```
CREATE DATABASE IF NOT EXISTS question_bank;

USE question_bank;

CREATE TABLE IF NOT EXISTS users(
   user_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   user_name VARCHAR(100) NOT NULL,
   blocking_reason VARCHAR(100),
   email VARCHAR(100) UNIQUE,
   user_status VARCHAR(15),
   last_loggedin_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
   is_admin BOOLEAN NOT NULL DEFAULT 0,
   image_url VARCHAR(150),
   FOREIGN KEY (reason_id)
      REFERENCES blocking_reasons (reason_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   CHECK (user_status IN ('blocked','unblocked'))
);

CREATE TABLE IF NOT EXISTS popularity(
   popularity_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   upvotes INT,
   downvotes INT,
   CHECK (upvotes >= 0 AND downvotes >= 0)
);

CREATE TABLE IF NOT EXISTS question(
   question_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   question_title TEXT NOT NULL,
   question_body TEXT NOT NULL,
   time_posted TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
   user_id INTEGER UNSIGNED,
   popularity_id INTEGER UNSIGNED,
   question_detail_id INT,
   FOREIGN KEY (user_id)
      REFERENCES users (user_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (question_detail_id)
      REFERENCES question_detail (question_detail_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (popularity_id)
```

```
      REFERENCES popularity (popularity_id)
      ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS question_details(
   question_deatil_id INTEGER UNSIGNED,
   closing_reason VARCHAR(50),
   question_status VARCHAR(20) DEFAULT 'open',
   CHECK (question_status IN ('open','closed')),
   CHECK (closing_reason IN (accepted','irrelevant')),
   FOREIGN KEY (reason_id)
      REFERENCES closing_reasons (reason_id)
      ON DELETE CASCADE ON UPDATE CASCADE
);



CREATE TABLE IF NOT EXISTS answer(
   answer_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   question_id INTEGER UNSIGNED,
   answer_body TEXT NOT NULL,
   answer_detail_id INTEGER UNSIGNED,
   time_posted TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
   user_id INTEGER UNSIGNED,
   popularity_id INTEGER UNSIGNED,
   FOREIGN KEY (user_id)
      REFERENCES users (user_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (question_id)
      REFERENCES question (question_id)
      ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS answer_details(
   answer_id INTEGER UNSIGNED,
   answer_status VARCHAR(20),
   CHECK (answer_status IN ('accepted','not accepted')),
);
```

```
CREATE TABLE IF NOT EXISTS tag(
   tag_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   tag_name VARCHAR(20) NOT NULL
);

INSERT INTO tag VALUES(1, 'Java');

CREATE TABLE IF NOT EXISTS question_tag(
   tag_id INTEGER UNSIGNED,
   question_id INTEGER UNSIGNED,
   PRIMARY KEY (tag_id,question_id),
   FOREIGN KEY (tag_id)
      REFERENCES tag (tag_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (question_id)
      REFERENCES question (question_id)
      ON DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE IF NOT EXISTS question_votes(
   answer_vote_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   answer_id INTEGER UNSIGNED NOT NULL,
   status INTEGER UNSIGNED,
   user_id INTEGER UNSIGNED NOT NULL,
   FOREIGN KEY (question_id)
      REFERENCES question (question_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (user_id)
      REFERENCES user (user_id)
      ON DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE IF NOT EXISTS answer_votes(
   answer_vote_id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   answer_id INTEGER UNSIGNED NOT NULL,
   status INTEGER UNSIGNED,
   user_id INTEGER UNSIGNED NOT NULL,
   FOREIGN KEY (answer_id)
      REFERENCES answer (answer_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (user_id)
```

```
    REFERENCES user (user_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```