

TEMA 5 – OPTIMIZACIÓN DE CONSULTAS DISTRIBUIDAS
SERIE DE EJERCICIOS – PARTE 1

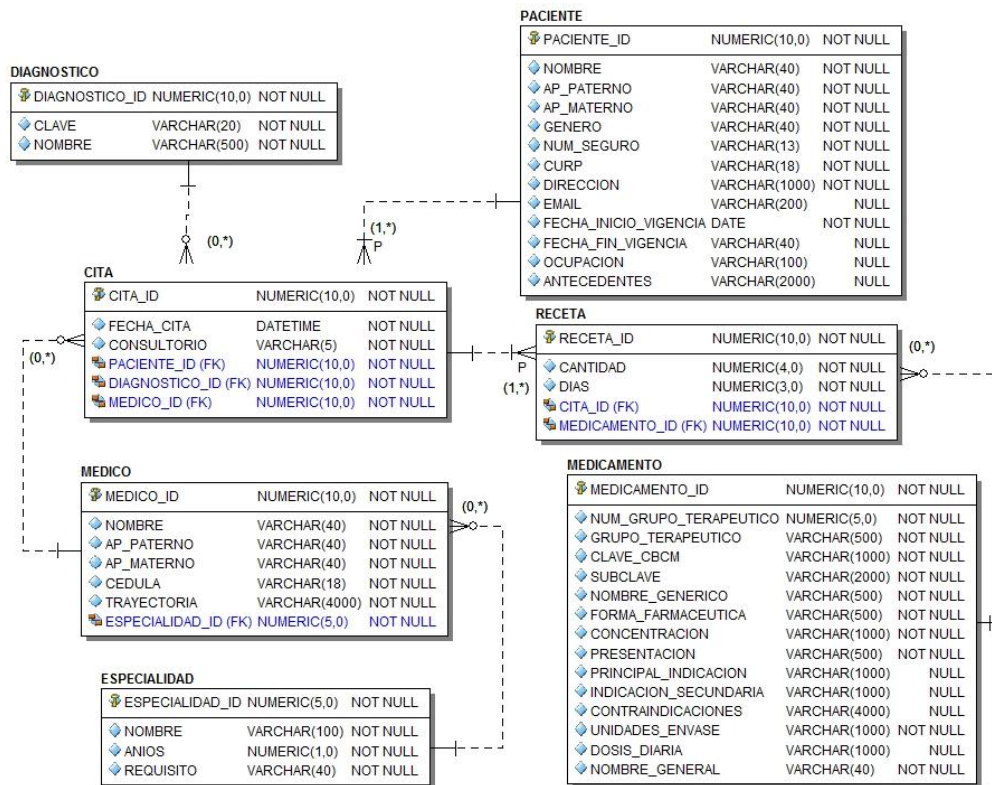
Nota: Los ejercicios se entregan de forma individual en cualquier formato.

TOTAL PUNTOS: _____ DE 100

1.1. PREGUNTAS.

1. Describa brevemente la función que desempeña cada uno de los componentes de un optimizador centralizado.
2. ¿Qué información de entrada requiere un estimador centralizado para operar de forma adecuada?
3. ¿Qué tipo de información produce como salida un estimador centralizado?
4. ¿Qué información se puede recuperar cuando se realiza un método de acceso `unique index scan`?

Para los siguientes ejercicios considere nuevamente el caso de estudio del registro de citas médicas.



- El código DDL y DML pueden descargarse de la carpeta compartida BDD. Ejecutar los scripts para resolver y verificar los resultados de los siguientes ejercicios.

1.2. PREDICCIÓN DE PLANES DE EJECUCIÓN.

10P por ejercicio.

Para cada una de las consultas de la siguiente tabla, indicar:

- La consulta SQL solicitada.
- Llenar una tabla con los métodos de acceso a datos y las fuentes de datos que se requieren para recuperar los datos solicitados. Notar que algunas consultas pueden requerir de varios pasos para obtener los datos solicitados. Ver ejemplo resuelto.

1.2.1. Ejercicio 1 (Resuelto)

- Obtener el nombre del paciente con curp 'ABC798701GYTOS7029'
- Considerar los siguientes índices:

```
create unique index paciente_curp_uk on paciente(curp);
create index paciente_nombre_ix on paciente(nombre);
```

Respuesta:

Num.	Consulta SQL	Orden de ejecución	Método de acceso a datos	Fuente de Datos.
1	<pre>select nombre from paciente where curp ='AAAAE344345WEDX'; on paciente(curp);</pre>	2	TABLE ACCESS BY INDEX ROW ID	PACIENTE
		1	UNIQUE INDEX SCAN	PACIENTE_NOMBRE_IX

En este ejercicio se requieren 2 pasos. El paso 1 se realiza un acceso al índice PACIENTE_NOMBRE_IX ya que el campo CURP está indexado y se tiene un predicado con el operador '=', y el tipo de índice es Unique. Del primer paso se obtendría un solo registro. Dicho registro está formado por la pareja (ROW_ID,ETIQUETA). En el paso 2 se emplea un acceso a la tabla empleando el ROW_ID del paso anterior para recuperar el registro de la tabla PACIENTE. Notar que a pesar de existir un índice en el campo nombre, este no se utiliza ya que el ROW_ID obtenido en el paso 1 solo es útil para acceder a la tabla.

1.2.2. Ejercicio 2

- Obtener los correos electrónicos de todos los pacientes. Solo se requiere mostrar los correos electrónicos que comiencen con 'r'.
- Notar que se trata de un índice tipo UNIQUE, y la columna email contiene valores nulos
- Considerar los siguientes índices:

```
create unique index paciente_emal_uk on paciente(email);
```

1.2.3. Ejercicio 3

- Obtener el nombre de todos los pacientes cuya ocupación sea 'GERENTE'
- Considerar los siguientes índices:

```
create index paciente_nombre_ix on paciente(nombre);
```

1.2.4. Ejercicio 4

- De la tabla Cita, obtener el identificador del médico que atenderá la cita con id 20.
- No olvidar que existe el índice tipo unique cita_pk asociado a la llave primaria de la tabla cita.
- Considerar los siguientes índices:

```
create index cita_medico_id_ix on cita(medico_id);
```

1.2.5. Ejercicio 5

- De la tabla Cita, obtener el identificador de la cita que va a ser atendida por el medico con id = 40
- Considerar los siguientes índices:

```
create index cita_medico_id_ix on cita(medico_id);
```

1.2.6. Ejercicio 6

- De la tabla medicamento, seleccionar la lista de sus identificadores.

1.2.7. Ejercicio 7

- Obtener el nombre y apellidos de todos los pacientes con genero igual a 'F'
- Considerar los siguientes índices:

```
create index paciente_genero_ix on paciente(genero);
```

1.2.8. Ejercicio 8

- Obtener los identificadores de los pacientes con genero igual a 'F'
- Considerar los siguientes índices:

```
create index paciente_genero_ix on paciente(genero);
```

1.2.9. Ejercicio 9

- Suponer que se lanza la siguiente consulta.

```
select cedula
from medico
where nombre = 'JUAN';
```

- Suponer los siguientes índices. Observar que se trata de un índice aplicado a la función upper.

```
create index medico_nombre_ix on medico (upper(nombre));
```

1.2.10. Ejercicio 10

- Suponer la siguiente consulta

```
select count(ocupacion)
from paciente;
```

- No olvidar que la columna ocupación tiene valores nulos, y los índices B Tree no soportan etiquetas nulas.
- Suponer los siguientes índices.

```
create index paciente_ocupacion_ix on paciente(ocupacion);
```