**Chapter 10 – Team Project - Planning for Distribution**

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

We will assume The Images and Photography Studio has expanded to three locations: Midtown (which is the original gallery), Uptown, and Downtown. We wish to distribute the database among the three locations. We will use the updated normalized set of relations developed from Chapter 6 as the global schema. The schema is:

1. Client (clientId, firstName, lastName)
2. Zip\_Code(zip,street,city,state)
3. Meeting (clientId, meetDate, meetTime, repname,zip)
4. Photographer (empId, firstName, lastName, street,zip)
5. Booking (clientId, bookingDate, bookingTime, duration, type, empId1, empId2, zip)
6. PackageMenu (packageNo, numWallet, albumType, albumPages, albumCover, price)
7. Job (contractNo, type, eventName, location, clientId, date, time, duration, cost, empId1, empId2, packageNoChosen, totalCost, amtPaid, amtDue)
8. Proof (contractNo, proofNo, quality)
9. Order (orderNo, dateOrdered, totalAmount, packageNoOrdered, contractNo)
10. OrderItem (orderNo, proofNo, size, quantity, dateDelivered).
11. Payment (contractNo, datePaid, payType, amount).
12. EmergencyContact(Clientid,eventid,Emergencyfirstname,Emergencylastname,Emergencyphonenumber).
13. Client\_Photographer (clientId, empId).

**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.**

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

The applications performed at each branch for that branch’s own data are:

1. Managing Photoshoot Records
2. Generating Invoices
3. Managing Invoice Records
4. Managing Client Records
5. Generating Photos for Sale Report
6. Generating Employee Performance Report
7. Generating Location-wise Photoshoot Report

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

1. Managing Photoshoot Records
2. Generating Invoices
3. Managing Invoice Records
4. Managing Client Records
5. Generating Photos for Sale Report
6. Generating Employee Performance Report
7. Generating Location-wise Photoshoot Report
8. Managing Photographer Records
9. Managing Employee Records
10. Producing Active Photographer Summary Report
11. Client Feedback
12. Photograph Sales Report
13. Photograph Feedback

**Step 10.3 - For each application, decide what tables are required.**

Let's associate the tables needed for each application based on the provided list:

1. Managing Photoshoot Records:

Tables: Photographer, Client, Zip\_Code, Meeting, Booking, PackageMenu, Job, Proof, Order, OrderItem, Payment, EmergencyContact, Client\_Photographer

1. Generating Invoices:

Tables: Client, Zip\_Code, Meeting, Booking, PackageMenu, Job, Proof, Order, OrderItem, Payment, EmergencyContact, Client\_Photographer

1. Managing Invoice Records:

Tables: Job, Order, OrderItem, Payment, EmergencyContact

1. Managing Client Records:

Tables: Client, Zip\_Code, Meeting, Booking, Job, EmergencyContact, Client\_Photographer

1. Generating Photos for Sale Report:

Tables: Photographer, Client, Zip\_Code, Meeting, Booking, PackageMenu, Job, Proof, Order, OrderItem, EmergencyContact, Client\_Photographer

1. Generating Employee Performance Report:

Tables: Photographer, Job, Payment

1. Generating Location-wise Photoshoot Report:

Tables: Zip\_Code, Meeting, Booking, Job, EmergencyContact, Client\_Photographer

1. Managing Photographer Records:

Tables: Photographer, Job, EmergencyContact, Client\_Photographer

1. Managing Employee Records:

Tables: Photographer, Job, Payment

1. Active Photographer Summary Report:

Tables: Photographer, Job, EmergencyContact, Client\_Photographer

1. Client Feedback:

Tables: Client

1. Photograph Sales Report:

Tables: Client, Photographer, Job, Order, OrderItem, EmergencyContact, Client\_Photographer

1. Photograph Feedback:

Tables: Client, Photographer

These associations operations involve joining tables to retrieve the required information. The relationships between these tables are essential for accurate and efficient data retrieval for each application.

**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.**

I. Managing Photoshoot Records:

PhotographerFragment1 = πphotographerId, firstName, lastName(Photographer)

PhotographerFragment2 = πphotographerId, firstName, lastName, street, zip, socialSecurityNumber(Photographer)

ClientFragment = πclientId, firstName, lastName(Client)

ZipCodeFragment = πzip, street, city, state(Zip\_Code)

MeetingFragment = πclientId, meetDate, meetTime, repname, zip(Meeting)

BookingFragment = πclientId, bookingDate, bookingTime, duration, type, photographerId1, photographerId2, zip(Booking)

PackageMenuFragment = πpackageNo, numWallet, albumType, albumPages, albumCover, price(PackageMenu)

JobFragment = πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue(Job)

ProofFragment = πcontractNo, proofNo, quality(Proof)

OrderFragment = πorderNo, dateOrdered, totalAmount, packageNoOrdered, contractNo(Order)

OrderItemFragment = πorderNo, proofNo, size, quantity, dateDelivered(OrderItem)

PaymentFragment = πcontractNo, datePaid, payType, amount(Payment)

EmergencyContactFragment = πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber(EmergencyContact)

ClientPhotographerFragment = πclientId, photographerId(Client\_Photographer)

ClientDowntown = σID LIKE ‘D%’ (Client)

ClientMidtown = σID LIKE ‘M%’ (Client)

ClientUptown =σID LIKE ‘U%’ (Client)

II. Generating Invoices:

ClientInvoiceFragment = πclientId, firstName, lastName, zip, dateOrdered, totalAmount, packageNoOrdered(Client, Order)

PaymentDowntown = σinvoiceNumber>0 and invoiceNumer <30000 (Payment)

PaymentMidtown = σ invoiceNumber>20000 and invoiceNumer <60000 (Payment)

PaymentUptown =σ invoiceNumber>40000 and invoiceNumer <90000 (Payment)

III. Managing Invoice Records:

JobInvoiceFragment = πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue(Job)

OrderInvoiceFragment = πorderNo, dateOrdered, totalAmount, packageNoOrdered, contractNo(Order)

OrderItemInvoiceFragment = πorderNo, proofNo, size, quantity, dateDelivered(OrderItem)

PaymentInvoiceFragment = πcontractNo, datePaid, payType, amount(Payment)

EmergencyContactInvoiceFragment = πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber(EmergencyContact)

IV. Managing Client Records:

ClientRecordFragment = πclientId, firstName, lastName, zip, meetDate, meetTime, repname, dateOrdered, totalAmount, packageNoOrdered(Client, Meeting, Order)

EmergencyContactRecordFragment = πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber(EmergencyContact)

ClientPhotographerRecordFragment = πclientId, photographerId(Client\_Photographer)

V. Generating Photos for Sale Report:

PhotographerSalesFragment = πphotographerId, firstName, lastName, dateOrdered, totalAmount, packageNoOrdered(Photographer, Order)

VI. Generating Employee Performance Report:

PhotographerPerformanceFragment = πphotographerId, firstName, lastName, datePaid, amount(Photographer, Payment)

JobPerformanceFragment = πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue(Job)

VII. Generating Location-wise Photoshoot Report:

ZipCodeReportFragment = πzip, street, city, state(Zip\_Code)

MeetingReportFragment = πclientId, meetDate, meetTime, repname, zip(Meeting)

BookingReportFragment = πclientId, bookingDate, bookingTime, duration, type, photographerId1, photographerId2, zip(Booking)

EmergencyContactReportFragment = πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber(EmergencyContact)

ClientPhotographerReportFragment = πclientId, photographerId(Client\_Photographer)

VIII. Managing Photographer Records:

PhotographerRecordFragment = πphotographerId, firstName, lastName, street, zip, datePaid, amount(Photographer, Payment)

JobPhotographerRecordFragment = πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue(Job)

EmergencyContactPhotographerRecordFragment = πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber(EmergencyContact)

ClientPhotographerPhotographerRecordFragment = πclientId, photographerId(Client\_Photographer)

IX. Managing Employee Records:

PhotographerEmployeeRecordFragment = πphotographerId, firstName, lastName, street, datePaid, amount(Photographer, Payment)

JobEmployeeRecordFragment = πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue(Job)

X. Active Photographer Summary Report:

ActivePhotographerSummaryFragment = πphotographerId, firstName, lastName, datePaid, amount(Photographer, Payment)

JobActivePhotographerSummaryFragment = πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue(Job)

EmergencyContactActivePhotographerSummaryFragment = πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber(EmergencyContact)

ClientPhotographerActivePhotographerSummaryFragment = πclientId, photographerId(Client\_Photographer)

XI. Client Feedback:

ClientFeedbackFragment = πclientId(Client)

XII. Photograph Sales Report:

PhotographerSalesReportFragment = πphotographerId, firstName, lastName, dateOrdered, totalAmount, packageNoOrdered(Photographer, Order)

ClientPhotographerSalesReportFragment = πclientId, photographerId(Client\_Photographer)

PhotographSales Downtown = σbranch=’Downtown’(PhotographSales)

PhotographSales Midtown= σbranch=’Midtown’(PhotographSales)

PhotographSales Uptown= σbranch=’Uptown’(PhotographSales)

XIII. Photograph Feedback:

ClientPhotographFeedbackFragment = πclientId, photographerId(Client\_Photographer)

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

I. Managing Photoshoot Records:

- PhotographerFragment1: πphotographerId, firstName, lastName (Midtown, Uptown, Downtown)

- PhotographerFragment2: πphotographerId, firstName, lastName, street, zip, socialSecurityNumber (Midtown)

- ClientFragment: πclientId, firstName, lastName (Midtown, Uptown, Downtown)

- ZipCodeFragment: πzip, street, city, state (Midtown, Uptown, Downtown)

- MeetingFragment: πclientId, meetDate, meetTime, repname, zip (Midtown, Uptown, Downtown)

- BookingFragment: πclientId, bookingDate, bookingTime, duration, type, photographerId1, photographerId2, zip (Midtown, Uptown, Downtown)

- PackageMenuFragment: πpackageNo, numWallet, albumType, albumPages, albumCover, price (Midtown, Uptown, Downtown)

- JobFragment: πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue (Midtown, Uptown, Downtown)

- ProofFragment: πcontractNo, proofNo, quality (Midtown, Uptown, Downtown)

- OrderFragment: πorderNo, dateOrdered, totalAmount, packageNoOrdered, contractNo (Midtown, Uptown, Downtown)

- OrderItemFragment: πorderNo, proofNo, size, quantity, dateDelivered (Midtown, Uptown, Downtown)

- PaymentFragment: πcontractNo, datePaid, payType, amount (Midtown, Uptown, Downtown)

- EmergencyContactFragment: πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber (Midtown, Uptown, Downtown)

- ClientPhotographerFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

II. Generating Invoices:

- ClientInvoiceFragment: πclientId, firstName, lastName, zip, dateOrdered, totalAmount, packageNoOrdered (Midtown, Uptown, Downtown)

III. Managing Invoice Records:

- JobInvoiceFragment: πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue (Midtown, Uptown, Downtown)

- OrderInvoiceFragment: πorderNo, dateOrdered, totalAmount, packageNoOrdered, contractNo (Midtown, Uptown, Downtown)

- OrderItemInvoiceFragment: πorderNo, proofNo, size, quantity, dateDelivered (Midtown, Uptown, Downtown)

- PaymentInvoiceFragment: πcontractNo, datePaid, payType, amount (Midtown, Uptown, Downtown)

- EmergencyContactInvoiceFragment: πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber (Midtown, Uptown, Downtown)

IV. Managing Client Records:

- ClientRecordFragment: πclientId, firstName, lastName, zip, meetDate, meetTime, repname, dateOrdered, totalAmount, packageNoOrdered (Midtown, Uptown, Downtown)

- EmergencyContactRecordFragment: πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber (Midtown, Uptown, Downtown)

- ClientPhotographerRecordFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

V. Generating Photos for Sale Report:

- PhotographerSalesFragment: πphotographerId, firstName, lastName, dateOrdered, totalAmount, packageNoOrdered (Midtown, Uptown, Downtown)

VI. Generating Employee Performance Report:

- PhotographerPerformanceFragment: πphotographerId, firstName, lastName, datePaid, amount (Midtown, Uptown, Downtown)

- JobPerformanceFragment: πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue (Midtown, Uptown, Downtown)

VII. Generating Location-wise Photoshoot Report:

- ZipCodeReportFragment: πzip, street, city, state (Midtown, Uptown, Downtown)

- MeetingReportFragment: πclientId, meetDate, meetTime, repname, zip (Midtown, Uptown, Downtown)

- BookingReportFragment: πclientId, bookingDate, bookingTime, duration, type, photographerId1, photographerId2, zip (Midtown, Uptown, Downtown)

- EmergencyContactReportFragment: πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber (Midtown, Uptown, Downtown)

- ClientPhotographerReportFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

VIII. Managing Photographer Records:

- PhotographerRecordFragment: πphotographerId, firstName, lastName, street, zip, datePaid, amount (Midtown, Uptown, Downtown)

- JobPhotographerRecordFragment: πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue (Midtown, Uptown, Downtown)

- EmergencyContactPhotographerRecordFragment: πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber (Midtown, Uptown, Downtown)

- ClientPhotographerPhotographerRecordFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

IX. Managing Employee Records:

- PhotographerEmployeeRecordFragment: πphotographerId, firstName, lastName, street, datePaid, amount (Midtown, Uptown, Downtown)

- JobEmployeeRecordFragment: πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue (Midtown, Uptown, Downtown)

X. Active Photographer Summary Report:

- ActivePhotographerSummaryFragment: πphotographerId, firstName, lastName, datePaid, amount (Midtown, Uptown, Downtown)

- JobActivePhotographerSummaryFragment: πcontractNo, type, eventName, location, clientId, date, time, duration, cost, photographerId1, photographerId2, packageNoChosen, totalCost, amtPaid, amtDue (Midtown, Uptown, Downtown)

- EmergencyContactActivePhotographerSummaryFragment: πclientId, eventId, Emergencyfirstname, Emergencylastname, Emergencyphonenumber (Midtown, Uptown, Downtown)

- ClientPhotographerActivePhotographerSummaryFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

XI. Client Feedback:

- ClientFeedbackFragment: πclientId (Midtown, Uptown, Downtown)

XII. Photograph Sales Report:

- PhotographerSalesReportFragment: πphotographerId, firstName, lastName, dateOrdered, totalAmount, packageNoOrdered (Midtown, Uptown, Downtown)

- ClientPhotographerSalesReportFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

XIII. Photograph Feedback:

- ClientPhotographFeedbackFragment: πclientId, photographerId (Midtown, Uptown, Downtown)

**Step 10.6 - Make a table showing a geographical network, listing nodes and applications and showing the data fragments at each node.**

|  |  |  |
| --- | --- | --- |
| Node | Applications | Data Fragments |
| Midtown | All applications | All Fragments |
| Uptown | Managing Photoshoot Records, Generating Invoices, Managing Client Records, Generating Photos for Sale Report, Generating Employee Performance Report, Generating Location-wise Photoshoot Report | Relevant Fragments |
| Downtown | Managing Photoshoot Records, Generating Invoices, Managing Client Records, Generating Photos for Sale Report, Generating Employee Performance Report, Generating Location-wise Photoshoot Report | Relevant Fragments |
| Midtown | All applications | All Fragments |
| Uptown | Managing Photoshoot Records, Generating Invoices, Managing Client Records, Generating Photos for Sale Report, Generating Employee Performance Report, Generating Location-wise Photoshoot Report | Relevant Fragments |
| Downtown | Managing Photoshoot Records, Generating Invoices, Managing Client Records, Generating Photos for Sale Report, Generating Employee Performance Report, Generating Location-wise Photoshoot Report | Relevant Fragments |

**Step 10.7 - For each application in the geographical network, determine whether access will be local, remote, or compound.**

|  |  |  |  |
| --- | --- | --- | --- |
| Site | Local Access | Remote Access | Compound Access |
| Midtown | All applications | None | None |
| Uptown | Managing Photoshoot Records, Generating Invoices, Managing Client Records, Generating Photos for Sale Report, Generating Employee Performance Report, Generating Location-wise Photoshoot Report | Managing Invoice Records, Photograph Sales Report, Photograph Feedback, Photograph Sales Report (Downtown, Midtown) | None |
| Downtown | Managing Photoshoot Records, Generating Invoices, Managing Client Records, Generating Photos for Sale Report, Generating Employee Performance Report, Generating Location-wise Photoshoot Report | Managing Invoice Records, Photograph Sales Report, Photograph Feedback, Photograph Sales Report (Uptown, Midtown) | None |

**Step l0.8 - For each of the non-local accesses, identify the application and the location of the data.**

Managing Invoice Records:

Remote Access: Uptown and Downtown

Data Location: Midtown

Access Frequency: Medium (Accessed for generating invoices, which might happen multiple times a day, but not as frequently as real-time operations.)

Photograph Sales Report:

Remote Access: Uptown and Downtown

Data Location: Midtown

Access Frequency: Medium (Accessed for generating reports, which might happen periodically.)

Photograph Feedback:

Remote Access: Uptown and Downtown

Data Location: Midtown

Access Frequency: Low (Accessed for feedback, which might not happen as frequently as other operations.)

Photograph Sales Report (Downtown, Midtown):

Remote Access: Uptown

Data Location: Midtown

Access Frequency: Low (Accessed for generating reports, which might not happen as frequently as other operations.)

Photograph Sales Report (Uptown, Midtown):

Remote Access: Downtown

Data Location: Midtown

Access Frequency: Low (Accessed for generating reports, which might not happen as frequently as other operations.)

**Step 10.9 - Make any adjustments indicated by your analysis of applications and traffic and plan a final geographical network.**

After analyzing applications and traffic, it seems that the data access requirements are reasonable, and the planned geographical network is suitable. No adjustments are indicated based on the current analysis. The final geographical network remains as described in Step 10.6.