

Technologies for Information Systems

Part I (10 points)

prof. L. Tanca – September 7th, 2018

Available time: 25 minutes

Last Name _____
First Name _____
Student ID _____ Signature _____

1. Briefly define data management in pervasive systems and the main problems that must be solved.
2. Describe the two techniques GAV and LAV for data integration in the case of the existence of a Global Schema, and underline the main pros and cons of the two approaches.

- During this part of the exam, students are not allowed to use books or notes.
- Students should answer the theoretical questions using their own words, in order for the teachers to be able to assess their real level of understanding.

Technologies for Information Systems

Part II (23 points)

prof. L. Tanca – September 7th, 2018

Available time: 2h 00m

Last Name _____	
First Name _____	
Student ID _____	Signature _____

PoliBeach is a company owning a chain of beach resorts. All the beach resorts owned by *PoliBeach* offer their customers the same set of packages, each associated with the rental of various objects like beach chairs and beach umbrellas. Each package has a daily rental cost, and the customer purchases the package for a certain number of days. The director of a beach resort may negotiate the sale price with the customers, deciding to apply a discount percentage that may be different for each sale.

The management of *PoliBeach* has now hired you to design a data warehouse to analyze the sales of the packages.

The following is the schema of the operational database used by *PoliBeach*:

CITY (CityName, Country)

BEACHRESORT (BeachResortId, BeachResortName, Director, CityName) // A director may supervise multiple beach resorts.

PACKAGE (Packageld, Description, Category, DailyPrice) // The packages are classified in categories describing their luxury level (e.g., standard, premium, ...). The daily price is expressed in EUR.

AVAILABLERENTALOBJECT (ObjectName, Description) // Examples of rental objects: beach chair, beach umbrella, ...

PACKAGECOMPOSITION (Packageld, RentalObjectName, Quantity) // The packages are associated with the rental of various objects.

CUSTOMER (CustomerId, Surname, GivenName, BirthDate, CityOfResidence)

SALE (CustomerId, Packageld, BeachResortId, StartDate, Duration, DiscountPercentage) // Duration represents the number of days of the rental. DiscountPercentage is a number between 0 and 100.

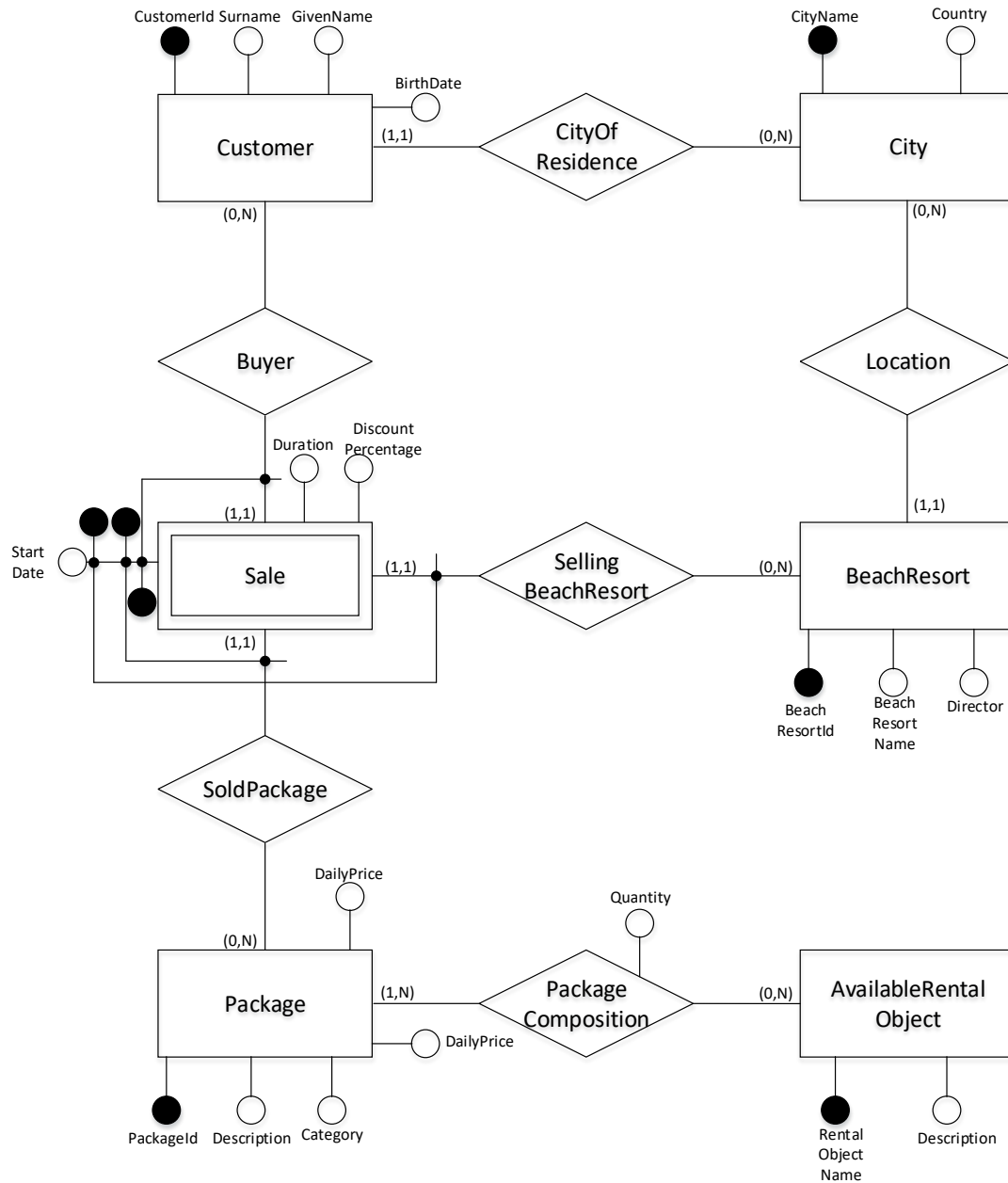
1. (3 points) Perform the reverse engineering of the given logical schema into a conceptual schema (Entity-Relationship model).
2. With respect to the produced ER diagram, discover the fact(s) that are useful specifically for answering the queries reported below. For each of these facts:
 - a. (3 points) Produce the attribute tree (with pruning and grafting).
 - b. (3 points) Produce the conceptual schema (fact schema).
 - c. (2.5 points) Produce the glossary.
3. (3 points) Produce a logical schema consistent with the conceptual schema.
4. Write in SQL the following queries against the designed logical schema:
 - a. (2 points) Compute the total number of sold packages for each package category, beach resort director and city of the customer. Include in the answer also the aggregations computed using only one and two of the three attributes.
 - b. (2.5 points) Compute the average duration of the packages purchased by Italian customers born in 1989 for each city of the beach resort and package id.
 - c. (2 points) Aggregate the total income by start date, month and year (include in the answer the aggregations computed only by start date, only by month and only by year).
 - d. (2 points) Find for each country the beach resort(s) (specify id and name) having earned the greatest income.

NOTE

In SQL (PostgreSQL syntax), given a column of type Date named *d*, you can extract the year from the column using the function **EXTRACT(YEAR FROM *d*)**.

SOLUTION

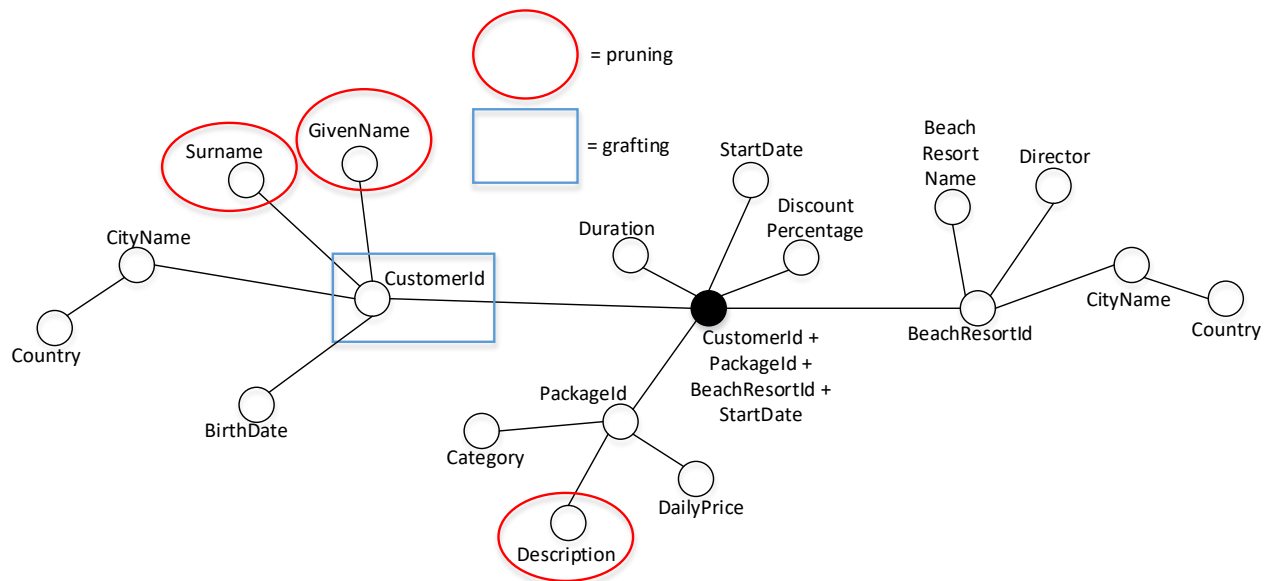
1. Reverse engineering



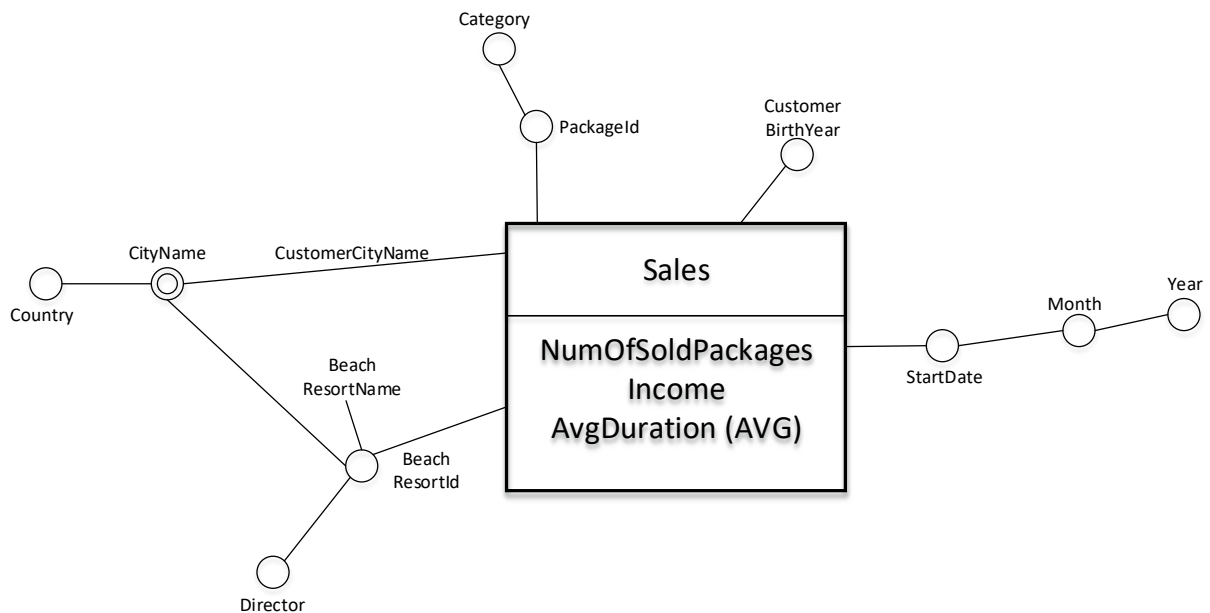
2. Conceptual design

Fact: Sales (Sale entity)

2a) Attribute tree



2b) Fact schema



2c) Glossary

NumOfSoldPackages

```

SELECT S.StartDate, S.BeachResortId, C.CityOfResidence, EXTRACT(YEAR FROM C.BirthDate), S.PackageId,
COUNT(*)
FROM Sales AS S, Customer AS C
WHERE S.CustomerId=C.CustomerId
GROUP BY S.StartDate, S.BeachResortId, C. CityOfResidence, EXTRACT(YEAR FROM C.BirthDate),
S.PackageId

```

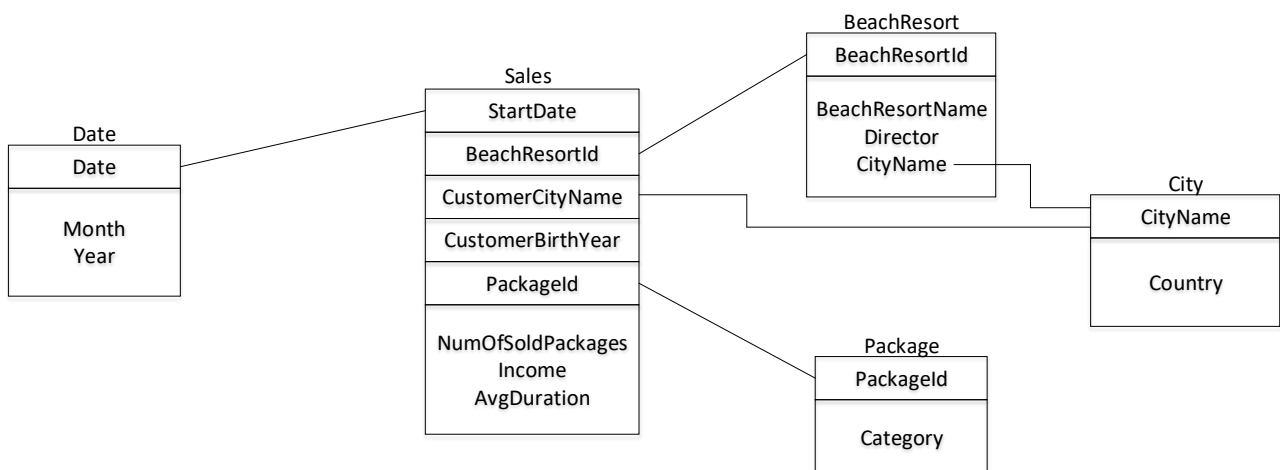
Income

```
SELECT S.StartDate, S.BeachResortId, C.CityOfResidence, EXTRACT(YEAR FROM C.BirthDate), S.PackageId,  
       SUM(S.Duration*P.DailyPrice*(1-0.01*S.PercentageDiscount))  
FROM Sales AS S, Customer AS C, Package AS P  
WHERE S.CustomerId=C.CustomerId AND S.PackageId=P.PackageId  
GROUP BY S.StartDate, S.BeachResortId, C. CityOfResidence, EXTRACT(YEAR FROM C.BirthDate),  
         S.PackageId
```

AvgDuration

```
SELECT S.StartDate, S.BeachResortId, C. CityOfResidence, EXTRACT(YEAR FROM C.BirthDate), S.PackageId,  
       AVG(S.Duration)  
FROM Sales AS S, Customer AS C  
WHERE S.CustomerId=C.CustomerId  
GROUP BY S.StartDate, S.BeachResortId, C. CityOfResidence, EXTRACT(YEAR FROM C.BirthDate),  
         S.PackageId
```

3. Logical design



4. Query answering

4a) Compute the total number of sold packages for each package category, beach resort director and city of the customer. Include in the answer also the aggregations computed using only one and two of the three attributes.

```
SELECT P.Category, B.Director, S.CustomerCityName, SUM(S.NumOfSoldPackages)  
FROM Sales AS S, BeachResort AS B, Package AS P  
WHERE S.BeachResortId=B.BeachResortId AND S.PackageId=P.PackageId  
GROUP BY P.Category, B.Director, S.CustomerCityName WITH CUBE
```

4b) Compute the average duration of the packages purchased by Italian customers born in 1989 for each city of the beach resort and package id.

```

SELECT B.CityName, S.PackageId,
        SUM(S.NumOfSoldPackages * S.AvgDuration)/SUM(S.NumOfSoldPackages)
FROM Sales AS S, BeachResort AS B, City AS C
WHERE S.BeachResortId=B.BeachResortId AND S.CustomerCityName=C.CityName AND C.Country='Italy'
        AND S.CustomerBirthYear=1989
GROUP BY B.CityName, S.PackageId

```

4c) Aggregate the total income by start date, month and year (include in the answer the aggregations computed only by start date, only by month and only by year).

```

SELECT D.Year, D.Month, D.Date, SUM(S.Income)
FROM Sales AS S, Date AS D
WHERE S.StartDate=D.Date
GROUP BY D.Year, D.Month, D.Date WITH ROLLUP

```

4d) Find for each country the beach resort(s) (specify id and name) having earned the greatest income.

```

CREATE VIEW BeachResortIncome (BeachResortId, BeachResortName, Country, TotIncome) AS (
    SELECT B.BeachResortId, B.BeachResortName, C.Country, SUM(S.Income)
    FROM Sales AS S, BeachResort AS B, City AS C
    WHERE S.BeachResortId=B.BeachResortId AND B.CityName=C.CityName
    GROUP BY B.BeachResortId, B.BeachResortName, C.Country
)
SELECT B.Country, B.BeachResortId, B.BeachResortName
FROM BeachResortIncome AS B
WHERE B.TotIncome = (
    SELECT MAX(B2.TotIncome)
    FROM BeachResortIncome AS B2
    WHERE B.Country=B2.Country
)

```