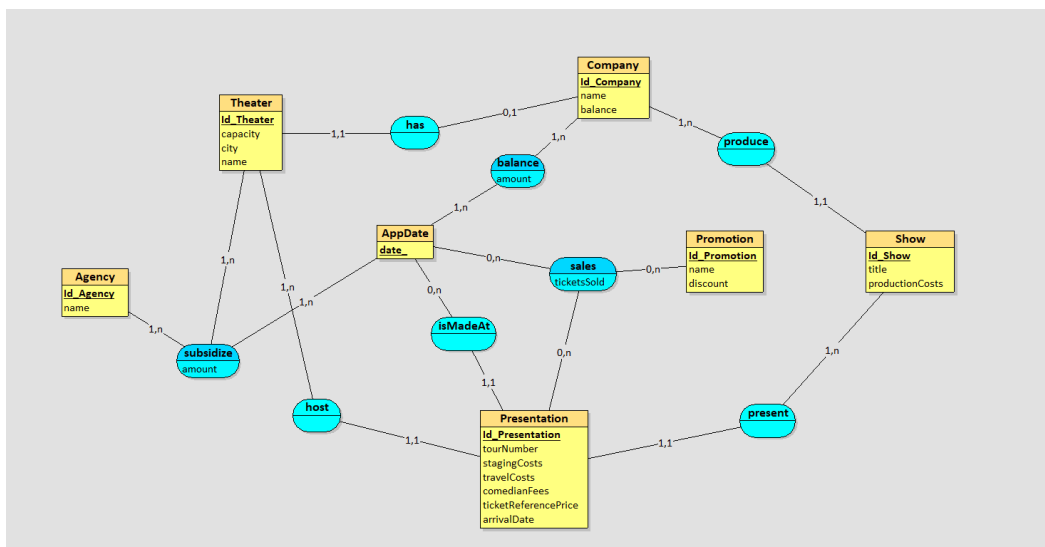


## Advanced Databases: Projet Farès FADILI

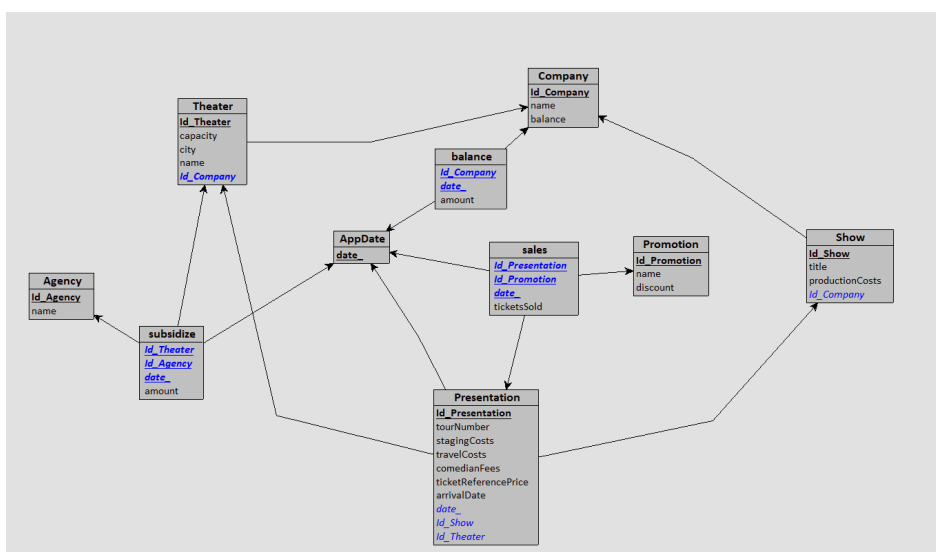
### MCD

Le modèle conceptuel des données (MCD) du projet se présente comme suit. Son élément central est l'entité présentation, qui est reliée à presque tous les autres acteurs de l'écosystème théâtral évoqués dans l'énoncé. Dans un souci de stockage de certaines données, nous avons pris soin d'ajouter des attributs à certaines associations (qui deviendront donc des entités dans le MLD). Nous avons également ajouté une table qui va contenir les dates de chaque jour.



### MLD

Le modèle logique des données (MLD) du projet se présente comme suit. Il intègre les clés étrangères dans son contenu, ce qui facilite la lecture des relations entre les différents acteurs.



## Création

Pour la création de chaque table, nous avons bien explicité chaque contrainte, notamment les clés étrangères, afin de bien appréhender le modèle.

```
CREATE TABLE Show(  
  Id_Show NUMBER,  
  title VARCHAR(50),  
  productionCosts NUMBER,  
  Id_Company INT NOT NULL,  
  CONSTRAINT pk_show PRIMARY KEY(Id_Show),  
  CONSTRAINT fk_company_show FOREIGN KEY(Id_Company) REFERENCES Company(Id_Company)  
);
```

## Insertion

Les insertions ont été effectuées de manière judicieuse afin de satisfaire les différentes problématiques soulevées par l'énoncé.

```
-- Show  
  
Insert into show (Id_Show, title, productionCosts, id_Company)  
values (1, 'Attack on Titan', 1500, 1);  
Insert into show (Id_Show, title, productionCosts, id_Company)  
values (2, 'Hunter x Hunter', 2000, 1);  
Insert into show (Id_Show, title, productionCosts, id_Company)  
values (3, 'Naruto', 2500, 3);  
Insert into show (Id_Show, title, productionCosts, id_Company)  
values (4, 'Dragon Ball', 3000, 4);  
Insert into show (Id_Show, title, productionCosts, id_Company)  
values (5, 'One Piece', 4500, 5);
```

Par exemple, nous avons volontairement attribué des bénéfices peu élevés ne pouvant pas satisfaire les dépenses trop élevées, afin que la « balance » soit négative, pour pouvoir donc répondre à plusieurs questions de l'énoncé.

De plus, la partie « Grant Management » a été traitée par ce biais. En effet, l'ensemble de subventions effectuées par les différents organismes y sont répertoriés. Par conséquent, on peut prendre connaissance de la fréquence des dons d'un organisme, ainsi que leurs montants.

Enfin, la partie « Insertion » est pour nous le moyen de traiter la notion de représentations internes/externes. Ainsi, une représentation est dite « interne » si ses « travel costs » et « staging costs » sont nuls. Par ailleurs, l'entreprise gagne sur la vente des tickets et perd sur les salaires des comédiens lors d'une représentation interne. À l'inverse, lors d'une représentation externe, l'entreprise gagne sur les « staging costs » et perd sur les « travel costs » et les salaires des comédiens.

## Requêtes

Les requêtes sont formulées pour répondre aux questions de l'énoncé, et classées par partie du sujet (« Organization », « Ticketing », « Accounting », « Network »).

```

-- Organization

-- Is no representation of company overlap another one ?
CREATE OR REPLACE TRIGGER overlap
BEFORE INSERT OR UPDATE
ON Presentation
FOR EACH ROW
DECLARE
    nb_rows NUMBER;
BEGIN
    SELECT NVL(count(DISTINCT id_Theater),0) INTO nb_rows
    FROM Company c
    INNER JOIN Show s USING(id_company)
    INNER JOIN Presentation p USING (id_show)
    WHERE id_company = (SELECT id_Company FROM company c INNER JOIN Show s USING(id_Company) WHERE s.id_show = :new.id_Show)
    AND p.date_ = :new.date_
    GROUP BY p.date_;

    IF nb_rows >= 1 THEN
        raise_application_error(-20001, 'this company already presents in a theater on this day');
    END IF;
EXCEPTION
    WHEN no_data_found THEN
        null;
END;
/

```

En outre, nous avons implémenté des triggers satisfaisant les besoins de l'énoncés concernant les promotions (intitulés « -20% », « -30% », « -50% »).

```

-- Promotions

-- 20%
CREATE OR REPLACE TRIGGER fifteenDays

--30%
CREATE OR REPLACE TRIGGER capacityLessFifty

--50%
CREATE OR REPLACE TRIGGER capacityLessThirty

```

En plus des requêtes demandées dans l'énoncé, nous avons également implémenté plusieurs triggers permettant de respecter quelques règles logiques, à savoir :

- le nombre de ventes pour une présentation ne peut pas dépasser la capacité de la salle où elle aura lieu (« capacity rule »),
- la date de vente d'un ticket pour une présentation ne peut pas survenir après la date de cette-même présentation (« time rule »).

```

-- Capacity rule
CREATE OR REPLACE TRIGGER noSizeExceed
BEFORE INSERT OR UPDATE ON Sales
FOR EACH ROW

DECLARE
    capacity NUMBER;
    tickets NUMBER;

BEGIN
    SELECT sum(TicketsSold) + :new.ticketsSold INTO tickets
    FROM sales WHERE id_presentation = :new.id_presentation
    GROUP BY id_Presentation;

    SELECT t.capacity INTO capacity
    FROM Presentation p
    INNER JOIN Theater t USING (id_Theater)
    WHERE p.id_Presentation = :new.id_Presentation;

    IF tickets > capacity THEN
        raise_application_error(-20003, 'the number of tickets for this presentation will exceed the theater capacity');
    END IF;

END;
/

```

```

-- Time rule
CREATE OR REPLACE TRIGGER datePassed
BEFORE INSERT OR UPDATE ON Sales
FOR EACH ROW

DECLARE
    date1 DATE;

BEGIN
    SELECT date_ INTO date1
    FROM presentation WHERE id_presentation = :new.id_presentation;

    IF :new.date_ > date1 THEN
        raise_application_error(-20006,'a ticket purchase cannot be recorded after the presentation date');
    END IF;

END;
/

```

## Tests

Voici la liste des requêtes à effectuer pour s'assurer que les triggers fonctionnent.

### Time Rule

```

insert into Sales (id_presentation, id_promotion, date_, ticketsSold)
values (1,2,'30-JAN-2022',10);

```

1	CREATE OR REPLACE TRIGGER datePassed	
2	BEFORE INSERT OR UPDATE ON Sales	
3	FOR EACH ROW	
4		
5	DECLARE	
6	date1 DATE;	
7		
8	BEGIN	
9	SELECT date_ INTO date1	
10	FROM presentation WHERE id_presentation = :new.id_presentation;	
11		
12	IF :new.date_ > date1 THEN	
13	raise_application_error(-20006,'a ticket purchase cannot be recorded after the presentation date');	
14	END IF;	
15		
16	END;	
17	/	
18		
19	insert into Sales (id_presentation, id_promotion, date_, ticketsSold)	
20	values (1,2,'30-JAN-2022',10);	

ORA-20006: a ticket purchase cannot be recorded after the presentation date  
 ORA-06512: at "SQL\_ITRGTPTJFYNNWAXACECBXVCKJIGR.DATEPASSED", line 9  
 ORA-06512: at "SYS.DBMS\_SQL", line 1721

### Capacity Rule

```

insert into Sales (id_presentation, id_promotion, date_, ticketsSold)
values (1,2,'01-JAN-2022',100);

```

```

1 CREATE OR REPLACE TRIGGER noSizeExceed
2 BEFORE INSERT OR UPDATE ON Sales
3 FOR EACH ROW
4
5 DECLARE
6     capacity NUMBER;
7     tickets NUMBER;
8
9 BEGIN
10    SELECT sum(TicketsSold) + :new.ticketsSold INTO tickets
11    FROM sales WHERE id_presentation = :new.id_presentation
12    GROUP BY id_Presentation;
13
14    SELECT t.capacity INTO capacity
15    FROM Presentation p
16    INNER JOIN Theater t USING (id_Theater)
17    WHERE p.id_Presentation = :new.id_Presentation;
18
19    IF tickets > capacity THEN
20        raise_application_error(-20003,'the number of tickets for this presentation will exceed the theater capacity');
21    END IF;
22
23 END;
24 /
25
26 insert into Sales (id_presentation, id_promotion, date_, ticketsSold)
27 values (1,2,'01-JAN-2022',100);

```

ORA-20003: the number of tickets for this presentation will exceed the theater capacity ORA-06512: at "SQL\_ITRGTPIFYNNWACECBXVCKJIGR.NOSIZEEXCEED", line 16  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

-20%

insert into Sales (id\_presentation, id\_promotion, date\_, ticketsSold)  
values (1,2,'19-JAN-2022',10);

```

1 CREATE OR REPLACE TRIGGER fifteenDays
2 BEFORE INSERT ON Sales
3 FOR EACH ROW
4
5 DECLARE
6     date1 DATE;
7
8 BEGIN
9     SELECT date_ INTO date1
10    FROM presentation WHERE id_presentation = :new.id_presentation;
11
12    IF date1 - :new.date_ < 15 AND :new.id_Promotion = 2 THEN
13        raise_application_error(-20002,'this promotion is no longer available for this presentation');
14    END IF;
15 END;
16 /
17
18 insert into Sales (id_presentation, id_promotion, date_, ticketsSold)
19 values (1,2,'19-JAN-2022',10);

```

ORA-20002: this promotion is no longer available for this presentation ORA-06512: at "SQL\_ITRGTPIFYNNWACECBXVCKJIGR.FIFTEENDAYS", line 9  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

-30%

insert into Sales (id\_presentation, id\_promotion, date\_, ticketsSold)  
values (1,3,'19-JAN-2022',55);

```

15 FROM Presentation p
16 INNER JOIN Theater t USING (id_Theater)
17 WHERE p.id_Presentation = :new.id_Presentation;
18
19 SELECT sum(TicketsSold) INTO tickets
20 FROM sales WHERE id_presentation = :new.id_presentation
21 GROUP BY id_Presentation;
22
23 IF :new.id_Promotion = 3 THEN
24     IF date1 = :new.date_ AND tickets < 0.5*capacity AND :new.id_Promotion = 3 THEN
25         NULL;
26     ELSE
27         raise_application_error(-20004,'this promotion is no longer available for this presentation');
28     END IF;
29 END IF;
30 END;
31 /
32
33 insert into Sales (id_presentation, id_promotion, date_, ticketsSold)
34 values (1,3,'19-JAN-2022',55);

```

ORA-20004: this promotion is no longer available for this presentation ORA-06512: at "SQL\_ITRGTPJFYNNWAXACEBXVCKJIGR.CAPACITYLESSFIFTY", line 23  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

-50%

insert into Sales (id\_presentation, id\_promotion, date\_, ticketsSold)  
values (1,4,'19-JAN-2022',55);

```

15 FROM Presentation p
16 INNER JOIN Theater t USING (id_Theater)
17 WHERE p.id_Presentation = :new.id_Presentation;
18
19 SELECT sum(TicketsSold) INTO tickets
20 FROM sales WHERE id_presentation = :new.id_presentation
21 GROUP BY id_Presentation;
22
23 IF :new.id_Promotion = 4 THEN
24     IF date1 = :new.date_ AND tickets < 0.3*capacity AND :new.id_Promotion = 4 THEN
25         NULL;
26     ELSE
27         raise_application_error(-20005,'this promotion is no longer available for this presentation');
28     END IF;
29 END IF;
30 END;
31 /
32
33 insert into Sales (id_presentation, id_promotion, date_, ticketsSold)
34 values (1,4,'19-JAN-2022',55);

```

ORA-20005: this promotion is no longer available for this presentation ORA-06512: at "SQL\_ITRGTPJFYNNWAXACEBXVCKJIGR.CAPACITYLESSTHIRTY", line 23  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

Is no representation of company overlap another one ?

Insert into presentation (Id\_Presentation, stagingCosts, travelCosts, date\_, Id\_Show, id\_Theater, tourNumber, comedianFees, ticketReferencePrice, arrivalDate)  
values (10, 0, 0, to\_date('20-01-2022','dd-mm-yyyy'), 1, 1, 1, 200, 16, to\_date('15-01-2022','dd-mm-yyyy'));

```

10 SELECT NVL(count(DISTINCT id_Theater),0) INTO nb_rows
11 FROM Company c
12 INNER JOIN Show s USING (id_company)
13 INNER JOIN Presentation p USING (id_show)
14 WHERE id_company = (SELECT id_Company FROM company c INNER JOIN Show s USING (id_Company) WHERE s.id_show = :new.id_Show)
15 AND p.date_ = :new.date_
16 GROUP BY p.date_;
17
18 IF nb_rows >= 1 THEN
19     raise_application_error(-20001, 'this company already presents in a theater on this day');
20 END IF;
21
22 EXCEPTION
23 WHEN no_data_found THEN
24     null;
25 END;
26 /
27
28 Insert into presentation (Id_Presentation, stagingCosts, travelCosts, date_, Id_Show, id_Theater, tourNumber, comedianFees, ticketReferencePrice, arrivalDate)
29 values (10, 0, 0, to_date('20-01-2022','dd-mm-yyyy'), 1, 1, 1, 200, 16, to_date('15-01-2022','dd-mm-yyyy'));

```

ORA-20001: this company already presents in a theater on this day ORA-06512: at "SQL\_ITRGTPJFYNNWAXACEBXVCKJIGR.OVERLAP", line 14  
ORA-06512: at "SYS.DBMS\_SQL", line 1721