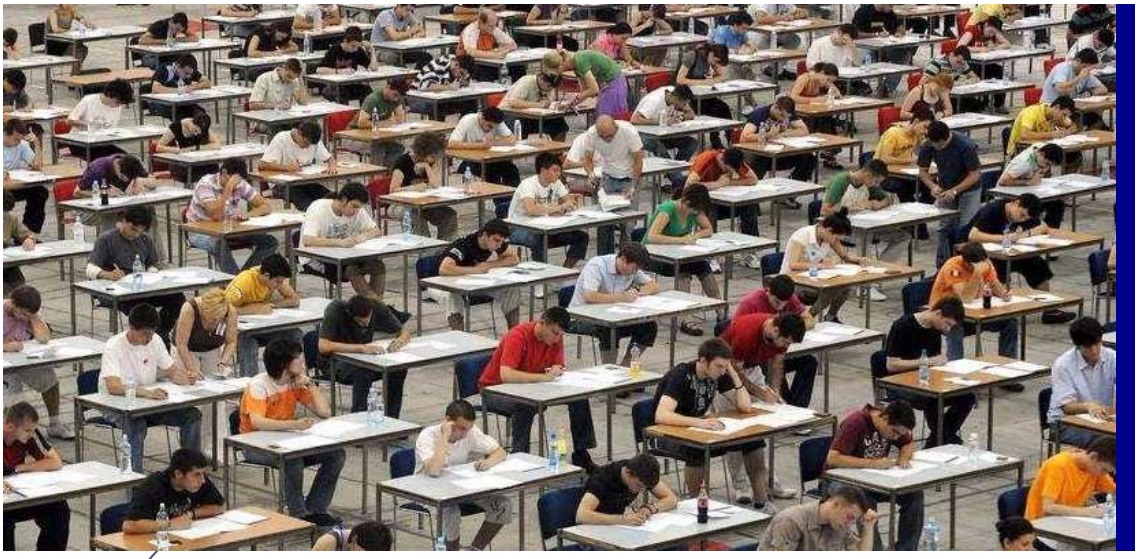
NYU TANDON SCHOOL OF ENGINEERING

18th Dec., 2021

Final Exam

CS-GY 6083 - B, FALL-2021.
Principles of Database Systems.



FINAL EXAM [100 points with 30% weight]

12/18/2020 10:00 AM to 12:30 PM EST (2 HRS 30 MINS)

Please read instructions carefully before writing exam

Write your name, student id, and net id below

- **Last Name: Li** **First Name: Minrui**
- **Net ID: ml7136** **Student ID: N17048569**

THIS IS AN ONLINE – OPEN BOOK EXAM. PLEASE LOGIN TO ZOOM MEETING USING YOUR NET ID (DO NOT LOGIN WITH YOUR PERSONAL EMAIL ACCOUNT). Find the Zoom meeting details under ZOOM top bar menu of the course website with title "Final Exam F2021". Join to the Zoom at least 15 minutes before the exam time.

- **WRITE YOUR ANSWERS UNDER EACH QUESTION IN THIS WORD DOCUMENT AND SUBMIT IT ON OR BEFORE 12:30PM TO Course site > ASSIGNMENTS > Final Exam. Save and submit the exam submission document in format <Your Net id>_Midterm_Fall_2021_6083B. YOU MAY RESUBMIT YOUR ASSIGMENT UNLIMITED TIME BEFORE THE SUBMISSION DEADLINE. The latest submission will be considered for the grading. PLEASE MUTE YOUR MICROPHONE DURING ENTIRE EXAM DURATION and KEEP VIDEO ON SINCE IT IS ONLINE EXAM.**

- **This exam has 5 sections A, B.C.D, and E. All sections and questions have grading points. There is NO negative points for any wrong answers.**

- **IF YOU HAVE ANY QUESTION DURING THE EXAM, PLESAE SEND YOUR QUESTION PRIVATELY TO PROFESSOR ON ZOOM MEETING CHAT WINDOW. DO NOT SPEAK IN MICROPHONE.**

- **USE Oracle Data Modeler for ERD diagrams, no hand drawing will be graded. Insert snapshot of database design models in same Word/PDF document. NO ZIP FILE WILL BE ACCEPTED. NO ANY TYPE OF COPY WILL BE GRADED.**

- **All Objects created in ERD should have your initial as prefix, e.g. AP_DEPT**

- **Keep practice schema (that we used for SQL tutorial) ready to use.**

GOOD LUCK!

A) Answer following questions briefly [20 points]

i) Disk performance and reliability are two major goals of RAID. State and explain methods that are used for achieving each of both goals.

To improve performance, we can use block striping, which is storing database blocks in multiple disks so that people might access to different blocks of data at the same time.

To improve reliability, we can use data mirroring, which is duplicate every disk. In this way, if one of the disks fails, people can get the same data from another one.

ii) What is Deadlock, give an example of the **deadlock situation**

Deadlock is two transactions are waiting for the other to unlock the resources.
Example: Suppose user A and B are reading the same resource at the same time, they both set write locks on the resource. Now they want to modify the resource, but neither A nor B can do it, because A is waiting for B to release the resource and B is waiting for A to release the resource. In this situation, we say A and B are in a deadlock.

iii) What is database constraint? Explain types of database constraints with example of each type in context of the below relational model of HRD database.

Database constraints are the rules enforced on the columns of table. There are lots of database constraints such as primary key constraint, foreign key constraint, index constraint, check constraint, not null constraint and etc.
For example: There is a primary key constraint on EMPNO in table ML_EMP, a foreign key constraint on DEPTNO in table ML_EMP.

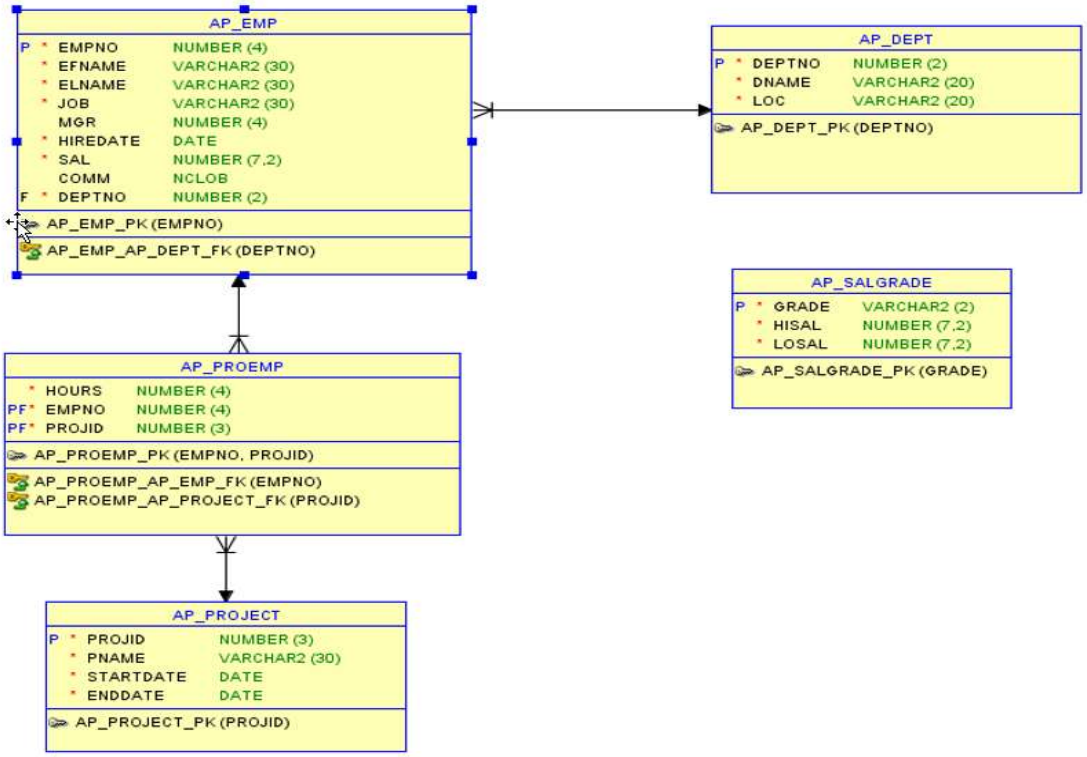
iv) Explain with example the difference between WHERE and HAVING clause of SQL statement in context of the below relational model of HRD database.

WHERE clause applies on the attributes in table, HAVING clause applies on the grouping results.

For example:
SELECT DEPTNO, AVE(SAL)
FROM ML_EMP
WHERE DEPTNO IN (10, 20, 30)
GROUP BY DEPTNO
HAVING AVE(SAL) >5000;

In the example above, WHERE clause applies on the DEPTNO, which in an attribute in table, whereas HAVING clause applies on AVE(SAL), which is a grouping result.

Consider following model of HRD database for Questions B, C, and D



RELATIONAL MODEL

B) For HRD database in relational model above, consider following transaction and answer the question [10 points]

```
CONNECT apatel/N3wY0rk21@HRD
UPDATE EMP SET COMM= COMM + SAL*0.08 WHERE DEPTNO=20;
DELETE FROM DEPT WEHRE DEPTNO=60;
ALTER TABLE DEPT MODIFY (LOC VARCHAR2 (30) );
DELETE FROM DEPT WEHRE DEPTNO=60;
ROLLBACK;
COMMIT;
```

i) For above set of database changes, list all transactions with transaction number e.g. TX1, TX2 etc., along with when each transaction starts and when it ends. [5 points]

Transaction Name	Started at Activity Number	Ended at Activity Number
TX1	CONNECT apatel/N3wY0rk21@HRD	ALTER TABLE DEPT MODIFY (LOC VARCHAR2 (30))
TX2	ALTER TABLE DEPT MODIFY (LOC VARCHAR2 (30))	ROLLBACK
TX3	ROLLBACK	COMMIT

ii) At the end of these transactions what changes will take place in HRD database and why? [5 points]

At the end of TX1, the COMM of people in table EMP whose DEPTNO is 20 was changed to COMM+SAL*0.08, and the department in table DEPT whose DEPTNO is 60 was deleted. TX2 started at the same time.

At the end of TX2, two queries were redo (ALTER TABLE DEPT MODIFY (LOC VARCHAR2 (30)); DELETE FROM DEPT WEHRE DEPTNO=60), which means the LOC in table DEPT were changed into VARCHAR(30), and DEPTNO=60 was deleted (Actually nothing was deleted here since it had been deleted at the previous transaction). At the same time, TX3 started

At the end of TX3, nothing happened, because there were nothing to commit.

C) For HRD database in relational model above, correct each of SQL [30 points]

Following are incorrect SQLs.

For each of these SQL, identify and list all mistakes and then write corrected SQL. State the purpose of the corrected query (what business question that query answers to)

i)

```
ALTER TABLE AP_EMP ADD CONSTRAINT ck_emp_job
CHECK (JOB =
('CLERK','ANALYST','SALESMAN','MANAGER','PRESIDENT'))
WHERE DEPTNO=20;
```

It should be:

```
ALTER TABLE AP_EMP ADD CONSTRAINT ck_emp_job
CHECK (JOB in ('CLERK','ANALYST','SALESMAN','MANAGER','PRESIDENT'))
WHERE DEPTNO=20;
```

Purpose: to make sure the value of JOB in table AP_EMP is 'CLERK' or 'ANALYST' or 'SALESMAN' or 'MANAGER' or 'PRESIDENT'.

ii)

```
SELECT JOB, DNAME, SUM(SAL) 'TOTAL SALARY'
FROM AP_EMP A JOIN AP_DEPT B ON
A.DEPTNO=B.DEPTNO WHERE DEPTNO<>50 AND
SUM(SAL)>=10000
ORDER BY DNAME;
```

It should be:

```
SELECT JOB, DNAME, SUM(SAL) 'TOTAL SALARY' FROM AP_EMP A JOIN
AP_DEPT B ON A.DEPTNO=B.DEPTNO
WHERE DEPTNO<>50
GROUP BY DNAME
HAVING SUM(SAL)>=1000
ORDER BY DNAME;
```

Purpose: select JOB, DNAME and TOTAL SALARY from table AP_EMP under the condition TOTAL SALARY >=1000, and order the result by DNAME.

iii)

```
SELECT EMPNO, EFNAME, DEPTNO, SAL
FROM AP_EMP WHERE SAL>=(SELECT DEPTNO, AVG(SAL) FROM
AP_EMP GROUP BY DEPTNO);
```

It should be:

```
SELECT EMPNO, EFNAME, DEPTNO, SAL
```

FROM AP_EMP WHERE SAL>=(SELECT AVG(SAL) FROM AP_EMP);

Purpose: list EMPNO, EFNAME, DEPTNO, SAL of a person whose salary is greater than the average salary from table AP_EMP

iv) **DELETE * FROM AP_PROEMP**
WHERE PROJID=100, HOURS<30;

It should be:
delete from ap_proemp
where projid=100 and hours<30;

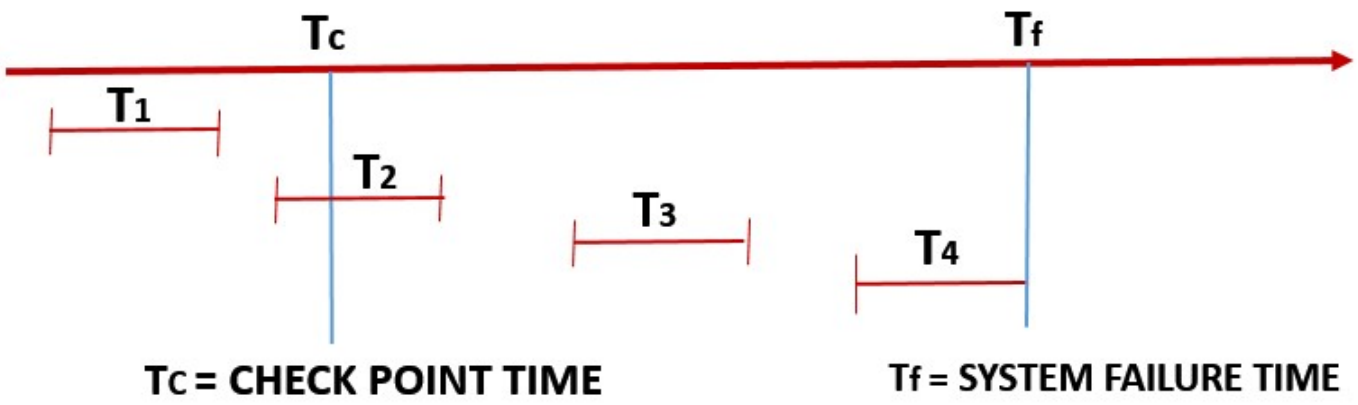
Purpose: Delete the record in table AP_PROEMP whose PROJID is 100 and HOURS is 30

v)
```
SELECT EMPNO, ENAME, DEPTNO, SAL
FROM AP_EMP
WHERE JOB= 'MANAGER'
ORDER BY SAL
INTERSECT
SELECT ENAME, EMPNO, DEPTNO, SAL
FROM AP_EMP
WHERE DEPTNO=20
AND COMM=NULL;
```

It should be:
```
SELECT ENAME, EMPNO, DEPTNO, SAL
FROM AP_EMP
WHERE JOB= 'MANAGER'
ORDER BY SAL
INTERSECT
SELECT ENAME, EMPNO, DEPTNO, SAL
FROM AP_EMP
WHERE DEPTNO=20 AND COMM=NULL;
```

Purpose: to intersect two tables, table A and table B. Table A is ENAME, EMPNO, DEPTNO, SAL whose JOB is ‘MANAGER’; Table B is ENAME, EMPNO, DEPTNO, SAL whose DEPTNO is 20 and COMM is NULL.

D) Consider following scenario of transactions in HRD database. [20 points]



Tc is the time when database has checkpoint.
Tf is the time when database has system failure
T1, T2, T3, and T4 are database transactions

Transactions T1, T2, T3, and T4 occurred in chronological order. The checkpoint in database happened at given time Tc and later on time at Tf, the system crashed on power failure.
Assume that, the employee DEVIN with employee id 7788 is working in department number 20 with monthly salary of $9000. DEVIN is currently not eligible to earn the commission.

Following is the details of work done in each transaction.

T1:

CONNECT apatel/N3wY0rk21@HRD

SELECT deptno, sal, comm FROM emp WHERE EMPNO='7788';

UPDATE emp SET sal=sal+500, comm=sal*0.1 WHRE EMPNO='7788'

CREATE TABLE EMP_TEST AS SELECT * FROM EMP WHERE 'Tc'='Tf';

T2:

DELETE FROM emp WHERE SAL BETWEEN 8000 and 9000;

ALTER TABLE EMP_TEST ADD CONSTRAINT

pk_emp_test PRIMARY KEY(EMPNO);

T3:

UPDATE emp SET deptno=30 WHERE empno='7788';

UPDATE emp SET sal=sal+sal*0.1 WHERE deptno=30;

COMMIT:

T4:

```
UPDATE emp
SET comm=nvl(comm,0)+100 , sal=sal+300
 WHERE empno=7788;
```

i) **Upon system recovery, which transaction(s) will undergo REDO operations and which transactions will undergo UNDO operations and why? [8 points]**

T1 will be ignored because it has been committed before check point time.
T2 and T3 will undergo REDO.
T4 will undergo UNDO because it has not been committed at system failure time.

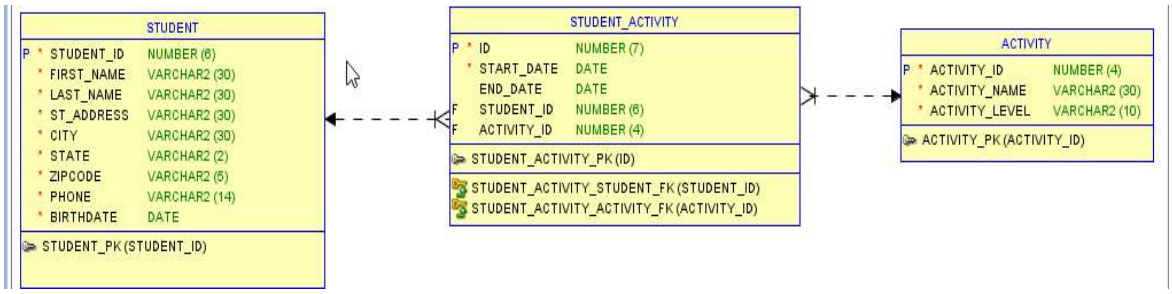
ii) **For transaction(s) that will undergo UNDO, what will be written out in transaction log? [8 points]**

Go backwards:
<T4, sal, 10450>, <T4, comm, 950>, <T4, abort>

iii) **What will the department number, salary, and commission of DEVIN after the system is recovered. [4 points]**

DEPTNO is 30, SAL is 10450 and COMM is 950.

E) Consider following relational model. [20 points]



Assume that, there are 50,000 students, 100 activities, 3 activity levels, and total 90,000 combination of students and activities.

The following query is frequently used by the department.

```
SELECT a.student_id, a.first_name, a.last_name,
          months_between(b.end_date. b..start_date), c.activity_name,
         c.activity_level
FROM student a JOIN student_activity b ON a. student_id=b.student_id
       JOIN activity c ON b.activity_id=c.activity_id
WHERE a.state in ('NY', 'NJ','CT') and c.activity_level='MODERATE' and
          INITCAP (activity_name)=' Piano Concert';
```

Answer following questions.

i) Which attributes are most suitable for index to improve the performance of this query?

STUDENT_ID in table STUDENT_ACTIVITY;
ACTIVITY_ID in table STUDENT_ACTIVITY;
STATE in table STUDENT;
ACTIVITY_LEVEL in table ACTIVITY;
ACTIVITY_NAME in table ACTIVITY.

ii) What type of index is appropriate for each column that you have intended to create index and why?

NON_UNIQUE INDEX: STUDENT_ID in table STUDENT_ACTIVITY; ACTIVITY_ID in table STUDENT_ACTIVITY; STATE in table STUDENT; ACTIVITY_NAME in table ACTIVITY.
BITMAP INDEX: ACTIVITY_LEVEL in table ACTIVITY;

iii) Write DDL command to create the index for each column that you have intended to create index.

CREATE INDEX idx_student_id ON STUDENT_ACTITITY (STUDENT_ID);

```
CREATE INDEX idx_activity_id ON STUDENT_ACTIVITY (ACTIVITY_ID);
CREATE INDEX idx_state ON STUDENT (STATE);
CREATE INDEX idx_activity_name ON ACTIVITY (AVTIVITY_NAME);
CREATE BITMAP INDEX idx_activity_level ON ACTIVITY (ACTIVITY_LEVEL);
```

iv) Which attributes in relational model has index created automatically, and what is the type of index they are.

STUDENT_ID in table STUDENT, ID in table STUDENT_ACTIVITY and ACTIVITY_ID in table ACYIVITY have index created automatically, because they are primary keys. The type is UNIQUE INDEX.