**Diccionario de Datos: Gimnasio “BendiFit”**

**Requerimientos y descripción:**

* El cliente especifica que tienen proyecciones de crecimiento edilicio y mayor oferta de clases.
* Un cliente del gimnasio es una persona que abona el pase libre mensual del mismo, este incluye todas las clases que se dicten.
* En cada sala pueden ingresar como máximo 25 personas y tomar una clase de una hora, o dividirla en dos clases. Estas están identificadas algunas con nombre y número. Existen actualmente 5 salas.
* El gimnasio cuenta con 5 instructores, y cada uno puede dar como mínimo una clase, y varias durante el día.

**Tablas**

**Clientes:** Posee datos de los clientes quienes abonaron un pase mensual al gimnasio. De cada cliente se desea registrar, nombre, apellido, DNI, fecha de nacimiento, teléfono, altura, peso, apto físico y obra social (si tiene o no).

**Nombre de la tabla en la base de datos:** clientes

**Primary Key**: idCliente

**Campos**

* idCiente

Tipo de Datos: int(3) **PK**

Descripción: identificador de datos.

* nombre

Tipo de Datos: varchar(30)

Descripción: nombre de pila del cliente

* apellido

Tipo de Datos: varchar(30)

Descripción: apellido paterno del cliente o de soltera.

* DNI

Tipo de Datos: int(8)

Descripción: numero de documento nacional de identidad

* fechaNac

Tipo de Datos: date

Descripción: día mes y año de nacimiento

* telefono

Tipo de Datos: varchar(15)

Descripción: número de contacto del cliente (fijo o celular)

* altura

Tipo de Datos: decimal (3,2)

Descripción: altura del cliente en “metros,centímetros” (1,65m)

* peso

Tipo de Datos: decimal (3,2)

Descripción: peso del cliente en “kilogramos,gramos” (Ej: 84,5 Kg)

* aptoFisico

Tipo de Datos: boolean

Descripción: registro si posee o no apto fisico

* obraSocial

Tipo de Datos: boolean

Descripción: registro si posee o no obra social

**Direcciones:** Posee datos del domicilio de los clientes.

**Nombre de la tabla en la base de datos:** direcciones

**Campos**

* idDireccion

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* calle

Tipo de Datos: varchar(30)

Descripción: parte de la dirección del cliente, se especifica el nombre de la calle donde vive

* alturaCalle

Tipo de Datos: int(5)

Descripción: parte de la dirección del cliente, se especifica la altura de la calle donde vive

* piso

Tipo de Datos: int(2)

Descripción: parte de la dirección del cliente, se especifica el piso en donde vive

* departamento

Tipo de Datos: int(3)

Descripción: parte de la dirección del cliente, se especifica el departamento en donde vive

* localidad

Tipo de Datos: varchar(40)

Descripción: parte de la dirección del cliente, se la localidad del domicilio.

* codigoPostal

Tipo de Datos: varchar(8)

Descripción: parte de la dirección del cliente, se especifica el código postal donde vive.

**Clientes y Salas**

**Descripción**: Registro de ingreso a clientes a una sala para tomar clases, interesa saber que sala es y que cantidad de clientes ingresan (máximo 25 personas por clase).

**Nombre de la tabla en la base de datos**: asistencia

**Primary Key**: idAsistencia

**Foreign Keys:** idSala, idCliente

**Campos:**

* idAsistencia

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* idSala

Tipo de Datos: int(2) **FK**

Descripción: clave foránea desde tabla “salas”, índice restringido “on delete” y “on update”.

* idCliente

Tipo de Datos: int(3) **FK**

Descripción: clave foránea desde tabla “clientes”, índice restringido “on delete” y “on update”.

**Salones de clases**

**Descripción**: Registro de datos de los salones de clases, interesa saber la capacidad de alumnos, el número y nombre de sala.

**Nombre de la tabla en la base de datos**: salas

**Primary Key**: idSala

**Campos:**

* idSala

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* numSala

Tipo de Datos: int(2)

Descripción: numero del salón de clases

* nombreSala

Tipo de Datos: varchar(10)

Descripción: nombre del salón de clases

* capacidad

Tipo de Datos: int(3)

Descripción: numero del personas totales en el salón.

**Clases que se dictan**

**Descripción**: Registro de clases dictadas por instructores, interesa saber la capacidad de alumnos, la duración y nombre de la clase.

**Nombre de la tabla en la base de datos**: clases

**Primary Key**: idClase

**Campos:**

* idClase

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* nombreClase

Tipo de Datos: vachar(30)

Descripción: nombre del tipo de ejercicio que se dicta en la clase.

* duracion

Tipo de Datos: int(2) cambiar en der y tabla

Descripción: cantidad en minutos de duración de la clase

**Instructores de gimnasia.**

**Descripción**: Registro de datos de los profesores o instructores encargados de dictar clases. Interesa saber el nombre del instructor, apellido, teléfono, DNI y email.

**Nombre de la tabla en la base de datos**: instructores

**Primary Key**: idInstructor

**Campos:**

* idInstructor

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* nombre

Tipo de Datos: varchar(30)

Descripción: nombre de pila de instructor/a

* apellido

Tipo de Datos: varchar(30)

Descripción: apellido paterno o de soltera de instructor/a

* DNI

Tipo de Datos: int(8)

Descripción: documento nacional de identidad de instructor/a

* email

Tipo de Datos: varchar(20)

Descripción: correo electrónico instructor/a

**Clases y Clientes.**

**Descripción**: Registro de datos de presentismo a clases que dictan los profesores, interesa saber en qué clase es y que cliente asiste.

**Nombre de la tabla en la base de datos**: clase\_salon

**Primary Key**: idCursada

**Foreign Keys:** idClase, idCliente

**Campos:**

* idCursada

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* idClase

Tipo de Datos: int(3) **FK**

Descripción: clave foránea desde tabla “clases”, índice restringido “on delete” y “on update”.

* idCliente

Tipo de Datos: int(3) **FK**

Descripción: clave foránea desde tabla “clientes”, índice restringido “on delete” y “on update”.

**Instructores y Clases.**

**Descripción**: Registro de datos de presentismo a clases que dictan los profesores, interesa saber en qué clase es y que cliente asiste.

**Nombre de la tabla en la base de datos**: dictado\_clases

**Primary Key**: idDictado

**Foreign Keys:** idClase, idInstructor

**Campos:**

* idDictado

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* idInstructor

Tipo de Datos: int(2) **FK**

Descripción: clave foránea desde tabla “instructores”, índice restringido “on delete” y “on update”.

* idClase

Tipo de Datos: int(3) **FK**

Descripción: clave foránea desde tabla “clases”, índice restringido “on delete” y “on update”.

**Roles asignados**

**Descripción**: Registro de usuarios que ingresarán al sistema.

**Nombre de la tabla en la base de datos**: usuarios

**Primary Key**: idUsuario

**Campos:**

* idUsuario

Tipo de Datos: int(2) **PK**

Descripción: identificador de datos.

* nombreUsuario

Tipo de Datos: varchar(50)

Descripción: nombre del usuario a ser validado

* contrasena

Tipo de Datos: varchar(100)

Descripción: contraseña del usuario a ser validada.

**Estructuras de las tablas – SQL**

**Estructura de tabla para la tabla `asistencia`**

CREATE TABLE `asistencia` (

`idAsistencia` int(2) NOT NULL,

`idSala` int(2) NOT NULL,

`IdCliente` int(3) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `clases`**

CREATE TABLE `clases` (

`idClase` int(2) NOT NULL,

`Capacidad` int(3) NOT NULL,

`NombreClase` varchar(30) NOT NULL,

`Duracion` int(3) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `clase\_salon`**

CREATE TABLE `clase\_salon` (

`idCursada` int(2) NOT NULL,

`idCliente` int(3) NOT NULL,

`idClase` int(3) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `clientes`**

CREATE TABLE `clientes` (

`idCliente` int(8) NOT NULL,

`Nombre` varchar(30) NOT NULL,

`Apellido` varchar(30) NOT NULL,

`DNI` int(8) NOT NULL,

`FechaNac` date NOT NULL,

`Telefono` varchar(15) NOT NULL,

`Altura` decimal(3,2) NOT NULL,

`Peso` decimal(3,2) NOT NULL,

`AptoFisico` tinyint(1) NOT NULL,

`ObraSocial` tinyint(1) NOT NULL,

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `direcciones`**

CREATE TABLE `direcciones` (

`idDireccion` int(2) NOT NULL,

`idCliente` int(3) NOT NULL,

`Calle` varchar(30) NOT NULL,

`AlturaCalle` int(5) NOT NULL,

`Piso` int(2) DEFAULT NULL,

`Departamento` int(3) DEFAULT NULL,

`Localidad` varchar(40) NOT NULL,

`CodigoPostal` varchar(8) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `dictado\_clases`**

CREATE TABLE `dictado\_clases` (

`idDictado` int(2) NOT NULL,

`idInstructor` int(2) NOT NULL,

`idClase` int(3) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `instructores`**

CREATE TABLE `instructores` (

`idInstructor` int(2) NOT NULL,

`Nombre` varchar(30) NOT NULL,

`Apellido` varchar(30) NOT NULL,

`Telefono` varchar(15) NOT NULL,

`DNI` int(8) NOT NULL,

`Email` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `salas`**

CREATE TABLE `salas` (

`idSala` int(2) NOT NULL,

`Capacidad` int(2) NOT NULL,

`NumeroSala` int(2) NOT NULL,

`NombreSala` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Estructura de tabla para la tabla `usuarios`**

CREATE TABLE usuarios (

`idUsuario` int(2) PRIMARY KEY AUTO\_INCREMENT,

`nombreUsuario` varchar(50) NOT NULL,

`contrasena` varchar(100) NOT NULL,

`rol` ENUM('administrativo', 'cliente', 'instructor') NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Índices para tablas volcadas**

**Indices de la tabla `asistencia`**

ALTER TABLE `asistencia`

ADD PRIMARY KEY (`idAsistencia`),

ADD KEY `IDSala` (`idSala`,`IdCliente`),

ADD KEY `IdCliente` (`IdCliente`);

**Indices de la tabla `clase\_salon`**

ALTER TABLE `clase\_salon`

ADD PRIMARY KEY (`idCursada`),

ADD KEY `IDCliente` (`idCliente`,`idClase`),

ADD KEY `idClase` (`idClase`);

**Indices de la tabla `direcciones`**

ALTER TABLE `direcciones`

ADD PRIMARY KEY (`idDireccion`),

ADD KEY `IDCliente` (`idCliente`);

**Indices de la tabla `clientes`**

ALTER TABLE `clientes`

ADD PRIMARY KEY (`idCliente`);

**Indices de la tabla `dictado\_clases`**

ALTER TABLE `dictado\_clases`

ADD PRIMARY KEY (`idDictado`),

ADD KEY `IDInstructor` (`idInstructor`,`idClase`),

ADD KEY `idClase` (`idClase`);

**Indices de la tabla `instructores`**

ALTER TABLE `instructores`

ADD PRIMARY KEY (`idInstructor`);

**Indices de la tabla `salas`**

ALTER TABLE `salas`

ADD PRIMARY KEY (`idSala`);

**Indices de la tabla `usuarios`**

ALTER TABLE `usuarios`

ADD PRIMARY KEY (`idUsuario`);

**AUTO\_INCREMENT de las tablas volcadas**

**AUTO\_INCREMENT de la tabla `asistencia`**

ALTER TABLE `asistencia`

MODIFY `idAsistencia` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `direcciones`**

ALTER TABLE `direcciones`

MODIFY `idDireccion` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `clases`**

ALTER TABLE `clases`

MODIFY `idClase` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `clase\_salon`**

ALTER TABLE `clase\_salon`

MODIFY `idCursada` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `clientes`**

ALTER TABLE `clientes`

MODIFY `idCliente` int(8) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `dictado\_clases`**

ALTER TABLE `dictado\_clases`

MODIFY `idDictado` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `instructores`**

ALTER TABLE `instructores`

MODIFY `idInstructor` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `salas`**

ALTER TABLE `salas`

MODIFY `idSala` int(2) NOT NULL AUTO\_INCREMENT;

**AUTO\_INCREMENT de la tabla `usuarios`**

ALTER TABLE `usuarios`

MODIFY `idUsuarios` int(2) NOT NULL AUTO\_INCREMENT;

**Restricciones para tablas volcadas**

**Filtros para la tabla `asistencia`**

ALTER TABLE `asistencia`

ADD CONSTRAINT `asistencia\_ibfk\_1` FOREIGN KEY (`IdCliente`) REFERENCES `clientes` (`idCliente`),

ADD CONSTRAINT `asistencia\_ibfk\_2` FOREIGN KEY (`idSala`) REFERENCES `salas` (`idSala`);

**Filtros para la tabla `** direcciones **`**

ALTER TABLE `direcciones`

ADD CONSTRAINT `direcciones\_ibfk\_1` FOREIGN KEY (`IdCliente`) REFERENCES `clientes` (`idCliente`);

**Filtros para la tabla `clase\_salon`**

ALTER TABLE `clase\_salon`

ADD CONSTRAINT `clase\_salon\_ibfk\_1` FOREIGN KEY (`idCliente`) REFERENCES `clientes` (`idCliente`),

ADD CONSTRAINT `clase\_salon\_ibfk\_2` FOREIGN KEY (`idClase`) REFERENCES `clases` (`idClase`);

**Filtros para la tabla `dictado\_clases`**

ALTER TABLE `dictado\_clases`

ADD CONSTRAINT `dictado\_clases\_ibfk\_1` FOREIGN KEY (`idInstructor`) REFERENCES `instructores` (`idInstructor`),

ADD CONSTRAINT `dictado\_clases\_ibfk\_2` FOREIGN KEY (`idClase`) REFERENCES `clases` (`idClase`);

COMMIT;

***Manejo de contraseñas***

INSERT INTO usuarios (nombreUsuario, contrasena, rol) VALUES ('usuario1', SHA2('password123', 256), 'cliente');

***STORED PROCEDURES***

* Procedimiento AltaClienteConDireccion para dar de alta clientes nuevos. Lo realizan los usuarios ‘administrativos’

DELIMITER //

CREATE PROCEDURE AltaClienteConDireccion (

IN p\_nombre VARCHAR(30),

IN p\_apellido VARCHAR(30),

IN p\_DNI INT,

IN p\_fechaNac DATE,

IN p\_telefono VARCHAR(15),

IN p\_altura DECIMAL(3,2),

IN p\_peso DECIMAL(3,2),

IN p\_aptoFisico TINYINT,

IN p\_obraSocial TINYINT,

IN p\_calle VARCHAR(30),

IN p\_alturaCalle INT,

IN p\_piso INT,

IN p\_departamento INT,

IN p\_localidad VARCHAR(40),

IN p\_codigoPostal VARCHAR(8)

)

BEGIN

DECLARE cliente\_id INT;

INSERT INTO clientes (nombre, apellido, DNI, fechaNac, telefono, altura, peso, aptoFisico, obraSocial)

VALUES (p\_nombre, p\_apellido, p\_DNI, p\_fechaNac, p\_telefono, p\_altura, p\_peso, p\_aptoFisico, p\_obraSocial);

SET cliente\_id = LAST\_INSERT\_ID();

INSERT INTO direcciones (idCliente, calle, alturaCalle, piso, departamento, localidad, codigoPostal)

VALUES (cliente\_id, p\_calle, p\_alturaCalle, p\_piso, p\_departamento, p\_localidad, p\_codigoPostal);

END //

DELIMITER ;

* Procedimiento TomarAsistencia para toma de asistencia de clientes, lo realizan los usuarios ‘instructores’

DELIMITER //

CREATE PROCEDURE TomarAsistencia(

IN p\_idClase INT,

IN p\_idCliente INT

)

BEGIN

DECLARE claseExistente INT;

SELECT COUNT(\*) INTO claseExistente FROM clases WHERE idClase = p\_idClase;

IF claseExistente > 0 THEN

-- Registrar asistencia del cliente a la clase

INSERT INTO clase\_salon (idCliente, idClase) VALUES (p\_idCliente, p\_idClase);

SELECT 'Asistencia registrada exitosamente.' AS mensaje;

ELSE

SELECT 'La clase especificada no existe.' AS mensaje;

END IF;

END //

DELIMITER ;

***VISTAS***

* VistaDatosCliente Se consultan los datos del cliente, a que clases asistió y que instructores dictaron las clases. La pueden visualizar los usuarios ‘clientes’ y ‘administrativos’

CREATE VIEW VistaDatosCliente AS

SELECT c.idCliente, c.nombre AS NombreCliente, c.apellido AS ApellidoCliente, d.idDireccion, d.calle, d.alturaCalle, d.piso, d.departamento, d.localidad, d.codigoPostal,

cs.idCursada, cl.nombreClase, i.nombre AS NombreInstructor, i.apellido AS ApellidoInstructor

FROM clientes c

JOIN clase\_salon cs ON c.idCliente = cs.idCliente

JOIN clases cl ON cs.idClase = cl.idClase

JOIN dictado\_clases dc ON cl.idClase = dc.idClase

JOIN instructores i ON dc.idInstructor = i.idInstructor

LEFT JOIN direcciones d ON c.idCliente = d.idCliente;