**Student Projects Chapter 10 - Planning for Distribution**

Read the sample project steps for this chapter and apply the same techniques to the student project that you are developing. Use the normalized set of relations developed in Chapter 6 as the global schema. The schema is:

**Member**(memberID, address, areaCode, email, firstName, lastName, phoneNumber, *zip*)

**Play**(playID, author, numberOfActs, title, type, numberOfSets)

**Sponsor** (businessName, income)

**Subscriber** (subscriberID, address, areaCode, email, firstName, lastName, pastPlays, phoneNumber, *zip*)

**Production** (playID*,* cost, yearOfTheProduction, seasonStartDate, seasonEndDate)

**Performance** (playID, date, time, year)

**TicketSale** (*subscriberID*, *playID,* ticketIncome)

**DuesPayment (**memberID*,* duesPaid, dateofDuesSubmission, duesAmount)

**Donation** ( businessName*,* donationNumber)

**Ticket﻿﻿﻿﻿﻿﻿﻿** (*qrCode*, playID subscriberID, playDate, *playNames*, playTime, price, seat, seatNumbers)

**ZipCode**(zip,city,state)

For the project you have chosen, assume that there are at least four locations or branches for the enterprise and that the processing is to be distributed to these locations. Identify the applications that will be performed at each of the locations, and then follow the steps below to plan the distribution of your database.

* Step 10.1 - Write out a set of end user locations and the applications performed at each.

Our locations will include Pleasantville (the main branch) along with offices spread throughout Westchester and NYC in Yorktown Heights, White Plains, and the Financial District.

1. Maintain tickets number records
2. Produce ticket sales receipt
3. Produce the Tickets sold per Performance
4. Maintain sponsor records
5. Maintain Income records
6. Maintain expenditures records
7. Producing active Subscribers report
8. Producing active Members report
9. Maintain ticket sales records
10. Producing the performance report
11. Producing the subscribers Sales Report

In addition, the following applications will only be performed at Pleasantville

1. Maintain subscriber list
2. Maintain member list
3. Produce DuesPayment report
4. Maintain Play list

* Step 10.2 - For each application, decide what tables are required.

1. Maintain tickets number records - Tickets
2. Produce ticket sales receipt - Tickets, TicketsSale, Subscriber
3. Produce the Tickets sold per Performance - Tickets, TicketsSale, Performance
4. Maintain sponsor records - Sponsors, Donations
5. Maintain Income records - DuesPayment, Donations, TicketSales
6. Maintain expenditures records - Production
7. Producing active Subscribers report - Subscribers
8. Producing active Members report - Members, ZipCode
9. Maintain ticket sales records - TicketSales, Subscribers
10. Producing the performance report - Susbscibers, TicketSales, Members, Sponsors, Performance
11. Producing the subscribers Sales Report - Subscribers, TicketSales, Tickets
12. Maintain subscriber list - Subscribers, ZipCode
13. Maintain member list - Members, ZipCode
14. Produce DuesPayment report - DuesPayment, Member
15. Maintain Play list - Play

* Step 10.3 - Using the normalized relations, perform selection and projection operations, to create the set of vertical, horizontal and mixed data fragments needed for each application.

**Member**: Parts of the table are used in all the sites for some of the applications(8, 10). Only the memberId is needed for most of these, but the names and phonenumber are needed for a few. The required data could be produced by projecting the Member table onto the required columns, creating a fragment we will call MemberFragment1.

MemberFragment1 = ΠmembererId, firstName, lastName(Member).

At Pleasantville only, the theater also requires the member’s email, address and phone number if the members didn’t pay the due in time to the theater. We form another fragment for this projection.

MemberFragment2 = ΠmembererId, firstName, lastName, address, email, phoneNumber(Member).

**Sponsor:** Each branch will have its own sponsor records (application4).

SponsorPleasantville = 𝞼ID LIKE ‘P’(Sponsor)

SponsorYorktownHeights = 𝞼ID LIKE ‘Y’(Sponsor)

SponsorWhitePlains = 𝞼ID LIKE ‘W’(Sponsor)

SponsorFinancialDistrict = 𝞼ID LIKE ‘F’(Sponsor)

**Subscriber:** parts of the Subscriber table will be used at all the different locations (for applications 2, 7, 9, 11, 12). The following fragment will be used for applications 2, 9, and 11, which all offices will be able to access for generic reports and recording ticket sales.

SubscriberFragment1 = ΠsubscriberId, firstName, lastName, email, phoneNumber(Subscriber).

For application 7, we will also need the subscribers past plays to see how active they are in the program. A past play that was just performed would show a more active subscriber. For this, we create another fragment that we will refer to as SubscriberFragment2.

SubscriberFragment2 = ΠsubscriberId, firstName, lastName, pastPlays(Subscriber).

Since the Subscriber table contains sensitive information, address, and only the main office should be able to view and update, we will create the following fragment which includes the address attribute.

SubscriberFragment3 = ΠsubscriberId, firstName, lastName, email, address, phoneNumber(Subscriber).

**ZipCode:** The table will not be updated unless a member or subscriber has an address change and is needed at every location, so we will replicate the entire table at each different branch.

**Play:** While this table will only be updated at Pleasantville (application 15) all offices will need access to it. This table will be updated somewhat infrequently (only when new plays are added) and every office will need access to it so we will replicate this table at each office.

**Production:** Each office uses this table for maintaining expenses related to a specific production (6). Expenses will be recorded by office so we will create an attribute, office, to the table to identify the office that made the purchase/booked the expense related to a production.

ProductionPleasantville = 𝞼office=’Pleasantville’(Production)

ProductionYorktownHeights = 𝞼office=’YorktownHeights’(Production)

ProductionWhitePlains = 𝞼office=’WhitePlains’(Production)

ProductionFinancialDistrict = 𝞼office=’FinancialDistrict’(Production)

**Performance:** Each office will use this table for the tickets sold per performance report (3) and the performance report (10). We can create the following fragment for application by using projection.

PerformanceFragment1 = ΠplayID, date(Performance).

We can also create the following fragment for application 10, which contains all information from the performance table and provides for a more detailed report.

PerformanceFragment2 = ΠplayID, date, time, year(Performance).

**TicketSale:** Each office will be able to use this table to produce ticket sales receipts (2), produce tickets sold per performance reports (3), maintain income records (5), produce performance reports (10), and produce subscribers sales reports (11).

For each ticket sale, a ticket sales receipt is produced for the subscriber so we create the following fragment for application 2. We can also use the fragment below for application 11 which is the internally generated subscribers sales report, and application 3 which is tickets sold by performance.

TicketSaleFragment1 = ΠsubscriberID, playID, ticketIncome(TicketSale).

Each branch will need to maintain their own ticket sales so we will use the following fragments for application 5. This will also help us keep track of ticket sale volume by office.

TicketSalePleasantville = 𝞼office=’Pleasantville’(TicketSale)

TicketSaleYorktownHeights = 𝞼office=’YorktownHeights’(TicketSale)

TicketSaleWhitePlains = 𝞼office=’WhitePlains’(TicketSale)

TicketSaleFinancialDistrict = 𝞼office=’FinancialDistrict’(TicketSale)

**DuesPayment:** Each office will use this table to accept dues from members, which will be maintained in the income records (5) and only the Pleasantville office will use this table to produce the DuesPayment report (14).

We can use the following fragment for both applications.

DuesPaymentFragment1 = ΠmemberID, duesPaid, dateofDuesSubmission, duesAmount(DuesPayment).

**Donation:** The donation table will be used by each office to maintain sponsor records (4) and maintain income records (5).

We can use the following projection for applications 4 and 5.

DonationFragment1 = ΠbusinessName, donationNumber(Donation).

**Ticket:** We will need a few different fragments for the Ticket table and each office will need access to different portions of the Ticket table depending on the application. For application 1, maintaining a record of ticket numbers, we need to view the qrCode, seat, and seatNumber, so we will use the following fragment.

TicketFragment1 = ΠqrCode, seat, seatNumber(Ticket). For application 2 and 11, producing a ticket sales receipt, and producing a subscribers sales report, we need the following fragment.

TicketFragment2 = ΠqrCode, subscriberID, playDate, playName, playTime, price(Ticket)

For application 3, producing the tickets sold per performance we will need to take the following steps. First we modify the Ticket table and add a primary/foreign key, playID, which serves as a unique identifier for each performance. We will also need the following fragment.

TicketFragment3 = ΠqrCode, playID, subscriberID(Ticket)

We can then use TicketFragment3 and the following SQL query.

SELECT \*

FROM Ticket

WHERE subscriberID IS NOT NULL;

* Step 10.4 - 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.

**Member:**

The Member table will be updated infrequently. Pleasantville (main office) will need access to the entire table so that all member information can be maintained, so we will store the entire table (MemberFragment2) at Pleasantville. All other offices will need only the memberID, firstName, and lastName attributes of the table, so we will replicate MemberFragment1 at the other branches.

**Sponsor:**

Each office will maintain its own list of sponsors, using fragments, SponsorPleasantville, SponsorYorktownHeights, SponsorWhitePlains, SponsorFinancialDistrict.

**Subscriber:**

The Subscriber table will only be updated when there is a new subscription, changes are made to an existing subscriber (last name, phone number, address), and when they attend a new play (past plays will be updated). Each office will require access to portions of the Subscriber table so we create SubscriberFragment1 which will be kept at all offices for viewing reports and maintaining records of ticket sales. The active subscribers report will require the past plays attribute from the Subscriber table so we create a second fragment, SubscriberFragment2. We created SubscriberFragment3, which contains all Subscriber attributes including address. Only the main office should be able to view/update/create a subscriber’s address so this fragment will be stored at the Pleasantville office.

**ZipCode:**

This table is needed at every location and is rarely updated. Also, this table doesn’t contain any sensitive information so we will replicate it at all offices.

**Play:**

Each office will need access to the entire table so we will replicate the Play table at each.

**Production:**

Each office will book/record their own production related expenses. Pleasantville office also keeps a copy of the entire table for administrative purposes.

**Performance:**

Each site stores a copy PerformanceFragment1, which has playID and performance date. For more detailed report of the performance, PerformanceFragment2 can be accessed as another important view which contains playID, performance date, performance time and performance year.

**TicketSale:**

Each site stores a copy of TicketSaleFragment1, which has subscriberID, playID, ticketincome. To know more detail about the ticket sale volume of each branch, each office keeps records of its own ticketsales, namely TicketSalePleasantville, TicketSaleYorktownHeights, TicketSaleWhitePlains, TicketSaleFinancialDistrict.

**DuesPayment:**

Each branch sites stores a copy of DuesPaymentFragment1, which has memberID, duesPaid, dateofDuesSubmission, duesAmount.

**Donation**

Each branch sites stores a copy of DonationFragment1 contains two important data, businessname and donationnumber.

**Ticket﻿﻿﻿﻿﻿﻿﻿**

Each office will need access to the different ticket fragments. These applications will be updated rarely but will need to be accessed on a consistent basis so we will replicate at each office.

* Step 10.5 - Make a table showing a geographical network, listing nodes and applications and showing the data fragments at each node.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| APPLICATION | Pleasantville | YorktownHeights | WhitePlains | FinancialDistrict |
| 1 Maintain tickets number records | TicketFragment1 | TicketFragment1 | TicketFragment1 | TicketFragment1 |
| 2 Produce ticket sales receipt | TicketFragment2TicketsSale  Subscriber | TicketFragment2  TicketSaleYorktownHeights  SubscriberFragment1 | TicketFragment2  TicketSaleWhitePlains  SubscriberFragment1 | TicketFragment2  TicketSaleFinancialDistrict  SubscriberFragment1 |
| 3 Produce the Tickets sold per Performance | TicketFragment3  TicketsSale Performance | TicketFragment3  TicketSaleYorktownHeights  PerformanceFragment1 | TicketFragment3  TicketSaleWhitePlains  PerformanceFragment1 | TicketFragment3  TicketSaleFinancialDistrict  PerformanceFragment1 |
| 4 Maintain sponsor records | Sponsors Donations | SponsorYorktownHeights  DonationFragment1 | SponsorWhitePlains  DonationFragment1 | SponsorFinancialDistrict  DonationFragment1 |
| 5 Maintain Income records | DuesPayment Donations TicketSales | DuesPaymentFragment1  DonationFragment1  TicketSaleYorktownHeights | DuesPaymentFragment1  DonationFragment1  TicketSaleWhitePlains | DuesPaymentFragment1  DonationFragment1  TicketSaleFinancialDistrict |
| 6 Maintain expenditures records | Production | ProductionYorktownHeights | ProductionWhitePlains | ProductionFinancialDistrict |
| 7 Producing active Subscribers report | Subscribers | SubscriberFragment1 | SubscriberFragment1 | SubscriberFragment1 |
| 8 Producing active Members report | Members ZipCode | MemberFragment1  ZipCode | MemberFragment1  ZipCode | MemberFragment1  ZipCode |
| 9 Maintain ticket sales records | TicketSales Subscribers | TicketSaleYorktownHeights  SubscriberFragment1 | TicketSaleWhitePlains  SubscriberFragment1 | TicketSaleFinancialDistrict  SubscriberFragment1 |
| 10 Producing the performance report | Susbscibers TicketSales Members Sponsors Performance | SubscriberFragment1  TicketSaleYorktownHeights  MemberFragment1  SponsorYorktownHeights  PerformanceFragment1 | SubscriberFragment1  TicketSaleWhitePlains  MemberFragment1  SponsorWhitePlains  PerformanceFragment1 | SubscriberFragment1  TicketSaleFinancialDistrict  MemberFragment1  SponsorFinancialDistrict  PerformanceFragment1 |
| 11 Producing the subscribers Sales Report | Subscribers TicketSales TicketFragment2 | SubscriberFragment1  TicketSaleYorktownHeights  TicketFragment2 | SubscriberFragment1  TicketSaleWhitePlains  TicketFragment2 | SubscriberFragment1  TicketSaleFinancialDistrict  TicketFragment2 |
| 12 Maintain subscriber list | Subscribers, ZipCode |  |  |  |
| 13 Maintain member list | Members, ZipCode |  |  |  |
| 14 Produce DuesPayment report | DuesPayment, Member |  |  |  |
| 15 Maintain Play list | Play |  |  |  |

* Step 10.6 - For each application in the geographical network, determine whether access will be local, remote, or compound. Make up a table showing each site, and the applications requiring local access, remote access, and compound access.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| APPLICATION | Pleasantville | YorktownHeights | WhitePlains | FinancialDistrict |
| 1 Maintain tickets number records- all local access | local | local | local | local |
| 2 Produce ticket sales receipt- all local access | local | local | local | local |
| 3 Produce the Tickets sold per Performance- all local access | local | local | local | local |
| 4 Maintain sponsor records- all local access | local | local | local | local |
| 5 Maintain Income records-all local access | local | local | local | local |
| 6 Maintain expenditures records-all local access | local | local | local | local |
| 7 Producing active Subscribers report- all local access | local | local | local | local |
| 8 Producing active Members report- all local access | local | local | local | local |
| 9 Maintain ticket sales records- all local access | local | local | local | local |
| 10 Producing the performance report- all local access | local | local | local | local |
| 11 Producing the subscribers Sales Report- all local access | local | local | local | local |
| 12 Maintain subscriber list- the subscriber list will be centralized in the main branch | local | remote | remote | remote |
| 13 Maintain member list-  the member list will be centralized in the main branch | local | remote | remote | remote |
| 14 Produce DuesPayment report - duesPaid will be remotely accessed by the branches | local | remote | remote | remote |
| 15 Maintain Play list - since Pleasantville is the only branch performing plays, remote access will be given to the other branches | local | remote | remote | remote |

* Step l0.7 - For each of the non-local accesses, identify the application and the location of the data. Estimate the number of accesses required per day using estimates such as low, medium, or high. If it is high, justify your choice of non-local storage.

The applications that will require remote access will be maintaining the subscriber and member lists, managing due payments, and maintaining the play lists. We have decided to maintain these only in our main branch to keep some of the information private. The members and subscribers list will be centralized in Pleasantville and the due payments will all go through there. Also, our Pleasantville location will be the only one performing plays as the other branches will primarily be ticket sale centers. For this reason, the other branches only need remote access to this data.

The amount of data per day will differ mainly based on how many different members register and how many new subscribers there are. As the majority of data will be centralized and in Pleasantville before opening the new branches, we don’t anticipate too many data transactions per day in regards to this. However, when due payments are due, there could be a large number of transactions on the deadline and days leading up to the deadline to show that the members have paid.

* Step 10.8 - Make any adjustments indicated by your analysis of applications and traffic, and plan a final geographical network.

Since most accesses are local, there is no need to adjust the geographical network shown in section 10.6