

Modelización de Datos

Diseño de bases de datos

© JMA 2010. All rights reserved

INTRODUCCIÓN

© JMA 2010. All rights reserved

Introducción

- El modelado de datos es un proceso que se utiliza para definir y analizar las necesidades de datos necesarios para soportar los procesos de negocio en los ámbitos correspondientes de los sistemas de información de las organizaciones.
- Las técnicas y metodologías de modelado de datos se utilizan para modelar los datos según un estándar, de manera consistente y predecible para ser gestionados como un recurso.
- El modelado de datos se puede realizar en diferentes tipos de proyectos y en múltiples fases de los proyectos.
- Los modelos de datos son progresivos; no hay un modelo de datos final para una empresa o una aplicación.
- En su lugar, un modelo de datos debe ser considerado como un documento vivo que va a cambiar en respuesta al modelo de negocio cambiante.

© JMA 2010. All rights reserved

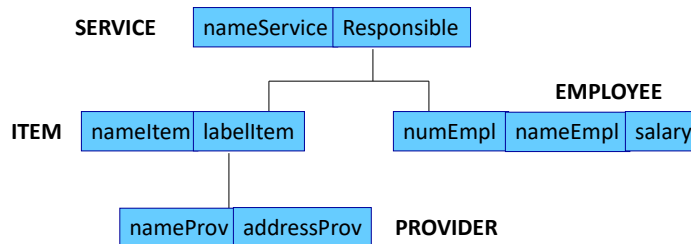
Modelos

- Se han propuesto diferentes modelos de bases de datos, por ejemplo:
 - Jerárquico
 - Red
 - Relacional
- Los modelos jerárquico y en red fueron los primeros modelos

© JMA 2010. All rights reserved

Modelo Jerárquico

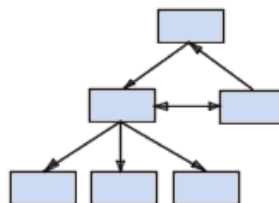
- 1965, es el primer modelo
- Modelo por composición, un dato está compuesto por otros datos que a su vez pueden estar compuestos por nuevos datos creando una estructura arborescente
- Información sólo es accesible a través de un punto de entrada único
- Obliga a tener información redundante
- Dificulta el mantenimiento de la coherencia de la BD



© JMA 2010. All rights reserved

Modelo en Red

- Es una variación del modelo jerárquico
- Un dato puede contener una referencia a un dato de otro nodo
- Los datos se organizan de acuerdo con un grafo
- Toda la información puede estar asociada con otra información
- Más rico que el modelo jerárquico, pero muy difícil de manejar



© JMA 2010. All rights reserved

Modelo Relacional

- Propuesto en 1970 por Codd
- Los datos se almacenan en tablas de dos dimensiones (filas y columnas)
- La manipulación de datos se realiza según la teoría matemática de las relaciones
- La ventaja más importante del modelo relacional es su simplicidad: sólo una estructura (relación o tabla)
- Una relación puede ser representada simplemente con una tabla relacional
- Los usuarios no necesitan tener un conocimiento de la posición física de los datos, la navegación por la base de datos se realiza de forma automática
- Lenguaje de acceso a los datos muy sencillo (SQL)

© JMA 2010. All rights reserved

Fases de la modelización

- **Modelo Conceptual** \Rightarrow **Información**
 - Define qué información hace falta para nuestro sistema
 - Estudio del problema
- **Modelo Lógico** \Rightarrow **Tablas Lógicas**
 - Define qué tablas lógicas necesitamos para almacenar la información
 - Estudio de la solución
- **Modelo Físico** \Rightarrow **Implementación**
 - Define cómo implementar el modelo lógico en un determinado Gestor de Bases de Datos Relacional

© JMA 2010. All rights reserved

Técnicas y utilidades

- Diseño del modelo Conceptual
 - ERD: Diagrama Entidad-Relación
 - ERM: Matriz entidades-roles
 - DD: Diccionario de datos
- Diseño del modelo Lógico
 - Paso a tablas
 - TD: Diagrama de tablas
 - ORM: Matriz objetos-roles
 - DD: Diccionario de datos
 - Normalización
- Diseño del modelo Físico
 - SQL

© JMA 2010. All rights reserved

Herramientas de diseño automatizado

- Utilizando herramientas de diseño automatizado ayuda a aumentar la productividad y ayuda a transformar un diseño conceptual de la base de datos en un modelo lógico y genera base de datos física y la documentación elaborada.
- Aunque las herramientas pueden generar los modelos lógicos desde los modelos conceptuales, algunos diseñadores prefieren crearlos manualmente.
- Puede asegurar calidad del proceso cuando se crea manualmente el diseño.
- Sin embargo, podrá beneficiarse de ambos enfoques usando una herramienta de diseño automatizado para generar el modelo lógico y revisar manualmente el modelo.

© JMA 2010. All rights reserved

DISEÑO DEL MODELO CONCEPTUAL

© JMA 2010. All rights reserved

En que consiste

- Proceso de describir los datos, las relaciones entre los datos y las restricciones sobre los datos.
 - Reunir todos los datos esenciales necesarios y entender cómo se relacionan los datos entre si
 - El foco está en los datos, en lugar de los procesos.
 - El resultado del diseño conceptual de la base de datos es un Modelo Conceptual de Datos (+ el Diccionario de datos)
-

© JMA 2010. All rights reserved

Recopilación de la información

- Dos perspectivas:
 - Top-down
 - El modelo de datos se deriva de una comprensión profunda de la empresa.
 - Bottom-up
 - El modelo de datos se deriva mediante la revisión de las especificaciones y documentos de negocios.

© JMA 2010. All rights reserved

El Modelo E-R

- El modelo entidad-relación (ER) fue originalmente propuesto por Peter Chen en 1976 como una forma de unificar la red y vistas de base de datos relacional.
- En pocas palabras, el modelo ER es un modelo de datos conceptual que considera el mundo real como entidades y relaciones.
- Un componente básico del modelo es el diagrama de entidad-relación que representa visualmente los objetos de datos.

© JMA 2010. All rights reserved

Ventajas

- Los elementos utilizados en el modelo ER pueden transformarse fácilmente en tablas relacionales.
- Es simple y fácil de entender con un mínimo de entrenamiento. Por lo tanto, el modelo puede utilizarse por el diseñador de base de datos para comunicar el diseño para el usuario final.
- Además, el modelo puede utilizarse como un plan de diseño por el desarrollador de base de datos para implementar un modelo de datos en un software de gestión de base de datos específica.

© JMA 2010. All rights reserved

Objetivos

- Comprender los datos de una organización y del funcionamiento de la misma.
- Ilustrar cómo las relaciones entre entidades son definidas y refinadas.
- Saber cómo las relaciones se incorporan en el proceso de diseño de base de datos.
- Describir cómo los componentes ERD afectan al diseño e implementación de la base de datos.
- Controlar los posibles errores desde el principio, o al menos, percibir las deficiencias lo antes posible.
- Mejorar el mantenimiento.

© JMA 2010. All rights reserved

Cómo Modelizar Datos

- **Modelo de Datos**
 - Conjunto de conceptos, reglas y convenios que nos permiten describir las entidades, atributos, relaciones y restricciones en el mundo real
- **Conceptos Básicos del Modelo Conceptual**
 - Entidad
 - Relación
 - Grado de una relación
 - Cardinalidad
 - Ocurrencia de una relación
 - Dominio
 - Atributos

© JMA 2010. All rights reserved

Entidad

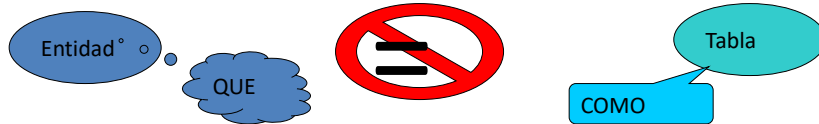
- **Definición**
 - Objeto, individuo o concepto acerca del cual necesitamos guardar información en el sistema
 - Ideas, acontecimientos, existencias... pueden llegar a ser ENTIDADES
- **Ejemplos:**
 - Cliente, Factura, Coche
- **Nombre**
 - Legible y conciso
 - Debe ser significativo respecto de lo que se guarde en la entidad
 - Sustantivo en singular
- **Tupla:** cada ocurrencia de una entidad

© JMA 2010. All rights reserved

Entidad

- **Cómo se dibuja**
 - Nomenclatura de Martin, propuesta por Chen
 - Caja rectangular de esquinas cuadradas
- **Errores Típicos**
 - Considerar entidades sólo cosas tangibles
 - A los conceptos no se les suele tratar como entidades (p.e., Vuelos)
 - Considerar que una entidad es una tabla o fichero
 - Una tabla es una estructura de datos muy apropiada para guardar información

Cliente



© JMA 2010. All rights reserved

Tipos de entidades

- **Fuerte:**
 - su existencia no depende de ninguna otra entidad.
- **Débil:**
 - su existencia depende de alguna otra entidad (doble recuadro)
- **Renacida:**
 - Surge de relaciones muchos a muchos (rombo o rombo en caja)

© JMA 2010. All rights reserved

Atributos

- Definición
 - Cualquier propiedad o característica de una entidad o relación
- Ejemplos
 - Cliente
 - Nombre, Apellidos, Fecha Nacimiento, DNI
- Errores comunes
 - Poner atributos a entidades que no les correspondan
- Representación
 - Dentro de la entidad o fuera unidos por conectores

© JMA 2010. All rights reserved

Tipos de atributos

- Simples:
 - Compuesto de un único componente con existencia independiente
- Compuestos:
 - Compuesto por múltiples componentes, cada uno de ellos con una existencia independiente (dirección)
- Univaluados:
 - Contiene un único valor para cada tupla
- Multivaluados:
 - Puede contener múltiples valores para cada tupla
- Derivados:
 - valor calculado con el valor de otro atributo o conjunto de atributos.

© JMA 2010. All rights reserved

Reglas de negocio

- Una declaración que impone cierta forma de restricción de elementos dentro de la especificación de los datos para un dato en particular o en la relación con otros datos.
- Están basadas en la forma de la organización percibe y utiliza sus datos.
- Esta percepción deriva de la manera en que la organización funciona o realiza sus negocios.
- Como las reglas de negocio dependen de cada organización, varias organizaciones podrían utilizar reglas de negocio con el mismo concepto pero aplicadas de formas completamente diferentes

© JMA 2010. All rights reserved

Dominio

- Definición
 - El conjunto de valores permitidos para uno o más atributos
- Pueden consistir en:
 - Tipos de datos (genéricos), tamaños, rangos, validaciones, ...
- El dominio puede ser:
 - Finito.
 - Infinito.
 - Continuo.
 - Discreto, ...

© JMA 2010. All rights reserved

Identificador Único

- Definición
 - Atributo o conjunto de atributos que identifican única o unívocamente cualquier ocurrencia de una entidad
- Representación
 - Se subrayan los atributos o se preceden de *
- Una entidad puede tener más de un identificador único
 - En ese caso, nos quedaremos con el más utilizado (Identificador Único Primario)
 - El resto se considerarán Identificadores Únicos Secundarios o Alternativos

© JMA 2010. All rights reserved

Identificador Único Principal

- IUP: se selecciona sobre el identificador candidato:
 - Con el número mínimo de atributos
 - Con valores que tienen menor probabilidad de cambio
 - El más corto: menor número de caracteres o valor numérico máximo más pequeño
 - El más fácil de utilizar desde el punto de vista del usuario

© JMA 2010. All rights reserved

Relación

- Definición
 - Cualquier asociación o nexo de unión entre entidades
 - En una relación entre dos entidades las asociaciones que aparecerán son
 - Asociación de A hacia B
 - Asociación de B hacia A
 - Tiene la misma o más importancia que una entidad
- Debe ir nombrada
 - Con un verbo en voz activa, o una frase verbal corta
 - Se escribe encima de la entidad
 - Para saber si es correcto el nombre
 - Debe describir con la mayor precisión la relación
 - Debe poderse poner en voz pasiva

© JMA 2010. All rights reserved

Ocurrencia de una Relación

- Definición
 - Caso concreto y perfectamente distinguible de dicha relación
- Atributos de una Relación
 - En Chen y Merise, las relaciones pueden tener atributos
 - En Martin, los atributos de una relación se simulan como en el caso de las relaciones ternarias, mediante una entidad renacida o asociativa

© JMA 2010. All rights reserved

Grado de una Relación

- Definición
 - Número de entidades que interconecta dicha relación
- Tipos
 - Grado 1: Unarias, Recursivas, Reflexivas
 - Grado 2: Binarias
 - Grado 3: Ternarias
- Observaciones
 - Las relaciones ternarias o superiores sólo pueden pintarse como tales en Chen o Merise, no en Martin
 - En Martin se simulan mediante relaciones binarias, y haciendo uso de lo que se conoce como entidad renacida

© JMA 2010. All rights reserved

Cardinalidad de una Relación

- Definición:
 - Número de tuplas implicadas en la relación
- Tipos:
 - Uno a uno (1:1)
 - Uno a muchos (1:*)
 - Muchos a muchos (*:*)
- Existen dos tipos de cardinalidades:
 - Cardinalidad mínima: Determina la obligatoriedad u opcionalidad de la relación (Valores posibles: 0 ó 1)
 - Cardinalidad Máxima: Determina con cuántos elementos como máximo está relacionado una ocurrencia (Valores posibles: 1 ó n)

© JMA 2010. All rights reserved

Normalización

- Proceso que realiza una optimización del diseño de la B. D. para construir una B. D. Relacional
- Proceso que permite:
 - Reducir o anular el nivel de redundancia del Modelo de Datos
 - Ajustar el Modelo a una serie de estándares
- Un modelo de datos puede encontrarse en diferentes estados de normalización

© JMA 2010. All rights reserved

Roles

- Un rol es un conjunto de comportamientos conectados, derechos y obligaciones conceptualizados para los diferentes actores de una situación social.
- Actividad:
 - Q: Query
 - C: Create
 - U: Update
 - D: Delete

© JMA 2010. All rights reserved

Matriz Entidades-Roles

| | Role 1 | Role 2 | ... | Role n |
|----------|--------|--------|-----|--------|
| Entity 1 | Q | QCU | | |
| Entity 2 | QCUD | - | | |
| Entity 3 | - | QC | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Entity n | | | | |

Q: Query, C: Create, U: Update, D: Delete, -: None

© JMA 2010. All rights reserved

¿Cómo evaluar un modelo de datos?

- Un modelo de datos de calidad debe cumplir lo siguiente:
 - Preciso y exhaustivo
 - Sin redundancia
 - Aplicación de las reglas de negocio
 - Reutilización de datos
 - Estabilidad y flexibilidad
 - Comunicación efectividad
 - Sencillez

© JMA 2010. All rights reserved

Normalización

- La normalización es una técnica para producir un conjunto de relaciones adecuadas que apoyen las necesidades de datos de una empresa y eliminar la redundancia.
- Principalmente es una herramienta para validar y mejorar un diseño lógico de manera que satisfaga ciertas restricciones que eviten la duplicación innecesaria de datos.
- El proceso descompone las relaciones con anomalías para producir relaciones más pequeñas y bien estructuradas.

© JMA 2010. All rights reserved

K Ibérica. Pedro J. Encabo

Relaciones bien estructuradas

- Una relación bien estructurada contiene una redundancia de datos mínima y permite a los usuarios insertar, eliminar y actualizar filas sin causar inconsistencias de datos.
- El objetivo es evitar las siguientes anomalías:
 - Redundancia: los mismos datos en diferentes lugares con diferentes valores
 - Inserción: añadir nuevas filas fuerza al usuario a crear datos duplicados.
 - Supresión: eliminar filas puede causar una pérdida de datos que se necesitaría para otras filas futuras.
 - Modificación: cambiar los datos en una fila obliga a cambios en otras filas debido a la duplicación.

© JMA 2010. All rights reserved

Ejemplo

EMPLOYEE2

| EmpID | Name | DeptName | Salary | CourseTitle | DateCompleted |
|-------|------------------|--------------|--------|--------------|---------------|
| 100 | Margaret Simpson | Marketing | 48,000 | SPSS | 6/19/201X |
| 100 | Margaret Simpson | Marketing | 48,000 | Surveys | 10/7/201X |
| 140 | Alan Beeton | Accounting | 52,000 | Tax Acc | 12/8/201X |
| 110 | Chris Lucero | Info Systems | 43,000 | Visual Basic | 1/12/201X |
| 110 | Chris Lucero | Info Systems | 43,000 | C++ | 4/22/201X |
| 190 | Lorenzo Davis | Finance | 55,000 | | |
| 150 | Susan Martin | Marketing | 42,000 | SPSS | 6/19/201X |
| 150 | Susan Martin | Marketing | 42,000 | Java | 8/12/201X |

Pregunta: ¿Es esta una relación?

Respuesta: Si, filas únicas y sin atributos multivalor

Pregunta: ¿Cuál es la clave principal?

Respuesta: Si, compuesta por EmpID, CourseTitle

© JMA 2010. All rights reserved

Anomalías

- Inserción:
 - no puede añadir un nuevo empleado si el empleado no ha recibido clase
- Supresión:
 - si quitamos empleado 140, perdemos la información sobre la existencia de una clase de Tax Acc
- Modificación:
 - al dar un aumento de sueldo al empleado 100 nos fuerza a modificar varios registros
- ¿Por qué existen estas anomalías?
 - Debido a que hay dos conceptos (tipos de entidades) en ésta relación. Esto da lugar a la duplicación de datos y una dependencia innecesaria entre las entidades.

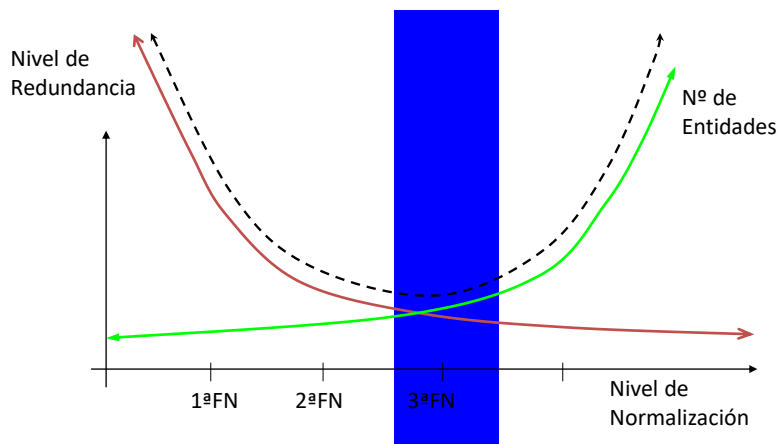
© JMA 2010. All rights reserved

Proceso de Normalización

- **Objetivo:**
 - Reducir o anular el nivel de redundancia de datos del modelo.
- **Como:**
 - Aplicando a cada entidad una serie de reglas consecutivas.
 - El incumplimiento de una regla puede derivar en la división de una entidad en varias entidades en cuyo caso es necesario iniciar el proceso de nuevo.

© JMA 2010. All rights reserved

Normalización



© JMA 2010. All rights reserved

Ejemplo

Una empresa quiere guardar la información de sus proveedores y de las piezas que le suministran anualmente.

Los proveedores suministran varias piezas y éstas pueden ser suministradas por varios proveedores.

Un proveedor sólo está ubicado o trabaja en una ciudad.

© JMA 2010. All rights reserved

Ejemplo

PR01

Nombre Proveedor: *Pedro*

Ciudad del Proveedor: *Madrid*

Cod. Ciudad: *28*

| Cod Pieza | Descripción | Cantidad |
|-----------|-------------|----------|
| PI01 | Tornillo | 2.000 |
| PI02 | Arandela | 3.000 |
| PI03 | Alicate | 200 |


PR02

Nombre Proveedor: *Javi*

Ciudad del Proveedor: *Madrid*

Cod. Ciudad: *28*

| Cod Pieza | Descripción | Cantidad |
|-----------|-------------|----------|
| PI01 | Tornillo | 100 |
| PI03 | Alicate | 50 |



Fichero de Proveedores

© JMA 2010. All rights reserved

Ejemplo

- Después de un estudio se determina que la información que se precisa guardar es:
 - Nombre del proveedor
 - Código del proveedor
 - Nombre de ciudad
 - Código de ciudad
 - Código de pieza
 - Nombre de pieza
 - Cantidad suministrada
- Se decide guardar toda esta información en la entidad “*Proveedor*”

© JMA 2010. All rights reserved

Ejemplo

proveedor

–Código del proveedor (IUP)

- Nombre del proveedor
- Código de ciudad
- Nombre de ciudad
- Código de pieza
- Nombre de pieza
- Cantidad suministrada

¿Cómo son las ocurrencias de la entidad proveedor?

© JMA 2010. All rights reserved

1ª Forma Normal

- Se dice que una entidad está en 1ª FN cuando:

1º. Tiene identificador único principal (IUP).

2º. No existen grupos repetitivos de datos.

© JMA 2010. All rights reserved

Normalización

proveedor

–Código del proveedor (IUP)

- Nombre del proveedor
- Código de ciudad
- Nombre de ciudad
- Código de pieza
- Nombre de pieza
- Cantidad suministrada

¿La entidad proveedor esta en 1ª F.N.?

¿Tiene identificador único principal (IUP)?

¿ Existen grupos repetitivos de datos ?

© JMA 2010. All rights reserved

| Código del proveedor | Nombre del proveedor | Código de ciudad | Nombre de ciudad | Código de pieza | Nombre de pieza | Cantidad suministrada |
|----------------------|----------------------|------------------|------------------|-----------------|-----------------|-----------------------|
| PR01 | Pedro | 28 | Madrid | PI01 | Tornillo | 2.000 |
| | | | | PI02 | Arandela | 3.000 |
| | | | | PI03 | Alicate | 200 |
| PR02 | Javi | 28 | Madrid | PI01 | Tornillo | 100 |
| | | | | PI03 | Alicate | 50 |

© JMA 2010. All rights reserved

| Código del proveedor | Nombre del proveedor | Código de ciudad | Nombre de ciudad | Código de pieza | Nombre de pieza | Cantidad suministrada |
|----------------------|----------------------|------------------|------------------|-----------------|-----------------|-----------------------|
| PR01 | Pedro | 28 | Madrid | PI01 | Tornillo | 2.000 |
| PR01 | Pedro | 28 | Madrid | PI02 | Arandela | 3.000 |
| PR01 | Pedro | 28 | Madrid | PI03 | Alicate | 200 |
| PR02 | Javi | 28 | Madrid | PI01 | Tornillo | 100 |
| PR02 | Javi | 28 | Madrid | PI03 | Alicate | 50 |

© JMA 2010. All rights reserved

Entidad Sin Normalizar

PR01


Nombre Proveedor: *Pedro*
 Ciudad del Proveedor: *Madrid*
 Cod. Ciudad: *28*

| Cod Pieza | Descripción | Cantidad |
|-----------|-------------|----------|
| PI01 | Tornillo | 2.000 |
| PI02 | Arandela | 3.000 |
| PI03 | Alicate | 200 |

PR02

Nombre Proveedor: *Javi*
 Ciudad del Proveedor: *Madrid*
 Cod. Ciudad: *28*

| Cod Pieza | Descripción | Cantidad |
|-----------|-------------|----------|
| PI01 | Tornillo | 100 |
| PI03 | Alicate | 50 |



Fichero de Proveedores

© JMA 2010. All rights reserved

Normalización

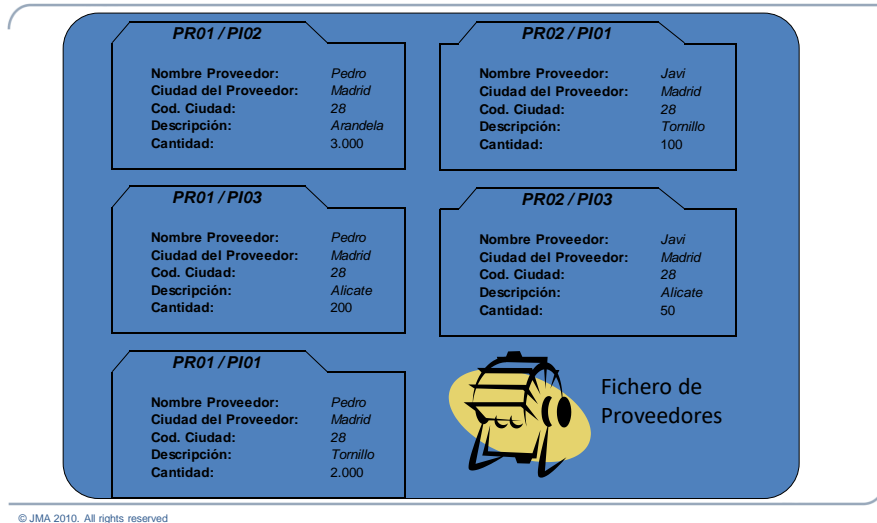
proveedor

- Código del proveedor (IUP)
- Código de pieza
- Nombre del proveedor
- Código de ciudad
- Nombre de ciudad
- Nombre de pieza
- Cantidad suministrada

¿Cómo son ahora las ocurrencias de la entidad *proveedor*?

© JMA 2010. All rights reserved

1ª Forma Normal

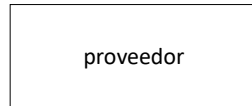


2ª Forma Normal

- Se dice que una entidad está en 2ª FN cuando:

- Está en 1ª Forma Normal.
- Todo atributo que no forma parte del IUP depende, funcionalmente, de todo el IUP.

Normalización



–Código del proveedor (IUP)

–Código de pieza

–Nombre del proveedor

–Código de ciudad

–Nombre de ciudad

–Nombre de pieza

–Cantidad suministrada

Está en 1ª Forma Normal

¿ Todo atributo que no forma parte del IUP depende, funcionalmente, de todo el IUP?

© JMA 201

Normalización



–Código del proveedor (IUP)

–Nombre del proveedor

–Código de ciudad

–Nombre de ciudad

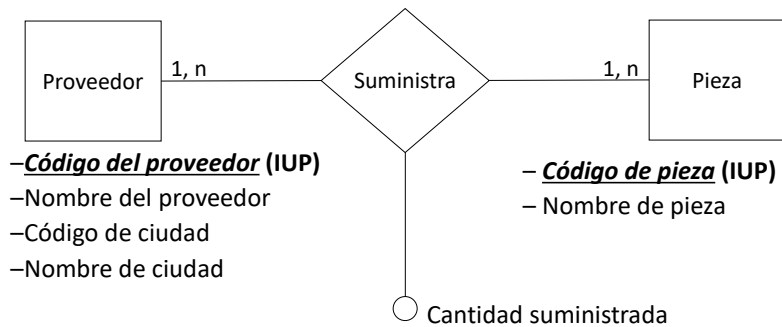
– Código de pieza (IUP)

– Nombre de pieza

– Cantidad suministrada

© JMA 2010. All rights reserved

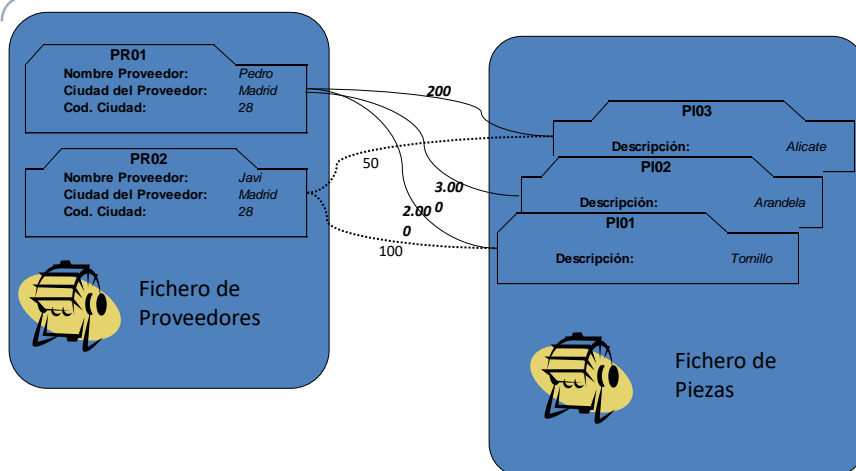
Normalización



¿Cómo son las ocurrencias ahora?

© JMA 2010. All rights reserved

2ª Forma Normal



© JMA 2010. All rights reserved

3ª Forma Normal

- Se dice que una entidad está en 3ª FN cuando:
 - 1º. Esta en 2ª Forma Normal.
 - 2º. No existen dependencias transitivas entre el IUP y ningún otro atributo que no forme parte del IUP

© JMA 2010. All rights reserved

Normalización

¿Está en 3ª FN?



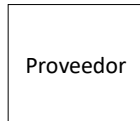
- Código de pieza (IUP)
- Nombre de pieza

Está en 3ª Forma Normal

© JMA 2010. All rights reserved

Normalización

¿Está en 3ª FN?



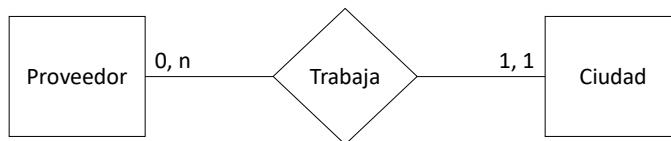
- Código del proveedor (IUP)
- Nombre del proveedor
- Código de ciudad
- Nombre de ciudad

No está en 3ª Forma Normal

Existen dependencias transitivas entre el Código de Proveedor, el código de ciudad y nombre de la ciudad.

© JMA 2010. All rights reserved

Normalización

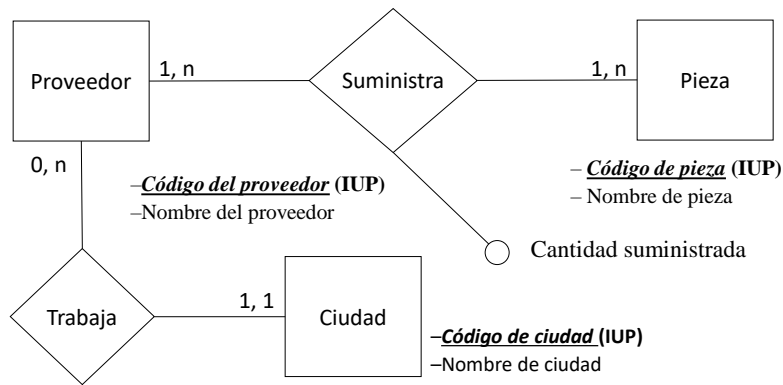


- Código del proveedor (IUP)
- Nombre del proveedor

- Código de ciudad (IUP)
- Nombre de ciudad

© JMA 2010. All rights reserved

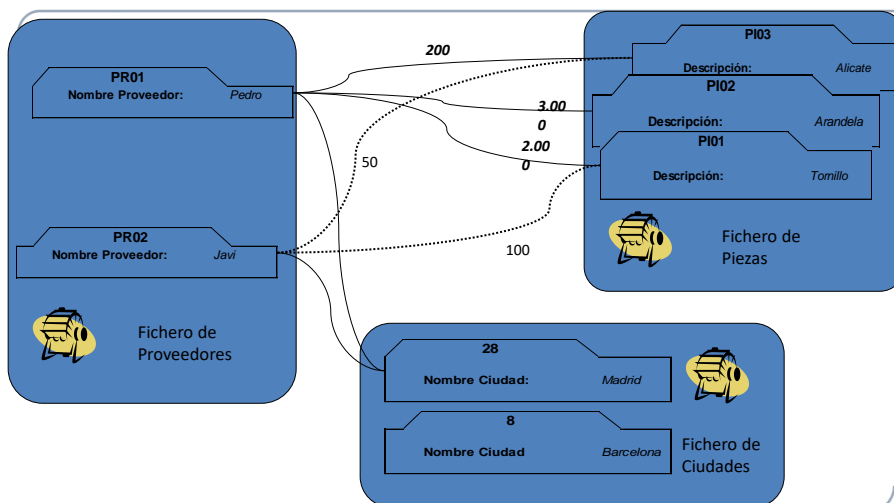
Normalización



¿Cómo son las ocurrencias?

© JMA 2010. All rights reserved

3ª Forma Normal



© JMA 2010. All rights reserved

Metodología de diseño conceptual

Entradas:

- Documentación de Requerimientos:
Requerimientos funcionales, de datos y no funcionales.
- Consultas a los usuarios.

© JMA 2010. All rights reserved

Metodología de diseño conceptual

Pasos:

1. Identificar entidades
2. Identificar relaciones
3. Identificar y asociar atributos a entidades y relaciones
4. Determinar los dominios de los atributos
5. Determinar los atributos de los identificadores candidatos, principales y alternativos
6. Comprobar si el modelo tiene redundancia mediante la normalización
7. Identificar roles de usuario y operaciones necesarias
8. Validar el modelo comprobando las transacciones de los usuarios
9. Repasar el modelo con los usuarios

© JMA 2010. All rights reserved

Metodología de diseño conceptual

Documentación de salida:

- Diagrama del MER
- Matriz Roles/Entidades
- Diccionario de datos
 - Entidades: Nombre, descripción, alias o sinónimos, identificadores únicos, volumetría
 - Relaciones: Nombre, descripción, cardinalidad
 - Atributos:
 - Nombre, descripción, alias o sinónimos
 - Tipo y tamaño
 - Dominio, valores únicos, requeridos, rangos, por defecto.
 - Si es compuesto: componentes
 - Si es multivaluado: cardinalidad
 - Si es derivado: formula de cálculo
 - Transacciones: Nombre, descripción, pasos
 - Roles: Nombre, descripción, alias o sinónimos, condición de pertenencia, operaciones

© JMA 2010. All rights reserved

DISEÑO DEL MODELO LÓGICO

© JMA 2010. All rights reserved

Modelos

- Modelo de almacenamiento
 - Tablas, columnas, claves, restricciones, filas
- Modelo de recuperación
 - Vistas, Índices
- Modelo funcional
 - Procedimientos almacenados, funciones y desencadenadores
- Modelo de seguridad
 - Permisos, roles

© JMA 2010. All rights reserved

Tablas

- Una tabla es un conjunto de elementos de datos (valores) que se organiza mediante un modelo de columnas verticales (que se identifican por su nombre) y filas horizontales.
- Una tabla tiene un número especificado de columnas, pero puede tener cualquier número de filas.
- Es la estructura de datos sobre la que se soportan las bases de datos relacionales
- Puede representar a una sola entidad o a una asociación de entidades

© JMA 2010. All rights reserved

Elementos de una Tabla

- COLUMNA (campo)
 - Representa un atributo de la entidad
 - Formada por un nombre y un conjunto de valores
 - Se identifica siempre por su nombre
 - Puede haber de 1 a N columnas
- FILA (tupla, registro)
 - Representa una ocurrencia de la entidad
 - Puede haber en una tabla de 0 a M filas
- CELDA
 - Representa el valor de un determinado atributo, el representado por la columna a la que pertenece la celda, para un registro de la tabla
 - Intersección de fila y columna
 - Este valor puede repetirse en otras intersecciones
 - Si no se conoce el valor de la intersección se dice que tiene valor nulo

© JMA 2010. All rights reserved

Restricciones

- Reglas que establecen los valores validos de una columna o tabla y son de obligado cumplimiento, para asegurar la integridad de los datos, evitando que la información sea incompleta, imprecisa o incoherente.
- Tipos de restricciones:
 - Valor Único: No puede estar repetido
 - Valor Requerido: No puede ser nulo
 - Valor Chequeado: Rango o condiciones de validación
 - Clave Primaria: Requerido y Único
 - Clave Ajena: Existe en otra tabla

© JMA 2010. All rights reserved

Integridad referencial

- Si existe la clave externa en una relación, el valor de clave externa debe coincidir con un valor clave candidata de alguna tupla en su relación de origen o el valor de clave externa debe ser nulo.
- Al exigir la integridad referencial, se impide a los usuarios las siguientes operaciones:
 - Agregar o cambiar filas a una tabla relacionada, si no hay fila asociada de la tabla principal.
 - Cambiar de valores en una tabla principal si provoca filas huérfanas en una tabla relacionada.
 - Eliminar filas de una tabla principal si están siendo usadas por filas relacionadas.
- Automatismos para las modificaciones y borrados:
 - NO PERMITIR, CASCADA, PONER NULOS, PONER VALORES POR DEFECTO, NO CHEQUEAR

© JMA 2010. All rights reserved

Claves (Integridad Referencial)

- Claves Candidatas
 - Todas las claves que pueden existir en una tabla
- Clave Primaria (Primary Key)
 - Columna o columnas que permiten distinguir unívocamente una fila de otra en la misma tabla
- Clave Ajena (Foreign Key)
 - Representa la relación entre tablas
 - En otra tabla es una clave primaria
 - Los valores que tome en nuestra tabla tienen que existir previamente en la tabla en la que es clave primaria
- Claves naturales vs. Claves artificiales

© JMA 2010. All rights reserved

Vistas

- Una vista consiste en una consulta almacenada accesible como una tabla virtual en una base de datos relacional.
- Las vistas son también tablas relacionales, pero sus datos se obtienen en tiempo de consulta.
- Las vistas pueden ofrecer ventajas sobre las tablas:
 - Pueden representar un subconjunto de los datos contenidos en una tabla
 - Pueden unir y simplificar varias tablas en una sola tabla virtual
 - Pueden actuar como tablas agregadas, donde el motor de base de datos presenta los resultados calculados como parte de los datos
 - Pueden ocultar la complejidad de los datos
 - Ocupan muy poco espacio; la base de datos sólo contiene la definición de una vista, no una copia de todos los datos que presentan
 - Pueden proporcionar seguridad adicional, pueden limitar el grado de exposición de una tabla o tablas con el mundo exterior

© JMA 2010. All rights reserved

Índices

- Un índice contiene claves (a partir de una o más columnas de la tabla o vista) y punteros que indican la ubicación de los datos en el almacenamiento.
- Se pueden mejorar significativamente el rendimiento de las consultas de bases de datos y las aplicaciones mediante la creación de índices bien diseñados (degradan la modificación de los datos).
- Los índices pueden reducir la cantidad de datos que se deben leer para devolver el conjunto de resultados de la consulta.
- Los índices también pueden exigir la unicidad de las filas de una tabla, lo que garantiza la integridad de los datos de los datos de la tabla.
- Se pueden almacenar independientemente de la tabla a la afectan.
- Se pueden crear de forma directa o indirecta
 - Cuando se crea una restricción de tipo CLAVE PRIMARIA o CLAVE ÚNICA, se genera implícitamente un índice asociado

© JMA 2010. All rights reserved

Procedimientos, Funciones, Disparadores

- Son similares a los procedimientos y funciones en otros lenguajes de programación, que permiten:
 - Aceptar los parámetros de entrada y devolver varios valores en forma de parámetros de salida al procedimiento de llamada o lote.
 - Contener la programación que realiza operaciones en la base de datos, incluyendo llamadas a otros procedimientos.
 - Devolver un valor de estado a un procedimiento de llamada o lote para indicar el éxito o el fracaso (y la razón del fracaso).
- Los beneficios del uso de código almacenado son:
 - Están registrados y centralizado en el servidor.
 - Permiten la programación modular.
 - Pueden mejorar la seguridad de la aplicación.
 - Pueden mejorar el rendimiento y reducir el tráfico de red.

© JMA 2010. All rights reserved

Transformando el ERD

| Conceptual | → | Lógico |
|-----------------|---|---------------------|
| Entidades | → | Tablas |
| Atributos | → | Columnas |
| Identificadores | → | Claves |
| Dominios | → | Restricciones |
| | → | Migración de claves |
| Relaciones | → | Restricciones |
| | → | Tablas renacidas |

© JMA 2010. All rights reserved

Paso de Entidades

- Las entidades se convierten en tablas.
- Reglas:
 - La tabla toma el nombre de la entidad en plural
 - Toda ocurrencia de la entidad se convierte en una fila (registro)
 - Todo atributo se convierte en columna
- Cardinalidad:
 - Una entidad (fuerte) pasa a una tabla
 - Una entidad pasa a varias tablas (con relaciones $1,1 \rightarrow 0,1$)
 - Varias entidades pasan a una tabla (desnormalizaciones y relaciones $1,1 \rightarrow 1,1$)

© JMA 2010. All rights reserved

Paso de Atributos

- Los atributos se convierten en columnas de las tablas.
- Reglas:
 - Un atributo simple y univaluado se convierte en una columna
 - Un atributo compuesto se convierte en tantas columnas como componentes, una por componente.
 - Un atributo multivaluado se convierte en función de la cardinalidad:
 - Fija: En tantas columnas como indique la cardinalidad
 - Variable: En una tabla ($1 \rightarrow n$)
 - Un atributo derivado se convierte en una columna calculada

© JMA 2010. All rights reserved

Paso de Dominios

- Los dominios se convierten en:
 - Definición de columnas
 - Reglas
 - Restricciones
 - Desencadenadores

© JMA 2010. All rights reserved

Paso de Identificadores Únicos

- Los identificadores únicos se convierten en claves y se implementan con restricciones
- Reglas:
 - El Identificador único Primario pasa a clave primaria con la restricción “Clave Primaria”
 - Los identificadores únicos Secundarios o Alternativos pasan a claves alternativas o candidatas con las restricciones “Requerido” y “Único”

© JMA 2010. All rights reserved

Migración de claves

- La clave primaria de una tabla, conjunto de columnas que forman la clave, migra a otra tabla, es decir, se le añaden las columnas de la clave primaria de la otra tabla y se establece una restricción de “Clave Ajena”.
- La tabla secundaria a la que migra la clave debe disponer del conjunto de columnas necesario para almacenar la clave del elemento con el que esta relacionado en la tabla primaria.

© JMA 2010. All rights reserved

Paso de Relaciones

- La conversión de las relaciones depende de la cardinalidad de la relación y se implementa mediante la migración de claves.
- Reglas:
 - 1:1 Se establecen la misma clave principal para las dos tablas, y en la tabla de menor cardinalidad mínima se establece la restricción de “Clave Ajena” sobre la clave primaria.
 - 1:* Se migra la clave de la entidad fuerte (lado 1) a la entidad débil (lado muchos). Si la cardinalidad mínima es 1 se establecen restricciones “Requerido” sobre las columnas de la clave ajena.
 - *: * Se crea una nueva tabla (renacida) a la que migran las claves de las entidades relacionadas, estableciéndose restricciones “Requerido” sobre las columnas de las claves ajenas. Los atributos de la relación, es caso de existir, se convierten en columnas de la tabla renacida.

© JMA 2010. All rights reserved

Matriz de Objetos-Roles

- Las entidades del modelo conceptual se han transformado en tablas y han aparecido nuevos objetos propios del modelo lógico:
 - Vistas, procedimientos almacenados, funciones
- Los permisos a aplicar sobre los objetos son:
 - Q: Query
 - C: Create
 - U: Update
 - D: Delete
 - X: Execute.

© JMA 2010. All rights reserved

Matriz de Objetos-Roles

| | Role 1 | Role 2 | ... | Role n |
|----------|--------|--------|-----|--------|
| Object 1 | Q | QCU | | |
| Object 2 | QCUD | - | | |
| Object 3 | - | QC | | |
| | | | | |
| SP 1 | X | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Object n | | | | |

Q: Query, C: Create, U: Update, D: Delete, X: Execute, -: None

© JMA 2010. All rights reserved

Metodología de diseño lógico

Entradas:

- Diagrama del MER
- Matriz Roles/Entidades
- Diccionario de datos del modelo conceptual
- Modelo funcional

© JMA 2010. All rights reserved

Metodología de diseño lógico

Pasos:

1. Paso a tablas
2. Revisar el modelo y completar con claves y restricciones
3. Comprobar si el modelo tiene redundancia mediante la normalización
4. Comprobar las restricciones de integridad
5. Identificar y definir las diferentes vistas para el usuario
6. Optimizar el modelo mediante la identificación y definición de índices
7. Identificar y definir los procesos: Procedimientos, Funciones y Disparadores
8. Revisar roles de usuario y transformar la matriz Roles/Entidades en la matriz Roles/Objetos
9. Validar el modelo comprobando las transacciones de los usuarios
10. Repasar el modelo con los usuarios

© JMA 2010. All rights reserved

Metodología de diseño lógico

Documentación de salida:

- Diagrama de tablas
- Matriz Roles/Objetos
- Diccionario de datos
 - Tablas: Nombre, descripción, alias o sinónimos, columnas, claves, restricciones, índices, disparadores y volumetría.
 - Relaciones: Nombre, descripción, integridad referencial
 - Columnas:
 - Nombre, descripción, alias o sinónimos
 - Tipo, precisión y longitud (fija o variable: tamaños máximo, mínimo y medio)
 - Restricciones y valor por defecto.
 - Si es calculada: formula de cálculo
 - Índices: Nombre, descripción, tabla, columnas y orden
 - Disparadores: Tabla, operaciones y descripción
 - Vistas: Nombre, descripción y columnas
 - Procedimientos y Funciones: Nombre, descripción, parámetros y resultados
 - Transacciones: Nombre, descripción, pasos
 - Roles: Nombre, descripción, alias o sinónimos, condición de pertenencia, operaciones

© JMA 2010. All rights reserved

DISEÑO DEL MODELO FÍSICO

© JMA 2010. All rights reserved

Introducción

- Proceso de producción mediante el cual se adapta la solución genérica del modelo lógico a una solución específica para un gestor de bases de datos relacionales concreto.
- El modelo físico debe aprovechar la fortalezas del gestor de base de datos escogido y subsanar las debilidades del mismo.
- Describe las relaciones de base, organización de archivos e los índices utilizados para lograr un acceso eficiente a los datos. También describe las restricciones de integridad asociadas y medidas de seguridad.
- Siempre tiene que estar adaptado a un sistema DBMS específico.

© JMA 2010. All rights reserved

Volumetría

- Tamaños máximo, mínimo y medio de filas
- Número inicial de filas
- Número máximo de filas
- Tasa de volatilidad = $((\text{altas} + \text{bajas}) / \text{inicial}) / \text{ud. tiempo}$
- Tasa de actividad = $((\text{Consultas} + \text{altas} + \text{bajas} + \text{actualizaciones}) / \text{inicial}) / \text{ud. tiempo}$
- Tasa de crecimiento = $((\text{altas} - \text{bajas}) / \text{inicial}) / \text{ud. Tiempo}$

© JMA 2010. All rights reserved

Metodología de diseño físico

Entradas:

- Diagrama de tablas
- Matriz Roles/Objetos
- Diccionario de datos

© JMA 2010. All rights reserved

Metodología de diseño físico

Pasos:

1. Traducción del modelo lógico al modelo del SGDB seleccionado
2. Implementar los Procedimientos, Funciones y Disparadores
3. Dimensionar la ocupación física en ficheros de la base de datos
4. Revisar y redefinir las vistas para el usuario
5. Optimizar el modelo mediante la revisión y redefinición de los índices
6. Estimar los requerimientos de espacio en disco
7. Diseñar los mecanismos de seguridad
8. Validar el modelo comprobando las transacciones de los usuarios
9. Diseñar los planes de mantenimiento y establecer los planes de contingencias

© JMA 2010. All rights reserved

Metodología de diseño físico

Documentación de salida:

- Requerimientos físicos para la implementación del modelo físico
- Secuencias de comandos de creación de la Base de Datos y carga inicial.
- Plan de mantenimiento
 - Operación, frecuencia, responsables y descripción
- Plan de contingencia
 - Tipo de incidencia, TMF, TMR, recursos, responsables y pasos a seguir en la recuperación