

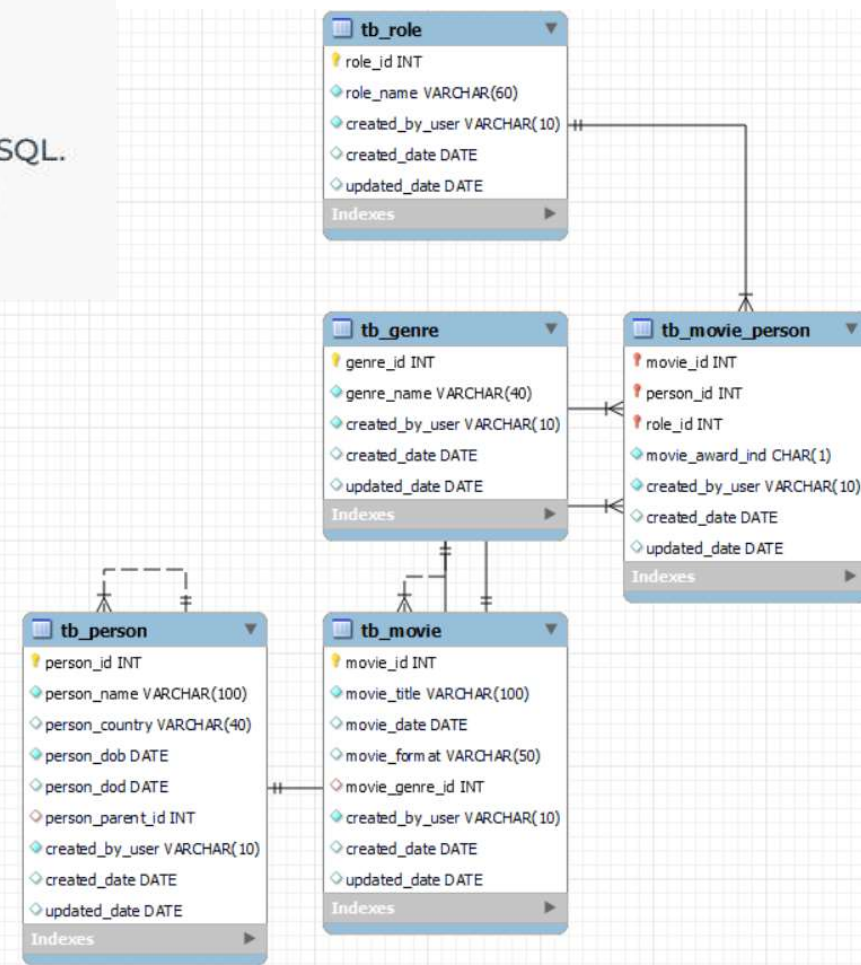
- Exercici 1

A partir dels documents adjunts (estructura i dades), crea una base de dades amb MySQL. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que hi ha.

La base de datos consta de 5 tablas diferentes

```
5 • USE movies;
6 • SHOW tables; -- La base de datos consta de 5 tablas diferentes
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Tables_in_movies			
tb_genre			
tb_movie			
tb_movie_person			
tb_person			
tb_role			



- Key: (Part of) Primary Key
- Filled Diamond: NOT NULL
- Not filled Diamond: NULL
- Red colored: (Part of) Foreign key
- Blue lined Diamond: Simple attribute (no key)

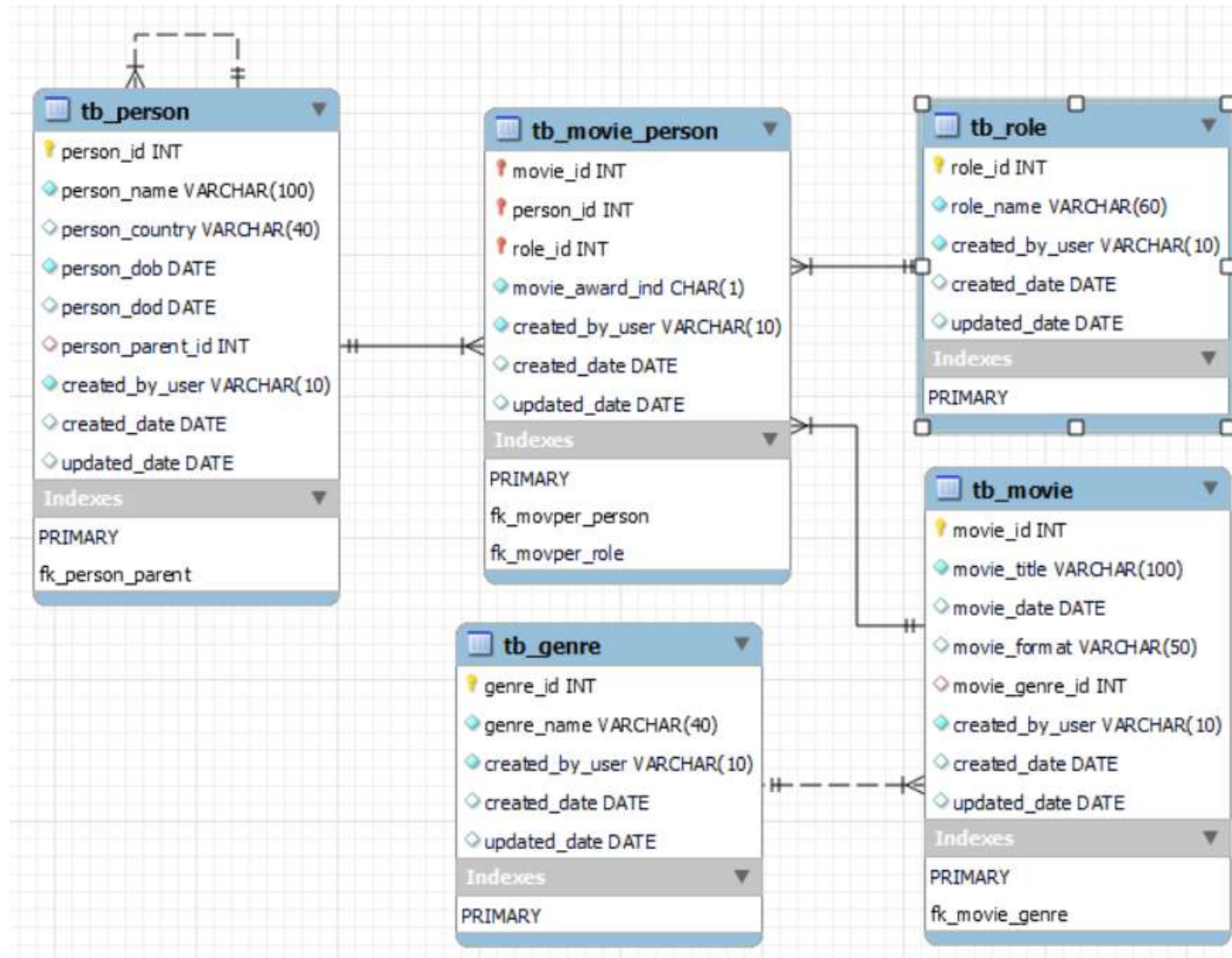
Can be combined for example:

- is a Red colored Key so it's a Primary Key which is also a Foreign Key
- is a Yellow (non Red) Key so it's only a Primary Key
- is a blue lined filled diamond so it's a NOT NULL simple attribute
- is a red colored filled diamond so it's a NOT NULL Foreign Key
- is a blue lined not filled diamond so it's a simple attribute which can be NULL
- is a red colored not filled diamond so it's a Foreign Key which can be NULL

- Ejercicio 1

Aquí podemos ver el diseño gráfico del esquema de la base de datos. Se puede apreciar:

- Las **5 tablas** que componen la base de datos
- Las variables que componen cada tabla
- Las relaciones entre cada tabla:
Ej: **tb_movie_person** esta relacionada con las tablas **tb_movie**, **tb_role** y **tb_person**.
- Las variables que son *PRIMARY KEY* de cada tabla se indican con una llave amarilla.
- La llave naranja significa que además de ser *PRIMARY KEY* son *FOREIGN KEY*



- Ejercicio 1

Tenemos diferentes tipos de variables, como ejemplo nos vamos a fijar en la tabla **tb_genre** :

Field	Type	Null	Key	Default	Extra
genre_id	int	NO	PRI	NULL	
genre_name	varchar(40)	NO		NULL	
created_by_user	varchar(10)	NO		OS_SGAD	
created_date	date	YES		NULL	
updated_date	date	YES		NULL	

- **genre_id** es una variable de tipo numérico sin decimales y además es la variable primaria, identificador único para cada fila de la tabla.
- **genre_name** Es una variable de tipo alfabético de máximo 40 caracteres y no puede estar vacía.
- **created_by_user** Es una variable de tipo alfabético de máximo caracteres y no puede estar vacía. Por defecto, si el usuario no indica nada, aparecerá "OS_SGAD" como valor.
- **created_date** Es una variable de tipo fecha y puede estar vacía.
- **updated_date** Es una variable de tipo fecha y puede estar vacía.

La tabla **tb_genre** consta de 11 entradas que son las siguientes:

```
10 • SELECT * from tb_genre;
```

genre_id	genre_name	created_by_user	created_date	updated_date
1	Acción	OS_SGAD	NULL	NULL
2	Ciencia Ficción	OS_SGAD	NULL	NULL
3	Comedia	OS_SGAD	NULL	NULL
4	Drama	OS_SGAD	NULL	NULL
5	Fantasía	apermag	NULL	NULL
6	Melodrama	apermag	2018-09-01	2018-09-27
7	Musical	OS_SGAD	NULL	NULL
8	Romance	OS_SGAD	NULL	NULL
9	Suspense	OS_SGAD	NULL	NULL
10	Terror	OS_SGAD	NULL	NULL
11	Bélico	OS_SGAD	NULL	NULL
69	Documental	Juanjo	NULL	NULL
NULL	NULL	NULL	NULL	NULL

- Exercici 3

Realitza la següent consulta sobre la base de dades acabada de crear:

Has d'obtenir el nom del gènere i el nombre total de pel·lícules d'aquest gènere i ordenar-ho per ordre descendent de nombre total de pel·lícules.

```
SELECT tb_genre.genre_name, COUNT(*) AS "Total"
FROM movies.tb_movie
LEFT JOIN movies.tb_genre
    ON movies.tb_genre.genre_id = movies.tb_movie.movie_genre_id
GROUP BY tb_movie.movie_genre_id
ORDER BY Total DESC;
```

genre_name	Total
Comedia	4
Acción	2
Ciencia Ficción	2
Drama	2
Suspense	2
Fantasia	1
Romance	1
Terror	1
Bélico	1

- Exercici 4

Realitza la següent consulta sobre la base de dades acabada de crear:

Has d'obtenir, per a cada persona, el seu nom i el nombre màxim de rols diferents que ha tingut en una mateixa pel·lícula.

Posteriorment, mostra únicament aquelles persones que hagin assumit més d'un rol en una mateixa pel·lícula.

-- Has d'obtenir, per a cada persona, el seu nom i el nombre màxim de rols
-- diferents que ha tingut en una mateixa pel·lícula.

```
SELECT tb_person.person_name, COUNT(*) AS Maximo_num_Roles
FROM tb_movie_person
LEFT JOIN tb_person
    ON tb_movie_person.person_id = tb_person.person_id
GROUP BY tb_person.person_name, tb_movie_person.movie_id;
```

person_name	Maximo_num_Roles
Francis Ford Coppola	3
Carmine Coppola	1
Marlon Brando	1
Robert Duvall	1
Martin Sheen	1
Harrison Ford	1
Harrison Ford	1
Harrison Ford	1
George Lucas	1
George Lucas	2
Gary Kurtz	1
Steven Spielberg	1
Steven Spielberg	2
Steven Spielberg	1
Steven Spielberg	1
John Williams	1
Tom Hanks	1
Catherine Zeta-Jones	1
Alfred Joseph Hitchc...	3
Anthony Perkins	1
Vera Miles	1
Emilio Martinez Lazaro	1
Emilio Martinez Lazaro	1
Emilio Martinez Lazaro	1

- Exercici 4

Realitza la següent consulta sobre la base de dades acabada de crear:

Has d'obtenir, per a cada persona, el seu nom i el nombre màxim de rols diferents que ha tingut en una mateixa pel·lícula.

Posteriorment, mostra únicament aquelles persones que hagin assumit més d'un rol en una mateixa pel·lícula.

```
-- Posteriorment, mostra únicament aquelles persones que hagin assumit més  
-- d'un rol en una mateixa pel·lícula.
```

```
SELECT tb_person.person_name, COUNT(*) AS Maximo_num_Roles  
FROM tb_movie_person  
LEFT JOIN tb_person  
    ON tb_movie_person.person_id = tb_person.person_id  
GROUP BY tb_person.person_name, tb_movie_person.movie_id  
    HAVING Maximo_num_Roles > 1;
```

person_name	Maximo_num_Roles
Francis Ford Coppola	3
George Lucas	2
Steven Spielberg	2
Alfred Joseph Hitchcock	3
Daniel Sanchez Arevalo	2
Mel Gibson	2