

Base de Datos - TUDAI

# Trabajo Práctico Especial Sistema Gestión Hotelera

---

Facultad de Ciencias Exactas

UNCPBA



Carlos Cabrera Gentile - Email: [carloscgentile@gmail.com](mailto:carloscgentile@gmail.com)

Maximiliano Guerra - Email: [guerramd@gmail.com](mailto:guerramd@gmail.com)

Fecha de entrega: Lunes 25 de mayo

## Descripción del sistema

Se necesita hacer un sistema sobre una base de datos existente de un sistema de Gestión de servicios de hospedaje. El sistema pertenece a un servicio de alquileres de departamentos premium disponible para ejecutivos que viajan a EEUU.

Hay dueños de casas o departamentos que los proponen para alquiler temporario. La administración consiste en el control y seguimiento completo de las reservas, cada una debe tener el tipo de departamento, fechas, preferencias, etc. El periodo mínimo que se puede reservar un departamento es un día. El sistema debe llevar control del mantenimiento del departamento (es decir si fue limpiado y/o si se encontró algún problema en el mismo)

Las habitaciones de los departamentos pueden cambiar su configuración de una reserva a otra, respecto de la cantidad o tipo de camas y/o comodidades (televisor, sillones, frigobar, escritorios, mesas, etc).

## Elaboración de restricciones

En esta parte del Trabajo Práctico Especial donde se pedía agregar restricciones a las tablas, las realizamos para cada

declarativo estándar los incisos

Inciso B-A:

**Que las fechas de las reservas sean consistentes, es decir que la fecha de inicio de la reserva sea menor que la fecha de finalización.**

Forma SQL declarativo estándar:

```
ALTER TABLE gr04_reserva  
ADD CONSTRAINT fechas_reserva  
CHECK(fecha_hasta>fecha_desde)
```

En este caso, como es una restricción sobre un atributo con sólo hacer un simple check alcanza para controlarlo. Cuando se intenta incorporar una fecha que no corresponde el DBMS

da menor que la fecha el **error:**

Inciso B-B:

**Que el detalle de las habitaciones sea consistente con el tipo de departamento, es decir que si el tipo de departamento es de 2 habitaciones, en el detalle se consideren como máximo 2 habitaciones.**

Forma SQL declarativo estándar:

```
ALTER TABLE id_tipo_depto
ADD CONSTRAINT
CREATE ASSERTION CHECK(NOT EXISTS(select td.*
FROM gr04_tipo_dpto td
JOIN gr04_departamento d ON(d.id_tipo_depto=td.id_tipo_depto)
WHERE td.cant_habitaciones!=(select count(*)
FROM gr04_habitacion h
GROUP BY id_dpto
HAVING id_dpto=d.id_dpto)))
```

En el ejemplo mencionado debemos realizar un `ALTER TABLE`, ya que la restricción incluye más de una tabla. El DBMS informa del error cuando un departamento que tiene un tipo de departamento el cual, tiene 1 habitación, no se puedan agregar más de una.

Inciso B-C:

**Que tanto la persona que realiza la reserva como los huéspedes no sea el propietario del departamento.**

Forma SQL declarativo estándar:

```
ALTER TABLE gr04_huesped_reserva
ADD CONSTRAINT
CREATE ASSERTION CHECK(NOT EXISTS(select 1
FROM gr04_departamento d
JOIN gr04_reserva r ON(r.id_dpto=d.id_dpto)
JOIN gr04_huesped_reserva hr ON(hr.id_reserva=r.id_reserva)
WHERE d.tipo_doc=hr.tipo_doc and d.nro_doc=hr.nro_doc)))
```

```
ALTER TABLE gr04_reserva
ADD CONSTRAINT
CREATE ASSERTION CHECK(NOT EXISTS(select 1
from gr04_departamento d
JOIN gr04_reserva r ON(r.id_dpto=d.id_dpto)
WHERE d.tipo_doc=r.tipo_doc AND d.nro_doc=r.nro_doc)))
```

Al igual que el inciso anterior, debemos realizar un `ALTER TABLE`, ya que la restricción incluye más de una tabla. Si quien hace la reserva y/o se hospeda en dicho departamento es propietario del mismo el DBMS informa del error.

Inciso B-D:

**Que la cantidad de huéspedes no supere la cantidad máxima de personas permitidas para una reserva.**

Forma SQL declarativo estándar:

```
ALTER TABLE gr04_huesped_reserva
ADD CONSTRAINT
CREATE ASSERTION CHECK(NOT EXISTS(select 1
FROM gr04_departamento d
JOIN gr04_tipo_dpto td ON(td.id_tipo_depto=d.id_tipo_depto)
WHERE td.cant_max_huespedes >=(
SELECT count(*) AS "cant huespedes"
FROM gr04_reserva r
JOIN gr04_huesped_reserva hr ON(hr.id_reserva=r.id_reserva)
GROUP BY r.id_reserva
HAVING id_dpto=d.id_dpto)
```

Al igual que las anteriores restricciones, como se involucran más de una tabla, debemos realizar un `ALTER TABLE`. El DBMS informa del error cuando en un tipo de departamento se quieren agregar más huéspedes de los permitidos.

Inciso C:

Consideramos que la mejor opción para la resolución del presente inciso es por medio de un procedimiento que `CALL` en la consigna de trabajo. Nos pareció la opción más simple de resolverlo ya que de otra manera, tendríamos que crear una tabla fuera del procedimiento y posteriormente, colocar los datos en la misma, teniendo que borrar los campos cada vez que llamemos al procedimiento `CALL`.

Inciso D:

Definición de vistas

Para esta parte del Trabajo Práctico Especial, generamos 2 vistas:

- Devuelva un listado de todos los departamentos del sistema junto con la recaudación de los mismos en los últimos 6 meses.
- Devuelva un listado con los departamentos ordenados por ciudad y por mejor rating (estrellas).

Ambas vistas son actualizables ya que las mismas contienen la clave primaria, en este caso, de la tabla departamento. Las vistas que se crean tienen todas las columnas e información de la tabla. Tampoco se están utilizando subconsultas adentro del select.