o normal.

Con los valores del ejemplo debería obtenerse

name

Yongsan-gu

Rta

### Ejercicio 3 (Funciones de Agregación y Clausula Group by/ Having)

3.1) Mostrar el máximo valor registrado en las mediciones

Verificarlo realizando un SELECT ORDER BY DESC con LIMIT 1

Rta

3.2) Mostrar cuántos son los contaminantes que tienen alguna medición. Con los datos del ejemplo deberíamos obtener 6

Rta

3.3) Calcular por cada estación, cuántas mediciones ha realizado. Es decir, la distribución de mediciones entre las estaciones. Se espera que se despliegue el nombre de la estación junto a dicha cantidad.

Con los datos del ejemplo deberíamos obtener

name	cantidad
Yongsan-gu	17
Jung-gu	18
Jongno-gu	18

Rta

3.4) Ha medido alguna estación más de un contaminante? Mostrar por cada estación cuántos contaminantes diferentes a medido.

Con los datos del ejemplo, debería obtenerse

name	cantidad
Jongno-gu	6
Jung-gu	6
Yongsan-gu	6

Rta

3.5) Mostrar los nombres de los contaminantes cuyo valor promedio de sus mediciones fue mayor que 10. Mostrar nombre y dicho promedio.

Con los datos del ejemplo debería obtenerse

name	promedio
PM10	74.44444444444444
PM2.5	55.125

Rta

3.6) Mostrar aquellas estaciones que en un mismo instante de tiempo midieron más de un contaminante. Desplegar el nombre de dicha estación, la fecha y dicha cantidad.

Con los datos del ejemplo deberíamos obtener

name	date	qty
Jung-gu	2017-01-01T02:00:00Z	6
Yongsan-gu	2017-01-01T02:00:00Z	6
Jongno-gu	2017-01-01T02:00:00Z	6
Jongno-gu	2017-01-01T00:00:00Z	6
Yongsan-gu	2017-01-01T01:00:00Z	5
Jung-gu	2017-01-01T01:00:00Z	6
Jung-gu	2017-01-01T00:00:00Z	6
Yongsan-gu	2017-01-01T00:00:00Z	6
Jongno-gu	2017-01-01T01:00:00Z	6

Rta

3.7) Se quiere obtener la misma agregación, pero por año. Es decir

name	date_part	qty
Jongno-gu	2017	18
Yongsan-gu	2017	17
Jung-gu	2017	18

La forma de obtener solo el año de una fecha es con la funcion extract:  
extract(year from valorfecha)

Explicar por qué al ejecutar la siguiente consulta ni siquiera compila

```
SELECT station.name, DATE, count( item_code) AS qty
FROM station , measurement
WHERE station.code= station_code
GROUP BY station.code, station.name, extract(year from date)
HAVING COUNT( item_code) > 1
```

¿Cómo debería escribirse para averiguar cuantos ítems diferentes midió una misma estación durante un mismo año?

Rta

3.8) Se quiere obtener para cada nombre de contaminante la máxima y mínima cantidad registrada. Pero si un contaminante todavía no registra mediciones, igual se lo quiere obtener en el listado, con valor 0

Con el ejemplo, la consulta devolvería

name	coalesce	min
dummy	0	0
bis	0	0
CO	2.5	1.2
PM10	93	67
O3	0.2	0.002
SO2	0.6	0.004
NO2	0.5900000000000001	0.03700000000000005
PM2.5	68	7

Rta