

UD2 – Examen (Modelo A)

LEE LAS INSTRUCCIONES ATENTAMENTE. DEBERÁS ENTREGAR ESTE MISMO DOCUMENTO EN FORMATO PDF (CUALQUIER OTRO FORMATO DISTINTO DE PDF NO SERÁ CORREGIDO). SE ESPERA QUE PUEDAS RESOLVER ESTAS ACTIVIDADES EMPLEANDO EXCLUSIVAMENTE TUS CONOCIMIENTOS Y LOS MATERIALES QUE HAYAS DESCARGADO PREVIAMENTE. EN CASO DE CUALQUIER EVIDENCIA DE COPIA, EL EXAMEN SE EVALUARÁ COMO SUSPENSO (0).

PARA ESTE EXAMEN, REUTILIZAREMOS EL SCRIPT QUE USAMOS PARA LA ACTIVIDAD 3.5 QUE PUEDES ENCONTRAR EN MOODLE. POR LO TANTO, NO OLVIDES CORREGIR EL ERROR (SUGERENCIA: ORDEN INCORRECTO PARA LOS INSERTS).

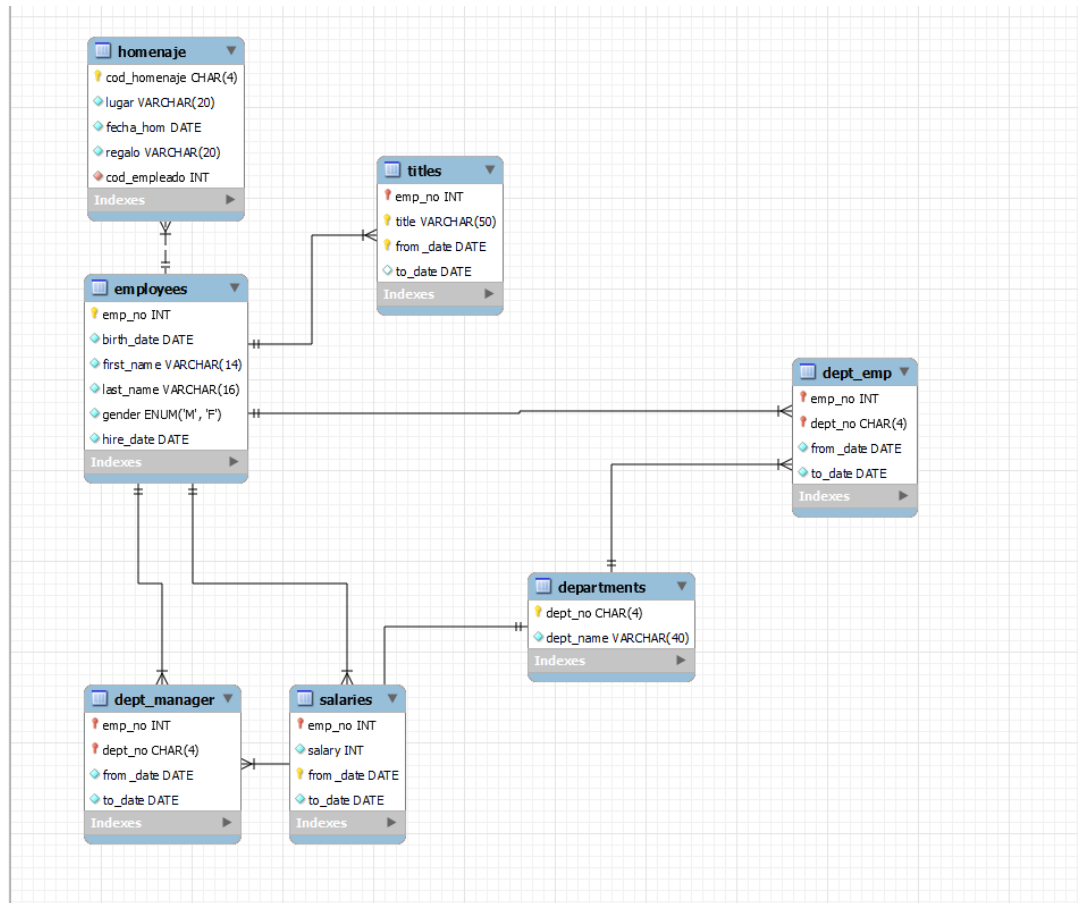
Pregunta 1. Teniendo en cuenta la base de datos dada, necesitamos agregar una nueva tabla llamada "homenaje". Dicha tabla debe estar conectada a los empleados, ya que un empleado puede recibir varios homenajes, sin embargo, se organiza un homenaje para un solo empleado. Cada registro de cada homenaje debe contener un lugar, una fecha y un regalo para el empleado homenajeado (puede ser una pluma, un reloj o un ordenador portátil). (2 puntos)

Se pide:

1. Código SQL para la creación de esta nueva tabla y diagrama de ingeniería inversa que muestra la base de datos actualizada. (0,5 puntos)

```
592 • CREATE TABLE homenaje(  
593     cod_homenaje CHAR(4) UNIQUE NOT NULL,  
594     lugar VARCHAR(20) NOT NULL,  
595     fecha_hom DATE NOT NULL,  
596     regalo VARCHAR(20) NOT NULL,  
597     cod_empleado INT NOT NULL,  
598     CONSTRAINT cod_homenaje PRIMARY KEY (cod_homenaje),  
599     CONSTRAINT cod_empleado FOREIGN KEY (cod_empleado) REFERENCES employees(emp_no)  
600  
601 );
```

1 Código de creación de la tabla.



2 Ingeniería Inversa

2. Crea un registro en esta tabla para cada uno de los 5 primeros empleados en función de su emp_no. Todos los homenajes ocurrieron el 10/02/2024 y hubo al menos 3 ubicaciones diferentes. Asigna los regalos al azar. (1 punto)

02

```
03 • INSERT INTO homenaje VALUES
04 ('H001','Hotel Real','2024-02-10','Reloj de Oro','10001'),
05 ('H002','Palacio de Exposiciones','2024-02-10','Pluma Estilográfica','10002'),
06 ('H003','Hotel Real','2024-02-10','Reloj de Oro','10003'),
07 ('H004','Restaurante El Riojano','2024-02-10','Pin de Oro y Brillantes','10004'),
08 ('H005','Sala Argenta','2024-02-10','Reloj de oro','10005');
```

3 Inserts

#	Time	Action	Message
99	10:14:09	select sum(s.salary) as 'Suma Salarios', d.dept_name as 'Departamento' from salaries s join employees e on s.emp_no = e.emp_no join dept_emp de on ...	1 row(s) returned
100	10:14:09	CREATE TABLE homenaje(cod_homenaje CHAR(4) UNIQUE NOT NULL, lugar VARCHAR(30) NOT NULL, fecha_hom DATE NOT NULL, re...	0 row(s) affected
101	10:14:09	INSERT INTO homenaje VALUES ('H001','Hotel Real','2024-02-10','Reloj de Oro','10001'), ('H002','Palacio de Exposiciones','2024-02-10','Pluma Estilog...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

4 Validación de los Inserts

3. Elimina el tributo para el primer empleado. Cambia todas las ubicaciones a Santander.
(0,5 puntos)

```
DELETE FROM homenaje WHERE cod_empleado = '10001';
```

Compruebo el resultado después de borrar el primer registro

	cod_homenaje	lugar	fecha_hom	regalo	cod_empleado
▶	H002	Palacio de Exposiciones	2024-02-10	Pluma Estilográfica	10002
	H003	Hotel Real	2024-02-10	Reloj de Oro	10003
	H004	Restaurante El Riojano	2024-02-10	Pin de Oro y Brillantes	10004
	H005	Sala Argenta	2024-02-10	Reloj de oro	10005

Ahora tengo que modificar los lugares como indica el enunciado.

```
UPDATE homenaje set lugar = 'Santander';
```

Lo pongo así que tardo menos que en hacer una captura. Con este comando voy a modificar todos los campos lugar de la tabla y voy a introducir como valor Santander para todos ellos.

Compruebo el resultado con SELECT * FROM homenaje;

Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	cod_homenaje	lugar	fecha_hom	regalo	cod_empleado
▶	H002	Santander	2024-02-10	Pluma Estilográfica	10002
	H003	Santander	2024-02-10	Reloj de Oro	10003
	H004	Santander	2024-02-10	Pin de Oro y Brillantes	10004
	H005	Santander	2024-02-10	Reloj de oro	10005

Pregunta 2. Desarrolla las siguientes consultas. Para cada una, proporciona tu implementación SQL y una captura de pantalla que muestre los resultados. (5,5 puntos)

Se pide:

1. Muestra nombres, apellidos, fechas de contratación y regalos de todos los empleados que hayan recibido un homenaje. (0,5 puntos)

```
SELECT e.first_name, e.last_name, e.hire_date, h.regalo  
FROM employees e  
JOIN homenaje h on e.emp_no = h.cod_empleado;
```

Este es el código que he implementado para realizar esta actividad.

	first_name	last_name	hire_date	regalo
►	Bezalel	Simmel	1985-11-21	Pluma Estilográfica
	Parto	Bamford	1986-08-28	Reloj de Oro
	Chirstian	Koblick	1986-12-01	Pin de Oro y Brillantes
	Kyoichi	Maliniak	1989-09-12	Reloj de oro

Este es el resultado de esa consulta.

2. Identifica al gerente de cada departamento junto con su salario actual. (1 punto)

```
SELECT e.first_name, e.last_name, d.dept_name, s.salary
FROM employees e
JOIN dept_manager dem on e.emp_no = dem.emp_no
JOIN salaries s on e.emp_no = s.emp_no
JOIN dept_emp on dept_emp.emp_no = e.emp_no
JOIN departments d on d.dept_no = dept_emp.dept_no
WHERE s.to_date = '9999-01-01' and dem.to_date = '9999-01-01';
```

Este es el código SQL. Tengo que realizar varios JOIN entre las tablas que necesito y luego tengo que validar que las fechas to_date de salaries y department manager sean igual a las del fin de los días.

	first_name	last_name	dept_name	salary
▶	Parto	Bamford	Production	43311
	Yinghua	Dredge	Sales	97830
	Yinghua	Dredge	Finance	97830

Aquí se ve el resultado.

3. Muestra el salario y la evolución del departamento, así como el nombre y apellidos del primer empleado contratado. (1 punto)

```
SELECT e.first_name, e.last_name, s.salary, d.dept_name
FROM employees e
JOIN dept_emp de on e.emp_no = de.emp_no
JOIN departments d ON d.dept_no = de.dept_no
JOIN salaries s ON s.emp_no = e.emp_no
WHERE e.hire_date =
(SELECT MIN(e.hire_date)
FROM employees e);
```

Código SQL

	first_name	last_name	salary	dept_name
►	Sumant	Peac	60929	Quality Management
	Sumant	Peac	64604	Quality Management
	Sumant	Peac	64780	Quality Management
	Sumant	Peac	66302	Quality Management
	Sumant	Peac	69042	Quality Management
	Sumant	Peac	70889	Quality Management
	Sumant	Peac	71434	Quality Management
	Sumant	Peac	74612	Quality Management

Resultado.

4. Enumera los nombres de todos los empleados y los nombres de sus gerentes directos (si es que tienen uno). Sólo considera a gerentes actuales. (1 punto)

5. Enumera todos los nombres de los empleados que nunca han cambiado de departamento. (1 punto)

La lógica me dice que para que un empleado no haya cambiado de departamento su fecha de contratación tiene que ser igual a la from_date de la tabla dept_emp

```
SELECT e.first_name, e.last_name
from employees e
JOIN dept_emp de on e.emp_no = de.emp_no
where de.from_date = e.hire_date
order by e.first_name;
```

	first_name	last_name
▶	Alain	Chappelet
	Alejandro	McAlpine
	Alejandro	Brender
	Amabile	Gomatam
	Anoosh	Peyn
	Arif	Merlo

63 Result 364 Result 365

6. Calcula el salario promedio de todos los empleados cuyo título actual contiene la palabra "Senior". (1 punto)

Pregunta 3. Implementa los siguientes cambios en la base de datos: (2,5 puntos)

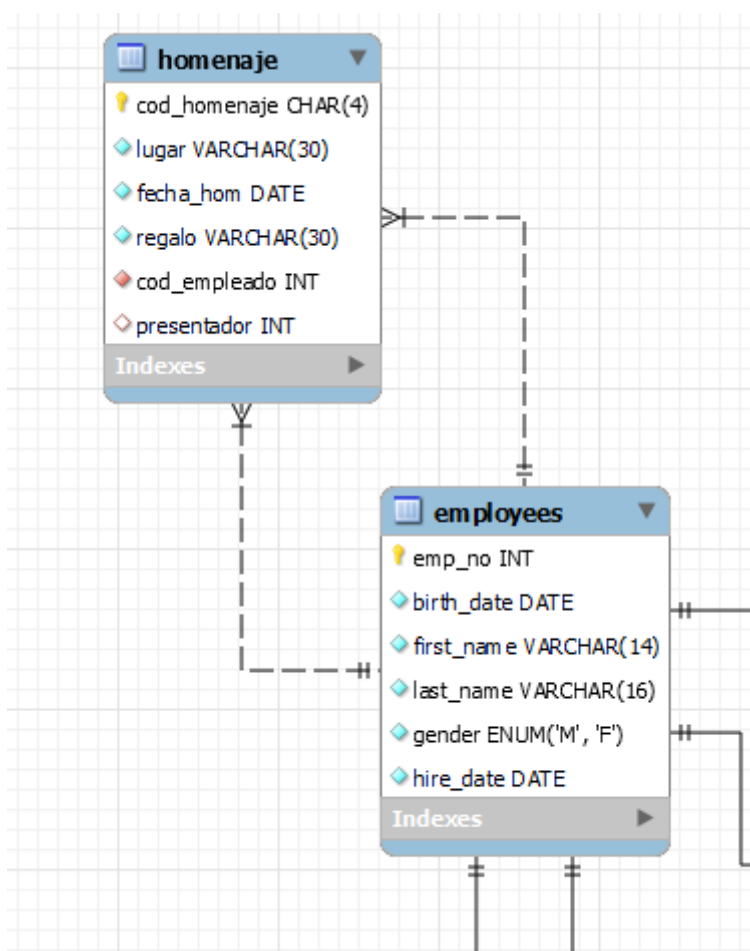
1. Añade una nueva columna a la tabla homenaje la cual se va a llamar "presentador". El presentador es otro empleado que entrega el regalo al empleado homenajeado. Por lo tanto, esta nueva columna debe tener el mismo tipo de datos que emp_no. (0,5 puntos)

```
ALTER TABLE homenaje ADD presentador INT;  
ALTER TABLE homenaje ADD CONSTRAINT presentador FOREIGN KEY (presentador) REFERENCES employees(emp_no);
```

Con estos dos comando inserto el nuevo atributo y lo indico como FK de la nueva tabla que creamos antes.

2. Añade la clave foránea necesaria al presentador, de modo que crees un nuevo vínculo entre la tabla de tributo y el empleado. Muestra el diagrama de ingeniería inversa actualizado. (0,5 puntos)

Vale, pensé que añadir el constraint tenía que ser en el ejercicio anterior, pero bueno ya lo tengo hecho. Solo voy a mostrar la ingeniería inversa.



Como podemos ver, ahora homenaje y employees están doblemente relacionadas, mediante dos foreign key.

3. Establece todos los presentadores al empleado de mayor edad que esté trabajando actualmente en la empresa. (1 punto)

```
UPDATE homenaje set presentador = (select e.emp_no  
FROM employees e  
order by e.birth_date asc  
limit 1);
```

Con este comando lo que hago es modificar todos los valores de presentador para que sean igual al resultado de la consulta, en este caso el trabajador más viejo de la empresa.

	cod_homenaje	lugar	fecha_hom	regalo	cod_empleado	presentador
▶	H002	Santander	2024-02-10	Pluma Estilográfica	10002	10097
	H003	Santander	2024-02-10	Reloj de Oro	10003	10097
	H004	Santander	2024-02-10	Pin de Oro y Brillantes	10004	10097
	H005	Santander	2024-02-10	Reloj de oro	10005	10097
*	NULL	NULL	NULL	NULL	NULL	NULL

4. Muestra el nombre y apellidos de todos los empleados, así como el nombre del regalo y presentador de todos los empleados homenajeados. (0,5 puntos)

NO OLVIDES AGREGAR CAPTURAS DE PANTALLA DE TU CÓDIGO, ASÍ COMO DE TODOS TUS RESULTADOS.