

Documentación de los Índices

1. Índice compuesto sobre la tabla *inscripcion*

```
CREATE INDEX idx_inscripcion_documento_convocatoria ON inscripcion  
(PER_DocumentoIdentidad, CON_IdConvocatoria);
```

Este índice está compuesto por la columnas *PER_DocumentoIdentidad* y *CON_IdConvocatoria* que son llaves foráneas en la tabla *inscripcion*. Estas llaves son fundamentales para las relaciones para establecer la inscripción de las personas de la comunidad universitaria en distintas convocatorias. La mayoría de las reglas del negocio, fundamentados en los procedimientos almacenados, vistas y funciones están basados en la relación entre las personas junto a sus inscripciones en sus respectivas convocatorias. Por tanto, es importante la implementación de un índice compuesto que tenga en cuenta estas dos columnas para hacer las búsquedas y consultas más rápidas, haciendo uso del algoritmo de búsqueda B-Tree.

2. Índice de texto sobre la columna *Per_CorreoElectronico* de la tabla *persona*

```
CREATE FULLTEXT INDEX idx_persona_email ON persona (PER_CorreoElectronico);
```

La columna *Per_CorreoElectronico* es de tipo VARCHAR() por tanto un índice de tipo FULLTEXT es una mejor opción de implementación, tal que es adecuado para columnas con tipos de dato texto. El correo electrónico de cualquier persona de la comunidad universitaria es uno de los datos más relevantes para llevar a cabo procesos del modelo.

3. Índice compuesto sobre las fechas de inicio y fin de las convocatorias.

```
CREATE INDEX idx_convocatoria_fecha_inicio ON convocatoria  
(CON_FechaInicio, CON_FechaFin);
```

Las columnas *CON_FechaInicio* y *CON_FechaFin* son muy importantes a la hora de realizar análisis y búsquedas sobre las convocatorias, tal que cada convocatoria debe cumplir con un período donde se hace posible la realización de cursos, proyectos ppg y beneficios. Así que la identificación de los períodos de cada convocatoria es fundamental en las reglas de negocio del sistema de bienestar. Por tanto, se implementa un índice compuesto entre estas dos columnas (*CON_FechaInicio* y *CON_FechaFin*) en la tabla *convocatoria* con el uso del

algoritmo de búsqueda B-Tree, ya que este es más eficiente al momento de realizar comparaciones de desigualdades ($>$ ó $<$), tal como ocurre en el las sentencias condicionadas con tipos de dato DATE().

4. Índice compuesto sobre las columnas que referencian el período de un beneficio.

```
CREATE INDEX idx_beneficio_fecha ON beneficio (BEN_Inicio, BEN_Finalizacion);
```

Este índice compuesto hace más eficientes y rápidas las búsquedas y sentencias sobre las tablas involucradas en el índice, tal y como puede ocurrir con consultas que establezcan condiciones a partir del período de aplicación de un beneficio, el cual es uno de los programas fundamentales ofrecidos por bienestar. Ambas columnas son de tipo de dato DATE(), entonces el algoritmo B-Tree es más adecuado para la creación del índice.

5. Índice sobre los ID de los requisitos para las convocatorias.

```
CREATE INDEX idx_requisito ON convocatoria_requisito (REQ_idRequisito);
```

La creación de este índice permite facilitar la búsqueda de requisitos por convocatoria. Cabe resaltar que cada requisito tiene un ID el cual permite la relación como llave primaria en la relación *convocatoria_requisito*, y además según las convocatoria hay una considerable cantidad de requisitos por una sola convocatoria. Por tanto, la implementación de un índice B-Tree hace más eficiente la respuesta a sentencias que involucran en primer lugar la columna *REQ_IdRequisito* en la tabla *convocatoria_requisito*.

6. Índice implementado sobre la fecha de la inscripción

```
CREATE INDEX idx_inscripcion_fecha ON inscripcion (INS_Fecha);
```

La columna *INS_Fecha* en la tabla *inscripción* desempeña un papel fundamental para registrar el día exacto donde se establece la relación entre la convocatoria con una persona de la comunidad universitaria a través de la inscripción. Así que el uso de un índice sobre esta columna de tipo de dato DATE() hace más eficiente las consultas en la tabla *inscripcion* con respecto a la fecha

7. Índice sobre el ID de los horarios en la tabla evento.

```
CREATE INDEX idx_evento_horario ON evento (HOR_IdHorario);
```

La implementación de un índice sobre una columna que es llave foránea en la relación *evento*, mediante el algoritmo B-tree o árbol balanceado hace más eficiente la relación entre las tablas que se da por la llave. Esta relación evento-horario es fundamental a la hora de establecer nuevos eventos o realizar análisis sobre eventos ya realizados.

8. Índice único sobre el documento de identidad de los empleados

```
CREATE UNIQUE INDEX idx_empleado_documento ON empleado  
(PER_DocumentoIdentidad);
```

La creación de un índice único sobre los documentos de identidad de los empleados hace más sólida y eficaz la relación por llave foránea a través de la columna *PER_DocumentoIdentidad* entre las tablas *persona* y *empleado*. Este índice es único tal que no se repiten valores sobre la columna afectada por el índice.

Nota: Al crear llaves primarias en el momento de definir las tablas y sus atributos, estamos definiendo índices primarios sobre las columnas establecidas como llaves primarias. Así mismo el uso de índices con el uso del algoritmo HASH se implementa en bases de datos con motores de almacenamiento MEMORY o InnoDB. Por lo tanto, se prefiere usar el tipo B-Tree en la definición de índices.