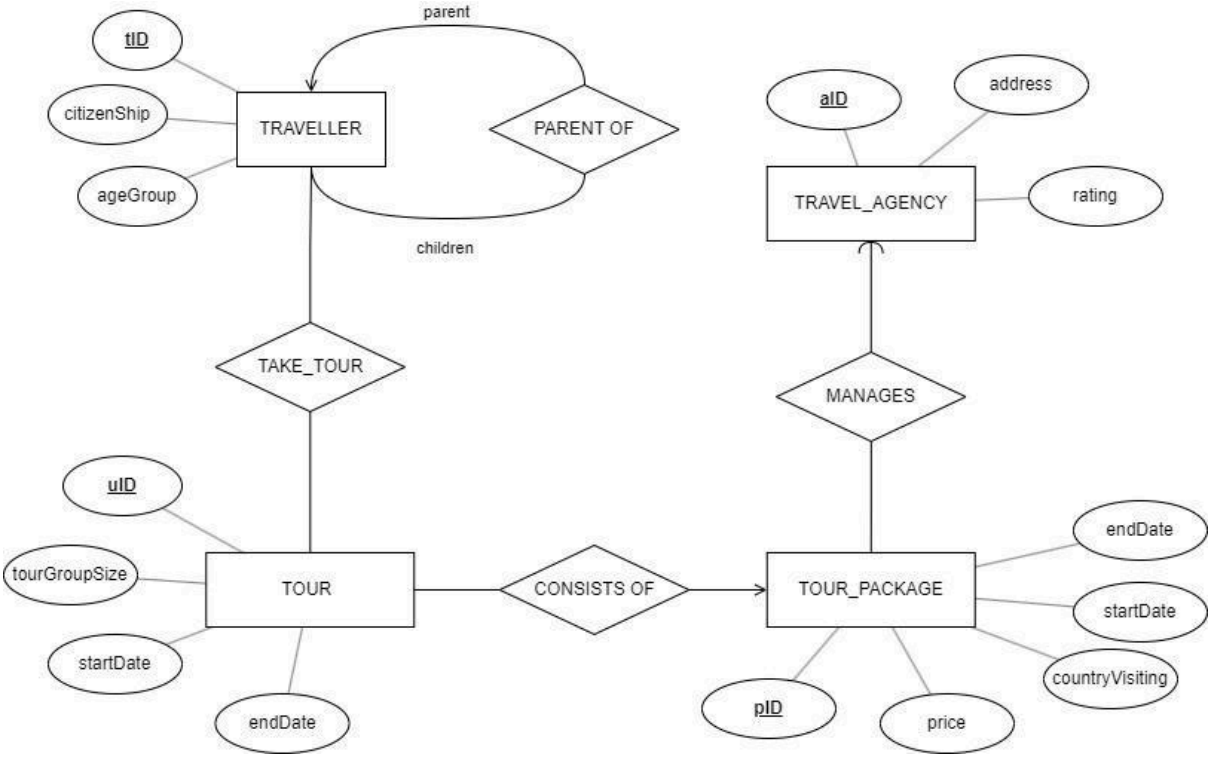
24th SCSE – Past Year Paper Solution (2023 – 2024 Semester 1)
SC2207/CZ2007 – Introduction to Databases

1 (a)



(b)

$LastYearTours := \sigma_{startDate \leq 31-01-2022 \; OR \; endDate \geq 01-01-2022} TOURS$

$R1 := \Pi_{uID, tID \rightarrow childID} TAKE\_TOUR$

$R2 := \Pi_{uID, tID \rightarrow parentID} TAKE\_TOUR$

$R3 := (R1 \bowtie R2)$

$ParentChild := R3 \bowtie_{PARENT.parentTravellerID = R3.parentID \; AND \; PARENT.childTravellerID = R3.childID} PARENT$

$ParentChildTours := LastYearTours \bowtie ParentChild$

$ParentIDs := \Pi_{tID}\left(TRAVELLER \bowtie_{ParentChildTours.parentID = TRAVELLER.tID} ParentChildTours\right)$

$UniqueParentIds := \delta(ParentIds)$

$Answer := TRAVELLER \bowtie UniqueParentIds$

(c)

$R4 := ParentChildTours \bowtie_{ParentChildTours.uID = TOURS.uID} TOURS$

$TourCount := \gamma_{aID,\, COUNT(uID) \rightarrow tourCount} R4$

$MaxTour := \gamma_{MAX(tourCount)} TourCount$

$MaxTourIDS := \Pi_{aID}\left(\sigma_{tourCount = MaxTour} TourCount\right)$

$Answer := TRAVEL\_AGENCY \bowtie MaxTourIDS$

2 (a) For $AB \rightarrow C$, each pair of AB given can be used to determine a single value of C:

- $(a1, b1, c1)$
- $(a2, b2, c2)$
- $(a3, b1, c1)$
- $(a4, b2, c2)$

For $C \rightarrow B$, each value of C can be used to determine a single value of B:

- $(c1, b1)$
- $(c2, b2)$

However, for $A \rightarrow D$, there is a case that shows that a value of A cannot be used to determine a single of D (more than one value of D is associated with a single value of A), that is $(a2, d2)$ and $(a2, d1)$ in line 2 and 6 respectively.

Thus, only the first two functional dependency holds with respect to the given table.

(b) The keys for this schema are AC and AB, as other pairs cannot be used to determine the other attributes of the schema.

For a schema to be in Third Normal Form (3NF), each functional dependency (FD) $X \rightarrow Y$ must be either:

- A trivial dependency
- X contains a key
- Every attribute in Y is contained in a key

However, for the FD $A \rightarrow D$, the attribute D is not contained in any of the keys above, violating the rules above. Thus, this table is not in 3NF.

There are several consequences if the table is not in 3NF, causing anomalies such as:

1. Redundancy
   AD pair values are repeated across the table.
2. Update
   Updating a value of D that is associated to a specific value of A could leave the other value of D unchanged.
3. Insertion
   Insertion a record to the table can be not consistent with the previous records (e.g. an inserted record with a specific value of A could have a different value of D)

(c) Applying 3NF decomposition onto the table will split it into two tables:

- $R_1(A, B, C)$
- $R_2(A, D)$

The schema is not necessarily smaller as there are repeated columns (in this case that is A). There is a way to improve the table R, that is by enforcing a stricter normalization like Boyce-Codd normal form (BCNF) which reduces the redundancy further at the cost of losing some FDs.

3 (a) (i)

```
SELECT *
FROM CLUB
ORDER BY yearFounded DESC
```

(ii)

```
SELECT name, gender, age
FROM STUDENT
WHERE name LIKE 'J%';
```

(iii)

```
SELECT name, gender
FROM STUDENT
WHERE age = (
      SELECT MIN(age)
      FROM STUDENT
);
```

(b)

```
SELECT s.name, c.name, m.position
FROM STUDENT s
JOIN MEMBERSHIP m ON m.sID = s.ID
JOIN CLUB c ON c.cID = m.cID
WHERE m.yearJoin <= 2020;
```

(c)

```
WITH AvailableActivites AS (
      SELECT m.sID AS sID, COUNT(*) AS count
      FROM ACTIVITY a
      JOIN MEMBERSHIP m ON m.cID = a.cID
      WHERE a.date >= CURRENT_DATE()
      GROUP BY m.sID
)
SELECT s.name, c.count
FROM AvailableActivities c
JOIN STUDENT s ON s.sID = c.sID;
```

(d)

```
SELECT s.name
FROM STUDENT s, MEMBERSHIP m
WHERE s.sID = m.sID AND s.age >= 21 AND m.yearJoin <= 2020
GROUP BY s.sID, s.name
HAVING COUNT(m.cID) > 4;
```

4 **(a)** **(i)**

```
ALTER TABLE STUDENT
ADD INDEX StudentIndex (gender, age);
```

(ii)

```
CREATE TRIGGER newStudent
AFTER INSERT
ON STUDENT
REFERENCING NEW ROW AS newRow
FOR EACH ROW
BEGIN
    IF newRow.gender = “female” THEN
        INSERT INTO MEMBERSHIP(cID, sID, position, yearJoin)
        VALUES (“YWS012”, newRow.sID, “member”, YEAR(GETDATE())),
        (“SU0001”, newRow.sID, “member”, YEAR(GETDATE()));
    ELSE
        INSERT INTO MEMBERSHIP(cID, sID, position, yearJoin)
        VALUES (“SU0001”, newRow.sID, “member”, YEAR(GETDATE());
    ENDIF
END
```

(b)

- Structured data has a predefined schema, while unstructured data does not have.
- Structured data is less flexible to accommodate data with a different structure, while unstructured data is much more flexible with different data formats.
- Structured data is easier to be queried and analyzed, while unstructured data is harder.

(c)

```
<!DOCTYPE hotels[
    <!ELEMENT hotels (hotel*)>
    <!ELEMENT hotel(room+)>
    <!ELEMENT room (#PCDATA)>
    <!ATTLIST hotel Hname ID>
    <!ATTLIST hotel UnitNo CDATA>
    <!ATTLIST hotel StreetName CDATA>
    <!ATTLIST hotel Zipcode CDATA>
    <!ATTLIST hotel Country CDATA>
    <!ATTLIST room RoomNo ID>
    <!ATTLIST room Type CDATA>
    <!ATTLIST room Prices CDATA>
    <!ATTLIST room Currency CDATA>
]>
```

(d) a

```
<Persons>
  <Person name=”Alan” age=”32”>
    <Roles>
      <Role name=”Teaching Asst.” school=”SCSE” course=”SC2207”>
      <Role name=”PhD Student” school=”MAE” supervisor=”Prof. David>
    </Roles>
  </Person>
  <Person name=”Belinda” age=”28”>
    <Roles>
      <Role name=”Student” school=”SCSE”>
    </Roles>
    <Courses>
      <Course name=”SC2207” grade=”A+”>
      <Course name=”SC2005” grade=”A”>
    </Courses>
  </Person>
</Persons>
```

Solver: Clayton Fernalo (cfernalo001@e.ntu.edu.sg)