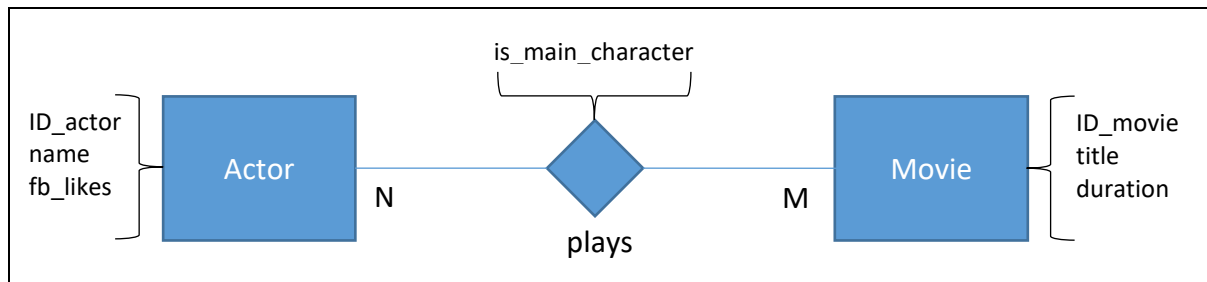


Sessió 6: “Because we all need to believe in movies, sometimes”

En aquesta sessió haureu de crear el model físic a partir dels models conceptual i lògic que es mostren a continuació, per tal de poder realitzar una **importació de dades complexa**.

Model conceptual:



Model lògic (schema):

- **Actor** (ID_actor(PK), name, fb_likes)
- **Plays** (ID_actor(PK/FK), ID_movie(PK/FK), is_main_character)
- **Movie** (ID_movie(PK), title, duration)

Trobareu el fitxer amb les dades corresponents per a realitzar la sessió a l’eStudy. Fixeu-vos que l’estructura d’aquest fitxer la trobareu en el seu HEADER (primera línia del mateix).

Què s’ha de fer:

A partir del model relacional que s’acaba de mostrar, cal fer un script que:

1. Creï les taules del model físic d’aquest model relacional (**amb les PK i FK**).
2. Creï una taula d’importació i hi carregui les dades (tractades) que conté el fitxer `movies.csv`. A l’hora d’importar la informació, és interessant tractar aquella informació necessària.
3. Reparteixi les dades de la taula d’importació per les 3 taules del model físic.

Noteu que el vostre model físic té més d’una taula, mentre que només teniu una taula d’importació. Això significa que cal fer un repartiment d’aquestes dades per les diferents taules. Per a més detalls d’aquest procés, consulteu el manual bàsic que teniu adjunt

Què caldrà entregar?

Caldrà entregar **individualment** l’script `.sql` o `.txt`. Aquest fitxer s’anomenarà de la següent manera: “*Login_S6.sql*”. El no compliment de la nomenclatura suposarà la no correcció de la sessió. No s’acceptaran entregues via e-mail. Recordeu que no podeu modificar cap dels fitxers donats.

Manual bàsic d'importació i repartiment de dades

En sessions anteriors ja s'ha vist com importar informació d'un CSV a una taula en SQL. Per ara, hem après que per poder importar informació, la taula que creem al SGBD ha de tenir la mateixa estructura que el CSV que volem importar.

Tot sovint, però, el nostre model físic tindrà una estructura diferent que la del CSV que volem importar, o bé el CSV contindrà tota la informació de la base de dades agrupada en una sola taula, com és el cas que es proposa en aquesta sessió: un sol fitxer CSV conté tota la informació de pel·lícules i actors.

Després de carregar el fitxer `movies.csv` a una taula d'importació, veurem que cada línia de la taula conté informació sobre la pel·lícula en la que ha participat un actor en concret. Si aquesta pel·lícula té 3 actors o actrius, aquesta apareixerà 3 vegades a la taula, una per cada participant. També veiem que tots els atributs referent a l'actor estan en la mateixa fila, així com tota la informació de la pel·lícula. És per això que necessitem extreure informació d'aquesta taula d'importació per tal d'omplir les taules del vostre model físic: *Actor*, *Plays* i *Movie*.

A continuació s'especifica el seguit de passos que hauríeu de seguir per acabar d'omplir el vostre model físic:

1. Trobar tots els **actors diferents** de la taula d'importació i afegir-los a la taula *Actor*.
2. Trobar totes les **pel·lícules diferents** de la taula d'importació i afegir-les a la taula *Movie*.
3. Omplir la taula *Plays* a partir de la taula d'importació i les dues anteriors taules tal i com s'indica més endavant.
4. Fer Query de comprovació (especificada al final del document).

Trobar diferents ocurrences dins d'una taula (DISTINCT)

Per a trobar els actors i pel·lícules diferents, cal utilitzar l'eina `DISTINCT`:

```
SELECT DISTINCT camp1, camp2 FROM taula;
```

Aquesta sentència retornarà totes que tinguin els camps especificats diferents. Així doncs, no ens mostrarà els repetits.

Inserir les relacions a la taula *Plays*

Per a codificar les relacions que existeixen entre els nostres actors i pel·lícules, caldrà emmagatzemar a la taula *Plays* cadascun dels actors que participa en una pel·lícula, i si aquest és actor principal o no. Tot i així, aquí no emmagatzemarem el títol de la pel·lícula ni el nom de l'actor: **en guardarem el seu PK** (que idealment ha d'haver estat autogenerat).

Per a fer aquesta inserció, necessitem executar una query del següent estil:

```
INSERT INTO taula_relacio (camp_PKFK1, camp_PKFK2, campX)
SELECT taula1.camp_PK_taula1, taula2.camp_PK_taula2, importacio.campX
FROM taula1 NATURAL JOIN taula2 NATURAL JOIN importacio
```

Aquesta query avançada, a partir de dues taules (*taula1* i *taula2*), ens omple la taula de la relació (*taula_relacio*) que relaciona aquestes dues taules amb totes les relacions que codifica la taula d'importació. A més, ens l'omple amb els PK corresponents de *taula1* i *taula2* en comptes de altres camps. Noteu que campX representa un altre camp que hagi d'emmagatzemar la taula de la relació, com podria ser el cas de *is_main_character*.

Sobre els NATURAL JOIN, de moment no necessitem saber el que fan i per ara només els farem servir sense entrar molt en detall en el seu funcionament.

Provar si la sessió s'ha fet correctament

D'aquesta sessió s'espera que un cop feta l'importació, la taula *Actor* tingui 224 files, la taula *Movie* en tingui 100 i la taula *Plays* sigui de 300. Un exemple de com hauria de ser el contingut d'aquestes taules és mostra a continuació:

Actor			Plays			Movie		
id_actor integer	actor_name character varying(255)	facebook_likes integer	id_actor integer	id_movie integer	is_main boolean	id_movie integer	title character varying(255)	duration integer
1	Hugh Jackman	20000	1	94	f	1	Maleficent	97
2	Julie Walters	838	1	58	t	2	Suicide Squad	123
3	Naomi Watts	6000	2	3	f	3	Brave	93
4	Thomas Robinson	604	3	78	t	4	Evan Almighty	96
5	Bruce Greenwood	981	4	49	f	5	X-Men: Apocalypse	144
6	Viola Davis	890	5	99	f	6	47 Ronin	128
7	Michael Fassbender	1200	6	26	t			

Cal realitzar una SOLA Query que mostri que l'actriu Julie Walters va participar a la pel·lícula Brave i no era el personatge principal. S'ha de mostrar el nom del actor, el nom de la pel·lícula i si era personatge principal.