

Pruebas de los métodos ABM y repositorio estadístico

Alberto Esparza Pérez
Grado de Ingeniería Informática
Bases de Datos

Jordi Ferrer Duran
Javier Jiménez Pelayo

Febrero 2023

1. Procedimiento

Tal y como se indica en el capítulo 5 para realizar las pruebas se ha generado un script que, de manera automática, simula una ejecución realista de 3 cursos académicos.

Para ello se hace uso de ciertas estructuras fijas (las inserciones de universidades, titulaciones y empresas) y de bucles con un número aleatorio de iteraciones para las personas responsables de las prácticas, el profesorado, el alumnado, las matriculaciones, candidaturas, entrevistas, colaboraciones, informes e inspecciones.

Para que los nombres de las personas sean medianamente realistas se generan mediante un array de nombres y otro de apellidos seleccionando un nombre completo de manera aleatoria.

Todo el resto de campos se insertan con textos y números aleatorios.

1. Se modifica el curso académico al '20 – '21.
2. Se insertan datos de 10 universidades.
3. Se insertan 14 titulaciones.
4. Se insertan 10 empresas.
5. Se insertan entre 15 y 20 personas responsables de prácticas.

6. Se insertan entre 10 y 20 profesores.
7. Se insertan entre 30 y 40 alumnos.
8. De dichos alumnos se matriculan entre el 80 y el 90% en una titulación.
9. Se presentan entre 30 y 40 ofertas con entre 2 y 5 plazas cada oferta.
10. Cada estudiante matriculado presenta candidatura a entre 1 y 3 ofertas.
11. Se realizan entrevistas de todas las candidaturas con un resultado aleatorio de aptitud.
12. De las entrevistas con resultado apto se generan colaboraciones de prácticas. Cada colaboración tiene entre 3 y 10 informes esperados.
13. Se realizan inspecciones aleatorias a todas las empresas con estudiantes.
14. Para cada colaboración se insertan todos los informes esperados de los que el último es de tipo final con un resultado aleatorio de aptitud de las prácticas.

15. Se modifica el curso académico al '21 – '22.
16. Se repiten los puntos del 6 al 14 aunque en este caso se insertan entre 5 y 10 profesores.

17. Se modifica el curso académico al '22 – '23
18. Se añaden 2 universidades
19. Se repiten los puntos del 6 al 14 aunque en este caso se dejan aleatoriamente algunas colaboraciones sin algunos de los informes para las explotaciones estadísticas.

2. Resultados

Mediante el script anterior se logra disponer de un conjunto de datos suficiente para poder comprobar el funcionamiento del repositorio estadístico y la exactitud de los datos obtenidos.

En la ejecución en la que se basará este anexo se obtuvieron los siguientes datos:

- 12 universidades
- 14 titulaciones
- 10 empresas
- 15 personas responsables de prácticas
- 25 profesores
- 99 estudiantes
- 93 matriculaciones
- 101 ofertas
- 194 candidaturas
- 194 entrevistas
- 101 colaboraciones
- 155 inspecciones
- 609 informes
- 73 informes finales

Para cada una de las consultas del repositorio estadístico se muestran los resultados de la consulta y los de la consulta de control sobre las tablas del modelo normalizado.

- Universidad que, analizando todos los datos de que disponemos, tiene un porcentaje más alto de estudiantes que superan las prácticas en empresas. Se consideran superadas cuando la persona responsable del estudiante en la empresa lo califica como apto.

```
SELECT univ_max_stdnts_sup FROM re_general;
```

	UNIV_MAX_STDNTS_SUP
1	Universidad Pompeu Fabra (UPF)

```
SELECT u.nombre
FROM
      (SELECT      c.universidad_id,      COUNT(1)      total,
SUM(TO_NUMBER(NVL(i.es_aptos,      '0')))      aptos,
NVL(SUM(TO_NUMBER(NVL(i.es_aptos,      '0'))) / NULLIF(COUNT(1), 0), 0)
prcnt
FROM colaboracion c
LEFT JOIN informe_final i ON (
      c.empresa_id = i.empresa_id
      AND c.universidad_id = i.universidad_id
      AND c.estudiante_id = i.estudiante_id
      AND c.oferta_id = i.oferta_id
)
GROUP BY c.universidad_id) prcnt
JOIN universidad u ON prcnt.universidad_id = u.id
ORDER BY prcnt.prcnt DESC, u.id
FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE
1	Universidad Pompeu Fabra (UPF)

- Top5 de empresas que han ofrecido más plazas para estudiantes en prácticas al CIC.

```
SELECT empresas_mas_plazas FROM re_general;
```

	EMPRESAS_MAS_PLAZAS
1	SEAT, Lidl, Grifols, Vueling Airlines, Caprabo

```
SELECT LISTAGG(nombre, ', ')
FROM
      (SELECT o.empresa_id, e.nombre, SUM(o.plazas) plzs
FROM oferta o
JOIN empresa e ON o.empresa_id = e.id
GROUP BY o.empresa_id, e.nombre
ORDER BY SUM(o.plazas) DESC, o.empresa_id
FETCH FIRST 5 ROWS ONLY);
```

	LISTAGG(NOMBRE,',')
1	SEAT, Caprabo, Vueling Airlines, Grifols, Lidl

- Empresa que, entre todos los datos de que disponemos, ha recibido más inspecciones de trabajo (sin considerar el resultado de la inspección).

```
SELECT empresa_mas_inspecciones FROM re_general;
```

	EMPRESA_MAS_INSPECCIONES
1	SEAT

```
SELECT e.nombre
FROM inspeccion i
JOIN empresa e ON i.empresa_id = e.id
GROUP BY i.empresa_id, e.nombre
ORDER BY COUNT(1) DESC, i.empresa_id
FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE
1	SEAT

- Dado un año universitario en concreto, porcentaje de estudiantes, de cualquier universidad, que no superan las prácticas en empresas. *Se muestran los resultados para los 3 años de los que hay datos.

```
SELECT prcnt_no_superan FROM re_any_univ WHERE anyo_univ = '<AÑO-UNIVERSITARIO>';
```

	ANYO_UNIV	PRCNT_NO_SUPERAN
1	21-22	53,85
2	22-23	13,16
3	20-21	62,5

```
SELECT NVL(SUM(DECODE(i.es_apto, '0', 1, 0)) / NULLIF(COUNT(1), 0) * 100, 0)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
LEFT JOIN informe_final i ON (
  c.empresa_id = i.empresa_id
  AND c.universidad_id = i.universidad_id
  AND c.estudiante_id = i.estudiante_id
  AND c.oferta_id = i.oferta_id
)
WHERE o.anyo_acad = '<AÑO-UNIVERSITARIO>';
```

	'20-21'	NVL(SUM(D	'21-22'	NVL(SUM(DEC	'22-23'	NVL(SUM(DECC
1	20-21	62,5	21-22	53,846	22-23	13,158

- En el año universitario en curso, sueldo medio que reciben los estudiantes durante sus prácticas.

```
SELECT sueldo_medio FROM re_any_actual;
```

	SUELDO_MEDIO
1	692

```
SELECT NVL(FLOOR(AVG(c.sueldo)), 0)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
WHERE o.anyo_acad = anyoActual();
```

	NVL(FLOOR(AVG(C.SUELDO)),0)
1	692

- Número total de estudiantes en prácticas en el momento de ejecutar la consulta. Se considerarán los estudiantes que han iniciado las prácticas y no las han acabado.

```
SELECT num_estudiantes_actual FROM re_any_actual;
```

	NUM_ESTUDIANTES_ACTUAL
1	28

```
SELECT COUNT(1)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
WHERE o.anyo_acad = anyoActual()
AND NOT EXISTS (SELECT 1
FROM informe_final i
WHERE c.empresa_id = i.empresa_id
AND c.universidad_id = i.universidad_id
AND c.estudiante_id = i.estudiante_id
AND c.oferta_id = i.oferta_id);
```

	COUNT(1)
1	28

- Durante el último año universitario, porcentaje de universidades controladas por el CIC que no tienen estudiantes en prácticas.

```
SELECT prcnt_no_stdnts FROM re_any_actual;
```

	PRCNT_NO_STDNTS
1	25

```
SELECT NVL((COUNT(1) - COUNT(universidad_id)) / NULLIF(COUNT(1), 0) * 100, 0)
FROM universidad u
LEFT JOIN (
  SELECT DISTINCT c.universidad_id
  FROM colaboracion c
  JOIN oferta o ON c.oferta_id = o.id
  LEFT JOIN universidad u ON c.universidad_id = u.id
  WHERE o.anyo_acad = anyoActual()
) prt ON u.id = prt.universidad_id;
```

	NVL((COUNT(1)-COUNT(UNIVERSIDAD_ID))/NULLIF(COUNT(1),0)*100,0)
1	25

- Titulación universitaria con un mayor número de estudiantes que han acabado satisfactoriamente las prácticas en empresas. Se tendrán en cuenta todos los datos de que se dispone en la BD.

```
SELECT titulacion FROM re_general;
```

	TITULACION
1	Ingeniería Náutica y Transporte Marítimo

```
SELECT t.nombre
FROM informe_final i
JOIN oferta o ON i.oferta_id = o.id
JOIN cursa c ON (
  i.universidad_id = c.universidad_id
  AND i.estudiante_id = c.estudiante_id
  AND o.anyo_acad = c.anyo_acad
)
JOIN titulacion t ON c.titulacion_id = t.codigo
WHERE i.es_apto = '1'
GROUP BY t.codigo, t.nombre
ORDER BY COUNT(1) DESC, t.codigo
FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE
1	Ingeniería Náutica y Transporte Marítimo

- Número medio de estudiantes que se presentan a cada oferta recibida por las diferentes universidades durante el año universitario en curso.

```
SELECT media_candidat_ofert FROM re_any_actual;
```

	MEDIA_CANDIDAT_OFERT
1	2,31

```
SELECT NVL(COUNT(c.oferta_id) / NULLIF(COUNT(DISTINCT o.id), 0), 0)
FROM oferta o
LEFT JOIN candidatura c ON o.id = c.oferta_id
WHERE o.anyo_acad = anyoActual();
```

	NVL(COUNT(C.OFERTA_ID)/NULLIF(COUNT(DISTINCTO.ID),0),0)
1	2,313

- Teniendo en cuenta todos los datos de que se dispone, duración media de los contratos de estudiantes con las empresas.

```
SELECT duracion_media_contrato FROM re_general;
```

	DURACION_MEDIA_CONTRATO
1	9,1

```
SELECT NVL(AVG(MONTHS_BETWEEN(fecha_final, fecha_inicio)), 0)
FROM colaboracion;
```

	NVL(AVG(MONTHS_BETWEEN(FECHA_FINAL,FECHA_INICIO)),0)
1	9,059

- Empresa que, en un año en concreto, ha realizado el mayor número de entrevistas. *Se muestran los resultados para los 3 años de los que hay datos.

```
SELECT empresa_max_intrvws FROM re_any_univ WHERE anyo_univ = '<AÑO-UNIVERSITARIO>';
```

	ANYO_UNIV	EMPRESA_MAX_INTRVWS
1	21-22	SEAT
2	22-23	Caprabo
3	20-21	Caprabo

```
SELECT m.nombre
FROM entrevista e
JOIN oferta o ON e.oferta_id = o.id
JOIN empresa m ON e.empresa_id = m.id
WHERE o.anyo_acad = '<AÑO-UNIVERSITARIO>'
GROUP BY m.id, m.nombre
ORDER BY COUNT(1) DESC, m.id
FETCH FIRST 1 ROWS ONLY;
```

	'20-21'	NOMBRE	'21-22'	NOMBRE	'22-23'	NOMBRE
1	20-21	Caprabo	21-22	SEAT	22-23	Caprabo

- En el año universitario en curso, profesor que lleva un mayor número de estudiantes que están realizando prácticas en empresas.

```
SELECT profesor_max_stdnts FROM re_any_actual;
```

	PROFESOR_MAX_STDNTS
1	Plinio Osuna Enríquez

```
SELECT TRIM(p.nombre || ' ' || p.apellido1 || ' ' || p.apellido2)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
JOIN cursa cr ON (
    c.universidad_id = cr.universidad_id
    AND c.estudiante_id = cr.estudiante_id
    AND o.anyo_acad = cr.anyo_acad
)
JOIN profesor_responsable p ON cr.profesor_id = p.id
WHERE o.anyo_acad = anyoActual()
GROUP BY p.id, p.nombre, p.apellido1, p.apellido2
ORDER BY COUNT(1) DESC, p.id
FETCH FIRST 1 ROWS ONLY;
```

	TRIM(P.NOMBRE " " P.APELLIDO1 " " P.APELLIDO2)
1	Plinio Osuna Enríquez

- En el momento de ejecutar la consulta, número total de informes de prácticas pendientes de ser realizados por las personas responsables de las empresas.

```
SELECT num_informes_pendents FROM re_general;
```

	NUM_INFORMES_PENDENTS
1	118

```
SELECT SUM(c.informes_esperados - NVL(i.num, 0))
FROM colaboracion c
LEFT JOIN (
    SELECT empresa_id, universidad_id, estudiante_id, oferta_id,
COUNT(1) num
    FROM informe
    GROUP BY empresa_id, universidad_id, estudiante_id, oferta_id
) i ON (
    c.empresa_id = i.empresa_id
    AND c.universidad_id = i.universidad_id
    AND c.estudiante_id = i.estudiante_id
    AND c.oferta_id = i.oferta_id
);
```

	SUM(C.INFORMES_ESPERADOS-NVL(I.NUM,0))
1	118

- Teniendo en cuenta el año universitario en curso y el anterior, porcentaje de empresas que ofrecieron plazas el año anterior y no las han ofrecido en el actual.

```
SELECT prcnt_emp_no_plazas_any_act FROM re_general;
```

	PRCNT_EMP_NO_PLAZAS_ANY_ACT
1	0

```
SELECT NVL(SUM(hits) / NULLIF(COUNT(1), 0) * 100, 0)
FROM (
    SELECT e.id, CASE WHEN MAX(Decode(o.anyo_acad,
TO_CHAR(TO_NUMBER(SUBSTR(anyoActual(), 1, 2)) - 1) || '-' ||
SUBSTR(anyoActual(), 1, 2), 1, 0)) - MAX(Decode(o.anyo_acad,
anyoActual(), 1, 0)) > 0 THEN 1 ELSE 0 END hits
    FROM empresa e
    LEFT JOIN oferta o ON e.id = o.empresa_id
    GROUP BY e.id
);
```

	NVL(SUM(HITS)/NULLIF(COUNT(1),0)*100,0)
1	0

3.Métodos de actualización

Una vez comprobado que mediante una serie de inserciones realistas el repositorio se comporta de la manera adecuada se realizan modificaciones y eliminaciones sobre los datos que mayor impacto puedan tener sobre el repositorio estadístico.

Se modifica el nombre de la universidad que tiene un porcentaje más alto de estudiantes que superan las prácticas en empresas.

```
SELECT univ_max_stdnts_sup FROM re_general;
```

```
UNIV_MAX_STDNTS_SUP
-----
Universidad Pompeu Fabra (UPF)
```

```
Transcurrido: 00:00:00.002
1 filas seleccionadas.
```

```
BEGIN
      DBMS_OUTPUT.PUT_LINE(pq_abm.modUniversidad(4, 'Nuevo nombre',
'Lorem ipsum', 'dolor sit amet'));
END;
/
```

OK

```
Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.012
```

```
SELECT * FROM universidad WHERE id = 4;
```

```
ID NOMBRE          DIRECCION  WEB
-----
4 Nuevo nombre Lorem ipsum dolor sit amet
```

```
Transcurrido: 00:00:00.002
1 filas seleccionadas.
```

```
SELECT univ_max_stdnts_sup FROM re_general;
```

```
UNIV_MAX_STDNTS_SUP
-----
Nuevo nombre
```

```
Transcurrido: 00:00:00.002
1 filas seleccionadas.
```

Se modifica la titulación universitaria con un mayor número de estudiantes que han acabado satisfactoriamente las prácticas en empresas.

```
SELECT titulacion FROM re_general;
```

```
TITULACION
-----
Ingeniería Náutica y Transporte Marítimo
```

```
Transcurrido: 00:00:00.002
1 filas seleccionadas.
```

```
BEGIN
      DBMS_OUTPUT.PUT_LINE(pq_abm.modTitulacion('COD08', 'Derecho'));
END;
/
```

OK

```
Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.022
```

```
SELECT * FROM titulacion WHERE codigo = 'COD08';
```

```
CODIGO NOMBRE
-----
COD08  Derecho
```

```
Transcurrido: 00:00:00.006
1 filas seleccionadas.
```

```
SELECT titulacion FROM re_general;
```

```
TITULACION
-----
Derecho
```

```
Transcurrido: 00:00:00.002
1 filas seleccionadas.
```

Se modifica el nombre del profesor que lleva un mayor número de estudiantes que están realizando prácticas en empresas en el año universitario en curso.

```
SELECT profesor_max_stdnts FROM re_any_actual;
```

```
PROFESOR_MAX_STDNTS
-----
Plinio Osuna Enríquez
```

```
Transcurrido: 00:00:00.002
1 filas seleccionadas.
```

```
BEGIN
      DBMS_OUTPUT.PUT_LINE(pq_abm.modProfesor (6, 18, 'Nombre',
'Apellido1', NULL));
END;
/
```

OK

```
Procedimiento PL/SQL terminado correctamente.
```

Transcurrido: 00:00:00.027

```
SELECT * FROM profesor_responsable WHERE id = 18;
```

```
UNIVERSIDAD_ID ID NOMBRE APELLIDO1 APELLIDO2
-----
          6 18 Nombre Apellido1
```

Transcurrido: 00:00:00.002

1 filas seleccionadas.

```
SELECT profesor_max_stdnts FROM re_any_actual;
```

```
PROFESOR_MAX_STDNTS
-----
Nombre Apellido1
```

Transcurrido: 00:00:00.001

1 filas seleccionadas.

Se modifica el nombre de un estudiante

```
SELECT * FROM estudiante WHERE id = 7;
```

```
UNIVERSIDAD_ID ID NOMBRE APELLIDO1 APELLIDO2
-----
          5 7 Simón Tomas Roma
```

Transcurrido: 00:00:00.002

1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.modEstudiante (5, 7, 'Estudiante',
'Pruebas', 'Probando') );
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.

Transcurrido: 00:00:00.185

```
SELECT * FROM estudiante WHERE id = 7;
```

```
UNIVERSIDAD_ID ID NOMBRE APELLIDO1 APELLIDO2
-----
          5 7 Estudiante Pruebas Probando
```

Transcurrido: 00:00:00.002

1 filas seleccionadas.

Se intenta eliminar una matriculación vinculada a una candidatura

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(pq_abm.bajaCursa (7, 'COD05', 1, 3) );
END;
/
```

ERROR EXISTEN REGISTROS VINCULADOS

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.406

Se elimina una matriculación nueva creada ex profeso para la prueba.

```
BEGIN
DBMS_OUTPUT.PUT_LINE(pq_abm.bajaCursa(1, 'COD14', 2, 78));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.582

Se modifica el nombre de la empresa que más inspecciones de trabajo ha recibido, también una de las 5 que más plazas ha ofrecido.

```
SELECT empresas_mas_plazas FROM re_general;
```

EMPRESAS_MAS_PLAZAS

```
-----
SEAT, Lidl, Grifols, Vueling Airlines, Caprabo
```

Transcurrido: 00:00:00.003
1 filas seleccionadas.

```
SELECT empresa_mas_inspecciones FROM re_general;
```

EMPRESA_MAS_INSPECCIONES

```
-----
SEAT
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
DBMS_OUTPUT.PUT_LINE(pq_abm.modEmpresa (1, 'CIF', 'NomEmpresa',
'Dirección', 'Lorem', 'Ipsum') );
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.036

```
SELECT * FROM empresa WHERE id = 1;
```

```
ID CIF NOMBRE DIRECCION TELEFONO RESPONSABLE_RRHH
```

```
-----  
1 CIF NomEmpresa Dirección Lorem Ipsum
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT empresas_mas_plazas FROM re_general;
```

```
EMPRESAS_MAS_PLAZAS  
-----  
NomEmpresa, Vueling Airlines, Caprabo, Lidl, Grifols
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT empresa_mas_inspecciones FROM re_general;
```

```
EMPRESA_MAS_INSPECCIONES  
-----  
NomEmpresa
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se modifican los datos de una persona responsable de las prácticas en una empresa.

```
SELECT * FROM solicitante WHERE id = 7;
```

```
EMPRESA_ID ID NOMBRE CARGO  
-----  
1 7 Aránzazu Gutierrez Alegria IHsGBzIQRSTqML
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN  
    DBMS_OUTPUT.PUT_LINE(pq_abm.modSolicitante (1, 7, 'María Pombo  
Bernat', 'Dirección') );  
END;  
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.019

```
SELECT * FROM solicitante WHERE id = 7;
```

```
EMPRESA_ID ID NOMBRE CARGO  
-----  
1 7 María Pombo Bernat Dirección
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se intenta eliminar a una persona responsable vinculada a una oferta.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaSolicitante (1, 7) );
END;
/
```

ERROR EXISTEN REGISTROS VINCULADOS

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.016

Se elimina una persona responsable creada ex profeso para la prueba.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaSolicitante (1, 21) );
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.019

Se modifica una oferta aumentando el número de plazas para que otra empresa entre al Top 5 de empresas ofertantes.

```
SELECT empresas_mas_plazas FROM re_general;
```

EMPRESAS_MAS_PLAZAS

```
-----
NomEmpresa, Vueling Airlines, PRUEBAS, Lidl, Grifols
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
SELECT * FROM oferta where id = 59;
```

```
ID EMPRESA_ID SOLICITANTE_ID DESCRIPCION          REQUISITOS
ANYO_ACAD PLAZAS
-- -----
-----
59          9          5 QXZHBtLoflaIKEiHscOGIPDAj0vUcf
21-22      3
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.modOferta (59, 'DESC', 'Idiomas:
Alemán', 99) );
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.151

```
SELECT * FROM oferta where id = 59;
```

ID	EMPRESA_ID	SOLICITANTE_ID	DESCRIPCION	REQUISITOS	ANYO_ACAD
59	9		5 DESC	Idiomas: Alemán	21-22

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT empresas_mas_plazas FROM re_general;
```

```
EMPRESAS_MAS_PLAZAS
-----
Puig, Grifols, Vueling Airlines, PRUEBAS, NomEmpresa
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se intenta eliminar una oferta que ya ha recibido candidaturas

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaOferta (97) );
END;
/
```

ERROR EXISTEN REGISTROS VINCULADOS

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.018

Se elimina una oferta sin candidaturas

```
SELECT empresas_mas_plazas FROM re_general;
```

```
EMPRESAS_MAS_PLAZAS
-----
Puig, Grifols, Vueling Airlines, PRUEBAS, NomEmpresa
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT * FROM oferta where id = 59;
```

ID	EMPRESA_ID	SOLICITANTE_ID	DESCRIPCION	REQUISITOS	ANYO_ACAD
----	------------	----------------	-------------	------------	-----------

59
99

9

5 DESC

Idiomas: Alemán 21-22

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaOferta (59) );
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.031

```
SELECT * FROM oferta where id = 59;
```

No se ha seleccionado ninguna fila
0 filas seleccionadas.

```
SELECT empresas_mas_plazas FROM re_general;
```

EMPRESAS_MAS_PLAZAS

```
-----
NomEmpresa, Lidl, Grifols, Vueling Airlines, PRUEBAS
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

Se modifica una candidatura existente

```
SELECT * FROM candidatura WHERE universidad_id = 4 AND estudiante_id =
73 AND oferta_id = 101;
```

```
UNIVERSIDAD_ID ESTUDIANTE_ID OFERTA_ID FECHA
-----
                4                73                101 14/2/2023, 0:00:00
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.modCandidatura (4, 73, 101,
TO_DATE('25/03/2023', 'dd/mm/yyyy')) );
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.020

```
SELECT * FROM candidatura WHERE universidad_id = 4 AND estudiante_id =
73 AND oferta_id = 101;
```

```
UNIVERSIDAD_ID ESTUDIANTE_ID OFERTA_ID FECHA
-----
```

4 73 101 25/3/2023, 0:00:00

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se intenta eliminar una candidatura que ya ha pasado por la entrevista.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaCandidatura (4, 73, 101) );
END;
/
```

ERROR EXISTEN REGISTROS VINCULADOS

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.026

Se elimina una candidatura creada ex profeso para la prueba.

```
SELECT media_candidat_ofert FROM re_any_actual;
```

```
MEDIA_CANDIDAT_OFERT
-----
                2.42
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaCandidatura(1, 90, 101));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.011

```
SELECT media_candidat_ofert FROM re_any_actual;
```

```
MEDIA_CANDIDAT_OFERT
-----
                2.39
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

Se intenta cambiar la aptitud de una entrevista favorable con colaboración asociada.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.modEntrevista (4, 73, 101, 1, 3,
  TO_DATE('01/11/2022', 'dd/mm/yyyy'), 'Comments', '0') );
END;
```


Se elimina una entrevista

```
SELECT empresa_max_intrvws FROM re_any_univ where anyo_univ =  
anyoActual();
```

```
EMPRESA_MAX_INTRVWS
```

```
-----
```

```
PRUEBAS
```

```
Transcurrido: 00:00:00.001
```

```
1 filas seleccionadas.
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE(pq_abm.bajaEntrevista (2, 84, 87, 6, 11) );
```

```
END;
```

```
/
```

```
OK
```

```
Procedimiento PL/SQL terminado correctamente.
```

```
Transcurrido: 00:00:00.015
```

```
SELECT empresa_max_intrvws FROM re_any_univ where anyo_univ =  
anyoActual();
```

```
EMPRESA_MAX_INTRVWS
```

```
-----
```

```
NomEmpresa
```

```
Transcurrido: 00:00:00.002
```

```
1 filas seleccionadas.
```

Se modifica una colaboración

```
SELECT sueldo_medio FROM re_any_actual;
```

```
SUELDO_MEDIO
```

```
-----
```

```
692
```

```
Transcurrido: 00:00:00.003
```

```
1 filas seleccionadas.
```

```
SELECT duracion_media_contrato, num_informes_pendents FROM re_general;
```

```
DURACION_MEDIA_CONTRATO NUM_INFORMES_PENDENTS
```

```
-----
```

```
9.1
```

```
118
```

```
Transcurrido: 00:00:00.008
```

```
1 filas seleccionadas.
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(pq_abm.modColaboracion (1, 6, 86, 101,
TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('31/12/2023',
'dd/mm/yyyy'), 9000, 99));
```

```
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.038

```
SELECT sueldo_medio FROM re_any_actual;
```

```
SUELDO_MEDIO
-----
          916
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT duracion_media_contrato, num_informes_pendents FROM re_general;
```

```
DURACION_MEDIA_CONTRATO NUM_INFORMES_PENDENTS
-----
                   9.2                   210
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

Se elimina una colaboración

```
SELECT prcnt_no_superan FROM re_any_univ WHERE anyo_univ =
anyoActual();
```

```
PRCNT_NO_SUPERAN
-----
          13.16
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
SELECT sueldo_medio, prcnt_no_stdnts, profesor_max_stdnts FROM
re_any_actual;
```

```
SUELDO_MEDIO PRCNT_NO_STDNTS PROFESOR_MAX_STDNTS
-----
          916                25 Nombre Apellido1
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
SELECT duracion_media_contrato, num_informes_pendents,
univ_max_stdnts_sup FROM re_general;
```

```
DURACION_MEDIA_CONTRATO NUM_INFORMES_PENDENTS UNIV_MAX_STDNTS_SUP
```

9.2 211 Nuevo nombre

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaColaboracion (9, 10, 76, 97));
```

```
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.645

```
SELECT prcnt_no_superan FROM re_any_univ WHERE anyo_univ =
anyoActual();
```

```
PRCNT_NO_SUPERAN
-----
          13.51
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT sueldo_medio, prcnt_no_stdnts, profesor_max_stdnts FROM
re_any_actual;
```

```
SUELDO_MEDIO PRCNT_NO_STDNTS PROFESOR_MAX_STDNTS
-----
          914          25 Nombre Apellido1
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
SELECT duracion_media_contrato, num_informes_pendents,
univ_max_stdnts_sup FROM re_general;
```

```
DURACION_MEDIA_CONTRATO NUM_INFORMES_PENDENTS UNIV_MAX_STDNTS_SUP
-----
          9.2          204 Nuevo nombre
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

Se modifica una inspección a una empresa

```
SELECT * FROM inspeccion WHERE id = 11;
```

```
ID EMPRESA_ID FECHA MUESTREOS
-----
11          1 26/10/2022, 0:00:00 rtjaTxsXdwXRNqprneEIaMISbfVDhM
```

Transcurrido: 00:00:00.003
1 filas seleccionadas.

```
BEGIN
          DBMS_OUTPUT.PUT_LINE(pq_abm.modInspeccion(11,
TO_DATE('15/03/2022', 'dd/mm/yyyy'), 'Muestras'));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.025

```
SELECT * FROM inspeccion WHERE id = 11;
```

ID	EMPRESA_ID	FECHA	MUESTREOS
11	1	15/3/2022, 0:00:00	Muestras

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se elimina una inspección

```
SELECT empresa_mas_inspecciones FROM re_general;
```

EMPRESA_MAS_INSPECCIONES
NomEmpresa

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
BEGIN
          DBMS_OUTPUT.PUT_LINE(pq_abm.bajaInspeccion(14));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.022

```
SELECT empresa_mas_inspecciones FROM re_general;
```

EMPRESA_MAS_INSPECCIONES
Grifols

Transcurrido: 00:00:00.001
1 filas seleccionadas.

Se modifica un informe


```
DBMS_OUTPUT.PUT_LINE(pq_abm.bajaInforme(1, 6, 86, 101, 3,
TO_DATE('12/6/2023', 'dd/mm/yyyy')));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.018

```
SELECT num_informes_pendents FROM re_general;
```

```
NUM_INFORMES_PENDENTS
-----
                        205
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se modifica un informe final pasándolo de no apto a apto.

```
SELECT * FROM re_universidades WHERE id = 4;
```

```
ID STDNTS_EN_PRACT STDNTS_SUPERAN PRCNT_STDNTS_SUP
-- -----
4          11          7          63.64
```

Transcurrido: 00:00:00.003
1 filas seleccionadas.

```
SELECT * FROM re_titulaciones WHERE codigo = 'COD07';
```

```
CODIGO ESTUDIANTES_APTOS
-----
COD07          0
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(pq_abm.modInformeFinal (1, 4, 73, 101, 3,
TO_DATE('25/06/2023', 'dd/mm/yyyy'), 'Nuevo Coment', 'Nueva val',
'1'));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.243

```
SELECT * FROM re_universidades WHERE id = 4;
```

```
ID STDNTS_EN_PRACT STDNTS_SUPERAN PRCNT_STDNTS_SUP
-- -----
4          11          8          72.73
```

Transcurrido: 00:00:00.003
1 filas seleccionadas.

```
SELECT * FROM re_titulaciones WHERE codigo = 'COD07';
```

CODIGO	ESTUDIANTES	APTOS
COD07		1

Transcurrido: 00:00:00.003
1 filas seleccionadas.

Se modifica un informe final pasándolo de apto a no apto.

```
SELECT * FROM re_universidades WHERE id = 4;
```

ID	STDNTS_EN_PRACT	STDNTS_SUPERAN	PRCNT_STDNTS_SUP
4	11	8	72.73

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT * FROM re_titulaciones WHERE codigo = 'COD01';
```

CODIGO	ESTUDIANTES	APTOS
COD01		1

Transcurrido: 00:00:00.008
1 filas seleccionadas.

```
BEGIN
    DBMS_OUTPUT.PUT_LINE(pq_abm.modInformeFinal (3, 4, 98, 89, 10,
TO_DATE('26/06/2023', 'dd/mm/yyyy'), 'Nuevo Coment', 'Nueva val',
'0'));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.023

```
SELECT * FROM re_universidades WHERE id = 4;
```

ID	STDNTS_EN_PRACT	STDNTS_SUPERAN	PRCNT_STDNTS_SUP
4	11	7	63.64

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT * FROM re_titulaciones WHERE codigo = 'COD01';
```

```
CODIGO ESTUDIANTES_APTOS
-----
COD01                      0
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

Se elimina un informe final con resultado de aptitud.

```
SELECT * FROM re_universidades WHERE id = 5;
```

```
ID STDNTS_EN_PRACT STDNTS_SUPERAN PRCNT_STDNTS_SUP
-----
5                      27                      11                      40.74
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
SELECT * FROM re_titulaciones WHERE codigo = 'COD02';
```

```
CODIGO ESTUDIANTES_APTOS
-----
COD02                      3
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(pq_abm.bajaInformeFinal (3, 5, 25, 89, 10,
TO_DATE('18/03/2023', 'dd/mm/yyyy')));
END;
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.018

```
SELECT * FROM re_universidades WHERE id = 5;
```

```
ID STDNTS_EN_PRACT STDNTS_SUPERAN PRCNT_STDNTS_SUP
-----
5                      27                      10                      37.04
```

Transcurrido: 00:00:00.001
1 filas seleccionadas.

```
SELECT * FROM re_titulaciones WHERE codigo = 'COD02';
```

```
CODIGO ESTUDIANTES_APTOS
-----
COD02                      2
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

Se elimina un informe final con resultado de no aptitud.

```
SELECT prcnt_no_superan FROM re_any_univ WHERE anyo_univ =  
anyoActual();
```

```
PRCNT_NO_SUPERAN  
-----  
10.81
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

```
BEGIN  
    DBMS_OUTPUT.PUT_LINE(pq_abm.bajaInformeFinal (6, 7, 60, 93, 15,  
TO_DATE('08/08/2023', 'dd/mm/yyyy')));  
END;  
/
```

OK

Procedimiento PL/SQL terminado correctamente.
Transcurrido: 00:00:00.017

```
SELECT prcnt_no_superan FROM re_any_univ WHERE anyo_univ =  
anyoActual();
```

```
PRCNT_NO_SUPERAN  
-----  
8.11
```

Transcurrido: 00:00:00.002
1 filas seleccionadas.

4. Conclusiones

Mediante este juego de pruebas se ha podido comprobar cómo los datos del repositorio estadístico se mantienen consistentes en todo momento.

Se ha demostrado cómo los resultados de las consultas simples (en tiempo constante 1) sobre el repositorio estadístico coinciden con los resultados de las consultas con uniones y funciones agregadas.

Se ha demostrado también que las operaciones de modificación y eliminación de datos realizan comprobaciones de integridad y de coherencia.