|  |  |
| --- | --- |
|  | **Universidad Nacional Autónoma de México** |
|  | **Facultad de Ingeniería** |
|  | **División de Ingeniería Eléctrica** |
|  | **Laboratorio de**  **Bases de Datos** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Practica 8 | Normalización | |
|  |  |  |  | |
|  | Profesor: | Lugowski Rivero Czeslaw Kristofer |  | |
|  | Grupo: | 06 |  | |
|  | Alumnos: | Guerrero López Uriel Ivan 320046045  Muñoz San Agustin Victoria Monserrat 320094187 | | |
|  |  |  | Calificación | |
|  |  | Fecha de entrega | Jueves 10 de abril de 2025 | |
|  |  |  | Semestre 2025-2 | |

**Objetivo:**

* El alumno comprenderá la importancia de la teoría de la normalización en el diseño de bases de datos y aplicará los conocimientos adquiridos. Conocerá e implementará las formas normales en un caso práctico apoyado de muestras de datos para el adecuado modelado de la base de datos.

**Introducción**

Normalizar es un proceso formado por distintas etapas que tiene como propósito eliminar redundancias y anomalías de diseño: anomalías de inserción,modificación y eliminación.

* Anomalía de inserción: Algunos atributos no pueden ser agregados a la relación sin la presencia de otros atributos.
* Anomalía de modificación: Al modificar datos duplicados, no se modifican todas las instancias
* Anomalía de eliminación: Algunos atributos se pierden por la eliminación de otros atributos

Normalización es un proceso secuencial, incremental, cíclico y repetitivo que consiste en aplicar las formas normales a todas y cada una de las entidades del modelo, con la finalidad de garantizar la Integridad de los datos, evitando la redundancia y corrigiendo las anomalías de diseño.

Dependencia funcional.

Las dependencias funcionales son muy útiles para analizar una tabla en búsqueda de redundancias innecesarias. Una dependencia funcional es una restricción entre dos conjuntos de atributos de la base de datos

Dada una relación R el atributo y(R) depende funcionalmente del atributo x(R) sí y solo sí un solo valor de y(R) está asociado a cada valor x(R) en cualquier momento dado

x(R) → y(R)

Por ejemplo, si conocemos el valor de la fecha de nacimiento podemos conocer el valor

de la edad.

fechaNacimiento → edad

De lo anterior, se puede concluir que “x” puede tomar el rol de llave primaria, siempre y

cuando x, también pueda determinar el valor de los demás atributos.

El proceso consiste en encontrar el o los campos que determinen de manera única a cada

uno de los registros de la tabla empleando el concepto de dependencia funcional.

Dependencia parcial

Alguno de los atributos que forma parte de la PK puede por sí solo actuar como atributo determinante de uno o más campos de la tabla Las dependencias parciales pueden existir únicamente en tablas con una llave primaria compuesta.

Dependencia transitiva.

Sucede cuando un campo que no forma parte de la PK puede actuar como atributo determinante de uno o más campos.

Primera Forma Normal. Una relación está en Primera Forma Normal (1NF) si y sólo si:

* No debe haber tuplas repetidas (grupos de repetición)
* No debe importar el orden de las tuplas
* Existencia de una Llave Primaria
* Todos los atributos atómicos

Segunda Forma Normal. Una Relación está en Segunda Forma Normal (2NF) si, y sólo si:

1. Está en Primera Forma Normal
2. Todos los atributos dependen de la llave primaria por completo. Ningún atributo depende únicamente de una parte de la llave primaria, es decir, se han eliminado las dependencias parciales.

Tercera Forma Normal. Una Relación está en Tercera Forma Normal (3NF) si, y sólo si:

1. Está en Segunda Forma Normal
2. Ningún atributo “no determinante” depende transitivamente de la llave primaria.

Ningún atributo “no primo” depende de otro atributo no clave, es decir, se han eliminado

las dependencias transitivas.

Forma Normal Boyce Codd. Esta forma normal es una variante de la 3FN, y fue desarrollada en 1974 por Raymond F. Boyce y Edgar F. Codd en la cual se resuelven algunas anomalías que no resuelve la

3FN.

La forma Normal Boyce-Codd puede aplicarse en tablas con PK compuesta, generalmente en tablas intermedias con atributos que por su naturaleza pudieran también formar parte de la PK.

La existencia de estos atributos puede causar anomalías en los datos, por ejemplo, redundancia.

Para poder eliminar estas anomalías se deberá verificar si es posible intercambiar atributos de tal forma que sea posible formar una dependencia parcial y así poder eliminarla.

Cuarta Forma Normal. Una tabla estará en su cuarta forma normal (4FN) cuando:

* Está en su tercera forma normal 3FN.
* Se han eliminado posibles dependencias multivalor.

Una dependencia multivalor ocurre cuando la PK de una tabla determina múltiples valores de 2 o más campos de la tabla y cada uno de estos campos son totalmente independientes entre sí, es decir, no hay relación alguna entre ellos.

Desnormalización.

Se refiere al proceso inverso de Normalización. En algunos escenarios un nivel alto de normalización pudiera generar posibles impactos en el desempeño. A mayor nivel de normalización, el número de tablas se incrementa y por lo tanto, la cantidad de operaciones join requeridas para reconstruir o realizar consultas en este nuevo conjunto de tablas aumenta de manera considerable.

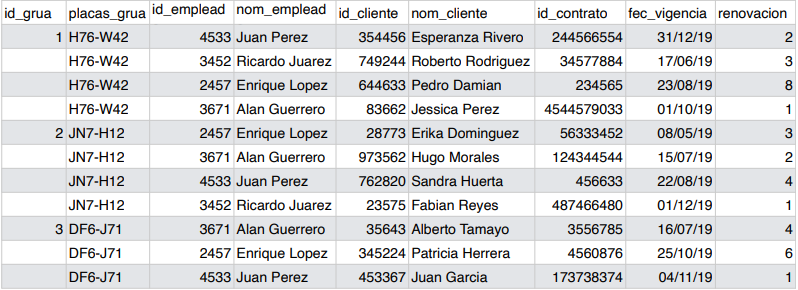
Esta condición pudiera generar dificultad para realizar consultas o analítica de datos adicional a posibles impactos en el desempeño, en especial para tablas de gran tamaño.

Por lo anterior, las tablas de una base de datos pudieran modificarse para bajar el nivel de normalización reduciendo así el número de tablas, asumiendo la posible existencia de anomalías, pero favoreciendo al desempeño.

La aparición de anomalías podría reducirse o evitarse a través de la aplicación que accede a los datos, es decir, el software o aplicación será la responsable de evitar la incorporación de anomalías, en especial la aparición de inconsistencias.

**Actividades propuestas por la academia**

1. A partir de una tabla de datos proporcionada realizar lo siguiente:



1. Aplicar la 1ra Forma Normal

Primero separamos la table para que solo quede en valores atómicos

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id\_grua | Placas\_grua | Id\_empleado | Nombre\_empleado | Apellido\_empleado | Id\_cliente | Nombre\_cliente | Apellido\_cliente | Id\_contrato | Fec\_vigencia | renovación |
| 1 | H76-W42 | 4533 | Juan | Pérez | 354456 | Esperanza | Rivero | 244566554 | 31/12/19 | 2 |
|  | H76-W42 | 3452 | Ricardo | Juárez | 749244 | Roberto | Rodriguez | 34577884 | 17/06/19 | 3 |
|  | H76-W42 | 2457 | Enrique | López | 644633 | Pedro | Damian | 234565 | 23/08/19 | 8 |
|  | H76-W42 | 3671 | Alan | Guerrero | 83662 | Jessica | Perez | 4544579033 | 01/10/19 | 1 |
| 2 | JN7-H12 | 2457 | Enrique | López | 28773 | Erika | Dominguez | 56333452 | 08/05/19 | 3 |
|  | JN7-H12 | 3671 | Alan | Guerrero | 973562 | Hugo | Morales | 124344544 | 15/07/19 | 2 |
|  | JN7-H12 | 4533 | Juan | Pérez | 762820 | Sandra | Huerta | 456633 | 22/08/19 | 4 |
|  | JN7-H12 | 3452 | Ricardo | Juárez | 23575 | Fabian | Reyes | 487466480 | 01/12/19 | 1 |
| 3 | DF6-J71 | 3671 | Alan | Guerrero | 35643 | Alberto | Tamayo | 3556785 | 16/07/19 | 4 |
|  | DF6-J71 | 2457 | Enrique | López | 345224 | Patricia | Herrera | 3556785 | 25/10/19 | 6 |
|  | DF6-J71 | 4533 | Juan | Pérez | 453367 | Juan | Garcia | 4560876 | 04/11/19 | 1 |

Inicialmente, tenemos que verificar que nuestra tabla está en 1era forma normal, por lo que checamos que todos los atributos deben ser atómicos, es decir sin multivaluados ni repetidos, en este caso, nuestra tabla ya se encuentra en esta condición, por lo que no será necesario modificar ningún atributo de nuestra tabla. Posteriormente, vamos a identificar nuestra llave primaria que nos hará mapear cualquier atributo de nuestra tabla, en este caso, sería la siguiente:

*Llave primaria = (id-grua, id\_empleado, id\_cliente, id\_contrato)*

De esta forma, podemos decir que nuestra tabla **ya está en 1FN**

1. Aplicar la 2da Forma Normal

Posteriormente, al tener nuestra tabla en 1FN, buscaremos que todos los atributos no primarios dependan totalmente de la clave primaria definida con anterioridad, es decir no hay dependencias parciales.

Análisis de dependencias:

|  |
| --- |
| Primary Key |

**Grúas**

|  |  |
| --- | --- |
| Id\_grua | Placas\_grua |
| 1 | H76-W42 |
| 1 | H76-W42 |
| 1 | H76-W42 |
| 1 | H76-W42 |
| 2 | JN7-H12 |
| 2 | JN7-H12 |
| 2 | JN7-H12 |
| 2 | JN7-H12 |
| 3 | DF6-J71 |
| 3 | DF6-J71 |
| 3 | DF6-J71 |

**Empleado**

|  |  |  |
| --- | --- | --- |
| Id\_empleado | Nombre\_empleado | Apellido\_empleado |
| 4533 | Juan | Pérez |
| 3452 | Ricardo | Juárez |
| 2457 | Enrique | López |
| 3671 | Alan | Guerrero |
| 2457 | Enrique | López |
| 3671 | Alan | Guerrero |
| 4533 | Juan | Pérez |
| 3452 | Ricardo | Juárez |
| 3671 | Alan | Guerrero |
| 2457 | Enrique | López |
| 4533 | Juan | Pérez |

**Cliente**

|  |  |  |
| --- | --- | --- |
| Id\_cliente | Nombre\_cliente | Apellido\_cliente |
| 354456 | Esperanza | Rivero |
| 749244 | Roberto | Rodriguez |
| 644633 | Pedro | Damian |
| 83662 | Jessica | Perez |
| 28773 | Erika | Dominguez |
| 973562 | Hugo | Morales |
| 762820 | Sandra | Huerta |
| 23575 | Fabian | Reyes |
| 35643 | Alberto | Tamayo |
| 345224 | Patricia | Herrera |
| 453367 | Juan | Garcia |

Ahora si ya **está en 2FN**, donde la llave primaria seleccionada en la anterior forma será nuestra tabla mayor (Servicios) pues resguarda las PKs que a su vez son FKs de las otras tablas

* 1. Aplicar la 3ra Forma Normal

Al analizar la tabla, podemos verificar que:

* Cada tabla tiene una clave primaria simple
* Cada atributo depende únicamente de su clave primaria
* No hay dependencias transitivas

Por lo tanto podemos aseverar que **ya está en 3FN**

1. Aplicar las formas normales FNBC y 4ta a un modelo relacional presentado, obteniendo un nuevo modelo relacional. En la herramienta Data Modeling Tool seleccionada generar el código de salida SQL(DDL).

**FNBC (Forma Normal de Boyce-Codd)**

Reglas:

* Estar en 3FN.
* Para toda dependencia funcional X → Y, X debe ser superclave.

No hay violaciones: todas las dependencias funcionales tienen como determinante una superclave.

**Resultado: Ya está en FNBC.**

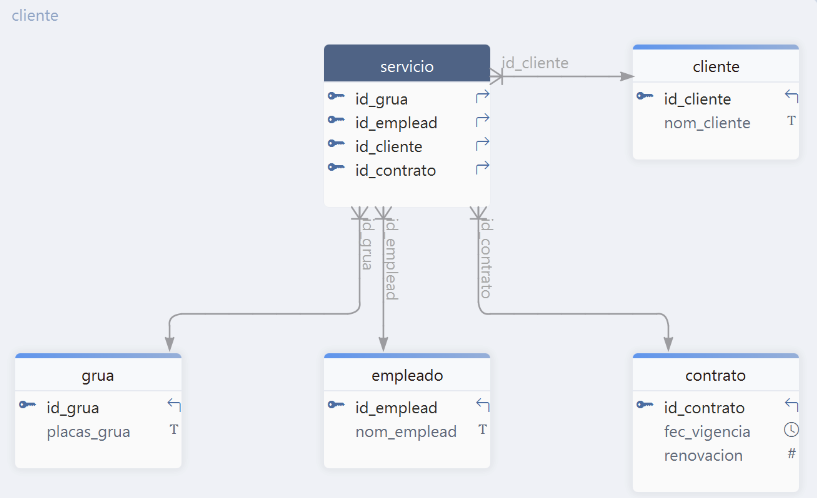
**Cuarta Forma Normal (4FN)**

Reglas:

* Estar en FNBC.
* No debe haber dependencias multivaluadas independientes.
* No hay multivaluadas independientes en este modelo. Cada contrato une un solo cliente, grúa y empleado.

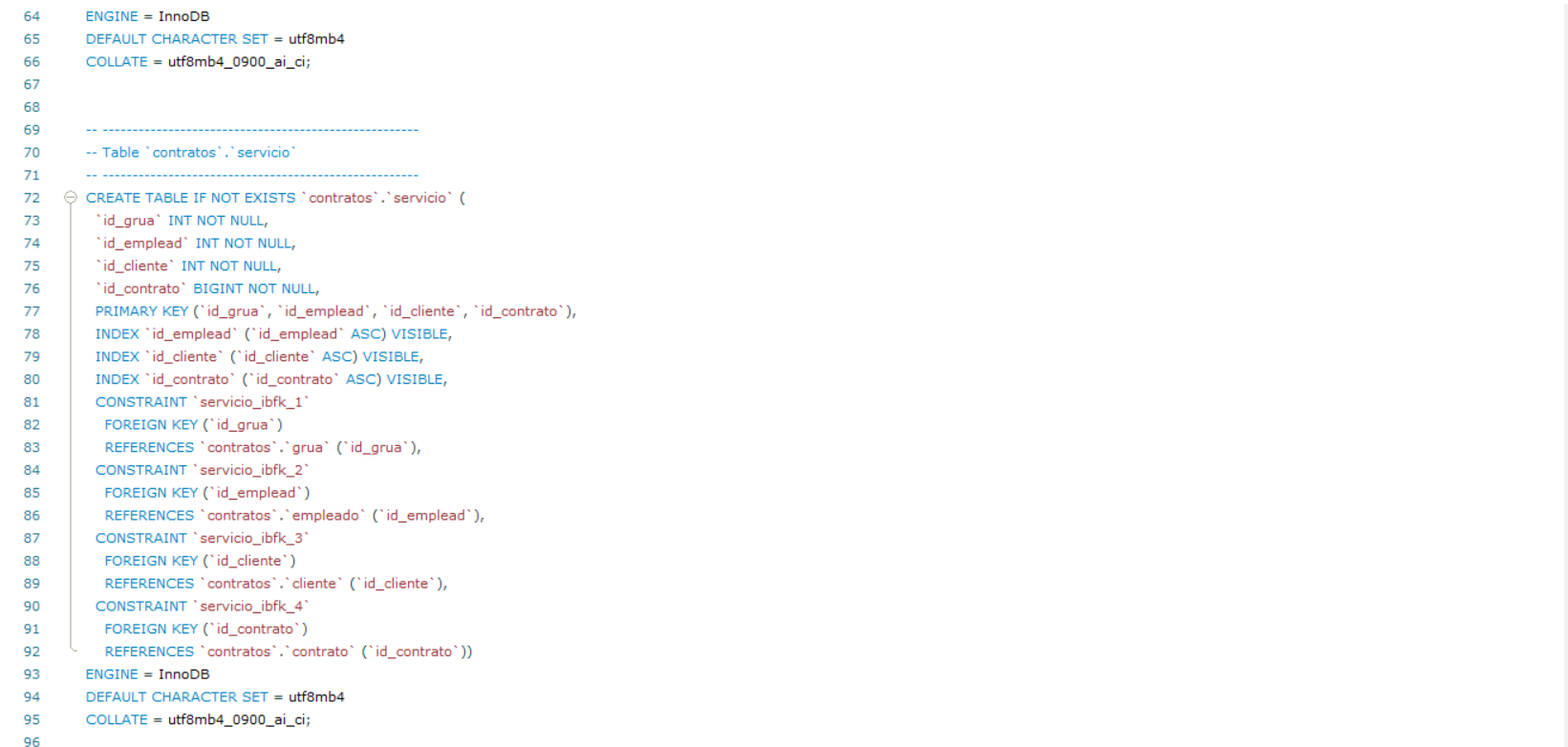
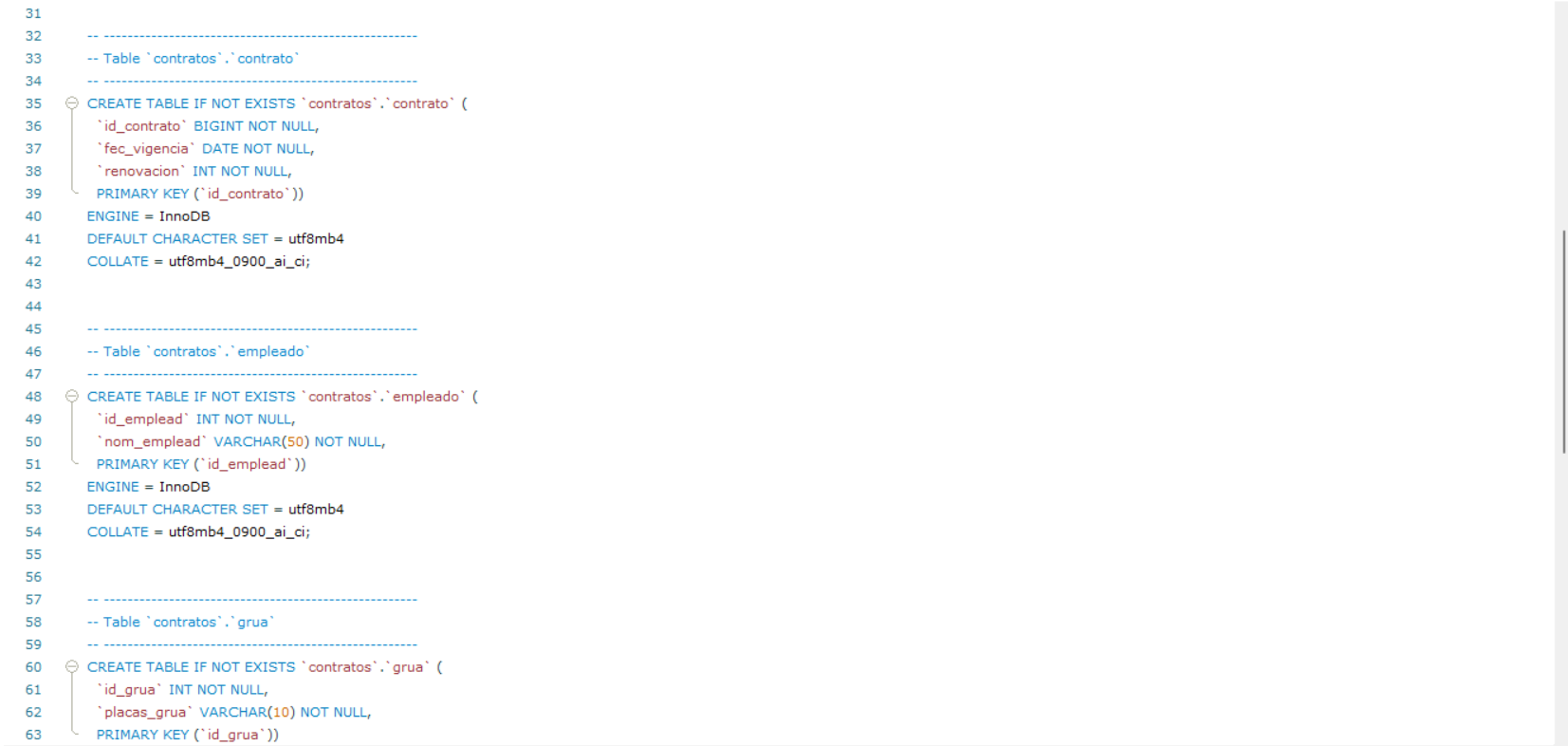
**Resultado: También está en 4FN.**

**Modelo relacional:**



**Código de salida:**







1. Realizar una propuesta para desnormalizar el modelo relacional presentado y fundamentar que ventaja tendría en determinado escenario.

Con la propuesta de desnormalización buscamos que se reduzca el número de tablas y optimizar la velocidad de lectura (consultas) en escenarios donde la integridad no sea tan crítica o donde los datos no cambien con frecuencia. La propuesta es la siguiente:

Combinar las 4 tablas en una sola tabla denormalizada que contenga todos los datos necesarios en una sola fila:

Tabla SERVICIO\_TOTAL

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id\_grua | placas\_grua | id\_empleado | nom\_empleado | id\_cliente | nom\_cliente | id\_contrato | fec\_vigencia | renovacion |

Esto significa eliminar la estructura modular de GRUA, EMPLEADO, CLIENTE y CONTRATO, y almacenar todo junto en una sola tabla.

*Escenario: Consultas frecuentes de reportes y estadísticas de servicios*

Por ejemplo, en una empresa de grúas donde se necesita mostrar constantemente en pantalla o imprimir reportes del tipo:

*“¿Qué grúa llevó a qué cliente, con qué operador, en qué fecha y cuántas renovaciones ha tenido?”*

En este caso:

* No es necesario hacer JOINs entre 4 tablas.
* Toda la información está en un solo SELECT.
* Las consultas son mucho más rápidas, lo cual es útil para dashboards o reportes de rendimiento.
* El sistema es más simple de mantener si los datos no cambian (por ejemplo, si los nombres de empleados o placas no se editan casi nunca).

**Desventajas**

* Redundancia de datos (ej. el nombre del empleado se repite cada vez que hace un servicio).
* Mayor riesgo de inconsistencias (si se actualiza el nombre del cliente en un lugar pero no en otro).
* Se rompe el principio de mantener la integridad referencial automáticamente.
* Desnormalizar el modelo relacional combinando GRUA, EMPLEADO, CLIENTE y CONTRATO en una sola tabla puede mejorar notablemente el rendimiento de lectura en sistemas de consulta intensiva, como reportes o dashboards en tiempo real, donde las operacionesde actualización son mínimas y la velocidad es crítica.

**Conclusiones**

* Guerrero López Uriel Ivan

A lo largo del desarrollo de este tema, logré comprender con mayor profundidad cómo la teoría de la normalización influye directamente en la calidad del diseño de una base de datos. Aprendí que normalizar no es solo un conjunto de pasos mecánicos, sino una herramienta poderosa para organizar la información de manera lógica, coherente y eficiente. A medida que fui aplicando las distintas formas normales, desde la Primera hasta la Tercera, me di cuenta de cómo cada una contribuye a reducir la redundancia, mejorar la integridad y facilitar futuras actualizaciones o consultas.

Uno de los mayores aprendizajes fue entender el papel de las dependencias funcionales, ya que son clave para decidir cómo dividir o reorganizar las tablas. Al trabajar con un caso práctico y muestras de datos reales, pude visualizar mejor cómo se aplican estos conceptos en escenarios concretos, lo que hizo más significativo el aprendizaje. No obstante, también enfrenté dificultades, especialmente al inicio, cuando intentaba diferenciar entre la Segunda y la Tercera Forma Normal. En varios momentos, me costó trabajo identificar si una dependencia era parcial o transitiva, lo cual afectaba el diseño. Me equivoqué al considerar ciertos atributos como dependientes directos de una clave primaria cuando en realidad dependían de otra combinación de campos. Fue necesario repasar definiciones y consultar ejemplos adicionales para poder corregir esos errores. A pesar de esos tropiezos, considero que he logrado dominar los fundamentos esenciales de la normalización y que ahora soy capaz de aplicarlos con mayor seguridad.

* Muñoz San Agustin Victoria Monserrat

Trabajar con la teoría de la normalización desde un enfoque práctico me permitió ver la diferencia entre diseñar una base de datos de manera intuitiva y hacerlo con fundamentos sólidos. Lo que más destaco de esta experiencia es haber aprendido a tomar decisiones justificadas sobre cómo organizar los datos, basándome en el análisis de las relaciones entre los atributos y no solamente en la lógica aparente de los datos. Gracias al ejercicio de implementar formas normales, pude desarrollar una mejor capacidad de análisis, especialmente al interpretar estructuras complejas o ambiguas dentro de una tabla.

Uno de los desafíos más grandes que enfrenté fue interpretar correctamente los datos de muestra. A veces, los valores parecían dar a entender una dependencia que no era real o era circunstancial, y eso me llevó a cometer errores en la elección de claves primarias o en la forma de descomponer las relaciones. También me costó establecer límites claros entre cuándo detener el proceso de normalización, ya que tenía la impresión de que mientras más lo hiciera, más "correcto" sería el modelo, cuando en realidad también es importante mantener un equilibrio con la eficiencia y la funcionalidad. A pesar de esas dificultades, considero que el proceso fue muy enriquecedor. Aprendí no solo a aplicar reglas, sino a razonar sobre por qué se aplican y cómo afectan al modelo general. Esta experiencia me dio herramientas concretas para mejorar el diseño de bases de datos en proyectos futuros, y también me ayudó a desarrollar un pensamiento más crítico al momento de estructurar la información.

**Referencias**

Chocobar, H. (2017). Cómo realizar la normalización de bases de datos y por qué. Recuperado el 09 de abril de 2025 de https://openwebinars.net/blog/como-realizar-la-normalizacion-de-bases-de-datos-y-por-que/

IBM (2025). Normalización en el diseño de base de datos. Recuperado el 09 de abril de 2025 de https://www.ibm.com/docs/es/db2-for-zos/13.0.0?topic=modeling-normalization-in-database-design

Pacheco, M. (2018). Normalización de Bases de Datos. Unidades de Apoyo para el Aprendizaje. CUAIEED/FCA-UNAM. Recuperado el 09 de abril de 2025 de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/872/mod\_resource/content/7/Contenido/index.html