**United College of Engineering and Research, Allahabad**

**Department of Computer Science & Information Technology**

**IInd Sessional Examination (2017-18)**

**B.Tech. (Vth Semester (CS & IT))**

**Database Management System**

**Subject Code:** NCS-502

**Time:** 2.00 hours **Max. Marks:** 30

**Note:** There are three sections in this paper. All sections are compulsory.

**Section-A**

**Note:** All questions are **compulsory**. Each question has equal marks. **10\*1=10**

1. Define Triggers.
2. What is meant by normalization of data?
3. Define 3NF.
4. Define multi-valued dependency.
5. What are Armstrong rules?
6. Define recoverable schedule.
7. Write name of all the states of a transaction.
8. Explain division operation in relational algebra with example.
9. What is view in database? Explain it with example.
10. Define canonical cover of set of FDs.

**Section-B**

**Note:** Attempt any **five** questions. Each question has equal marks. **5\*2=10**

1. Give a set of FDs for the relation schema *R(A,B,C,D)* with primary key *AB* under which *R* is in 2NF but not in 3NF.
2. Explain about the following clauses with example queries.

(i) Group by

(ii) Order by

(iii) Aggregation functions.

1. Is the decomposition in 4NF always dependency preserving and lossless? Explain with an example.
2. Consider the following relation schema R(A,B,C,D,E) and FD’s A🡪BC, C🡪A,D🡪E, F🡪A, E🡪D

Is the decomposition of R into R1(A, C, D), R2(B,C, D) AND R3(E,F,D) lossless?

1. Define transaction properties.
2. Define conflict and view serializability.

**Section-C**

**Note:** Attempt any **two** questions. Each question has equal marks. **2\*5=10**

1. Consider the following relations containing airline flight information:

Flights(*flno:* integer, *from:* string, *to:* string, *distance:* integer, *departs:* time, *arrives:* time)

Aircraft(*aid:* integer, *aname:* string, *cruisingrange:* integer)

Certified(*eid:* integer, *aid:* integer)

Employees(*eid:* integer, *ename:* string, *salary:* integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in relational algebra:-

1. Find the *aid*s of all aircraft that can be used on non-stop flights from Bonn to Madras.
2. Identify the flights that can be piloted by every pilot whose salary is more than $100,000.
3. Find the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.
4. Find the *eid*s of employees who make the highest salary.
5. Find the *eid*s of employees who make the second highest salary.
6. Consider the following relations:

Student(*snum:* integer, *sname:* string, *major:* string, *level:* string, *age:* integer)

Class(*name:* string, *meets at:* string, *room:* string, *fid:* integer)

Enrolled(*snum:* integer, *cname:* string)

Faculty(*fid*: integer, *fname:* string, *deptid:* integer)

The meaning of these relations is straightforward; for example, Enrolled has one record

per student-class pair such that the student is enrolled in the class.

Write the following queries in SQL. No duplicates should be printed in any of the answers.

1. Find the names of faculty members who teach in every room in which some class is taught.
2. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.
3. For each level, print the level and the average age of students for that level.
4. For all levels except JR, print the level and the average age of students for that level.
5. For each faculty member that has taught classes only in room R128, print the faculty member’s name and the total number of classes she or he has taught.
6. Suppose you are given a relation *R* with four attributes *ABCD*. For each of the following sets of FDs, assuming those are the only dependencies that hold for *R*, do the following: (a) Identify the candidate key(s) for *R*. (b) Identify the best normal form that *R* satisfies (1NF, 2NF, 3NF, or BCNF). (c) If *R* is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies.
7. F={ *A → B, BC → D, A → C }*
8. F={ *AB → C, AB → D, C → A, D → B}*