


Práctica Grupal Base de Datos

[PDF Tarea Grupal BBDD](#)

 App Entornos

[Github](#)

Crea 2 consultas monotabla que se ejecuten regularmente en la aplicación. Por ejemplo, si al conectarnos la app nos muestra las ofertas de los productos “SELECT ofertas FROM productos;”

```
SELECT * FROM juego;
```

```
SELECT * FROM usuario;
```

Consultas de GitHub 2 por persona

```
SELECT * FROM administrador;
```

```
SELECT * FROM comentarios;
```

```
SELECT * FROM logsComentarios;
```

```
SELECT juegos_creados FROM usuario
```

```
SELECT nombre AS 'Usuario con más juegos', apellido1, apellido2 FROM  
usuario ORDER BY juegos_creados DESC LIMIT 1;
```

```
SELECT nombre, apellido1, apellido2 FROM administrador;
```

```
SELECT nombre AS 'Juegos +18' FROM juego WHERE clasificacion_edades  
>= 18;
```

```
SELECT nombre FROM juego ORDER BY valoracion DESC;
```

Igual que en el apartado anterior, crea 2 consultas multitablas que se ejecutarán de manera frecuente en la aplicación (JOIN)

Esta consulta muestra toda la información de los juegos que pertenezcan a un usuario concreto.

```
SELECT juego.*  
FROM juego
```

```
JOIN usuario ON juego.creador = usuario.id_usuario;
```

Esta consulta muestra toda la información de los comentarios de un juego concreto

```
SELECT comentarios.*  
FROM comentarios  
JOIN juego ON juego.nombre = comentarios.juego;
```

Crea 2 inserciones, 2 modificaciones y 2 eliminaciones frecuentes que se llevarán a cabo en la aplicación.

```
INSERT INTO usuario VALUES(7, 'Paco', 'Perez', 'Garcia',  
'12345678A', 'paco.perezg@example.com',0);  
INSERT INTO juego VALUES ('Carrera Runner', 7, 'El maravilloso y  
famoso juego Carrera Runner', 'Carreras', 10, 16);
```

```
UPDATE juego SET clasificacion_edades=16 WHERE categoria =  
"Carreras";  
UPDATE usuario SET correo='maria.gonzalez96@example.com' WHERE  
id_usuario = 3;
```

```
DELETE FROM juego WHERE creador LIKE 2;  
DELETE FROM comentarios WHERE autor LIKE 3;
```

Imaginaros que vienen dos personas nuevas a vuestra empresa, os ayudarán a mejorar aspectos de la aplicación ¿Qué vistas le proporcionaríais teniendo en cuenta las funciones que van a desarrollar (define también qué función desarrollarán)? crea mínimo 2 vistas. [PDF Vistas](#)

Esta consulta muestra los juegos más populares basandose en el número de comentarios:

```
CREATE VIEW juegos_populares AS  
SELECT juego.nombre, COUNT(comentarios.mensaje) AS total_comentarios  
FROM juego  
LEFT JOIN comentarios ON juego.nombre = comentarios.juego  
GROUP BY juego.nombre  
ORDER BY total_comentarios DESC;
```

Esta consulta muestra los todos los usuarios y administradores de la base de datos:

```
CREATE VIEW usuarios_y_administradores AS  
SELECT id_usuario AS id, nombre, apellido1, apellido2, dni, correo,  
'usuario' AS tipo  
FROM usuario
```

UNION

```
SELECT cod_administrador AS id, nombre, apellido1, apellido2, dni,  
correo, 'administrador' AS tipo  
FROM administrador;
```

Define qué usuarios crearías para que puedan acceder al SGBD y qué permisos tendrán.

Crearíamos usuarios administradores y tendrían permisos de lectura de los datos, modificación de los datos y eliminación de los datos.

¿Has detectado alguna consulta que se realiza constantemente? ¿echas de menos algún índice? por ejemplo, si se busca siempre información por ciudad sería recomendable crear un índice en el campo ciudad.

Si, hemos detectado varias consultas que se realizan constantemente. Si, hemos echado de menos algún índice y hemos modificado la base de datos para añadirlos (No había FOREIGN KEYS).

¿Qué triggers pensáis que necesita la base de datos?, crea 2 trigger por tabla.
[PDF Triggers y Excepciones](#)

```
CREATE TRIGGER TR_JUEGOS_01  
AFTER INSERT ON juego  
FOR EACH ROW  
    UPDATE usuario  
    SET juegos_creados = juegos_creados + 1  
    WHERE id_usuario = NEW.creador;
```

```
CREATE TRIGGER TR_JUEGOS_02  
AFTER INSERT ON comentarios  
FOR EACH ROW  
    INSERT INTO logsComentarios (autor_comentario, nombre_juego,  
    fecha_anyadido, mensaje)  
    VALUES (NEW.autor, NEW.juego, NOW(), NEW.mensaje);
```

¿Qué procedimientos almacenados podría necesitar la base de datos?, crea 2 procedimientos almacenados. [PDF Procedimientos almacenados](#)

Obtiene los comentarios que tiene cada juego, seleccionando el nombre y apellidos del usuario y su mensaje

DELIMITER //

```
CREATE PROCEDURE ObtenerComentariosPorJuego(  
    IN nombre_juego VARCHAR(30)  
)  
BEGIN  
    SELECT u.nombre, u.apellido1, u.apellido2, c.mensaje  
    FROM comentarios AS c  
    INNER JOIN usuario AS u ON c.autor = u.id_usuario  
    WHERE c.juego = nombreJuego;  
END //  
DELIMITER ;
```

-- Para ejecutarlo: CALL ObtenerComentariosPorJuego('Carreras
Extremas');

Obtiene los comentarios que tiene cada autor, seleccionando el nombre, categoría y clasificación de edades del juego y su mensaje

DELIMITER //

```
CREATE PROCEDURE ObtenerComentariosPorAutor(  
    IN cod_usuario INT  
)  
BEGIN  
    SELECT j.nombre, j.categoria, j.clasificacion_edades,  
c.mensaje  
    FROM comentarios AS c  
    INNER JOIN juego AS j ON c.juego = j.nombre  
    WHERE c.autor = cod_usuario;  
END //  
DELIMITER ;
```

-- Para ejecutarlo: CALL ObtenerComentariosPorAutor(5);

Base de datos:

```
DROP DATABASE IF EXISTS app_entornos;
```

```
CREATE DATABASE app_entornos;
```

```
USE app_entornos;
```

```
CREATE TABLE usuario(  
    id_usuario INT PRIMARY KEY auto_increment,  
    nombre VARCHAR (30) NOT NULL,  
    apellido1 VARCHAR (30) NOT NULL,  
    apellido2 VARCHAR (30),  
    dni VARCHAR (9) NOT NULL,  
    correo VARCHAR (30) NOT NULL,  
    juegos_creados INT  
);
```

```
CREATE TABLE administrador(  
    cod_administrador INT (5) PRIMARY KEY ,  
    nombre VARCHAR (30) NOT NULL,  
    apellido1 VARCHAR (30) NOT NULL,  
    apellido2 VARCHAR (30),  
    dni VARCHAR (9) NOT NULL,  
    correo VARCHAR (30) NOT NULL  
);
```

```
CREATE TABLE juego(  
    nombre VARCHAR (30) PRIMARY KEY ,  
    creador INT ,  
    descripcion VARCHAR (300) NOT NULL,  
    categoria VARCHAR (30) NOT NULL,  
    valoracion INT (2) ,  
    clasificacion_edades INT (2) NOT NULL,  
    CONSTRAINT creador FOREIGN KEY (creador) REFERENCES  
    usuario(id_usuario) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE comentarios(  
    autor INT,  
    juego VARCHAR (30) NOT NULL,  
    mensaje VARCHAR (30) NOT NULL PRIMARY KEY,
```

```
CONSTRAINT juego FOREIGN KEY (juego) REFERENCES juego(nombre)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT autor FOREIGN KEY (autor) REFERENCES
usuario(id_usuario) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE logsComentarios(
    autor_comentario INT NOT NULL,
    nombre_juego VARCHAR (30) NOT NULL,
    fecha_anyadido DATETIME NOT NULL,
    mensaje VARCHAR (30) NOT NULL

);
```

```
INSERT INTO usuario VALUES(1, 'Pepe', 'García', 'Pérez',
'83363997Z', 'pepe.gaarcia@example.com',3);
INSERT INTO usuario VALUES(2, 'Juan', 'Pérez', NULL, '12345678Z',
'juan.perez@example.com',0);
INSERT INTO usuario VALUES(3, 'María', 'González', 'Fernández',
'87654321X', 'maria.gonzalez@example.com',2);
INSERT INTO usuario VALUES(4, 'Pedro', 'Martínez', 'Sánchez',
'42047996V', 'pedro.martinez@example.com',5);
INSERT INTO usuario VALUES(5, 'Lucía', 'Sanz', 'Gómez', '10810834Y',
'lucia.sanz@example.com',1);
INSERT INTO usuario VALUES(6, 'Javier', 'López', 'Hernández',
'90043060S', 'javier.lopez@example.com',2);
```

```
INSERT INTO administrador VALUES(1, 'Jorge', 'Ruiz', 'García',
'12345678A', 'juan.perez@ejemplo.com');
INSERT INTO administrador VALUES(2, 'Eva', 'Rodríguez', 'Fernández',
'87654321B', 'maria.gonzalez@ejemplo.com');
INSERT INTO administrador VALUES(3, 'Pablo', 'Jimenez', 'Sánchez',
'13579246C', 'pedro.martinez@ejemplo.com');
INSERT INTO administrador VALUES(4, 'Paula', 'Ruiz', 'Gómez',
'86429731D', 'ana.ruiz@ejemplo.com');
INSERT INTO administrador VALUES(5, 'Alberto', 'Hernández',
'Jiménez', '24681357E', 'luis.hernandez@ejemplo.com');
INSERT INTO administrador VALUES(6, 'Sandra', 'Díaz', 'Pérez',
'97531864F', 'sara.diaz@ejemplo.com');
```

```
INSERT INTO juego VALUES('Carreras Extremas', 1, 'Un juego de
carreras de coches', 'Carreras', 8, 18);
INSERT INTO juego VALUES('Aventuras en la Selva', 4, 'Un juego de
aventuras en la selva', 'Aventuras', 7, 16);
INSERT INTO juego VALUES('Estrategia Militar', 3, 'Un juego de
estrategia militar', 'Estrategia', 9, 21);
INSERT INTO juego VALUES('Saltos y Plataformas', 4, 'Un juego de
plataformas con un personaje de dibujos animados', 'Plataformas', 6,
12);
INSERT INTO juego VALUES('Medieval Fantasy', 1, 'Un juego de rol
medieval', 'Rol', 10, 15);
INSERT INTO juego VALUES('Puzzles de Colores', 5, 'Un juego de
puzzles con bloques', 'Puzzles', 5, 10);
```

```
INSERT INTO comentarios VALUES(5, 'Puzzles de Colores', '¡Me encanta
este juego!');
INSERT INTO comentarios VALUES(6, 'Saltos y Plataformas', 'No me
gusta mucho el diseño');
INSERT INTO comentarios VALUES(5, 'Estrategia Militar', '¡Es muy
adictivo!');
INSERT INTO comentarios VALUES(5, 'Aventuras en la Selva', '¡Muy buen
juego!');
INSERT INTO comentarios VALUES(4, 'Medieval Fantasy', 'No puedo dejar
de jugarlo.');
```

```
INSERT INTO comentarios VALUES(1, 'Carreras Extremas', '¡Es el mejor
juego!');
```