# CS374 -- ASSIGNMENT #4

## NAME: Aaron Borjas DUE: 13 Oct Received: .

**GRADE:**

|  |  |  |
| --- | --- | --- |
| **CATEGORY** | **POINTS** |  |
| EX04\_01 |  | 20 |
| EX04\_02 |  | 40 |
| EX04\_03 |  | 10 |
| EX04\_04 |  | 10 |
| EX04\_05 |  | 10 |
| EX04\_06 |  | 10 |
| BONUS |  | 10 |
| BONUS #2 |  | 10 |
|  |  |  |
| **TOTAL** |  | 100 |

## READING: Elmasri Section 7.3

## EXERCISES:

**EX04\_01 –** Convert the SQL you used in Exercise 7.5 to stored procedures in the DBMS of your choice

CREATE PROCEDURE DeptSalary

AS

BEGIN

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

END

GO

CREATE PROCEDURE MaleSalary

AS

BEGIN

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

END

GO

**EX04\_02 –** Exercise 7.8, page 237

**a)**

CREATE VIEW V1

AS SELECT D.Dname, E.Fname, E.Lname, E.Salary

FROM Department AS D, Employee AS E

WHERE D.Mgrssn=E.Ssn

**b)**

CREATE VIEW V2

AS SELECT E.Fname, E.Lname, S.Fname, S.Lname, E.Salary

FROM Department AS D, Employee AS E, Employee AS S

WHERE Dname = 'Research' AND D.Dnumber = E.Dno AND E.Super\_ssn = S.Ssn

**c)**

CREATE VIEW V3

AS SELECT P.Pname, D.Dname, COUNT(\*) AS "Employees", SUM(W.Hours) AS "Hours per Week"

FROM Project P, Department D, Works\_On W

WHERE P.Dnum = D.Dnumber AND P.Pnumber = W.Pno

GROUP BY Pname, Dname

**d)**

CREATE VIEW V4

AS SELECT P.Pname, D.Dname, COUNT(\*), SUM(W.HOURS)

FROM Project P, Department D, Works\_On W

WHERE P.PNumber = P.Pno AND D.Dnum = Dnumber

GROUP BY Pname, Dname

HAVING COUNT(\*) > 1

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

**For 03-04, 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.**

**EX04\_03 –** SELECT property, COUNT(\*) AS cnt FROM person GROUP BY property;

Go through the list of people and sum the total number of people in each category of people in relation to the library.

| **Query2** | |
| --- | --- |
| **property** | **cnt** |
|  | 1 |
| "Student" | 1 |
| 0347834 | 1 |
| Post Grad. | 25 |
| Post. Grad | 1 |

**EX04\_04 –** SELECT S.SubjectType, COUNT(\*) FROM Book\_Description BD, Subject S

WHERE BD.SubjectID=S.SubjectID GROUP BY S.SubjectType;

Count the number of books in each subject.

| **Query2** | |
| --- | --- |
| **SubjectType** | **Expr1001** |
| Algorithms | 20 |
| Artificial Intelligence | 24 |
| Communications | 21 |
| Computer Hardware | 56 |
| Computer Software | 33 |

**For 05-06, 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.**

**EX04\_05 –** List the names of all authors who wrote more than one book in 1991. Be careful here, two authors may have the same name.

SELECT A.Firstname, A.Lastname, Q.bookcount

FROM Author A, (

SELECT RA.AuthorID, Count(\*) AS bookcount

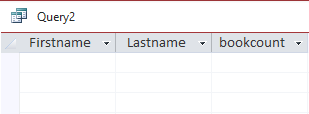
FROM RelAuth RA, Book\_Description B

WHERE B.Year = 1991 AND B.BookDescID = RA.BookDescId

GROUP BY RA.AuthorID

) as Q

WHERE A.AuthorID = Q.AuthorID AND bookcount > 1



**EX04\_06 –** List the number of distinct book authors for each year a book was published.

SELECT B.Year, Count(R.AuthorID)

FROM Book\_Description AS B

INNER JOIN RelAuth R on B.BookDescId = R.BookDescId

GROUP BY Year

| **Query2** | |
| --- | --- |
| **Year** | **Num Authors** |
| 1966 | 5 |
| 1967 | 10 |
| 1968 | 9 |
| 1969 | 8 |
| 1970 | 14 |
| 1971 | 16 |
| 1972 | 18 |

**BONUS –** For each book, list its title. If that book has been borrowed since 1/1/2000, list also the date that the book was borrowed. (Hint: You will need OUTER JOIN and nested sub queries.)

**BONUS –** See In-class Exercise in Complex SQL slides, slide #50