UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

BASES DE DATOS
NoSQL

GUÍA DE EJERCICIOS

1 Conceptuales

1.1. Describa brevemente limitaciones de las base de datos relacionales y con qué características las base de datos NoSql mitigan este problema.

1.2. Describa los cuatro tipos de base de datos NoSql.

1.3. ¿Qué es un espacio de nombres o bucket en una base de datos *Key-Value*?

1.4. ¿De ejemplos de uso de TTL (time to live) en una base *Key-Value*?

1.5. Compare *Key-Value* con *Document Database*, de ventajas y desventajas de una u otra.

1.6. Discuta ventajas y desventajas de que una *Document Database* sea *schemaless*.

1.7. ¿En qué casos puede ser conveniente *desnormalizar*?

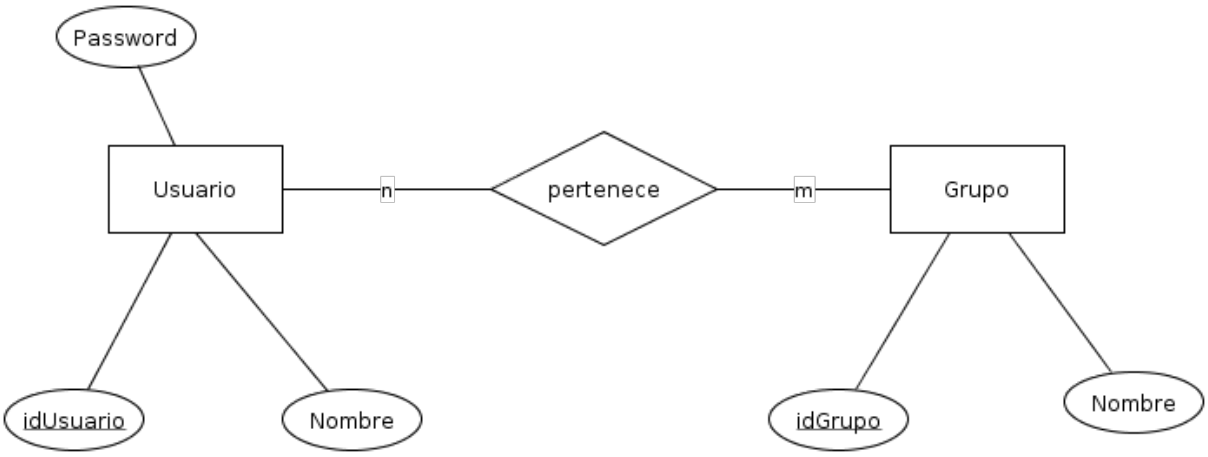
1.8. Compare las *Column Family Databases* con otros tipos de bases de datos NoSQL.

1.9. ¿A que se denomina consistencia eventual?

1.10. Explique el teorema CAP.

2 Modelización y Diseño

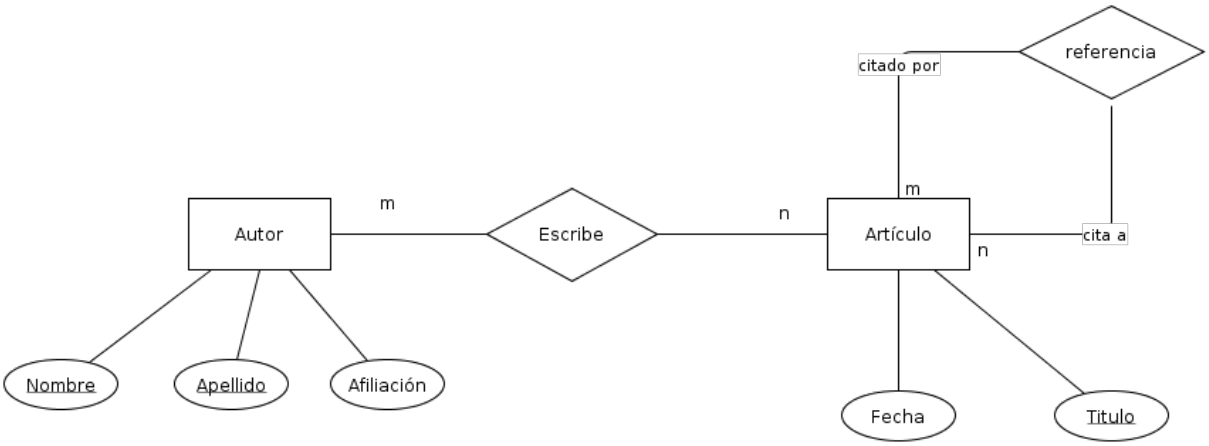
2.1. Dado el siguiente DER realizar por lo menos 3 diferentes esquemas para *Document Databases*.



2.2. Un congreso de ciencias de la computación almacena datos de las publicaciones realizadas. Se guardan autores, artículos y además las relaciones entre artículos (es decir artículos que citan otros artículos). El siguiente DER modela la situación. Se pide que realice un modelo para una base de datos *Column-Family* tal que responda las siguientes consultas.

(a) Buscar todos los artículos que citan a un artículo dado. Se debe devolver fecha y titulo de los artículos citados.

(b) Buscar todos los autores que escribieron en una fecha dada.



2.3. Una empresa de video juegos realiza un juego en línea y necesita guardar el estado de las partidas de los jugadores. Dicho estado debe almacenar: posición, nivel de vida, objetos encontrados y enemigos abatidos. El jugador deberá poder jugar desde cualquier estado guardado eligiendo la fecha y hora en el que lo guardo. Se pide realizar un modelo para una base *Key-Value* que soporte lo descrito.

2.4. El servicio meteorlógico nacional debe guardar para análisis posteriores la información de las mediciones de temperatura y presión atmosférica de diversas estaciones meteorológicas. Cada estación meteorológica se identifica con un número y dos letras que representan la ubicación geográfica. La información obtenida se guarda cada 10 minutos. Es necesario poder graficar la evolución de la temperatura y/o de la presión en una estación dada en un día dado. También puede ser necesario comparar esa evolución en dos fechas diferentes. Se pide que:

(a) Realice un modelo para una base de datos de *Column-Family* tal que sea adecuado para lo enunciado.

(b) Realice un modelo para una base de datos de *Key-Value* tal que sea adecuado para el enunciado

(c) Realice un modelo para una base de datos *Document-Based* tal que sea adecuado para el enunciado

Al decir adecuado para el enunciado se pide que se optimice para las consultas planteadas y que no se generen estructuras innecesarias.

2.5. Dado el siguiente ejemplo en JSON sobre editoriales libros

```
{
    "titulo": "MongoDB: The Definitive Guide",
    "autor": [ "Kristina Chodorow", "Mike Dirolf" ],
    "fecha_pub": ISODate("2010-09-24"),
    "paginas": 216,
    "idioma": "English",
    "editorial": {
            "nombre": "O'Reilly Media",
            "fundada": 1980,
            "ubicacion": "CA"
    }
}
```

```
{
    "titulo": "50 Tips and Tricks for MongoDB Developer",
    "autor": "Kristina Chodorow",
    "fecha_pub": ISODate("2011-05-06"),
    "paginas": 68,
    "idioma": "English",
    "editorial": {
            "nombre": "O'Reilly Media",
            "fundada": 1980,
            "ubicacion": "CA"
    }
}
```

(a) Modificar los documentos para que no se repitan los datos de las editoriales

(b) Discutir en que contexto es mejor la representación dada sobre la resuelta en el ítem anterior.

2.6. Dado el siguiente documento JSON, definir un esquema para soportar esos datos en un modelo para *Column Databases*

```
{
   "person": {
            "name": "Lindsay Bassett",
            "heightInInches": 66,
            "head": {
                    "hair": {
                            "color": "light blond",
                            "length": "short",
                            "style": "A-line"
                    },
                    "eyes": "green"
            }
    }
}
```

2.7. Se necesita almacenar información sobre la cantidad de veces que diversos usuarios visitan determinados sitios web. Desarrolle un modelo para una base de datos *Column Family* que soporte esta información.

2.8. Una instituto educativo está desarrollando una aplicación para realizar encuestas on-line.

(a) Definir un esquema JSON para almacenar una encuesta con múltiples respuestas.

(b) Expandir el esquema del ejercicio anterior si se quiere guardar el resultado de las elecciones que cada alumno hizo al realizar la encuesta. Analice las opciones disponibles y discuta sus ventajas y desventajas.

2.9. Se desea guardar la información de llamadas telefónicas para poder procesarlas posteriormente. Cada llamada tiene: el abonado origen, el abonado destino, el momento de inicio y de fin de la llamada y por último el estado de la llamada: 'A' activa, 'I' inactiva que dice si la comunicación se debe facturar o no. Cada abonado tiene datos particulares: el número, el nombre y el domicilio completo. Las consultas a realizar son:

- Listado de llamadas de un abonado en particular,
- Conocer si entre dos abonados hubo una llamada en algún rango de tiempo.

Diseñe los esquemas generales para:

(a) Bases de datos *Column-Family.*

(b) Bases de datos *Document Databases.*

(c) Bases de datos *Key-Value.*

2.10. Una aplicación desea guardar información de sus usuarios, los mismos son identificados por su dirección de correo. Podemos asumir dos tamaños de la información guardar pequeño y grande. La información de tamaño pequeño es:

- Nombre
- Genero
- Dirección
- Teléfono

La información de tamaño grande incluye:

- Imagen
- Clave pública 1
- Clave pública 2
- Mensaje de voz de saludos

Diseñar para una base de datos *Key-Value* la forma de organizar la clave para los siguientes casos:

(a) La aplicación requiere que se lean todas las propiedades cada vez que se accede al usuario

(b) Las forma de acceso es:
- Las propiedades de tamaño pequeño son accedidas generalmente en conjunto.
- Las propiedades de tamaño grande son leídas cuando un usuario busca su información
- Al actualizar los datos las claves públicas son siempre actualizadas al mismo tiempo
- Tanto la imagen como el mensaje de voz puede ser actualizados en forma independiente de las otras cosas.

2.11. Se desea organizar el soporte de almacenamiento para un blog, el mismo debe ser multiusuario, es decir varios usuarios pueden publicar en él. Además cada artículo en el blog puede tener archivos adjuntos para que los lectores descarguen. Hay que soportar dos tipos de comentarios: autenticados y no autenticados. Los primeros son de usuarios registrados y los segundos de visitas ocasionales. Las consultas mas frecuentes son ver los artículos de un usuario, ver artículos por categoría y ver comentarios de un artículo (cuando se muestra el artículo se presenta la opción de ver los comentarios).Se pide

(a) Desarrollar un modelo conceptual y un modelo de documentos para soportar la información.

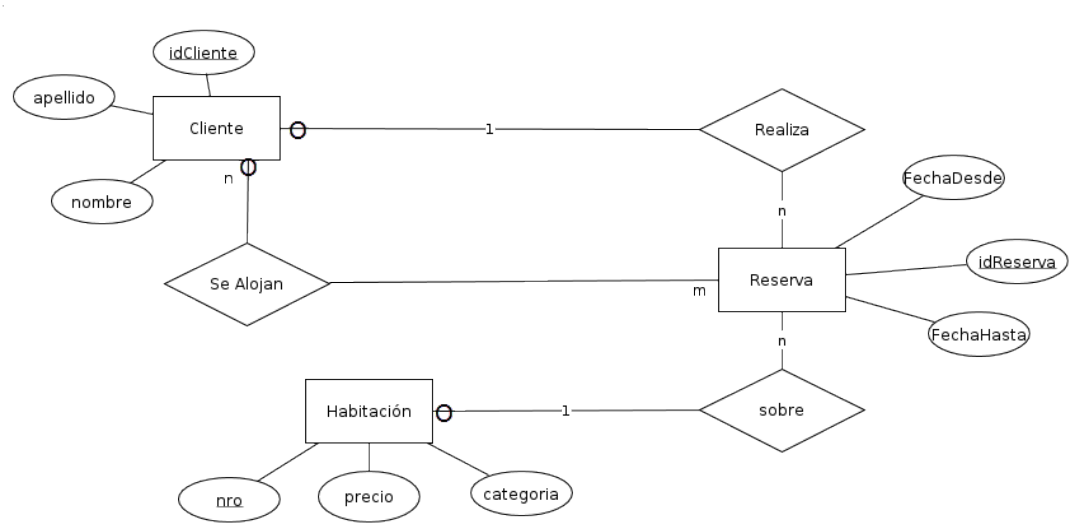
2.12. Una empresa provee acceso mediante tarjetas con lectores. Cada tarjeta tiene un identificador que es leído por los lectores para saber si tiene acceso en cada lugar. Cada puerta esta identificada con un número y cuenta con un lector. Si el identificador de tarjeta tiene permiso la puerta se abre y sino no. Para saber si un identificador tiene permiso los mismos se agrupan en roles. Los roles son los que tienen el permiso de acceso. Un identificador de tarjeta puede pertenecer a varios roles. Una vez que una persona entra en la recepción se le da una tarjeta y se activa la misma, cuando se retira de la empresa la devuelve en recepción y se desactiva.

Se pide:

(a) Realizar un diseño para mantener el recorrido las tarjetas activas en una base clave-valor. Es decir debe guardarse la tarjeta con las puertas por las que paso y por las que intento pasar. También debe saberse en que momento abrió o intento abrir una puerta. Además la base debe poder usarse para determinar si una tarjeta puede o no acceder a una puerta.

(b) Realizar un diseño basado en documentos para modelar el problema. Debe realizar primero un modelo conceptual. Explicar fundamentadamente las decisiones de diseño tomadas. Cuando la persona que tiene la tarjeta se retira toda la información asociada a la tarjeta que está en la base clave valor (según ítem i)debe pasarse a la base de documentos y es eliminada de la base clave valor. Tome en cuenta que esa información es enormememente grande a lo largo del tiempo. No debería perder información al modelar los documentos.

2.13. Dado el siguiente DER , donde se modela la reserva de habitaciones en un hotel



(a) Se pide que se realice un modelo completo del mismo para una base de datos *Document Based*. El modelo ***debe*** incluir diagrama de interrelación de documentos y JSONSchema completo del diseño. Tome en cuenta lo siguiente: Un cliente realiza una reserva para una habitación en un rango de fechas. La reserva es sobre una habitación y para una cantidad de personas (son los que se alojan y también son clientes). El precio que figura en habitación es el precio por noche. Cuando se consulta una reserva se necesita conocer el detalle de la habitación reservada y los datos del cliente que hizo la reserva y eventualmente la lista de pasajeros. El precio de la habitación puede cambiar con el tiempo por lo que se necesita saber a que precio se reservó la habitación es su momento. Debe fundamentar brevemente las elecciones de diseño que realice y deben ser coherentes con la funcionalidad descrita.

(b) Se pide que realice el diseño (utilizando el diagrama Chebotko) para una base de datos *Column-Family* tal que responda la siguiente consulta:

Los datos de las habitaciones que para una fecha dada comienzan una reserva en esa fecha ordenadas por precio descendente.

(c) Diseñe una base clave-valor para poder consultar si hay alguna habitación de una categoría dada reservada en una fecha. Es decir se debe saber las habitaciones de una categoría y si esa habitación esta reservada en una fecha dada.

2.14. Dado el DER de la figura, donde se modela un blog con artículos y acciones de usuarios sobre ellos (calificar o comentar)

(a) Se pide que se realice un modelo para una base de datos *Document Based*. Tome en cuenta que:

i. Se quiere *particionar* las actividades por Fecha para cada usuario, es decir acceder en una única consulta para un usuario y fecha dadas a todas las actividades correspon-dientes.

ii. Para el caso de los artículos se quiere mostrar todas las actividades sobre él cada vez que se accede al mismo independientemente de la fecha.

Realizar el *JSONSchema* correspondiente.

(b) Se pide que realice el diseño de las tablas para una base de datos de Column-Family tal que responda las siguientes consultas:

i. Todos los artículos (título y contenido) de un Blog dado para una Fecha dada.

ii. Todos los comentarios de un usuario en un rango de fechas.

3 Map-Reduce

3.1. Suponga un archivo distribuido con datos de ciudades en cada linea los datos de cada una. Diseñar algoritmos MapReduce que sean equivalentes a las siguientes sentencia SQL:

(a) SELECT país, SUM(población) FROM Ciudad GROUP BY país

(b) SELECT país, provincia, AVG(población) FROM Ciudad GROUP BY pais, provincia

3.2. Definir las funciones Map y Reduce para las operaciones del algebra relacional: selección, proyección, resta y junta

3.3. Diseñar algoritmos MapReduce que tomen como entrada un archivo de enteros muy grande y produzcan como resultado:

(a) El entero más grande

(b) El promedio de todos los números

(c) El mismo conjunto de números pero cada uno apareciendo solo una vez

(d) La cantidad de números distintos

3.4. Definir las funciones Map y Reduce que permitan construir un índice invertido sobre una lista de *tweets*. El resultado debería permitir ubicar los tweets donde se utilizaron determinadas palabras.

3.5. Para el ejercicio **2.11.** diseñar algoritmos MapReduce que permitan obtener:

(a) La cantidad total de comentarios por autor

(b) El usuario que mayor cantidad de comentarios realizó

(c) La cantidad de comentarios en una categoría

(d) Los usuarios que publicaron articulos en al menos 3 categorías.

3.6. Una compañía de música online permite a los usuarios escuchar diferentes pistas, los datos se guardan en archivos con el formato UserId,PistaID, Compartido, Radio, Omitida Diseñar algoritmos MapReduce que permitan obtener

(a) Número de usuarios únicos

(b) Número de veces que una pista fue compartida con otros

(c) Número de veces que una pista fue escuchada en la radio

(d) Número de veces que una pista fue escuchada en total

Nota: Compartido, Radio y Omitida son valores que pueden ser 0 o 1 que indican falso y verdadero respectivamente.

3.7. Tomando en cuenta el ejercicio **2.12.** Suponga que la base de documentos contiene una colección de documentos donde cada uno se compone de: identificación de tarjeta, fecha, una lista de puertas accedidas con su timestamp y una lista de puertas no accedidas (que intento pero no abrió) con su timestamp. Realice una consulta map reduce que devuelva la cantidad de accesos no permitidos y la cantidad de accesos permitidos por puerta durante el año pasado. O sea la puerta 1 tendrá por ejemplo 3000 accesos permitidos y 5000 no permitidos. El reducer debe ser **combinable**. Optimice el tráfico en red.

3.8. Realice el diseño de una consulta Map-Reduce (sobre el modelo de documentos del ejercicio **2.14.a** que devuelva la cantidad de artículos por cantidad de comentarios. El resultado sería una lista de pares $< NroComentarios, CantArticulos >$. Es decir, por ejemplo: $< 5, 10 >$ sería uno de los resultados donde el 5 es la cantidad de comentarios y el 10 es la cantidad de artículos que tienen 5 comentarios.