XJTLU Entrepreneur College (Taicang) Cover Sheet

Module code and Title	Database Development and Design (CPT201TC)
School Title	School of AI and Advanced Computing
Assignment Title	002: Assessment Task 2 (CW)
Submission Deadline	17:00, 17th Dec (Friday)
Final Word Count	NA
If you agree to let the university use your work anonymously for teaching and learning purposes, please type **"yes"** here.	**Yes**

I certify that I have read and understood the University's Policy for dealing with Plagiarism, Collusion and the Fabrication of Data (available on Learning Mall Online). With reference to this policy I certify that:

- My work does not contain any instances of plagiarism and/or collusion.
  My work does not contain any fabricated data.

By uploading my assignment onto Learning Mall Online, I formally declare that all of the above information is true to the best of my knowledge and belief.

Scoring – For Tutor Use	
Student ID	**1930080**

Stage of Marking	**Marker Code**	**Learning Outcomes Achieved (F/P/M/D) (please modify as appropriate)**			**Final Score**
		A	**B**	**C**	
1st Marker – red pen					
Moderation	**IM**	The original mark has been accepted by the moderator (please circle as appropriate):			Y / N

<table>
<tr><td colspan="2">– green pen</td><td>Initials</td><td colspan="3"></td><td></td></tr>
<tr><td colspan="2"></td><td></td><td colspan="3">Data entry and score calculation have been checked by another tutor (please circle):</td><td>Y</td></tr>
<tr><td colspan="2">2nd Marker if needed – green pen</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="3">For Academic Office Use</td><td colspan="4">Possible Academic Infringement (please tick as appropriate)</td></tr>
<tr><td>Date Received</td><td>Days late</td><td>Late Penalty</td><td colspan="2">☐ Category A</td><td colspan="2" rowspan="2">Total Academic Infringement Penalty (A,B, C, D, E, Please modify where necessary)
_____</td></tr>
<tr><td rowspan="4"></td><td rowspan="4"></td><td rowspan="4"></td><td colspan="2">☐ Category B</td></tr>
<tr><td colspan="2">☐ Category C</td></tr>
<tr><td colspan="2">☐ Category D</td></tr>
<tr><td colspan="2">☐ Category E</td></tr>
</table>

Students

The assignment must be typed in an MS Word document and submitted via Learning Mall Online to the correct drop box. Only electronic submission is accepted and no hard copy submission.

All students must download their file and check that it is viewable after submission. Documents may become corrupted during the uploading process (e.g. due to slow internet connections). However, students themselves are responsible for submitting a functional and correct file for assessments.

I. DBMS Indexing

Q1(a)

Record has fields: Name (30 bytes), ID (10 bytes), email (10 bytes), Address (40 bytes), Phone (10 bytes), and Birth_date (10 bytes)

`Record_Length`: R = (30+10+10+10+10+10+40)+1 = 111 `bytes`

`Blocking factor`: bfr = floor($\frac{B}{R}$)=floor($\frac{512}{111}$)=4 `record per block`

`number of block`: b=ceiling($\frac{r}{bfr}$)=ceiling($\frac{40000}{4}$)=10000

Q1(b)

If data file is unorder a linear search on the data file would need approximately $\frac{b}{2} = 5000$ block access

If data file is ordered on ID, a binary search on the data file would need approximately $log_2{}^{b} = log_2{}^{10000} = 14$ block access

Q1(c)

$R` = (ID + P) = (10 + 6) = 16$ bytes

ID Index blocking factor $bfr` = floor(\frac{512}{16}) = 32$ records per block

Number of Blocks $b` = ceiling(\frac{r}{bfr}) = \frac{10000}{32} = 313$

number of blocks to search: $ceiling(log_2{}^{b`}) = log_2{}^{313} = 10$

Q1(d)

- **First-Level Index Entries**:

  $r_1 =$ `number of file blocks` $b = 10000$ entries

- **First-Level Index Blocks:**

  $b_1 =$ ceiling($\frac{r_1}{bfr_i}$)=ceiling($\frac{10000}{32}$)=313 blocks

- **Second-Level Index Entries**:

  r_2= `number of first-level blocks` $b_1 = 313$ entries

- **Second-Level Index Blocks:**

  b_2=ceiling($\frac{r_2}{bfr_i}$)=ceiling($\frac{313}{32}$)=10 blocks

- **Third-Level Index Entries**:

  $r_3 =$`number of second-level index blocks` b_2=10 entries

- **Third-Level Index Blocks:**

  b_3 =ceiling($\frac{r_3}{bfr_i}$)=ceiling($\frac{10}{32}$)=1

Total number of blocks for the index: $b_i = b_1 + b_2 + b_3 = 313 + 10 + 1 = 324$ blocks

Number of block accesses to search for a record: 3+1=4

Q1(e)

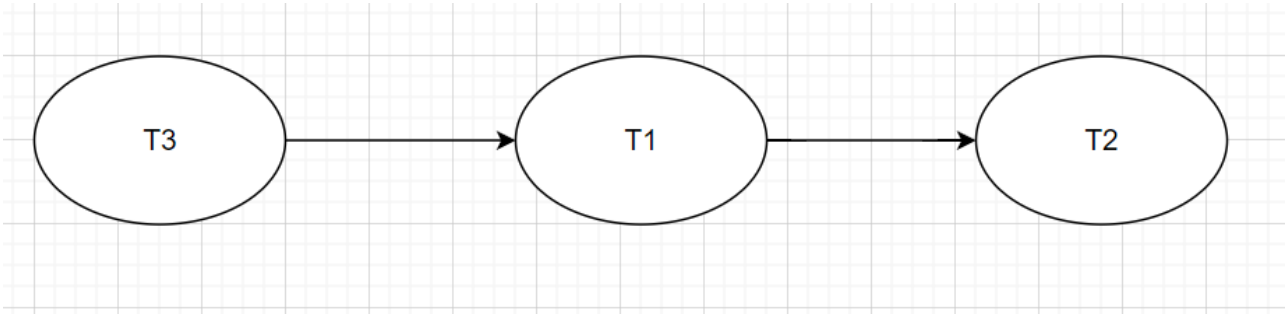
Maximum Capacity of B+ tree of height 4

- $49 \times 50 \times 50 \times 50 = 612500$

It requires 4 index-block read and 1 data-block read, so $4 + 1 = 5$ blocks in total

II. Transaction Management

Q2(a)

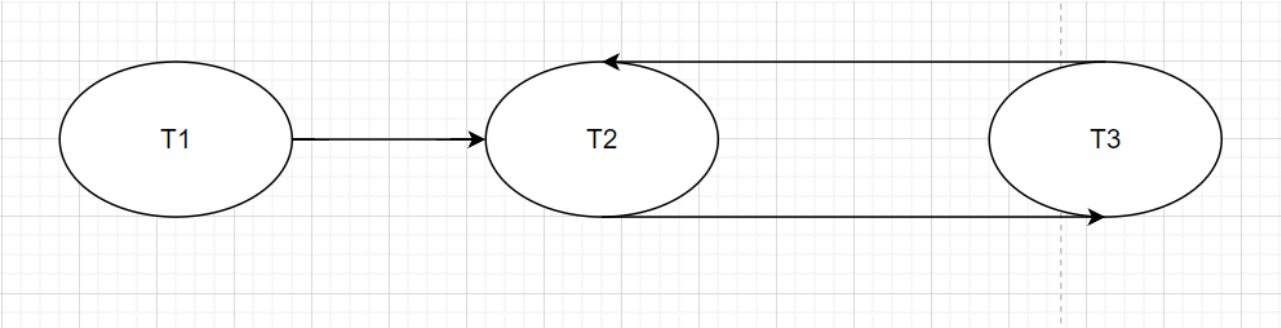


S1 is conflict-serializable. T1's read1(z) and T2's write2(z) are om conflict. T3's r3(x) and w1(x)for T1 are in conflict

	S1	
T1	T2	T3
R1(x) R1(z) W1(x)	R2(z) R2(y) W2(z) W2(y)	R3(x) R3(y) W3(y)

The equivalent serial schedule is: **T3 ------> T1 ------> T2**

Q2(b)



Schedule S2 is not serializable. transaction 3 r3(x) and transaction 1 w1(x) are in conflict.

The r1(z) operation for transaction1 conflicts with the w2(z) fro transaction2

w3(y) conflicts with w2(y)

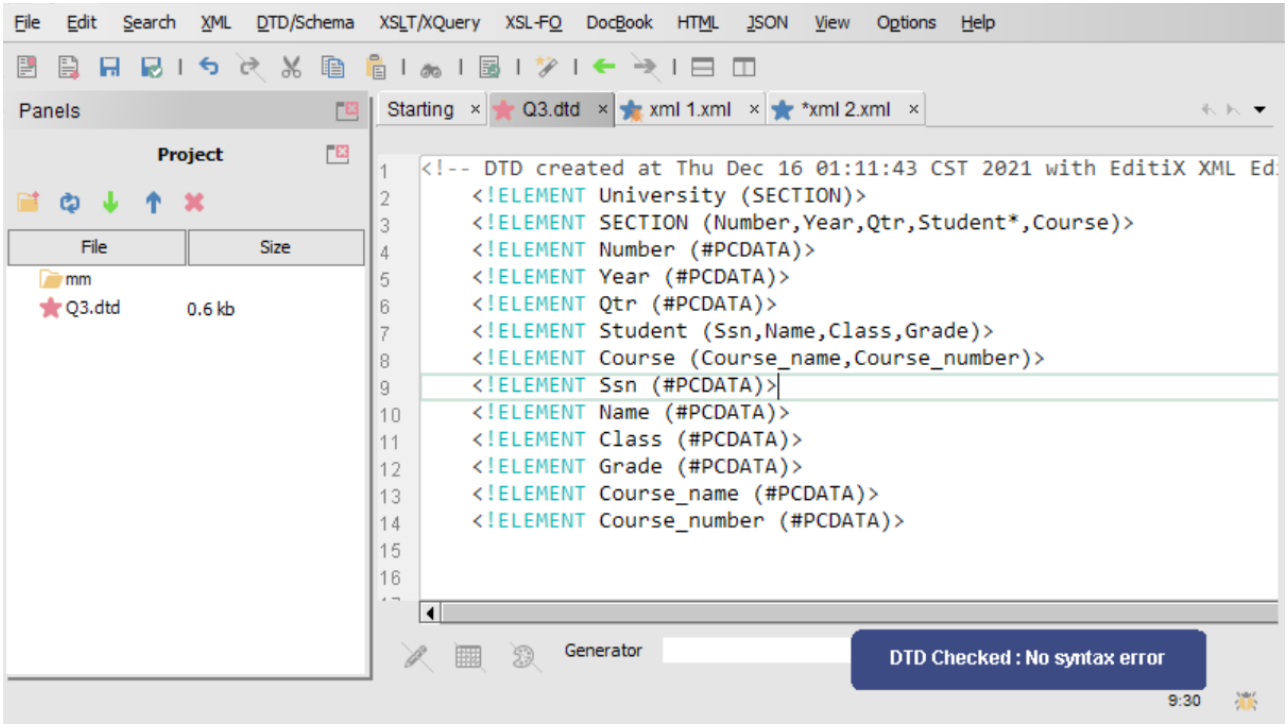
iii. it's not serialized, so the output `Y` value after serialization is different from the original Y value.

III. XML Data Modeling

Q3(a)

```
<!ELEMENT University (SECTION)>
<!ELEMENT SECTION (Number,Year,Qtr,Student*,Course)>
<!ELEMENT Number (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Qtr (#PCDATA)>
<!ELEMENT Student (Ssn,Name,Class,Grade)>
<!ELEMENT Course (Course_name,Course_number)>
<!ELEMENT Ssn (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Class (#PCDATA)>
<!ELEMENT Grade (#PCDATA)>
<!ELEMENT Course_name (#PCDATA)>
<!ELEMENT Course_number (#PCDATA)>
```

The below image shows `DTD file` can be validated by EditiX

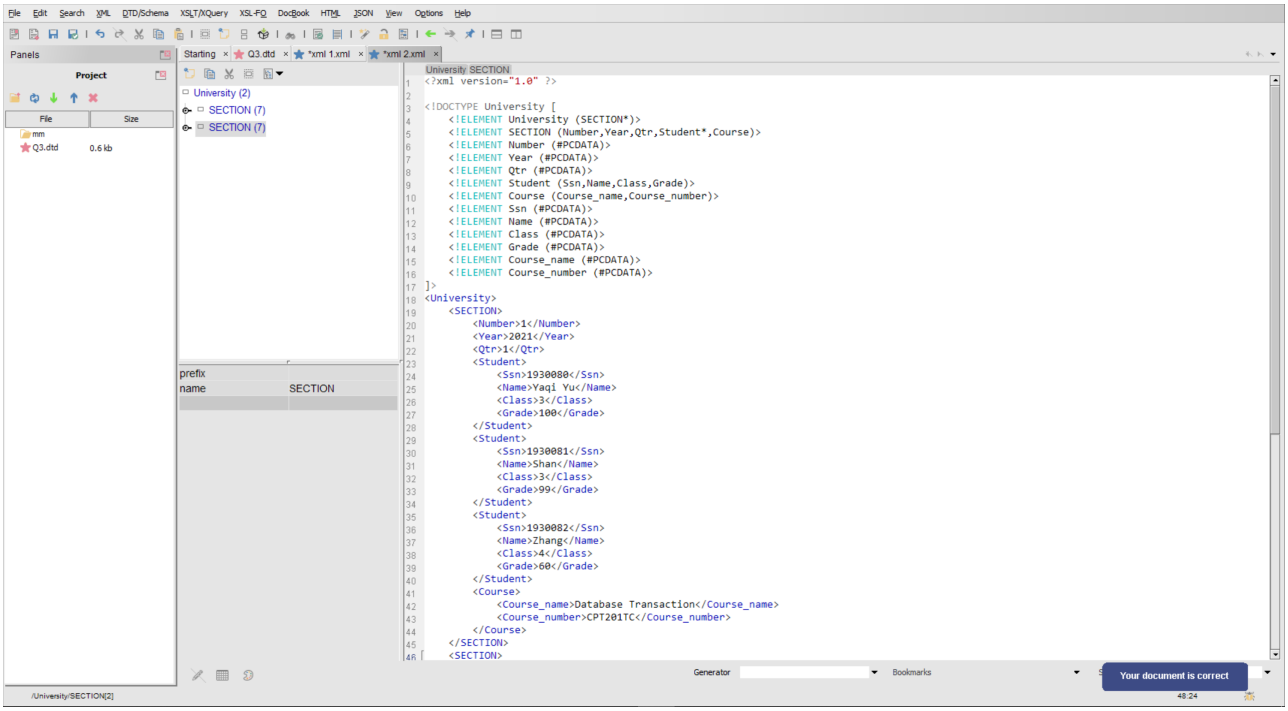


Q3(b)

```
<?xml version="1.0" ?>

<!DOCTYPE University [
    <!ELEMENT University (SECTION*)>
    <!ELEMENT SECTION (Number,Year,Qtr,Student*,Course)>
    <!ELEMENT Number (#PCDATA)>
    <!ELEMENT Year (#PCDATA)>
    <!ELEMENT Qtr (#PCDATA)>
    <!ELEMENT Student (Ssn,Name,Class,Grade)>
    <!ELEMENT Course (Course_name,Course_number)>
    <!ELEMENT Ssn (#PCDATA)>
    <!ELEMENT Name (#PCDATA)>
    <!ELEMENT Class (#PCDATA)>
    <!ELEMENT Grade (#PCDATA)>
    <!ELEMENT Course_name (#PCDATA)>
    <!ELEMENT Course_number (#PCDATA)>
]>
<University>
    <SECTION>
        <Number>1</Number>
        <Year>2021</Year>
        <Qtr>1</Qtr>
        <Student>
            <Ssn>1930080</Ssn>
            <Name>Yaqi Yu</Name>
            <Class>3</Class>
            <Grade>100</Grade>
        </Student>
        <Student>
            <Ssn>1930081</Ssn>
            <Name>Shan</Name>
            <Class>3</Class>
            <Grade>99</Grade>
        </Student>
        <Student>
            <Ssn>1930082</Ssn>
            <Name>Zhang</Name>
            <Class>4</Class>
            <Grade>60</Grade>
        </Student>
        <Course>
```

```
            <Course_name>Database Transaction</Course_name>
            <Course_number>CPT201TC</Course_number>
        </Course>

    </SECTION>
    <SECTION>
        <Number>2</Number>
        <Year>2021</Year>
        <Qtr>2</Qtr>
        <Student>
            <Ssn>1930080</Ssn>
            <Name>Yaqi Yu</Name>
            <Class>3</Class>
            <Grade>98</Grade>
        </Student>
        <Student>
            <Ssn>1930081</Ssn>
            <Name>Shan</Name>
            <Class>3</Class>
            <Grade>60</Grade>
        </Student>
        <Student>
            <Ssn>1930082</Ssn>
            <Name>Zhang</Name>
            <Class>4</Class>
            <Grade>100</Grade>
        </Student>
        <Course>
            <Course_name>Advanced Linear Algebra</Course_name>
            <Course_number>MTH314TC</Course_number>
        </Course>

    </SECTION>
</University>
```

I create Two section instance, each section have one course and three students. The below image shows XML file can be validated.



IV. Object-Relational Database

Q4(a)

Note

1. Instructor

`Instructor` is abstract class, which has 2 attributes:Name as Primary key(String type) and office_number is integer type.

2. Professor

`Instructor` is `professor`'s super class, which means `professor` inherited from `instructor` class.

`Professor` class has some attributes from `Instructor` class (Name, office_Number). And has only professor class attributes: Rank, which is rank type. In this model, rank type is enumeration class, it has 3 literals: lecturer, associate and full_professor

3. Teaching_assistant

`Instructor` is `Teaching_assistant`'s super class, which means `Teaching_assistant` inherited from `instructor` class

`Teaching_assistant` class has some attributes from `Instructor` class (Name, office_Number). And has only professor class attributes: number_year, which is integer type.

4. Course

`Course` class is association class of `Teaching_assistant` and `Professor` class. `Course` class has one attribute: Course code is a primary key.

And one course instance have 0-2 teaching_assistants. one TA has 3-6 courses.

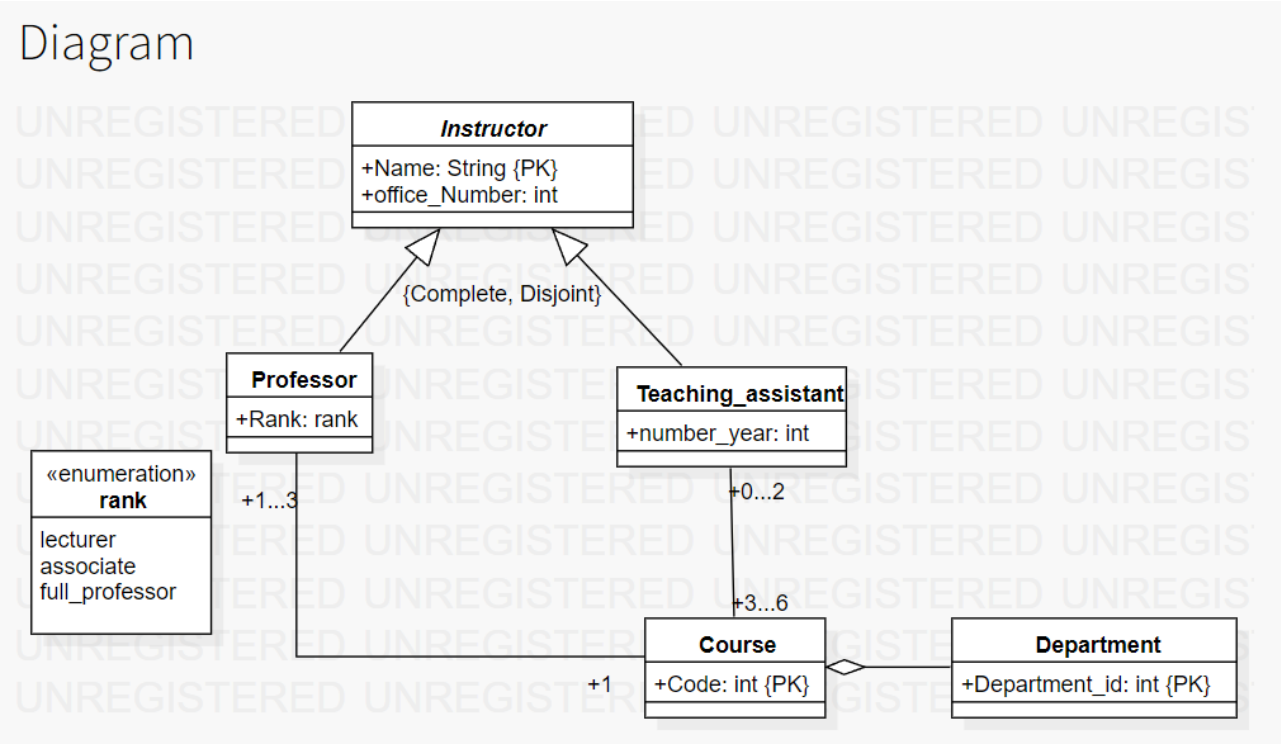
One course instance have 1-3 professor, one professor has only one course.

5. Department

`Course` class is offered by a department, so `department` is aggregated class of `Course` class, which have one attribute: department_id(not from cw2 question, writted by the author).

Diagram

Notation`+` represent the element is public



Q4(b)

User(user_id, first_name, last_name, dob);

Instructor(instructor_id, Is_full_time, total_learners);

Course(course_code, description,instructors,Learners);

Learner(learner_id, occupation, social_link);

Lesson(lesson_id, title, course, content);

The above converts the classes in the uml diagram into tables, and there is a relationship between tables and tables. And the elements of the primary key are added to each table and marked with an underscore.

V. Data Warehousing

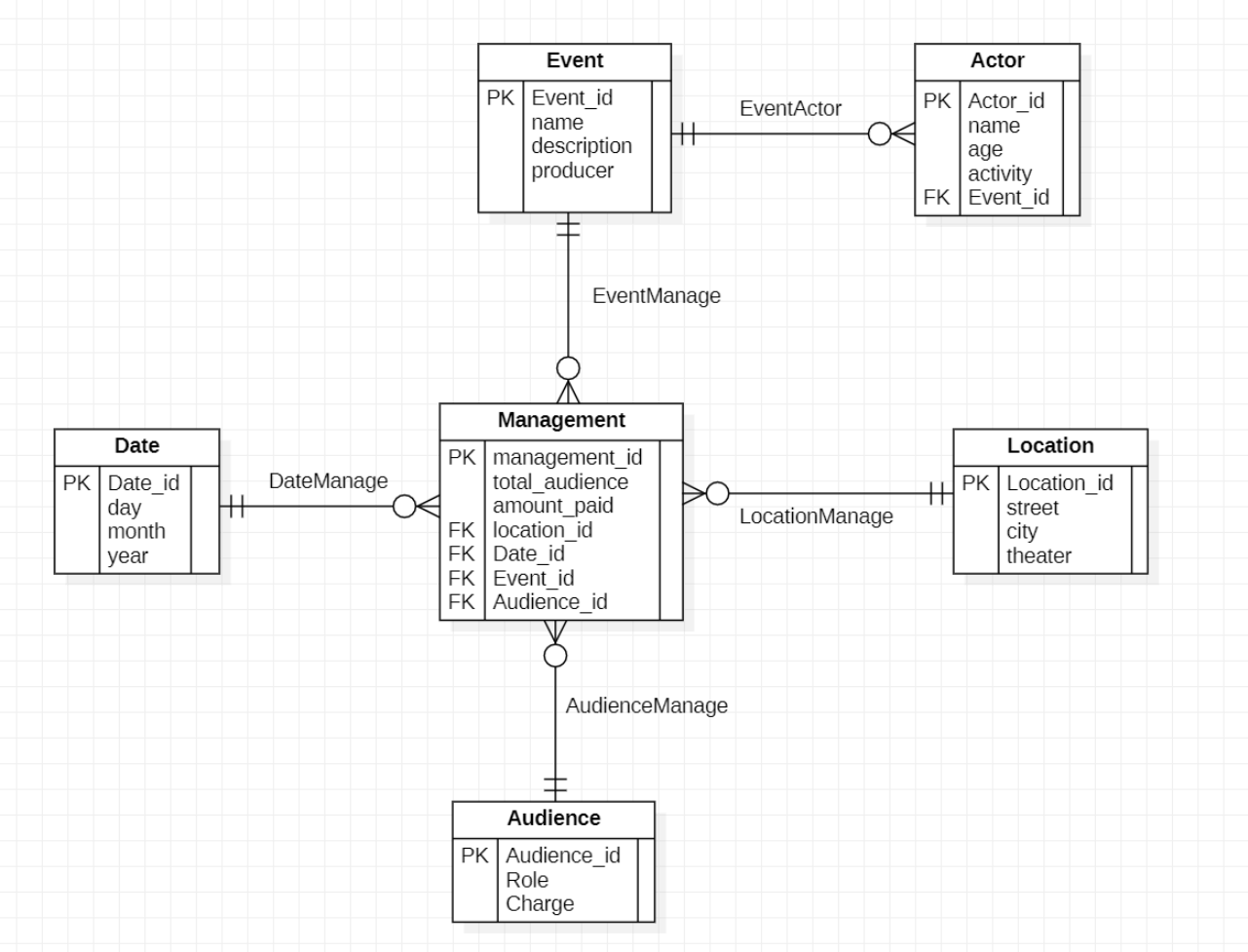
Q5(a)

I think snowflake schema is the most appropriate to model this data warehouse.

Because multiple levels of dimension tables. Only the fact table and the dimension tables are joined in star schemas, resulting in simpler and faster SQL queries. Snowflake schemas are easier to maintain because they done't have any redudant data. Data warehouses benefit from snowflake schemas, while datamarts with basic relationships benefit from star schemas.

It is more empirically useful mainly because they are independent and also separable in a scenario whereby the relation schemas are in pairs and incorporable in nature.

Q5(b)



Dimension table:

1. management(magement_id, total_audience, amount_paid, location_id, Date_id, Event_id, Audience_id)
2. Event(Event_id,name,description,producer,Actor_id)

Fact table:

1. Actor(Actor_id,name,age,activity)
2. Location(Location_id,street,city,theater)
3. Audience(Audience_id, Role, Charge)
4. Date(Date_id, day, month, year)

Relationship:

1. DateManage: Management table(Date_id)->Date table
2. EventManage: Management table(Event_id)-> Event table
3. LocationManage: Management table(location_id)->Location table
4. AudienceManage: Management table(Audience_id)->Audience table
5. ActorEvent: Event table(actor_id)->Actor table