>>>>>>>>>>>>>>>>>>CREATING Movie\_rental DATABASE<<<<<<<<<<<<<<<

-- Database: Movie\_rental

-- DROP DATABASE "Movie\_rental";

CREATE DATABASE "Movie\_rental"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

>>>>>>>>>>>>>>>>>>CREATING **actor** TABLE<<<<<<<<<<<<<<

-- Table: public.actor

-- DROP TABLE public.actor;

CREATE TABLE public.actor

(

actor\_id integer NOT NULL DEFAULT nextval('actor\_actor\_id\_seq'::regclass),

first\_name character varying(45) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(45) COLLATE pg\_catalog."default" NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT actor\_pkey PRIMARY KEY (actor\_id)

)

TABLESPACE pg\_default;

ALTER TABLE public.actor

OWNER to postgres;

-- Index: idx\_actor\_last\_name

-- DROP INDEX public.idx\_actor\_last\_name;

CREATE INDEX idx\_actor\_last\_name

ON public.actor USING btree

(last\_name COLLATE pg\_catalog."default" ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.actor;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.actor

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>CREATING **address** TABLE<<<<<<<<<<<<<<

-- Table: public.address

-- DROP TABLE public.address;

CREATE TABLE public.address

(

address\_id integer NOT NULL DEFAULT nextval('address\_address\_id\_seq'::regclass),

address character varying(50) COLLATE pg\_catalog."default" NOT NULL,

address2 character varying(50) COLLATE pg\_catalog."default",

district character varying(20) COLLATE pg\_catalog."default" NOT NULL,

city\_id smallint NOT NULL,

postal\_code character varying(10) COLLATE pg\_catalog."default",

phone character varying(20) COLLATE pg\_catalog."default" NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT address\_pkey PRIMARY KEY (address\_id),

CONSTRAINT fk\_address\_city FOREIGN KEY (city\_id)

REFERENCES public.city (city\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.address

OWNER to postgres;

-- Index: idx\_fk\_city\_id

-- DROP INDEX public.idx\_fk\_city\_id;

CREATE INDEX idx\_fk\_city\_id

ON public.address USING btree

(city\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.address;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.address

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>CREATING category TABLE<<<<<<<<<<<<<<

-- Table: public.category

-- DROP TABLE public.category;

CREATE TABLE public.category

(

category\_id integer NOT NULL DEFAULT nextval('category\_category\_id\_seq'::regclass),

name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT category\_pkey PRIMARY KEY (category\_id)

)

TABLESPACE pg\_default;

ALTER TABLE public.category

OWNER to postgres;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.category;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.category

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>CREATING **city** TABLE<<<<<<<<<<<<<<<<

-- Table: public.city

-- DROP TABLE public.city;

CREATE TABLE public.city

(

city\_id integer NOT NULL DEFAULT nextval('city\_city\_id\_seq'::regclass),

city character varying(50) COLLATE pg\_catalog."default" NOT NULL,

country\_id smallint NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT city\_pkey PRIMARY KEY (city\_id),

CONSTRAINT fk\_city FOREIGN KEY (country\_id)

REFERENCES public.country (country\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.city

OWNER to postgres;

-- Index: idx\_fk\_country\_id

-- DROP INDEX public.idx\_fk\_country\_id;

CREATE INDEX idx\_fk\_country\_id

ON public.city USING btree

(country\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.city;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.city

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>CREATING **country** TABLE<<<<<<<<<<<<

-- Table: public.country

-- DROP TABLE public.country;

CREATE TABLE public.country

(

country\_id integer NOT NULL DEFAULT nextval('country\_country\_id\_seq'::regclass),

country character varying(50) COLLATE pg\_catalog."default" NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT country\_pkey PRIMARY KEY (country\_id)

)

TABLESPACE pg\_default;

ALTER TABLE public.country

OWNER to postgres;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.country;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.country

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>CREATING **customer** TABLE<<<<<<<<<<<<<<<<<<

-- Table: public.customer

-- DROP TABLE public.customer;

CREATE TABLE public.customer

(

customer\_id integer NOT NULL DEFAULT nextval('customer\_customer\_id\_seq'::regclass),

store\_id smallint NOT NULL,

first\_name character varying(45) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(45) COLLATE pg\_catalog."default" NOT NULL,

email character varying(50) COLLATE pg\_catalog."default",

address\_id smallint NOT NULL,

activebool boolean NOT NULL DEFAULT true,

create\_date date NOT NULL DEFAULT ('now'::text)::date,

last\_update timestamp without time zone DEFAULT now(),

active integer,

CONSTRAINT customer\_pkey PRIMARY KEY (customer\_id),

CONSTRAINT customer\_address\_id\_fkey FOREIGN KEY (address\_id)

REFERENCES public.address (address\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.customer

OWNER to postgres;

-- Index: idx\_fk\_address\_id

-- DROP INDEX public.idx\_fk\_address\_id;

CREATE INDEX idx\_fk\_address\_id

ON public.customer USING btree

(address\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: idx\_fk\_store\_id

-- DROP INDEX public.idx\_fk\_store\_id;

CREATE INDEX idx\_fk\_store\_id

ON public.customer USING btree

(store\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: idx\_last\_name

-- DROP INDEX public.idx\_last\_name;

CREATE INDEX idx\_last\_name

ON public.customer USING btree

(last\_name COLLATE pg\_catalog."default" ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.customer;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.customer

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>CREATING **film** TABLE<<<<<<<<<<<<<

-- Table: public.film

-- DROP TABLE public.film;

CREATE TABLE public.film

(

film\_id integer NOT NULL DEFAULT nextval('film\_film\_id\_seq'::regclass),

title character varying(255) COLLATE pg\_catalog."default" NOT NULL,

description text COLLATE pg\_catalog."default",

release\_year year,

language\_id smallint NOT NULL,

rental\_duration smallint NOT NULL DEFAULT 3,

rental\_rate numeric(4,2) NOT NULL DEFAULT 4.99,

length smallint,

replacement\_cost numeric(5,2) NOT NULL DEFAULT 19.99,

rating mpaa\_rating DEFAULT 'G'::mpaa\_rating,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

special\_features text[] COLLATE pg\_catalog."default",

fulltext tsvector NOT NULL,

CONSTRAINT film\_pkey PRIMARY KEY (film\_id),

CONSTRAINT film\_language\_id\_fkey FOREIGN KEY (language\_id)

REFERENCES public.language (language\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.film

OWNER to postgres;

-- Index: film\_fulltext\_idx

-- DROP INDEX public.film\_fulltext\_idx;

CREATE INDEX film\_fulltext\_idx

ON public.film USING gist

(fulltext)

TABLESPACE pg\_default;

-- Index: idx\_fk\_language\_id

-- DROP INDEX public.idx\_fk\_language\_id;

CREATE INDEX idx\_fk\_language\_id

ON public.film USING btree

(language\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: idx\_title

-- DROP INDEX public.idx\_title;

CREATE INDEX idx\_title

ON public.film USING btree

(title COLLATE pg\_catalog."default" ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: film\_fulltext\_trigger

-- DROP TRIGGER film\_fulltext\_trigger ON public.film;

CREATE TRIGGER film\_fulltext\_trigger

BEFORE INSERT OR UPDATE

ON public.film

FOR EACH ROW

EXECUTE PROCEDURE tsvector\_update\_trigger('fulltext', 'pg\_catalog.english', 'title', 'description');

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.film;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.film

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>CREATING **film\_actor** TABLE<<<<<<<<<<<<<<<<

-- Table: public.film\_actor

-- DROP TABLE public.film\_actor;

CREATE TABLE public.film\_actor

(

actor\_id smallint NOT NULL,

film\_id smallint NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT film\_actor\_pkey PRIMARY KEY (actor\_id, film\_id),

CONSTRAINT film\_actor\_actor\_id\_fkey FOREIGN KEY (actor\_id)

REFERENCES public.actor (actor\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT,

CONSTRAINT film\_actor\_film\_id\_fkey FOREIGN KEY (film\_id)

REFERENCES public.film (film\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.film\_actor

OWNER to postgres;

-- Index: idx\_fk\_film\_id

-- DROP INDEX public.idx\_fk\_film\_id;

CREATE INDEX idx\_fk\_film\_id

ON public.film\_actor USING btree

(film\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.film\_actor;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.film\_actor

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>>>>>CREATING **film\_category** TABLE<<<<<<<<<<<<<<<<<

-- Table: public.film\_category

-- DROP TABLE public.film\_category;

CREATE TABLE public.film\_category

(

film\_id smallint NOT NULL,

category\_id smallint NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT film\_category\_pkey PRIMARY KEY (film\_id, category\_id),

CONSTRAINT film\_category\_category\_id\_fkey FOREIGN KEY (category\_id)

REFERENCES public.category (category\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT,

CONSTRAINT film\_category\_film\_id\_fkey FOREIGN KEY (film\_id)

REFERENCES public.film (film\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.film\_category

OWNER to postgres;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.film\_category;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.film\_category

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>>CREATING **inventory** TABLE<<<<<<<<<<<<<<<

-- Table: public.inventory

-- DROP TABLE public.inventory;

CREATE TABLE public.inventory

(

inventory\_id integer NOT NULL DEFAULT nextval('inventory\_inventory\_id\_seq'::regclass),

film\_id smallint NOT NULL,

store\_id smallint NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT inventory\_pkey PRIMARY KEY (inventory\_id),

CONSTRAINT inventory\_film\_id\_fkey FOREIGN KEY (film\_id)

REFERENCES public.film (film\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.inventory

OWNER to postgres;

-- Index: idx\_store\_id\_film\_id

-- DROP INDEX public.idx\_store\_id\_film\_id;

CREATE INDEX idx\_store\_id\_film\_id

ON public.inventory USING btree

(store\_id ASC NULLS LAST, film\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.inventory;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.inventory

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>CREATING **language** TABLE<<<<<<<<<<<<<

-- Table: public.language

-- DROP TABLE public.language;

CREATE TABLE public.language

(

language\_id integer NOT NULL DEFAULT nextval('language\_language\_id\_seq'::regclass),

name character(20) COLLATE pg\_catalog."default" NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT language\_pkey PRIMARY KEY (language\_id)

)

TABLESPACE pg\_default;

ALTER TABLE public.language

OWNER to postgres;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.language;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.language

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>CREATING payment TABLE<<<<<<<<<<<<<<<

-- Table: public.payment

-- DROP TABLE public.payment;

CREATE TABLE public.payment

(

payment\_id integer NOT NULL DEFAULT nextval('payment\_payment\_id\_seq'::regclass),

customer\_id smallint NOT NULL,

staff\_id smallint NOT NULL,

rental\_id integer NOT NULL,

amount numeric(5,2) NOT NULL,

payment\_date timestamp without time zone NOT NULL,

CONSTRAINT payment\_pkey PRIMARY KEY (payment\_id),

CONSTRAINT payment\_customer\_id\_fkey FOREIGN KEY (customer\_id)

REFERENCES public.customer (customer\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT,

CONSTRAINT payment\_rental\_id\_fkey FOREIGN KEY (rental\_id)

REFERENCES public.rental (rental\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE SET NULL,

CONSTRAINT payment\_staff\_id\_fkey FOREIGN KEY (staff\_id)

REFERENCES public.staff (staff\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.payment

OWNER to postgres;

-- Index: idx\_fk\_customer\_id

-- DROP INDEX public.idx\_fk\_customer\_id;

CREATE INDEX idx\_fk\_customer\_id

ON public.payment USING btree

(customer\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: idx\_fk\_rental\_id

-- DROP INDEX public.idx\_fk\_rental\_id;

CREATE INDEX idx\_fk\_rental\_id

ON public.payment USING btree

(rental\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: idx\_fk\_staff\_id

-- DROP INDEX public.idx\_fk\_staff\_id;

CREATE INDEX idx\_fk\_staff\_id

ON public.payment USING btree

(staff\_id ASC NULLS LAST)

TABLESPACE pg\_default;

>>>>>>>>>>>>>>>>>CREATING **rental** TABLE<<<<<<<<<<<<<<<<

-- Table: public.rental

-- DROP TABLE public.rental;

CREATE TABLE public.rental

(

rental\_id integer NOT NULL DEFAULT nextval('rental\_rental\_id\_seq'::regclass),

rental\_date timestamp without time zone NOT NULL,

inventory\_id integer NOT NULL,

customer\_id smallint NOT NULL,

return\_date timestamp without time zone,

staff\_id smallint NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT rental\_pkey PRIMARY KEY (rental\_id),

CONSTRAINT rental\_customer\_id\_fkey FOREIGN KEY (customer\_id)

REFERENCES public.customer (customer\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT,

CONSTRAINT rental\_inventory\_id\_fkey FOREIGN KEY (inventory\_id)

REFERENCES public.inventory (inventory\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT,

CONSTRAINT rental\_staff\_id\_key FOREIGN KEY (staff\_id)

REFERENCES public.staff (staff\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.rental

OWNER to postgres;

-- Index: idx\_fk\_inventory\_id

-- DROP INDEX public.idx\_fk\_inventory\_id;

CREATE INDEX idx\_fk\_inventory\_id

ON public.rental USING btree

(inventory\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: idx\_unq\_rental\_rental\_date\_inventory\_id\_customer\_id

-- DROP INDEX public.idx\_unq\_rental\_rental\_date\_inventory\_id\_customer\_id;

CREATE UNIQUE INDEX idx\_unq\_rental\_rental\_date\_inventory\_id\_customer\_id

ON public.rental USING btree

(rental\_date ASC NULLS LAST, inventory\_id ASC NULLS LAST, customer\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.rental;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.rental

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>CREATING **staff** TABLE<<<<<<<<<<<<<

-- Table: public.staff

-- DROP TABLE public.staff;

CREATE TABLE public.staff

(

staff\_id integer NOT NULL DEFAULT nextval('staff\_staff\_id\_seq'::regclass),

first\_name character varying(45) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(45) COLLATE pg\_catalog."default" NOT NULL,

address\_id smallint NOT NULL,

email character varying(50) COLLATE pg\_catalog."default",

store\_id smallint NOT NULL,

active boolean NOT NULL DEFAULT true,

username character varying(16) COLLATE pg\_catalog."default" NOT NULL,

password character varying(40) COLLATE pg\_catalog."default",

last\_update timestamp without time zone NOT NULL DEFAULT now(),

picture bytea,

CONSTRAINT staff\_pkey PRIMARY KEY (staff\_id),

CONSTRAINT staff\_address\_id\_fkey FOREIGN KEY (address\_id)

REFERENCES public.address (address\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.staff

OWNER to postgres;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.staff;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.staff

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>>>>CREATING **store** TABLE<<<<<<<<<<<<<<<<

-- Table: public.store

-- DROP TABLE public.store;

CREATE TABLE public.store

(

store\_id integer NOT NULL DEFAULT nextval('store\_store\_id\_seq'::regclass),

manager\_staff\_id smallint NOT NULL,

address\_id smallint NOT NULL,

last\_update timestamp without time zone NOT NULL DEFAULT now(),

CONSTRAINT store\_pkey PRIMARY KEY (store\_id),

CONSTRAINT store\_address\_id\_fkey FOREIGN KEY (address\_id)

REFERENCES public.address (address\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT,

CONSTRAINT store\_manager\_staff\_id\_fkey FOREIGN KEY (manager\_staff\_id)

REFERENCES public.staff (staff\_id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE RESTRICT

)

TABLESPACE pg\_default;

ALTER TABLE public.store

OWNER to postgres;

-- Index: idx\_unq\_manager\_staff\_id

-- DROP INDEX public.idx\_unq\_manager\_staff\_id;

CREATE UNIQUE INDEX idx\_unq\_manager\_staff\_id

ON public.store USING btree

(manager\_staff\_id ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: last\_updated

-- DROP TRIGGER last\_updated ON public.store;

CREATE TRIGGER last\_updated

BEFORE UPDATE

ON public.store

FOR EACH ROW

EXECUTE PROCEDURE public.last\_updated();

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

RESTORING THE DATA ENTRIES

The data for all the table structures is compressed in a **.tar file** and was imported from the local folder using the **RESTORE function** on the pgAdmin.