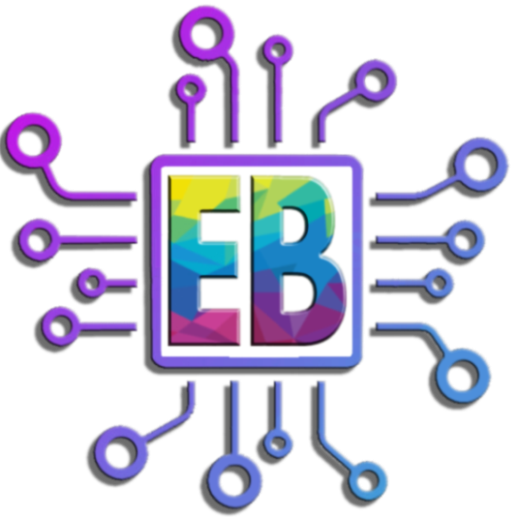
**Tercera Entrega Sistemas de Base de Datos II**

**Solicitante:**

**I.T.S. – Instituto Tecnológico Superior Arias - Balparda**

**Nombre de Fantasía del Proyecto: Eureka Bit  
Grupo de Clase: 3° ID  
Turno: Vespertino**

**Materia: Sistemas de Base de Datos II**

**Nombre de los Integrantes del Grupo: Luciana De Matteis (representante), Alberto Pérez, Ignacio Roland, e Irina Sena**

**Fecha de entrega: 25/10/2019**

Instituto Tecnológico Superior F. Arias – L. Balparda  
***Gral. Flores 3591 esq. Bvar. José Batlle y Ordoñez - Montevideo***

**Objetivo**

El objetivo de la realización de esta última entrega para la asignatura Sistemas de Bases de Datos II es estudiar qué permisos le pertenecen a cada usuario según su rol en el sistema e implementarlos en la Base de datos, también elegir una forma de respaldar los datos de los usuarios e igualmente aplicar ese o esos métodos a nuestra la BD.

**Alcance**

Esta entrega es importante para asegurarnos de que los usuario no podrán ver o borrar información de la Base de datos a menos que tenga los permisos correspondientes para realizar esas acciones y para asegurarnos que toda la información de nuestro cliente este correctamente respaldada ante cualquier problema o desperfecto.

Índice

**1. Modelo conceptual- MER 4**

1.1 RNE 4

**2. Esquema Relacional (E.R) 5**

2.1 Pasaje a tablas 5

2.2 Normalización (3FN) 6

**3. Justificacion de cambios en MER 7**

**4. Diccionario de datos 8**

**5. Sentencias DDL y DML 12**

**6. Usuarios 16**

**7. Permisos 17**

7.1 Permisos de los diferentes roles 17

7.2 Sentencias utilizadas para la asignacion de permisos 17

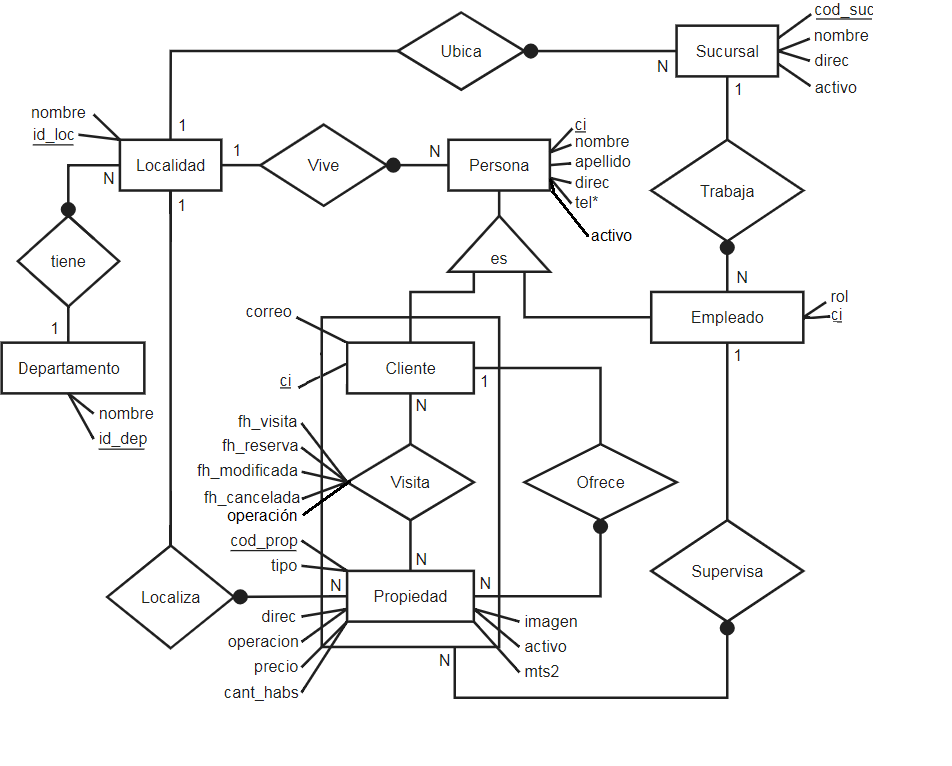
**8. Respaldo de datos 19**

**9. Recuperación de datos 20**

**10. Consultas 21**

**Hoja Testigo 23**

**1. Modelo conceptual- MER:**



**1.1 RNE:**

* Propiedad.tipo = {“casa”, “apartamento”, “local comercial”}
* Propiedad.operación = {“alquiler, “venta”, “ambas”}
* Visita.operación = {“alquiler", “venta”}
* Visita.fech\_hor\_reserva < Visita.fech\_hor\_visita
* Empleado.Rol = {"agente", "gerente"}
* Cliente.correo es único
* Teléfono.número es único

**2. Esquema Relacional (E.R):**

**2.1 Pasaje a tablas:**

**Sucursal** (CodSuc, NomSuc, Direc, Activo, Id\_Loc)

Id\_Loc -> Localidad

**Persona** (CI, Nom, Ape, Direc, Tel\*, Id\_Loc, Activo)

Id\_Loc -> Localidad

**Empleado** (CIEmp, Rol, CodSuc , CodProp)

CIEmp -> Persona CodSuc - > Sucursal

CodProp-> Propiedad

**Cliente** (CICli, Correo, CodProp)

CIInt -> Persona CodProp-> Propiedad

**Propiedad** (CodProp, Tipo, Direc, Operación, Precio, Cant.Habit,, Imagen , Activo, CICli, Id\_Loc)

CICli -> Cliente Id\_Loc -> Localidad

**Visita** (CICli, CodProp, FechVisita, FechReserva, FechCancelada, FechModificada, Operación, CIEmp)

CICli -> Cliente CodProp -> Propiedad CIEmp-> Empleado

**Localidad** (Id\_Loc, Nombre, Id\_Dep)

Id\_Dep -> Departamento

**2.2 Normalización (3FN):**

Se eliminan compuestos y/o multivaluados. Todos los atributos no primos dependen de la clave primaria, a su vez ningún atributo no primo depende de otro atributo no primo.

**Sucursal** (CodSuc, NomSuc, Direc, Activo, Id\_Loc)

Id\_Loc -> Localidad

**Persona** (CI, Nom, Ape, Direc, Id\_Loc, Activo)

Id\_Loc -> Localidad

**Tel\_Persona** (CI, Numero)

CI -> Persona

**Empleado** (CIEmp, Rol, CodSuc , CodProp)

CIEmp -> Persona CodSuc - > Sucursal

CodProp-> Propiedad

**Cliente** (CICli, Correo, CodProp)

CIInt -> Persona CodProp-> Propiedad

**Propiedad** (CodProp, Tipo, Direc, Operación, Precio, Cant.Habit,, Imagen , Activo, CICli, Id\_Loc)

CICli -> Cliente Id\_Loc -> Localidad

**Visita** (CICli, CodProp, FechVisita, FechReserva, FechCancelada, FechModificada, Operación, CIEmp)

CICli -> Cliente CodProp -> Propiedad CIEmp-> Empleado

**Localidad** (Id\_Loc, Nombre, Id\_Dep)

Id\_Dep -> Departamento

**3. Justificación de cambios en el MER:**

Hemos añadido dos tablas (Departamento y Localidad), su razón es por inconvenientes que se presentaron desarrollando el programa para la asignatura Programación, para que los usuarios no ingresen datos, sino que se seleccione de la lista ya definida, ya cargada en la base de datos. De esta forma evitar la redundancia de datos, validaciones innecesarias, y sobrecargar el programa con datos que no le corresponde abarcar.

Por ejemplo, un usuario ingresa una localidad con mayúscula y otro se refiere a la misma pero en minúscula, uno ingresa con tilde y otro no, o con faltas ortográficas y sin. Y esto produce que los datos se repitan, entorpece las búsquedas a la hora de hacer consultas, o brinda datos erróneos.

**4. Diccionario de datos:**

|  |
| --- |
| Tabla Departamento |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| id\_dep | tinyint | - | Refiere al código único propio de cada departamento. |
| nombre | varchar | 50 | Refiere al nombre propio de cada departamento. |

|  |
| --- |
| Tabla Localidad |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| id\_loc | smallint | - | Refiere al código único propio de cada localidad |
| nombre | varchar | 50 | Refiere al nombre propio de cada localidad. |
| id\_dep | tinyint | - | Refiere al código único propio de cada departamento. |

|  |
| --- |
| Tabla Sucursal |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| Cod\_Suc | int | 11 | Refiere al código propio de cada sucursal que lo identifica de forma única. |
| Nombre | varchar | 50 | Refiere al nombre propio de cada sucursal. |
| Direc | varchar | 75 | Dirección en la que esta la sucursal. |
| Id\_Loc | smallint | - | Refiere al código de la localidad en la cual esta. |
| Activo | boolean | - | Indica si esta o no activa. |

|  |
| --- |
| Tabla Persona |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| CI | int | 11 | Refiere a la cedula de identidad de la persona |
| Nombre | varchar | 30 | Refiere al nombre propio de la persona. |
| Apellido | varchar | 30 | Refiere al apellido propio de la persona. |
| Direc | varchar | 75 | Dirección de residencia de la persona |
| id\_loc | smallint | - | Refiere al código de la localidad en la cual vive. |
| Activo | boolean | - | Indica si está o no activo. |

|  |
| --- |
| Tabla Teléfono |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| CI | int | 11 | Refiere a la cedula de identidad de la persona. |
| Numero | varchar | 45 | Refiere al teléfono de la persona. |

|  |
| --- |
| Tabla Cliente |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| CI\_cli | int | 11 | Refiere a la cedula de identidad del cliente. |
| Correo | varchar | 75 | Correo electrónico del cliente. |

|  |
| --- |
| Tabla Empleado |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| CI\_Empleado | int | 11 | Refiere a la cedula de identidad del empleado. |
| CodSucursal | int | 11 | Código de la sucursal en la cual trabaja el empleado. |
| Rol | varchar | 15 | Su cargo en la inmobiliaria, tal puede ser agente o gerente. |

|  |
| --- |
| Tabla Propiedad |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| CodProp | int | 11 | Refiere al código propio de la propiedad. |
| Tipo | varchar | 25 | Categoría en la cual está, tal puede ser apartamento, casa o local comercial. |
| Direc | varchar | 75 | Dirección de residencia de la propiedad |
| Operación | varchar | 25 | Condición en la cual está, tal puede ser venta, alquiler o ambas. |
| Precio | int | 11 | Precio de la propiedad. |
| Cant\_Habs | int | 11 | Cantidad de habitaciones que posee. |
| MTS^2 | int | 11 | Medidas de la propiedad. |
| Imagen | mediumblob | - | Ruta de la foto de la propiedad. |
| id\_loc | smallint | - | Refiere al código de la localidad en la cual vive. |
| Ci\_cli | int | 11 |  |
| activo | boolean | - | Indica si esta o no activa. |

|  |
| --- |
| Tabla Visita |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Descripción |
| CI\_Cliente | int | 11 | Refiere a la cedula de identidad del interesado. |
| CodProp | int | 11 | Refiere al código propio de la propiedad. |
| CI\_Emp | int | 11 | Refiere a la cedula de identidad del empleado. |
| FechVisita | datetime | - | Fecha para la que fue agendada la visita. |
| FechReserva | datetime | - | Fecha en la que fue reservada la visita. |
| FechModificada | datetime | - | Fecha de modificación de visita |
| FechCancelada | datetime | - | Fecha de cancelación de la visita |
| Operación | varchar | 25 | Condición en la cual está, tal puede ser venta, alquiler o ambas. |

**5. Sentencias DDL y DML:**

CREATE DATABASE IF NOT EXISTS `proyecto\_eurekabit`;

USE `proyecto\_eurekabit`;

DROP TABLE IF EXISTS `departamento`;

CREATE TABLE `departamento` (

`id\_dep` tinyint(4) NOT NULL,

`nombre` varchar(50) NOT NULL,

PRIMARY KEY (`id\_dep`),

UNIQUE KEY `nombre` (`nombre`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `localidad`;

CREATE TABLE `localidad` (

`id\_loc` smallint(6) NOT NULL,

`nombre` varchar(50) NOT NULL,

`id\_dep` tinyint(4) NOT NULL,

PRIMARY KEY (`id\_loc`),

UNIQUE KEY `nombre` (`nombre`,`id\_dep`),

KEY `id\_dep` (`id\_dep`),

CONSTRAINT `localidad\_ibfk\_1` FOREIGN KEY (`id\_dep`) REFERENCES `departamento` (`id\_dep`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `sucursal`;

CREATE TABLE `sucursal` (

`cod\_suc` int(11) NOT NULL AUTO\_INCREMENT,

`nombre` varchar(50) NOT NULL,

`direc` varchar(75) NOT NULL,

`id\_loc` smallint(6) NOT NULL,

`activo` tinyint(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`cod\_suc`),

KEY `id\_loc` (`id\_loc`),

CONSTRAINT `sucursal\_ibfk\_1` FOREIGN KEY (`id\_loc`) REFERENCES `localidad` (`id\_loc`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `persona`;

CREATE TABLE `persona` (

`ci` int(11) NOT NULL,

`nombre` varchar(30) NOT NULL,

`apellido` varchar(30) NOT NULL,

`direc` varchar(75) NOT NULL,

`id\_loc` smallint(6) NOT NULL,

`activo` tinyint(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`ci`),

KEY `id\_loc` (`id\_loc`),

CONSTRAINT `persona\_ibfk\_1` FOREIGN KEY (`id\_loc`) REFERENCES `localidad` (`id\_loc`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `telefono`;

CREATE TABLE `telefono` (

`ci` int(11) NOT NULL,

`numero` varchar(45) NOT NULL,

UNIQUE KEY `numero` (`numero`),

KEY `ci` (`ci`),

CONSTRAINT `telefono\_ibfk\_1` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `cliente`;

CREATE TABLE `cliente` (

`ci\_cli` int(11) NOT NULL,

`correo` varchar(75) NOT NULL,

PRIMARY KEY (`ci\_cli`),

UNIQUE KEY `correo` (`correo`),

CONSTRAINT `cliente\_ibfk\_1` FOREIGN KEY (`ci\_cli`) REFERENCES `persona` (`ci`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `empleado`;

CREATE TABLE `empleado` (

`ci\_emp` int(11) NOT NULL,

`cod\_suc` int(11) NOT NULL,

`rol` varchar(15) NOT NULL,

PRIMARY KEY (`ci\_emp`),

KEY `cod\_suc` (`cod\_suc`),

CONSTRAINT `empleado\_ibfk\_1` FOREIGN KEY (`ci\_emp`) REFERENCES `persona` (`ci`) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT `empleado\_ibfk\_2` FOREIGN KEY (`cod\_suc`) REFERENCES `sucursal` (`cod\_suc`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `propiedad`;

CREATE TABLE `propiedad` (

`cod\_prop` int(11) NOT NULL AUTO\_INCREMENT,

`tipo` varchar(25) NOT NULL,

`direc` varchar(75) NOT NULL,

`operacion` varchar(25) NOT NULL,

`precio` int(11) NOT NULL,

`cant\_habs` int(11) NOT NULL,

`mts2` int(11) NOT NULL,

`imagen` mediumblob DEFAULT NULL,

`ci\_cli` int(11) NOT NULL,

`id\_loc` smallint(6) NOT NULL,

`activo` tinyint(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`cod\_prop`),

KEY `id\_loc` (`id\_loc`),

KEY `ci\_cli` (`ci\_cli`),

CONSTRAINT `propiedad\_ibfk\_1` FOREIGN KEY (`id\_loc`) REFERENCES `localidad` (`id\_loc`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `propiedad\_ibfk\_2` FOREIGN KEY (`ci\_cli`) REFERENCES `cliente` (`ci\_cli`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=23 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `visita`;

CREATE TABLE `visita` (

`ci\_cli` int(11) NOT NULL,

`cod\_prop` int(11) NOT NULL,

`ci\_emp` int(11) NOT NULL,

`fh\_visita` datetime NOT NULL,

`fh\_reserva` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`fh\_modificada` datetime DEFAULT NULL,

`fh\_cancelada` datetime DEFAULT NULL,

`operacion` varchar(25) NOT NULL,

PRIMARY KEY (`ci\_cli`,`cod\_prop`),

KEY `ci\_emp` (`ci\_emp`),

KEY `cod\_prop` (`cod\_prop`),

CONSTRAINT `visita\_ibfk\_1` FOREIGN KEY (`ci\_emp`) REFERENCES `empleado` (`ci\_emp`) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT `visita\_ibfk\_2` FOREIGN KEY (`ci\_cli`) REFERENCES `cliente` (`ci\_cli`) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT `visita\_ibfk\_3` FOREIGN KEY (`cod\_prop`) REFERENCES `propiedad` (`cod\_prop`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**6. Usuarios:**

Se contará con un total de 4 usuarios:

**Administrador**

Nombre: dbaeurekabit

Contraseña: administrador

**Gerente**

Nombre: gerenteeurekabit

Contraseña: gerente

**Agente inmobiliario**

Nombre: agenteeurekabit

Contraseña: agente

**Cliente**

Nombre: clienteeurekabit

Contraseña: cliente

**7. Permisos**

**7.1 Permisos de los diferentes roles:**

**Administrador:**

* Todas las acciones posibles en la base de datos

**Gerente:**

* Consulta las propiedades
* Consulta todas las agendas del día
* Consulta todas las agendas canceladas
* Consulta todas las agendas modificadas
* Cancela las visitas agendadas
* Modifica las visitas de la agenda (día, hora)
* Modifica y agrega agentes.
* Modifica y agrega clientes.

**Agente Inmobiliario:**

* Consulta las propiedades
* Cancelar las visitas agendadas
* Modifica las visitas de la agenda (día, hora)
* Modifica y agrega clientes.

**Cliente:**

* Consulta las propiedades
* Agenda visitas

**7.2 Sentencias utilizadas para la asignación de permisos:**

**Administrador (dbaeurekabit):**

* Grant all on proyecto\_eurekabit.\* to dbaeurekabit ;

**Gerente (gerenteeurekabit):**

* Grant insert, select, update, delete on visita to gerenteeurekabit;
* Grant select on propiedad to gerenteeurekabit;
* Grant insert, update on cliente to gerenteeurekabit;
* Grant insert, update on empleado to gerenteeurekabit;
* Grant insert, update on Tel\_Persona to gerenteeurekabit;

**Agente Inmobiliario (agenteeurekabit):**

* Grant select, update on visita to agenteeurekabit;
* Grant insert, update on cliente to agenteeurekabit;
* Grant insert, update on Tel\_Persona to agenteeurekabit;

**Cliente (clienteeurekabit):**

* Grant insert on visita to clienteeurekabit;
* Grant select on propiedad to clienteeurekabit;

**8. Respaldo de datos:**

Consideramos de suma importancia realizar siempre el respaldo de una base de datos.

En este caso la base de datos de la inmobiliaria es el pilar fundamental para poder brindar sus servicios de manera eficiente y eficaz, la importancia de los datos almacenados allí radican en lo esencial que resulta mantener el respaldo actualizado de los mismos, así como también de toda la base de datos en general.

Hemos tenido en cuenta que la magnitud de los datos es amplia ya que, día a día se gestionaran varias operaciones que modificaran los existentes y agregaran nuevos. A su vez, a arquitectura que posee la base de datos desarrollada hace que los datos se caractericen por ser históricos, es decir, que jamás se borrarán los mismos, en cuyo caso quedaran inactivos.

Teniendo en cuenta todas estas cualidades de este caso en particular hemos decidido que lo ideal es realizar dos copias de seguridad semanales, a una hora en la que el servidor no esté muy sobrecargado de transacciones, y la base de datos, no esté siendo utilizada en ese momento.

El respaldo estará automatizado en un script al que solo tendrá acceso el usuario root, así garantizamos la seguridad de la base de datos. El mismo se ejecutará en segundo plano todos los viernes y martes a las 7 am.

#Le asigna a la variable la fecha y hora del sistema

FECHA\_HORA=$(date "+%Y-%m-%d\_%H-%M-%S")

#Le asigna a la variable el nombre del archivo

NOMBRE=$(echo $FECHA\_HORA.copia\_seguridad.sql)

#Crea un respaldo de la base de datos incluyendo los datos y las sentencias de creación de la misma. Cabe destacar que se borrará la anterior y se sustituirá por la nueva copia de seguridad en la carpeta Backup

mysqldump -e - root –pEurekabit2019! proyecto\_eurekabit > /home/Backup/$(echo $NOMBRE.sql)

**9. Recuperación de datos:**

En el caso de necesitar hacer una recuperación de los datos de la base de datos, simplemente se deberá restaurar la última copia de seguridad almacenada, de la siguiente manera:

mysql -u root –pEurekabit2019! proyecto\_eurekabit < /home/Backup/\*.sql

De esta forma se va a restaurar toda la base de datos, incluyendo los datos que se encontraban almacenados en ella.

**10. Consultas:**

1. **Cantidad de propiedades por categoría**

Select propiedad.tipo, count(\*)

From propiedad

Group by Tipo;

1. **Cantidad de reservas históricas para cada número de dormitorios**

Select visita.cod\_prop, propiedad.cant\_habs

From propiedad, visita

Where visita.cod\_prop=propiedad.cod\_prop;

1. **Lista de propiedades y todas las visitas que tuvo en el pasado cada una**

Select propiedad.cod\_prop, count(visita.cod\_prop)

From propiedad, visita

Where visita.cod\_prop=propiedad.cod\_prop and date(fh\_visita)<curdate() and fh\_cancelada is null;

**4. Historial de visitas (fecha, hora (separados), nombre propietario, nombre agente, dirección de propiedad)**

Select DATE\_FORMAT(visita.fh\_Visita,'%Y-%m-%d') Fecha\_Visita, DATE\_FORMAT(visita.fh\_Visita,'%H:%i:%s') Hora\_Visita, propiedad.ci\_cli, visita.ci\_emp, propiedad.Direc

From propiedad, visita

Where visita.cod\_prop=propiedad.cod\_prop;

**5. Visitas programadas (fecha, hora (separados), nombre propietario, nombre agente, dirección de propiedad)**

Select DATE\_FORMAT(visita.fh\_Visita,'%Y-%m-%d') Fecha\_Visita, DATE\_FORMAT(visita.fh\_Visita,'%H:%i:%s') Hora\_Visita, propiedad.ci\_cli, visita.ci\_emp, propiedad.Direc

From propiedad, visita

Where visita.cod\_prop=propiedad.cod\_prop and date(fh\_visita)>curdate();

**6. Vistas canceladas (fecha, hora (separados), nombre propietario, nombre agente, dirección de propiedad)**

Select DATE\_FORMAT(visita.fh\_Visita,'%Y-%m-%d') Fecha\_Visita, DATE\_FORMAT(visita.fh\_Visita,'%H:%i:%s') Hora\_Visita, propiedad.ci\_cli, visita.ci\_emp, propiedad.Direc

From propiedad, visita

Where visita.cod\_prop=propiedad.cod\_prop and fh\_cancelada is not null;

**Hoja Testigo**