

# Universidad ORT Uruguay

## Facultad de Ingeniería

Bernard Wand Polak

### Base de Datos 1

#### Obligatorio 2

Clavijo, Tomás

Nro est. 235426

Daneri, Franco

Nro est. 260284

Nuccio, Marco

Nro est. 264844

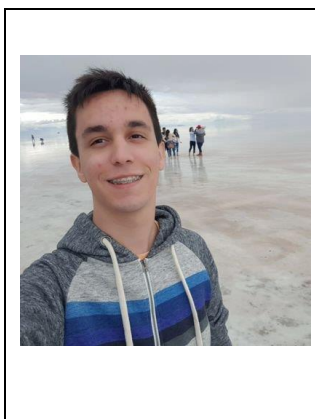
Grupo: M4C

Docente: Enrique Latorres

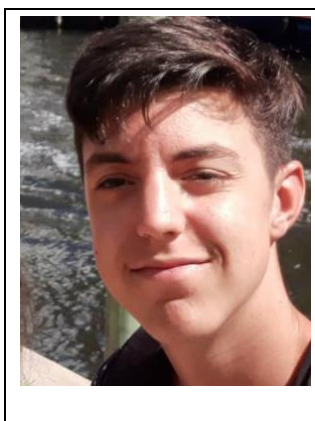
## Formulario de Antecedentes

### Curso Base de Datos I

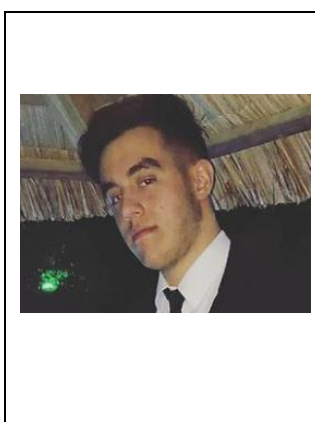
<b>Nro. Estudiante</b>	235426
<b>Nombre:</b>	Tomás Andrés
<b>Apellido:</b>	Clavijo Vitale
<b>Grupo / Turno:</b>	M4C
<b>¿Trabaja en algo relacionado con la carrera?</b>	Por el momento no.
<b>¿Qué tareas desempeña?</b>	



<b>Nro. Estudiante</b>	260284
<b>Nombre:</b>	Franco
<b>Apellido:</b>	Daneri
<b>Grupo / Turno:</b>	M4C
<b>¿Trabaja en algo relacionado con la carrera?</b>	Por el momento no.
<b>¿Qué tareas desempeña?</b>	



<b>Nro. Estudiante</b>	264844
<b>Nombre:</b>	Marco
<b>Apellido:</b>	Nuccio
<b>Grupo / Turno:</b>	M4C
<b>¿Trabaja en algo relacionado con la carrera?</b>	Por el momento no.
<b>¿Qué tareas desempeña?</b>	



## Tabla de contenido

Ejercicio 1	1
Ejercicio 2	3
Ejercicio 3	5
Ejercicio 4	5
Ejercicio 5	8
Ejercicio 6	10
Ejercicio 7	11
Ejercicio 8	12
Ejercicio 9	13
Ejercicio 10	14

## Ejercicio 1

### Problema:

Obtener el nombre y localidad de aquellos vacunatorios que tuvieron solicitudes agendadas o finalizadas, pero no ambos estados, en julio de 2021

### Solución

#### SQL:

```
SELECT V.NOMBREVACUNATORIO, L.LOCALIDAD
FROM LOCALIDAD L, VACUNATORIO V, SOLICITUD S
WHERE L.CODLOC = V.CODLOC
AND S.IDVACUNATORIO = V.IDVACUNATORIO
AND S.FECHA >= TO_DATE('01/07/2021', 'DD/MM/YYYY')
AND S.FECHA <= TO_DATE('31/07/2021', 'DD/MM/YYYY')
AND S.ESTADOSOL = 'Agendada'
AND V.NOMBREVACUNATORIO NOT IN (SELECT DISTINCT V.NOMBREVACUNATORIO
                                FROM LOCALIDAD L, VACUNATORIO V, SOLICITUD S
                                WHERE L.CODLOC = V.CODLOC
                                AND S.IDVACUNATORIO = V.IDVACUNATORIO
                                AND S.FECHA >= TO_DATE('01/07/2021', 'DD/MM/YYYY')
                                AND S.FECHA <= TO_DATE('31/07/2021', 'DD/MM/YYYY')
                                AND S.ESTADOSOL = 'Finalizada')

UNION

SELECT V.NOMBREVACUNATORIO, L.LOCALIDAD
FROM LOCALIDAD L, VACUNATORIO V, SOLICITUD S
WHERE L.CODLOC = V.CODLOC
AND S.IDVACUNATORIO = V.IDVACUNATORIO
AND S.FECHA >= TO_DATE('01/07/2021', 'DD/MM/YYYY')
AND S.FECHA <= TO_DATE('31/07/2021', 'DD/MM/YYYY')
AND S.ESTADOSOL = 'Finalizada'
AND V.NOMBREVACUNATORIO NOT IN (SELECT DISTINCT V.NOMBREVACUNATORIO
                                FROM LOCALIDAD L, VACUNATORIO V, SOLICITUD S
                                WHERE L.CODLOC = V.CODLOC
                                AND S.IDVACUNATORIO = V.IDVACUNATORIO
                                AND S.FECHA >= TO_DATE('01/07/2021', 'DD/MM/YYYY')
                                AND S.FECHA <= TO_DATE('31/07/2021', 'DD/MM/YYYY')
                                AND S.ESTADOSOL = 'Agendada');
```

**Justificación:** Para la resolución del SQL realizamos una consulta donde buscamos los vacunatorios que tuvieron solicitudes agendadas y a través de una subconsulta que no se encuentren en aquellos vacunatorios que tienen solicitudes finalizadas; y viceversa. Generando una unión entre ambas consultas, obteniendo como resultados aquellos vacunatorios que tienen agendadas, pero no finalizadas, y los que tienen finalizadas, pero no agendadas.

**AR:** (Notación de Relax)

```
solAgendada = ( $\pi$  NombreVacunatorio, Localidad ( $\sigma$  EstadoSol='Agendada'  $\wedge$  Fecha>=date('2021-05-01')
 $\wedge$  Fecha<=date('2021-07-30') (Localidad  $\bowtie$  Localidad.CodLoc = Vacunatorio.CodLoc (Vacunatorio  $\bowtie$ 
Vacunatorio.IdVacunatorio = Solicitud.IdVacunatorio Solicitud))))

solFinalizada = ( $\pi$  NombreVacunatorio, Localidad ( $\sigma$  EstadoSol='Finalizada'  $\wedge$  Fecha>=date('2021-07-
30')  $\wedge$  Fecha<=date('2021-07-01') (Localidad  $\bowtie$  Localidad.CodLoc = Vacunatorio.CodLoc (Vacunatorio  $\bowtie$ 
Vacunatorio.IdVacunatorio = Solicitud.IdVacunatorio Solicitud))))

(solAgendada  $\cup$  solFinalizada) - (solAgendada  $\cap$  solFinalizada)
```

**Justificación:** La idea fue similar a la parte de SQL. Solo que, al finalizar, unimos los vacunatorios que tienen agendadas y finalizadas y le restamos los que tienen ambas (la intersección)

## Ejercicio 2

### Problema:

Obtener el nombre de los vacunatorios donde solamente se haya vacunado con Pfizer y que dichos vacunatorios no tengan solicitudes canceladas durante mayo de 2021.

### Solución

#### SQL:

```
SELECT DISTINCT V.NOMBREVACUNATORIO
FROM VACUNATORIO V, SOLICITUD S
WHERE V.IDVACUNATORIO = S.IDVACUNATORIO
AND V.NOMBREVACUNATORIO NOT IN (SELECT DISTINCT V.NOMBREVACUNATORIO
                                FROM VACUNATORIO V, SOLICITUD S
                                WHERE V.IDVACUNATORIO = S.IDVACUNATORIO
                                AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
                                AND S.FECHAAGENDA <= TO_DATE('31/05/2021', 'DD/MM/YYYY')
                                AND S.ESTADOSOL = 'Cancelada')

INTERSECT

SELECT DISTINCT V.NOMBREVACUNATORIO
FROM VACUNATORIO V, SOLICITUD S, TIPOVACUNA T
WHERE V.IDVACUNATORIO = S.IDVACUNATORIO
AND S.IDVACUNA = T.IDVACUNA
AND T.CODIGO = 'PFZ'
AND V.NOMBREVACUNATORIO NOT IN (SELECT DISTINCT V.NOMBREVACUNATORIO
                                FROM VACUNATORIO V, SOLICITUD S, TIPOVACUNA T
                                WHERE V.IDVACUNATORIO = S.IDVACUNATORIO
                                AND S.IDVACUNA = T.IDVACUNA
                                AND T.CODIGO <> 'PFZ');
```

**Justificación:** Dividimos la consulta en dos partes: En la primera, buscamos los vacunatorios que no tengan solicitudes canceladas durante mayo de 2021 a través de una subconsulta realizada; en la segunda, buscamos los vacunatorios que solamente vacunaron con Pfizer, viendo que el código sea 'PFZ' y dicho vacunatorio no haya vacunado con otro tipo de vacuna distinto de Pfizer a través de la subconsulta.

#### AR: (Notación de Relax)

```
VacunatoriosPfizer =  $\pi$  NombreVacunatorio ((( $\sigma$  Codigo = 'PFZ' (TipoVacuna))  $\bowtie$  IdVacuna = IdVacuna (VacunatorioVacuna))  $\bowtie$  IdVacunatorio = IdVacunatorio (Vacunatorio))

VacunatoriosNoPfizer =  $\pi$  NombreVacunatorio ((( $\sigma$  Codigo  $\neq$  'PFZ' (TipoVacuna))  $\bowtie$  IdVacuna = IdVacuna (VacunatorioVacuna))  $\bowtie$  IdVacunatorio = IdVacunatorio (Vacunatorio))

SolicitudesCanceladas =  $\pi$  NombreVacunatorio ((( $\sigma$  EstadoSol = 'Cancelada'  $\wedge$  Fecha  $\geq$  date('2021-05-01')  $\wedge$  Fecha  $\leq$  date('2021-05-31') (Solicitud))  $\bowtie$  IdVacunatorio = IdVacunatorio (Vacunatorio)))

((VacunatoriosPfizer) - (VacunatoriosNoPfizer)) - (SolicitudesCanceladas)
```

**Justificación:** Para la solución AR planteamos hallar los vacunatorios que vacunaron con Pfizer y los que no, así como aquellos que tuvieron solicitudes canceladas en el período establecido. Para finalizar, a los primeros les restamos aquellos que vacunaron con alguna vacuna distinta de Pfizer para cumplir con la condición: “únicamente Pfizer”, y luego a ese conjunto le restamos los vacunatorios que tuvieron solicitudes canceladas.

## Ejercicio 3

### Problema:

Obtener el nombre, fecha de nacimiento y localidad de las personas que se anotaron con mayor antelación dentro de las solicitudes realizadas mediante agenda web. Considerar solo solicitudes no canceladas de personas a las que les corresponde la vacuna Sinovac.

### Solución

#### SQL:

```
SELECT P.NOMBRE, P.FECHANAC, L.LOCALIDAD
FROM LOCALIDAD L, PERSONA P, SOLICITUD S, TIPOVACUNA T
WHERE L.CODLOC = P.CODLOC
AND P.CI = S.CI
AND S.IDVACUNA = T.IDVACUNA
AND S.ESTADOSOL <> 'Cancelada'
AND T.CODIGO = 'SVC'
AND S.FECHA IN (SELECT MIN(S.FECHA)
                FROM SOLICITUD S
                WHERE S.VIASOLICITUD = 'Web');
```

**Datos de prueba:** En archivo .txt

**Justificación:** Planteamos una consulta con las respectivas condiciones establecidas por letra acerca del estado de la solicitud y el código de la vacuna, y a través de una subconsulta exigimos que la fecha sea la mínima fecha de todas las solicitudes vía web.

**AR:** (Notación de Relax)

```
ConsultaWeb = (π Nombre, FechaNac, Localidad, Fecha ((σ ViaSolicitud = 'Web' (Solicitud)) ⋈
Persona ⋈ Localidad))
```

```
Consulta = (π Nombre, FechaNac, Localidad, Fecha ((σ EstadoSol ≠ 'Cancelada' ∧ ViaSolicitud =
'Web' (Solicitud ⋈ (σ Codigo = 'Sinovac' (TipoVacuna)))) ⋈ Persona ⋈ Localidad))
```

```
Antelacion = (π m1.Nombre, m1.FechaNac, m1.Localidad, m1.Fecha (ρ (ConsultaWeb) m1 ⋈ m1.Fecha >
m2.Fecha ρ(ConsultaWeb) m2))
```

```
π Nombre, FechaNac, Localidad (Consulta - Antelacion)
```

**Justificación:** Con “ConsultaWeb” buscamos aquellos vacunatorios cuya vía de solicitud fue “Web” para utilizarla en “Antelación” buscando todos los que superan a la fecha mínima para posteriormente restárselo a “Consulta” que tiene todos los vacunatorios que cumplen con las características mencionadas en letra.



## Ejercicio 4

### Problema:

Obtener el nombre de los vacunatorios para los cuales existen solicitudes canceladas y que dichas solicitudes hayan sido realizadas por más de una vía de agenda. Considerar los vacunatorios de Montevideo y las cancelaciones durante la primera quincena de mayo de 2021.

### Solución

#### SQL:

```
SELECT DISTINCT V.NOMBREVACUNATORIO
FROM VACUNATORIO V, SOLICITUD S, LOCALIDAD L
WHERE V.CODLOC = L.CODLOC
AND V.IDVACUNATORIO = S.IDVACUNATORIO
AND S.ESTADOSOL = 'Cancelada'
AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
AND S.FECHAAGENDA <= TO_DATE('15/05/2021', 'DD/MM/YYYY')
AND L.LOCALIDAD = 'Montevideo'
AND S.VIASOLICITUD = 'Web'
AND V.NOMBREVACUNATORIO IN (SELECT V.NOMBREVACUNATORIO
                             FROM VACUNATORIO V, SOLICITUD S, LOCALIDAD L
                             WHERE V.CODLOC = L.CODLOC
                             AND V.IDVACUNATORIO = S.IDVACUNATORIO
                             AND S.ESTADOSOL = 'Cancelada'
                             AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
                             AND S.FECHAAGENDA <= TO_DATE('15/05/2021', 'DD/MM/YYYY')
                             AND L.LOCALIDAD = 'Montevideo'
                             AND S.VIASOLICITUD <> 'Web')

UNION

SELECT DISTINCT V.NOMBREVACUNATORIO
FROM VACUNATORIO V, SOLICITUD S, LOCALIDAD L
WHERE V.CODLOC = L.CODLOC
AND V.IDVACUNATORIO = S.IDVACUNATORIO
AND S.ESTADOSOL = 'Cancelada'
AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
AND S.FECHAAGENDA <= TO_DATE('15/05/2021', 'DD/MM/YYYY')
AND L.LOCALIDAD = 'Montevideo'
AND S.VIASOLICITUD = 'Whatsapp'
AND V.NOMBREVACUNATORIO IN (SELECT V.NOMBREVACUNATORIO
                             FROM VACUNATORIO V, SOLICITUD S, LOCALIDAD L
                             WHERE V.CODLOC = L.CODLOC
                             AND V.IDVACUNATORIO = S.IDVACUNATORIO
                             AND S.ESTADOSOL = 'Cancelada'
                             AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
                             AND S.FECHAAGENDA <= TO_DATE('15/05/2021', 'DD/MM/YYYY')
                             AND L.LOCALIDAD = 'Montevideo'
                             AND S.VIASOLICITUD <> 'Whatsapp')
```

```

UNION
SELECT DISTINCT V.NOMBREVACUNATORIO
FROM VACUNATORIO V, SOLICITUD S, LOCALIDAD L
WHERE V.CODLOC = L.CODLOC
AND V.IDVACUNATORIO = S.IDVACUNATORIO
AND S.ESTADOSOL = 'Cancelada'
AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
AND S.FECHAAGENDA <= TO_DATE('15/05/2021', 'DD/MM/YYYY')
AND L.LOCALIDAD = 'Montevideo'
AND S.VIASOLICITUD = 'Teléfono'
AND V.NOMBREVACUNATORIO IN (SELECT V.NOMBREVACUNATORIO
                             FROM VACUNATORIO V, SOLICITUD S, LOCALIDAD L
                             WHERE V.CODLOC = L.CODLOC
                             AND V.IDVACUNATORIO = S.IDVACUNATORIO
                             AND S.ESTADOSOL = 'Cancelada'
                             AND S.FECHAAGENDA >= TO_DATE('01/05/2021', 'DD/MM/YYYY')
                             AND S.FECHAAGENDA <= TO_DATE('15/05/2021', 'DD/MM/YYYY')
                             AND L.LOCALIDAD = 'Montevideo'
                             AND S.VIASOLICITUD <> 'Teléfono');

```

Para la resolución de este ejercicio planteamos las distintas combinaciones: En el caso de que el vacunatorio presente solicitudes que cumplen con las condiciones establecidas y fueron mediante vía Web, a través de una subconsulta buscar si se encuentra entre los vacunatorios que tuvieron solicitudes que también cumplieran con dichas condiciones pero fueron realizadas mediante una vía distinta a Web (en este caso particular, WhatsApp o Teléfono), y así asegurar que el vacunatorio cumple con más de una vía de agenda.

Otra posible solución pudo haber sido intersecar 2 a 2 {Web, WhatsApp}, {WhatsApp, Teléfono} {Teléfono, Web}, cada consulta individual; idea aplicada en Álgebra Relacional a continuación:

**AR:** (Notación de Relax)

```

solCanceladasWeb = (π NombreVacunatorio ((σ EstadoSol = 'Cancelada' ∧ FechaAgenda<=date('2020-05-15') ∧ FechaAgenda>=date('2020-05-01') ∧ ViaSolicitud = 'Web' (Solicitud)) ⋈ (σ Localidad = 'MONTEVIDEO' (Localidad ⋈ Vacunatorio))))

```

```

solCanceladasWhatsapp = (π NombreVacunatorio ((σ EstadoSol = 'Cancelada' ∧ FechaAgenda<=date('2020-05-15') ∧ FechaAgenda>=date('2020-05-01') ∧ ViaSolicitud = 'Whatsapp' (Solicitud)) ⋈ (σ Localidad = 'MONTEVIDEO' (Localidad ⋈ Vacunatorio))))

```

```

solCanceladasTelefono = (π NombreVacunatorio ((σ EstadoSol = 'Cancelada' ∧ FechaAgenda<=date('2020-05-15') ∧ FechaAgenda>=date('2020-05-01') ∧ ViaSolicitud = 'Telefono' (Solicitud)) ⋈ (σ Localidad = 'MONTEVIDEO' (Localidad ⋈ Vacunatorio))))

```

```

(solCanceladasWeb ⋈ solCanceladasWhatsapp) ∪ (solCanceladasWhatsapp ⋈ solCanceladasTelefono) ∪ (solCanceladasTelefono ⋈ solCanceladasWeb)

```

## Ejercicio 5

### Problema:

Obtener las localidades que tengan a todos los ciudadanos que hayan solicitado vacunarse y cuya fecha de nacimiento sea entre el 01/01/1991 y el 31/12/1996 con estado de solicitud Finalizada.

### Solución

#### SQL:

```
(
SELECT distinct L.CODLOC, L.LOCALIDAD
FROM LOCALIDAD L, PERSONA P, SOLICITUD S
WHERE L.CODLOC = P.CODLOC
AND S.CI = P.CI
AND P.FECHANAC >= TO_DATE('01/01/1991', 'DD/MM/YYYY')
AND P.FECHANAC <= TO_DATE('31/12/1996', 'DD/MM/YYYY')
AND S.ESTADOSOL = 'Finalizada'
)
MINUS
(
SELECT distinct L.CODLOC, L.LOCALIDAD
FROM LOCALIDAD L, PERSONA P, SOLICITUD S
WHERE L.CODLOC = P.CODLOC
AND S.CI = P.CI
AND S.ESTADOSOL <> 'Finalizada'
)
MINUS
(
SELECT distinct L.CODLOC, L.LOCALIDAD
FROM PERSONA P, LOCALIDAD L, SOLICITUD S
WHERE L.CODLOC = P.CODLOC
AND S.CI = P.CI
AND S.ESTADOSOL = 'Finalizada'
AND (P.FECHANAC < TO_DATE('01/01/1991', 'DD/MM/YYYY')
OR P.FECHANAC > TO_DATE('31/12/1996', 'DD/MM/YYYY'))
);
```

**Datos de prueba:** En archivo .txt

**CR:**

```
{
  L.CODLOC, L.LOCALIDAD |
  LOCALIDAD(L) AND PERSONA(P) AND SOLICITUD (S)
  AND L.CODLOC = P.CODLOC
  AND S.CI = P.CI
  AND P.FECHANAC >= TO_DATE('01/01/1991', 'DD/MM/YYYY')
  AND P.FECHANAC <= TO_DATE('31/12/1996', 'DD/MM/YYYY')
  AND S.ESTADOSOL = 'Finalizada'
  AND (
    not (∃ l) and not (∃ p) and not (∃ s) | WHERE L.CODLOC = P.CODLOC AND S.CI = P.CI AND
    S.ESTADOSOL <> 'Finalizada'
  )
  AND (
    not (∃ l) and not (∃ p) and not (∃ s) |
    WHERE L.CODLOC = P.CODLOC AND S.CI = P.CI AND S.ESTADOSOL = 'Finalizada'
    AND (P.FECHANAC < TO_DATE('01/01/1991', 'DD/MM/YYYY') OR P.FECHANAC >
    TO_DATE ('31/12/1996', 'DD/MM/YYYY'))
  )
}
```

**Justificación:** Nos basamos en traducción de SQL a CR. En el caso de resta: se utiliza el “not”

## Ejercicio 6

### Problema:

Mostrar los datos de las personas y el estado de sus solicitudes. Además, para las solicitudes en estado Agendada o Finalizada, mostrar la fecha de agenda. Para aquellas solicitudes pendientes mostrar el texto “Pendiente de agenda”. No incluir aquellas solicitudes que han sido canceladas.

### Solución

```
SELECT P.*, S.ESTADOSOL, NVL(CAST(S.FECHAAGENDA AS VARCHAR(10)), 'Pendiente de agenda') AS FECHAAGENDA
FROM PERSONA P, SOLICITUD S
WHERE S.CI = P.CI
AND S.ESTADOSOL <> 'Cancelada';
```

**Justificación:** Utilizamos CAST para pasar la fecha de agenda a un Varchar de 10 caracteres, evaluándolo con NVL y en caso de que sea NULL se muestra el texto “Pendiente de Agenda”.

## Ejercicio 7

### Problema:

Obtener para cada localidad la cantidad de solicitudes por cada vía de solicitud. Tener en cuenta solo las localidades que tienen la menor cantidad de tipos de vacunas diferentes. Ordenar el resultado alfabéticamente por nombre de localidad y por cantidad de solicitudes ascendente.

### Solución

```
SELECT L.LOCALIDAD, S.VIASOLICITUD, COUNT(S.VIASOLICITUD)
FROM SOLICITUD S, LOCALIDAD L, VACUNATORIO V
WHERE S.IDVACUNATORIO = V.IDVACUNATORIO
AND V.CODLOC = L.CODLOC
AND V.IDVACUNATORIO IN (SELECT V.IDVACUNATORIO
                        FROM VACUNATORIO V, VACUNATORIOVACUNA W, LOCALIDAD L
                        WHERE V.IDVACUNATORIO = W.IDVACUNATORIO
                        AND V.CODLOC = L.CODLOC
                        GROUP BY V.NOMBREVACUNATORIO, V.IDVACUNATORIO
                        ORDER BY COUNT (W.IDVACUNA)
                        FETCH NEXT 1 ROWS WITH TIES)
GROUP BY L.LOCALIDAD, S.VIASOLICITUD
ORDER BY L.LOCALIDAD;
```

**Justificación:** En la subconsulta realizada, utilizamos un “Fetch next 1 rows” para obtener la primera fila de la cantidad de tipos de vacuna (IdVacuna), la que tiene menor valor, y utilizamos la opción “with ties” para obtener el resto de las filas que tienen el mismo valor (el resultado son todas las filas con menor cantidad de vacunas). Tanto en la subconsulta como al finalizar, utilizamos Group By para agrupar en base a cierta característica, y Order By para ordenarlos.

## Ejercicio 8

### Problema:

Obtener las localidades que tengan la mayor cantidad de personas vacunadas. Se entiende que las personas que recibieron la dosis son aquellas cuya solicitud está finalizada. Considerar solo las localidades que tienen más de 10.000 habitantes.

### Solución

```
SELECT V.LOCALIDAD
FROM SOLICITUD S, VACUNATORIO V, LOCALIDAD L
WHERE S.IDVACUNATORIO = V.IDVACUNATORIO
AND L.CODLOC = V.CODLOC
AND V.CODLOC IN (SELECT CODLOC
                  FROM PERSONA
                  GROUP BY CODLOC
                  HAVING COUNT(*) >= 10000)
GROUP BY V.CODLOC
ORDER BY COUNT(*) DESC
FETCH NEXT 1 ROWS WITH TIES;
```

**Datos de prueba:** En archivo .txt

**Justificación:** Realizamos una subconsulta con las personas agrupadas por localidad para identificar si dicha localidad tiene más de 10.000 habitantes, utilizando Having Count(\*).

## Ejercicio 9

### Problema:

Obtener el nombre de los vacunatorios donde se vacune con la menor cantidad de tipos de vacuna y que dichos vacunatorios hayan tenido la mayor cantidad de solicitudes de agenda en junio de 2021.

### Solución

```
(SELECT V.NOMBREVACUNATORIO
FROM VACUNATORIO V, VACUNATORIOVACUNA W
WHERE V.IDVACUNATORIO = W.IDVACUNATORIO
GROUP BY V.NOMBREVACUNATORIO, V.IDVACUNATORIO
ORDER BY COUNT (W.IDVACUNA)
FETCH NEXT 1 ROWS WITH TIES)
INTERSECT
(SELECT V.NOMBREVACUNATORIO
FROM SOLICITUD S, VACUNATORIO V
WHERE S.IDVACUNATORIO = V.IDVACUNATORIO
AND S.FECHA >= TO_DATE('01/06/2021', 'DD/MM/YYYY')
AND S.FECHA <= TO_DATE('30/06/2021', 'DD/MM/YYYY')
GROUP BY V.NOMBREVACUNATORIO, V.IDVACUNATORIO
ORDER BY COUNT (*)
FETCH NEXT 1 ROWS WITH TIES);
```

**Justificación:** Separamos la consulta en dos partes, intersecándolas entre ellas. En la primera consulta, halla los vacunatorios con menor cantidad de tipos de vacuna, y en la segunda los vacunatorios que hayan tenido la mayor cantidad de solicitudes en el período dado.



## Ejercicio 10

### Problema:

Obtener para cada tipo de vacuna la cantidad total de solicitudes por vía de solicitud. Obtener el porcentaje que representan estas cantidades sobre el total general de solicitudes. Además, obtener el nombre del vacunatorio donde se realizaron la mayor cantidad de solicitudes del tipo de vacuna asociado.

### Solución

```
select tv.codigo, s.viasolicitud, count(*) CantSol, round(100*(count(*) / sum(count(*) over ()), 1) || '%' PrcSol, v.nombrevacunatorio from Solicitud s
join Tipovacuna tv on tv.idvacuna = s.idvacuna
join Vacunatorio v on v.idvacunatorio = s.idvacunatorio
join Vacunatoriovacuna vv on vv.idvacunatorio = s.idvacunatorio
where v.nombrevacunatorio in (select vl.nombrevacunatorio from Vacunatorio vl
                             join Solicitud sl on vl.idvacunatorio = sl.idvacunatorio
                             where sl.idvacuna = s.idvacuna
                             group by vl.nombrevacunatorio
                             having count(*) = (select max(count(*) from Vacunatorio v2
                             join Solicitud s2 on v2.idvacunatorio = s2.idvacunatorio
                             where s2.idvacuna = s.idvacuna
                             group by v2.nombrevacunatorio))
group by tv.codigo, s.viasolicitud, v.nombrevacunatorio
order by tv.codigo, s.viasolicitud, v.nombrevacunatorio;
```

**Datos de prueba:** En archivos .txt