

# Relational Schema, Validation and Schema Refinement - Artifact 5

This artifact contains the relational schema obtained by mapping from the conceptual data model already provided in the artifact 4. The relational schema includes the relation schemas, attributes, domains, primary keys, foreign keys and other integrity rules: UNIQUE, DEFAULT, NOT NULL, CHECK.

Glossary:

- PK - PRIMARY KEY
- FK - FOREIGN KEY
- UK - UNIQUE
- NN - NOT NULL
- CK - CHECK
- DF - DEFAULT

## Entity Tables

<b>User(userID PK, userUsername NN UK, userPassword NN, userName NN, userEmail UK NN, userPhone DF, userIsBanned NN DF, userMunicipalityID FK DF)</b>	
<b>Candidate keys:</b>	userID, userUsername, userEmail
<b>Primary key:</b>	userID
<b>Foreign keys:</b>	userMunicipality → Municipality(municipalityID)
<b>Integrity rule: UNIQUE</b>	userUsername, userEmail
<b>Integrity rule: NOT NULL</b>	userUsername, userPassword, userName, userEmail, userIsBanned
<b>Integrity rule: CHECK</b>	
<b>Integrity rule: DEFAULT</b>	userPhone NULL, userIsBanned false, userMunicipalityID NULL
<b>Domain:</b>	userID: bigint userUsername: varchar(64) userPassword: varchar(256) userName: varchar(64) userEmail: varchar(64) userPhone: varchar(20) userIsBanned: boolean userMunicipalityID: bigint
<b>Municipality(municipalityID PK, municipalityName UK NN)</b>	
<b>Candidate keys:</b>	municipalityID, municipalityName
<b>Primary key:</b>	municipalityID
<b>Foreign keys:</b>	
<b>Integrity rule: UNIQUE</b>	municipalityName
<b>Integrity rule: NOT NULL</b>	municipalityName
<b>Integrity rule: CHECK</b>	
<b>Integrity rule: DEFAULT</b>	

<b>Domain:</b>	municipalityID: serial municipalityName: varchar(64)
<b>Admin(adminID PK, adminUsername UK NN, adminPassword NN, adminAccepted NN)</b>	
<b>Candidate keys:</b>	adminID, adminUsername
<b>Primary key:</b>	adminID
<b>Foreign keys:</b>	
<b>Integrity rule: UNIQUE</b>	adminUsername
<b>Integrity rule: NOT NULL</b>	adminUsername, adminPassword, adminAccepted
<b>Integrity rule: CHECK</b>	
<b>Integrity rule: DEFAULT</b>	
<b>Domain:</b>	adminID: serial adminUsername: varchar(64) adminPassword: varchar(256) adminAccepted: boolean
<b>SportsComplex(complexID PK, complexName NN, complexLocation NN, complexOpeningHour NN CK, complexClosingHour NN CK, complexOpenOnWeekends NN, complexEmail NN, complexPhone NN, complexPaypal NN, complexDescription DF, complexMunicipalityID FK DF, complexInactive NN)</b>	
<b>Candidate keys:</b>	complexID
<b>Primary key:</b>	complexID
<b>Foreign keys:</b>	complexMunicipalityID → Municipality(municipalityID)
<b>Integrity rule: UNIQUE</b>	
<b>Integrity rule: NOT NULL</b>	complexName, complexLocation, complexOpeningHour, complexClosingHour, complexOpenOnWeekends, complexEmail, complexPhone, complexPaypal, complexMunicipalityID, complexInactive
<b>Integrity rule: CHECK</b>	complexClosingHour > complexOpeningHour
<b>Integrity rule: DEFAULT</b>	complexDescription NULL, complexInactive false
<b>Domain:</b>	complexID: serial complexName: varchar(64) complexLocation: varchar(256) complexOpeningHour: time complexClosingHour: time complexOpenOnWeekends: boolean complexEmail: varchar(64) complexPhone : varchar(20) complexPaypal: varchar(64) complexDescription: text complexMunicipalityID: bigint complexInactive: boolean
<b>Equipment(equipmentID PK, equipmentName NN, equipmentQuantity NN CK, equipmentDetails DF, equipmentQuantityUnavailable NN CK, equipmentPrice NN CK, equipmentComplexID NN FK, equipmentInactive NN)</b>	
<b>Candidate keys:</b>	equipmentID
<b>Primary key:</b>	equipmentID
<b>Foreign keys:</b>	equipmentComplexID → SportsComplex(complexID)
<b>Integrity rule: UNIQUE</b>	
<b>Integrity rule: NOT NULL</b>	equipmentName, equipmentQuantity, equipmentQuantityUnavailable, equipmentPrice, equipmentComplexID, equipmentInactive

Integrity rule: CHECK	equipmentQuantityUnavailable ← equipmentQuantity equipmentPrice > 0	
Integrity rule: DEFAULT	equipmentDetails NULL, equipmentInactive false	
Domain:	equipmentID: bigserial equipmentName: varchar(64) equipmentQuantity: smallint equipmentDetails: varchar(256) equipmentQuantityUnavailable: smallint equipmentPrice: money equipmentComplexID: bigint equipmentInactive: boolean	
Space(spaceID PK, spacelsCovered NN, spacelsAvailable NN, spacePrice NN CK, spaceSurfaceType NN, spaceComplexID FK NN)		
Candidate keys:	spaceID	
Primary key:	spaceID	
Foreign keys:	spaceComplexID → SportsComplex(complexID)	
Integrity rule: UNIQUE		
Integrity rule: NOT NULL	spacelsCovered, spacelsAvailable, spacePrice, spaceSurfaceType, spaceComplexID	
Integrity rule: CHECK	spacePrice > 0	
Integrity rule: DEFAULT		
Domain:	spaceID: bigserial spacelsCovered: boolean spacelsAvailable: boolean spacePrice: money spaceSurfaceType: surfaceType spaceComplexID: bigint	
Sport(sportID PK, sportName UK NN)		
Candidate keys:	sportID, sportName	
Primary key:	sportID	
Foreign keys:		
Integrity rule: UNIQUE	sportName	
Integrity rule: NOT NULL	sportName	
Integrity rule: CHECK		
Integrity rule: DEFAULT		
Domain:	sportID: serial sportName: varchar(64)	
Issue(issueID PK, issueSubject NN, issueDescription NN, issueForAdmin NN CK, issueForManager NN CK, issueCategory NN, issueRentalID FK NN, issueResolved NN, issueComplexID FK, issueUserID FK)		
Candidate keys:	issueID	
Primary key:	issueID	
Foreign keys:	issueRentalID → Rental(rentalID)	
Integrity rule: UNIQUE		
Integrity rule: NOT NULL	issueSubject, issueDescription, issueForAdmin, issueForManager, issueCategory, issueRentalID, issueResolved	
Integrity rule: CHECK	issueForAdmin    issueForManager	
Integrity rule: DEFAULT		

<b>Domain:</b>	issueID: bigserial issueSubject: text issueDescription: text issueForAdmin: boolean issueForManager: boolean issueCategory: issueCategory issueRentalID: bigint issueResolved: boolean issueComplexID: bigint	issueUserID: bigint
<b>Rental(rentalID PK, rentalRating CK, rentalStartTime NN, rentalDurationInMinutes NN CK, rentalDate NN, rentalState NN, rentalUserID FK NN, rentalSpaceID FK NN)</b>		
<b>Candidate keys:</b>	rentalID	
<b>Primary key:</b>	rentalID	
<b>Foreign keys:</b>	rentalUserID → User(userID) rentalSpaceID → Space(spaceID)	
<b>Integrity rule: UNIQUE</b>		
<b>Integrity rule: NOT NULL</b>	rentalStartTime, rentalDurationInMinutes, rentalDate, rentalState, rentalUserID, rentalSpaceID, rentalUserID	
<b>Integrity rule: CHECK</b>	rentalDurationInMinutes > 0 rentalRating >= 1 AND rentalRating <= 5	
<b>Integrity rule: DEFAULT</b>	rentalRating NULL, rentalState 'RESERVED'	
<b>Domain:</b>	rentalID: bigserial rentalRating: smallint rentalStartTime: time rentalDuration: interval rentalDate: date rentalState: rentalState rentalUserID: bigint rentalSpaceID: bigint rentalUserID: bigint	
<b>issueCategory</b>	ENUM ('Refunds', 'Complex Issues', 'Last Minute Cancelation', 'Other')	
<b>rentalState</b>	ENUM ('Reserved', 'Reserved By Manager', 'Canceled by User', 'Canceled by Manager', 'Canceled by Admin', 'Waiting User Feedback', 'Suspended By Admin', 'Concluded', 'Disabled by Admin')	
<b>surfaceType</b>	ENUM ('Synthetic Grass', 'Grass', 'Dirt', 'Floating Wooden Floor', 'Hardwood', 'Resilient', 'Rubber', 'Linoleum', 'Vinyl', 'Other')	

## Relation Tables

<b>Manager(managerID → User, managerComplexID → SportsComplex)</b>	
<b>Candidate keys:</b>	(managerID, managerComplexID)
<b>Primary key:</b>	(managerID, managerComplexID)
<b>Foreign keys:</b>	managerID → User(userID) managerComplexID → SportsComplex(complexID)
<b>Integrity rule: UNIQUE</b>	
<b>Integrity rule: NOT NULL</b>	
<b>Integrity rule: CHECK</b>	
<b>Integrity rule: DEFAULT</b>	

<b>Manager(managerID → User, managerComplexID → SportsComplex)</b>	
<b>Domain:</b>	managerID: bigint managerComplexID: integer
<b>SpaceSports(spaceID → Space, sportID → Sport)</b>	
<b>Candidate keys:</b>	(spaceID, sportID)
<b>Primary key:</b>	(spaceID, sportID)
<b>Foreign keys:</b>	spaceID → Space(spaceID) sportID → Sport(sportID)
<b>Integrity rule: UNIQUE</b>	
<b>Integrity rule: NOT NULL</b>	
<b>Integrity rule: CHECK</b>	
<b>Integrity rule: DEFAULT</b>	
<b>Domain:</b>	spaceID: bigint sportID: bigint
<b>RentalEquipment(rentalID → Rental, equipmentID → Equipment, quantity NN)</b>	
<b>Candidate keys:</b>	(rentalID, equipmentID)
<b>Primary key:</b>	(rentalID, equipmentID)
<b>Foreign keys:</b>	rentalID → Rental(rentalID) equipmentID → Equipment(equipmentID)
<b>Integrity rule: UNIQUE</b>	
<b>Integrity rule: NOT NULL</b>	quantity
<b>Integrity rule: CHECK</b>	quantity > 0
<b>Integrity rule: DEFAULT</b>	
<b>Domain:</b>	rentalID: bigint equipmentID: bigint quantity: smallint
<b>EquipmentSport(equipmentID → Equipment, sportID → Sport)</b>	
<b>Candidate keys:</b>	(equipmentID, sportName)
<b>Primary key:</b>	(equipmentID, sportName)
<b>Foreign keys:</b>	equipmentID → Equipment(equipmentID) sportName → Sport(sportID)
<b>Integrity rule: UNIQUE</b>	
<b>Integrity rule: NOT NULL</b>	
<b>Integrity rule: CHECK</b>	
<b>Integrity rule: DEFAULT</b>	
<b>Domain:</b>	equipmentID: bigint sportID: bigint

## Functional Dependencies and Normalization

To validate the Relational Schema obtained from the Conceptual Model, all functional dependencies are identified and the normalization of all relation schemas is accomplished.

A functional dependency is a set of constraints between two sets of attributes from the database. It says that if two tuples have the same values for attributes A1, A2, ..., An then those two tuples must have to have the same values for attributes B1, B2, ..., Bn. It is represented by the arrow sign (→) so that  $X \rightarrow Y$  means that X functionally determines Y. The left-hand side attributes determine the values

of the attributes on the right-hand side.

Normalization is a method to remove anomalies in the database and make it consistent.

The Boyce-Codd Normal Form (BCNF) is used in database normalization and states that for any non-trivial functional dependency,  $X \rightarrow Y$ ,  $X$  must be a super-key.

The following tables demonstrate that the relational schema is indeed in the Boyce-Codd Normal Form (BCNF) so all the redundancy based on functional dependency is removed.

<b>User</b>	<u>userID</u> → userUsername, userPassword, userName, userEmail, userPhone, userIsBanned, userMunicipalityID
<b>Admin</b>	<u>adminID</u> → adminUsername, adminPassword
<b>SportsComplex</b>	<u>complexID</u> → complexName, complexLocation, complexOpeningHour, complexClosingHour, complexOpenOnWeekends, complexEmail, complexPhone, complexPaypal, complexDescription
<b>Space</b>	<u>spaceID</u> → spaceIsCovered, spaceIsAvailable, spacePrice, spaceSurfaceType, spaceComplexID
<b>Equipment</b>	<u>equipmentID</u> → equipmentName, equipmentQuantity, equipmentDetails, equipmentQuantityUnavailable, equipmentPrice, equipmentComplexID
<b>Rental</b>	<u>rentalID</u> → rentalRating, rentalStartTime, rentalDurationInMinutes, rentalDate, rentalState, rentalUserID, rentalSpaceID
<b>Issue</b>	<u>issueID</u> → issueSubject, issueDescription, issueForAdmin, issueForManager, issueCategory, issueRentalID

## SQL Code

[db1653.sql](#)

```
--
-- PostgreSQL database dump
--

SET statement_timeout = ;
SET lock_timeout = ;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;

SET search_path = public, pg_catalog;

--
-- Name: issueCategory; Type: TYPE; Schema: public; Owner: lbaw1653
--

CREATE TYPE "issueCategory" AS ENUM (
    'REFUNDS',
    'COMPLEXISSUES',
    'LASTMINUTECANCELLATION',
    'OTHER'
);
```

```
ALTER TYPE "issueCategory" OWNER TO lbaw1653;

--
-- Name: rentalState; Type: TYPE; Schema: public; Owner: lbaw1653
--

CREATE TYPE "rentalState" AS ENUM (
    'RESERVED',
    'RESERVEDBYMANAGER',
    'CANCELEDBYUSER',
    'CANCELEDBYMANAGER',
    'WAITINGUSERFEEDBACK',
    'SUSPENDEDADMIN',
    'CONCLUDED',
    'CANCELEDBYADMIN'
);

ALTER TYPE "rentalState" OWNER TO lbaw1653;

--
-- Name: surface; Type: TYPE; Schema: public; Owner: lbaw1653
--

CREATE TYPE surface AS ENUM (
    'Synthetic Grass',
    'Grass',
    'Dirt',
    'Floating Wooden Floor',
    'Hardwood',
    'Resilient',
    'Rubber',
    'Linoleum',
    'Vinyl',
    'Other'
);

ALTER TYPE surface OWNER TO lbaw1653;

--
-- Name: surfaceType; Type: TYPE; Schema: public; Owner: lbaw1653
--

CREATE TYPE "surfaceType" AS ENUM (
    'Synthetic Grass',
    'Grass',
    'Dirt',
    'Floating Wooden Floor',
    'Hardwood',
```

```
'Resilient',
'Rubber',
'Linoleum',
'Vinyl',
'Other'
);

ALTER TYPE "surfaceType" OWNER TO lbaw1653;

--
-- Name: banneduser(); Type: FUNCTION; Schema: public; Owner: lbaw1653
--

CREATE FUNCTION banneduser() RETURNS trigger
    LANGUAGE plpgsql
    AS $$BEGIN
IF (OLD."userIsBanned" = false AND New."userIsBanned" = true)
THEN
UPDATE "Rental" SET "rentalState" = 'SUSPENDED_BYADMIN'::"rentalState"
WHERE "rentalUserID" = New."userID";
END IF;
RETURN NEW;
END;
$$;

ALTER FUNCTION public.banneduser() OWNER TO lbaw1653;

--
-- Name: concludedissue(); Type: FUNCTION; Schema: public; Owner:
lbaw1653
--

CREATE FUNCTION concludedissue() RETURNS trigger
    LANGUAGE plpgsql
    AS $$BEGIN
IF(OLD."issueResolved" = false AND New."issueResolved" = true)
THEN
UPDATE "Rental" SET "rentalState" = 'CONCLUDED'::"rentalState" WHERE
"rentalID" = New."issueRentalID";
END IF;
RETURN NEW;
END;
$$;

ALTER FUNCTION public.concludedissue() OWNER TO lbaw1653;

SET default_tablespace = '';
```



```
SET default_with_oids = false;

--
-- Name: Admin; Type: TABLE; Schema: public; Owner: lbaw1653;
-- Tablespace:
--

CREATE TABLE "Admin" (
    "adminUsername" character varying(64) NOT NULL,
    "adminPassword" character varying(256) NOT NULL,
    "adminID" integer NOT NULL,
    "adminAccepted" boolean DEFAULT false NOT NULL
);

ALTER TABLE "Admin" OWNER TO lbaw1653;

--
-- Name: Admin_adminID_seq; Type: SEQUENCE; Schema: public; Owner:
-- lbaw1653
--

CREATE SEQUENCE "Admin_adminID_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE "Admin_adminID_seq" OWNER TO lbaw1653;

--
-- Name: Admin_adminID_seq; Type: SEQUENCE OWNED BY; Schema: public;
-- Owner: lbaw1653
--

ALTER SEQUENCE "Admin_adminID_seq" OWNED BY "Admin"."adminID";

--
-- Name: Equipment; Type: TABLE; Schema: public; Owner: lbaw1653;
-- Tablespace:
--

CREATE TABLE "Equipment" (
    "equipmentName" character varying(64) NOT NULL,
    "equipmentQuantity" smallint NOT NULL,
    "equipmentDetails" character varying(256) DEFAULT NULL::character
    varying,
```

```
"equipmentQuantityUnavailable" smallint DEFAULT NOT NULL,
"equipmentID" bigint NOT NULL,
"equipmentPrice" real NOT NULL,
"equipmentComplexID" integer NOT NULL,
"equipmentInactive" boolean DEFAULT false NOT NULL,
CONSTRAINT "equipmentCheckPrice" CHECK (("equipmentPrice" >
)::real)),
CONSTRAINT "equipmentCheckUnavailable" CHECK
(("equipmentQuantityUnavailable" <= "equipmentQuantity"))
);

ALTER TABLE "Equipment" OWNER TO lbaw1653;

--
-- Name: EquipmentSports; Type: TABLE; Schema: public; Owner: lbaw1653;
--
Tablespace:

CREATE TABLE "EquipmentSports" (
    "equipmentSportsEquipmentID" bigint NOT NULL,
    "equipmentSportsSportID" bigint NOT NULL
);

ALTER TABLE "EquipmentSports" OWNER TO lbaw1653;

--
-- Name: Equipment_equipmentID_seq; Type: SEQUENCE; Schema: public;
--
Owner: lbaw1653

CREATE SEQUENCE "Equipment_equipmentID_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE "Equipment_equipmentID_seq" OWNER TO lbaw1653;

--
-- Name: Equipment_equipmentID_seq; Type: SEQUENCE OWNED BY; Schema:
--
public; Owner: lbaw1653

ALTER SEQUENCE "Equipment_equipmentID_seq" OWNED BY
"Equipment"."equipmentID";
```

```
--
-- Name: Issue; Type: TABLE; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE TABLE "Issue" (
    "issueID" bigint NOT NULL,
    "issueSubject" text NOT NULL,
    "issueDescription" text NOT NULL,
    "issueForAdmin" boolean DEFAULT false NOT NULL,
    "issueForManager" boolean DEFAULT false NOT NULL,
    "issueCategory" "issueCategory" NOT NULL,
    "issueRentalID" bigint,
    "issueResolved" boolean DEFAULT false NOT NULL,
    "issueComplexID" bigint,
    "issueUserID" bigint
);

ALTER TABLE "Issue" OWNER TO lbaw1653;

--
-- Name: Issue_issueID_seq; Type: SEQUENCE; Schema: public; Owner:
lbaw1653
--

CREATE SEQUENCE "Issue_issueID_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE "Issue_issueID_seq" OWNER TO lbaw1653;

--
-- Name: Issue_issueID_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: lbaw1653
--

ALTER SEQUENCE "Issue_issueID_seq" OWNED BY "Issue"."issueID";

--
-- Name: Manager; Type: TABLE; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE TABLE "Manager" (
    "managerComplexID" integer NOT NULL,
```

```
        "managerID" bigint NOT NULL
    );

ALTER TABLE "Manager" OWNER TO lbaw1653;

--
-- Name: Municipality; Type: TABLE; Schema: public; Owner: lbaw1653;
-- Tablespace:
--

CREATE TABLE "Municipality" (
    "municipalityName" character varying(64) NOT NULL,
    "municipalityID" integer NOT NULL
);

ALTER TABLE "Municipality" OWNER TO lbaw1653;

--
-- Name: Municipality_municipalityID_seq; Type: SEQUENCE; Schema:
-- public; Owner: lbaw1653
--

CREATE SEQUENCE "Municipality_municipalityID_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE "Municipality_municipalityID_seq" OWNER TO lbaw1653;

--
-- Name: Municipality_municipalityID_seq; Type: SEQUENCE OWNED BY;
-- Schema: public; Owner: lbaw1653
--

ALTER SEQUENCE "Municipality_municipalityID_seq" OWNED BY
"Municipality"."municipalityID";

--
-- Name: Notification; Type: TABLE; Schema: public; Owner: lbaw1653;
-- Tablespace:
--

CREATE TABLE "Notification" (
    "notificationUserID" bigint NOT NULL,
    "notificationText" character varying NOT NULL
```

```
);

ALTER TABLE "Notification" OWNER TO lbaw1653;

--
-- Name: Rental; Type: TABLE; Schema: public; Owner: lbaw1653;
-- Tablespace:
--

CREATE TABLE "Rental" (
    "rentalID" bigint NOT NULL,
    "rentalRating" smallint,
    "rentalStartTime" time without time zone NOT NULL,
    "rentalDuration" interval NOT NULL,
    "rentalDate" date NOT NULL,
    "rentalState" "rentalState" DEFAULT 'RESERVED'::"rentalState" NOT
NULL,
    "rentalSpaceID" bigint NOT NULL,
    "rentalUserID" bigint NOT NULL,
    CONSTRAINT "rentalValidDuration" CHECK (("rentalDuration" >
'00:00:00'::interval)),
    CONSTRAINT "rentalValidRating" CHECK (((("rentalRating" >= 1) AND
("rentalRating" <= 5))))
);

ALTER TABLE "Rental" OWNER TO lbaw1653;

--
-- Name: RentalEquipment; Type: TABLE; Schema: public; Owner: lbaw1653;
-- Tablespace:
--

CREATE TABLE "RentalEquipment" (
    "rentalEquipmentRentalID" bigint NOT NULL,
    "rentalEquipmentEquipmentID" bigint NOT NULL,
    "rentalEquipmentQuantity" smallint NOT NULL,
    CONSTRAINT "rentalEquipmentQuantityPositive" CHECK
(("rentalEquipmentQuantity" > ))
);

ALTER TABLE "RentalEquipment" OWNER TO lbaw1653;

--
-- Name: Rental_rentalID_seq; Type: SEQUENCE; Schema: public; Owner:
lbaw1653
--

CREATE SEQUENCE "Rental_rentalID_seq"
```

```
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;

ALTER TABLE "Rental_rentalID_seq" OWNER TO lbaw1653;

--
-- Name: Rental_rentalID_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: lbaw1653
--

ALTER SEQUENCE "Rental_rentalID_seq" OWNED BY "Rental"."rentalID";

--
-- Name: Space; Type: TABLE; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE TABLE "Space" (
    "spaceID" bigint NOT NULL,
    "spaceIsCovered" boolean NOT NULL,
    "spaceIsAvailable" boolean NOT NULL,
    "spacePrice" real NOT NULL,
    "spaceComplexID" integer NOT NULL,
    "spaceName" character varying(64),
    "spaceSurfaceType" "surfaceType" NOT NULL,
    CONSTRAINT "spaceCheckPositivePrice" CHECK (("spacePrice" >
)::real))
);

ALTER TABLE "Space" OWNER TO lbaw1653;

--
-- Name: SpaceSports; Type: TABLE; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE TABLE "SpaceSports" (
    "spaceSportsSpaceID" bigint NOT NULL,
    "spaceSportsSportID" bigint NOT NULL
);

ALTER TABLE "SpaceSports" OWNER TO lbaw1653;

--
```

```
-- Name: Space_spaceID_seq; Type: SEQUENCE; Schema: public; Owner:
lbaw1653
--

CREATE SEQUENCE "Space_spaceID_seq"
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE "Space_spaceID_seq" OWNER TO lbaw1653;

--
-- Name: Space_spaceID_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: lbaw1653
--

ALTER SEQUENCE "Space_spaceID_seq" OWNED BY "Space"."spaceID";

--
-- Name: Sport; Type: TABLE; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE TABLE "Sport" (
  "sportName" character varying(64) NOT NULL,
  "sportID" integer NOT NULL
);

ALTER TABLE "Sport" OWNER TO lbaw1653;

--
-- Name: Sport_sportID_seq; Type: SEQUENCE; Schema: public; Owner:
lbaw1653
--

CREATE SEQUENCE "Sport_sportID_seq"
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE "Sport_sportID_seq" OWNER TO lbaw1653;

--
```

```
-- Name: Sport_sportID_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: lbaw1653
--

ALTER SEQUENCE "Sport_sportID_seq" OWNED BY "Sport"."sportID";

--
-- Name: SportsComplex; Type: TABLE; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE TABLE "SportsComplex" (
    "complexID" integer NOT NULL,
    "complexName" character varying(64) NOT NULL,
    "complexLocation" character varying(256) NOT NULL,
    "complexOpenOnWeekends" boolean NOT NULL,
    "complexEmail" character varying(64) NOT NULL,
    "complexPhone" character varying(20) NOT NULL,
    "complexPaypal" character varying(64) NOT NULL,
    "complexDescription" character varying(2000) DEFAULT
NULL::character varying,
    "complexOpeningHour" time without time zone NOT NULL,
    "complexClosingHour" time without time zone NOT NULL,
    "complexMunicipalityID" bigint NOT NULL,
    "complexInactive" boolean DEFAULT false NOT NULL,
    CONSTRAINT "complexValidHours" CHECK (("complexClosingHour" >
"complexOpeningHour"))
);

ALTER TABLE "SportsComplex" OWNER TO lbaw1653;

--
-- Name: SportsComplex_complexId_seq; Type: SEQUENCE; Schema: public;
Owner: lbaw1653
--

CREATE SEQUENCE "SportsComplex_complexId_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE "SportsComplex_complexId_seq" OWNER TO lbaw1653;

--
-- Name: SportsComplex_complexId_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: lbaw1653
```



```
--  
  
ALTER SEQUENCE "SportsComplex_complexId_seq" OWNED BY  
"SportsComplex"."complexID";  
  
--  
-- Name: User; Type: TABLE; Schema: public; Owner: lbaw1653;  
Tablespace:  
--  
  
CREATE TABLE "User" (  
    "userUsername" character varying(64) NOT NULL,  
    "userPassword" character varying(256) NOT NULL,  
    "userName" character varying(64) NOT NULL,  
    "userEmail" character varying(64) NOT NULL,  
    "userPhone" character varying(20) DEFAULT NULL::character varying,  
    "userIsBanned" boolean DEFAULT false NOT NULL,  
    "userID" integer NOT NULL,  
    "userMunicipalityID" bigint  
);  
  
ALTER TABLE "User" OWNER TO lbaw1653;  
  
--  
-- Name: User_userID_seq; Type: SEQUENCE; Schema: public; Owner:  
lbaw1653  
--  
  
CREATE SEQUENCE "User_userID_seq"  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER TABLE "User_userID_seq" OWNER TO lbaw1653;  
  
--  
-- Name: User_userID_seq; Type: SEQUENCE OWNED BY; Schema: public;  
Owner: lbaw1653  
--  
  
ALTER SEQUENCE "User_userID_seq" OWNED BY "User"."userID";  
  
--  
-- Name: adminID; Type: DEFAULT; Schema: public; Owner: lbaw1653  
--
```

```
ALTER TABLE ONLY "Admin" ALTER COLUMN "adminID" SET DEFAULT
nextval('"Admin_adminID_seq"'::regclass);

--
-- Name: equipmentID; Type: DEFAULT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Equipment" ALTER COLUMN "equipmentID" SET DEFAULT
nextval('"Equipment_equipmentID_seq"'::regclass);

--
-- Name: issueID; Type: DEFAULT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Issue" ALTER COLUMN "issueID" SET DEFAULT
nextval('"Issue_issueID_seq"'::regclass);

--
-- Name: municipalityID; Type: DEFAULT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Municipality" ALTER COLUMN "municipalityID" SET
DEFAULT nextval('"Municipality_municipalityID_seq"'::regclass);

--
-- Name: rentalID; Type: DEFAULT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Rental" ALTER COLUMN "rentalID" SET DEFAULT
nextval('"Rental_rentalID_seq"'::regclass);

--
-- Name: spaceID; Type: DEFAULT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Space" ALTER COLUMN "spaceID" SET DEFAULT
nextval('"Space_spaceID_seq"'::regclass);

--
-- Name: sportID; Type: DEFAULT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Sport" ALTER COLUMN "sportID" SET DEFAULT
nextval('"Sport_sportID_seq"'::regclass);
```

```
--  
-- Name: complexID; Type: DEFAULT; Schema: public; Owner: lbaw1653  
--  
  
ALTER TABLE ONLY "SportsComplex" ALTER COLUMN "complexID" SET DEFAULT  
nextval('"SportsComplex_complexId_seq"'::regclass);  
  
--  
-- Name: userID; Type: DEFAULT; Schema: public; Owner: lbaw1653  
--  
  
ALTER TABLE ONLY "User" ALTER COLUMN "userID" SET DEFAULT  
nextval('"User_userID_seq"'::regclass);  
  
--  
-- Name: Admin_pkey; Type: CONSTRAINT; Schema: public; Owner: lbaw1653;  
Tablespace:  
--  
  
ALTER TABLE ONLY "Admin"  
    ADD CONSTRAINT "Admin_pkey" PRIMARY KEY ("adminID");  
  
--  
-- Name: EquipmentSports_pkey; Type: CONSTRAINT; Schema: public; Owner:  
lbaw1653; Tablespace:  
--  
  
ALTER TABLE ONLY "EquipmentSports"  
    ADD CONSTRAINT "EquipmentSports_pkey" PRIMARY KEY  
("equipmentSportsEquipmentID", "equipmentSportsSportID");  
  
--  
-- Name: Equipment_pkey; Type: CONSTRAINT; Schema: public; Owner:  
lbaw1653; Tablespace:  
--  
  
ALTER TABLE ONLY "Equipment"  
    ADD CONSTRAINT "Equipment_pkey" PRIMARY KEY ("equipmentID");  
  
--  
-- Name: Issue_pkey; Type: CONSTRAINT; Schema: public; Owner: lbaw1653;  
Tablespace:  
--
```

```
ALTER TABLE ONLY "Issue"
    ADD CONSTRAINT "Issue_pkey" PRIMARY KEY ("issueID");

--
-- Name: Manager_pkey; Type: CONSTRAINT; Schema: public; Owner:
lbaw1653; Tablespace:
--

ALTER TABLE ONLY "Manager"
    ADD CONSTRAINT "Manager_pkey" PRIMARY KEY ("managerComplexID",
"managerID");

--
-- Name: Municipality_pkey; Type: CONSTRAINT; Schema: public; Owner:
lbaw1653; Tablespace:
--

ALTER TABLE ONLY "Municipality"
    ADD CONSTRAINT "Municipality_pkey" PRIMARY KEY ("municipalityID");

--
-- Name: Notification_pkey; Type: CONSTRAINT; Schema: public; Owner:
lbaw1653; Tablespace:
--

ALTER TABLE ONLY "Notification"
    ADD CONSTRAINT "Notification_pkey" PRIMARY KEY
("notificationUserID", "notificationText");

--
-- Name: RentalEquipment_pkey; Type: CONSTRAINT; Schema: public; Owner:
lbaw1653; Tablespace:
--

ALTER TABLE ONLY "RentalEquipment"
    ADD CONSTRAINT "RentalEquipment_pkey" PRIMARY KEY
("rentalEquipmentRentalID", "rentalEquipmentEquipmentID");

--
-- Name: Rental_pkey; Type: CONSTRAINT; Schema: public; Owner:
lbaw1653; Tablespace:
--

ALTER TABLE ONLY "Rental"
    ADD CONSTRAINT "Rental_pkey" PRIMARY KEY ("rentalID");
```

```
--  
-- Name: SpaceSports_pkey; Type: CONSTRAINT; Schema: public; Owner:  
lbaw1653; Tablespace:  
--  
  
ALTER TABLE ONLY "SpaceSports"  
    ADD CONSTRAINT "SpaceSports_pkey" PRIMARY KEY  
    ("spaceSportsSpaceID", "spaceSportsSportID");  
  
--  
-- Name: Space_pkey; Type: CONSTRAINT; Schema: public; Owner: lbaw1653;  
Tablespace:  
--  
  
ALTER TABLE ONLY "Space"  
    ADD CONSTRAINT "Space_pkey" PRIMARY KEY ("spaceID");  
  
--  
-- Name: Sport_pkey; Type: CONSTRAINT; Schema: public; Owner: lbaw1653;  
Tablespace:  
--  
  
ALTER TABLE ONLY "Sport"  
    ADD CONSTRAINT "Sport_pkey" PRIMARY KEY ("sportID");  
  
--  
-- Name: SportsComplex_pkey; Type: CONSTRAINT; Schema: public; Owner:  
lbaw1653; Tablespace:  
--  
  
ALTER TABLE ONLY "SportsComplex"  
    ADD CONSTRAINT "SportsComplex_pkey" PRIMARY KEY ("complexID");  
  
--  
-- Name: User_pkey; Type: CONSTRAINT; Schema: public; Owner: lbaw1653;  
Tablespace:  
--  
  
ALTER TABLE ONLY "User"  
    ADD CONSTRAINT "User_pkey" PRIMARY KEY ("userID");  
  
--  
-- Name: User_userEmail_key; Type: CONSTRAINT; Schema: public; Owner:  
lbaw1653; Tablespace:  
--
```

```
ALTER TABLE ONLY "User"
    ADD CONSTRAINT "User_userEmail_key" UNIQUE ("userEmail");

--
-- Name: User_userUsername_key; Type: CONSTRAINT; Schema: public;
Owner: lbaw1653; Tablespace:
--

ALTER TABLE ONLY "User"
    ADD CONSTRAINT "User_userUsername_key" UNIQUE ("userUsername");

--
-- Name: equipmentInfoIndex; Type: INDEX; Schema: public; Owner:
lbaw1653; Tablespace:
--

CREATE INDEX "equipmentInfoIndex" ON "Equipment" USING btree
("equipmentComplexID", "equipmentID", "equipmentQuantity",
"equipmentQuantityUnavailable");

--
-- Name: issueRentalIndex; Type: INDEX; Schema: public; Owner:
lbaw1653; Tablespace:
--

CREATE INDEX "issueRentalIndex" ON "Issue" USING hash
("issueRentalID");

--
-- Name: rentalSpaceTimeIndex; Type: INDEX; Schema: public; Owner:
lbaw1653; Tablespace:
--

CREATE INDEX "rentalSpaceTimeIndex" ON "Rental" USING btree
("rentalID", "rentalSpaceID", "rentalStartTime", "rentalDuration");

--
-- Name: rentalUserIndex; Type: INDEX; Schema: public; Owner: lbaw1653;
Tablespace:
--

CREATE INDEX "rentalUserIndex" ON "Rental" USING hash ("rentalUserID");

--
```

```
-- Name: sportComplexSearchIndex; Type: INDEX; Schema: public; Owner:
lbaw1653; Tablespace:
--

CREATE INDEX "sportComplexSearchIndex" ON "SportsComplex" USING hash
("complexLocation");

--

-- Name: sportcomplexsearchindexfulltext; Type: INDEX; Schema: public;
Owner: lbaw1653; Tablespace:
--

CREATE INDEX sportcomplexsearchindexfulltext ON "SportsComplex" USING
gin (to_tsvector('english'::regconfig, (((("complexName")::text || '
'::text) || ("complexLocation")::text) || ' '::text) ||
("complexDescription")::text)));

--

-- Name: sportcomplexsearchindexfulltextweighted; Type: INDEX; Schema:
public; Owner: lbaw1653; Tablespace:
--

CREATE INDEX sportcomplexsearchindexfulltextweighted ON "SportsComplex"
USING gin (((setweight(to_tsvector('english'::regconfig,
("complexName")::text), 'A'::"char") ||
setweight(to_tsvector('english'::regconfig, ("complexLocation")::text),
'B'::"char"))) || setweight(to_tsvector('english'::regconfig,
("complexDescription")::text), 'C'::"char"))));

--

-- Name: sportcomplexsearchindexname; Type: INDEX; Schema: public;
Owner: lbaw1653; Tablespace:
--

CREATE INDEX sportcomplexsearchindexname ON "SportsComplex" USING gin
(to_tsvector('english'::regconfig, ("complexName")::text));

--

-- Name: banneduser; Type: TRIGGER; Schema: public; Owner: lbaw1653
--

CREATE TRIGGER banneduser AFTER UPDATE OF "userIsBanned" ON "User" FOR
EACH ROW EXECUTE PROCEDURE banneduser();

--

-- Name: concludedissue; Type: TRIGGER; Schema: public; Owner: lbaw1653
```

```
--

CREATE TRIGGER concludedissue AFTER UPDATE OF "issueResolved" ON
"Issue" FOR EACH ROW EXECUTE PROCEDURE concludedissue();

--
-- Name: EquipmentSports_equipmentsportSportID_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "EquipmentSports"
    ADD CONSTRAINT "EquipmentSports_equipmentsportSportID_fkey" FOREIGN
KEY ("equipmentSportsSportID") REFERENCES "Sport"("sportID");

--
-- Name: EquipmentSports_equipmentsportsEquipmentID_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "EquipmentSports"
    ADD CONSTRAINT "EquipmentSports_equipmentsportsEquipmentID_fkey"
FOREIGN KEY ("equipmentSportsEquipmentID") REFERENCES
"Equipment"("equipmentID");

--
-- Name: Equipment_equipmentsComplexID_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "Equipment"
    ADD CONSTRAINT "Equipment_equipmentsComplexID_fkey" FOREIGN KEY
("equipmentComplexID") REFERENCES "SportsComplex"("complexID");

--
-- Name: Issue_issueRentalID_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: lbaw1653
--

ALTER TABLE ONLY "Issue"
    ADD CONSTRAINT "Issue_issueRentalID_fkey" FOREIGN KEY
("issueRentalID") REFERENCES "Rental"("rentalID");

--
-- Name: Manager_managerComplexID_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: lbaw1653
--
```



```
ALTER TABLE ONLY "Manager"
    ADD CONSTRAINT "Manager_managerComplexID_fkey" FOREIGN KEY
("managerComplexID") REFERENCES "SportsComplex"("complexID");

--
-- Name: Manager_managerID_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: lbaw1653
--

ALTER TABLE ONLY "Manager"
    ADD CONSTRAINT "Manager_managerID_fkey" FOREIGN KEY ("managerID")
REFERENCES "User"("userID");

--
-- Name: RentalEquipment_rentalEquipmentEquipmentID_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "RentalEquipment"
    ADD CONSTRAINT "RentalEquipment_rentalEquipmentEquipmentID_fkey"
FOREIGN KEY ("rentalEquipmentEquipmentID") REFERENCES
"Equipment"("equipmentID");

--
-- Name: RentalEquipment_rentalEquipmentRentalID_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "RentalEquipment"
    ADD CONSTRAINT "RentalEquipment_rentalEquipmentRentalID_fkey"
FOREIGN KEY ("rentalEquipmentRentalID") REFERENCES
"Rental"("rentalID");

--
-- Name: Rental_rentalSpaceID_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: lbaw1653
--

ALTER TABLE ONLY "Rental"
    ADD CONSTRAINT "Rental_rentalSpaceID_fkey" FOREIGN KEY
("rentalSpaceID") REFERENCES "Space"("spaceID");

--
-- Name: Rental_rentalUserID_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: lbaw1653
```

```
--

ALTER TABLE ONLY "Rental"
    ADD CONSTRAINT "Rental_rentalUserID_fkey" FOREIGN KEY
("rentalUserID") REFERENCES "User"("userID");

--
-- Name: SpaceSports_spaceID_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: lbaw1653
--

ALTER TABLE ONLY "SpaceSports"
    ADD CONSTRAINT "SpaceSports_spaceID_fkey" FOREIGN KEY
("spaceSportsSpaceID") REFERENCES "Space"("spaceID");

--
-- Name: SpaceSports_sportID_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: lbaw1653
--

ALTER TABLE ONLY "SpaceSports"
    ADD CONSTRAINT "SpaceSports_sportID_fkey" FOREIGN KEY
("spaceSportsSportID") REFERENCES "Sport"("sportID");

--
-- Name: Space_spaceComplexID_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: lbaw1653
--

ALTER TABLE ONLY "Space"
    ADD CONSTRAINT "Space_spaceComplexID_fkey" FOREIGN KEY
("spaceComplexID") REFERENCES "SportsComplex"("complexID");

--
-- Name: SportsComplex_complexMunicipalityID_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: lbaw1653
--

ALTER TABLE ONLY "SportsComplex"
    ADD CONSTRAINT "SportsComplex_complexMunicipalityID_fkey" FOREIGN
KEY ("complexMunicipalityID") REFERENCES
"Municipality"("municipalityID");

--
-- Name: User_userMunicipalityID_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: lbaw1653
```

```
--
--
ALTER TABLE ONLY "User"
  ADD CONSTRAINT "User_userMunicipalityID_fkey" FOREIGN KEY
    ("userMunicipalityID") REFERENCES "Municipality"("municipalityID");

--
-- Name: public; Type: ACL; Schema: -; Owner: lbaw1653
--

REVOKE ALL ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON SCHEMA public FROM lbaw1653;
GRANT ALL ON SCHEMA public TO lbaw1653;

--
-- PostgreSQL database dump complete
--
```

## Revision History

Date	Description
2017/03/29	<ul style="list-style-type: none"> <li>- changed "User" primary key from userUsername to userID</li> <li>- changed "Admin" primary key from adminUsername to adminID</li> <li>- changed "Municipality" primary key from municipalityName to municipalityID</li> <li>- changed "Sport" primary key from sportName to sportID</li> <li>- complexDescription's type changed to "text"</li> <li>- equipmentPrice's type changed to "money"</li> <li>- added the "Enumeration" tables</li> <li>- functional dependencies text was added</li> <li>- equipmentQuantityOccupied removed from the table Equipment</li> <li>- added issueResolved to the table Issue</li> </ul>
2017/05/04	- added 'Disabled by Admin' as a new type of rentalState
2017/05/18	<ul style="list-style-type: none"> <li>- changed userName from UK NN to only NN</li> <li>- changed municipalityName from NN to UK NN</li> <li>- changed adminUsername from NN to UK NN</li> <li>- changed sportName from NN to UK NN</li> </ul>

From:

<http://lbaw.fe.up.pt/201617/> - **L B A W :: WORK**

Permanent link:

<http://lbaw.fe.up.pt/201617/doku.php/lbaw1653/a5>

Last update: **2017/06/01 18:37**

