UNIVERSIDAD POLITÉCNICA DE TULANCINGO

Modelado lógico caso práctico "Alquiler-Vivienda" ejemplo

por

2230229 Gonzalez Castro Elizabeth

2230738 Hurtado Vega Maria Estephania

2234539 Trejo Romo Adriana

2230016 Badillo Gonzalez Ambar

Ingeniería en Sistemas Computacionales

Asignatura:

Base de Datos

Nombre del Catedrático:

Mtro. Victor Hugo Fernández Cruz

Quinto Cuatrimestre

ISC 54

Tulancingo de Bravo, Hidalgo

U

P

T

Contenido

Modelado lógico caso práctico “Alquiler-Vivienda” ejemplo 3
Resumen 3
Introducción 4
Enunciado 5
Diagrama MER 5
Diccionario de Datos 6
Esquema de Transición 7
Diagrama ER 8
Descripción de primera, segunda y tercera forma normal, según Diagrama ER 8
Primera forma 8
Segunda forma normal 11
3ra Forma de la Normalización: 13
Código de creación de la base de datos según esquemas de transición 15
Datos de registro mínimo 10, máximo 15 Tuplas 23
Queries de Prueba (Joins) 28
Conclusión 30
Tabla de Ilustraciones ..... ¡Error! Marcador no definido.

Modelado lógico caso práctico “Alquiler-Vivienda” ejemplo

Resumen

El diseño y gestión de una base de datos para alquileres de viviendas implica varios pasos clave. Inicialmente, se emplea un Diagrama de Entidad-Relación (DER) para visualizar la estructura de la base de datos, junto con un Diccionario de Datos que detalla cada elemento del diagrama. Luego, se transita a un modelo relacional mediante un esquema de transición, que define las tablas, atributos y relaciones. MySQL Workbench es una herramienta útil para crear y visualizar estos diagramas ER. Es crucial asegurar la normalización de la base de datos hasta la tercera forma normal (3FN) para evitar redundancias y anomalías. La implementación se realiza mediante SQL, creando las tablas y estableciendo las relaciones necesarias. Finalmente, se insertan datos de ejemplo y se realizan consultas de prueba utilizando JOIN para combinar información de diferentes tablas. Este proceso proporciona una base sólida para gestionar eficazmente los alquileres de viviendas, garantizando la integridad y eficiencia de la base de datos.

Introducción

La gestión de alquileres de viviendas es un proceso complejo que involucra la interacción entre propietarios, inquilinos y las propias propiedades. Para llevar a cabo esta gestión de manera eficiente y organizada, es crucial contar con una base de datos bien diseñada que pueda almacenar y gestionar toda la información relevante de manera estructurada y accesible.

En este contexto, el diseño y la implementación de una base de datos para alquileres de viviendas se convierte en un aspecto fundamental. Este proceso implica la creación de un modelo de datos que represente adecuadamente las entidades involucradas, como propietarios, inquilinos y viviendas, así como las relaciones entre ellas, como los contratos de alquiler.

En esta tarea, se emplean herramientas como los Diagramas de Entidad-Relación (DER), que permiten visualizar la estructura de la base de datos de manera gráfica, junto con un Diccionario de Datos que proporciona detalles específicos sobre cada elemento del modelo.

Una vez diseñada la base de datos, se procede a su implementación, utilizando lenguajes de consulta como SQL para crear las tablas y establecer las relaciones entre ellas. Es importante asegurar que la base de datos esté normalizada hasta la tercera forma normal (3FN), lo que garantiza la integridad y eficiencia de los datos almacenados.

Finalmente, se realizan pruebas para verificar el correcto funcionamiento de la base de datos, insertando datos de ejemplo y ejecutando consultas para comprobar su funcionamiento. En resumen, el diseño y la implementación de una base de datos para alquileres de viviendas son procesos fundamentales para gestionar de manera efectiva este tipo de transacciones inmobiliarias.

Enunciado

Se desea gestionar los alquileres de viviendas de particulares, se requiere tener información de los propietarios que ofrecen sus viviendas en alquiler, así como de los inquilinos que se interesan por alquilar estas, obteniendo información sobre los alquilers y las viviendas alquiladas correspondientemente, saber de la duración de cada alquiler (histórico) y la información de renovación de un alquiler, por él inquilino.

<u>Diagrama MER</u>

Ilustración 1 Diagrama MER del caso Vivienda-Alquiler

Diccionario de Datos

Nombre de la base de datos:	Gestión del Alquiler de una Vivienda				Preparado por:		ISC-53				Fecha de preparación:
1. Identificación del archivo de datos					2. Reglas de validación						
Nombre del Archivo	Descripción del Archivo	Tabla a la que Pertenece	Nombre del Campo	Descripción del Campo	Llave Primaria	Llave Foránea	Campo obligatorio	Dominio o lista de valores	Tipo de datos	Longitud del campo	Regla de validación (en lenguaje natural)
					[Indicar si el campo hace parte de la llave primaria] [SI/NO]	[Si el campo hace parte de una llave foránea, indique el archivo y el campo de referencia]	[SI/NO]	[Liste los valores válidos, patrón o rango para el campo]	[Numérico, Cadena, Carácter, Entero, otro]	[Según el tipo de dato indicar la longitud del campo]	[Enumere claramente cada regla que se debe aplicar a la Variable]
GESTION ALQIILER-VIVIENDA	Diccionario de datos donde se registran las tablas, campos y sus caracteristicas	PROPIETARIO	ID-Propietario	ID de Identificación del propietario	SI	SI (ID-Propietario es clave foranea a PERSONA_P)	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
		PERSONA_P	INE	Credencial de elector del propietario	SI	SI (INE es clave foranea a NOMBRE_P y DIRECCION_P)	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	15	1. Campo requerido, NO DEBE SER NULO.
			Edad	Edad del propietario	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	3	1. Campo requerido, NO DEBE SER NULO.
			Teléfono	Teléfono para contactar al propietario	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	15	1. Campo requerido, NO DEBE SER NULO.
			Correo_Electrónico	Correo para contactar al propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
		NOMBRE_P	Nombre	Nombre del Propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, NO DEBE SER NULO.
			Nombre_2	2do nombre del Propietario	NO	NO	NO	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, DEBE SER NULO.
			Apellido_1	Apellido Paterno del Propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, NO DEBE SER NULO.
			Apellido_2	Apellido Materno del Propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, NO DEBE SER NULO.
		DIRECIÓN_P	Calle	Nombre de la calle de la casa del propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			Num_Ext	Número de la casa del propietario	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, NO DEBE SER NULO.
			Num_Int	2do Número de la casa del propietario	NO	NO	NO	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, DEBE SER NULO.
			Colonia	Nombre de la colonia de la casa del propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			C.P	Identificador del Área Geográfica Precisa del domicilio	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
			Entidad	Estado en el que se encuetra la casa.	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			Municipio	Ubicación geografica de la casa del propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
		INQUILINO	ID_Inquilino	ID de Identificación del propietario	SI	SI (ID-Inquilino es clave foranea a PERSONA_I)	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
			Comprobante_Ingresos	Comprobante de ingresos del inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	30	1. Campo requerido, NO DEBE SER NULO.
		NOMBRE_I	Nombre	Nombre del Inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, NO DEBE SER NULO.
			Nombre_2	2do nombre del Inquilino	NO	NO	NO	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, DEBE SER NULO.
			Apellido_1	Apellido Paterno del Inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, NO DEBE SER NULO.
			Apellido_2	Apellido Materno del Inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	10	1. Campo requerido, NO DEBE SER NULO.
		DIRECCIÓN_I	Calle	Nombre de la calle de la casa del Inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			Num_Ext	Número de la casa del Inquilino	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, NO DEBE SER NULO.
			Num_Int	2do Número de la casa del Inquilino	NO	NO	NO	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, DEBE SER NULO.
			Colonia	Nombre de la colonia de la casa del Inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			C.P	Identificador del Área Geográfica Precisa del domicilio	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
			Entidad	Estado en el que se encuetra la casa.	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			Municipio	Ubicación geografica de la casa del propietario	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
		PERSONA_I	INE	Credencial de elector del Inquilino	SI	SI (INE es clave foranea a NOMBRE_I y DIRECCION_I)	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	15	1. Campo requerido, NO DEBE SER NULO.
			Edad	Edad del Inquilino	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	3	1. Campo requerido, NO DEBE SER NULO.
			Teléfono	Teléfono para contactar al Inquilino	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	15	1. Campo requerido, NO DEBE SER NULO.
			Correo_Electrónico	Correo para contactar al Inquilino	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
		VIVIENDA	ID_Vivienda	Identificación	SI	SI (ID-Vivienda es clave foranea a DIRECCION_V, SERVICIOS_V y	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
		DIRECCIÓN_V	Calle	Nombre de la calle de la casa de la vivienda que se va	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			Num_Ext	Número de la casa de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, NO DEBE SER NULO.
			Num_Int	2do Número de la casa de la vivienda que se va alquilar	NO	NO	NO	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, DEBE SER NULO.
			Colonia	Nombre de la colonia de la casa de la vivienda que se va	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			C.P	Identificador del Área Geográfica Precisa del domicilio	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
			Entidad	Estado en el que se encuetra la vivienda que se va	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
			Municipio	Ubicación geografica de la casa de la vivienda que se va	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	20	1. Campo requerido, NO DEBE SER NULO.
		DESCRIPCIÓN_V	Extensión_Territorial	Medida del lugar de la vivienda que se va alquilar que se	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	4	1. Campo requerido, NO DEBE SER NULO.
			Num.Pisos	Información Casa de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	2	1. Campo requerido, NO DEBE SER NULO.
			Num.Baños	Información Casa de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	2	1. Campo requerido, NO DEBE SER NULO.
			Num.Recamaras	Información Casa de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	2	1. Campo requerido, NO DEBE SER NULO.
		SERVICIOS	Garaje	Información Casa de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	2	1. Campo requerido, NO DEBE SER NULO.
			Agua	Servicio de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	2	1. Campo requerido, NO DEBE SER NULO.
			Luz	Servicio de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	2	1. Campo requerido, NO DEBE SER NULO.
			Gas	Servicio de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	2	1. Campo requerido, NO DEBE SER NULO.
			Internet	Servicio de la vivienda que se va alquilar	NO	NO	SI	Valores Válidos: (A,B,C,D,E,F,...),(a,b,c,d,...)	Cadena de Caracteres	2	1. Campo requerido, NO DEBE SER NULO.
		ALQUILER	ID-Alquiler	Identificación del alquiler	SI	SI (ID-ALQUILER es clave foranea a PAGO_A)	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
			No_Pago	Información del numero de pago	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.
			Fecha_Fin	Fecha de inicio en la que se alquila la casa	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	6	1. Campo requerido, NO DEBE SER NULO.
			Fecha_Inicio	Fecha final en la que se entrega la casa	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Fecha	6	1. Campo requerido, NO DEBE SER NULO.
		PAGO_A	Monto	Cantidad a pagar del alquiler	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	10	1. Campo requerido, NO DEBE SER NULO.
			Fecha_Limite	Fecha limite de pago	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Fecha	6	1. Campo requerido, NO DEBE SER NULO.
			Fecha_Pago	Fecha en la que se pagó	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Fecha	6	1. Campo requerido, NO DEBE SER NULO.
			Folio	Número de identificación del pago	NO	NO	SI	Valores Válidos: (1,2,3,4...)	Entero	5	1. Campo requerido, NO DEBE SER NULO.

Ilustración 2 Diccionario de Datos del caso Vivienda-Alquiler

Esquema de Transición

- PROPIETARIO (ID-Propietario, No_Escritura), PERSONA_P (INE, Edad, Teléfono, Correo_Electrónico) PERSONA_P. ID_Propietario es una clave ajena a PROPIETARIO, NOMBRE (Nombre_1, Nombre_2, Apellido_1, Apellido_2) NOMBRE_P.INE es una clave ajena a PERSONA_P, DIRECCIÓN (Calle, Num_Ext, Num_Int, Colonia, C.P, Entidad, Municipio) DIRECCIÓN_P.INE es una clave ajena a PERSONA_P.

- VIVIENDA (ID_vivienda), DIRECCIÓN_V (Calle, Num_Ext, Num_Int, Colonia, C.P, Entidad, Municipio) DIRECCION_V.ID-Vivienda es una clave ajena a VIVIENDA, DESCRIPCIÓN_V (Extensión_Territorial, Num.Pisos, Num.Baños, Num.Recamaras, Garaje), DESCRIPCIÓN_V.ID-Viviendas es una clave ajena a VIVIENDA, SERVICIOS_V (Agua, Luz, Gas, Internet) SERVICIOS_V.ID-viviendas es una clave ajena a VIVIENDA.

- ALQUILER (ID_Alquiler, No.Pago, Fecha_Fin, Fecha_Inicio), PAGO_A (Monto, Fecha_Limite, Fecha_Pago, Folio) PAGO_A.ID-alquiler es una clave ajena a ALQUILER.

- INQUILINO (ID_Inquilino, Comprobante_Ingresos), PERSONA_I (INE, Edad, Teléfono, Correo_Electrónico) PERSONA_I. ID_Inquilino es una clave ajena a INQUILINO, NOMBRE_I (Nombre, Nombre_2, Apellido_1, Apellido_2) NOMBRE_I.INE es una clave ajena a PERSONA_I, DIRECCIÓN_I (Calle, Num_Ext, Num_Int, Colonia, C.P, Entidad, Municipio) DIRECCIÓN_I.INE es una clave ajena a PERSONA_I.

Diagrama ER

Ilustración 3 Diagrama ER del caso Vivienda-Alquiler

<u>Descripción de primera, segunda y tercera forma normal, según Diagrama ER</u>

Primera forma

La primera forma normal (1FN) en bases de datos es un principio fundamental que establece que cada celda de una tabla debe contener un solo valor atómico, es decir, un valor que no se pueda dividir en partes más pequeñas con significado independiente. Esto implica que no debe haber grupos repetitivos de datos dentro de ninguna celda, lo que garantiza la integridad y la coherencia de los datos almacenados.

Al analizar el esquema de transición proporcionado con respecto a la 1FN, es importante examinar cada relación y sus atributos para determinar si cumplen con este principio. Vamos a detallar cada parte del esquema:

PROPIETARIO:

• Los atributos ID-Propietario y No_Escritura parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación parece cumplir con la 1FN.

PERSONA_P:

• Los atributos INE, Edad, Teléfono y Correo_Electrónico parecen ser valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación también parece cumplir con la 1FN.

NOMBRE:

• Sin embargo, los atributos Nombre_1, Nombre_2, Apellido_1 y Apellido_2 representan conjuntos de datos relacionados con el nombre completo de una persona. Esto no cumple con la 1FN, ya que cada celda debería contener un solo valor atómico. Se sugiere dividir estos atributos en partes más pequeñas, como Nombre y Apellido, para cumplir con la 1FN.

DIRECCIÓN:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

VIVIENDA:

• El atributo ID_vivienda parece ser un valor individual y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

DESCRIPCIÓN_V:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

SERVICIOS_V:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación también cumple con la 1FN.

ALQUILER:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

PAGO_A:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

INQUILINO:

• Los atributos ID_Inquilino y Comprobante_Ingresos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

PERSONA_I:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación también cumple con la 1FN.

NOMBRE_I:

• Similar a la relación NOMBRE, los atributos Nombre, Nombre_2, Apellido_1 y Apellido_2 representan conjuntos de datos relacionados con el nombre completo de una persona. Esto no cumple con la 1FN y se sugiere dividir estos atributos en partes más pequeñas para cumplir con este principio.

DIRECCIÓN_I:

• Todos los atributos parecen representar valores individuales y no hay grupos repetitivos de datos en ninguna celda. Por lo tanto, esta relación cumple con la 1FN.

En resumen, la mayoría de las relaciones en el esquema de transición parecen cumplir con la primera forma normal (1FN), con la excepción de las relaciones NOMBRE y NOMBRE_I, donde se sugiere una reestructuración de los atributos para garantizar que cada celda contenga solo un valor atómico. Esto es fundamental para mantener la integridad y la coherencia de los datos en la base de datos.

Segunda forma normal

- PROPIETARIO:

Esta entidad representa a los propietarios de las viviendas. Contiene atributos como ID_Propietario y No_Escritura, y tiene una relación con la entidad PERSONA_P a través de la clave ajena ID_Propietario.

- PERSONA_P:

Esta entidad contiene información personal de los propietarios, como INE (Identificación Nacional Electoral), Edad, Teléfono y Correo_Electrónico. La clave ajena INE hace referencia a la entidad NOMBRE_P.

- NOMBRE_P:

Esta entidad almacena los nombres de los propietarios. La clave ajena INE la relaciona con la entidad PERSONA_P.

- DIRECCIÓN_P:

Esta entidad representa la dirección de los propietarios. La clave ajena INE la con la entidad PERSONA_P.

- VIVIENDA:

Esta entidad representa las viviendas disponibles para alquiler. Contiene atributos como ID_Vivienda y tiene relaciones con otras entidades como DIRECCIÓN_V, DESCRIPCIÓN_V y SERVICIOS_V.

- DIRECCIÓN_V:

Almacena la dirección de las viviendas. La clave ajena ID_Vivienda la relaciona con la entidad VIVIENDA.

- DESCRIPCIÓN_V:

Esta entidad contiene detalles sobre las viviendas, como Extensión_Territorial, Num.Pisos, Num.Baños, Num.Recamaras, Garaje. La clave ajena ID_Vivienda la relaciona con la entidad VIVIENDA.

- SERVICIOS_V:

Almacena información sobre los servicios disponibles en las viviendas, como Agua, Luz, Gas e Internet. La clave ajena ID_Vivienda la relaciona con la entidad VIVIENDA.

- ALQUILER:

Esta entidad representa los contratos de alquiler. Contiene atributos como ID_Alquiler, No.Pago, Fecha_Fin y Fecha_Inicio, y tiene una relación con la entidad PAGO_A a través de la clave ajena ID_Alquiler.

- PAGO_A:

Almacena información sobre los pagos asociados con los contratos de alquiler, como Monto, Fecha_Limite, Fecha_Pago y Folio. La clave ajena ID_Alquiler la relaciona con la entidad ALQUILER.

- INQUILINO:

Representa a los inquilinos que alquilan las viviendas. Contiene atributos como ID_Inquilino y Comprobante_Ingresos, y tiene una relación con la entidad PERSONA_I a través de la clave ajena ID_Inquilino.

- PERSONA_I:

Contiene información personal de los inquilinos, como INE, Edad, Teléfono y Correo_Electrónico. La clave ajena INE la relaciona con la entidad NOMBRE_I.

- NOMBRE_I:

Almacena los nombres de los inquilinos. La clave ajena INE la relaciona con la entidad PERSONA_I.

- DIRECCIÓN_I:

Representa la dirección de los inquilinos. La clave ajena INE la relaciona con la entidad PERSONA_I.

3ra Forma de la Normalización:

Tabla PROPIETARIOS:

- ID_Propietario (clave primaria)
- Nombre
- Apellido
- Dirección
- Teléfono
- Correo electrónico

Tabla VIVIENDA:

- ID_Vivienda (clave primaria)
- ID_Propietario (clave foránea referenciando a Propietarios)
- Dirección
- Tipo de vivienda
- Número de habitaciones
- Precio de alquiler

Tabla INQUILINO:

- ID_Inquilino (clave primaria)
- Nombre
- Apellido
- Dirección
- Teléfono
- Correo electrónico

Tabla ALQUILER:

- ID_Alquiler (clave primaria)
- ID_Vivienda (clave foránea referenciando a Viviendas)
- ID_Inquilino (clave foránea referenciando a Inquilinos)
- Fecha de inicio
- Duración (en meses)
- Precio mensual

- Fecha de finalización

Código de creación de la base de datos según esquemas de transición

Este es el código necesario para crear cada una de las tablas en nuestra base de datos.

- Iniciar sesión en MySQL y seleccionar la base de datos Alquiler-Vivienda

  mysql -u usuario -p contraseña

  USE Alquiler-Vivienda;

- Crear la tabla PROPIETARIO

  CREATE TABLE IF NOT EXISTS PROPIETARIO (

  -> ID_Propietario INT NOT NULL,

  ->No_Escrituras INT NOT NULL,

  ->PRIMARY KEY (ID_Propietario)

  ->);

- Crear la tabla PERSONA_P

  CREATE TABLE IF NOT EXISTS PERSONA_P (

  -> INE VARCHAR(25) NOT NULL,

  -> Edad INT NOT NULL,

  ->Teléfono VARCHAR(15) NOT NULL,

  ->Correo_Electrónico VARCHAR(30) NULL,

->PROPIETARIO_ID_Propietario INT NOT NULL,

->PRIMARY KEY (INE),

->FOREIGN KEY (PROPIETARIO_ID_Propietario) REFERENCES PROPIETARIO(ID_Propietario)

->);

- Crear la tabla NOMBRE_P

CREATE TABLE IF NOT EXISTS NOMBRE_P (

->Nombre VARCHAR(10) NOT NULL,

->Nombre_2 VARCHAR(10) NULL,

->Apellido_1 VARCHAR(45) NOT NULL,

->Apellido_2 VARCHAR(45) NOT NULL,

->PERSONA_P_INE VARCHAR(25) NOT NULL,

->PRIMARY KEY (PERSONA_P_INE),

->FOREIGN KEY (PERSONA_P_INE) REFERENCES PERSONA_P(INE)

->);

- Crear la tabla ALQUILER

CREATE TABLE IF NOT EXISTS ALQUILER (

-> ID_Alquiler INT NOT NULL,

-> No_Pago INT NOT NULL,

->Fecha_Fin DATE NOT NULL,

->Fecha_Inicio VARCHAR(45) NOT NULL,

->PRIMARY KEY (ID_Alquiler)

->);

- Crear la tabla VIVIENDA

CREATE TABLE IF NOT EXISTS VIVIENDA (

->ID_Vivienda INT NOT NULL,

->PROPIETARIO_ID_Propietario INT NOT NULL,

->ALQUILER_ID_Alquiler INT NOT NULL,

->PRIMARY KEY (ID_Vivienda, ALQUILER_ID_Alquiler),

->FOREIGN KEY (PROPIETARIO_ID_Propietario) REFERENCES PROPIETARIO(ID_Propietario),

->FOREIGN KEY (ALQUILER_ID_Alquiler) REFERENCES ALQUILER(ID_Alquiler)

->);

- Crear la tabla DIRECCIÓN_V

CREATE TABLE IF NOT EXISTS DIRECCIÓN_V (

->Calle VARCHAR(20) NOT NULL,

->Num_Ext INT NOT NULL,

-> Num_Int INT NULL,

```
->Colonia VARCHAR(20) NOT NULL,

->C_P INT NOT NULL,

->Entidad VARCHAR(15) NOT NULL,

->Municipio VARCHAR(20) NOT NULL,

->VIVIENDA_ID_Vivienda INT NOT NULL,

->PRIMARY KEY (VIVIENDA_ID_Vivienda),

->FOREIGN KEY (VIVIENDA_ID_Vivienda) REFERENCES VIVIENDA(ID_Vivienda)

->);
```

- Crear la tabla DESCRIPCIÓN_V

```
CREATE TABLE IF NOT EXISTS DESCRIPCIÓN_V (

->Extensión_Territorial INT NOT NULL,

-> Num_Pisos INT NOT NULL,

->Num_Baños INT NOT NULL,

->Num_Habit INT NOT NULL,

->Garage TINYINT NOT NULL,

->VIVIENDA_ID_Vivienda INT NOT NULL,

->PRIMARY KEY (VIVIENDA_ID_Vivienda),

->FOREIGN KEY (VIVIENDA_ID_Vivienda) REFERENCES VIVIENDA(ID_Vivienda)
```

->);

- Crear la tabla SERVICIOS_V

```
CREATE TABLE IF NOT EXISTS SERVICIOS_V (
    ->Agua VARCHAR(2) NOT NULL,
    ->Luz VARCHAR(2) NOT NULL,
    ->Gas VARCHAR(2) NOT NULL,
    ->Internet VARCHAR(2) NULL,
    ->VIVIENDA_ID_Vivienda INT NOT NULL,
    ->PRIMARY KEY (VIVIENDA_ID_Vivienda),
    ->FOREIGN KEY (VIVIENDA_ID_Vivienda) REFERENCES VIVIENDA(ID_Vivienda)
    ->);
```

- Crear la tabla PAGO_A

```
CREATE TABLE IF NOT EXISTS PAGO_A (
    ->Monto FLOAT NOT NULL,
    ->Fecha_Limite DATE NOT NULL,
    ->Fecha_Pago DATE NOT NULL,
    ->Folio INT NOT NULL,
    ->ALQUILER_ID_Alquiler INT NOT NULL,
```

```
->PRIMARY KEY (Folio),
->FOREIGN KEY (ALQUILER_ID_Alquiler) REFERENCES ALQUILER(ID_Alquiler)
->);
```

- Crear la tabla INQUILINO

```
CREATE TABLE IF NOT EXISTS INQUILINO (
->ID_Inquilino INT NOT NULL,
->Comprobante_Ingresos VARCHAR(45) NOT NULL,
->ALQUILER_ID_Alquiler INT NOT NULL,
->PRIMARY KEY (ID_Inquilino, ALQUILER_ID_Alquiler),
->FOREIGN KEY (ALQUILER_ID_Alquiler) REFERENCES ALQUILER(ID_Alquiler)
->);
```

- Crear la tabla PERSONA_I

```
CREATE TABLE IF NOT EXISTS PERSONA_I (
-> INE VARCHAR(25) NOT NULL,
->Edad INT NOT NULL,
->Teléfono VARCHAR(15) NOT NULL,
->Correo_Electrónico VARCHAR(30) NULL,
```

```
-> INQUILINO_ID_Inquilino INT NOT NULL,
->PRIMARY KEY (INE),
->FOREIGN KEY (INQUILINO_ID_Inquilino) REFERENCES INQUILINO(ID_Inquilino)
->);
```

- Crear la tabla NOMBRE_I

```
CREATE TABLE IF NOT EXISTS NOMBRE_I (
->Nombre VARCHAR(10) NOT NULL,
->Nombre_2 VARCHAR(10) NULL,
->Apellido_1 VARCHAR(45) NOT NULL,
->Apellido_2 VARCHAR(45) NOT NULL,
->PERSONA_I_INE VARCHAR(25) NOT NULL,
->PRIMARY KEY (PERSONA_I_INE),
->FOREIGN KEY (PERSONA_I_INE) REFERENCES PERSONA_I(INE)
->);
```

- Crear la tabla DIRECCIÓN_I

```
CREATE TABLE IF NOT EXISTS DIRECCIÓN_I (
->Calle VARCHAR(20) NOT NULL,
-> Num_Ext INT NOT NULL,
```

```
->Num_Int INT NULL,

->Colonia VARCHAR(20) NOT NULL,

->C_P INT NOT NULL,

->Entidad VARCHAR(15) NOT NULL,

->Municipio VARCHAR(20) NOT NULL,

->PERSONA_I_INE VARCHAR(25) NOT NULL,

->PRIMARY KEY (PERSONA_I_INE),

->FOREIGN KEY (PERSONA_I_INE) REFERENCES PERSONA_I(INE)

->);
```

- Salir del cliente MySQ es EXIT;

Datos de registro mínimo 10, máximo 15 Tuplas

```
mysql> use alquiler;
Database changed
mysql> CREATE TABLE IF NOT EXISTS PROPIETARIO (
    -> ID_Propietario INT NOT NULL,
    -> No_Escrituras INT NOT NULL,
    -> PRIMARY KEY (ID_Propietario)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE IF NOT EXISTS PERSONA_P (
    -> INE VARCHAR (25) NOT NULL,
    -> Edad INT NOT NULL,
    -> Telefono VARCHAR(15) NOT NULL,
    -> Correo_Electronico VARCHAR (30) NULL,
    -> PROPIETARIO_ID_Propietario INT NOT NULL,
    -> PRIMARY KEY (INE),
    -> FOREIGN KEY (PROPIETARIO_ID_Propietario) REFERENCES PROPIETARIO(ID_Propietario)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE IF NOT EXISTS NOMBRE_P (
    -> Nombre VARCHAR(10) NOT NULL,
    -> Nombre_2 VARCHAR(10) NULL,
    -> Apellido_1 VARCHAR(45) NOT NULL,
    -> Apellido_2 VARCHAR(45) NOT NULL,
    -> PERSONA_P_INE VARCHAR(25) NOT NULL,
    -> PRIMARY KEY (PERSONA_P_INE),
    -> FOREIGN KEY (PERSONA_P_INE) REFERENCES PERSONA_P(INE)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE IF NOT EXISTS ALQUILER (
    -> ID_Alquiler INT NOT NULL,
    -> No_Pago INT NOT NULL,
    -> Fecha_Inicio VARCHAR(45) NOT NULL,
    -> Fecha_Fin DATE NOT NULL,
    -> PRIMARY KEY (ID_Alquiler)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

Ilustración 4 Creación de las tablas: PROPIETARIO, PERSONA_P, NOMBRE_P Y ALQUILER

```
mysql> INSERT INTO PROPIETARIO VALUES(2, 876543219);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PROPIETARIO VALUES(3, 765432198);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(4, 654321987);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(5, 543219876);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(6, 432198765);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(7, 321987654);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(8, 219876543);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(9, 198765432);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPIETARIO VALUES(10, 987654321);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * from PROPIETARIO;
+----------------+---------------+
| ID_Propietario | No_Escrituras |
+----------------+---------------+
|              1 |     987654321 |
|              2 |     876543219 |
|              3 |     765432198 |
|              4 |     654321987 |
|              5 |     543219876 |
|              6 |     432198765 |
|              7 |     321987654 |
|              8 |     219876543 |
|              9 |     198765432 |
|             10 |     987654321 |
+----------------+---------------+
```

Ilustración 5 Código Para llenar la tabla PROPIETARIO junto con la tabla con los datos ya ingresados

```
mysql> INSERT INTO PERSONA_P VALUES ("8BCD246A", 35, '7751203464', '2230260@
upt.edu.mx', 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("68CD246AB", 42, '7751703964', '2230453
@upt.edu.mx', 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("68CD246AB", 27, '7751047224', '2230913
@upt.edu.mx', 4);
ERROR 1062 (23000): Duplicate entry '68CD246AB' for key 'persona_p.PRIMARY'
mysql> INSERT INTO PERSONA_P VALUES ("468D2ABC", 27, '7751047224', '2230913@
upt.edu.mx', 4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("4682ABCD", 29, '7751041224', '2230343@
upt.edu.mx', 5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("D682ABC4", 40, '7751290374', '2230600@
upt.edu.mx', 6);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("CD68AB42", 35, '7711250374', '2230690@
upt.edu.mx', 7);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("BCD8AB426", 39, '7711250076', '2230543
@upt.edu.mx', 8);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("ABCD4268", 23, '7712479902', '2230476@
upt.edu.mx', 9);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PERSONA_P VALUES ("ABC4268D", 44, '7714054050', '2230630@
upt.edu.mx', 10);
Query OK, 1 row affected (0.00 sec)
```

Ilustración 6 Código para llenar los campos de la tabla de PERSONA_P

INE	Edad	Telefono	Correo_Electronico	PROPIETARIO_ID_Propietario
4682ABCD	29	7751041224	2230343@upt.edu.mx	5
468D2ABC	27	7751047224	2230913@upt.edu.mx	4
68CD246AB	42	7751703964	2230453@upt.edu.mx	3
8BCD246A	35	7751203464	2230260@upt.edu.mx	2
ABC4268D	44	7714054050	2230630@upt.edu.mx	10
ABCD2468	33	7711634137	2230161@upt.edu.mx	1
ABCD4268	23	7712479902	2230476@upt.edu.mx	9
BCD8AB426	39	7711250076	2230543@upt.edu.mx	8
CD68AB42	35	7711250374	2230690@upt.edu.mx	7
D682ABC4	40	7751290374	2230600@upt.edu.mx	6

10 rows in set (0.00 sec)

Ilustración 7 Tabla PERSONA_P llena con los datos ingresados en cada campo

```
mysql> INSERT INTO NOMBRE_P VALUES ("Juan", "Manuel", "Gomez", "Perez", "ABC4268D");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO NOMBRE_P VALUES ("Jennifer", NULL, "Jimenez", "Arredondo"
, "D682ABC4");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO NOMBRE_P VALUES ("Alejandro", NULL, "Lopez", "Martinez",
"CD68AB42");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO NOMBRE_P VALUES ("Juan", "Mauricio", "Soto", "Anaya", "BC
D8AB426");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO NOMBRE_P VALUES ("Cristian", NULL, "Gomez", "Gonzales", "
ABCD4268");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO NOMBRE_P VALUES ("Enrique", NULL, "Lopez", "Hernandez", "
ABCD2468");
Query OINSERT INTO NOMBRE_P VALUES ("Fatima", "Paola", "Cuellar", "Blanco",

mysql> INSERT INTO NOMBRE_P VALUES ("Fatima", "Paola", "Cuellar", "Blanco",
"8BCD246A");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO NOMBRE_P VALUES ("Fatima", "Paola", "Cuellar", "Blanco", "8BCD246A");
```

Ilustración 8 Código con los campos llenos de datos ingresados de la tabla NOMBRE_P

```
mysql> SELECT * from NOMBRE_P;
+-----------+----------+------------+------------+---------------+
| Nombre    | Nombre_2 | Apellido_1 | Apellido_2 | PERSONA_P_INE |
+-----------+----------+------------+------------+---------------+
| Fredy     | Ivan     | Angeles    | Gonzales   | 4682ABCD      |
| Martin    | NULL     | Jardinez   | Romero     | 468D2ABC      |
| Adrian    | NULL     | Felipe     | Susano     | 68CD246AB     |
| Fatima    | Paola    | Cuellar    | Blanco     | 8BCD246A      |
| Juan      | Manuel   | Gomez      | Perez      | ABC4268D      |
| Enrique   | NULL     | Lopez      | Hernandez  | ABCD2468      |
| Cristian  | NULL     | Gomez      | Gonzales   | ABCD4268      |
| Juan      | Mauricio | Soto       | Anaya      | BCD8AB426     |
| Alejandro | NULL     | Lopez      | Martinez   | CD68AB42      |
| Jennifer  | NULL     | Jimenez    | Arredondo  | D682ABC4      |
+-----------+----------+------------+------------+---------------+
10 rows in set (0.00 sec)
```

Ilustración 9 Tabla con los datos ingresados en el campo correspondiente de la tabla NOMBRE_P

Queries de Prueba (Joins)

```
mysql> SELECT *
    -> FROM PROPIETARIO p
    -> JOIN PERSONA_P pp ON p.ID_Propietario = pp.PROPIETARIO_ID_Propietario
    -> JOIN NOMBRE_P np ON pp.INE = np.PERSONA_P_INE;
```

ID_Propietario	No_Escrituras	INE	Edad	Telefono	Correo_Electronico	PROPIETARIO_ID_Propietario	Nombre	Nombre_2	Apellido_1	Apellido_2	PERSONA_P_INE
5	543219876	4682ABCD	29	7751041224	2230343@upt.edu.mx	5	Fredy	Ivan	Angeles	Gonzales	4682ABCD
4	654321987	468D2ABC	27	7751047224	2230913@upt.edu.mx	4	Martin	NULL	Jardinez	Romero	468D2ABC
3	765432198	68CD246AB	42	7751703964	2230453@upt.edu.mx	3	Adrian	NULL	Felipe	Susano	68CD246AB
2	876543219	8BCD246A	35	7751203464	2230260@upt.edu.mx	2	Fatima	Paola	Cuellar	Blanco	8BCD246A
10	987654321	ABC4268D	44	7714054050	2230630@upt.edu.mx	10	Juan	Manuel	Gomez	Perez	ABC4268D
1	987654321	ABCD2468	33	7711634137	2230161@upt.edu.mx	1	Enrique	NULL	Lopez	Hernandez	ABCD2468
9	198765432	ABCD4268	23	7712479902	2230476@upt.edu.mx	9	Cristian	NULL	Gomez	Gonzales	ABCD4268
8	219876543	BCD8AB426	39	7711250076	2230543@upt.edu.mx	8	Juan	Mauricio	Soto	Anaya	BCD8AB426
7	321987654	CD68AB42	35	7711250374	2230690@upt.edu.mx	7	Alejandro	NULL	Lopez	Martinez	CD68AB42
6	432198765	D682ABC4	40	7751290374	2230600@upt.edu.mx	6	Jennifer	NULL	Jimenez	Arredondo	D682ABC4

```
10 rows in set (0.00 sec)
```

Ilustración 10 Join de Prueba del caso Vivienda-Alquiler

```
mysql> SELECT *
    -> FROM PROPIETARIO p
    -> JOIN PERSONA_P pp ON p.ID_Propietario = pp.PROPIETARIO_ID_Propietario;
+----------------+---------------+-----------+------+------------+--------------------+----------------------------+
| ID_Propietario | No_Escrituras | INE       | Edad | Telefono   | Correo_Electronico | PROPIETARIO_ID_Propietario |
+----------------+---------------+-----------+------+------------+--------------------+----------------------------+
|              5 |     543219876 | 4682ABCD  |   29 | 7751041224 | 2230343@upt.edu.mx |                          5 |
|              4 |     654321987 | 468D2ABC  |   27 | 7751047224 | 2230913@upt.edu.mx |                          4 |
|              3 |     765432198 | 68CD246AB |   42 | 7751703964 | 2230453@upt.edu.mx |                          3 |
|              2 |     876543219 | 8BCD246A  |   35 | 7751203464 | 2230260@upt.edu.mx |                          2 |
|             10 |     987654321 | ABC4268D  |   44 | 7714054050 | 2230630@upt.edu.mx |                         10 |
|              1 |     987654321 | ABCD2468  |   33 | 7711634137 | 2230161@upt.edu.mx |                          1 |
|              9 |     198765432 | ABCD4268  |   23 | 7712479902 | 2230476@upt.edu.mx |                          9 |
|              8 |     219876543 | BCD8AB426 |   39 | 7711250076 | 2230543@upt.edu.mx |                          8 |
|              7 |     321987654 | CD68AB42  |   35 | 7711250374 | 2230690@upt.edu.mx |                          7 |
|              6 |     432198765 | D682ABC4  |   40 | 7751290374 | 2230600@upt.edu.mx |                          6 |
+----------------+---------------+-----------+------+------------+--------------------+----------------------------+
10 rows in set (0.00 sec)
```

Ilustración 11 Join de prueba con ON en el caso Vivienda-Alquiler

```
mysql> SELECT *
    -> FROM PERSONA_P pp
    -> JOIN NOMBRE_P np ON pp.INE = np.PERSONA_P_INE;
+-----------+------+------------+--------------------+----------------------------+----------+----------+------------+------------+---------------+
| INE       | Edad | Telefono   | Correo_Electronico | PROPIETARIO_ID_Propietario | Nombre   | Nombre_2 | Apellido_1 | Apellido_2 | PERSONA_P_INE |
+-----------+------+------------+--------------------+----------------------------+----------+----------+------------+------------+---------------+
| 4682ABCD  |   29 | 7751041224 | 2230343@upt.edu.mx |                          5 | Fredy    | Ivan     | Angeles    | Gonzales   | 4682ABCD      |
| 468D2ABC  |   27 | 7751047224 | 2230913@upt.edu.mx |                          4 | Martin   | NULL     | Jardinez   | Romero     | 468D2ABC      |
| 68CD246AB |   42 | 7751703964 | 2230453@upt.edu.mx |                          3 | Adrian   | NULL     | Felipe     | Susano     | 68CD246AB     |
| 8BCD246A  |   35 | 7751203464 | 2230260@upt.edu.mx |                          2 | Fatima   | Paola    | Cuellar    | Blanco     | 8BCD246A      |
| ABC4268D  |   44 | 7714054050 | 2230630@upt.edu.mx |                         10 | Juan     | Manuel   | Gomez      | Perez      | ABC4268D      |
| ABCD2468  |   33 | 7711634137 | 2230161@upt.edu.mx |                          1 | Enrique  | NULL     | Lopez      | Hernandez  | ABCD2468      |
| ABCD4268  |   23 | 7712479902 | 2230476@upt.edu.mx |                          9 | Cristian | NULL     | Gomez      | Gonzales   | ABCD4268      |
| BCD8AB426 |   39 | 7711250076 | 2230543@upt.edu.mx |                          8 | Juan     | Mauricio | Soto       | Anaya      | BCD8AB426     |
| CD68AB42  |   35 | 7711250374 | 2230690@upt.edu.mx |                          7 | Alejandro | NULL     | Lopez      | Martinez   | CD68AB42      |
| D682ABC4  |   40 | 7751290374 | 2230600@upt.edu.mx |                          6 | Jennifer | NULL     | Jimenez    | Arredondo  | D682ABC4      |
+-----------+------+------------+--------------------+----------------------------+----------+----------+------------+------------+---------------+
10 rows in set (0.00 sec)
```

Ilustración 12 Join de prueba con el INE en ON en el caso Vivienda-Alquiler

Conclusión

El diseño y la implementación de una base de datos para la gestión de alquileres de viviendas es un proceso fundamental para asegurar la eficiencia, organización y precisión en la administración de este tipo de transacciones inmobiliarias. A lo largo de este proceso, se han abordado diversas etapas clave, desde la creación de un modelo conceptual mediante Diagramas de Entidad-Relación (DER) y un detallado Diccionario de Datos, hasta la implementación práctica utilizando lenguajes de consulta como SQL.

La utilización de herramientas como MySQL Workbench facilita la visualización y diseño del modelo de datos, permitiendo una representación clara de las entidades, atributos y relaciones que componen el sistema. Además, se ha destacado la importancia de la normalización de la base de datos hasta la tercera forma normal (3FN), lo que garantiza la integridad y consistencia de los datos almacenados, evitando redundancias y anomalías.

Una vez implementada la base de datos, es crucial llevar a cabo pruebas exhaustivas para verificar su correcto funcionamiento. Esto incluye la inserción de datos de ejemplo y la ejecución de consultas para comprobar la precisión y eficiencia en la recuperación de la información.

En resumen, el diseño y la implementación de una base de datos para alquileres de viviendas proporciona una sólida base para gestionar eficazmente este tipo de transacciones, mejorando la organización, accesibilidad y precisión de la información, y contribuyendo a una gestión más eficiente y satisfactoria tanto para propietarios como para inquilinos.