

PROFESOR

PROFESOR ROGELIO PETENNÓN

BASE DE DATOS

IEI

INSTANCIA
EVALUATIVA
INTEGRADORA

Práctico Integrador

Trabajo rintegrador
realizado para las mate-
rias de Base de Datos y
Prácticas Profesionali-
zantes.
Esperamos les sea de su
agrado.

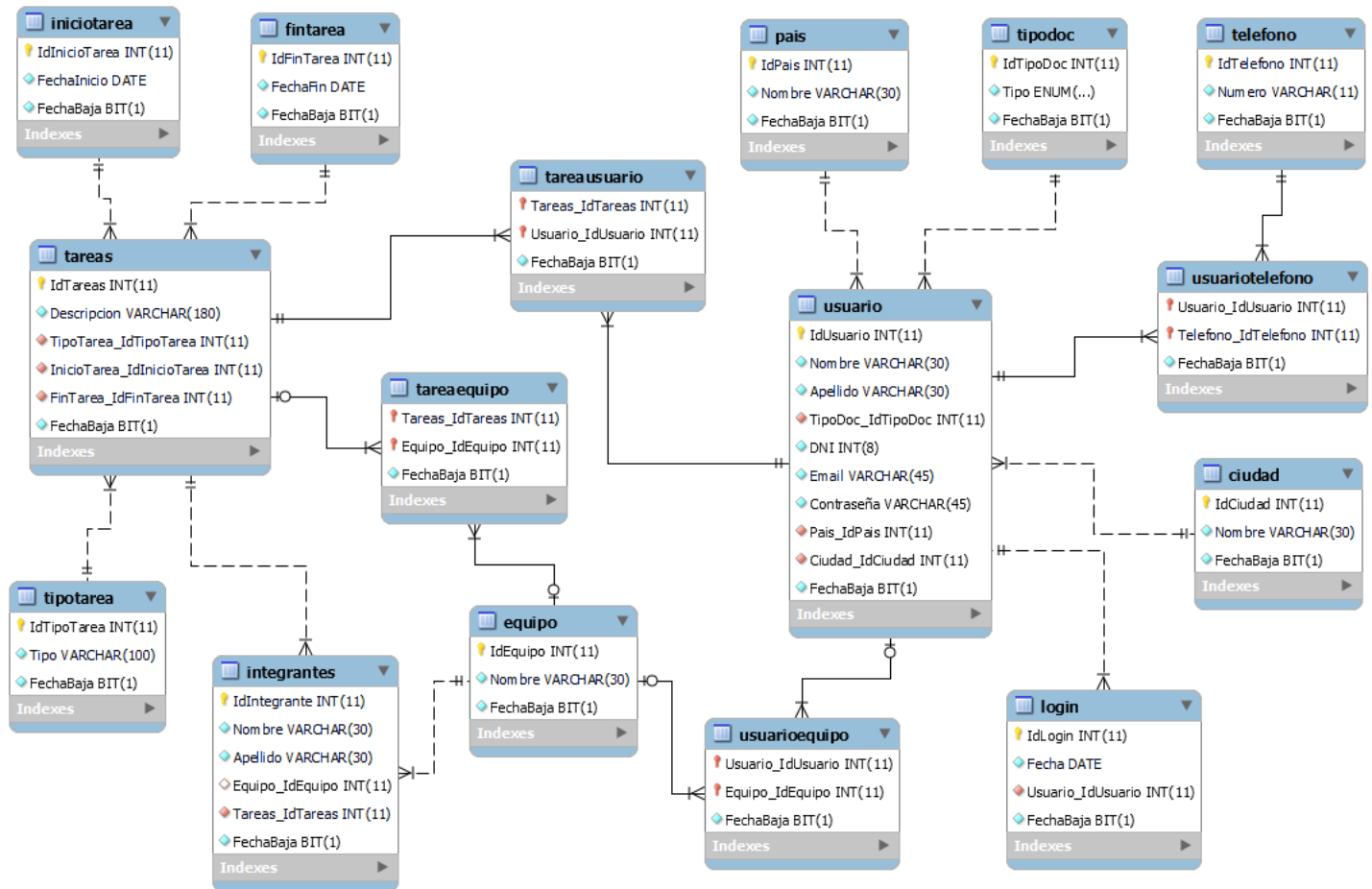
ALUMNOS

MÓNICA BEATRIZ ANTICH
ESCOBAR BRENDA GISELLE
IVAN MASSIMINO
DIEGO MARCEL SILVA
LÓPEZ JUAN MANUEL

AULA 4
2021

- > 01 Imagen del
modelo
relacional
generado
- > 02 Script DDL
- > 03 Consultas
SQL

MODELO RELACIONAL



SCRIPT DDL

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----
```

```
-- Schema Service
```

```
-- -----
```

```
drop schema if exists `ready_bd`;
```

```
-- -----
```

```
-- Schema Service
```

```
-- -----
```

```
create schema if not exists `ready_bd` default character set utf8;
```

```
use `ready_bd`;
```

```
-- -----
```

```
-- Tablas hijas
```

```
-- -----
```

```
-- -----
```

```
-- Tabla `ready_bd`.`TipoTarea`
```

```
-- Las tareas se van a dividir inicialmente en 3 tipos: "Tarea de Planificación", "Tarea de Extensión", "tarea de Práctica".
```

```
-- -----
```

```
create table if not exists `ready_bd`.`TipoTarea`
```

```
(
```

```

`IdTipoTarea` int not null auto_increment,
`Tipo` varchar(100) not null,
-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
registros.
`FechaBaja` bit(1) not null,
primary key(`IdTipoTarea`)
);

```

```

-----
-- Tabla `ready_bd`.`InicioTarea`
-----

```

```

create table if not exists `ready_bd`.`InicioTarea`
(
`IdInicioTarea` int not null auto_increment,
`FechaInicio` date not null,
-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
registros.
`FechaBaja` bit(1) not null,
primary key(`IdInicioTarea`)
);

```

```

-----
-- Tabla `ready_bd`.`FinTarea`
-----

```

```

create table if not exists `ready_bd`.`FinTarea`
(
`IdFinTarea` int not null auto_increment,
`FechaFin` date not null,
-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
registros.
`FechaBaja` bit(1) not null,

```

```
primary key(`IdFinTarea`)
```

```
);
```

```
-----
```

```
-- Tabla `ready_bd`.`TipoDoc`
```

```
-----
```

```
create table if not exists `ready_bd`.`TipoDoc`
```

```
(
```

```
`IdTipoDoc` int not null auto_increment,
```

```
`Tipo` enum('DNI','LC','LE') not null,
```

```
-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los registros.
```

```
`FechaBaja` bit(1) not null,
```

```
primary key(`IdTipoDoc`)
```

```
);
```

```
-----
```

```
-- Tabla `ready_bd`.`Pais`
```

```
-----
```

```
create table if not exists `ready_bd`.`Pais`
```

```
(
```

```
`IdPais` int not null auto_increment,
```

```
`Nombre` varchar(30) not null,
```

```
-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los registros.
```

```
`FechaBaja` bit(1) not null,
```

```
primary key(`IdPais`)
```

```
);
```

```
-----
```

```
-- Tabla `ready_bd`.`Ciudad`
```

```
-- -----  
  
create table if not exists `ready_bd`.`Ciudad`
```

```
(  
  `IdCiudad` int not null auto_increment,
```

```
  `Nombre` varchar(30) not null,
```

```
  -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los  
  registros.
```

```
  `FechaBaja` bit(1) not null,
```

```
  primary key(`IdCiudad`)
```

```
);
```

```
-- -----  
  
-- Tabla `ready_bd`.`Telefono`  
  
-- -----
```

```
create table if not exists `ready_bd`.`Telefono`
```

```
(  
  `IdTelefono` int not null auto_increment,
```

```
  `Numero` varchar(11) not null,
```

```
  -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los  
  registros.
```

```
  `FechaBaja` bit(1) not null,
```

```
  primary key(`IdTelefono`)
```

```
);
```

```
-- -----  
  
-- Tablas padres  
  
-- -----  
  
-- -----  
  
-- Tabla `ready_bd`.`Equipo`  
  
-- -----
```

```

create table if not exists `ready_bd`.`Equipo`
(
  `IdEquipo` int not null auto_increment,
  `Nombre` varchar(30) not null,
  `Usuario_IdUsuario` int null,
  -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
  registros.
  `FechaBaja` bit(1) not null,
  primary key(`IdEquipo`),
  constraint `Equipo_Usuario`
    foreign key(`Usuario_IdUsuario`)
    references `ready_bd`.`Usuario`(`IdUsuario`)
    on delete no action
    on update no action
);

```

```

-----
-- Tabla `ready_bd`.`Integrantes`
-----

```

```

create table if not exists `ready_bd`.`Integrantes`
(
  `IdIntegrante` int not null auto_increment,
  `Nombre` varchar(30) not null,
  `Apellido` varchar(30) not null,
  `Equipo_IdEquipo` int null,
  `Tareas_IdTareas` int not null,
  -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
  registros.
  `FechaBaja` bit(1) not null,
  primary key(`IdIntegrante`),
  constraint `Integrante_Equipo`

```

```
        foreign key(`Equipo_IdEquipo`)
references `ready_bd`.`Equipo`(`IdEquipo`)
on delete no action
on update no action,
constraint `Integrante_Tareas`
        foreign key(`Tareas_IdTareas`)
references `ready_bd`.`Tareas`(`IdTareas`)
on delete no action
on update no action
);
```

```
-- -----
-- Tabla `ready_bd`.`Login`
-- -----
```

```
create table if not exists `ready_bd`.`Login`
(
`IdLogin` int not null auto_increment,
`Fecha` date not null,
`Usuario_IdUsuario` int not null,
-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
registros.
`FechaBaja` bit(1) not null,
primary key(`IdLogin`),
constraint `Login_Usuario`
        foreign key(`Usuario_IdUsuario`)
references `ready_bd`.`Usuario`(`IdUsuario`)
on delete no action
on update no action
);
```

```
-- -----
```


-- Tabla `ready_bd`.`Usuario`

create table if not exists `ready_bd`.`Usuario`

(

 `IdUsuario` int not null auto_increment,

 `Nombre` varchar(30) not null,

 `Apellido` varchar(30) not null,

 `TipoDoc_IdTipoDoc` int not null,

 `DNI` int(8) not null,

 `Email` varchar(45) not null,

 `Contraseña` varchar(45) not null,

 `Pais_IdPais` int not null,

 `Ciudad_IdCiudad` int not null,

-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los registros.

 `FechaBaja` bit(1) not null,

primary key(`IdUsuario`),

constraint `Usuario_TipoDoc`

 foreign key(`TipoDoc_IdTipoDoc`)

 references `ready_bd`.`TipoDoc`(`IdTipoDoc`)

 on delete no action

 on update no action,

constraint `Usuario_Pais`

 foreign key(`Pais_IdPais`)

 references `ready_bd`.`Pais`(`IdPais`)

 on delete no action

 on update no action,

constraint `Usuario_Ciudad`

 foreign key(`Ciudad_IdCiudad`)

 references `ready_bd`.`Ciudad`(`IdCiudad`)

 on delete no action

```

        on update no action
    );

-----

-- Tabla `ready_bd`.`Tareas`
-----

create table if not exists `ready_bd`.`Tareas`
(
    `IdTareas` int not null auto_increment,
    `Descripcion` varchar(180) not null,
    `TipoTarea_IdTipoTarea` int not null,
    `InicioTarea_IdInicioTarea` int not null,
    `FinTarea_IdFinTarea` int not null,
    -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
    registros.
    `FechaBaja` bit(1) not null,
    primary key(`IdTareas`),
    constraint `Tareas_TipoTarea`
        foreign key(`TipoTarea_IdTipoTarea`)
        references `ready_bd`.`TipoTarea`(`IdTipoTarea`)
        on delete no action
        on update no action,
    constraint `Tareas_InicioTarea`
        foreign key(`InicioTarea_IdInicioTarea`)
        references `ready_bd`.`InicioTarea`(`IdInicioTarea`)
        on delete no action
        on update no action,
    constraint `Tareas_FinTarea`
        foreign key(`FinTarea_IdFinTarea`)
        references `ready_bd`.`FinTarea`(`IdFinTarea`)
        on delete no action

```

```

on update no action

);

-----

-- Tablas compuestas

-----

-----

-- Tabla `ready_bd`.`UsuarioTelefono`

-----

create table if not exists `ready_bd`.`UsuarioTelefono`
(
  `Usuario_IdUsuario` int not null,
  `Telefono_IdTelefono` int not null,
  -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los
  registros.
  `FechaBaja` bit(1) not null,
  primary key(`Usuario_IdUsuario`, `Telefono_IdTelefono`),
  constraint `UsuarioTelefono_Usuario`
    foreign key(`Usuario_IdUsuario`)
    references `ready_bd`.`Usuario`(`IdUsuario`)
    on delete no action
    on update no action,
  constraint `UsuarioTelefono_Telefono`
    foreign key(`Telefono_IdTelefono`)
    references `ready_bd`.`Telefono`(`IdTelefono`)
    on delete no action
    on update no action
);

-----

```

```
-- Tabla `ready_bd`.`TareaUsuario`
```

```
-----
```

```
create table if not exists `ready_bd`.`TareaUsuario`
```

```
(
```

```
`Tareas_IdTareas` int not null,
```

```
`Usuario_IdUsuario` int not null,
```

-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los registros.

```
`FechaBaja` bit(1) not null,
```

```
primary key(`Tareas_IdTareas`, `Usuario_IdUsuario`),
```

```
constraint `TareaUsuario_Tareas`
```

```
foreign key(`Tareas_IdTareas`)
```

```
references `ready_bd`.`Tareas`(`IdTareas`)
```

```
on delete no action
```

```
on update no action,
```

```
constraint `TareaUsuario_Usuario`
```

```
foreign key(`Usuario_IdUsuario`)
```

```
references `ready_bd`.`Usuario`(`IdUsuario`)
```

```
on delete no action
```

```
on update no action
```

```
);
```

```
-----
```

```
-- Tabla `ready_bd`.`TareaEquipo`
```

```
-----
```

```
create table if not exists `ready_bd`.`TareaEquipo`
```

```
(
```

```
`Tareas_IdTareas` int null,
```

```
`Equipo_IdEquipo` int null,
```

-- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los registros.

```

`FechaBaja` bit(1) not null,
primary key(`Tareas_IdTareas`, `Equipo_IdEquipo`),
constraint `TareaEquipo_Tareas`
    foreign key(`Tareas_IdTareas`)
    references `ready_bd`.`Tareas`(`IdTareas`)
    on delete no action
    on update no action,
constraint `TareaEquipo_Equipo`
    foreign key(`Equipo_IdEquipo`)
    references `ready_bd`.`Equipo`(`IdEquipo`)
    on delete no action
    on update no action
);

```

```

-----
-- Tabla `ready_bd`.`UsuarioEquipo`
-----

```

```

create table if not exists `ready_bd`.`UsuarioEquipo`
(
    `Usuario_IdUsuario` int null,
    `Equipo_IdEquipo` int null,
    -- los campos FechaBaja fueron generados para crear una sentencia logica que permita "borrar los datos" sin eliminar los registros.
    `FechaBaja` bit(1) not null,
    primary key(`Usuario_IdUsuario`, `Equipo_IdEquipo`),
    constraint `UsuarioEquipo_Usuario`
        foreign key(`Usuario_IdUsuario`)
        references `ready_bd`.`Usuario`(`IdUsuario`)
        on delete no action
        on update no action,
    constraint `UsuarioEquipo_Equipo`

```

```
foreign key(`Equipo_IdEquipo`)  
references `ready_bd`.`Equipo`(`IdEquipo`)  
on delete no action  
on update no action  
);  
SET SQL_MODE=@OLD SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD UNIQUE_CHECKS;
```

CONSULTAS DDL

a. Registrar un nuevo usuario a la base de datos.

```
insert into Usuario (Nombre,Apellido,TipoDoc_IdTipoDoc,DNI,Email,Contraseña,Pais_IdPais,Ciudad_IdCiudad,FechaBaja)
values ('Juan','López','1','36125641','juanem@hotmail.com','123456','1','1',1);
```

b. Consultar si existe el usuario “Juan” en la base de datos y si la contraseña “123456” es correcta.

```
SELECT `Nombre`, `Contraseña` FROM `Usuario` WHERE `Nombre` LIKE 'Juan' AND Contraseña = '123456';
```

c. Insertar un registro en nuestra entidad principal. Siguiendo con el ejemplo anterior, sería la entidad Vehículo.

```
insert into `Tareas`
(`Descripcion`,`TipoTarea_IdTipoTarea`,`InicioTarea_IdInicioTarea`,`FinTarea_IdFinTarea`,`FechaBaja`)
values ('Sacar la basura','3','1','1',1);
```

d. Actualizar un dato en nuestra entidad principal para un registro específico.

```
update `Tareas` set `TipoTarea_IdTipoTarea` = '2' where `IdTareas` = '1' and `TipoTarea_IdTipoTarea` = 3;
```

e. Actualizar un dato en nuestra entidad principal para un grupo de registros.

```
update `Tareas` set `TipoTarea_IdTipoTarea` = '1' where `TipoTarea_IdTipoTarea` >= 2;
```

f. Eliminar un registro específico de la base de datos.

```
SET SQL_SAFE_UPDATES = 0;
delete from `Tareas` where `Descripcion` LIKE 'Sacar%';
SET SQL_SAFE_UPDATES = 1;
```