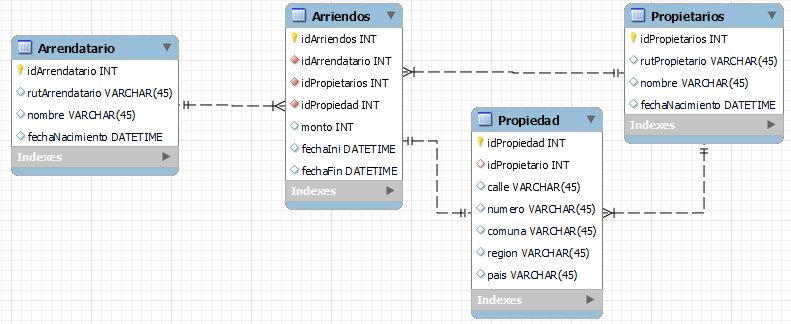
|  |
| --- |
|  |
|  |
| **CHALLENGE ANALISTA DESARROLLADOR BASE DE DATOS** |
|  |
|  |
| **2022** |

1.- Acorde al siguiente diagrama resolver las preguntas:



1. Indicar cuales arrendatarios sus arriendos vencen el próximo mes.

SELECT B.nombre

FROM Arriendos as A

INNER JOIN Arrendatario as B

ON A.idArrendatario=B. idArrendatario

WHERE A.fechaFin BETWEEN dateadd(dd, 1, EOMONTH(getdate())) and EOMONTH(dateadd(dd, 1, EOMONTH(getdate())))

1. Indicar cuales propietarios tienen al menos una propiedad sin arrendar.

SELECT A.nombre

FROM Propietarios A

INNER JOIN

(

SELECT idPropietarios, count(1) cantidad

FROM Arriendos

GROUP BY idPropietarios

) X

ON A.idPropietarios=X. idPropietarios

WHERE X.cantidad=0

1. Indicar cuantas propiedades tiene cada propietario por cada país.

SELECT A.nombre, B.pais, B.cantidad

FROM Propietarios A

INNER JOIN

(

SELECT idPropietarios, pais, count(1) cantidad

FROM Propiedad A

GROUP BY idPropietarios, país

) B

ON A. idPropietarios=B. idPropietarios

1. Indicar cuales propietarios son también arrendatarios.

SELECT P.nombre

FROM Propietarios P

INNER JOIN

(

SELECT A.idPropietarios, B.rutArrendatario

FROM Arriendos A

INNER JOIN Arrendatario B

ON A.idArrendatario=B.idArrendatario

) X

ON P.IdPropietarios=X.IdPropietarios

AND P.rutPropietario=X.rutArrendatario

1. Indicar cuales arrendatarios arriendan fuera de chile.

Select C.nombre

FROM Arrendatario C

INNER JOIN

(

SELECT A.idArrendatario

FROM Arriendo A

INNER JOIN Propiedad B

ON A.idPropiedad=B.idPropiedad

WHERE B.pais<>’CHILE’

) D

ON C.idArrendatario=D.idArrtendatario

1. Indicar cuales son los 3 países que el monto promedio de arriendo son los más altos.

SELECT TOP 3 P.pais, R.monto

FROM Propiedad P

INNER JOIN

(

SELECT idPropiedad, AVG(monto) monto

FROM Arriendos

GROUP BY idArriendos

) R

ON P.idPropiedad=R.idPropiedad

ORDER BY R.monto

1. Indicar el monto promedio, mínimo y máximo que pagan arrendatarios que también son propietarios.

SELECT P.nombre, R.monto\_promedio, R.monto\_minimo, R.monto\_maximo

FROM Propietarios P

INNER JOIN

(

SELECT A.IdPropietario, A.idArrendatario, B.rutArrendatario, AVG(A.monto) monto\_promedio, MIN(A.monto) monto\_minimo, MAX(A.monto) monto\_maximo

FROM Arriendos A, Arrendatario B

WHERE A.idArrendatario=B.idArrendatario

GROUP BY A.IdPropietario, A.idArrendatario

) R

ON P.IdPropietario=R.IdPropietario

AND P.rutPropietario=R.rutArrendatario

2.- Modelamiento de datos:

                La caja de compensación administra las licencias médicas a distintas empresas que son afiliadas. Estas licencias médicas pueden ser de distinto tipo (maternal, sanna etc) y también estas licencias se pueden presentar de forma manual, electrónica o mixta. Cada licencia médica tiene definido una serie de documentos que se deben enviar para realizar el cálculo, como por ejemplo las 3 últimas liquidaciones de sueldo y el contrato de trabajo. Para el caso de licencias manuales y mixtas, la documentación requerida se presenta de forma presencial en las sucursales en donde es digitada al sistema y se asigna un estado inicial, donde posteriormente se valida que estén todos los datos necesarios para realizar su cálculo. En caso que falte algún dato, se solicita el documento faltante para reiniciar el proceso. En el caso de licencias electrónicas todos los datos llegan sin problemas para realizar el cálculo.

Se necesita que se desarrolle un modelo de datos que soporte el flujo antes descrito y que responda a las siguientes preguntas (query’s):

Diagrama

Descripción generada automáticamente

1. Top de empresas que sus trabajadores presentan más licencias.

SELECT TOP 10

E,nombreEmpresa, L.cantidad

FROM [Empresa] E

LEFT JOIN

(

SELECT idEmpresa, COUNT(idLicencia) cantidad

FROM [Licencia]

GROUP BY idEmpresa

) L

ON E.idEmpresa=L.idEmpresa

ORDER BY L.cantidad desc

1. Las sucursales que reciben más documentación, segmentados por región o comuna, así como sucursales que no están aptas para recibir documentación.

SELECT idSucursal, comunaSucursal as Comuna, count(1) cantidad,

'aptas para recibir documentación' comentario

FROM [Sucursal]

WHERE licenciaFEisica=1

GROUP BY idSucursal, comunaSucursal

UNION ALL

SELECT idSucursal, comunaSucursal as Comuna, count(1) cantidad,

'NO están aptas para recibir documentación' comentario

FROM [Sucursal]

WHERE licenciaFEisica=0

GROUP BY idSucursal, comunaSucursal

ORDER BY 3 desc

1. Top de documentos que hacen que la licencia reinicie su flujo.

SELECT TOP 10 PL.ComentarioRechazo, count(1) cantidadDocumentos

FROM [PagoLicencias] PL

LEFT JOIN [EstadoLicencvias] EL

ON PL.idEstado=EL.idEstado

WHERE EL.nombreEstado<>'PAGADA'

GROUP BY PL.ComentarioRechazo

ORDER BY 2 DESC

1. Tiempos promedios, mínimos y máximos, desde el inicio del proceso hasta el cálculo del monto a pagar por cada licencia

SELECT R.idLicencia, DATEDIFF(dd, R.fecha\_recepcion, A.fecha\_resolucion) dias\_proceso

FROM

(

SELECT idLicencia, fecha\_recepcion

FROM [PagoLicencias] PL

LEFT JOIN [EstadoLicencvias] EL

ON PL.idEstado=EL.idEstado

WHERE EL.nombreEstado='RECEPCIONADA'

GROUP BY PL.ComentarioRechazo

) R

INNER JOIN

(

SELECT idLicencia, fecha\_resolucion

FROM [PagoLicencias] PL

LEFT JOIN [EstadoLicencvias] EL

ON PL.idEstado=EL.idEstado

WHERE EL.nombreEstado='APROBADA'

GROUP BY PL.ComentarioRechazo

) A

ON R.idLicencia=A.idLicencia

1. Estadísticas de licencias manuales vs electrónicas vs mixtas

SELECT T.nombreTipoLicencia, count(L.idLicencia)

FROM [Licencia] L

INNER JOIN TipoLicencia T

ON L.idLicencia=T.idLicencia

1. Los estados del proceso que almacenan la mayor cantidad de licencias así como los cambios de estado que tardan más tiempo en ser modificados

SELECT EL.nombreEstado, count(1) cantidad

FROM [PagoLicerncias] PL

LEFT JOIN [EstadoLicencias] EL

ON PL.idEstado=El.idEstado

1. Trabajadores que tienen licencia y son desafiliados

SELECT L.idLicencia

FROM [Licencia] L

LEFT JOIN [Trabajador] T

ON L.idTrabajador=T.idTrabajador

LEFT JOIN [PagoLicencias] P

ON L.idLicencia=P.idLicencia

WHERE T.estadoAfiliacion=0

1. Manejo de data histórica, de validación de procesos y log’s de cambios de estado o actualización de data relevante.

SELECT LEL.idLog, EL.nombreEstado, PL.fecha\_recepcion, PL.fecha\_resolucion, PL.comentarioRechazo,

LEL.idEstado, LEL.fecha

FROM [PagoLicencias] PL

INNER JOIN [LogEstadoLicencias] LEL

ON PL.idLicencia=LEL.idLicencia

INNER JOIN [EstadoLicencias] EL

ON PL.idEstado=El.idEstado

GROUP BY LEL.idLog, EL.nombreEstado, PL.fecha\_recepcion, PL.fecha\_resolucion, PL.comentarioRechazo,

LEL.idEstado, LEL.fecha

ORDER BY LEL.idLog, EL.nombreEstado, PL.fecha\_recepcion, PL.fecha\_resolucion, PL.comentarioRechazo,

LEL.idEstado, LEL.fecha

Se recomienda modelar hasta 3FN, se valorará la mayor cantidad de detalles en cuanto a entidades, atributos y relaciones.

Entregables:

* Diagrama de la elaboración del punto 2
* Repositorio Git con las querys de cada ejercicio

Suma puntos:

* Ambientar en servicio de nube (a elección del participante) BD relacional para probar las querys enviadas (enviar datos de acceso).

Tecnología:

* BD relacional de elección del participante

**Tiempos:**

**El ejercicio propuesto debe ser desarrollado en 3 días máximo.**

|  |
| --- |
|  |