

# HOTEL

# Roma

Realidad N°5

Realizado por:

- Facundo Gabriel León
- Garces Brocal María Pía

Materia:

Base de Datos



## INDICE

• Universo del Discurso	2
▪ Realidad Asignada	2
▪ Supuestos Realizados	2
• Modelo Entidad-Relación	3-4
▪ Diagrama Completo	3
▪ Definición formal	4
▪ Análisis de Propiedades	4
• Modelo Relacional	5
• Creación de la Base de Datos	6-7
• Planteo y Resolución de Consultas	8



## Universo del Discurso:

### *Realidad Asignada: N°5 Hotel ROMA*

El hotel ROMA necesita armar una base de datos que facilite la administración del mismo. El hotel cuenta con un gran número de habitaciones y una sección de cocheras.

El personal que trabaja en el hotel se organiza en personal administrativo, personal de limpieza y personal de mantenimiento. De los administrativos se registra DNI, nombre, dirección, año de ingreso, una dirección de mail y el máximo nivel de estudios alcanzados (primario, secundario, universitario). De los empleados de limpieza se registra DNI, nombre, dirección, año de ingreso y un teléfono de contacto.

De cada habitación se registra su número, el piso donde se encuentra, la capacidad, el tipo de habitación (estándar, superior, premium), el valor diario y los empleados de limpieza encargados de la limpieza de la misma. Pueden existir habitaciones con el mismo número siempre que se encuentren en distintos pisos.

El hotel cuenta con cocheras que se asignan en el momento de la reserva a petición del cliente. Las cocheras tienen un costo adicional que no es el mismo para todas las cocheras. De cada cochera se registra el número que la identifica, para qué tipo de vehículo sirve (moto, auto, camioneta, van, etc) y el valor diario de la misma.

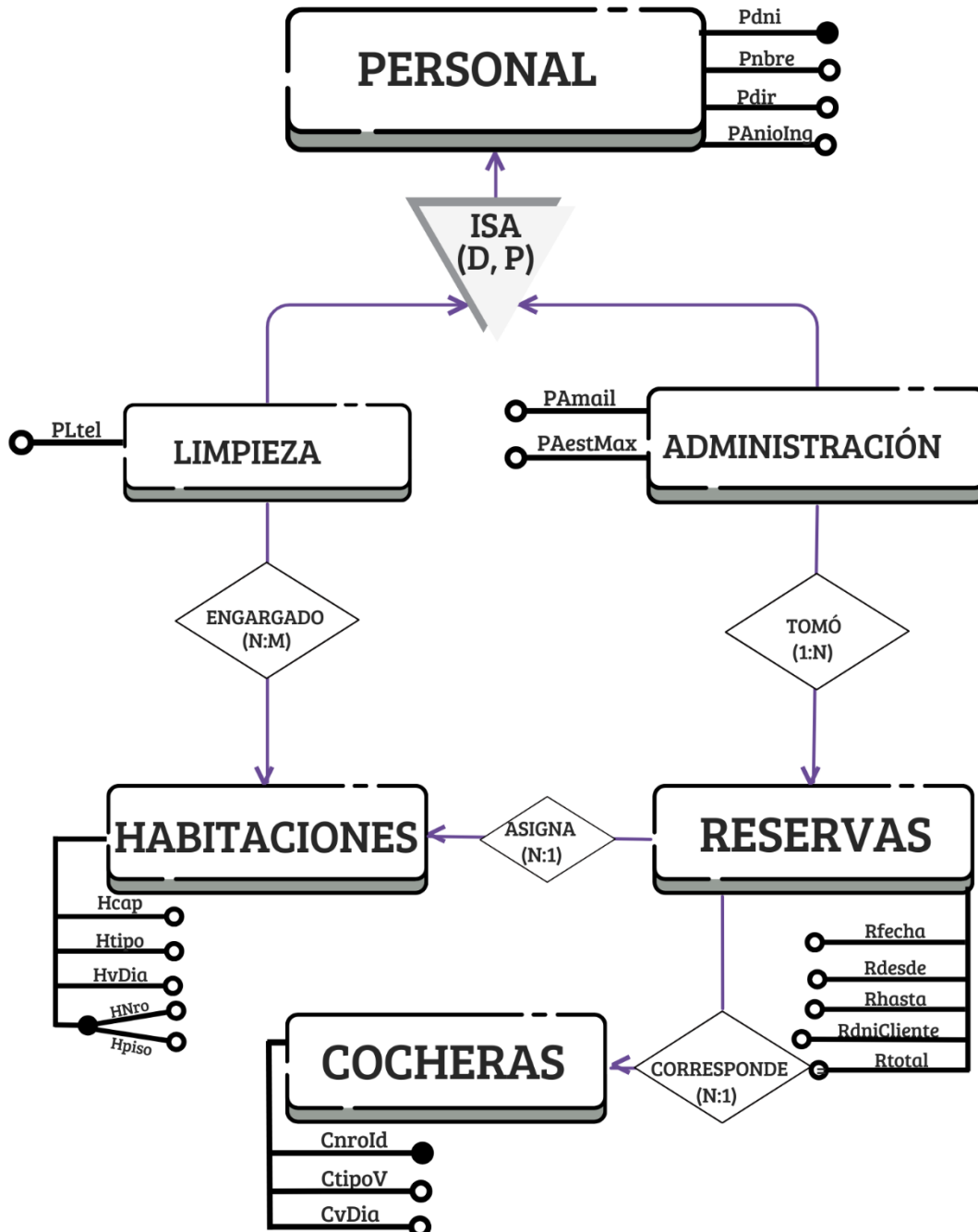
Se desea además mantener información sobre las reservas realizadas en el hotel. De cada reserva se registra la habitación reservada, si correspondiera la cochera reservada, la fecha en que se realizó la reserva, el período por el cual se reservó, el empleado administrativo que tomó la reserva, el DNI del cliente que realizó la reserva y el monto total a pagar.

### *Supuestos Realizados*

- Dado que una habitación puede tener varios empleados de limpieza encargados de la limpieza de la misma, suponemos que un empleado puede encargarse de la limpieza de más de una habitación.
- Las reservas realizadas deben ser tomadas por un empleado administrativo, por lo tanto, suponemos que un empleado administrativo puede tomar más de una reserva.
- Dado que en el relato no queda totalmente explícito, suponemos que en una reserva solo se le puede asignar una sola habitación y en el caso de que correspondiera, una sola cochera.

### Modelo Entidad-Relación:

#### Diagrama Completo



#### Propiedades:

- **Encargado**: No total, sobreyectiva.
- **Tomo**: No total, sobreyectiva.
- **Asigna**: Total, no sobreyectiva.
- **Corresponde**: No total, no sobreyectiva.



### *Definición formal*

#### ❖ Entidades:

Personal = {x/x es un personal de trabajo del Hotel Roma}

Dom(Pdni) =  $\mathbb{N}$

Dom(Pnbre) = Alfa+

Dom(Pdir) = AlfaNco+

Dom(PanioIng) =  $\mathbb{N}$

Cocheras = {x/x es una cochera del Hotel Roma}

Dom(CnroId) =  $\mathbb{N}$

Dom(CtipoV) = Alfa+

Dom(CvDia) =  $\mathbb{R}$

#### ❖ Relaciones:

Tomo = {(x,y)/x $\in$ Administrativos  $\wedge$  y $\in$ Reservas  $\wedge$  “El Administrativo x toma la reserva y”}

Asigna = {(x,y)/x $\in$ Reservas  $\wedge$  y $\in$ Habitaciones  $\wedge$  “A la Reserva x se le asigna la Habitación y”}

### *Análisis de las propiedades*

#### ❖ **Encargado:** (Entre Limpieza y Habitaciones)

No Total dado que puede haber al menos un personal de limpieza que no se encargue de limpiar ninguna habitación.

Sobreyectiva dado que toda habitación necesita de al menos un personal de limpieza.

#### ❖ **Tomo:** (Entre Administrativos y Reservas)

No Total dado que puede haber un personal administrativo que no haya tomado ninguna reserva.

Sobreyectiva dado que toda reserva debe ser tomada por al menos un personal administrativo.

#### ❖ **Asigna:** (Entre Reservas y Habitaciones)

Total dado que toda reserva debe tener asignada al menos una Habitación.

No Sobreyectiva dado que puede haber alguna Habitación que todavía no haya sido Reservada

#### ❖ **Corresponde:** (Entre Reservas y Cocheras)

No Total dado que puede haber Reservas que no le correspondan ninguna Cochera.

No Sobreyectiva dado que hay Cocheras que pueden no corresponder a ninguna Reserva.



### Modelo Relacional:

**Personal** = {Pcod, Pdni, Pnbre, Pdir, PanioIng, Pes}

Dom(Pcod) = |N

Dom(Pes) = {'LIMPIEZA', 'ADMINISTRATIVO', 'OTRO'}

**Limpieza** = {PLcod, PLtel}

Dom(PLcod) = |N

FK(PLcod) → Personal(Pcod)

**Administración** = {PAcod, PAmail, EMcod}

Dom(PAcod) = |N

FK(PAcod) → Personal(Pcod)

FK(EMcod) → Estudios\_Max(EMcod)

**Estudios\_Max** = {EMcod, EstMax}

Dom(EMcod) = {'PR', 'SC', 'UN'}

Dom(EstMax) = {'PRIMARIO', 'SECUNDARIO', 'UNIVERSITARIO'}

**Habitaciones** = {Hcod, Hnro, Hpiso, Hcap, Tcod, HvDia}

Dom(Hcod) = |N

FK(Tcod) → Tipo\_Habitacion(Tcod)

**Tipo\_Habitacion** = {Tcod, Tipo}

Dom(Tcod) = {'STD', 'SUP', 'PREM'}

Dom(Tipo) = {'ESTANDAR', 'SUPERIOR', 'PREMIUM'}

**Cocheras** = {CnroId, CtipoV, CvDia}

**Reservas** = {Rnum, Rfecha, Rdesde, Rhasta, RdniCliente, Rtotal,  
HcodAsigna, CnroIdCorresp, PAcodTomo}

Dom(Rnum) = |N

FK(HcodAsigna) → Habitaciones(Hcod)

FK(CnroIdCorresp) → Cocheras(CnroId)

FK(PAcodTomo) → Administración(PAcod)

**Encargado** = {PLcodEncargado, HcodEncargado}

FK(PLcodEncargado) → Limpieza(PLcod)

FK(HcodEncargado) → Habitaciones(Hcod)

### Aclaraciones:

La relación Estudios\_Max fue creada para registrar el dominio por extensión del atributo PA-N\_EST en el que guardamos el nivel de estudios máximos alcanzados por un personal administrativo (Primario, Secundario, Universitario)

La relación Tipo\_Habitacion fue creada para registrar el dominio por extensión del atributo H-TIPO en el que guardamos el tipo de cada habitación (Estándar, Superior, Premium)



### Creación de la Base de Datos:

```
CREATE TABLE Personal(  
    Pcod INTEGER NOT NULL,  
    Pdni INTEGER NOT NULL,  
    Pnbre TEXT NOT NULL,  
    Pdir TEXT NOT NULL,  
    PanioIng INTEGER NOT NULL,  
    Pes TEXT NOT NULL,  
    PRIMARY KEY(Pdni)  
    UNIQUE(Pcod)  
);  
  
CREATE TABLE Limpieza(  
    PLcod INTEGER NOT NULL,  
    PLtel INTEGER NOT NULL,  
    PRIMARY KEY(PLcod)  
    FOREIGN KEY(PLcod) REFERENCES Personal(Pcod)  
);  
  
CREATE TABLE Administracion(  
    PAcod INTEGER NOT NULL,  
    PAmail TEXT NOT NULL,  
    EMcod TEXT NOT NULL,  
    PRIMARY KEY(PAcod)  
    FOREIGN KEY(PAcod) REFERENCES Personal(Pcod)  
    FOREIGN KEY(EMcod) REFERENCES Estudios_Max(EMcod)  
);  
  
CREATE TABLE Estudios_Max(  
    EMcod TEXT NOT NULL,  
    EstMax TEXT NOT NULL,  
    PRIMARY KEY(EMcod)  
);  
  
CREATE TABLE Habitaciones(  
    Hcod INTEGER NOT NULL,  
    Hnro INTEGER NOT NULL,  
    Hpiso INTEGER NOT NULL,  
    Hcap INTEGER NOT NULL,  
    HvDia REAL NOT NULL,  
    Tcod TEXT NOT NULL,  
    PRIMARY KEY(Hcod)  
    UNIQUE (Hnro, Hpiso)  
    FOREIGN KEY(Tcod) REFERENCES Tipo_Habitacion(Tcod)  
);  
  
CREATE TABLE Tipo_Habitacion(  
    Tcod TEXT NOT NULL,  
    Tipo TEXT NOT NULL,  
    PRIMARY KEY(Tcod)  
);
```



Trabajo Integrador-1°Año-2°Cuatrimestre-2024

```
CREATE TABLE Cocheras(  
    CnroId INTEGER NOT NULL,  
    CtipoV TEXT NOT NULL,  
    CvDia REAL NOT NULL,  
    PRIMARY KEY(CnroId)  
);
```

```
CREATE TABLE Reservas (  
    Rnum INTEGER NOT NULL,  
    Rfecha TEXT NOT NULL,  
    Rdesde TEXT NOT NULL,  
    Rhasta TEXT NOT NULL,  
    RdniCliente INTEGER NOT NULL,  
    Rtotal REAL NOT NULL,  
    HcodAsigna INTEGER NOT NULL,  
    CnroIdCorresp INTEGER,  
    PAcodTomo INTEGER NOT NULL,  
    PRIMARY KEY(Rnum)  
    FOREIGN KEY(HcodAsigna) REFERENCES Habitaciones(Hcod)  
    FOREIGN KEY(CnroIdCorresp) REFERENCES Cocheras(CnroId)  
    FOREIGN KEY(PAcodTomo) REFERENCES Administracion(PAcod)  
);
```

```
CREATE TABLE Encargado(  
    PLcodEncargado INTEGER NOT NULL,  
    HcodEncargado INTEGER NOT NULL,  
    PRIMARY KEY(PLcodEncargado, HcodEncargado)  
    FOREIGN KEY(PLcodEncargado) REFERENCES Limpieza(PLcod)  
    FOREIGN KEY(HcodEncargado) REFERENCES Habitaciones(Hcod)  
);
```





### Planteo y Resolución de Consultas:

Semántica: Obtener la información del personal administrativo que tomó la reserva número 256, junto con el código de la habitación y de la cochera asignada.

- ❖ SELECT PAcod, Pnbre, Pdni, Pdir, PanioIng, PAmail, EstMAX AS 'Estudios Maximos', HcodAsigna, CnroIdCorresponde  
FROM Administracion  
INNER JOIN Personal ON PAcod=Pcod  
INNER JOIN Estudios\_Max ON Administracion.EMcod=Estudios\_Max.EMcod  
INNER JOIN Reservas ON PAcod=PAcodTomo  
WHERE Rnum=256;

Semántica: Obtener, por cada personal de administración el código, nombre y la cantidad de reservas tomadas.

- ❖ SELECT PAcodTomo, Pnbre, COUNT(Rnum) AS 'Cant Res Tomadas'  
FROM Administracion  
INNER JOIN Personal ON PAcod=Pcod  
INNER JOIN Reservas ON PAcod=PAcodTomo  
GROUP BY PAcodTomo, Pnbre;

Semántica: Obtener el código, número y piso de las habitaciones reservadas del día 19/10/2024 al 25/10/2024.

- ❖ SELECT HcodAsigna, Hnro, Hpiso  
FROM Reservas INNER JOIN Habitaciones ON HcodAsigna=Hcod  
WHERE Rdesde= '2024/10/19' AND Rhasta='2024/10/25';

Semántica: Obtener código, nombre, teléfono y año de ingreso del personal de limpieza encargado de las habitaciones 7 y 92, ordenados por nombre de forma ascendente.

- ❖ SELECT PLcod, Pnbre, PLtel, PanioIng  
FROM Encargado  
INNER JOIN Limpieza ON PLcodEncargado=PLcod  
INNER JOIN Personal ON PLcod=Pcod  
WHERE HcodEncargado=7 OR HcodEncargado=92  
ORDER BY Pnbre ASC;

Semántica: Obtén el código y nombre de cada personal administrativo junto con la cantidad total de reservas que ha tomado, mostrando solo aquellos que hayan tomado más de 5 reservas y hayan ingresado al Hotel a partir del 2014. Ordena los resultados por la cantidad de reservas en orden descendente.

- ❖ SELECT PAcod, Pnbre, COUNT(Rnum) AS 'Cant Res Tomadas'  
FROM Administracion  
INNER JOIN Personal ON PAcod=Pcod  
INNER JOIN Reservas ON PAcod=PAcodTomo  
WHERE PanioIng >= 2014  
GROUP BY PAcod, Pnbre  
HAVING COUNT(Rnum) > 5  
ORDER BY COUNT(Rnum) DESC;