

Step 1: Analyze the Current Schema

Description: First, list all the tables in the database to understand the current schema structure.

SQL Command:

```
-- List all tables in the database  
SELECT name FROM sqlite_master WHERE type='table';
```

Step 2: Review the Schema of Each Table

Description: Check the schema of each table to identify fields and their data types.

```
-- Get the schema of a table (repeat for each table)  
PRAGMA table_info(inventories);  
PRAGMA table_info(inventory_parts);  
PRAGMA table_info(inventory_minifigs);  
...
```

Step 3: Identify and Remove Redundancies

Description: Review the tables for any redundancies and ensure that the tables are in First Normal Form (1NF).

1NF: Ensure that each column contains only atomic values.

- Ensure no multi-valued attributes exist.
- Each field should contain only one value.

Step 4: Create the New Normalized Schema

Description: Create new tables that adhere to 3NF, ensuring no partial or transitive dependencies.

```
-- Create new normalized tables
```

```
CREATE TABLE new_inventories (  
    id INTEGER PRIMARY KEY,  
    version INTEGER,  
    set_num VARCHAR(20)  
);
```

```
CREATE TABLE new_inventory_parts (  
    inventory_id INTEGER,  
    part_num VARCHAR(20),  
    color_id INTEGER,  
    quantity INTEGER,  
    is_spare BOOLEAN,  
    PRIMARY KEY (inventory_id, part_num, color_id),  
    FOREIGN KEY (inventory_id) REFERENCES new_inventories(id),  
    FOREIGN KEY (part_num) REFERENCES parts(part_num),  
    FOREIGN KEY (color_id) REFERENCES colors(id)  
);
```

```
CREATE TABLE new_inventory_minifigs (  
    inventory_id INTEGER,  
    fig_num VARCHAR(20),  
    quantity INTEGER,  
    PRIMARY KEY (inventory_id, fig_num),  
    FOREIGN KEY (inventory_id) REFERENCES new_inventories(id),
```

```
FOREIGN KEY (fig_num) REFERENCES minifigs(fig_num)
);
...
```

Step 5: Migrate Data to the New Schema

Description: Copy the data from the old tables to the new normalized tables.

-- Migrate data to the new tables

```
INSERT INTO new_inventories (id, version, set_num)
```

```
SELECT id, version, set_num FROM inventories;
```

```
INSERT INTO new_inventory_parts (inventory_id, part_num, color_id, quantity, is_spare)
```

```
SELECT inventory_id, part_num, color_id, quantity, is_spare FROM inventory_parts;
```

```
INSERT INTO new_inventory_minifigs (inventory_id, fig_num, quantity)
```

```
SELECT inventory_id, fig_num, quantity FROM inventory_minifigs;
```

...

Step 6: Verify the Data

Description: Ensure that the data has been correctly migrated by querying the new tables.

-- Verify the data in the new tables

```
SELECT * FROM new_inventories LIMIT 10;
```

```
SELECT * FROM new_inventory_parts LIMIT 10;
```

```
SELECT * FROM new_inventory_minifigs LIMIT 10;
```

...

By following these steps, we can create a normalized schema that adheres to the principles of 3NF, migrate your existing data to the new schema, and verify that the data has been correctly

transferred. This process helps improve data integrity, reduce redundancy, and enhance the overall structure of our LEGO database.