

## **Business Rules for Curve Electric E-Bikes Management System**

### **1. Customer Information Requirements**

Rule: Customer records must include Customer\_ID, Name, Email, Phone, Address, City, State Code, ZIP Code, and Password as mandatory fields.

Validation: Enforced through NOT NULL constraints on required fields.

### **2. Unique Email ID for Customers**

Rule: Each customer must have a unique and valid email address.

Validation: Enforced using UNIQUE constraint and REGEXP\_LIKE pattern for format validation.

### **3. Valid Phone and ZIP Code Formats**

Rule: Phone number must be exactly 10 digits; ZIP Code must be 5 characters.

Validation: Enforced using REGEXP\_LIKE for phone and LENGTH check for ZIP Code.

### **4. Payment Details Association**

Rule: Each payment must be associated with a valid customer. No pre-dated or post-dated payments are allowed. Payment method is mandatory.

Validation: Enforced using FOREIGN KEY on Customer\_ID and NOT NULL on required fields. Additional checks required via application logic.

### **5. Valid Model Information**

Rule: Each bike model must have both a brand and a model name.

Validation: Enforced using NOT NULL constraints.

### **6. Employee Information Requirements**

Rule: Each employee must have Customer\_ID, Name, Email, Phone, Address, City, State Code, ZIP Code, and Designation.

Validation: Enforced using NOT NULL and CHECK constraints; Email must be unique.

### **7. Gender Validation**

Rule: Gender must be one of: 'Male', 'Female', or 'Transgender'.

Validation: Enforced using a CHECK constraint on the Gender field.

### **8. Dock Identity and Assignment**

Rule: Each dock must have a unique ID, a name, a location, and be assigned to an employee.

Validation: Enforced via PRIMARY KEY, NOT NULL, and FOREIGN KEY to the Employee table.

### **9. Dock Capacity Management**

Rule: The number of available bikes must not exceed the dock's capacity, and neither value can be negative.

Validation: Enforced using a CHECK constraint on Bike\_Available  $\leq$  Bike\_Capacity and  $\geq$  0.

#### **10. Required Bike Attributes**

Rule: Each bike must include a unique ID, a mandatory model ID, and a dock ID. Current location is optional.

Validation: Enforced via NOT NULL and FOREIGN KEY constraints.

#### **11. Valid Rental Status**

Rule: The default bike status is 'N' (available). The status must be either 'Y' (rented) or 'N' (available) and cannot be null.

Validation: Enforced using a CHECK constraint on Rental\_Status.

#### **12. Accessory Requirements**

Rule: Each accessory must have a name and a cost.

Validation: Enforced using NOT NULL and a CHECK constraint to ensure Item\_Cost  $\geq$  0.

#### **13. Accessory-Bike Association**

Rule: Accessories may be linked to bikes.

Validation: Implemented through the Bike\_Accessory table with a composite primary key.

#### **14. Maintenance Record Requirements**

Rule: Each maintenance record must include Date, Repair Cost, Bike ID, and Employee ID. Description is optional. While a bike is under maintenance, it should not be available for rental.

Validation: Enforced through NOT NULL, FOREIGN KEY constraints, and a CHECK constraint on Repair\_Cost  $\geq$  0.

#### **15. Rental Information Requirements**

Rule: Each rental must include Customer ID, Bike ID, Transaction ID, Start Dock ID, End Dock ID, Start and End Date/Time.

Validation: Enforced using NOT NULL and FOREIGN KEY constraints.

#### **16. Rental Time Constraints**

Rule: Rental start time must be today or within the next 10 days. End time must be after the start and no later than 30 days from the start.

Validation: Enforced via BEFORE INSERT/UPDATE trigger: trg\_validate\_rental\_dates.

#### **17. Rental Duration Calculation**

Rule: Rental duration (in hours) must be calculated automatically based on the start and end times.

Validation: Enforced using a trigger (trigger\_set\_rental\_time) and function (CalculateRentalTime).