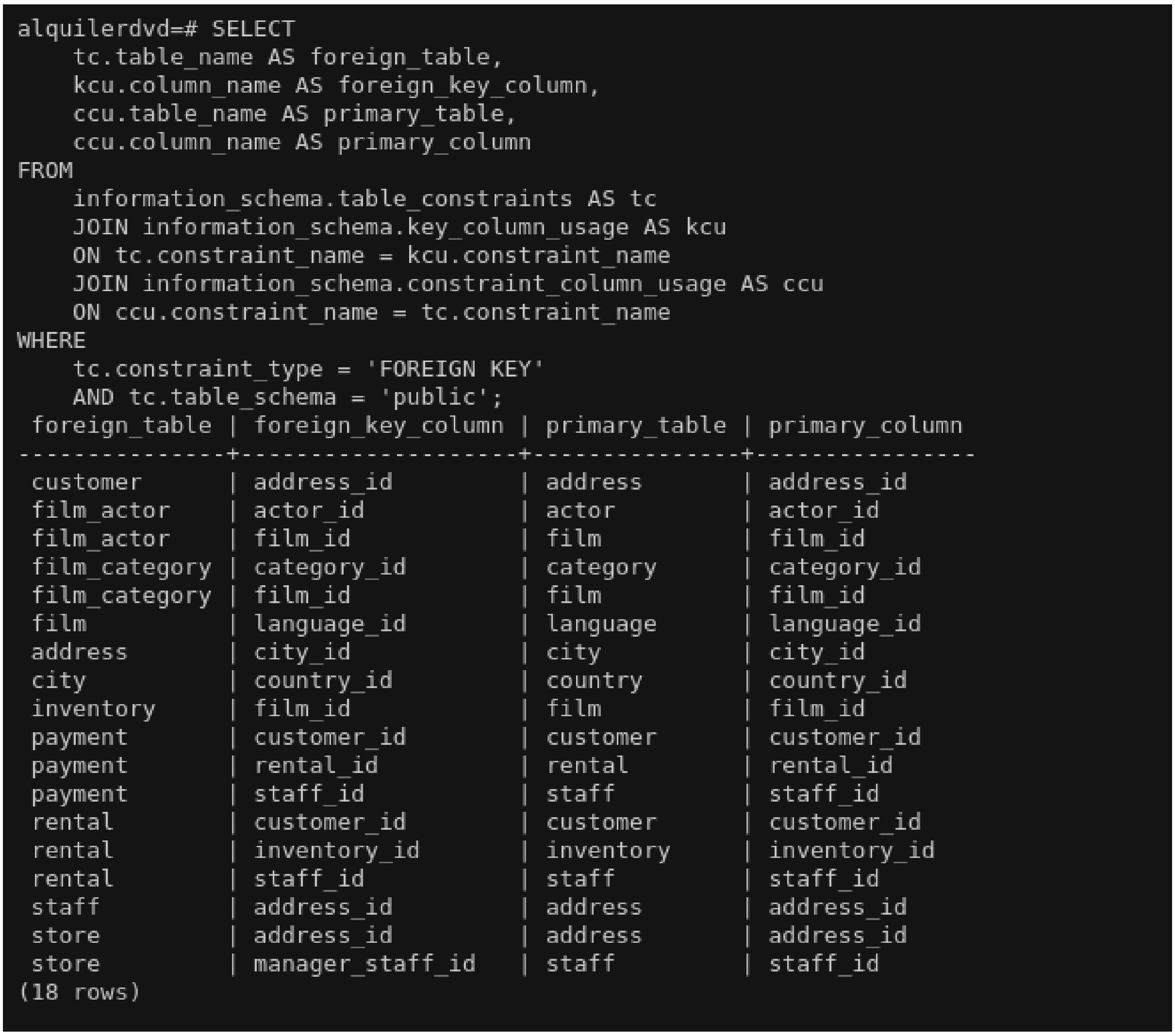
Identificar las tablas principales

Para identificar las tablas principales, tenemos que encontrar aquellas tablas que no dependan de otras, es decir, que no tengan claves foráneas. Para ello, usaremos el comando:

```
SELECT
   table_name,
   column_name,
   data_type,
   is_nullable,
   column_default
FROM
   information_schema.columns
WHERE
   table_schema = 'public'
ORDER BY table_name, ordinal_position;
```



```
alquilerdvd=# SELECT
    tc.table_name AS foreign_table,
    kcu.column_name AS foreign_key_column,
    ccu.table_name AS primary_table,
    ccu.column_name AS primary_column
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
    ON tc.constraint_name = kcu.constraint_name
    JOIN information_schema.constraint_column_usage AS ccu
    ON ccu.constraint_name = tc.constraint_name
WHERE
    tc.constraint_type = 'FOREIGN KEY'
    AND tc.table_schema = 'public';
 foreign_table | foreign_key_column | primary_table | primary_column
---------------+--------------------+---------------+----------------
 customer      | address_id         | address       | address_id
 film_actor    | actor_id           | actor         | actor_id
 film_actor    | film_id            | film          | film_id
 film_category | category_id        | category      | category_id
 film_category | film_id            | film          | film_id
 film          | language_id        | language      | language_id
 address       | city_id            | city          | city_id
 city          | country_id         | country       | country_id
 inventory     | film_id            | film          | film_id
 payment       | customer_id        | customer      | customer_id
 payment       | rental_id          | rental        | rental_id
 payment       | staff_id           | staff         | staff_id
 rental        | customer_id        | customer      | customer_id
 rental        | inventory_id       | inventory     | inventory_id
 rental        | staff_id           | staff         | staff_id
 staff         | address_id         | address       | address_id
 store         | address_id         | address       | address_id
 store         | manager_staff_id   | staff         | staff_id
(18 rows)
```

De la siguiente consulta, escogeremos las tablas que aparezca en primary_table pero no en foreign_table. En cuyo caso, serían las siguientes 5 tablas:

actor
category
country
language
film

Claves primarias

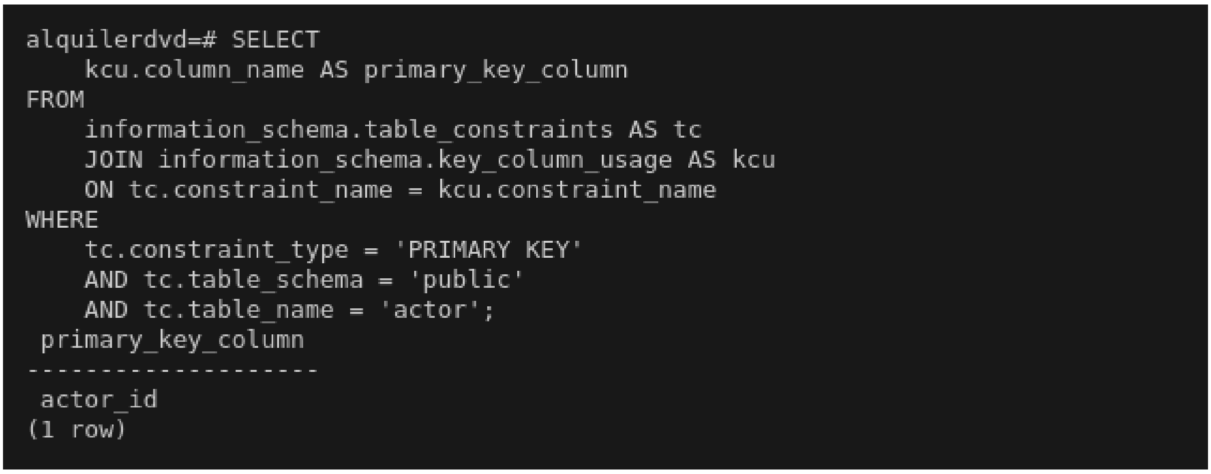
Se pueden obtener de dos formas:

1. Ejecutando el siguiente comando para filtrar la clave primaria:

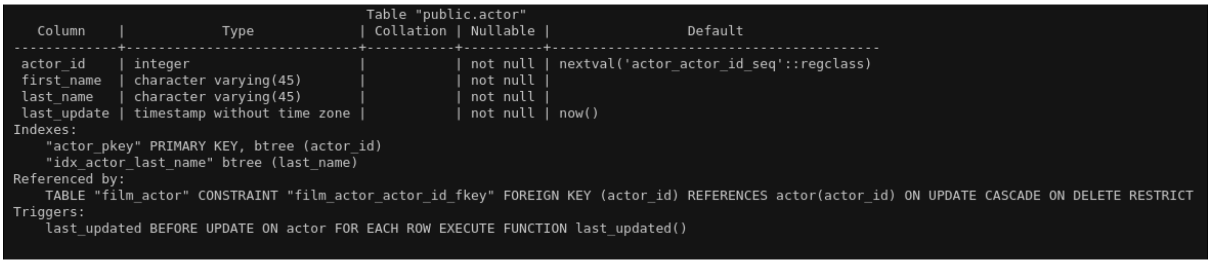
```
SELECT
   kcu.column_name AS primary_key_column
FROM
   information_schema.table_constraints AS tc
   JOIN information_schema.key_column_usage AS kcu
   ON tc.constraint_name = kcu.constraint_name
WHERE
   tc.constraint_type = 'PRIMARY KEY'
   AND tc.table_schema = 'public'
   AND tc.table_name = 'NOMBRE_TABLA';
```

2. Ejecutar el comando \d NOMBRE_TABLA, que nos devuelve un resumen de la tabla y ahí identificar sus claves primarias

Tabla actor - Clave primaria: actor_id
Capturas de los comandos:

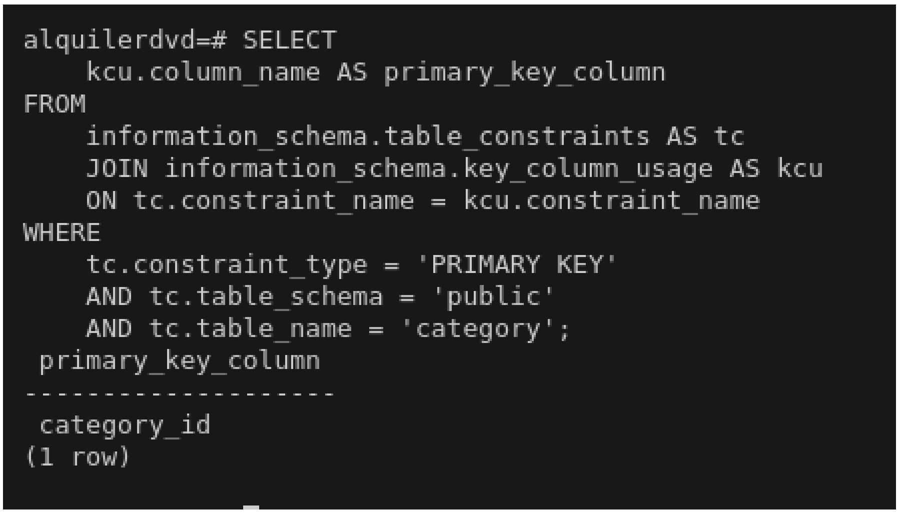


```
alquilerdvd=# SELECT
    kcu.column_name AS primary_key_column
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
    ON tc.constraint_name = kcu.constraint_name
WHERE
    tc.constraint_type = 'PRIMARY KEY'
    AND tc.table_schema = 'public'
    AND tc.table_name = 'actor';
 primary_key_column
--------------------
 actor_id
(1 row)
```

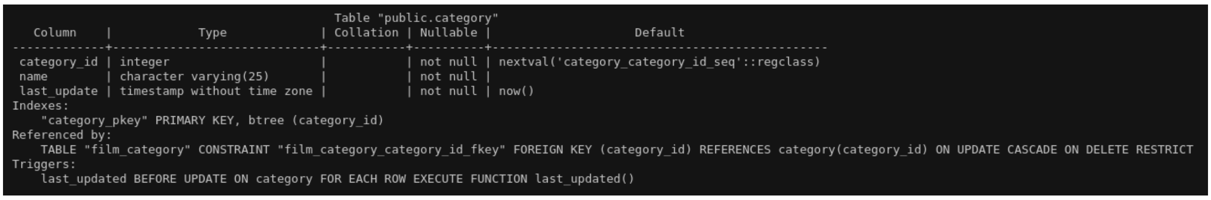


```
                                    Table "public.actor"
   Column    |            Type             | Collation | Nullable |                 Default
-------------+-----------------------------+-----------+----------+-----------------------------------------
 actor_id    | integer                     |           | not null | nextval('actor_actor_id_seq'::regclass)
 first_name  | character varying(45)       |           | not null |
 last_name   | character varying(45)       |           | not null |
 last_update | timestamp without time zone |           | not null | now()
Indexes:
    "actor_pkey" PRIMARY KEY, btree (actor_id)
    "idx_actor_last_name" btree (last_name)
Referenced by:
    TABLE "film_actor" CONSTRAINT "film_actor_actor_id_fkey" FOREIGN KEY (actor_id) REFERENCES actor(actor_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    last_updated BEFORE UPDATE ON actor FOR EACH ROW EXECUTE FUNCTION last_updated()
```

Tabla category - Clave primaria:
Capturas de los comandos:

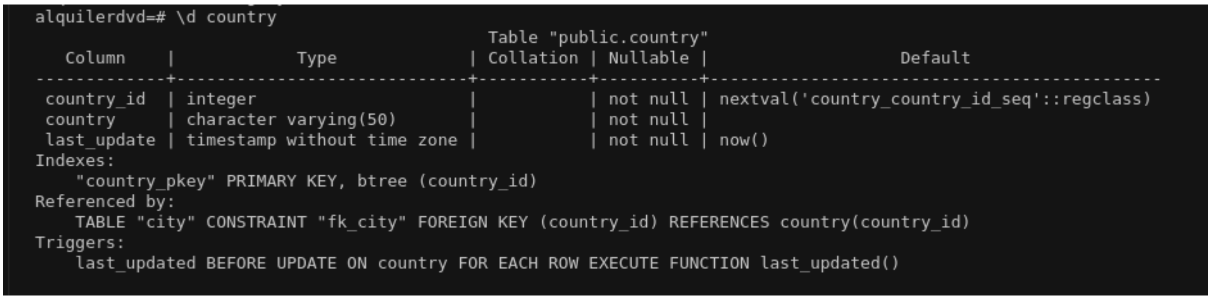


```
alquilerdvd=# SELECT
    kcu.column_name AS primary_key_column
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
    ON tc.constraint_name = kcu.constraint_name
WHERE
    tc.constraint_type = 'PRIMARY KEY'
    AND tc.table_schema = 'public'
    AND tc.table_name = 'category';
 primary_key_column
--------------------
 category_id
(1 row)
```

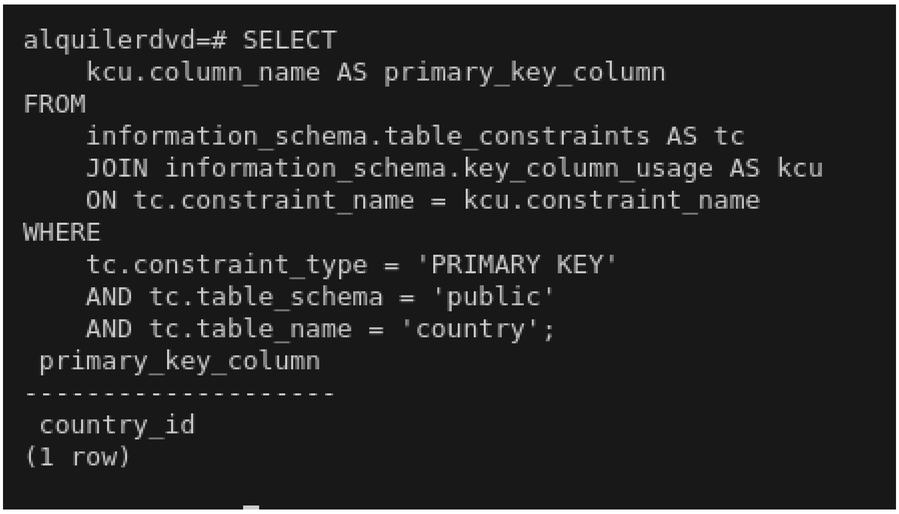


```
                                     Table "public.category"
   Column    |            Type             | Collation | Nullable |                    Default
-------------+-----------------------------+-----------+----------+-----------------------------------------------
 category_id | integer                     |           | not null | nextval('category_category_id_seq'::regclass)
 name        | character varying(25)       |           | not null |
 last_update | timestamp without time zone |           | not null | now()
Indexes:
    "category_pkey" PRIMARY KEY, btree (category_id)
Referenced by:
    TABLE "film_category" CONSTRAINT "film_category_category_id_fkey" FOREIGN KEY (category_id) REFERENCES category(category_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    last_updated BEFORE UPDATE ON category FOR EACH ROW EXECUTE FUNCTION last_updated()
```

Tabla country - Clave primaria:
Capturas de los comandos:

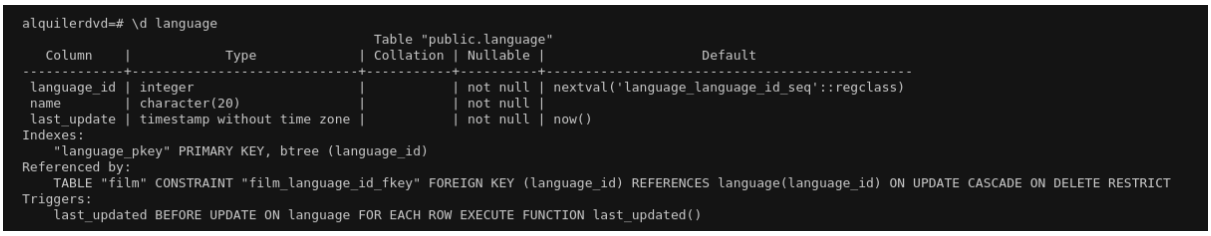


```
alquilerdvd=# \d country
                                      Table "public.country"
   Column    |            Type             | Collation | Nullable |                   Default
-------------+-----------------------------+-----------+----------+---------------------------------------------
 country_id  | integer                     |           | not null | nextval('country_country_id_seq'::regclass)
 country     | character varying(50)       |           | not null |
 last_update | timestamp without time zone |           | not null | now()
Indexes:
    "country_pkey" PRIMARY KEY, btree (country_id)
Referenced by:
    TABLE "city" CONSTRAINT "fk_city" FOREIGN KEY (country_id) REFERENCES country(country_id)
Triggers:
    last_updated BEFORE UPDATE ON country FOR EACH ROW EXECUTE FUNCTION last_updated()
```

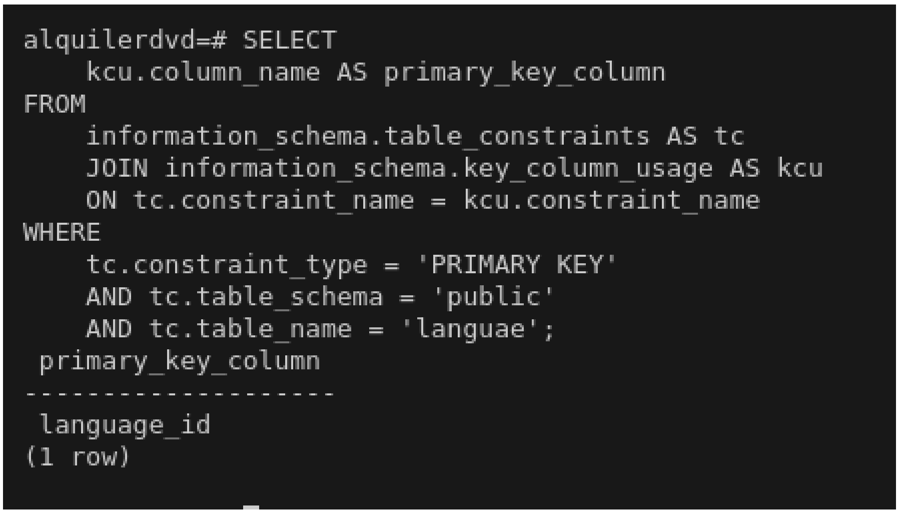


```
alquilerdvd=# SELECT
    kcu.column_name AS primary_key_column
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
    ON tc.constraint_name = kcu.constraint_name
WHERE
    tc.constraint_type = 'PRIMARY KEY'
    AND tc.table_schema = 'public'
    AND tc.table_name = 'country';
 primary_key_column
--------------------
 country_id
(1 row)
```

Tabla language - Clave primaria:
Capturas de los comandos:

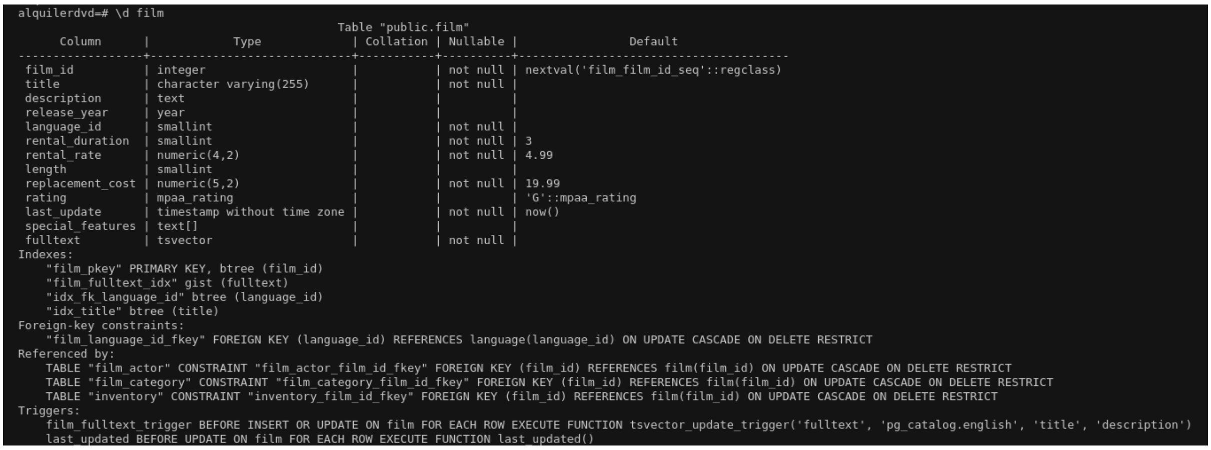


```
alquilerdvd=# \d language
                                      Table "public.language"
   Column    |            Type             | Collation | Nullable |                    Default
-------------+-----------------------------+-----------+----------+-----------------------------------------------
 language_id | integer                     |           | not null | nextval('language_language_id_seq'::regclass)
 name        | character(20)               |           | not null |
 last_update | timestamp without time zone |           | not null | now()
Indexes:
    "language_pkey" PRIMARY KEY, btree (language_id)
Referenced by:
    TABLE "film" CONSTRAINT "film_language_id_fkey" FOREIGN KEY (language_id) REFERENCES language(language_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    last_updated BEFORE UPDATE ON language FOR EACH ROW EXECUTE FUNCTION last_updated()
```

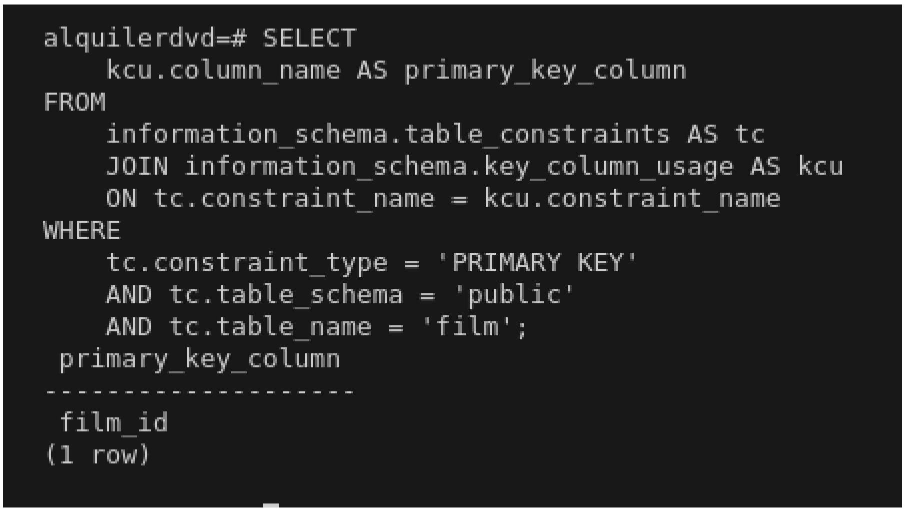


```
alquilerdvd=# SELECT
    kcu.column_name AS primary_key_column
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
    ON tc.constraint_name = kcu.constraint_name
WHERE
    tc.constraint_type = 'PRIMARY KEY'
    AND tc.table_schema = 'public'
    AND tc.table_name = 'languae';
 primary_key_column
--------------------
 language_id
(1 row)
```

Tabla film - Clave primaria: film_id
Capturas de los comandos:



```
alquilerdvd=# \d film
                                             Table "public.film"
      Column      |            Type             | Collation | Nullable |                Default
------------------+-----------------------------+-----------+----------+---------------------------------------
 film_id          | integer                     |           | not null | nextval('film_film_id_seq'::regclass)
 title            | character varying(255)      |           | not null |
 description      | text                        |           |          |
 release_year     | year                        |           |          |
 language_id      | smallint                    |           | not null |
 rental_duration  | smallint                    |           | not null | 3
 rental_rate      | numeric(4,2)                |           | not null | 4.99
 length           | smallint                    |           |          |
 replacement_cost | numeric(5,2)                |           | not null | 19.99
 rating           | mpaa_rating                 |           |          | 'G'::mpaa_rating
 last_update      | timestamp without time zone |           | not null | now()
 special_features | text[]                      |           |          |
 fulltext         | tsvector                    |           | not null |
Indexes:
    "film_pkey" PRIMARY KEY, btree (film_id)
    "film_fulltext_idx" gist (fulltext)
    "idx_fk_language_id" btree (language_id)
    "idx_title" btree (title)
Foreign-key constraints:
    "film_language_id_fkey" FOREIGN KEY (language_id) REFERENCES language(language_id) ON UPDATE CASCADE ON DELETE RESTRICT
Referenced by:
    TABLE "film_actor" CONSTRAINT "film_actor_film_id_fkey" FOREIGN KEY (film_id) REFERENCES film(film_id) ON UPDATE CASCADE ON DELETE RESTRICT
    TABLE "film_category" CONSTRAINT "film_category_film_id_fkey" FOREIGN KEY (film_id) REFERENCES film(film_id) ON UPDATE CASCADE ON DELETE RESTRICT
    TABLE "inventory" CONSTRAINT "inventory_film_id_fkey" FOREIGN KEY (film_id) REFERENCES film(film_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    film_fulltext_trigger BEFORE INSERT OR UPDATE ON film FOR EACH ROW EXECUTE FUNCTION tsvector_update_trigger('fulltext', 'pg_catalog.english', 'title', 'description')
    last_updated BEFORE UPDATE ON film FOR EACH ROW EXECUTE FUNCTION last_updated()
```



```
alquilerdvd=# SELECT
    kcu.column_name AS primary_key_column
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
    ON tc.constraint_name = kcu.constraint_name
WHERE
    tc.constraint_type = 'PRIMARY KEY'
    AND tc.table_schema = 'public'
    AND tc.table_name = 'film';
 primary_key_column
--------------------
 film_id
(1 row)
```