FUNDAMENTOS DE BASE DE DATOS.

Tema Nº4:Normalización I.

Indicador de logro Nº4:Identifica los tipos de normalizaciones considerando llaves principales y foráneas.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº4:**

Normalización I.

**Subtema 4.1:**

Introducción a la normalización, tipos de normalización: 1ra Forma Normal, 2da Forma Normal,3ra Forma Normal.

**NORMALIZACIÓN:**

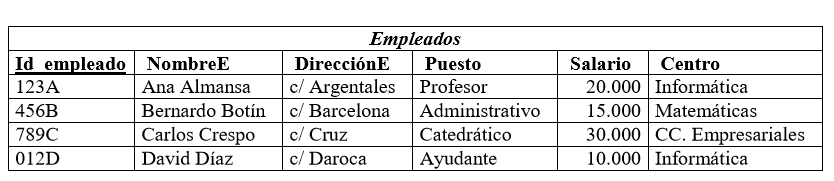
La traducción del esquema conceptual al lógico no es única. Es necesario contar con una medida de la calidad de la agrupación de los atributos en relaciones. Como herramienta principal se usan las dependencias funcionales para agrupar los atributos en esquemas de relación, que se dice que se encuentran en una determinada forma normal.

La forma normal de un esquema de relación, determina su grado de calidad con respecto a reducir dos efectos no deseados: **la redundancia de datos** y **las anomalías que produce esta redundancia**. Es importante determinar en qué forma normal se encuentra un esquema de relación y el procedimiento, conocido como normalización, para descomponerlo en otros esquemas de relación que se encuentren en formas normales más exigentes.

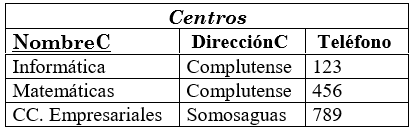
**REDUNDANCIA DE DATOS**

Un objetivo del diseño de bases de datos relacionales, es agrupar atributos en relaciones de forma que se reduzca la redundancia de datos y así el espacio de almacenamiento necesario. **Por ejemplo**, los siguientes dos esquemas

**Empleados** (Id\_empleado, NombreP, DirecciónP, Puesto, Salario, Centro)

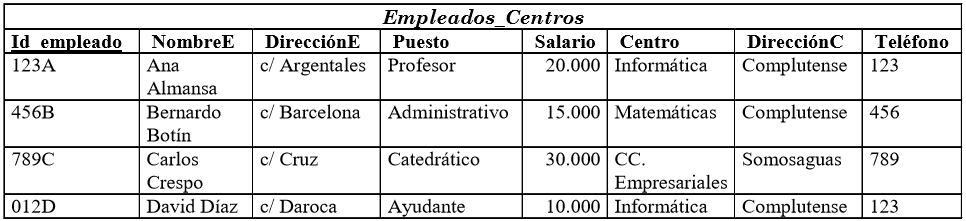


**Centros** (NombreC, DirecciónC, Teléfono)



**Contienen la misma información con el siguiente esquema:**

**Empleados\_Centros** (Id\_empleado, NombreP, DirecciónP, Puesto, Salario, NombreC, DirecciónC, Teléfono)



La relación **Empleados\_Centros** presenta redundancia de datos porque se repite para cada empleado la información asociada al centro. Las relaciones con datos redundantes presentan diferentes anomalías de actualización: son las anomalías de inserción, borrado y modificación.

**ANOMALÍAS DE ACTUALIZACIÓN**

**Anomalías de inserción.** Se produce en dos casos. En primer lugar, cuando se inserta una nueva fila sin respetar las dependencias funcionales. En el ejemplo anterior puede ocurrir si se añade una fila de un empleado adscrito a Informática y con un teléfono distinto de 123. En segundo lugar, la imposibilidad de añadir nuevos datos para el consecuente de la dependencia funcional sin que exista un antecedente para ella. En el ejemplo anterior no se puede dar de alta un centro a menos que exista un empleado destinado en él. Sería necesario dejar valores nulos en la clave (Id\_empleado).

**Anomalías de modificación.** Se produce cuando se modifican las columnas con datos redundantes de sólo un subconjunto de las filas con el mismo dato. En el ejemplo puede ocurrir cuando se modifica el teléfono de Informática sólo en la primera fila.

**Anomalías de eliminación.** Se produce cuando se eliminan todas las filas en las que aparecen los datos redundantes por lo que se pierde los datos de la dependencia funcional. Si se elimina la segunda fila porque el empleado se da de baja, se pierden también los datos del centro.

Las anomalías de actualización aparecen también en los modelos de red y jerárquico, y se resuelven con campos virtuales y tipos de registros virtuales implementados con punteros. Los modelos orientados a objetos evitan el problema mediante la referencia en lugar de la copia.

**FORMAS NORMALES Y NORMALIZACIÓN**

La forma normal de una relación se refiere a la mejor forma normal que satisface un esquema de relación indicando así el grado hasta el que se ha normalizado. La indicación del grado de calidad de un esquema de relación se refiere en general en el contexto global del esquema de la base de datos relacional, es decir, en el conjunto de todos los esquemas de relación de la base de datos. Dos propiedades que se deben cumplir para poder asegurarlo son:

* La propiedad de preservación de dependencias, que asegura que las dependencias funcionales originales se mantienen en algún esquema de relación después de la descomposición.
* La propiedad de la posibilidad de reproducir la información de la tabla antes de su descomposición a partir de las tablas resultado de ella. Las formas normales más habituales, por orden ascendente de exigencia de las propiedades deseadas, son: Primera (1FN) Segunda (2FN) Tercera (3FN)

**Subtema 4.2:**

Identificar llaves primarias y foráneas y el primer caso de normalización de datos.

**DESARROLANDO EL PRIMER CASO DE NORMALIZACIÓN**

**FORMA NORMAL CERO**:

En esta etapa el **usuario modelador** recibe los datos tal como se los han proporcionado, en otras palabras, son datos completamente desordenados y redundantes.

Ejemplo:







**PRIMERA FORMA NORMAL 1FN, CONSISTE:**

1. Eliminar los Grupos Repetitivos de las tablas individuales. Garantizando que los datos sean Atómicos (únicos)
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con la clave primaria

**1**. Buscamos los datos que provocan la redundancia. (Grupos Repetitivos)



**2**. Crear una tabla separada por cada grupo de datos relacionados

Estas tablas quedan de esta manera al momento de la separación.

**PRINCIPAL**



**SECUNDARIO**



**3**. Identificar cada grupo de datos relacionados con la clave primaria



**PRINCIPAL**



**SECUNDARIO**



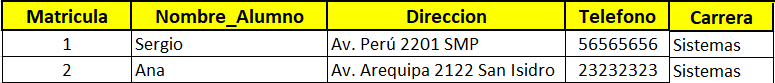
Traslada la clave

La clave de la parte principal, pasa a la parte secundaria.

**De esta manera, en la parte principal se podrán eliminar la redundancia de datos. quedando el diseño de nuestra base de datos de la siguiente manera:**



**PRINCIPAL**



**SECUNDARIO**



En la parte secundaria aún existen 🡪

redundancias. entonces se debe

aplicar la 2FN.

**SEGUNDA FORMA NORMAL 2FN, CONSISTE:**

**Regla:** Ahora para normalizar en 2da forma normal, la tabla deberá estar ya en **1ra** forma normal. Y localizar las dependencias **Funcionales** y **Transitivas**.

|  |  |
| --- | --- |
| **DEPENDENCIAS** | |
| **FUNCIONAL** | **TRANSITIVA** |
| es una relación entre uno o más atributos, es decir los atributos depende exclusivamente de a clave principal.  **Por ejemplo**, si se conoce el valor de DNI (Documento Nacional de Identidad) este tiene una conexión explicita con Apellido o Nombre. | es una relación entre uno o más atributos, es decir los atributos depende de otros atributos y no clave principal.  **Por ejemplo**, si se conoce el valor de DNI (Documento Nacional de Identidad) este tiene una conexión explicita con Apellido o Nombre. Pero no con un producto. |

**NOTA:** en la 2FN se deben eliminar aquellos atributos que tengan **dependencia** **TRANSITIVA**.

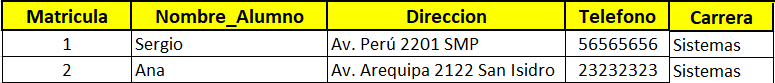


Vamos a tener varios atributos **A, B, C**, donde **B** y **C** dependen Funcionalmente del Atributo **A** el Primary Key.

Aquí es distinto, vamos a tener los mismos atributos, donde **B** depende Total y Funcionalmente de **A**, en el caso de **C**, tiene dependencia transitivamente con **B** y con **A**,



Este es el resultado de nuestro último análisis (1FN) donde observamos que en la parte secundaria, existen redundancias, es ahí donde se debe aplicar las dependencias.







Verificamos que atributos tienen dependencia **FUNCIONAL** y **TRANSITIVA**.



**Depende**

**No depende**

**No depende**

Como ya se localizaron las dependencias **TRANSITIVAS**, se procede a eliminarlas, en modo de separación de tablas, de esta manera, se evita la redundancia.



**PRINCIPAL SECUNDARIO**



La clave de la parte principal, pasa a la parte secundaria.



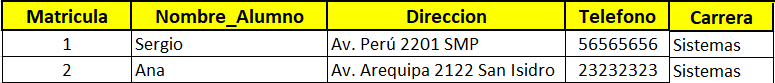


**PRINCIPAL**

**SECUNDARIO**

**De esta manera, en la parte principal se podrán eliminar la redundancia de datos. Quedando el diseño de nuestra base de datos, de la siguiente manera:**











Aún siguen existiendo

Redundancia. 🡪

**TERCERA FORMA NORMAL 3FN, CONSISTE:**

* Eliminar aquellos campos o atributos que no dependan de la clave.

**Esta tablilla contiene datos redundantes, la carrera.**



Buscamos aquellos atributos que no dependan de la llave (**TRANSITIVA**), en este caso es **carrera** no depende den **Num Materia**.



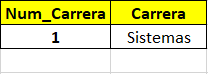
**Aplicamos la separación en tablas.**

**PRINCIPAL SECUNDARIO**



En este caso la parte **PRINCIPAL** tiene sus atributos completos, en el caso de la parte **SECUNDARIO** sólo tiene un atributo descriptivo, aquí se debe aplicar la creación de un atributo clave, para evitar la redundancia, si fuera caso. Quedando de la siguiente manera:

**PRINCIPAL SECUNDARIO**

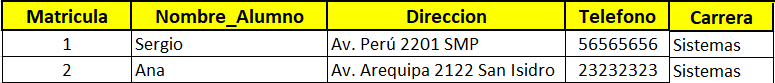


Si vemos nuestro modelo relacional por completo, después de haber pasado todas las formas normales de modelamiento, esto quedaría de la siguiente manera:

**CADA TABLA DEBE LLEVAR UN NOMBRE Y RELACIONARSE.**



**ALUMNO**





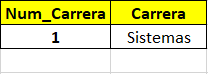
**¿Qué pasaría con la tabla Carrera?**

**No puede quedar solitaria.**

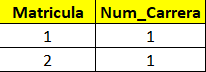
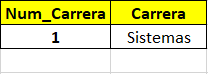
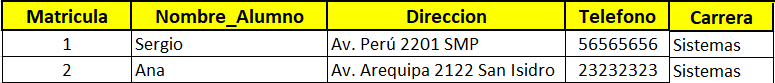


**MATERIA CARRERA**





En el caso de la tabla **CARRERA** aplicaremos el mismo principio de combinación de datos, de las tablas **ALUMNO**, **MATERIA**, es decir crearemos una tabla **intermedia** el cual pueda enlazar a las tablas **ALUMNO** y **CARRERA**, tal como se observa en la siguiente imagen:



**FK**

**FK**

**ALUMNO**

**ALUMNO\_MATERIA ALUMNO\_CARRERA**

**FK**

**MATERIA CARRERA**



**NORMALIZACIÓN TERMINADA.**

**CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA**

* Nos ayuda a prevenir la redundancia de los datos
* Nos ayuda a evitar la falta de integridad e inconsistencia en la base de datos.

**ACTIVIDAD VIRTUAL**

Analizar y revisar los siguientes enlaces; luego responde las preguntas propuestas.

<https://www.youtube.com/watch?v=bO18omSzeR4>

* ¿Cuáles son las ventajas cuando tenemos normalizado la base de datos?
* ¿Cuáles son los tipos de formas normales que se tiene en una base de datos?
* ¿Cuál es la diferencia entre dependencia funcional y dependencia transitiva?

**NORMALIZAR EL SUGUIENTE CASO:**

