A5: Relational Schema, validation and schema refinement

The product consists of an online auction website where you are able to bid on and create auctions.

In this artifact we have the objective of creating a blueprint for what the tables of the database should look like. This should contain information about the attributes, relations, primary and foreign keys etc.

1. Relational Schema

Relation reference	**Relation Compact Notation**
R01	auction(**id**, name NN, description NN, species_name NN, age NN, starting_price NN, buyout_price CK buyout_price > starting_price, current_price CK current_price > starting_price, ending_date CK ending_date >= Today, rating_seller CK rating_seller >= 1, rating_seller <= 5, id_category → category NN, id_main_color → main_color, id_dev_stage → development_stage, id_payment_method → payment_method, id_shipping_method → shipping_method, id_seller → seller NN, id_winner → buyer, id_status -> auction_status NN)
R02	bids(**id**, value NN, maximum CK maximum >= value, id_auction → auction, id_buyer → buyer)
R03	notification(**id**, message NN, type NN, read DF false, id_auction → auction NN, id_buyer → buyer)
R04	user(**id**, name NN, email NN UK, hashed_password NN)
R05	admin(**id** → user)
R06	buyer(**id** → user)
R07	seller(**id** → user, rating CK rating >= 1, rating <= 5)
R08	blocks(**id**, end_date CK end_date > Today, id_admin → admin NN, id_seller → seller NN)
R09	shipping_method(**id**, type NN CK type IN shipping)
R10	ships(**id_seller** → seller NN, **id_shipping_method** → shipping_method NN)
R11	payment_method(**id**, type NN CK type IN payment)
R12	accepts(**id_seller** → seller NN, **id_payment_method** → payment_method NN)
R13	reports(**id**, date NN DF Today, id_buyer → buyer NN, id_seller → seller NN, id_status -> report_status NN)
R14	report_status(**id**, type NN CK type IN report_status_name)
R15	watchlists(**id_auction** → auction NN, **id_buyer** → buyer NN)
R16	skill(**id**, type NN CK type IN skill_name)
R17	features(**id_auction** → auction NN, **id_skill** → skill NN)

Relation reference	Relation Compact Notation
R18	main_color(**id**, type NN CK type IN color)
R19	development_stage(**id**, type NN CK type IN dev_stage)
R20	category(**id**, type NN CK type IN category_name)
R21	auction_status(**id**, type NN CK type IN auction_status_name)
R22	image(**id**, url NN)
R23	profile_photo(**id** → image, id_user → user)
R24	animal_photo(**id** → image, id_auction → auction)

2. Domains

Domain Name	Domain Specification
Today	DATE DEFAULT CURRENT_DATE
category_name	ENUM (Mammals, Insects, Reptiles, Fishes, Birds, Amphibians)
skill_name	ENUM (Climbs, Jumps, Talks, Skates, Olfaction, Moonlight Navigation, Echolocation, Acrobatics)
color	ENUM (Blue, Brown, Black, Yellow, Green, Red, White)
dev_stage	ENUM (Baby, Child, Teen, Adult, Elderly)
shipping	ENUM (Standard Mail, Express Mail, Urgent Mail)
payment	ENUM (Debit Card, PayPal)
report_status_name	ENUM (Pending, Approved, Denied)
auction_status_name	ENUM (Ongoing, Cancelled, Finished)

3. Functional Dependencies and schema validation

TABLE R01	**auction**
Keys	{ id }
Functional Dependencies:	
FD0101	id → { name, description, species_name, age, starting_price, buyout_price, current_price, ending_date, rating_seller, id_category, id_payment_method, id_shipping_method, id_seller, id_winner, id_status }
NORMAL FORM	BCNF

TABLE R02	**bids**
Keys	{ id }
Functional Dependencies:	
FD0201	id → { value, maximum, id_auction, id_buyer }
NORMAL FORM	BCNF

TABLE R03	**notification**
Keys	{ id }
Functional Dependencies:	
FD0301	id → { message, type, read, id_auction, id_buyer }
NORMAL FORM	BCNF

TABLE R04	**user**
Keys	{ id }, { email }
Functional Dependencies:	
FD0401	id → { name, email, hashed_password }
FD0402	email → { name, id, hashed_password }
NORMAL FORM	BCNF

TABLE R05	**admin**
Keys	{ id }
Functional Dependencies:	
NORMAL FORM	BCNF

TABLE R06	**buyer**
Keys	{ id }
Functional Dependencies:	
NORMAL FORM	BCNF

TABLE R07	**seller**
Keys	{ id }
Functional Dependencies:	
FD0701	id → { rating }

TABLE R07	**seller**
NORMAL FORM	BCNF

TABLE R08	**blocks**
Keys	{ id }
Functional Dependencies:	
FD0801	id → { end_date, id_admin, id_seller }
NORMAL FORM	BCNF

TABLE R09	**shipping_method**
Keys	{ id }
Functional Dependencies:	
FD0901	id → { type }
NORMAL FORM	BCNF

TABLE R10	**ships**
Keys	{ id_seller, id_shipping_method }
Functional Dependencies:	
NORMAL FORM	BCNF

TABLE R11	**payment_method**
Keys	{ id }
Functional Dependencies:	
FD1101	id → { type }
NORMAL FORM	BCNF

TABLE R12	**accepts**
Keys	{ id_seller, id_payment_method }
Functional Dependencies:	
NORMAL FORM	BCNF

TABLE R13	**reports**
Keys	{ id }

TABLE R13	**reports**
Functional Dependencies:	
FD1301	id → { id_buyer, id_seller }
NORMAL FORM	BCNF

TABLE R14	**report_status**
Keys	{ id }
Functional Dependencies:	
FD1401	id → { type }
NORMAL FORM	BCNF

TABLE R15	**watchlists**
Keys	{ id_auction, id_buyer }
Functional Dependencies:	
NORMAL FORM	BCNF

TABLE R16	**skill**
Keys	{ id }
Functional Dependencies:	
FD1901	id → { type }
NORMAL FORM	BCNF

TABLE R17	**features**
Keys	{ id_auction, id_skill }
Functional Dependencies:	
NORMAL FORM	BCNF

TABLE R18	**main_color**
Keys	{ id }
Functional Dependencies:	
FD2101	id → { type }
NORMAL FORM	BCNF

TABLE R19	**development_stage**
Keys	{ id }
Functional Dependencies:	
FD2201	id → { type }
NORMAL FORM	BCNF

TABLE R20	**category**
Keys	{ id }
Functional Dependencies:	
FD2301	id → { type }
NORMAL FORM	BCNF

TABLE R21	**auction_status**
Keys	{ id }
Functional Dependencies:	
FD2401	id → { type }
NORMAL FORM	BCNF

TABLE R22	**image**
Keys	{ id }
Functional Dependencies:	
FD2801	id → { url }
NORMAL FORM	BCNF

TABLE R23	**profile_photo**
Keys	{ id }
Functional Dependencies:	
FD2901	id → { id_user }
NORMAL FORM	BCNF

TABLE R24	**animal_photo**
Keys	{ id }
Functional Dependencies:	

TABLE R24	**animal_photo**
FD3001	id → { id_auction }
NORMAL FORM	BCNF

There were no changes necessary to the relational schema so that it would fit the BCNF, since every functional dependency had a left hand side consisting of a key to the table.

4. SQL Code

SQL Script

```
DROP TABLE IF EXISTS watchlists;
DROP TABLE IF EXISTS ships;
DROP TABLE IF EXISTS profile_photo;
DROP TABLE IF EXISTS features;
DROP TABLE IF EXISTS animal_photo;
DROP TABLE IF EXISTS accepts;
DROP TABLE IF EXISTS skill;
DROP TABLE IF EXISTS reports;
DROP TABLE IF EXISTS report_status;
DROP TABLE IF EXISTS "notification";
DROP TABLE IF EXISTS bids;
DROP TABLE IF EXISTS auction;
DROP TABLE IF EXISTS auction_status;
DROP TABLE IF EXISTS main_color;
DROP TABLE IF EXISTS "image";
DROP TABLE IF EXISTS blocks;
DROP TABLE IF EXISTS development_stage;
DROP TABLE IF EXISTS category;
DROP TABLE IF EXISTS shipping_method;
DROP TABLE IF EXISTS payment_method;
DROP TABLE IF EXISTS "admin";
DROP TABLE IF EXISTS seller;
DROP TABLE IF EXISTS buyer;
DROP TABLE IF EXISTS "user";

DROP TYPE IF EXISTS skill_name;
DROP TYPE IF EXISTS category_name;
DROP TYPE IF EXISTS shipping;
DROP TYPE IF EXISTS payment;
DROP TYPE IF EXISTS dev_stage;
DROP TYPE IF EXISTS color;
DROP TYPE IF EXISTS report_status_name;
DROP TYPE IF EXISTS auction_status_name;

-----------------------------------------
-- TYPES
-----------------------------------------
CREATE TYPE shipping AS ENUM ('Standard Mail', 'Express Mail', 'Urgent Mail');
```

```
CREATE TYPE payment AS ENUM ('Debit Card', 'PayPal');
CREATE TYPE skill_name AS ENUM ('Climbs', 'Jumps', 'Talks', 'Skates', 'Olfaction',
'Moonlight Navigation', 'Echolocation', 'Acrobatics');
CREATE TYPE color AS ENUM ('Blue', 'Brown', 'Black', 'Yellow', 'Green', 'Red',
'White');
CREATE TYPE dev_stage AS ENUM ('Baby', 'Child', 'Teen', 'Adult', 'Elderly');
CREATE TYPE category_name AS ENUM ('Mammals', 'Insects', 'Reptiles', 'Fishes',
'Birds', 'Amphibians');
CREATE TYPE report_status_name as ENUM('Pending', 'Approved', 'Denied');
CREATE TYPE auction_status_name as ENUM('Ongoing', 'Finished', 'Cancelled');

-----------------------------------------
-- TABLES
-----------------------------------------
CREATE TABLE "user"
(
    id SERIAL PRIMARY KEY,
    name text NOT NULL,
    email text NOT NULL UNIQUE,
    hashed_password text NOT NULL
);

CREATE TABLE "admin"
(
    id integer NOT NULL PRIMARY KEY REFERENCES "user" (id) ON UPDATE CASCADE ON
DELETE RESTRICT
);

CREATE TABLE buyer
(
    id integer NOT NULL PRIMARY KEY REFERENCES "user" (id) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE seller
(
    id integer NOT NULL PRIMARY KEY REFERENCES "user" (id) ON UPDATE CASCADE ON
DELETE CASCADE,
    rating NUMERIC(3, 2) CHECK (rating >= 1 AND rating <= 5)
);

CREATE TABLE skill
(
    id SERIAL PRIMARY KEY,
    TYPE skill_name NOT NULL
);

CREATE TABLE main_color
(
    id SERIAL PRIMARY KEY,
    TYPE color NOT NULL
);

CREATE TABLE development_stage
```

```
(
    id SERIAL PRIMARY KEY,
    TYPE dev_stage NOT NULL
);

CREATE TABLE category
(
    id SERIAL PRIMARY KEY,
    TYPE category_name NOT NULL
);

CREATE TABLE payment_method
(
    id SERIAL PRIMARY KEY,
    TYPE payment NOT NULL
);

CREATE TABLE shipping_method
(
    id SERIAL PRIMARY KEY,
    TYPE shipping NOT NULL
);

CREATE TABLE auction_status
(
    id integer PRIMARY KEY,
    TYPE auction_status_name NOT NULL
);

CREATE TABLE auction
(
    id SERIAL PRIMARY KEY,
    name text NOT NULL,
    description text NOT NULL,
    species_name text NOT NULL,
    age integer NOT NULL,
    starting_price integer NOT NULL,
    buyout_price integer,
    current_price integer,
    ending_date date NOT NULL,
    rating_seller integer CHECK (rating_seller >= 1 AND rating_seller <= 5)
DEFAULT NULL,
    id_category integer NOT NULL REFERENCES category (id) ON UPDATE CASCADE ON
DELETE RESTRICT,
    id_main_color integer NOT NULL REFERENCES main_color (id) ON UPDATE CASCADE ON
DELETE RESTRICT,
    id_dev_stage integer NOT NULL REFERENCES development_stage (id) ON UPDATE
CASCADE ON DELETE RESTRICT,
    id_payment_method integer REFERENCES payment_method (id) ON UPDATE CASCADE ON
DELETE RESTRICT,
    id_shipping_method integer REFERENCES shipping_method (id) ON UPDATE CASCADE
ON DELETE RESTRICT,
    id_seller integer NOT NULL REFERENCES seller (id) ON UPDATE CASCADE ,
    id_winner integer REFERENCES buyer (id) ON UPDATE CASCADE ,
```

```
    id_status integer NOT NULL REFERENCES auction_status (id) ON UPDATE CASCADE,
    CONSTRAINT "buyout_price_ck" CHECK (buyout_price > starting_price),
    CONSTRAINT "current_price_ck" CHECK (current_price >= starting_price),
    CONSTRAINT "ending_date_ck" CHECK ((ending_date > 'now'::text::date) OR
(id_status = 1 OR id_status = 2))
);

CREATE TABLE bids
(
    id SERIAL PRIMARY KEY,
    value integer NOT NULL,
    maximum integer,
    id_auction integer NOT NULL REFERENCES auction (id) ON UPDATE CASCADE ON
DELETE CASCADE,
    id_buyer integer REFERENCES buyer (id) ON UPDATE CASCADE,
    CONSTRAINT "maximum_ck" CHECK (maximum >= value)
);

CREATE TABLE "notification"
(
    id SERIAL PRIMARY KEY,
    "message" text NOT NULL,
    "read" boolean DEFAULT FALSE,
    id_auction integer NOT NULL REFERENCES auction (id) ON UPDATE CASCADE ON
DELETE CASCADE,
    id_buyer integer REFERENCES buyer (id) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE blocks
(
    id SERIAL PRIMARY KEY,
    end_date date NOT NULL CHECK (end_date > 'now'::text::date),
    id_admin integer NOT NULL REFERENCES "admin" (id) ON UPDATE CASCADE,
    id_seller integer NOT NULL REFERENCES seller (id) ON UPDATE CASCADE ON DELETE
CASCADE
);

CREATE TABLE ships
(
    id_seller integer NOT NULL REFERENCES seller (id) ON UPDATE CASCADE ON DELETE
CASCADE,
    id_shipping_method integer NOT NULL REFERENCES shipping_method (id) ON UPDATE
CASCADE ON DELETE CASCADE,
    PRIMARY Key(id_seller, id_shipping_method)
);

CREATE TABLE accepts
(
    id_seller integer NOT NULL REFERENCES seller (id) ON UPDATE CASCADE ON DELETE
CASCADE,
    id_payment_method integer NOT NULL REFERENCES payment_method (id) ON UPDATE
CASCADE ON DELETE CASCADE,
    PRIMARY Key(id_seller, id_payment_method)
);
```

```
CREATE TABLE report_status
(
    id integer PRIMARY KEY,
    TYPE report_status_name NOT NULL
);

CREATE TABLE reports
(
    id SERIAL PRIMARY KEY,
    "date" date NOT NULL DEFAULT 'now'::text::date,
    id_buyer integer NOT NULL REFERENCES buyer (id) ON UPDATE CASCADE,
    id_seller integer NOT NULL REFERENCES seller (id) ON UPDATE CASCADE ON DELETE
CASCADE,
    id_status integer NOT NULL REFERENCES report_status ON UPDATE CASCADE
);

CREATE TABLE watchlists
(
    id_auction integer NOT NULL REFERENCES auction (id) ON UPDATE CASCADE ON
DELETE CASCADE,
    id_buyer integer NOT NULL REFERENCES buyer (id) ON UPDATE CASCADE ON DELETE
CASCADE,
    PRIMARY Key(id_auction, id_buyer)
);

CREATE TABLE features
(
    id_auction integer NOT NULL REFERENCES auction (id) ON UPDATE CASCADE ON
DELETE CASCADE,
    id_skill integer NOT NULL REFERENCES skill (id) ON UPDATE CASCADE ON DELETE
CASCADE,
    PRIMARY Key(id_auction, id_skill)
);

CREATE TABLE "image"
(
    id SERIAL PRIMARY KEY,
    url text NOT NULL
);

CREATE TABLE profile_photo
(
    id integer NOT NULL PRIMARY KEY REFERENCES "image" (id) ON UPDATE CASCADE,
    id_user integer NOT NULL REFERENCES "user" (id) ON UPDATE CASCADE ON DELETE
CASCADE
);

CREATE TABLE animal_photo
(
    id integer NOT NULL PRIMARY KEY REFERENCES "image" (id) ON UPDATE CASCADE ,
    id_auction integer NOT NULL REFERENCES auction (id) ON UPDATE CASCADE ON
DELETE CASCADE
);
```

Revision history

1. Removed IS-A relations, added current_price and id_status parameters to the auction relation, added id_status to the reports relation. This last two changes made the id_auction and id_reports attributes in the auction_status and report_status tables disappear too.

GROUP2053, 12/04/2020

- Carlos Miguel Sousa Vieira, up201606868@fe.up.pt (Editor)
- João Alberto Preto Rodrigues Praça, up201704748@fe.up.pt
- Lucas Tomás Martins Ribeiro, up201705227@fe.up.pt
- Silvia Jorge Moreira da Rocha, up201704684@fe.up.pt