

# Entrega 2: Diseño de una aplicación

## 1. Descripción general

Finalmente ha llegado el momento de trabajar en tu aplicación!. En esta entrega debes diseñar una buena base de datos y una interfaz web para hacer consultas. El objetivo de la entrega es modelar correctamente un problema relacionado con el manejo de datos. Luego se evaluará la habilidad para crear consultas SQL en un entorno realista, y aprenderás a trabajar con una conexión a una base de datos local.

## 2. Enunciado

### 2.1. Crear modelo

En esta parte deberás crear un modelo relacional usando los conceptos aprendidos en clases. Debes hacer lo siguiente:

- Construir un modelo de Entidad/Relación que represente de manera fiel la situación planteada en la entrega 1 (recuerda que la aplicación es distinta para los grupos par o impar). Se deben ilustrar las entidades, asociaciones y cardinalidades.
- Traspasar el modelo Entidad/Relación a un esquema relacional especificando los *datatypes* correctos e incluyendo llaves primarias y llaves foráneas como restricciones de integridad (no es necesario agregar otro tipo de restricciones adicionales).

### 2.2. Justificar modelo

En esta parte debes justificar que la situación está bien modelada y tu modelo es correcto según la teoría. Específicamente:

- Debes mostrar que tu modelo está en BCNF, o si prefieres en 3NF. Para ello debes listar las dependencias funcionales de cada tabla y mostrar que está todo normalizado.

Puede que necesites hacer algunos ajustes a tu modelo para normalizarlo. El modelo final es el que debe ser entregado en el reporte.

## 2.3. Importar datos

Una vez que tengas el modelo preciso, es tiempo de importar los datos. Junto con esta entrega se subirán archivos CSV con todos los datos que debes guardar en tu base de datos. Ojo: esos datos no están normalizados, por lo que deberás trabajar un poco para importarlos<sup>1</sup>. Puedes usar una interfaz externa<sup>2</sup> o desde la consola.<sup>3</sup>

Específicamente, hay que:

- Pre-procesar los datos sin pérdida de información para que se ajusten a tu esquema final.
- Ingresar a la base de datos del servidor llamada **grupoXe2** (por ejemplo grupo166e2 si tu grupo es 166) y crear todas las tablas de tu modelo relacional.<sup>4</sup> Posteriormente, importar los datos de manera que la importación sea sin pérdida de información.

## 2.4. Consultas en SQL

En esta sección se detallan las consultas para cada grupo. **¡No son las mismas consultas de la entrega 1!** (Pero sí son parecidas.) Algunas consultas tienen parámetros (quiere decir que la consulta cambiará dependiendo de su input). Cómo manejar los inputs y selecciones se explica más abajo.

### Consultas grupo impar:

1. Seleccione una categoría del producto. Muestre los nombres de los cinco productos más carros de esta categoría.
2. Ingrese una comuna. Liste todas las tiendas ubicadas en esta comuna, indicando su dirección y el número de teléfono.
3. Ingrese el id de un producto. Liste el nombre del producto, el número de cajas necesarias para entregar este producto, y el peso total de todas las cajas necesarias para guardar este producto.
4. Ingrese nombre de un producto y seleccione una región. Listar todas las tiendas de la región que tienen dicho producto en stock. Adicionalmente, hay que mostrar el precio del producto. Si hay una oferta para el producto en una tienda, mostrar esta información.
5. Seleccione una región. Compute el número total de personas trabajando en las tiendas de esta región.

---

<sup>1</sup>En general, puedes procesar estos *csv* con Python o tu lenguaje de programación favorito para facilitar la importación.

<sup>2</sup>Como DataGrip.

<sup>3</sup>Como se explica en la Wiki.

<sup>4</sup>En cada entrega de proyecto tendrán que usar una BD distinta. Esto con el objetivo que congelar las anteriores para poder corregir.

6. Seleccione una región. Para cada tienda de esta región listar el número de personas de cada género trabajando en esta tienda, y su edad promedio (para cada género).
7. Seleccione una región. Listar, para cada tienda de esta región, su dirección, y el valor total del stock disponible en esta tienda. Valor del stock de un producto es su precio multiplicado por el número de unidades de este item en el stock. El valor del stock completo es suma de los valores de stock de cada producto que la tienda tiene en el stock. Para este cómputo **no** es necesario aplicar el descuento que producen las ofertas.

### **Consultas grupo par:**

1. Ingrese una fecha. Para cada despacho programado para esta fecha, liste el nombre del cliente y su dirección.
2. Ingrese el id de una compra. Para la compra listar el nombre del cliente, su dirección, y el valor de la compra.
3. Ingrese el id de una compra. Para la compra computar el número total de las cajas que se ocuparán para guardar/entregar los productos comprados. Es importante notar que un producto se puede comprar más de una vez (por ejemplo si compro cinco lámparas). Se pide computar el número total de las cajas.
4. Ingrese una fecha y la patente de un vehículo. Computar el peso total de todos los despachos programados para dicha fecha y asignadas a este vehículo.
5. Seleccione una región. Para esta región compute el número total de repartidores de cada género trabajando en esta región y su edad promedio (para cada género).
6. Ingrese el id de un cliente. Liste todas las compras realizada por este cliente. Para cada compra listar su id, el número de productos comprados (si un mismo producto se compra 5 veces, esto sí cuenta), y el número de cajas necesarias para guardar todos estos productos (de nuevo, ojo con productos comprados 2 o más veces).
7. Seleccione una región. Para esta región, liste los cinco cliente que gastaron la mayor cantidad de plata en todas sus compras. Quiere decir, para cada cliente, computar el valor de todas las compras realizadas por este cliente, y entregar cinco clientes con el mayor valor de esta suma.

**Ojo:** Las búsquedas que solicitan ingresar un dato, deben ser case-insensitive y con matching parcial. Esto quiere decir, por ejemplo, que las búsquedas por “Lámpara LED”, “lampara led”, “LAMPARA led” o “Lampar LED” (dónde falta la última letra de lampara) deben entregar los mismos resultados. Además recuerden que ingresar y seleccionar requieren tipos de input distintos, algo que debe ser reflejado en su página web (y explicado abajo).

## 2.5. Página Web

Tu grupo deberá implementar una interfaz web en PHP para visualizar sus consultas. La interfaz puede ser simple; para las consultas que no tienen input, solo se requiere que cada consulta despliegue el resultado en la misma página, o un **link de ruta relativa**<sup>5</sup> a una nueva página con el resultado de la consulta. Para las consultas con input, tu interfaz debe ser capaz de recibir sólo el parámetro necesario (no la consulta SQL entera). Por ejemplo, si la consulta requiere un mueble, el input debe funcionar solamente escribiendo el nombre o identificador de ese mueble.

Adicionalmente, se debe hacer la distinción entre los dos tipos de inputs con los que deberían trabajar:

- aquellos en los que se solicita seleccionar información, los cuales requieren del uso de un dropdown; y
- aquellos campos que piden ingresar un dato, en los que el usuario debe escribir el parámetro a utilizar.

## 3. Detalles adicionales

### Ubicación de su entrega

1. Los archivos de la app web deben estar ubicados en el directorio `/home/grupoX/Sites`
2. El *homepage* debe ser `/home/grupoX/Sites/index.php`, es decir, ubicarse en el directorio del item anterior y llamarse `index.php`.
3. El reporte se debe entregar en la carpeta `/home/grupoX/Entrega2`. El nombre del archivo no es relevante.

Es importante que siga estas tres instrucciones, de lo contrario su trabajo **no será corregido**. También asegúrense de que el archivo o carpeta se subió correctamente, navegando dentro de ella en el servidor y accediendo a su página web desde un navegador <sup>6</sup>. Recuerden que al final de la rúbrica contarán con una imagen de como debería verse la distribución de sus carpetas al ingresar al servidor.

### Reporte

Además de la aplicación web, deberá entregar un archivo PDF que contenga:

- Un **diagrama** E/R de su dominio incluyendo las cardinalidades de las relaciones entre entidades. Si alguna entidad tiene un nombre no sugerente deberá explicar su significado.

---

<sup>5</sup>Por ejemplo, link a `/resultado1.php` y no link a `http://codd.ing.puc.cl/grupoXX/resultado1.php`

<sup>6</sup>Como se indica en la Wiki del curso

- El esquema relacional que resulta del diagrama E/R anterior, incluyendo el tipo de dato de cada atributo y todas las restricciones de llaves primarias y foráneas de las relaciones.
- La justificación de por qué ese esquema se halla en BCNF o 3NF.
- Todas las consultas en SQL que implementaron en su aplicación.
- Incluir cualquier supuesto que hayan realizado sobre la entrega, mientras sea razonable.
- Deseablemente, incluir cualquier detalle que facilite la corrección estilo `readme`.

## Corrección de las consultas

Debes asegurarte que el resultado de las consultas sea consistente con los datos. Para corregir las consultas, se probarán 4 casos de prueba predeterminados en aquellas que reciban parámetros. Las otras se prueban *one-time*. La nota dependerá de la correctitud del resultado en cada caso. Para automatizar la corrección y tener su nota lo antes posible, les pedimos que respeten el formato de *output* que se enseñará en la **ayudantía del viernes, 28 de abril**.

## Bonificación +0.5

Para los alumnos interesados, se ofrecerá 0.5 puntos extra en esta entrega a aquellos grupos cuya página sea sobresaliente en su diseño (a juicio del corrector). Se sugiere buscar plantillas HTML en la web. <sup>7</sup>

## 4. Detalles Académicos

Deberán trabajar según los grupos asignados. Expliquen adecuadamente su trabajo. El equipo corrector se reserva el derecho de bajar la nota de aquellos trabajos que no estén bien explicados, que cuenten con demasiadas faltas de ortografía o trabajos en que se dificulte la corrección (por ejemplo, al no adjuntar el informe). Además, el corrector puede optar por **no asignar puntaje a una consulta** si esta no se logra correr en la aplicación web.

Finalmente, deben exponer su interfaz utilizado el servidor otorgado en la entrega anterior. La interfaz debe estar construida en PHP, y debe permitir realizar todas las consultas que se señalan. Para más detalles sobre esta entrega, haremos una ayudantía el **viernes, 28 de Abril**.

**El plazo para esta entrega vence el día viernes, 19 de Mayo a las 23:59 hrs.**

---

<sup>7</sup>Un sitio muy utilizado es Bootstrap.