Resumen de Bases de Datos

Última clase auxiliar
de CC42A / CC55A

Repaso para el examen

Mauricio Monsalve M.

Principales temas a manejar

- Modelamiento de datos: modelo ER y modelo relacional.
- Normalización y formas normales.
- Álgebra relacional.
- SQL.
- Evaluación de consultas.
- Indexación.
- Transacciones concurrentes.
- Recuperación ante caídas.

Modelos de datos

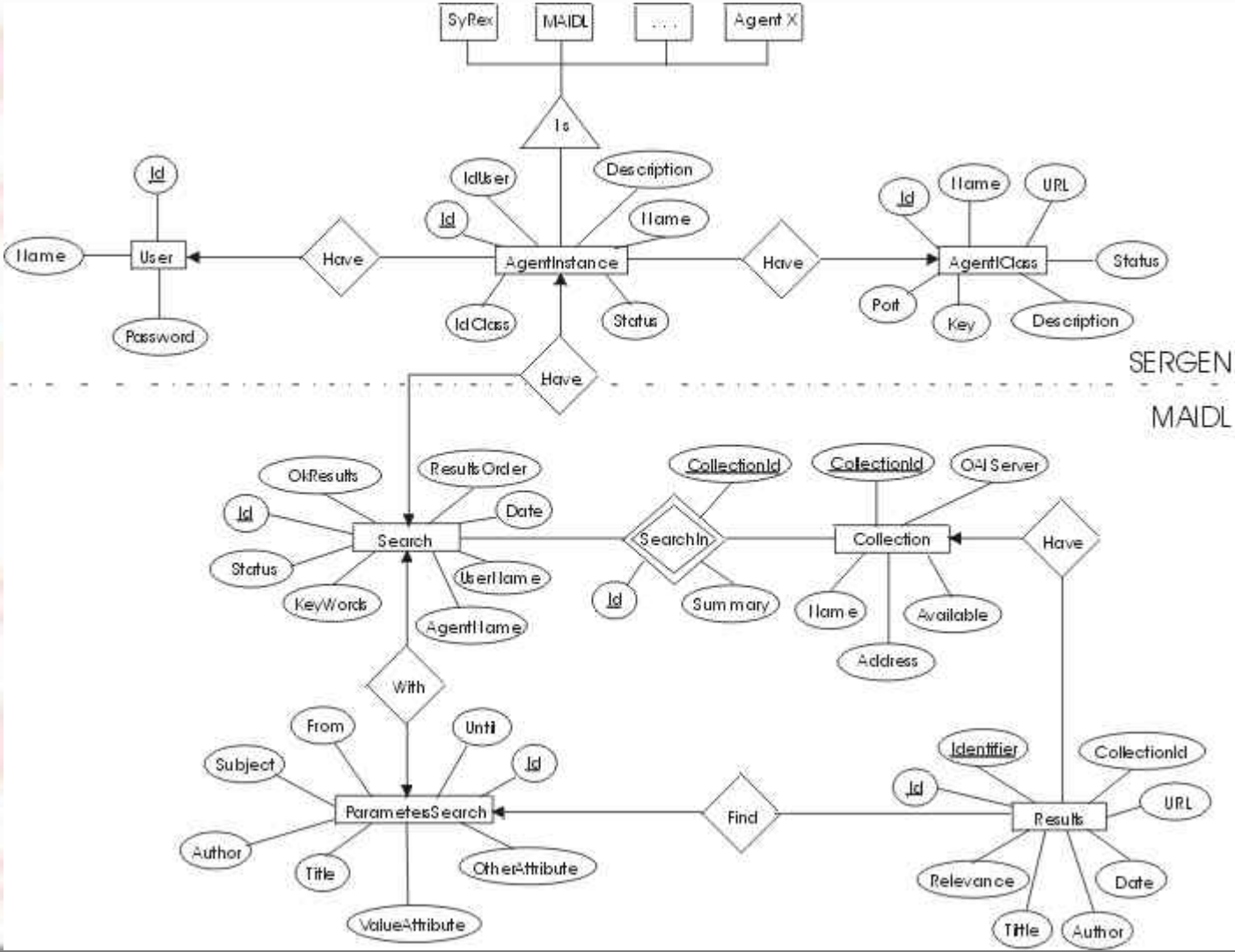
Entidad Relación

- El de las cajas.
- Tener claro cómo se escriben las cardinalidades y el concepto de entidad débil.

Relacional

- Relaciones matemáticas.
- El modelo de las tablas y las llaves externas (referencias).

Ej. entidad relación



Dependencias funcionales

- *X*-->*Y* significa que, implícitamente, *Y=f(X)*.
- Para iguales x deben haber iguales y.

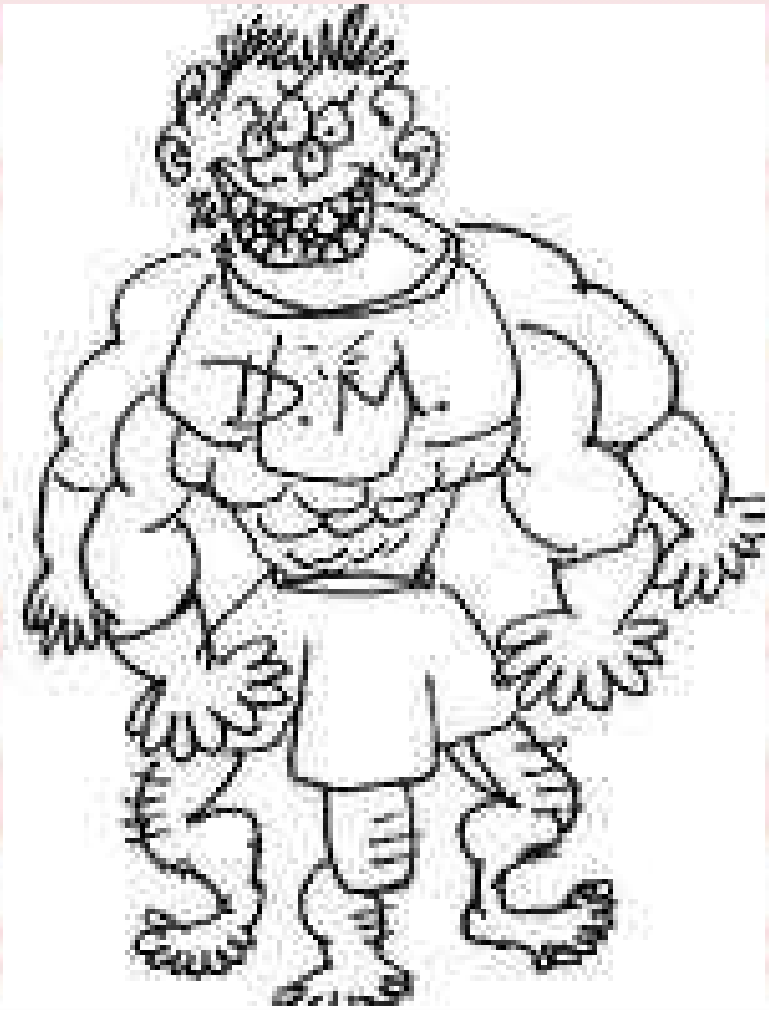
Dependencias funcionales

¿Qué dfs se cumplen?

- (a, b, c, d)
- (10, 2, +, A)
- (13, 3, - , A)
- (11, 2, +, B)
- (7 , 3, - , B)

- Axiomas de Armstrong
- Reflexividad: *ab->a*
- Amplificación: *a->b* y *a->c* => *a->bc*
- Transitividad: *a->b* y *b->c* => *a->c*

Dependencias multivaluadas



- A->>B significa que para cada valor de A, deberán apare-cer B(A) valores asociados.
- *a1b1c1* y *a1b2c2* llevan a *a1b1c2* y *a1b2c1*.

Dependencias multivaluadas

Dependencias multivaluadas

- A donde va la mamá, van **todos** los hijos.
- A->>B significa que para cada A siem-pre se asocian los mismos B.

Ejercicio: Paseos R(mami,hijo,lugar).

Tenemos:

(Rosa,Juan,Arica),
(Luisa,Diego,Osorno),
(Rosa,Pablo,Copiapó)

Completar.

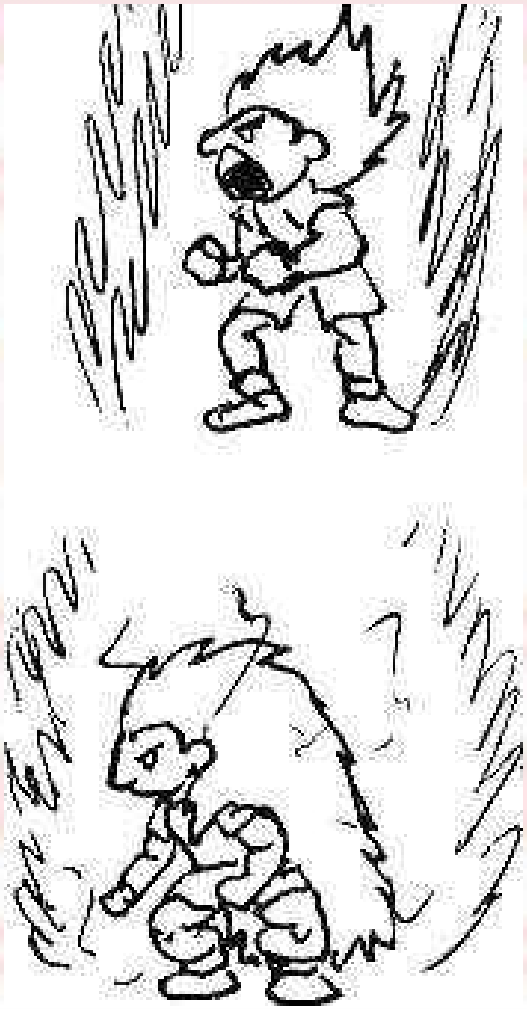
Formas normales

- 1FN: Atributos atómicos en las relaciones. Lo aho-ra inviolable.
- 2FN: Dependencia completa de la llave (no de una parte de ésta).

Formas normales

- 3FN: X->Y (no tri-vial), X superllave o Y atributo primo.
- FNBC: Más simple, X->Y no trivial, X es superllave.
- BC puede romper dependencias.

Formas normales



- 4FN: X->>Y (no trivial), X es superllave. FNBC generalizado.
- 5FN: Producto reunión: R=P*Q*T, entonces sólo guardar P, Q y T.

Formas normales

Como ejercicio *ayudamemoria*, pruebe las siguientes propiedades:

- Pruebe que FNBC => 3FN.
- Pruebe que 4FN => FNBC.
- Pruebe que 3FN => 2FN.
- Pruebe que 5FN => 4FN.

Normalización

- Las formas nor-males más impor-tantes son 3FN y FNBC.
- Encontrar árbol cobertor mí-nimo (sacar dependen-cias redundantes).
- Para 3FN hacer cada dependencia una relación. Si no está la llave agre-garla.
- Para FNBC partir la relación sistemáti-camente.

Consultas SQL

- La consulta SQL es el SELECT.
- MUY importante.
- Tratar de que cada consulta requiera sólo una instrucción (dejar holgura al optimizador).

Consultas SQL

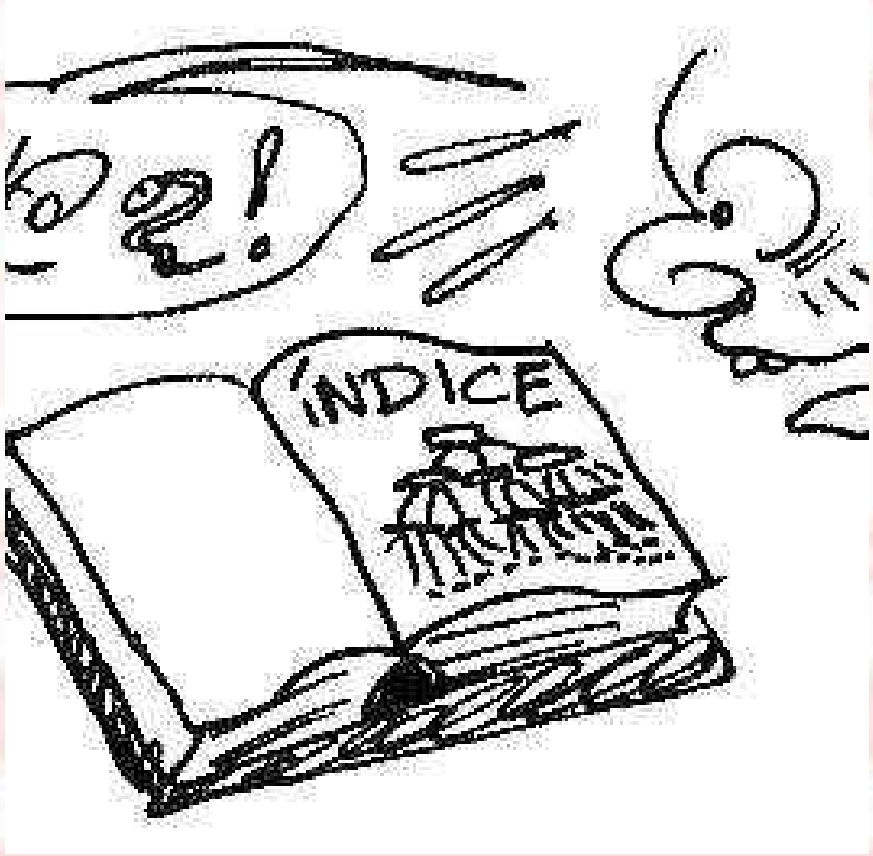
Tenemos:

```
Auto(Patente,Año,Kms,RUT)
Dueño(RUT,Nombre,Apellido)
Domicilio(RUT,Nro,Calle,Comuna)
```

Conteste:

- Personas con a lo más tres autos.
- Personas que hayan usado todos sus autos y que tengan residencia en todas las comunas.

Almacenamiento en disco



- Básicamente hay 3 tipos de archivo sin índices.
- Heap: desordenado.
- Sorted: ordenado.
- Clustered: agrupado, similar a ordenado.

Almacenamiento en disco

- Índices: B+Tree y Hashing.
- B+Tree es un TDA árbol con nodos internos que son guía y con nodos hoja que son punteros al archivo principal.
- Hashing se basa en una función que arroja un número. H(atrib)=N. Es la función de hashing.
- Saber más es más conviene.

Almacenamiento en disco

- B+Tree permite búsquedas en igualdad y en rango.
- Hashing sólo permite búsquedas en igualdad. Más eficiente en eso.
- Índice agrupado: es “casi” ordenado en relación a la tabla que indexa. Su uso es más eficiente.
- No agrupado: orden aleatorio (ineficiencia secuencial).

Evaluación de consultas

- Es útil el álgebra relacional. ¿Por qué?
- Optimización:
  - Álgebra.
  - Clase de equivalencia.
  - Función de costo.
- Método: al ver árboles, sólo considerar left-joins.
- Ver la posibilidad de usar índices.
- Estimar. De forma grosa, pero hacerlo.

Transacciones

- El objetivo es permitir el uso concurrente de la base de datos.
- Scheduling de transacciones.
- Protocolos de bloqueo.
- Niveles de aislamiento.
- Propiedades ACID.

Transacciones

- Serializabilidad (**view serializability**): Cuando una ejecución concurrente es equivalente a una serial, en cuanto a resultados finales.
- Problemas de consistencia: RW, WW, WR.
- Aislamientos:

SERIALIZABLE

READ_REPETIBLE

READ_COMMITED

READ_UNCOMMITED

Serializabilidad de conflicto

Serializabilidad de conflicto

- Dibujar gráfico y dependencias de un plan. Las dependencias son conflictos.
- Grafo cíclico: malas noticias.
- No serializable por conflicto => No serializable.

Recuperabilidad

Recuperabilidad

- El bandido va a envenenar a los caballos de la competencia.
- El tipo que sabe, apuesta todo su dinero al caballo que quedará sano.
- Pero el bandido no cumple su cometido, pues es atrapado en el acto.
- El otro tipo ya apostó, no hay manera de deshacer su acción.

Recuperabilidad y recuperación

- Como el apostador llegó a commit, no se recupera. El bandido cayó en el abort, así que sus “cambios” fueron deshechos.
- Recuperabilidad: si T1 lee algo escrito por T2, entonces T2 hace commit antes que T1.
- ¿Por qué? Por la recuperación.

Recuperación

- Existen LOGs en las bases de datos.
- Guardan las transac-ciones.
- Ante una caída, se deshace todo lo que no haya hecho com-mit, hacia atrás.
- Recuperabilidad: (T1 lee de T2) Si T2 no ha hecho commit, será deshecho. Entonces T1 también deberá ser deshecho. Por eso, si T2 no hace commit, T1 tampoco.

¡Eso ha sido!

- Mucha suerte en el examen.
- Comer y dormir bien.
- Nada de “carretes tóxicos” el día anterior.
- Estudien de las tantas guías disponibles.

Vean:

- http://www.dcc.uchile.cl/~mnmonsal/BD
- http://www.dcc.uchile.cl/~cgutierr/cursos/BD
- http://www.dcc.uchile.cl/~cc42a