LO2 Bases de datos



**Índice**

[**P2**](#_ryo0gtb5hcrq) **3**

[**P3**](#_c065orn6axyk) **11**

[**M2**](#_e53g032x4151) **17**

[**M3**](#_6uyqy1k4wel4) **27**

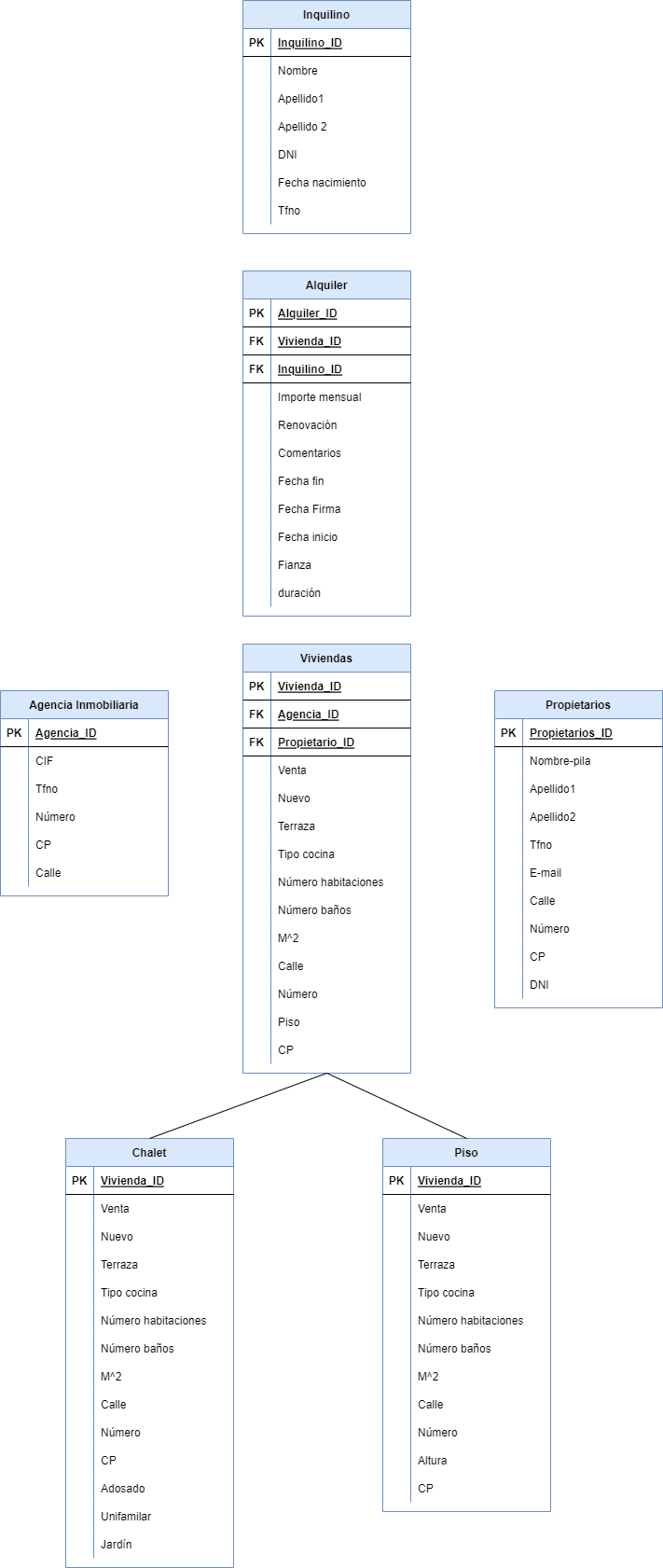
[**Bibliografía**](#_evyx7cr2it1t) **31**

# P2

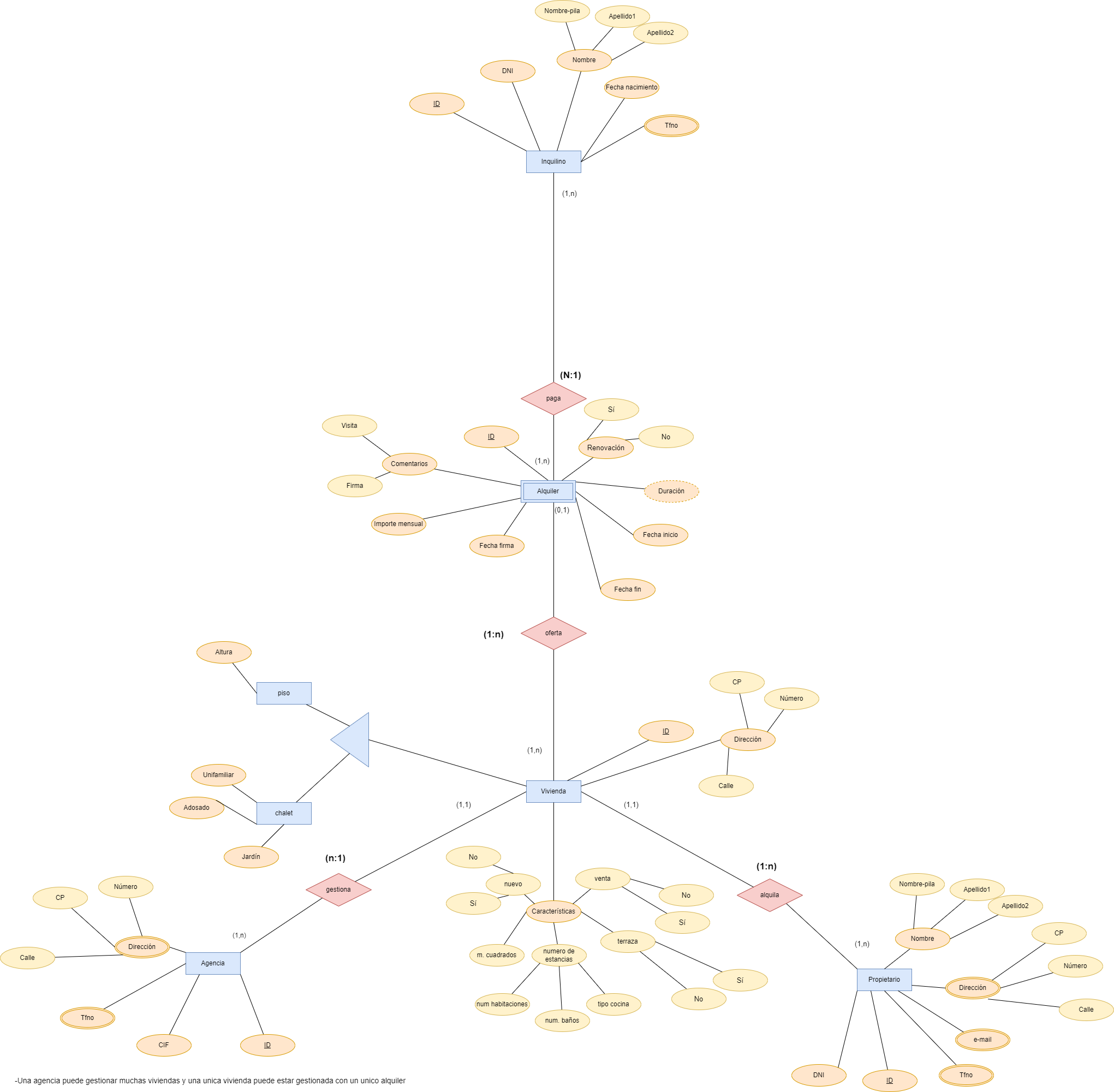
**Desarrollar un sistema de bases de datos concreto que incluya interfaz de usuario, salida de datos y validaciones de datos y realizar consultas en dicho sistema que incluya consultas a varias tablas.**

1. Incluir el Modelo Relacional final, que describa las necesidades expuestas en el Documento de Requerimientos dado y en el Diagrama Entidad Relación ya generado en el P1.

-Modelo Relacional:

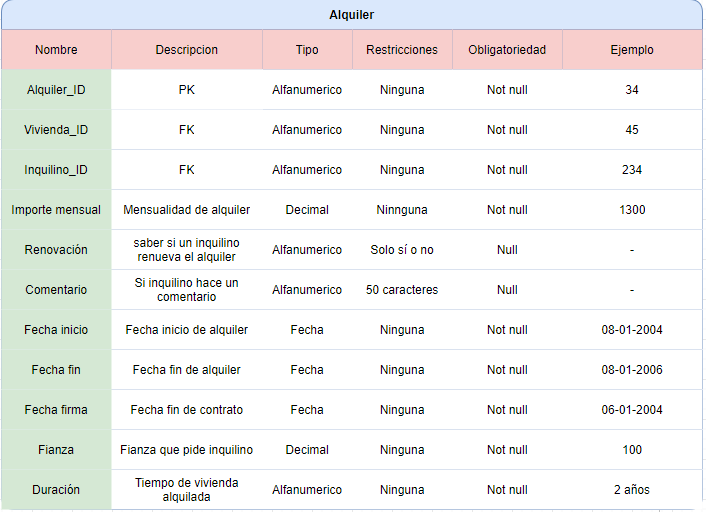


-Modelo Entidad Relación:



2. Breve explicación de los datos de cada entidad, generando un Modelo de Especificación de Datos con el nombre de atributo, una descripción, tipo de dato, restricciones, obligatoriedad y ejemplo (Se recomienda en formato tabla, una por cada entidad).















3. En el diseño del formulario de entrada propuesto en el M1, ¿qué mejoras podría incluir?. Si no lo realizó proponga ideas genéricas para dicho formulario.

Adjunto de formulario de entrada M1:

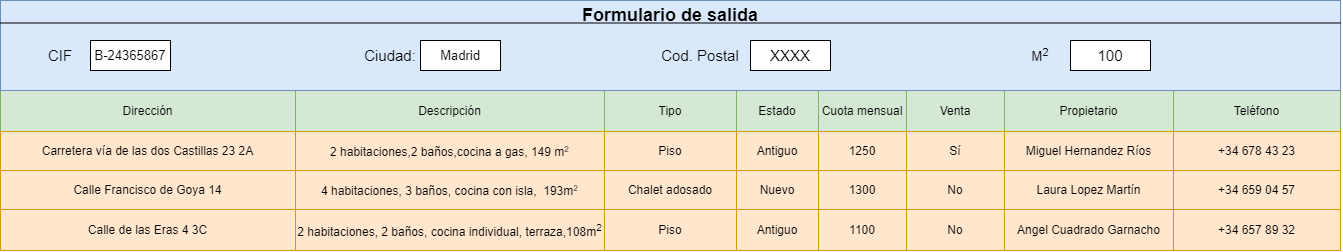


Se podrían añadir más campos que añadan información acerca del inquilino y que aporten a la hora de buscar un alquiler para que la búsqueda del mismo sea más eficiente como por ejemplo, un campo donde introducir la dirección de correo electrónico para que, por ejemplo, se le mande información actualizada de la búsqueda de viviendas para alquilar. Además , sería interesante saber la dirección de la vivienda actual en la que resides ya sea para hacer un estudio de mercado por zonas o para el envío de información por vía postal.

Por último, sería interesante añadir un campo más opcional donde el usuario pueda introducir algunos datos extras que se puedan tener en cuenta para el alquiler, por ejemplo, especificar si tiene alguna imposibilidad física o si es dueño de alguna mascota.

4. En el diseño de la interfaz de salida de datos propuesto en el M1, explique la información que se ofrece y cómo se tendría que obtener (qué tablas y qué campos deberían accederse, y cómo se conectaría esa información). Si no realizó esa interfaz, lea la consulta que se pedía resolver en dicho M1 y responda igualmente.

Adjunto formulario de salida M1:



Se deberían acceder al atributo cif de la tabla agencia, la ciudad y el CP se accede por medio de la dirección de la entidad Vivienda y los metros cuadrados al propio atributo de la misma tabla.

La dirección concreta de la vivienda se accede por los atributos de la calle , ´CP y número de la calle. La descripción se accede gracias a los atributos: número de habitaciones, número de baños y metros cuadrados, el tipo se accede dependiendo de que si los datos de la vivienda se han introducido en uno de los dos subtipos de la super entidad vivienda(piso o chalet). El estado hace referencia al atributo booleano nuevo, la cuota mensual del atributo con el mismo nombre proveniente de la tabla alquiler, la venta se accede por el mismo medio que el estado, por el atributo booleano venta , el nombre del propietario y el teléfono se accede por los campos nombre, ape1, ape2 y Tfno de la tabla propietario.

5. Proponer, de forma teórica y textual, al menos 5 consultas/informes a una tabla y 5 consultas/informes a varias tablas (del modelo relacional), que nuestro futuro sistema debiera poder responder (necesidades de usuario/cliente final no resueltas en las consultas que se proponen en los apartados siguientes).

1.Listado de propietarios con su nombre, apellidos y el número total de viviendas que poseen

2 Total de alquileres realizados por cada agencia inmobiliaria.

3.Informe sobre el promedio de importe mensual de los alquileres de viviendas en una determinada población.

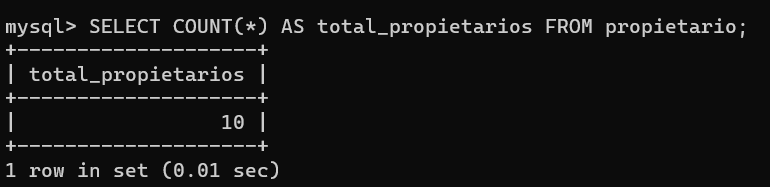
4.Listado de viviendas con un determinado número de habitaciones, ordenadas por metros cuadrados.

5.Informe sobre el promedio de edad de los inquilinos en una determinada población.

# P3

**Implementar consultas en el sistema de bases de datos relacional.**

**1. ¿Cuántos Propietarios tengo en la empresa?**

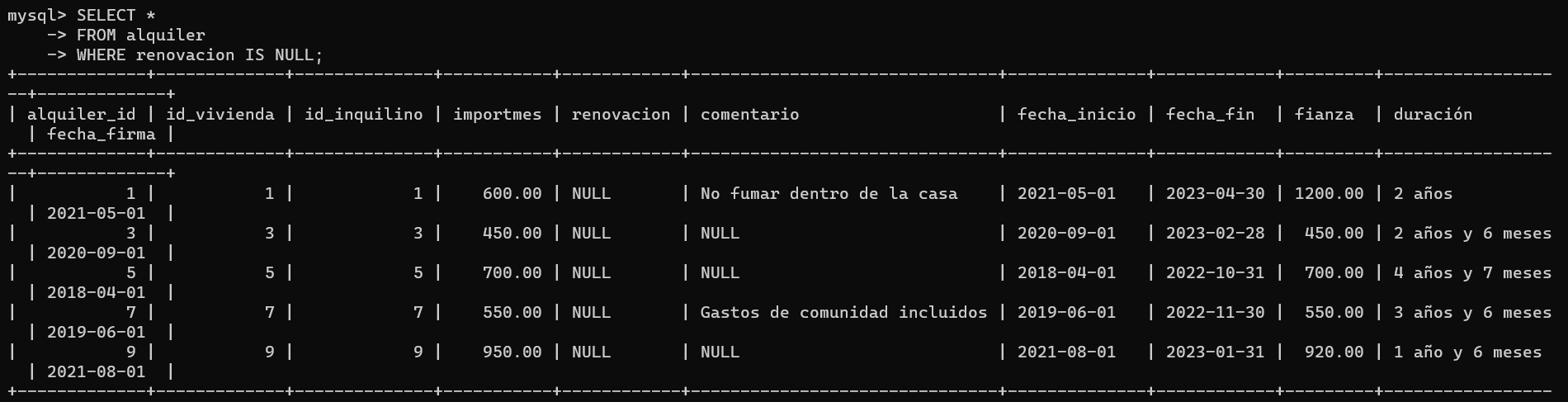
****

La consulta cuenta todas las filas de la tabla propietario y las muestra en una columna llamada total\_propietarios.

**2. Queremos un listado de los inquilinos, incluyendo su nombre, sus apellidos y su teléfono de contacto (si lo tuviera), ordenado por apellido.**

Esta consulta selecciona los campos de interés y los ordena por apellido (ape1). Además, incluye el teléfono de contacto en caso de que esté disponible.

**3. Listado de alquileres que no sean renovaciones.**



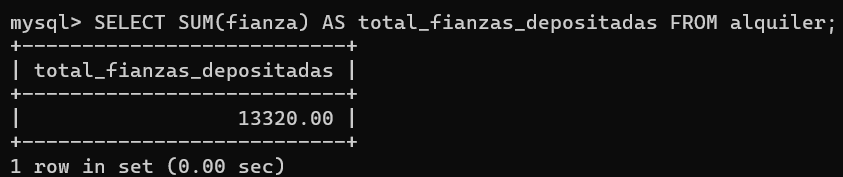
Con select \* selecciona todos los campos de la tabla alquiler con from alquiler, donde el campo renovación es null.

**4. Informe sobre los alquileres, ordenados por su fecha de inicio**

****

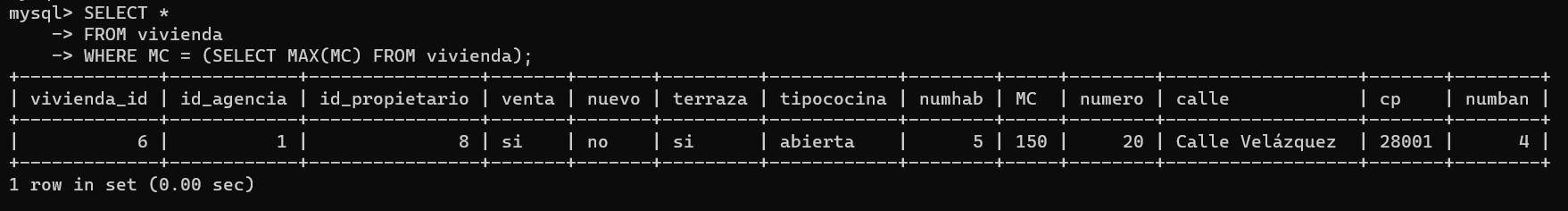
La consulta selecciona todos los alquileres ordenados de forma ascendente (ASC).

**5. ¿Cuánto dinero se supone que tiene que haber por las fianzas depositadas de los alquileres que llevamos registrados hasta el momento?**



La consulta suma (SUM) todas las fianzas, con (AS) se le da un nombre a la consulta de la tabla (FROM) alquiler.

**6. ¿Qué características tienen las viviendas de mayor metros cuadrados? (La vivienda más grande y sus iguales en metros totales)**

****

Esta consulta nos devolverá todas las viviendas que tengan el valor máximo de metros cuadrados (MC), es decir, la vivienda más grande y sus iguales en metros totales.

**7.Listado de agencias con todos sus datos. ¿en qué orden deberían presentarse y por qué?**

Se podrían presentar en diversos órdenes según, por ejemplo, el orden alfabético del cif , la cantidad de vivienda que tiene cada agencia o por el orden de registro de agencias en la base de datos.

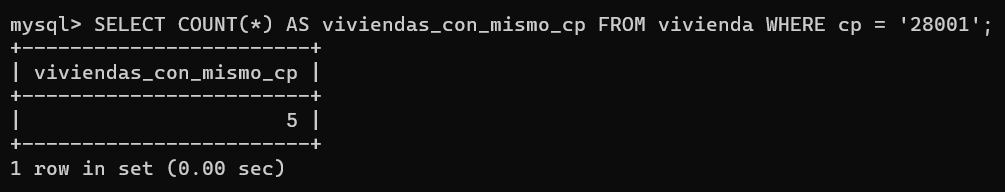
Esta consulta ordena según la cantidad de viviendas que tiene cada agencia de forma ascendente, siendo la más representativa.



En esta consulta, se seleccionan todos los datos de la tabla agencia, se cuenta el número total de viviendas asociadas a cada agencia con el JOIN. Después del JOIN, se relacionan la clave primaria de agencia a la izquierda del igual con su clave foránea en la tabla vivienda y se ordena la columna creada con ORDER BY y en forma ascendente con ASC.

**8. Dado un determinado código postal, ¿cuántas viviendas hay en alquiler**

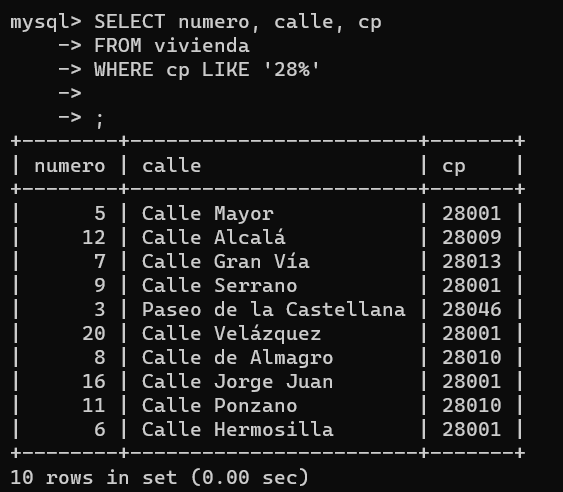
**en él?**



La consulta cuenta todos los registros de la tabla vivienda que tiene el código postal 28001

**9. ¿Cuál es la dirección de todos las viviendas de Barcelona (o cualquier otra provincia dada de alta)?**

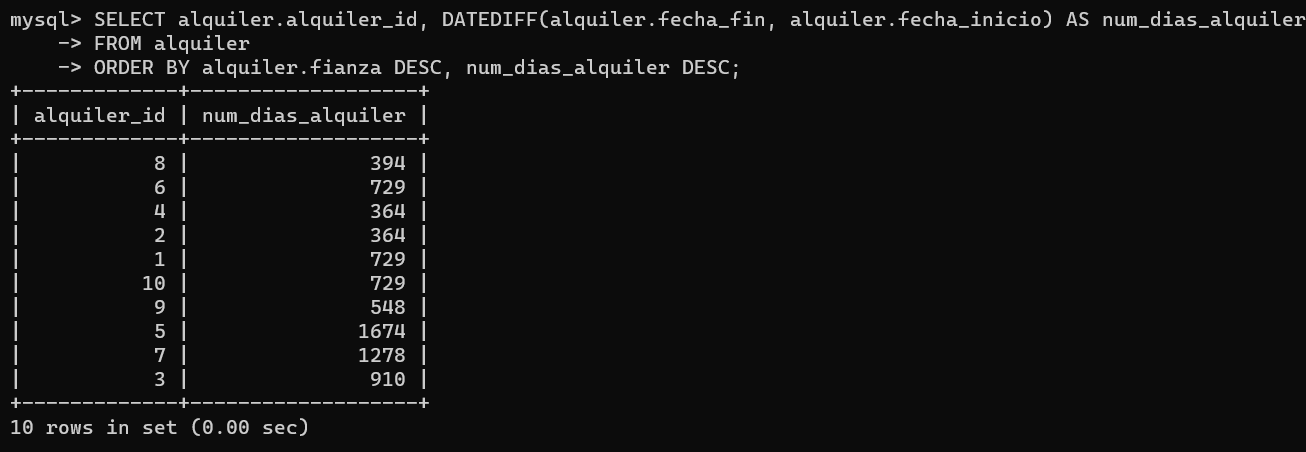
La provincia dada de alta es Madrid

****

La consulta selecciona el número, la calle y el cp de todas las viviendas cuyo cp empiece en 28 gracias al comodín %

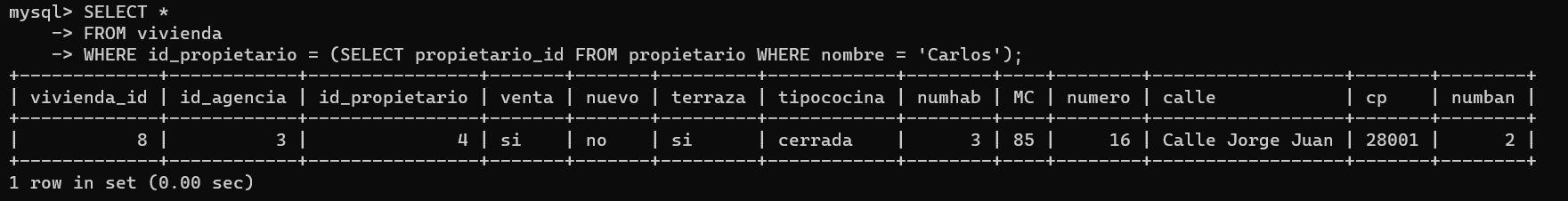
**10. Ordenar los alquileres en orden descendente por la fianza recibida, e**

**indicando el número de días de alquiler.**

****

La consulta selecciona el alquiler\_id y los ordena de manera descendente según su fianza. Aparte, con DATEDIFF hace la diferencia de la ficha de fin de alquiler con la de comienzo y la muestra en días.

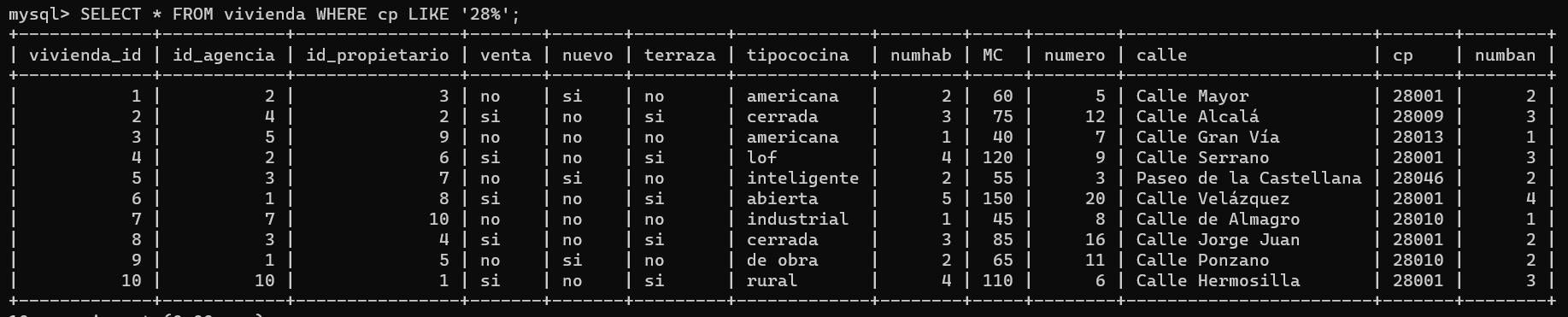
**11. Dado el nombre de un propietario en concreto, ¿qué viviendas proporciona para alquiler?**

****

Selecciona todo de la tabla vivienda que comparta la clave foránea id\_propietario con su primaria en la tabla propietarios donde el nombre es “Carlos”.

**12. Listados de códigos de las viviendas de una determinada población**

La única provincia dada de alta en la base de datos es Madrid, nos podemos dar cuenta de esto gracias a esta consulta:



La consulta devuelve las viviendas cuyo cp empiece en 28 gracias al comodín %

# M2

**Implementar, y explicar, un sistema de bases de datos relacional completo que incluya medidas de seguridad de la base de datos y mantenimiento de la base de datos.**

1. Modelo Físico (código y explicación de las restricciones o tipos de datos utilizados) que represente todo lo anteriormente expuesto en el apartado P1 y en los dos primeros puntos del apartado P2. Nos debemos asegurar que se incluyan medidas de seguridad básica de la base de datos (en la entrada de datos) y mantenimiento de la base de datos (con integridad referencial; y justificación de borrados y

actualizaciones de referencias)

**Leyenda de datos utilizados:**

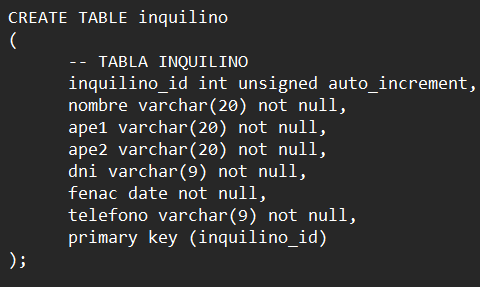
**-Varchar:** Se trata de un dato que admiten letras,números y caracteres especiales. A continuación se le pone la restricción, que se indica entre paréntesis. Dentro del paréntesis pondremos el número máximo de caracteres que puede haber en ese dato.

-**Int:**El dato int solo admite números enteros, después de el dato se le puede añadir restricciones como unsigned, que especifica que solo se aceptarán números enteros positivos.

**-Date:**Sirve para introducir fechas en la base de dato, este dato es muy útil ya que se puede operar con él de diversas formas.Es importante saber que para introducir una fecha debe ser en formato americano, es decir, AÑO-MES-DIA.

-**Decimal:**Similar al int ya que solo acepta números pero este tipo de dato se pueden introducir decimales.Se le pueden añadir las mismas restricciones que a los datos de tipo int. La estructura para declarar los datos decimales es la siguiente. Decimal(precisión,escala). Precisión representa el número total de dígitos que pueden almacenarse y escala el número de decimales

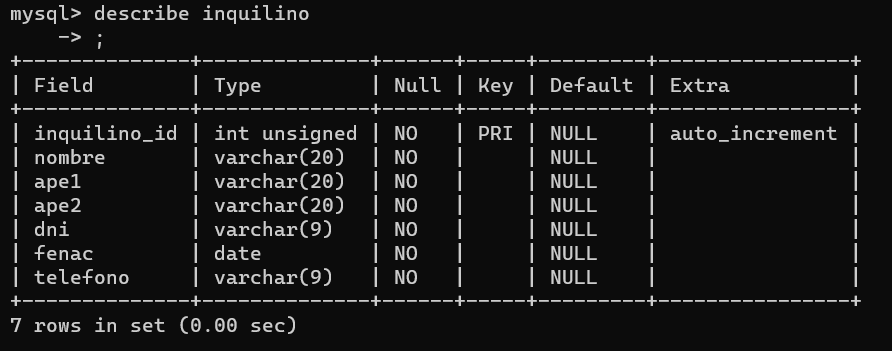
**Inquilino:**

****

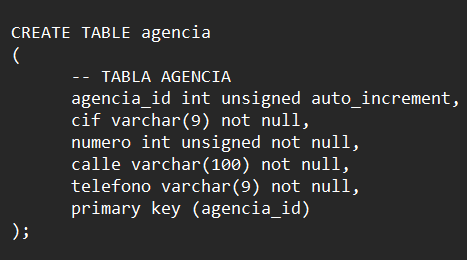
Este es el código utilizado para crear la tabla inquilino. En primer lugar declaramos el dato inquilino\_id , después declaramos el tipo de dato (int) con el extra auto\_increment. Este extra hace que, a la hora de hacer los inserts no tengas que introducir el id, si no que a cada insert que haces se suma uno, lo que agiliza un poco el trabajo.Con la última linea de codigo (primary key (inquilino\_id)) hace que el programa referencie este dato como clave primaria de la tabla. Esta estructura será la misma para todas las claves primarias de todas las tablas.

Después, declaramos el nombre, el primer apellido (ape1) y el segundo apellido (ape2) del inquilino con un dato tipo varchar, los tres con una restricción los tres de 20 caracteres. Después el dni que también es un varchar de 9 caracteres, igual que los DNI en España. En fenac guardamos la fecha de nacimiento del inquilino, por lo que es un dato de tipo date y en teléfono guardaremos el número de teléfono del inquilino. Después añadimos a los datos not null, que sirve para indicar que a la hora de hacer un insert, el dato que tenga not null tiene la obligación de figurar.

Este código implementado en modelo físico se vería así:

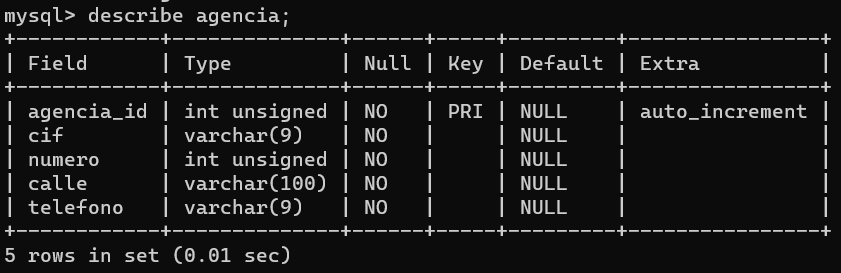


**Agencia inmobiliaria:**

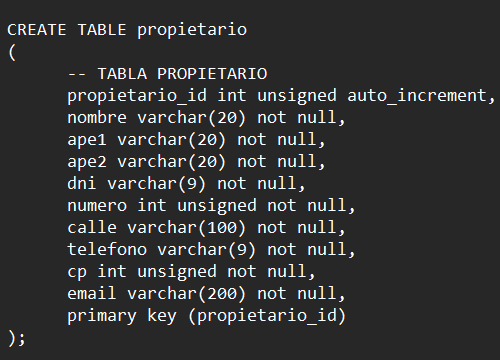


Repetimos la misma estructura en cuanto a la clave primaria (agencia\_id).

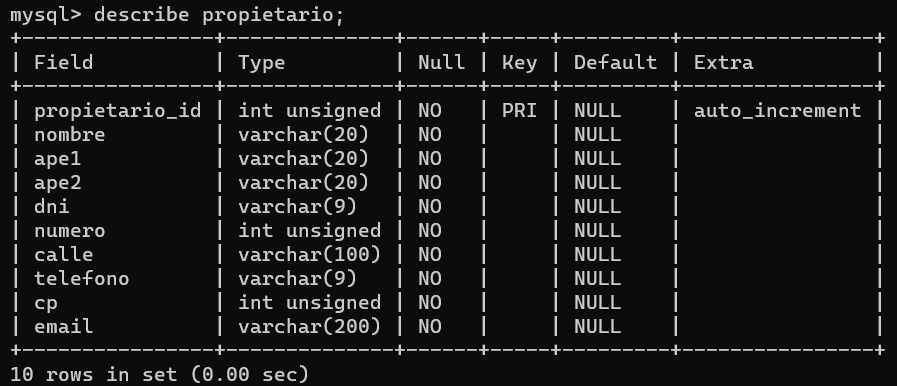
Introducimos el dato cif que es un varchar con una restricción de 9 dígitos. El número hace referencia a la altura de la calle, que la declaramos como un varchar que puede llegar a ser muy largo con 100 caracteres y por último el teléfono.Por último, se le añade a todos los datos la obligatoriedad not null y es representado de esta manera:



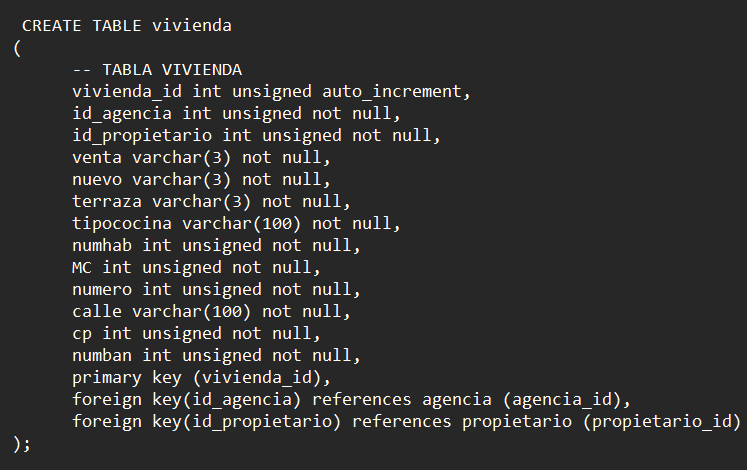
**Propietario:**



La clave primaria sigue la misma estructura que las anteriores. Nombre, ape1,ape2,dni número calle y teléfono son exactamente iguales que los de la tabla inquilino. Se le añaden los datos cp que referencia el código postal de la dirección. (La dirección es el dato compuesto formado por número, calle y cp). Y el email es otro varchar que puede llegar a ser largo por lo que se le añade una restricción muy larga. De nuevo, se le añade a todos los datos la obligatoriedad not null. Esa tabla implementada en modelo físico se vería de esta forma:



**-Vivienda:**



La clave primaria sigue la misma estructura que las tablas anteriores lo datos de direccion(numero,calle y cp) son los mismos que en la tabla de propietario.Existen datos como venta , nuevo y terraza que son de tipo varchar con una restricción de tres caracteres, tiene tan poco caracteres para indicar solo si existe o no con un SÍ o NO.

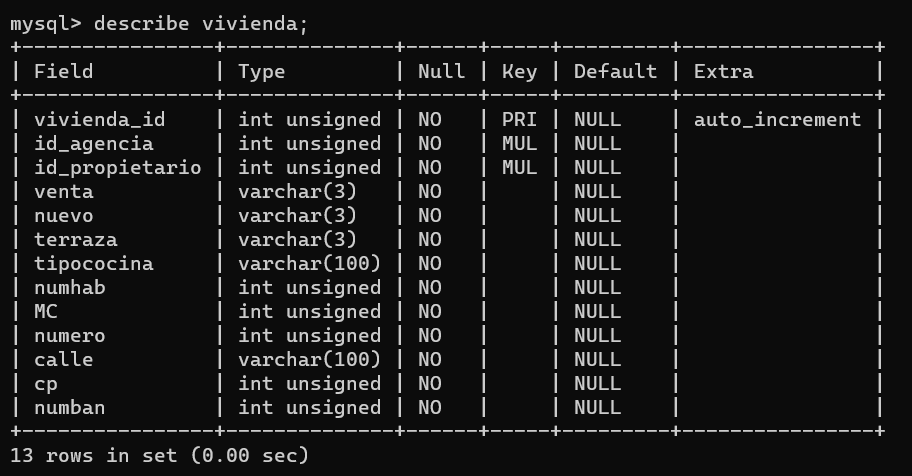
Luego también existen otros como tipococina que es otro varchar que sirve para indicar de manera general el tipo de cocina que tiene la vivienda, num hab y numban que indican cuántas habitaciones y cuántos baños tiene la vivienda respectivamente, por lo que son datos de tipo int con la restricción unsigned y por último MC, que es del mismo tipo que los dos anteriores, en el que se indican cuántos metros cuadrados tiene, a todos los datos se le añade la obligatoriedad not null.

Por último, hay que destacar Id\_agencia y id\_propietario de tipo int unsigned, que sirven para referenciar el registro de una vivienda con un registro de una agencia y con un propietario. Esta referencia se consigue con las ultimas dos lineas:

foreign key(id\_agencia) references agencia(agencia\_id)

foreign key(id\_propietario) references propietario(propietario\_id)

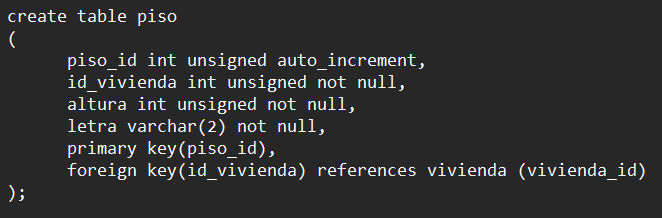
Se declaran las que son foráneas en la primera parte, después se hace referencia a la tabla y entre paréntesis la clave primaria de dicha tabla. Implementada en modelo físico se vería así:



La entidad vivienda tiene dos subtipos, piso y chalet.Las función de estas dos tablas es complementar a la de vivienda, dependiendo de que sean piso o chalet, se le añaden los datos para complementar a la tabla vivienda.

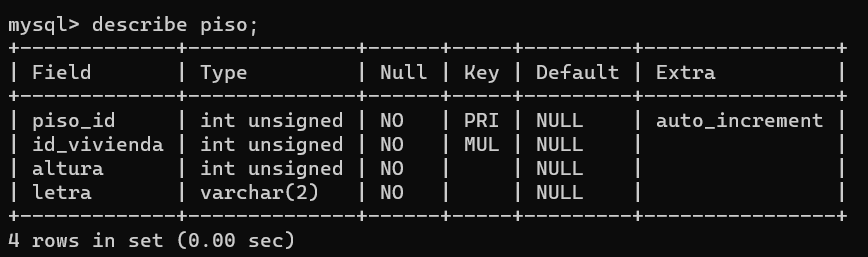
Las cuales se representan así:

**-Piso:**

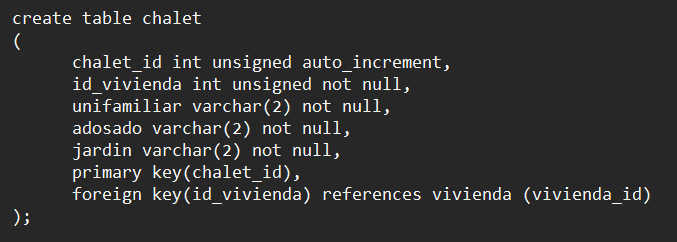
****

La clave primaria primaria piso\_id sigue la misma estructura que las anteriores y también tiene la clave foránea id\_vivienda que hace referencia a la clave primaria vivienda\_id de la tabla vivienda.

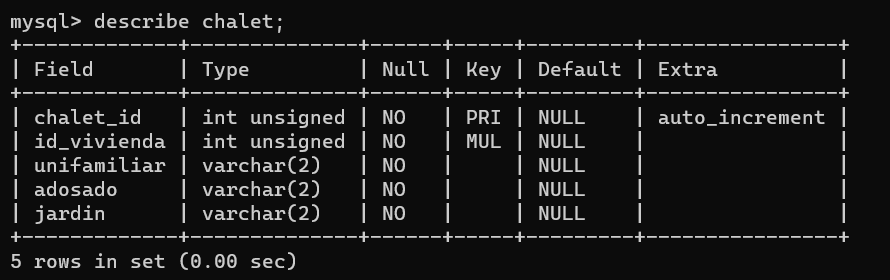
La altura indica a qué altura está el piso dentro de un edificio, por lo que es un dato de tipo int unsigned y la la letra es de tipo varchar para especificar cual es el piso dentro de la altura de un edificio. Implementada en modelo físico:



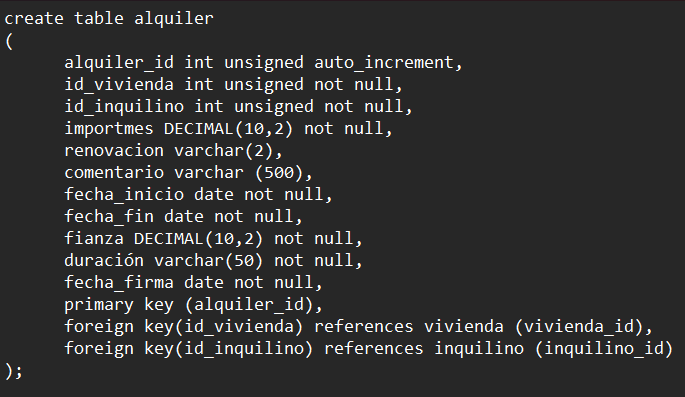
**-Chalet:**



En cuanto a la clave primaria chalet\_id sigue la misma estructura que el resto y la foránea id\_vivienda es exactamente igual que la de la tabla pisó.Los datos unifamiliar , adosado y jardín sirven únicamente para indicar si tienen o no jardín, por lo que son datos de tipo varchar de 2 caracteres para poner SI o NO.



**-Alquiler:**

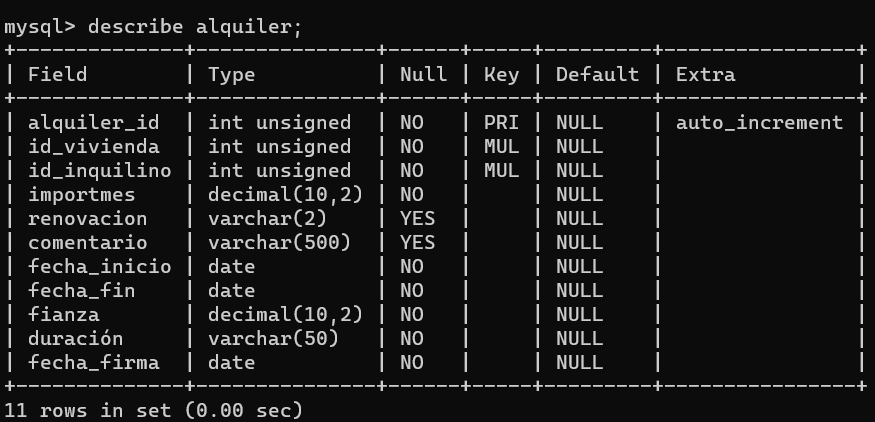
****

La estructura de la clave primaria alquiler\_id es la misma que las demás. Además, las claves foráneas id\_vivienda y id\_inquilino hacen referencia a las claves primarias de las tablas vivienda e inquilino respectivamente.

Esta tabla guarda los datos de cada alquiler, por lo que tenemos un dato que es importmes, que es el importe mensual del alquiler de una vivienda. Como se representa en euros lo declaramos en tipo decimal con la restricción (10,2) que indica que como máximo pueden haber diez dígitos con dos decimales. Este es el mismo caso que el dato fianza.

Fecha\_inicio indica el día que se empezó a alquiler la vivienda, fecha\_fin el día que se terminó el alquiler y fecha\_firma el día que se firmó el contrato de alquiler. Ya que son fechas se indica el dato de tipo date. Relacionados con estos últimos, encontramos el dato duración, que es de tipo varchar y recoge la diferencia entre la fecha\_inicio y fecha\_fin. Es de 50 caracteres para especificar esta diferencia en años,meses y días.

Por último, están los datos renovación y comentarios, que son los únicos que no llevan la obligatoriedad not null. La renovación puede o no darse, por lo que, si se da la renovación de alquiler se indicará un SÍ, ya que un varchar de sólo 2 caracteres y en el caso que no haya se registrará con NULL. Según el modelo de ejercicio podrán o no haber comentario, por lo que sí hay, se escribirá ya que es un varchar con una restricción de 500 caracteres, más que necesarios y en el caso de que no haya se indicará con NULL, igual que en el dato renovación.



2. Proponga diferentes perfiles de usuario (mínimo tres) que podrían acceder al sistema, e investigue y resuelva cómo limitar su acceso a la información, con la gestión de permisos en MySQL.

Dentro del modelo de esta base de datos, se podrían proponer diversos perfiles, estos son algunos:

-Inquilinos: este usuario solo tendría acceso para leer y escribir solo en su registro dentro de la tabla inquilinos, podrían escribir algunos campos dentro de la tabla alquiler, leer la tabla alquiler completa, vivienda y sus subtipos. No tendría acceso a la tabla agencia ni a la tabla propietario.

-Agencia Inmobiliaria: tendrá acceso a la lectura y escritura únicamente de su registro. Pudiendo leer y escribir en la tabla vivienda, propietario y alquiler.

-Administrador (DBA): este usuario tendría un acceso y permisos totales a toda la base de datos.

La forma de limitar accesos a los usuarios es la siguiente:

1ºDebemos crear un usuario para luego poder otorgarlos o restringir permisos. Esto se hace con el siguiente comando:

CREATE USER 'nombre\_usuario'@'localhost' IDENTIFIED BY 'contraseña';

Es importante ingresar a MYSql con privilegios de administrador en el servidor. “nombre\_usuario” lo cambiariamos por el nombre de usuario que queremos y la contraseña igual.

2ºUna vez tenemos los usuarios creados, con este comando podremos cambiar los privilegios de cada usuario a voluntad:

GRANT SELECT, INSERT, UPDATE ON vivienda, propietario TO 'propietario1'@'localhost';

Con este ejemplo, lo que estamos haciendo es otorgar permisos al propietario1 para poder hacer querings, insertar datos nuevos en la tabla y actualizar los ya existentes en las tablas vivienda y propietario.

Para otros casos, solo hace falta poner o quitar los permisos y tablas a los que queremos dar acceso e indicar el usuario. Es importante colocar siempre la palabra GRANT al principio.

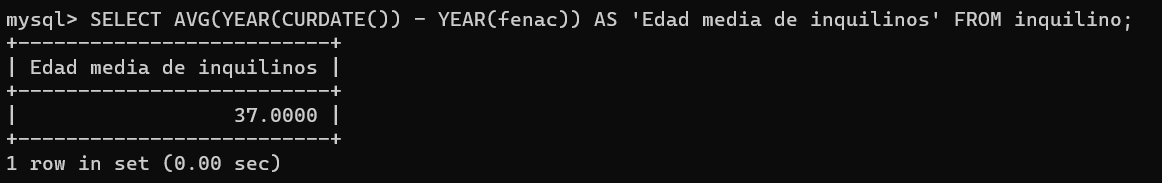
# 

# M3

**Asesorar sobre la extracción de datos importantes de la base de datos mediante el uso de consultas para producir información de gestión apropiada.**

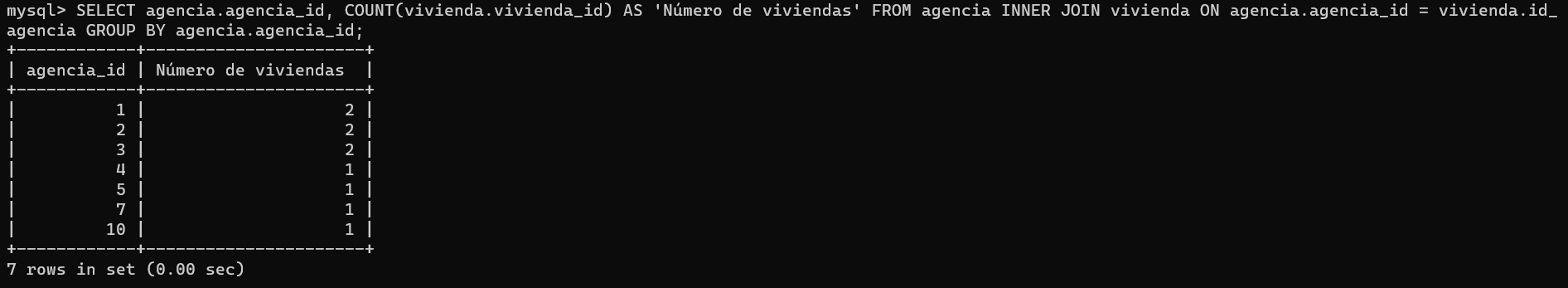
1. **¿Cuántos años tienen de media nuestros inquilinos?**

Esta consulta permitirá conocer la edad promedio de nuestros inquilinos y nos puede ayudar a entender mejor su perfil y necesidades, lo que nos permitiría tomar decisiones sobre cómo mejorar nuestros servicios para satisfacer mejor sus necesidades.



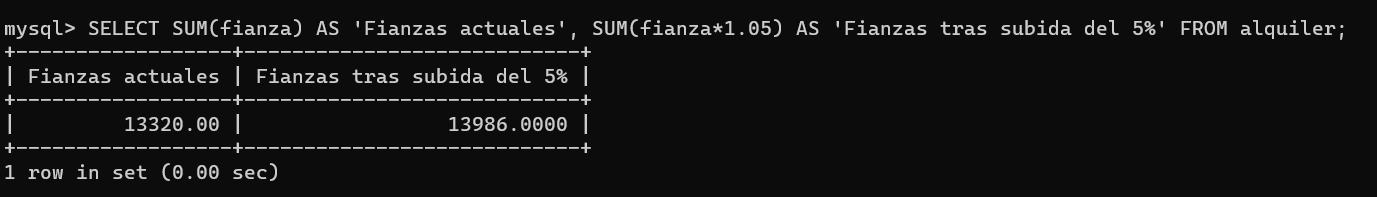
**2.¿Cuántas viviendas hay de cada una de las agencias registradas?**

Esta consulta permitirá conocer cuántas viviendas tenemos registradas para cada una de las agencias y nos permitiría tomar decisiones sobre cómo distribuir mejor nuestros recursos y esfuerzos de marketing.



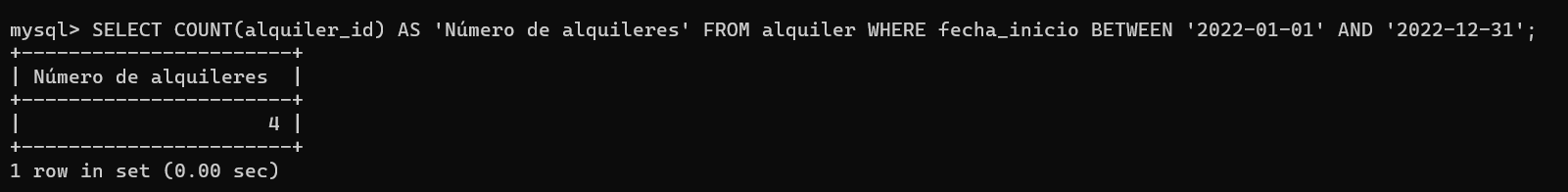
**3. ¿Cómo quedarían las fianzas de los alquileres que tenemos si procederemos a subirlas un 5%? No ejecutar el cambio, sólo mostrar cómo es ahora y cómo sería tras la subida.**

Esta consulta permitirá conocer cómo afectaría una posible subida de las fianzas de alquileres en nuestra cartera de alquileres y nos permitiría tomar decisiones sobre si es una medida viable y cómo llevarla a cabo.



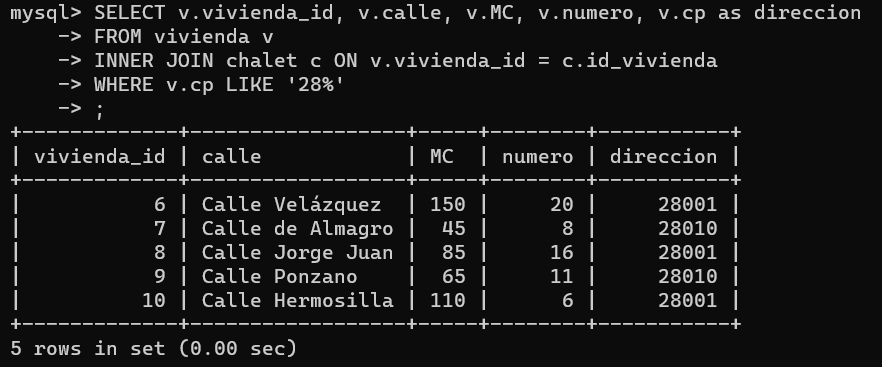
**4. Número de alquileres que se inician entre dos fechas concretas.**

Esta consulta permitirá conocer cuántos alquileres se iniciaron en un periodo de tiempo determinado, lo que nos permitiría tener una idea de la demanda en determinados momentos del año y nos ayudaría a planificar mejor nuestros esfuerzos de marketing y gestión de recursos.



**5. Dirección y metros cuadrados de los chalets de una población concreta.**

Esta consulta permitirá conocer la información básica de los chalets en una población concreta, lo que nos permitiría tomar decisiones sobre cómo enfocar nuestros esfuerzos de marketing y gestión de recursos para este tipo de propiedades.



**6. ¿Cómo podríamos comprobar el nombre y dirección de aquellos inquilinos de los que no tenemos teléfono?**

Esta consulta nos permitiría encontrar a los inquilinos que no tienen número de teléfono registrado, lo que nos permitiría tomar decisiones sobre cómo mejorar la calidad de nuestros datos y cómo contactar con estos inquilinos para fines de marketing y gestión de recursos.



El resultado muestra que no existe ningún inquilino que no tenga teléfono registrado.

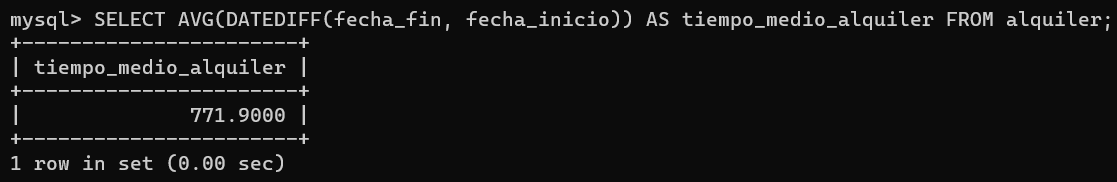
**7. ¿Cuál es la información que tenemos sobre el alquiler más duradero?**

Esta consulta nos permitiría conocer cuál es el alquiler que ha durado más tiempo, lo que nos permitiría tomar decisiones sobre cómo mejorar la calidad de nuestros servicios y cómo atender mejor las necesidades de nuestros clientes más leales.



**8. ¿Cuánto tiempo medio duran los alquileres?**

Esta consulta permite estimar la cantidad de tiempo medio de los alquileres, lo que suele ser útil en cuanto a planificación y presupuestos.



**9. Cuál es la población, de las que tenemos registradas, con menor número de viviendas (destinadas a alquiler)?**

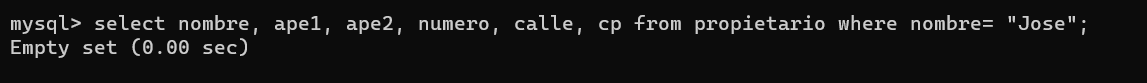
Esta consulta es útil para el departamento de marketing y ventas, ya que permite identificar cuáles son las poblaciones en las que se puede trabajar más para aumentar la oferta de viviendas en alquiler.

Ya que de momento solo tenemos registradas viviendas en la población de Madrid, esta consulta no sería totalmente solvente al problema.

**10. ¿A cuántos de nuestros propietarios deberíamos mandar un mensaje de felicitación por su santo el próximo 19 de marzo? Generar además un listado con su nombre completo y dirección.**

Esta consulta es útil para el departamento de atención al cliente, ya que permite identificar a los propietarios a los que se les puede enviar un mensaje personalizado, lo que puede mejorar la relación entre el cliente y la empresa.

Dado que el 19 de Marzo es el dia de San José, buscamos en la tabla de propietarios a lo que se llaman Jose pero no se devuelve ninguno porque no hay ningún registro con ese nombre:



# Bibliografía

<https://www.w3schools.com/sql/>

<http://www.sqlcourse.com/>

<https://sqlzoo.net/>

<https://www.tutorialspoint.com/sql/index.htm>

<https://mode.com/sql-tutorial/>

<https://dev.mysql.com/doc/mysql-tutorial-excerpt/5.7/en/>

<https://www.postgresqltutorial.com/>

<https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

<https://www.sqlitetutorial.net/>