

# Entrega 2: Diseño de una aplicación

## 1. Descripción general

Finalmente ha llegado el momento de trabajar en la aplicación web. 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** 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 tipo de grupo. Ojo que no son las mismas consultas que en la entrega 1.

### Consultas grupo impar:

1. Muestre todos los *username* junto a su correo.
2. Ingrese el nombre de un país. Muestre todos los nombres de las ciudades del país con ese nombre en su base de datos.
3. Ingrese un *username*. Muestre todos los nombres distintos de países en los que ha hospedado el usuario con ese *username* mediante hoteles de la agencia.
4. Ingrese el identificador de un usuario. Muestre la cantidad de dinero que ha gastado el usuario con ese identificador en todos los tickets que ha comprado.
5. Entregue el identificador y nombre de usuario junto a la fecha de inicio en formato YYYY-MM-DD, la fecha de término y el nombre del hotel de las reservas que parten desde el 01 de enero del 2020 y terminan antes del 31 de marzo del 2020, ambas fechas inclusive.
6. Ingrese dos fechas en formato YYYY-MM-DD: una de inicio y una de fin. Muestre el id, nombre de usuario y el total de dinero gastado en tickets entre esas dos fechas, ambas inclusive.

---

<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.

### Consultas grupo par:

1. Muestre todos los nombres distintos de las obras de arte.
2. Muestre todos los nombres de las plazas que contengan al menos una escultura de “Gian Lorenzo Bernini”.
3. Ingrese el nombre de un país. Muestre el nombre de todos los museos de ese país que tengan obras del renacimiento.
4. Para cada artista, entregue su nombre y el número de obras en las que ha participado.
5. Ingrese una hora de apertura en formato **hh:mm:ss**, una hora de cierre y una ciudad. Muestre los nombres de las iglesias ubicadas en esa ciudad, abiertas entre esas horas (inclusive) junto a todos los nombres de los frescos que encuentra en cada una de ellas. Una fila por cada tupla.
6. Encuentre el nombre de cada museo, plaza o iglesia que tenga obras de todos los periodos del arte que existan en la base de datos.

**Ojo:** Las búsquedas deben ser *case-insensitive* y con *matching* parcial. Esto quiere decir, por ejemplo, que las búsquedas por “Estados Unidos”, “estados unidos”, “ESTADOS unidos” o “Estados Uni” deben entregar los mismos resultados en una consulta.

## 2.5. Página Web

Tu grupo deberá implementar una interfaz web en PHP para visualizar sus consultas. La interfaz puede ser simple; solo se requerirá 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. No se espera que la página web tenga un cuadro de input para recibir toda la consulta SQL, sino que debe ser capaz de recibir sólo el parámetro necesario. Por ejemplo, si la consulta requiere el identificador del usuario, el input debe funcionar solamente escribiendo el identificador.

## 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 ítem anterior y llamarse `index.php`.

---

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

## 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.
- 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 3 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>

---

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

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

## 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 utilizando 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 3 de Abril**.

El plazo para esta entrega vence el día viernes 17 de abril a las 23:59 hrs. Cualquier entrega fuera de este plazo tendrá la nota mínima, sin derecho a corrección.