

Introduccion a la Bioinformática

Información biológica en formato electrónico

Bases de datos

Fernán Agüero
Instituto de Investigaciones Biotecnológicas
UNSAM

Bases de datos: introducción: conceptos básicos

Qué es una base de datos?

Una colección de datos

Cómo colecciono los datos?

Decisión del usuario. Diseño de la base de datos.

Puedo usar:

Procesador de texto? (Word)

Si. Permite sólo búsqueda y ordenamiento simples.

Planilla de Cálculo? (Excel)

También. Como los datos están en columnas independientes, se puede ordenar en formas más complejas. Las búsquedas siguen siendo simples.

Bases de datos: introducción: conceptos básicos: registros

- Una colección de registros (records).
 - Cada registro tiene varios campos.
 - Cada campo contiene información específica.
 - Cada campo contiene datos de un tipo determinado.
 - Ej: dinero, texto, números enteros, fechas, direcciones
 - Cada registro tiene una **clave primaria**. Un identificador **único** que define al registro sin ambigüedad.

Planilla

Versión simple de una base de datos

gi	Accession	version	date	Genbank Division	taxid	organims	Number of Chromosomes
6226959	NM_000014	3	01/06/2000	PRI	9606	homo sapiens	22 diploid + X+Y
6226762	NM_000014	2	12/10/1999	PRI	9606	homo sapiens	22 diploid + X+Y
4557224	NM_000014	1	04/02/1999	PRI	9606	homo sapiens	22 diploid + X+Y
41	X63129	1	06/06/1996	MAM	9913	bos taurus	29+X+Y

Tipos de datos

- Cada campo de una base de datos contiene un tipo particular de datos
 - 211203
 - Es un numero?
 - Es texto?
 - Es una fecha?
- Ejemplo de una busqueda: buscar todos los registros en donde el valor almacenado sea mayor que 211203
 - Es obvio que para poder comparar los valores almacenados tenemos que saber qe tipo de valores estamos comparando.
 - Si es una fecha: 21 12 03 < 2 12 04
 - Si es un numero: 211 203 > 21 204
 - Si es texto: 211203 \neq 21204, las comparaciones < y > pueden dar distintos resultados (evaluan orden o longitud)

Tipos de datos

- Numericos (enteros, decimales)
- Texto
- Fechas (DD/MM/YYYY, HH:MM:SS)
- Logicos (boolean) = verdadero / falso
- Geometricos (punto, linea, circulo, poligonos, etc.)

Bases de datos: conceptos básicos: clave primaria

gi	Accession	version	date	Genbank Division	taxid	organims	Number of Chromosomes
6226959	NM_000014	3	01/06/2000	PRI	9606	homo sapiens	22 diploid + X+Y
6226762	NM_000014	2	12/10/1999	PRI	9606	homo sapiens	22 diploid + X+Y
4557224	NM_000014	1	04/02/1999	PRI	9606	homo sapiens	22 diploid + X+Y
41	X63129	1	06/06/1996	MAM	9913	bos taurus	29+X+Y

gi = Genbank Identifier: Clave única : Clave primaria

Cambia con cada actualización del registro correspondiente a la secuencia

Accession Number: Clave secundaria

Refiere al mismo locus y secuencia, a pesar de los cambios en la secuencia.

Accession + Version es equivalente al **gi** (representa un identificador único)

Ejemplo: AF405321.2 Accession: AF405321 Version: 2

Bases de datos: bases de datos relacionales

gi	Accession	version	date	Genbank Division	taxid	organims	Number of Chromosomes
6226959	NM_000014	3	01/06/2000	PRI	9606	homo sapiens	22 diploid + X+Y
6226762	NM_000014	2	12/10/1999	PRI	9606	homo sapiens	22 diploid + X+Y
4557224	NM_000014	1	04/02/1999	PRI	9606	homo sapiens	22 diploid + X+Y
41	X63129	1	06/06/1996	MAM	9913	bos taurus	29+X+Y

Base de datos relacional:


Normalizar una base de datos: repartir sub-elementos repetidos en varias tablas, relacionadas a través de un identificador único (clave primaria).

gi	Accession	version	date	Genbank Division	taxid
6226959	NM_000014	3	01/06/2000	PRI	9606
6226762	NM_000014	2	12/10/1999	PRI	9606
4557224	NM_000014	1	04/02/1999	PRI	9606
41	X63129	1	06/06/1996	MAM	9913
taxid	organims	Number of Chromosomes			
9606	homo sapiens	22 diploid + X+Y			
9913	bos taurus	29+X+Y			

Bases de datos: distribucion de la informacion

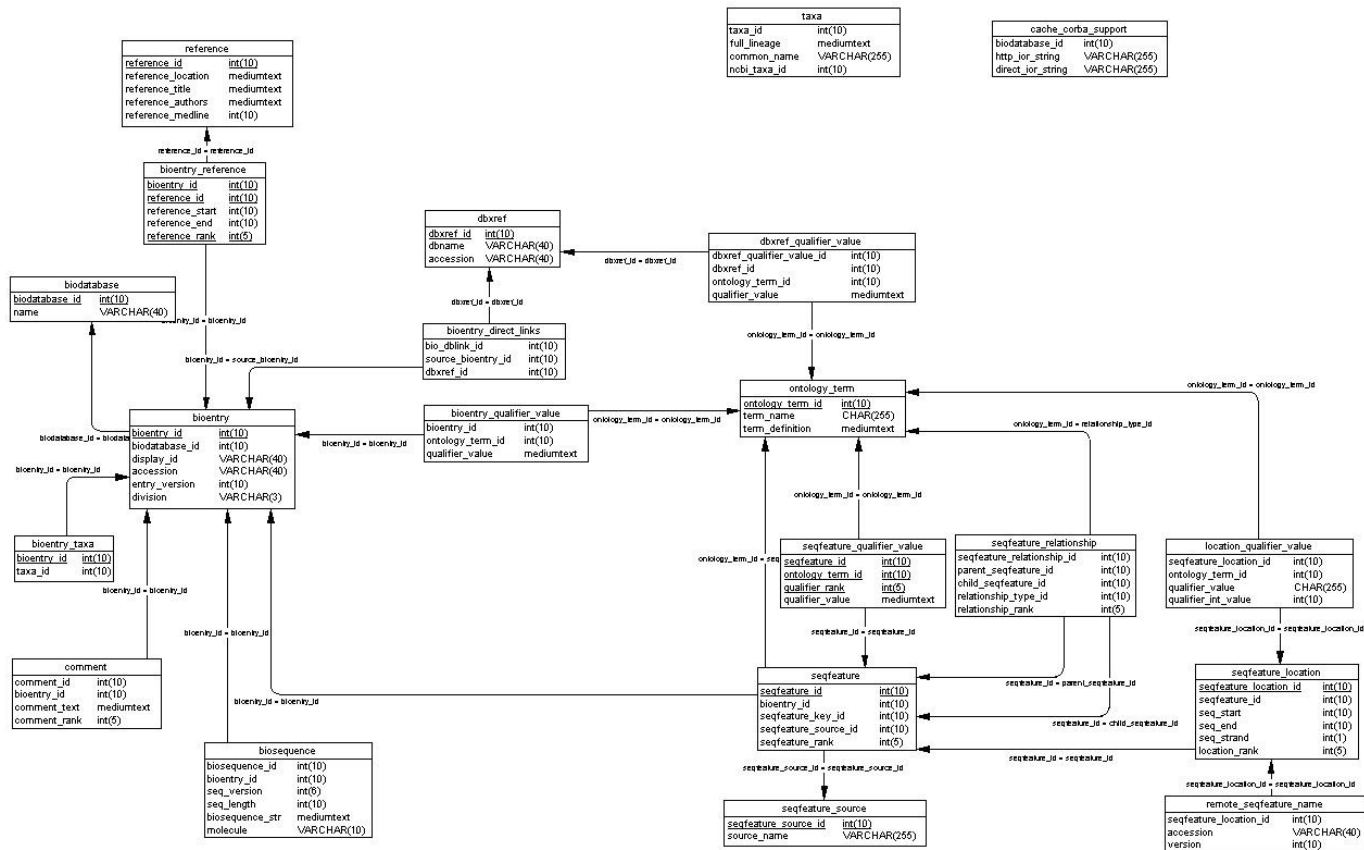
gi	annotation
5693	Trypanosoma cruzi chromosome 3, ORF 1234, similar to gi 12345 AF934567 caseine kinase (Candida albicans)
5694	Candida albicans hypothetical protein in region 21922..24568
5695	Sarcocystis cruzi 16SRNA gene
5696	Lutzomyia cruzi cytochrome b; best similarity to gi 1234568

gi	Organism	Annotation	similar to
5693	Trypanosoma cruzi	Chromosome 3, ORF 1234	12345
5694	Candida albicans	Hypothetical protein in region 21922..24568	
5695	Sarcocystis cruzi	16S RNA gene	786512
5696	Lutzomyia cruzi	Cytochrome b	1234568

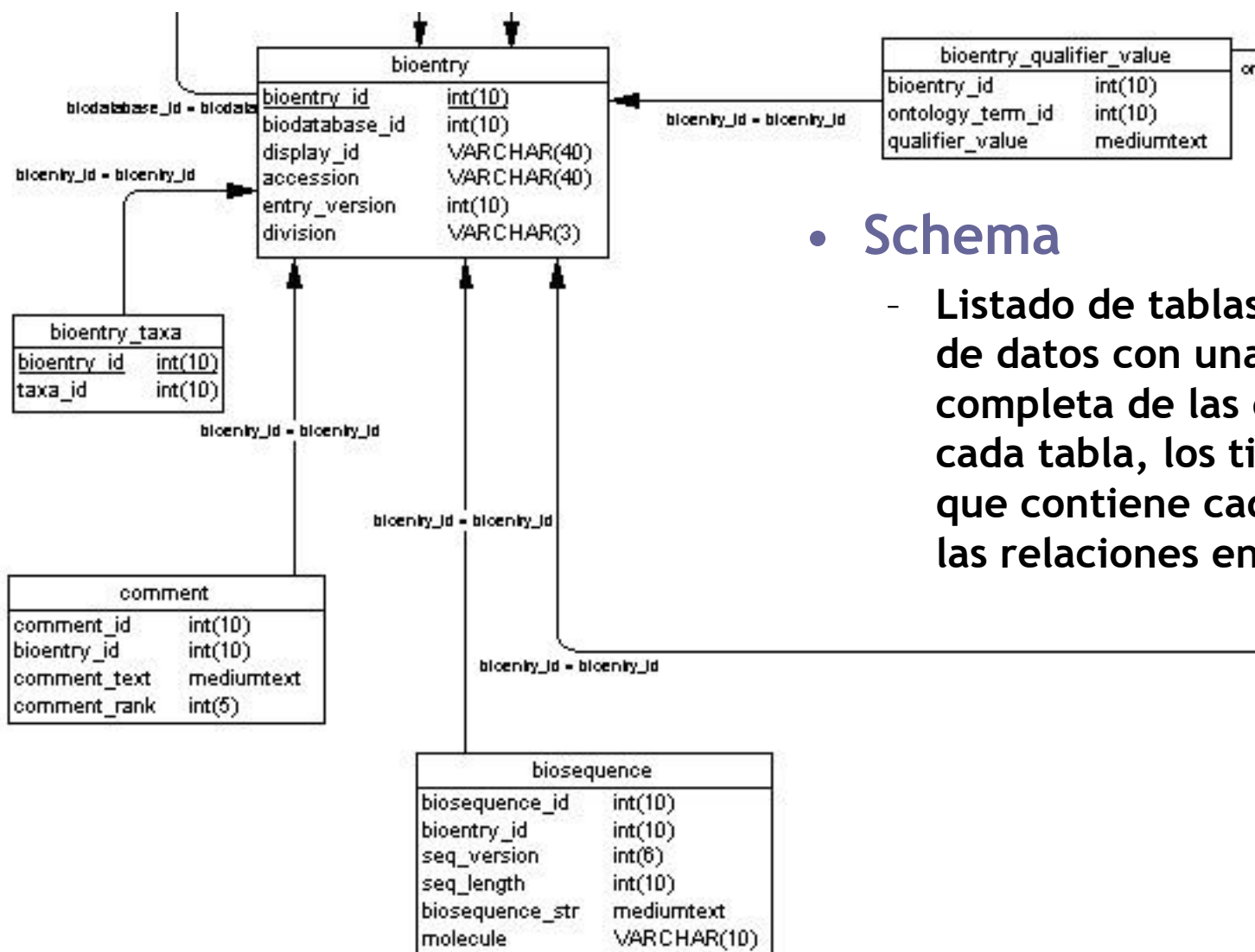


Schemas

- La distribución de los datos en campos dentro de una tabla y de las relaciones entre tablas y sus campos es lo que se llama el diseño o **schema**



Schemas (cont)



- Schema

- Listado de tablas de una base de datos con una descripción completa de las columnas de cada tabla, los tipos de datos que contiene cada columna y las relaciones entre tablas.

Representación relacional de la información

- Qué criterio usamos para diseñar el schema?
- Cómo distribuimos los datos en tablas/columnas?
- Distintas cosas a tener en cuenta:
 - Eficiencia (economía) al almacenar datos: normalización
 - Consultas que planeamos hacer sobre nuestra base de datos y en el tipo de datos.

Relaciones entre los datos

- Ejemplos de relaciones
 - Proteins ↔ Bibliographic references

Proteins

Accession	Description	MW	pI
AF1234	Malate dehydrogenase	36000	6.4
AM44432	Cysteine proteinase	45000	4.5

Linking table

Accession	PubMed ID
AF1234	1234556
AF1234	23445

Bibliographic References

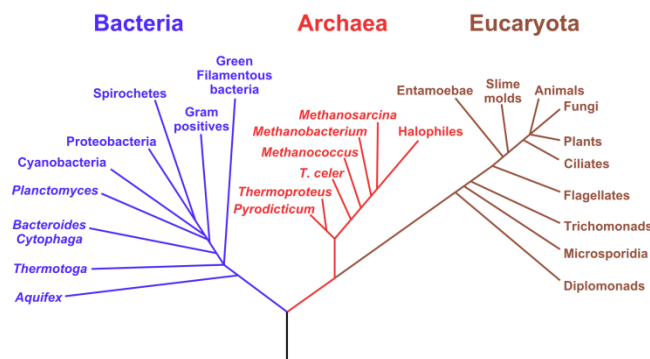
PubMed ID	Journal	Year	Title	Vol
1234556	J Biol Chem	1978	The malate dehydrogenase ...	5
23445	Biochem J	1982	A malate dehydrogenase from ...	13

Representación de árboles y grafos

- Ejemplo: representación en forma relacional de árboles y grafos
 - Información estructurada jerárquicamente
 - Taxonomy (NCBI), SCOP (Structural Classification of Proteins)



Phylogenetic Tree of Life



Relational modeling of biological data: trees and graphs. Aaron J. Mackey. <http://www.oreillynet.com/pub/a/network/2002/11/27/bioconf.html>

Ejemplo: adjacency list

	Campo	Tipo de dato
PK	Taxon_id	Entero
FK	Parent_id	Entero (ref a PK)
	Nombre	texto

Este tipo de representación se conoce como 'adjacency list':

Cada relación jerárquica 'padre-hijo' está definida en forma explícita.

Taxon_id	Parent_id	nombre
1	-	raíz
2	1	Bacteria
2157	1	Archaea
2759	1	Eukaryota
1224	2	Proteobacteria
...
543	1236	Enterobacteriaceae
561	543	Escherichia
562	561	Escherichia coli
83333	562	Escherichia coli K12

Adjacency list: consultas

- Qué consultas podemos hacer sobre los datos organizados en forma de 'adjacency list'?
 - Podemos encontrar el taxón inmediatamente superior de cualquier elemento taxonómico.
 - Podemos encontrar taxones terminales sin 'hijos'
 - Podemos encontrar un taxón (o taxones) buscándolos por nombre
- Y cuáles son difíciles de hacer con esta representación de los datos?
 - Podemos encontrar todos los taxones 'hijos' de un determinado taxón?
 - Ejemplos típicos de este tipo de consultas: buscar todos los mamíferos, todos los vertebrados, o todos los miembros del orden Apicomplexa.
 - Cómo harían esta consulta? Es posible responder estas preguntas con una única consulta sobre la base de datos? Cuántas consultas deberían hacer?

Eukaryota; Fungi/Metazoa group; Metazoa; Eumetazoa; Bilateria; Coelomata; Deuterostomia; Chordata; Craniata; Vertebrata; Gnathostomata; Teleostomi; Euteleostome; Sarcopterygii; Tetrapoda; Amniota; Mammalia; Theria; Eutheria; Primates; Catarrhini; Hominidae; Homo/Pan/Gorilla Group; Homo; Homo sapiens

Representación relacional de árboles: nested set

	Campo	Tipo
PK	Taxon_id	entero
FK	Parent_id	entero
	Left_id	entero
	Right_id	entero
	Nombre	texto

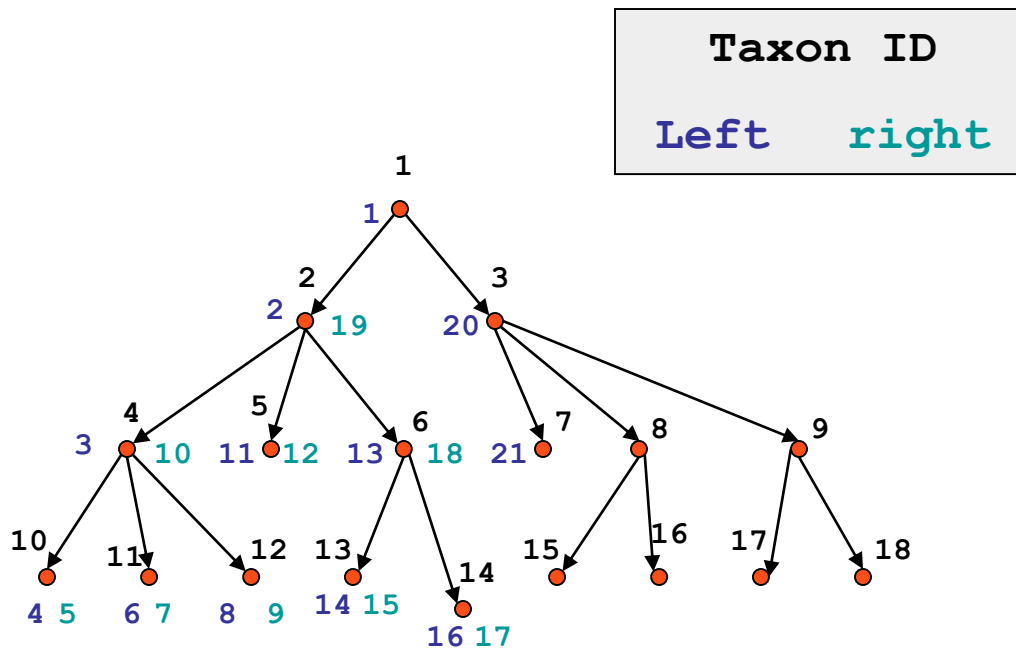
Los valores **left** y **right** son números arbitrarios, pero deben cumplir con la siguiente propiedad:

Para cada par 'padre-hijo' los valores del hijo tienen que estar dentro de los valores del padre.

Taxon	Nombre	Parent	Left	Right
1	Root	NULL	1	323458
2	Bacteria	1	21703	87862
3	Archaea	1	87863	92266
4	Eukaryota	1	92267	323456
1224	Proteobacteria	2	23982	49591
...
543	Enterobacteriaceae	1236	26681	27938
561	Escherichia	543	26852	26891
562	Escherichia coli	561	26853	26868
83333	Escherichia coli K12	562	26856	26857

Nested set representation: como calcular left/right?

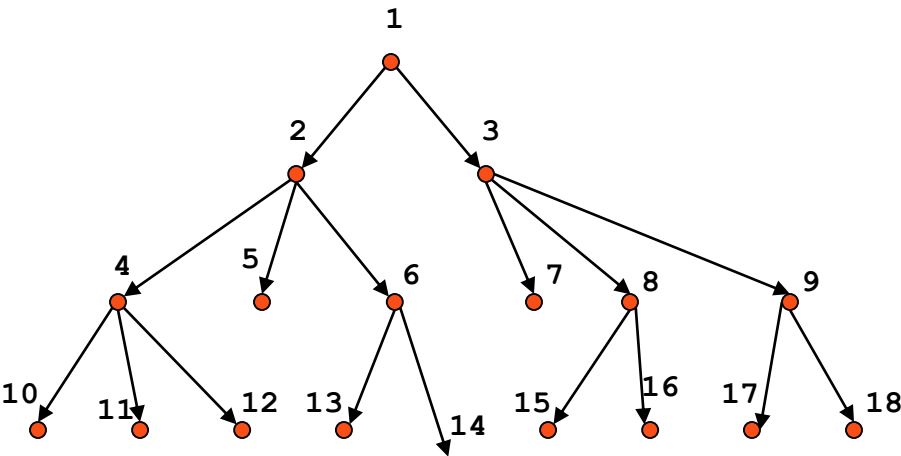
- Cómo se generan los valores para **left** y **right**?
 - Hay que recorrer el árbol asignando estos valores



- Arboles / Grafos
 - Hay distintas maneras de recorrerlos
 - Depth-first
 - Breadth-first

Materialized Paths

	Campo	Tipo
PK	Taxon_id	entero
	Name	Texto
	Path	Texto



Taxon	Nombre	Path
1	Root	1
2	Bacteria	1.2
3	Archaea	1.3
4	Proteobacteria	1.2.4
5	Cyanobacteria	1.2.5
6	Actinobacteria	1.2.6
7	Crenarchaeota	1.3.7
8	Euryarchaeota	1.3.8
9	Thaumarchaeota	1.3.9
10	Alfa-Proteobacteria	1.2.4.10
11	Gamma-proteobacteria	1.2.4.11
12	Delta-proteobacteria	1.2.4.12
13	Coriobacteridae	1.2.6.13
14	Actinobacteria	1.2.6.14
15	Methanobacteria	1.3.8.15
16	Thermococci	1.3.8.16
17	Cenarchaeales	1.3.9.17
18	Nitrosopumiales	1.3.9.18

En este diseño, el **camino (path)** hacia cada nodo del árbol, está incluido en forma explícita en la información asociada a cada nodo.

Materialized Paths: consultas

- **Buscar un nodo y todos sus parentales**
 - Ej: buscar todos los parentales del nodo 'Thermococci'
 - Buscar todos los registros cuyo Path **esté contenido** dentro del nodo de interés
 - Path del nodo 'thermococci' = 1.3.8.16
 - Lista de Paths que cumplen la condición,
 - 1.3.8 (Euryarchaeota), 1.3 (Archaea), 1 (root)
- **Buscar un nodo y todos sus descendientes (directos o indirectos)**
 - Ej: buscar todos los descendientes del nodo 'Bacteria'
 - Buscar todos los registros cuyo Path **contenga** al del nodo de interés
 - Path del nodo de 'Bacteria' = 1.2
 - Lista de Paths que cumplen con la condición,
 - 1.2.4 (Proteobacteria), 1.2.5 (cyanobacteria), 1.2.6 (Actinobacteria), 1.2.4.10 (alpha-proteobacteria), 1.2.4.11 (gamma-proteobacteria), 1.2.4.12 (delta-proteobacteria), etc.

Entity-Attribute-Value

- También: Object-Attribute-Value
- Usado en casos en donde el número de **atributos** (propiedades, parámetros) utilizados para describir **algo** (un objeto o entidad) es muy grande pero el número de atributos que realmente se utilizan es variable y pequeño.
- El caso más común es el de *historias clínicas* de pacientes
 - Cientos de miles de atributos que se pueden medir, diagnosticar, o evaluar
 - En la consulta el médico pregunta de acuerdo a los síntomas que describe el paciente (filtra atributos) y finalmente se almacena en la base de datos aquellos que son **relevantes**.

Entity-Attribute-Value

- **Modelar estos datos de la manera tradicional**
 - Una tabla con miles de columnas (una por cada posible atributo)
 - El seguimiento en el tiempo de un paciente implica agregar una fila por cada consulta.
 - En cada fila hay sólo unos pocos hallazgos (positivos), el resto de las columnas están vacías (NULL).
- **Modelar estos datos usando el modelo Entity-Attribute-Value**
 - Una única tabla con tres columnas:

Tabla de objetos (entidades)

ID	Nombre	Apellido	...
Paciente 1	Tito	Chocola	

Tabla de datos

Entity	Attribute	Value
Paciente 1	1	33
Paciente 1	15	230
Paciente 1	56	

Tabla de atributos

Attr ID	Name	Description	Data type	Units	Input validation
1	Edad	Edad ...	Integer	Años	\d+
15	Colesterol en sangre	Descripcion ...	Float	Mg/ml	\d+\.\d*

Structured Query Language

- **SQL - Structured Query Language**

- Es un lenguaje utilizado por todos los sistemas de manejo de bases de datos relacionales
 - Oracle, Sybase, PostgreSQL, MySQL, SQLite, etc.
- Permite definir tablas, relaciones (DDL)
- Y hacer consultas (DML)

- **DDL - Data Definition Language**

- Subset de SQL utilizado para crear bases de datos, tablas, definir campos, etc.
- CREATE DATABASE, CREATE TABLE
- DROP DATABASE, DROP TABLE,
- ALTER TABLE,

- **DML - Data Manipulation Language**

- Subset de SQL utilizado para hacer consultas, insertar y actualizar datos, etc.
- SELECT FROM TABLE, INSERT INTO TABLE
- UPDATE TABLE
- DELETE FROM TABLE

SQL - Un ejemplo de consulta

Proteins

Accession	Description	MW	pI
AF1234	Malate dehydrogenase	36000	6.4
AM44432	Cysteine proteinase	45000	4.5

Linking table

Accession	PubMed ID
AF1234	1234556
AF1234	23445

Bibliographic References

PubMed ID	Journal	Year	Title	Vol
1234556	J Biol Chem	1978	The malate dehydrogenase ...	5
23445	Biochem J	1982	A malate dehydrogenase from ...	13

```
SELECT accession, description, journal, year, vol, pages, ...
FROM proteins, bibliographic_references, linking_table
WHERE linking_table.accession = proteins.accession
AND linking_table.pubmed_id = bibliographic_references.pubmed_id
AND proteins.mw <= 36000;
```

SQL - Un ejemplo de manipulación de datos

Proteins

Accession	Description	MW	pI
AF1234	Malate dehydrogenase	36000	6.4
AM44432	Cysteine proteinase	45000	4.5

Linking table

Accession	PubMed ID
AF1234	1234556
AF1234	23445

Bibliographic References

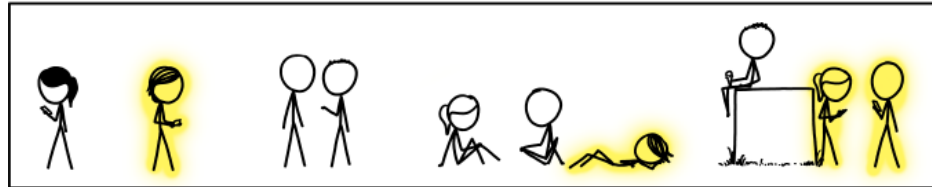
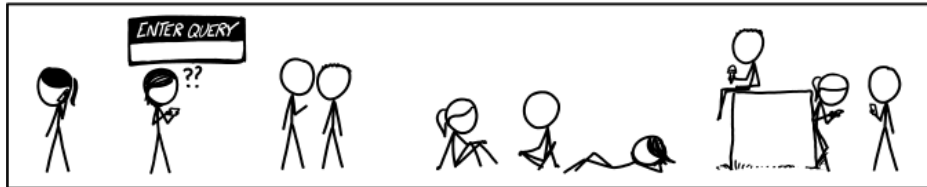
PubMed ID	Journal	Year	Title	Vol
1234556	J Biol Chem	1978	The malate dehydrogenase ...	5
23445	Biochem J	1982	A malate dehydrogenase from ...	13

INSERT INTO proteins (accession, description, mw, ...)
VALUES ('AF1234', 'Malate dehydrogenase', '36000', ...);

UPDATE proteins **SET** mw = 45000 **WHERE** accession = AM44432;

DELETE FROM proteins **WHERE** accession = AF1234;

SQL Query (comic)



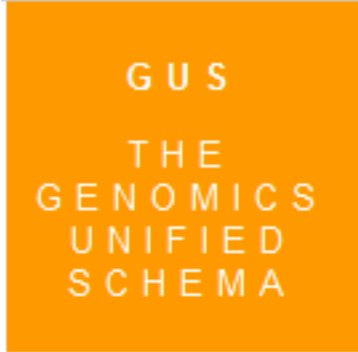
XKCD: <http://xkcd.com/1409>

- Reinventar la rueda

- Cuántas maneras hay de organizar información biológica en forma de tablas en una base de datos relacional?
- secuencias + anotación?
- Secuencias + features (propiedades de la secuencia localizables)
- ...

- Después de haber reinventado muchas ruedas ...

- GUS, Genomics Unified Schema
 - PlasmoDB, ToxoDB, CryptoDB (ApiDB), TcruziDB, Allgenes.org,
- Chado, The GMOD Database Schema
 - Wormbase, FlyBase, TaiR, Gramene, SGD, DictyBase



- **Qué es?**
 - Extensive relational database schema
 - Associated application framework
- **Para que se usa?**
 - Para almacenar, integrar, analizar y presentar datos genómicos
- **Modular schema:**
 - Core: tablas conteniendo información de GUS (housekeeping)
 - DOTS: tablas para almacenar información sobre secuencias, genes,
 - SRES: resource tables (to store external resources, controlled vocabularies)
 - RAD: microarray data
 - TESS: transcription binding, transcription factors
 - PROT: proteomics

CHADO: The GMOD schema

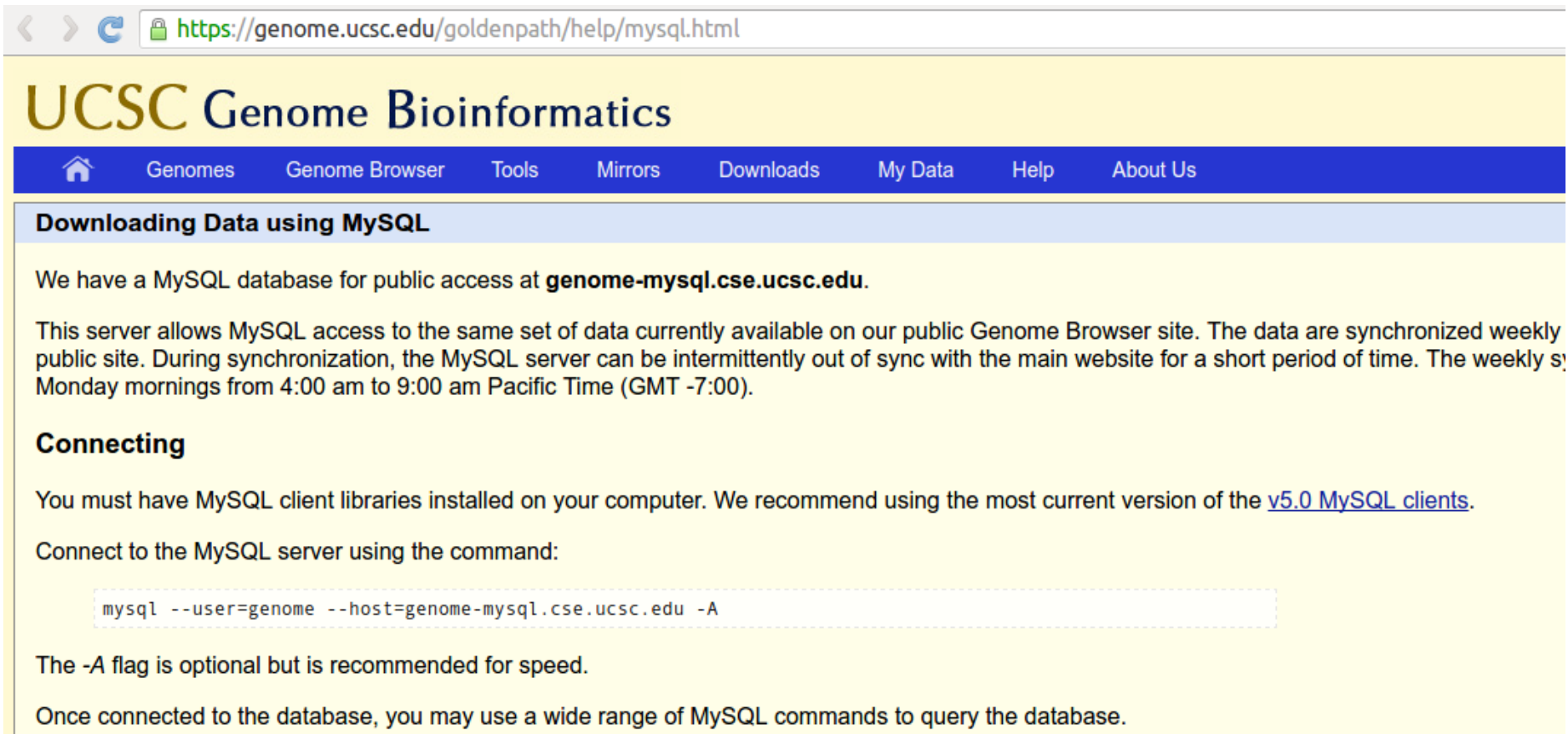


- GMOD = Generic Model Organism Database
- CHADO = DB Schema that underlies many GMOD installations
- Capable of representing many of the general classes of data frequently encountered in modern biology

- **Modular schema**

- Companalysis, for data derived from computational analysis
- Contact, for people, groups, organizations
- Controlled vocabularies
- Expression
- General (for accession numbers and identifiers)
- Genetic
- MAGE, microarray data
- Phenotype,
- Organism, for taxonomic data)
- Publication, for publication references
- Sequence, for sequence, annotation, and features
- Stock, for specimens and biological collections

UCSC Genome Browser is one of the Human Genome Reference sites (also has many other genomes)



The screenshot shows a web browser window with the address bar displaying <https://genome.ucsc.edu/goldenpath/help/mysql.html>. The page header features the UCSC Genome Bioinformatics logo and a navigation menu with links: Home, Genomes, Genome Browser, Tools, Mirrors, Downloads, My Data, Help, and About Us. The main content area is titled "Downloading Data using MySQL" and contains the following text:

We have a MySQL database for public access at **genome-mysql.cse.ucsc.edu**.

This server allows MySQL access to the same set of data currently available on our public Genome Browser site. The data are synchronized weekly with the public site. During synchronization, the MySQL server can be intermittently out of sync with the main website for a short period of time. The weekly synchronization occurs on Monday mornings from 4:00 am to 9:00 am Pacific Time (GMT -7:00).

Connecting

You must have MySQL client libraries installed on your computer. We recommend using the most current version of the [v5.0 MySQL clients](#).

Connect to the MySQL server using the command:

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A
```

The -A flag is optional but is recommended for speed.

Once connected to the database, you may use a wide range of MySQL commands to query the database.

- **Relational Database Management Systems**
 - **Comerciales**
 - Oracle, Sybase
 - **Open source, gratuitos**
 - PostgreSQL, MySQL
- **Todos usan SQL (standard query language) para**
 - **crear tablas, índices, etc.**
 - CREATE TABLE **taxon** (**taxon_id** integer, **name** text, PRIMARY KEY(**taxon_id**))
 - ALTER TABLE **taxon** INDEX (name)
 - **ingresar datos**
 - INSERT INTO **taxon** (**taxon_id**, **name**) VALUES (1, root);
 - UPDATE **taxon** SET **name** = "Trypanosoma cruzi" WHERE **name** = "Schizotrypanum cruzi"
 - **consultar**
 - SELECT **name** FROM **taxon** WHERE **taxon_id** = 1;
 - SELECT **taxon_id**, **name** FROM **taxon** WHERE **taxon_id** IN ('12', '15', '345', '1823')

Búsquedas en una base de datos: índices

- Para facilitar las búsquedas en una base de datos, se construyen índices.
- Un índice es una lista de claves primarias asociadas a un determinado campo (o grupo de campos)

gi	Accession	version	date	Genbank Division	taxid	organims	Number of Chromosomes
6226959	NM_000014	3	01/06/2000	PRI	9606	homo sapiens	22 diploid + X+Y
6226762	NM_000014	2	12/10/1999	PRI	9606	homo sapiens	22 diploid + X+Y
4557224	NM_000014	1	04/02/1999	PRI	9606	homo sapiens	22 diploid + X+Y
41	X63129	1	06/06/1996	MAM	9913	bos taurus	29+X+Y

gi Accession
6226959 NM_000014
6226762 NM_000014
4557224 NM_000014
41 X63129

Indices (cont)

- Un ejemplo más complejo: buscar todos los records que contengan la palabra 'kinase' en la descripción de la secuencia

gi	acc	def
214734077	20	Xenopus laevis rhodopsin n
123456743	567	Mus musculus casein kinase

•Indexar la columna 'def'

word	list of GIs
casein	1234, 3245, 43678, 123456 ...
kinase	432, 5678, 32456, 123456 ...
laevis	36314, 214734, ...
mus	23467, 98732, 123456, 312456, 567
muscu	123467, 98732, 123456, 567983 ...
rhodopsin	214734, 223466, 873212, 23587, 29
xenopus	23462, 36314, 98476, 214734 ...

Indexar es costoso

- El proceso de indexación es costoso en términos computacionales, pero se realiza una única vez (en realidad cada vez que se actualizan los datos)
- Desde el punto de vista de la base de datos, los índices no son otra cosa que nuevas tablas relacionadas con la tabla que contiene el campo indexado
- Ejemplo más obvio: buscadores de páginas de internet (Google, Altavista). Visitan páginas e indexan los términos que encuentran
 - keyword: url1, url2, url3, url4, etc.

- **Son estructuras de datos utilizadas para acelerar la búsqueda de relaciones (tuples) que cumplan alguna determinada condición**
 - **Igualdad: encontrar Discos donde Banda = Tipitos**
 - **Otras condiciones son posibles: rangos**
 - **Encontrar Discos donde Año de Lanzamiento (AL) sea**
 - **$AL < 1990$ y $AL > 1980$**
- **Hay muchos tipos de Indices**
 - **Convencionales**
 - **B-Trees**
 - **Hashing indexes**
- **Se evalúan de acuerdo a**
 - **Tiempo de acceso**
 - **Tiempo que lleva insertar un dato**
 - **Tiempo que lleva borrar un dato**
 - **Espacio en disco que ocupan**

Indices convencionales

- **Similares al índice de un libro**
 - El índice contiene una entrada, con un puntero (número de página) al lugar donde están los datos
- **Sparse vs Dense (más o menos densos)**
 - Dense: hay una entrada para cada clave asociada a un objeto
 - Sparse: hay una entrada para algunas claves solamente

Indice (genes)	Genes	Tiempo	Expresión
ABC1	ABC1	1	0.2
BRC2	BRC2	1	0.8
CAM3	BRC2	2	0.3
DHFR	CAM3	2	0.25
EGF-1	DHFR	1	0.1
	DHFR	2	0.3
	DHFR	2	0.4
	EGF-1	1	0.3

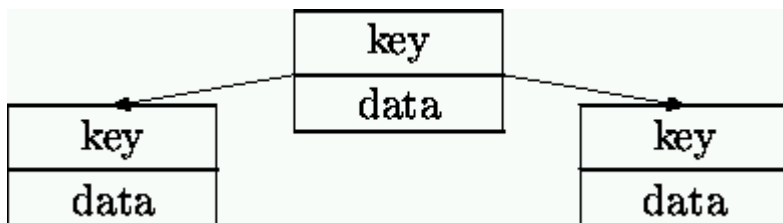
- **Ventajas / Desventajas?**
 - Los índices densos son más rápidos
 - Los índices dispersos ocupan menos espacio
 - **Cuál es el límite a partir del cuál un índice *disperso* se vuelve *denso* ?**
 - Los índices dispersos pueden ajustarse
 - Cuantas claves nos salteamos?
 - Densidad de claves
 - Evaluar # total de filas, # de entradas por cada clave

Multi-level indexes

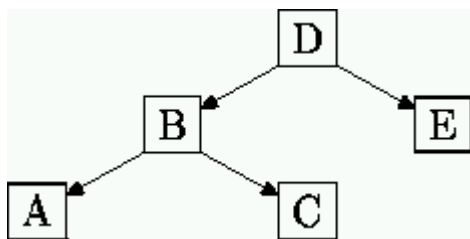
- Los índices convencionales pueden ser muy grandes
- La idea de estos índices es que reduzcan el acceso a disco
- La unidad mínima de I/O en una computadora es mover un bloque de datos del disco a memoria
- Ejemplo
 - Un archivo con 100,000 registros, con 10 datos x gen
 - Un índice disperso, con una entrada x gen: tendríamos 10,000 filas
 - Si asumimos que en un bloque de I/O entran 100 filas, necesitamos acceder a 100 bloques.
- Es deseable mantener los índices en memoria RAM
- Los índices se vuelven costosos cuando crecen los datos
 - Que pasaría en el caso de tener millones de registros?

Binary trees

- Arboles: nodos conectados con vértices (grafos)



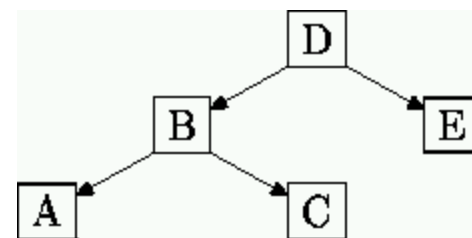
- Para hacer búsquedas, los datos se ignoran.
- Es como si el árbol solo tuviera 'claves'



Binary trees

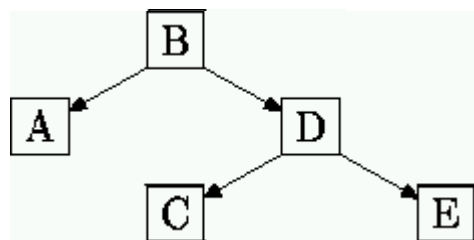
Reglas para moverse en el árbol

- Ejemplo, buscar A
- Empezar en la raíz, si encontramos A listo!
- Si no, nos movemos, de esta forma:
 - Si la clave del nodo es $< A$, a la izquierda
 - Si la clave del nodo es $> A$, a la derecha



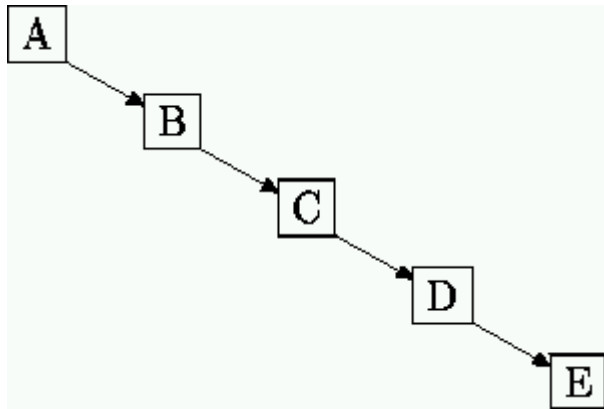
- **Armando un árbol: BDCAE**

- Empezamos por la raíz
- Agregamos nodos siguiendo las mismas reglas de movimiento



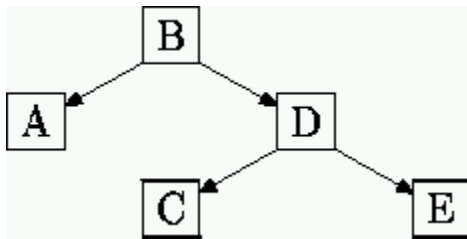
Profundidad promedio de este árbol: **1.5**
 $(1 + 1 + 2 + 2) / 4$

- **Armando un árbol: ABCDE**
 - El orden de los datos afecta el balance del árbol
(la distribución de las ramas)



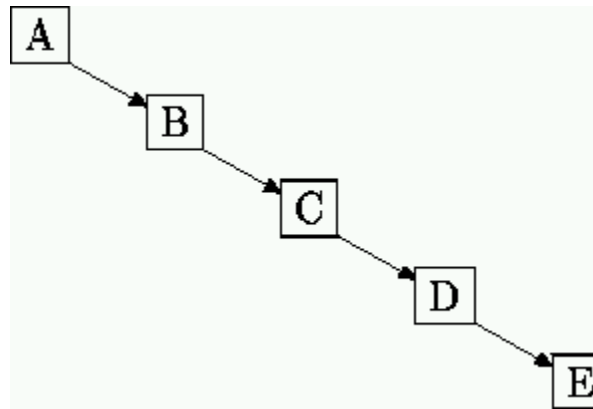
Más sobre 'binary trees'

- **Arboles balanceados vs no balanceados**
 - Profundidad es inversamente proporcional a la velocidad de las búsquedas



2

1.5



4

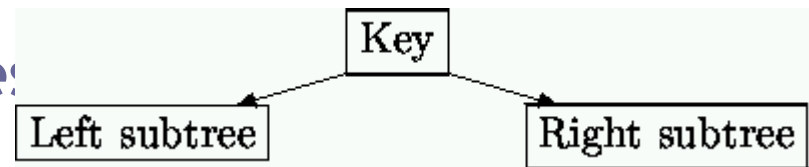
2.5

Profundidad máxima

Profundidad promedio

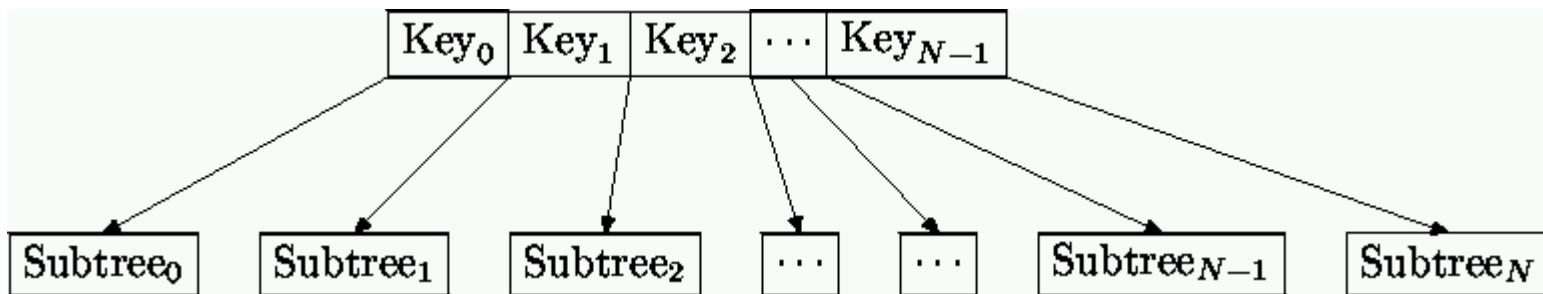
- B-Trees are not 'binary trees'

- Bushy Trees, B

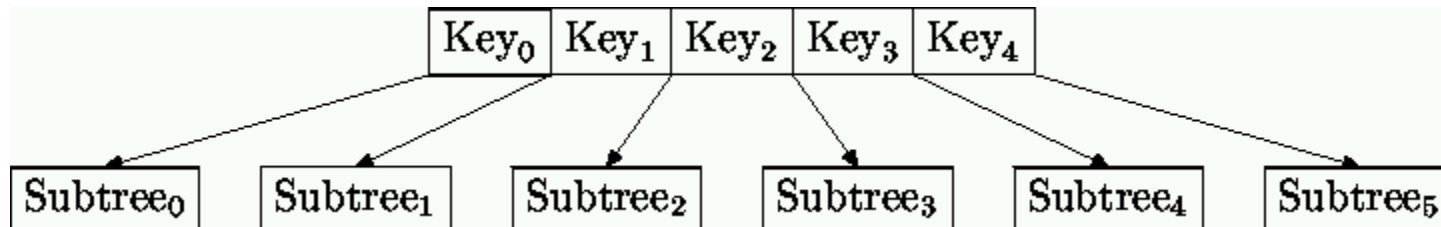


- Son una generalización de los árboles binarios

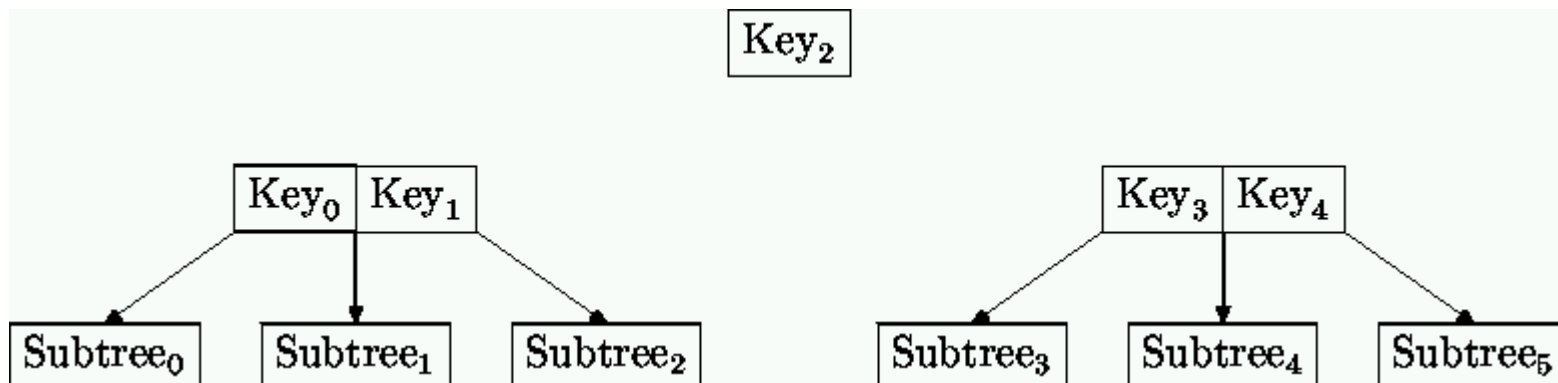
- Los nodos pueden contener más de una clave
 - Las claves dentro de un nodo están **ordenadas**
 - B-Tree de orden 2 => cada nodo contiene a lo sumo 3 claves
 - B-Tree de orden 3 => cada nodo contiene a lo sumo 4 claves
 - B-Tree de orden n => a lo sumo $n-1$ claves



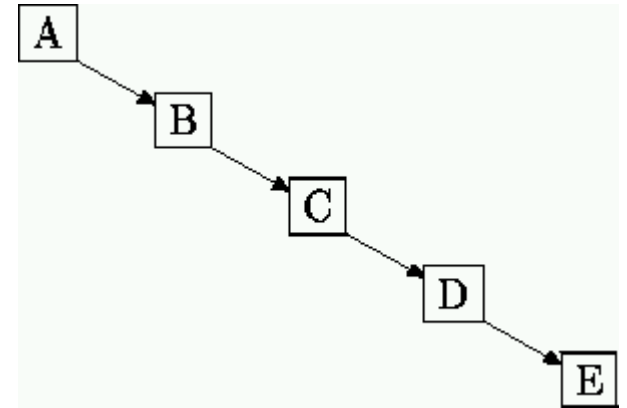
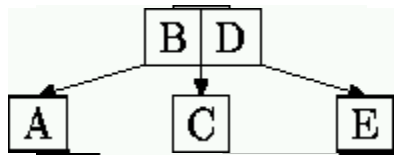
- **B-Tree de orden 5**
 - A lo sumo 4 claves por nodo
 - Agrego una nueva clave (un nuevo objeto/elemento)



- Como me excedo del límite, parto el nodo a la mitad



- **Ejemplo: ABCDE**
 - Esta cadena de texto, en un árbol binario, da un árbol desbalanceado
 - Probemos usando un B-Tree de orden 3



- **B-trees son estructuras de datos especializadas**
 - Uso en discos (lento)
 - Almacenamiento de grandes volúmenes de datos
- **Permiten realizar búsquedas extremadamente rápidas**
 - No se recorren todos y cada uno de los nodos para obtener una respuesta

Recorriendo árboles

- **Depth-first**
 - Recorrido en profundidad primero
 - Se visita cada nodo 3 veces
 - Al visitarlo por primera vez (desde el nodo parental)
 - Al visitarlo por segunda vez desde el nodo hijo izquierdo
 - Al visitarlo nuevamente (desde el nodo hijo derecho)
- **Breadth-first (level order)**
 - Recorrido exhaustivo de cada nivel de profundidad del árbol (hacia lo ancho)
- **Ejemplos interactivos:**
 - <http://nova.umuc.edu/~jarc/idsv/lesson1.html>
 - <https://www.cs.usfca.edu/~galles/visualization/BTree.html>

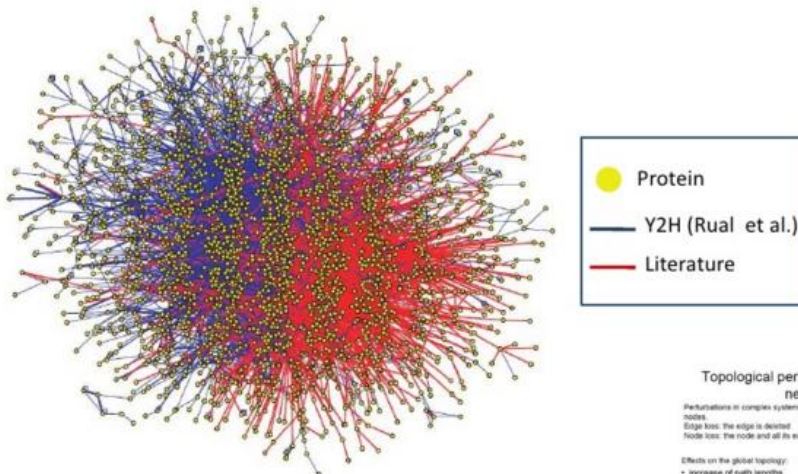
What is NoSQL?

- It's not “No to SQL”
- It's not “Never SQL”
 - **It's “Not Only SQL”**

Graph Databases: Neo4j

Nodes and **Edges** are used to represent and store information

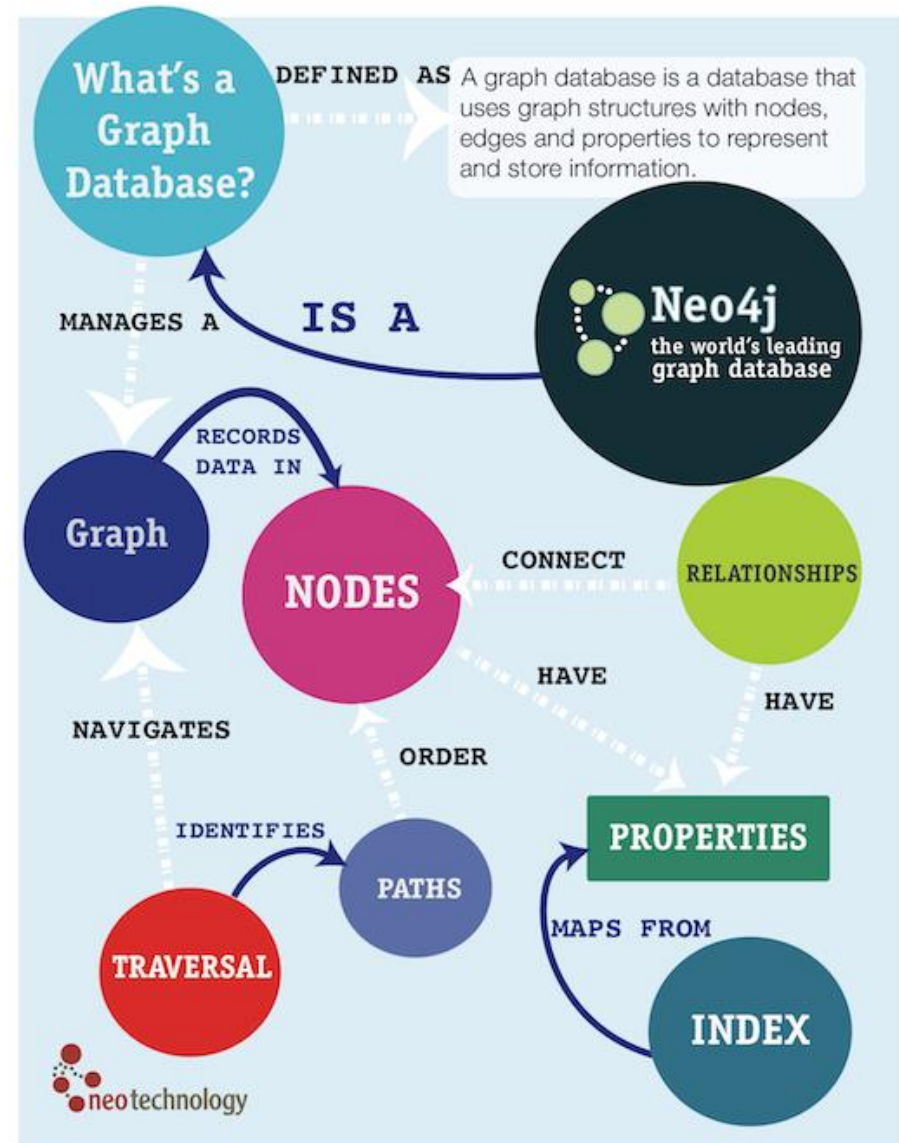
Human Interactome



Human: Rual et al 2005; Stelzl et al 2005.
Drosophila: Giot et al. 2003.
C. elegans: Li et al 2004.

Yeast: Yu et al. 2008; Kr
2006; Gavin et al. 2006;
2001; Uetz et al 2000.

Topological pertu
netv
Perturbations in complex systems o
nodes.
Edge loss: the edge is deleted
Node loss: the node and all its edge
Effects on the global topology:
- increase of path lengths,
- separation into isolated
clusters.
More connected network - less
But bridges are definite point
The effect of a node removal
characteristics of its edges.



Importante: no se busca en el total de los datos disponibles, sino sobre un subset pre-computado.

- Buscadores de páginas en internet
- PubMed / Entrez / EMBL-EBI
- BLAST

Motores de búsqueda: búsquedas simples

- Los motores de búsqueda ofrecen búsquedas simples
- No imponen restricciones
- El usuario tipea palabras libremente
- Usan estrategias para intentar “adivinar” la intención del usuario (sobre qué campo de la base de datos buscar)

Ejemplo: term mapping - Entrez (PubMed)

- Entrez busca en una serie de listas para ver si la palabra que ingresaron se encuentra en alguna
- **MeSH (Medical Subject Headings):** vocabulario controlado utilizado para indexar artículos en PubMed.
- **Journals:** nombre completo del journal, abreviaturas usadas en MEDLINE y números ISSN.
- **Lista de frases:** cientos de miles de frases generadas a partir de MeSH y otros vocabularios controlados similares.
- **Indice de autores:** apellido e iniciales.
- **Stopwords:** palabras comunes, presentes en casi todos los registros de la base de datos (a, an, by, of, the ...)

Búsquedas simples: pros / cons

- **Ventajas**

- rápidas de formular
- no hay que leer el manual
- ni hacer un curso 😊

- **Desventajas**

- poco selectivas

Búsquedas avanzadas

- Presuponen un cierto conocimiento sobre la organización subyacente de los datos
- Hay que especificar sobre qué campos buscar:
⇒ hay que conocer los campos
- **Entrez:** se especifican entre corchetes
- Tags predefinidos (hay que conocerlos)
 - `Escherichia coli[organism]`
 - `review[publication type]`
 - `attenuator[feature key]`
- **EMBL-EBI:** formulario avanzado (no hay que conocer términos o tags)

Advanced UniProt searches at EMBL-EBI

www.uniprot.org/#

UniProtKB

Advanced Search

BLAST Align Retrieve/ID Searching in UniProtKB ? Help

The mission of UniProt is to

UniProtKB
UniProt Knowledgebase

Swiss-Prot (549,008)
Manually annotated and reviewed.

TrEMBL (50,011,027)
Automatically annotated and not reviewed.

Organism [OS] Trypanosoma cruzi [569]

Protein name [DE] mucin

AND

Supporting data

- Literature citations
- Cross-ref. databases
- Taxonomy
- Diseases
- Subcellular locations
- Keywords

UniProt release 2015_07

Coding-non-coding RNAs: a game of hide-and-seek | Cross-references to ESTHER and Genevisible | Removal of

News archive

Búsquedas avanzadas: Entrez

- Entrez

- L

- bú

- los

- Hi

- En

- so

- Pr

- y

- y

- De

- de

- et

Nucleotide

Nucleotide

mucin

Search

Create alert Advanced

Display Settings: Summary, 20 per page, Sorted by Default order

Send to:

Items: 1 to 20 of 37617

<< First < Prev Page 1 of 1881 Next > Last >>

Found 40931 nucleotide sequences. Nucleotide (37617) EST (3307) GSS (7)

Species

Animals (27,096)

Plants (553)

Fungi (1,027)

Protists (4,804)

Bacteria (2,897)

Archaea (18)

Viruses (19)

Customize ...

Molecule types

genomic DNA/RNA (22,019)

mRNA (15,169)

rRNA (2)

Customize ...

Source databases

INSDC (GenBank) (11,997)

RefSeq (25,527)

Customize ...

Genetic compartments

Mitochondrion (1)

Plasmid (39)

Sequence length

Custom range...

Release date

Custom range...

Revision date

Custom range...

Clear all

Show additional filters

1. 1,341 bp linear mRNA

Accession: M76740.1 GI: 205545

[GenBank](#) [FASTA](#) [Graphics](#)

2. 884 bp linear mRNA

Accession: M57417.1 GI: 188877

[GenBank](#) [FASTA](#) [Graphics](#)

3. 1,800 bp linear mRNA

Accession: S78981.1 GI: 1042036

[GenBank](#) [FASTA](#) [Graphics](#)

4. 1,431 bp linear mRNA

Accession: Z34277.1 GI: 563374

[GenBank](#) [FASTA](#) [Graphics](#)

5. 1,541 bp linear mRNA

Accession: L42292.1 GI: 808736

[GenBank](#) [FASTA](#) [Graphics](#)

6. 3,811 bp linear mRNA

55

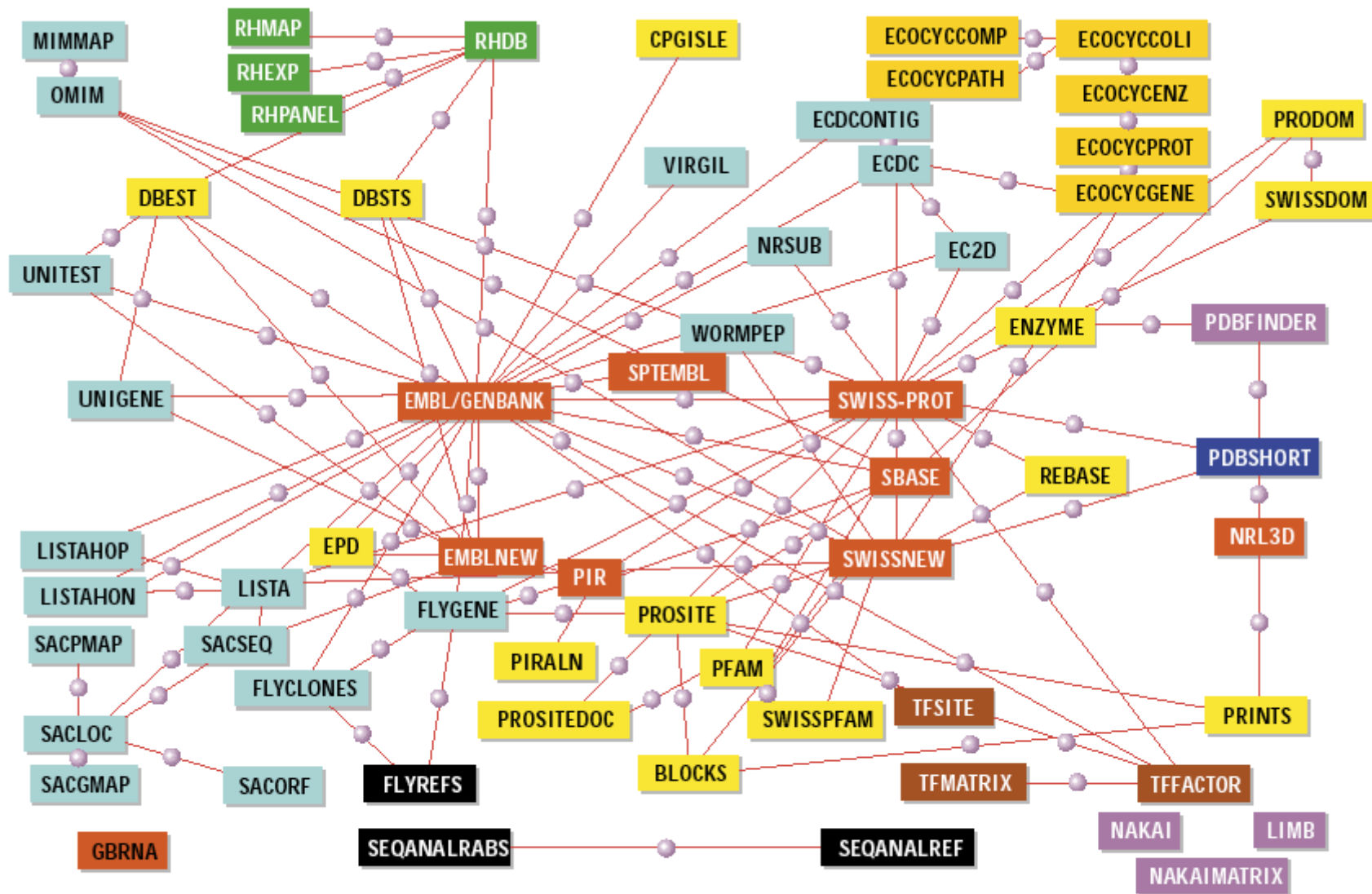
Fernán Agüero

Operadores lógicos

- En búsquedas simples o avanzadas siempre tienen a disposición operadores lógicos para encadenar términos
- **AND (intersección)**
 - human AND genome
 - +human +genome
 - human && genome
- **OR (unión)**
 - human OR genome
 - human | | genome
- **NOT (subconjunto)**
 - human NOT genome

Orden de los términos en un query

- El orden de los términos es importante
- Un query se evalúa de izquierda a derecha
 - **human NOT genome** no es lo mismo que **genome NOT human**
- Si el query tiene muchos términos pueden forzar el orden de evaluación usando paréntesis
 - **human AND cancer AND (cell OR science OR nature)**
 - **casein kinase NOT (human OR mouse)**



■ Sequence
 ■ Protein Structure
 ■ SeqRelated
 ■ Genome
 ■ Metabolic
 ■ Literature
 ■ Others
 ■ Transfac
 ■ Mapping