

EXAMEN ORDINARIO (EJEMPLO)

– Administración de Base de Datos

SECCIÓN 1 – DISEÑO Y NORMALIZACIÓN (20 puntos)

1. (10 pts) Genera un diagrama ER normalizado ($\geq 3FN$) para una aplicación que gestiona proyectos, tareas y usuarios. Entregar: diagrama, tablas y script CREATE TABLE.
2. (10 pts) Explica por qué tu modelo cumple con 1FN, 2FN y 3FN con ejemplos.

SECCIÓN 2 – ADMINISTRACIÓN DE MySQL EN DOCKER (30 puntos)

3. (10 pts) Levanta MySQL en Docker usando Dockerfile o docker-compose. Incluye usuarios, permisos y volúmenes.
4. (10 pts) Crea índices (simple, compuesto, único). Usa EXPLAIN antes/después y explica mejoras.
5. (10 pts) Crea un script de transacción con inserciones, error provocado y rollback. Entrega transaccion.sql y evidencia desde el contenedor.

SECCIÓN 3 – AWS RDS CON MySQL (20 puntos)

6. (10 pts) Conecta tu MySQL en Docker a AWS RDS. Entrega capturas y comandos.
7. (15 pts) Realiza backup y rollback en RDS usando mysqldump desde Docker.

SECCIÓN 4 – MONITOREO Y ADMINISTRACIÓN (15 puntos)

8. (10 pts) Explica cómo monitorear MySQL en Docker: espacio, memoria, logs, estado general.
9. (15 pts) Realiza una prueba de estrés con mysqlslap o similar. Entrega comando, resultados y análisis.

Problemática real para desarrollar el modelo de Proyectos–Tareas–Usuarios

Contexto

La empresa **LogiStore**, un almacén de distribución de productos, constantemente realiza mejoras internas: reorganización de inventario, implementación de software, mantenimiento de racks, auditorías, etc.

Para controlar estas actividades, la empresa quiere una **aplicación que gestione proyectos internos**, las **tareas asociadas** y los **usuarios involucrados**.

Problemática

Actualmente, LogiStore administra sus proyectos en hojas de cálculo.

Esto genera problemas:

- No existe un historial claro de quién hizo qué.
- Los responsables de tareas cambian, pero no queda registro.
- Un proyecto puede tener muchas tareas, pero no existe control del avance.
- Las fechas se pierden o se duplican.
- Los empleados no tienen claridad sobre tareas pendientes.

Por eso se requiere una **base de datos normalizada** ($\geq 3FN$) que permita:

1. Registrar proyectos (mantenimiento de racks, reorganización de pasillos, auditorías, etc.).
2. Registrar tareas por proyecto (inspección, limpieza, etiquetado, codificación, instalación, etc.).
3. Asignar usuarios/empleados responsables de ejecutar dichas tareas.
4. Controlar estados, fechas y descripciones.
5. Evitar duplicidad de datos y garantizar consistencia.

Examen — martes 2 de diciembre

1. CAMPOS DAGUER EMILIO
2. CERVANTES ALONZO JUAN FELIPE
3. BARREDO MENDOZA GUSTAVO ALBERTO
4. MARTINEZ RAMON EMILIANO
5. GARCIA KURI ALEJANDRO
6. GARABITA CORDOVA KEVIN EMANUEL
7. MATUS SALVADOR ELIAM
8. CARRERA VARGUEZ VICENTE
9. YUPIT BENITEZ ROBERTO
10. TRACONIZ SALAZAR JULIO FRANCISCO

Examen — viernes 5 de diciembre

1. TAMAYO BRICEÑO DULMY ANAHI
2. ORTEGA PERERA SANTIAGO MARTIN
3. SALDAÑA PACHECO ANDRES
4. AVALOS MARTINEZ RUBEN ALEJANDRO
5. KUUK TUN CARLOS GABRIEL
6. VARGUEZ DURAN JOSE EMILIO
7. CASTELLANOS YAÑEZ SANTIAGO
8. SALAZAR BERZUNZA DANIEL
9. ESPINOSA RODRIGUEZ ALEJANDRO
10. GOMEZ MONTAÑEZ VLADIMIR ABRAHAM

Examen — martes 9 de diciembre

1. ROSADO SANTIAGO ANGEL EFREN
2. BECERRA BENAVIDES VICTOR
3. CANTO MARTINEZ CARLOS ENRIQUE
4. GARCIA LOYOLA CHRISTIAN ENRIQUE
5. VICENTE ROSALDO KEVIN ALEXANDER
6. GONZALEZ BRITO ABRAHAM ELOY
7. GUILLERMO PIÑA ANDERSON
8. AKE CANUL JUAN RAMON
9. COBOS BRACAMONTE JOSHUA ARTURO