EXERCISE 1: Single Table Queries Using Indexes

Database Instance and Indexes:

Table	Entries	Number of Entries per Block	Number of Blocks	Index Fields	Index Type	Keys per Node	Depth of B+ Tree
Customer	10,000	10	1,000	accountId	B+ tree and ordered sequential file	100	3
				lastName	B+ tree	50	3
				zipcode	hash	100	
Rental	1,000,000	100	10,000	accountId	B+ tree	100	3
				movieId	B+ tree	100	3
				date	B+ tree	100	2
Movie	10,000	20	500	movieId	B+ tree and ordered sequential file	100	3
				title	B tree	20	4
				genre	Hash	100	

from: *Principles of Database Systems* by Greg Riccardi
Addison Wesley 2001

Discuss execution strategies for the following queries:

SELECT genre FROM Movie WHERE title LIKE “Iron Man”

SELECT * FROM Rental WHERE date = ‘2008-10-31’

SELECT * FROM Rental WHERE accountId = 834 AND date = ‘2008-10-31’

SELECT * FROM Rental WHERE accountId = 834 AND date > ‘2008-10-31’

SELECT * FROM Rental WHERE accountId = 834
AND date > ‘2008-07-01’ AND date < ‘2008-10-31’

SELECT Title FROM Movie WHERE genre LIKE ‘documentary’

SELECT Title FROM Movie WHERE genre LIKE ‘comedy’
OR genre LIKE ‘action’ OR genre LIKE ‘drama’

SELECT MAX(date) FROM Rental WHERE accountID = 398

SELECT MAX(date) FROM Rental WHERE movieID = 7834

SELECT date FROM Rental WHERE accountId = 398 AND movieID = 7834

EXERCISE 2: Query Optimization

Estimated storage data on Big Hit Video relations:

Movie
10000 records
500 blocks
B+tree (3) on movieId
B+tree (3) on title

Videotape
100,000 records
800 blocks
B+tree (3) on movieId
primary index (3) on videoId

Store
100 records
20 blocks
primary index (2) on storeId

Customer
10000 records
1000 blocks
B+tree (3) on accountId
B+tree (3) on lastName

OtherUsers
3000 records
120 blocks
primary index (2) on accountId

Reservation
2000 records
75 blocks
no indexes

Employee
1000 records
100 blocks
primary index (2) on ssn
B+tree (2) on lastName

HourlyEmployee
800 records
20 blocks
B+tree (2) on ssn

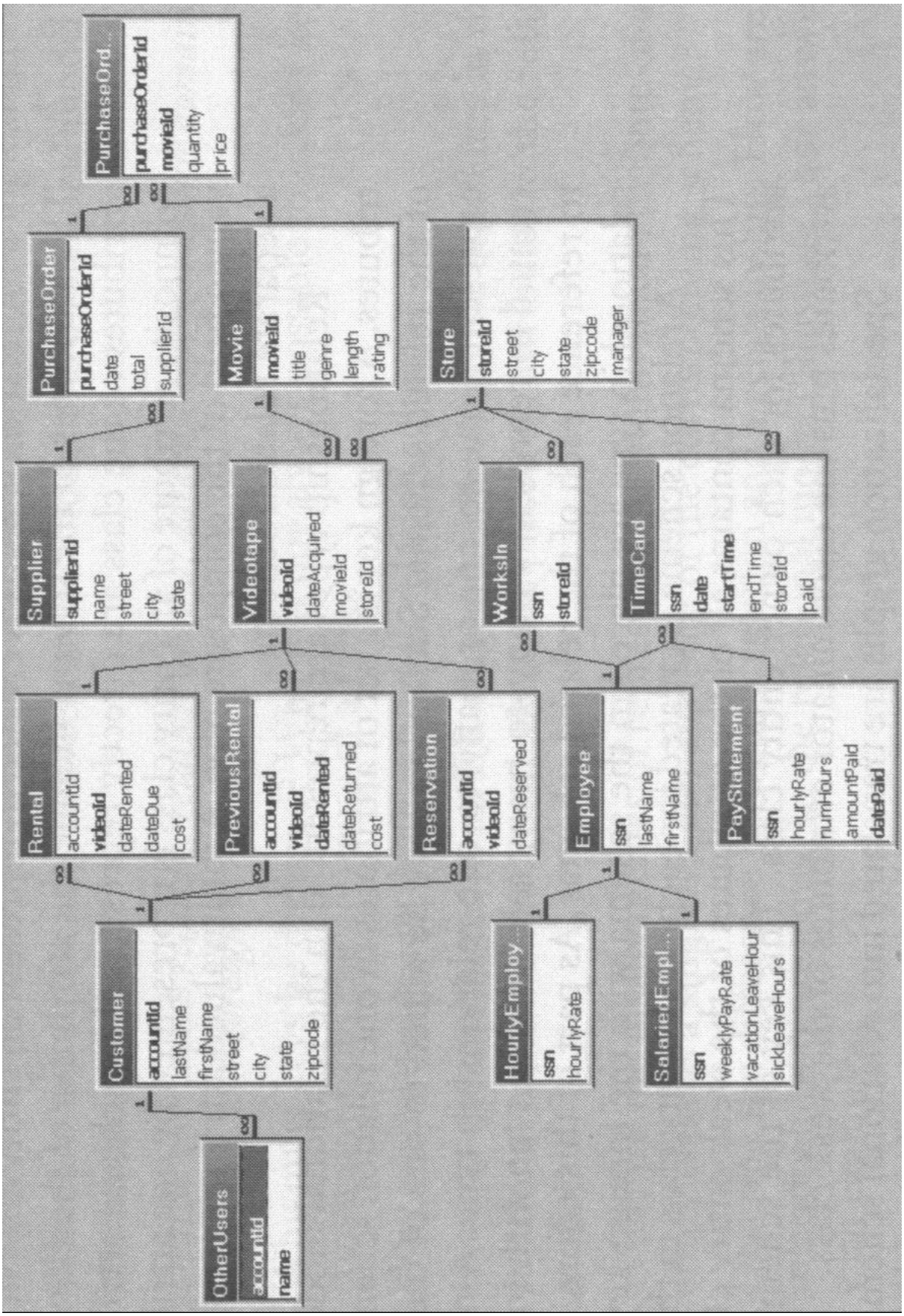
SalariedEmployee
200 records
8 blocks
B+tree (2) on ssn

WorksIn
1100 records
20 blocks
B+tree (2) on storeId

PurchaseOrder
5000 records
150 blocks
B+tree (3) on purchaseOrderId

PurchaseOrderDetail
70,000 records
2200 blocks
B+tree (3) on purchaseOrderId
B+tree (3) on movieId

Supplier
50 records
15 blocks
no indexes


PurchaseOrd...
purchaseOrderId
movieId
quantity
price
PurchaseOrder
purchaseOrderId
date
total
supplierId
Movie
movieId
title
genre
length
rating
Store
storeId
street
city
state
zipcode
manager
Supplier
supplierId
name
street
city
state
Videotape
videoId
dateAcquired
movieId
storeId
WorksIn
ssn
storeId
TimeCard
ssn
date
startTime
endTime
storeId
paid
Rental
accountId
videoId
dateRented
dateDue
cost
PreviousRental
accountId
videoId
dateRented
dateReturned
cost
Reservation
accountId
videoId
dateReserved
Employee
ssn
lastName
firstName
PayStatement
ssn
hourlyRate
numHours
amountPaid
datePaid
Customer
accountId
lastName
firstName
street
city
state
zipcode
HourlyEmploy...
ssn
hourlyRate
SalariedEmpl...
ssn
weeklyPayRate
vacationLeaveHour
sickLeaveHours
OtherUsers
accountId
name

Analyze each query and determine the following:

1) an SQL command to implement the query,
2) a literal interpretation of SQL command as an algebra expression tree
3) an efficient query plan (expression tree)
4) a rough estimate of the cost (disk accesses) of the efficient plan

Group 1: Supplier's address and purchase price for all purchases of the movie "Kung Fu Panda".

Group 2: Employee name and pay rate for all employees of the store managed by "Jeremy Pierce".

Group 3: Store address and reservation date for all reservations of the movie "Speed Racer".

Group 4: Account ID and movie title for all movies that have been rented more than once by the same account.