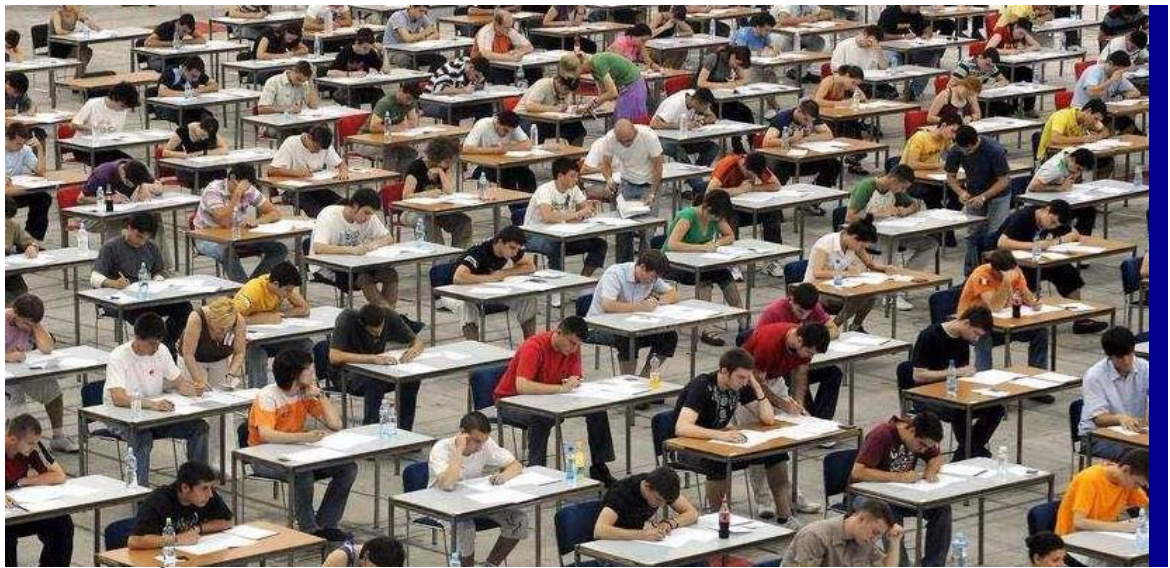


**NYU****TANDON SCHOOL  
OF ENGINEERING****Final Exam**

**16th Dec. B, FALL-  
2023.  
Principles of Database  
Systems.**



**FINAL EXAM [100 points with 20% weight]**  
**Saturday, 12/16/2023 11:00 AM to 1:30 PM EST (2 ½ HRS.)**

**Please read the instructions carefully before writing the exam**

Write your name, student ID, and net ID below.

? Last Name:

First Name:

? Net ID:

Student ID:

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

- **Write your answers under each question in this Word document and submit it on or before 1:30 pm to course site > assignments > Final Exam. Save and submit the exam submission document in format <your net id>\_final\_fall\_2023\_6083b.pdf. **Unlimited submission is allowed before the deadline. The latest submission will be considered for grading.** Please mute your microphone during the entire exam duration and keep video on since it is an online exam.**
- **This exam has 5 sections A, B, C, D, and E. There are no negative points for any wrong answers.**
- **If you have any questions during the exam, please send your question privately to the course assistant on zoom meeting chat window. Do not speak in the microphone.**
- **Please keep saving your exam document frequently.**
- **Please keep ready the practice schema (that we used for the SQL tutorial) with all tables with your initial.**

**GOOD LUCK!**

**A) Answer the following questions briefly [10 points]**

- i) **Explain the difference between Data, Metadata, and Information with an example of each.**

**Ans:**

**Data:** Data is plain facts concerning meaningful objects (tables) in a database. In an e-commerce database, data could include product names, prices, customer IDs, order numbers, and quantities.

**Metadata:** Metadata is the description and context of user data. It helps organize, find, and understand data. It is data about data, including data type, size, optional/mandatory, constraints, and description. For example, in a library database, metadata for the "Book Title" column might include data type (e.g., varchar), size (e.g., 30 characters), and constraints (e.g., cannot be null).

**Information:** Information is processed, interpreted, organized, and structured to derive meaningful and actionable insight that can help in making decisions. Example: From a sales database, information could be the total revenue for each product category over a specific time.

- ii) **Explain the difference between Delete, Truncate, and Drop SQL statements.**

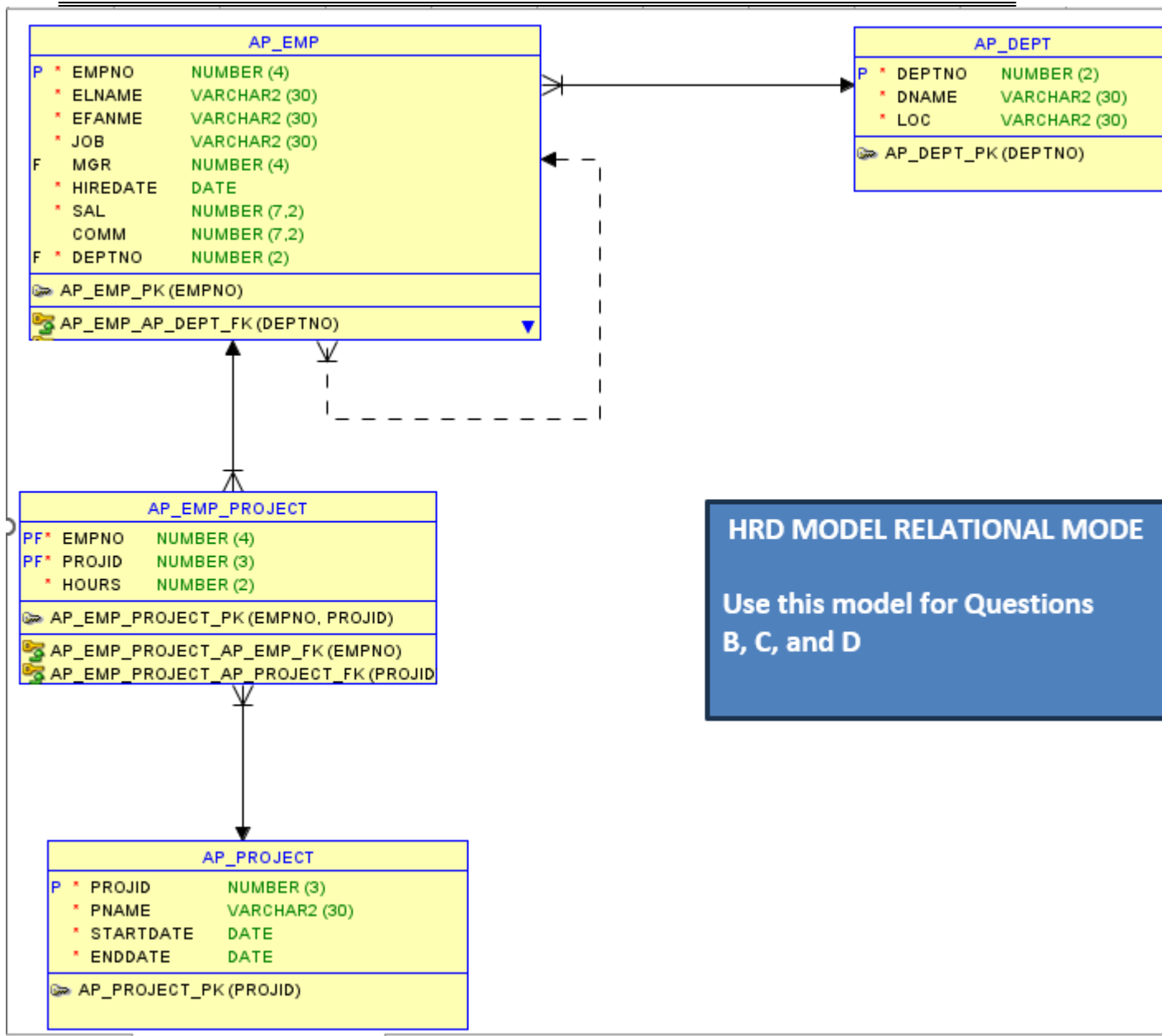
**Ans:**

**Delete:** The DELETE is a data manipulation command. It can be rolled back. We can delete one or more rows of the table using it. We can add conditions in the delete statement to delete only the rows we want to delete. For example, DELETE from table WHERE x = 123.

**Truncate:** Truncate is a Data Definition Language Command. It cannot be rolled back. The TRUNCATE statement will delete all the data (rows) of the table but not delete the table. The columns, indexes, and constraints are not deleted.

**Drop:** DROP is also a DDL (Data Definition Language) command. It also cannot be rolled back. And it is used to drop

the entire table, which includes rows, columns, indexes, constraints, etc. The table ceases to exist once dropped.



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

1. **CONNECT** apatel/Fa1I2021@HRD
2. **UPDATE** AP\_EMP SET COMM= COMM + SAL\*0.12 WHERE DEPTNO=20;
3. **DELETE** \* FROM AP\_DEPT WHERE DEPTNO=40;
4. **ALTER TABLE** AP\_DEPT MODIFY (DNAME VARCHAR2 (40) );
5. **ROLLBACK**;
6. **CREATE INDEX** index\_ename ON ap\_emp(substr(ename,-2,2));
7. **COMMIT**;
8. **ROLLBACK**;

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

Transaction Name	Started at Activity Number	Ended at Activity Number
TX1	1	4
TX2	4	5
TX3	5	6
TX4	6	7
TX5	7	8

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

At the end of these transactions, employees who are eligible for commission (NOT NULL) in department number 20 will have 12% salary added to their commission.

The DNAME attribute of DEPT table will have size of VARCHAR modified to 40.

Employees of department number 40 will be removed.

An function based index will be created on substr(ename,-2,2)

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

The 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)**

- i. **ALTER TABLE AP\_DEPT ADD CONSTRAINT ck\_dept\_loc  
CHECK (LOC = ('NEW YORK','CHICAGO','BOSTON'))  
WHERE DEPTNO=10 OR 20 OR 30 ;**

we cannot use "=" operator for a list of values. We need to use the IN operator instead. Also, where clause cannot be used with alter query.

Correct query:

**ALTER TABLE AP\_DEPT ADD CONSTRAINT ck\_dept\_loc  
CHECK (LOC IN ('NEW YORK','CHICAGO','BOSTON'));**

Adds a constraint to check that the LOC column in the AP\_DEPT only contains these values 'NEW YORK', 'CHICAGO', or 'BOSTON'

- ii. **SELECT JOB, DNAME, AVG(SAL) 'AVERAGE SALARY'  
FROM AP\_EMP A JOIN AP\_DEPT B  
A.DEPTNO=B.DEPTNO  
WHERE DEPTNO<>50 AND AVG(SAL)>=2000  
ORDER BY 4;**

To use a column alias, we need to put it in double quotes(""), instead of single quotes. To specify the column that should be used to join tables we use the ON keyword Since avg(sal) is an aggregate function, we need to group by columns in the select statement.

Finally, there is no fourth column in the table, so we should use "AVERAGE SALARY"

**SELECT JOB, DNAME, AVG(SAL) AS "AVERAGE SALARY"  
FROM AP\_EMP A JOIN AP\_DEPT B  
ON A.DEPTNO = B.DEPTNO  
WHERE A.DEPTNO <> 50  
GROUP BY JOB, DNAME  
HAVING AVG(SAL) >= 2000  
ORDER BY "AVERAGE SALARY";**

This query gives us the job, department name, and avg salary of each job within a department for all employees that are not in department number 50, and avg salary is greater than 2000.

- iii. **SELECT EMPNO, EFNAME, DEPTNO, SAL  
FROM AP\_EMP WHERE SAL>=(SELECT DEPTNO, MIN(SAL) FROM  
AP\_EMP GROUP BY DEPTNO);**

Here, the subquery returns multiple columns and multiple columns cannot be used for comparison

**SELECT EMPNO, EFNAME, DEPTNO, SAL  
FROM AP\_EMP e1 WHERE SAL>=(SELECT MIN(SAL) from AP\_EMP  
e2 where e1.DEPTNO = e2.DEPTNO);**

This query selects details for all employees in a department whose salary is greater than or equal to the minimum salary for the department.

- iv. **DELETE \* FROM AP\_PROEMP  
WHERE PROJID=100, HOURS<30;**

Syntax for delete query is wrong, should not use \*. Also, we need to use AND in the where clause instead of comma(,).

**DELETE FROM AP\_PROEMP  
WHERE PROJID=100 AND HOURS<30;**

this query deletes records from the intersect table between project and employee that have project id 100 and hours less than 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;**

The order and data types of the columns should be the same on both sides of the intersect query. Also, to check if COMM is equal to null, we should use IS NULL instead of = null. Order By SAL should be after the second select query.

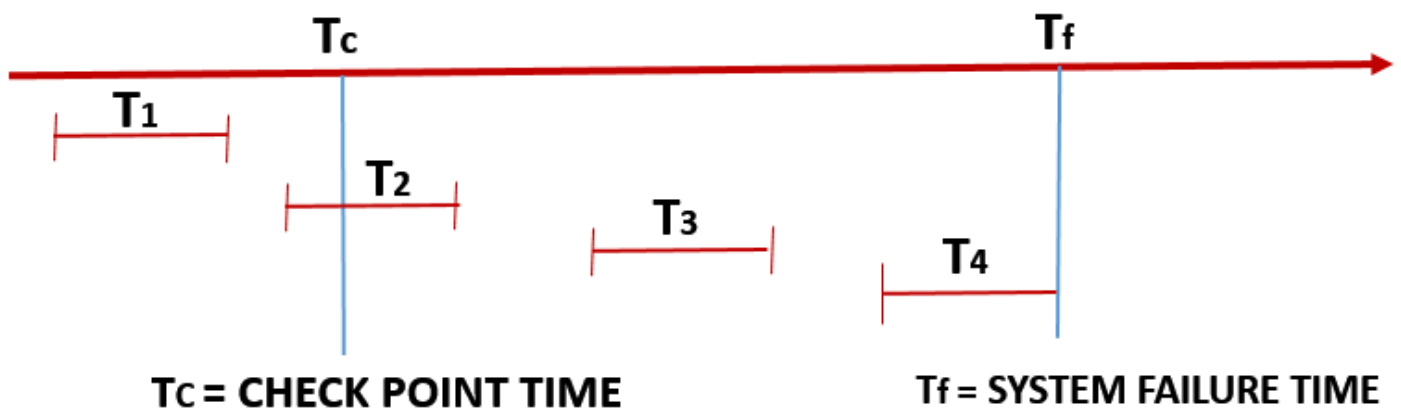
```
SELECT EMPNO, ENAME, DEPTNO,  
SAL FROM AP_EMP  
WHERE JOB= 'MANAGER'
```

**INTERSECT**

```
SELECT EMPNO, ENAME, DEPTNO, SAL  
FROM AP_EMP  
WHERE DEPTNO=20  
AND  
COMM IS NULL  
ORDER BY SAL;
```

The purpose of this query is to find employees who are both managers, belong to department number 20 and do not have any commission. By using INTERSECT, we only get those records which are present in both queries.

**D) Consider the following scenario of transactions in the HRD database. [24 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 the database happened at a given time Tc and later on time at Tf, the system crashed due to power failure.**



Assume that the employee MELVIN with employee ID 7799 is working in department number 20 with a monthly salary of \$7000. MELVIN is currently not eligible to earn the commission.

The following are the details of the work done in each transaction.

**T1:**

```
CONNECT apatel/N3wY0rk23@HRD
```

```
SELECT deptno, sal, comm FROM ap_emp WHERE EMPNO='7799';
```

```
UPDATE ap_emp SET sal=sal+500, comm=sal*0.1 WHERE EMPNO='7799';
```

```
CREATE TABLE AP_EMP_TEST AS SELECT * FROM EMP WHERE "T1"="TX 1";
```

**T2:**

```
DELETE FROM ap_emp WHERE SAL BETWEEN 8000 and 9000;
```

```
ALTER TABLE EMP_TEST ADD CONSTRAINT  
pk_emp_test PRIMARY KEY(EMPNO);
```

**T3:**

```
UPDATE ap_emp SET deptno=30 WHERE empno='7799';
```

```
UPDATE ap_emp SET sal=sal+sal*0.1 WHERE deptno=30;
```

```
COMMIT;
```

**T4:**

```
UPDATE ap_emp  
SET comm=nvl(comm,0)+100 , sal=sal+300  
WHERE empno=7799;
```

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

**Ans.** Since T2 began but did not finish at check point Tc, REDO operations will be performed on T2 and T3 transactions upon system recovery. T3 began following the checkpoint as well and finished before the system crashed at Tf, T3 will also undergo REDO. Since transaction T4 began after checkpoint Tc but was never finished, it will undergo a UNDO operation.

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

Ans. <T4, comm, 750> <T4, sal, 8250> <T4, abort>

iii) What will the department number, salary, and commission of MELVIN be after the system is recovered? [8 points]

Ans. After the system is recovered, Melvin's salary will be \$8250, department no. 30. And the commission of Melvin will be \$750.

E) Consider the following dataset. [26 points]

GENDER	AGE	INCOME	CHURN
Female	<50	<100K	NO
Female	>50	>100K	YES
Male	<50	<100K	YES
Male	<50	<100K	YES
Male	>50	<100K	YES
Male	>50	>100K	NO
Male	<50	<100K	YES
Female	<50	<100K	NO
Female	<50	<100K	NO
Male	<50	<100K	YES
Female	<50	<100K	NO
Female	>50	<100K	YES
Male	>50	>100K	NO

This dataset is about employee churn (decision to leave the company) based on gender, age, and income criteria.

a) Draw bitmaps for the attributes GENDER, AGE, INCOME, and CHURN. Submit each bitmap. [8 points]

ANS.

1. Gender: -

- Female 1100000110110  
- Male 0011111001001

2. Age: -

- >50 0100110000011  
- <50 1011001111100

3. Income: -

- >100K 0100010000001

- <100K      1011101111110

**4. Churn: -**

- Yes      0111101001010

- No      1000010110101

**b) Using the bitmaps created in step a above, answer the following questions by resolving the bitmaps.**

**Total number of male employees of the age less than 50 and with salary less than 100K who left (CHURN) the company.**

**Submit intermediate and final bitmaps, and a total number of employees as per given criteria. [8 points]**

**Ans.** Considering, Male employees, age >50 and <100K, left the company (Yes).

**Bitmaps for these values –**

- Male      0011111001001

- <50      1011001111100

- <100K      1011101111110

- Yes      0111101001010

So, the Total number of male employees of the age less than 50 and with salary less than 100K who left (CHURN) the company, will be solved by the intersection of these bitmap values.

- Male AND <50 AND <100K AND <Yes

Male AND <50

```
      0011111001001
AND  1011001111100
=    0011001001000
```

Male AND <50 AND <100

```
      0011001001000
AND  1011101111110
=    0011001001000
```

Male AND <50 AND <100K AND <Yes

```
      0011001001000
AND  0111101001010
=    0011001001000
```

4 bits are set to 1, hence there are 4 males of the age <50 and with salary <100K who left the company.

**c) Use the attached DDL and DML to create a table and data by replacing asp13 with your Net ID. Write a SQL query that finds**

the answer to the same question (Total number of male employees of the age less than 50 and with salary less than 100K who left (CHURN) the company).

**Submit the SQL and screenshot of the result. [8 points]**

**Ans.** SELECT COUNT(\*) AS EMP  
FROM asp13\_emp\_churn  
WHERE GENDER = 'Male'  
AND AGE = '<50'  
AND INCOME = '<100K'  
AND CHURN = 'YES'

The screenshot shows an SQL Worksheet interface. On the left is a sidebar with navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a SQL query:   
1 /\* Male Employees, age <50, sal <100K, left the company \*/  
2  
3  
4 SELECT COUNT(\*) AS EMP  
5 FROM asp13\_emp\_churn  
6 WHERE GENDER = 'Male'  
7 AND AGE = '<50'  
8 AND INCOME = '<100K'  
9 AND CHURN = 'YES'  
Below the query editor, the result is displayed in a table with one column 'EMP' and one row containing the value '4'. A 'Download CSV' button is located below the table. At the bottom of the interface, there is a footer with text: '2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf · 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym' and 'Built with ❤️ using Oracle APEX · Privacy · Terms of Use'.

**d) Write DDL code to create bitmap indexes for each attribute.**

**Submit the DDL code. [2 points]**

**ANS.**

```
CREATE BITMAP INDEX idx_gender_role ON asp13_emp_churn(GENDER);  
CREATE BITMAP INDEX idx_age_range ON asp13_emp_churn(AGE);  
CREATE BITMAP INDEX idx_income_level ON asp13_emp_churn(INCOME);  
CREATE BITMAP INDEX idx_churn_status ON asp13_emp_churn(CHURN);
```