Base de Datos

Luis Monge

[lmoge@ucenfotec.ac.cr](mailto:lmoge@ucenfotec.ac.cr)



**Fechas Importantes**

13 de Octubre: Primer Laboratorio

27 de Octubre: Especificación del Proyecto

05 de Noviembre: Primer Examen

24 de Noviembre: Pegudo Laboratorio

10 de Diciembre: Segundo Examen

15 de Diciembre: Entrega de Proyecto

Maquina Virtual VMware Player (CENFOTEC)

Open Data Base

Browse: Disco (D)/MV Bases de Datos I-2015.

**Base de Datos**

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información,

Es una una colección de información ordenada y estructurada.

“Sistema de Archivos Electrónico”

Ventajas:

* Es Multiusuario (acceso concurrente por parte de múltiples usuarios)
* Mejora la seguridad e integridad de datos
* Portable y escalable
* Control sobre la redundancia de datos
* Consistencia de datos
* Mantenimiento de estándares
* Capacidad de adaptarse al crecimiento

### Vulnerable a los fallos:

### El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ella que deben tenerse copias de seguridad, (Backup).

**sistema gestor de base de datos (SGBD)**



Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una [**base de datos**](https://es.wikipedia.org/wiki/Base_de_datos), además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

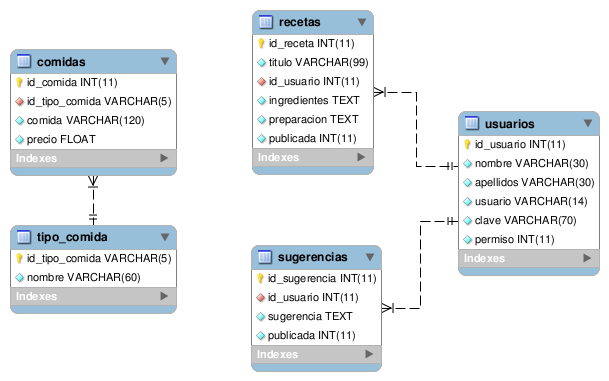
**RDBMS**

**Plataformas de programación de Base de Datos.**

Debe ser Jerárquico, Reticular, relacional.

* ORACLE
* SQL Server
* MySql
* SQL
* DB2
* InformIx
* RBD
* BORLAND
* ASE
* ASA
* ODBC
* Sybase

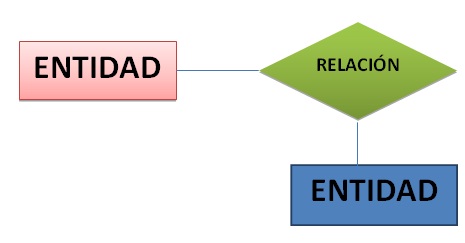
**modelo entidad relación (M.E.R.)**



**Elementos**

* Entidades.
  + Todo aquello que necesitamos del Sistema.
  + Se recomiendo usar nombres en Singular.
  + Una ocurrencia o instancia, un caso o ejemplo.
* Atributos.
  + Son las características de las entidades.
* Relaciones.
  + Es un elemento bidireccional . cuando hay una relación entre 2 o más elementos.
  + La relación entre una misma entidad, se denomina **Relación Circular o Recursiva.**
  + Opcionalidad y Cardinalidad.
* Identificadores.

***relational database* management system (*RDBMS*)**



Es un sistema de gestión de Base de Datos que se basa en el modelo racional inventado por por [**EF Codd**,](https://translate.googleusercontent.com/translate_c?depth=1&hl=es&prev=search&rurl=translate.google.com&sl=en&u=https://en.wikipedia.org/wiki/Edgar_F._Codd&usg=ALkJrhjhQ_XwU1hYlLUn4mxqGu39rbxKng) de los de IBM [**en San José Laboratorio de Investigación**](https://translate.googleusercontent.com/translate_c?depth=1&hl=es&prev=search&rurl=translate.google.com&sl=en&u=https://en.wikipedia.org/wiki/IBM_Almaden_Research_Center&usg=ALkJrhhcJNH8BwpLyku5f41nvUsU5Ne1rw).



**Entre 2 entidades**, pueden haber **múltiples relaciones**, lo que se conocen como **opcionalidad**.

* Si la relación es **Opcional no** tiene contraparte.
* Si la relación es **Mandatoria debe** tener contraparte.



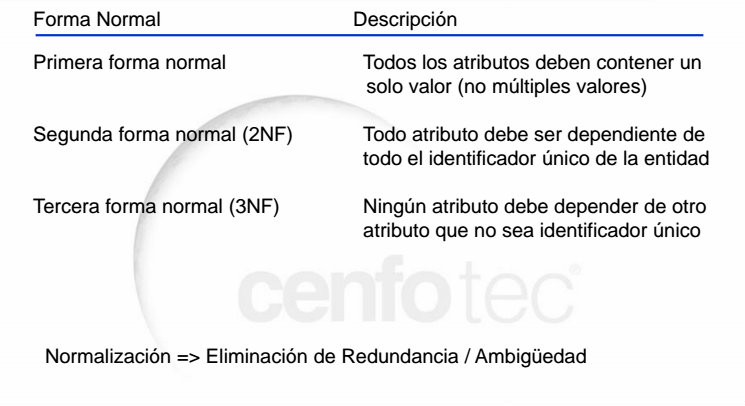


**cardinalidad**

**opcionalidad**

**Atributo Locales**

**Atributos Externos**



**4FN**

Se asegura de que las dependencias multivaluadas independientes estén correctas y eficientemente representadas en un diseño de base de datos.



**5FN**

Es un nivel de normalización de bases de datos diseñado para reducir redundancia en las bases de datos relacionales que guardan hechos multi-valores aislando semánticamente relaciones múltiples relacionadas. Una tabla se dice que está en 5NF si y sólo si está en 4NF y cada dependencia de unión (join) en ella es implicada por las claves candidatas.



El proceso de **normalización de** [**bases de datos**](https://es.wikipedia.org/wiki/Base_de_datos) consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del [modelo entidad-relación](https://es.wikipedia.org/wiki/Modelo_E-R) al[modelo relacional](https://es.wikipedia.org/wiki/Modelo_relacional).

Las bases de datos relacionales se normalizan para:

* Evitar la [redundancia](https://es.wikipedia.org/wiki/Redundancia) de los datos.
* Disminuir problemas de actualización de los datos en las tablas.
* Proteger la [integridad](https://es.wikipedia.org/wiki/Integridad_de_datos) de los datos.

En el modelo relacional es frecuente llamar [*tabla*](https://es.wikipedia.org/wiki/Tabla_(base_de_datos)) a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

* Cada tabla debe tener su nombre único.
* No puede haber dos [filas](https://es.wikipedia.org/wiki/Fila_(base_de_datos)) iguales. No se permiten los duplicados.
* Todos los datos en una [columna](https://es.wikipedia.org/wiki/Columna_(base_de_datos)) deben ser del mismo tipo.

Trabajo (Código, Nombre, Posición, Salario), donde Código es la Clave Primaria.

* Relación = [tabla](https://es.wikipedia.org/wiki/Tabla_(base_de_datos)) o archivo
* Registro = [registro](https://es.wikipedia.org/wiki/Registro_(base_de_datos)), fila , renglón o tupla
* Atributo = [columna](https://es.wikipedia.org/wiki/Columna_(base_de_datos)) o campo
* Clave = llave o código de identificación
* Clave Candidata = superclave mínima
* Clave Primaria = clave candidata elegida
* Clave Ajena (o foránea) = clave externa o clave foránea
* Clave Alternativa = clave secundaria
* Dependencia Multivaluada = dependencia multivalor
* RDBMS = Del inglés *Relational Data Base Manager System* que significa, *Sistema Gestor de Bases de Datos Relacionales*.
* 1FN = Significa, *Primera Forma Normal* o 1NF del inglés *First Normal Form*.



El identificador único es la función que tiene un atributo para identificar una entidad. Ejm:

ced, ced\_jur, num\_pas, code.

Puede estar compuesto por identificadores externos (entidades relacionadas)

En el modelo entidad relación se diferencia con ´#´.

Una entidad puede tener dos identificadores, con la exclusividad que sea dependiente uno del otro, de lo contrario, no es aceptado para crear una base de datos eficaz y libre de ambigüedad.

Conversión del Modelo de Identidad de una Base de Datos.

**6 Pasos**

**Paso1**

**Entidad = table**

Es buena práctica nombrarlas en plural

**Paso2**

**Columnas =**

Se convierten en columnas de la tabla

**Paso3**

**FK´s = FOREIGN KEY**

Llave foránea es un atributo que debo conocer de otra entidad, al cual se le conoce como atributo externo. (el identificador de la entidad que importa el atributo)

**Paso4**

**PK´s = PRYMARYKEY**

LLave primaria, es decir identificador único de la entidad.

**Paso5**

**Tipos = DATETIME, NUMERIC(9), CHAR(5), VARCHAR(20)**

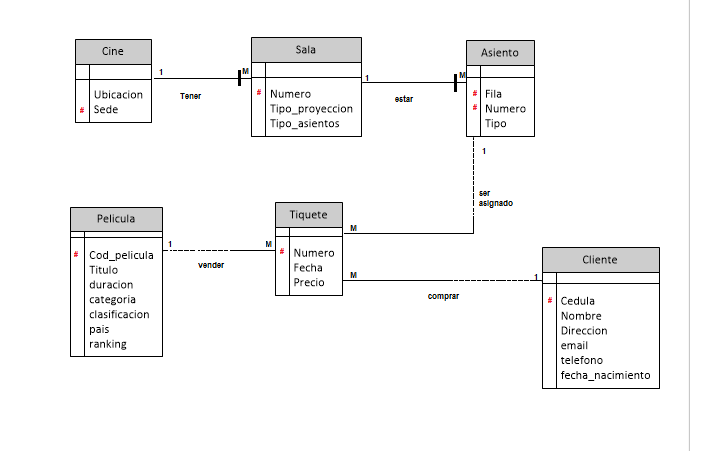
Tipo de datos que presenta una columna, longitud, precisión, escala, fechas, caracteres, cantidad de números.

**Paso6**

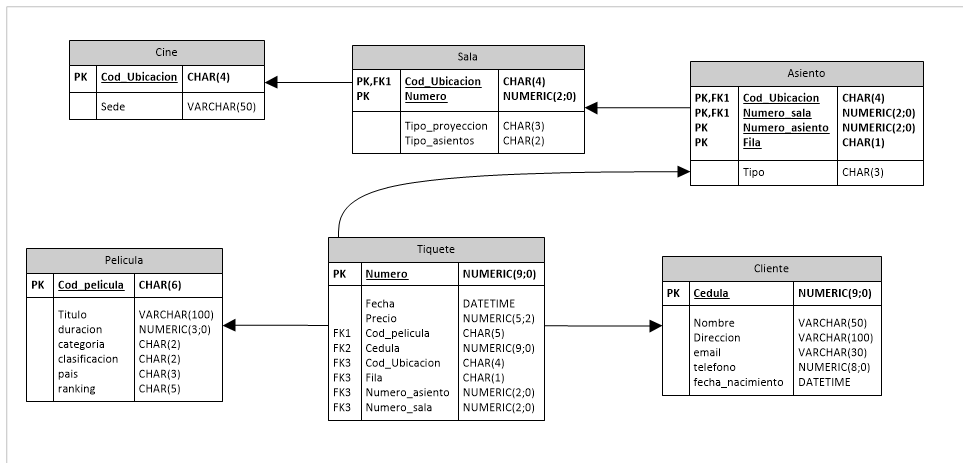
**NN = NOT NULL**

Se refiere a que el espacio que ocupa el atributo, debe tener un valor asignado.

**MER (Modelo Entidad Relación) to DBD**



**DataBase Cines (DBD)**



**Base de Datos.**

**Oracle.**

**Cinema** (Visto en clase).

|  |  |
| --- | --- |
| Create Table CINES(  cod\_cine char(4) Not Null,  sede varchar2(50) Not Null,  Constraint PK\_cines Primary Key(cod\_cine)  ); | Create Table Peliculas(  cod\_pelicula Varchar(6) Not Null,  titulo Varchar(100) Not Null,  duracion Numeric(3) Not Null,  categoria Char(2) Not Null,  clasificacion Char(2) Not Null,  pais Char(3) Null,  rankig Char(3) Null,  Constraint PK\_PELICULAS Primary Key  (cod\_pelicula)  ) |
| Create Table SALAS(  code\_cine char(4) not null,  numero numeric(2) not null,  tipo\_proyeccion char(3) null,  tipo\_asientos char(2) null,  Constraint PK\_SALAS Primary key  (code\_cine, numero),  Constraint FK\_SALAS Foreign Key  (code\_cine) References CINES (cod\_cine)  ); | Create Table CLIENTE(  cedula Numeric(9) Not Null,  nombre Varchar(50) Not Null,  direccion Varchar(100) Not Null,  email Varchar(30) Not Null,  telefono Numeric(9) Not Null,  fecha\_nacimiento date,  Constraint PK\_CLIENTE Primary Key  (cedula)  ); |
| Create Table Asiento(  code\_cine char(4) not null,  numero\_sala numeric(2) not null,  numero\_asiento numeric(2) not null,  fila char(1) not null,  tipo char(3) not null,  Constraint PK\_Asiento Primary Key  (code\_cine, numero\_sala, numero\_asiento, fila),  Constraint FK\_Asiento\_SALAS Foreign Key  (code\_cine, numero\_sala) References SALAS(code\_cine, numero),  Constraint FK\_Asiento\_CINES Foreign Key  (code\_cine) References CINES(cod\_cine)  ) | Create Table TIQUETE(  numero Numeric(9) Not Null,  cod\_pelicula Varchar(6) Not Null,  cod\_ubicacion Char(4) Not Null,  numero\_sala Numeric(2) Not Null,  numero\_asiento Numeric(2) Not Null,  fila Char(1) Not Null,  fecha date,  cedula Numeric(9) Not Null,  precio Numeric(5) Not Null,  Constraint PK\_TIQUETE Primary Key  (numero),  Constraint FK\_TIQUETE\_ASIENTO Foreign Key  (cod\_ubicacion, numero\_sala, numero\_asiento, fila)  References Asiento(code\_cine, numero\_sala, numero\_asiento, fila),  Constraint FK\_TIQUETE\_PELICULA Foreign Key  (cod\_pelicula) References PELICULAS(cod\_pelicula),  Constraint FK\_TIQUETE\_CLIENTE Foreign Key  (cedula) References CLIENTE(cedula)  ) |

**SQLPLUS**

Herramienta para interactuar con la base de datos en línea de código

Para entrar a la consola

**“inicio + r, cmd”**

C:\Users\alego>sqlplus

SQL\*Plus: Release 11.2.0.2.0 Production on Mar Oct 6 20:18:39 2015

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: system

Enter password:

Connected to:

Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production

SQL>

**Comandos Básicos**

* **clear screen**
* **create user nombre, ejm:**

**SQL> create user CINES**

**2 identified by cines;**

**User created.**

* **alter user CINES, ejm:**

**SQL> alter user CINES**

**2 identified by new\_pass;**

**User altered.**

* **password, ejm:**

**SQL> password**

**Changing password for CINES**

**Old password:**

**New password:**

**Retype new password:**

**Password changed**

* **grant, ejm:**

**SQL> grant connect, resource**

**2 to CINES;**

**Grant succeeded.**

* **connect, ejm:**

**SQL> connect**

**Enter user-name: CINES**

**Enter password:**

**Connected.**

Los comandos de SQL se deben cerrar con (punto y coma, ‘ ; ’) para que ejecuten.

|  |  |
| --- | --- |
| **SqlPlus** | **SQL** |
| show user  connect  host  host cd  clear screen  exit  describe  edit  start  list | select \* from cat;  drop table name; |

Es recomendable asignar el nombre de las tablas en **plural**, además, si hay un cambio de palabra utilizar “**underscore**”.

Datos en una tabla (Registros, filas u(o) ocurrencia).

Los atributos (Columnas).

**SqlPlus**

**DML's (Data Manipulation Language)**, "permite manipular registros de la Base de Datos".

-Insert

-Update

-Delete

**DLL's (Data Definition Language)**, "".

-Create

-Alter

-Drop

**DCL's (Data Lenguaje)**, "Manipulan los permisos (privilegios), de los usuarios sobre los objetos".

-Grant

-Revoke

**Trans ()**, "Permite confirmar o deshacer una serie transacciones".

-Commit

-Rollback

-Se ejecutan sobre los **DCL's**

**SELECT**

**Select**, es el comando más utilizado en la creación de Base de Datos, el 70% de comandos ejecutados, son Selects.

Tiene muchas variantes.

Los Query's "Se programan en base al **Select**".

Para extraer los datos requeridos, dependiendo de los requerimientos de la consulta y mostrarlos.

Los **Comandos** se componen de:

Palabras reservadas, identificadores y símbolos

lista\_cols = {colum | 'constante' | expresión}

lista\_cols = {col1, 10, col2 \* col3}

**Operadores de Comparación**

=

>

>=

<

<=

<>

BETWEEN col2 AND col3

IN(x, 'thing', 3)

LIKE "A"

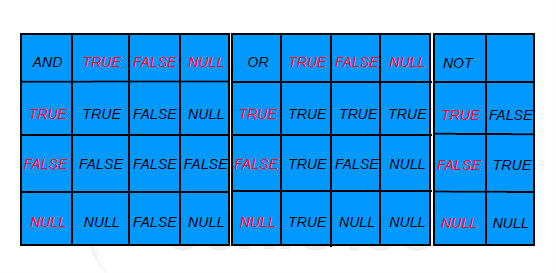
IS NULL

**Operadores Lógicos**

AND

OR

NOT



Truncate no tiene rollback

Purge Recyclebin;

Borrar todo lo que está en la papelera por el drope \* (all)

Select es solo un comando de consulta, no modifica los registros en la base de datos.

**Group by**

**Having**

Me permite filtrar los grupos.

* Las filas son agrupadas
* Se aplican las funciones de grupo
* Los grupos que cumplan con la condición (HAVING), son desplegados

**Order By**

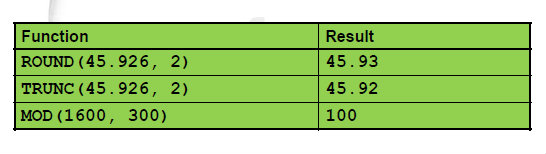
**Funciones**

**Funciones de tipo Char**

* (**Concat**)
* (**Substr**) Recibe un parámetro, su rango inicial y su rango final. (**Substr(parámetro, 1, 1)**)
* (**Length**)
* (**Instr**)
* (**Lpad**)
* (**Rpad**)
* (**Trim**)
* (**Replace**)
* (**Lower**)
* (**Upper**)
* (**Initcap**)

**Funciones Numéricas.**

* (**Round**) Redondea
* (**Trunc**) Trunca
* (**Mod**) Residuo



**Funciones para Fechas.**

* **(Select Sysdate from dual)** Traer Fecha Actual
* **(Months\_BETWEEN)** # de meses entre dos fechas
* **(Add\_MONTHS)** Agrega meses a una fecha
* **(Next\_DAY)** Siguiente día de la fecha indicada
* **(Last\_DAY)** Último día del mes indicado
* **(Round)** Redondea una fecha
* **(Trunc)** Trunca una fecha

**Funciones de agrupamiento.**

* **(AVG) ???**
* **(Count)** Cuenta registros válidos
* **(MAX)** Muestra el registro más alto
* **(MIN)** Muestra el registro más bajo
* **(STDDEV) ???**
* **(SUM)** Sumatoria de filas
* **(Variance) ???**

**Función Single Row**

Que modifica un dato

* **DECODE(to\_char(fecha\_ingreso, 'yy'), '90', true, false) as “90”**
* **DECODE(X, V1, R1, V2, R2, V3, R3, Rd)**

**CASE X**

**When '90' then 1**

**else 0**

**Manage Page Size**

* set pageSize 100
* col nombre format a9
* col nombre format ‘9999999999’
* set lines 80
* set pages 40
* set echo on en el archivo queda el script y el resultado

**Top-N analisis**

Un Select de un Select

La tablas tienen una pseudo columna que se llama Rownum es el orden cronológico todos los registros tienen su Rownum

Cuando se creo y las tablas tienen también un Rownum y el resultado va ha tener un Rownum

Eje:

|  |
| --- |
| Select nombre  From (Select nombre, apellido x  From empleados  order by apellido ASC)  where rownum <= 3;  Select nombre  From (Select nombre, apellido, num\_dept nd  From empleados  order by apellido ASC) y  join departamentos d on(y.nd = d.num\_dept)  where rownum <= 3; |

**Para salvar, editar, mostrar y correr un query, podemos utilizar:**

* **save pr4-1**
* **edit pr4-1**
* **get pr4-1**
* **start pr4-1**

**Consulta:**

**Mostrar el porcentaje de personas y la cantidad que ingresaron por mes.**

|  |
| --- |
| Select AVG(salario) Porcentajes, Count(1) Cantidad, To\_Char(fecha\_ingreso, 'mm') Mes  from empleados  group by to\_char(fecha\_ingreso, 'mm')  order by Mes; |

**Consulta:**

**Mostrar los nombres que empiezan con una letra ‘x’.**

Select COUNT(nombre)

from empleados

Where nombre LIKE 'A%';

**Consulta:**

**Mostrar la cantidad de nombres que empiezan con una Vocal.**

|  |
| --- |
| Select Substr(nombre,1,1) Vocal, Count ('x') Cant  from empleados  Where Substr (nombre,1,1) IN('A','E','I','O','U')  Group By Substr (nombre,1,1)  Order By 1; |



10/11/15

Practica

-Quien es el empleado que gana mas

-Cuales son los nombres de los 3 empleados mas antiguos

-Cual es emleado que ingreso un viernes mas nuevo

set lines 80

set pages 40

set echo on en el archivo queda el scrip y el resultado

cual es el primer empleado por apellido

Matriz

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select to\_char(fecha\_ingreso, 'd'), "#", to\_char(fecha\_ingreso, 'day') DIA, count(1) CANT,

sum(decode(to\_char(fecha\_ingreso, 'yy'), '90', 1, 0)) "90",

sum(decode(to\_char(fecha\_ingreso, 'yy'), '91', 1, 0)) "91",

sum(decode(to\_char(fecha\_ingreso, 'yy'), '92', 1, 0)) "92",

sum(decode(to\_char(fecha\_ingreso, 'yy'), '93', 1, 0)) "93",

sum(decode(to\_char(fecha\_ingreso, 'yy'), '94', 1, 0)) "94"

From empleados

--where to\_char(fecha\_ingreso, 'd') not in('7','1')

group by to\_char(fecha\_ingreso, 'd'), to\_char(fecha\_ingreso, 'day')

order by 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Bibliografía**

<https://translate.google.com/translate?hl=es&sl=en&u=https://en.wikipedia.org/wiki/Relational_database_management_system&prev=search>

<https://es.wikipedia.org/wiki/Sistema_de_Gesti%C3%B3n_de_Bases_de_Datos>

<http://www.eet2mdp.edu.ar/alumnos/MATERIAL/MATERIAL/info/infonorma.pdf>

<https://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos>