

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3088 - Base de Datos 1

Sección 20

Ing. Alexander Bolaños



Proyecto #3 - Reflexión Individual

Gadiel Ocaña, 231270

GUATEMALA, 05 de mayo de 2025

Reflexión Individual

1. ¿Qué criterios usaron para decidir qué entidades y relaciones debían formar parte del modelo? Justifiquen cómo identificaron qué debía representarse en la base de datos y qué decisiones tomaron para simplificar o abstraer ciertos aspectos del sistema real.

Para decidir qué entidades y relaciones debían formar parte del modelo de base de datos, partimos de analizar las funciones principales del sistema que queríamos representar, en este caso, la gestión de empleados, sus turnos y el control de su salario. Identificamos como entidades esenciales a los empleados, los turnos, los puestos y el historial salarial, ya que cada una cumple un rol importante dentro del funcionamiento del sistema. Por ejemplo, los empleados son el eje central, los puestos definen el rango salarial permitido y los turnos permiten registrar cuántas horas trabaja cada persona a la semana. Además, el historial salarial nos ayuda a mantener un registro de los cambios en el salario de cada empleado. Decidimos incluir estas entidades porque representan información que cambia con el tiempo y que es necesario controlar con precisión. Para simplificar el modelo, optamos por no representar detalles como horarios exactos de entrada y salida en los turnos, sino solo las horas asignadas, lo cual nos permite calcular fácilmente las horas semanales sin complicar demasiado la estructura. También abstrajimos los motivos de cambios salariales usando un valor fijo por defecto, con la posibilidad de extenderlo más adelante si se requiere.

2. ¿Qué tan adecuadas fueron las claves primarias y foráneas que definieron en su diseño? Evalúen si estas claves facilitaron las consultas, mantuvieron la integridad de los datos y permitieron modelar correctamente las dependencias entre tablas.

Las claves primarias y foráneas que definimos en el diseño resultaron adecuadas para representar correctamente las relaciones entre las distintas entidades del sistema. Las claves primarias se asignaron a cada tabla de forma que identificaran de manera única a cada registro, como por ejemplo `id_empleado` en la tabla de empleados o `id_turno` en la tabla de turnos. Por otro lado, las claves foráneas permitieron establecer conexiones claras entre tablas, como la relación entre empleado y puesto, o entre empleado_turno y empleado, lo cual fue fundamental para mantener la integridad referencial. Estas claves facilitaron las consultas porque nos permitieron hacer uniones (JOINS) eficientes y precisas entre tablas, sin ambigüedades. Además, al tener bien definidas las claves foráneas, el sistema puede evitar que se ingresen datos inválidos, como asignar un turno a un empleado que no existe o registrar un salario para un puesto

inexistente. También nos ayudaron a modelar las dependencias de forma correcta, por ejemplo, asegurando que no se pueda borrar un empleado si todavía tiene registros asociados en empleado_turno o historial_salario, a menos que se tomen medidas explícitas. En general, consideramos que las claves utilizadas sí cumplieron su función de forma eficiente y coherente con el modelo que necesitábamos representar.

3. ¿En qué medida aplicaron la normalización? ¿Qué beneficios y limitaciones experimentaron? Aplicar 1FN, 2FN y 3FN fue suficiente o si surgieron situaciones donde tuvieran que decidir entre rendimiento y diseño teórico.

Aplicamos la normalización hasta la Tercera Forma Normal (3FN), asegurándonos de eliminar datos repetidos y dependencias parciales o transitivas. Esto ayudó a mantener la integridad y claridad del modelo, facilitando el mantenimiento y evitando inconsistencias. En general, fue suficiente para nuestro caso, aunque en algunas consultas que requerían varias uniones (JOINS) notamos que el rendimiento podía verse afectado. Sin embargo, decidimos priorizar un buen diseño teórico, ya que el volumen de datos no era tan grande como para justificar una desnormalización en esta etapa.

4. ¿Qué restricciones y reglas del negocio implementaron directamente en la base de datos y por qué? Describe el uso de CHECK, DEFAULT, NOT NULL, UNIQUE, claves foráneas y triggers, y justifica su implementación.

Implementamos restricciones como 'NOT NULL' para evitar datos incompletos, 'CHECK' para validar rangos como el salario, y 'DEFAULT' para asignar valores automáticos. Usamos claves foráneas para mantener relaciones válidas entre tablas y 'UNIQUE' cuando era necesario evitar duplicados. También añadimos triggers para aplicar reglas más complejas, como actualizar horas semanales al asignar turnos o registrar cambios salariales. Estas medidas aseguran la integridad de los datos y reducen errores directamente desde la base de datos.

5. ¿Qué ventajas o desventajas identificas del modelo que construyeron al momento de hacer consultas complejas? Piensen en consultas con múltiples filtros, joins, subconsultas o agrupaciones, y comenten si el modelo fue flexible y escalable.

Al momento de hacer consultas complejas, el modelo resultó bastante flexible y claro gracias a su buen diseño y normalización. Las relaciones entre tablas estaban bien definidas, lo que facilitó el uso de 'JOINS', subconsultas y filtros sin ambigüedad. Esto permitió hacer consultas detalladas, como obtener el total

de horas trabajadas por empleado o el historial de cambios salariales. Sin embargo, en algunos casos, el uso de varias uniones puede afectar ligeramente el rendimiento, especialmente si las tablas crecen mucho. Aun así, consideramos que el modelo es escalable y mantiene un buen equilibrio entre claridad y eficiencia.

6. ¿Qué cambiarían en el diseño de la base de datos si tuvieran que escalar este sistema a un entorno de producción? Reflexionen sobre aspectos como volumen de datos, rendimiento, integridad y escalabilidad.

Si el sistema se escalara a producción, optimizaríamos índices para mejorar el rendimiento, consideraríamos desnormalizar algunas tablas muy consultadas y agregaríamos auditorías más robustas. También reforzaríamos la integridad con más validaciones y pensaríamos en particionar tablas si el volumen de datos creciera mucho.