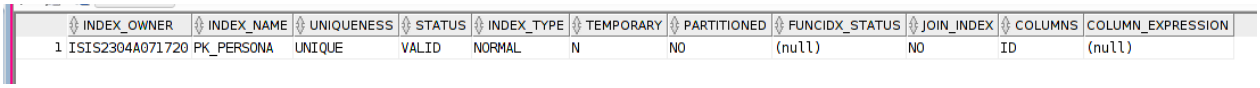
Uso de índices para mejorar la eficiencia en base a los requerimientos funcionales.

Para lograr tan funcionalidad, a pesar de que se observo y analizo la forma en que la base de datos estaba diseñada y la estructura que tienen las tablas, a nosotros nos pareció mucho más relevante las diferente consultas que se hacían sobre estas, pues sin importar cómo esté la DB, los las consultas las que definen el funcionamiento real de la aplicación y las peticiones que tendrá la DB.

En base las consultas se considero cual seria el mejor tipo de índice que se debería utilizar para cada tabla y cada columna, donde consideramos que fuera necesario.

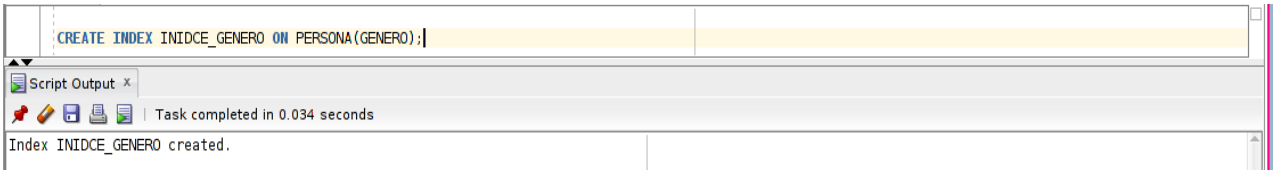
- Personas:

Para la tabla de personas se considero que el mejor índice que se podría utilizar sería uno el cual se basará en alta selectividad, como lo sería un Hash Index. Este se aplica a la columna de ID, la cual es diferente para cada uno de las personas.



	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS	COLUMN_EXPRESSION
1	ISIS2304A071720	PK_PERSONA	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	ID	(null)

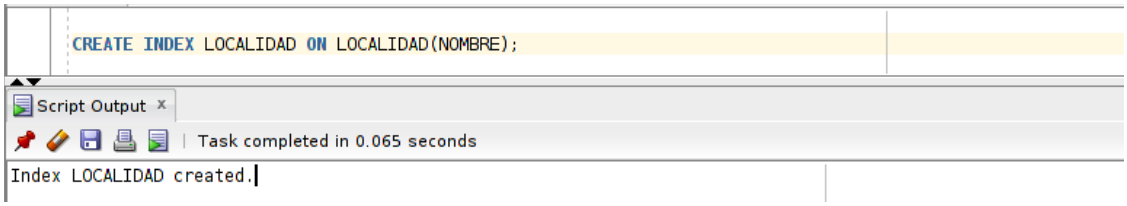
Por otro lado, para lo que es el GÉNERO es bueno usar un BitMap, pues este se basa en atributos que tengan baja cardinalidad y como el género es el mejor caso, es decir un Booleano, es la mejor opción.



- Localidad:

La localidad tiene la ventaja de que está separada por tipo (nombre), esto implica que no hay muchos tipos, es decir, la variabilidad en la información no es muy alta.

Por tal razón es otra vez la mejor opción hacer un índice BitMap sobre la columna de NOMBRE que hace referencia al tipo de localidad.



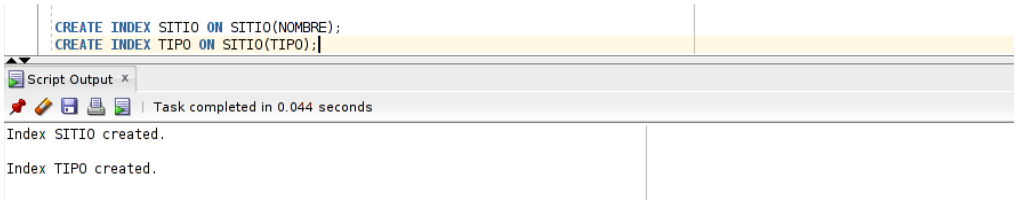
- Sitio:

Para sitio nos basamos en el RFC de "Consultar un sitio", esto implica que la consulta se basa en obtener información sobre un lugar en específico.

Por tal motivo para este se utilizó un índice basado en una alta selectividad. Pues se busca es un sitio en específico.

Para esto se usa el index que ofrece Oracle, el cual permite una búsqueda más eficiente.

El índice se hace no sobre el ID, sino por el NOMBRE, pues se presume que no hay sitios con un mismo nombre.



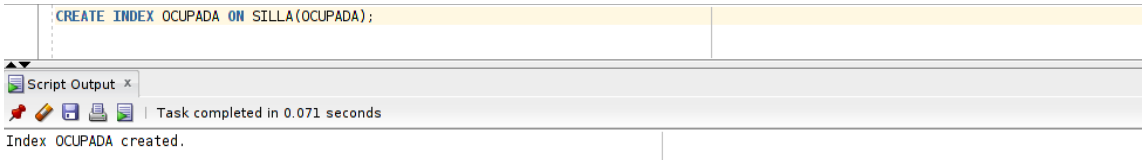
- Sillas

Las sillas más allá de pertenecer a una localidad, sitio, entre otras; lo más importante es saber si están disponibles. Esto se reduce a tener un valor booleano asociado a las sillas.

Por tanto la mejor opción vuelve a ser un índice BitMap ya que no solo es eficiente para atributos con baja cardinalidad, sino también su mejor caso es un booleano.

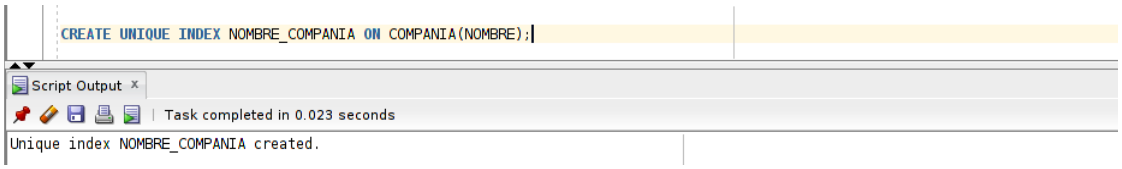
La columna a la cual se le pondría tal índice sería la de OCUPADO.

Esto ayuda al momento de hacer una reservación.



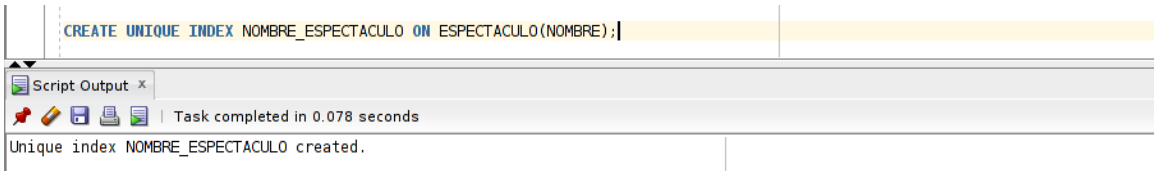
- Compañías:

En la aplicación se puede buscar por una sola empresa para conocer diferentes indicadores económicos, esto implica que la búsqueda debe ser específica. Por tal, se debe usar un índice que permita una mejor eficiencia para consultas específicas, tal como una Hash o un Unique (Oracle).



- Espectáculo:

Al igual que con las compañías, se debe poder hacer una búsqueda por un solo espectaculo para que, también, de un reporte económico del mismo. Usando la misma estratégia se aplica un indice sobre la columna de nombre; pues se esperaría que es mucho mas posible que los nombres sean distintos. Para esto un Hash o Unique (Oracle).



- Abono:

Para el abono se debe tener un índice que permita una busqueda facil en caso de que quiera hacer alguna devolución. Para esto se usa el proporcionado por la base de datos. Esto permite busquedas más eficientes.

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS	COLUMN_EXPRESSION
1	ISIS2304A071720	ABONO_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	ID	(null)