

Library Management System - Part - 4

(Queries, updates and deletes)

Updations made for Part - 4:

Below are the queries for Create and Insert statements, which include all global and fragmented inserts. Highlighted are essential new changes on the tables. Updated the inserts as per the new table changes as well.

Tables:

```
create table Author(
    Auth_ID NUMBER(8) NOT NULL,
    Auth_name VARCHAR2(25),
    Qualification VARCHAR2(50),
    PRIMARY KEY (Auth_ID)
);

create table Publisher(
    Pub_code NUMBER(8) NOT NULL,
    Pub_name VARCHAR2(25) NOT NULL,
    Pub_country VARCHAR2(50),
    Year_of_publication NUMBER(4) check (Year_of_publication >= 1440),
    PRIMARY KEY (Pub_code)
);

create table Vendor(
    Ven_code NUMBER(8) NOT NULL,
    Ven_name VARCHAR2(25),
    Ven_phone NUMBER(20) NOT NULL,
    PRIMARY KEY (Ven_code)
);

create table Library(
    Lib_code NUMBER(8) NOT NULL,
    Lib_name VARCHAR2(25),
    Library_street VARCHAR2(40) NOT NULL,
    Library_city VARCHAR2(15) NOT NULL,
    Library_State VARCHAR2(15) NOT NULL,
    Library_country VARCHAR2(40),
```

```

        Library_zip_code VARCHAR2(9),
        PRIMARY KEY (Lib_code)
    );

create table Administrator(
    Admin_ID NUMBER(8) NOT NULL,
    Admin_name VARCHAR2(25),
    Admin_phone NUMBER(20) NOT NULL UNIQUE,
    PRIMARY KEY (Admin_ID)
);

create table Users(
    User_ID NUMBER(8) NOT NULL,
    User_name VARCHAR2(25),
    User_contact NUMBER(20) NOT NULL UNIQUE,
    User_street VARCHAR2(40) NOT NULL,
    User_city VARCHAR2(15) NOT NULL,
    User_State VARCHAR2(15) NOT NULL,
    User_country VARCHAR2(40),
    User_zip_code VARCHAR2(9),
    Late_Factor NUMBER(2),
    Lost_books NUMBER(2),
    PRIMARY KEY (User_ID)
);

create table Staff(
    Emp_ID NUMBER(8) NOT NULL,
    Emp_name VARCHAR2(25),
    Emp_contact NUMBER(20) NOT NULL,
    Salary NUMBER(10,3) check (Salary > 25 and Salary < 999999999),
    Date_of_employment DATE NOT NULL,
    Designation VARCHAR2(25),
    Emp_street VARCHAR2(40) NOT NULL,
    Emp_city VARCHAR2(15) NOT NULL,
    Emp_state VARCHAR2(15) NOT NULL,
    Emp_country VARCHAR2(40),
    Emp_zip_code VARCHAR2(9),
    Admin_ID NUMBER(8),
    PRIMARY KEY (Emp_ID),
    FOREIGN KEY (Admin_ID) REFERENCES Administrator(Admin_ID)
);

```

```

create table Sponsor(
    Sponsor_ID NUMBER(8) NOT NULL,
    Lib_code NUMBER(8) NOT NULL,
    Percentage NUMBER(3, 2) check (Percentage > 0),
    Govt_name VARCHAR2(25),
    Period VARCHAR2(15),
    PRIMARY KEY (Sponsor_ID),
    FOREIGN KEY (Lib_code) REFERENCES Library(Lib_code)
);

```

```

create table Books(
    Book_ID NUMBER(8) NOT NULL,
    Ven_code NUMBER(8),
    Pub_code NUMBER(8),
    Auth_ID NUMBER(8),
    Lib_code NUMBER(8),
    Book_name VARCHAR2(85),
    Book_genre VARCHAR2(70),
    Price NUMBER(6, 2) check (Price > 0),
    No_of_books NUMBER(3) check (No_of_books >= 0),
    Status VARCHAR2(20) check (Status in ('available','reserved','NA')),
    Return_date DATE,
    Reserve_date DATE,
    Book_location VARCHAR2(50),
    PRIMARY KEY (Book_ID),
    FOREIGN KEY (Lib_code) REFERENCES Library(Lib_code),
    FOREIGN KEY (Ven_code) REFERENCES Vendor(Ven_code),
    FOREIGN KEY (Pub_code) REFERENCES Publisher(Pub_code),
    FOREIGN KEY (Auth_ID) REFERENCES Author(Auth_ID)
);

```

Data Insertion:

```

insert into Author values (90011, 'Leo Tolstoy', 'law and oriental languages');
Insert into Author values (90012, 'Tarantino', 'Cinema');
Insert into Author values (90013, 'Hunter Thompson', 'Journalism');
Insert into Author values (90014, 'Burt Macklin', 'Crime');
Insert into Author values (90015, 'William Shakesphere', 'Playwright');
Insert into Author values (90016, 'Bruce Wayne', 'Crime');
Insert into Author values (90017, 'Frank Herbert', 'SciFi');
Insert into Author values (90018, 'George Orwell', 'Social');
Insert into Author values (90019, 'Tolkein', 'Literature');
Insert into Author values(90020, 'Mark Twain', 'Literature');

```

Insert into Author values(90030, 'Martin Chatwin', 'Cookery');

insert into Publisher values (80011, 'The Russian Messenger', 'Russia', 1869);

Insert into Publisher values (80012, 'Penguin', 'UK', 1800);

Insert into Publisher values (80013, 'harpercollins', 'US', 1899);

Insert into Publisher values (80014, 'Macmillan', 'UK', 1843);

Insert into Publisher values (80015, 'Hachette', 'Europe', 1992);

Insert into Publisher values (80016, 'Scholastic', 'India', 1989);

Insert into Publisher values (80017, 'Pearson', 'US', 1901);

Insert into Publisher values (80018, 'Mcgraw Hill', 'US', 1900);

Insert into Publisher values (80019, 'Pearson', 'UK', 1954);

Insert into Publisher values (80020, 'Random', 'UK', 1984);

Insert into Publisher values (80030, 'Cook Books', 'Italy', 1992);

insert into Vendor values (70011, 'Victor', 1234567890);

insert into Vendor values (70012, 'Huey', 1236540987);

insert into Vendor values (70013, 'Duey', 4561237890);

insert into Vendor values (70014, 'Humpty', 1237894560);

insert into Vendor values (70015, 'Dumpty', 3214569999);

insert into Vendor values (70016, 'Deshaun', 9998431234);

insert into Vendor values (70017, 'The Rock', 9408438430);

insert into Vendor values (70018, 'Vlad', 9408436969);

insert into Vendor values (70019, 'Chad', 4086904200);

insert into Vendor values (70020, 'Jennifer', 4567891234);

insert into Vendor values (70030, 'Penny', 9877891234);

insert into Library values (20011, 'Sycamore', 'Sycamore St', 'San Antonio', 'Texas', 'US', '78015');

insert into Library values (20012, 'Willis', 'W Prairie St', 'Denton', 'Texas', 'US', '76201');

insert into Library values (20013, 'Discovery Park', 'N Elm St', 'Denton', 'Texas', 'US', '76201');

insert into Library values (20014, 'Library Of Alexandria', 'Ptolemy St', 'Alexandria', 'Athens', 'Greece', '11001');

insert into Library values (20015, 'Library of Congress', 'President St', 'Washington', 'DC', 'US', '11007');

insert into Library values (20016, 'Bodleian', 'Tea St', 'Oxford', 'London', 'UK', '34808');

insert into Library values (20017, 'Yale', 'Teasley St', 'New Haven', 'Connecticut', 'US', '30240');

insert into Library values (20018, 'Vatican', 'Pope Ln', 'Square', 'Vatican City', 'Rome', '01066');

insert into Library values (20019, 'Boston Public', 'Oak Street', 'Boston', 'Massachusetts', 'US', '69020');

insert into Library values (20020, 'Seattle Central', 'Fry St', 'Seattle', 'Washington', 'US', '42069');

insert into Library values (20030, 'San Jose Live', 'Southern St', 'Prairie', 'Italy', 'EUR', '98142');

Update Library

set LIBRARY_STREET='Oak Street',

LIBRARY_CITY='Boston' ,

```
LIBRARY_STATE='Massachusetts' ,  
LIBRARY_COUNTRY='US' ,  
LIBRARY_ZIP_CODE='69020',  
Lib_name='Boston Public'  
where Lib_code=20016;
```

```
insert into Sponsor values (10011, 20011, 5, 'govt_xx', '2006-2008');  
Insert into Sponsor values (10012, 20012, 7, 'govt_x1', '1993-1999');  
Insert into Sponsor values (10013, 20013, 8, 'govt_x3', '2020-2030');  
Insert into Sponsor values (10014, 20014, 4.5, 'govt_z', '3000-3033');  
Insert into Sponsor values (10015, 20015, 1.1, 'govt_u', '1939-1945');  
Insert into Sponsor values (10016, 20016, 2.5, 'govt_s', '1914-1918');  
Insert into Sponsor values (10017, 20017, 5.6, 'govt_r', '1888-1898');  
Insert into Sponsor values (10018, 20018, 7.3, 'govt_tt', '2010-2015');  
Insert into Sponsor values (10019, 20019, 6, 'govt_xy', '2015-2018');  
Insert into Sponsor values (10020, 20020, 5, 'govt_xz', '2018-2023');
```

```
insert into Books values (50011, 70011, 80011, 90011, 20011, 'War and Peace, first edition',  
'literary', 15.03, 12, 'available', 'FL_02 RACK33');  
insert into Books values (50012, 70012, 80012, 90012, 20012, 'Hells Angels', 'mystery', 19.99, 3,  
'available', 'Fl_01 Aisle 4');  
insert into Books values (50013, 70013, 80013, 90013, 20013, 'Kitchen Confidential',  
'education', 21.99, 1, 'reserved', 'Fl_01 Aisle 2');  
insert into Books values (50014, 70014, 80014, 90014, 20014, 'Dune 4th', 'fiction', 10.99, 2,  
'available', 'Fl_11 Aisle 6');  
insert into Books values (50015, 70015, 80015, 90015, 20015, 'Programming for Dummies',  
'education', 5.99, 0, 'reserved', 'Fl_01 Aisle 3');  
insert into Books values (50016, 70016, 80016, 90016, 20016, 'College Textbook', 'education',  
999.89, 1, 'reserved', 'Fl_00 Aisle1');  
insert into Books values (50017, 70017, 80017, 90017, 20017, 'Dune Messiah', 'fiction', 10.99, 10,  
'available', 'Fl_10 Aisle 20');  
insert into Books values (50018, 70018, 80018, 90018, 20018, 'College Book 2', 'education',  
1999.99, 0, 'reserved', 'Fl_01 Aisle 1');  
insert into Books values (50019, 70019, 80019, 90019, 20019, 'Murder Handbook', 'mystery', 1.99,  
10, 'available', 'Fl_99 Aisle 2');  
insert into Books values (50020, 70020, 80020, 90020, 20020, 'Cooking Meth', 'education', 59.99, 1,  
'available', 'Fl_69 Aisle 3');  
insert into Books values (50030, 70030, 80030, 90030, 20030, 'Chef Skills', 'education', 68, 5,  
'available', 'Fl_2 Aisle 9');  
insert into Books values (50031, 70018, 80018, 90018, 20018, 'Naughtiest Girl 1', 'Novel', 26,  
33, 'available', 'Fl_0 Aisle 9');
```

```
insert into Administrator values (30011, 'A. Alex', 0110981234);
```

```
insert into Administrator values (30012, 'B. Alex', 0120981234);
insert into Administrator values (30013, 'C. Alex', 0130981234);
insert into Administrator values (30014, 'D. Alex', 0140981234);
insert into Administrator values (30015, 'E. Alex', 0150981234);
insert into Administrator values (30016, 'F. Alex', 0160981234);
insert into Administrator values (30017, 'G. Alex', 0170981234);
insert into Administrator values (30018, 'H. Alex', 0180981234);
insert into Administrator values (30019, 'I. Alex', 0190981234);
insert into Administrator values (30020, 'J. Alex', 0200981234);

insert into Users values (40011, 'K. Christina', 3426757979, 'W. Oak', 'San Antonio', 'Texas', 'US', '78015', 2, 14);
insert into Users values (40012, 'Kirk', 7896757979, 'A. Oak', 'San Francisco', 'California', 'US', '76207', 3, 2);
insert into Users values (40013, 'Mary', 0136757979, 'A. Oak', 'San Francisco', 'California', 'US', '76207', 5, 9);
insert into Users values (40014, 'Rosy', 0146757979, 'A. Oak', 'San Francisco', 'California', 'US', '76207', 1, 5);
insert into Users values (40015, 'Chris', 0156757979, 'A. Oak', 'San Francisco', 'California', 'US', '76207', 4, 15);
insert into Users values (40016, 'Tracy', 0166757979, 'Fry St', 'Galveston', 'Texas', 'US', '76207', 8, 11);
insert into Users values (40017, 'Bruce', 0176757979, 'Bleeker St', 'Gotham City', 'Michigan', 'US', '76207', 2, 5);
insert into Users values (40018, 'Garg', 0186757979, '10 Downing St', 'Boston', 'Massachusetts', 'US', '76207', 4, 16);
insert into Users values (40019, 'Henry', 0196757979, 'A. Oak', 'Houston', 'Texas', 'US', '76207', 8, 12);
insert into Users values (40020, 'Feng', 0206757979, 'JFK st', 'Queens', 'New York', 'US', '76207', 6, 8);
insert into Users values (60018, 'Charles lokey', 0182344545, 'Avenue A', 'New Jersey', 'Texas', 'US', '78015', 6, 8);
insert into Users values (60021, 'De Caprio', 1202344545, 'Avenue A', 'California', 'California', 'US', '79015', 6, 8);
insert into Users values (60022, 'Ahmed', 9872344545, 'Ave G', 'Denton', 'Texas', 'US', '76201', 6, 8);


```

```
insert into Staff values (60011, 'A. Cooper', 0112344545, 12345, TO_DATE('10/10/21', 'mm/dd/yy'), 'librarian', 'Avenue A', 'San Antonio', 'Texas', 'US', '78015', 30011);
insert into Staff values (60012, 'James Lovey', 0122344545, 35000, TO_DATE('03/10/21', 'mm/dd/yy'), 'Manager', 'Avenue A', 'san francisco', 'Texas', 'US', '78015', 30012);
insert into Staff values (60013, 'Alexander', 0132344545, 24000, TO_DATE('10/10/21', 'mm/dd/yy'), 'lead', 'Avenue A', 'Houston', 'Texas', 'US', '78015', 30013);
```

```

insert into Staff values (60014, 'Roopa K', 0142344545, 40078, TO_DATE('10/09/21','mm/dd/yy'),
'Manager', 'Avenue A', 'San Antonio', 'Texas', 'US', '78015', 30014);
insert into Staff values (60015, 'Kim Jim', 0152344545, 67899, TO_DATE('10/02/22','mm/dd/yy'),
'Manager', 'Avenue A', 'san francisco', 'Texas', 'US', '78015', 30015);
insert into Staff values (60016, 'Fiaz', 0162344545, 34567, TO_DATE('09/10/21','mm/dd/yy'),
'Manager', 'Avenue A', 'Houston', 'Texas', 'US', '78015', 30016);
insert into Staff values (60017, 'Tom Jom', 0172344545, 24567, TO_DATE('08/10/21','mm/dd/yy'),
'librarian', 'Avenue A', 'New Jersey', 'Texas', 'US', '78015', 30017);
insert into Staff values (60018, 'Charles lokey', 0182344545, 56766,
TO_DATE('08/10/21','mm/dd/yy'), 'Manager', 'Avenue A', 'New Jersey', 'Texas', 'US', '78015',
30018);
insert into Staff values (60019, 'Riveria', 0192344545, 23456, TO_DATE('09/10/21','mm/dd/yy'),
'librarian', 'Avenue A', 'California', 'Texas', 'US', '78015', 30019);
insert into Staff values (60020, 'Lionel cab', 0202344545, 20778,
TO_DATE('10/28/21','mm/dd/yy'), 'librarian', 'Avenue A', 'California', 'Texas', 'US', '78015', 30020);
insert into Staff values (60021, 'De Caprio', 1202344545, 67078,
TO_DATE('11/28/22','mm/dd/yy'), 'Director', 'Avenue A', 'California', 'California', 'US', '79015',
30020);

```

Relation Tables and Data Insertion:

```

create table MaintainsTrackOf(
    Emp_ID NUMBER(8),
    User_ID NUMBER(8),
    PRIMARY KEY (Emp_ID, User_ID)
);

```

```

create table Reserve(
    Book_ID NUMBER(8),
    User_ID NUMBER(8),
    Reserve_date DATE,
    PRIMARY KEY (Book_ID, User_ID)
);

```

```

create table Return(
    Book_ID NUMBER(8),
    User_ID NUMBER(8),
    Expected_Return_date DATE,
    Actual_Return_date DATE,
    Late NUMBER(1),
    PRIMARY KEY (Book_ID, User_ID)
);

```

```

CREATE OR REPLACE TRIGGER TRG_INCREMENT_LATE_FACTOR
AFTER INSERT ON RETURN
FOR EACH ROW
WHEN (NEW.Late = 1)
BEGIN
    UPDATE Users
    SET Late_Factor = Late_Factor + 1
    WHERE User_ID = :NEW.User_ID;
END;
/

```

```

Insert into MaintainsTrackOf values (60011,40011);
Insert into MaintainsTrackOf values (60011,40012);
Insert into MaintainsTrackOf values (60019,40013);
Insert into MaintainsTrackOf values (60019,40014);
Insert into MaintainsTrackOf values (60017,40015);
Insert into MaintainsTrackOf values (60017,40016);
Insert into MaintainsTrackOf values (60012,40017);
Insert into MaintainsTrackOf values (60013,40018);
Insert into MaintainsTrackOf values (60014,40019);
Insert into MaintainsTrackOf values (60015,40020);

```

```

Insert into Reserve values (50011,40011,TO_DATE('11/20/22','mm/dd/yy'));
Insert into Reserve values (50012,40011,TO_DATE('10/11/22','mm/dd/yy'));
Insert into Reserve values (50013,40011,TO_DATE('01/23/23','mm/dd/yy'));
Insert into Reserve values (50014,40011,TO_DATE('09/18/22','mm/dd/yy'));
Insert into Reserve values (50015,40011,TO_DATE('10/10/22','mm/dd/yy'));
Insert into Reserve values (50011,40016,TO_DATE('12/20/22','mm/dd/yy'));
Insert into Reserve values (50011,40012,TO_DATE('09/27/22','mm/dd/yy'));
Insert into Reserve values (50011,40013,TO_DATE('10/19/22','mm/dd/yy'));
Insert into Reserve values (50011,40014,TO_DATE('05/26/22','mm/dd/yy'));
Insert into Reserve values (50011,40015,TO_DATE('10/10/22','mm/dd/yy'));
Insert into Reserve values (50019,60018,TO_DATE('01/19/23','mm/dd/yy'));
Insert into Reserve values (50016,60018,TO_DATE('12/10/22','mm/dd/yy'));
Insert into Reserve values (50018,60021,TO_DATE('12/21/22','mm/dd/yy'));

```

```

Insert into Return values
(50011,40011,TO_DATE('11/25/22','mm/dd/yy'),TO_DATE('11/25/22','mm/dd/yy'),0);
Insert into Return values
(50012,40011,TO_DATE('10/18/22','mm/dd/yy'),TO_DATE('10/28/22','mm/dd/yy'),1);
Insert into Return values
(50013,40011,TO_DATE('01/30/23','mm/dd/yy'),TO_DATE('01/30/23','mm/dd/yy'),0);

```

```

Insert into Return values
(50014,40011,TO_DATE('09/28/22','mm/dd/yy'),TO_DATE('10/01/22','mm/dd/yy'),1);
Insert into Return values
(50015,40011,TO_DATE('10/20/22','mm/dd/yy'),TO_DATE('10/20/22','mm/dd/yy'),0);
Insert into Return values
(50011,40016,TO_DATE('12/25/22','mm/dd/yy'),TO_DATE('12/25/22','mm/dd/yy'),0);
Insert into Return values
(50011,40012,TO_DATE('09/30/22','mm/dd/yy'),TO_DATE('09/30/22','mm/dd/yy'),0);
Insert into Return values
(50011,40013,TO_DATE('10/29/22','mm/dd/yy'),TO_DATE('10/29/22','mm/dd/yy'),0);
Insert into Return values
(50011,40014,TO_DATE('05/29/22','mm/dd/yy'),TO_DATE('05/29/22','mm/dd/yy'),0);
Insert into Return values
(50011,40015,TO_DATE('10/12/22','mm/dd/yy'),TO_DATE('10/12/22','mm/dd/yy'),0);
Insert into Return values (50018,60021,TO_DATE ('12/22/22', 'mm/dd/yy'),
TO_DATE('12/12/22','mm/dd/yy'),0);
Insert into Return values
(50018,60021,TO_DATE('12/22/22','mm/dd/yy'),TO_DATE('12/27/22','mm/dd/yy'),0);
Insert into Return values (50015,60015,TO_DATE('12/22/22','mm/dd/yy'),NULL,0);
Insert into Return values (50012,60015,TO_DATE('11/22/21','mm/dd/yy'),NULL,0);
Insert into Return values (50014,60015,TO_DATE('10/02/22','mm/dd/yy'),NULL,0);
Insert into Return values (50019,60015,TO_DATE('10/19/20','mm/dd/yy'),NULL,0);

```

We have created user profiles for generating local schemas for the same, also made the global schema into various fragments and ran the below queries as suggested from the requirements of project part 4.

Samples of user profiles from oracle user cloud :

The screenshot displays two user profiles in the Oracle User Cloud interface:

- NAVYA** (REST Enabled):
 - ORDS Alias: navya
 - >Password Expires in 359 days
 - Link: <https://ge467f9b7caf6ad-akhilboga18db1.adb.us-phoenix-1.oraclecloudapps.co...>
- VKP** (REST Enabled):
 - ORDS Alias: KRISHNA
 - Last Login: 4/25/2023, 2:18:59 AM
 - Password Expires in 359 days
 - Link: <https://ge467f9b7caf6ad-akhilboga18db1.adb.us-phoenix-1.oraclecloudapps.co...>

The screenshot shows the Oracle Database Actions SQL Worksheet interface. In the top-left, the Navigator pane shows 'GLOBAL_SCHEMA' and 'Tables'. The main area contains a SQL query:

```
1 SELECT * FROM ADMIN.LIBRARY
```

The results are displayed in a table:

	LIB_CODE	LIB_NAME	LIBRARY_STREET	LIBRARY_CITY	LIBRARY_STATE	LIBRARY_COUNTRY	LIBRARY_ZIP_CODE
1	20011	Sycamore	Sycamore St	San Antonio	Texas	US	78015
2	20012	Willis	W Prairie St	Denton	Texas	US	76201
3	20013	Discovery Park	N Elm St	Denton	Texas	US	76201
4	20014	Library Of Alexandria	Ptolemy St	Alexandria	Athens	Greece	11001
5	20015	Library of Congress	President St	Washington	DC	US	11007
6	20016	Bodleian	Tea St	Oxford	London	UK	34808

Queries solved from Local Schema:

1. List the name and address of library users with 4 or more book reservations.

```
select user_name,USER_STREET,    USER_CITY      ,USER_STATE,  USER_COUNTRY,
USER_ZIP_CODE
from users where user_id in (
select user_id from reserve group by user_id
having count(*)>4);
```

The screenshot shows the Oracle Database Actions SQL Worksheet interface. In the top-left, the Navigator pane shows 'NAVYA' and 'Tables'. The main area contains a SQL query:

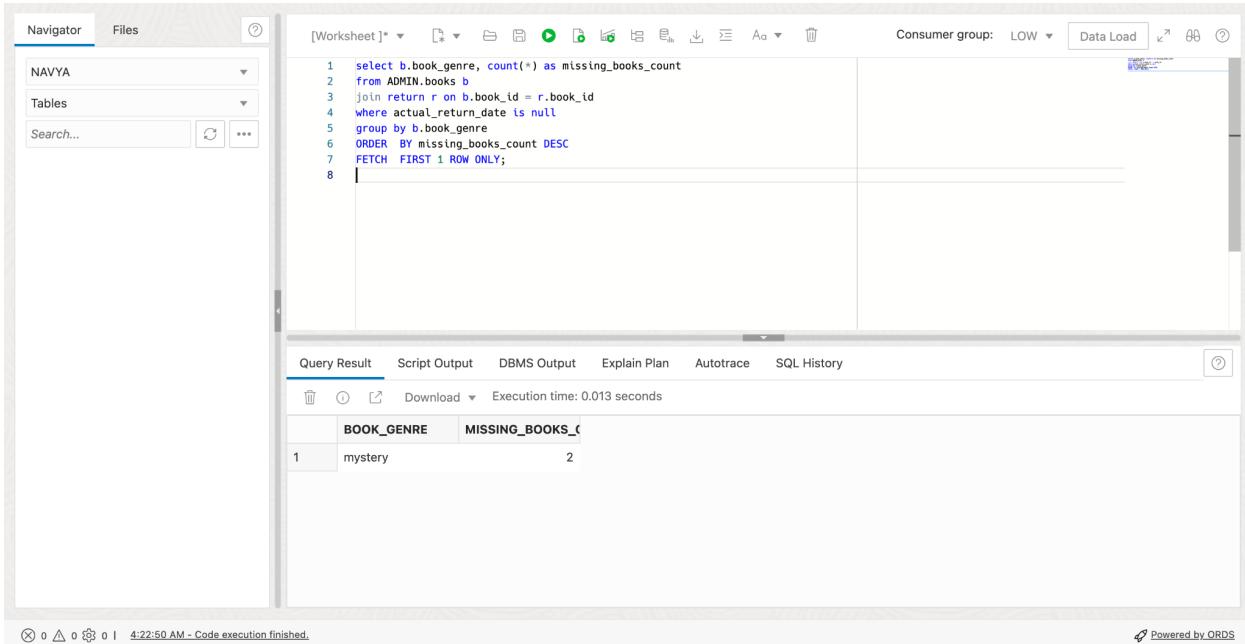
```
1 select user_name,USER_STREET,    USER_CITY      ,USER_STATE,  USER_COUNTRY,
2 USER_ZIP_CODE from users where user_id in (
3 select user_id from ADMIN.reserve group by user_id
4 having count(*)>4);
```

The results are displayed in a table:

	USER_NAME	USER_STREET	USER_CITY	USER_STATE	USER_COUNTRY	USER_ZIP_CODE
1	K. Christina	W. Oak	San Antonio	Texas	US	78015

2. List genre(classification) of book(s) with the most missing reports

```
select b.book_genre, count(*) as missing_books_count
from books b
join return r on b.book_id = r.book_id
where actual_return_date is null
group by b.book_genre
ORDER BY missing_books_count DESC
FETCH FIRST 1 ROW ONLY;
```



The screenshot shows a database worksheet interface. In the top-left, there's a 'Navigator' pane with 'NAVYA' selected. The main area contains a SQL script:

```
1 select b.book_genre, count(*) as missing_books_count
2 from books b
3 join return r on b.book_id = r.book_id
4 where actual_return_date is null
5 group by b.book_genre
6 ORDER BY missing_books_count DESC
7 FETCH FIRST 1 ROW ONLY;
8
```

Below the script, the 'Query Result' tab is active, showing the output:

	BOOK_GENRE	MISSING_BOOKS_C
1	mystery	2

Execution time: 0.013 seconds

At the bottom left, status icons show 0 errors, 0 warnings, and 0 informational messages. At the bottom right, it says 'Powered by ORDS'.

3. List genre(classification) of book(s) with the most reservations.

```
select count(*) as books_count,b.book_id,b.book_genre from books b
join reserve r on b.book_id = r.book_id
group by b.book_id,b.book_genre
ORDER BY books_count DESC
FETCH FIRST 1 ROW ONLY;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays 'VKP' and 'Tables'. The top-right shows 'Consumer group: LOW' and 'Data Load' buttons. The main area is a worksheet containing the following SQL code:

```

1 select count(*) as books_count,b.book_id,b.book_genre from books b
2 join reserve r on b.book_id = r.book_id
3 group by b.book_id,b.book_genre
4 ORDER BY books_count DESC
5 FETCH FIRST 1 ROW ONLY;

```

Below the worksheet is a 'Query Result' tab showing the execution time: 'Execution time: 0.014 seconds'. The result table has columns 'BOOKS_COUNT', 'BOOK_ID', and 'BOOK_GENRE', with one row: 6, 50011, literary.

At the bottom left, there are status icons: a red circle with a white exclamation mark, a green triangle, a blue square, and a yellow circle. The status bar at the bottom indicates: '4:32:25 AM - REST call resolved successfully.'

4. What is the library location with the least number of books reservations

```

select l.lib_name,count(*) as books_count from library l
join books b on l.lib_code = b.lib_code
join reserve r on r.book_id = b.book_id
group by lib_name
ORDER BY books_count Asc
FETCH FIRST 1 ROW ONLY;

```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays 'AKHILA' and 'Tables'. The top-right shows 'Consumer group: LOW' and 'Data Load' buttons. The main area is a worksheet containing the following SQL code:

```

1 select l.lib_name,count(*) as books_count from ADMIN.library l
2 join books b on l.lib_code = b.lib_code
3 join reserve r on r.book_id = b.book_id
4 group by lib_name
5 ORDER BY books_count Asc
6 FETCH FIRST 1 ROW ONLY;

```

Below the worksheet is a 'Query Result' tab showing the execution time: 'Execution time: 0.016 seconds'. The result table has columns 'LIB_NAME' and 'BOOKS_COUNT', with one row: Discovery Park, 1.

5. List library locations with employees making more than \$50,000 and having less than 3 book reservations.

```

select count(*),r.user_id,u.user_name,l.lib_name,l.LIBRARY_STREET,
l.LIBRARY_CITY, l.LIBRARY_STATE, l.LIBRARY_COUNTRY, l.LIBRARY_ZIP_CODE from staff s
join users u on s.emp_id = u.user_id
join reserve r on r.user_id = u.user_id
join books b on b.book_id = r.book_id
join library l on l.lib_code = b.lib_code
where s.salary > 50000
group by r.user_id,u.user_name,l.lib_name,l.LIBRARY_STREET, l.LIBRARY_CITY
,l.LIBRARY_STATE, l.LIBRARY_COUNTRY, l.LIBRARY_ZIP_CODE
having count(*) < 3;

```

The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL code with line numbers. The bottom pane shows the 'Query Result' tab with the following data:

COUNT(*)	USER_ID	USER_NAME	LIB_NAME	LIBRARY_STREET	LIBRARY_CITY	LIBRARY_STATE	LIBRARY_COUNTRY	
1	2	60018	Charles lokey	Boston Public	Oak Street	Boston	Massachusetts	US
2	1	60021	De Caprio	Vatican	Pope Ln	Square	Vatican City	Rome

6. What is the genre(classification) of books with the least number of reservations?

```

select count(*) as books_count,b.book_id,b.book_genre from books b
join reserve r on b.book_id = r.book_id
group by b.book_id,b.book_genre
ORDER BY books_count ASC
FETCH FIRST 1 ROW ONLY;

```

```

1 select count(*) as books_count,b.book_id,b.book_genre from books b
2 join reserve r on b.book_id = r.book_id
3 group by b.book_id,b.book_genre
4 ORDER BY books_count ASC
5 FETCH FIRST 1 ROW ONLY;

```

BOOKS_COUNT	BOOK_ID	BOOK_GENRE
1	50014	fiction

Curated 10 Queries:

- Find the name of the user who made highest number of reservations.

```

select u.user_id,u.user_name, count(*) as books_count
from books b
join return r on b.book_id = r.book_id
join users u on r.user_id = u.user_id
group by u.user_id,u.user_name
ORDER BY books_count DESC
FETCH FIRST 1 ROW ONLY;

```

```

1 select u.user_id,u.user_name, count(*) as books_count
2 from books b
3 join return r on b.book_id = r.book_id
4 join users u on r.user_id = u.user_id
5 group by u.user_id,u.user_name
6 ORDER BY books_count DESC
7 FETCH FIRST 1 ROW ONLY;
8

```

USER_ID	USER_NAME	BOOKS_COUNT
40011	K. Christina	5

- Who is the most experienced Employee, when did he/she join.

```

SELECT emp_name,MAX(sysdate-DATE_OF_EMPLOYMENT) as max_date
FROM staff

```

```

GROUP BY emp_name
order by max_date desc
fetch first 1 row only;

```

Navigator Files

[Worksheet] * Consumer group: LOW Data Load

```

1  SELECT emp_name,MAX(sysdate-DATE_OF_EMPLOYMENT) as max_date
2  FROM staff
3  GROUP BY emp_name
4  order by max_date desc
5  fetch first 1 row only;
6

```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History

EMP_NAME MAX_DATE

1	James Lovey	776.204212962963
---	-------------	------------------

3. List the names of cities where libraries are located
 select library_city from library;

Navigator Files

[Worksheet] * Consumer group: LOW Data Load

```

1  select library_city from library;

```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History

LIBRARY_CITY

1	San Antonio
2	Denton
3	Denton
4	Alexandria
5	Washington
6	Boston
7	New Haven
8	Square
9	Boston

4. List all books borrowed by a particular user 40011.
 select b.* from books b
 join reserve r on b.book_id = r.book_id
 join users u on u.user_id = r.user_id

where u.user_id = 40011;

The screenshot shows the Oracle SQL Developer interface. In the top right, the consumer group is set to 'LOW' and the data load status is shown. The central area is a worksheet containing the following SQL code:

```
1 select b.* from books b
2 join reserve r on b.book_id = r.book_id
3 join users u on u.user_id = r.user_id
4 where u.user_id = 40011;
5
```

Below the worksheet is a 'Query Result' tab showing the execution time (0.028 seconds). The results are displayed in a table:

BOOK_ID	VEN_CODE	PUB_CODE	AUTH_ID	LIB_CODE	BOOK_NAME	BOOK_GENRE	PRICE
1	50011	70011	80011	90011	20011	War and Peace, first	literary
2	50012	70012	80012	90012	20012	Hells Angels	mystery
3	50013	70013	80013	90013	20013	Kitchen Confidential	education
4	50014	70014	80014	90014	20014	Dune 4th	fiction
5	50015	70015	80015	90015	20015	Programming for Dummies	education

5. List all overdue books borrowed by a particular user.

select * from return where actual_return_date is null;

The screenshot shows the Oracle SQL Developer interface. In the top right, the consumer group is set to 'LOW' and the data load status is shown. The central area is a worksheet containing the following SQL code:

```
1 select * from return where actual_return_date is null;
2
```

Below the worksheet is a 'Query Result' tab showing the execution time (0.012 seconds). The results are displayed in a table:

BOOK_ID	USER_ID	EXPECTED_RETURN_DATE	ACTUAL_RETURN_DATE	LATE
1	50015	60015 12/22/2022, 12:00:00 (null)		0
2	50012	60015 11/22/2021, 12:00:00 (null)		0
3	50014	60015 10/2/2022, 12:00:00 (null)		0
4	50019	60015 10/19/2020, 12:00:00 (null)		0

6. List all books that have been borrowed more than 3 times.

select count(*),b.book_id,b.book_name from reserve r
join books b on r.book_id = b.book_id
group by b.book_id,b.book_name
having count(*)>3;

The screenshot shows the Oracle SQL Developer interface. The Navigator pane on the left lists 'NAVYA' and 'Tables'. The Worksheet pane contains the following SQL code:

```

1 select count(*),b.book_id,b.book_name from reserve r
2 join books b on r.book_id = b.book_id
3 group by b.book_id,b.book_name
4 having count(*)>3;
5

```

The Query Result pane shows the output:

COUNT(*)	BOOK_ID	BOOK_NAME
1	6	50011 War and Peace, first

7. List all users who have borrowed a book more than once.

```

select count(*),b.book_id,b.book_name,u.user_name from reserve r
join books b on r.book_id = b.book_id
join users u on u.user_id = r.user_id
group by b.book_id,b.book_name,u.user_name
having count(*)>1;

```

The screenshot shows the Oracle SQL Developer interface. The Navigator pane on the left lists 'NAVYA' and 'Tables'. The Worksheet pane contains the following SQL code:

```

1 select count(*),b.book_id,b.book_name,u.user_name from reserve r
2 join books b on r.book_id = b.book_id
3 join users u on u.user_id = r.user_id
4 group by b.book_id,b.book_name,u.user_name
5 having count(*)>1
6

```

The Query Result pane shows the output:

COUNT(*)	BOOK_ID	BOOK_NAME	USER_NAME
No rows selected			

8. List all users who have not borrowed any books.

```
select * from users where user_id not in (select distinct user_id from reserve);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, there's a Navigator pane with 'NAVYA' selected. The main area is a 'Worksheet' tab with the following SQL code:

```
1 select * from users where user_id not in (select distinct user_id from reserve)
```

The results are displayed in a table titled 'Query Result' with the following data:

	USER_ID	USER_NAME	USER_CONTACT	USER_STREET	USER_CITY	USER_STATE	USER_COUNTRY	USER_Z
1	40017	Bruce	176757979	Bleeker St	Gotham City	Michigan	US	76207
2	40018	Jane Doe	186757979	123 Main St	Anytown	Massachusetts	US	02360
3	40019	Henry	196757979	A. Oak	Houston	Texas	US	76207
4	40020	Feng	206757979	JFK st	Queens	New York	US	76207
5	60021	De Caprio	1202344545	Avenue A	California	California	US	79015

9. List the names of all authors who have written books in more than one genre.

```
select count(*),a.AUTH_NAME from books b
join author a on b.AUTH_ID = a.AUTH_ID
group by a.AUTH_NAME;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, there's a Navigator pane with 'NAVYA' selected. The main area is a 'Worksheet' tab with the following SQL code:

```
1 select count(*),a.AUTH_NAME from books b
2 join author a on b.AUTH_ID = a.AUTH_ID
3 group by a.AUTH_NAME
4
```

The results are displayed in a table titled 'Query Result' with the following data:

	COUNT(*)	AUTH_NAME
1	1	William Shakesphere
2	1	George Orwell
3	1	Tolkein
4	1	Frank Herbert
5	1	Hunter Thompson
6	1	Leo Tolstoy
7	1	Bruce Wayne
8	1	Burt Macklin
9	1	Tarantino

10. List the names and contacts of all library users who have reserved books by Leo Tolstoy.

```
select u.user_name,u.USER_CONTACT from users u
join reserve r on u.user_id = r.user_id
join books b on r.book_id = b.book_id
join author a on a.auth_id = b.auth_id
where a.auth_name = 'Leo Tolstoy'
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays 'NAVYA' and 'Tables'. The top-right shows consumer group 'LOW' and data load status. The central area is a worksheet containing the following SQL code:

```

1 select u.user_name,u.USER_CONTACT from users u
2 join reserve r on u.user_id = r.user_id
3 join books b on r.book_id = b.book_id
4 join author a on a.auth_id = b.auth_id
5 where a.auth_name = 'Leo Tolstoy'
6
7

```

Below the worksheet is a toolbar with tabs: Query Result, Script Output, DBMS Output, Explain Plan, Autotrace, and SQL History. The 'Query Result' tab is selected, showing the execution time: 0.012 seconds. The results are displayed in a table:

	USER_NAME	USER_CONTACT
1	K. Christina	3426757979
2	Kirk	7896757979
3	Mary	136757979
4	Rosy	146757979
5	Chris	156757979
6	Tracy	166757979

Update statements:

1. Update the title of the book with Book_id 50030 to 'Chef Skills by Cook Books'.

```

UPDATE Books
SET Book_name='Chef Skills by Cook Books'
WHERE Book_ID=50030;

```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays 'AKHILA' and 'Tables'. The top-right shows consumer group 'LOW' and data load status. The central area is a worksheet containing the following SQL code:

```

1 UPDATE Books
2 SET Book_name='Chef Skills by Cook Books'
3 WHERE Book_ID=50030;
4

```

Below the worksheet is a toolbar with tabs: Query Result, Script Output, DBMS Output, Explain Plan, Autotrace, and SQL History. The 'Script Output' tab is selected. The results show several error messages indicating unique constraint violations:

- ORA-00001: unique constraint (ADMIN.SYS_C0021961) violated
Error at Line: 24 Column: 0
- ORA-00001: unique constraint (ADMIN.SYS_C0021961) violated
Error at Line: 25 Column: 0
- ORA-00001: unique constraint (ADMIN.SYS_C0021961) violated
Error at Line: 26 Column: 0
- ORA-00001: unique constraint (ADMIN.SYS_C0021961) violated
Error at Line: 27 Column: 0

At the bottom of the script output, it says '1 row updated.' and 'Elapsed: 00:00:00.006'. The bottom right corner of the interface shows the full command: 'UPDATE Books SET Book_name='Chef Skills by Cook Books' WHERE Book_ID=50030'.

2. Update the expected return date for book reservations with ID '50018' to '12/29/22'.

```
UPDATE Return  
SET Expected_Return_date=TO_DATE('12/29/22','mm/dd/yy')  
WHERE Book_ID=50018;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, the Navigator shows the schema 'AKHILA'. The main workspace contains the following SQL code:

```
1 UPDATE Return  
2 SET Expected_Return_date=TO_DATE('12/29/22','mm/dd/yy')  
3 WHERE Book_ID=50018;  
4
```

Below the code, the 'Query Result' tab is selected, displaying the output:

```
1 row updated.  
Elapsed: 00:00:00.010
```

3. Update the name of the user with ID '40018' to 'Jane Doe'.

```
UPDATE Users  
SET User_name='Jane Doe'  
WHERE User_ID=40018;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, the Navigator shows the schema 'AKHILA'. The main workspace contains the following SQL code:

```
1 UPDATE Users  
2 SET User_name='Jane Doe'  
3 WHERE User_ID=40018;  
4
```

Below the code, the 'Query Result' tab is selected, displaying the output:

```
1 row updated.  
Elapsed: 00:00:00.010
```



```
1 row updated.  
Elapsed: 00:00:00.009
```

4. Update the address of the user with ID '40018' to '123 Main St, Anytown, Massachusetts, US, 02360'.

```
UPDATE Users  
SET User_street='123 Main St',  
User_city='Anytown',  
User_State='Massachusetts',  
User_Country='US',  
User_Zip_code='02360'  
WHERE User_ID=40018;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays 'AKHILA' and 'Tables'. The main area is a 'Worksheet' containing the following SQL code:

```
1 UPDATE Users
2 SET User_street='123 Main St',
3 User_city='Anytown',
4 User_State='Massachusetts',
5 User_Country='US',
6 User_Zip_code='02360'
7 WHERE User_ID=40018;
```

The bottom panel shows the 'Query Result' tab with the output:

```
1 row updated.
Elapsed: 00:00:00.010
```

5. Update the number of copies available for the book with ID '50030' to 33.

```
UPDATE Books
SET No_of_books=33
WHERE Book_ID=50030;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays 'AKHILA' and 'Tables'. The main area is a 'Worksheet' containing the following SQL code:

```
1 UPDATE Books
2 SET No_of_books=33
3 WHERE Book_ID=50030;
4 |
```

The bottom panel shows the 'Query Result' tab with the output:

```
1 row updated.
Elapsed: 00:00:00.010
```

6. Update the reserved date for all book reservations made by the user with ID '60021' to '12/19/22'.

```
UPDATE Reserve
SET Reserve_date=TO_DATE('12/19/22','mm/dd/yy')
WHERE User_ID=60021;
```

The screenshot shows the Oracle SQL Developer interface. In the top worksheet, an UPDATE statement is being run:

```
1 UPDATE Reserve
2 SET Reserve_date=TO_DATE('12/19/22','mm/dd/yy')
3 WHERE User_ID=60021;
4
```

In the bottom Query Result tab, the output shows:

```
1 row updated.
Elapsed: 00:00:00.010
```

Delete Statements:

1. Delete book with Book_id 50030.

```
DELETE
FROM
Books
WHERE
Book_ID=50030;
```

The screenshot shows the Oracle SQL Developer interface. In the top worksheet, a DELETE statement is being run:

```
1 DELETE
2 FROM
3 Books
4 WHERE
5 Book_ID=50030;
6
```

In the bottom Query Result tab, the output shows:

```
1 row deleted.
Elapsed: 00:00:00.014
```

2. Delete book reservations with ID book id '50018'.

```
DELETE
FROM
Reserve
WHERE
Book_ID=50018;
```

The screenshot shows the Oracle SQL Developer interface. In the top workspace, a script is being run:

```
1 DELETE
2   FROM ...
3     Reserve
4 WHERE ...
5   Book_ID=56018;
```

In the bottom workspace, the results of the query are displayed under the "Query Result" tab:

1 row deleted.
Elapsed: 00:00:00.014

3. Delete user with ID '60022'.

```
DELETE
FROM
  Users
WHERE
  User_ID=60022;
```

The screenshot shows the Oracle SQL Developer interface. In the top workspace, a script is being run:

```
1 DELETE
2   FROM ...
3     Users
4 WHERE ...
5   User_ID=60022;
```

In the bottom workspace, the results of the query are displayed under the "Query Result" tab:

1 row deleted.
Elapsed: 00:00:00.014

4. Delete entry in the books table where the number of copies reserved for the book is 33.

```
DELETE
FROM
  Books
WHERE
  No_of_books=33 and Status='reserved';
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays a connection named 'NAVYA' and a 'Tables' section. The main workspace contains a worksheet with the following SQL code:

```
1 DELETE
2 FROM
3 Books
4 WHERE
5 No_of_books=33 and Status='reserved';
6
```

Below the worksheet, the 'Script Output' tab is selected in the results panel, which shows the output: '1 row deleted.' and 'Elapsed: 00:00:00.006'.

5. There was some confusion about the return dates for Book id 50018, hence the entry.

```
DELETE
FROM
    Return
WHERE
    Book_ID=50018;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, the Navigator pane displays a connection named 'AKHILA' and a 'Tables' section. The main workspace contains a worksheet with the following SQL code:

```
1 DELETE
2 FROM
3 Return
4 WHERE
5 Book_ID=50018;
6
```

Below the worksheet, the 'Script Output' tab is selected in the results panel, which shows the output: '1 row deleted.' and 'Elapsed: 00:00:00.010'.

6. Accidentally reserved a book with Id 50016, and hence admin asked to remove the entry to reduce unnecessary entries in the database.

```
DELETE
FROM
    Reserve
WHERE
    Book_ID=50016;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, the Navigator pane displays a schema named 'AKHILA' with a 'Tables' section. The main workspace contains a worksheet with the following SQL code:

```

1  DELETE
2  FROM
3    Reserve
4  WHERE
5    Book_ID=50016;
6

```

Below the worksheet, the 'Script Output' tab is selected, showing the results of the query execution:

```

1 row deleted.
Elapsed: 00:00:00.010

```

Individual Contribution:

Navyatha Yamsani: I'm responsible for updating and inserting the data into the existing database according to the queries which need to be executed. Involved in developing fragmentation using Oracle Cloud. Then as per the queries given, I have participated in writing the statements which need to be queried to retrieve information from a local schema. I have also been involved in writing queries for Update and Delete statements according to the updated database.

Akhila Battula: I have participated in updating the existing database and writing the queries for the statements which were given and also written additional queries which involve Update, Delete statements. And then along with that, worked on existing databases to match the assumptions made for new queries. Involved in developing fragmentation using Oracle Cloud. Written Update statements and queried using Oracle SQL code.

Akhil Boga: I have written the queries for fragmenting the database. After which, I have written code for the 1st , 2nd and 3rd queries asked in the assignment. First, I have returned the list of names and addresses of library users with four or more book reservations using vertical fragmentation. Next, I've returned the list of genre books with the most missing reports. This was done by cross validating the Reserve, Return, Books and the Maintainstrackof tables.. Finally, I have written the query for returning the genre of books with the most reservations. This was done by dropping and fragmenting the tables Books, Reserves, Maintainstrackof tables.

Krishna Praveen Velagapalli: I am responsible for creating the global schema which is further fragmented into the local schemas when accessed by the other groupmates. I did this by loading my database into my oracle cloud account which will act as the global schema to which the other users will connect and act as a local schema.

Rakshith Dyavari Shetty: I have worked the queries for the 4th,5th and 6th parts asked in the assignment. Initially, I have written the query to return the location of the library with the least number of book reservations. This was done by accessing the library table from gcs and dropping all the attributes other than the location and book id. Then , I vertically fragmented the reserves and the library table. For my second query, I have returned the list of locations with employees with more than 50K in salaries and less than 3 book reservations. This was done by fragmenting the tables Employees,Books, reserves wherein the unnecessary columns were dropped. Finally, I returned the genre of books with the least number of reservations.