**Co-Working Company
Nombre y Apellido:** Valentina Avariano

**Comisión del curso:** 47370

**Fecha:** 17/12/2023

**SQL**

**Proyecto Final SQL**

***Co-working Company***

* **Introducción**

Con el fin de realizar el presente trabajo, se creó una base de datos con información proveniente de una empresa de co-working, la cual pretende habilitar y establecer un modelo de negocio viable para ampliar la base de clientes.

La compañia se cataloga como una startup, por lo que el período a examinar se basa en datos generados en el 2023.

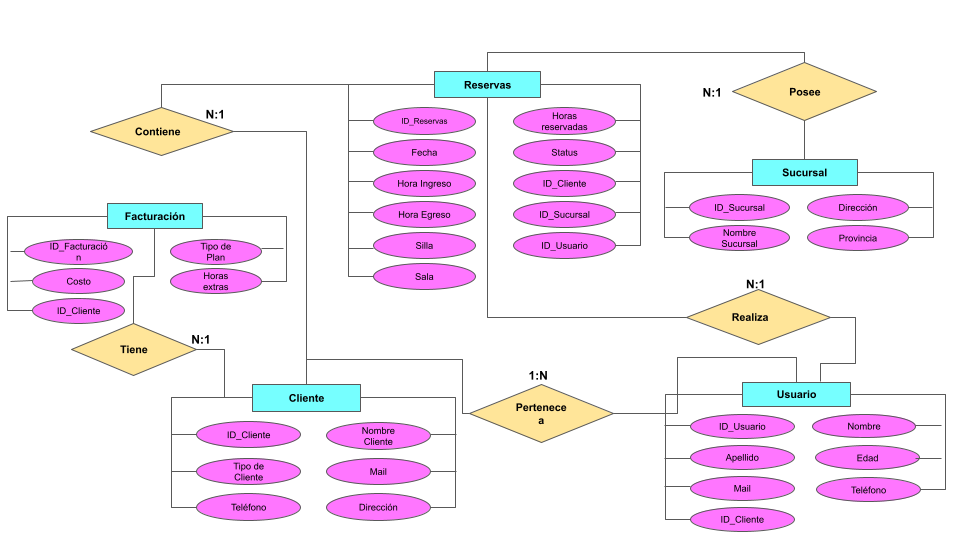
* **Objetivo**

Realizar una base de datos de una empresa en la que se alquilan espacios físicos de trabajo a distintos tipos de compañías.

* **Modelo de negocio**

Los datos creados se tomaron a partir de un negocio basado en el alquiler de sillas/escritorios/salas de reunión para diferentes empresas y/o trabajadores particulares que requieran espacios físicos para realizar sus jornadas laborales.

* **Diagrama Entidad-Relación**

****

* **Descripción de Tablas**

*Tabla Cliente*

| **CLIENTE** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Contiene información sobre las empresas que contratan el servicio de Co-working Company. | | | | | |
| **Campo** | **Tipo de Campo** | **Tipo de Clave** | **NOT NULL** | **Unique** | **Descripción** |
| ID\_Cliente | int | PK | TRUE | TRUE | Código del cliente |
| Nombre Cliente | varchar(50) | - | TRUE | - | Nombre del cliente |
| Tipo de Cliente | varchar(50) | - | - | - | Tipo de empresa cliente: large, medium, small o particular |
| Mail | varchar(100) | - | - | - | Correo electrónico del cliente |
| Telefono | int | - | - | - | Teléfono del cliente |
| Direccion | varchar(100) | - | - | - | Dirección del cliente |

*Tabla Facturación*

| **FACTURACIÓN** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Posee la información acerca de los saldos cargados a los clientes. | | | | | |
| **Campo** | **Tipo de Campo** | **Tipo de Clave** | **NOT NULL** | **Unique** | **Descripción** |
| ID\_Facturacion | int | PK | TRUE | TRUE | Código la factura mensual de los servicios solicitados por parte del cliente |
| Tipo de Plan | varchar(30) | - | TRUE | - | Tipo de plan adherido del cliente:  -Corporativo (más de 100 sillas al mes + 2 salas de reunión)  -Premium (50 a 99 sillas al mes + 1 sala de reunión)  -Plus (11 a 49 sillas al mes)  -Básico (5 a 10 sillas al mes)  -Sin Plan. |
| Costo | Float | - | DEFAULT 0 | - | Saldo mensual a cargar por los servicios solicitados por el cliente |
| ID\_Cliente | int | FK | TRUE | TRUE | Código del cliente |
| Horas extras | tinyint | - | - | - | Horas extras realizadas mensualmente por cliente |

*Tabla Usuario*

| **USUARIO** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Contiene información sobre los usuarios que asisten al espacio de Co-Working, podrían ser empleados de las empresas asociadas como clientes o particulares (también clientes). | | | | | |
| **Campo** | **Tipo de Campo** | **Tipo de Clave** | **NOT NULL** | **Unique** | **Descripción** |
| ID\_Usuario | int | PK | TRUE | TRUE | ID de los usuarios |
| Nombre | varchar(50) | - | - | - | Nombre de los usuarios |
| Apellido | varchar(50) | - | - | - | Apellido de los usuarios |
| Edad | tinyint | - | - | - | Edad de los usuarios |
| Mail | varchar(100) | - | - | - | Correo electrónico de los usuarios |
| Teléfono | int | - | - | - | Teléfono de los usuarios |
| ID\_Cliente | int | FK | TRUE | TRUE | Código del cliente |

*Tabla Sucursal*

| **SUCURSAL** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Posee la información sobre la sucursal reservada por los clientes | | | | | |
| **Campo** | **Tipo de Campo** | **Tipo de Clave** | **NOT NULL** | **Unique** | **Descripción** |
| ID\_Sucursal | int | PK | TRUE | TRUE | ID de la sucursal |
| Nombre Sucursal | varchar(50) | - | - | - | Nombre de la sucursal |
| Dirección | varchar(100) | - | TRUE | - | Dirección de la sucursal |
| Provincia | varchar(50) | - | - | - | Provincia donde se encuentra la sucursal |

*Tabla Reservas*

| **RESERVAS** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Contiene la información sobre las reservas realizadas por los clientes. | | | | | |
| **Campo** | **Tipo de Campo** | **Tipo de Clave** | **NOT NULL** | **Unique** | **Descripción** |
| ID\_Reservas | int | PK | TRUE | TRUE | ID de las reservas realizadas por los clientes |
|  |  |  |  |  |  |
| Fecha Hora Ingreso | TimeStamp | - | TRUE | - | Fecha y hora de ingreso al espacio de Co-working |
| Fecha Hora Egreso | TimeStamp | - | TRUE | - | Fecha y hora de egreso al espacio de Co-working |
| Silla | tinyint | - | - | - | Silla reservada |
| Estado | varchar(50) | - | TRUE | - | Estado de la reserva: Realizada, Cancelada, Pendiente.. |
| Horas reservadas | tinyint | - | - | - | Cantidad de horas reservadas |
| Sala | varchar(50) | - | - | - | Sala de reunión reservada:  -Sala Big VIP  -Sala Happy break  -Sala Spot  -Sala Innovation  -Sala Free |
| ID\_Sucursal | int | FK | TRUE | TRUE | ID de la sucursal |
| ID\_Cliente | int | FK | TRUE | TRUE | Código del cliente |
| ID\_Usuario | int | FK | TRUE | TRUE | ID de los usuarios |

* **Script - Creación de Tablas**

[**Script Formato SQL**](https://drive.google.com/file/d/1gkgKqSK00j4NNJyz7euEsFI13PG8rd_D/view?usp=sharing)

**Script:**

----------Tabla CLIENTE ---------

CREATE TABLE Cliente (

ID\_Cliente INT NOT NULL PRIMARY KEY,

Nombre\_Cliente VARCHAR(50) NOT NULL,

Tipo\_de\_Cliente VARCHAR(50),

Mail VARCHAR(100),

Teléfono INT,

Dirección VARCHAR(100)

);

----------Tabla FACTURACIÓN---------

CREATE TABLE Facturacion (

ID\_Facturacion INT NOT NULL PRIMARY KEY,

Tipo\_de\_Plan VARCHAR(30) NOT NULL,

Costo Decimal,

ID\_Cliente INT NOT NULL,

Horas\_extras TINYINT,

FOREIGN KEY(ID\_Cliente) REFERENCES Cliente(ID\_Cliente)

);

----------Tabla USUARIO---------

CREATE TABLE Usuario (

ID\_Usuario INT NOT NULL PRIMARY KEY,

Nombre VARCHAR(50),

Apellido VARCHAR(50),

Edad INT,

Mail VARCHAR(100),

Teléfono INT,

ID\_Cliente INT NOT NULL,

FOREIGN KEY(ID\_Cliente) REFERENCES Cliente(ID\_Cliente)

);

----------Tabla SUCURSAL---------

CREATE TABLE Sucursal (

ID\_Sucursal INT NOT NULL PRIMARY KEY,

Nombre\_Sucursal VARCHAR(50),

Direccion VARCHAR(100) NOT NULL,

Provincia VARCHAR(50)

);

----------Tabla RESERVAS---------

CREATE TABLE Reservas (

ID\_Reservas INT NOT NULL PRIMARY KEY,

Fecha TimeStamp NOT NULL,

Hora\_Ingreso TIME NOT NULL,

Hora\_Egreso TIME NOT NULL,

Silla TINYINT,

Estado VARCHAR(50),

Horas\_reservadas TINYINT,

Sala TINYINT,

ID\_Sucursal INT NOT NULL,

ID\_Cliente INT NOT NULL,

ID\_Usuario INT NOT NULL,

FOREIGN KEY(ID\_Sucursal) REFERENCES Sucursal(ID\_Sucursal),

FOREIGN KEY(ID\_Cliente) REFERENCES Cliente(ID\_Cliente),

FOREIGN KEY(ID\_Usuario) REFERENCES Usuario(ID\_Usuario)

);

* **Creación de Vistas**

Se crearon 5 vistas:

* *View FechaRes:* Muestra todas las reservas realizadas al momento junto a sus estados, sala (si corresponde), el usuario y la fecha.

Tablas: Reservas

Objetivo: Permite traer la información sobre las sillas reservadas por los usuarios junto al estado actual de la misma según las fechas, por lo que muestra el historial de los bookings.

* *View Vista\_Factu:* Muestra los datos necesarios para realizar la facturación a los clientes sobre las reservas realizadas. Contiene el tipo de plan al que se encuentra adherido el cliente, costo de sus reservas, ID Cliente junto a las horas extras que contienen un cargo adicional de su plan.

Tablas: Facturación

Objetivo: Ayuda a visualizar ágilmente los componentes necesarios para la facturación del cliente.

* View Vista\_ReservasSucu: Muestra las reservas realizadas exitosamente acorde a las sucursales elegidas por los usuarios.

Tablas: Reservas y Sucursal

Objetivo: Sirve para determinar la sucursal con mayor tráfico de reservas realizadas.

* View Vista\_FactuCliente: Muestra el costo mensual facturado a cada cliente según su tipo de plan.

Tablas: Facturación y Cliente

Objetivo: Permite ver los costos finales de facturación de los clientes.

* View Vista\_ResUsuario: Muestra la cantidad de horas reservadas exitosamente por cada usuario.

Tablas: Reservas y Usuario

Objetivo: Visualizar las horas reservadas por usuario.

* **Creación de Funciones**
  + Nombre: PorcentajeProfit (deterministic)
    - Nos permite calcular el porcentaje de ganancias que representa un valor con respecto al total
    - Se puede utilizar con cualquier tabla
    - Objetivo: facilitar el cálculo de porcentaje mediante dos valores (profit int, total int)
  + Nombre: ReservasXyear (reads sql data)
    - Nos permite hacer un conteo agrupado por año de la cantidad de reservas realizadas mediante el ingreso de un año (ano int)
    - Tabla que interviene: reservas
    - Objetivo: permitir saber el total de reservas para un año específico.
* **Creación de Stored Procedures**
  + Nombre: spOrdenCliente
    - Nos permite ordenar la tabla de artículos por la columna que le indiquemos de manera asc o desc
    - Tabla que interviene: Cliente
    - Objetivo: Facilitar una consulta simple a la tabla cientes ordenada por una columna de manera asc o desc
  + Nombre: spInsertarSucursal
    - Nos permite agregar a la tabla de sucursales un nuevo registro y retorna una consulta ordenada por id de sucursal
    - Tabla que interviene: Sucursal
    - Objetivo: Facilitar la creación de una sucursal nueva
* **Creación de Triggers**
  + Se crearon dos tabla de log y cuatro triggers. Dos para la tabla Sucursal y dos para la tabla Usuario
    - **Tabla:** Sucursal
      * **Tabla de LOG**: logSucursal
      * **Trigger BEFORE INSERT**: trigger\_log\_before\_insert
        + Este Trigger se ejecuta antes de una inserción de datos, coloca en la tabla de logSucursal quien hizo el insert y en qué fecha
      * **Trigger AFTER UPDATE**: trigger\_log\_after\_update
        + Este Trigger se ejecuta después de una actualización de datos, coloca en la tabla de logSucursal quien hizo el update y en qué fecha
    - **Tabla:** Usuario
      * **Tabla de LOG**: logUsuario
      * **Trigger BEFORE INSERT**: trigger\_logUSER\_before\_insert
        + Este se ejecuta antes de una inserción de datos, coloca en la tabla de logUsuario quien hizo el insert y en qué fecha
      * **Trigger AFTER INSERT**: trigger\_logUSER\_after\_update
        + Este Trigger se ejecuta después de una actualización de datos, coloca en la tabla de logUsuario quien hizo el update y en qué fecha
* **Herramientas y tecnologías usada**

Se hizo uso de los siguientes programas para dar forma a este proyecto:

-MySQL Workbench 8.0

-Google Sheets + Google Documents.

-Presentaciones de Google: para el diagrama DER

* **Conclusiones y futuras líneas**

La finalización de este proyecto y el análisis presentado en este informe subraya la importancia del adecuado uso de herramientas analíticas. Las tendencias observadas en los datos indican la necesidad de una rápida adaptación a conocimiento de los clientes.

En el futuro, se alienta a la empresa a centrarse en el análisis del tipo de reservas versus los clientes, de esta forma se podría trabajar en impulsar las ventas según el segmento de clientes y así evitar reservas canceladas, para seguir siendo competitiva y sostener el crecimiento.