# Databases Advanced Retake Exam: Weddings Planner

# Data Model Definition

You are requested to model a database for Bulgarian government that would help keep monitoring marriages around the country. The **database you need to design** has to keep information about:

##### Venue

* **Name**
* **Capacity** – number of people who can celebrate in that venue
* **Town**

##### Agency

* **Name**
* **Employees Count**
* **Town**

##### Person

* **First Name** - must be between **1 and**  **60 symbols** long. Mandatory information
* **Middle Name Initial** – first letter of the middle name of the person. Must be **exactly 1 symbol**. Mandatory information
* **Last Name** – must be **at least** **2 symbols** long. Mandatory information
* **Full Name** – should not be present in the database. Full name is concatenation of first name, middle name initial and last name separated with single space
* **Gender** – can have 3 valid values – **Male**, **Female** or **Not Specified.** Mandatory information
* **Birthdate**
* **Age** – should not be present in the database. It should be calculated only if needed
* **perPhone**
* **Email** – must be in format **{user}@{host}** where:
  + **{user}** - can contain **only alphanumeric characters, ignoring the casing**
  + **{host}** - can contain **only lowercase latin letters** and **one dot**. The dot cannot be at the beginning or at the end of the host
  + **Valid Emails** - john99JJ@abv.bg, john@abv.bg, 9john@abvbg.bg, JJ@abv.com
  + **Invalid Emails** - john.99@abv.bg, john-J@abv.bg, john99@.abv.bg, john99@abvbg., jj<3@a.b.bg.

##### Wedding

* **Bride** – any person. Mandatory information
* **Bridegroom** – any person. Mandatory information
* **Date –** mandatory information
* **Agency –** agency that organizes the wedding

##### Invitation

* **Wedding** – wedding where the guest were invited. Mandatory information
* **Guest** – can be any person beside the bride or bridegroom. Mandatory information
* **Present** – can be eighter cash or gift
* **Attending** – information about whether the **guest accepted the invitation** to attend at the wedding or not
* **Family** – family from which the guest is coming from. Can be only from **Bride’s family** or **Bridegroom’s family.** Mandatory information

##### Cash

* **Owner** – would be person marked as guest in an invitation
* **Cash Amount.** Mandatory information

##### Gift

* **Owner** – would be person marked as guest in an invitation
* **Name –** name of the gift. Mandatory information
* **Size** - size of the gift can be only **Small, Medium** or **Large or Not Specified**

Asume that a **wedding** can be celebrated in **many venues** and each venue can host **many wedding celebrations**. Each **invitation** can contain only **one present** (cash or gift) and a **present** can be only related to **one invitation**. If a **field** is not told to be mandatory that means it **can be null**.

Design a database using the **Entity Framework** and **Code First** approach. You will also need to write several data-driven applications in C# for importing, querying and exporting data from the database.

# Importing Data

You are provided with some **JSON** and **XML** files to **populate the database with some data**. Import all the data from those files into the database. When **importing agencies and venues, no data transfer object is required** (but it is recommended). But for all other imports **use data transfer objects.**

You are **not allowed to modify** the provided **JSON** and **XML** files.

### Importing Data from JSON

#### Import Agencies

Using the file **agencies.json** create a console application that **imports the data from that file into the database** you created in the previous section. After each iteration of importing agency **print information of the imported agency** in the format provided in the example below.

### Example

#### Input

|  |
| --- |
| **agencies.json** |
| [  { "name" : "A Perfect Event Productions", "employeesCount" : 5, "town" : "Sofia" },  { "name" : "A Spice of Life Catering Services", "employeesCount" : 10, "town" : "Sofia" },  { "name" : "Abbey Party Rents", "employeesCount" : 8, "town" : "Plovdiv" },  { "name" : "Allies Party Equipment Rental Inc.", "employeesCount" : 13, "town" : "Varna" },  { "name" : "A-One Weddings & Events", "employeesCount" : 20, "town" : "Burgas" },  { "name" : "Arrangers DMC", "employeesCount" : 2, "town" : "Pleven" },  ...  ] |

#### Output

|  |
| --- |
| Successfully imported A Perfect Event Productions  Successfully imported A Spice of Life Catering Services  Successfully imported Abbey Party Rents  Successfully imported Allies Party Equipment Rental Inc.  Successfully imported A-One Weddings & Events  Successfully imported Arrangers DMC  ... |

#### Import People

Using the file **people.json** create a console application that **imports the data from that file into the database** you created in the previous section. After each iteration of importing person **print information of the imported person** in the format provided in the example below.

This time you need to **validate a record before inserting** it into the database. A record is valid only if **contains first name, last name, middle name initial**. If a gender is wrong or missing at all for given person, set it to Not Specified and consider that record as a valid one if contains all other required fields. If a record is not valid **print an error message and continue with the next record**.

### Example

#### Input

|  |
| --- |
| **people.json** |
| [  {  "firstName": "Amal",  "lastName": "Devivo",  "middleInitial": "A",  "gender": "Male",  "birthday": "2016-12-09T00:00:00",  "phone": "089331533",  "email": "amal@ama.l"  },  {  "firstName": "Willette",  "lastName": "Grana",  "middleInitial": "K",  "gender": "Female",  "birthday": "2016-12-09T10:00:00",  "phone": "0884234634",  "email": "grana@gran.gr"  },  {  "firstName": "Stefan",  "lastName": "Shankles",  "middleInitial": "L",  "gender": "NotSpecified",  "birthday": "2016-12-09T10:00:00",  "phone": "0885312",  "email": "stef6@gmail.com"  },  {  "firstName": "Lindsay",  "lastName": "Suen",  "middleInitial": "J",  "gender": "Female"  },  {  "firstName": "Linnea",  "lastName": "Hasegawa",  "middleInitial": "P",  "birthday": "2016-12-09T10:00:00",  "phone": "08859855551",  "email": "invalid\_email@jj.j."},  {  "firstName": "Juana",  "lastName": "Tulley",  "gender": "Female",  "birthday": "2016-12-09T10:00:00",  "phone": "0885321334"  },  {  "firstName": "Margaret",  "lastName": "Prentice",  "middleInitial": "FF",  "gender": "Female"  },  {  "firstName": "Carry",  "lastName": "Grube",  "middleInitial": "P",  },  ...  ] |

#### Output

|  |
| --- |
| Successfully imported Amal A Devivo  Successfully imported Willette K Grana  Successfully imported Stefan L Shankles  Successfully imported Lindsay J Suen  Error. Invalid data provided  Error. Invalid data provided  Error. Invalid data provided  Successfylly imported Carry P Grube  ... |

#### Import Weddings and Invitations

Using the file **weddings.json** create a console application that **imports the data from that file into the database** you created. After each iteration of importing weddings **print information of the imported wedding** in the format provided in the example below.

Again you need to validate the data before importing it in the database. The **bride** and the **bridegroom would always be valid** if present. **Date** and **agency might be omitted** from the record. In the list of guests, some guest might not be present in the database. In that case ignore that guest and continue with the next one. The RSVP means whether the guest confirmed his/her attendance at the wedding.

A **valid record** must contain at least **bride, bridegroom, agency** and **date**. Guests are optional. If a record is not valid **print an error message and continue with the next record**.

### Example

#### Input

|  |
| --- |
| **weddings.json** |
| [  {  "Bride": "Demetria G Whitney",  "Bridegroom": "Elidia Q Dotts",  "Date": "2016-07-09T00:00:00",  "Agency": "The Manor House",  "Guests": [  {  "Name": "Lynetta W Collie",  "RSVP": false,  "Family": "Bridegroom"  },  {  "Name": "Marylynn T Schueler",  "RSVP": false,  "Family": "Bridegroom"  },  ...]  },  {  "Bride": "Nikia T Sepulveda",  "Date": "2020-02-04T00:00:00",  "Agency": "Monrreal & Company",  },  {  "Bride": "Nancy Q Rodreguez",  "Bridegroom": "Wade V Mcinnis",  "Date": "2020-08-01T00:00:00",  "Agency": "Everything Entertainment"  },  ...  ] |

#### Output

|  |
| --- |
| Successfully imported wedding of Demetria and Elidia  Error. Invalid data provided.  Succesfully imported wedding of Nancy and Wade  ... |

### Importing Data from XML

#### Import Venues

Using the file **venues.xml** create a console application that **imports the data from that file into the database**. After each iteration of importing agency **print information of the imported agency** in the format provided in the example below. For each wedding **add 2 random venues**./ef core bug – not possible for many to many tables/

### Example

#### Input

|  |
| --- |
| **venues.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <venues>  <venue name="Glasshouse">  <capacity>300</capacity>  <town>Sofia</town>  </venue>  <venue name="Christ Community Church">  <capacity>50</capacity>  <town>Plovdiv</town>  </venue>  <venue name="Baku Crystal Hall">  <capacity>600</capacity>  <town>Burgas</town>  </venue>  ...  </venues> |

#### Output

|  |
| --- |
| Successfully imported Glasshouse  Successfully imported Christ Community Church  Successfully imported Baku Crystal Hall  ... |

#### Import Presents

Using the file **presents.xml** create a console application that **imports the data from that file into the database**. After each iteration of importing present **print information of the imported present** in the format provided in the example below.

A **valid recod** for import must contain at least present’s **type** and **invitation-id**. Also if the type of the present is **cash** its **amount must be mentioned**. If the type of the present is **gift** the **name of the gift is a mandatory information** (the **size is optional**, default size is Not Specified) . If any of those elements is missing or it has not valid value (eg. invitation with given Id is not in the database) this present must be ignored and not inserted into the database.

If size attribute is missing size to the present must be set to Not Specified. If the value of the size is wrong that present must be ignored.

If a record is not valid **print an error message and continue with the next record**.

### Example

#### Input

|  |
| --- |
| **presents.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <presents>  <present type="cash" invitation-id="35" amount="9866" />  <present type ="gift" invitation-id="136" present-name="Giant Piano Mat" size="Medium" />  < present invitation-id="309" amount ="1405" />  < present type="cash" invitation-id="62" amount ="29" />  < present invitation-id="192" present-name="Learn to Play The Acoustic Guitar" size="NotSpecified" />  < present type="cash" invitation-id="131" />  < present type="gift" invitation-id="-10" present-name="Cloud B Turtle Night Light" size="NotSpecified" />  < present type="cash" invitation-id="320" amount="2539" />  < present type="cash" invitation-id="19" amount="6831" />  < present type="cash" invitation-id="86" amount="709" />  < present type="cash" invitation-id="63" amount="7144" />  < present type="gift" invitation-id="233" present-name="Leather Journal" size="XL" />  < present type="cash" invitation-id="74" amount ="5178" /> ...  </presents> |

#### Output

|  |
| --- |
| Succesfully imported gift from Yevette F Phinney  Succesfully imported gift from Carmelo Q Morace  Error. Invalid data provided  Succesfully imported gift from Hermelinda X Creel  Error. Invalid data provided  Error. Invalid data provided  Error. Invalid data provided  Succesfully imported gift from Luanne W Messerly  Succesfully imported gift from Bettie P Arrowsmith  Succesfully imported gift from Ada W Basham  Succesfully imported gift from Shayne X Rester  Error. Invalid data provided  Succesfully imported gift from Albertina D Jeter  ... |

# Exporting Data

Now let’s provide some statistics to the government. It need those statistics **exported in JSON** and **XML format.**

### Exporting to JSON

#### Ordered Agencies

Write a program that creates file **agencies-ordered.json** that would contain **all agencyies’ name, employees count** and **town**. Order them **by employees count descending** and then **by name alphabetically**.

### Example

#### Output

|  |
| --- |
| **agencies-ordered.json** |
| [  {  "name": "Exceptional Events",  "count": 30,  "town": "Montana"  },  {  "name": "Ice Occasions",  "count": 24,  "town": "Vratsa"  },  {  "name": "Hannibals Catering & Events",  "count": 23,  "town": "Montana"  },  ...  ] |

#### Guest Lists

Write a program that creates file **guests.json** that would contain:

* bride and bridesgrooms full names
* agency that organizes the wedding (its name and town)
* count of **all invited guests**
* count of **guest from bride’s family**
* count of **guest from bridegroom’s family**
* count of **guest that confirmed they would attend at the wedding**
* the **list of full names** of guests who would attend at the wedding

Order the list of weddings by all invited guests descending and then by attending guests count ascending.

### Example

#### Output

|  |
| --- |
| **guests.json** |
| [  {  "bride": "Eleanore I Windley",  "bridegroom": "Hong G Cusack",  "agency": {  "name": "Splash Cafe",  "town": "Plovdiv"  },  "invitedGuests": 42,  "brideGuests": 16,  "bridegroomGuests": 26,  "attendingGuests": 30,  "guests": [  "Portia P Wile",  "Emerson Y Rauscher",  "Elidia X Schrecengost",  ...  ]  },  ...  ] |

### Exporting to XML

#### Venues in Sofia

Write a program that creates file **sofia-venues.xml** that would include information about **all venues located in Sofia** that have **at least 3 weddings** celebrated in them, ordered **by capacity ascending**. Use the format provided below to **structure your XML file**.

### Format

<venues town=”Sofia”>

<venue name=”{Venue Name}” capacity=”{Capacity}”>

<weddings-count>{Number of Weddings}</weddings-count>

</venue>

...

</venues>

### Example

#### Output

|  |
| --- |
| **sofia-venues.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <venues town="Sofia">  <venue name="Sand Spring Restaurant" capacity="500">  <weddings-count>4</weddings-count>  </venue>  <venue name="St John the Divine" capacity="4100">  <weddings-count>4</weddings-count>  </venue>  <venue name="St John the Baptist" capacity="4100">  <weddings-count>3</weddings-count>  </venue>  </venues> |

#### Agencies by Town

Write a program that creates file **agencies-by-town.xml** that would include information about **all agencies grouped by town and the weddings they organized.** Use the format provided below to **structure your XML file**.

Include only **towns whose name is at least 6 symbols long**. Include only **agencies** which **organized at least 2 weddings**. For each wedding include **bride** **and bridegroom’s full name** and the **list of all invited guests**. For each **guest include the family** which he/she is coming from (Bride or Bridegroom). For the wedding **calculate the total amount given cash** as presents and also the **total count of all small, medium and large gifts**.

Also, include the profit for each agency. **Profit is 20% of the total amount of cash for each wedding the agency has organized**. For example: Agency X organized 3 weddings an each received as cash gifts respectively 1000, 2000 and 3000. So, the profit for agency X would be (1000 + 2000 + 3000) x 20% = 1200.

The **order** of towns, agencies, weddings and guests **does not matter**.

### Format

<towns>

<town name={Town Name}>

<agencies>

<agency name={Agency Name} profit={Agency Profit}>

<wedding cash={Count} presents={Count}>

<bride>{Full Name}</bride>

<bridegroom>{Full Name}</bridegroom>

<guests>

<guest family={Bride or Bridegroom}>{Full Name}</guest>

...

</guests>

</wedding>

...

</agency>

...

</agencies>

</town>

...

</towns>

### Example

#### Output

|  |
| --- |
| agencies-by-town**.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <towns>  <town name="Plovdiv">  <agencies>  <agency name="Forevermore Events" profit="46671.200">  <wedding cash="33280.00" presents="3">  <bride>Lashunda Q Montero</bride>  <bridegroom>Denese D Bartel</bridegroom>  <guests>  <guest family="Bridegroom">Isobel V Kreamer</guest>  <guest family="Bridegroom">Bess D Zheng</guest>  <guest family="Bridegroom">Candie H Withrow</guest>  <guest family="Bride">Cleopatra J Cull</guest>  ...  </guests>  </wedding>  <wedding cash="41834.00" presents="8">  ...  </wedding>  <wedding cash="22548.00" presents="7">  ...  </wedding>  <wedding cash="135694.00" presents="12">  ...  </wedding>  </agency>  <agency name="Brown Brothers Catering" profit="39855.200">  ...  </agency>  ...  </agencies>  </town>  ...  </towns> |