

Entrega 3: Integración de datos de distintas fuentes

1. Descripción General

El objetivo de esta entrega es unir los datos de un grupo par y un grupo impar, para armar un sistema integrado de venta de muebles, y usar SQL avanzado y procedimientos almacenados para ayudar a la gestión automática de datos.

En esta fase aprenderán a implementar una aplicación que requiere consumir dos bases de datos distintas. También conocerán de primera mano los problemas de manejo de datos que ocurren al integrar bases de datos en el mundo real y los problemas al recibir bases de datos creadas por otra persona. Los grupos deberán afrontar una serie de desafíos ligados a estos problemas.

2. Detalles Académicos

Se deberá juntar un grupo **par** con uno **impar**. Los grupos que deseen trabajar juntos podrán inscribirse en este **formulario** hasta el día 9 de Junio, de lo contrario serán asignadas al azar. Las parejas de grupos definitivas serán publicadas el día **Miércoles 14 de Junio**.

Ubicación de su entrega

1. La nueva versión de la app debe estar ubicada en el directorio **/home/grupoXX/Sites/**. Esto significa que el homepage debe ser **/home/grupoXX/Sites/index.php**. Usar cualquier otra ruta arriesga que el proyecto no sea corregido.
2. El procedimiento almacenado debe estar (o debe haber una copia) en la carpeta **/home/grupoXX/Entrega3/**

donde XX es el número del **MENOR** de los dos grupos unidos. Se asumirá también que la página principal de su sitio está ubicada en

`codd.ing.puc.cl/~grupoXX/index.php`

Importante: Esta entrega requiere una coordinación entre los grupos que no siempre es sencilla. Se recomienda planificar con anterioridad y no dejar todo el trabajo para la última semana.

3. Requerimientos

Su aplicación va a tener que integrar la base de datos de la empresa de muebles (grupos impar) con la base de datos de la empresa de logística (grupos par). Para trabajar en esta entrega **no se permite** migrar todo el esquema a la base de datos de uno de los grupos, sino que se deben manejar cada una en el servidor respectivo. La idea de esto es mantener dos conexiones abiertas.

Su página Web tiene que soportar las siguientes acciones y requerimientos. Lógicamente, todo lo realizado vía Web deberá verse reflejado en sus bases de datos.

3.1. Usuarios y Acceso

Sistema de usuarios: Su página debe manejar un sistema de usuarios, para lo que deben crear una tabla llamada **Usuarios** con esta información en el servidor del grupo **PAR**. Cada usuario tendrá un id, una contraseña, y puede ser de distintos tipos. Su aplicación soportará tres tipos de usuarios:

- *Admin*: que podrá revisar todos los datos, agregar productos nuevos a la oferta de la tienda, asignar vehículos a despachos, etc.
- *Cliente*: que puede revisar los productos que haya comprado, revisar el despacho asignado a sus compras, o generar una nueva compra.

Su aplicación necesita tener un botón designado **importar usuarios** que hará lo siguiente:

- Creará (si ya no existe) un usuario llamado **ADMIN** con la contraseña **admin** y de tipo *Admin*.
- Convertirá en usuarios de tipo *Cliente*, a todos los clientes del grupo par (si ya no existen en la tabla **Usuarios**). El id debería ser el id del cliente (de la tabla **Clientes** del grupo par), y le deberían asignar una contraseña al azar.

Importante: Esta funcionalidad se debe crear usando php o procedimientos almacenados (o ambos). La asignación de contraseñas no pueden hacer manualmente (ingresándolos a la tabla **Usuarios** previamente a ejecutar este import). Deben reportar el método con el que hicieron el import en el informe o en un readme para que lo podamos revisar.

Información personal y Login: Su página tendrá dos campos, usuario, y contraseña, donde los usuarios pueden hacer un login. Al hacer el login, y dependiendo del tipo de usuario, se permitirán distintas funciones especificadas abajo.

En resumen, su página inicial tendrá dos funcionalidades:

1. El botón para importar los usuarios;

2. Dos campos para que un usuario existente pueda hacer el login (probablemente necesitarán un botón extra para realizar el login mismo). Al realizar login con éxito, un usuario tendrá distintas opciones dependiendo de tipo de usuario.

3.2. Navegación

Dependiendo del tipo de usuario que realizó el login, se debe desplegar una página distinta, dependiendo de tipo de usuario.

Admin: Este usuario puede generar una oferta para ciertos productos en una tienda, o actualizar su stock.

Para esto, el admin debe ver un dropdown dónde puede seleccionar la región de la tienda, después de qué se le listarán todas las tiendas de esta región.

De allá, el admin puede seleccionar una tienda en particular. Al hacer esto, se les muestra otro dropdown dónde puede seleccionar la categoría del producto, y se le mostrarán todos los productos de esta categoría vendidos por la tienda seleccionada.

De allá, el admin puede seleccionar un producto, y tener dos opciones: (i) actualizar su stock; o (ii) armar una oferta para este producto. Las dos funcionalidades pueden realizar con un form con el botón al lado. Al ingresar los datos y presionar el botón, se deben actualizar las tablas correspondientes. Estas funcionalidades deben hacer ocupando procedimientos almacenados. Quiere decir, al hacer el click al botón, se ejecuta un procedimiento en PL/pgSQL, para actualizar las tablas que corresponden. Deben tratar el caso cuando el usuario no ingresa a los datos, pero sí presiona uno de los dos botones.

Al actualizar la información debe aparecer un mensaje informativo.

Cliente: Para este tipo de usuario deben visualizar su nombre, su dirección, y un listado de todas sus compras realizadas.

Adicionalmente, debe existir una sección dónde el usuario puede realizar una compra, y una sección dónde el usuario puede revisar una compra existente. Las secciones pueden estar en la misma página, o ser alcanzable a través de links en la misma página del usuario.

Revisar compra: Aquí el usuario puede visualizar todas sus compras, y acceder a una de ellas. Para acceder a la información de la compra, pueden tener links en el listado de todas compras realizadas a cada compra particular, o poner un form dónde se ingresa el id de la compra con un botón que al hacer click permite revisar la compra. Cómo mínimo, la información que se debe desplegar para la compra seleccionada es: id y nombre de cada producto comprado, precio de cada producto, número de cajas para cada producto, el precio total, y la fecha de despacho programada, si corresponde (si no existe esta información se considera que el cliente retiró los productos personalmente).

Nueva compra: En esta sección el usuario puede generar una compra nueva. Para esto, el usuario puede ingresar en un form el nombre del producto, y se le muestra un listado de productos con este nombre (la búsqueda debe ser case insensitive, y permitir que faltan uno o más caracteres al final de la palabra). Se muestran solo los primeros 10 productos que hacen match con el nombre (si hay 10 o más). Para cada producto se muestra el nombre completo, id, y el precio (aplicando ofertas). Nóten que un producto puede tener distintos precios en distintas tiendas. Deben reflejar esto en el listado mostrado al usuario. Seleccionando un producto, el usuario lo puede agregar a la compra.

Al agregar un producto, el usuario debe poder repetir el proceso con un producto distinto. Pueden asumir que en cada búsqueda se agrega un producto, pero que el mismo producto puede ser agregado varias veces.

Finalmente, para terminar la compra, existe un botón para seleccionar la fecha de despacho. Aquí se selecciona una fecha de despacho, y se calcula el costo. Si el usuario compró todos los productos en una misma tienda, el valor de despacho es igual a 2000. Para cada tienda adicional, el precio sube en 3000. Quiere decir, si el usuario compra tres productos en tres tiendas distintas, el precio del despacho será 8000.

Finalmentem al seleccionar la fecha de despacho se presenta un botón para finalizar la compra. Al presionar este botón, el usuario puede generar una compra con los productos seleccionados. Esta funcionalidad (finalizar la compra) deben hacer ocupando procedimientos almacenados. Quiere decir, al hacer el click al botón para generar la compra, se ejecuta un procedimiento en PL/pgSQL, para actualizar las tablas que corresponden.

Para guardar la información de los productos en una compra mientras el usuario los está seleccionando, pueden ocupar php o crear una tabla nueva. El el caso de crear una nueva tabla deben borrar tuplas de esta tabla al finalizar la compra.

Al finalizar una compra debe aparecer un mensaje informativo con la información de la compra.

3.3. Procedimiento Almacenado

Importante: la funcionalidad de generar la compra (el último botón descrito para la página de clientes), o la funcionalidad de actualizar stock/oferta (para el admin), deben ser ejecutadas por medio de un procedimiento almacenado en lenguaje **PL/pgSQL**: en php se debe invocar a ese procedimiento con los datos de entrada de la consulta (además de otros necesarios), y el procedimiento debe ejecutar y insertar las tuplas a las tablas de su base de datos. Por supuesto, otras funcionalidades también pueden ser logradas usando procedimientos almacenados si lo prefieren realizar de esta manera.

3.4. Creatividad

Funcionalidad Adicional: Debe agregar una funcionalidad adicional que no esté listada en el enunciado y que aporte valor a la aplicación. Por ejemplo, no se ha hablado de opción de generar una compra de más de un producto, o generar un despacho para la compra (con las tablas del grupo Par), etc.

4. Bonus

No se evaluará la calidad de la página, ni su atractivo visual; sin embargo, los ayudantes tendrán la facultad de ofrecer un bonus de 5 décimas extra en la nota de esta entrega a aquellos grupos que presenten una página con una navegación sobresaliente. Deberá notarse un trabajo adicional al de la entrega anterior.

5. Entrega

El plazo para esta entrega es el **Viernes 23 de Junio a las 23:59 horas**.

Debe entregar los siguientes items:

- **Aplicación Web:** Todos los archivos necesarios para el correcto funcionamiento de la aplicación en su carpeta Sites.
- **Readme.md o Reporte.pdf:** Donde indique cómo logearse en la aplicación (contraseñas), cómo corregir la funcionalidad adicional que implementaron y cómo corregir cada requerimiento de esta entrega si no es intuitivo desde la *homepage*. Adicionalmente, deberían especificar cómo asignaron contraseñas a los usuarios existentes, y cómo registraron los clientes como usuarios. Los ayudantes se reservan el derecho a descontar décimas en una entrega en la que se haya dificultado la corrección. Archivo ubicado en la carpeta Entrega3 (afuera de Sites).
- **Procedimiento Almacenado:** Archivo(s) con la definición de su procedimiento almacenado, en lenguaje **PL/pgSQL**. Ubicado junto al Readme en la carpeta Entrega3.

6. Evaluación de pares

Cómo aquí trabajarán en grupos, también existe una evaluación de pares. Esto puede afectar a su nota (y disminuirla) si sus pares encuentran qué no aportaron al proyecto. Para aprobar el proyecto, deben contar con una nota promedio de al menos 4 (de 7) entre todos sus pares.