## MINI PROJECT DBMS-II

(IMPLEMENTATION)

SHAIL N PARIKH
U18CO082@COED.SVNIT.AC.IN



Mini project DBMS

SHAIL N PARIKH

COED-II@SVNIT

2020

Sem IV

# Step 1

## LIBRARY SYSTEM

MANAGEMENT AND DATABASE HANDLING

## **Physical Designing**

## CREATING TABLES WITH PROPER CONSTRAINTS:

- Creation of Tables
- Declaring Primary Key
- Defining Foreign Key as per ER Diagram
- Populating Tables with Data

```
--Constraint and ordered Tables
--Order of deletion of tables
--drop table Book_Authors;
--drop table Book_Copies;
--drop table Book_Lending;
--drop table Published_by;
--drop table Pub_Phone;
--drop table Card_Phone;
--drop table Library_Branch;
--drop table Publisher;
--drop table Card;
--drop table Book;
--1 Book
create table Book (
  Book_ID varchar2(15) not null constraint book_pk primary
key,
  Title varchar2(50) not null,
  Pub_Year number(4)
);
--Book Table Insertion
INSERT INTO Book VALUES ('B100001', 'DBMS', 2017);
INSERT INTO Book VALUES ('B100002', 'ADBMS', 2016);
INSERT INTO Book VALUES ('B100003', 'CN', 2016);
```

--Inserting data into the tables.

```
INSERT INTO Book VALUES ('B100004', 'CG', 2015);
INSERT INTO Book VALUES ('B100005', 'OS', 2016);
--2 Library_Branch
create table Library_Branch (
  Branch_ID varchar2(15) not null constraint library_branch_pk
primary key,
  Branch_Name varchar2(30) not null,
  Address varchar2(50) not null
);
--Library_Branch Table Insertion
INSERT INTO Library_Branch VALUES ('LB100001', 'RR
NAGAR', 'BANGALORE');
INSERT INTO Library_Branch VALUES
('LB200001', 'RNSIT', 'BANGALORE');
INSERT INTO Library_Branch VALUES ('LB300001', 'RAJAJI
NAGAR', 'BANGALORE');
INSERT INTO Library_Branch VALUES
('LB400001', 'NITTE', 'MANGALORE');
INSERT INTO Library_Branch VALUES
('LB500001', 'MANIPAL', 'UDUPI');
--3 Publisher
create table Publisher (
  Pub_ID varchar2(25) not null constraint publisher_pk
primary key,
  Name varchar2(50) not null,
  Address varchar2(100)
```

```
);
--Publisher Table Insertion
INSERT INTO Publisher VALUES ('P100001', 'MCGRAW-
HILL', 'BANGALORE');
INSERT INTO Publisher VALUES ('P100002', 'PEARSON', 'NEW
DELHI');
INSERT INTO Publisher VALUES ('P100003', 'RANDOM'
HOUSE', 'HYDERABAD');
INSERT INTO Publisher VALUES ('P100004', 'HACHETTE
LIVRE', 'CHENNAI');
INSERT INTO Publisher VALUES ('P100005', 'GRUPO
PLANETA', 'BANGALORE');
--4 Card
create table Card (
  Card_No varchar2(20) not null constraint card_pk primary
key,
 Name varchar2(30) not null,
 Address varchar2(50),
 Expiry_date date not null
);
--Card Table Insertion
INSERT INTO Card VALUES ('M100100', 'CHANDLER', 'NEW
DELHI', '25-AUG-2020');
INSERT INTO Card VALUES
('M100101', 'ROSS', 'BANGALORE', '14-NOV-2020');
```

INSERT INTO Card VALUES ('M100102', 'OTIS', 'CHENNAI', '27-

NOV-2020');

```
INSERT INTO Card VALUES
('M100103', 'MAEVE', 'HYDERABAD', '10-JAN-2021');
INSERT INTO Card VALUES
('M100104', 'JOEY', 'BANGALORE', '31-DEC-2020');
--5 Book Authors
create table Book_Authors (
  Book_ID varchar2(15) not null,
  constraint fk_book_id foreign key(Book_ID) references Book
(Book_ID) on delete cascade,
  Author_Name varchar2(30)
);
alter table Book_Authors add constraint book_authors_pk
primary key (Book_ID, Author_Name);
--Book Authors Table Insertion
INSERT INTO Book_Authors VALUES ('B100001', 'NAVATHE');
INSERT INTO Book_Authors VALUES ('B100002', 'NAVATHE');
INSERT INTO Book_Authors VALUES ('B100003', 'TANENBAUM');
INSERT INTO Book_Authors VALUES ('B100004', 'TANENBAUM');
INSERT INTO Book_Authors VALUES ('B100003', 'EDWARD
ANGEL');
INSERT INTO Book_Authors VALUES ('B100005', 'GALVIN');
INSERT INTO Book_Authors VALUES ('B100001', 'EDWARD
ANGEL');
--6 Published_by
create table Published_by (
```

```
Book ID varchar2(15) not null,
  constraint fk_book2_id foreign key(Book_ID) references
Book (Book_ID) on delete cascade,
  Pub_ID varchar2(25) not null,
  constraint fk_pub2_id foreign key(Pub_ID) references
Publisher (Pub ID) on delete cascade
);
alter table Published_by add constraint published_by_pk
primary key (Book_ID, Pub_ID);
--Published_by Table Insertion
INSERT INTO Published_by VALUES ('B100001', 'P100001');
INSERT INTO Published_by VALUES ('B100001', 'P100002');
INSERT INTO Published_by VALUES ('B100002', 'P100002');
INSERT INTO Published_by VALUES ('B100003', 'P100003');
INSERT INTO Published_by VALUES ('B100003', 'P100005');
INSERT INTO Published_by VALUES ('B100004', 'P100004');
INSERT INTO Published_by VALUES ('B100005', 'P100005');
-- 7 Book Lending
create table Book_Lending (
  Book_ID varchar2(15) not null,
  constraint fk_book3_id foreign key(Book_ID) references
Book (Book_ID) on delete cascade,
  Card_No varchar2(20) not null,
  constraint fk_card3_id foreign key(Card_No) references
```

Card (Card No) on delete cascade,

```
Issue_Date date not null,

Return_Date date,

Due_Date date not null
);

alter table Book_Lending add constraint book_lending_pk
primary key (Book_ID, Card_No);
```

#### --Book\_Lending Table Insertion

INSERT INTO Book\_Lending VALUES ('B100001', 'M100100', '01-JAN-17', '01-JUN-17', '15-MAY-17');

INSERT INTO Book\_Lending VALUES ('B100001', 'M100103', '11-JAN-17', '11-MAR-17', '15-MAR-17');

INSERT INTO Book\_Lending VALUES ('B100003', 'M100100', '21-FEB-17', '21-APR-17', '15-APR-17');

INSERT INTO Book\_Lending VALUES ('B100004', 'M100104', '15-MAR-17', '15-JUL-17', '15-JUL-17');

INSERT INTO Book\_Lending VALUES ('B100004', 'M100102', '12-APR-17', '12-MAY-17', '15-MAY-17');

INSERT INTO Book\_Lending VALUES ('B100004', 'M100103', '12-APR-19', '7-MAY-19', '15-MAY-19');

INSERT INTO Book\_Lending VALUES ('B100004', 'M100101', '20-APR-19', '25-MAY-19', '15-MAY-19');

INSERT INTO Book\_Lending VALUES ('B100005', 'M100104', '7-JAN-20', '7-MAY-20', '23-APR-20');

INSERT INTO Book\_Lending VALUES ('B100005', 'M100103', '12-APR-20', '12-MAY-20', '15-MAY-20');

INSERT INTO Book\_Lending VALUES ('B100002', 'M100101', '20-APR-19', '25-MAY-19', '15-MAY-19');

```
create table Book Copies (
  Book_ID varchar2(15) not null,
  constraint fk book4 id foreign key(Book ID) references
Book (Book_ID) on delete cascade,
  Branch_ID varchar2(15) not null,
  constraint fk_branch4_id foreign key(Branch_ID) references
Library_Branch (Branch_ID) on delete cascade,
  No_of_Copies number
);
alter table Book_Copies add constraint book_copies_pk
primary key (Book_ID, Branch_ID);
--Book_Copies Table Insertion
INSERT INTO Book_Copies VALUES ('B100001', 'LB100001', 0);
INSERT INTO Book_Copies VALUES ('B100001', 'LB200001', 11);
INSERT INTO Book_Copies VALUES ('B100001', 'LB500001', 12);
INSERT INTO Book_Copies VALUES ('B100002', 'LB100001', 13);
INSERT INTO Book_Copies VALUES ('B100002', 'LB300001', 14);
INSERT INTO Book_Copies VALUES ('B100002', 'LB400001', 10);
INSERT INTO Book Copies VALUES ('B100003', 'LB200001', 11);
INSERT INTO Book_Copies VALUES ('B100003', 'LB300001', 7);
INSERT INTO Book_Copies VALUES ('B100003', 'LB400001', 6);
INSERT INTO Book_Copies VALUES ('B100004', 'LB100001', 5);
INSERT INTO Book_Copies VALUES ('B100004', 'LB500001', 8);
INSERT INTO Book_Copies VALUES ('B100004', 'LB200001', 10);
INSERT INTO Book_Copies VALUES ('B100005', 'LB500001', 8);
```

INSERT INTO Book\_Copies VALUES ('B100005', 'LB300001', 4);

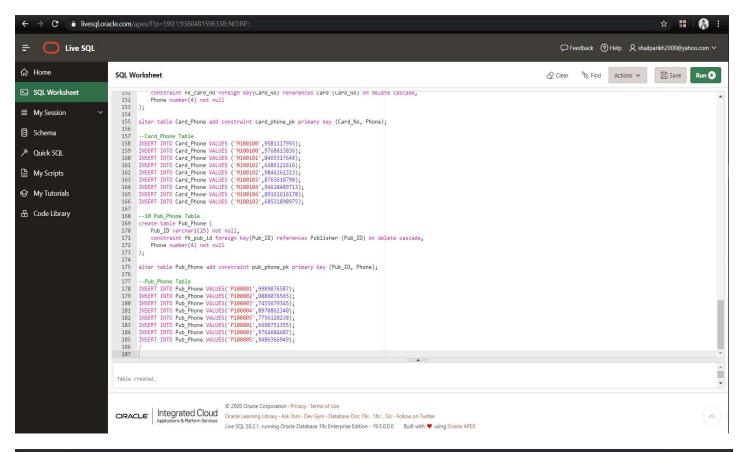
```
--9 Card Phone
create table Card_Phone (
  Card_No varchar2(20) not null,
  constraint fk_card_no foreign key(Card_No) references Card
(Card_No) on delete cascade,
  Phone number not null
);
alter table Card_Phone add constraint card_phone_pk primary
key (Card_No, Phone);
-- Card Phone Table Insertion
INSERT INTO Card_Phone VALUES ('M100100',9581117955);
INSERT INTO Card_Phone VALUES ('M100100',9768613836);
INSERT INTO Card_Phone VALUES ('M100101',8465317649);
INSERT INTO Card_Phone VALUES ('M100102',6486121616);
INSERT INTO Card_Phone VALUES ('M100102',9846161313);
INSERT INTO Card_Phone VALUES ('M100103',8765618798);
INSERT INTO Card_Phone VALUES ('M100104',94618489713);
INSERT INTO Card_Phone VALUES ('M100104',89161616178);
INSERT INTO Card_Phone VALUES ('M100103',68531898975);
--10 Pub_Phone
create table Pub_Phone (
  Pub_ID varchar2(25) not null,
```

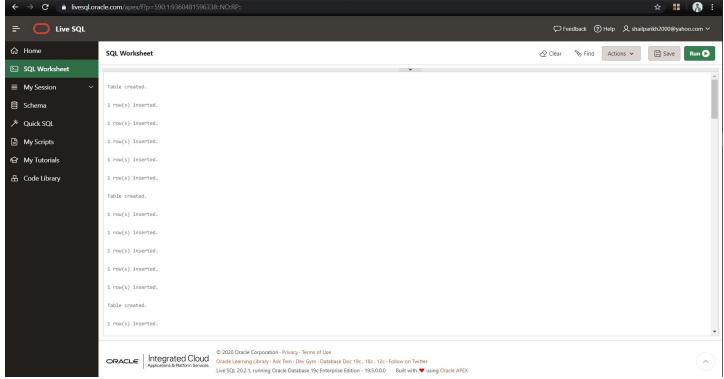
```
constraint fk_pub_id foreign key(Pub_ID) references
Publisher (Pub_ID) on delete cascade,
Phone number not null
);
```

alter table Pub\_Phone add constraint pub\_phone\_pk primary key (Pub\_ID, Phone);

#### --Pub\_Phone Table Insertion

```
INSERT INTO Pub_Phone VALUES('P100001',9989076587);
INSERT INTO Pub_Phone VALUES('P100002',9889076565);
INSERT INTO Pub_Phone VALUES('P100003',7455679345);
INSERT INTO Pub_Phone VALUES('P100004',8970862340);
INSERT INTO Pub_Phone VALUES('P100005',7756120238);
INSERT INTO Pub_Phone VALUES('P100001',6698751355);
INSERT INTO Pub_Phone VALUES('P100003',9764684687);
INSERT INTO Pub_Phone VALUES('P100005',9486366949);
/
```





# Step 2

## LIBRARY SYSTEM

MANAGEMENT AND DATABASE HANDLING

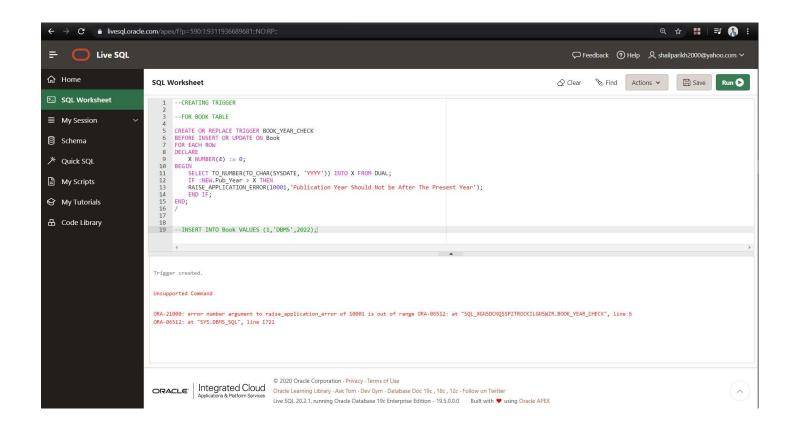
## **Physical Designing**

## **CREATING TRIGGERS:**

- Check Publishing Year
- Validate the value of Number of Copies of Book
- Creating Default Date for Return\_Date
- Checking Expiry\_Date

```
--PL SQL CODES:
--CREATING TRIGGER
--FOR BOOK TABLE: TO VALIDATE THE PUBLICTION YEAR OF
BOOK
CREATE OR REPLACE TRIGGER BOOK_YEAR_CHECK
BEFORE INSERT OR UPDATE ON Book
FOR EACH ROW
DECLARE
  X NUMBER(4) := 0;
BEGIN
  SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) INTO X
FROM DUAL;
      IF :NEW.Pub_Year > X THEN
      RAISE_APPLICATION_ERROR(10001, 'Publication Year
Should Not be After The Present Year');
      END IF;
END;
--INSERT INTO Book VALUES (1, 'DBMS', 2022);
```

--INSERT INTO Book VALUES (1,'DBMS',2017);



#### --FOR BOOK\_COPIES TABLE : CHECKING VALUE OF NO OF COPIES

CREATE OR REPLACE TRIGGER BOOK\_COPIES\_CHECK

BEFORE INSERT OR UPDATE ON Book\_Copies

FOR EACH ROW

```
BEGIN
```

```
IF :NEW.No_of_Copies < 0 THEN
```

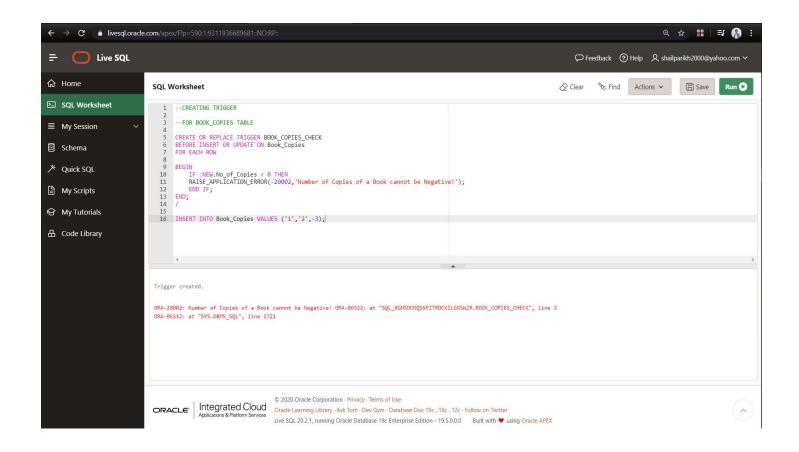
RAISE\_APPLICATION\_ERROR(-20002,'Number of Copies of a Book cannot be Negative!');

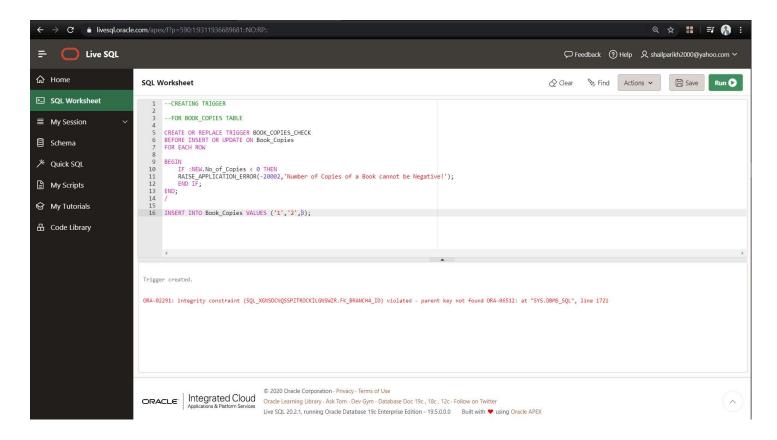
END IF;

END;

/

--INSERT INTO Book\_Copies VALUES ('1','2',-3);





## --FOR BOOK\_LENDING : TO SET DEFAULT DATE AS PRESENT DATE

#### CREATE OR REPLACE TRIGGER CURRENT\_DATE

BEFORE INSERT OR UPDATE ON Book\_Lending

FOR EACH ROW

#### **BEGIN**

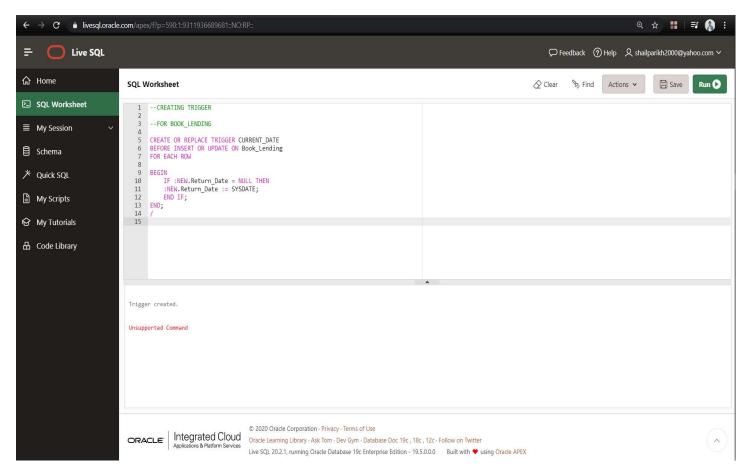
IF :NEW.Return\_Date = NULL THEN

:NEW.Return\_Date := SYSDATE;

END IF;

END;

/



## --FOR BOOK\_LENDING TABLE : EXPIRY\_DATE CHECK IN CORRESPONDENCE WITH DUE\_DATE

```
CREATE OR REPLACE TRIGGER EXPIRY_CHECK
```

BEFORE INSERT OR UPDATE ON Book\_Lending

FOR EACH ROW

```
DECLARE

X DATE := SYSDATE;

BEGIN

SELECT Expiry_Date INTO X FROM Card WHERE Card_No = :NEW.Card_No;

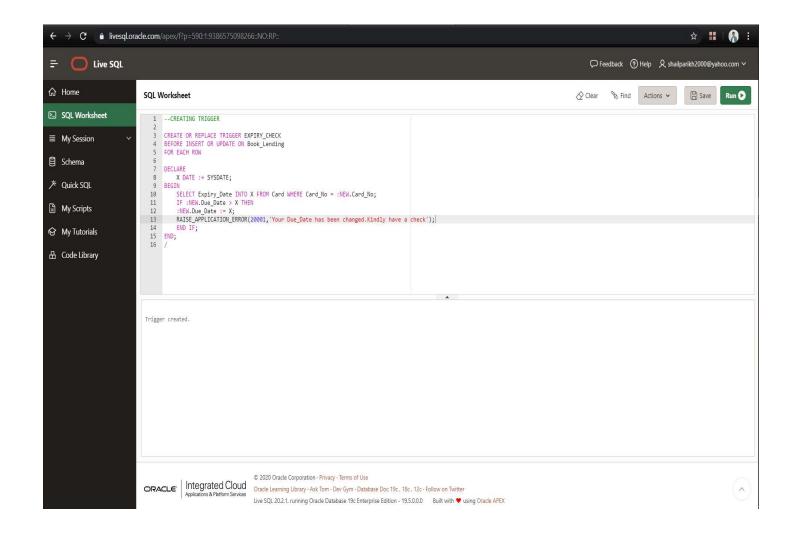
IF :NEW.Due_Date > X THEN

:NEW.Due_Date := X;

RAISE_APPLICATION_ERROR(20001, 'Your Due_Date has been changed.Kindly have a check');

END IF;

END;
```



## LIBRARY SYSTEM

MANAGEMENT AND DATABASE HANDLING

### **Physical Designing**

# CREATING PROCEDURES AND FUNCTIONS:

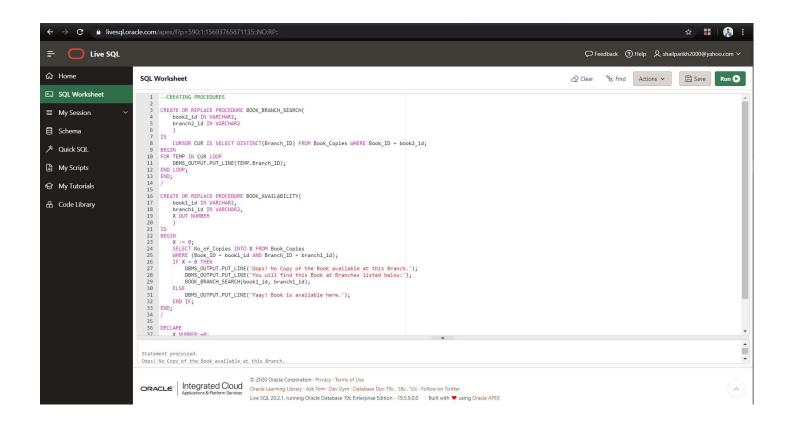
- Book Search
- Book Availability
- Fine Due to Single Book
- Total Fine Calculator

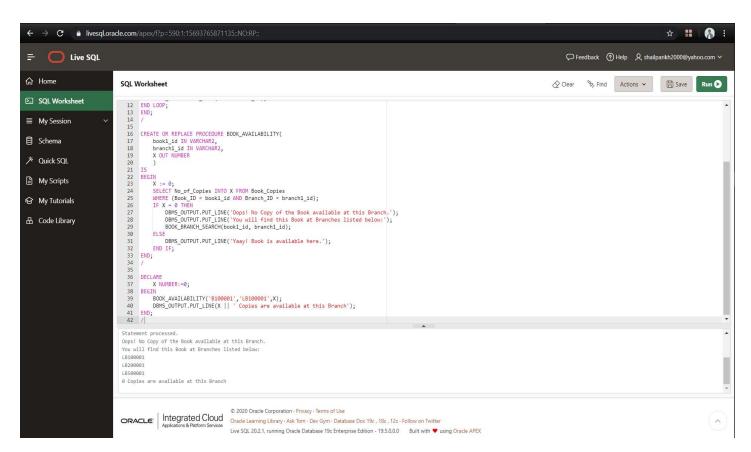
Step 3

#### SALIENT FEATURE

```
--PROCEDURE TO SEARCH A BOOK AT A PATICULAR BRANCH
CREATE OR REPLACE PROCEDURE BOOK_BRANCH_SEARCH(
 book2_id IN VARCHAR2,
 branch2 id IN VARCHAR2
 )
IS
  CURSOR CUR IS SELECT DISTINCT(Branch_ID) FROM
Book_Copies WHERE Book_ID = book2_id;
BEGIN
FOR TEMP IN CUR LOOP
 DBMS_OUTPUT.PUT_LINE(TEMP.Branch_ID);
END LOOP;
END;
--PROCEDURE TO CHECK WHETHER A BOOK IS AVAILABLE
OR NOT AT EVERY BRANCH
CREATE OR REPLACE PROCEDURE BOOK_AVAILABILITY(
 bookl_id IN VARCHAR2,
 branchl_id IN VARCHAR2,
 X OUT NUMBER
 )
IS
BEGIN
```

```
X := 0;
  SELECT No_of_Copies INTO X FROM Book_Copies
  WHERE (Book_ID = bookl_id AND Branch_ID = branchl_id);
  IF X = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Oops! No Copy of the Book
available at this Branch.');
    DBMS_OUTPUT.PUT_LINE('You will find this Book at
Branches listed below:');
    BOOK_BRANCH_SEARCH(bookl_id, branchl_id);
  ELSE
    DBMS_OUTPUT_LINE('Yaay! Book is available here.');
  END IF;
END;
DECLARE
  X NUMBER:=0;
BEGIN
  BOOK_AVAILABILITY('B100001','LB100001',X);
  DBMS_OUTPUT.PUT_LINE(X | | 'Copies are available at this
Branch');
END;
```

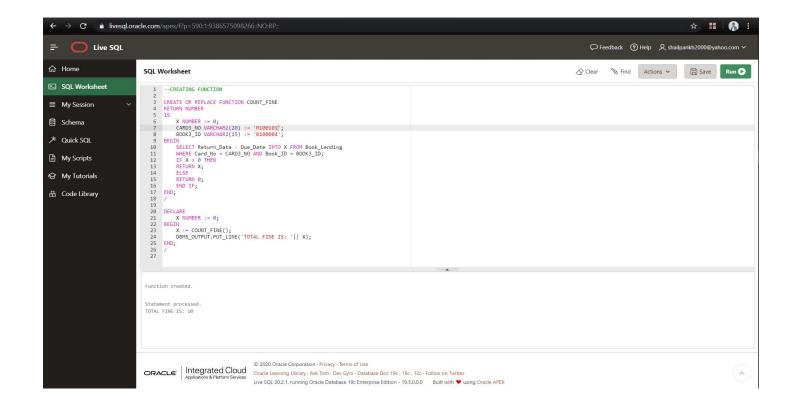




#### SALIENT FEATURE

#### --FUNCTION TO CALCULATE FINE DUE TO A BOOK

```
CREATE OR REPLACE FUNCTION COUNT_FINE
RETURN NUMBER
IS
      X NUMBER := 0;
      CARD3_NO VARCHAR2(20) := 'M100102';
      BOOK3_ID VARCHAR2(15) := 'B100004';
BEGIN
      SELECT Return_Date - Due_Date INTO X FROM
Book_Lending
      WHERE Card_No = CARD3_NO AND Book_ID =
BOOK3_ID;
      IF X > 0 THEN
      RETURN X;
      ELSE
      RETURN 0;
      END IF;
END;
DECLARE
 X NUMBER := 0;
BEGIN
 X := COUNT_FINE();
 DBMS_OUTPUT_LINE('TOTAL FINE IS: '| | X);
END;
```



## --CREATING A FUNCTION TO FIND TOTAL\_FINE OF A MEMBER

CREATE OR REPLACE FUNCTION TOTAL FINE

#### RETURN NUMBER

IS

```
CARD4_NO VARCHAR2(20) := 'M100101';

CURSOR CUR IS SELECT Book_ID FROM Book_Lending
WHERE Card_No = CARD4_NO;

X NUMBER:=0;

TOT NUMBER:=0;
```

**BEGIN** 

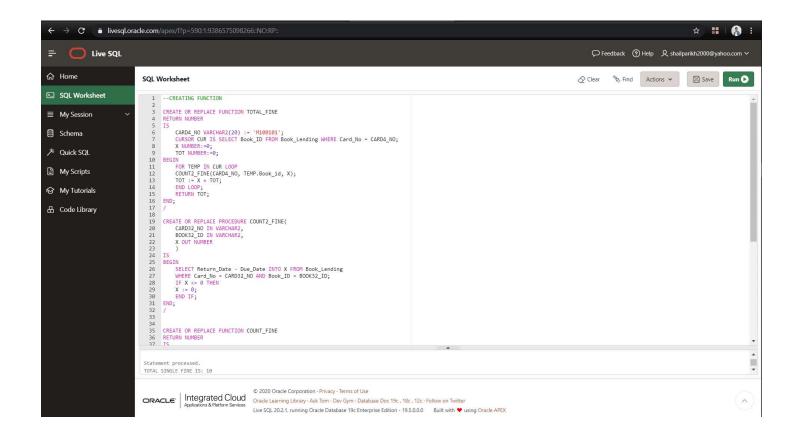
FOR TEMP IN CUR LOOP

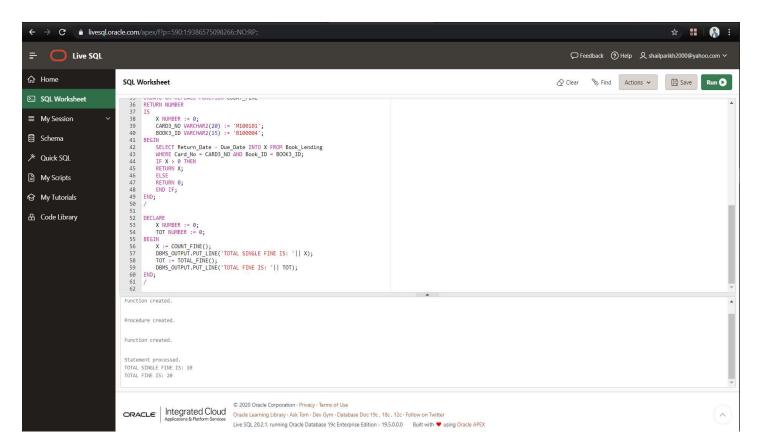
COUNT2\_FINE(CARD4\_NO, TEMP.Book\_id, X);

TOT := X + TOT;

END LOOP;

```
RETURN TOT;
END;
--PROCEDURE TO COUNT FINE DUE TO EACH BOOK
CREATE OR REPLACE PROCEDURE COUNT2_FINE(
 CARD32_NO IN VARCHAR2,
      BOOK32_ID IN VARCHAR2,
      X OUT NUMBER
      )
IS
BEGIN
      SELECT Return_Date - Due_Date INTO X FROM
Book_Lending
      WHERE Card_No = CARD32_NO AND Book_ID =
BOOK32_ID;
      IF X <= 0 THEN
      X := 0;
      END IF;
END;
```





## LIBRARY SYSTEM

MANAGEMENT AND DATABASE HANDLING

### **Physical Designing**

## **DEFINING VIEWS:**

- Showing Book details
- Showing Publisher details
- Showing details of Members and their Card

Step 4

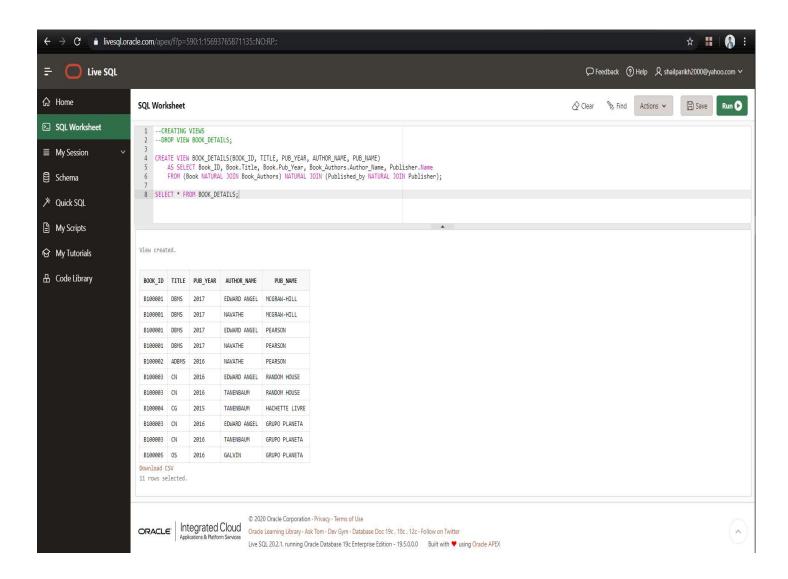
#### --CREATING VIEWS

#### -- VIEW SHOWING DETAILS OF BOOKS

CREATE VIEW BOOK\_DETAILS(BOOK\_ID, TITLE, PUB\_YEAR, AUTHOR\_NAME, PUB\_NAME)

AS SELECT Book\_ID, Book.Title, Book.Pub\_Year, Book\_Authors.Author\_Name, Publisher.Name

FROM (Book NATURAL JOIN Book\_Authors) NATURAL JOIN (Published\_by NATURAL JOIN Publisher);

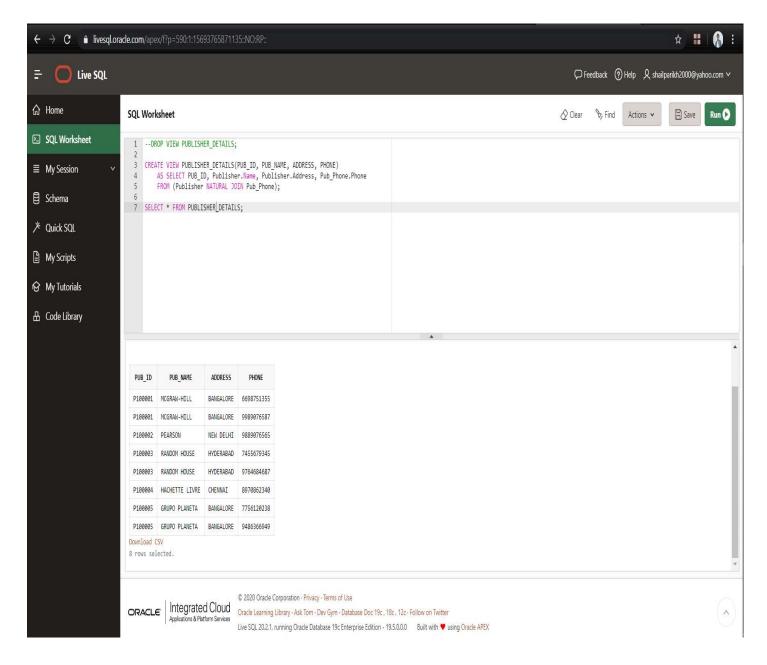


#### --VIEW DISPLAYING DETAILS OF PUBLISHERS

CREATE VIEW PUBLISHER\_DETAILS(PUB\_ID, PUB\_NAME, ADDRESS, PHONE)

AS SELECT PUB\_ID, Publisher.Name, Publisher.Address, Pub\_Phone.Phone

FROM (Publisher NATURAL JOIN Pub\_Phone);

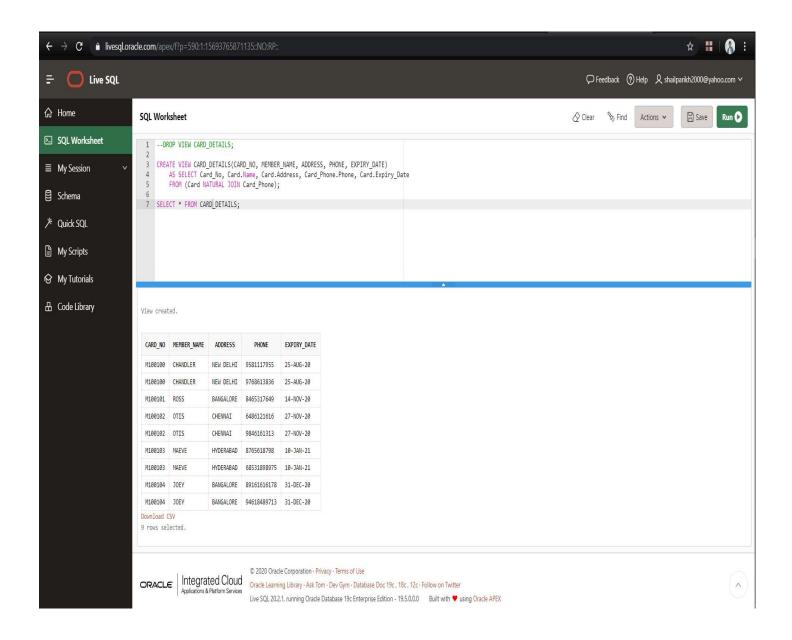


#### --VIEW DISPLAYING DETAILS OF MEMBERS AND THEIR CARD

CREATE VIEW CARD\_DETAILS(CARD\_NO, MEMBER\_NAME, ADDRESS, PHONE, EXPIRY\_DATE)

AS SELECT Card\_No, Card.Name, Card.Address, Card\_Phone.Phone, Card.Expiry\_Date

FROM (Card NATURAL JOIN Card\_Phone);



## LIBRARY SYSTEM

MANAGEMENT AND DATABASE HANDLING

## **Physical Designing**

## TYPES OF JOINS:

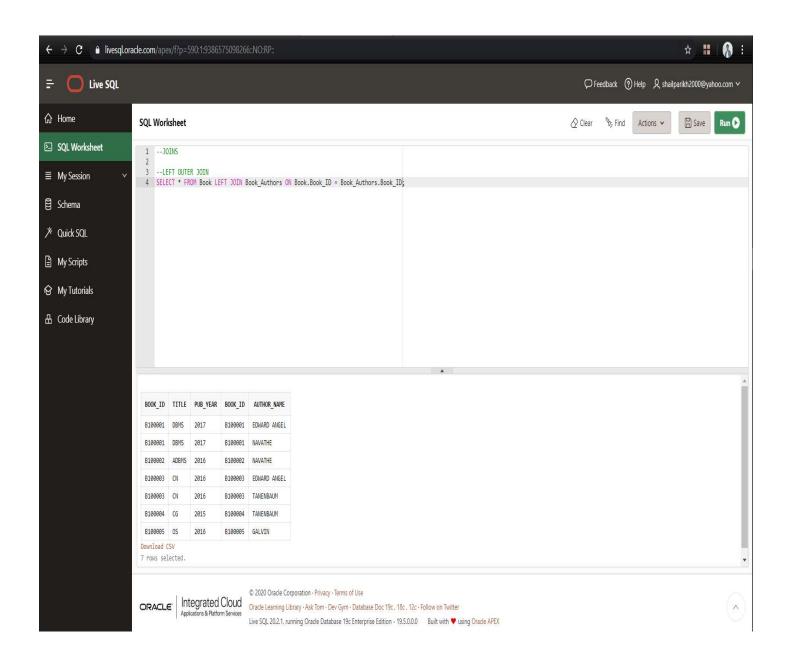
- Left Outer Join
- Right Outer Join
- Full Outer Join
- Natural Join

Step 5

#### --JOINS

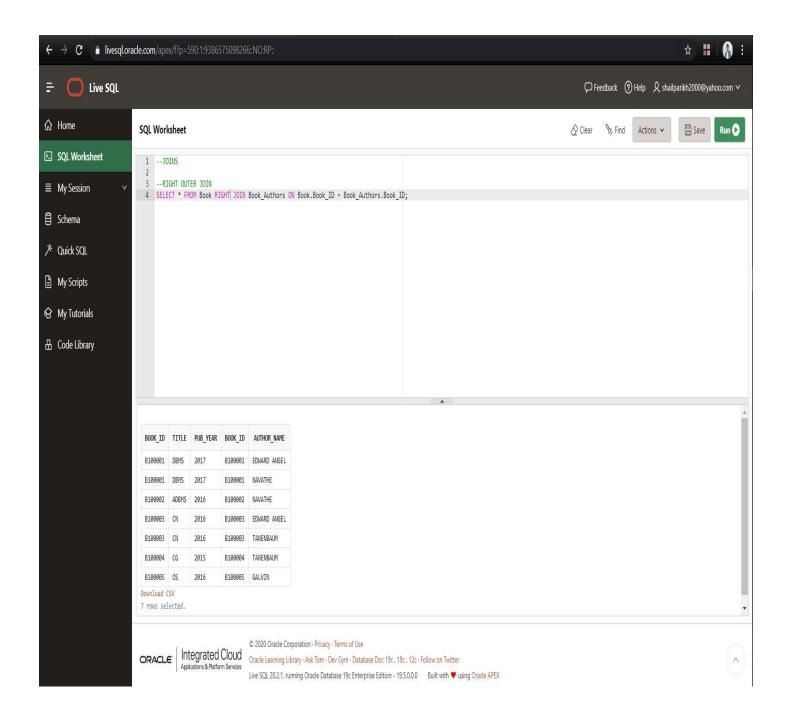
#### --LEFT OUTER JOIN

SELECT \* FROM Book LEFT JOIN Book\_Authors ON Book\_Book\_ID = Book\_Authors.Book\_ID;



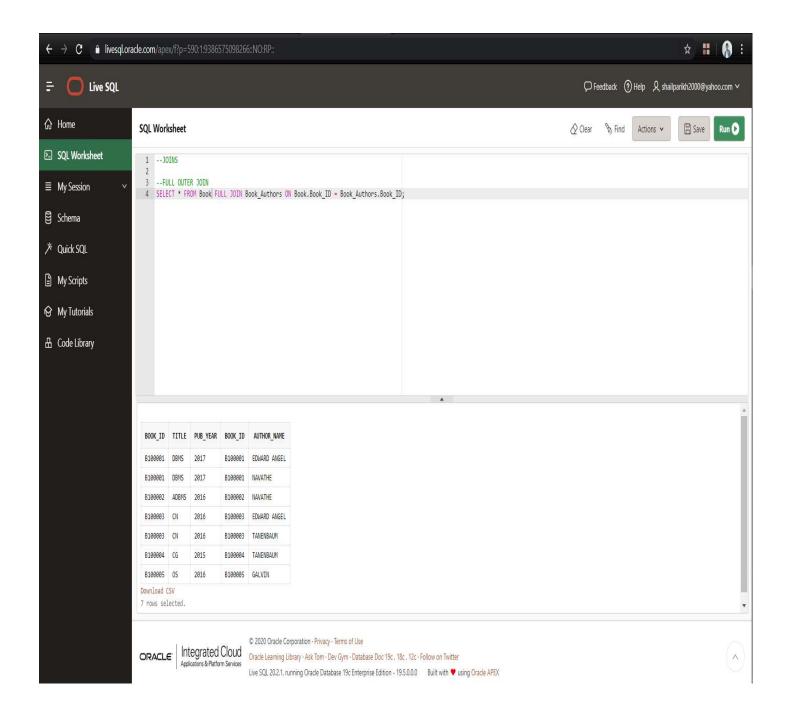
#### --RIGHT OUTER JOIN

SELECT \* FROM Book RIGHT JOIN Book\_Authors ON
Book\_Book\_ID = Book\_Authors.Book\_ID;



#### --FULL OUTER JOIN

SELECT \* FROM Book FULL JOIN Book\_Authors ON Book\_Book\_ID = Book\_Authors.Book\_ID;



#### --NATURAL JOIN

SELECT \* FROM (Book NATURAL JOIN Book\_Authors)
NATURAL JOIN (Publisher NATURAL JOIN Published\_by);

