**CS 457 – Data modeling and Implementation techniques (Lab) Venue: Home**

**Time Duration: 12.45 pm to 3 pm**

**Instructions: 1) Open book, Open notes**

**2) Try using Advanced SQL queries (JOIN, INTERSECT, GROUP BY, ORDER BY, USE, IN, NOT IN, LIKE, NOT LIKE, etc) where applicable.**

**3) Answer the questions in the Quiz in the best possible way you can.**

**Quiz 2: [Max: 50 points]**

1. **[6 x 3 = 18 points]** Write a SQL query that returns any two rows out of the list of 9 rows given as:

(1, 1, 1, 0), (1, 1, 1, 0), (1, 1, 1, 0), (1, 1, 2, 1), (1, 1, 2, 0), (1, 1, 2, 0), (1, 1, 3, 0), (1, 1, 3, 0), (1, 1, 3, 0);

1. From the original list of nine rows, display the first 3 column entries.
2. Sum the column entries which does not have a zero in it (for instance, the last column entries have zero in it. You may not want to add the column entries on the last column, since it has all zeros in it).
3. Create a view (subset of a Table) which contains the first 3 columns. Display the view.
4. Now, insert into the view the following rows: (4, 5, 6), (5, 7, -3), (6, 4, -8).

e) From the updated view in part (d), display only those rows which have one of their elements as negative.

(f) Now, rearrange the rows in the View so that they are listed in the descending order of their row sum. (Note: Rows which have higher row sum will be listed on the top).

1. Answer the following questions (you need to show the source code + output for each subdivision) on the basis of the entries in the Tables that you see in Chapter 6 (Slides 6-12, 6-13): **[18 points]**

(a) Implement a Three-way Join (can be any type of Joins) between different Tables. [4 points]

(b) Implement a Three-way Join (one inner join, two outer) between different Tables comprising two sub-queries. [4 points]

(c) Implement a Four-way Join (two inner and two outer) between different Tables comprising three sub-queries. [4 points]

(d) Implement a Cross-Join between any two distinct Tables. [2 points]

(e) Implement a Self-Join between two Tables [2 points]

(f) Implement a Natural Join between two Tables [2 points]

III. [**14 points]** Write SIX queries (each with different combinations of Four way Joins – Inner, Left outer, Right outer, Natural join, Equi join, Cross join) to use the following Tables to implement Four-way joins using all types of joins introduced using the University database.

**Instructors, Students, Grade reports, Pre-requisites, Course, Section**