Question 1. [18 MARKS]

Consider the following schema for tracking customers' ratings of books for a bookstore website.

- Books(ISBN, title, author, year, length). ISBN is a string used internationally for identifying books.
- Authors(AID, name, city)
- Customers(CID, name)
- Rates(ISBN, CID, rating). Rating is an integer.

The following inclusion dependencies hold:

- Books(author) $\subseteq$ Authors(AID)
- Rates(ISBN) $\subseteq$ Books(ISBN)
- Rates(CID) $\subseteq$ Customers(CID)

Throughout this question, we will use only the basic Relational Algebra operators $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho$, :=. Assume the set semantics (not bag semantics) for Relational Algebra.

Part (a) [6 MARKS]

For each query below, explain in plain English what it computes.

1. $(\rho_{R(CID)}\Pi_{R1.CID}\sigma_{R1.CID=R2.CID \wedge R1.ISBN \neq R2.ISBN}((\rho_{R1}Rates) \times (\rho_{R2}Rates)))$
   $\cup$
   $(\Pi_{CID}\sigma_{length>500}(Rates \bowtie Books))$

   Solution:

   The CID of all customers who rated at least two books or rated at least one book that has over 500 pages.

2. $R0 := \Pi_{AID}\sigma_{city=\text{“}Iqaluit\text{”}}Author$
   $R1(CID, AID) := \Pi_{CID,author}\sigma_{rating=10}(Rates \bowtie Books)$
   $R2 := (\Pi_{CID}Customers) \times (R0)$
   $R3 := R2 - R1$
   $Answer := (\Pi_{CID}Customer) - (\Pi_{CID}R3)$

   Solution:

   The CID of all customers who gave a rating of 10 to every author from Iqaluit (for one or more of their books).

3. $R1(CID, AID) := \Pi_{CID,author}(Rates \bowtie Books)$

   $R2(CID, AID) := \Pi_{CID,author}\sigma_{length>500}(Rates \bowtie Books)$

   $R3 := R1 - R2$

   $Answer := (\Pi_{CID}Customer) - (\Pi_{CID}R3)$

   Solution:

   The CID of all customers who didn't rate any book of under 500 pages. (They may have rated no books at all.)

Part (b) [12 MARKS]

Write the following queries in relational algebra.

1. Find the name and AID of the author of the second shortest book with a rating of 6 or more. If there is a tie, report them all.

Here is that schema again, for easy reference:

- Books(ISBN, title, author, year, length). ISBN is a string used internationally for identifying books.
- Authors(AID, name, city)
- Customers(CID, name)
- Rates(ISBN, CID, rating). Rating is an integer.

The following inclusion dependencies hold:

- Books(author) $\subseteq$ Authors(AID)
- Rates(ISBN) $\subseteq$ Books(ISBN)
- Rates(CID) $\subseteq$ Customers(CID)

2. Find the CID and name of customers who have rated at most two books and have never rated one by any author named "Laurence".

Question 2. [9 MARKS]

Answer the following questions. **Do not guess** on the true-false questions; there is one point for each correct answer, -1 for each incorrect answer, and 0 points if you leave the answer blank.

1. If R has one tuple, $R \bowtie R$ has no tuples.

   True [False]

2. For any relation R, $R \bowtie R = R$.

   [True] False

3. If relation R has arity 1, $R \bowtie R$ has cardinality 0.

   True [False]

4. Suppose we have a relation R with attributes ABCD and that the following is a legal instance of this relation:

A	B	C	D
1	3	9	6
2	3	9	6
1	3	4	6
5	7	9	6

   Consider the possible functional dependencies below. Circle all that definitely do *not* hold in R.

   [$CD \rightarrow B$] $BC \rightarrow D$ [$A \rightarrow C$] [$C \rightarrow A$]

5. Assume the same relation and instance as in the previous question. Can you identify an FD not listed above that does hold? If yes, state it. If not, explain why not.

   Solution:

   You can never determine that an FD holds based on an instance of the relation. (You can, on the other hand, determine that one *doesn't* hold based on a single instance if that instance violates it.) The only way to determine that an FD holds is by knowing the domain.

6. Consider the relation the relation S with attributes MNOP only one key: MO. Give a set of functional dependencies under which S is in 3NF but not BCNF.

   Solution: If the FDs are:

   $MO \rightarrow NP$ and $P \rightarrow M$

   Then the relation is not in BCNF because P is not a superkey. But it is in 3NF because M is prime (part of the key MO).

Question 3. [7 MARKS]

Consider the following table definitions.

```
CREATE TABLE S (
    P INT PRIMARY KEY,
    Q INT
);

CREATE TABLE R (
    A INT PRIMARY KEY,
    B INT,
    C INT,
    FOREIGN KEY (B) REFERENCES S(P)
         ON UPDATE SET NULL ON DELETE CASCADE
);

CREATE TABLE T (
    M INT,
    N INT,
    O INT,
    PRIMARY KEY (M, N),
    FOREIGN KEY (M) REFERENCES S(P)
         ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (N) REFERENCES R(A)
         ON DELETE CASCADE ON UPDATE CASCADE
);
```

Suppose the tables have been populated as follows:

R:

A	B	C
11	2	76
15	8	71
18	8	75

S:

P	Q
2	140
9	149
8	142
6	140

T:

M	N	O
2	11	1112
8	11	1219
6	18	1219
9	15	1317

1. Write a query to delete the tuple (2,11,1112) from T.

   Solution:

   ```
   delete from t where m=2;
   ```

2. Show the contents of the three relations after your query is executed.

   Solution:

   ```
   csc343h-dianeh=> select * from r;
    a  | b | c
   ----+---+----
    11 | 2 | 76
    15 | 8 | 71
    18 | 8 | 75
   (3 rows)

   csc343h-dianeh=> select * from s;
    p |  q
   ---+-----
    2 | 140
    9 | 149
    8 | 142
    6 | 140
   (4 rows)

   csc343h-dianeh=> select * from t;
    m | n  |  o
   ---+----+------
    8 | 11 | 1219
    6 | 18 | 1219
    9 | 15 | 1317
   (3 rows)
   ```

3. Suppose, instead, we were to start with the original tables shown above and make two changes to R: delete the tuple (18,8,75) and change the tuple (15,8,71) to (15,6,71). Show the contents of the three relations after these changes. **Solution:**

```
csc343h-dianeh=> select * from r;
 a  | b | c
----+---+----
 11 | 2 | 76
 15 | 6 | 71
(2 rows)

csc343h-dianeh=> select * from s;
 p |  q
---+-----
 2 | 140
 9 | 149
 8 | 142
 6 | 140
(4 rows)

csc343h-dianeh=> select * from t;
 m | n  |  o
---+----+------
 2 | 11 | 1112
 8 | 11 | 1219
 9 | 15 | 1317
(3 rows)
```

4. Suppose, instead, we were to start with the original tables shown above and delete the tuple (8,142) from S. Show the contents of the 3 relations after these changes. **Solution:**

```
csc343h-dianeh=> select * from r;
 a  | b | c
----+---+----
 11 | 2 | 76
(1 row)

csc343h-dianeh=> select * from s;
 p |  q
---+-----
 2 | 140
 9 | 149
 6 | 140
(3 rows)

csc343h-dianeh=> select * from t;
 m | n  |  o
---+----+------
 2 | 11 | 1112
(1 row)
```

Question 4. [12 MARKS]

Consider the following schema for a grocery store. Note that it does not represent that quantity of an item included in a purchase. Assume that no one every buys more than one of anything in a single purchase. (This will make it easy to total up the cost of a purchase.)

```
-- Category can have values such as 'food', 'drugs', or 'toiletries'.
CREATE TABLE Product (
   id INT PRIMARY KEY,
   name TEXT,
   category TEXT
);

CREATE TABLE Price (
   id INT PRIMARY KEY,
   price FLOAT,
   FOREIGN KEY (id) REFERENCES Product(id)
);

-- A purchase on this day by the customer at this phone number.
CREATE TABLE Purchase (
   id INT PRIMARY KEY,
   phone TEXT,
   day DATE
);

-- This item was included in this purchase.
CREATE TABLE included (
   item INT,
   purchase INT,
   PRIMARY KEY (item, purchase),
   FOREIGN KEY (item) REFERENCES Product(id),
   FOREIGN KEY (purchase) REFERENCES Purchase(id)
);
```

Write queries in SQL to get the following information.

1. The phone number, day and total cost of the most expensive purchase not counting the cost of any drugs (*i.e.*, any product whose category is 'drugs').

   Solution:

```
create view totals as (
      select purchase, sum(price) as total
      from included join product on item=id natural join price
      where category <> 'drugs'
      group by purchase
);

select phone, day, total
from Purchase join
      (select purchase, total
      from totals t1
      where not exists
            (select * from totals t2 where t2.total > t1.total)
      ) biggest
on id=purchase;
```

Here is the same schema again, for easy reference:

```
-- Category can have values such as 'food', 'drugs', or 'toiletries'.
CREATE TABLE Product (
   id INT PRIMARY KEY,
   name TEXT,
   category TEXT
);

CREATE TABLE Price (
   id INT PRIMARY KEY,
   price FLOAT,
   FOREIGN KEY (id) REFERENCES Product(id)
);

-- A purchase on this day by the customer at this phone number.
CREATE TABLE Purchase (
   id INT PRIMARY KEY,
   phone TEXT,
   day DATE
);

-- This item was included in this purchase.
CREATE TABLE included (
   item INT,
   purchase INT,
   PRIMARY KEY (item, purchase),
   FOREIGN KEY (item) REFERENCES Product(id),
   FOREIGN KEY (purchase) REFERENCES Purchase(id)
);
```

2. For each product, the number of distinct customers (identified by their phone numbers) who have purchased it. Include products even if no one ever purchased them.

   Solution:

```
create view counts as (
      select item, count(distinct phone)
      from purchase join included on id=purchase
      group by item
);

create view unpopular as(
      select id as item, 0 as count
      from product p1
      where not exists (
            select *
            from purchase join included on id=purchase
            where item=p1.id)
);

(select * from counts) union (select * from unpopular);
```

Question 5. [10 MARKS]

Assume that the following valid DTD is defined in file `party.dtd`:

```
<!ELEMENT GuestList (Invitee*)>
<!ELEMENT Invitee (Who, Response)>
<!ATTLIST Invitee nickname ID #REQUIRED>
<!ELEMENT Who (First, Last)>
<!ATTLIST Who age CDATA #IMPLIED>
<!ELEMENT First (#PCDATA)>
<!ELEMENT Last (#PCDATA)>
<!ELEMENT Response (#PCDATA)>
```

Part (a) [6 MARKS]

For each XML document below, the first two lines are valid. Circle the correct answer to indicate whether the entire document is valid or not. If it is not valid, circle each error and explain why it is an error.

1.
```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
   <Invitee nickname="sweetPea">
      <Who age="almost 4">
         <First>Catherine</First>
         <Last>Fairgrieve</Last>
      </Who>
      <Response>It's my party!!!</Response>
   </Invitee>
   <Invitee nickname="sunshine">
      <Who age="3" First="Avery" Last="McLaughlin"/>
      <Response>Love to come!</Response>
   </Invitee>
   <Invitee nickname="sunshine">
      <Who>
         <First>Sam</First>
         <Last>Kingston</Last>
      </Who>
      <Response>Can't wait!</Response>
   </Invitee>
</GuestList>
<GuestList>
   <Invitee nickname="dub">
      <Who age="12">
<First>William</First>  <Last>Fairgrieve</Last>
      </Who>
      <Response>Count me in</Response>
   </Invitee>
</GuestList>
```

Valid Not Valid

Solution

It's not valid. There are 3 problems: the nickname "sunshine" occurs twice; Avery's first and last shold be subelements not attributes; and it's not legal two have two GuestLists. Here is a corrected version:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
   <Invitee nickname="sweetPea">
      <Who age="almost 4">
         <First>Catherine</First>
         <Last>Fairgrieve</Last>
      </Who>
      <Response>It's my party!!!</Response>
   </Invitee>
   <Invitee nickname="sunshine">
      <Who age="3">
         <First>Avery</First>
         <Last>McLaughlin</Last>
      </Who>
      <Response>Love to come!</Response>
   </Invitee>
   <Invitee nickname="something else">
      <Who>
         <First>Sam</First>
         <Last>Kingston</Last>
      </Who>
      <Response>Can't wait!</Response>
   </Invitee>
   <Invitee nickname="dub">
      <Who age="12">
<First>William</First>  <Last>Fairgrieve</Last>
      </Who>
      <Response>Count me in</Response>
   </Invitee>
</GuestList>
```

2.
```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
</GuestList>
```

Valid Not Valid

Solution

Valid

3.
```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
   <Invitee nickname="dub">
      <Who age="12">
         <First>William</First>
         <Last>Fairgrieve</Last>
      </Who>
      <Response>Count me in</Response>
   </Invitee>
   <Invitee nickname="sunshine">
      <Response>Can't wait!</Response>
      <Who>
         <First>Sam</First>
         <Last>Kingston</Last>
      </Who>
   </Invitee>
</GuestList>
```

Valid Not Valid

Solution:

It's not valid. For Sam, the Response element comes before the Who element. Here's a corrected version:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
   <Invitee nickname="dub">
      <Who age="12">
         <First>William</First>
         <Last>Fairgrieve</Last>
      </Who>
      <Response>Count me in</Response>
   </Invitee>
   <Invitee nickname="sunshine">
      <Who>
         <First>Sam</First>
         <Last>Kingston</Last>
      </Who>
      <Response>Can't wait!</Response>
   </Invitee>
</GuestList>
```

Part (b) [4 MARKS]

Suppose we want to keep the same structure in our XML files, but to enforce additional rules. Which of these can be expressed in our DTD. **Do not quess.** Each question is worth 1 mark if correct, -1 if incorrect, and 0 if you don't respond. Your minimum mark is 0.

1. An invitee's age must be between 0 and 100.

   Can be Cannot be

2. An invitee's last name is optional.

   Can be Cannot be

3. A guest list must have at least 3 guests.

   Can be Cannot be

4. A response must be either "yes" or "no".

   Can be Cannot be

Here is a revised DTD that has all of these features:

```
<!ELEMENT GuestList (Invitee, Invitee, Invitee+)>
<!ELEMENT Invitee (Who)>
<!ATTLIST Invitee nickname ID #REQUIRED>
<!ATTLIST Invitee response (yes|no) #REQUIRED>
<!ELEMENT Who (First, Last?)>
<!ATTLIST Who age (
    0|1|2|3|4|5|6|7|8|9|
    10|11|12|13|14|15|16|17|18|19|
    20|21|22|23|24|25|26|27|28|29|
    30|31|32|33|34|35|36|37|38|39|
    40|41|42|43|44|45|46|47|48|49|
    50|51|52|53|54|55|56|57|58|59|
    60|61|62|63|64|65|66|67|68|69|
    70|71|72|73|74|75|76|77|78|79|
    80|81|82|83|84|85|86|87|88|89|
    90|91|92|93|94|95|96|97|98|99|
    100
 ) #IMPLIED>
<!ELEMENT First (#PCDATA)>
<!ELEMENT Last (#PCDATA)>
```

It is not elegant, but it does express the above restrictions.

Question 6. [11 MARKS]

Consider the following well-formed XML document, stored in a file called "minutes.xml".

```
<proceedings>
   <members>
      <member rid="96124" reps="Trinity">
         <name>Barbara Reid</name>
         <votes>
            <vote mid="784" choice="for"/>
            <vote mid="899" choice="against"/>
            <vote mid="955" choice="for"/>
         </votes>
      </member>
      <member rid="11234" reps="Vic">
         <name>Quentin Blake</name>
         <votes>
            <vote mid="784" choice="for"/>
            <vote mid="899" choice="for"/>
            <vote mid="955" choice="against"/>
         </votes>
      </member>
      <member rid="67890" reps="Woodsworth">
         <name>Kenneth Oppel</name>
         <votes>
            <vote mid="784" choice="for"/>
            <vote mid="899" choice="for"/>
            <vote mid="955" choice="against"/>
         </votes>
      </member>
   </members>
   <motions>
         <motion mid="784" passed="true" madeby="96124">
            The committee should set aside $100 for budget overruns.
         </motion>
         <motion mid="899" passed="true" madeby="67890">
            The chair should discuss space issues with the Dean.
         </motion>
         <motion mid="955" passed="false" madeby="96124">
            The party planned for May 12th should be cancelled.
         </motion>
   </motions>
</proceedings>
```

Part (a) [3 MARKS]

What does the following XQuery code return? Whitespace in your answer does not matter.

```
let $d := doc("minutes.xml")
for $x in $d/proceedings/members/member//vote[@choice="for"]
return
   $x
```

Solution:

```
<vote mid="784" choice="for"/>,
<vote mid="955" choice="for"/>,
<vote mid="784" choice="for"/>,
<vote mid="899" choice="for"/>,
<vote mid="784" choice="for"/>,
<vote mid="899" choice="for"/>
```

Part (b) [3 MARKS]

What does the following XQuery code return? Whitespace in your answer does not matter.

```
let $d := doc("minutes.xml")
for $x in $d/proceedings/*/motion/@madeby
return
   <made> $x </made>
```

Solution:

```
<made> $x </made>, <made> $x </made>, <made> $x </made>
```

Here is the contents of "minutes.xml" again, for easy reference.

```
<proceedings>
   <members>
      <member rid="96124" reps="Trinity">
         <name>Barbara Reid</name>
         <votes>
            <vote mid="784" choice="for"/>
            <vote mid="899" choice="against"/>
            <vote mid="955" choice="for"/>
         </votes>
      </member>
      <member rid="11234" reps="Vic">
         <name>Quentin Blake</name>
         <votes>
            <vote mid="784" choice="for"/>
            <vote mid="899" choice="for"/>
            <vote mid="955" choice="against"/>
         </votes>
      </member>
      <member rid="67890" reps="Woodsworth">
         <name>Kenneth Oppel</name>
         <votes>
            <vote mid="784" choice="for"/>
            <vote mid="899" choice="for"/>
            <vote mid="955" choice="against"/>
         </votes>
      </member>
   </members>
   <motions>
         <motion mid="784" passed="true" madeby="96124">
            The committee should set aside $100 for budget overruns.
         </motion>
         <motion mid="899" passed="true" madeby="67890">
            The chair should discuss space issues with the Dean.
         </motion>
         <motion mid="955" passed="false" madeby="96124">
            The party planned for May 12th should be cancelled.
         </motion>
   </motions>
</proceedings>
```

Part (c) [5 MARKS]

What does the following XQuery code return? Whitespace in your answer does not matter.

```
<passed>
{
let $d := doc("minutes.xml")
for $passed in $d//motions/motion[@passed="true"]
return
(
   <motion> {data($passed)} </motion>,
   <infavour>
   { for $member in $d//member
     for $vote in $member//vote
     where $vote/@mid=$passed/@mid and $vote/@choice="for"
     return <person>{data($member/name)}</person>
   }
   </infavour>
)
}
</passed>
```

Solution:

(The formatting has been modified for easier reading.)

```
<passed>
     <motion>
          The committee should set aside $100 for budget overruns.
     </motion>
     <infavour>
          <person>Barbara Reid</person>
          <person>Quentin Blake</person>
          <person>Kenneth Oppel</person>
     </infavour>
     <motion>
          The chair should discuss space issues with the Dean.
     </motion>
     <infavour>
          <person>Quentin Blake</person>
          <person>Kenneth Oppel</person>
     </infavour>
</passed>
```

Question 7. [13 MARKS]

Part (a) [7 MARKS]

Suppose we have a relation $R(A, B, C, D)$ with functional dependency: $B \to C$. Suppose that we are going to decompose R into two new relations S and T.

1. Give schemas for S and T that will yield a lossy join.

   Solution:

   AB, BCD
   (There are other solutions.)

2. Give an example demonstrating that the join is lossy. Explain your answer.

   Solution:

   The chase test yields an example:

A	B	C	D
a	b	c_1	d_1
a_1	b	c	d

   If you were to decompose then rejoin, the result would include the tuple (a, b, c, d). In other words, $(a, b, c, d) \in (\Pi_{A,B}R) \bowtie (\Pi_{B,C,D}R)$. Yet this tuple is not in R itself. Therefore the decomposition is a lossy join decomposition.

3. Give schemas for S and T that will yield a lossless join.

   Solution:

   ABD, BC
   (There are other solutions.)
   The chase test can be used to confirm this answer.

Part (b) [3 MARKS]

Is the following a valid rule about FDs: If $AB \to D$ and $BC \to D$ then $AC \to D$. Circle one answer.

Valid **Not valid**

If it is valid, prove it. If it is not valid, demonstrate this by giving an instance of a relation that satisfies $AB \to D$ and $BC \to D$ but not $AC \to D$.

Solution:

Here's a small example that satisfies $AB \to D$ and $BC \to D$ but not $AC \to D$.

A	B	C	D
1	3	9	6
1	22	9	11

This may seem like cheating since the first two FDs are never "exercised". But it does do the job. And here's an example that does exercise the first two:

A	B	C	D
1	3	9	6
1	3	8	6
5	7	8	4
2	7	8	4
1	22	9	11

Part (c) [3 MARKS]

Suppose we have a relation R with attributes ABCDE and functional dependencies $A \to D$, $B \to A$, $C \to A$, $D \to CE$. Project the functional dependencies of R onto the attribute set ABD.

Solution:

A	B	D	closure	FDs in the projection
		✓	$D^+ = DCEA$	$D \to A$
	✓		$B^+ = BADCE$	$B \to AD$
✓			$A^+ = ADCE$	$A \to D$
no need to look at supersets of B				
✓		✓	$AD^+ = ADCE$	none

Final answer: $D \to A$, $B \to AD$, $A \to D$.

(If doing the projection as part of a BCNF decomposition, you definitely need to find a minimal basis before proceeding.)

Question 8. [10 MARKS]

Suppose we have a relation R with attributes ABCD and functional dependencies $AB \rightarrow C$, $C \rightarrow A$, and $C \rightarrow D$.

1. Circle the FD(s) that violate BCNF: $AB \rightarrow C$ $\boxed{C \rightarrow A}$ $\boxed{C \rightarrow D}$.

2. Perform BCNF decomposition on relation R. Show your rough work here and your final answer at the bottom of the next page.

   Solution:

   Decompose based on the violating FD $C \rightarrow A$. This yields two new relations: R1 with attributes BC and R2 with attributes ACD. (A common mistake is to think this yields BCD and CA, but remember that you are to build one of the descendant relations using the closure of the LHS attributes, not just the attributes in the FD.)

   Project the FDs onto R1:

B	C	closure	FDs to add
	✓	$C^+ = CAD$	none
✓		$B^+ = B$	none

   R1 is in BCNF.

   Project the FDs onto R2:

A	C	D	closure	FDs to add
		✓	$D^+ = D$	none
	✓		$C^+ = CAD$	$C \rightarrow AD$
✓			$A^+ = A$	none
no need to look at supersets of C				
✓		✓	$AD^+ = AD$	none

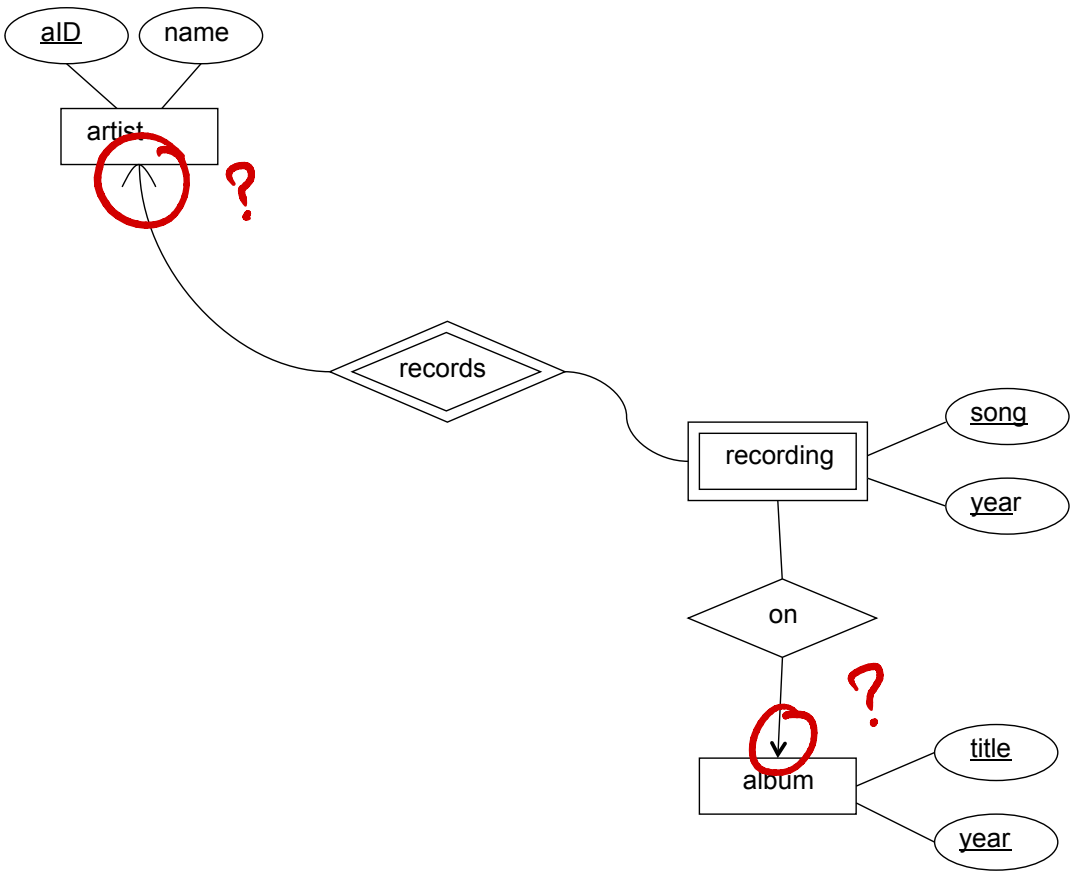
   R2 is also in BCNF.

3. Show your final answer here: List the relations that are in your final decomposition of R and state the FDs of each.

   Solution:
   R1 with attributes BC and no FDs.
   R2 with attributes ACD and FD $C \rightarrow AD$.

Question 9. [12 MARKS]

Below is an Entity/Relationship diagram about the music industry. It may or may not represent the domain well.



Part (a) [7 MARKS]

Which of the following statements are true according to this Entity/Relationship diagram? **Do not guess.** There is one point for each correct answer, -1 for each incorrect answer, and 0 points if you leave the answer blank.

1. The same recording can be on more than one album.

   True [False]

2. There can be two albums in the same year as long as they have different titles.

   [True] False

3. For every recording, there must be an artist who recorded it.

   [True] False

4. For every recording, there must be an album that it is on.

   True [False]

5. Album titles are unique.

   True [False]

6. There cannot be two recordings of the same song in the same year.

   True [False]

7. A recording can be by several artists.

   True [False]

Part (b) [2 MARKS]

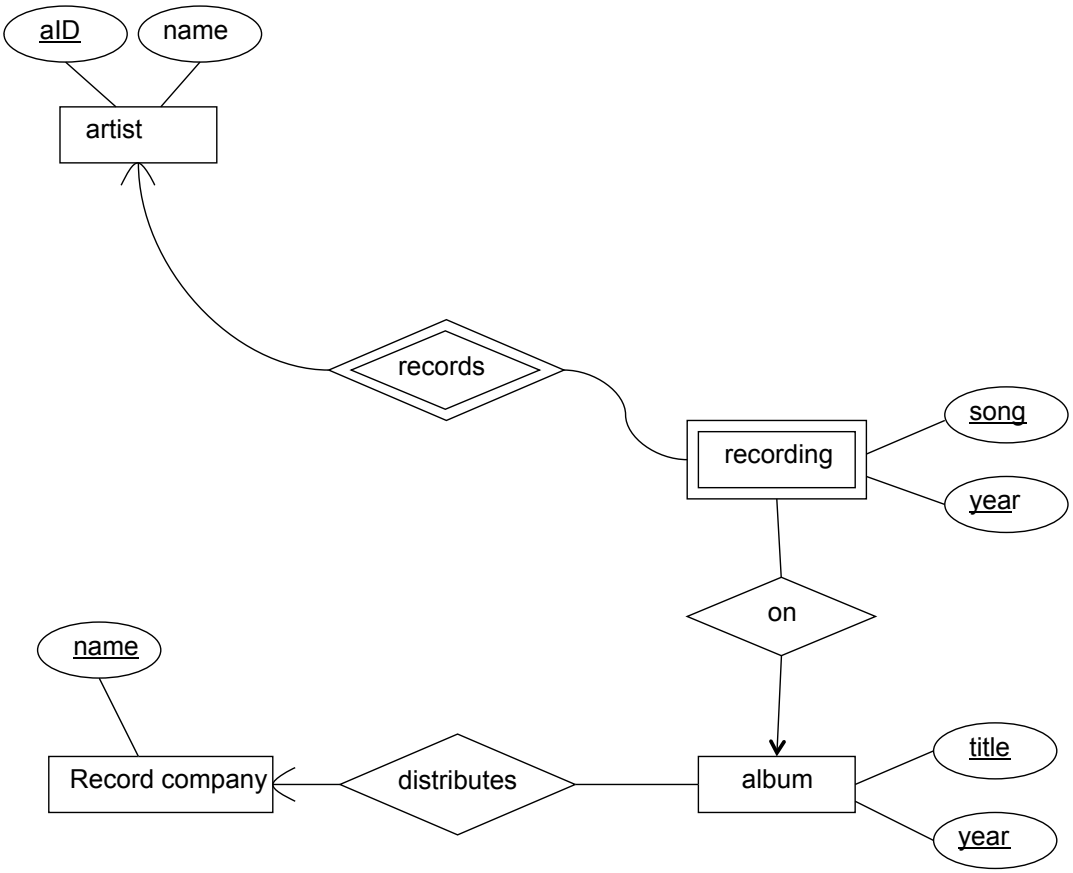
Name a weak entity set in this diagram.
Solution: There is only one: recording

Name its supporting entity set(s):
Solution: There is only one: artist

Part (c) [3 MARKS]

Extend the diagram above to represent the following information: Record companies (such as Sony Music or EMI) distribute albums. Record companies have names, which are unique. Every album is distributed by exactly one record company, but a record company can distribute many albums.

Solution:


aID
name
artist
records
recording
song
year
on
name
Record company
distributes
album
title
year