

16th Dec. 2023

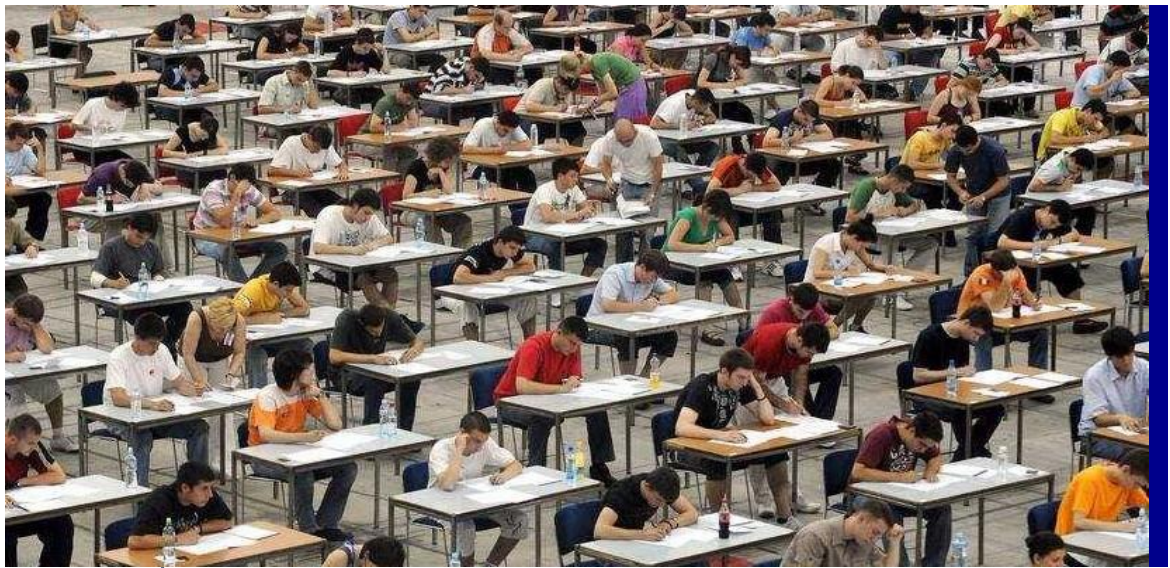
Final Exam



NYU

TANDON SCHOOL
OF ENGINEERING

CS-GY 6083 - 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: Jie
- Net ID: jc12300

First Name: Cheng
Student ID: N14410995

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.

Data refers to raw, unprocessed facts and figures. It consists of symbols, characters, or quantities that can be processed to extract meaningful information. For example, consider a set of numbers: 5, 10, 15, 20. These individual values are data. Without context or interpretation, they are just raw pieces of information.

Metadata provides information about data. It describes the characteristics, context, and structure of data, making it easier to manage, understand, and use. Metadata is essentially data about data. For example, for the set of numbers mentioned earlier (5, 10, 15, 20), metadata could include details such as the data type (integer), the range of values, the date the data was recorded, and any other relevant information that helps provide context or meaning to the raw data.

Information is the result of processing and interpreting data. It is meaningful, organized, and contextualized data that can be used for decision-making or understanding. For example, if we calculate the average of the set of numbers (5, 10, 15, 20), the result (12.5) is information. It provides a meaningful insight derived from the raw data. In this case, the average gives us an understanding of the central tendency of the values.

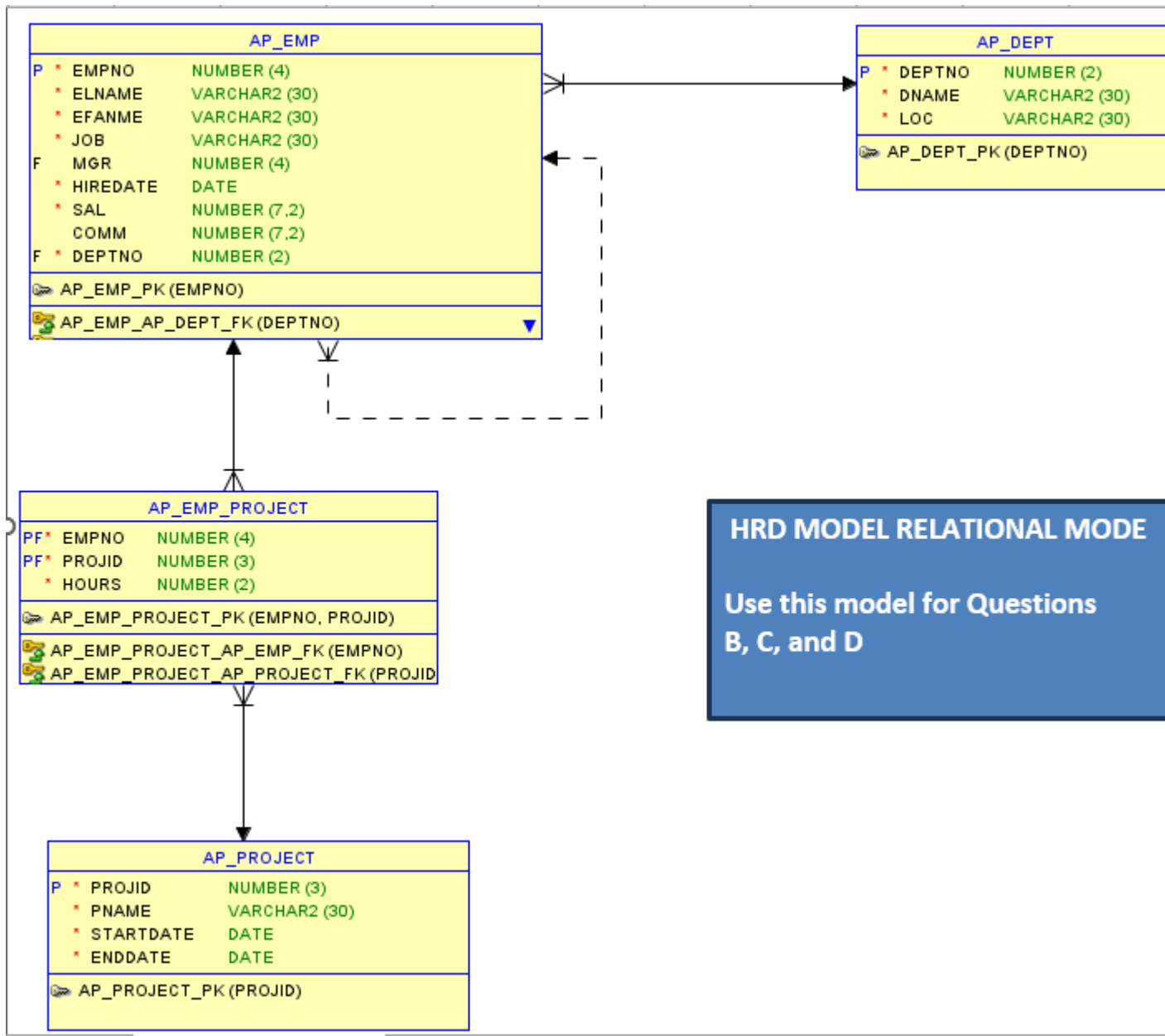
ii) Explain the difference between Delete, Truncate, and Drop SQL statements.

The DELETE statement is used to remove rows from a table based on a specified condition. You can use the DELETE statement to selectively remove specific rows from a table that meet a certain criterion. The DELETE statement is transactional, meaning you can roll back the changes if needed. It also generates individual row delete operations, which can be resource-intensive for large datasets.

The TRUNCATE statement is used to remove all rows from a table quickly. It is a more efficient operation than DELETE for removing all records from a table. TRUNCATE removes all the rows from a table, but it does not log individual row deletions, making it faster than DELETE. TRUNCATE is not transactional (in most databases), and it cannot be rolled back. It is also less flexible than DELETE because it does not allow a WHERE clause to specify conditions.

The DROP statement is used to remove database objects such as tables, views, indexes, or even entire databases. DROP is used to delete the entire structure and data associated with the specified object. For example, you can drop a table to remove both the table and its data. The DROP statement is a powerful and

irreversible operation. Once an object is dropped, it cannot be recovered.



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

1. **CONNECT** apatel/Fa1l2021@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	7
TX4	7	8
TX5		

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, the HRD database will reflect the changes made by the committed transactions. The AP_EMP table will have updated COMM values for employees in department 20. The AP_DEPT table will have no rows where DEPTNO is 40. The AP_DEPT table will have a modified data type for the DNAME column. The AP_EMP table will have a new index (index_ename) on the last two characters of the ename column. The changes made by the rolled back transactions will not be reflected in the database since the rollbacks undo those changes.

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;**

Mistakes:

The **CHECK** constraint syntax is incorrect and the **WHERE** clause is not applied correctly. As we are using ALTER TABLE, we cannot use the WHERE clause.

CORRECT SQL:

**ALTER TABLE AP_DEPT ADD CONSTRAINT ck_dept_loc
CHECK (LOC in ('NEW YORK','CHICAGO','BOSTON'));**

Purpose: Apply a constraint such that the location has one of the values as 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;**

Mistakes:

AVERAGE SALARY should be in double quote instead of single quote. DEPTNO<>50 is incorrect. It should be DEPTNO IS NOT NULL. Null values cannot be compared with comparison operator. There should be an ON after joining two tables. AVG(SAL)>=300 cannot be used in the where clause, it should be used in the HAVING clause. The ORDER BY clause should reference the column position or alias, not the column expression.

CORRECT SQL:

**SELECT JOB, DNAME, AVG(SAL) "AVERAGE SALARY"
FROM AP_EMP A JOIN AP_DEPT B ON
A.DEPTNO = B.DEPTNO
WHERE DEPTNO IS NOT NULL
GROUP BY JOB, DNAME
HAVING AVG(SAL) >= 2000
ORDER BY DNAME;**

Purpose:

Retrieve the average salary for each job and department combination in the **AP_EMP** table, excluding department 50 and those whose average salary is

greater than or equal to 2000.

- iii) **SELECT EMPNO, EFNAME, DEPTNO, SAL
FROM AP_EMP WHERE SAL>=(SELECT DEPTNO, MIN(SAL) FROM
AP_EMP GROUP BY DEPTNO);**

Mistakes:

It should have ANY/ANY/IN operator in where clause as the single-row subquery is returning two columns. So, it should only have MIN(SAL).

CORRECT SQL:

**SELECT EMPNO, EFNAME, DEPTNO, SAL
FROM AP_EMP
WHERE SAL >= ANY (SELECT MIN(SAL) FROM AP_EMP GROUP BY DEPTNO);**

Purpose:

Retrieve employee number, first name, department number, and salary for those whose salary is greater than or equal to the minimum salary in their respective departments.

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

Mistakes:

The * after DELETE is unnecessary. Multiple conditions in the WHERE clause should be connected using AND or OR.

CORRECT SQL:

**DELETE FROM AP_PROEMP
WHERE PROJID = 100 AND HOURS < 30;**

Purpose:

Delete records from the **AP_PROEMP** table where the project ID is 100 and the hours are 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;

Mistakes:

The position of Order by clause is incorrect. It should be used as the last clause.
COMM=NULL is incorrect. It should be COMM IS NULL.

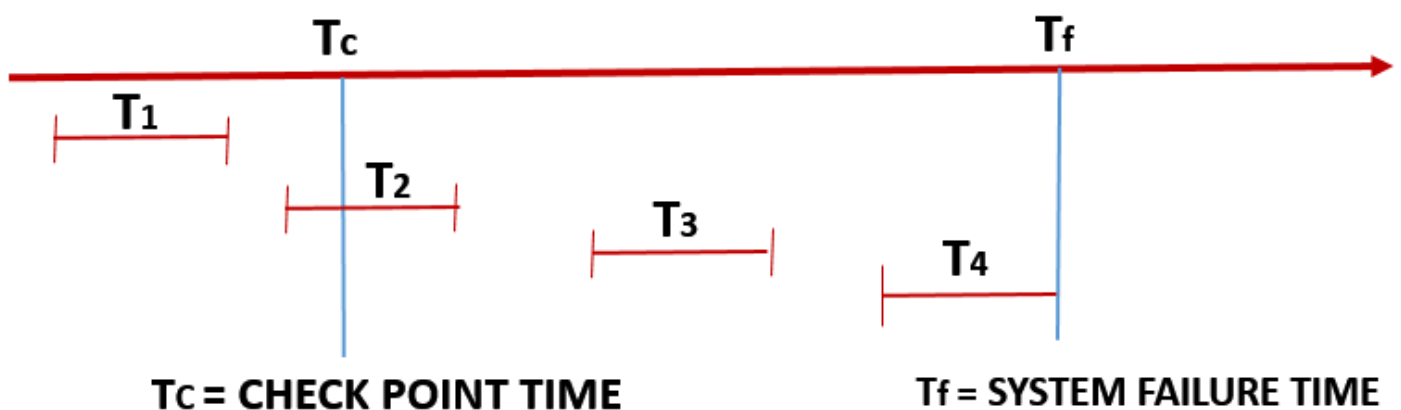
CORRECT SQL:

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

Purpose:

Retrieve the employee number, employee name, department number, and salary of employees who are managers and have a NULL value in the COMM column, with additional filtering for employees in department 20.

D) Consider the following scenario of transactions in the HRD database. [24 points]



T_c is the time when database has checkpoint.
 T_f is the time when database has system failure
 T_1 , T_2 , T_3 , and T_4 are database transactions.

Transactions T1, T2, T3, and T4 occurred in chronological order. The checkpoint in the database happened at a given time T_c and later on time at T_f, 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]

Following system recovery, the REDO operation will be applied to transactions T2 and T3. This is because T2 initiated but did not finish by the time of the Tc checkpoint. T3 began after the checkpoint but concluded before the system crash at Tf. The UNDO operation will be applied to transaction T4, as it commenced after the Tc checkpoint but was never successfully completed.

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

<T4, COMM, 700>, <T4, SAL 8250>, T4, ABORT>

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

After the system is recovered MELVIN's department number will be 30, salary will be 8250 and commission will be 700.

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]

GENDER

Female: 1100000110110

Male: 0011111001001

AGE

<50: 1011001111100

>50: 0100110000011

INCOME

<100K: 1011101111110

>100K: 0100010000001

CHURN

NO: 1000010110101

YES: 0111101001010

- 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.

0011111001001 AND
1011001111100 AND
1011101111110 AND
0111101001010 =
0011001001000

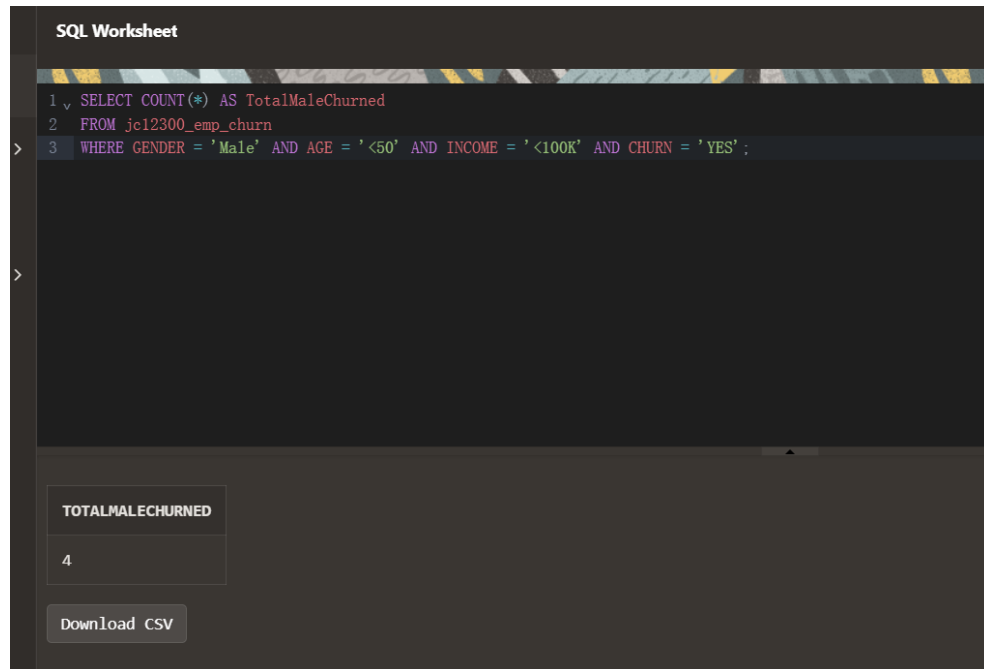
Since 4 ones exist in the result, there are 4 employees who satisfy the conditions above.

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

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]

```
SELECT COUNT(*) AS TotalMaleChurned
FROM jc12300_emp_churn
WHERE GENDER = 'Male' AND AGE = '<50' AND INCOME =
'<100K' AND CHURN = 'YES';
```



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

Submit the DDL code. [2 points]

```
CREATE BITMAP INDEX idx_gender ON
jc12300_emp_churn(GENDER);
```

```
CREATE BITMAP INDEX idx_age ON
jc12300_emp_churn(AGE);
```

```
CREATE BITMAP INDEX idx_income ON
jc12300_emp_churn(INCOME);
```

```
CREATE BITMAP INDEX idx_churn ON
jc12300_emp_churn(CHURN);
```