

22511 - Bases de datos para *Data Warehousing*

Actividad P2: De la creación a la manipulación de una base de datos relacional

TechShowEvents es una empresa especializada en la gestión del control de acceso para eventos organizados por empresas tecnológicas. Su actividad principal se centra en la organización de encuentros anuales de clientes que incluyen actividades como *networking*, conferencias, presentaciones de productos y más.

Recientemente, el responsable de TechShowEvents ha recibido el encargo de gestionar una convención de 4 días para una *startup* que ofrece soluciones *cloud*. Este evento contará con múltiples actividades programadas, la *startup* proporcionará un listado detallado de los eventos, los datos de los participantes y las invitaciones asignadas a cada uno.

Con esta información, TechShowEvents será responsable de implementar un sistema eficaz para controlar el acceso a los eventos programados. Esto garantizará una experiencia óptima para los asistentes y permitirá reportar toda la información necesaria a la empresa contratante.

Con este objetivo en mente, se ha puesto en contacto con la UOC, solicitando que los estudiantes de la asignatura **Bases de datos para *Data Warehousing*** se encarguen de desarrollar esta solución.

Consideraciones para la entrega y realización de la actividad:

- Todo lo que se pide en esta actividad está explicado en el reto 2 (salvo que se trate de un ejercicio de investigación, cuyo enunciado lo especificará). No es necesario adelantar el estudio del material de otros retos para la realización de esta actividad.
- Se recomienda la utilización de **pgAdmin** para la implementación de toda la actividad. Existe otra alternativa que es **psql** (línea de comandos), pero es preferible que utilicéis pgAdmin, ya que es una interfaz gráfica que os permitirá editar y crear sentencias SQL (así como mostrar los resultados) de forma más sencilla que **psql**.
- Tal y como se indica en el enunciado, la respuesta de cada ejercicio tiene que entregarse en un fichero **.sql** diferente, con el nombre correspondiente. Se evaluará el código entregado en estos ficheros **.sql** y **NO el código que aparezca en el documento o en los pantallazos adjuntos**.
- Las capturas de pantalla de los ejercicios (y explicaciones pertinentes) han de proporcionarse en un documento aparte en PDF (se proporciona una plantilla para el caso, **indicad vuestro nombre en el documento**, por favor).

Consideraciones para la evaluación de la actividad:

- Los **scripts** proporcionados por el estudiante con las soluciones de los ejercicios han de ejecutarse correctamente. El estudiante ha de asegurarse de que lanzando el **script** completo de cada ejercicio no se produzca ningún error.

- **Importante:** Las sentencias SQL proporcionadas en los *scripts* han de ser creadas de forma manual y no mediante asistentes que PostgreSQL/pgAdmin pueda proporcionar. Se pretende aprender SQL y no la utilización de asistentes.
- **Adicionalmente:**
En esta actividad **no está permitido el uso de herramientas de inteligencia artificial**. En el plan docente y en la web sobre integridad académica y plagio de la UOC encontraréis información sobre qué se considera conducta irregular en la evaluación y las consecuencias que puede tener.

Las sentencias SQL proporcionadas en los ejercicios han de ser **solamente** aquellas que pide el enunciado y ninguna otra más. Cualquier sentencia innecesaria, si está mal o provoca que el *script* no se ejecute correctamente a la hora de corregirlo, penalizará la calificación del ejercicio.

Esta actividad tiene un peso de **25 puntos del total de todas las actividades**. En cada una de las 4 preguntas de esta actividad tenéis anotado su valor en puntos.

Pregunta 1 (6 puntos)

a) Se pide proporcionar las sentencias SQL para crear los componentes especificados a continuación. El resultado de este ejercicio ha de entregarse en el fichero **Actividad2_ej1.sql**.

dbdw_pec2: Nombre de la base de datos donde se deben de crear el esquema y los componentes requeridos. El comando de creación de la base de datos ha de proporcionarse **comentado** en el *script*.

erp: Nombre del esquema de base de datos donde se deben de crear los componentes especificados a continuación.

Descripción de las tablas:

tb_guest_type

Esta tabla está diseñada para almacenar información sobre las diferentes clasificaciones de los invitados que asistirán a los eventos, permitiendo la implementación de reglas específicas para su acceso a los eventos según su tipo. Además del identificador único, se debe registrar el nombre de la clasificación.

Tabla: tb_guest_type Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
guest_type_id	Cadena de 3 caracteres	No	Sí		
name	Cadena de 40 caracteres variable	No	No		

tb_guest

Esta tabla almacena los datos de los invitados registrados para asistir a la convención. Los campos necesarios incluyen: un identificador único, el nombre completo, correo electrónico (**que debe ser único**), teléfono, fecha de nacimiento y la clasificación asignada al invitado.

Tabla: tb_guest Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
guest_id	Número entero	No	Sí		

name	Cadena de 40 caracteres variable	No	No		
email	Cadena de 40 caracteres variable	No	No		
phone_number	Cadena de 40 caracteres variable	No	No		
date_of_birth	Fecha (sin tiempo)	Sí	No		
guest_type_id	Cadena de 3 caracteres	Sí	No	tb_guest_type	

tb_event

Esta tabla almacena la información clave de los eventos que se gestionarán durante la convención. Los campos necesarios incluyen: un identificador único, el nombre del evento, la fecha en que se llevará a cabo, el número máximo de participantes permitidos, dónde se realiza el evento y el evento padre. **El campo de fecha del evento debe contener siempre una fecha superior a 01 de enero de 2025.**

Tabla: tb_event Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
event_id	Numérico entero	No	Sí		
name	Cadena de 100 caracteres variable	No	No		
event_date	Fecha (sin tiempo)	No	No		
max_guest_count	Numérico entero	No	No		
event_location	Cadena de 50 caracteres variable	No	No		
event_parent	Numérico entero	Sí	No	tb_event	

¿Qué es el evento padre?

Cuando hablamos de un evento padre dentro de un evento, nos referimos a que, para asistir, es necesario superar dos controles de acceso: el del evento principal y el del evento específico al que se desea ingresar.

En nuestro caso, los días 4 y 5 de marzo se llevarán a cabo los eventos “General Entrance Day 1” y “General Entrance Day 2”, los cuales permiten la entrada al recinto de la feria (Exhibition Hall PII) mediante su propio control de acceso. Dentro de este evento principal, se llevarán a cabo conferencias. Para asistir a cualquiera de ellas, será necesario pasar un segundo control de acceso,

de manera que si invitamos a una persona a una de las conferencias, también le debemos dar acceso a la entrada del recinto de la feria.

tb_event_guest

Tabla destinada a almacenar la información de a qué eventos pueden acceder los invitados, los campos son el identificador de evento, el identificador de invitado y si el invitado ha confirmado o no la asistencia, de manera que si el valor es S ha confirmado la asistencia, si el valor es N indica que ha rechazado la invitación y si es nulo significa que aún no ha respondido.

Tabla: tb_event_guest Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
event_id	Numérico entero	No	Sí	tb_event	
guest_id	Numérico entero	No	Sí	tb_guest	
confirmed	Cadena de 1 carácter	Sí	No		

tb_access_log

Tabla para guardar información del control de acceso real al evento, en concreto debemos registrar el identificador único, el identificador del evento, el identificador del invitado, la fecha y hora de acceso al evento y el tipo de acceso que **solo puede contener dos valores “IN” o “OUT”** para gestionar la entrada y la salida al evento.

Tabla: tb_access_log Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
access_log_id	Numérico entero	No	Sí		
event_id	Numérico entero	No	No	tb_event	
guest_id	Numérico entero	No	No	tb_guest	
access_log_date	Fecha (con tiempo)	No	No		Fecha y hora actual
access_log_type	Cadena de 3 caracteres	No	No		

Una vez ejecutadas las sentencias SQL de creación de las estructuras que se requieren, se pide ejecutar el *script* **BBDD_ERP_data.sql** adjunto. Este fichero NO puede ser modificado para la

realización de la actividad y es necesario que todas las sentencias SQL del fichero se ejecuten correctamente sin producir errores. **Si se encuentran indicios de que el fichero se ha modificado o que los datos están incompletos, toda la pregunta será considerada como no válida.**

b) Nos piden que, en el *script* de creación de la base de datos, eliminemos las tablas antes de crearlas en caso de que ya existan. Hemos añadido el siguiente código en el *script*, pero nos aparece un error. ¿A qué se debe?

```
-----  
--  
-- Drop tables  
--  
-----  
  
DROP TABLE IF EXISTS erp.tb_event;  
DROP TABLE IF EXISTS erp.tb_access_log;  
DROP TABLE IF EXISTS erp.tb_event_guests;  
DROP TABLE IF EXISTS erp.tb_guest_type;  
DROP TABLE IF EXISTS erp.tb_guest;
```

Debéis modificar el *script* de creación de la base de datos para que elimine las tablas antes de recrearlas, pero no podéis utilizar la opción **DROP CASCADE**.

Pregunta 2 (7,5 puntos)

Se pide proporcionar, para cada apartado, **una única consulta en lenguaje SQL, y el resultado de dicha consulta mediante una captura del resultado en pgAdmin**. Para poder realizar este ejercicio, el primer ejercicio ha de estar finalizado de forma correcta.

Las consultas SQL de este ejercicio han de entregarse en un fichero llamado **Actividad2_ej2.sql** (las capturas de pantalla y explicaciones, si son necesarias, han de entregarse en un documento adjunto, revisad el apartado final de **Formato y fecha de entrega**).

Los resultados de las consultas no deben contener repeticiones en ningún caso, por lo que la eliminación de resultados puede ser necesaria en alguna de las consultas. Por otro lado, el uso innecesario de la cláusula DISTINCT será penalizado.

Nota: La instalación por defecto de PostgreSQL es case sensitive, esto es, la cadena de caracteres "ALUMNO UOC" es diferente de "Alumno UOC" y de "alumno uoc".

Nota: Es posible que para alguna de las consultas exista más de una solución. En este caso, cualquiera de las soluciones se considerará válida, siempre y cuando cumpla con las especificaciones proporcionadas en el enunciado.

- a) Nos piden obtener los eventos que se realizan el día 4 de marzo de 2025, con capacidad igual o mayor de 20 y que no se realicen en el "Auditorium". Se requiere visualizar el identificador del evento, su nombre, la fecha de realización, la capacidad máxima de invitados y dónde se realiza. Los resultados deben estar ordenados por nombre del evento alfabéticamente.
- b) Nos piden obtener la lista de invitados cuyo tipo sea "VIP Guests", "Government & Regulatory Officials" o "Media & Press", excluyendo aquellos cuyo número de teléfono comience con el prefijo "+34". Se requiere visualizar el identificador del invitado, su nombre, correo electrónico, número de teléfono y la descripción del tipo de invitado. Los resultados deben estar ordenados alfabéticamente por el nombre del invitado.
- c) Los organizadores quieren cerrar el tema de la comida de los CEO y nos piden obtener la lista de invitados al evento "CEO gala dinner" que no han confirmado su asistencia o cuya confirmación es nula, para poderlos contactar. Se requiere visualizar el nombre del invitado, el nombre del evento, su número de teléfono y el estado de la confirmación.
- d) Ha empezado la convención, el primer evento se ha realizado con éxito y en la tabla de control de accesos tenemos la información de los invitados que asistieron al "Opening party". Se quiere obtener la lista de invitados que fueron invitados al evento pero **no han asistido**. Se requiere visualizar el nombre y el correo electrónico.

- e) *Marketing* quiere mirar de ampliar la gente apuntada a los eventos. Por ello nos piden obtener un listado de invitados, junto con el número total de eventos en los que están registrados. Solo se deben incluir aquellos invitados que han sido registrados en **menos de 2 eventos. En el caso que alguien no haya sido invitado a ningún evento, también debe aparecer.** El resultado de la consulta debe incluir el identificador del invitado, su nombre, su correo electrónico y el número total de eventos a los que está asociado. Además, el resultado debe estar ordenado por el número total de eventos en los que han participado de mayor a menor.

Pregunta 3 (7.5 puntos)

Proporcionar, para cada apartado, **las sentencias SQL y el resultado de dichas sentencias mediante una captura del resultado en pgAdmin**. Para poder realizar este ejercicio, el ejercicio 1 ha de estar finalizado.

Las sentencias SQL de este ejercicio han de entregarse en un fichero llamado **Actividad2_ej3.sql** (las capturas de pantalla han de entregarse en un documento adjunto).

- a) Nos han informado de que se ha subcontratado a una empresa para gestionar el *catering* de los eventos. Esta empresa cobra un coste fijo de 6.452 € por el servicio, además de un coste variable que depende del número de invitados y del tipo de evento. Nos han indicado los precios por evento e invitado:

Evento	Precio
Opening party	73,67 € invitado
CEO gala dinner	158,00 € invitado
Closing party	73,67 € invitado
General Entrance Day 1	8,76 € invitado
General Entrance Day 2	8,76 € invitado

Nos solicitan añadir un campo numérico con dos decimales en la tabla **tb_event**, denominado **catering_cost_guest** para almacenar esta información. **Este campo nuevo puede tener valor nulo, pero no puede contener valores negativos**.

Tabla: tb_event Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
catering_cost_guest	Numérico decimal (6,2)	Sí	No		

Debéis generar una sentencia para crear el campo requerido y las instrucciones para actualizar su valor utilizando los datos de la tabla de precios proporcionada.

Una vez creado el campo y actualizados los valores, debéis realizar una consulta para calcular el coste total del *catering*. Para ello, tened en cuenta el coste fijo y el número total de invitados por evento, incluyendo sólo aquellas invitaciones que han sido confirmadas ('S') o que aún no han recibido respuesta (NULL).

- b) El CEO de la empresa organizadora ha decidido, a última hora, realizar un nuevo evento con la prensa el mismo día de la 'Opening Party' y en el mismo lugar. Se requiere crear un evento denominado 'Press Briefing' con el identificador 16. Este evento no tendrá un evento padre, y la capacidad máxima será de 20 personas.

Una vez creado el evento en la tabla correspondiente, es necesario invitar a todos los asistentes de la categoría 'Media & Press', insertándolos en la tabla correspondiente. Para ello, puedes consultar la documentación de PostgreSQL sobre cómo realizar una inserción (*INSERT*) a partir de una consulta (*SELECT*).

<https://www.postgresql.org/docs/16/dml-insert.html>

El CEO ha revisado la lista de invitados, y nos ha pedido que no se invite a "Katie Andrews". Por favor, borrarla de la tabla de invitados para este evento.

- c) Se requiere crear una vista denominada **event_occupation** que proporcione, para cada evento, un resumen del número total de asistentes junto con el porcentaje de ocupación respecto a su capacidad máxima.

La vista debe incluir los siguientes campos:

- Nombre del evento
- Capacidad máxima de asistentes
- Número total de asistentes registrados
- Porcentaje de ocupación en relación con la capacidad máxima

Mostrar el resultado de la consulta de la vista.

- d) Puedes responder las siguientes preguntas relacionadas con la concesión y revocación de privilegios sobre una base de datos:

- 1) ¿Qué sentencia SQL usarías para conceder a los usuarios user1 y user2 permisos para realizar consultas (*SELECT*) e inserciones (*INSERT*) en la tabla tb_access_log?
- 2) Previamente se ha concedido al usuario user3 el permiso de *UPDATE* sobre la tabla tb_event. ¿Cómo podrías revocar este permiso?
- 3) ¿Qué hace la opción *WITH GRANT OPTION* en una sentencia *GRANT*? Escribe un ejemplo de uso.
- 4) El usuario admin ha concedido permisos *SELECT* e *INSERT* sobre la tabla tb_event_guest a user5 con *WITH GRANT OPTION*, y este a su vez ha concedido los mismos permisos a user6.

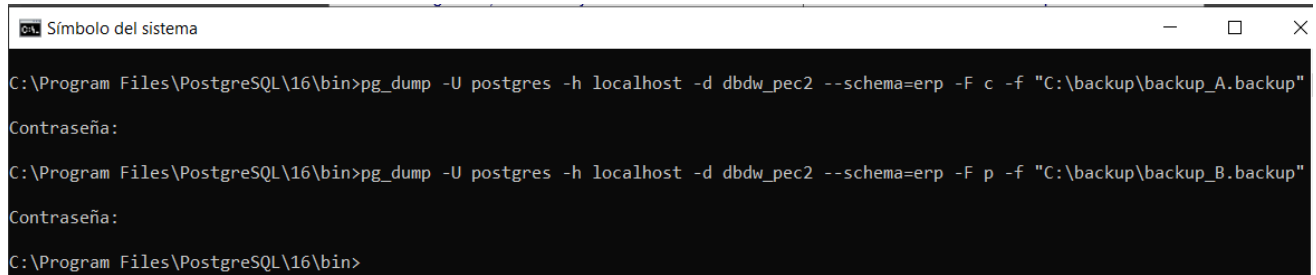
Si ahora admin ejecuta:

```
REVOKE SELECT, INSERT ON tb_event_guest FROM user5 CASCADE;
```

¿Qué ocurrirá con los permisos de user6?

Pregunta 4 (4 puntos)

Desde la empresa nos han solicitado que verifiquemos la copia de seguridad (*backup*) diaria de su base de datos PostgreSQL, la cual se ejecuta de manera automática en el servidor con dos comandos que son los siguientes:



```

C:\Program Files\PostgreSQL\16\bin>pg_dump -U postgres -h localhost -d dbdw_pec2 --schema=erp -F c -f "C:\backup\backup_A.backup"
Contraseña:
C:\Program Files\PostgreSQL\16\bin>pg_dump -U postgres -h localhost -d dbdw_pec2 --schema=erp -F p -f "C:\backup\backup_B.backup"
Contraseña:
C:\Program Files\PostgreSQL\16\bin>
  
```

- ¿Podrías explicar qué hace cada uno de estos comandos y qué opciones utilizan? Además, ¿cuáles son las diferencias entre los dos tipos de copia de seguridad generados?
- Nos solicitan restaurar la tabla **erp.tb_access_log** desde las copias de seguridad generados. ¿Cómo deberíamos proceder para recuperar esta tabla en cada uno de los dos tipos de *backup*?

Criterios de evaluación

En el enunciado se indica el peso/valoración de cada ejercicio.

En **todos los ejercicios** se valorará que se aplique el conjunto de **buenas prácticas en SQL** vistas en este reto; se penalizará su no aplicación. Asimismo, y tal como se ha indicado en los enunciados, se espera que vuestras sentencias no generen errores de ejecución y que la entrega siga las indicaciones dadas.

- En el **ejercicio 1** se valorará:
 - Que el *script* de creación cree la base de datos con los requerimientos solicitados.
 - Que haga correctamente la carga de los datos.
 - No son necesarios comentarios en este ejercicio a menos que queráis aclarar algún tema.
- En el **ejercicio 2** se valorará:
 - Que las propuestas de solución devuelvan los resultados solicitados en el enunciado.
 - Que la solución no haga uso inadecuado de cláusulas como *DISTINCT* o *GROUP BY* y que no realice operaciones de *JOIN* innecesarias.
 - Que se documente el resultado final de las *SELECT*, con una captura de pantalla en el documento a entregar.
 - No son necesarios comentarios de la solución a menos que queráis aclarar algún tema.
- En el **ejercicio 3** se valorará:
 - Que la solución se ajuste al enunciado.
 - Documentar las pruebas realizadas para verificar la solución.
 - Explicación clara en los apartados que se requieran.
 - Citar todas las fuentes de información utilizadas.
- En el **ejercicio 4** se valorará:
 - La claridad en las explicaciones de la solución propuesta, basándose en la documentación consultada.
 - La utilidad de los ejemplos propuestos para apoyar e ilustrar las explicaciones proporcionadas.
 - Documentar las pruebas realizadas para verificar la solución.
 - Citar todas las fuentes de información utilizadas.

Formato

Se debe entregar como resultado de esta actividad de evaluación:

- Un archivo con vuestras propuestas de solución, capturas de pantalla y las explicaciones a las preguntas planteadas. Este archivo **obligatoriamente debe ser un PDF**. Es **necesario utilizar la plantilla proporcionada** para elaborar vuestra propuesta de solución.
- Los archivos **.SQL solicitados** en los ejercicios, subidos uno a uno como ficheros separados.

Para realizar la entrega de la actividad, tenéis que seguir el enlace **Entrega de la actividad P2**, disponible en el **Reto 2**:

▼ Reto 2. De la creación a la manipulación de una base de datos relacional			
📄	Planteamiento y competencias R2	📄 ✓	⋮
📄	Recursos de aprendizaje R2	📄 ✓	⋮
💬	Foro de dudas Reto 2 y PEC2	📄 ✓	⋮
📄	PEC2	✓	⋮
📄	Planteamiento y competencias P2	📄 ✓	⋮
📄	Enunciado de la actividad P2	📄 ✓	⋮
📄	Entrega de la actividad P2 14 de abr 25 puntos	📄 ✓	⋮
📄	Solución P2	📄 ✓	⋮

Fechas clave

- La fecha límite para entregar vuestra propuesta de solución para la actividad es el **14/04/2025**.
- La propuesta de solución del profesorado se hará pública el **22/04/2025**.
- La publicación de notas se hará el **28/04/2025**.