

Fundamentos de Bases de Datos.

Proyecto final.

Profesora: Dra. Amparo López Gaona
alg@ciencias.unam.mx

Laboratorio: Lic. Carlos Augusto Escalona Navarro
caen@ciencias.unam.mx

13 de enero de 2022

Se dan a conocer especificaciones de entrega para el proyecto final.

1. Caso de uso

Para el proyecto final vamos a utilizar el caso de uso visto en el laboratorio.

2. Actividad

El proyecto final deberá contener las siguientes consideraciones:

- Deberás utilizar el SDBD **PostgreSQL** para su implementación.
- Es necesario utilizar **todos los aspectos** revisados en el curso.
- Crear el modelo E/R asociado al caso de uso especificado. Se debe justificar el diseño de manera concisa e indicar de qué forma su esquema empata con las reglas del caso de uso. Recuerda que un buen diseño de base de datos es muy importante.
- Traducir al modelo relacional.
- Indicar cuáles son las dependencias funcionales, evita aquellas que son triviales y/o redundantes.
- Normalizar tu diseño de base de datos del punto anterior, utilizando el conjunto de dependencias funcionales que identificaste. Para fines de este proyecto basta con llegar a 3NF. Para calificar este apartado no basta con indicar únicamente las relaciones resultantes, debes indicar el procedimiento y justificación.

- Plasmar las relaciones de la salida en un nuevo esquema relacional. Para esta parte sólo puedes hacer uso de los **ALTER TABLE**, para la modificación de tus tablas.
- Construir la base de datos del esquema del punto anterior. Deberás asegurar los 3 tipos de integridad.
- Generar al menos 5 funciones, 5 procedimientos almacenados y 5 disparadores para implementar las reglas de negocio dentro de tu base de datos.
- Poblar la base de datos. Para esta parte haz uso de los procedimientos almacenados que se te solicitaron en el punto anterior, al menos debe de haber 15 registros en cada una de tus tablas.
- Deberás de proponer al menos 7 consultas no triviales que deben poner a prueba tu manejo en SQL para recuperación de información, las cuales deben ser ad-hoc para proporcionar información interesante, por ejemplo, conocer la cantidad de ventas realizadas por los cajeros, el número de ventas realizadas en una sucursal; ventas por sucursal y por forma de pago, cajeros que realizan más ventas, historial de ventas por productos, etcétera (este punto se deja a tu completa elección, deben tratar de que la información a entregar sea de interés para el caso de uso). Es importante que no se cree la base de datos teniendo en mente la satisfacción de los reportes listados arriba. Se considera un error grave de diseño orientar el mismo a procesos y no a datos.
- Construir un diccionario de datos que contenga definiciones de campos, las abreviaturas o formatos de la base de datos y restricciones creadas en cada atributo. El formato es libre y pueden utilizar la herramienta de edición que deseen.

3. Entregables

1. El diseño de la base de datos deberá estar en un archivo llamado **diseño.pdf**, el modelo entidad – relación se deberá realizar con DIA y es necesario justificar las relaciones y cómo es que están relacionadas entre sí, no olvides indicar tus llaves y discriminantes.
2. Al documento del punto anterior agrega el esquema de la base de datos del modelo relacional, indicando el conjunto de dependencias funcionales válidas para el problema que se te plantea y justificando el proceso de traducción, no olvides indicar las llaves primarias, foráneas y candidatas.
3. El esquema de la base de datos normalizado deberá estar en un documento llamado **normalizacion.pdf** bajo la tercera forma normal, es imprescindible que muestres todos los pasos de la normalización. Se deberá generar el nuevo esquema relacional de la base de datos después de este proceso.

4. Un script llamado **DDL.sql** que contenga todas las instrucciones necesarias para crear el esquema de la base de datos. Debes considerar que deberá ser un esquema restringido que asegure los 3 tipos de integridad. El sistema debe rechazar toda inserción de valores que no estén permitidos en los campos en los que se ha listado el dominio válido. Lo mismo debe ocurrir con las actualizaciones. Se debe vigilar la integridad referencial, de dominio y de entidad. Se deben especificar las políticas para el manejo de dicha vigilancia y reflejarse en la base de datos de manera automática bajo las actualizaciones y borrados.
5. Un script llamado **funciones.sql** con las 5 funciones solicitadas. Deberán estar correctamente documentadas e indicar el objetivo que tiene cada una de ellas.
6. Un script llamado **precedimientos_almacenados.sql** con los 5 procedimientos almacenados solicitados. Deberán estar correctamente documentados e indicar el objetivo que tiene cada uno de ellos.
7. Un script llamado **disparadores.sql** con los 5 disparadores solicitados. Deberán estar correctamente documentados e indicar el objetivo que tiene cada uno de ellos.
8. Un script llamado **DML.sql** con las instrucciones para poblar la base de datos con los datos que utilizaste para hacer tus pruebas. La cantidad de datos se deja a tu elección, pero deberá ser la mínima requerida para probar el funcionamiento completo del sistema (un número pobre de tuplas será penalizado). Recuerda que esta parte **la debes hacer utilizando los procedimientos implementados anteriormente**.
9. Un script llamado **consultas.sql** donde especifiques todas y cada una de las consultas SQL que hayas considerado en tu proyecto final, recuerda que es importante que las funciones que implementaste anteriormente se utilicen en esta parte.

NOTA:

Recuerda que para la entrega de tu proyecto deben de seguir los lineamientos de laboratorio para su entrega.

Todos los documentos a entregar deben tener excelente presentación, recuerda que forma parte de las buenas prácticas de desarrollo.

La entrega deberá ser el día 28 de enero de 2022.