

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, pero ahora respaldados por datos provenientes tanto de los artistas como de las productoras de eventos (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 justifica 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 importarlos¹. Puedes usar una interfaz externa ² o desde la consola. ³

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.⁴ 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 que la entrega 1.**

Consultas grupo par:

1. Entregue los nombres y números de contacto de todas las productoras de eventos
2. Entregue el nombre de las productoras, junto a la cantidad de eventos que han producido
3. Dado el nombre de una productora, entregue los datos del ultimo evento que ha producido
4. Dada una productora, liste a los artistas con los que ha trabajado dicha productora
5. Dado un evento, entregue la suma total de ingresos por ventas de entradas de dicho evento
6. Liste todos los eventos, junto al numero total de artistas que se presentaran en cada uno
7. Muestre el nombre del evento que tiene la mayor cantidad de entradas vendidas

Consultas grupo impar:

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

²Como DataGrip.

³Como se explica en la Wiki.

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

1. Entregue un listado del nombre y teléfono de contacto de todos los artistas
2. Dado un artista, entregue el número de entradas de cortesía que ha entregado
3. Dado un artista, entregue los datos de su ultimo tour (el más reciente)
4. Dado un tour, liste los países que serán visitados en dicho tour
5. Dado un artista, liste todas las productoras con las que ha trabajado dicho artista
6. Dado un artista, liste todos los hoteles en los que se ha hospedado y cuantas veces se ha hospedado en cada uno (con códigos de reserva distinto, no cantidad de noches)
7. Muestre al artista que ha entregado la mayor cantidad de entradas de cortesía

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 “Depeche Mode”, “depeche mode”, “DEPECHE mode” o “Depech mode” 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.

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**⁵ 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 una comuna, el input debe funcionar solamente escribiendo o seleccionando el nombre o identificador de esa comuna.

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 ítem anterior y llamarse `index.php`.

⁵Por 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 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 ⁶. 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.
- 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, 30 de septiembre.

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. ⁷

⁶Como se indica en la Wiki del curso

⁷Un 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 30 de septiembre**.

El plazo para esta entrega vence el día **viernes 21 de octubre a las 23:59 hrs.**