A5: Relational Schema, validation and schema refinement

This artifact contains the Relational Schema obtained by mapping from the Conceptual Data Model.

1 Relation Schema

Relation schemas are specified in the following compact notation, where UK means UNIQUE KEY and NN means NOT NULL.

R01	admin(id, username NN UK, email NN UK, hashed_pass NN)
R02	answer(id, date NN, message NN, question_id → question NN, user_id → user NN, auction_id → auction NN)
R03	answer_report(id, date NN, message NN, answer_id → answer NN)
R04	auction(id, start_bid NN, curr_bid NN, start_date NN, end_date NN, date NN, type NN, user_id → user NN, product_id → product NN)
R05	auction_report(id, date NN, message NN, auction_id → auction NN)
R06	bid(id, amount NN, date NN, user_id → user NN, auction_id → auction NN)
R07	city(id, name NN UK, country_id → country NN)
R08	country(id, name NN UK)
R09	follow(user_followed_id → user, user_following_id → user, date NN)
R10	image(id, filename NN UK, product_id → product NN)
R11	location(id, address NN, city_id → city NN)
R12	notification(id, message NN, type, user_id → user NN, is_new NN, date NN)
R13	product(id, name NN, description NN, type NN)
R14	question(id, date NN, message NN, title NN, user_id → user NN, auction_id → auction NN)
R15	question_report(id, date NN, message NN, question_id → question NN)
R16	review(id, rating NN, message NN, date NN, bid_id → bid NN)
R17	review_report(id, date NN, message NN, review_id → review NN)
R18	user(id, username NN UK, email NN UK, name NN, short_bio NN, full_bio, hashed_pass NN, phone, register_date NN, profile_pic, location_id → location, rating)
R19	user_report(id, date NN, message NN, user_id → user NN)
R20	watchlist(auction_id → auction, user_id → user, notifications NN, date NN)

Table 1: Relation Schemas.

D01	auction_type	ENUM('Default', 'Dutch', 'Sealed Bid')
D02	notification_type	ENUM('Auction', 'Question', 'Answer', 'Win', 'Warning')
D03	category_type	ENUM('Art', 'Tickets and Trips', 'Dolls and Bears', 'Toys and Hobbies', 'Cars and Vehicles', 'Sports Souvenirs', 'Home and Garden', 'Collectibles', 'Electronics and Computers', 'Movies and DVDs', 'Musical Instruments', 'Jewelry', 'Books', 'Cloths and Accessories', 'Health and Beauty', 'Video Games', 'Sexual Toys')

Table 2: Additional Domains.

2 Functional Dependencies and Normalization

For the relational model presented above, consider the following non-trivial functional dependencies for each table.

Table R01 admin	
Keys: {id}, {username}, {email}	
Functional Dependencies	
DF0101	id → username, email, hashed_pass
DF0102	username → id, email, hashed_pass
DF0103	email → username, id, hashed_pass
NORMAL FORM	BCNF

Table R02 answer	
Keys: {id}	
Functional Dependencies	
DF0201	id → date, message, question_id, user_id, auction_id
NORMAL FORM	BCNF

Table R03 answer_report	
Keys: {id}	
Functional Dependencies	
DF0301	id → date, message, answer_id
NORMAL FORM	BCNF

Table R04 auction	
Keys: {id}	
Functional Dependencies	
DF0401	id → start_bid, curr_bid, start_date, end_date, date, type, user_id, product_id
NORMAL FORM	BCNF

Table R05 auction_report	
Keys: {id}	
Functional Dependencies	
DF0501	id → date, message, auction_id
NORMAL FORM	BCNF

Table R06 bid	
Keys: {id}	
Functional Dependencies	
DF0601	id → amount, date, user_id, auction_id
NORMAL FORM	BCNF

Table R07 city	
Keys: {id}, {name}	
Functional Dependencies	
DF0701	id → name, country_id
DF0702	name →id, country_id
NORMAL FORM	BCNF

Table R08 country
Keys: {id}, {name}

Functional Dependencies	
DF0801	id → name
DF0802	name → id
NORMAL FORM	BCNF

Table R09 follow	
Keys: {user_followed_id, user_following_id}	
Functional Dependencies	
DF0901	user_followed_id, user_following_id → date
NORMAL FORM	BCNF

Table R10 image	
Keys: {id}	
Functional Dependencies	
DF1001	id → filename, product_id
NORMAL FORM	BCNF

Table R11 location	
Keys: {id}	
Functional Dependencies	
DF1101	id → address, city_id
NORMAL FORM	BCNF

Table R12 notification	
Keys: {id}	
Functional Dependencies	
DF1201	id → message, type, user_id, is_new, date
NORMAL FORM	BCNF

Table R13 product	
Keys: {id}	
Functional Dependencies	
DF1301	id → name, description, type
NORMAL FORM	BCNF

Table R14 question	
Keys: {id}	
Functional Dependencies	
DF1401	id → date, message, title, user_id, auction_id
NORMAL FORM	BCNF

Table R15 question_report	
Keys: {id}	
Functional Dependencies	
DF1501	id → date, message, question_id
NORMAL FORM	BCNF

Table R16 review	
Keys: {id}	
Functional Dependencies	
DF1601	id → rating, message, date, bid_id
NORMAL FORM	BCNF

Table R17 review_report
Keys: {id}

Functional Dependencies	
DF1701	id → date, message, review_id
NORMAL FORM	BCNF

Table R18 user	
Keys: {id}, {username}, {email}	
Functional Dependencies	
DF1801	id → username, email, name, short_bio, full_bio, hashed_pass, phone, register_date, profile_pic, location_id, rating
DF1802	username → id, email, name, short_bio, full_bio, hashed_pass, phone, register_date, profile_pic, location_id, rating
DF1803	email → username, id, name, short_bio, full_bio, hashed_pass, phone, register_date, profile_pic, location_id, rating
NORMAL FORM	BCNF

Table R19 user_report	
Keys: {id}	
Functional Dependencies	
DF1901	id → date, message, user_id
NORMAL FORM	BCNF

Table R20 watchlist	
Keys: {auction_id, user_id}	
Functional Dependencies	
DF2001	auction_id, user_id → notifications, date
NORMAL FORM	BCNF

A relational schema R is considered to be in ***Boyce–Codd* normal form (BCNF)** if, for every one of its dependencies X → Y, one of the following conditions holds true:

- X → Y is a trivial functional dependency (i.e., Y is a subset of X)
- X is a superkey for schema R

By the analysis of the functional dependencies, it is confirmed that all the attributes of the left side are super keys. The *id* attribute is a primary key, then it's a super key. The other attributes on the left side of the relations (*username*, *email*, etc) have the **UNIQUE** constraint, then they are also super keys.

The schema is already in the Normal Form of Boyce-Codd and does not need to be normalized.

3 SQL Code

lbaw1662_data.sql

```
--
-- PostgreSQL database
--


--
-- Name: public; Type: SCHEMA; Schema: -; Owner: lbaw1662
```

```
--

CREATE SCHEMA public;

--
-- Name: auction_type; Type: TYPE; Schema: public; Owner: lbaw1662
--

CREATE TYPE auction_type AS ENUM (
    'Default',
    'Dutch',
    'Sealed Bid'
);

--
-- Name: category_type; Type: TYPE; Schema: public; Owner: lbaw1662
--

CREATE TYPE category_type AS ENUM (
    'Art',
    'Tickets and Trips',
    'Dolls and Bears',
    'Toys and Hobbies',
    'Cars and Vehicles',
    'Sports Souvenirs',
    'Home and Garden',
    'Collectibles',
    'Electronics and Computers',
    'Movies and DVDs',
    'Musical Instruments',
    'Jewelry',
    'Books',
    'Cloths and Accessories',
    'Health and Beauty',
    'Video Games',
    'Sexual Toys'
);

--
-- Name: notification_type; Type: TYPE; Schema: public; Owner: lbaw1662
--

CREATE TYPE notification_type AS ENUM (
    'Auction',
    'Question',
    'Answer',
    'Win',
    'Warning'
);

--
```

```
-- Name: admin; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE admin (
    id SERIAL NOT NULL,
    username CHARACTER VARYING(32) NOT NULL,
    email CHARACTER VARYING(64) NOT NULL,
    hashed_pass CHARACTER VARYING(32) NOT NULL,
    CONSTRAINT admin_pkey PRIMARY KEY (id),
    CONSTRAINT admin_email_uindex UNIQUE (email),
    CONSTRAINT admin_username_uindex UNIQUE (username)
);

--
-- Name: answer; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE answer (
    id SERIAL NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    question_id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    auction_id INTEGER NOT NULL,
    CONSTRAINT answer_pkey PRIMARY KEY (id),
    CONSTRAINT answer_auction_fk FOREIGN KEY (auction_id) REFERENCES
auction(id) ON DELETE CASCADE,
    CONSTRAINT answer_question_fk FOREIGN KEY (question_id) REFERENCES
question(id) ON DELETE CASCADE,
    CONSTRAINT answer_user_fk FOREIGN KEY (user_id) REFERENCES USER(id)
ON DELETE SET NULL
);

--
-- Name: answer_report; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE answer_report (
    id SERIAL NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    answer_id INTEGER NOT NULL,
    CONSTRAINT answer_report_pkey PRIMARY KEY (id),
    CONSTRAINT answer_report_fk FOREIGN KEY (answer_id) REFERENCES
answer(id) ON DELETE CASCADE
);

--
```

```
-- Name: auction; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE auction (
    id SERIAL NOT NULL,
    start_bid DOUBLE PRECISION NOT NULL,
    curr_bid DOUBLE PRECISION NOT NULL,
    start_date TIMESTAMP WITHOUT TIME zone NOT NULL,
    end_date TIMESTAMP WITHOUT TIME zone NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    TYPE auction_type NOT NULL,
    user_id INTEGER NOT NULL,
    product_id INTEGER NOT NULL,
    CONSTRAINT auction_pkey PRIMARY KEY (id),
    CONSTRAINT auction_user_fk FOREIGN KEY (user_id) REFERENCES
USER(id) ON DELETE CASCADE,
    CONSTRAINT auction_product_fk FOREIGN KEY (product_id) REFERENCES
product(id) ON DELETE CASCADE,
    CONSTRAINT auction_date_ck CHECK (DATE < start_date AND start_date
< end_date)
);

--
-- Name: auction_report; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE auction_report (
    id SERIAL NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    auction_id INTEGER NOT NULL,
    CONSTRAINT auction_report_pkey PRIMARY KEY (id),
    CONSTRAINT auction_report_fk FOREIGN KEY (auction_id) REFERENCES
auction(id) ON DELETE CASCADE
);

--
-- Name: bid; Type: TABLE; Schema: public; Owner: lbaw1662; Tablespace:
--

CREATE TABLE bid (
    id SERIAL NOT NULL,
    amount DOUBLE PRECISION NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    user_id INTEGER NOT NULL,
    auction_id INTEGER NOT NULL,
    CONSTRAINT bid_pkey PRIMARY KEY (id),
    CONSTRAINT bid_auction_fk FOREIGN KEY (auction_id) REFERENCES
auction(id) ON DELETE CASCADE,
```

```
    CONSTRAINT bid_user_fk FOREIGN KEY (user_id) REFERENCES USER(id) ON
DELETE SET NULL,
    CONSTRAINT bid_amount_ck CHECK (amount > ()::DOUBLE PRECISION)
);

--
-- Name: city; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE city (
    id SERIAL NOT NULL,
    country_id INTEGER NOT NULL,
    name CHARACTER VARYING(32) NOT NULL,
    CONSTRAINT city_pkey PRIMARY KEY (id),
    CONSTRAINT city_fk FOREIGN KEY (country_id) REFERENCES country(id)
ON DELETE CASCADE
);

--
-- Name: country; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE country (
    id SERIAL NOT NULL,
    name CHARACTER VARYING(64) NOT NULL,
    CONSTRAINT country_pkey PRIMARY KEY (id),
    CONSTRAINT country_name_uindex UNIQUE (name)
);

--
-- Name: follow; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE follow (
    user_followed_id INTEGER NOT NULL,
    user_following_id INTEGER NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    CONSTRAINT follow_pk PRIMARY KEY (user_following_id,
user_followed_id),
    CONSTRAINT follow_user_followed_fk FOREIGN KEY (user_followed_id)
REFERENCES USER(id) ON DELETE CASCADE,
    CONSTRAINT follow_user_following_fk FOREIGN KEY (user_following_id)
REFERENCES USER(id) ON DELETE CASCADE
);

--
-- Name: image; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
```

```
--

CREATE TABLE image (
    id SERIAL NOT NULL,
    filename text NOT NULL,
    product_id INTEGER NOT NULL,
    CONSTRAINT image_pkey PRIMARY KEY (id),
    CONSTRAINT image_filename_uindex UNIQUE (filename),
    CONSTRAINT image_fk FOREIGN KEY (product_id) REFERENCES product(id)
ON DELETE CASCADE
);

--
-- Name: location; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE location (
    id SERIAL NOT NULL,
    city_id INTEGER NOT NULL,
    address CHARACTER VARYING(64) NOT NULL,
    CONSTRAINT location_pkey PRIMARY KEY (id),
    CONSTRAINT location_fk FOREIGN KEY (city_id) REFERENCES city(id)
);

--
-- Name: notification; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE notification (
    id SERIAL NOT NULL,
    message CHARACTER VARYING(128) NOT NULL,
    TYPE notification_type NOT NULL,
    user_id INTEGER NOT NULL,
    is_new BOOLEAN NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    CONSTRAINT notification_pkey PRIMARY KEY (id),
    CONSTRAINT notification_fk FOREIGN KEY (user_id) REFERENCES
USER(id) ON DELETE CASCADE
);

--
-- Name: product; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE product (
    id SERIAL NOT NULL,
    TYPE category_type[] NOT NULL,
    name CHARACTER VARYING(64) NOT NULL,
```

```
    description text NOT NULL,
    CONSTRAINT product_pkey PRIMARY KEY (id)
);

--
-- Name: question; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE question (
    id SERIAL NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    title CHARACTER VARYING(64) NOT NULL,
    user_id INTEGER NOT NULL,
    auction_id INTEGER NOT NULL,
    CONSTRAINT question_pkey PRIMARY KEY (id),
    CONSTRAINT question_auction_fk FOREIGN KEY (auction_id) REFERENCES
auction(id) ON DELETE CASCADE,
    CONSTRAINT question_user_fk FOREIGN KEY (user_id) REFERENCES
USER(id) ON DELETE SET NULL
);

--
-- Name: question_report; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE question_report (
    id SERIAL NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    question_id INTEGER NOT NULL,
    CONSTRAINT question_report_pkey PRIMARY KEY (id),
    CONSTRAINT question_report_fk FOREIGN KEY (question_id) REFERENCES
question(id) ON DELETE CASCADE
);

--
-- Name: review; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE review (
    id SERIAL NOT NULL,
    rating INTEGER NOT NULL,
    message text NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    bid_id INTEGER NOT NULL,
    CONSTRAINT review_pkey PRIMARY KEY (id),
    CONSTRAINT review_fk FOREIGN KEY (bid_id) REFERENCES bid(id),
```

```
    CONSTRAINT review_rating_ck CHECK ((rating >= ) AND (rating <= 10))
);

--
-- Name: review_report; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE review_report (
    id SERIAL NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    review_id INTEGER NOT NULL,
    CONSTRAINT review_report_pkey PRIMARY KEY (id),
    CONSTRAINT review_report_fk FOREIGN KEY (review_id) REFERENCES
review(id) ON DELETE CASCADE
);

--
-- Name: user; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE USER (
    id SERIAL NOT NULL,
    username CHARACTER VARYING(64) NOT NULL,
    email CHARACTER VARYING(64) NOT NULL,
    name CHARACTER VARYING(64) NOT NULL,
    short_bio CHARACTER VARYING(255) NOT NULL,
    full_bio text,
    hashed_pass CHARACTER VARYING(64) NOT NULL,
    phone CHARACTER VARYING(20),
    register_date TIMESTAMP WITHOUT TIME zone NOT NULL,
    location_id INTEGER,
    profile_pic CHARACTER VARYING(72),
    rating INTEGER,
    CONSTRAINT user_pkey PRIMARY KEY (id),
    CONSTRAINT user_email_uindex UNIQUE (email),
    CONSTRAINT user_username_uindex UNIQUE (username),
    CONSTRAINT user_rating_ck CHECK ((rating >= ) AND (rating <= 10)),
    CONSTRAINT user_fk FOREIGN KEY (location_id) REFERENCES
location(id) ON DELETE SET NULL
);

--
-- Name: user_report; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE user_report (
    id SERIAL NOT NULL,
```

```
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    message text NOT NULL,
    user_id INTEGER NOT NULL,
    CONSTRAINT user_report_pkey PRIMARY KEY (id),
    CONSTRAINT user_report_fk FOREIGN KEY (user_id) REFERENCES USER(id)
ON DELETE CASCADE
);

--
-- Name: watchlist; Type: TABLE; Schema: public; Owner: lbaw1662;
Tablespace:
--

CREATE TABLE watchlist (
    auction_id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    notifications BOOLEAN NOT NULL,
    DATE TIMESTAMP WITHOUT TIME zone NOT NULL,
    CONSTRAINT watchlist_pk PRIMARY KEY (auction_id, user_id),
    CONSTRAINT watchlist_auction_fk FOREIGN KEY (auction_id) REFERENCES
auction(id) ON DELETE CASCADE,
    CONSTRAINT watchlist_user_fk FOREIGN KEY (user_id) REFERENCES
USER(id) ON DELETE CASCADE
);
```

— *LBAW*

[SeekBid]