Technologies for Information Systems

Part I (10 points)

prof. L. Tanca – February 1st, 2017

Available time: 25 minutes

Last Name ____

First Name ____

Student ID _______________ **Signature**_____________________________________

1) Describe the dimensional fact model used in the data warehouse context and define its main elements. Provide a small example.

2) Describe the problem of personalization and list the various kinds of personalization with a small explanation of each.

- **During this part of the exam, students are not allowed to consult 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 (22 points)

prof. L. Tanca – February 1st, 2017

Available Time: 2h 00m

Last Name ____

First Name ____

Student ID ________________ **Signature**_____________________________________

PoliRestaurants is a chain of Italian restaurants operating in Lombardy which allows customers to order meals through its website. The customer needs to register, and then can navigate to the website, where he/she can choose one of the restaurants of the chain. At this point it is possible to order dishes and beverages from the chosen restaurant, specifying for each dish or beverage the required number of items. All the PoliRestaurants restaurants offer take-away service, and in some cities also home delivery is available.

UniRestaurants, on the contrary, is a restaurant chain featuring Chinese and Japanese cuisine, which offers comparable services in Piedmont and Liguria. Analogously to PoliRestaurants, all the UniRestaurants restaurants offer the take-away service. Home delivery is available just in some of them. In addition, UniRestaurants does not permit to order beverages.

The two companies have now merged into a unique one named ***UniPoliRestaurants***. The UniPoliRestaurants ownership asks you to integrate the relational databases of the two companies into a unique relational database. You must perform the integration ensuring to lose the least possible amount of information.

The original relational schemas of the two sources are reported below.

PoliRestaurants

CITY (<u>CityName</u>, HomeDeliveryService) *// HomeDeliveryService is a boolean attribute that is true if the restaurants in this city allow the home delivery.*

MANAGER (<u>ManagerSSN</u>, Surname, GivenName, PhoneNumber)

RESTAURANT (<u>RestaurantCode</u>, Name, Address, CityName, ManagerSSN) *// A manager may supervise multiple restaurants.*

CUSTOMER (<u>CustomerSSN</u>, Surname, GivenName, BirthDate)

ORDER (<u>OrderCode</u>, Timestamp, OrderType, ServiceDate, ServiceTime, DeliveryAddress*, RestaurantCode, CustomerSSN) *// OrderType is either 'TakeAway' or 'HomeDelivery'. The DeliveryAddress attribute is null for for take-away orders.*

DISH (<u>DishCode</u>, Name, Description, Cost, IsVegan, WeightGrams) *// IsVegan is a boolean attribute that is true if this dish is vegan. WeightGrams represents the weight of the dish expressed in grams.*

BEVERAGE (<u>BeverageCode</u>, Name, Description, Price, IsAlcoholic) // *IsAlcoholic is a boolean attribute that is true if this beverage is alcoholic.*

ORDEREDDISH (<u>OrderCode</u>, <u>DishCode</u>, Quantity)

ORDEREDBEVERAGE (<u>OrderCode</u>, <u>BeverageCode</u>, Quantity)

UniRestaurants

CITY (CityName, Region) *// Region is either 'Piedmont' or 'Liguria'.*

RESTAURANT (RestaurantCode, Name, Address, HomeDeliveryService, ManagerPhoneContact, CityName) *// HomeDeliveryService is a boolean attribute that is true if the restaurant allows the home delivery.*

CUSTOMER (CustomerRegistrationNr, CustomerSSN, LastName, FirstName, PhoneNumber)

COURSE (CourseCode, Name, Description, CuisineType, Price, WeightKilos) *// CuisineType is either 'Chinese' or 'Japanese'. WeightKilos represents the weight of the course expressed in kilograms.*

INGREDIENT (IngredientName, Category, Supplier) *// The ingredient category may be vegetable, fruit, dairy, …*

COURSEINGREDIENT (CourseCode, IngredientName, Quantity)

BOOKING (Timestamp, CustomerRegistrationNr, BookingType, ServiceDate, ServiceTime, DeliveryAddress*, CourseCode, RestaurantCode, Quantity) *// BookingType is either 'TakeAway' or 'HomeDelivery'. The DeliveryAddress attribute is null for take-away bookings.*

Notes:

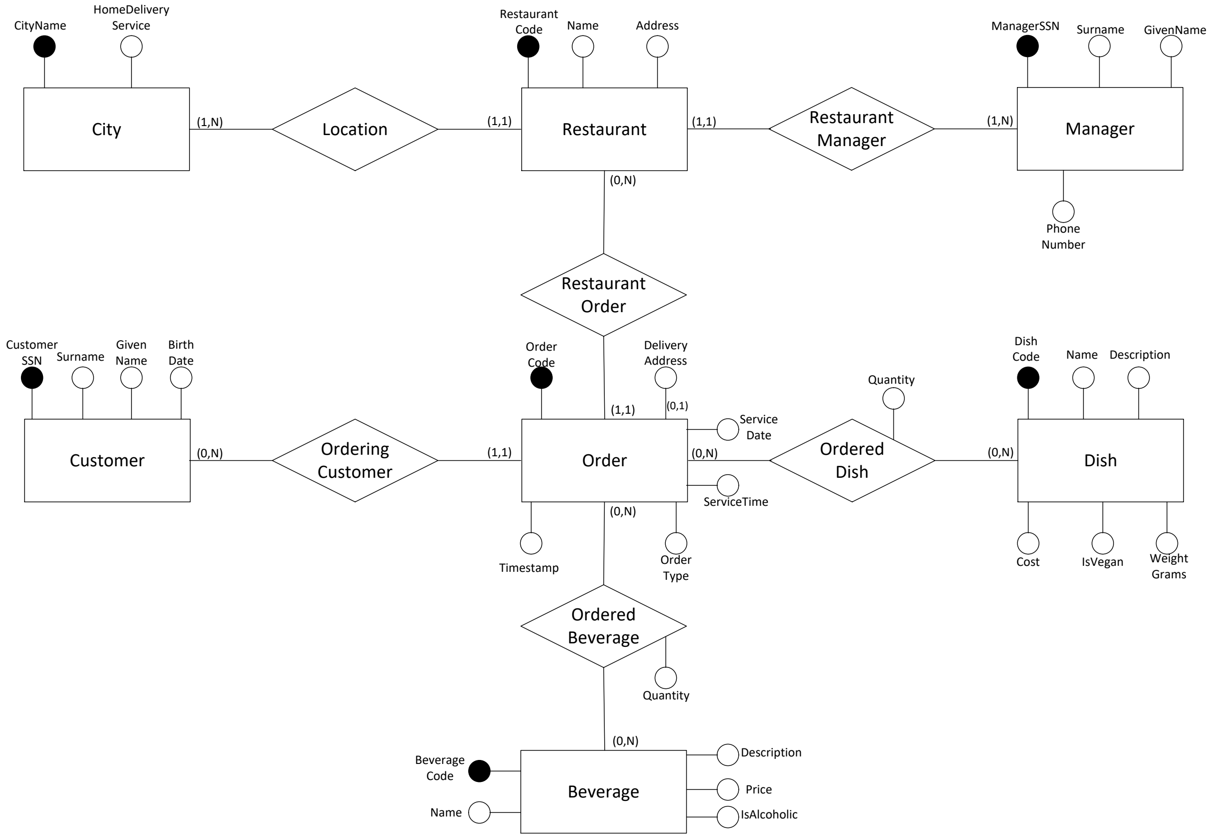
- Since PoliRestaurants and UniRestaurants operate in different geographic areas, you can assume that the customers in the two data sources are disjoint.
- Since PoliRestaurants and UniRestaurants deal with different kinds of cuisine, you can assume that the sets of dishes included in the two data sources are disjoint.

1. **Source schema reverse engineering**. Provide, for each input data source, the reverse engineering from the logical schema to the conceptual model (ER graph). (5 points)
2. **Schema integration**. Design an integrated global conceptual schema (ER graph) for *UniPoliRestaurants* capturing all the data coming from both *PoliRestaurants* and *UniRestaurants*, and provide the corresponding global logical schema. In more detail, follow these steps:
   a. *Related concept identification and conflict analysis and resolution*. Write a table as shown in the exercise sessions, using the following columns: "PoliRestaurants concept", "UniRestaurants concept", "Conflict", "Solution". (3.5 points)
   b. *Integrated conceptual schema* (ER graph). (3.5 points)
   c. *Conceptual to logical translation of the integrated schema*. (2 points)
3. **Query answering and mapping definition**. Consider the query Q "Find timestamp and customer SSN of the take-away orders to restaurants located in Milan or in Turin, which have involved at least one dish with weight greater than 200 grams and price greater than 20 euros".
   a. *Query formulation*. Consider query Q posed on the logical schema of *UniPoliRestaurants* and write it in SQL. (1.5 points)
   b. *Mapping definition*. Write the GAV mappings between the schema of *UniPoliRestaurants* and the two sources using SQL. Write the mappings only for the tables used to answer query Q**.** (4 points)
   c. *Query rewriting*. Show the rewriting of Q on the two data sources using SQL. (2.5 points)

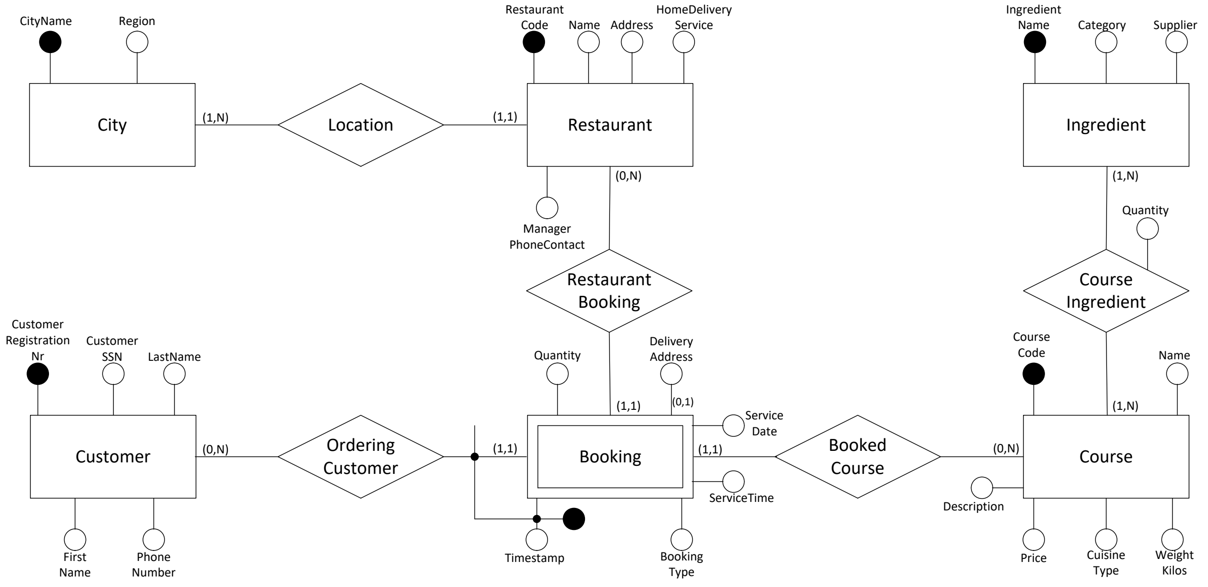
SOLUTION

1. Source schema reverse engineering

PoliRestaurants



UniRestaurants

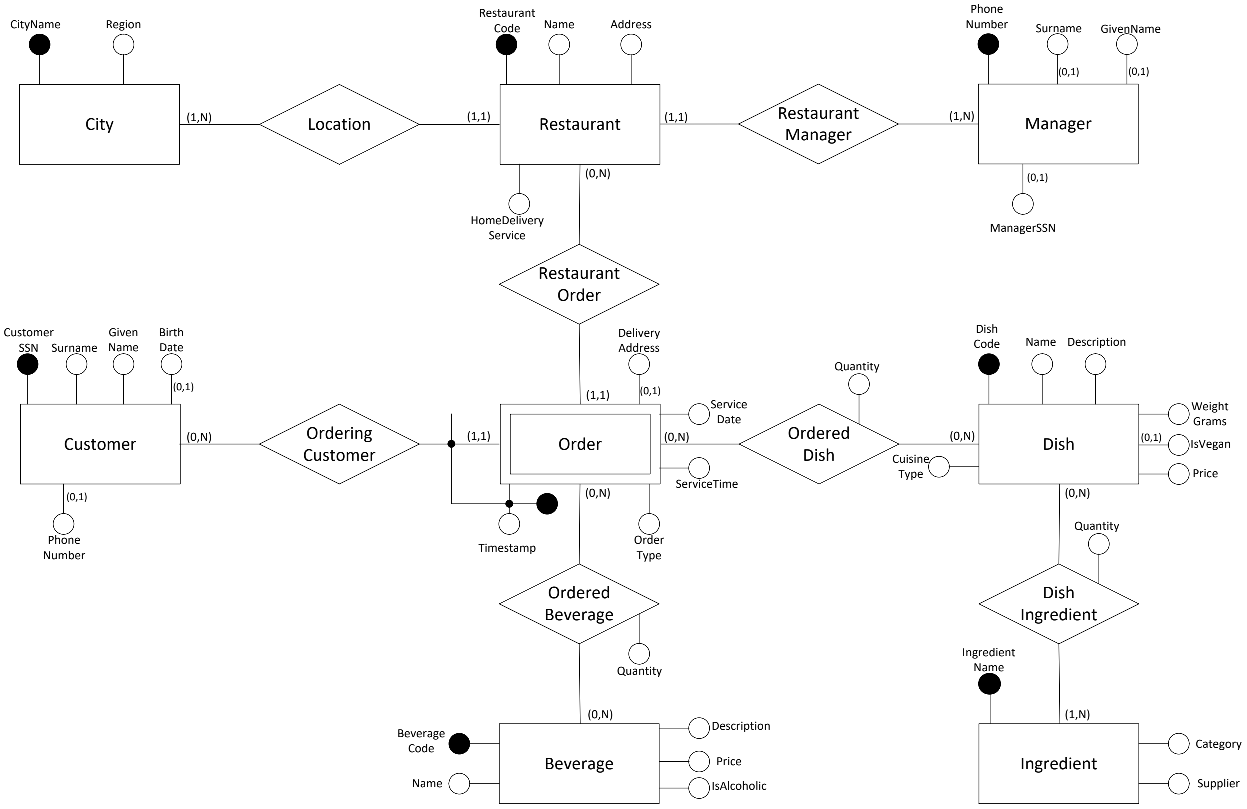


2. Schema integration

2a) Related concept identification + conflict analysis and resolution

PoliRestaurants	UniRestaurants	Conflict	Solution
City	City	Structure conflicts	
		- HomeDeliveryService is an attribute of City → HomeDeliveryService is an attribute of Restaurant	HomeDeliveryService is an attribute of Restaurant
Restaurant	Restaurant	Structure conflicts	
		- Manager is an entity → The manager (phone contact) is an attribute of Restaurant	Manager is an entity
Customer	Customer	Name conflicts	
		- Surname → LastName	Surname
		- GivenName → FirstName	GivenName
		Key conflict	
		- CustomerSSN → CustomerRegistrationNr	CustomerSSN
Dish	Course	Name conflicts	
		- Entity name	Dish
		- DishCode → CourseCode	DishCode
		- Cost → Price	Price
		Data semantics conflicts	
		- WeightInGrams → WeightInKilos	WeightInGrams
Order	Booking	Name conflicts	
		- Entity name	Order
		- OrderType → BookingType	OrderType
		Key conflict	
		- OrderCode →Timestamp + key of the Customer	Timestamp + key of the Customer
		Cardinality conflicts	
		- Each order may be associated with multiple dishes → Each booking is associated with just one course	Each order may be associated with multiple dishes

2b) Global conceptual schema



2c) Conceptual to logical translation

CITY (CityName, Region)
MANAGER (PhoneNumber, ManagerSSN*, Surname*, GivenName*)
RESTAURANT (RestaurantCode, Name, Address, HomeDeliveryService, CityName, ManagerPhoneNumber)
CUSTOMER (CustomerSSN, Surname, GivenName, BirthDate*, PhoneNumber*)
ORDER (CustomerSSN, Timestamp, OrderType, ServiceDate, ServiceTime, DeliveryAddress*, RestaurantCode)
DISH (DishCode, Name, Description, Price, IsVegan*, WeightGrams, CuisineType)
BEVERAGE (BeverageCode, Name, Description, Price, IsAlcoholic)
ORDEREDDISH (CustomerSSN, Timestamp, DishCode, Quantity)
ORDEREDBEVERAGE (CustomerSSN, Timestamp, BeverageCode, Quantity)
INGREDIENT (IngredientName, Category, Supplier)
DISHINGREDIENT (DishCode, IngredientName, Quantity)

3. Query answering and mapping definition

3a) Query formulation

Find timestamp and customer SSN of the take-away orders to restaurants located in Milan or in Turin, which have involved at least one dish with weight greater than 200 grams and price greater than 20 euros.

```
SELECT DISTINCT O.CustomerSSN, O.Timestamp
FROM Order AS O, Restaurant AS R, OrderedDish AS OD, Dish AS D
WHERE O.RestaurantCode=R.RestaurantCode AND O.OrderCode=OD.OrderCode AND
        OD.DishCode=D.DishCode AND O.OrderType='TakeAway'  AND (R.CityName='Milan' OR
        R.CityName='Turin') AND D.WeightGrams>200 AND D.Price>20
```

3b) GAV mapping definition

The KeyGen(_, _) functions generate univocal identifiers.

```
CREATE VIEW UniPoliRestaurants.Restaurant (RestaurantCode, Name, Address, HomeDeliveryService,
CityName, ManagerPhoneNumber) AS (
        SELECT KeyGenRestaurant(R.RestaurantCode, 'PoliRestaurants'), R.Name, R.Address,
                C.HomeDeliveryService, R.CityName, M.ManagerPhoneNumber
        FROM PoliRestaurants.Restaurant AS R, PoliRestaurants.City AS C, PoliRestaurants.Manager AS M
        WHERE R.CityName=C.CityName AND R.ManagerSSN=M.ManagerSSN

        UNION

        SELECT KeyGenRestaurants(RestaurantCode, 'UniRestaurants'), Name, Address,
                HomeDeliveryService, CityName, ManagerPhoneContact
        FROM UniRestaurants.Restaurant
)
```

```
CREATE VIEW UniPoliRestaurants.Dish (DishCode, Name, Description, Price, IsVegan, WeightGrams,
CuisineType) AS (
        SELECT KeyGenDish(DishCode, 'PoliRestaurants'), Name, Description, Cost, IsVegan, WeightGrams,
                'Italian'
        FROM PoliRestaurants.Dish

        UNION

        SELECT KeyGenDish(CourseCode, 'UniRestaurants'), Name, Description, Price, null,
                WeightKilos*1000, CuisineType
        FROM UniRestaurants.Course
)
```

```
CREATE VIEW UniPoliRestaurants.Order (CustomerSSN, Timestamp, OrderType, ServiceDate, ServiceTime,
DeliveryAddress, RestaurantCode) AS (
        SELECT CustomerSSN, Timestamp, OrderType, ServiceDate, ServiceTime, DeliveryAddress,
                KeyGenRestaurant(RestaurantCode, ‘PoliRestaurants’)
        FROM PoliRestaurants.Order

        UNION

        SELECT C.CustomerSSN, B.Timestamp, B.BookingType, B.ServiceDate, B.ServiceTime,
                B.DeliveryAddress, KeyGenRestaurant(B.RestaurantCode, ‘UniRestaurants’)
        FROM UniRestaurants.Booking AS B, UniRestaurants.Customer AS C
        WHERE B.CustomerRegistrationNr=C.CustomerRegistrationNr
)


CREATE VIEW UniPoliRoads.OrderedDish (CustomerSSN, Timestamp, DishCode, Quantity) AS (
        SELECT O.CustomerSSN, O.Timestamp, KeyGenDish(OD.DishCode, ‘PoliRestaurants’), OD.Quantity
        FROM PoliRestaurants.OrderedDish AS OD, PoliRestaurants.Order AS O
        WHERE OD.OrderCode=O.OrderCode

        UNION

        SELECT C.CustomerSSN, B.Timestamp, KeyGenDish(B.CourseCode, ‘UniRestaurants’), B.Quantity
        FROM UniRestaurants.Booking AS B, UniRestaurants.Customer AS C
        WHERE B.CustomerRegistrationNr=C.CustomerRegistrationNr
)
```

3c) Query rewriting

Find timestamp and customer SSN of the take-away orders to restaurants located in Milan or in Turin, which have involved at least one dish with weight greater than 200 grams and price greater than 20 euros.

```
SELECT O.CustomerSSN, O.Timestamp
FROM Order AS O, Restaurant AS R, OrderedDish AS OD, Dish AS D
WHERE O.RestaurantCode=R.RestaurantCode AND O.OrderCode=OD.OrderCode AND
        OD.DishCode=D.DishCode AND O.OrderType=’TakeAway’ AND (R.CityName=’Milan’ OR
        R.CityName=’Turin’) AND D.WeightGrams>200 AND D.Cost>20

UNION

SELECT Cu.CustomerSSN, B.Timestamp
FROM Booking AS B, Customer AS Cu, Restaurant AS R, Course AS Co
WHERE B.CustomerRegistrationNr=Cu.CustomerRegistrationNr AND B.RestaurantCode=R.RestaurantCode
        AND B.CourseCode=Co.CourseCode AND B.BookingType=’TakeAway’ AND (R.CityName=’Milan’ OR
        R.CityName=’Turin’) AND D.WeightKilos>(200/1000) AND D.Price>20
```