|  |  |  |
| --- | --- | --- |
| *Diagramme de classes* | *Schéma logique relationnel de données* | |
| 1 - Composition : Une classe est identifiée par rapport à une autre. | | |
|  |  | |
| *Programmation SQL*  La clé primaire est composée de plusieurs champs. Dans les jointures, il faut penser à mettre une ligne de jointure par champ.  Liste des films du CELTIC  Capacité des salles qui projettent Gran Torino  SELECT \*  FROM FILM, SALLE, CINEMA  WHERE FILM.idFilm = PROJETER.idFilm AND PROJETER.idCine= CINEMA.idCine AND nom = ‘Celtic’ ;  SELECT capacite  FROM FILM, PROJETER, SALLE  WHERE FILM.idFilm = PROJETER.idFilm AND PROJETER.idSalle=SALLE.idSalle AND PROJETER.idCine = CINEMA.idCine AND FILM.titre = ‘Grand Torino’ ; | | |
| *Diagramme de classes* | | *Schéma logique relationnel de données* |
| 2 - Plusieurs associations entre 2 mêmes entités | | |
|  | |  |
| *Programmation SQL*  Si, dans une requête, on a besoin 2 fois de la même table (PERSONNE ou FILM) pour faire les jointures, il faut renommer ces tables par des alias de table.  Exemple ici :  Afficher pour chaque film **le nom** des réalisateurs et **le nom** de tous ses acteurs.  SELECT REALISATEUR.nom, ACTEUR.nom  FROM FILM,JOUE, REALISE, PERSONNE AS ACTEUR, PERSONNE AS REALISATEUR  where FILM.idfilm = REALISE.idfilm  and REALISE.id = REALISATEUR.id  and FILM.idFilm=JOUE.idfilm  and JOUE.id = ACTEUR.id ; | | |

|  |  |  |
| --- | --- | --- |
| *Diagramme de classes* | | *Schéma logique relationnel de données* |
| 3 - Association n-aire (ternaire, quaternaire) | | |
|  | |  |
| *Programmation SQL*  Passage au modèle logique : on conserve toujours les mêmes règles. Il peut y avoir n entités reliées par la même association (n rarement supérieur à 3 ou 4).  **Cas typique des HISTORIQUES :**  Schéma typique des associations pour lesquelles il faut garder un historique (associations qui évoluent dans le temps)  La table JOUR ne sera pas conservée dans le modèle physique (dans la base de données).  Afficher pour chaque film **le nom** du film, le nom, le prénom de l’internaute, la date et la note. La liste est triée dans l’ordre alphabétique des films, internautes et date.  SELECT titre, nom, prenom, dte, note  FROM FILM F, EVALUER E, INTERNAUTE I  WHERE F.idFilm = E.idFilm  AND E. email = I.email  ORDER BY titre, I.email, dte ; | | |
| *Diagramme de classes* | *Schéma logique relationnel de données* | |
| 4–Association réflexive (Boucle sur la même classe) | | |
|  |  | |
| *Programmation SQL*  Si, dans une requête, on a besoin 2 fois de la même table (ACTEUR) pour faire les jointures, il faut renommer cette table par un alias de table.  Afficher le nom d’un enfant et celui de son parent.  SELECT ENFANT.nom, ENFANT.prenom, PARENT.nom, PARENT.prenom  FROM ACTEUR AS ENFANT, ACTEUR AS PARENT  WHERE ENFANT.id\_pere = PARENT.id ; | | |