

DATA WAREHOUSE EXERCISE (Travel agency)

A travel agency organizes guided trips for tourists of different nationalities. The agency wants to design a data warehouse in order to know the main trends about the participation in trips with respect to the characteristics of trips and participants.

The following is the relational schema of the operational database.

TRIP (CodeT, DestinationName, CategoryName, CodeGuide, DepartureDate) // *Examples of categories: business trip, adventure travel, ...*

DESTINATION (DestinationName, Description, TypeName, NationName) // *Examples of destination types: seaside, mountain, art city, ...*

TRAVELING (CodeTrip, CodeParticipant, PaidPrice, Discount) // *PaidPrice is the price paid by a specific participant for the trip; it may differ for each participant. Discount is the percentage of discount obtained by the participant.*

PARTICIPANT (CodeP, Name, Surname, Age, NationName)

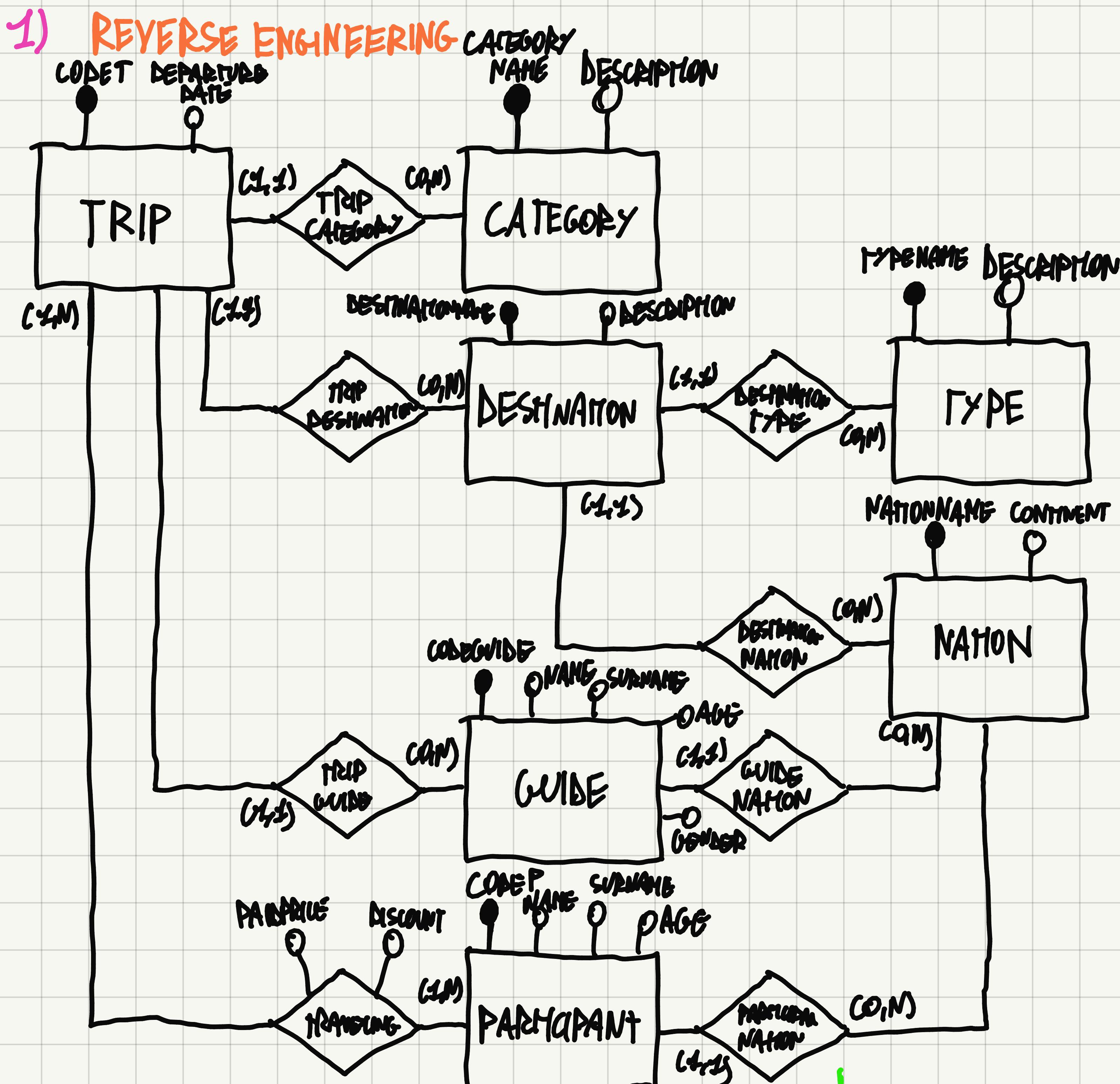
NATION (NationName, Continent)

GUIDE (CodeGuide, Name, Surname, Age, Gender, NationName)

TYPE (TypeName, Description)

CATEGORY (CategoryName, Description)

1. Perform the reverse engineering of the given logical schema into a conceptual schema (Entity-Relationship model).
2. With respect to the produced ER diagram, discover the fact(s) that are useful specifically to answer the queries reported below. For each of these facts:
 - a. Produce the attribute tree (with pruning and grafting).
 - b. Produce the conceptual schema (fact schema).
 - c. Produce the glossary.
3. Produce a logical schema consistent with the conceptual schema and such that it allows performing the queries below.
4. Write in SQL the following queries against the designed logical schema:
 - a. Total income and total number of participations with respect to departure date, type and nation of the destination, and age and continent of the participants.
 - b. Average discount with respect to destination continent, month, and participant nationality.
 - c. Total number of participations with respect to destination, semester and guide's nationality.
Include in the answer also the aggregations computed using only one and two of the three attributes.
 - d. Aggregate the total income by month, by semester and by year (include in the answer the aggregations computed only by month, only by semester and only by year).
 - e. Code, name and surname of the guide(s) having supervised the largest number of participants, among the guides from Africa.

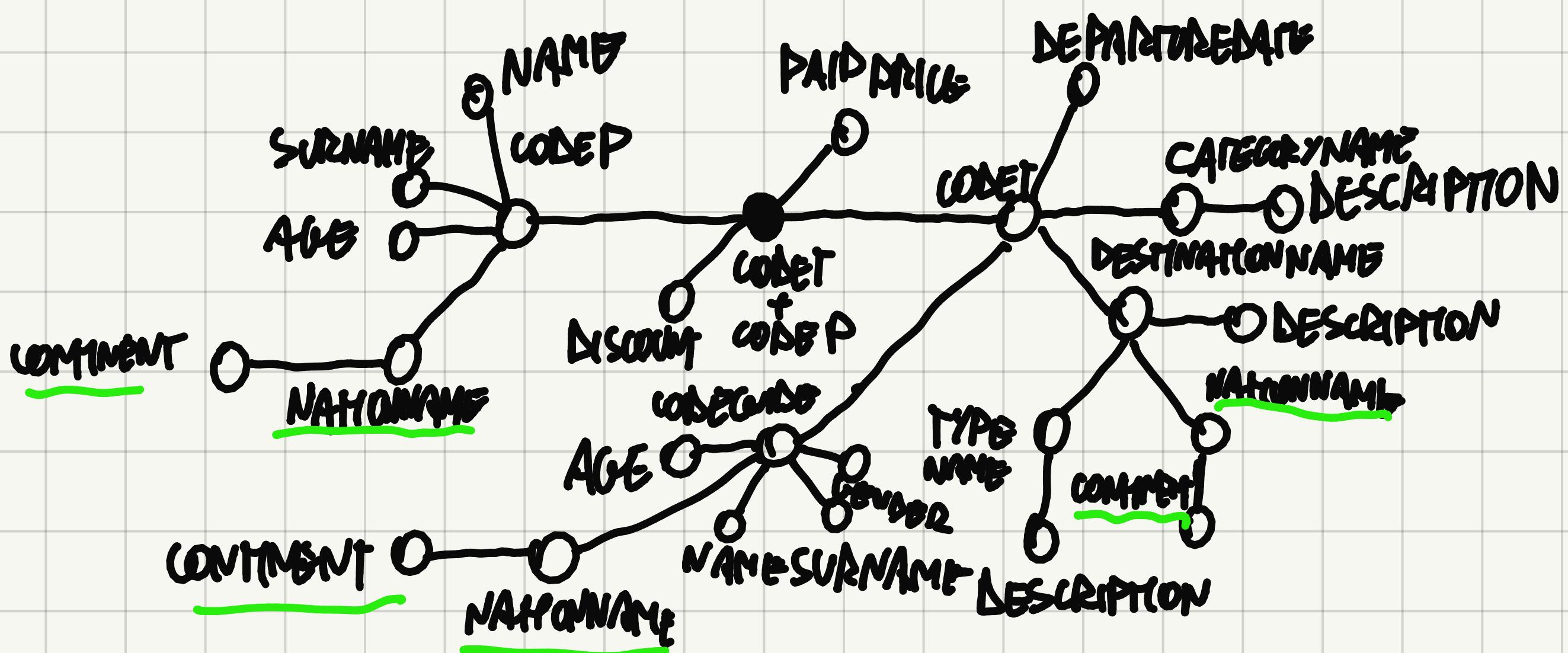


2)

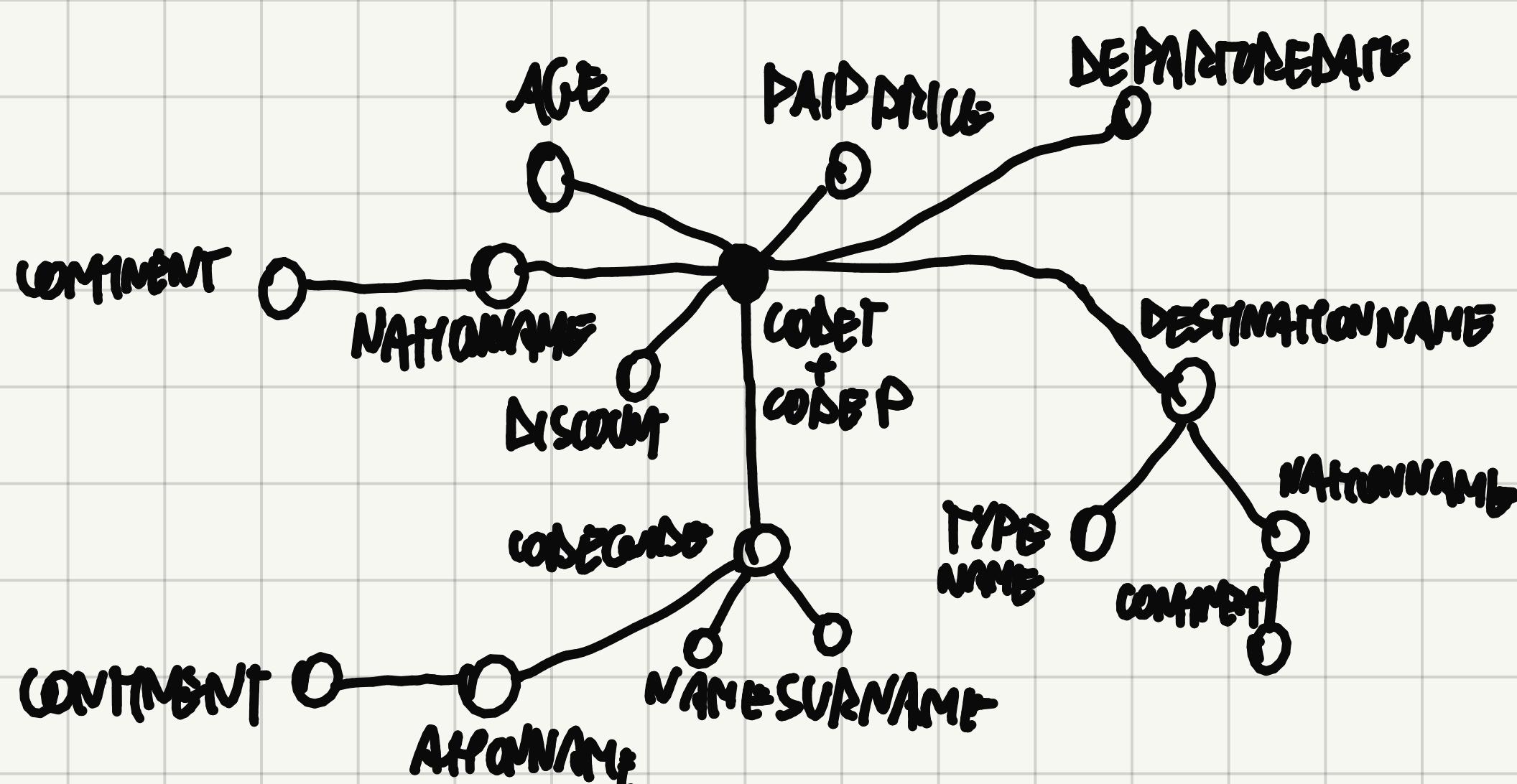
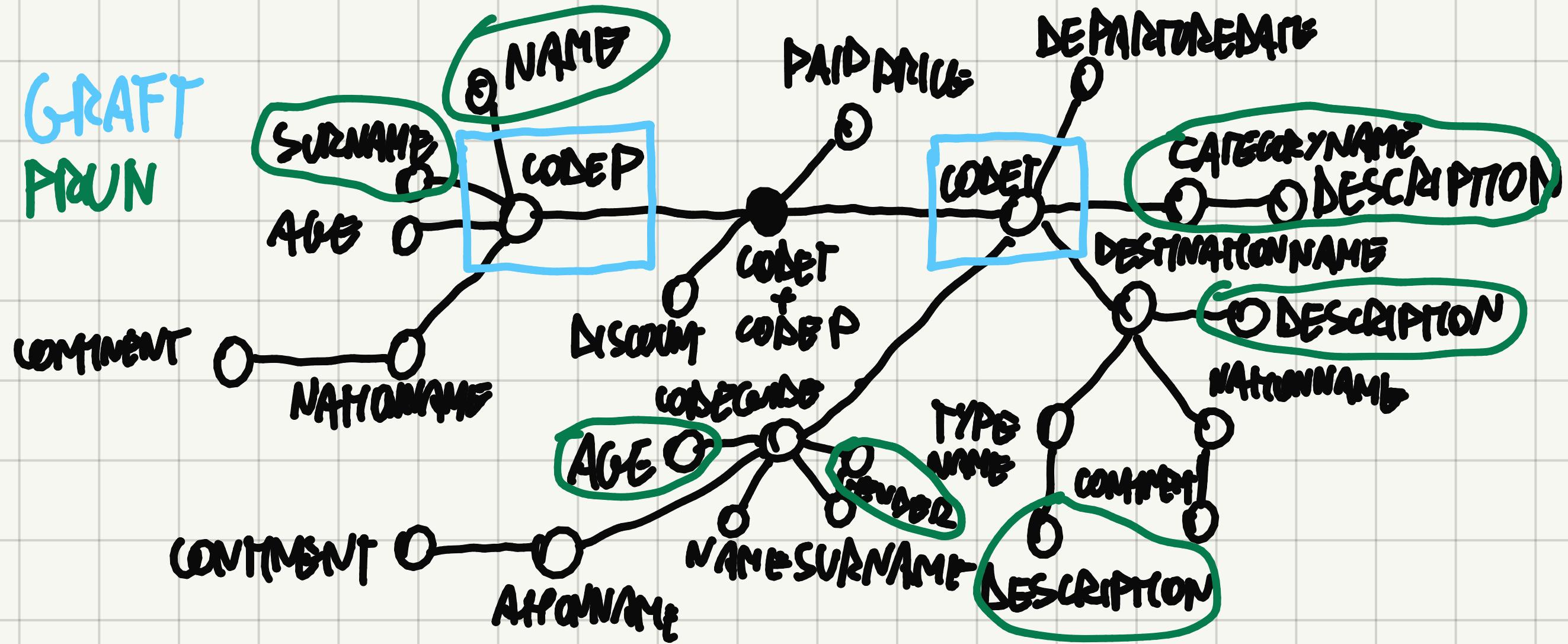
(a) ATTRIBUTE TREE

FACTS : TRAVELING (n:n RELATIONSHIP)

! NOTICE THAT
NATION APPEARS
MORE THAN ONE

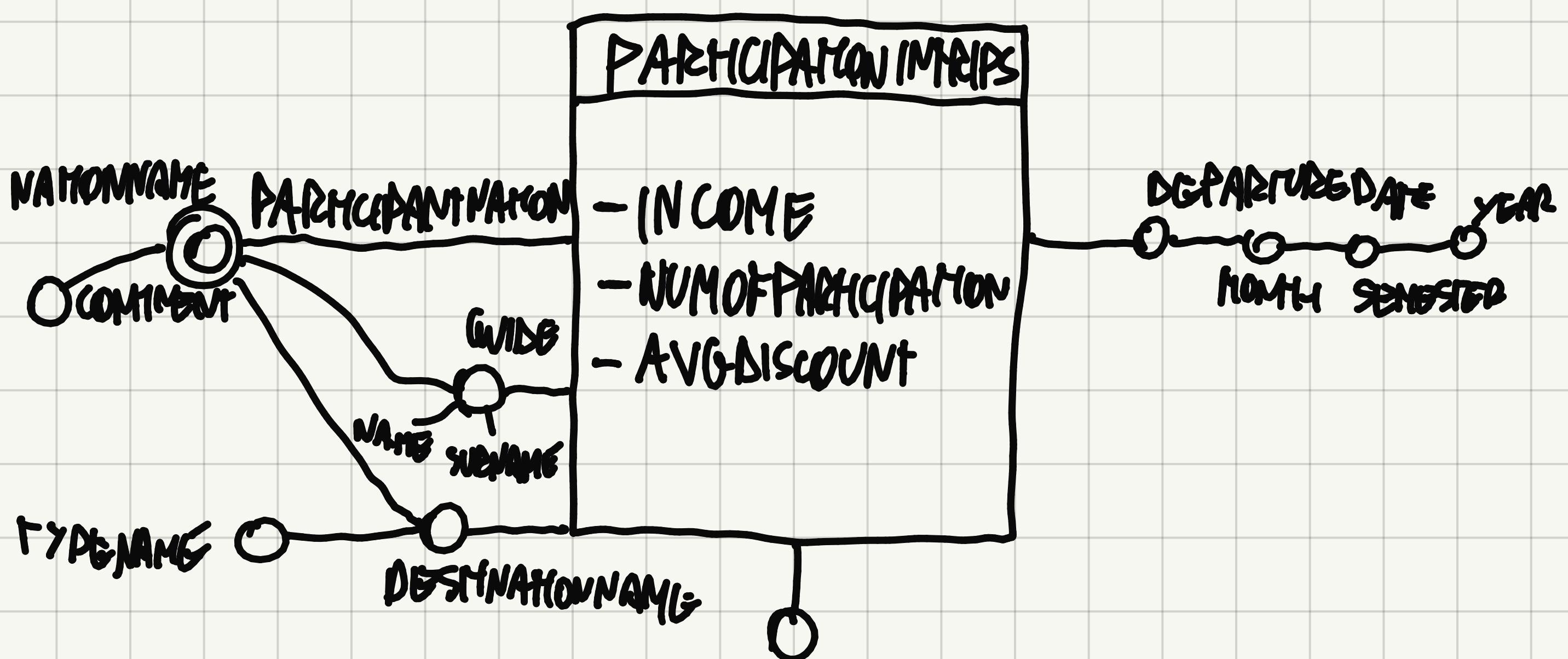


**GRAFT
PRUN**



b) **FACT SCHEMA**

④ **SHARED HIERARCHY**



c) **GLOSSARY**

NUMOFPARTICIPATIONS

```

SELECT T.DESTINATIONNAME, T.DEPARTUREDATE, T.CODEGUIDE, P.AGE
FROM TRAVELING AS T, TRIP AS T, PARTICIPANT AS P
WHERE T.CODETRIP = T.CODET AND T.CODEPARTICIPANT = P.CODEP
GROUP BY T.DESTINATIONNAME, T.DEPARTUREDATE, T.CODEGUIDE, P.AGE,
        P.NATIONALNAME, COUNT(*)
    
```

INCOME

SELECT
FROM
WHERE
GROUP BY

T.DESTINATIONNAME, T.DEPARTUREDATE, T.CODEGUIDE, P.AGE,
TRAVELING AS TV, TRIP AS T, PARTICIPANT AS P
TV.CODETRIP = T.CODE AND TV.CODEPARTICIPANT = P.CODE
T.DESTINATIONNAME, T.DEPARTUREDATE, T.CODEGUIDE, P.AGE,

P.NATIONNAME, SUM(N.PAIDPRICE)

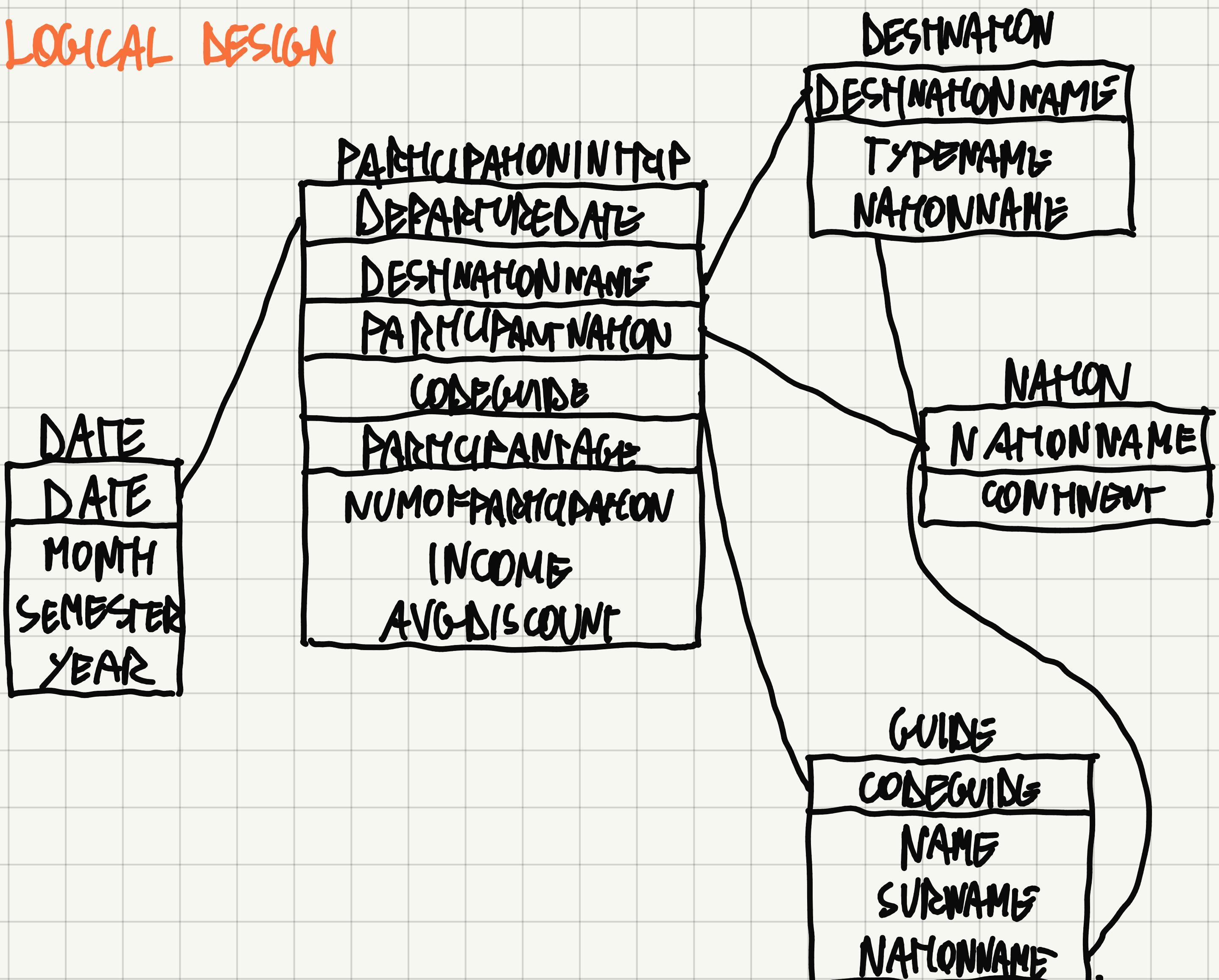
Avg-DISCOUNT

SELECT
FROM
WHERE
GROUP BY

T.DESTINATIONNAME, T.DEPARTUREDATE, T.CODEGUIDE, P.AGE,
TRAVELING AS TV, TRIP AS T, PARTICIPANT AS P
TV.CODETRIP = T.CODE AND TV.CODEPARTICIPANT = P.CODE
T.DESTINATIONNAME, T.DEPARTUREDATE, T.CODEGUIDE, P.AGE,

P.NATIONNAME, AVG(TV.DISCOUNT)

3) LOGICAL DESIGN



4) QUERIES

N.COMMNGT, SUM(P.INCOME), SUM(P.NUMOFPARTICIPATION)

(2) SELECT P.DEPARTUREDATE, D.TYPENAME, D.NATIONNAME, P.PARTICIPANTAGE,
FROM PARTICIPATIONINTRIPS AS P, DESTINATION AS D, NATION AS N
WHERE P.DESTINATIONNAME = D.DESTINATIONNAME AND P.PARTICIPATION
N.COMMNGT
GROUP BY P.DEPARTUREDATE, D.TYPENAME, D.NATIONNAME, P.PARTICIPANTAGE,

b) SELECT N.COMMNGT, DA.MONTH, P.PARTICIPANTNATION,
SUM(P.NUMOFPARTICIPATION * P.AVGDISCOUNT) / SUM(P.NUMOFPARTICIPATION)
AS DA
FROM PARTICIPATIONINTRIPS AS P, DESTINATION AS D, NATION AS N, DATE
= N.NATIONNAME AND P.DEPARTUREDA.DATE
WHERE P.DESTINATIONNAME = DA.DESTINATIONNAME AND D.NATIONNAME =
N.COMMNGT, DA.MONTH, P.PARTICIPANTNATION

c) SELECT P.DESTINATIONNAME, D.SEMESTER, G.NATIONNAME,
SUM(P.NUMOFPARTICIPATION)
FROM PARTICIPATIONINTRIPS AS P, DATE AS D, GUIDE AS G
WHERE P.DEPARTUREDATE = D.DATE AND P.GUIDEGUIDE = G.GUIDEID

GROUP BY P.DESTINATIONNAME, D.SEMESTER, G.NATIONNAME WITH

DIFFERENT HIERARCHIES AND IF

* WE CAN CHOOSE BETWEEN:

CUBE → GENERATES ALL THE POSSIBLE COMBINATION OF ATTRIBUTES

ROLL UP → ALL THE ATTRIBUTES OF THE GROUPBY BELONG TO THE SAME HIERARCHY

OUR FINAL RESULT IS :

```
SELECT P.DESTINATIONNAME, D.SEMESTER, G.NATIONNAME,  
SUM(P.NUMOFPARTICIPANTS)  
FROM PARTICIPATIONINTRIPS AS P, DATE AS D, GUIDE AS G  
WHERE P.DEPARTUREDATE=D.DATE AND P.CODEGUIDE=G.CODEGUIDE  
GROUP BY P.DESTINATIONNAME, D.SEMESTER, G.NATIONNAME WITH CUBE
```

d) SELECT D.YEAR, D.SEMESTER, D.MONTH, SUM(P.INCOME)

FROM PARTICIPATIONINTRIPS AS P, DATE AS D

WHERE P.DEPARTUREDATE=D.DATE

GROUP BY D.YEAR, D.SEMESTER, D.MONTH WITH ROLLUP

e) WE NEED A PROPER VIEW

```
CREATE VIEW GLOBENUMPARTICIPANTS(CODEGUIDE, NAME, SURNAME, NUM) AS
```

```
SELECT G.CODEGUIDE, G.NAME, G.SURNAME, SUM(P.NUMOFPARTICIPANTS)
```

FROM PARTICIPATIONINTRIPS AS P, GUIDE AS G, NATION AS N

WHERE P.CODEGUIDE=G.CODEGUIDE AND G.NATIONNAME=N.NATIONNAME

AND N.CONTINENT='AFRICA'

GROUP BY G.CODEGUIDE, G.NAME, G.SURNAME

)

SELECT CODEGUIDE, NAME, SURNAME

FROM GUIDENUMPARTICPANTS

WHERE NUM = (

SELECT MAX(NUM)

FROM GUIDENUMPARTICPANTS

)