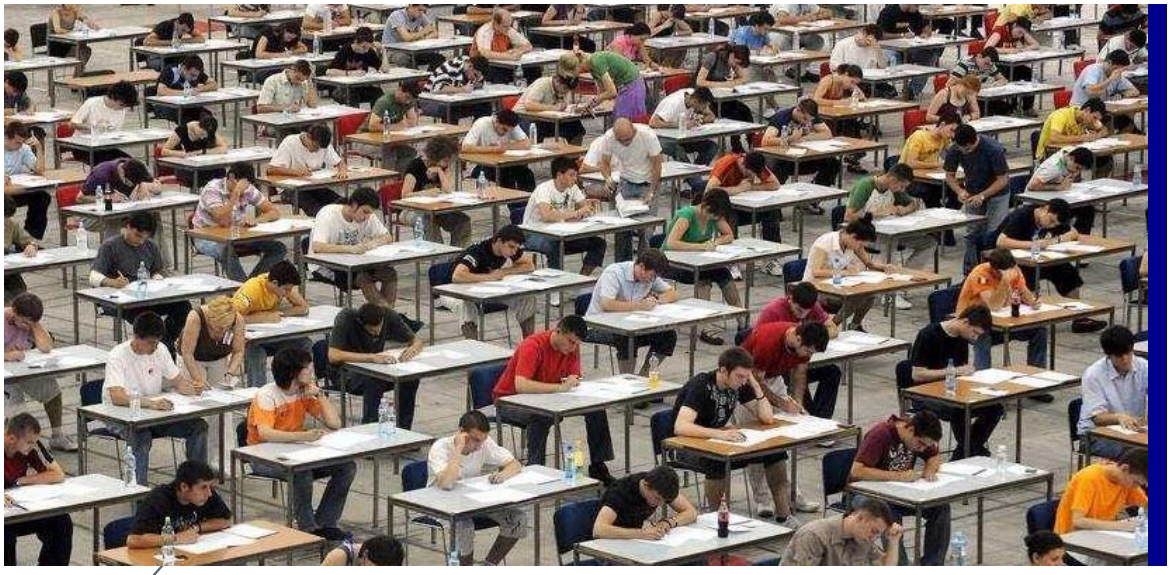
NYU TANDON SCHOOL OF ENGINEERING

19th Dec., 2020

CS-GY 6083 - B, FALL-2020.
Principles of Database Systems.

Final Exam



FINAL EXAM [100 points with 40% weight]

12/19/2020 10:00 AM to 12:15 PM EST (2 HRS 15 MINS)

Please read instructions carefully before writing exam

Write your name, student id, and net id below

- **Last Name:** **First Name:**
- **Net ID:** **Student ID:**

THIS IS AN ONLINE EXAM. PLEASE LOGIN TO ZOOM MEETING USING YOUR NET ID (MANDATORY, DO NOT LOGIN WITH YOUR PERSONAL ACCOUNT).

Join Zoom Meeting:

https://nyu.zoom.us/j/98424560126?pwd=OFhSL3gyNWZDalY1MnIxakpnWnNFUT09

Meeting ID: 984 2456 0126 Passcode: 381435

DO NOT COPY FROM ANY SOURCES INCLUDING BUT NOT LIMITED TO BOOK, CLASS MATERIALS, ONLINE RESOURCES, FROM OTHER STUDENTS.

- **WRITE YOUR ANSWERS UNDER EACH QUESTION IN THIS WORD DOCUMENT AND SUBMIT IT ON OR BEFORE 12:15PM TO NYU CLASSES > ASSIGNMENTS > FINAL_EXAM. Save and submit the exam submission document in format <Your Net id>_Final_Exam_FALL_2020_6083B..**
- **This exam has four sections A, B.C.D, E and each section has multiple questions. All sections and questions have grading points. There is NO negative points for any wrong answers. All sections and questions are expected to answer.**
- **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. FOR ANY REASON, YOU CAN NOT COMMUNICATE IN ZOOM CHAT. SEND EMAIL TO PROFESSOR WITH CC TO ALL TAs. PLEASE MUTE YOUR MICROPHONE DURING ENTIRE EXAM DURATION.**
- **If your Data Modeler software is giving you trouble during exam, you can hand draw diagram with all details expected in logical and relational model.**

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.

Achieving reliability via redundancy: Redundantly storing data to rebuild the information lost in disk failure. Mirroring is done to achieve redundancy; every disk is duplicated and all the writes are performed on original disk and the mirror disk simultaneously. In this way, even if the original disk fails, we will have the backup data on mirror disk.

Achieving performance via parallelism: Parallel structuring the disk will help in increasing the throughput by load balancing the access requests to disks. Parallelizing disk also reduce response time. Parnellism is achieved through block striping across available disks.

Different schemes of disk organization are used to achieve performance and reliability:

RAID level 0: It is used for high performance, no mirroring only block is stripped.
RAID level 1: Blocks are mirrored to achieve redundancy
RAID level 1 +0: Blocks are both mirrored and stripped
RAID level 2: Additional parity bit is stored for error detection and correction. Memory style error correction code with bit stripping.
RAID level 3: When data is written a parity, bit is calculated and stored in parity bit disk as a single parity bit is enough of error correction.
RAID level 4: Block Interleaved parity, a separate parity block is kept on separate disk for corresponding blocks from rest of the disks. On every write, block parity bits are calculated and written to parity disk.
RAID level 5: Block interleaved distribute parity, partitions data and parity are stored among all disks instead of storing parity on a specific parity disk.
RAID level 6: Same as raid level 5, but here instead of storing one parity block, and additional mirrored parity block information is redundantly stored to guard against multiple disk failure.

[Students are not required to explain each RAID levels. It is listed here just for students to review on possible solution]

.

ii) What is database trigger, function, and procedure and how they differ from each other?

Trigger: A trigger is PL/SQL code which contains a set of SQL statements in database which is automatically executed whenever any special event occurs specific to condition defined, such as before or after of insert, delete, update etc. Triggers are used to implement complex business rules that cannot be defined using constraints.

Function: A function is PL/SQL code that accepts input parameter, and produces output based upon SQL instructions and calculation performed as coded.

Procedure: A procedure is also PL/SQL code and set of pre-compiled queries which can

be used and shared among multiple programs. It can access and modify data in database as coded.

Trigger	Function	Procedure
Trigger is executed automatically on actions like update, delete, insert.	We can call a function whenever required but function can not be executed as it is not precompiled.	A procedure can be executed whenever we want.
Trigger can not be called from either function or procedure	Function can be called from trigger and procedure	Procedure can not be called from function. It can be called from trigger.
It never returns a value on execution.	It must return a value.	It may or may not return a value on execution

iii) What is Deadlock and state at least three possible methods to prevent dead lock.

Deadlock is a situation when two or more transactions have put a lock on a common shared resource and each one of them is waiting for others to release their locks first.

The three possible methods to prevent a dead lock are:

- Wait- die scheme – Rollback the younger transaction, older transaction may wait for the younger transaction to release the data item. Here older transaction means a transaction with smaller time stamp.
- Wound- wait scheme – Older transaction forces rollback of younger transaction, younger transaction may wait for older ones.
- Timeout-Based scheme – Break the deadlock on detection. Wait for the lock for a specified time and if lock is not granted then rollback the transaction and redo.

[Students may have another method, i.e. Locking all records that are intended to update in advance and thus putting exclusive lock on all such records and releasing them after updates are done. e.g. SELECT FOR UPADATE statement before actual UPDATE

```
SELECT * FROM EMP FOR UPDATE WHERE DEPTNO=10;
UDATE EMP
SET SAL=SAL+10
WHERE DEPTNO=10;
COMMIT;

VS.

UPDATE EMP
SET SAL=SAL+10;
COMMIT;
```

]

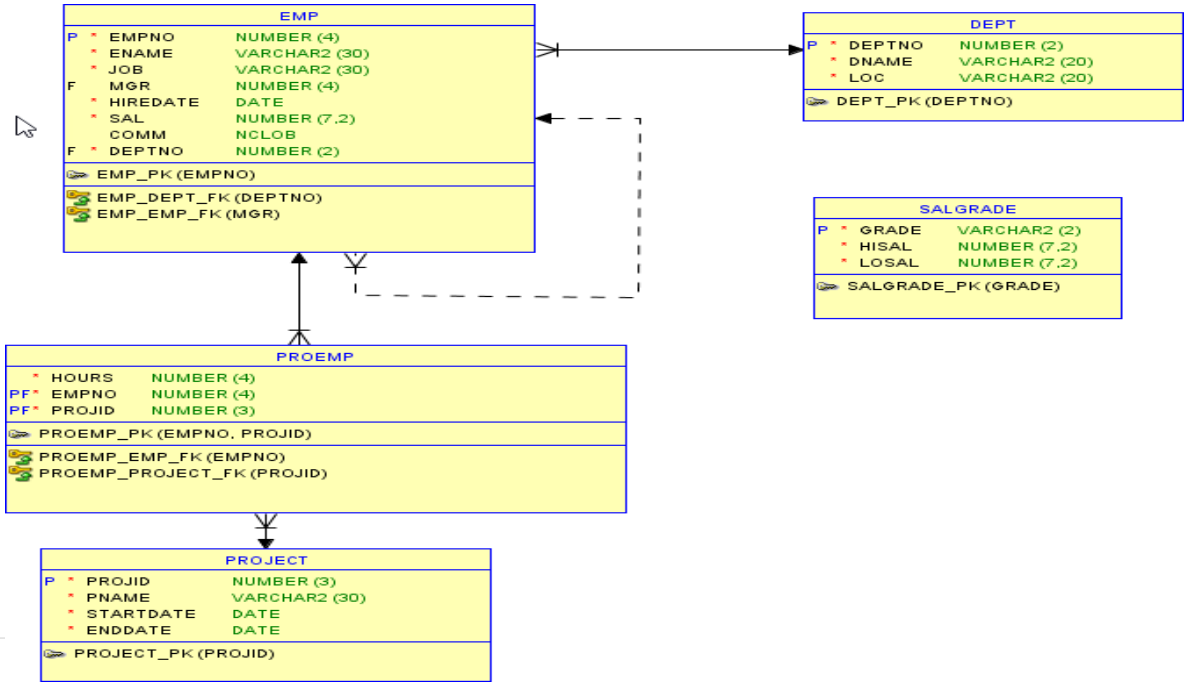
iv) Explain composite attribute, multivalued attribute, and derived attribute with example and how to resolve them in database design.

Composite Attribute: An attribute that contains multiple meaningful component/attribute in itself is called composite attribute.
Example: Address, it contains multiple components such as street address, city, state, country etc.
It can be resolved by breaking down the composite attributes into simple and atomic attributes.

Multivalued Attribute: An attribute that can take more than one value for a record.
Example: If skill is the attribute of STUDENTS table, students may have multiple skills, such as snorkeling, dancing, singing etc.
It can be resolved by creating a new table for skill, consisting of all the possible values the multivalued attribute can take, and then defining relationship between parent (SDTUENT) and child table(SKILL)

Derived Attribute: An attribute that can be calculated from other related attributes of database.
Example: Number of employment years of an employee working in a company.
To resolve it, have a column in table that requires to determine derived value and calculate the derived value when required and not store its value in database. In this example, have HIREDATE attribute in table and then calculate experience in years/ month/ weeks/ days etc. when running data analysis, or generating reports.

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



B) For HRD database, consider following transaction and answer the question [15 points]

```
CONNECT apatel/N3wY0rk20@HRD
UPDATE EMP SET COMM= COMM + SAL*0.05 WHERE DEPTNO=20;
ALTER TABLE DEPT MODIFY (LOC VARCHAR2(30) );
DELETE FROM DEPT WEHRE DEPTNO=50;
ROLLBACK;
```

At the end of this transaction what changes will take place in HRD database and why?

CONNECT apatel/N3wY0rk20@HRD

- Database connections will be established and transaction will start

UPDATE EMP SET COMM= COMM + SAL*0.05 WHERE DEPTNO=20;

- Commission of the employees whose DEPTNO is 20, is being increased by this factor -- COMM = COMM + SAL*0.05

ALTER TABLE DEPT MODIFY (LOC VARCHAR2(30));

- Datatype for the attribute LOC from table DEPT is being modified to Varchar2(30)

DELETE FROM DEPT WEHRE DEPTNO=50;

- Department record for DEPTNO =50 will be deleted.

ROLLBACK;

- Any uncommitted changes will be rollbacked

At the end of the transactions, following will happen.

a. Commission of the employees whose DEPTNO is 20, is being increased by this

b. Datatype for the attribute LOC from table DEPT is being modified to Varchar2(30)

c. If AUTO COMMIT IS OFF then, delete statement will be rollbacked and changes above a, b will be committed since DDL (ALTER TABLE in this problem set) is auto commit, and that commits all previously executed DML statements from the beginning of the transaction i.e. after DB connections was made in this problem set.

If AUTO COMMIT IS ON then, delete statement will be committed as well and there will be NO effect of ROLLBACK.

C) For HRD database, correct each of SQL [20 points}

Following are incorrect SQL. For each of these SQL, explain why it is incorrect and then write correct SQL

i) **ALTER TABLE EMP ADD CONSTRAINT ck_emp_job**
CHECK (JOB = ('CLERK','ANALYST','SALESMAN','MANAGER'))

IN operator should be used instead of comparative operator (=). Comparative operator can be used to compare only one data value. In this SQL statement, IN should be used since JOB can be any of "CLRK" or "ANALYST" or "SALESMAN" or "MANAGER".

Corrected SQL:

```
ALTER TABLE EMP ADD CONSTRAINT ck_emp_job
CHECK (JOB IN ('CLERK','ANALYST','SALESMAN','MANAGER'));
```

ii) **SELECT JOB, DNAME, SUM(SAL) "TOTAL SALARY"**
FROM EMP A JOIN DEPT B ON A.DEPTNO=B.DEPTNO
WHERE DEPTNO<>50 AND SUM(SAL)>=10000
ORDER BY DEPTNO;

Column names should be qualified with table alias, such as A.JOB, B.DNAME etc.

Any condition that filters result of group functions such as max, sum, etc., it requires to use HAVING clause.

Since group function SUM(SAL) is used in SELECT statement, any other columns used in SELEC clause must be referred in GROUP BY clause.

Corrected SQL:

```
SELECT B.DEPTNO,A.JOB, B.DNAME, SUM(A.SAL) as "TOTAL SALARY"
FROM EMP A JOIN DEPT B ON A.DEPTNO=B.DEPTNO
WHERE B.DEPTNO<>50
GROUP BY B.DNAME, B.DEPTNO, A.JOB
HAVING SUM(A.SAL)>=10000
ORDER BY B.DEPTNO;
```

iii) **SELECT EMPNO, ENAME, DEPTNO, SAL**
FROM EMP WHERE SAL>=(SELECT DEPTNO, AVG(SAL) FROM EMP
GROPUP BY DEPTNO));

GROUPUP BY is the typo, it should be GROUP BY
Inner query is returning more than one row, so comparative operators cannot be used. Instead it should use one of the operators ALL, IN, ANY etc.
Outer query has only one column SAL, so it cannot compare two column DEPTNO and AVG(SAL)

There is an extra closing bracket at the end.

Corrected SQL:

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

iv) **SELECT EMPNO, ENAME, DEPTNO, SAL**

FROM EMP
WHERE JOB=’ANALYST’
ORDER BY SAL
INTERSECT
SELECT ENAME, EMPNO, DEPTNO, SAL
FROM EMP
WHERE DEPTNO=20
ORDER BY SAL;

The ORDER BY statement should only be used at the end of the entire query.

ENAME, EMPNO should be swapped in fifth line to be consistent with the first query else we’ll get an error, meaning that datatype of columns must match.

Corrected SQL:

```
SELECT EMPNO, ENAME, DEPTNO, SAL
FROM AP_EMP
WHERE JOB='ANALYST'
INTERSECT
SELECT EMPNO, ENAME, DEPTNO, SAL
FROM AP_EMP
WHERE DEPTNO=20
ORDER BY SAL;
```

D) For HRD database, answer questions [25 points}

For HRD database assume following

- **EMP table has 10,000 total number of records**
- **EMPNO is the primary key**
- **There are 6 distinct jobs**
- **There are 10 distinct departments**
- **Salary is in range of $4000 and $15000 USD**

With above mentioned assumptions answer following questions.
[Questions I, ii, and iii has 5 points, question iv has 10 points]

i) Upon creation of EMP table, which column will have index automatically created? What is the type of that index? Write a DDL command that would have used for creating that index

Primary key EMPNO will be automatically indexed. This index will be unique index (primary index or cluster index and of type dense).

```
CREATE UNIQUE INDEX idx_emp_empno ON EMP (EMPNO);
```

ii) What are the candidate columns for creating bitmap index and why? Write DDL command to create such bitmap indexes.

Columns JOB and DEPTNO can be used to create bitmap index, because they have low cardinality. (small number of distinct values)

```
CREATE BITMAP INDEX emp_jobs_bitmap ON EMP(JOB);
CREATE BITMAP INDEX emp_deptno_bitmap ON EMP(DEPTNO);
```

iii) If you create an index on SAL column, what type of the index it will be? Write DDL command to create index on SAL column.

It would be secondary index (of type dense).

```
CREATE INDEX idx_emp_sal ON EMP (SAL);
```

iv) For following SQL command, provide at least two possible ways the query execution plan will be created. Explain your answer with which indexes will be used or not used in possible execution plan. Suggest if any additional index you may want to create to further optimize the execution plan.

SELECT EMPNO, ENAME, HIREDATE, SAL, JOB, DEPTNO
FROM EMP
WHERE SAL>=3000 AND (JOB='ANALYST' or DEPTNO=30)
ORDER BY HIREDATE;

P.S.: While determining alternative execution plans you may consider indexes created in D.I, D.II, and D.III or discard any of those indexes based upon assumptions stated.

Plan 1.

a. Use bitmap index on JOB to find rows with JOB = 'ANALYST'
b. Use bitmap index on DEPTNO to find rows with DEPTNO = 30
c. Create UNION of bitmaps from result of a and b above
d. Use index on SAL filter result of c above for SAL>= 3000 (this step will be in plan only if data statistics are not correct or up to date, since as stated in case all employees have salary in range of $4000 to $15000, so as per case stated, all employees qualifies for this filter, so need no use index on SAL
e. Use index on HIREDATE to order results.

Plan 2:

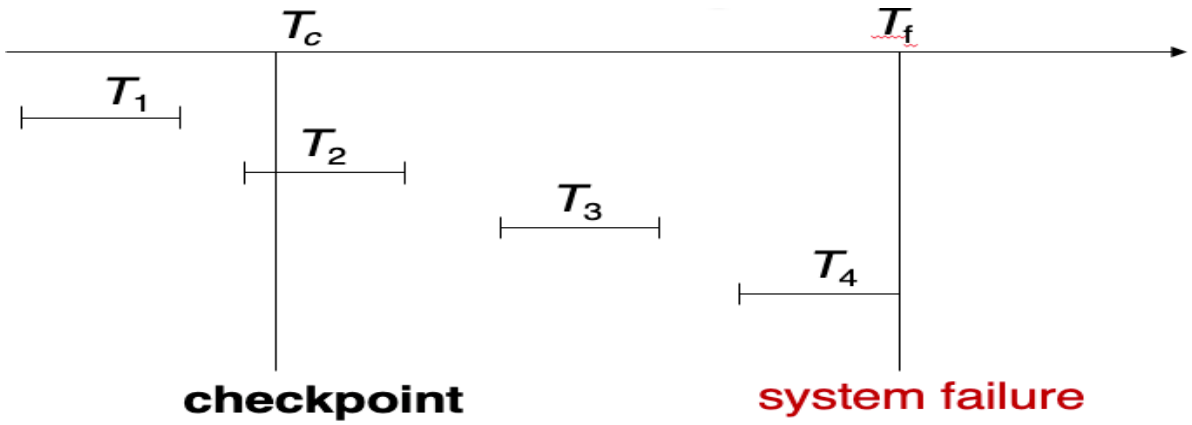
a. Use bitmap index on JOB to find rows with JOB = 'ANALYST'
b. Use bitmap index on DEPTNO to find rows with DEPTNO = 30
c. Create UNION of bitmaps from result of a and b above
d. Do not use index on SAL, and Ignore condition SAL >= 3000(From data statistics, optimizer knows minimum salary is greater than 3000)
e. Use index on HIREDATE to sort results

Plan 3: Same as Plan 1 without using HIREDATE index, if there is no index on HIREDATE
Plan 4: Same as Plan 2 without using HIREDATE index, if there is no index on HIREDATE
Plan 5: Same as plan 1, with following order changes

a. Use index on SAL first, and find qualified records (in that case all records)
b. Use bitmap index on JOB to find rows with JOB = 'ANALYST'
c. Use bitmap index on DEPTNO to find rows with DEPTNO = 30
d. Create UNION of bitmaps from result of b and c above
e. Use index on HIREDATE to order results.

Plan 6: Same as plan 5, without using HIREDATE index, if there is no index on HIREDATE

E) Consider following scenario of transactions for a problem set. [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 Tf, the system crashed on power failure. Following is the details of work done in each transaction.

Transaction T1	Transaction T2	Transaction T3	Transaction T4
< T1 start>	< T2 start>	< T3 start>	< T4 start>
< T1, X, 5000, 3000 >	< T2, X, 3000, 2500 >	< T3, A, 1800, 1200 >	< T4, X, 3000, 4000 >
< T1, Y, 3000, 2000 >	< T2, Y, 2000, 1000 >	< T3, B, 800, 600>	< T4, Y, 2000, 2500 >
< T1, Z, 1000, NULL >	< T2, Z, NULL, 500 >	< T3, C, 250 , NULL >	< T4, Z, NULL, 700 >
<T1, COMMIT >	<T2, ROLLBACK>	<T3, COMMIT >	< T4, A , 1200, 800 >
			< T4, B, 600, 1200>

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

T2 and T3 will undergo REDO because they are committed between checkpoint and failure.
T4 will undergo UNDO because it has not finished before the failure.

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

Undo T4 :

<T4, B, 600>
<T4, A,1200>
<T4, Z, NULL>
<T4, Y, 2000>
<T4, X, 3000>
<T4, abort>

The order should be the same as above

iii) What will be data values of data items X, Y, Z, A, B, and C after the system is recovered. [4 points]

X : 3000, Y : 2000, Z : NULL, A : 1200, B : 600, C : NULL