

Ejercicios de Evaluación Continua UF2175: Diseño de Bases de Datos Relacionales

Descripción del Contexto:

En un futuro cercano, la humanidad ha comenzado a expandirse bajo los océanos y ha creado varias ciudades submarinas autosuficientes conocidas como "AquaCities". Estas ciudades albergan tanto residentes permanentes como turistas que vienen a experimentar la vida en el fondo del océano. Cada ciudad cuenta con diferentes distritos, estaciones de investigación, hoteles, transportes submarinos y sistemas avanzados de suministro de recursos.

La compañía "DeepBlue Ventures" se encarga de gestionar todas las operaciones de "AquaCities" y necesita implementar un sistema de gestión eficiente para supervisar todas las actividades que se desarrollan en estas ciudades submarinas. Te han contratado para diseñar una base de datos que permita gestionar tanto las operaciones diarias como la infraestructura de las ciudades de manera eficaz.

Escenario:

"AquaCities" es un conglomerado de varias ciudades submarinas situadas en diferentes partes del océano. Cada ciudad tiene sus propios residentes, turistas, transportes submarinos, centros de investigación científica, y sistemas de suministro de alimentos, energía y agua. La base de datos que diseñes debe ser capaz de gestionar todos estos aspectos.

Requerimientos generales:

Gestión de Residentes y Turistas: Cada ciudad submarina tiene residentes permanentes que viven allí y turistas que la visitan temporalmente. Se debe gestionar la información de ambos grupos, incluyendo sus datos personales, sus habitaciones o residencias asignadas, y las actividades que realizan dentro de la ciudad.

Gestión de Distritos y Edificios: Cada ciudad submarina está dividida en distritos (residencial, comercial, investigación, etc.), y cada distrito tiene varios edificios (residencias, laboratorios, hoteles, tiendas). Es necesario gestionar la ubicación y capacidad de cada edificio, así como el personal que trabaja en él y los residentes/turistas asignados.

Gestión de Vehículos Submarinos: Las ciudades están conectadas por vehículos submarinos que transportan tanto personas como mercancías entre ellas. Se debe registrar el estado de cada vehículo, su capacidad, las rutas que siguen, y los viajes que realizan.

Suministro de Recursos: Cada ciudad necesita gestionar eficientemente sus suministros de alimentos, agua, energía y oxígeno. Estos recursos se almacenan en centros de distribución en cada ciudad y deben ser monitoreados para asegurarse de que los niveles sean adecuados. Es necesario rastrear el consumo de recursos por distrito y por edificio.

Estaciones de Investigación: Las estaciones de investigación dentro de "AquaCities" realizan estudios sobre la vida marina, la tecnología de energías renovables y la sostenibilidad del entorno submarino. Cada estación debe gestionar proyectos de investigación, asignar científicos, y registrar los recursos utilizados en cada proyecto.

Gestión del Personal: Cada ciudad cuenta con un equipo de personal que trabaja en diversas áreas, como la administración, investigación, operaciones y mantenimiento. Se

debe registrar su información personal, su rol, el edificio en el que trabajan, y su horario de trabajo.

Tareas a Realizar:

Análisis de Requerimientos Funcionales:

Analiza el escenario y define los requerimientos funcionales que el sistema debe satisfacer. Describe la información que necesita ser almacenada y cómo debe ser organizada para reflejar la estructura de "AquaCities". Identifica qué procesos deben gestionarse en la base de datos y cómo los diferentes elementos (residentes, turistas, distritos, vehículos, recursos, personal) interactúan entre sí.

Diseño del Esquema de la Base de Datos:

Basado en los requerimientos funcionales, diseña un esquema de base de datos relacional que incluya:

Tablas que representen a los residentes, turistas, edificios, distritos, vehículos submarinos, recursos, proyectos de investigación, personal, entre otros.

Claves primarias (PK) para identificar de manera única cada registro en las tablas.

Claves foráneas (FK) para establecer las relaciones entre las tablas, como por ejemplo, la asignación de residentes a sus edificios o de vehículos a sus rutas.

Definición de las relaciones entre las tablas, incluyendo relaciones uno a uno, uno a muchos y muchos a muchos, cuando sea necesario.

Restricciones de integridad para garantizar que, por ejemplo, un residente no pueda estar asignado a más de un edificio o que un vehículo submarino no sea asignado a varias rutas al mismo tiempo.

Explicación del Diseño:

Proporciona una explicación detallada del diseño de la base de datos. Justifica las decisiones de diseño que has tomado, incluyendo la creación de tablas, la definición de claves, restricciones y las relaciones entre las entidades. Explica cómo tu diseño garantiza que todos los aspectos del sistema sean gestionados de manera eficiente.

Explica cómo la base de datos gestiona los aspectos críticos del sistema, como la seguridad de los datos, la eficiencia de las consultas, y la capacidad de manejar grandes volúmenes de datos relacionados con los habitantes, las operaciones y los suministros.

Entregable:

Un documento PDF que incluya los 3 apartados indicados en Tareas a Realizar. También se puede entregar el documento mwb (diseño realizado en my SQL workbench) para proporcionar un mayor detalle.

Instrucciones Adicionales:

El documento debe ser claro y conciso, proporcionando detalles suficientes para entender el diseño de la base de datos y su alineación con los requisitos de "AquaCities".

Se valorará la correcta identificación de los procesos de negocio, la claridad del diagrama de base de datos y la profundidad de la justificación del diseño.

Respuesta:

Análisis de Requerimientos Funcionales

Los requerimientos funcionales incluyen la gestión de residentes y turistas, distritos y edificios, vehículos submarinos, suministro de recursos, estaciones de investigación y personal. La base de datos debe almacenar información personal, asignaciones de habitaciones, actividades, ubicación y capacidad de edificios, estado de vehículos, rutas, consumo de recursos, proyectos de investigación y roles del personal.

Gestión de Personas:

Residentes permanentes con datos personales y asignación de residencia

Turistas con estadías temporales y asignación de alojamiento

Registro de actividades realizadas por ambos grupos

Gestión de Infraestructura:

División de ciudades en distritos (residencial, comercial, investigación)

Catastro de edificios con ubicación, capacidad y función

Asignación de personal a edificios

Transporte Submarino:

Registro de vehículos submarinos con estado y capacidad

Gestión de rutas entre ciudades

Control de viajes realizados

Suministro de Recursos:

Monitoreo de alimentos, agua, energía y oxígeno

Centros de distribución por ciudad

Seguimiento de consumo por distrito/edificio

Investigación Científica

Gestión de estaciones de investigación

Asignación de proyectos y científicos

Control de recursos utilizados en investigación

Gestión de Personal:

Registro de empleados con roles y horarios

Asignación a áreas específicas (administración, mantenimiento, etc.)

Diseño del Esquema de la Base de Datos

El esquema incluirá tablas para residentes, turistas, edificios, distritos, vehículos, recursos, proyectos de investigación y personal. Las claves primarias identificarán registros únicos, y las claves foráneas establecerán relaciones entre tablas. Por ejemplo, un residente se asignará a un edificio, y un vehículo a una ruta. Las relaciones pueden ser uno a uno, uno a muchos o muchos a muchos.

Explicación del Diseño

Estructura General

El diseño sigue los principios de normalización hasta la tercera forma normal (3FN) para evitar redundancias y anomalías en los datos. Las tablas principales representan las entidades fundamentales del sistema (ciudades, distritos, edificios, personas, vehículos, etc.), mientras que las tablas de relación gestionan las interacciones complejas entre ellas.

Relaciones Clave

1. Jerarquía Geográfica: Ciudades → Distritos → Edificios
 - Cada ciudad contiene varios distritos
 - Cada distrito contiene varios edificios
2. Asignación de Personas:
 - Residentes asignados a edificios residenciales
 - Turistas asignados a hoteles (que son un tipo de edificio)
3. Transporte:
 - Vehículos submarinos asociados a viajes específicos
 - Viajes que siguen rutas predefinidas entre ciudades
4. Gestión de Recursos:
 - Inventario por ciudad para cada tipo de recurso
 - Seguimiento de consumo por distrito
5. Investigación:
 - Proyectos asociados a edificios de investigación
 - Científicos (que son residentes) asignados a proyectos
6. Personal:
 - Empleados (residentes) con roles específicos
 - Horarios de trabajo por empleado

Consideraciones de Integridad

- Claves primarias: Cada tabla tiene una clave primaria auto_incremental para la identificación única.
- Claves foráneas: Todas las relaciones están debidamente implementadas con claves foráneas.
- Restricciones CHECK: Se utilizan para validar dominios de valores (tipos de distrito, estado de vehículos, etc.).
- Restricciones de nulidad: Campos obligatorios marcados como NOT NULL.

Optimización de Consultas

- Índices implícitos en claves primarias y foráneas
- Tipos de datos apropiados para cada campo
- Esta estructura evita redundancias, mantiene la integridad referencial y permite escalabilidad, cumpliendo con los principios de normalización hasta 3FN

Código en SQL:

```
DROP SCHEMA IF EXISTS aquacities;
CREATE SCHEMA aquacities;
USE aquacities;
/* Tabla principal para ciudades submarinas*/
CREATE TABLE Ciudades (
    ciudad_id INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    ubicacion_geografica VARCHAR(255),
    fecha_fundacion DATE,
    capacidad_maxima INT
);
/*Tabla para distritos dentro de cada ciudad*/
CREATE TABLE Distritos (
    distrito_id INT PRIMARY KEY,
    ciudad_id INT NOT NULL,
    nombre VARCHAR(100) NOT NULL,
    tipo VARCHAR(50) CHECK (tipo IN ('Residencial', 'Comercial', 'Investigación', 'Industrial',
'Gobierno')),
    capacidad INT,
    FOREIGN KEY (ciudad_id) REFERENCES Ciudades(ciudad_id)
);
/*Tabla para edificios*/
CREATE TABLE Edificios (
    edificio_id INT PRIMARY KEY,
    distrito_id INT NOT NULL,
    nombre VARCHAR(100) NOT NULL,
    tipo VARCHAR(50) CHECK (tipo IN ('Residencial', 'Hotel', 'Laboratorio', 'Tienda', 'Almacén',
'Estación de Energía')),
    capacidad INT,
    niveles INT,
    FOREIGN KEY (distrito_id) REFERENCES Distritos(distrito_id)
);
/*Tabla para residentes permanentes*/
CREATE TABLE Residentes (
    residente_id INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE,
    profesion VARCHAR(100),
    edificio_id INT NOT NULL,
    unidad_habitacional VARCHAR(20),
    fecha_registro DATE,
    FOREIGN KEY (edificio_id) REFERENCES Edificios(edificio_id)
);
/*Tabla para turistas*/
CREATE TABLE Turistas (
    turista_id INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE,
    ciudad_origen VARCHAR(100),
    hotel_id INT NOT NULL,
```

```

    habitacion VARCHAR(20),
    fecha_llegada DATE,
    fecha_salida DATE,
    FOREIGN KEY (hotel_id) REFERENCES Edificios(edificio_id)
);
/*Tabla para vehículos submarinos*/
CREATE TABLE VehiculosSubmarinos (
    vehiculo_id INT PRIMARY KEY,
    modelo VARCHAR(100) NOT NULL,
    capacidad_pasajeros INT,
    capacidad_carga DECIMAL(10,2),
    fecha_adquisicion DATE,
    estado VARCHAR(20) CHECK (estado IN ('Operativo', 'Mantenimiento', 'Fuera de Servicio')),
    ultimo_mantenimiento DATE
);
/*Tabla para rutas entre ciudades*/
CREATE TABLE Rutas (
    ruta_id INT PRIMARY KEY,
    ciudad_origen INT NOT NULL,
    ciudad_destino INT NOT NULL,
    distancia_km DECIMAL(10,2),
    tiempo_estimado_min INT,
    FOREIGN KEY (ciudad_origen) REFERENCES Ciudades(ciudad_id),
    FOREIGN KEY (ciudad_destino) REFERENCES Ciudades(ciudad_id)
);
/*Tabla para viajes realizados*/
CREATE TABLE Viajes (
    viaje_id INT PRIMARY KEY,
    ruta_id INT NOT NULL,
    vehiculo_id INT NOT NULL,
    fecha_salida DATETIME,
    fecha_llegada DATETIME,
    pasajeros INT,
    carga DECIMAL(10,2),
    FOREIGN KEY (ruta_id) REFERENCES Rutas(ruta_id),
    FOREIGN KEY (vehiculo_id) REFERENCES VehiculosSubmarinos(vehiculo_id)
);
/*Tabla para recursos*/
CREATE TABLE Recursos (
    recurso_id INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    tipo VARCHAR(50) CHECK (tipo IN ('Alimento', 'Agua', 'Energía', 'Oxígeno', 'Materiales')),
    unidad_medida VARCHAR(20)
);
/*Tabla para inventario en ciudades*/
CREATE TABLE InventarioCiudades (
    inventario_id INT PRIMARY KEY,
    ciudad_id INT NOT NULL,
    recurso_id INT NOT NULL,
    cantidad_disponible DECIMAL(10,2),
    capacidad_almacenamiento DECIMAL(10,2),
    ultima_actualizacion DATETIME,
    FOREIGN KEY (ciudad_id) REFERENCES Ciudades(ciudad_id),
    FOREIGN KEY (recurso_id) REFERENCES Recursos(recurso_id)

```

```

);
/*Tabla para consumo de recursos*/
CREATE TABLE ConsumoRecursos (
    consumo_id INT PRIMARY KEY,
    distrito_id INT NOT NULL,
    recurso_id INT NOT NULL,
    cantidad_consumida DECIMAL(10,2),
    periodo DATE,
    FOREIGN KEY (distrito_id) REFERENCES Distritos(distrito_id),
    FOREIGN KEY (recurso_id) REFERENCES Recursos(recurso_id)
);
/*Tabla para proyectos de investigación*/
CREATE TABLE ProyectosInvestigacion (
    proyecto_id INT PRIMARY KEY,
    nombre VARCHAR(200) NOT NULL,
    descripcion TEXT,
    edificio_id INT NOT NULL,
    fecha_inicio DATE,
    fecha_fin_estimada DATE,
    presupuesto DECIMAL(15,2),
    estado VARCHAR(20) CHECK (estado IN ('Planificación', 'En Progreso', 'Completado',
'Cancelado')),
    FOREIGN KEY (edificio_id) REFERENCES Edificios(edificio_id)
);
/*Tabla para personal científico*/
CREATE TABLE Cientificos (
    cientifico_id INT PRIMARY KEY,
    residente_id INT NOT NULL,
    especialidad VARCHAR(100),
    nivel_academico VARCHAR(100),
    proyecto_id INT,
    FOREIGN KEY (residente_id) REFERENCES Residentes(residente_id),
    FOREIGN KEY (proyecto_id) REFERENCES ProyectosInvestigacion(proyecto_id)
);
/*Tabla para empleados*/
CREATE TABLE Empleados (
    empleado_id INT PRIMARY KEY,
    residente_id INT NOT NULL,
    rol VARCHAR(100) NOT NULL,
    departamento VARCHAR(100),
    edificio_id INT NOT NULL,
    salario DECIMAL(12,2),
    fecha_contratacion DATE,
    FOREIGN KEY (residente_id) REFERENCES Residentes(residente_id),
    FOREIGN KEY (edificio_id) REFERENCES Edificios(edificio_id)
);
/*Tabla para horarios de trabajo*/
CREATE TABLE Horarios (
    horario_id INT PRIMARY KEY,
    empleado_id INT NOT NULL,
    dia_semana VARCHAR(10) CHECK (dia_semana IN ('Lunes', 'Martes', 'Miércoles', 'Jueves',
'Viernes', 'Sábado', 'Domingo')),
    hora_entrada TIME,
    hora_salida TIME,

```

```

FOREIGN KEY (empleado_id) REFERENCES Empleados(empleado_id)
);
/*Tabla para actividades*/
CREATE TABLE Actividades (
    actividad_id INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    tipo VARCHAR(50) CHECK (tipo IN ('Recreativa', 'Cultural', 'Educativa', 'Científica')),
    edificio_id INT NOT NULL,
    capacidad_maxima INT,
    FOREIGN KEY (edificio_id) REFERENCES Edificios(edificio_id)
);
/*Tabla de relación para participación en actividades*/
CREATE TABLE ParticipacionActividades (
    participacion_id INT PRIMARY KEY,
    actividad_id INT NOT NULL,
    residente_id INT,
    turista_id INT,
    fecha_participacion DATE,
    CHECK ((residente_id IS NOT NULL AND turista_id IS NULL) OR (residente_id IS NULL AND
turista_id IS NOT NULL)),
    FOREIGN KEY (actividad_id) REFERENCES Actividades(actividad_id),
    FOREIGN KEY (residente_id) REFERENCES Residentes(residente_id),
    FOREIGN KEY (turista_id) REFERENCES Turistas(turista_id)
);
/*Insertar ciudades*/
INSERT INTO Ciudades (ciudad_id, nombre, ubicacion_geografica, fecha_fundacion,
capacidad_maxima) VALUES
(1, 'Neptunia', 'Océano Pacífico, Fosa de las Marianas', '2023-06-15', 5000),
(2, 'Atlantida', 'Océano Atlántico, Placa Euroasiática', '2024-01-20', 3500),
(3, 'Poseidonia', 'Mar Mediterráneo', '2023-11-05', 2500);
/*Insertar distritos*/
INSERT INTO Distritos (distrito_id, ciudad_id, nombre, tipo, capacidad) VALUES
(1, 1, 'Central', 'Gobierno', 1000),
(2, 1, 'Residencial Norte', 'Residencial', 2000),
(3, 1, 'Científico Este', 'Investigación', 500),
(4, 2, 'Puerto', 'Comercial', 800),
(5, 2, 'Habitat Alpha', 'Residencial', 1500),
(6, 3, 'Estación Biológica', 'Investigación', 600);
-- Insertar edificios
INSERT INTO Edificios (edificio_id, distrito_id, nombre, tipo, capacidad, niveles) VALUES
(1, 1, 'Torre del Gobierno', 'Estación de Energía', 300, 10),
(2, 2, 'Complejo Residencial A', 'Residencial', 500, 15),
(3, 2, 'Complejo Residencial B', 'Residencial', 500, 15),
(4, 3, 'Laboratorio Marino', 'Laboratorio', 200, 8),
(5, 4, 'Hotel del Puerto', 'Hotel', 300, 12),
(6, 5, 'Habitat Principal', 'Residencial', 700, 20),
(7, 6, 'Centro de Investigación', 'Laboratorio', 400, 10);
SELECT * FROM edificios;
/*Insertar residentes*/
INSERT INTO Residentes (residente_id, nombre, apellido, fecha_nacimiento, profesion, edificio_id,
unidad_habitacional, fecha_registro) VALUES
(1, 'Marta', 'González', '1985-04-12', 'Ingeniera Oceánica', 2, 'A-205', '2023-07-10'),
(2, 'Carlos', 'Suárez', '1990-11-25', 'Biólogo Marino', 2, 'B-312', '2023-08-15'),

```



```

(3, 'Lucía', 'Fernández', '1988-05-30', 'Doctora', 3, 'C-104', '2023-09-20'),
(4, 'David', 'Martínez', '1982-02-18', 'Técnico de Mantenimiento', 6, 'D-408', '2024-02-01');
SELECT * FROM Residentes;
-- Insertar turistas
INSERT INTO Turistas (turista_id, nombre, apellido, fecha_nacimiento, ciudad_origen, hotel_id,
habitacion, fecha_llegada, fecha_salida) VALUES
(1, 'Ana', 'López', '1995-07-22', 'Madrid, España', 5, 'H-302', '2025-03-10', '2025-03-17'),
(2, 'James', 'Wilson', '1987-09-14', 'Nueva York, USA', 5, 'H-305', '2025-03-12', '2025-03-19');
SELECT * FROM Turistas;
/*Insertar vehículos submarinos*/
INSERT INTO VehiculosSubmarinos (vehiculo_id, modelo, capacidad_pasajeros, capacidad_carga,
fecha_adquisicion, estado, ultimo_mantenimiento) VALUES
(1, 'DeepExplorer X', 50, 2000.00, '2023-05-10', 'Operativo', '2025-02-15'),
(2, 'AquaTransit 3000', 100, 5000.00, '2024-01-20', 'Operativo', '2025-03-01'),
(3, 'OceanShuttle II', 30, 1500.00, '2023-11-15', 'Mantenimiento', '2025-01-10');
SELECT * FROM vehiculosSubmarinos;
/*Insertar rutas*/
INSERT INTO Rutas (ruta_id, ciudad_origen, ciudad_destino, distancia_km, tiempo_estimado_min)
VALUES
(1, 1, 2, 1200.00, 180),
(2, 2, 3, 800.00, 120),
(3, 1, 3, 1500.00, 210);
SELECT * FROM Rutas;
/*Insertar viajes*/
INSERT INTO Viajes (viaje_id, ruta_id, vehiculo_id, fecha_salida, fecha_llegada, pasajeros, carga)
VALUES
(1, 1, 1, '2025-03-10 08:00:00', '2025-03-10 11:00:00', 45, 1800.00),
(2, 2, 2, '2025-03-11 10:00:00', '2025-03-11 12:00:00', 85, 4200.00);
SELECT * FROM Viajes;
/*Insertar recursos*/
INSERT INTO Recursos (recurso_id, nombre, tipo, unidad_medida) VALUES
(1, 'Algas cultivadas', 'Alimento', 'kg'),
(2, 'Agua potable', 'Agua', 'litros'),
(3, 'Energía termal', 'Energía', 'MWh'),
(4, 'Oxígeno comprimido', 'Oxígeno', 'm³');
SELECT * FROM Recursos;
/*Insertar inventario*/
INSERT INTO InventarioCiudades (inventario_id, ciudad_id, recurso_id, cantidad_disponible,
capacidad_almacenamiento, ultima_actualizacion) VALUES
(1, 1, 1, 5000.00, 10000.00, '2025-03-15 08:00:00'),
(2, 1, 2, 200000.00, 500000.00, '2025-03-15 08:00:00'),
(3, 2, 1, 3000.00, 8000.00, '2025-03-15 08:00:00'),
(4, 3, 4, 5000.00, 10000.00, '2025-03-15 08:00:00');
SELECT * FROM InventarioCiudades;
/*Insertar consumo de recursos*/
INSERT INTO ConsumoRecursos (consumo_id, distrito_id, recurso_id, cantidad_consumida, periodo)
VALUES
(1, 2, 1, 200.00, '2025-02-01'),
(2, 2, 2, 5000.00, '2025-02-01'),
(3, 5, 1, 150.00, '2025-02-01');
SELECT * FROM ConsumoRecursos;
/*Insertar proyectos de investigación*/
INSERT INTO ProyectosInvestigacion (proyecto_id, nombre, descripcion, edificio_id, fecha_inicio,
fecha_fin_estimada, presupuesto, estado) VALUES

```

```

(1, 'Biominería Marina', 'Extracción sostenible de minerales de fuentes marinas', 4, '2024-01-15',
'2025-12-31', 2500000.00, 'En Progreso'),
(2, 'Cultivos en Alta Presión', 'Desarrollo de cultivos alimenticios en condiciones de alta presión', 7,
'2024-03-10', '2026-06-30', 1800000.00, 'En Progreso');
SELECT * FROM ProyectosInvestigacion;
/*Insertar científicos*/
INSERT INTO Cientificos (cientifico_id, residente_id, especialidad, nivel_academico, proyecto_id)
VALUES
(1, 2, 'Biología Marina', 'Doctorado', 1),
(2, 4, 'Agronomía Submarina', 'Máster', 2);
SELECT * FROM Cientificos;
-- Insertar empleados
INSERT INTO Empleados (empleado_id, residente_id, rol, departamento, edificio_id, salario,
fecha_contratacion) VALUES
(1, 1, 'Ingeniera Jefe', 'Mantenimiento', 1, 6500.00, '2023-07-15'),
(2, 3, 'Directora Médica', 'Salud', 1, 7200.00, '2023-09-25');
SELECT * FROM Empleados;
/*Insertar horarios*/
INSERT INTO Horarios (horario_id, empleado_id, dia_semana, hora_entrada, hora_salida) VALUES
(1, 1, 'Lunes', '08:00:00', '16:00:00'),
(2, 1, 'Martes', '08:00:00', '16:00:00'),
(3, 2, 'Lunes', '09:00:00', '17:00:00'),
(4, 2, 'Miércoles', '09:00:00', '17:00:00');
SELECT * FROM Horarios;
/*Insertar actividades*/
INSERT INTO Actividades (actividad_id, nombre, descripcion, tipo, edificio_id, capacidad_maxima)
VALUES
(1, 'Tour de la Biosfera', 'Visita guiada a los ecosistemas artificiales', 'Educativa', 1, 20),
(2, 'Observación de Tiburones', 'Sesión de observación en cúpula submarina', 'Recreativa', 5, 15);
SELECT * FROM Actividades;
/*Insertar participación en actividades*/
INSERT INTO ParticipacionActividades (participacion_id, actividad_id, residente_id, turista_id,
fecha_participacion) VALUES
(1, 1, NULL, 1, '2025-03-11'),
(2, 2, 1, NULL, '2025-03-12');
SELECT * FROM ParticipacionActividades;
SELECT
    c.ciudad_id, c.nombre AS ciudad, c.ubicacion_geografica,
    d.distrito_id, d.nombre AS distrito, d.tipo AS tipo_distrito,
    e.edificio_id, e.nombre AS edificio, e.tipo AS tipo_edificio, e.capacidad
FROM
    Ciudades c
LEFT JOIN
    Distritos d ON c.ciudad_id = d.ciudad_id
LEFT JOIN
    Edificios e ON d.distrito_id = e.distrito_id
ORDER BY
    c.ciudad_id, d.distrito_id, e.edificio_id

SELECT * FROM Edificios WHERE edificio_id = [el edificio_id que estás intentando insertar en
Residentes];

```

SELECT c.ciudad_id, c.nombre AS ciudad, cubicacion_geografica

Enter a SQL expression to filter results (use Ctrl+Space)

Grid	123 ciudad_id	A.z ciudad	A.z ubicacion_geografica	123 distrito_id	A.z distrito	A.z tipo_distrito	123 edificio_id	A.z edificio	A.z tipo_edificio	123 capacidad
1	1	Neptunia	Océano Pacífico, Fosa de las Marianas	1	Central	Gobierno	1	Torre del Gobierno	Estación de Energía	300
2	1	Neptunia	Océano Pacífico, Fosa de las Marianas	2	Residencial Norte	Residencial	2	Complejo Residencial A	Residencial	500
3	1	Neptunia	Océano Pacífico, Fosa de las Marianas	2	Científico Este	Investigación	3	Laboratorio Marino	Laboratorio	200
4	1	Neptunia	Océano Pacífico, Fosa de las Marianas	3	Puerto	Comercial	4	Hotel del Puerto	Hotel	300
5	2	Atlantida	Océano Atlántico, Placa Euroasiática	4	Habitat Alpha	Residencial	5	Habitat Principal	Residencial	700
6	2	Atlantida	Océano Atlántico, Placa Euroasiática	5	Estación Biológica	Investigación	6	Centro de Investigación	Laboratorio	400
7	3	Poseidonia	Mar Mediterráneo	6			7			