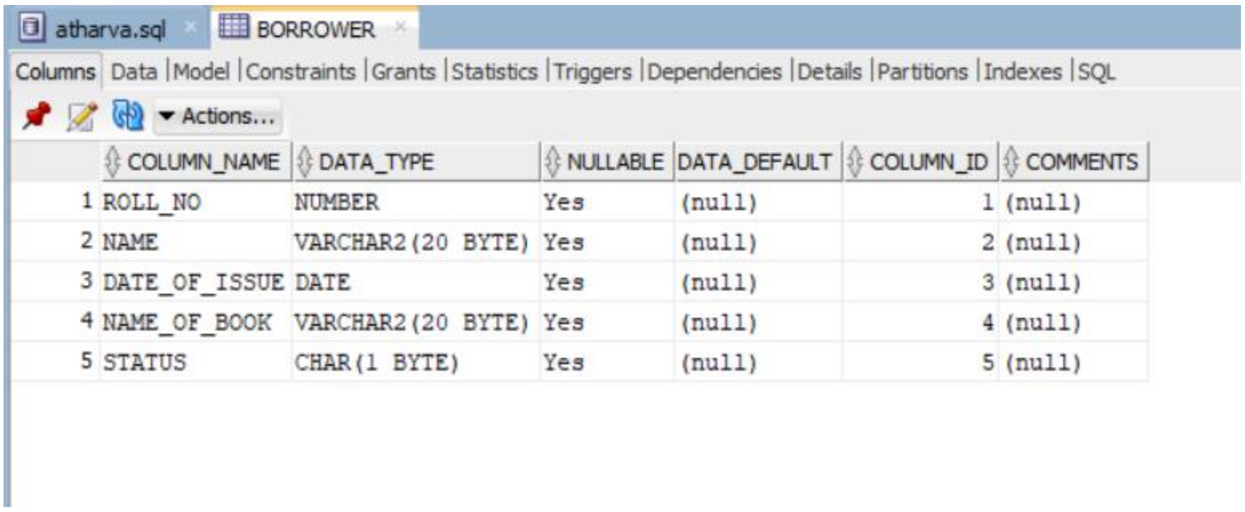
DBMS LAB CHIT SOLUTIONS

(1) Chit 2 –

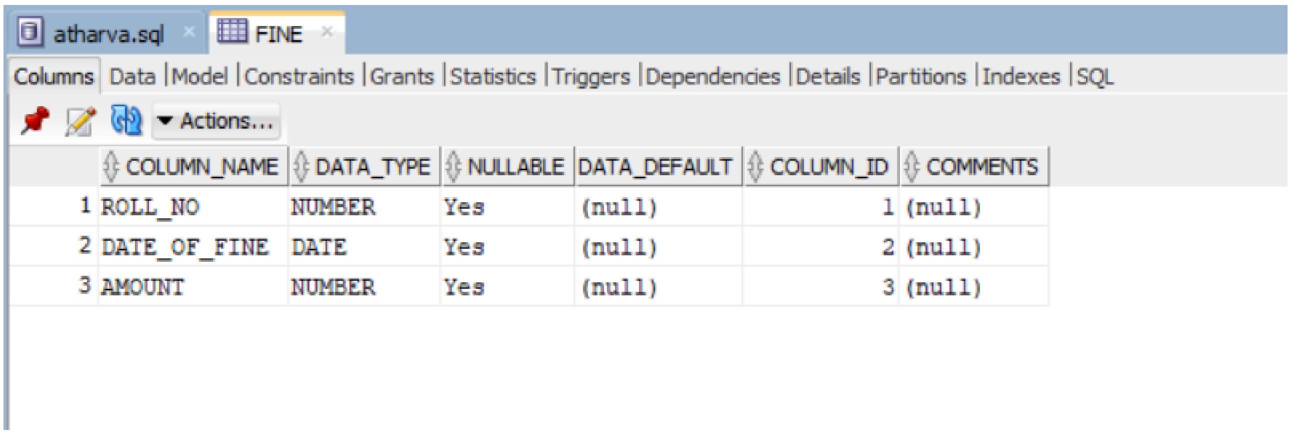
Borrower Table



atharva.sql
BORROWER
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL
Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ROLL_NO	NUMBER	Yes	(null)	1	(null)
2	NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	DATE_OF_ISSUE	DATE	Yes	(null)	3	(null)
4	NAME_OF_BOOK	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	STATUS	CHAR(1 BYTE)	Yes	(null)	5	(null)

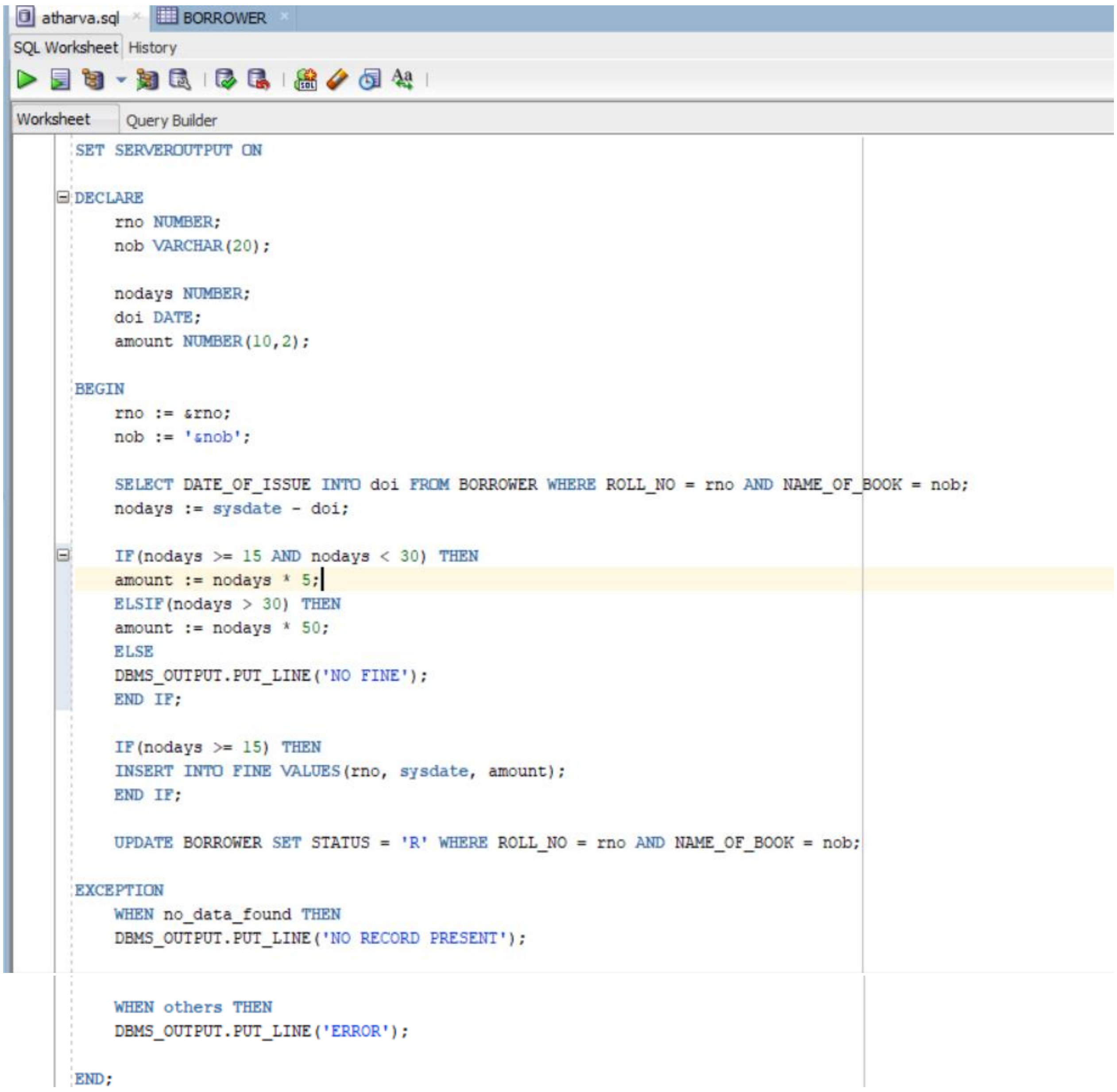
Fine Table



atharva.sql
FINE
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL
Actions...

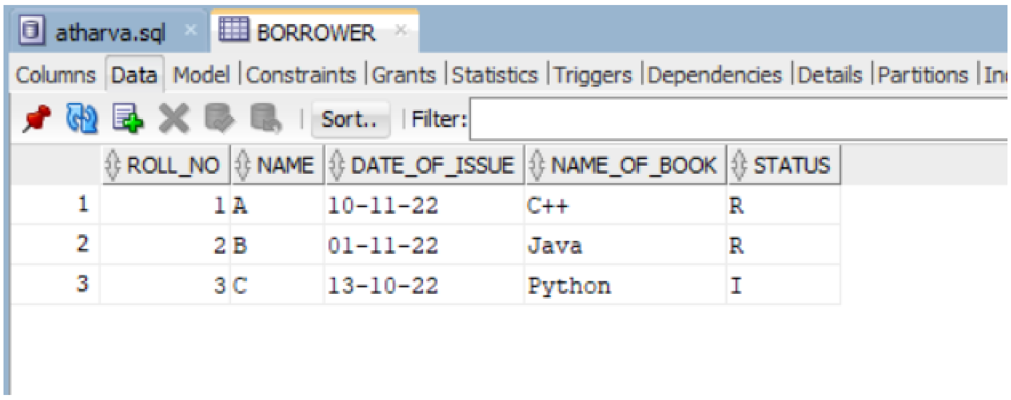
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ROLL_NO	NUMBER	Yes	(null)	1	(null)
2	DATE_OF_FINE	DATE	Yes	(null)	2	(null)
3	AMOUNT	NUMBER	Yes	(null)	3	(null)

PL/SQL Code

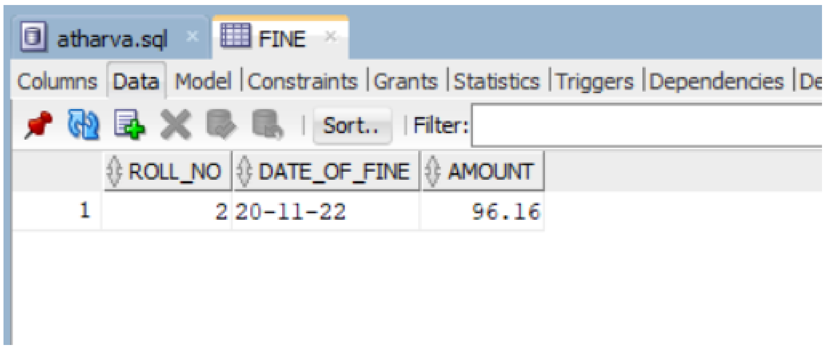


```
SET SERVEROUTPUT ON

DECLARE
    rno NUMBER;
    nob VARCHAR(20);

    nodays NUMBER;
    doi DATE;
    amount NUMBER(10,2);

BEGIN
    rno := &rno;
    nob := '&nob';

    SELECT DATE_OF_ISSUE INTO doi FROM BORROWER WHERE ROLL_NO = rno AND NAME_OF_BOOK = nob;
    nodays := sysdate - doi;

    IF(nodays >= 15 AND nodays < 30) THEN
    amount := nodays * 5;
    ELSIF(nodays > 30) THEN
    amount := nodays * 50;
    ELSE
    DBMS_OUTPUT.PUT_LINE('NO FINE');
    END IF;

    IF(nodays >= 15) THEN
    INSERT INTO FINE VALUES(rno, sysdate, amount);
    END IF;

    UPDATE BORROWER SET STATUS = 'R' WHERE ROLL_NO = rno AND NAME_OF_BOOK = nob;

EXCEPTION
    WHEN no_data_found THEN
    DBMS_OUTPUT.PUT_LINE('NO RECORD PRESENT');


    WHEN others THEN
    DBMS_OUTPUT.PUT_LINE('ERROR');

END;
```

Output



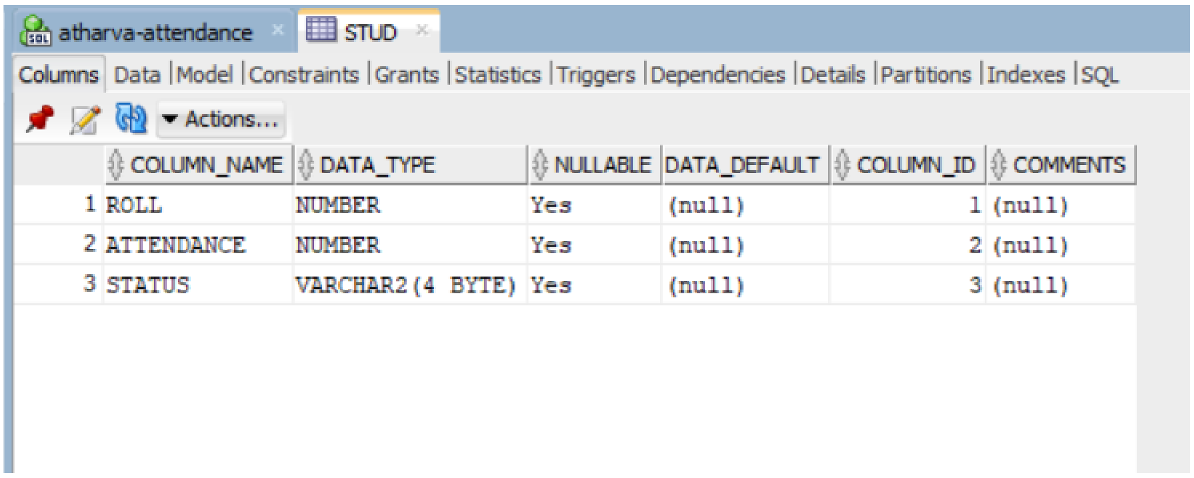
	ROLL_NO	NAME	DATE_OF_ISSUE	NAME_OF_BOOK	STATUS
1	1	A	10-11-22	C++	R
2	2	B	01-11-22	Java	R
3	3	C	13-10-22	Python	I



	ROLL_NO	DATE_OF_FINE	AMOUNT
1	2	20-11-22	96.16

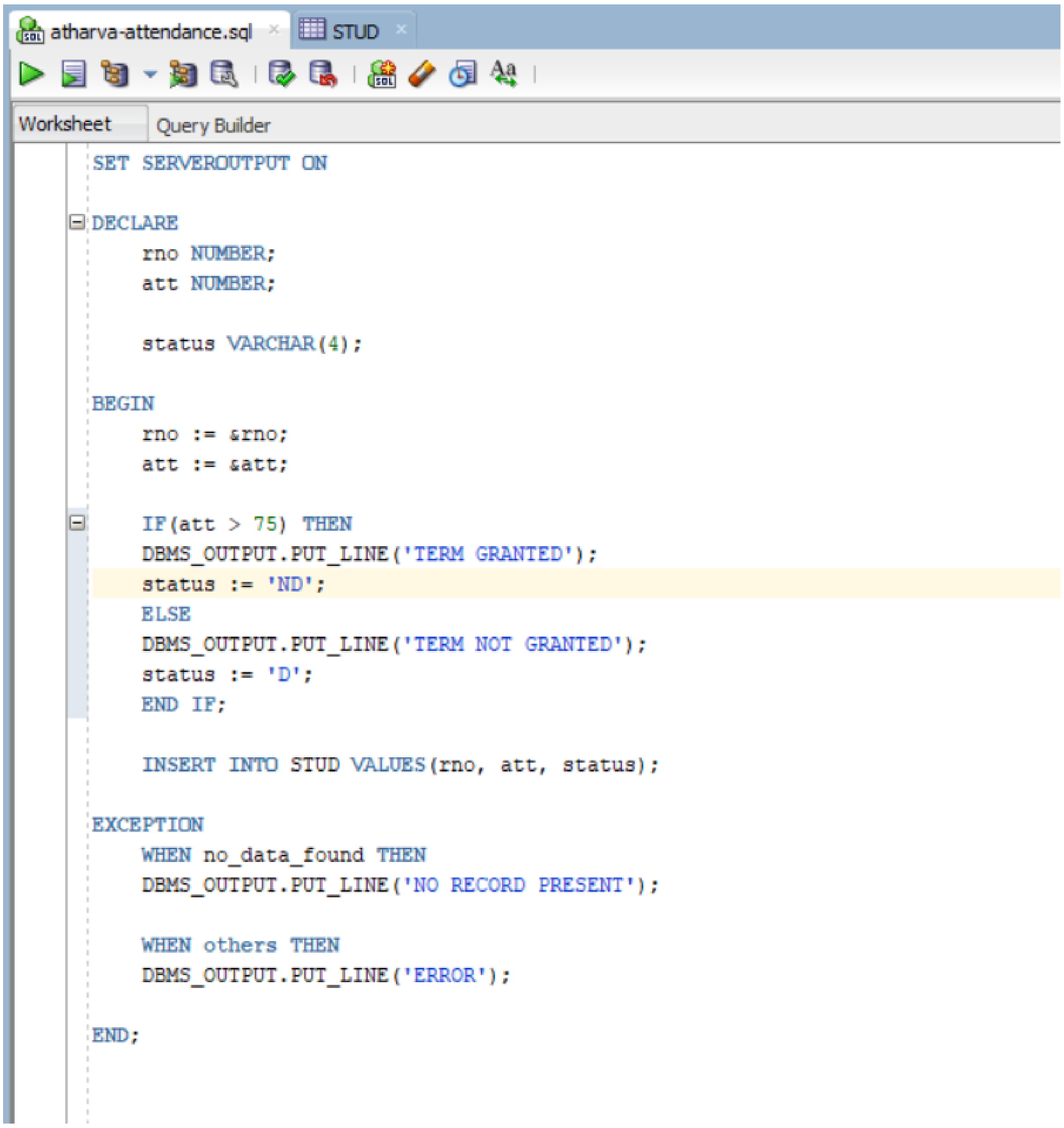
(2) **Chit 3 –**

Student Table


atharva-attendance
STUD
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL
Actions...

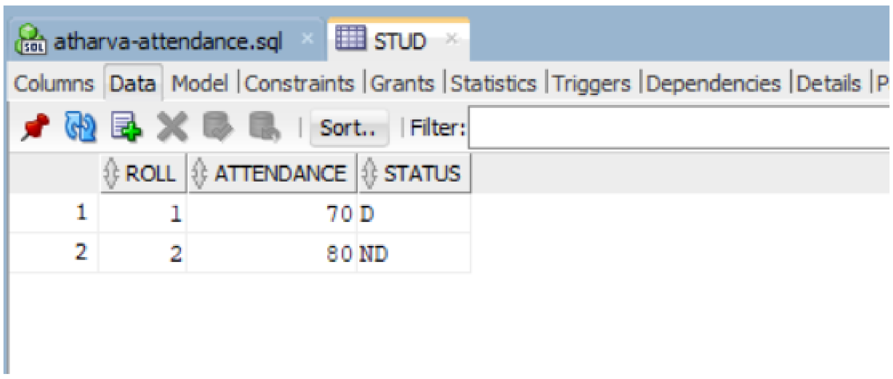
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ROLL	NUMBER	Yes	(null)	1	(null)
2	ATTENDANCE	NUMBER	Yes	(null)	2	(null)
3	STATUS	VARCHAR2(4 BYTE)	Yes	(null)	3	(null)

PL/SQL Code



```
SET SERVEROUTPUT ON

DECLARE
    rno NUMBER;
    att NUMBER;

    status VARCHAR(4);

BEGIN
    rno := &rno;
    att := &att;

    IF(att > 75) THEN
    DBMS_OUTPUT.PUT_LINE('TERM GRANTED');
    status := 'ND';
    ELSE
    DBMS_OUTPUT.PUT_LINE('TERM NOT GRANTED');
    status := 'D';
    END IF;

    INSERT INTO STUD VALUES(rno, att, status);

EXCEPTION
    WHEN no_data_found THEN
    DBMS_OUTPUT.PUT_LINE('NO RECORD PRESENT');

    WHEN others THEN
    DBMS_OUTPUT.PUT_LINE('ERROR');

END;
```

Output

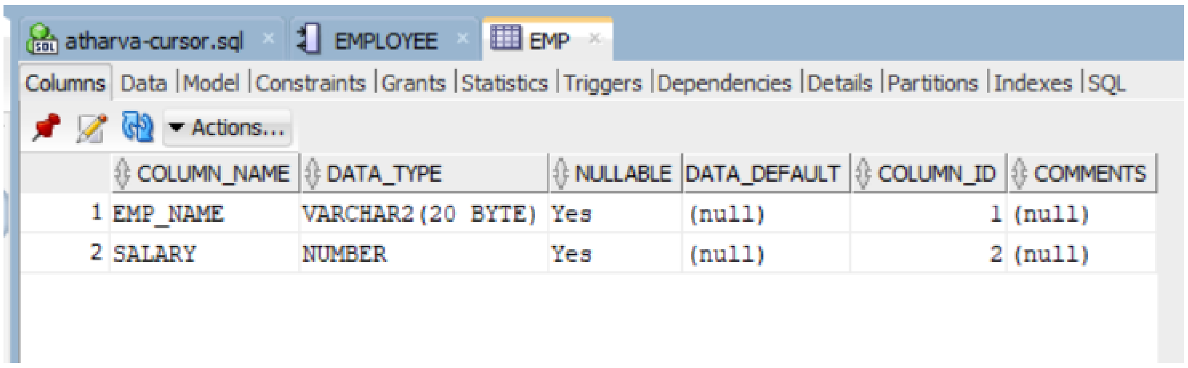


atharva-attendance.sql
STUD
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | P
Sort..
Filter:

	ROLL	ATTENDANCE	STATUS
1	1	70	D
2	2	80	ND

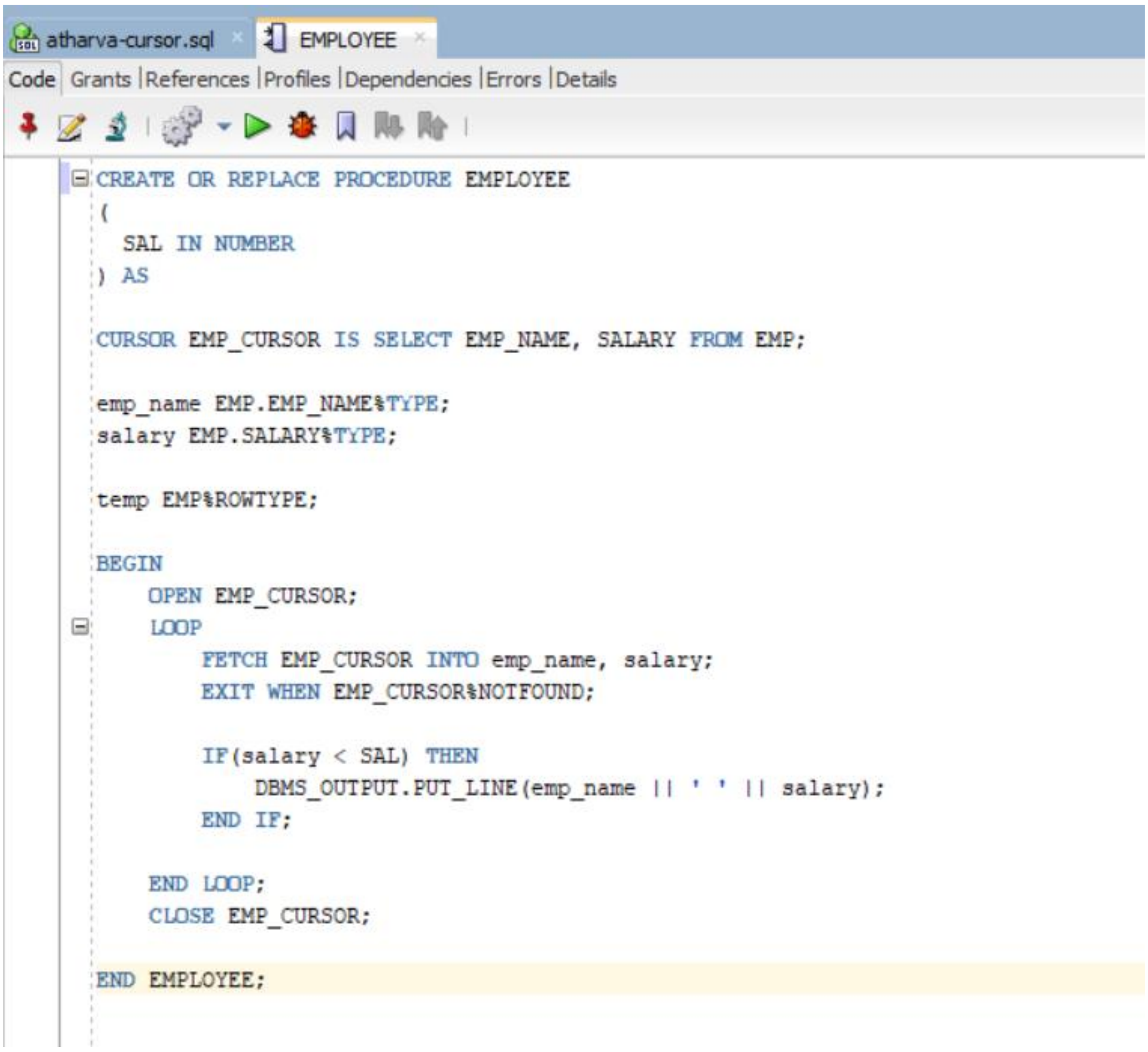
(3) Chit 5 –

EMP Table


atharva-cursor.sql
EMPLOYEE
EMP
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL
Actions...

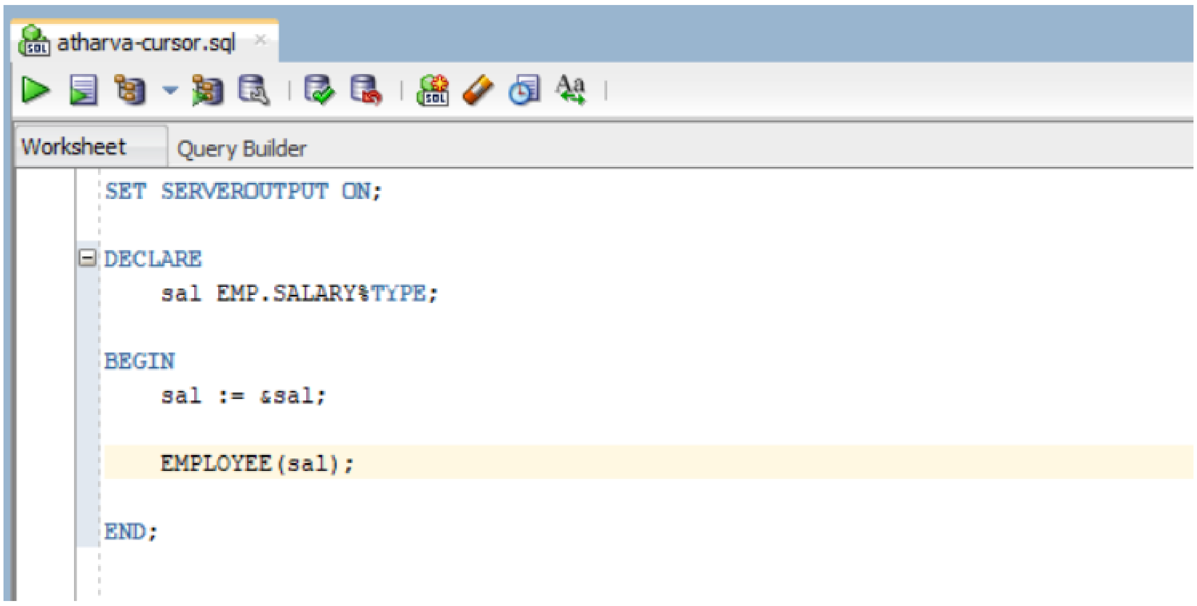
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMP_NAME	VARCHAR2(20 BYTE)	Yes	(null)	1	(null)
2	SALARY	NUMBER	Yes	(null)	2	(null)

EMPLOYEE Cursor


atharva-cursor.sql
EMPLOYEE
Code | Grants | References | Profiles | Dependencies | Errors | Details

```
CREATE OR REPLACE PROCEDURE EMPLOYEE
(
  SAL IN NUMBER
) AS

CURSOR EMP_CURSOR IS SELECT EMP_NAME, SALARY FROM EMP;

emp_name EMP.EMP_NAME%TYPE;
salary EMP.SALARY%TYPE;

temp EMP%ROWTYPE;

BEGIN
    OPEN EMP_CURSOR;
    LOOP
        FETCH EMP_CURSOR INTO emp_name, salary;
        EXIT WHEN EMP_CURSOR%NOTFOUND;

        IF(salary < SAL) THEN
            DBMS_OUTPUT.PUT_LINE(emp_name || ' ' || salary);
        END IF;

    END LOOP;
    CLOSE EMP_CURSOR;

END EMPLOYEE;
```

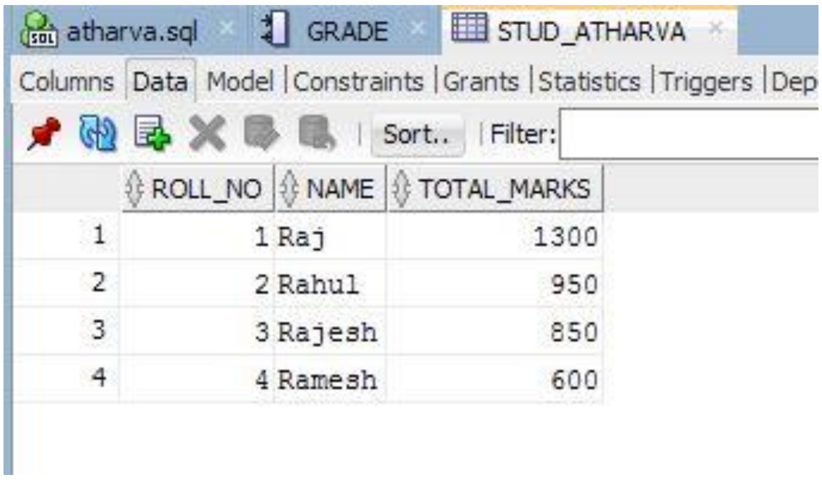
PL/SQL Code



```
SET SERVEROUTPUT ON;

DECLARE
    sal EMP.SALARY%TYPE;

BEGIN
    sal := &sal;

    EMPLOYEE(sal);

END;
```

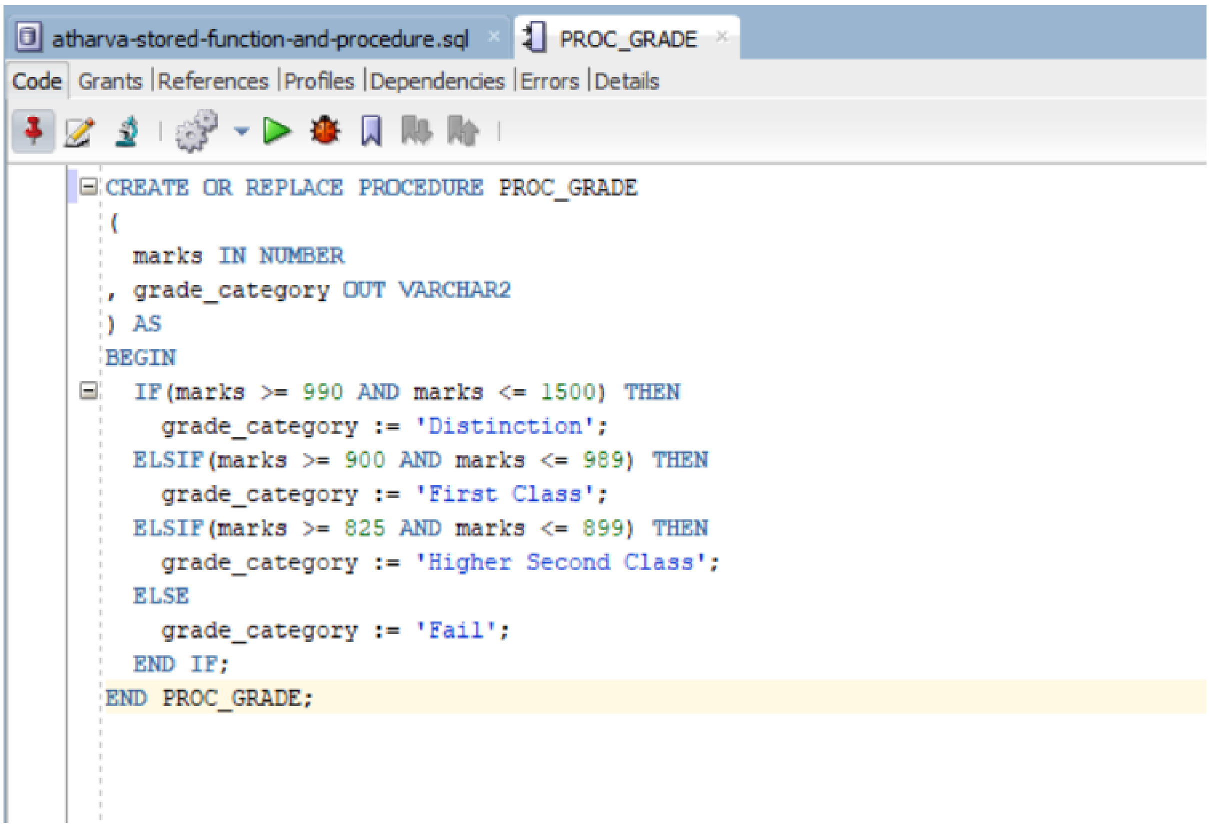
(4) Chit 6 –

STUDENT Table



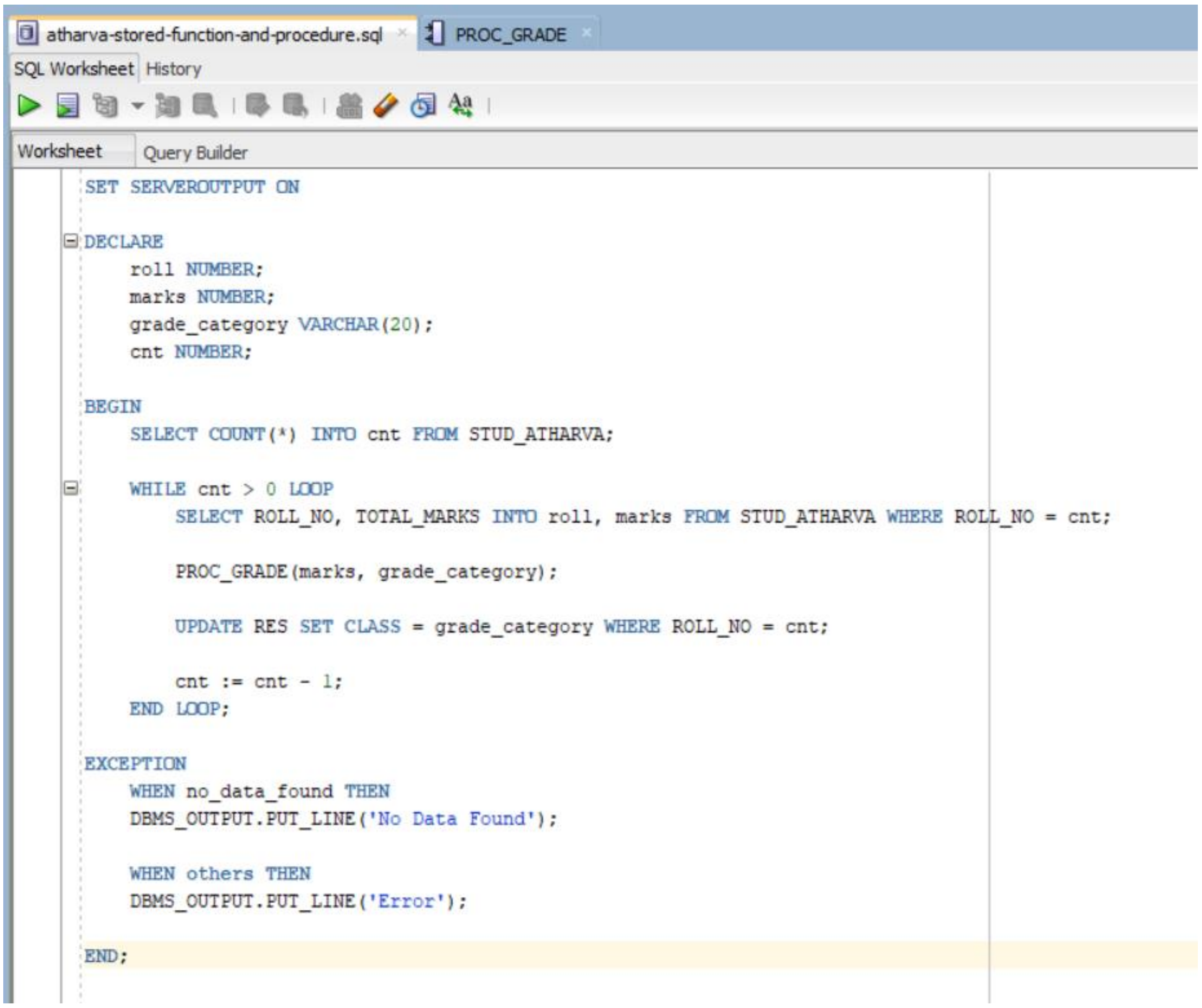
	ROLL_NO	NAME	TOTAL_MARKS
1	1	Raj	1300
2	2	Rahul	950
3	3	Rajesh	850
4	4	Ramesh	600

Stored Procedure



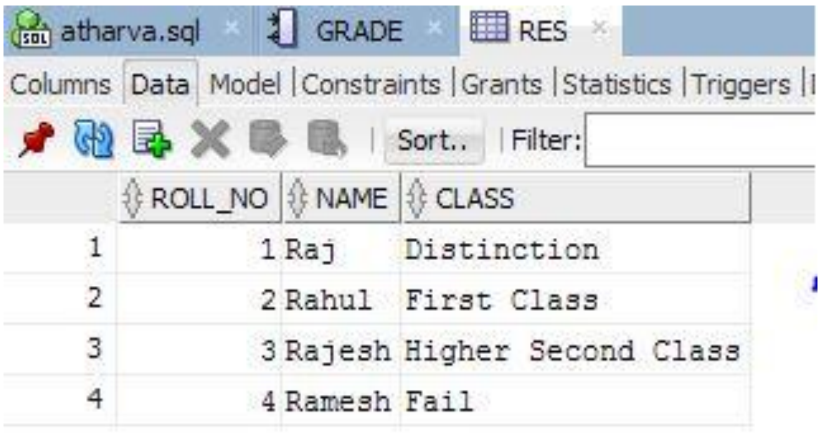
```
CREATE OR REPLACE PROCEDURE PROC_GRADE
(
  marks IN NUMBER
, grade_category OUT VARCHAR2
) AS
BEGIN
  IF(marks >= 990 AND marks <= 1500) THEN
    grade_category := 'Distinction';
  ELSIF(marks >= 900 AND marks <= 989) THEN
    grade_category := 'First Class';
  ELSIF(marks >= 825 AND marks <= 899) THEN
    grade_category := 'Higher Second Class';
  ELSE
    grade_category := 'Fail';
  END IF;
END PROC_GRADE;
```

PL/SQL Code



```
SET SERVEROUTPUT ON

DECLARE
    roll NUMBER;
    marks NUMBER;
    grade_category VARCHAR(20);
    cnt NUMBER;

BEGIN
    SELECT COUNT(*) INTO cnt FROM STUD_ATHARVA;

    WHILE cnt > 0 LOOP
        SELECT ROLL_NO, TOTAL_MARKS INTO roll, marks FROM STUD_ATHARVA WHERE ROLL_NO = cnt;

        PROC_GRADE(marks, grade_category);

        UPDATE RES SET CLASS = grade_category WHERE ROLL_NO = cnt;

        cnt := cnt - 1;
    END LOOP;

EXCEPTION
    WHEN no_data_found THEN
    DBMS_OUTPUT.PUT_LINE('No Data Found');

    WHEN others THEN
    DBMS_OUTPUT.PUT_LINE('Error');

END;
```

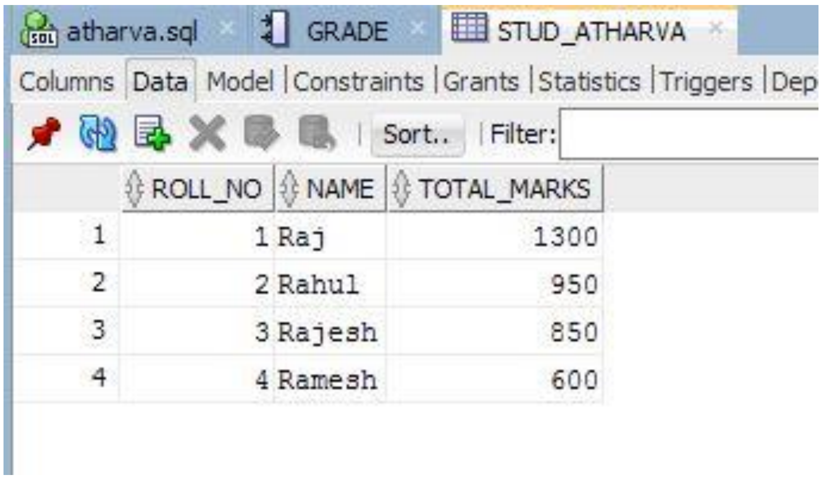
Output



	ROLL_NO	NAME	CLASS
1	1	Raj	Distinction
2	2	Rahul	First Class
3	3	Rajesh	Higher Second Class
4	4	Ramesh	Fail

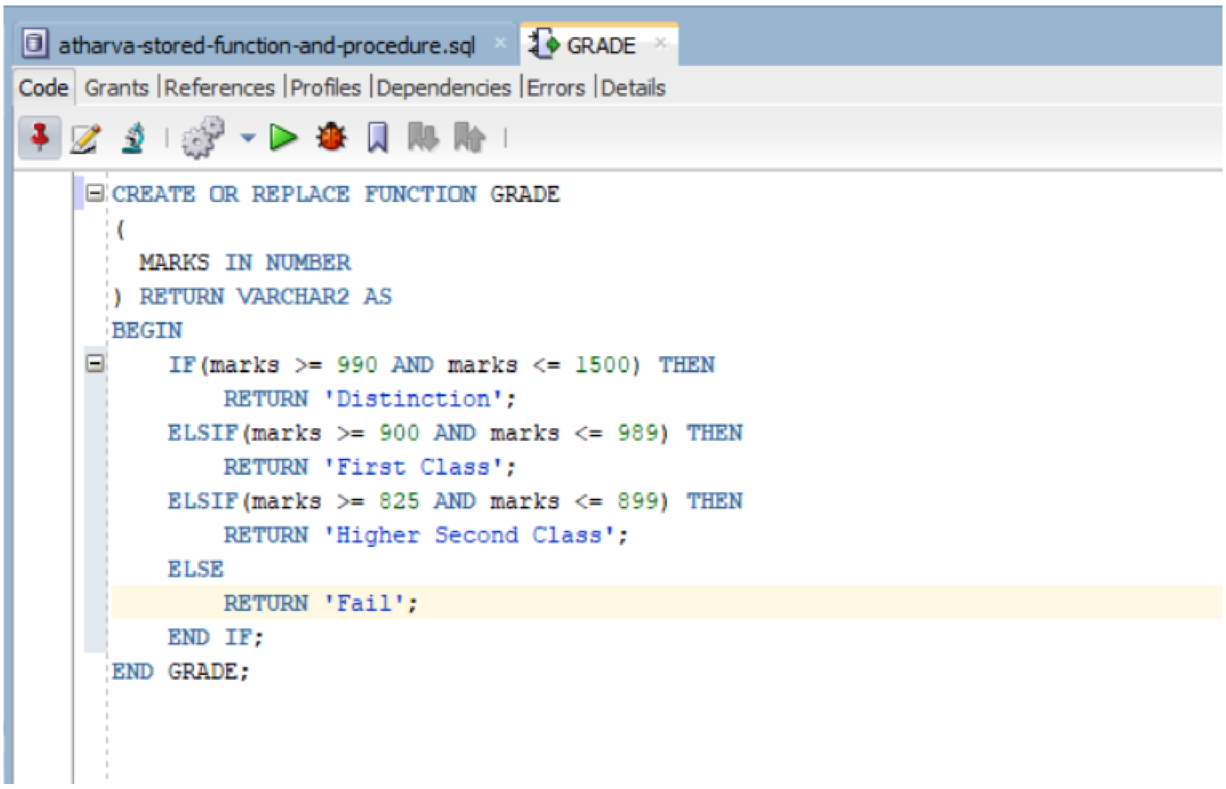
(5) Chit 7 –

STUDENT Table



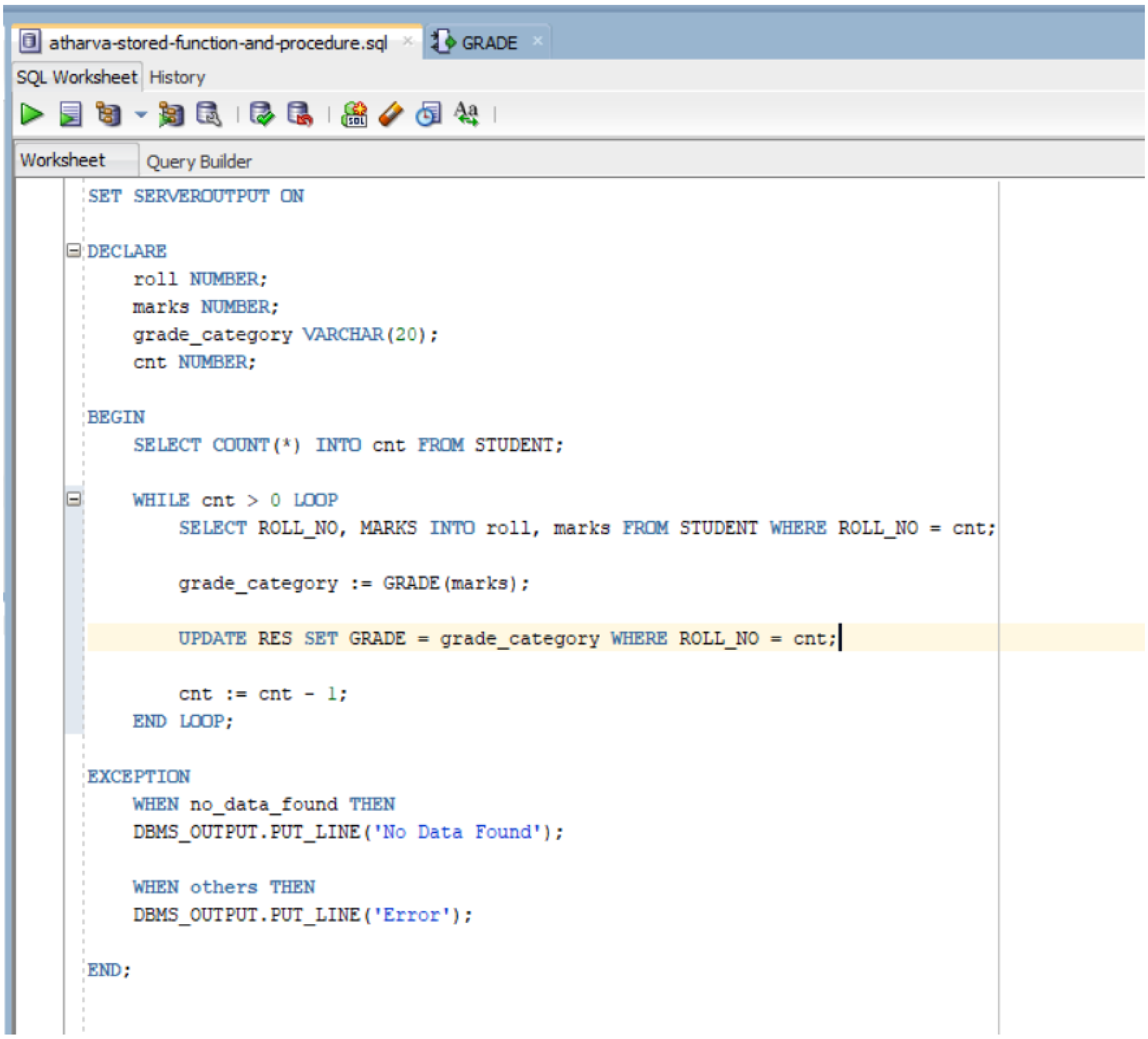
	ROLL_NO	NAME	TOTAL_MARKS
1	1	Raj	1300
2	2	Rahul	950
3	3	Rajesh	850
4	4	Ramesh	600

Stored Function



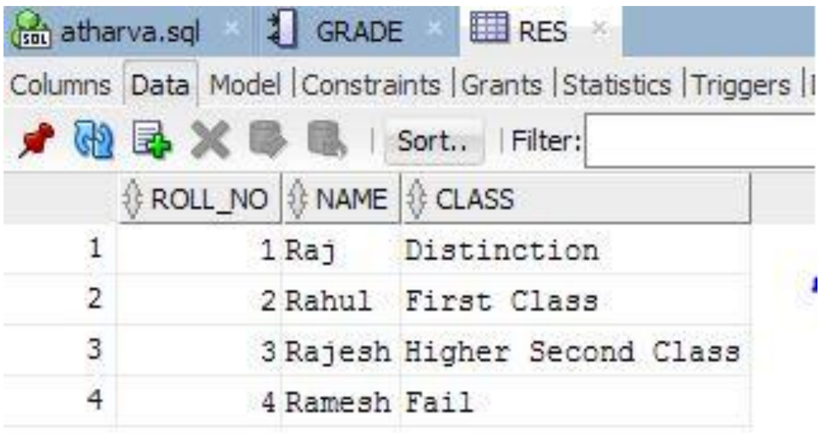
```
CREATE OR REPLACE FUNCTION GRADE
(
  MARKS IN NUMBER
) RETURN VARCHAR2 AS
BEGIN
    IF(marks >= 990 AND marks <= 1500) THEN
        RETURN 'Distinction';
    ELSIF(marks >= 900 AND marks <= 989) THEN
        RETURN 'First Class';
    ELSIF(marks >= 825 AND marks <= 899) THEN
        RETURN 'Higher Second Class';
    ELSE
        RETURN 'Fail';
    END IF;
END GRADE;
```

PL/SQL Code



```
SET SERVEROUTPUT ON

DECLARE
    roll NUMBER;
    marks NUMBER;
    grade_category VARCHAR(20);
    cnt NUMBER;

BEGIN
    SELECT COUNT(*) INTO cnt FROM STUDENT;

    WHILE cnt > 0 LOOP
        SELECT ROLL_NO, MARKS INTO roll, marks FROM STUDENT WHERE ROLL_NO = cnt;

        grade_category := GRADE(marks);

        UPDATE RES SET GRADE = grade_category WHERE ROLL_NO = cnt;

        cnt := cnt - 1;
    END LOOP;

EXCEPTION
    WHEN no_data_found THEN
    DBMS_OUTPUT.PUT_LINE('No Data Found');

    WHEN others THEN
    DBMS_OUTPUT.PUT_LINE('Error');

END;
```

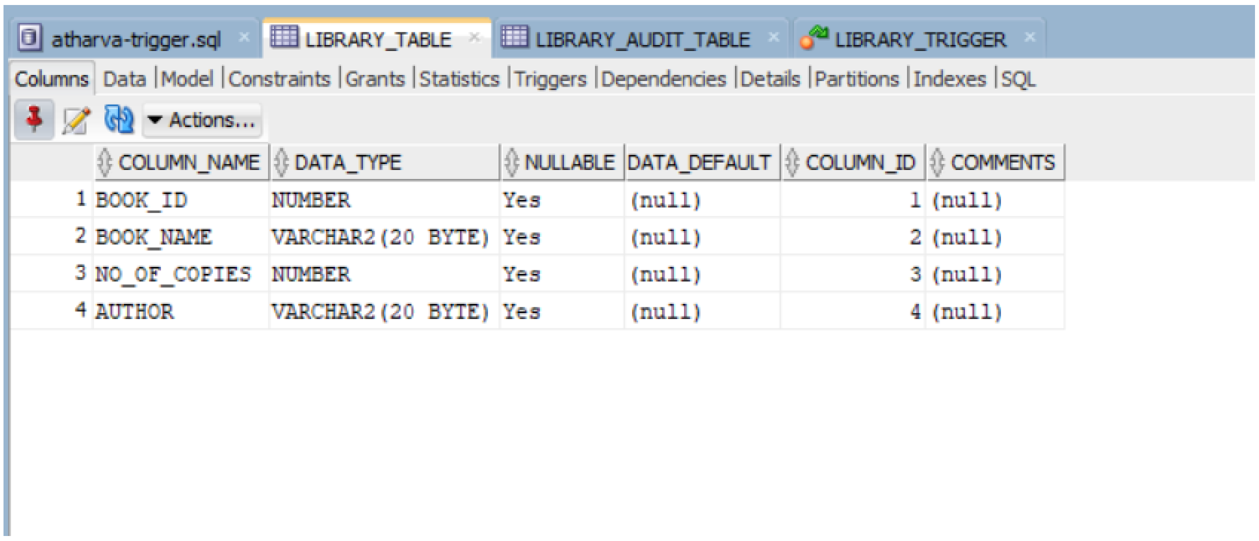
Output



	ROLL_NO	NAME	CLASS
1	1	Raj	Distinction
2	2	Rahul	First Class
3	3	Rajesh	Higher Second Class
4	4	Ramesh	Fail

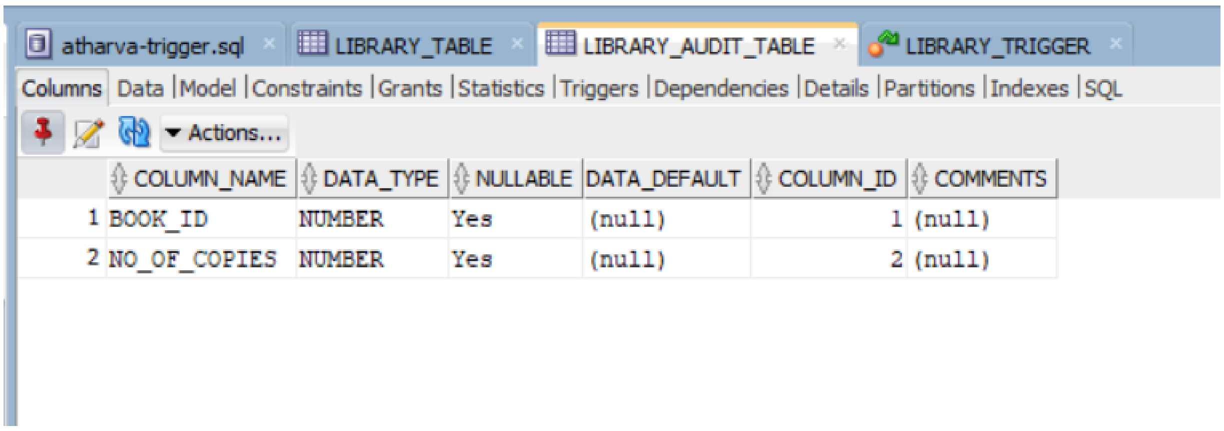
(6) Chit 11 –

LIBRARY_TABLE


atharva-trigger.sql
LIBRARY_TABLE
LIBRARY_AUDIT_TABLE
LIBRARY_TRIGGER
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL
Actions...

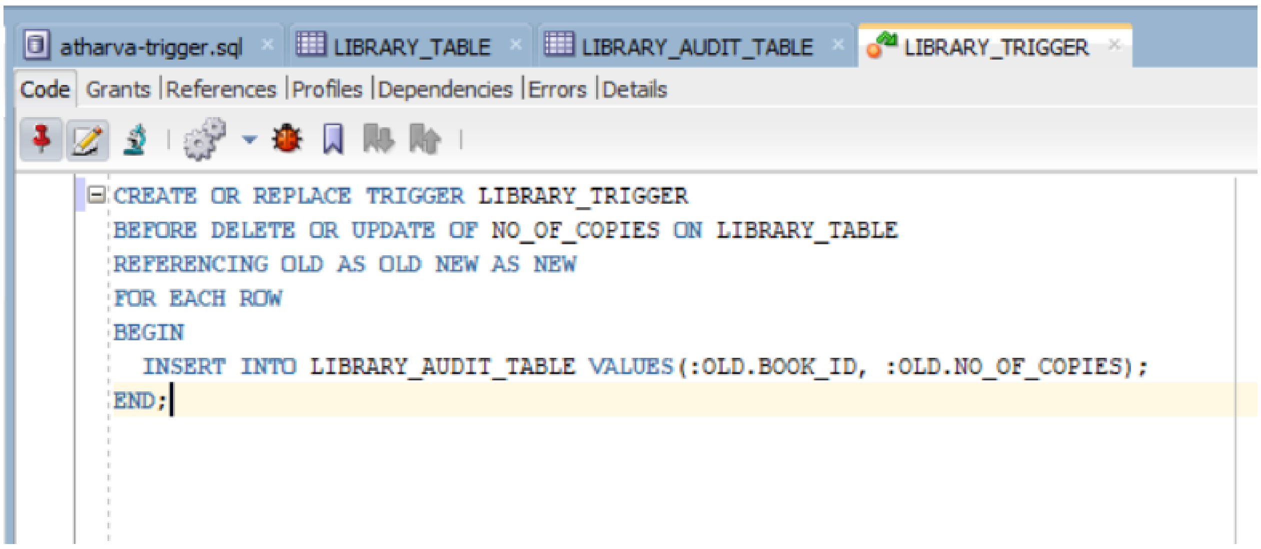
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BOOK_ID	NUMBER	Yes	(null)	1	(null)
2	BOOK_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	NO_OF_COPIES	NUMBER	Yes	(null)	3	(null)
4	AUTHOR	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)

LIBRARY_AUDIT_TABLE


atharva-trigger.sql
LIBRARY_TABLE
LIBRARY_AUDIT_TABLE
LIBRARY_TRIGGER
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL
Actions...

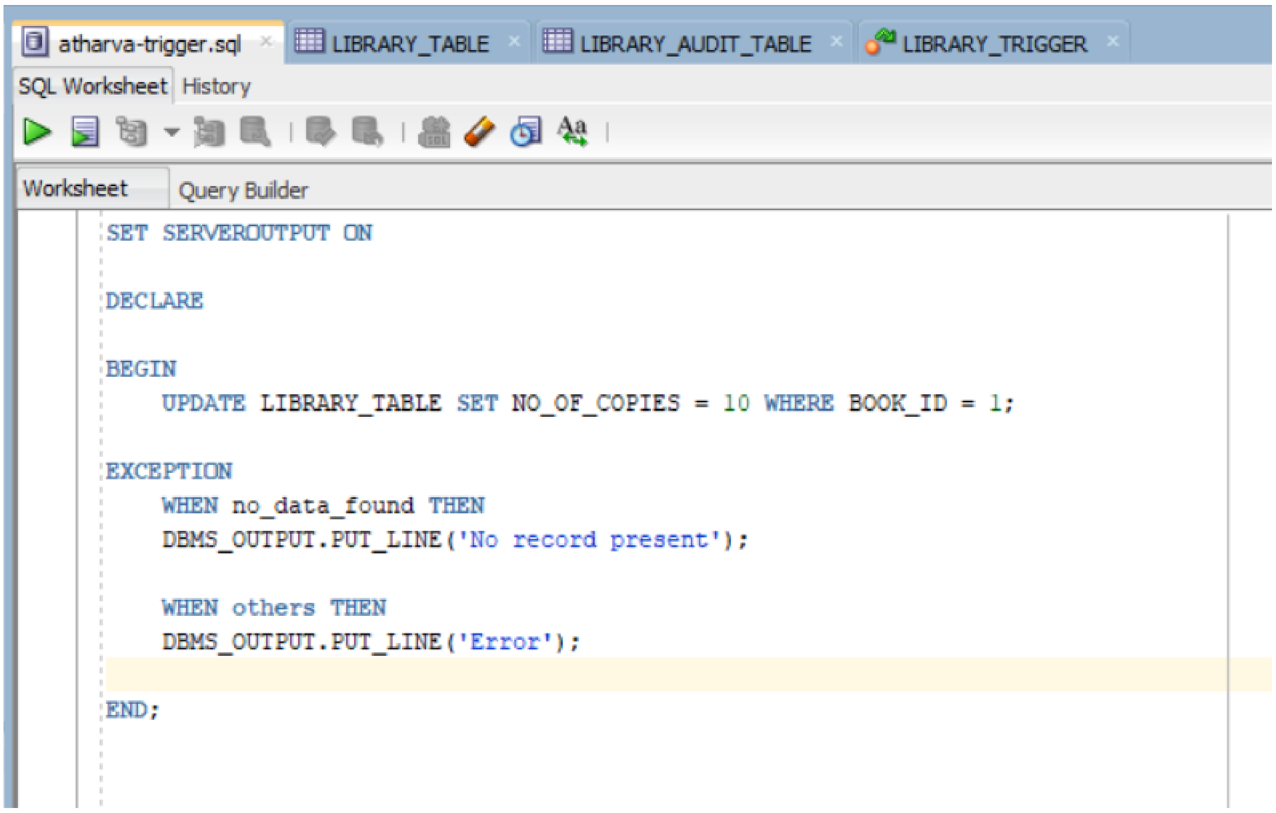
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BOOK_ID	NUMBER	Yes	(null)	1	(null)
2	NO_OF_COPIES	NUMBER	Yes	(null)	2	(null)

LIBRARY_TRIGGER


atharva-trigger.sql
LIBRARY_TABLE
LIBRARY_AUDIT_TABLE
LIBRARY_TRIGGER
Code | Grants | References | Profiles | Dependencies | Errors | Details

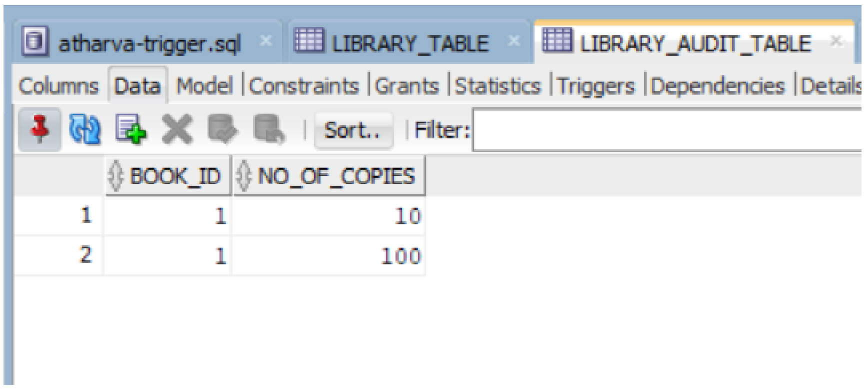
```
CREATE OR REPLACE TRIGGER LIBRARY_TRIGGER
BEFORE DELETE OR UPDATE OF NO_OF_COPIES ON LIBRARY_TABLE
REFERENCING OLD AS OLD NEW AS NEW
FOR EACH ROW
BEGIN
  INSERT INTO LIBRARY_AUDIT_TABLE VALUES(:OLD.BOOK_ID, :OLD.NO_OF_COPIES);
END;
```

PL/SQL CODE

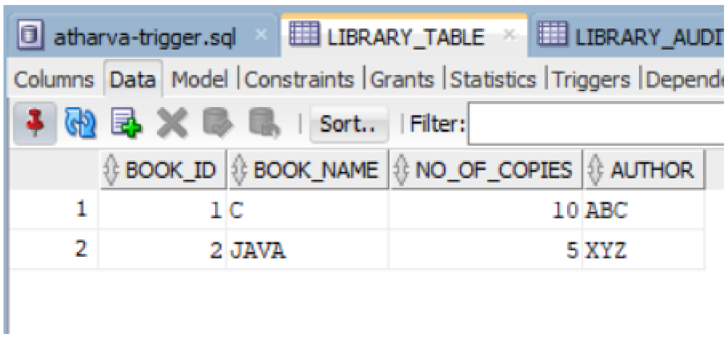


```
SET SERVEROUTPUT ON

DECLARE

BEGIN
    UPDATE LIBRARY_TABLE SET NO_OF_COPIES = 10 WHERE BOOK_ID = 1;

EXCEPTION
    WHEN no_data_found THEN
    DBMS_OUTPUT.PUT_LINE('No record present');

    WHEN others THEN
    DBMS_OUTPUT.PUT_LINE('Error');

END;
```

Output



	BOOK_ID	NO_OF_COPIES
1	1	10
2	1	100



	BOOK_ID	BOOK_NAME	NO_OF_COPIES	AUTHOR
1	1	C	10	ABC
2	2	JAVA	5	XYZ