



## sql schema please

```
-- Complete SQL Schema for SKU Encoder/Decoder System
-- Run this in MySQL after: CREATE DATABASE sku_app; USE sku_app;

-- =====
-- 1. LOOKUP TABLES (for encoder dropdowns/options)
-- =====

-- Stone Types (J/A/G/Z)
CREATE TABLE stones (
  id INT AUTO_INCREMENT PRIMARY KEY,
  code CHAR(1) NOT NULL UNIQUE COMMENT 'J=Mined, A=Lab Grown, G=Gemstone, Z=Lab Created',
  label VARCHAR(50) NOT NULL,
  description TEXT,
  active TINYINT(1) DEFAULT 1,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  INDEX idx_code (code)
);

-- Styles (E/P/N/R/B/S)
CREATE TABLE styles (
  id INT AUTO_INCREMENT PRIMARY KEY,
  code CHAR(1) NOT NULL UNIQUE COMMENT 'E=Earring, P=Pendant, N=Necklace, R=Ring, B=Bangle',
  label VARCHAR(50) NOT NULL,
  needs_size TINYINT(1) DEFAULT 0 COMMENT '1 if N/R/B',
  active TINYINT(1) DEFAULT 1,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  INDEX idx_code (code)
);

-- Gemstones (3-char codes)
CREATE TABLE gemstones (
  id INT AUTO_INCREMENT PRIMARY KEY,
  code CHAR(3) NOT NULL UNIQUE,
  name VARCHAR(100) NOT NULL,
  active TINYINT(1) DEFAULT 1,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  INDEX idx_code (code)
);

-- Metals (metal_type + code)
CREATE TABLE metals (
  id INT AUTO_INCREMENT PRIMARY KEY,
  metal_type VARCHAR(10) NOT NULL COMMENT '10KT,14KT,18KT,9KT,SIL,PLT,SPC',
  code CHAR(1) NOT NULL,
  label VARCHAR(100) NOT NULL,
  active TINYINT(1) DEFAULT 1,
```

```

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY uniq_metal (metal_type, code),
    INDEX idx_metal_type (metal_type)
);

-- Sizes (style-specific)
CREATE TABLE sizes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    style_code CHAR(1) NOT NULL COMMENT 'N/R/B only',
    code CHAR(1) NOT NULL,
    label VARCHAR(50) NOT NULL COMMENT '16 inches (A), 4 (A), etc.',
    active TINYINT(1) DEFAULT 1,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY uniq_size (style_code, code),
    INDEX idx_style_code (style_code)
);

-- =====
-- 2. CENTRAL SKUS TABLE (stores all generated SKUs)
-- =====
CREATE TABLE skus (
    id INT AUTO_INCREMENT PRIMARY KEY,

    -- Full SKU (unique, indexed)
    sku CHAR(13) NOT NULL UNIQUE COMMENT 'JEA010AABC123A',

    -- Decomposed components (for decoder)
    stone_code CHAR(1) NOT NULL,
    style_code CHAR(1) NOT NULL,
    gemstone_code CHAR(3) NULL,
    carat_code CHAR(3) NULL COMMENT '010,850,12T',
    metal_type VARCHAR(10) NOT NULL,
    metal_code CHAR(1) NOT NULL,
    unique_id VARCHAR(6) NOT NULL,
    size_code CHAR(1) NULL,

    -- Extra: raw gem carat if gemstone had carat
    gem_carat_raw DECIMAL(5,2) NULL COMMENT '0.50, 1.25 etc.',

    -- Audit
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

    INDEX idx_sku (sku),
    INDEX idx_stone_style (stone_code, style_code),
    INDEX idx_unique_id (unique_id)
);

-- =====
-- 3. SEED DATA (your existing options)
-- =====

-- Stones
INSERT INTO stones (code, label) VALUES
('J', 'Mined (J)'),
('A', 'Lab Grown (A)'),

```

```

('G', 'Gemstone (G)'),
('Z', 'Lab Created (Z)');

-- Styles (with needs_size flag)
INSERT INTO styles (code, label, needs_size) VALUES
('E', 'Earring (E)', 0),
('P', 'Pendant (P)', 0),
('N', 'Necklace (N)', 1),
('R', 'Ring (R)', 1),
('B', 'Bangle/Bracelet (B)', 1),
('S', 'Set (S)', 0);

-- Gemstones (first 10 - add all 100+ from your JSON)
INSERT INTO gemstones (code, name) VALUES
('ABA', 'ABALONE'),
('AGT', 'AGATE'),
('ALE', 'ALEXANDRITE'),
('AME', 'AMETHYST'),
('AQM', 'AQUAMARINE'),
('EMR', 'EMERALD'),
('GAR', 'GARNET'),
('RUB', 'RUBY'),
('SAP', 'SAPPHIRE'),
('TAN', 'TANZANITE');

-- Metals (from your metals.json)
INSERT INTO metals (metal_type, code, label) VALUES
('10KT', 'A', '10KY - YELLOW (A)'),
('10KT', 'B', '10KW - WHITE (B)'),
('10KT', 'C', '10KR - ROSE (C)'),
('10KT', 'D', '10KTT - TWO-TONE (D)'),
('14KT', 'P', '14KY - YELLOW (P)'),
('14KT', 'Q', '14KW - WHITE (Q)'),
('14KT', 'R', '14KR - ROSE (R)'),
('14KT', 'T', '14KTT - TWO-TONE (T)'),
('SIL', 'J', 'RPS - RHODIUM PLATED (J)'),
('PLT', 'U', 'PLAT - Platinum (U)'),
('SPC', 'S', 'SPEC - Special (S)');

-- Sizes (style-specific)
INSERT INTO sizes (style_code, code, label) VALUES
-- Necklace
('N', 'A', '16 inches (A)'),
('N', 'B', '17 inches (B)'),
('N', 'C', '18 inches (C)'),
('N', 'D', '20 inches (D)'),
('N', 'E', '22 inches (E)'),
('N', 'F', '24 inches (F)'),
-- Ring
('R', 'A', '4 (A)'),
('R', 'B', '4.5 (B)'),
('R', 'C', '5 (C)'),
('R', 'D', '5.5 (D)'),
('R', 'E', '6 (E)'),
('R', 'F', '6.5 (F)'),
('R', 'G', '7 (G)'),

```

```

('R', 'H', '7.5 (H)'),
('R', 'I', '8 (I)'),
('R', 'J', '8.5 (J)'),
('R', 'K', '9 (K)'),
('R', 'L', '9.5 (L)'),
('R', 'M', '10 (M)'),
('R', 'N', '10.5 (N)'),
('R', 'O', '11 (O)'),
('R', 'P', '11.5 (P)'),
('R', 'Q', '12 (Q)'),
-- Bangle/Bracelet
('B', 'A', '6 inches (A)'),
('B', 'B', '6.25 inches (B)'),
('B', 'C', '6.5 inches (C)'),
('B', 'D', '7 inches (D)'),
('B', 'E', '7.25 inches (E)'),
('B', 'F', '7.5 inches (F)'),
('B', 'G', '8 inches (G)');

-- =====
-- 4. VIEWS for easy querying (optional)
-- =====

-- View for decoder: full SKU details with labels
CREATE VIEW sku_details AS
SELECT
    s.sku,
    st.label AS stone_label,
    sty.label AS style_label,
    g.name AS gemstone_name,
    s.gem_carat_raw,
    CONCAT(m.metal_type, ' - ', m.label) AS metal_label,
    sz.label AS size_label,
    s.unique_id,
    s.carat_code,
    s.created_at
FROM skus s
LEFT JOIN stones st ON st.code = s.stone_code
LEFT JOIN styles sty ON sty.code = s.style_code
LEFT JOIN gemstones g ON g.code = s.gemstone_code
LEFT JOIN metals m ON m.metal_type = s.metal_type AND m.code = s.metal_code
LEFT JOIN sizes sz ON sz.style_code = s.style_code AND sz.code = s.size_code;

-- =====
-- 5. USAGE EXAMPLES
-- =====

-- Encoder: Check if SKU exists
SELECT sku FROM skus WHERE sku = 'JEA010AABC123A';

-- Encoder: Insert new SKU
INSERT INTO skus (sku, stone_code, style_code, gemstone_code, carat_code, metal_type, metal_label, size_code)
VALUES ('JEA010AABC123A', 'J', 'E', 'AME', '010', '14KT', 'P', 'ABC123', NULL, 0.10);

-- Decoder: Get full details
SELECT * FROM sku_details WHERE sku = 'JEA010AABC123A';

```

```
-- Admin: List active gemstones for dropdown
SELECT code, name FROM gemstones WHERE active = 1 ORDER BY name;

-- Admin: Deactivate a gemstone (UI updates automatically)
UPDATE gemstones SET active = 0 WHERE code = 'ABA';
```

## How to use:

1. **Save as** `schema.sql` → Run: `mysql -u root -p sku_app < schema.sql`
2. **Populate remaining gemstones** from your JSON (add `INSERT INTO gemstones...` statements)
3. **Your backend endpoints** now query these tables instead of JSON
4. **Encoder:** `INSERT` only if `sku` doesn't exist
5. **Decoder:** `SELECT * FROM sku_details WHERE sku = ?`

## Benefits:

- ✓ Update lookup tables → UI updates automatically
- ✓ SKU uniqueness enforced
- ✓ Gem carat preserved for gemstones
- ✓ Decoder gets full decoded info with labels
- ✓ Admin can deactivate options without code changes

Ready for **Node.js backend code** that implements this schema?

✱

1. <https://www.linkedin.com/pulse/designing-robust-database-schema-e-commerce-shirsh-sinha-7orsc>
2. <https://stackoverflow.com/questions/31747273/mysql-database-to-store-product-color-size-and-stock>
3. [https://www.reddit.com/r/SQL/comments/1j3nawn/schema\\_for\\_db\\_of\\_products\\_with\\_different/](https://www.reddit.com/r/SQL/comments/1j3nawn/schema_for_db_of_products_with_different/)
4. <https://www.scribd.com/document/923095410/Guide-To-Design-Database-For-Shopping-Cart-In-MySQL>
5. <https://github.com/janet-dev/jewellery-designers-database>
6. <https://www.facebook.com/groups/WebDevCommunity/posts/2507639409281051/>
7. <https://www.youtube.com/watch?v=RbKEYDtKAJl>
8. <https://www.mysql.com/products/workbench/design/>