



GUÍA DE TRABAJO VIRTUAL

ASIGNATURA: PROGRAMACIÓN 11

SEMANA DE TRABAJO: AGOSTO 9

Guía elaborada por: Diego Eddy Hurtado Quintero

METAS DE APRENDIZAJE / COMPETENCIAS A DESARROLLAR

- Comprender e implementar el Modelo Relacional de Bases de datos en un proyecto sencillo.

LECTURAS

LECTURA 1

MODELO RELACIONAL DE BASES DE DATOS

El modelo relacional de bases de datos se rige por algunas normas sencillas:

- Todos los datos se representan en forma de tablas (también llamadas "relaciones", ver nota anterior). Incluso los resultados de consultar otras tablas. La tabla es además la unidad de almacenamiento principal.
- Las **tablas** están compuestas por filas (o **registros**) y columnas (o **campos**) que almacenan cada uno de los registros (la información sobre una entidad concreta, considerados una unidad).
- Las filas y las columnas, en principio, carecen de orden a la hora de ser almacenadas. Aunque en la implementación del diseño físico de cada SGBD esto no suele ser así. Por ejemplo, en SQL Server si añadimos una clave de tipo "Clustered" a una tabla haremos que los datos se ordenen físicamente por el campo correspondiente.
- El orden de las columnas lo determina cada consulta (que se realizan usando SQL).
- Cada tabla debe poseer una clave primaria, esto es, un identificador único de cada registro compuesto por una o más columnas.
- Para establecer una relación entre dos tablas es necesario incluir, en forma de columna, en una de ellas la clave primaria de la otra. A esta columna se le llama clave externa. Ambos conceptos de clave son extremadamente importantes en el diseño de bases de datos.

Basándose en estos principios se diseñan las diferentes bases de datos relacionales, definiendo un diseño conceptual y un diseño lógico, que luego se implementa en el diseño físico usando para ello el **gestor de bases de datos** de nuestra elección (por ejemplo MySQL).

Por ejemplo, consideremos la conocida base de datos Northwind de Microsoft. Esta base de datos representa un sistema sencillo de gestión de pedidos para una empresa ficticia. Existen conceptos que hay que manejar como: proveedores, empleados, clientes, empresas de transporte, regiones geográficas, y por supuesto pedidos y productos.

El diseño conceptual de la base de datos para manejar toda esta información se puede ver en la siguiente figura, denominada diagrama Entidad/Relación o simplemente diagrama E-R:



GUÍA DE TRABAJO VIRTUAL

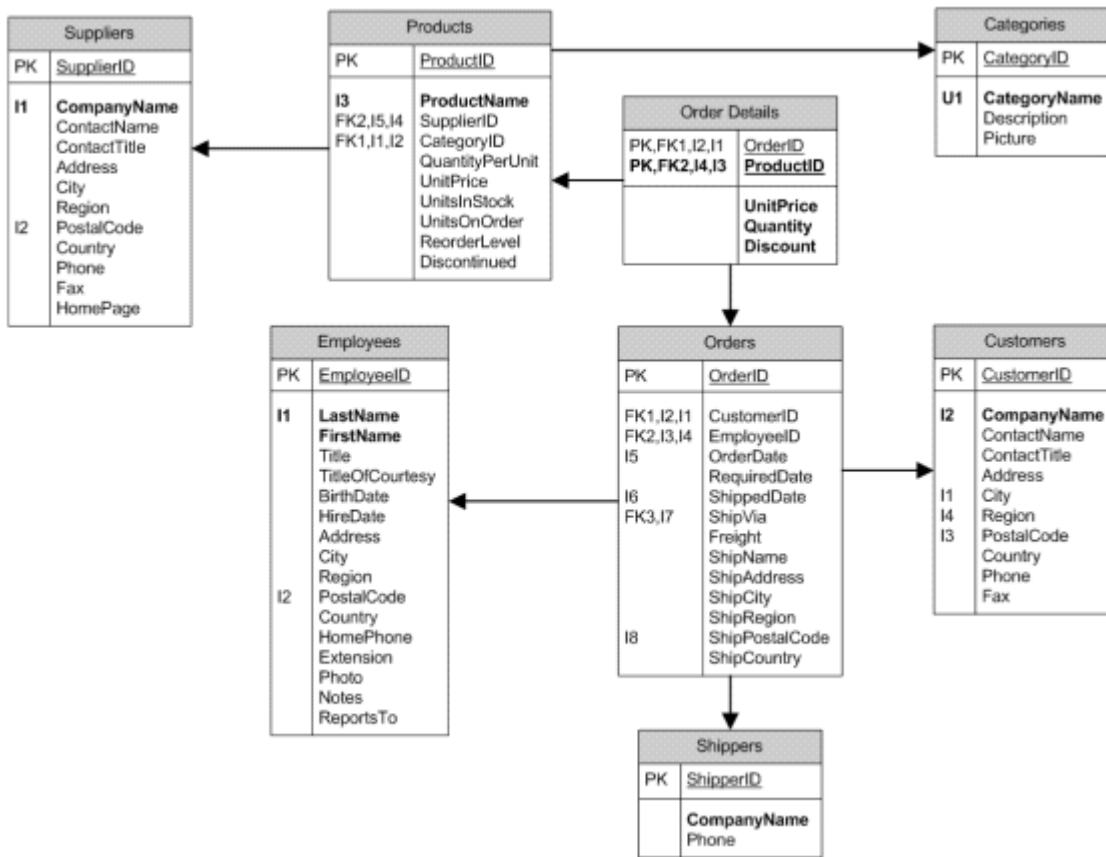


Imagen1. Modelo Lógico Base de datos Northwind_EF

Como vemos existen tablas para representar cada una de estas entidades del mundo real: Proveedores (Suppliers), Productos, Categorías de productos, Empleados, Clientes, Transportistas (Shippers), y Pedidos (Orders) con sus correspondientes líneas de detalle (Order Details). Además están relacionadas entre ellas de modo que, por ejemplo, un producto pertenece a una determinada categoría (se relacionan por el campo CategoryID) y un proveedor (SupplierID), y lo mismo con las demás tablas.

Cada tabla posee una serie de campos que representan valores que queremos almacenar para cada **entidad**. Por ejemplo, un producto posee los siguientes **atributos** que se traducen en los campos correspondientes para almacenar su información: Nombre (ProductName), Proveedor (SupplierID, que identifica al proveedor), Categoría a la que pertenece (CategoryID), Cantidad de producto por cada unidad a la venta (QuantityPerUnit), Precio unitario (UnitPrice), Unidades que quedan en stock (UnitsInStock), Unidades de ese producto que están actualmente en pedidos (UnitsOnOrder), qué cantidad debe haber para que se vuelva a solicitar más producto al proveedor (ReorderLevel) y si está descatalogado o no (Discontinued).

Los campos marcados con "PK" indican aquellos que son **claves primarias**, es decir, que identifican de manera única a cada entidad. Por ejemplo, ProductID es el identificador único del producto, que será por regla general un número entero que se va incrementando cada vez que introducimos un nuevo producto (1, 2, 3, etc..).



GUÍA DE TRABAJO VIRTUAL

Los campos marcados como "FK" son **claves foráneas** o claves externas. Indican campos que van a almacenar claves primarias de otras tablas de modo que se puedan relacionar con la tabla actual. Por ejemplo, en la tabla de productos el campo CategoryID está marcado como "FK" porque en él se guardará el identificador único de la categoría asociada al producto actual. En otras palabras: ese campo almacenará el valor de la clave primaria (PK) de la tabla de categorías que identifica a la categoría en la que está ese producto.

Los campos marcados con indicadores que empiezan por "I" (ej: "I1") se refieren a índices. Los índices generan información adicional para facilitar la localización más rápida de registros basándose en esos campos. Por ejemplo, en la tabla de empleados (Employees) existe un índice "I1" del que forman parte los campos Nombre y Apellidos (en negrita además porque serán también valores únicos) y que indica que se va a facilitar la locación de los clientes mediante esos datos. También tiene otro índice "I2" en el campo del código postal para localizar más rápidamente a todos los clientes de una determinada zona.

Los campos marcados con indicadores que empiezan con "U" (por ejemplo U1) se refieren a campo que deben ser únicos. Por ejemplo, en la tabla de categorías el nombre de ésta (CategoryName) debe ser único, es decir, no puede haber -lógicamente- dos categorías con el mismo nombre.

Como vemos, un diseño conceptual no es más que una representación formal y acotada de entidades que existen en el mundo real, así como de sus restricciones, y que están relacionadas con el dominio del problema que queremos resolver.

El Modelo lógico

Una vez tenemos claro el modelo E-R debemos traducirlo a un modelo lógico directamente en el propio sistema gestor de bases de datos (Oracle, MySQL, SQL Server...). Si hemos utilizado alguna herramienta profesional para crear el diagrama E-R, seguramente podremos generar automáticamente las instrucciones necesarias para crear la base de datos.

La mayoría de los generadores de diagramas E-R (por ejemplo Microsoft Visio) ofrecen la capacidad de exportar el modelo directamente a los SGBD más populares.

Entonces, todo este modelo conceptual se traduce en un modelo lógico que trasladaremos a la base de datos concreta que estemos utilizando y que generalmente será muy parecido. Por ejemplo, este es el mismo modelo anterior, mostrado ya como tablas en un diagrama de SQL Server:



GUÍA DE TRABAJO VIRTUAL

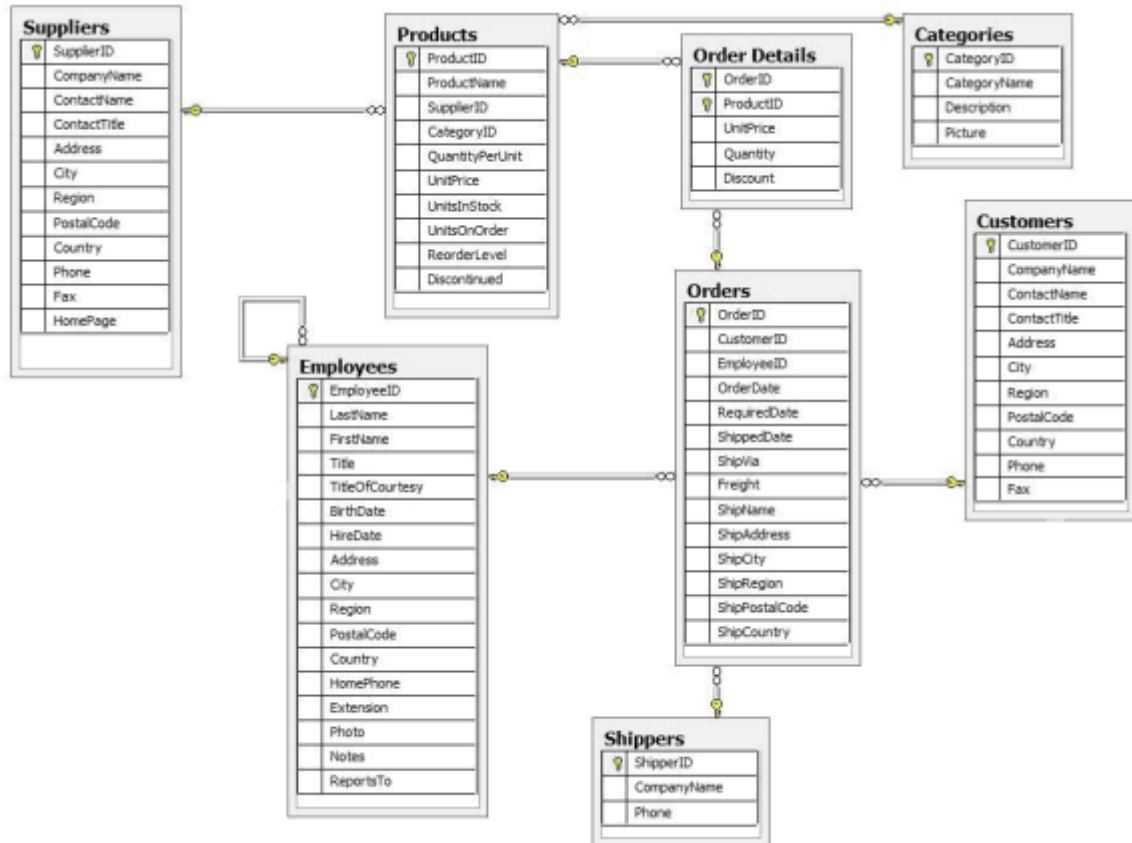


Imagen2. Modelo Relacional de Northwind

En este caso hemos creado cada tabla, una a una, siguiendo lo identificado en el diagrama E-R y estableciendo índices y demás elementos según las indicaciones de cada uno de los campos. Además hemos decidido el mejor tipo de datos que podemos aplicar a cada campo (texto, números, fechas... que se almacenan para cada registro).

Su representación gráfica en la base de datos es muy similar, sin embargo el modelo físico (cómo se almacena esto físicamente), puede variar mucho de un SGBD a otro y según la configuración que le demos.

En resumen

Según Thomas H. Grayson, un buen diseño de base de datos debe poseer siempre las siguientes cualidades, aunque algunas puede llegar a ser contradictorias entre sí:

- Reflejar la estructura del problema en el mundo real.
- Ser capaz de representar todos los datos esperados, incluso con el paso del tiempo.
- Evitar el almacenamiento de información redundante.
- Proporcionar un acceso eficaz a los datos.
- Mantener la integridad de los datos a lo largo del tiempo.
- Ser claro, coherente y de fácil comprensión.



GUÍA DE TRABAJO VIRTUAL

Como hemos visto el diseño de una base de datos parte de un problema real que queremos resolver y se traduce en una serie de modelos, conceptual, lógico y físico, que debemos implementar.

El primero, el diseño conceptual, es el que más tiempo nos va a llevar pues debemos pensar muy bien cómo vamos a representar las entidades del mundo real que queremos representar, qué datos almacenaremos, cómo los relacionaremos entre sí, etc...

El diseño lógico es mucho más sencillo puesto que no es más que pasar el diseño anterior a una base de datos concreta. De hecho muchas herramientas profesionales nos ofrecen la generación automática del modelo, por lo que suele ser muy rápido.

El diseño físico por regla general recae en la propia base de datos, a partir del diseño lógico, aunque si dominamos bien esa parte elegiremos cuidadosamente índices, restricciones o particiones así como configuraciones para determinar cómo se almacenará físicamente esa información, en qué orden, cómo se repartirá físicamente en el almacenamiento, etc...

Tomado y adaptado de: <https://www.campusmvp.es/recursos/post/Disenando-una-base-de-datos-en-el-modelo-relacional.aspx>

ACTIVIDADES

ACTIVIDAD 1 (TRABAJO EN EQUIPO)

Después de realizar la lectura, resuelvan las siguientes preguntas:

1. Defina usando fuentes bibliográficas confiables y escriba un ejemplo de cada una de las palabras resaltadas con rojo en la lectura.
2. Elabore el modelo relacional de bases de datos de su proyecto productivo, usando la herramienta MySQL Workbench.
3. Cree la base de datos a partir del modelo (Una vez el docente lo apruebe), usando PhpMyAdmin.

EVALUACIONES

EVALUACIÓN 1

1. Envíen el informe con las evidencias de todo el trabajo en un solo documento en formato PDF al whatsapp del docente.
2. Envía únicamente un estudiante por equipo de trabajo.

CRITERIOS DE EVALUACIÓN Y PLAZOS DE ENTREGA

1. Se evaluarán los criterios de puntualidad, originalidad, presentación, aplicación y apropiación.
2. El plazo máximo de entrega del trabajo será el viernes 20 de agosto de 2021.

INFORMACIÓN DE CONTACTO

DOCENTE

- Nombre: Diego Eddy Hurtado Quintero
- Grupos: 11° 1 - 11A
- Teléfono: 3042458020