>>>>>>>>>>>>>>>>>>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;
```

```sql
CREATE TRIGGER last_updated
    BEFORE UPDATE
    ON public.film
    FOR EACH ROW
    EXECUTE PROCEDURE public.last_updated();
```

>>>>>>>>>>>>>>>>>>>CREATING **film_actor** TABLE<<<<<<<<<<<<<<<<

```sql
-- 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.