# CS374 -- ASSIGNMENT #3

## NAME: Aaron Borjas DUE: 06 Oct Received: .

**GRADE:**

|  |  |  |
| --- | --- | --- |
| **CATEGORY** | **POINTS** |  |
| EX03\_01 |  | 20 |
| EX03\_02 |  | 20 |
| EX03\_03 |  | 30 |
| EX03\_04 |  | 5 |
| EX03\_05 |  | 5 |
| EX03\_06 |  | 5 |
| EX03\_07 |  | 5 |
| EX03\_08 |  | 5 |
| EX03\_09 |  | 5 |
|  |  |  |
| **TOTAL** |  | 100 |

## READING: Elmasri Sections 7.1, 8.1-8.5

## EXERCISES:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EX03\_01** | Exercise 7.5, page 236. You may use SQL or relational algebra.  **a)**  SELECT D.Dname, COUNT(\*) as "Employee Count", AVG(Salary) AS "Average Salary"  FROM Department AS D, Employee as E  WHERE D.Dnumber = E.Dno  GROUP BY Dname  HAVING AVG(Salary) > 30000   | **Query1** | | | | --- | --- | --- | | **Dname** | **"Employee Count"** | **"Average Salary"** | | Research | 4 | 42000 |   **b)**  **We can specify a query for male employees in each department making more than $30,000. This is done with a nested query, as so:**  SELECT D.Dname, COUNT(\*) as "Employee Count", AVG(Salary) AS "Average Salary"  FROM Department AS D, Employee as E  WHERE D.Dnumber = E.Dno AND E.Sex = "M" AND E.Dno IN  (SELECT E.Dno  FROM Employee  GROUP BY E.Dno  HAVING AVG(E.Salary > 30000))  GROUP BY D.Dname   | **Query1** | | | | --- | --- | --- | | **Dname** | **"Employee Count"** | **"Average Salary"** | | 倁 | 1 | 38000 | | 倂 | 2 | 47500 | |
| **EX03\_02** | Exercise 7.6, page 237. You may use SQL or relational algebra.  **a)**  SELECT S.FName, S.Major  FROM Student AS S  WHERE NOT EXISTS (  SELECT \*  FROM Grade\_Report AS G  WHERE G.Student\_number = S.Student\_Number AND G.Grade <> 'A'  **)**   | **Query1** | | | --- | --- | | **FName** | **Major** | | Stud | CS |   **b)**  SELECT S.FName, S.Major  FROM Student AS S  WHERE NOT EXISTS (  SELECT \*  FROM Grade\_Report AS G  WHERE G.Student\_number = S.Student\_Number AND G.Grade = 'A'  )   | **Query1** | | | --- | --- | | **FName** | **Major** | | Smith | CS | |
| **EX03\_03** | Exercise 7.7, page 237. You may use SQL or relational algebra.  **a)**  SELECT Fname, Minit, Lname  FROM Employee  WHERE Dno = (  SELECT Dno  FROM Employee  WHERE Salary = (  SELECT MAX(Salary)  FROM Employee  )  )   | **Query1** | | | | --- | --- | --- | | **Fname** | **Minit** | **Lname** | | John | B | Smith | | Franklin | T | Wong | | James | E | Borg | | Jennifer | S | Wallace |   **b)**  SELECT Fname, Minit, Lname  FROM Employee  WHERE Super\_ssn IN (  SELECT Ssn  FROM Employee  WHERE Super\_ssn = 888665555  )   | **Query1** | | | | --- | --- | --- | | **Fname** | **Minit** | **Lname** | | John | B | Smith | | Joyce | A | English | | Ramesh | K | Narayan | | Ahmad | V | Jabbar | | Alicia | J | Zelaya |   **c)**  SELECT Fname, Minit, Lname  FROM Employee  WHERE Salary >= (  SELECT MIN(Salary)  FROM Employee  ) + 10000   | **Query1** | | | | --- | --- | --- | | **Fname** | **Minit** | **Lname** | | Franklin | T | Wong | | Ramesh | K | Narayan | | James | E | Borg | | Jennifer | S | Wallace | |

**The queries below use the library database, which can be obtained from our Teams site (Access) or SQL Server on CS1. See me for help getting into the Library database if you are having trouble.**

**For 04-05, describe what the queries are doing in English. Note, do not use phrases such as “we join persons with orders”. Instead, say “we output persons with their corresponding orders.” Turn in the query in English and up to the first five result rows when the SQL is executed.**

**EX03\_04 –** SELECT lastname, address, zipcode FROM person WHERE property="professor" OR property="visitor";

Output the last name, address, and zipcode of any person who is a professor or visitor of the library.

| **Query2** | | |
| --- | --- | --- |
| **lastname** | **address** | **zipcode** |
| Ege |  |  |
| Muthusamy | 23 95th Avenue | 67946 |
| Vireday | 2 8th Avenue | 67894 |
| Wu | 20 10th Avenue | 89647 |
| DeFreez | 17 22nd Avenue | 11522 |

**EX03\_05 –** SELECT \* FROM person P, Action A WHERE P.personid=A.personid AND A.returndate>=#1999-3-19# AND firstname="George";

Output all the information on any person named “George” who returned a book on or after March 3, 1999

| **Query2** | | | | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **P.PersonId** | **FirstName** | **MiddleName** | **LastName** | **ParentName** | **Address** | **City** | **ZipCode** | **PhoneNumber** | **EMail** | **Property** | **StudentNo** | **IDCardNo** | **TransactionId** | **A.PersonId** | **BorrowDate** | **DueDate** | **ReturnDate** |
| 116 | George |  | Berkling | Bill | 3 71st Avenue | Portland | 11855 | 3424812 | gbb@softlab.lib.ece. | Student |  | M113586 | 153 | 116 | 12/5/2000 | 12/15/2000 | 12/7/2000 |
| 248 | George |  | Kuzas | Dawn | 34 64th Avenue | Portland | 17563 | 7722289 | george@esd.ece.un.de | Visitor | 0963796 | M159156 | 265 | 248 | 1/10/2000 | 1/20/2000 | 1/11/2000 |
| 447 | George |  | Micelbart | Tom |  | Seattle |  | 7721511 | gemit@cs.un.de | Student | 7862518 | K907849 | 742 | 447 | 10/8/2000 | 10/18/2000 | 10/18/2000 |
| 450 | George |  | Tavaglione | John | 1 6th Avenue | Portland | 17856 | 9078435 |  | Professor |  |  | 747 | 450 | 10/8/2000 | 10/18/2000 | 10/18/2000 |
| 462 | George |  | Ashmore | John | 4 33th Avenue | Paris | 78875 | 7994378 |  | Student | 9089786 |  | 815 | 462 | 10/25/2000 | 11/4/2000 | 10/27/2000 |

**For 06-07, write SQL queries for the following English queries. Turn in the query in SQL and up to the first five result rows when the SQL is executed.**

**EX03\_06 –** List all information on students who live in London

SELECT \*

FROM Person P

WHERE P.property = "student" AND P.City = "London"

| **Query2** | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PersonId** | **FirstName** | **MiddleName** | **LastName** | **ParentName** | **Address** | **City** | **ZipCode** | **PhoneNumber** | **EMail** | **Property** | **StudentNo** | **IDCardNo** |
| 138 | Pete |  | Gupta | Ethan |  | London |  | 8044033 |  | Student | 0906783 | U89674 |
| 257 | Nicholas |  | Korn | George | 9 71st Avenue | London | 89645 | 6831803 |  | Student |  |  |
| 327 | Elias |  | Breitkreuz | Peter | 53 18th Avenue | London | 10437 | 5246887 |  | Student | 0985673 | X166036 |
| 388 | Nickolas |  | Mark | Manning | 7 60th Avenue | London | 15122 | 8060973 | nimama@cs.un.de | Student | 8946732 | M678943 |
| 414 | Hercules |  | Wadell | Pete | 21 78th Avenue | London | 89043 | 4323830 | hewa@cs.un.de | Student | 9009785 | D119949 |

**EX03\_07 –** List the names of all students who returned a book after its due date (that is, its ReturnDate is later than its DueDate).

SELECT Firstname, MiddleName, Lastname

FROM Person P, Action A

WHERE P.personid = A.personid AND A.returndate >A.duedate

| **Query2** | | |
| --- | --- | --- |
| **Firstname** | **MiddleName** | **Lastname** |
| Jim |  | Ganta |
| Helen |  | Thamann |
| Nick |  | House |
| Jason |  | Noel |
| Williams |  | Schill |
| Mary |  | Zhang |

**For 08-09, write queries in relational algebra for the following English queries**

**EX03\_08 –** EX03\_06

π\*(σProperty = “Student” and City = “London”(Person))

**EX03\_09 –** EX03\_07

πfirstname, middlename, lastname(σreturndate > duedate(πpersonid (Person) ∪ πpersonid (Action)))