**Planning for Distribution - Beta University Annual Fund**

**Step 10.1 – Write out the normalized set of relations developed in Chapter 6 as the global schema.**

**Answer:**

**PotentialDonor**(donorId, firstName, lastName, *street*, *zip*, country, countryCode, *telNumber*, donorCircle, spouseName, amountDonatedLastYear, amountPledgedThisYear, amountPaidThisYear, *volunteerFirstName*, *volunteerLastName*)

**Zip**(zip, city, state)

**Phone**(areaCode, telNumber, street, *zip*)

**MatchingCorp**(matchCorpName, *zip*, country, countryCode, *telNumber*)

**Event**(eventName, eventDate, eventTime, eventLocation, eventOrganizer, eventTotalPledged eventTotalPaid)

**ClassRep**(graduationYear, firstName, lastName, *zip*, country,  countryCode, *telNumber*)

**Volunteer**(firstName, lastName, *zip*, country, countryCode, *telNumber*)

**Pledge**(pledgeNumber,  pledgeAmount, pledgeDate, numPaymentsChosen, pledgeAmountPaid, numPaymentsMade, *donorId, eventName*)

**Payment**(*pledgeNumber*, datePaid, amountPaid, paymentMethod, creditCardType, creditCardNumber, *matchCorpName*)

**PotentialDonor-Event**(*donorId, eventName*)

**Pledge-MatchingCorp**(*pledgeNumber, matchCorpName,*spouse)

**Donor-GradYear**(*donorId*, graduationYear)

**Donor-Categories**(*donorId*, donorCategory)

**Step 10.2 - Write out a set of end user locations and the applications performed at each. Provide a reason to justify why you chose this data distribution plan.**

**Answer:**

The three locations are Midtown (the main origin), Uptown, and Downtown.

**Applications for each branch’s own data:**

1. **Maintaining Donor Records:** Keeping track of donor personal details, donation history, and pledge amounts.
2. **Managing Events:** Keeping track of event details, such as the event name, date, location, and pledged amounts.
3. **Maintaining Volunteer Records:** Tracking volunteer personal details, the amount of time volunteered, and event participation.
4. **Managing Pledges:** Tracking donor pledges, payment details, and whether they are matched by corporate donors.
5. **Managing Payments:** Managing the details of payments made toward pledges, including payment methods.
6. **Tracking Donor Categories:** Classifying donors into specific categories based on amount donated or other criteria.

In addition, the following applications are performed at Midtown only:

1. **Generating Reports on Donors**: Creating reports on donors based on categories, donation amounts, and pledge statuses.
2. **Generating Event Summaries**: Summarizing event totals for pledges, payments made, and matched donations.
3. **Generating Volunteer Reports**: To tracking volunteer participation and correlating volunteer work with pledges or donations.
4. **Payment Tracking**: Reporting on payment statuses and pledges across multiple locations.
5. **Corporate Matching Reports**: Reports showing matched pledges by corporate donors.

**Step 10.3 - For each application, determine and write out the required tables.**

**Answer:**

|  |  |
| --- | --- |
| Maintaining Donor Records: | PotentialDonor  Zip  Phone  MatchingCorp |
| Managing Events: | Event  PotentialDonor-Event  Pledge |
| Maintaining Volunteer Records: | Volunteer  PotentialDonor  Zip |
| Managing Pledges: | Pledge  PotentialDonor  MatchingCorp  Pledge-MatchingCorp |
| Managing Payments: | Payment  Pledge  MatchingCorp  Donor-GradYear |
| Tracking Donor Categories: | Donor-Categories  PotentialDonor |
| Generating Reports on Donors: | PotentialDonor  Donor-Categories  Pledge  Payment  MatchingCorp |
| Generating Event Summaries: | Event  Pledge  Payment  MatchingCorp |
| Generating Volunteer Reports: | Volunteer  PotentialDonor  Event |
| Payment Tracking: | Payment  Pledge  MatchingCorp |
| Corporate Matching Reports: | MatchingCorp  Pledge-MatchingCorp  PotentialDonor |

**Step 10.4 - Using the normalized relations, perform selection and projection operations, to create the set of vertical, horizontal and mixed data fragments needed for each application.**

**Answer:**

**PotentialDonor:** Here, each branch may only need a subset of donor information.

* PotentialDonorFragment1 = πdonorId, firstName, lastName(PotentialDonor)  
  Here, Basic donor identification for use across applications
* PotentialDonorFragment2 = πdonorId, firstName, lastName, street, zip, country, telNumber(PotentialDonor)  
  Full donor information, including address and contact details, for specific donor records
* Each branch could use specific ID prefixes for donor identification:
  + PotentialDonorDowntown = σdonorId LIKE 'D%'(PotentialDonor)
  + PotentialDonorMidtown = σdonorId LIKE 'M%'(PotentialDonor)
  + PotentialDonorUptown = σdonorId LIKE 'U%'(PotentialDonor)

**Zip:** Since this table is rarely updated and needed at all locations, replicate the entire table across branches.

**Phone**: Here, select only records needed for each branch based on the area code.

* PhoneDowntown = σareaCode='D'(Phone)
* PhoneMidtown = σareaCode='M'(Phone)
* PhoneUptown = σareaCode='U'(Phone)

**MatchingCorp**: Fragment this based on matching corporate sponsors for each branch.

* MatchingCorpDowntown = σzip='D'(MatchingCorp)
* MatchingCorpMidtown = σzip='M'(MatchingCorp)
* MatchingCorpUptown = σzip='U'(MatchingCorp)

**Event**: Branch-specific fragments based on location or organizer.

* EventDowntown = σeventLocation='Downtown'(Event)
* EventMidtown = σeventLocation='Midtown'(Event)
* EventUptown = σeventLocation='Uptown'(Event)

**PotentialDonor-Event**: Each branch only needs event participation records for its location.

* PotentialDonorEventDowntown = σdonorId LIKE 'D%'(PotentialDonor-Event)
* PotentialDonorEventMidtown = σdonorId LIKE 'M%'(PotentialDonor-Event)
* PotentialDonorEventUptown = σdonorId LIKE 'U%'(PotentialDonor-Event)

**Pledge**: Filter by branch based on donor association.

* PledgeDowntown = σdonorId LIKE 'D%'(Pledge)
* PledgeMidtown = σdonorId LIKE 'M%'(Pledge)
* PledgeUptown = σdonorId LIKE 'U%'(Pledge)

**Volunteer**: Create fragments by location to manage volunteer records.

* VolunteerDowntown = σzip='D'(Volunteer)
* VolunteerMidtown = σzip='M'(Volunteer)
* VolunteerUptown = σzip='U'(Volunteer)

**Payment**: Separate fragments based on branch using pledge information.

* PaymentDowntown = σpledgeNumber > 0 AND pledgeNumber < 20000(Payment)
* PaymentMidtown = σpledgeNumber > 20000 AND pledgeNumber < 40000(Payment)
* PaymentUptown = σpledgeNumber > 40000 AND pledgeNumber < 60000(Payment)

**Donor-Categories**: Fragment based on donor records in each branch.

* DonorCategoriesDowntown = σdonorId LIKE 'D%'(Donor-Categories)
* DonorCategoriesMidtown = σdonorId LIKE 'M%'(Donor-Categories)
* DonorCategoriesUptown = σdonorId LIKE 'U%'(Donor-Categories)

**Step 10.5 - Map the fragments to the applications and locations.**

For each fragment that is required at more than one application location, decide whether the fragment can be replicated, by considering frequency of use and of update.

**Answer:**

**PotentialDonor**:

* The PotentialDonorFragment1 containing donor ID, first name, and last name is used at multiple branches and updated infrequently. We will replicate this fragment at all branches to support frequent use.
* PotentialDonorFragment2, containing sensitive information such as address and contact details, will be stored only at Midtown, where other branches can access it via a view for applications that require donor details with sensitive information, such as for specific donor records or reports.

**Zip**:

* This table is required at all locations, is rarely updated, and does not contain sensitive information, so it will be fully replicated at each branch.

**Phone**:

* Each branch will have its own fragment for phone records based on area codes (e.g., PhoneDowntown, PhoneMidtown, PhoneUptown) since this data is used only for branch-specific applications and rarely updated.

**MatchingCorp**:

* Since branches require only records for their local corporations, we will use fragments based on location like MatchingCorpDowntown, MatchingCorpMidtown, MatchingCorpUptown, rather than replicating the entire table.

**Event**:

* Each branch will keep a fragment of its own events that is, EventDowntown, EventMidtown, EventUptown. This is frequently used within a branch and updated as new events are created.

**PotentialDonor-Event**:

* The table is fragmented by branch, with each branch storing only records of donor-event associations relevant to its location which are PotentialDonorEventDowntown, PotentialDonorEventMidtown, PotentialDonorEventUptown.

**Pledge**:

* Each branch will keep a fragment of pledges made by local donors. These fragments are frequently accessed and updated within the branch e.g., PledgeDowntown, PledgeMidtown, PledgeUptown, allowing branches to work independently with minimal cross-branch dependencies.

**Volunteer**:

* Each branch will maintain its own fragment for volunteer records which are VolunteerDowntown, VolunteerMidtown, VolunteerUptown, based on the branch’s zip code, as this data is frequently accessed and updated locally. here replication across branches is unnecessary.

**PotentialDonor** and **Zip**:

* These fragments are the same as in “Maintaining Donor Records.”

**Pledge (For Managing Pledges)**:

* Each branch will have its own fragment which are PledgeDowntown, PledgeMidtown, PledgeUptown based on donor association, as this data is frequently used and updated within each branch.

**PotentialDonor** **and** **MatchingCorp**:

* We will use fragments for each branch, as in the “Maintaining Donor Records” section, based on the donor’s location.

**Payment**:

* Each branch will maintain its own fragment of payment records, identified by pledge association (PaymentDowntown, PaymentMidtown, PaymentUptown). This allows each branch to keep independent records of payments for local pledges without cross-references.

**Pledge** **and** **MatchingCorp**:

* These will use the same fragments as in the “Managing Pledges” section, with branch-specific records.

**Donor-Categories (For Tracking Donor Categories)**:

* Each branch will store a fragment containing records of local donors' categories which are DonorCategoriesDowntown, DonorCategoriesMidtown, DonorCategoriesUptown, allowing them to generate category-specific reports independently.

**PotentialDonor**, **Donor-Categories**, **Pledge**, **Payment**, **MatchingCorp (For Generating Reports on Donors)**:

* These tables use the previously defined fragments, ensuring each branch has access to the necessary data for donor reporting while minimizing data redundancy.

**Event**, **Pledge**, **Payment**, **MatchingCorp (For Generating Event Summaries)**:

* Each branch will use its own fragments as defined in “Managing Events” and “Managing Payments” to generate event summaries, using only local data for event-specific reports.

**Volunteer**, **PotentialDonor**, **Event**, **Payment (For Generating Volunteer Reports)**:

* Each branch will use previously defined fragments to generate volunteer reports, ensuring local data access for volunteer-related applications.

**Payment**, **Pledge**, **MatchingCorp (For Payment Tracking)**:

* Payment tracking at each branch will use the branch-specific fragments defined under “Managing Payments” to ensure each branch has access to the necessary data for local payment tracking.

**MatchingCorp**, **Pledge-MatchingCorp**, **PotentialDonor (For Corporate Matching Reports)**:

* These tables use fragments specific to each branch, with MatchingCorp and Pledge-MatchingCorp containing only local records.

**Step 10.6 – Create a table (within Microsoft Word) showing a geographical network, listing the locations and applications. Write out the data fragments for each location.**

**Answer:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Application** | **Downtown** | **Midtown** | **Uptown** |
| Maintaining Donor Records | PotentialDonorFragment1,  Zip,  PhoneDowntown | PotentialDonorFragment2,  Zip,  PhoneMidtown | PotentialDonorFragment1,  Zip,  PhoneUptown |
| Managing Events | EventDowntown, PotentialDonorEventDowntown, PledgeDowntown | EventMidtown, PotentialDonorEventMidtown, PledgeMidtown | EventUptown, PotentialDonorEventUptown, PledgeUptown |
| Maintaining Volunteer Records | VolunteerDowntown, PotentialDonorFragment1,  Zip | VolunteerMidtown, PotentialDonorFragment2,  Zip | VolunteerUptown, PotentialDonorFragment1, Zip |
| Managing Pledges | PledgeDowntown, PotentialDonorFragment1, MatchingCorpDowntown, Pledge-MatchingCorp | PledgeMidtown, PotentialDonorFragment2, MatchingCorpMidtown, Pledge-MatchingCorp | PledgeUptown, PotentialDonorFragment1, MatchingCorpUptown, Pledge-MatchingCorp |
| Managing Payments | PaymentDowntown, PledgeDowntown, MatchingCorpDowntown, Donor-GradYear | PaymentMidtown, PledgeMidtown, MatchingCorpMidtown, Donor-GradYear | PaymentUptown, PledgeUptown, MatchingCorpUptown, Donor-GradYear |
| Tracking Donor Categories | DonorCategoriesDowntown, PotentialDonorFragment1 | DonorCategoriesMidtown, PotentialDonorFragment2 | DonorCategoriesUptown, PotentialDonorFragment1 |
| Generating Reports on Donors | PotentialDonorFragment1, DonorCategoriesDowntown, PledgeDowntown, PaymentDowntown, MatchingCorpDowntown | PotentialDonorFragment2, DonorCategoriesMidtown, PledgeMidtown, PaymentMidtown, MatchingCorpMidtown | PotentialDonorFragment1, DonorCategoriesUptown, PledgeUptown, PaymentUptown, MatchingCorpUptown |
| Generating Event Summaries | EventDowntown, PledgeDowntown, PaymentDowntown, MatchingCorpDowntown | EventMidtown, PledgeMidtown, PaymentMidtown, MatchingCorpMidtown | EventUptown, PledgeUptown, PaymentUptown, MatchingCorpUptown |
| Generating Volunteer Reports | VolunteerDowntown, PotentialDonorFragment1, EventDowntown, PaymentDowntown | VolunteerMidtown, PotentialDonorFragment2, EventMidtown, PaymentMidtown | VolunteerUptown, PotentialDonorFragment1, EventUptown, PaymentUptown |
| Payment Tracking | PaymentDowntown, PledgeDowntown, MatchingCorpDowntown | PaymentMidtown, PledgeMidtown, MatchingCorpMidtown | PaymentUptown, PledgeUptown, MatchingCorpUptown |
| Corporate Matching Reports | MatchingCorpDowntown, Pledge-MatchingCorp, PotentialDonorFragment1 | MatchingCorpMidtown, Pledge-MatchingCorp, PotentialDonorFragment2 | MatchingCorpUptown, Pledge-MatchingCorp, PotentialDonorFragment1 |

**Step 10.7 – Create a table (within Microsoft Word) showing a geographical network, listing the locations and applications. Determine and write out whether access will be local, remote, or compound.**

Make up a table showing each location and the applications requiring local access, remote access, and compound access.

**Answer:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Application** | **Downtown** | **Midtown** | **Uptown** |
| Maintaining Donor Records | remote | local | remote |
| Managing Events | local | local | local |
| Maintaining Volunteer Records | remote | local | remote |
| Managing Pledges | remote | local | remote |
| Managing Payments | remote | local | remote |
| Tracking Donor Categories | remote | local | remote |
| Generating Reports on Donors | Compound | local | Compound |
| Generating Event Summaries | Compound | local | Compound |
| Generating Volunteer Reports | Compound | local | Compound |
| Payment Tracking | Compound | local | Compound |
| Corporate Matching Reports | Compound | local | Compound |

**Step l0.8 - For each of the non-local accesses, identify the application and the location of the data. Provide a reason to justify your choice of non-local storage.**

Estimate the number of accesses required per day using estimates such as low, medium, or high. Provide a reason to justify your choice of non-local storage.

**Answer:**

Since the **Downtown** and **Uptown** can have access from **Midtown**, **Downtown** and **Uptown** must be allowed to access the records of **donors**. **Midtown** is chosen for **data storage** since this would ensure that there is **consistency** and **privacy**. The access in both **Downtown** and **Uptown** should be **remote** because the data will be kept **centralized** in **Midtown** for all locations to reference the **latest information**, especially regarding **sensitive donor information**. Access to this is **medium** in frequency because it continuously updates **donors' information**; however, this also doesn't require **constant access**.  
The **management of events** is limited to the **local** levels in each location. Each branch needs access to the details of events right away to **plan** and **manage** the events taking place at each site locally. The **local access** would suffice since it is not needed for event data to be **shared** among or accessed **remotely** by branches. The frequency of access is **high** because **event planning** is a continuous process.  
The **central data** for maintaining **volunteer records** is kept in **Midtown**, and **Downtown** and **Uptown** would have access to it **remotely**. This ensures that each branch has **identical records** of its volunteers, with the majority of the editing being done from **Midtown**. For volunteer data, which hardly changes frequently compared to event details or donor records, the frequency of access is **medium**, which would occur on occasions when updates are necessary or when new volunteers are being registered.  
The access model to **Managing Pledges** is similar in that **pledge data** rests in **Midtown**, while **Downtown** and **Uptown** need to access it **remotely**. Since the **pledges** are tracked centrally, the remote branches must be given **remote access** to view the pledge information for their respective locations; however, **Midtown** retains **local access** to the most recent records. Access is also **medium** in frequency because pledges are updated on a **regular schedule** but not updated excessively.  
The function of managing the **payment system** is also **centralized** in **Midtown**, and all other branches such as **Downtown** and **Uptown** access these **payment records**. It shall be received in **Midtown** for proper **financial accuracy**, but the branches must have **remote access** to review these **payment statuses** pertaining to their locations. This **remote access** has a **medium frequency**, with **periodic tracking** of payments.  
In **Donor Categories Tracking**, the **donor categories** are maintained only at **Midtown**, but **Downtown** and **Uptown** access them **remotely**. In this respect, it ensures that the branches are consistently classifying donors in a **similar way** in their **reporting** and **tracking**. The frequency is **low** since donor categories are normally updated **infrequently**.  
**Donor Reports Creation** involves **compound access**. Other examples include **Event Summaries Generation**, **Volunteer Reports Generation**, and **Payment Tracking**. **Midtown** is the repository for **donor**, **event**, **volunteer**, and **payment information**. **Downtown** and **Uptown** must also **remotely access** their respective information. The **compound access** is necessary because such reports draw from a **variety of locations** for the reports to make sense of the **operations** in each of the branches. These processes occur at **medium frequency** depending on the **reporting cycle**, whether **monthly**, **quarterly**, etc.  
Another similar type of **composite access model** is that of **Corporate Matching Reports**. **Midtown** maintains the matching donation data, but **Downtown** and **Uptown** have a need to access information regarding their own **corporate matching donations**. The data is **less frequently updated**, thus the access frequency will be **low**, as reports are typically generated on a **periodic basis** based on the **matching donation cycles**.  
**Summary**: Overall, most applications would adapt to a **remote** or **compound access model**, as the needs are to **centralize data** in **Midtown** for **consistency**, yet still allow **Downtown** and **Uptown** branches to access **location-specific data**. The frequency of access will range from **high**, such as event management, to **low**, such as in corporate matching, reflecting different operational imperatives in the branches.

**Step 10.9 - Create a table (within Microsoft Word) showing a final geographical network, listing any adjustments made by your analysis of applications and traffic.**

If there are no changes, relist the geographical network table from Step 10.7.

|  |  |  |  |
| --- | --- | --- | --- |
| **Application** | **Downtown** | **Midtown** | **Uptown** |
| Maintaining Donor Records | remote | local | remote |
| Managing Events | local | local | local |
| Maintaining Volunteer Records | remote | local | remote |
| Managing Pledges | remote | local | remote |
| Managing Payments | remote | local | remote |
| Tracking Donor Categories | remote | local | remote |
| Generating Reports on Donors | Compound | local | Compound |
| Generating Event Summaries | Compound | local | Compound |
| Generating Volunteer Reports | Compound | local | Compound |
| Payment Tracking | Compound | local | Compound |
| Corporate Matching Reports | Compound | local | Compound |