customer
payment
payment_id
customer_id
staff_id
rental_id
amount
payment_date
customer_list
id
name
address
zip code
phone
city
country
notes
sid
customer
customer_id
store_id
first_name
last_name
email
address_id
activebool
create_date
last_update
active
rental
rental_id
rental_date
inventory_id
customer_id
return_date
staff_id
last_update
inventory
inventory_id
film_id
store_id
last_update
payment_p2007_06
payment_id
customer_id
staff_id
rental_id
amount
payment_date
store
store_id
manager_staff_id
address_id
last_update
staff
staff_list
id
name
address
zip code
phone
city
country
sid
payment_p2007_05
payment_id
customer_id
staff_id
rental_id
amount
payment_date
staff
staff_id
first_name
last_name
address_id
email
store_id
active
username
password
last_update
picture
payment_p2007_01
payment_id
customer_id
staff_id
rental_id
amount
payment_date
payment_p2007_04
payment_id
customer_id
staff_id
rental_id
amount
payment_date
payment_p2007_03
payment_id
customer_id
staff_id
rental_id
amount
payment_date
payment_p2007_02
payment_id
customer_id
staff_id
rental_id
amount
payment_date
nicer_but_slower_film_list
nicer_but_slower_film_list
fid
title
description
category
price
length
rating
actors
sales_by_film_category
category
total_sales
sales_by_store
store
manager
total_sales
address
address
address_id
address
address2
district
city_id
postal_code
phone
last_update
city
city_id
city
country_id
last_update
country
country_id
country
last_update
film
film_list
fid
title
description
category
price
length
rating
actors
film
film_id
title
description
release_year
language_id
original_language_id
rental_duration
rental_rate
length
replacement_cost
rating
last_update
special_features
fulltext
deleted_at
film_actor
actor_id
film_id
last_update
actor
actor_id
first_name
last_name
last_update
actor_info
actor_id
first_name
last_name
film_info
film_category
film_id
category_id
last_update
category
category_id
name
last_update
language
language_id
name
last_update
staff_id
rental_id
customer_id
inventory_id
store_id
address_id
manager_staff_id
film_id
city_id
country_id
actor_id
original_language_id
language_id
category_id

Default Layout

Table actor

* Pk	actor_id	integer
*	first_name	varchar(45)
* Idx	last_name	varchar(45)
*	last_update	timestamp
Indexes		
Pk	actor_pkey	actor_id
	idx_actor_last_name	last_name
Triggers		

Table address

* Pk	address_id	integer
*	address	varchar(50)
	address2	varchar(50)
*	district	varchar(20)
* Idx	city_id	smallint
	postal_code	varchar(10)
*	phone	varchar(20)
*	last_update	timestamp
Indexes		
Pk	address_pkey	address_id
	idx_fk_city_id	city_id
Foreign Keys		
	address_city_id_fkey (city_id) ref city (city_id)	
Triggers		

Table category

* Pk	category_id	integer
*	name	varchar(25)
*	last_update	timestamp
Indexes		
Pk	category_pkey	category_id
Triggers		

Table city

* Pk	city_id	integer
*	city	varchar(50)
* Idx	country_id	smallint
*	last_update	timestamp
Indexes		
Pk	city_pkey	city_id
	idx_fk_country_id	country_id
Foreign Keys		
	city_country_id_fkey (country_id) ref country (country_id)	
Triggers		

Table country

* Pk	country_id	integer
*	country	varchar(50)
*	last_update	timestamp
Indexes		

Table country

Pk	country_pkey	country_id
Triggers		

Table customer

* Pk	customer_id	integer
* Idx	store_id	smallint
*	first_name	varchar(45)
* Idx	last_name	varchar(45)
	email	varchar(50)
* Idx	address_id	smallint
*	activebool	boolean
*	create_date	date
	last_update	timestamp
	active	integer
Indexes		
Pk	customer_pkey	customer_id
	idx_fk_address_id	address_id
	idx_fk_store_id	store_id
	idx_last_name	last_name
Foreign Keys		
	customer_address_id_fkey (address_id) ref address (address_id)	
	customer_store_id_fkey (store_id) ref store (store_id)	
Triggers		

Table film

* Pk	film_id	integer
* Idx	title	varchar(255)
	description	text
	release_year	"public"."year"
* Idx	language_id	smallint
Idx	original_language_id	smallint
*	rental_duration	smallint
*	rental_rate	numeric(4,2)
	length	smallint
*	replacement_cost	numeric(5,2)
	rating	"public".mpaa_rating
*	last_update	timestamp
	special_features	text[]
* Idx	fulltext	tsvector
	deleted_at	timestamp
Indexes		
Pk	film_pkey	film_id
	film_fulltext_idx	fulltext
	idx_fk_language_id	language_id
	idx_fk_original_language_id	original_language_id
	idx_title	title
Foreign Keys		
	film_language_id_fkey (language_id) ref language (language_id)	
	film_original_language_id_fkey (original_language_id) ref language (language_id)	
Triggers		
	film_fulltext_trigger	

Table film_actor

* Pk	actor_id	smallint
* Pk	film_id	smallint
*	last_update	timestamp
Indexes		
Pk	film_actor_pkey	actor_id, film_id
	idx_fk_film_id	film_id
Foreign Keys		
	film_actor_actor_id_fkey (actor_id) ref actor (actor_id)	
	film_actor_film_id_fkey (film_id) ref film (film_id)	
Triggers		

Table film_category

* Pk	film_id	smallint
* Pk	category_id	smallint
*	last_update	timestamp
Indexes		
Pk	film_category_pkey	film_id, category_id
Foreign Keys		
	film_category_category_id_fkey (category_id) ref category (category_id)	
	film_category_film_id_fkey (film_id) ref film (film_id)	
Triggers		

Table inventory

* Pk	inventory_id	integer
* Idx	film_id	smallint
* Idx	store_id	smallint
*	last_update	timestamp
Indexes		
Pk	inventory_pkey	inventory_id
	idx_store_id_film_id	store_id, film_id
Foreign Keys		
	inventory_film_id_fkey (film_id) ref film (film_id)	
	inventory_store_id_fkey (store_id) ref store (store_id)	
Triggers		

Table language

* Pk	language_id	integer
*	name	char(20)
*	last_update	timestamp
Indexes		
Pk	language_pkey	language_id
Triggers		

Table payment

* Pk	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer
*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
Pk	payment_pkey	payment_id
	idx_fk_customer_id	customer_id

Table payment

	idx_fk_staff_id	staff_id
Foreign Keys		
	payment_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_staff_id_fkey (staff_id) ref staff (staff_id)	

Table payment_p2007_01

*	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer
*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
	idx_fk_payment_p2007_01_customer_id	customer_id
	idx_fk_payment_p2007_01_staff_id	staff_id
Foreign Keys		
	payment_p2007_01_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_p2007_01_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_p2007_01_staff_id_fkey (staff_id) ref staff (staff_id)	
Constraints		
	payment_p2007_01_payment_date_check	CHECK (((payment_date >= '2007-01-01 00:00:00'::timestamp without time zone) AND (payment_date < '2007-02-01 00:00:00'::timestamp without time zone)))

Table payment_p2007_02

*	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer
*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
	idx_fk_payment_p2007_02_customer_id	customer_id
	idx_fk_payment_p2007_02_staff_id	staff_id
Foreign Keys		
	payment_p2007_02_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_p2007_02_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_p2007_02_staff_id_fkey (staff_id) ref staff (staff_id)	
Constraints		
	payment_p2007_02_payment_date_check	CHECK (((payment_date >= '2007-02-01 00:00:00'::timestamp without time zone) AND (payment_date < '2007-03-01 00:00:00'::timestamp without time zone)))

Table payment_p2007_03

*	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer
*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
	idx_fk_payment_p2007_03_customer_id	customer_id

Table payment_p2007_03

	idx_fk_payment_p2007_03_staff_id	staff_id
Foreign Keys		
	payment_p2007_03_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_p2007_03_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_p2007_03_staff_id_fkey (staff_id) ref staff (staff_id)	
Constraints		
	payment_p2007_03_payment_date_check	CHECK (((payment_date >= '2007-03-01 00:00:00'::timestamp without time zone) AND (payment_date < '2007-04-01 00:00:00'::timestamp without time zone)))

Table payment_p2007_04

*	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer
*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
	idx_fk_payment_p2007_04_customer_id	customer_id
	idx_fk_payment_p2007_04_staff_id	staff_id
Foreign Keys		
	payment_p2007_04_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_p2007_04_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_p2007_04_staff_id_fkey (staff_id) ref staff (staff_id)	
Constraints		
	payment_p2007_04_payment_date_check	CHECK (((payment_date >= '2007-04-01 00:00:00'::timestamp without time zone) AND (payment_date < '2007-05-01 00:00:00'::timestamp without time zone)))

Table payment_p2007_05

*	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer
*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
	idx_fk_payment_p2007_05_customer_id	customer_id
	idx_fk_payment_p2007_05_staff_id	staff_id
Foreign Keys		
	payment_p2007_05_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_p2007_05_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_p2007_05_staff_id_fkey (staff_id) ref staff (staff_id)	
Constraints		
	payment_p2007_05_payment_date_check	CHECK (((payment_date >= '2007-05-01 00:00:00'::timestamp without time zone) AND (payment_date < '2007-06-01 00:00:00'::timestamp without time zone)))

Table payment_p2007_06

*	payment_id	integer
* Idx	customer_id	smallint
* Idx	staff_id	smallint
*	rental_id	integer

Table payment_p2007_06

*	amount	numeric(5,2)
*	payment_date	timestamp
Indexes		
	idx_fk_payment_p2007_06_customer_id	customer_id
	idx_fk_payment_p2007_06_staff_id	staff_id
Foreign Keys		
	payment_p2007_06_customer_id_fkey (customer_id) ref customer (customer_id)	
	payment_p2007_06_rental_id_fkey (rental_id) ref rental (rental_id)	
	payment_p2007_06_staff_id_fkey (staff_id) ref staff (staff_id)	
Constraints		
	payment_p2007_06_payment_date_check	CHECK (((payment_date >= '2007-06-01 00:00:00'::timestamp without time zone) AND (payment_date < '2007-07-01 00:00:00'::timestamp without time zone)))

Table rental

* Pk	rental_id	integer
* Unq	rental_date	timestamp
* Unq	inventory_id	integer
* Unq	customer_id	smallint
	return_date	timestamp
*	staff_id	smallint
*	last_update	timestamp
Indexes		
Pk	rental_pkey	rental_id
Unq	idx_unq_rental_rental_date_inventory_id_customer_id	rental_date, inventory_id, customer_id
	idx_fk_inventory_id	inventory_id
Foreign Keys		
	rental_customer_id_fkey (customer_id) ref customer (customer_id)	
	rental_inventory_id_fkey (inventory_id) ref inventory (inventory_id)	
	rental_staff_id_fkey (staff_id) ref staff (staff_id)	
Triggers		

Table staff

* Pk	staff_id	integer
*	first_name	varchar(45)
*	last_name	varchar(45)
*	address_id	smallint
	email	varchar(50)
*	store_id	smallint
*	active	boolean
*	username	varchar(16)
	password	varchar(40)
*	last_update	timestamp
	picture	bytea
Indexes		
Pk	staff_pkey	staff_id
Foreign Keys		
	staff_address_id_fkey (address_id) ref address (address_id)	
	staff_store_id_fkey (store_id) ref store (store_id)	
Triggers		

Table store

* Pk	store_id	integer

Table store

* Unq	manager_staff_id	smallint
*	address_id	smallint
*	last_update	timestamp
Indexes		
Pk	store_pkey	store_id
Unq	idx_unq_manager_staff_id	manager_staff_id
Foreign Keys		
	store_address_id_fkey (address_id) ref address (address_id)	
	store_manager_staff_id_fkey (manager_staff_id) ref staff (staff_id)	
Triggers		

View actor_info

```
CREATE VIEW ${schema}.${name} AS  SELECT a.actor_id,
    a.first_name,
    a.last_name,
    group_concat(DISTINCT (((c.name)::text || ': '::text) || ( SELECT group_concat((f.title)::text) AS group_concat
           FROM ((film f
             JOIN film_category fc_1 ON ((f.film_id = fc_1.film_id)))
             JOIN film_actor fa_1 ON ((f.film_id = fa_1.film_id)))
          WHERE ((fc_1.category_id = c.category_id) AND (fa_1.actor_id = a.actor_id))
          GROUP BY fa_1.actor_id))) AS film_info
   FROM (((actor a
     LEFT JOIN film_actor fa ON ((a.actor_id = fa.actor_id)))
     LEFT JOIN film_category fc ON ((fa.film_id = fc.film_id)))
     LEFT JOIN category c ON ((fc.category_id = c.category_id)))
  GROUP BY a.actor_id, a.first_name, a.last_name;
```

View customer_list

```
CREATE VIEW ${schema}.${name} AS  SELECT cu.customer_id AS id,
    (((cu.first_name)::text || ' '::text) || (cu.last_name)::text) AS name,
    a.address,
    a.postal_code AS "zip code",
    a.phone,
    city.city,
    country.country,
        CASE
            WHEN cu.activebool THEN 'active'::text
            ELSE ''::text
        END AS notes,
    cu.store_id AS sid
   FROM (((customer cu
     JOIN address a ON ((cu.address_id = a.address_id)))
     JOIN city ON ((a.city_id = city.city_id)))
     JOIN country ON ((city.country_id = country.country_id)));
```

View film_list

```
CREATE VIEW ${schema}.${name} AS  SELECT film.film_id AS fid,
    film.title,
    film.description,
    category.name AS category,
    film.rental_rate AS price,
    film.length,
    film.rating,
    group_concat((((actor.first_name)::text || ' '::text) || (actor.last_name)::text)) AS actors
   FROM ((((category
     LEFT JOIN film_category ON ((category.category_id = film_category.category_id)))
     LEFT JOIN film ON ((film_category.film_id = film.film_id)))
     JOIN film_actor ON ((film.film_id = film_actor.film_id)))
     JOIN actor ON ((film_actor.actor_id = actor.actor_id)))
  GROUP BY film.film_id, film.title, film.description, category.name, film.rental_rate, film.length, film.rating;
```

View nicer_but_slower_film_list

```
CREATE VIEW ${schema}.${name} AS  SELECT film.film_id AS fid,
    film.title,
    film.description,
    category.name AS category,
    film.rental_rate AS price,
    film.length,
    film.rating,
    group_concat((((upper("substring"((actor.first_name)::text, 1, 1)) || lower("substring"((actor.first_name)::text, 2))) || upper("substring"((actor.last_name)::text, 1, 1))) || lower("substring"((actor.last_name)::text, 2)))) AS actors
   FROM ((((category
     LEFT JOIN film_category ON ((category.category_id = film_category.category_id)))
     LEFT JOIN film ON ((film_category.film_id = film.film_id)))
     JOIN film_actor ON ((film.film_id = film_actor.film_id)))
     JOIN actor ON ((film_actor.actor_id = actor.actor_id)))
  GROUP BY film.film_id, film.title, film.description, category.name, film.rental_rate, film.length, film.rating;
```

View sales_by_film_category

```
CREATE VIEW ${schema}.${name} AS  SELECT c.name AS category,
    sum(p.amount) AS total_sales
   FROM (((((payment p
     JOIN rental r ON ((p.rental_id = r.rental_id)))
     JOIN inventory i ON ((r.inventory_id = i.inventory_id)))
     JOIN film f ON ((i.film_id = f.film_id)))
     JOIN film_category fc ON ((f.film_id = fc.film_id)))
     JOIN category c ON ((fc.category_id = c.category_id)))
  GROUP BY c.name
  ORDER BY (sum(p.amount)) DESC;
```

View sales_by_store

```
CREATE VIEW ${schema}.${name} AS  SELECT (((c.city)::text || ','::text) || (cy.country)::text) AS store,
    (((m.first_name)::text || ' '::text) || (m.last_name)::text) AS manager,
    sum(p.amount) AS total_sales
   FROM (((((((payment p
     JOIN rental r ON ((p.rental_id = r.rental_id)))
     JOIN inventory i ON ((r.inventory_id = i.inventory_id)))
     JOIN store s ON ((i.store_id = s.store_id)))
     JOIN address a ON ((s.address_id = a.address_id)))
     JOIN city c ON ((a.city_id = c.city_id)))
     JOIN country cy ON ((c.country_id = cy.country_id)))
     JOIN staff m ON ((s.manager_staff_id = m.staff_id)))
  GROUP BY cy.country, c.city, s.store_id, m.first_name, m.last_name
  ORDER BY cy.country, c.city;
```

View staff_list

```
CREATE VIEW ${schema}.${name} AS  SELECT s.staff_id AS id,
    (((s.first_name)::text || ' '::text) || (s.last_name)::text) AS name,
    a.address,
    a.postal_code AS "zip code",
    a.phone,
    city.city,
    country.country,
    s.store_id AS sid
   FROM (((staff s
     JOIN address a ON ((s.address_id = a.address_id)))
     JOIN city ON ((a.city_id = city.city_id)))
     JOIN country ON ((city.country_id = country.country_id)));
```