

S.I.G.T.

BASE DE DATOS II TESTING DUMMIES

Rol	Apellido	Nombre	C.I	Email	Tel/Cel.
Coordinador	Torterolo	Alejo	5549974-9	alejotorterolo.com@gmail.com	097 301 357
Sub-Coordiador	Franca de Lima	Matías	5472569-0	matiasfdlv@gmail.com	094 153 894
Integrante 1	Olsztyn	Mario	5537713-5	olsztyn322@gmail.com	095 937 863
Integrante 2	Cabrera	Santiago	5547462-4	sgcr987@gmail.com	095 032 337

Docente: Carámbula, Leonardo

**Fecha de
culminación
13/11/2023**

Tercera entrega

I.S.B.O.

3BG



Testing Dummies

Montevideo, 13/11/2023

Carámbula Leonardo

Asignatura: Sistemas de Base de Datos II

Instituto Superior Brazo Oriental

Presente.

A continuación, los alumnos de 3BG del turno matutino del Instituto Superior Brazo Oriental nos presentamos ante usted, con el fin de informar la creación del grupo

“Testing Dummies”. Los correspondientes integrantes con sus roles son los siguientes:

A continuación, se detalla dicha integración y roles del grupo:

ROL	C.I	APELLIDO	NOMBRE	E-MAIL	TEL/CEL
Coordinador	55499749	Torterolo	Alejo	alejotorterolo.com@gmail.com	097301357
Subcoordinador	54725690	Franca de Lima	Matías	matiasfdlv@gmail.com	094153894
Integrante 1	55377135	Olsztyn	Mario	olsztyn322@gmail.com	095937863
Integrante 2	55474624	Cabrera	Santiago	sgcr987@gmail.com	095032337

Por contacto al correo: testingdummies0@gmail.com

Firmas:

COORDINADOR

SUBCOORDINADOR

INTEGRANTE 1

INTEGRANTE 2

ISBO

3BG

2



Torterolo, Alejo

COORDINADOR



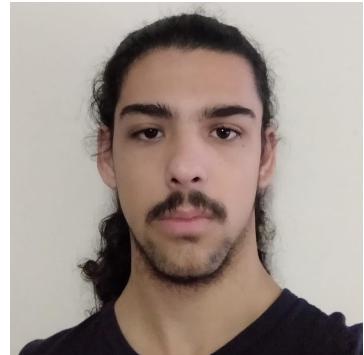
Franca de Lima, Matías

SUBCOORDINADOR



Olsztyn, Mario

INTEGRANTE 1



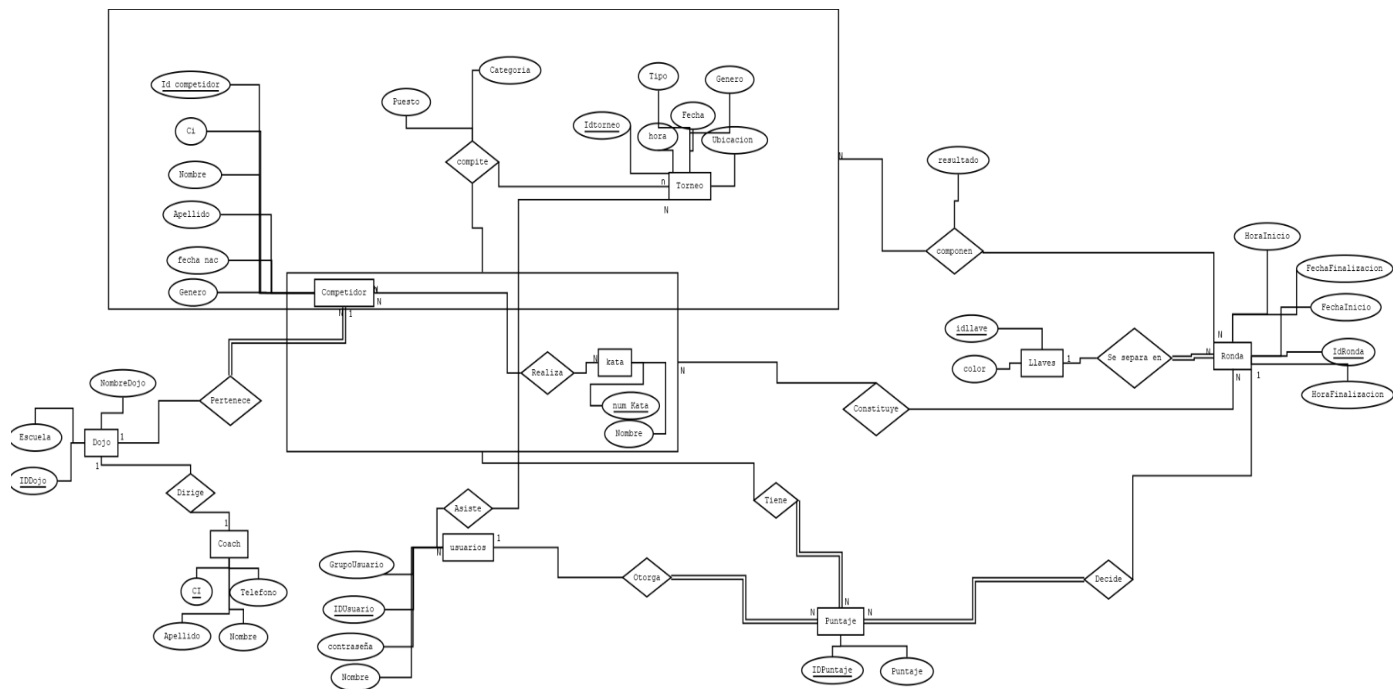
Cabrera, Santiago

INTEGRANTE 2

Índice

D.E-R.....	5
Pasaje a tablas.....	6
Normalización.....	7
Diccionario de datos.....	10
Sentencias DDL para la creación de tablas y estructuras necesarias.....	15
Sentencias DCL de permisos a la Base de Datos.....	29
Estudio de los permisos sobre BD, tablas y columnas, considerando los diferentes roles.....	30
Respaldar y recuperar datos de la base de datos.....	31
Consultas SQL.....	32

D.E-R.



RNE

RNE

- Competidor solo puede competir en pooles del torneo en el que se encuentra
- Un solo competidor no puede realizar 2 veces el mismo kata por torneo
 - El puntaje es de cada juez a cada pool
- El número 1 y número 2 del ranking o campeonato anterior, deben estar en semi llaves (pool) distintos (uno con AKA el otro con AO). Esto para evitar que se eliminen entre ellos en las primeras rondas.
- Se debe evitar que competidores pertenecientes a la misma escuela no queden en el mismo pool; por tal motivo, al momento de la inscripción, se debe ingresar la escuela a la pertenece. Esto no siempre es posible, ya que cuando son pocos competidores, esta casuística suele darse con frecuencia. Ej: 5 competidores anotados y 3 pertenecen a la misma escuela.
- Color es ao(azul) y aka(rojo) compiten contra el mismo color, y en la ronda final pueden competir contra el otro color
 - Los jueces no pueden puntuar mas de una ronda a la vez
- Limite de puntaje de 5 a 10(Puede ingresar 9.9, por ejemplo). Si alguien tiene 0 es eliminado.
 - Usuarios (Juez1 tiene mayores privilegios que el resto de jueces)
- Segun la fecha de nacimiento del competidor, se le añadira a una categoria de edad u otra.
 - Si un competidor cumple antes de la mitad del año se le cambia de categoria.
 - Si cumple despues de la mitad del año no se cambia de categoria

Pasaje a tablas.

Entidades:

Competidor (IDCompetidor, CI, Nombre, apellido, Fecha_nac, sexo, IDDojo)

IDDojo → Dojo

Dojo(IDDojo, NombreDojo, escuela)

Kata (NumKata, Nombre)

Torneo (IDTorneo, Ubicación, fecha, hora, Género, Tipo)

Llaves (IDLlave, color)

Ronda (IDRonda, FechaInicio, FechaFinalizacion, HoraInicio, HoraFinalizacion, IDLlave)

IDLlave → Llaves

Usuarios (IdUsuario, Contraseña, Nombre, GrupoUsuario)

Puntaje(IDPuntaje, puntaje, idusuarios, idcompetidor, idronda, numKata)

IdUsuario → Usuario

IDRonda → Ronda

Idcompetidor → Competidor

NumKata → Kata

Coach(CI, Teléfono, Nombre, Apellido)

Relaciones:

Componen (IDRonda, IDLlave, IDTorneo, IDCompetidor , Resultado)

IDRonda, IDLlave → Se separa en

IDTorneo, IDCompetidor → Compite

Compite (IDTorneo, IDCompetidor, NumKata, Puesto, Categoría)

IDTorneo → Torneo

IDCompetidor → Competidor

NumKata → Kata

Realiza (NumKata, IDCompetidor)

NumKata → Kata

IDCompetidor → Competidor

Constituye (NumKata, IDCompetidor, IDRonda)

NumKata, IDCompetidor → Realiza

IDRonda → Ronda

Asiste (IdUsuario, IDTorneo)

IDTorneo → Torneo

IdUsuario → Usuario

Dirige (IDDojo, CI)

CI → Coach

IDDojo → Dojo

Normalización.

1era forma normal:

Competidor (IDCompetidor, CI, Nombre, apellido. Fecha_nac, sexo, IDDojo)

IDDojo → Dojo

Dojo(IDDojo, NombreDojo, escuela)

Kata (NumKata, Nombre)

Torneo (IDTorneo, Ubicación, fecha, hora, Género, Tipo)

Llaves (IDLlave, color)

Ronda (IDRonda, FechaInicio, FechaFinalizacion, HoraInicio, HoraFinalizacion, IDLlave)

IDLlave → Llaves

Usuarios (IdUsuario, Contraseña, Nombre, GrupoUsuario)

Puntaje(IDPuntaje, puntaje, idusuarios, idcompetidor, idronda, NumKata)

IdUsuario → Usuario

IDRonda → Ronda

Idcompetidor → Competidor

NumKata → Kata

Coach(CI, Teléfono, Nombre, Apellido)

Relaciones 1era forma normal:

Componen (IDRonda, IDTorneo, IDCompetidor , Resultado)

IDRonda → Ronda

IDTorneo, IDCompetidor → Compite

Compite (IDTorneo, IDCompetidor, NumKata, Puesto, Categoría)

IDTorneo → Torneo

IDCompetidor → Competidor

NumKata → Kata

Realiza (NumKata, IDCompetidor)

NumKata → Kata

IDCompetidor → Competidor

Constituye (NumKata, IDCompetidor, IDRonda)

NumKata, IDCompetidor → Realiza

IDRonda → Ronda

Asiste (IdUsuario, IDTorneo)

IDTorneo → Torneo

IdUsuario → Usuario

Dirige (IDDojo, CI)

CI → Coach

IDDojo → Dojo

2da forma normal:

Competidor (IDCompetidor, CI, Nombre, apellido, Fecha_nac, sexo, IDDojo)

IDDojo → Dojo

Dojo(IDDojo, NombreDojo, escuela)

Kata (NumKata, Nombre)

Torneo (IDTorneo, Ubicación, fecha, hora, Género, Tipo)

Llaves (IDLlave, color)

Ronda (IDRonda, FechaInicio, FechaFinalizacion, HoraInicio, HoraFinalizacion, IDLlave)

IDLlave → Llaves

Usuarios (IdUsuario, Contraseña, Nombre, GrupoUsuario)

Puntaje(IDPuntaje, puntaje, idusuarios, idcompetidor, idronda, NumKata)

IdUsuario → Usuario

IDRonda → Ronda

Idcompetidor → Competidor

NumKata → Kata

Coach(CI, Teléfono, Nombre, Apellido)

Relaciones 2da forma normal:

Componen (IDRonda, IDTorneo, IDCompetidor , Resultado)

IDRonda → Ronda

IDTorneo, IDCompetidor → Compite

Compite (IDTorneo, IDCompetidor, NumKata, Puesto , Categoría)

IDTorneo → Torneo

IDCompetidor → Competidor

NumKata → Kata

Realiza (NumKata, IDCompetidor)

NumKata → Kata

IDCompetidor → Competidor

Constituye (NumKata, IDCompetidor, IDRonda)

NumKata, IDCompetidor → Realiza

IDRonda → Ronda

Asiste (IdUsuario, IDTorneo)

IDTorneo → Torneo

IdUsuario → Usuario

Dirige (IDDojo, CI)

CI → Coach

IDDojo → Dojo

3era forma normal:

Competidor (IDCompetidor, CI, Nombre, apellido, Fecha_nac, sexo, IDDojo)

IDDojo → Dojo

Dojo(IDDojo, NombreDojo, escuela)

Kata (NumKata, Nombre)

Torneo (IDTorneo, Ubicación, fecha, hora, Género, Tipo)

Llaves (IDLlave, color)

Ronda (IDRonda, FechaInicio, FechaFinalizacion, HoraInicio, HoraFinalizacion, IDLlave)

IDLlave → Llaves

Usuarios (IdUsuario, Contraseña, Nombre, GrupoUsuario)

Puntaje(IDPuntaje, puntaje, idusuarios, idcompetidor, idronda, NumKata)

IdUsuario → Usuario

IDRonda → Ronda

Idcompetidor → Competidor

NumKata → Kata

Coach(CI, Teléfono, Nombre, Apellido)

Relaciones 3era forma normal:

Componen (IDRonda, IDTorneo, IDCompetidor , Resultado)

IDRonda → Ronda

IDTorneo, IDCompetidor → Compite

Compite (IDTorneo, IDCompetidor, NumKata, Puesto, Categoría)

IDTorneo → Torneo

IDCompetidor → Competidor

NumKata → Kata

Realiza (NumKata, IDCompetidor)

NumKata → Kata

IDCompetidor → Competidor

Constituye (NumKata, IDCompetidor, IDRonda)

NumKata, IDCompetidor → Realiza

IDRonda → Ronda

Asiste (IdUsuario, IDTorneo)

IDTorneo → Torneo

IdUsuario → Usuario

Dirige (IDDojo, CI)

CI → Coach

IDDojo → Dojo

Diccionario de datos.

Tabla	Atributos	Tipo	Largo	Restricciones	Descripción
Asiste	IDTorneo	int	20	clave primaria clave foránea	identificador de torneo
	IDusuario	int	20	clave primaria clave foránea	Identificador de usuario
Coach	Apellido	varchar	20	No vacío	apellido del coach
	CI	int	8	Clave primaria	Cedula de identidad del coach
	Nombre	varchar	20	No vacío	Nombre del coach
	Teléfono	Int	20	No vacío	Teléfono del coach
Competidor	Apellido	varchar	20	No vacío	Apellido del competidor
	CI	int	8	No vacío	Cédula de identidad del competidor
	Fecha_nac	date	-	No vacío	Fecha de nacimiento del competidor
	IDCompetidor	int	Auto Increment	Clave Primaria	Identificador del competidor
	Nombre	varchar	20	No vacío	Nombre del competidor
	Género	Enum	-	No vacío	Sexo del competidor(Masculino o femenino)

	IDDojo	int	20	Clave foránea	Identificador de dojo
Compite	Categoría	Enum	-	No vacío	La categoría en la que esta inscripto el competidor(12/13, 14/15, 16/17 o mayores)
	IDCompetidor	int	20	clave primaria clave foránea	identificador de competidor
	Idtorneo	int	20	clave primaria clave foránea	identificador de torneo
	NumKata	int	20	Clave primaria clave foránea	Número de kata.
Componen	IDCompetidor	int	20	clave primaria clave foránea	Identificador competidor
	IDRonda	int	20	clave primaria clave foránea	Identificador de ronda
	IDTorneo	int	20	clave primaria clave foránea	Identificador de torneo
	Resultado	Boolean	-	No vacío	Gano o perdio
Constituye	IDCompetidor	int	20	clave primaria clave foránea	Identificador de competidor
	IDRonda	int	20	clave primaria clave foránea	Identificador de ronda
	NumKata	int	20	clave primaria clave foránea	Número de kata
Dirige	CI	int	8	clave foránea	Cédula de identidad del Coach
	IDDojo	int	20	clave foránea	Identificador de Dojo

Dojo	IDDojo	int	Auto Increment	Clave Primaria	IDentificado r de Dojo
	NombreDojo	varchar	30	No vacío	Nombre del Dojo
	escuela	varchar	30	No vacío	Nombre escuela
Kata	Nombre	varchar	20	No vacío	Nombre del kata
	NumKata	int	20	clave primaria	Número del kata
Llaves	Color	varchar	10	No vacío	Color de la llave (Aka o Ao)
	IDLlave	int	Auto-Increment	clave primaria	Identificado r de Llave
Puntaje	IdCompetidor	int	20	Clave Foránea	Identificado r de competidor
	IDPuntaje	int	Auto Increment	clave primaria	Identificado r puntaje
	Puntaje	decimal(4,2)	10	No vacío	El puntaje dado por un juez a un competidor
	IdUsuario	int	20	Clave foránea	identificado r de usuario
	IdRonda	int	20	clave foránea	identificado r de ronda
	NumKata	int	20	Clave foránea	Número del kata
Ronda	FechaFinalizacion	Date	-	No vacío	Fecha en la que finaliza la ronda
	FechaInicio	Date	-	No vacío	Fecha inicio de la ronda
	HoraFinalizacion	Time	-	No vacío	Hora en la que finaliza la ronda

	Horainicio	Time	-	No vacío	Hora en la que inicia la ronda
	IDLlave	int	20	clave foránea	identificador de llave
	IDRonda	int	Auto Increment	Clave Primaria	Identificador de ronda
Torneo	Fecha	Date	-	No vacío	Fecha en la que se realiza el torneo
	Género	Enum	-	No vacío	Tipo de torneo(Masculino o Femenino)
	Hora	Time	-	No vacío	Hora en la que empieza el torneo
	IDTorneo	int	Auto Increment	Clave Primaria	Identificador de torneo
	Tipo	Enum	-	No vacío	Tipo (Individual o grupal)
	Ubicación	Varchar	30	No vacío	Lugar en el que se realiza el torneo
Realiza	IDUsuario	int	20	clave foránea clave primaria	identificador de usuario
	Num kata	int	20	clave foránea clave primaria	Número de kata
Usuarios	Contraseña	Varchar	20	No vacío	Contraseña del usuario
	GrupoUsuarios	Enum	-	No vacío	Grupo al que pertenece el usuario (AdministradorBDKarate, juez,

					administrador)
	IDUsuario	int	Auto Increment	clave primaria	Identificador del usuario
	Nombre	varchar	20	No vacío	Nombre del usuario

Sentencias DDL para la creación de tablas y estructuras necesarias.

```
-- Base de datos: `testingdummies`  
  
--  
  
-----  
  
--  
  
-- Estructura de tabla para la tabla `asiste`  
--  
  
CREATE TABLE `asiste` (  
  `idUserario` int(11) NOT NULL,  
  `idTorneo` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `coach`  
--  
  
CREATE TABLE `coach` (  
  `CI` int(8) NOT NULL,  
  `Telefono` int(20) NOT NULL,  
  `Nombre` varchar(20) NOT NULL,  
  `Apellido` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;
```



```
-- Estructura de tabla para la tabla `competidor`
--

CREATE TABLE `competidor` (
  `idcompetidor` int(11) NOT NULL,
  `CI` int(8) NOT NULL,
  `nombre` varchar(20) NOT NULL,
  `apellido` varchar(20) NOT NULL,
  `fecha_nac` date NOT NULL,
  `genero` enum('Masculino','Femenino') NOT NULL,
  `IDDojo` int(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

-- Estructura de tabla para la tabla `compiten`
--

CREATE TABLE `compiten` (
  `idcompetidor` int(11) NOT NULL,
  `idTorneo` int(11) NOT NULL,
  `puesto` int(11) NOT NULL,
  `NumKata` int(11) NOT NULL,
  `Categoria` enum('12/13','14/15','16/17','Mayores') NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

--

-- Estructura de tabla para la tabla `componen`
```




```
--  
  
CREATE TABLE `componen` (  
    `idRonda` int(11) NOT NULL,  
    `idTorneo` int(11) NOT NULL,  
    `idcompetidor` int(11) NOT NULL,  
    `resultado` tinyint(1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `contituye`  
--  
  
CREATE TABLE `contituye` (  
    `NumKata` int(11) NOT NULL,  
    `idcompetidor` int(11) NOT NULL,  
    `idRonda` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `dirige`  
--  
  
CREATE TABLE `dirige` (  
    `CI` int(20) NOT NULL,  
    `IDDojo` int(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;
```



```
-- Estructura de tabla para la tabla `dojo`
--

CREATE TABLE `dojo` (
  `idDojo` int(11) NOT NULL,
  `nombre` varchar(30) NOT NULL,
  `escuela` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

-- Estructura de tabla para la tabla `kata`
--

CREATE TABLE `kata` (
  `NumKata` int(11) NOT NULL,
  `Nombre` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

-- Estructura de tabla para la tabla `llave`
--

CREATE TABLE `llave` (
  `idLlave` int(11) NOT NULL,
  `color` enum('rojo(aka)','azul(ao)') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

-- Estructura de tabla para la tabla `puntaje`
```



```
--  
  
CREATE TABLE `puntaje` (  
  `idpuntaje` int(11) NOT NULL,  
  `puntaje` decimal(4,2) NOT NULL,  
  `IdCompetidor` int(11) DEFAULT NULL,  
  `IdUsuario` int(11) DEFAULT NULL,  
  `IdRonda` int(11) DEFAULT NULL,  
  `NumKata` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `realiza`  
--  
  
CREATE TABLE `realiza` (  
  `NumKata` int(11) NOT NULL,  
  `idcompetidor` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `ronda`  
--  
  
CREATE TABLE `ronda` (  
  `idRonda` int(11) NOT NULL,  
  `fechaInicio` date NOT NULL,  
  `fechaFinilizacion` date NOT NULL,
```



```
`horaInicio` time NOT NULL,  
  
`horaFinilizacion` time NOT NULL,  
  
`Idllave` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `torneo`  
--  
  
CREATE TABLE `torneo` (  
  
  `idTorneo` int(11) NOT NULL,  
  
  `ubicacion` varchar(50) NOT NULL,  
  
  `fecha` date NOT NULL,  
  
  `hora` time NOT NULL,  
  
  `genero` enum('Masculino','Femenino','','') NOT NULL,  
  
  `tipo` enum('Individual','Grupal') NOT NULL,  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;  
  
-- Estructura de tabla para la tabla `usuario`  
--  
  
CREATE TABLE `usuario` (  
  
  `idUsuario` int(11) NOT NULL,  
  
  `nombre`  
enum('Juez1','Juez2','Juez3','Juez4','Juez5','Juez6','Juez7','Administrador','AdministradorBDKarate') NOT NULL,  
  
  `contraseña` varchar(30) NOT NULL,
```



```
`grupoUsuario`
enum('Juez','Administrador','AdministradorBDKarate','Juez1') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

-- Indices de la tabla `asiste`
--
ALTER TABLE `asiste`
  ADD PRIMARY KEY (`idUsuario`),
  ADD KEY `fk_asiste_torneo` (`idTorneo`);

--
-- Indices de la tabla `coach`
--
ALTER TABLE `coach`
  ADD PRIMARY KEY (`CI`);

--
-- Indices de la tabla `competidor`
--
ALTER TABLE `competidor`
  ADD PRIMARY KEY (`idcompetidor`),
  ADD KEY `IDDojo` (`IDDojo`);

--
-- Indices de la tabla `compiten`
--
ALTER TABLE `compiten`
```

```
ADD PRIMARY KEY (`idcompetidor`,`idTorneo`,`NumKata`),

ADD KEY `fk_compiten_torneo` (`idTorneo`),

ADD KEY `fk_compiten_kata`(`NumKata`),

ADD KEY `fk_compiten_competidor` (`IDCompetidor`);


--

-- Indices de la tabla `componen`

--

ALTER TABLE `componen`

  ADD PRIMARY KEY (`idRonda`,`idTorneo`,`idcompetidor`);


--

-- Indices de la tabla `contituye`

--

ALTER TABLE `contituye`

  ADD PRIMARY KEY (`NumKata`,`idcompetidor`,`idRonda`),

  ADD KEY `fk_contituye_ronda` (`idRonda`);


--

-- Indices de la tabla `dirige`

--

ALTER TABLE `dirige`

  ADD KEY `CI` (`CI`),

  ADD KEY `IDDojo` (`IDDojo`);
```



```
--  
  
-- Indices de la tabla `dojo`  
  
--  
ALTER TABLE `dojo`  
  
    ADD PRIMARY KEY (`idDojo`);  
  
--  
  
-- Indices de la tabla `kata`  
  
--  
ALTER TABLE `kata`  
  
    ADD PRIMARY KEY (`NumKata`);  
  
--  
  
-- Indices de la tabla `llave`  
  
--  
ALTER TABLE `llave`  
  
    ADD PRIMARY KEY (`idLlave`);  
  
--  
  
-- Indices de la tabla `puntaje`  
  
--  
ALTER TABLE `puntaje`  
  
    ADD PRIMARY KEY (`idpuntaje`),  
  
    ADD KEY `IdCompetidor` (`IdCompetidor`),  
  
    ADD KEY `IdUsuario` (`IdUsuario`),  
  
    ADD KEY `IdRonda` (`IdRonda`);
```



```
--  
  
-- Indices de la tabla `realiza`  
  
--  
ALTER TABLE `realiza`  
  
  ADD PRIMARY KEY (`NumKata`,`idcompetidor`),  
  
  ADD KEY `fk_realiza_competidor` (`idcompetidor`);  
  
--  
  
-- Indices de la tabla `ronda`  
  
--  
ALTER TABLE `ronda`  
  
  ADD PRIMARY KEY (`idRonda`),  
  
  ADD KEY `Idllave` (`Idllave`);  
  
--  
  
-- Indices de la tabla `torneo`  
  
--  
ALTER TABLE `torneo`  
  
  ADD PRIMARY KEY (`idTorneo`);  
  
--  
  
-- Indices de la tabla `usuario`  
  
--  
ALTER TABLE `usuario`  
  
  ADD PRIMARY KEY (`idUserio`);
```




```
--  
-- AUTO_INCREMENT de las tablas volcadas  
--  
--  
-- AUTO_INCREMENT de la tabla `competidor`  
--  
ALTER TABLE `competidor`  
    MODIFY `idcompetidor` int(11) NOT NULL AUTO_INCREMENT,  
    AUTO_INCREMENT=7;  
--  
-- AUTO_INCREMENT de la tabla `dojo`  
--  
ALTER TABLE `dojo`  
    MODIFY `idDojo` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;  
--  
-- AUTO_INCREMENT de la tabla `kata`  
--  
ALTER TABLE `kata`  
    MODIFY `NumKata` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=103;  
--  
-- AUTO_INCREMENT de la tabla `puntaje`  
--
```



```
ALTER TABLE `puntaje`

  MODIFY `idpuntaje` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--

-- AUTO_INCREMENT de la tabla `ronda`
--

ALTER TABLE `ronda`

  MODIFY `idRonda` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--

-- AUTO_INCREMENT de la tabla `usuario`
--

ALTER TABLE `usuario`

  MODIFY `idUseruario` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--

-- Restricciones para tablas volcadas
--

-- Filtros para la tabla `asiste`
--

ALTER TABLE `asiste`

  ADD CONSTRAINT `fk_asiste_torneo` FOREIGN KEY (`idTorneo`) REFERENCES
`torneo` (`idTorneo`),

  ADD CONSTRAINT `fk_asiste_usuario` FOREIGN KEY (`idUseruario`)
REFERENCES `usuario` (`idUseruario`) ON UPDATE CASCADE;
```

```
--  
-- Filtros para la tabla `competidor`  
--  
ALTER TABLE `competidor`  
    ADD CONSTRAINT `competidor_ibfk_1` FOREIGN KEY (`IDDojo`) REFERENCES  
`dojo` (`idDojo`);  
  
--  
-- Filtros para la tabla `compiten`  
--  
ALTER TABLE `compiten`  
    ADD CONSTRAINT `compiten_ibfk_1` FOREIGN KEY (`idcompetidor`) REFERENCES  
`competidor` (`idcompetidor`),  
    ADD CONSTRAINT `compiten_ibfk_2` FOREIGN KEY (`idTorneo`) REFERENCES  
`torneo` (`idTorneo`);  
  
--  
-- Filtros para la tabla `dirige`  
--  
ALTER TABLE `dirige`  
    ADD CONSTRAINT `dirige_ibfk_1` FOREIGN KEY (`CI`) REFERENCES `coach`  
(`CI`),  
    ADD CONSTRAINT `dirige_ibfk_2` FOREIGN KEY (`IDDojo`) REFERENCES  
`dojo` (`idDojo`);  
  
--  
-- Filtros para la tabla `puntaje`
```

```
--  
  
ALTER TABLE `puntaje`  
  
    ADD CONSTRAINT `puntaje_ibfk_1` FOREIGN KEY (`IdCompetidor`) REFERENCES `competidor` (`idcompetidor`),  
  
    ADD CONSTRAINT `puntaje_ibfk_2` FOREIGN KEY (`IdUsuario`) REFERENCES `usuario` (`idUsuario`),  
  
    ADD CONSTRAINT `puntaje_ibfk_3` FOREIGN KEY (`IdRonda`) REFERENCES `ronda` (`idRonda`);  
  
    ADD CONSTRAINT `puntaje_ibfk_4` FOREIGN KEY (`NumKata`) REFERENCES `kata` (`NumKata`);  
  
COMMIT;
```

Sentencias DCL de permisos a la Base de Datos.

```
1 create user 'Root(DBA)'@'localhost' identified by 'Root';
2 create user 'Administrador'@'localhost' identified by 'Administrador';
3 create user 'AdministradorBDKarate'@'localhost' identified by 'AdministradorBDKarate';
4 create user 'Juez'@'localhost' identified by 'Juez';
5
6 grant select, update, insert, delete on *.* to 'Root(DBA)'@'localhost';

1 grant SELECT, UPDATE, INSERT, DELETE on testingdummies.usuario to 'AdministradorBDKarate'@'localhost';
2 grant SELECT, UPDATE, INSERT, DELETE on testingdummies.ronda to 'Administrador'@'localhost';
3 grant SELECT, UPDATE, INSERT, DELETE on testingdummies.llave to 'Administrador'@'localhost';
4
5
6 grant SELECT, UPDATE, INSERT, DELETE on testingdummies.torneo to 'Administrador'@'localhost';
7 grant INSERT on testingdummies.puntaje to 'Juez'@'localhost';
8 grant SELECT on testingdummies.competidor to 'Juez'@'localhost';

1 grant SELECT, UPDATE, INSERT, DELETE on testingdummies.compiten to 'Administrador'@'localhost';
```



Estudio de los permisos sobre BD, tablas y columnas, **considerando los diferentes roles.**

El usuario “ROOT(DBA)” tendrá todos los permisos. (Select, Insert, Update, Delete)
(En todas las bases de datos y sus respectivas tablas.)

El usuario “Operador de base de datos” tendrá permisos para hacer backup y restore.

El usuario “AdministradorBDKarate” tendrá todos los permisos en la base de datos
“testingdummies”(Select, Insert, Update, Delete) en la tabla “Usuario”.

El usuario “Administrador” tendrá los permisos en la base de datos “testingdummies”
(Select, Insert, Update y Delete) en la tabla “Torneo”, “Compiten”, “Ronda” y
“Llave”.

El usuario “Juez” tendrá los permisos en la base de datos “proyecto” (Insert) en la
tabla “Puntua”. Permiso (Select) en “competidor”.

Respaldar y recuperar datos de la base de datos

Para el proyecto queremos respaldar los datos de nuestra base de datos tanto en cron como con scripts y a mano.

Para eso podemos hacer:

```
“mysqldump -u usuariobd -p nombrebd | ssh usuarioserver@ipserver >  
/directorio/respaldokarate_$(date +"%Y%m%d").sql”
```

Para poder hacer la rutina cron tendremos que poner esto en un script y ejecutar dicho script de la forma que queramos.

En este caso queremos respaldar cada viernes y cada domingo, por lo que después de hacer el script en nuestra rutina cron (a la que se puede entrar con **crontab -e**)

La forma sería: **“0 0 * * 5,0 scriptrespaldo.sh”**

y para recuperar:

```
“mysqldump -u usuariobd -p nombrebd | ssh usuarioserver@ipserver <  
/directorio/respaldokarate_fecha.sql”
```

Consultas SQL.

1- Cantidad de usuarios por tipo (rol). Ordenados por cantidad de mayor a menor.

```
select grupoUsuario, count(*) 'cantidad' from usuario
group by grupoUsuario
order by cantidad;
```

2- Jurados de un evento dado.

```
select usuario.IdUsuario, usuario.nombre, torneo.idtorneo from usuario join torneo join asiste
on usuario.idUsuario=asiste.idUsuario and torneo.idTorneo=asiste.idTorneo
where usuario.grupoUsuario like 'Juez' AND torneo.idTorneo = X;
```

3-Cantidad de competidores por Dojo y por categoría para un evento dado. Ordenados por cantidad de mayor a menor.

```
select count(*) 'CantidadCompetidores', dojo.nombre, compiten.Categoria from torneo JOIN
dojo JOIN competidor join compiten
ON competidor.idcompetidor=compiten.idcompetidor and
torneo.idTorneo=compiten.idTorneo and dojo.idDojo=competidor.IDDojo
where torneo.idTorneo = X
GROUP by compiten.Categoria , dojo.idDojo
order by CantidadCompetidores DESC;
```

4-Ranking anual de competidores por puntos obtenidos para una categoría dada

```
SELECT competidor.nombre, sum(puntaje.puntaje) 'PuntosTotales'FROM compiten JOIN
competidor JOIN puntaje join torneo
ON competidor.idcompetidor=compiten.idcompetidor and
competidor.idcompetidor=puntaje.IdCompetidor and torneo.idTorneo=compiten.idTorneo
WHERE compiten.Categoria like '12/13' AND
year(torneo.fecha)=year(CURRENT_DATE())
GROUP BY competidor.nombre
ORDER by PuntosTotales DESC;
```


5- Ganadores por categoría para un evento dado.

(Ganadores en general:

```
select COUNT(*)'Ganadores', compiten.Categoria, torneo.idTorneo from competidor join
compiten join torneo
on compiten.idcompetidor=competidor.idcompetidor and
torneo.idTorneo=compiten.idTorneo
WHERE compiten.puesto = 1
GROUP by compiten.Categoria;
)
```

(Ganadores con nombre:

```
select compiten.Categoria, competidor.nombre, torneo.idTorneo from competidor join
compiten join torneo
on compiten.idcompetidor=competidor.idcompetidor and
torneo.idTorneo=compiten.idTorneo
WHERE compiten.puesto = 1
GROUP by compiten.Categoria, competidor.nombre;
)
```

6- Puntajes para un encuentro dado. Ordenado cronológicamente.

```
select puntaje.puntaje, ronda.idRonda, torneo.fecha from ronda join puntaje join componen
join torneo
on puntaje.IdRonda=ronda.idRonda and componen.idRonda=ronda.idRonda and
componen.idTorneo=torneo.idTorneo
WHERE ronda.idRonda = X
GROUP BY puntaje.puntaje
ORDER BY torneo.fecha;
```

7-Historial anual de un competidor dado..

a- Cantidad de Katas realizadas

```
select count(*)'KatasRealizadas' from competidor join kata join realiza join compiten join
torneo
on competidor.idcompetidor=realiza.idcompetidor and kata.numkata=realiza.numkata and
competidor.idcompetidor=compiten.idcompetidor and torneo.idtorneo=compiten.idtorneo
where competidor.idcompetidor like X AND year(torneo.fecha)=year(CURRENT_DATE())
group by torneo.fecha;
```

b- Posición por evento

```
SELECT torneo.idTorneo, compiten.puesto, torneo.fecha from competidor join torneo JOIN compiten  
on competidor.idcompetidor=compiten.idcompetidor AND  
torneo.idTorneo=compiten.idTorneo  
where competidor.idcompetidor = 1 and year(torneo.fecha)=year(CURRENT_DATE())  
group by compiten.puesto, torneo.fecha;
```

c- Mejores puntajes por Kata realizada

```
SELECT kata.Nombre AS NombreKata, MAX(puntaje.puntaje) AS MejorPuntaje  
FROM puntaje JOIN componen join kata join ronda join torneo join competidor join compiten join realiza ON competidor.IdCompetidor=compiten.IdCompetidor AND torneo.idTorneo = compiten.IdTorneo and realiza.IdCompetidor = competidor.IdCompetidor and componen.IdTorneo = torneo.idTorneo and puntaje.IdRonda = ronda.idRonda and puntaje.NumKata = kata.NumKata and ronda.IdRonda = componen.IdRonda WHERE puntaje.IdCompetidor = 1 AND YEAR(torneo.fecha) = YEAR(CURRENT_DATE()) GROUP BY kata.Nombre;  
(No sé porque no funciona)
```

```
SELECT k.Nombre AS NombreKata, MAX(p.puntaje) AS MejorPuntaje FROM puntaje p  
JOIN kata k ON p.NumKata = k.NumKata WHERE p.IdCompetidor = X  
GROUP BY k.Nombre;  
(Este funciona, pero no muestra por año actual)
```

8-. Historial de un Dojo en particular:

a-cantidad de Katas realizadas

```
SELECT dojo.nombre, COUNT(*)'CantidadDeKatasRealizados' from dojo join competidor  
join kata join realiza  
on dojo.idDojo=competidor.IDDojo and kata.NumKata=realiza.NumKata and  
competidor.idcompetidor=realiza.idcompetidor  
WHERE dojo.idDojo like X;
```

b-posición por evento y por categoría

```
SELECT  
torneo.idTorneo AS Evento,  
compiten.Categoria AS Categoria,  
compiten.puesto as Puesto
```

```
FROM
  compiten
JOIN competidor ON compiten.idcompetidor = competidor.idcompetidor
JOIN torneo ON compiten.idTorneo = torneo.idTorneo
WHERE competidor.IDDojo = X
      AND YEAR(torneo.fecha) = YEAR(CURRENT_DATE())
GROUP BY torneo.idTorneo, compiten.Categoria
ORDER BY compiten.puesto;
```

c-Mejores puntajes por Kata realizada

```
SELECT c.nombre AS NombreCompetidor, k.Nombre AS NombreKata, MAX(p.puntaje)
AS MejorPuntaje
FROM competidor c
JOIN realiza r ON c.idcompetidor = r.idcompetidor
JOIN kata k ON r.NumKata = k.NumKata
JOIN puntaje p ON c.idcompetidor = p.IdCompetidor AND r.NumKata = p.NumKata
JOIN compiten comp ON c.idcompetidor = comp.idcompetidor
JOIN torneo t ON comp.idTorneo = t.idTorneo
WHERE c.IDDojo = X
      AND YEAR(t.fecha) = YEAR(CURDATE())
GROUP BY c.nombre, k.Nombre
ORDER BY MejorPuntaje DESC;
```

9- Ranking por estilo

```
SELECT dojo.escuela, sum(puntaje.puntaje)'PuntosTotales' FROM dojo join competidor join
puntaje
on dojo.idDojo=competidor.IDDojo and competidor.idcompetidor=puntaje.IdCompetidor
GROUP by dojo.escuela
ORDER by PuntosTotales;
```

10-Cantidad de Dojos por estilo

```
select escuela, COUNT(*) 'CantidadDeDojos' from dojo
GROUP BY escuela;
```

11- Cantidad de cometidores por Dojo y por estilo

```
select dojo.escuela, dojo.nombre, COUNT(*) 'CantidadCompetidores' from dojo join
competidor
```

```
ON competidor.IDDojo=dojo.idDojo  
GROUP BY dojo.escuela, dojo.nombre;
```

12-Record anual por Kata

```
SELECT k.Nombre AS NombreKata, YEAR(t.fecha) AS Año, MAX(p.puntaje) AS  
MejorPuntaje  
FROM kata k  
JOIN compiten c ON k.NumKata = c.idcompetidor  
JOIN torneo t ON c.idTorneo = t.idTorneo  
JOIN puntaje p ON c.idcompetidor = p.IdCompetidor AND k.NumKata = p.NumKata  
WHERE YEAR(t.fecha) = YEAR(CURDATE())  
GROUP BY NombreKata, Año  
ORDER BY NombreKata, Año, MejorPuntaje DESC;
```

(Creo que funciona), preguntarle al profesor para estar seguro.

o este:

```
SELECT k.Nombre AS NombreKata, YEAR(t.fecha) AS Año, MAX(p.puntaje) AS  
MejorPuntaje FROM kata k JOIN realiza r ON k.NumKata = r.NumKata JOIN puntaje p ON  
r.idcompetidor = p.IdCompetidor AND r.NumKata = p.NumKata JOIN ronda ro ON  
p.IdRonda = ro.idRonda JOIN torneo t ON ro.Idllave = t.idTorneo GROUP BY k.Nombre,  
YEAR(t.fecha) ORDER BY k.Nombre, Año, MejorPuntaje DESC;
```

13- Cantidad de veces realizada cada Kata en el año actual. Las Katas que no se han realizado deben figurar con cantidad 0 (cero)

```
select COUNT(*)'CantidadRealizada', kata.Nombre from kata join competidor join realiza
join torneo join compiten
on competidor.idcompetidor=realiza.idcompetidor and kata.NumKata=realiza.NumKata and
competidor.idcompetidor=compiten.idcompetidor and torneo.idTorneo=compiten.idTorneo
WHERE torneo.fecha like current_date
GROUP BY kata.Nombre;
```

(Falta que aparezcan el resto que no se hicieron en 0, preguntarle al profe.)

(Aparecen los katas realizados por competidores que estan anotados en el torneo de la fecha actual)

(Creemos que esto podria ser correcto:

```
SELECT k.Nombre AS NombreKata, IFNULL(COUNT(r.NumKata), 0) AS
CantidadRealizada
FROM competidor c
CROSS JOIN kata k
LEFT JOIN realiza r ON c.idcompetidor = r.idcompetidor AND k.NumKata = r.NumKata
LEFT JOIN compiten co ON c.idcompetidor = co.idcompetidor
LEFT JOIN torneo t ON co.idTorneo = t.idTorneo
WHERE YEAR(t.fecha) = YEAR(CURDATE())
GROUP BY c.Nombre, k.Nombre
ORDER BY `CantidadRealizada` DESC;
```

)

14- Cantidad de Katas realizadas por evento.

```
SELECT torneo.idTorneo, COUNT(*)'CantidadKatas' from torneo join kata join competidor
join compiten join realiza
on torneo.idTorneo=compiten.idTorneo and
competidor.idcompetidor=compiten.idcompetidor and
competidor.idcompetidor=realiza.idcompetidor and kata.NumKata=realiza.NumKata
GROUP BY torneo.idTorneo;
```

15-Puntaje máximo por Kata de un competidor dado. Las Katas que no las ha realizado deben figurar con cantidad 0 (cero).

```
SELECT k.Nombre AS NombreKata, IFNULL(MAX(p.puntaje), 0) AS PuntajeMaximo
FROM kata k
LEFT JOIN puntaje p ON k.NumKata = p.NumKata AND p.IdCompetidor = 1
GROUP BY k.Nombre
ORDER BY `PuntajeMaximo` DESC;
```



Glosario:

CROSS JOIN:

CROSS JOIN no tiene una condición específica; simplemente combina todas las filas de una tabla con todas las filas de la otra.

LEFT JOIN:

Devuelve todas las filas de la tabla de la izquierda (tabla principal) y las filas coincidentes de la tabla de la derecha (tabla secundaria). Si no hay coincidencias en la tabla de la derecha, se devuelven valores NULL.