El Trabajo Práctico Especial consiste en la resolución de un conjunto de controles y servicios sobre una base de datos que mantiene un sistema de “Reservas de Departamentos Premiun” según las consignas que se indican en cada caso. El esquema inicial es provisto por la Cátedra.

**Consideraciones Importantes:**

* Se podrá resolver en grupos (de dos integrantes), que deberán **inscribirse previamente** en el [formulario](https://docs.google.com/forms/d/1GJlwYajx0T2kicP5w-17doPZ5-NnA_IgErJZIdtBTO0) hasta el 31/05/2018
* Se recomienda **leer el enunciado completo** del trabajo antes de resolverlo.
* Se deberán **respetar las pautas** de entrega indicadas en cada parte del enunciado y la **convención de nombres** indicada al final de este documento,para todos los objetos del esquema de la base de datos. Se descontarán hasta 2 puntos de la nota final por no respetar las pautas de entrega y/o reglas de nombres solicitadas.
* Los scripts entregados se probarán y testear en el DBMS que se usa habitualmente para las prácticas durante el curso (dbases.exa.unicen.edu.ar), vía web, utilizando el *PhpPgAdmin* o mediante la aplicación de escritorio *PgAdminIII*. Cada grupo debe asegurarse que los scripts **ejecuten sin errores para que la cátedra pueda proceder a la corrección del TPE**, en el orden establecido; por esto es recomendable que los chequee debidamente a fin de asegurar que los mismos se ejecuten sin inconvenientes.
* Todo el código SQL entregado debe estar **optimizado**, es decir utilizando estrictamente los atributos, tablas y/u operaciones necesarias, de acuerdo a lo requerido.

**Entrega del TPE:** El trabajo podrá entregarse hasta el día **18 de Junio a las 23:59.** Se deberá completar una tarea (ENTREGA) en la plataforma Moodle en la cual se deberán subir los archivos scripts solicitados en forma **separada**. En la misma tarea se confirmará el día siguiente a la entrega, si los scripts corrieron sin errores o, de haber advertido algún inconveniente en su ejecución, se informará en dicha respuesta el primer error encontrado.

**Re-entrega del TPE o entrega fuera de término:** En este caso se aceptará que el grupo re-entregue **-por única vez-** los scripts corregidos hasta el **22 de Junio a las 23:59.** Se deberá completar una tarea (RE-ENTREGA) en la plataforma Moodle en la cual se deberán subir los archivos scripts solicitados en forma **separada**. Se **descontarán 1 (un) punto** en la nota final del TPE, la misma pauta se aplicará para quienes entreguen fuera de término. En la misma tarea se confirmará el día siguiente a la re-entrega, si los scripts corrieron sin errores o, de haber advertido algún inconveniente en su ejecución, se informará en dicha respuesta el primer error encontrado. En caso de que la re-entrega aún tuviera algún error de ejecución, el trabajo **NO** será evaluado.

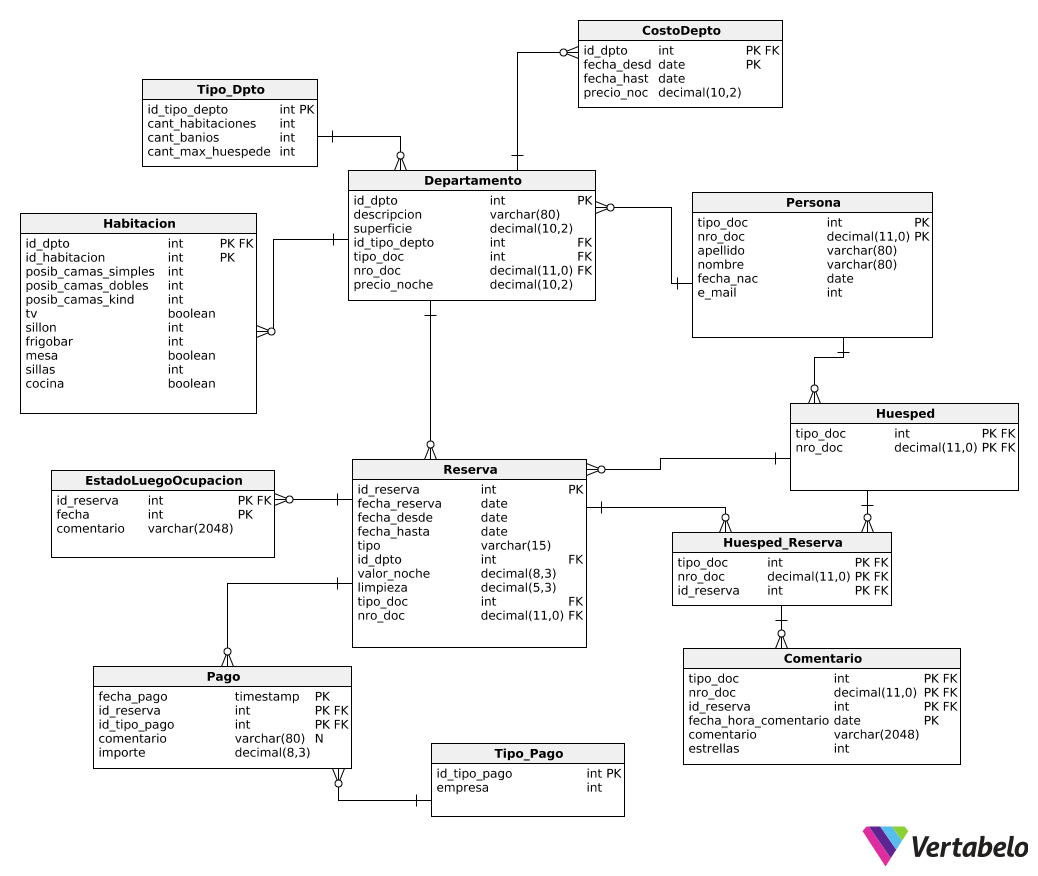
**Descripción resumida del Sistema de “Gestión Hotelera”**

Una descripción breve del contexto del problema es la siguiente:

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).

**Implementación de controles y servicios**

## A partir del anterior esquema original de la BD se requiere:

## AJUSTE DEL ESQUEMA

* Debe crear su esquema a partir del archivo XML de Vertabelo provisto por la cátedra.
* Debe ajustar dicho esquema según las pautas de nomenclatura indicada en la sección Reglas de Nombres.
* En caso de que lo considere necesario, debe modificar el esquema original, sin borrar tablas o atributos ya existentes, e incorporar el soporte para todos los aspectos definidos por el usuario que aún no estén plasmados en el esquema provisto.
* Debe generar un script de generación del esquema denominado **GXX\_Creación.sql** que cree todas las tablas con sus restricciones de nulidad, primary key y foreign keycon las tablas del diagrama de la cátedra y las que cada grupo agregó para las modificaciones.
* Debe agregar por lo menos 5 filas a cada tabla del esquema (algunas tablas les quedarán con más de 5 filas) y éstas deben ser agregadas antes de todas las restricciones a implementar.

## ELABORACIÓN DE RESTRICCIONES

## Para cada una de las restricciones/reglas del negocio enunciadas a continuación:

## 1) Escriba la restricción de la manera que considere más apropiada en *SQL estándar declarativo*, indicando su tipo y características correspondientes. NO se tendrán en cuenta las tipificaciones correctas pero con implementaciones inadecuadas.(*incluya el desarrollo en el informe*)

## 2) Impleméntela en PostgreSQL de la forma más adecuada, según las posibilidades que ofrece el DBMS (*incorpore la solución al script GXX\_Cambios.sql* )

3) Provea una sentencia de modificación, borrado y/o inserción según corresponda, de modo que promueva la activación de la restricción (***incorpórela comentada al script*** *GXX\_Cambios.sql de modo que la ejecución del mismo no de error*). Explique brevemente la respuesta por parte del DBMS (*en el informe)*.

Se deben realizar los siguientes controles en el esquema de datos:

1. 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.
2. 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.
3. Que tanto la persona que realiza la reserva como los huéspedes no sea el propietario del departamento
4. Que la cantidad de huéspedes no supere la cantidad máxima de personas permitidas para una reserva

**C. SERVICIOS**

Implemente lo siguiente con el recurso que crea más conveniente (triggers, procedimientos y/o vistas). Si necesita para este ítem crear tablas adicionales al esquema creado en el script de creación (**GXX\_Creacion.sql)** incorpórelas en el script **GXX\_Cambios.sql** .

1. Por cada departamento en el sistema, de el estado en una fecha determinada, esto es si el mismo está Ocupado o Libre.
2. Dada una rango de fechas y una ciudad, devuelva una lista de departamentos disponibles.

## D. DEFINICIÓN DE VISTAS

Escriba la sentencia SQL para crear las vistas detalladas a continuación e incorpórelas en el script **GXX\_Cambios.sql** . En cada caso comente y ejemplifique (*en el informe*) si es actualizable o no, indicando la/s causas/s:

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

**E. TUDAI - SITIO A REALIZAR**

Los alumnos de la carrera TUDAI deberán implementar un sitio en PHP que acceda a la base de datos y que permita realizar interactivamente una consulta que para un determinado departamento en un mes muestre los días que está disponible y los que está ocupado..

**Pautas de entrega**

Se deberá entregar un informe en pdf (de **no más de 5 páginas**), explicando concisamente el trabajo desarrollado y 3 (tres) scripts, que se deben **poder ejecutar en orden (detallados en los ítems 1, 2 y 3) y sin producir errores**:

1. script ***GXX\_Creación.sql***, que deberá contener la creación de lo solicitado en el punto **A**. Nota: cambiar los nombres de los objetos para ajustarlos a los de cada grupo (GXX\_).
2. script **GXX\_Cambios.sql**, que deberá incluir la implementación de las restricciones, las vistas y los servicios detallados en los puntos **B, C y D**. Nota: Incluir como comentario dentro del script el ítem que resuelve en cada caso.
3. script ***GXX\_Borrado.sql****,* que incluya las sentencias de borrado de todos los objetos del esquema (tanto los creados por el script GXX\_*Creacion.sql* como los objetos incorporados mediante el script *GXX\_Cambios.sql )*
4. Informe ***GXX\_Informe.pdf****,* en el que se expliquen y justifiquen las decisiones asociadas a la implementación de cada una de las restricciones de integridad, tanto en forma declarativa según el estándar SQL como de forma procedural (para aquellas que declarativamente no soporta PostgreSQL), la definición de las vistas y de los servicios requeridos.

**Orden de la Ejecución de Prueba**

El orden de la ejecución de los scripts de la prueba que realizará la cátedra será el siguiente:

GXX\_Creacion.sql

GXX\_Cambios.sql

GXX\_Borrado.sql

**Reglas de nombres**

Para el desarrollo y corrección de los esquemas se seguirán estos estándares que permiten identificar cada trabajo y objeto con nombres completos y descriptivos.

**Consideraciones Generales**

El prefijo de grupo <GRXX> se usará en tablas, vistas, procedimientos almacenados, triggers, índices y constraints. XX se deberá reemplazar por el número de grupo asignado por la cátedra en cada caso.La nomenclatura a seguir es:

1. **Tablas/Vistas**: <GRXX>\_<nombre\_tabla o vista>
2. **Índices: IDX**\_<GRXX>\_<nombre\_indice>
3. **Constraints**

Clave Primaria: **PK**\_<GRXX>\_<nnnnn>

donde: nnnn = el nombre de la tabla para la que la constraint se construye (Por ejemplo: PK\_GRXX\_Beneficiario)

Clave Extranjera: **FK**\_<GRXX>\_<pppp>\_<cccc>

donde pppp = el nombre de la tabla referenciante u original, cccc = el nombre de la tabla referenciada.

Unique: **UQ**\_<GRXX>\_<nnnn>\_<cccc>

donde: nnnn = el nombre de la tabla para la que la constraint se construye, cccc = el nombre del campo interviniente.

1. **Procedimientos, Triggers y Funciones**

Triggers: **TR\_**<GRXX>\_<tbl>\_<nnnn>

siendo <nnnn> el nombre que considere adecuado para el trigger y <tbl> la tabla a la cual está asociada el trigger.

Procedimientos: **PR\_**<GRXX>\_<nnnn>

siendo <nnnn> el nombre que considere adecuado para el procedimiento.

Funciones: **FN\_**<GRXX>\_<nnnn>

siendo <nnnn> el nombre que considere adecuado para la función.

Funciones de triggers: **TRFN\_**<GRXX>\_<nnnn>

siendo <nnnn> el nombre que considere adecuado para la función de trigger.