



MINI PROJECT DBMS-II

(IMPLEMENTATION)

SHAIL N PARIKH

U18CO082@COED.SVNIT.AC.IN





U18CO082

Mini project DBMS

SHAIL N PARIKH

COED-II@SVNIT

2020

Sem IV

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

Step 1

--Inserting data into the tables.

--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);

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');
```

--8 Book_Copies

```

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);

```

```
INSERT INTO Book_Copies VALUES ('B100005', 'LB400001', 18);
```

--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);

/

← → ↻ livesql.oracle.com/apex/?p=590:1:9360481596338:NO:RP::

Live SQL Feedback Help shailparikh2000@yahoo.com

Home SQL Worksheet Clear Find Actions Save Run

SQL Worksheet

```
151 constraint fk_card_no_foreign key(Card_No) references Card (Card_No) on delete cascade,
152 Phone number(4) not null
153 );
154
155 alter table Card_Phone add constraint card_phone_pk primary key (Card_No, Phone);
156
157 --Card_Phone Table
158 INSERT INTO Card_Phone VALUES ('M100100',9581117955);
159 INSERT INTO Card_Phone VALUES ('M100100',9768613836);
160 INSERT INTO Card_Phone VALUES ('M100101',8465317649);
161 INSERT INTO Card_Phone VALUES ('M100102',6486121616);
162 INSERT INTO Card_Phone VALUES ('M100102',9846161313);
163 INSERT INTO Card_Phone VALUES ('M100103',8765618798);
164 INSERT INTO Card_Phone VALUES ('M100104',94618489713);
165 INSERT INTO Card_Phone VALUES ('M100104',89161616178);
166 INSERT INTO Card_Phone VALUES ('M100103',68531898975);
167
168 --10 Pub_Phone Table
169 create table Pub_Phone (
170 Pub_ID varchar2(25) not null,
171 constraint fk_pub_id_foreign key(Pub_ID) references Publisher (Pub_ID) on delete cascade,
172 Phone number(4) not null
173 );
174
175 alter table Pub_Phone add constraint pub_phone_pk primary key (Pub_ID, Phone);
176
177 --Pub_Phone Table
178 INSERT INTO Pub_Phone VALUES('P100001',9989076587);
179 INSERT INTO Pub_Phone VALUES('P100002',9889076565);
180 INSERT INTO Pub_Phone VALUES('P100003',7455679345);
181 INSERT INTO Pub_Phone VALUES('P100004',8970862340);
182 INSERT INTO Pub_Phone VALUES('P100005',7756120238);
183 INSERT INTO Pub_Phone VALUES('P100001',6698751355);
184 INSERT INTO Pub_Phone VALUES('P100003',9764684687);
185 INSERT INTO Pub_Phone VALUES('P100005',9486366949);
186 /
187
```

Table created.

ORACLE Integrated Cloud Applications & Platform Services © 2020 Oracle Corporation - Privacy - Terms of Use Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ♥ using Oracle APEX

← → ↻ livesql.oracle.com/apex/?p=590:1:9360481596338:NO:RP::

Live SQL Feedback Help shailparikh2000@yahoo.com

Home SQL Worksheet Clear Find Actions Save Run

SQL Worksheet

Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

Table created.

1 row(s) inserted.

ORACLE Integrated Cloud Applications & Platform Services © 2020 Oracle Corporation - Privacy - Terms of Use Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ♥ using Oracle APEX

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

Step 2

--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);

The screenshot shows the Live SQL web interface. On the left is a navigation menu with options: 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 code editor with the following SQL script:

```

1  --CREATING TRIGGER
2
3  --FOR BOOK TABLE
4
5  CREATE OR REPLACE TRIGGER BOOK_YEAR_CHECK
6  BEFORE INSERT OR UPDATE ON Book
7  FOR EACH ROW
8  DECLARE
9      X NUMBER(4) := 0;
10 BEGIN
11     SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) INTO X FROM DUAL;
12     IF :NEW.Pub_Year > X THEN
13         RAISE_APPLICATION_ERROR(10001,'Publication Year Should Not be After The Present Year');
14     END IF;
15 END;
16 /
17
18
19 --INSERT INTO Book VALUES (1,'DBMS',2022);

```

Below the code editor, the execution results are shown. The first message is 'Trigger created.' followed by an 'Unsupported Command' error. The error details are:

```

ORA-21000: error number argument to raise_application_error of 10001 is out of range ORA-06512: at "SQL_XGHSDCNQSPZTROCKILGHISWZR.BOOK_YEAR_CHECK", line 6
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

At the bottom of the interface, there is a footer with Oracle and Integrated Cloud logos, copyright information (© 2020 Oracle Corporation), and links to Oracle Learning Library, Ask Tom, Dev Gym, and Database Doc.

--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);

← → ↻

livesql.oracle.com/apex/?p=590:19311936689681:NO:RP::

🔍 ☆ 🗑️ 📄 👤

Live SQL

Feedback ? Help 👤 shailparikh2000@yahoo.com

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

Clear Find Actions Save Run

```
1  --CREATING TRIGGER
2
3  --FOR BOOK_COPIES TABLE
4
5  CREATE OR REPLACE TRIGGER BOOK_COPIES_CHECK
6  BEFORE INSERT OR UPDATE ON Book_Copies
7  FOR EACH ROW
8
9  BEGIN
10     IF :NEW.No_of_Copies < 0 THEN
11         RAISE_APPLICATION_ERROR(-20002,'Number of Copies of a Book cannot be Negative!');
12     END IF;
13 END;
14 /
15
16 INSERT INTO Book_Copies VALUES ('1','2',-3);
```

Trigger created.

ORA-20002: Number of Copies of a Book cannot be Negative! ORA-06512: at "SQL_XGHSDCNQSSPZTROCKILGHSHZR.BOOK_COPIES_CHECK", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

ORACLE Integrated Cloud

© 2020 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ♥ using Oracle APEX

← → ↻

livesql.oracle.com/apex/?p=590:19311936689681:NO:RP::

🔍 ☆ 🗑️ 📄 👤

Live SQL

Feedback ? Help 👤 shailparikh2000@yahoo.com

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

Clear Find Actions Save Run

```
1  --CREATING TRIGGER
2
3  --FOR BOOK_COPIES TABLE
4
5  CREATE OR REPLACE TRIGGER BOOK_COPIES_CHECK
6  BEFORE INSERT OR UPDATE ON Book_Copies
7  FOR EACH ROW
8
9  BEGIN
10     IF :NEW.No_of_Copies < 0 THEN
11         RAISE_APPLICATION_ERROR(-20002,'Number of Copies of a Book cannot be Negative!');
12     END IF;
13 END;
14 /
15
16 INSERT INTO Book_Copies VALUES ('1','2',B);
```

Trigger created.

ORA-02291: integrity constraint (SQL_XGHSDCNQSSPZTROCKILGHSHZR.FK_BRANCH4_ID) violated - parent key not found ORA-06512: at "SYS.DBMS_SQL", line 1721

ORACLE Integrated Cloud

© 2020 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ♥ using Oracle APEX

*--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;

/

The screenshot shows the Live SQL web interface. The left sidebar contains 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 the following SQL code:

```
1  --CREATING TRIGGER
2
3  --FOR BOOK_LENDING
4
5  CREATE OR REPLACE TRIGGER CURRENT_DATE
6  BEFORE INSERT OR UPDATE ON Book_Lending
7  FOR EACH ROW
8
9  BEGIN
10     IF :NEW.Return_Date = NULL THEN
11         :NEW.Return_Date := SYSDATE;
12     END IF;
13 END;
14 /
15
```

Below the code editor, the output area shows the message 'Trigger created.' and an error message 'Unsupported Command' in red text.

The footer of the interface includes the Oracle logo, 'Integrated Cloud Applications & Platform Services', copyright information '© 2020 Oracle Corporation - Privacy - Terms of Use', and links to 'Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter'. It also states 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0' and 'Built with ❤️ using Oracle APEX'.

*--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;

/

← → ↻

livesql.oracle.com/apex/?p=590:1:9386575098266:NO:RP::

☆

≡

Live SQL

Feedback Help shailparikh2000@yahoo.com

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

Clear Find Actions Save Run

```
1  --CREATING TRIGGER
2
3  CREATE OR REPLACE TRIGGER EXPIRY_CHECK
4  BEFORE INSERT OR UPDATE ON Book_Lending
5  FOR EACH ROW
6
7  DECLARE
8      X DATE := SYSDATE;
9  BEGIN
10     SELECT Expiry_Date INTO X FROM Card WHERE Card_No = :NEW.Card_No;
11     IF :NEW.Due_Date > X THEN
12         :NEW.Due_Date := X;
13     RAISE_APPLICATION_ERROR(20001,'Your Due_Date has been changed.Kindly have a check');
14     END IF;
15 END;
16 /
```

Trigger created.

ORACLE

Integrated Cloud
Applications & Platform Services

© 2020 Oracle Corporation · Privacy · Terms of Use

Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c , 18c , 12c · Follow on Twitter

Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ❤️ using Oracle APEX

⬆

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

--CREATING PROCEDURES

SALIENT FEATURE

--PROCEDURE TO SEARCH A BOOK AT A PARTICULAR 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(  
    book1_id IN VARCHAR2,  
    branch1_id IN VARCHAR2,  
    X OUT NUMBER  
)  
IS  
BEGIN
```

```

X := 0;

SELECT No_of_Copies INTO X FROM Book_Copies
WHERE (Book_ID = book1_id AND Branch_ID = branch1_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(book1_id, branch1_id);

ELSE

    DBMS_OUTPUT.PUT_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;

/
```


livesql.oracle.com/apex/?p=590:1:15693765871135:NO:RP::

Live SQL

Feedback Help shailpankh2000@yahoo.com

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

Clear Find Actions Save Run

```

1  --CREATING PROCEDURES
2
3  CREATE OR REPLACE PROCEDURE BOOK_BRANCH_SEARCH(
4      book2_id IN VARCHAR2,
5      branch2_id IN VARCHAR2
6  )
7  IS
8      CURSOR CUR IS SELECT DISTINCT(Branch_ID) FROM Book_Copies WHERE Book_ID = book2_id;
9  BEGIN
10     FOR TEMP IN CUR LOOP
11         DBMS_OUTPUT.PUT_LINE(TEMP.Branch_ID);
12     END LOOP;
13 END;
14 /
15
16 CREATE OR REPLACE PROCEDURE BOOK_AVAILABILITY(
17     book1_id IN VARCHAR2,
18     branch1_id IN VARCHAR2,
19     X OUT NUMBER
20 )
21 IS
22 BEGIN
23     X := 0;
24     SELECT No_of_Copies INTO X FROM Book_Copies
25     WHERE (Book_ID = book1_id AND Branch_ID = branch1_id);
26     IF X = 0 THEN
27         DBMS_OUTPUT.PUT_LINE('Oops! No Copy of the Book available at this Branch. ');
28         DBMS_OUTPUT.PUT_LINE('You will find this Book at Branches listed below: ');
29         BOOK_BRANCH_SEARCH(book1_id, branch1_id);
30     ELSE
31         DBMS_OUTPUT.PUT_LINE('Yaay! Book is available here. ');
32     END IF;
33 END;
34 /
35
36 DECLARE
37     X NUMBER:=0;
38

```

Statement processed.
Oops! No Copy of the Book available at this Branch.

ORACLE Integrated Cloud
Applications & Platform Services

© 2020 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with using Oracle APEX

livesql.oracle.com/apex/?p=590:1:15693765871135:NO:RP::

Live SQL

Feedback Help shailpankh2000@yahoo.com

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

Clear Find Actions Save Run

```

12 END LOOP;
13 END;
14 /
15
16 CREATE OR REPLACE PROCEDURE BOOK_AVAILABILITY(
17     book1_id IN VARCHAR2,
18     branch1_id IN VARCHAR2,
19     X OUT NUMBER
20 )
21 IS
22 BEGIN
23     X := 0;
24     SELECT No_of_Copies INTO X FROM Book_Copies
25     WHERE (Book_ID = book1_id AND Branch_ID = branch1_id);
26     IF X = 0 THEN
27         DBMS_OUTPUT.PUT_LINE('Oops! No Copy of the Book available at this Branch. ');
28         DBMS_OUTPUT.PUT_LINE('You will find this Book at Branches listed below: ');
29         BOOK_BRANCH_SEARCH(book1_id, branch1_id);
30     ELSE
31         DBMS_OUTPUT.PUT_LINE('Yaay! Book is available here. ');
32     END IF;
33 END;
34 /
35
36 DECLARE
37     X NUMBER:=0;
38 BEGIN
39     BOOK_AVAILABILITY('B100001','LB100001',X);
40     DBMS_OUTPUT.PUT_LINE(X || ' Copies are available at this Branch');
41 END;
42 /

```

Statement processed.
Oops! No Copy of the Book available at this Branch.
You will find this Book at Branches listed below:
LB100001
LB200001
LB500001
0 Copies are available at this Branch

ORACLE Integrated Cloud
Applications & Platform Services

© 2020 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with using Oracle APEX

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.PUT_LINE('TOTAL FINE IS: ' || X);

END;

/

The screenshot shows the Live SQL web interface. On the left is a navigation menu with options: 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 text editor with the following SQL code:

```

1  --CREATING FUNCTION
2
3  CREATE OR REPLACE FUNCTION COUNT_FINE
4  RETURN NUMBER
5  IS
6
7      X NUMBER := 0;
8      CARD3_NO VARCHAR2(20) := 'M100101';
9      BOOK3_ID VARCHAR2(15) := 'B100004';
10
11  BEGIN
12      SELECT Return_Date - Due_Date INTO X FROM Book_Lending
13      WHERE Card_No = CARD3_NO AND Book_ID = BOOK3_ID;
14      IF X > 0 THEN
15          RETURN X;
16      ELSE
17          RETURN 0;
18      END IF;
19  END;
20 /
21
22 DECLARE
23     X NUMBER := 0;
24 BEGIN
25     X := COUNT_FINE();
26     DBMS_OUTPUT.PUT_LINE('TOTAL FINE IS: ' || X);
27 END;
28 /

```

Below the code editor, the execution results are displayed:

```

Function created.

Statement processed.
TOTAL FINE IS: 10

```

At the bottom of the interface, there is a footer with Oracle branding and version information: 'ORACLE Integrated Cloud Applications & Platform Services', '© 2020 Oracle Corporation · Privacy · Terms of Use', 'Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c, 18c, 12c · Follow on Twitter', 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0', and 'Built with ❤️ using Oracle APEX'.

--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;

/

```

← → ↻ livesql.oracle.com/apex/?p=590:19386575098266:NO:RP::

Live SQL

Feedback Help shailparikh2000@yahoo.com

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

Clear Find Actions Save Run

```

1  --CREATING FUNCTION
2
3  CREATE OR REPLACE FUNCTION TOTAL_FINE
4  RETURN NUMBER
5  IS
6      CARD4_NO VARCHAR2(20) := 'M100101';
7      CURSOR CUR IS SELECT Book_ID FROM Book_Lending WHERE Card_No = CARD4_NO;
8      X NUMBER:=0;
9      TOT NUMBER:=0;
10 BEGIN
11     FOR TEMP IN CUR LOOP
12         COUNT2_FINE(CARD4_NO, TEMP.Book_id, X);
13         TOT := X + TOT;
14     END LOOP;
15     RETURN TOT;
16 END;
17 /
18
19 CREATE OR REPLACE PROCEDURE COUNT2_FINE(
20     CARD32_NO IN VARCHAR2,
21     BOOK32_ID IN VARCHAR2,
22     X OUT NUMBER
23 )
24 IS
25 BEGIN
26     SELECT Return_Date - Due_Date INTO X FROM Book_Lending
27     WHERE Card_No = CARD32_NO AND Book_ID = BOOK32_ID;
28     IF X <= 0 THEN
29         X := 0;
30     END IF;
31 END;
32 /
33
34
35 CREATE OR REPLACE FUNCTION COUNT_FINE
36 RETURN NUMBER
37 IS

```

Statement processed.
TOTAL SINGLE FINE IS: 10

ORACLE Integrated Cloud
Applications & Platform Services

© 2020 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c, 18c, 12c · Follow on Twitter
Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ❤️ using Oracle APEX

← → ↻ livesql.oracle.com/apex/?p=590:19386575098266:NO:RP::

Live SQL

Feedback Help shailparikh2000@yahoo.com

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

Clear Find Actions Save Run

```

36 CREATE OR REPLACE FUNCTION COUNT_FINE
37 RETURN NUMBER
38 IS
39     X NUMBER := 0;
40     CARD3_NO VARCHAR2(20) := 'M100101';
41     BOOK3_ID VARCHAR2(15) := 'B100004';
42 BEGIN
43     SELECT Return_Date - Due_Date INTO X FROM Book_Lending
44     WHERE Card_No = CARD3_NO AND Book_ID = BOOK3_ID;
45     IF X > 0 THEN
46         RETURN X;
47     ELSE
48         RETURN 0;
49     END IF;
50 END;
51 /
52
53 DECLARE
54     X NUMBER := 0;
55     TOT NUMBER := 0;
56 BEGIN
57     X := COUNT_FINE();
58     DBMS_OUTPUT.PUT_LINE('TOTAL SINGLE FINE IS: ' || X);
59     TOT := TOTAL_FINE();
60     DBMS_OUTPUT.PUT_LINE('TOTAL FINE IS: ' || TOT);
61 END;
62 /

```

Function created.

Procedure created.

Function created.

Statement processed.
TOTAL SINGLE FINE IS: 10
TOTAL FINE IS: 20

ORACLE Integrated Cloud
Applications & Platform Services

© 2020 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c, 18c, 12c · Follow on Twitter
Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ❤️ using Oracle APEX

LIBRARY SYSTEM

MANAGEMENT AND DATABASE
HANDLING

Physical Designing

DEFINING VIEWS:

Step 4

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

--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);
```

The screenshot shows the Oracle Live SQL web interface. The left sidebar contains 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 the following SQL code:

```
1 --CREATING VIEWS  
2 --DROP VIEW BOOK_DETAILS;  
3  
4 CREATE VIEW BOOK_DETAILS(BOOK_ID, TITLE, PUB_YEAR, AUTHOR_NAME, PUB_NAME)  
5 AS SELECT Book_ID, Book.Title, Book.Pub_Year, Book_Authors.Author_Name, Publisher.Name  
6 FROM (Book NATURAL JOIN Book_Authors) NATURAL JOIN (Published_by NATURAL JOIN Publisher);  
7  
8 SELECT * FROM BOOK_DETAILS;
```

Below the code editor, a message states 'View created.' followed by a table of results. The table has 5 columns: BOOK_ID, TITLE, PUB_YEAR, AUTHOR_NAME, and PUB_NAME. It contains 11 rows of data. Below the table, a link 'Download CSV' is visible, along with the text '11 rows selected.'.

BOOK_ID	TITLE	PUB_YEAR	AUTHOR_NAME	PUB_NAME
B100001	DBMS	2017	EDWARD ANGEL	MCGRAW-HILL
B100001	DBMS	2017	NAVATHE	MCGRAW-HILL
B100001	DBMS	2017	EDWARD ANGEL	PEARSON
B100001	DBMS	2017	NAVATHE	PEARSON
B100002	ADBMS	2016	NAVATHE	PEARSON
B100003	CN	2016	EDWARD ANGEL	RANDOM HOUSE
B100003	CN	2016	TANENBAUM	RANDOM HOUSE
B100004	CG	2015	TANENBAUM	HACHETTE LIVRE
B100003	CN	2016	EDWARD ANGEL	GRUPO PLANETA
B100003	CN	2016	TANENBAUM	GRUPO PLANETA
B100005	OS	2016	GALVIN	GRUPO PLANETA

At the bottom of the page, the Oracle logo and 'Integrated Cloud Applications & Platform Services' are displayed. Copyright information for 2020 Oracle Corporation is provided, along with links to the Oracle Learning Library, Ask Tom, Dev Gym, and Database Doc. The footer also mentions 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0' and 'Built with ♥ using Oracle APEX'.

--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);
```

The screenshot shows the Live SQL web interface. The left sidebar contains 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 the following SQL code:

```
1 --DROP VIEW PUBLISHER_DETAILS;  
2  
3 CREATE VIEW PUBLISHER_DETAILS(PUB_ID, PUB_NAME, ADDRESS, PHONE)  
4 AS SELECT PUB_ID, Publisher.Name, Publisher.Address, Pub_Phone.Phone  
5 FROM (Publisher NATURAL JOIN Pub_Phone);  
6  
7 SELECT * FROM PUBLISHER_DETAILS;
```

Below the code editor, the results of the last query are displayed in a table:

PUB_ID	PUB_NAME	ADDRESS	PHONE
P100001	MCGRAN-HILL	BANGALORE	6690751355
P100001	MCGRAN-HILL	BANGALORE	9989076587
P100002	PEARSON	NEW DELHI	9889076565
P100003	RANDOM HOUSE	HYDERABAD	7455679345
P100003	RANDOM HOUSE	HYDERABAD	9764684687
P100004	HACHETTE LIVRE	CHENNAI	8970862340
P100005	GRUPO PLANETA	BANGALORE	7756120238
P100005	GRUPO PLANETA	BANGALORE	9486366949

Below the table, it says "Download CSV" and "8 rows selected." The footer of the interface includes the Oracle logo, "Integrated Cloud Applications & Platform Services", copyright information "© 2020 Oracle Corporation · Privacy · Terms of Use", and links to "Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c, 18c, 12c · Follow on Twitter". It also states "Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0" and "Built with ❤️ using Oracle APEX".

--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);
```

The screenshot shows the Live SQL web interface. The left sidebar contains 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 the following SQL code:

```
1 --DROP VIEW CARD_DETAILS;  
2  
3 CREATE VIEW CARD_DETAILS(CARD_NO, MEMBER_NAME, ADDRESS, PHONE, EXPIRY_DATE)  
4 AS SELECT Card_No, Card.Name, Card.Address, Card_Phone.Phone, Card.Expiry_Date  
5 FROM (Card NATURAL JOIN Card_Phone);  
6  
7 SELECT * FROM CARD_DETAILS;
```

Below the code editor, a message states 'View created.' followed by a table of results. The table has 5 columns: CARD_NO, MEMBER_NAME, ADDRESS, PHONE, and EXPIRY_DATE. It contains 9 rows of data. Below the table, it says 'Download CSV' and '9 rows selected.'

CARD_NO	MEMBER_NAME	ADDRESS	PHONE	EXPIRY_DATE
M100100	CHANDLER	NEW DELHI	9581117955	25-AUG-20
M100100	CHANDLER	NEW DELHI	9760613836	25-AUG-20
M100101	ROSS	BANGALORE	8465317649	14-NOV-20
M100102	OTIS	CHENNAI	6486121616	27-NOV-20
M100102	OTIS	CHENNAI	9846161313	27-NOV-20
M100103	MAEVE	HYDERABAD	8765618798	10-JAN-21
M100103	MAEVE	HYDERABAD	68531898975	10-JAN-21
M100104	JOEY	BANGALORE	89161616178	31-DEC-20
M100104	JOEY	BANGALORE	94618489713	31-DEC-20

At the bottom of the interface, there is a footer with the Oracle logo, 'Integrated Cloud Applications & Platform Services', copyright information for 2020 Oracle Corporation, and links to the Oracle Learning Library, Ask Tom, Dev Gym, Database Doc, and Twitter. It also mentions 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0' and 'Built with ❤️ using Oracle APEX'.

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;
```

The screenshot shows the Live SQL web application interface. The top navigation bar includes a hamburger menu, the 'Live SQL' logo, and links for Feedback, Help, and a user profile. The left sidebar contains navigation options: 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 text editor with the following SQL query:

```
1 --JOINS  
2  
3 --LEFT OUTER JOIN  
4 SELECT * FROM Book LEFT JOIN Book_Authors ON Book.Book_ID = Book_Authors.Book_ID;
```

Below the editor, the query results are displayed in a table with 5 columns: BOOK_ID, TITLE, PUB_YEAR, BOOK_ID, and AUTHOR_NAME. The table contains 7 rows of data. Below the table, there is a 'Download CSV' link and a message '7 rows selected.'.

BOOK_ID	TITLE	PUB_YEAR	BOOK_ID	AUTHOR_NAME
B100001	DBMS	2017	B100001	EDWARD ANGEL
B100001	DBMS	2017	B100001	NAVATHE
B100002	ADBMS	2016	B100002	NAVATHE
B100003	CN	2016	B100003	EDWARD ANGEL
B100003	CN	2016	B100003	TANENBAUM
B100004	CG	2015	B100004	TANENBAUM
B100005	OS	2016	B100005	GALVIN

At the bottom of the page, there is a footer with the Oracle logo, 'Integrated Cloud Applications & Platform Services', copyright information for 2020 Oracle Corporation, and links to Oracle Learning Library, Ask Tom, Dev Gym, Database Doc, and Twitter. It also mentions 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0' and 'Built with ❤️ using Oracle APEX'.

--RIGHT OUTER JOIN

```
SELECT * FROM Book RIGHT JOIN Book_Authors ON  
Book.Book_ID = Book_Authors.Book_ID;
```

The screenshot shows the Live SQL web application interface. The top navigation bar includes a home icon, the 'Live SQL' logo, and links for Feedback, Help, and a user profile. A left sidebar contains navigation options: 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 text editor with the following SQL query:

```
1 -- JOINS  
2  
3 --RIGHT OUTER JOIN  
4 SELECT * FROM Book RIGHT JOIN Book_Authors ON Book.Book_ID = Book_Authors.Book_ID;
```

Below the editor, the query results are displayed as a table with 7 rows and 5 columns:

BOOK_ID	TITLE	PUB_YEAR	BOOK_ID	AUTHOR_NAME
B100001	DBMS	2017	B100001	EDWARD ANGEL
B100001	DBMS	2017	B100001	NAVATHE
B100002	ADBMS	2016	B100002	NAVATHE
B100003	CN	2016	B100003	EDWARD ANGEL
B100003	CN	2016	B100003	TANENBAUM
B100004	CG	2015	B100004	TANENBAUM
B100005	OS	2016	B100005	GALVIN

Below the table, there is a 'Download CSV' link and a message '7 rows selected.' The footer contains the Oracle logo, 'Integrated Cloud Applications & Platform Services', copyright information for 2020 Oracle Corporation, and links to the Oracle Learning Library, Ask Tom, Dev Gym, and Database Doc. It also mentions 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0' and 'Built with ❤️ using Oracle APEX'.

--FULL OUTER JOIN

```
SELECT * FROM Book FULL JOIN Book_Authors ON  
Book.Book_ID = Book_Authors.Book_ID;
```

The screenshot shows the Oracle Live SQL web interface. The top navigation bar includes 'Home', 'SQL Worksheet', 'My Session', 'Schema', 'Quick SQL', 'My Scripts', 'My Tutorials', and 'Code Library'. The 'SQL Worksheet' section is active, displaying a SQL query in a text editor. The query is: `SELECT * FROM Book FULL JOIN Book_Authors ON Book.Book_ID = Book_Authors.Book_ID;`. Below the editor, the results of the query are displayed as a table with 5 columns: BOOK_ID, TITLE, PUB_YEAR, BOOK_ID, and AUTHOR_NAME. The table contains 7 rows of data. At the bottom of the results section, there is a 'Download CSV' link and a message '7 rows selected.'.

BOOK_ID	TITLE	PUB_YEAR	BOOK_ID	AUTHOR_NAME
B100001	DBMS	2017	B100001	EDWARD ANGEL
B100001	DBMS	2017	B100001	NAVATHE
B100002	ADBMS	2016	B100002	NAVATHE
B100003	CN	2016	B100003	EDWARD ANGEL
B100003	CN	2016	B100003	TANENBAUM
B100004	CG	2015	B100004	TANENBAUM
B100005	OS	2016	B100005	GALVIN

Download CSV
7 rows selected.

© 2020 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c, 18c, 12c · Follow on Twitter
Live SQL 202.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0 Built with ❤️ using Oracle APEX

--NATURAL JOIN

```
SELECT * FROM (Book NATURAL JOIN Book_Authors)
NATURAL JOIN (Publisher NATURAL JOIN Published_by);
```

The screenshot shows the Live SQL web application interface. The top navigation bar includes a 'Home' button and a 'Live SQL' logo. The left sidebar contains a menu with '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 text editor with the following SQL query:

```
-- JOINS
-- NATURAL JOIN
SELECT * FROM (Book NATURAL JOIN Book_Authors) NATURAL JOIN (Publisher NATURAL JOIN Published_by);
```

Below the editor, the results of the query are displayed as a table with 7 columns: BOOK_ID, TITLE, PUB_YEAR, AUTHOR_NAME, PUB_ID, NAME, and ADDRESS. The table contains 11 rows of data. Below the table, there is a 'Download CSV' link and a message '11 rows selected.'.

BOOK_ID	TITLE	PUB_YEAR	AUTHOR_NAME	PUB_ID	NAME	ADDRESS
B100001	DBMS	2017	EDWARD ANGEL	P100001	MCGRAW-HILL	BANGALORE
B100001	DBMS	2017	NAVATHE	P100001	MCGRAW-HILL	BANGALORE
B100001	DBMS	2017	EDWARD ANGEL	P100002	PEARSON	NEW DELHI
B100001	DBMS	2017	NAVATHE	P100002	PEARSON	NEW DELHI
B100002	ADBMS	2016	NAVATHE	P100002	PEARSON	NEW DELHI
B100003	CN	2016	EDWARD ANGEL	P100003	RANDOM HOUSE	HYDERABAD
B100003	CN	2016	TANENBAUM	P100003	RANDOM HOUSE	HYDERABAD
B100004	CG	2015	TANENBAUM	P100004	HACHETTE LIVRE	CHENNAI
B100003	CN	2016	EDWARD ANGEL	P100005	GRUPO PLANETA	BANGALORE
B100003	CN	2016	TANENBAUM	P100005	GRUPO PLANETA	BANGALORE
B100005	OS	2016	GALVIN	P100005	GRUPO PLANETA	BANGALORE

At the bottom of the page, there is a footer with the Oracle logo, 'Integrated Cloud' text, and copyright information: '© 2020 Oracle Corporation · Privacy · Terms of Use'. It also includes links to 'Oracle Learning Library', 'Ask Tom', 'Dev Gym', and 'Database Doc 19c, 18c, 12c', and a Twitter link. The footer also mentions 'Live SQL 20.2.1, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0' and 'Built with ❤️ using Oracle APEX'.

END