# Directions & Deliverable – Tim Capehart

Complete the following exercises in a single document. Label each question and/or letter clearly. This assignment requires the use of relational symbols. You may either do the assignment electronically or handwritten on paper. Regardless, please submit your work electronically to Blackboard by the due date specified.

#### Exercises

*The following tables form part of a database held in a relational DBMS:*

Formulate the following queries in relational algebra, tuple relational calculus, and domain relational calculus.

A screenshot of a cell phone

Description generated with very high confidence

1. (4 pts.) List all employees.

RA: Employee

TRC: { e | Employee(e) }

DRC: { E | E(Employee(empNo, fName, lName, address, DOB, sex, position, deptNo)) }

1. (4 pts.) List all the details of employees who are female.

RA: sex = ‘female’(Employee)

TRC: { e | (Employee(e) ^ e.sex = ‘female’) }

DRC: { E |  (E(Employee(empNo, fName, lName, address, DOB, sex, position, deptNo) ^ sex = ‘female’ }

1. (4 pts.) List the names and addresses of all employees who are managers.

RA: fName, lName, address ( position = ‘manager’ (Employee))

TRC: {e.fName, e.lName, e.address | (Employee(e) ^ e.position = ‘manager’)}

DRC: { fname, lName, address | (E(Employee(empNo, fName, lName, address, DOB, sex, position, deptNo) ^ position = ‘manager’ }

1. (4 pts.) Produce a list of the names and addresses of all employees who work for the IT department.

RA: fName, lName, address (Employee Employee.deptNo = Department.deptNo (deptName = ‘IT’ (Department))

TRC: { E.fName, E.lName, E.address | (Employee(E) ^ ( D) (Department(D)) ^ (E.deptNo = D.deptNo) ^ D.deptName = ‘IT’) }

DRC: { fName, lName, address | ( E(Employee(empNo, fName, lName, address, DOB, sex, position, deptNo) ^ D(Department(deptNo, deptName, mgrEmpNo)) ^( E.deptNo = D.deptNo ^ D.deptName = ‘IT’) }

1. (4 pts.) Produce a list of the names of all employees who work on the SCCS project.

RA: fName, lName (Employee Employee.deptNo = Project.deptNo ( projName = ‘SCCS’ (Project))) }

TRC: { E.fName, E.lName | (Employee(E) ^ ( P) (Project(P)) ^ E.deptNo = P.deptNo ^ P.projName = ‘SCCS’) }

DRC: { fName, lName | E(Employee(empNo, fName, lName, address, DOB, sex, position, deptNo) ^ P(Project(projNo, projName, deptNo)) ^ (E.deptNo = P.deptNo ^ P.projName = ‘SCCS’ )}

1. (4 pts.) Produce a complete list of all managers who are due to retire this year, in alphabetical order of surname.

RA: ( lName)( DOB = ‘%1953’ ^ position = ‘Manager’ (Employee))

TRC: { e ( e.lName) | Employee(e) ^ (e.DOB = ‘%1953’ ^ e.position = ‘Manager’) }

DRC: { E ( E.lName)|  E(Employee(empNo, fName, lName, address, DOB, sex, position, deptNo) ^ E.DOB = ‘%1953’ ^ E.position = ‘Manager’ ) }

1. (4 pts.) Find out how many employees are managed by “James Adam”.

RA: E 🡨 (EmployeeEmployee.deptNo = Department.deptNo(DepartmentDepartment.mgrEmpNo = Employee.empNo(Employee.fName = ‘James’ ^ Employee.lName = ‘Adam’(Employee))) COUNT (E)

TRC: { COUNT (E) | E1 (Employee(E1)) ^  E2 (Employee(E2)) ^ D (Department(D)) ^ E1.deptNo = D.deptNo ^ D.mgrEmpNo = E2.empNo ^ E2.fName = ‘James’ ^ E2.lName = ‘Adam’ }

DRC: { COUNT (E) |  (empNo1, fName1, lName1, address1, DOB1, sex1, position1, deptNo1, empNo2, fName2, lName2, address2, DOB2, sex2, posiotion2, deptNo2, deptNo3, deptName, mgrEmpNo) ^ Employee1(empNo1, fName1, lName1, address1, DOB1, sex1, position1, deptNo1) ^ Employee2(empNo2, fName2, lName2, address2, DOB2, sex2, position2, deptNo2)^Department(deptNo3, deptName, mgrEmpNo) ^ deptNo1 = deptNo3 ^ mgrEmpNo = empNo2 ^ fName2 = ‘James’ ^ lName2 = ‘Adam’ }

1. (4 pts.) Produce a report of the total hours worked by each employee.

RA: H 🡨 (EmployeeEmployee.empNo = WorksOn.empNo(hoursWorked(WorksOn)) SUM(H)

TRC: { SUM(H.hoursWorked) | E (Employee(E)) ^  W (WorksOn(W)) ^ E.empNo = W.empNo }

DRC: { SUM(H.hoursWorked) |  (empNo1, fName, lName, address, DOB, sex, position, deptNo, empNo2, projNo, dateWorked, hoursWorked) ^ Employee(empNo1, fName, lName, address, DOB, sex, position, deptNo) ^ WorksOn(empNo2, projNo, dateWorked,hoursWorked) ^ empNo1 = empNo2 }

1. (4 pts.)  For each project on which more than two employees worked, list the project number, project name, and the number of employees who work on that project.

RA: Can’t do this anymore.

TRC: NOPE

DRC: Nuh uh

1. (5 pts.) List the total number of employees in each department for those departments with more than 10 employees. Create an appropriate heading for the columns of the results table.

*The following tables form part of a Library database held in an RDBMS:*

A screenshot of a cell phone

Description generated with very high confidence

1. (4 pts.) List all book titles.

RA: title(Book)

TRC: { B.title | Book(B) }

DRC: { title | B(Book(ISBN, title, edition, year) ) }

1. (4 pts.) List all borrower details.

RA: Borrower

TRC: { B | Borrower(B) }

DRC: { B | B(Borrower(borrowerNo, borrowerName, borrowerAddress)) }

1. (4 pts.) List all book titles published in the year 2012.

RA: title(year = 2012(Book))

TRC: { B.title | Book(B) ^ B.year = 2012 }

DRC: { title | Book(ISBN, title, edition, year) ^ year = 2012 }

1. (4 pts.) List all copies of book titles that are available for borrowing.

RA: title(BookBook.ISBN = BookCopy.ISBN(available = true(BookCopy)))

TRC: { B.title | Book(B) ^ (BC)(BookCopy(BC)) ^ B.ISBN = BC.ISBN ^ BC.available = true }

DRC: { title | Book(ISBN1, title, edition, year) ^ BookCopy(copyNo, ISBN2, available)) ^ ISBN1 = ISBN2 ^ available = true }

1. (4 pts.) List all copies of the book title “Lord of the Rings” that are available for borrowing.

RA: title = ‘Lord of the Rings’ (Book Book.ISBN = BookCopy.ISBN(available = true (BookCopy)))

TRC: { BC | BookCopy(BC) ^ B (Book(B)) ^ B.title = ‘Lord of the Rings’ ^ B.ISBN = BC.ISBN ^ BC.available = true }

DRC: { BC | BC(BookCopy(copyNo, ISBN2, available)) ^ Book(ISBN1, title, edition, year) ^ ISBN1 = ISBN2 ^ available = true }

1. (4 pts.) List the names of borrowers who currently have the book title “Lord of the Rings” on loan.

RA: borrowerName(BorrowerBorrower.borrowerNo = BookLoan.borrowerNo(BookLoanBookLoan.copyNo = BookCopy.copyNo(BookCopyBookCopy.ISBN = Book.ISBN(title = ‘Lord of the Rings’(Book))))) }

TRC: { Bo.borrowerName | Borrower(Bo) ^ B (Book(B)) ^ (BC)(BookCopy(BC)) ^ (BL)(BookLoan(BL)) ^ B.ISBN = BC.ISBN ^ BL.copyNo = BC.copyNo ^ BO.borrowerNo = BL.BorrowerNo ^ B.title = ‘Lord of the Rings’ }

DRC: { BN | ((ISBN, title, edition, year, copyNo, ISBN2, available, borrowerNo1, borrowerName, borrowerAddress, copyNo2, dateOut, dateDue, borrowerNo2)^(Book(ISBN1, title, edition, year)^(BookCopy(copyNo1, ISBN2, available)^(Borrower(borrowerNo1, borrowerName, borrowerAddress,)^(BookLoan(copyNo2, dateOut, dateDue, borrowerNo2) ^ ISBN1 = ISBN2 ^ borrower1 = borrower2 ^ copyNo2 = copyNo2 ^ title = ‘Lord of the Rings’ }

1. (4 pts.)  List the names of borrowers with overdue books.

RA: borrowerName(BorrowerBorrower.borrowerNo = BookLoan.borrowerNo(currentDate > dateDue(BookLoan))

TRC: { BO.borrowerName |(BO)(Borrower(BO)) ^ (BL)(BookLoan(BL)) ^ BL.borrowerNo = BO.borrowerNo ^ currentDate > dateDue }

DRC: { BN | (( borrowerNo1, borrowerName, borrowerAddress, copyNo, dateOut, dateDue, borrowerNo2) ^ (Borrower(borrowerNo1, borrowerName, borrowerAddress) ^ (BookLoan(copyNo, dateOut, dateDue, borrowerNo2) ^ borrwerNo1 = borrowerNo2 ^ currentDate > dateDue }

1. (4 pts.)  How many copies of ISBN ‘0-321-52306-7’ are there?

RA: COUNT (BookCopy)) (BookBookCopy.ISBN = ‘0-321-52306-7’(BookCopy))

TRC: { BC🡨 COUNT | Book(B) ^  BC BookCopy(BC) COUNT (BC.ISBN = ‘0-321-52306-7’) }

DRC: { BC 🡨 COUNT | (ISBN1, title, edition, year, copyNo, ISBN2, available)^(Book(ISBN1, title, edition, year) ^ BookCopy(copyNo, ISBN2, available) ^ COUNT (ISBN1 = ISBN2 = ‘0-321052306-7’) }

1. (4 pts.)  How many copies of ISBN “0-321-52306-7” are currently available?

RA: BC 🡨 (BookBookCopy.ISBN = ‘0-321-52306-7’(available = true(BookCopy)) COUNT(BC)

TRC: { BC 🡨 COUNT | | Book(B) ^  BC BookCopy(BC)COUNT (BC.ISBN = ‘0-321-52306-7’ ^ BC.available = true) }

DRC: { BC 🡨 COUNT |  (ISBN1, title, edition, year, copyNo, ISBN2, available) ^ Book(ISBN1, title, edition, year) ^ BookCopy(coypNo, ISBN2, available) ^ COUNT (ISBN1 = ISBN2 = ‘0-321052306-7’ ^ BC.available = true) }

1. (4 pts.)  How many times has the book title with ISBN “0-321-52306-7” been borrowed?

RA: BC 🡨 (BookBook.ISBN = BookCopy.ISBN = ‘0-321-52306-7’(BookCopyBookCopy.copyNo = BookLoan.copyNo(dateOut != NULL(BookLoan))) COUNT(BC)

TRC: { BC 🡨 COUNT | Book(B) ^  BC BookCopy(BC) ^  BL BookLoan(BL) ^ COUNT(BC.ISBN = ‘0-321-52306-7’ ^ BC.copyNo = BL.copyNo ^ BL.dateOut != NULL) }

DRC: { BC 🡨 COUNT |  (ISBN1, title, edition, year, copyNo1, ISGN2, available, copyNo2, dateOut, dateDue, borrowerNo) ^ (Book(ISBN1, title, edition, year) ^ BookCopy(copyNo1, ISBN2, available) ^ BookLoan(copyNo2, dateOut, dateDue, borrowerNo) ^ COUNT(ISBN1 = ISBN2 = ‘0-321-52306-7’ ^ copyNo1 = copyNo2 ^ dateOut != NULL) }

1. (4 pts.)  Produce a report of book titles that have been borrowed by “Peter Bloomfield”.

RA: title(BookBook.ISBN = BookCopy.ISBN(BookCopyBookCopy.copyNo = BookLoan.copyNo(BookLoanBookLoan.borrowerNo = Borrower.borrowerNo(borrowerName = ‘Peter Bloomfield’(Borrower)))))

TRC: { B.title | Book(B) ^  BC BookCopy(BC) ^  BL BookLoan(BL) ^  BO Borrower(BO) ^ B.ISBN = BC.ISBN ^ BC.copyNo = BL.copyNo ^ BL.borrowerNo = BO.borrowerNo ^ BO.borrowerName = ‘Peter Bloomfield’ }

DRC: { bookTitle | (ISBN1, title, edition, year, copyNo1, ISBN2, available, borrowerNo1, borrowerName, borrowerAddress, copyNo2, dateOut, dateDue, borrowerNo2) ^ Book(ISBN1, title, edition, year) ^ BookCopy(copyNo1, ISBN2, available) ^ Borrower(borrowerNo1, borrowerName, borrowerAddress) ^ BookLoan(copyNo2, dateOut, dateDue, borrowerNo2) ^ ISBN1 = ISBN2 ^ copyNo1 = copyNo2 ^ borrowerNo1 = borrowerNo2 ^ borrowerName = ‘Peter Bloomfield’ }

1. (5 pts.)  For each book title with more than 3 copies, list the names of library members who have borrowed them.

RA: B 🡨 (BookBook.ISBN = BookCopy.ISBN(BookCopy) COUNT > 3(B) 🡪 Z borrowerName(BookCopyZ.copyNo = BookLoan.copyNo(BookLoanBookLoan.borrowerNo = Borrower.borrowerNo(Borrower)))

TRC: { BO.borrowerName | Book(B) ^  BC BookCopy(BC) ^  BL BookLoan(BL) ^  BO Borrower(BO) ^ COUNT > 3 (B.ISBN = BC.ISBN ^ BC.copyNo = BL.copyNo ^ BL.borrowerNo = BO.borrowerNo) }

DRC: { borrowerName |  (ISBN1, title, edition, year, copyNo1, ISBN2, available, borrowerNo1, borrowerName, borrowerAddress, copyNo2, dateOut, dateDue, borrowerNo2) ^ Book(ISBN, title, edition, year) ^ BookCopy(copyNo1, ISBN2, available) ^ BookLoan(borrowerNo1, borrowerName, borrowerAddress) ^ COUNT > 3(ISBN1 = ISBN2 ^ copyNo1 = copyNo2 ^ borrowerNo1 = borrowerNo2) }

1. (5 pts.)  Produce a report with the details of borrowers who currently have books overdue.

RA: borrowerNo, borrowerName, borrowerAddress(BorrowerBorrower.borrowerNo = BookLoan.borrowerNo(currentDate > dateDue(BookLoan)))

TRC: { BO |  BL BookLoan(BL) ^  BO Borrower(BO) ^ BO.borrowerNo = BL.borrowerNo ^ currentDate > BL.dueDate }

DRC: { BO |  (borrowerNo1, borrowerName, borrwerAddress, copyNo, dateOut, dateDue, borrowerNo2) ^ Borrower(borrowerNo1, borrowerName, borrowerAddress) ^ BookLoan(copyNo, dateOut, dateDue, borrowerNo2) ^ borrowerNo1 = borrowerNo2 ^ currentDate > dateDue }

1. (5 pts.)  Produce a report detailing how many times each book title has been borrowed.

RA: BC 🡨 (BookBook.ISBN = BookCopy.ISBN(BookCopyBookCopy.copyNo = BookLoan.copyNo(dateOut != NULL(BookLoan)) COUNT 🡪 Z COUNT(Z)

TRC: { BC 🡨 COUNT | Book(B) ^  BC BookCopy(BC) ^  BL BookLoan(BL) ^ COUNT(B.ISBN = BC.ISBN ^ BC.copyNo = BL.copyNo ^ BL.dateOut != NULL }

DRC: { BC 🡨 COUNT |  (ISBN1, title, edition, year, copyNo1, ISBN2, available, copyNo2, dateOut, dateDue, borrowerNo) ^ Book(ISBN1, title, edition, year) ^ BookLoan(copyNo2, dateOut, dateDue, borrowerNo) ^ COUNT(ISBN1 = ISBN2 ^ copyNo1 = copyNo2 ^ dateOut != NULL) }