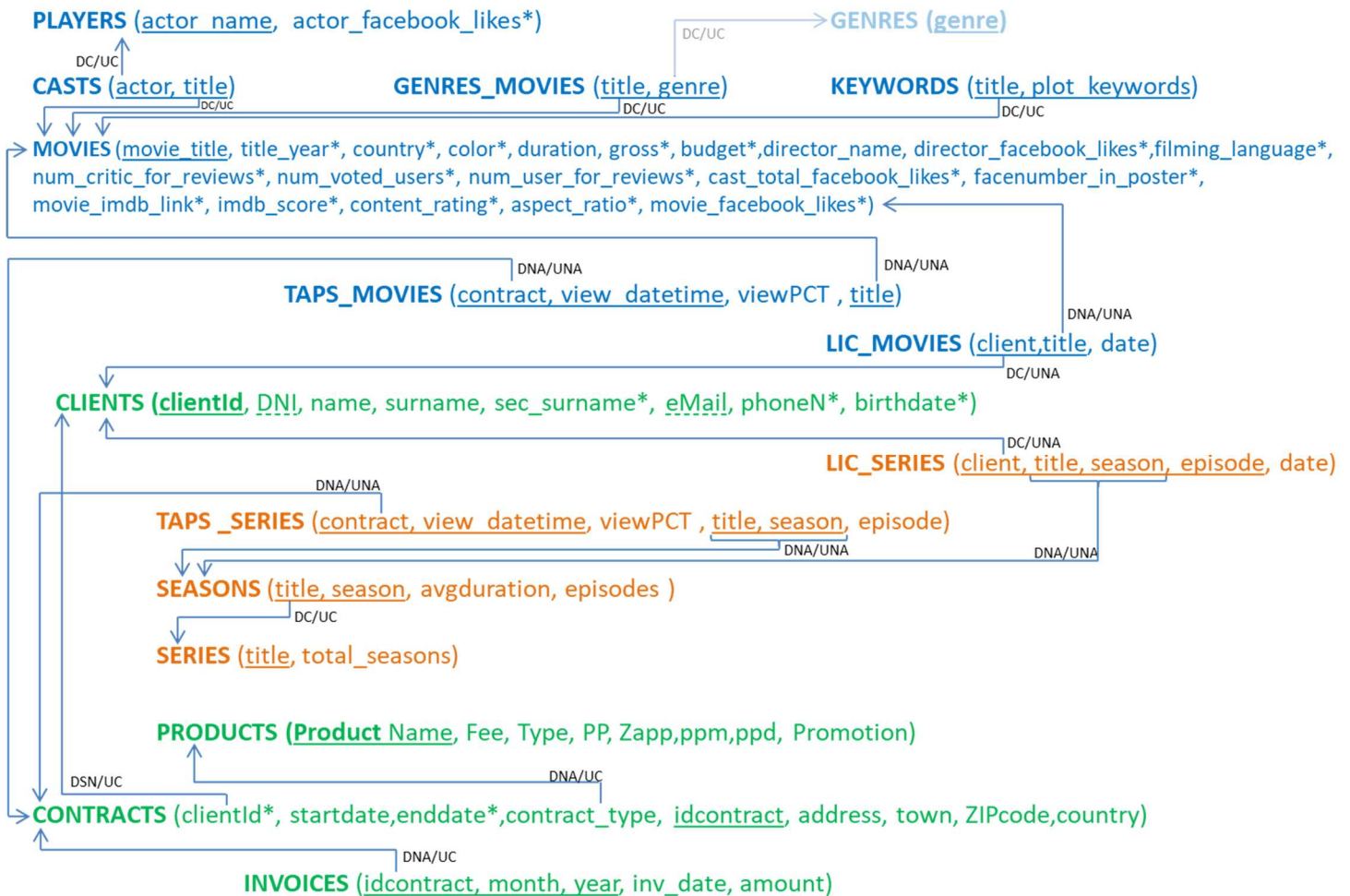


Sea el siguiente grafo relacional:



Resuelve las siguientes consultas en Álgebra Relacional y SQL:

σ	π	ρ	\equiv	\cup	\cap	$-$	\times	θ	$*$	$ *$	$* $	$ * $	$ *$	$* $	$ >$	$< $	\mathcal{G}	\mathcal{T}	$\overline{\mathcal{T}}$	\div
----------	-------	--------	----------	--------	--------	-----	----------	----------	-----	------	------	-------	------	------	------	------	---------------	---------------	--------------------------	--------

Consultas Analíticas:

1. Top 5 de artistas involucrados en más películas estadounidenses (USA).

$$C \equiv \sigma_{\text{country} = 'USA'}(\text{MOVIES})$$

$$Q \equiv \sigma_{\text{rownum} \leq 5} \bar{T}_{\text{count}('x')} \pi_{\text{actor}} G_{\text{actor}}(\text{CAST} *_{\text{title}} C)$$

2. Top-Star: actor/actriz con más visualizaciones (sumando todas las de sus películas) para cada mes.

$$A \equiv \pi_{\text{actor, mes, count}('x')} \text{ cuenta } G_{\text{actor, mes}}(\text{CAST} *_{\text{title}} \text{TAP_MOVIES})$$

$$B \equiv \pi_{\text{mes, max(cuenta) tope}} G_{\text{mes}}(A)$$

$$Q \equiv \pi_{\text{actor, mes, cuenta}} (A *_{A.\text{mes}=B.\text{mes} \text{ AND } A.\text{cuenta}=B.\text{tope}} B)$$

3. Películas con más *Facebook likes* que la media de *likes* de sus estrellas

$$A \equiv \pi_{\text{title, AVG(facebook_likes) as likes}} G_{\text{title}}(\text{CASTS} *_{\text{actor}} \text{PLAYERS})$$

$$Q \equiv \pi_{\text{title}} \sigma_{\text{likes} < \text{movie_facebook_likes}} (A *_{\text{title}} \text{MOVIES})$$

o bien

$$A \equiv \pi_{\text{title, AVG(facebook_likes) as likes}} G_{\text{title}}(\text{CASTS} *_{\text{actor}} \text{PLAYERS})$$

$$Q \equiv \pi_{\text{title}} (A \theta_{A.\text{title}=M.\text{title} \text{ AND } A.\text{likes} < M.\text{movie_facebook_likes}} \text{MOVIES } M)$$

4. Películas con más *Facebook likes* que la media de *likes* de sus estrellas y su director

$$A \equiv \pi_{\text{title, SUM(facebook_likes) as likes, COUNT('C') as cuenta}} G_{\text{title}}(\text{CASTS} *_{\text{actor}} \text{PLAYERS})$$

$$Q \equiv \pi_{\text{title}} \sigma_{((\text{likes} + \text{director_facebook_likes}) / (\text{cuenta} + 1)) < \text{movie_facebook_likes}} (A *_{\text{title}} \text{MOVIES})$$

5. Bacalá: top 5 de películas abandonadas (interrumpidas) después de ver la mitad o más de su metraje.

$ABANDON \equiv \pi_{title} \sigma_{viewPCT > 50 \text{ and } viewPCT < 97} (TAP_MOVIES)$

$Q \equiv \sigma_{rownum < 6} \bar{T}_{count('x')} \pi_{title, count('x')} G_{title} (ABANDON)$

6. Series-lovers: clientes con licencia para una serie completa (cliente y sus serie/s)

$EPI_TOTAL \equiv \pi_{title, sum(episodes) \text{ as total}} G_{title} (SEASONS)$

$EPI_COMPRA \equiv \pi_{client, title, count('x') \text{ as comprados}} G_{client, title} (LIC_SERIES)$

$Q \equiv \pi_{client, title} (EPI_COMPRA \Theta_{comprados = total \text{ AND } title = title} EPI_TOTAL)$

7. Top-Content: para cada mes, indicar que 'contenido' ha tenido mayor cantidad de visualizaciones.

$VISUAL_S \equiv \pi_{to_char(view_datetime, 'YYYY/MM') \text{ mes}, title || season || episode \text{ title}} (TAPS_SERIES)$

$VISUAL_M \equiv \pi_{to_char(view_datetime, 'YYYY/MM') \text{ mes}, title} (TAPS_MOVIES)$

$VISUAL \equiv \pi_{mes, title, count('c') \text{ as total}} G_{mes, title} (VISUAL_S \cup VISUAL_M)$

$MAX_V \equiv \pi_{mes, max(total) \text{ as maximo}} G_{mes} (VISUAL)$

$Q \equiv \pi_{mes, total, title} (MAX_V \Theta_{maximo = total \text{ AND } mes = mes} VISUAL)$

8. Pigeonholed: estrellas con más de la mitad de sus películas (al menos tres) en un género dado; en caso de varios géneros que cumplan esta condición, dar todos.

$ACT_GENS \equiv \pi_{actor, genre, count('x') \text{ AS cuantas}} \sigma_{count('x') >= 3} G_{actor, genre} (CASTS *_{title} MOVIES_GENRE)$

$ACT_TOT \equiv \pi_{actor, count('x') \text{ AS total}} G_{actor} (CASTS)$

$Q \equiv \pi_{actor, genre, (cuantas * 100 / total) \text{ as percentage}} (ACT_TOT \Theta_{actor = actor \text{ AND } cuantas > (total / 2)} ACT_GENS)$

9. Predict-TVseries: para el cliente '*John Doe*', siguiente episodio que probablemente vaya este a ver, que será el siguiente al último que vio; si este era el último de su temporada, se tomaría el que vio inmediatamente antes (saltándose los de la temporada ya completada) y así sucesivamente. .

- Paso 1: contratos de John Doe y episodios que ha visto

$CL_ID \equiv \pi_{clientId} \sigma_{name='John' \text{ and } surname='Doe'} (CLIENTS)$

$CON_ID \equiv \pi_{idcontract} (CONTRACT * CL_ID)$

$VISIO_JD \equiv \pi_{title,season,episode,view_datetime} (TAPS_SERIES * CON_ID)$

- Paso 2: De cada temporada, el último capítulo que ha visto (con la fecha)

$UL_FEC_JD \equiv \pi_{title,season, \max(view_datetime)} G_{title,season} (VISIO_JD)$

$UC \equiv \pi_{title,season,episode,view_datetime} (UL_FEC_JD *_{title,season,view_datetime} VISIO_JD)$

- Paso 3: combinamos lo anterior con el número de episodios de cada temporada. Considerando que ha terminado una serie si ha visto el último capítulo, nos quedamos sólo con aquellos en que el último capítulo visto es menor que el total de episodios de la temporada

$CAND \equiv UC \ominus_{title=title \text{ and } season=season \text{ and } episode < episodes} \pi_{title,season,episodes} (SEASONS)$

- Paso 4: ordenamos por fecha de visualización para elegir el correspondiente a la serie que ha visto más recientemente, y al proyectarlo le sumamos 1 (significa que proponemos ver el siguiente al último episodio visto, episodio+1)

$Q \equiv \pi_{title,season,episode+1} \sigma_{rownum \leq 1} \overline{T}_{view_datetime} (CAND)$

10. Predict-movie: película que se espera que vea el cliente 'John Doe'. Para cualquier cliente, un género tiene una puntuación (número de películas de ese género que ha visto), y una película también (igual a la suma de puntuaciones de los géneros de esa película para ese cliente).

- Paso 1: tomar las películas (distintas) que ha visto ese cliente, y recabamos también las películas que no ha visto ese cliente

$CL_ID \equiv \pi_{clientId} \sigma_{name='John' \text{ and } surname='Doe'}(CLIENTS)$
 $CON_ID \equiv \pi_{idcontract}(CONTRACT * CL_ID)$
 $VIEWED \equiv \pi_{title}(CON_ID * TAP_MOVIES)$
 $NOT_VIEWED \equiv \pi_{title}(MOVIES) - VIEWED$

- Paso 2: calculamos la puntuación de cada género (igual al número de películas de ese género que ha visualizado)

$PT_GENRE \equiv \pi_{genre, count('x') \text{ points}} \mathcal{G}_{genre}(VIEWED *_{title} GENRES_MOVIES)$

- Paso 3: para cada película no vista, calculamos su puntuación sumando las puntuaciones de cada género de esa película

$PT_MOV \equiv \pi_{title, sum(points) \text{ total}} \mathcal{G}_{title}(NOT_VIEWED *_{title} GENRES_MOVIES *_{genre} PT_GENRE)$

- Paso 4: ordenamos por puntuación y nos quedamos con la de mayor puntuación (la primera fila)

$Q \equiv \sigma_{rownum \leq 1} \overline{T}_{total}(PT_MOV)$