Database Systems / Design and Programming Exam

Kristoffer Fischer Klokker
16/08/2002
krklo21

2022

Contents

1 E/R Diagraml

A company wants to make a database of music. They want the following objects modeledin their system.

1. **Track** having Title, Length
2. **Album** having Title, Year, List of Tracks
3. **Artist** having Name, Telephone Number, Email Address, and List of Albums
4. **Concert** having Location, Date, Artist, List of Tracks

They also describe the relationships between the objects. Each track appears on one album. Each album is produced by one artist. Each concert is held by one artist, who plays a number of tracks from his/her albums.

1.1 Create an E/R diagram capturing the objects and relationships described above. Ensure that your model does not contain redundant information. Describe all your design choices and constraints

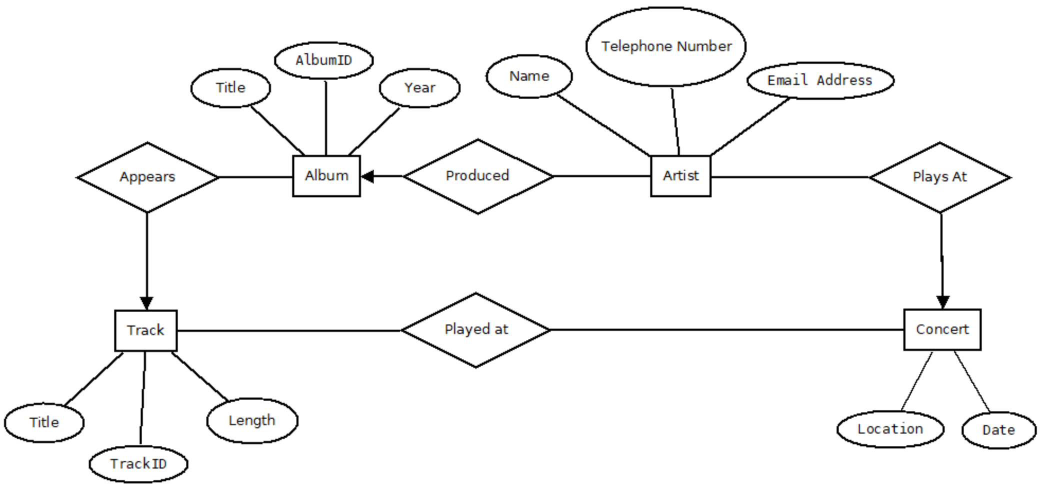


Figure 1: E/R model with relations and objects

For the the relation Appears from Track to Album, a many one relation is made and the List of Tracks is therefore not needed.

The relation Produced is also many one and eleminate the need for List of Albums on an artist.
Plays at is also a many one relation due to the text describing only one artist being able to play at an concert.
The concert also has a relation Played at, from track to concert. This relation was described as tracks from the artists album, but if the relation should be at the album the whole album needed to be played. Therefore it is a many many relation between track and concert.

1.2 For each entity set, determine an appropriate key and underline it. If you feel the attributes of an entity set do not form an appropriate key, you are allowed to introduce an ID.

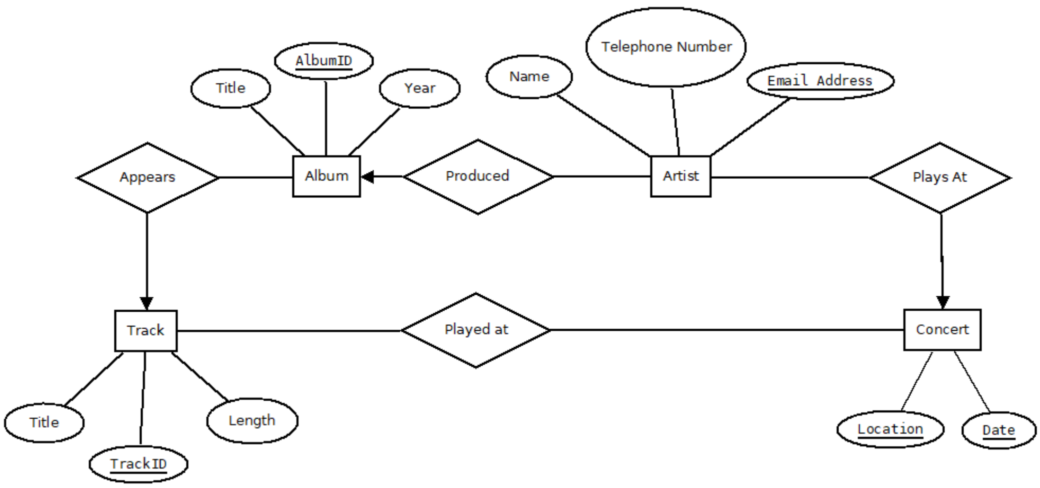
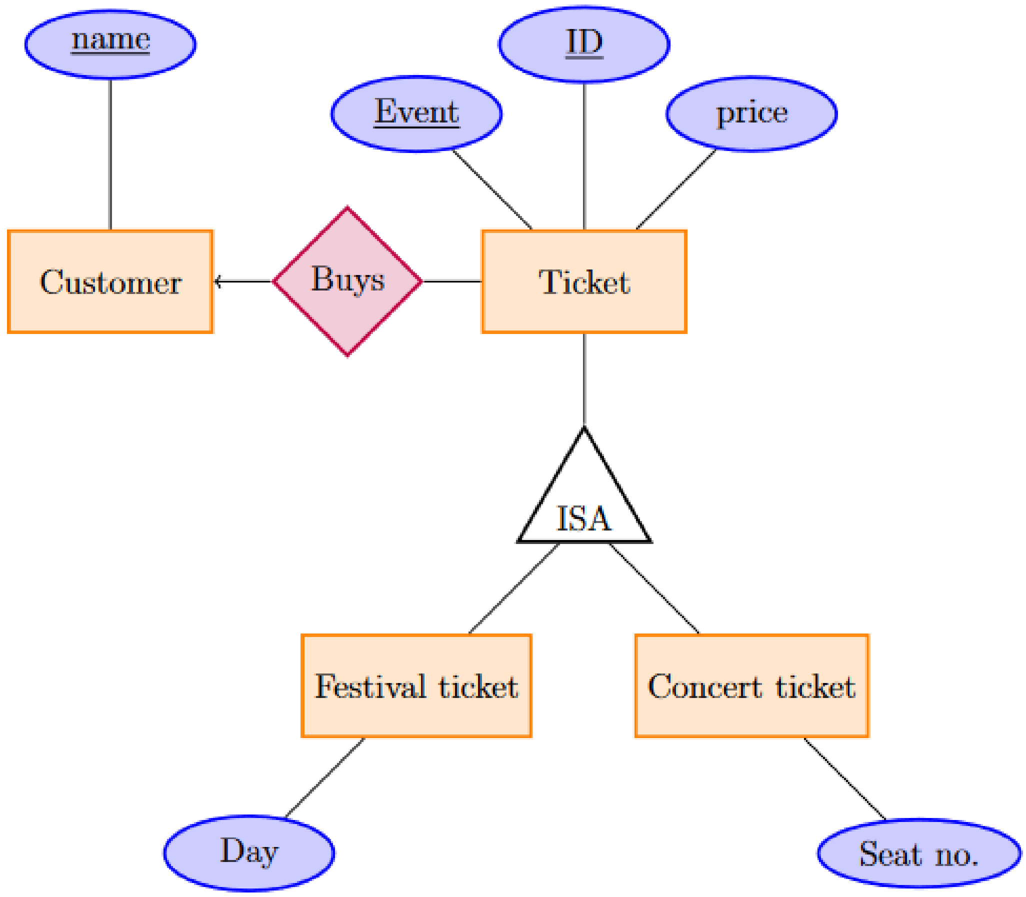


Figure 2: Figure 1 but with keys for the objects

For Album I choose to create a unique ID, due to an artist may revise and rerelease an album the same year.
The same goes for a track, it may be remixed or revised, so a unique ID is also required.
For the artist the Email Address can be used, due to it already being unique. The only problem may be if the artist have a personal and a business email, but that would require a whole new object. So therefore it is assumed the artist only have 1 email.
The concert key is Location and Date due to no two concert can be played at the exact same address and date if time is included in date.

2 From E/R diagram torelational model



A ticket agency has modeled their database system using the above E/R diagram. Now they need your help to translate it to relational schemata. They are unsure which translation to use and give you the following tasks.

2.1 The relational schemata must be created using the E/R-style translation

Ticket(<u>Event</u>,<u>ID</u>,price)
Customer(<u>name</u>)
Buys(<u>name</u>,<u>Event</u>,<u>ID</u>)
FestivalTicket(<u>Event</u>,<u>ID</u>,day)
ConcertTicket(<u>Event</u>,<u>ID</u>,Seat no.)

2.2 The relational schemata must be created using the object-oriented translation

Ticket(Event,ID,price)
Customer(name)
Buys(name,Event,ID)
Festival ticket(Event,ID,price,day)
ConcertTicket(Event,ID,price,Seat no.)
ConcertFestivalTicket(Event,ID,price,Seat no., Day)

2.3 The relational schemata must be created using the null translation

Ticket(Event,ID,price,Day,Seat no.)
Customer(name)
Buys(name,Event,ID)
The buys relation still has to be its own relation, due to the customer may buy ticket for different event and thus create redundencies.

2.4 Write CREATE TABLE statements for the second translation (b), namely for the object-oriented translation

```
CREATE TABLE Ticket(
   Event VARCHAR(40),
   ID INT,
   Price FLOAT,
   PRIMARY KEY(Event,ID)
);
CREATE TABLE Customer(
   Name VARCHAR(40) PRIMARY KEY
);
CREATE TABLE Buys(
   Name VARCHAR(40) REFERENCES Customer(Name),
   Event VARCHAR(40) REFERENCES Ticket(Event),
   ID INT REFERENCES Ticket(ID)
);
CREATE TABLE FestivalTicket(
   Event VARCHAR(40) REFERENCES Ticket(Event),
   ID INT REFERENCES Ticket(ID),
   Price FLOAT,
   Day DATE,
   PRIMARY KEY(Event,ID)
```

```
    );
    CREATE TABLE ConcertTicket(
       Event VARCHAR(40) REFERENCES Ticket(Event),
       ID INT REFERENCES Ticket(ID),
       Price FLOAT,
       SeatNo INT,
       PRIMARY KEY(Event,ID)
    );
    CREATE TABLE ConcertFestivalTicket(
       Event VARCHAR(40) REFERENCES Ticket(Event),
       ID INT REFERENCES Ticket(ID),
       Price FLOAT,
       SeatNo INT,
       Day DATE,
       PRIMARY KEY(Event,ID)
    );
```

For the ISA object, the keys have been choosen to be references to Ticket relation.

3 SQL queries

BeerID	Name	Alcohol
1	Stone IPA	6.9
2	Two Hearted Ale	7.0
3	Voodoo Ranger IPA	7.0
4	Founders All Day IPA	4.7
5	Rebel IPA	6.5
6	Racer 5 IPA	7.5
7	Dogfish Head IPA	6.0
8	Ranger IPA	6.5
9	Founders Centennial IPA	7.2
10	Odell IPA	7.0
⋮	⋮	⋮

Tap	BeerID	From	To
red	1	14	16
blue	4	14	16
red	1	16	18
blue	5	16	18
red	2	18	20
blue	6	18	20
red	2	20	22
blue	7	20	22
red	3	22	24
blue	8	22	24
⋮	⋮	⋮	⋮

Figure 3: To the left, tablebeers. To the right, tabletaps.

This task is set in a pub specialising in Indian Pale Ales (IPAs). The pub has far more kegs of IPAs than taps to connect them to. Thus, each kegs is at most connected to a tap some of the time. To keep track of this and to inform the customers, the pub maintains a database system.
Figure 3 shows the schemas and some example data for two important tables of the database. The left table,beers, contains information about the beers available. Each beer has a unique numeric BeerID. It also has a name and its alcohol percentage.
The right table,taps, contains information about which beer (keg) is connected to which tap at what time. The left-most column shows the name of tap, the second the numeric id of the beer connected between the two times specified in the third and fourth column. Not all taps have to have a beer connected and not all beers need to be connected to a tap.
In the following tasks, you should write a single SQL expression.

3.1 Write an SQL SELECT query without subqueries, which lists the names of all beers that are served from at least one tap Only those names should occur, and they should only occur once

```sql
SELECT DISTINCT Name FROM beers , taps WHERE beers .
 BeerID = taps . BeerID
```

3.2 Write an SQL SELECT query with subqueries, which lists the alcohol percentages served from the red tap for all beers that have the string "IPA" as a substring.

```
SELECT Alcohol FROM beers WHERE beers.BeerID IN(
 SELECT BeerID FROM taps WHERE Tap = 'red') AND Name
 Like '\%IPA\%'
```

3.3 Write an SQL SELECT query, which lists all taps together with the average alcohol percentage of the beers that they are connected to. There should be exactly one row for each tap in the result. If a beer is connected to a tap multiple times per day, it should contribute only once

```
SELECT Tap,AVG(Alcohol) FROM
(SELECT DISTINCT BeerID,Tap FROM taps) AS DisBeer,
beers WHERE DisBeer.BeerID = beers.BeerID GROUP BY
 Tap
```

4 Relational algebra

4.1 For each of the following expressions, give the resulting relation as a table similar to the ones for R and S above.

A	B	C
1	2	3
9	8	3
4	7	1

R

A	B	D
2	3	1
3	1	3
4	7	9

S

4.1.a $\sigma_{C>D}(\sigma_{R.A<S.A}(R \times S))$

R.A	R.B	C	S.A	S.B	D
1	2	3	2	3	1

4.1.b $\Pi_{A,B}(S) \cap \Pi_{C,A}(R)$

A	B
2	3
4	7

4.1.c $\gamma_{C,sum(A)}(R)$

C	SUM(A)
3	10
1	4

4.1.d $\sigma_{A>D}(S)$

A	B	D
2	3	1

4.1.e $R \bowtie S$

A	B	C	D
4	7	1	9

4.2 For each of the following SQL statements, write an expression of extended relational algebra which is equivalent in its effect to the following SQL statement independent of the actual tuples present in R and S

4.2.a SELECT R.B,S.A,D,C FROM S JOIN R ON R.B=S.B AND C > D

$\sigma_{R.B,S.A,D,C}(S \bowtie_{R.B=S.B \wedge C>D} R)$

4.2.b SELECT DISTINCT D, avg(A) FROM S WHERE D= 4 OR B= 2 GROUP BY D

$\gamma_{D,AVG(A)}(\sigma_{D=4 \vee B=2}(S))$
The distinct part of the query is not needed, due to the grouping, which will not result in any possible duplicates.

5 Normalization

Consider the relation

$R(SenderID, SenderName, ReceiverID, ReceiverName, External, Message, Hash)$

with the following functional dependencies:

1. $SenderID \rightarrow SenderName$
2. $ReceiverID \rightarrow ReceiverName$
3. $SenderID\ ReceiverID \rightarrow External$
4. $Message \rightarrow Hash$
5. $SenderID\ ReceiverID\ Hash \rightarrow Message$

5.1 List all keys of R and argue why there can be no other keys

There are two possible keys:
$\{SenderID, ReceiverID, Message\}$
$\{SenderID, ReceiverID, Hash\}$
Superkeys are possible from these, but only two keys are possible due to the rest of the attributes are on the rightside of the FD's

5.2 Analyze whether R is BCNF. If it is, show that there are no BCNF violations. If it is not, show that there is at least one BCNF violation and decompose R until it is in BCNF

The first violation is $SenderID \rightarrow SenderName$
$R_1 = SenderID^+ = \{SenderName, SenderID\}$
$R_2 = R - (SenderID^+ - SenderID)$
$R_2 = \{SenderID, ReceiverID, ReceiverName, External, Message, Hash\}$
Where R_1 has the FD $SenderID \rightarrow SenderName$ and is in BCNF now
And R_2 is not BCNF due to $ReceiverID \rightarrow ReceiverName$

The decomposition is just like for R_1 and R_2 but with receiver resulting in:

$R_3 = ReceiverID^+ = \{ReceiverID, ReceiverName\}$
$R_4 = R_2 - (ReceiverID^+ - ReceiverID)$
$R_4 = \{SenderID, ReceiverID, External, Message, Hash\}$
Now R_3 is BCNF with the FD $ReceiverID \rightarrow ReceiverName$
R_4 is not BCNF due to the violation of $SenderID\ ReceiverID \rightarrow External$

$R_5 = \{SenderID, ReceiverID\}^+ = \{SenderID, ReceiverID, External\}$
$R_6 = R_4 - (\{SenderID, ReceiverID\}^+ - \{SenderID, ReceiverID\})$
$R_6 = \{SenderID, ReceiverID, Message, Hash\}$
R_5 is BCNF with the FD $SenderID\ ReceiverID\ Hash \rightarrow Message$
While R_6 is not BCNF due to $Message \rightarrow Hash$

$R_7 = Message^+ = \{Message, Hash\}$
$R_8 = R_6 - (Message^+ - Message) = \{SenderID, ReceiverID, Message\}$
R_7 is BCNF with the FD $Message \rightarrow Hash$
R_8 is BCNF with no FD

Therefore the final relation is:
$R_1 = \{SenderName, SenderID\}$
$R_3 = \{ReceiverID, ReceiverName\}$
$R_5 = \{SenderID, ReceiverID, External\}$
$R_7 = \{Message, Hash\}$
$R_8 = \{SenderID, ReceiverID, Message\}$

5.3 What would change if we would consider 3NF instead of BCNF? Explain your answer

It would result in R_6 not getting decomposed. This is due to R_6 having the FD's:
$Message \rightarrow Hash$
$SenderID\ ReceiverID\ Hash \rightarrow Message$
BCNF would not allow the first FD, but in 3NF it is allowed due to $Hash$ being part of a key. The other FD is allowed due to the left side being a superkey.