--film dimension

select r.film\_id as film\_pk,

r.title,

r.release\_year,

l.name::varchar as language,

c.name as category,

r.rental\_duration,

r.rental\_rate::numeric,

r.length,

r.replacement\_cost::numeric,

r.rating::varchar

from film r

join language l using(language\_id)

join film\_category using(film\_id)

join category c using(category\_id)

--store dimension

select

store\_id as store\_pk,

c.city ||', '||ctry.country||' store' as store\_name

from store

join address using(address\_id)

join city c using(city\_id)

join country ctry using(country\_id)

--territory dimension

select city\_id as territory\_pk, city, c.country from city

join country c using(country\_id)

--hour dimension

select row\_number() over(order by bb)::int hour\_pk, bb as hour from

(select \* from generate\_series(0,23)bb)x

--date dimension

select replace(dt::varchar,'-','')::bigint as date\_pk, dt as date from(

select '2005-01-01'::date + sequence.day as dt from

generate\_series(0,1100) as sequence(day)) x order by date

--customer dimension

Imported from mockaroo csv file

---insert for rental hour view

insert into core.rental\_date\_view

select rental\_date\_view\_fk from core.sales\_fact\_t

order by rental\_date

---insert for rental hour view

insert into core.rental\_hour\_view

select rental\_hour\_view\_fk from core.sales\_fact\_t

order by rental\_date

---insert for return date view

insert into core.return\_date\_view

select return\_date\_view\_fk, case when return\_date\_view\_fk = 999 then 'Yes' else 'No'

end as return\_date\_nulls

from core.sales\_fact\_t

order by rental\_date

---insert for return hour view

insert into core.return\_hour\_view

select return\_hour\_view\_fk, case when return\_hour\_view\_fk = 999 then 'Yes' else 'No'

end as return\_hour\_nulls

from core.sales\_fact\_t

order by rental\_date

---insert for paymemnt date view

insert into core.payment\_date\_view

select payment\_date\_view\_fk, case when payment\_date\_view\_fk = 999 then 'Yes' else 'No'

end as payment\_date\_nulls

from core.sales\_fact\_t

order by rental\_date

---insert for payment hour view

insert into core.payment\_hour\_view

select payment\_hour\_view\_fk, case when payment\_hour\_view\_fk = 999 then 'Yes' else 'No'

end as payment\_hour\_nulls

from core.sales\_fact\_t

order by rental\_date

Rental fact table

select distinct rental\_id,

f.film\_id as film\_fk,

replace(date(rental\_date)::varchar,'-','')::bigint as rental\_date\_view\_fk,

extract(hour from rental\_date)::int + 1 as rental\_hour\_view\_fk,

rental\_date,

coalesce(replace(date(return\_date)::varchar,'-','')::bigint,999) as return\_date\_view\_fk,

coalesce(extract(hour from return\_date)::int + 1,999) as return\_date\_hour\_fk,

return\_date,

coalesce(replace(date(p.payment\_date)::varchar,'-','')::bigint,999) as payment\_date\_view\_fk,

coalesce(extract(hour from p.payment\_date)::int + 1,999) as payment\_hour\_view\_fk,

p.payment\_date,

r.customer\_id::int as customer\_fk,

c.city\_id as territory\_fk, ---using city\_id as customer\_territory\_fk

s.store\_id as store\_fk,

p.amount as purchase\_price

from rental r

join customer using(customer\_id)

join address using(address\_id)

join city c using(city\_id)

join store s using(store\_id)

join inventory using(inventory\_id)

join film f using(film\_id)

left join payment p using(rental\_id)

order by rental\_date