

2024 - 2025

**Máster en Big Data y Data Science**

**02MBID –Sistemas de Almacenamiento y Gestión Big Data**

**Nombre: Gonzalo Antonio Delgado Rubio**

**Fecha: 13/05/2024**

**Curso 2022 – Ed. Abril**

**ACTIVIDAD 1**

**1. Creación de Esquemas de una Base de Datos Orienta a Columnas**

Bajo el modelo conceptual se pide satisfacer las siguientes consultas

Diagrama

Descripción generada automáticamente

**Consultas a satisfacer**

1. Obtener toda la información de libros publicados en un año en concreto.

|  |  |
| --- | --- |
| **Tabla1** |  |
| **Columna** | **Primary Key** |
| Libro\_año | PK – Primary Key |
| Libro\_ISBN | CK – Clustering Key |
| Libro\_Titulo |  |
| Libro\_Temas |  |

A partir de la consulta a satisfacer se toma en consideración que año será la columna de partición y para mantener la unicidad de la tabla se debe considerar como clustering key la columna ISNB el cual corresponde al identificador único de libro según el modelo conceptual.

De este modo para satisfacer la consulta, se toma únicamente información de la entidad libros.

1. Obtener toda la información de los ejemplares de un libro según el título de este.

|  |  |
| --- | --- |
| **Tabla2** |  |
| **Columna** | **Primary Key** |
| Libro\_Titulo | PK – Partition Key |
| Libro\_ISBN | CK – Clustering Key |
| Ejemplar\_Nro | CK – Clustering Key |
| Ejemplar\_Status |  |

Dado que el título de libro será clave para satisfacer una consulta de tipo

Select \* from tabla2 where libro\_titulo=?

Debemos considerar a libro\_titulo como la columna partición a manera de balancear los datos de esta tabla, también debemos adicional las columnas libro\_isbn y ejemplar\_nro como clustering key dado que por la relación 1-n establecido deberán formar parte de nuestra clave primaria.

Se refuerza el hecho que al ser el titulo de libro un nombre que pueda repetirse para establecer verdaderamente la unicidad deberíamos añadir el campo Libro\_isbn.

Ejemplo

|  |  |  |
| --- | --- | --- |
| Harry Potter | ISBN123 | Ejemplar1 |
| Harry Potter | ISBN123 | Ejemplar2 |

Quizás si Ejemplar\_nro fuera una combinación de ISBN+Nro podría considerar únicamente a esta como clustering key.

1. Obtener toda la información de los libros escritos por un autor buscando por el nombre del autor.

|  |  |
| --- | --- |
| **Tabla3** |  |
| **Columna** | **Primary Key** |
| Autor\_nombre | PK – Partition Key |
| Libro\_ISBN | CK – Clustering Key |
| Autor\_Cod | CK – Clustering Key |
| Libro\_Titulo |  |
| Libro\_Año |  |
| Libro\_Temas |  |

Dada que la consulta a satisfacer tendrá una forma similar a la siguiente

Select \* from tabla3 where autor\_nombre=?

Debemos considerar a esta columna como PK y a las columnas Libro ISBN y Autor cod como clustering key para mantener la unicicidad en los registros considerando la relación de las entidades n-m entre libro y autor. Esto siguiente la metodología estudiada en el curso.

1. Obtener los usuarios que han tomado prestado el ejemplar de un libro según el título de un libro.

|  |  |
| --- | --- |
| **Tabla4** |  |
| **Columna** | **Primary Key** |
| Libro\_Titulo | PK – Partition Key |
| Usuario\_DNI | CK – Clustering Key |
| Libro\_ISBN | CK – Clustering Key |
| Ejemplar\_Nro | CK – Clustering Key |
| Usuario\_nombre |  |
| Usuario\_calle |  |
| Usuario\_ciudad |  |

La consulta más probable para resolver será la siguiente:

Select \* from tabla4 where libro\_titulo = ?

Esto nos indica que la partition key para este caso será el titulo de libro. Este campo nos ayudará en la distribución de nuestros datos sobre la db. En este caso en particular, a manera de mantener la unicidad deberíamos añadir las columnas Usuario\_DNI, Libro\_ISBN y Ejemplar\_Nro debido que la consulta nos pide información de los usuarios que han solicitado prestado algún ejemplar de un libro. Aquí se indica entonces que debemos ver la relación de las entidades Libro – Ejemplar – Usuario; que de acuerdo con el modelo conceptual se establece de la siguiente manera:

* Libro – Ejemplar: 1- n
* Ejemplar – Usuario: n – m

Tal y como se siguió en la consulta 2, en el caso de las entidades Libro-Ejemplar se están considerando ambas llaves de tabla como parte del cluster para evitar duplicidad. Para el caso Ejemplar – Usuario, se sigue la metodología establecida.

1. Considerando que el 60% de los usuarios están en la ciudad de Nueva York, haga la tabla óptima según rendimiento en la que se consulte por la ciudad del usuario su información

|  |  |
| --- | --- |
| **Tabla5** |  |
| **Columna** | **Primary Key** |
| Usuario\_ciudad | PK – Partition Key |
| Usuario\_DNI | CK |
| Usuario\_ Nombre |  |
| Usuario\_calle |  |

Para este caso nos solicitan satisfacer una consulta como la siguiente:

Select \* from tabla5 where ciudad=?

El problema describe que debemos optimizar la búsqueda según la ciudad, para lo cual nos indican que la mayor parte de los datos, 60%, se encuentra en la ciudad de Nueva York. Para este caso al no tener una mejor partición a ciudad esta se mantiene, dado que calle usuario sería un dato que no ayudaría mucho en la distribución de los datos. Quizás si se tuviera un campo del tipo zonificación o código de área podría ayudar a una mejor distribución de los datos.

Siguiente esta línea, debemos añadir la clave de la entidad como clustering key para mantener la unicidad de los registros.

1. Obtener cuantas veces se ha prestado un ejemplar según su número (nro).

|  |  |
| --- | --- |
| **Tabla6** |  |
| **Columna** | **Primary Key** |
| Ejemplar\_nro | PK – Partition Key |
| Usuario\_DNI | CK – Clustering Key |
| Cantidad | + |

Para esta consulta nos solicitan cuantas veces se ha prestado un ejemplar según el nro de este. Aquí nos damos cuenta de que nos están pidiendo una agregación de los datos a manera de cantidad de veces que un libro ha sido prestado.

Para este caso consideraremos el ejemplar\_nro como la clave partición para los datos y usuario dni como su clustering key a manera de llevar una cuenta de los prestamos por usuario sin que sea demasiado complicado poder realizar una sumarización total de los prestamos por ejemplar.

1. Obtener la información de los autores que hayan ganado un premio específico (p. ej. Premio Planeta)

|  |  |
| --- | --- |
| **Tabla7** |  |
| **Columna** | **Primary Key** |
| Premios\_Premio | PK – Partition Key |
| Autor\_Cod | CK – Clustering Key |
| Autor\_nombre |  |

Para este caso en particular y según el modelo conceptual el tipo de dato premios pertenece a un conjunto de datos por lo que si quisiéramos satisfacer una consulta de este tipo debemos generar una tabla por separado considerando a premios\_premio como la partición de la tabla y al autor\_cod como su clustering key para contener la unicicidad de los datos.

Finalmente se resolverá una consulta como la siguiente:

Select \* from tabla7 where premio=?

1. Buscar según la fecha de préstamo los ejemplares prestados y el usuario que lo tomó prestado.

|  |  |
| --- | --- |
| Tabla8 |  |
| Fecha | PK – Partition Key |
| Ejemplar\_nro | CK – Clustering Key |
| Usuario\_DNI | CK – Clustering Key |
| Ejemplar\_status |  |
| Usuario\_nombre |  |
| Usuario\_ciudad |  |
| Usuario\_calle |  |

La consulta que debemos resolver es la siguiente

Select \* from tabla8 where fecha=?

De aqui nos damos cuenta que fecha debería ser la clave partición de la tabla. Para descubrir los campos de tipo cluster que nos ayuden a mantener la unicidad de la tabla debemos ver la relación de las entidades. Para lo cual se tiene

* Usuario – Ejemplar: n – m

Por lo que la teoría nos indica que debemos traer las claves de estas entidades y añadir como clustering key. Para lo cual tenemos a ejemplar\_nro y usuario\_dni como tal.