

ALUMNO: FEDERICO SABATINI

TURNO: MAÑANA

PARCIAL N°1 DIAGRAMA ENTIDAD RELACION
PROYECTO FINAL LABORATORIO IV: "EL BUEN SABOR".

Definiciones:

¿Qué es una Base de Datos?

Es un Almacén que nos permite guardar grandes cantidades de información (datos) de forma organizada, para que luego podamos encontrar la misma y utilizarla fácilmente.

¿Qué es un Gestor de Base de Datos?

Se utiliza para poder interactuar con la Base de Datos a través del lenguaje SQL, podemos encontrar diferentes gestores, los mas reconocidos en la actualidad (MySQL, Oracle, PostgreSQL, MariaDB, SQLserver, etc).

¿Qué es SQL?

Lenguaje que nos permite comunicarnos con la base de datos a través de peticiones (Administrar una base de datos) su significado es Lenguaje de Consultas Estructurado.

¿Qué es un Query?

Comando SQL que genera una petición a la base de datos.

¿Qué es una Base de Datos Relacional?

Es uno de los tantos modelos existentes, donde las tablas se relacionan unas con otras permitiendo la relación entre los datos existentes en el modelo.

¿Qué es una Entidad?

Es la representación de Objetos de la vida real, se representa en la Base de Datos como una Tabla.

¿Qué son los Campos?

Son los Atributos que presentan dichos objetos, se presentan en la Base de Datos como campos columnas en la tabla (Entidad).

¿Qué son los Registros?

Son los datos que representan ese atributo (campo) se representan en la base de datos como filas de la tabla (Entidad).

¿Qué es la Clave Primaria?

Es un campo único dentro de la tabla (Entidad) que no se puede repetir es único, permite identificar cada registro de la tabla como único, facilita la relación entre tablas, solo puede existir una sola llave primaria, se identifica dentro del modelo como PK.

¿Qué es la Clave Foránea?

Es un campo que debe llevar el mismo nombre de la clave Primaria PK de la tabla que establece la relación, se identifica como FK, debe ser del mismo tipo, permite establecer la relación entre las tablas. La llave FK de una tabla (Entidad) es la llave primaria PK en la otra tabla (Entidad) de la relación.

¿Qué es la Clave Candidata?

Es otro campo de la tabla (Entidad) que también permite identificar un registro (fila) como único ej (DNI-CUIT) una tabla puede tener + de una clave candidata. Se presenta como unique.

¿Qué Tipos de Relaciones Existen entre tablas en el Modelo Relacional?

Para que los datos de las tablas se puedan seleccionar, encontrar y procesar de forma rápida y eficaz, es necesario que las tablas se encuentren relacionadas unas con otras.

Relación Uno a Uno (1:1):

Se da cuando un registro o fila de una tabla (Entidad) esta relacionado con otro único registro de la otra tabla (Entidad) y a su vez este último se relaciona con un solo registro de la primera tabla (Entidad).

Ej: Un Producto solo puede tener un Código de Barra, como un Código de Barra solo puede tener un producto.

Importante => En la relación uno a uno la clave FK se puede ubicar en cualquiera de las 2 tablas (Entidades).

Relación Uno a Muchos (1: N):

A cada registro de la tabla (Entidad A) se le puede asociar varios registros de la tabla (Entidad B) y cada registro de la tabla (Entidad B) se le puede asociar con solo un registro de la tabla (Entidad A)

Ej: Una categoría puede estar relaciona con mas de un Producto y este Producto puede solo tener una Categoría.

Importante => En las relaciones 1 a N la clave FK se debe ubicar en la tabla (Entidad) N muchos.

Relación Muchos a Muchos (N:M):

A cada registro de la tabla (Entidad A) se le puede asociar varios registros de la tabla (Entidad B) y cada registro de la tabla (Entidad B) se le puede asociar con varios registros de la tabla (Entidad A)

Ej: Un Producto puede ser traído por muchos Proveedores y un Proveedor puede traer muchos Productos.

Importante => Cuando existe una relación N:M se debe crear una tercera tabla `UNION_PRODUCTO_PROVEEDOR` que contenga el `id_Union` (PK), `id_Producto` (FK), `id_Proveedor` (FK).

¿Qué es un CRUD?

Son las operaciones principales con las que se puede interactuar con una Base de datos independientemente del modelo utilizado.

CREATE => Insertar datos en la base.

READ => Obtener buscar datos de la base ONE-ALL.

UPDATE => Actualizar los datos en la base.

DELETE => Eliminar los datos de la base.

¿Qué es una Transacción en la Base de Datos?

Son unidades o secuencias de trabajo realizadas de forma ordenada y separada en una base de datos. Normalmente representan cualquier cambio en la base de datos.

Una transacción es una propagación de uno o más cambios en la base de datos, ya sea cuando se crea, se modifica o se elimina un registro.

En la práctica suele consistir en la agrupación de consultas SQL y su ejecución como parte de una transacción.

Las transacciones siguen cuatro propiedades básicas, bajo el acrónimo ACID (Atomicity, Consistency, Isolation, Durability):

Atomicidad: aseguran que todas las operaciones dentro de la secuencia de trabajo se completen satisfactoriamente. Si no es así, la transacción se abandona en el punto del error y las operaciones previas retroceden a su estado inicial.

Consistencia: aseguran que la base de datos cambie estados en una transacción exitosa.

Aislamiento: permiten que las operaciones sean aisladas y transparentes unas de otras.

Durabilidad: aseguran que el resultado o efecto de una transacción completada permanezca en caso de error del sistema.

Existen tres comandos básicos de control en las transacciones SQL:

COMMIT. Para guardar los cambios.

ROLLBACK. Para abandonar la transacción y deshacer los cambios que se hubieran hecho en la transacción.

SAVEPOINT. Crea checkpoints, puntos concretos en la transacción donde poder deshacer la transacción hasta esos puntos.

Los comandos de control de transacciones se usan sólo con INSERT, DELETE y UPDATE. No pueden utilizarse creando tablas o vaciándolas porque las operaciones se guardan automáticamente en la base de datos.

¿Cómo se representa una Relación de Herencia en una Base de Datos?

La clase Padre que estaría representada por una tabla (Entidad) presenta sus campos atributos propios, estos serían heredados por la clase hija tabla (Entidad) con sus propios atributos (campos), la herencia se representa por una relación 1:1 donde la tabla (Entidad) que representa la clase hija, recibe la FK.

¿Qué es un Diagrama Entidad Relación?

Permite establecer a través de un lenguaje grafico definido de formas geométricas y flechas el diseño de una Base de Datos, Planificación de la Estructura.

Entidad: se representa con un cuadrado => en su interior el nombre de la entidad representada.

Atributo => se representa como un circulo => en su interior los nombres, el PK debe ir subrayado.

Relaciones => se representan a través de un Rombo, donde se describe en su interior la acción ej Cursar, y se describe el sentido a través de flechas.

Cardinalidad => se coloca en una esquina del cuadrado Entidad y puede ser representada con 1 o N.

¿Qué es la Normalización de una Base de Datos?

Se aplica una serie de Reglas al Diseño de la Base de Datos (Formas Normales) el objetivo del mismo es evitar la redundancia de datos innecesarios protegiendo la integridad de los mismos y evitando inconsistencias en la BD (Facilitando las Operaciones posteriores).

Primera Forma Normal (1FN) o Primera Regla:

Cada columna (Atributo) campo puede contener un único valor.

No pueden ser valores null (nulos).

No deben existir campos repetidos, grupos ordenados.

La tabla (Entidad) debe contener un único campo PK.

Segunda Forma Normal (2FN) o Segunda Regla:

Debe cumplir sin excepción la 1FN.

Cualquier campo (campo) no clave debe ser dependiente de la clave PK.

Si se utilizan en el diseño de la BD tablas (Entidad) con PK simples, se puede saltar esta regla o forma.

Tercera Forma Normal (3FN) o Tercera Regla:

Debe cumplir sin excepción la 2FN.

Ningún campo (columna) no clave, dependerá de ningún otro campo no clave.

Cuando el Diseño de la Base de Datos da cumplimiento de forma correcta a la 3FN (Tercer Forma Normal), podemos garantizar que estarán libres de anomalías, evitando repetición e inconsistencia de datos.

Modelo Entidad Relación Proyecto Final

Configuración del Modelo Entidades (Tablas) y Atributos (Campos):

Referencias:

COLOR: Entidad

COLOR: Clave Primaria PK

COLOR: Clave Foránea FK

COLOR: Clave Candidata

Cliente: **idCliente (INT-PK Primary Key-not null- autoIncrement)**, nombre (varchar-not null), apellido (varchar-not null), dni (varchar-not null- unique-Clave Candidata), fechaNacimiento (date-not null), telefono (varchar-not null), email (varchar-not null), fechaAlta (date-not null), fechaBaja (date-null- default '0000-00-00'), estado (varchar-null, default 'activo').

Usuario: **idUsuario (INT-PK Primary Key-not null- autoIncrement)**, usuario (varchar-not null- unique-Clave Candidata), contraseña (varchar-not null), rol (varchar-not null), fechaAlta (date-not null), fechaBaja (date-null- default '0000-00-00'), estado (varchar-null, default 'activo'), **idCliente (INT – FK Foreign Key-not null)**.

Domicilio: idDomicilio (INT-PK Primary Key-not null- autoIncrement), calle (varchar-not null), numero (varchar-not null), localidad (varchar-not null), fechaAlta (date-not null), fechaBaja (date-null- default '0000-00-00'), estado (varchar-null, default 'activo'), idCliente (INT – FK Foreign Key-not null).

Pedido: idPedido (INT-PK Primary Key-not null- autoIncrement), codigo (INT-not null- unique-Clave Candidata), horaEstimadaFin (date-not null), tipoEnvio (INT-not null), fechaAlta (date-not null), fechaBaja (date-null- default '0000-00-00'), estado (varchar-null, default 'activo'), idCliente (INT – FK Foreign Key-not null), idDomicilio (INT – FK Foreign Key-not null).

#SE CREAL LA TABLA UNION (DETALLE_PEDIDO) DE LA RELACION N:M ENTRE ENTIDAD ARTICULO_MANUFACTURADO Y ENTIDAD PEDIDO.

DetallePedido: idDetallePedido (INT-PK Primary Key-not null- autoIncrement), cantidad (INT- not null), subTotal(double-not null), idPedido (INT – FK Foreign Key-not null), idArticuloManufacturado (INT – FK Foreign Key-not null).

Factura: idFactura (INT-PK Primary Key-not null- autoIncrement), codigo (varchar-not null- unique-Clave Candidata), montoDescuento (double-not null), formaPago (varchar-not null), fechaAlta (date-not null), fechaBaja (date-null- default '0000-00-00'), estado (varchar-null, default 'activo'), idPedido (INT – FK Foreign Key-not null).

#SE CREAL LA TABLA UNION (DETALLE_FACTURA) DE LA RELACION N:M ENTRE ENTIDAD ARTICULO_MANUFACTURADO Y ENTIDAD FACTURA:

DetalleFactura: idDetalleFactura (INT-PK Primary Key-not null- autoIncrement), cantidad (INT- not null), subTotal(double-not null), idFactura (INT – FK Foreign Key-not null), idArticuloManufacturado (INT – FK Foreign Key-not null).

MercadoPago: **idMercado (INT-PK Primary Key-not null, autoIncrement)**, **codigo (varchar-not null- unique-Clave Candidata)**, **fechaAprobacion (date-not null)**, **nroTarjeta (varchar-not null)**, **metodoPago (varchar, not null)**, **estadoPago (varchar, not null)**, **fechaAlta (date-not null)**, **fechaBaja (date-null- default '0000-00-00')**, **estado (varchar-null, default 'activo')**, **idPedido (INT – FK Foreign Key-not null)**.

ArticuloManufacturado: **idArtManufacturado (INT-PK Primary Key-not null, autoIncrement)** **tiempoEstimado (INT – not null)**, **denominacion (varchar-not null)**, **precioVenta (double-not null)**, **imagen (varchar-not null)**, **fechaAlta (date-not null)**, **fechaBaja (date-null- default '0000-00-00')**, **estado (varchar-null, default 'activo')** **idRubroGeneral (INT – FK Foreign Key-not null)**.

#SE CREAL LA TABLA UNION (ARTICULO_MANUFACTURADO_DETALLE) DE LA RELACION N:M ENTRE ENTIDAD ARTICULO_MANUFACTURADO Y ENTIDAD ARTICULO_INSUMO:

ArticuloManufacturadoDetalle: **idArtManufacturadoDetalle (INT-PK Primary Key-not null)**, **cantidad (double-not null)**, **idArtManufacturado (INT – FK Foreign Key-not null)**, **idArtInsumo (INT – FK Foreign Key-not null)**.

RubroGeneral: **idRubroGeneral (INT-PK Primary Key-not null, autoIncrement)**, **fechaAlta (date-not null)**, **fechaBaja (date-null- default '0000-00-00')**, **estado (varchar-null, default 'activo')**, **denominacion (varchar-not null)**.

ArticuloInsumo: **idArtInsumo (INT-PK Primary Key-not null, autoIncrement)**, **denominacion (varchar-not null)**, **precioCompra (double-not null)**, **precioVenta (double-not null)**, **stockActual (double-not null)**, **stockMinimo (double-not null)**, **unidadMedida (varchar-not null)**, **fechaAlta (date-not null)**, **fechaBaja (date-null- default '0000-00-00')**, **estado (varchar-null, default 'activo')**, **idRubroArticulo (INT – FK Foreign Key-not null)**.

RubroArticulo: idRubroArticulo (INT-PK Primary Key-not null, autoIncrement), denominacion (varchar-not null), fechaAlta (date-not null), fechaBaja (date-null- default '0000-00-00'), estado (varchar-null, default 'activo').