**Exercise 1: Create a "Students" Table**

**Create a table named "Students" with the following columns:**

student\_id (Primary Key)

first\_name

last\_name

birth\_date (Must be before or on the current date)

admission\_date (Must be after birth\_date)

major (Can only be 'Computer Science' or 'Engineering')

Solution:

CREATE TABLE Students (

student\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

birth\_date DATE CHECK (birth\_date <= CURDATE()),

admission\_date DATE CHECK (admission\_date > birth\_date),

major ENUM('Computer Science', 'Engineering')

);

-- Insert data into Students table

INSERT INTO Students (student\_id, first\_name, last\_name, birth\_date, admission\_date, major)

VALUES

(1, 'John', 'Doe', '2000-01-15', '2022-09-01', 'Computer Science'),

(2, 'Jane', 'Smith', '2001-03-20', '2022-08-15', 'Engineering');

**Exercise 2: Create a "Books" Table**

**Create a table named "Books" with the following columns:**

book\_id (Primary Key)

title

author

publication\_year (Must be between 1900 and the current year)

category (Must be one of 'Fiction', 'Non-Fiction', or 'Sci-Fi')

Solution:

CREATE TABLE Books (

book\_id INT PRIMARY KEY,

title VARCHAR(200),

author VARCHAR(100),

publication\_year INT CHECK (publication\_year BETWEEN 1900 AND YEAR(CURDATE())),

category ENUM('Fiction', 'Non-Fiction', 'Sci-Fi')

);

-- Insert data into Books table

INSERT INTO Books (book\_id, title, author, publication\_year, category)

VALUES

(1, 'The Great Gatsby', 'F. Scott Fitzgerald', 1925, 'Fiction'),

(2, 'Cosmos', 'Carl Sagan', 1980, 'Non-Fiction');

**Exercise 3: Create a "Employees" Table**

**Create a table named "Employees" with the following columns:**

employee\_id (Primary Key)

first\_name

last\_name

birth\_date (Must be before or on the current date)

hire\_date (Must be after birth\_date)

department\_id (Foreign Key referencing department\_id from Departments table)

Solution:

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

birth\_date DATE CHECK (birth\_date <= CURDATE()),

hire\_date DATE CHECK (hire\_date > birth\_date),

department\_id INT,

CONSTRAINT fk\_department FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

-- Insert data into Employees table

INSERT INTO Employees (employee\_id, first\_name, last\_name, birth\_date, hire\_date, department\_id)

VALUES

(1, 'Michael', 'Johnson', '1995-02-18', '2022-01-22', 101),

(2, 'Emily', 'Brown', '1990-06-12', '2021-08-10', 102);

**Exercise 4: Create a "Orders" Table**

**Create a table named "Orders" with the following columns:**

order\_id (Primary Key)

customer\_id (Foreign Key referencing customer\_id from Customers table)

order\_date (Must be before or on the current date)

total\_amount (Must be greater than 0)

Solution:

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE CHECK (order\_date <= CURDATE()),

total\_amount DECIMAL(10, 2) CHECK (total\_amount > 0),

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Insert data into Orders table

INSERT INTO Orders (order\_id, customer\_id, order\_date, total\_amount)

VALUES

(1, 101, '2023-08-10', 150.00),

(2, 102, '2023-08-12', 200.00);

**Exercise 5: Create a "Products" Table**

**Create a table named "Products" with the following columns:**

product\_id (Primary Key)

product\_name

unit\_price (Must be greater than 0)

units\_in\_stock (Must be greater than or equal to 0)

supplier\_id (Foreign Key referencing supplier\_id from Suppliers table)

Solution:

CREATE TABLE Products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100),

unit\_price DECIMAL(10, 2) CHECK (unit\_price > 0),

units\_in\_stock INT CHECK (units\_in\_stock >= 0),

supplier\_id INT,

CONSTRAINT fk\_supplier FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

);

-- Insert data into Products table

INSERT INTO Products (product\_id, product\_name, unit\_price, units\_in\_stock, supplier\_id)

VALUES

(1, 'Smartphone', 499.99, 100, 201),

(2, 'T-shirt', 19.99, 500, 202);

**Exercise 6: Create a "Customers" Table**

**Create a table named "Customers" with the following columns:**

customer\_id (Primary Key)

first\_name

last\_name

email (Must be unique)

phone\_number

birth\_date (Must be before or on the current date)

registration\_date (Must be before or on the current date)

city\_id (Foreign Key referencing city\_id from Cities table)

Solution:

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(20),

birth\_date DATE CHECK (birth\_date <= CURDATE()),

registration\_date DATE CHECK (registration\_date <= CURDATE()),

city\_id INT,

CONSTRAINT fk\_city FOREIGN KEY (city\_id) REFERENCES Cities(city\_id)

);

-- Insert data into Customers table

INSERT INTO Customers (customer\_id, first\_name, last\_name, email, phone\_number, birth\_date, registration\_date, city\_id)

VALUES

(101, 'John', 'Doe', 'john@example.com', '123-456-7890', '1990-05-15', '2022-01-10', 301),

(102, 'Jane', 'Smith', 'jane@example.com', '987-654-3210', '1988-10-20', '2021-07-05', 302);

**Exercise 7: Create an "Inventory" Table**

**Create a table named "Inventory" with the following columns:**

item\_id (Primary Key)

item\_name

quantity (Must be greater than or equal to 0)

category\_id (Foreign Key referencing category\_id from Categories table)

Solution:

CREATE TABLE Inventory (

item\_id INT PRIMARY KEY,

item\_name VARCHAR(100),

quantity INT CHECK (quantity >= 0),

category\_id INT,

CONSTRAINT fk\_category FOREIGN KEY (category\_id) REFERENCES Categories(category\_id)

);

-- Insert data into Inventory table

INSERT INTO Inventory (item\_id, item\_name, quantity, category\_id)

VALUES

(1, 'Laptop', 25, 401),

(2, 'Desk Chair', 10, 402);

**Exercise 8: Create a "Employees" Table with Supervisor**

**Create a table named "Employees" with the following columns:**

employee\_id (Primary Key)

first\_name

last\_name

birth\_date (Must be before or on the current date)

hire\_date (Must be after birth\_date)

supervisor\_id (Foreign Key referencing employee\_id from Employees table)

Solution:

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

birth\_date DATE CHECK (birth\_date <= CURDATE()),

hire\_date DATE CHECK (hire\_date > birth\_date),

supervisor\_id INT,

CONSTRAINT fk\_supervisor FOREIGN KEY (supervisor\_id) REFERENCES Employees(employee\_id)

);

-- Insert data into Employees table

INSERT INTO Employees (employee\_id, first\_name, last\_name, birth\_date, hire\_date, supervisor\_id)

VALUES

(201, 'Michael', 'Johnson', '1995-02-18', '2022-01-22', NULL),

(202, 'Emily', 'Brown', '1990-06-12', '2021-08-10', 201);

**Exercise 9: Create a "Orders" Table with Shipment Date**

**Create a table named "Orders" with the following columns:**

order\_id (Primary Key)

customer\_id (Foreign Key referencing customer\_id from Customers table)

order\_date (Must be before or on the current date)

total\_amount (Must be greater than 0)

shipment\_date (Must be after order\_date)

Solution:

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE CHECK (order\_date <= CURDATE()),

total\_amount DECIMAL(10, 2) CHECK (total\_amount > 0),

shipment\_date DATE CHECK (shipment\_date > order\_date),

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Insert data into Orders table

INSERT INTO Orders (order\_id, customer\_id, order\_date, total\_amount, shipment\_date)

VALUES

(501, 101, '2023-08-10', 150.00, '2023-08-15'),

(502, 102, '2023-08-12', 200.00, '2023-08-16');

**Exercise 10: Create a "Feedback" Table**

**Create a table named "Feedback" with the following columns:**

feedback\_id (Primary Key)

customer\_id (Foreign Key referencing customer\_id from Customers table)

feedback\_text

rating (Must be between 1 and 5)

Solution:

CREATE TABLE Feedback (

feedback\_id INT PRIMARY KEY,

customer\_id INT,

feedback\_text TEXT,

rating INT CHECK (rating BETWEEN 1 AND 5),

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Insert data into Feedback table

INSERT INTO Feedback (feedback\_id, customer\_id, feedback\_text, rating)

VALUES

(1001, 101, 'Great service and fast delivery!', 5),

(1002, 102, 'Product quality could be better.', 3);